

VIRUS CONTROL

COLLABORATORS

	<i>TITLE :</i> VIRUS CONTROL		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 9, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	VIRUS CONTROL	1
1.1	VIRUS CONTROL 5.0	1
1.2	bedienungsanleitung	1
1.3	hauptmerkmale	3
1.4	schnelleinstieg	4
1.5	installation	6
1.6	selbsttest	7
1.7	arbeitsfenster öffnen	8
1.8	schließsymbol	9
1.9	eigener schirm	9
1.10	reset	10
1.11	kill	10
1.12	df0df1df2df3	10
1.13	zeige dfx	11
1.14	installiere dfx	11
1.15	bootblock->datei	12
1.16	bootblock->puffer	12
1.17	puffer ->bootblock	12
1.18	datei ->bootblock	12
1.19	f.prüfsum->noboot	13
1.20	r.prüfsum->boot	13
1.21	personal bootblock	13
1.22	vergleichen	14
1.23	zeige datei	15
1.24	dateiveränderungen	15
1.25	Linkviren	17
1.26	fileviren, mailboxviren, trojanische pferde	18
1.27	Linkviren	18
1.28	Disk-Validatorviren	19
1.29	automatisches entpacken von programmen	20

1.30	automatisches entpacken von dateiarchiven	21
1.31	unsichtbare zeichen in filenames anzeigen	23
1.32	beschädigte .info-files anzeigen	23
1.33	beschädigte programme anzeigen	23
1.34	alle dateien prüfen	23
1.35	automatisch	24
1.36	dateiauswahlfenster wählen und vorbelegen	24
1.37	unterverzeichnisse	24
1.38	protokoll	24
1.39	filenamenpuffer	24
1.40	statistik	25
1.41	linkvirus von file abtrennen	25
1.42	fensterleiste	25
1.43	diskeinlegen->schnelltest	26
1.44	diskeinlegen->kompletttest	26
1.45	boot-menü	27
1.46	disk-boot verhindern	28
1.47	beenden	28
1.48	laufwerke ändern	29
1.49	speichermedien	30
1.50	bootblock-archivierung	32
1.51	bootblock-analyse	36
1.52	disassembler, analyser	36
1.53	systemtest	38
1.54	virusentfern-dialogfenster	40
1.55	systemübersicht	41
1.56	lern-modus	45
1.57	speicheranzeige	45
1.58	benutzung von betriebssystemfunktionen kontrollieren	46
1.59	schreibzugriff auf device melden	47
1.60	startup-sequence-kontrolle	47
1.61	bootblock-schreibzugriffkontrolle	48
1.62	zero-location-bug	48
1.63	move sr,<ea> handler	48
1.64	uhr aktivieren	49
1.65	palntsc	49
1.66	chip-speicher bevorzugen	51
1.67	fast-speicher-zugriff erlauben	51
1.68	arbeitsfensterfarben	52

1.69 farbsignale	52
1.70 warnton	52
1.71 tastaturbelegung	53
1.72 einstellungen speichern	54
1.73 befehlsdateimodus	55
1.74 commodity	55
1.75 appicon, drag and drop	56
1.76 multiselect	56
1.77 amigaguide-hilfe	56
1.78 zukünftige viren	56
1.79 warnung	57
1.80 allgemeine einführung in die virenproblematik	57
1.81 geschichte	58
1.82 verbreitung	58
1.83 bootblockviren	59
1.84 beseitigung	60
1.85 rigiddiskblock beschädigen	60
1.86 fileviren	61
1.87 disk-validatorviren	61
1.88 linkviren	62
1.89 Beseitigung der Fileviren, Disk-Validatorviren und Linkviren	62
1.90 virusanzeichen	63
1.91 packerproblematik	63
1.92 uhrvirus	63
1.93 lauffähigkeit von viren	64
1.94 festplatte	64
1.95 speicherausbau	64
1.96 höhere prozessoren	65
1.97 feste kickstart-adressen	65
1.98 bootdisk-bug	66
1.99 drivebit-bug	66
1.100schlußfolgerung	67
1.101zukunftsansichten	67
1.102autoradresse	68
1.103konkrete virenbeschreibungen	68
1.104Bootblockviren	69
1.10516bit crew	72
1.1066uldv8	73
1.107a.h.c.	73

1.108aek	74
1.109aids	74
1.110aids-hiv	74
1.111alien new beat	74
1.112amigafreak	74
1.113amiga-master	75
1.114angel	75
1.115austral.parasite	76
1.116bamigasectorone	76
1.117blackflash	76
1.118black-knight	77
1.119bladerunners	77
1.120blf	77
1.121blowjob	77
1.122butonic's 1.1	78
1.123bytebandit	79
1.124bytebandit 1	79
1.125bytebandit 2	80
1.126bytebanditimitation	80
1.127bytebanditvips	80
1.128bytevoyager i	80
1.129bytevoyager ii	81
1.130cccp	81
1.131chaos-taipan	81
1.132claas abraham	82
1.133clist-lamer(uk-lamerstyle)	82
1.134clonk	82
1.135cobra	82
1.136coder	83
1.137copylock	83
1.138cracker-extermin.	84
1.139diskdoks	84
1.140creeping-eel	85
1.141dafgderfehy	85
1.142dasa-bytewarrior	85
1.143data crime	85
1.144dat-89	86
1.145datalock	86
1.146derk-mallander	87

1.147destructor	87
1.148detlef	87
1.149digitalemotions	88
1.150digital dream	88
1.151diskherpes	88
1.152divina extermin.	89
1.153dotty	90
1.154dumdum	90
1.155dum2dum	90
1.156electro-vision	91
1.157eleni!	91
1.158eleni-clock	92
1.159excrement	92
1.160exterminator ii	93
1.161extreme	93
1.162fast	93
1.163fast 1	94
1.164fasteddie	94
1.165fasteddie-infector	94
1.166f.i.c.a	94
1.167forpib	95
1.168frenchkiss	95
1.169freshmaker	95
1.170frity	96
1.171fuck.device	96
1.172fuck-lamer(ingo`s return)	96
1.173future-disaster	96
1.174gadaffi	97
1.175gandalf	97
1.176genestealer	98
1.177glasnost	98
1.178graffiti	99
1.179gremlin	99
1.180guardiansbootaids	99
1.181gx.team	100
1.182gyros	100
1.183hcs	100
1.184heil	101
1.185hilly	101

1.186hoden v33.17	101
1.187hulksters	102
1.188ice	102
1.189incognito	102
1.190inger.iq.virus	102
1.191jinx	103
1.192jitr	103
1.193joshua	103
1.194joshua 1	103
1.195julietick	104
1.196kako	104
1.197kauki	105
1.198killed	105
1.199l.a.d.s	105
1.200l.a.d.s - a.i.d.s	106
1.201lameblame-taipan(lameblame,cheater-hijacker,polish)	106
1.202lamer-bootblockviren	107
1.203laureline v1.0	108
1.204leviathan	109
1.205little sven	109
1.206loverboy&sexmachine	109
1.207lsd	109
1.208mad	109
1.209mad ii	110
1.210mad iii	110
1.211mad iv	110
1.212megamaster	110
1.213metamorphosis1.0	110
1.214mexx	111
1.215mg's virus v1.0	111
1.216micromaster	111
1.217microsystems	111
1.218morbid.angel.virus	111
1.219mosh	112
1.220mosh 2	112
1.221mount-eleni	112
1.222mutilator	113
1.223nasty	114
1.224no.bandit.any.more	114

1.225obelisk	114
1.226obelisk ii	114
1.227opapa	114
1.228overkill	115
1.229paradox i	116
1.230paradox ii	117
1.231paramount	117
1.232paratax i	117
1.233paratax ii	118
1.234paratax iii	118
1.235pentagonlayer	118
1.236perverse i	118
1.237powerbomb	119
1.238powerteam	119
1.239pvl	119
1.240revenge bootloader	120
1.241revenge	120
1.242ripper	120
1.243riska	121
1.244sachsen no.1	121
1.245sachsen no.3	121
1.246saddamhussein	122
1.247sao paulo	122
1.248satan	122
1.249sca	123
1.250sca-2001	124
1.251sca-aids	124
1.252sca-dag	124
1.253sca-kefrens	124
1.254sca-max	124
1.255scarface	125
1.256sendarian	125
1.257sentinel-ussr492	125
1.258shit	126
1.259sonja	126
1.260ss	127
1.261starcom	127
1.262starfire-north	128
1.263starfire-eaststar	128

1.264suicide	128
1.265superboy	129
1.266systemz	129
1.267systemz 6.1,6.3,6.4,6.5	129
1.268tai	130
1.269target	130
1.270telstar(systemz-v6.0)	130
1.271termigator	131
1.272t.f.c. revenge virus	131
1.273time-bomb-v1.0	131
1.274tomatesgentechnicservice	132
1.275traveller1.0	132
1.276triplex	132
1.277trisector 911	132
1.278turk virus 1.3	133
1.279twinz santa claus	133
1.280uhr	133
1.281ultra-fox	134
1.282umyj dupe	134
1.283vccoftnt	134
1.284vermin	135
1.285virusconstr.i	135
1.286virusconstr.ii	135
1.287virus fighter v1.0	136
1.288virusv1	136
1.289vkill 1.0	136
1.290waft	137
1.291wahnfried	137
1.292warhawk	138
1.293warsaw avenger	138
1.294zaccess v1.0	138
1.295zaccess v2.0	139
1.296zaccess v3.0	139
1.297z.e.s.t	139
1.298zenker	139
1.299zombi i	140
1.300Fileviren	141
1.301amigaknights	144
1.302anti-euromail-virus	145

1.303bgs9 i+ii+iii	146
1.304bluebox-icon.library	147
1.305bret-hawnes	147
1.306butonic-jeff	148
1.307compuphagozyte4	151
1.308compuphagozyte5	151
1.309compuphagozyte6	152
1.310compuphagozyte8	152
1.311compuphagozyte3	153
1.312compuphagozyte7	154
1.313darth vader v1.1	154
1.314disaster-master v2	154
1.315leviathan-bootblock+filevirus	156
1.316liberator1.21-memcheck	157
1.317liberator3.0-cv	157
1.318liberator5.01-pv	158
1.319lupo	159
1.320nano1	159
1.321nano2	160
1.322nast	160
1.323novi	160
1.324purge	161
1.325revengeofthelamerexterm.	161
1.326scsi	162
1.327sepultura	163
1.328sepultura (v2.26)	163
1.329telecom	164
1.330terrorists	164
1.331challenger-fish622	165
1.332clonk-installer	166
1.333colorsviruscarrierurkbb	166
1.334crime!++-trojan.pferd	166
1.335excrement-installer	167
1.336generalhunter v3.2	167
1.337intro-maker v1.00 by tcr	167
1.338jeff3.10-trojan.pferd	168
1.339lads-mvk	168
1.340lamer-bomb(gotcha lamer)	168
1.341lamerbb-trojan.pferd	169

1.342messangel	169
1.343modemcheck-fuck-virus	169
1.344scan.x	171
1.345t.f.c.-loadwb	172
1.346the smily cancer ii	172
1.347virus terminatorv6.0	172
1.348virus-install v2.0	173
1.349little sven-trojan.pferd	173
1.350??? \$4eb9-virus ???	173
1.351??? \$4eb9-4ef9-virus ???	175
1.352??? hunklab-virus ???	175
1.353??? xlink-virus ???	176
1.354bootblock-massacre	176
1.355bootshop	176
1.356dag-virus-infector	176
1.357virusmaker v1.0	177
1.358virusconstructionseti	177
1.359virusconstructionsetii	178
1.360aaa-enhancer	178
1.361a.i.s.f. interlamer	179
1.362aibon	180
1.363aibon2	180
1.364bootx-updater	181
1.365byteparasitei	181
1.366byteparasiteii	182
1.367byteparasiteiii	182
1.368chaos-master v0.5	183
1.369commodore-virus	183
1.370compuphagozyte1	184
1.371compuphagozyte2	185
1.372conman-trojan	185
1.373d&a	186
1.374decompiler	186
1.375degrad	186
1.376descriptor v3.0	187
1.377disk-killer v1.0	188
1.378diskspeedcheckv1.01ß	188
1.379disktroyer	189
1.380d-structure(a,b,c)	189

1.381	elien	190
1.382	excreminator v1.0	190
1.383	freedom	191
1.384	timebomb v0.9	191
1.385	timebomber(virustest)	192
1.386	virusblaster	193
1.387	unnamed.1	193
1.388	vmk v3.00	194
1.389	ae-registrator	194
1.390	amipatch v1.0a	195
1.391	dm-trash	195
1.392	doom	195
1.393	door_bells	195
1.394	dopusrt	196
1.395	easy-e	196
1.396	lhacheck 1.1	196
1.397	look-bbs	196
1.398	m_chat v2.3	196
1.399	modemspeederv2.1	197
1.400	mongo	197
1.401	noguruv2.0	197
1.402	showsysops	198
1.403	swiftware-devildoor8	198
1.404	sysinfov2.2	198
1.405	timer	198
1.406	top util v1.0	199
1.407	trojan killer v3.0	199
1.408	xpr-speederv3.2	200
1.409	Disk-Validatorviren	201
1.410	orange-disk-validatorvirus(=diskval1234)	202
1.411	returnofthelamerexterminator-disk-validatorvirus	202
1.412	saddam-hussein-disk-validatorvirus und abkömmlinge	203
1.413	Linkviren	204
1.414	antichrist	205
1.415	burn	205
1.416	christmas violator	206
1.417	cccp-bootblock+linkvirus	207
1.418	hochofen(=trabbi)	207
1.419	irq-linkvirus i+ii	207

1.420megalink	208
1.421menem's revenge	208
1.422metamorphosisv1.0-bootblock+linkvirus	209
1.423qrdl	209
1.424smilycancercenturions	210
1.425thetravelingjack	212
1.426viewtek	213
1.427bestial devastation	213
1.428new age	214
1.429xeno-virus i+ii	214
1.430crime!	215
1.431crime!++	215
1.432crime'92	216
1.433fileghost	217
1.434goldenrider	217
1.435lz-linkvirus	218
1.436commander	220
1.437darkavenger	220
1.438infiltrator	221
1.439polyzygotronifikator	221
1.440red october v1.7	222
1.441debugger	223
1.442lazarus	223
1.443myindex	224

Chapter 1

VIRUS CONTROL

1.1 VIRUS CONTROL 5.0

```
*****
*                               VIRUS CONTROL 5.0                               *
*   © Pius Nippgen, Bergstr.12, 66453 Gersheim, Germany   *
*               letzte Version, eine Legende stirbt ...               *
*****
```

Bedienungsanleitung

Allgemeine Einführung in die Virenproblematik

Konkrete Virenbeschreibungen

1.2 bedienungsanleitung

Bedienungsanleitung zu VIRUS CONTROL 5.0

```
-----
Hauptmerkmale
Schnelleinstieg
Installation
Selbsttest
Arbeitsfenster öffnen
Schließsymbol
Schirm wählen
Reset
Kill
DF0: DF1: DF2: DF3:
Zeige DFX:
Installiere DFX:
Bootblock->Datei
Bootblock->Puffer
Puffer    ->Bootblock
Datei     ->Bootblock
f.Prüfsum->NoBoot
r.Prüfsum->Boot
Personal Bootblock
```

Vergleichen
Zeige Datei
Dateiveränderungen erkennen
File/Linkviren suchen und entfernen
Fileviren, Mailboxviren, Trojanische Pferde
Linkviren
Disk-Validatorviren, (dekodieren, umbenennen)
automatisches Entpacken von Programmen
automatisches Entpacken von Dateiarhiven
unsichtbare Zeichen in Filenamen anzeigen
beschädigte .info-files anzeigen
beschädigte Programme anzeigen
alle Dateien prüfen
automatisch
Dateiauswahlfenster wählen und vorbelegen
Unterverzeichnisse
Protokoll
Filenamespuffer
Statistik
Linkvirus von File abtrennen
Fensterleiste
Diskeinlegen->Schnelltest
Diskeinlegen->Komplettest
Boot-Menü
Disk-Boot verhindern
beenden
Laufwerke ändern
Speichermedien, RigidDiskBlock-Verwaltung
Bootblock-Archivierung
Bootblock-Analyse
Disassembler, Analyser
Systemveränderungen kontrollieren
Virusentfern-Dialogfenster
Systemübersicht
Lern-Modus
Speicheranzeige
Benutzung von Betriebssystemfunktionen kontrollieren
Schreibzugriff auf Device melden
Startup-Sequence-Kontrolle
Bootblock-Schreibzugriffkontrolle
Zero-Location-Bug
move sr, <ea> Handler
Uhr aktivieren
PAL/NTSC
Chip-Speicher bevorzugen
Fast-Speicher-Zugriff erlauben
Arbeitsfensterfarben
Farbsignale
Warnton
Tastaturbelegung
Einstellungen speichern
Befehlsdateimodus
commodity
AppIcon, Drag and Drop
Multiselect
AmigaGuide-Hilfe
zukünftige Viren

Warnung
Autoradresse

1.3 hauptmerkmale

Hauptmerkmale von VIRUS CONTROL

erkennt und entfernt auf bequemste Art alle Bootblockviren ,
Fileviren , Linkviren , Disk-Validatorviren , Mailboxviren und
Trojanischen Pferde .

läuft auf allen Amigas und unter allen Bedingungen, wobei ab
Kickstart 2.0 automatisch die neuen Möglichkeiten wie z.B. AppIcon ,
Commodity , Schirmauswahl , Notify usw. ausgenutzt werden.

bietet Ihnen optimalen Systemschutz , denn aufgrund der Resetfestigkeit
und der Unabhängigkeit von externen libraries, kann VIRUS CONTROL jederzeit,
so auch bereits vor dem Booten, effektiv arbeiten. VIRUS CONTROL ist also
völlig unkompliziert einzusetzen, einfach starten und VIRUS CONTROL bleibt
bis zum nächsten Ausschalten aktiv, egal was zwischenzeitlich passiert,
egal von welcher Diskette oder Festplatte im Nachhinein gebootet wird.

besitzt einen Lern-Modus , mit dem Sie einfach per Mausklick VIRUS CONTROL
dauerhaft davon in Kenntnis setzen können, daß gewisse System-Veränderungen
harmlos sind, alle anderen System-Veränderungen werden aber weiterhin
sofort erkannt, da VIRUS CONTROL z.B. sekundlich Ihr System sehr wirksam
auf Virus-Befall überprüft.

kann auch in Dateiarchiven (z.B. lha,arc usw.) oder Diskettenarchiven
(z.B. dms,zoom usw.) automatisch Viren aufspüren und ist somit auch ideal für
z.B. Dafütreibende. Darüber hinaus werden natürlich auch Viren in gepackten
startbaren Programmen gefunden.

kann alle virustypischen Betriebssystemfunktionsaufrufe kontrollieren ,
wodurch sehr einfach das womöglich schädliche Verhalten neuer Programme
erkannt werden kann.

kann Speicherbereiche und Bootblöcke disassemblieren und analysieren ,
wodurch konkrete Aussagen über die Arbeitsweise und Gefährlichkeit der
Daten möglich werden.

wartet mit sehr umfassenden Vireninformationen auf.
Sowohl Anfänger wie Profis werden Nutzen aus den zum Teil
einzigartigen Informationen ziehen. Die im AmigaGuide-Format
vorliegenden Vireninformationen werden bei einem gefundenen Virus
auch automatisch angezeigt.

weist noch viele weitere interessante Funktionen auf,
wie z.B. Installerbenutzung , komfortable Rigiddiskblockverwaltung ,
Dateiprüfsummenfunktionen , Bootblockarchivierung , eine mächtige
Vergleichsfunktion mit der sehr bequem Dateien mit Dateien oder
Dateien mit Disketten oder Disketten mit Disketten verglichen werden
können, wobei auch eine Offsetangabe möglich ist.
Eine sehr komfortable und durchdachte Bedienungsoberfläche,
beliebig viele Bootblock-, Speicheranzeige-, Vergleichen- und

Disassemblierfenster gleichzeitig bedienbar, auf Wunsch automatischer Kompletcheck jeder eingelegten Diskette, Verwaltung beliebig großer Bootblöcke, flexible Tastaturbelegung, Fensterkoordinatenspeicherung, Dateipfadpufferung, Warntonoption, Multiselect, Befehlsdateimodus, Online-Hilfe und vieles mehr....

1.4 schnelleinstieg

Schnelleinstieg für Anfänger

Herzlichen Glückwunsch zum Erwerb von VIRUS CONTROL. Sie halten soeben eines der leistungsfähigsten Antivirensysteme in der Hand. VIRUS CONTROL ist völlig unkompliziert einzusetzen, einfach starten und VIRUS CONTROL bleibt bis zum nächsten Ausschalten aufgrund seiner Resetfestigkeit aktiv, wobei VIRUS CONTROL völlig unauffällig im Hintergrund arbeitet. Sie werden nicht andauernd von speicherplatzfressenden Fenstern belästigt. Beim Einlegen einer mit Viren verseuchten Diskette erscheint ein Fenster, in dem Sie auf diesen Umstand hingewiesen werden. Hier können Sie dann verschiedene Aktionen auslösen. Ebenso wird auch der Arbeitsspeicher permanent überwacht, damit sich auch dort kein Virus einnisten kann. VIRUS CONTROL bietet Ihnen also ständigen Schutz, ohne viel Rechenzeit zu beanspruchen.

Zu Beginn möchte ich einige Fachbegriffe erklären. Neben dem Arbeitsfenster von VIRUS CONTROL wird des öfteren auch von Dialogfenstern gesprochen. Ein Dialogfenster ist ein Fenster, in welchem man nur zwischen zwei Aktionen auswählen kann, also z.B. zwischen JA oder NEIN.

Weiterhin ist des öfteren von Load-Files die Rede. Load-Files sind Files, auch Programme genannt, die ausführbar bzw. startbar sind. Man kann diese Files oder Programme in der Shell(=CLI) durch Eintippen oder über die Workbench durch Anklicken starten. Nicht-Load-Files sind demnach Files, auch Dateien genannt, welche vom Betriebssystem nicht geladen und nicht gestartet werden können. Nicht-Load-Files können Daten aller Art beinhalten. Diese Daten werden dann oftmals von gewissen Programmen verwendet. Der Begriff File oder Datei wird oftmals als Oberbegriff für alle Arten von Files oder Dateien verwandt.

```
Load-Files (=startbare Programme)
/
File(=Datei) -
\
Nicht-Load-Files (=nicht startbare Dateien, z.B. Texte)
```

Nun aber zu dem eigentlichen Thema, den Viren.
Es gibt hauptsächlich zwei Gruppen von Viren auf dem Amiga.

Die Bootblockviren werden von VIRUS CONTROL automatisch beim Einlegen einer Diskette erkannt. Es erscheint dann das Arbeitsfenster von VIRUS CONTROL. In der schwarzen Tafel in der Mitte können Sie lesen, ob es sich um einen Bootblockvirus handelt. Der Virus wird in den allermeisten Fällen namentlich erkannt und angezeigt.

Mit Installiere DFX: entfernen Sie nun den Bootblockvirus. Vor dem Entfernen des womöglichen Bootblockvirus sollten Sie sicherheitshalber

eine Kopie der Diskette anfertigen, da z.B. manche Spiele-Disketten einen Spezial-Spiellade-Bootblock besitzen, der zwar auch etwas verdächtig aussieht, aber dennoch nicht mit 'Installiere DFX:' vernichtet werden darf. Die Mehrzahl aller Disketten kann man aber problemlos mit 'Installiere DFX:' behandeln.

Zu der anderen Virengruppe zählen die Fileviren, Linkviren, Mailboxviren, Trojanischen Pferde und Disk-Validatorviren. Es handelt sich hierbei also um echte ausführbare Programme. Diese Viren werden bei eingeschaltetem Diskeinlegen->Schnelltest zu circa 75% auch automatisch beim Einlegen einer Diskette erkannt. Um aber ganz sicher zu gehen, muß man jedoch das VIRUS CONTROL-Arbeitsfenster aufrufen, was z.B. jederzeit durch erneuten Aufruf des VIRUS CONTROL-Programms oder einfach durch gleichzeitiges Drücken der linken <ALT>-Taste und der 0-Taste aus dem Zehnerblock möglich ist. In dem VIRUS CONTROL-Arbeitsfenster klicken Sie nun Viren suchen+entf. an. In dem nun erscheinenden Dateiauswahlfenster geben Sie z.B. DF0: oder DH0: ein. Danach werden dann alle Programme auf DF0: oder DH0: nach Fileviren, Linkviren, Mailboxviren, Trojanischen Pferden und Disk-Validatorviren durchsucht. Sollte ein Virus gefunden werden, dann werden Sie gefragt ob er entfernt werden soll.

Ich wiederhole: Mit 'Installiere DFX:' entfernt man Bootblockviren und mit 'Viren suchen+entf.' sucht und entfernt man die restlichen File-, Link-, Mailbox-, Trojanischen Pferde und Disk-Validator-Viren.

Rufen Sie also das VIRUS CONTROL-Arbeitsfenster auf, legen Sie dann die zu untersuchende Diskette ein, wobei dann sofort ein eventueller Bootblockvirus-Befall in der schwarzen Tafel angezeigt wird. Entfernen Sie den Bootblockvirus nach einer eventuellen vorherigen Sicherheitskopie mit 'Installiere DFX:'. Danach sollten Sie noch 'Viren suchen+entf.' anklicken, damit die komplette Diskette auch noch nach File-, Link-, Mailbox-, Trojanischen Pferden und Disk-Validator-Viren durchsucht wird.

Es empfiehlt sich 'Entpackversuch ...' anzuwählen, damit VIRUS CONTROL auch in gepackten Programmen und Archiven nach Viren sucht. Alle sonstigen Einstellungen sollten Sie zunächst einfach ignorieren.

Wenn Sie eine Festplatte auf Viren überprüfen wollen, dann müssen Sie folgendes bedenken. Eine Festplatte ist nicht direkt mit einer Diskette vergleichbar. Es gibt bei einer Festplatte anstatt eines Bootblocks einen Rigid-Disk-Block, in welchem nähere Informationen über die Festplatte gespeichert werden. Dieser Rigid-Disk-Block kann nicht wie bei Disketten von Bootblockviren infiziert werden, sondern er kann lediglich beschädigt werden, was aber noch schlimmer ist, da dann die Festplatte nicht mehr erkannt wird.

Also bei einer Diskette benützt man 'Installiere DFX:' zum Entfernen von Bootblockviren und 'Viren suchen+entf.' zum Entfernen der übrigen Viren.

Bei einer Festplatte benutzt man ebenfalls 'Viren suchen+entf.' zum Entfernen der File-, Link-, Trojanischen Pferde und Mailboxviren. Disk-Validator-Viren gibt es bei Festplatten nicht. Anstatt der Bootblock-Funktionen gibt es aber für die Festplatte spezielle Funktionen, die Sie unter Speichermedien finden. Hier können Sie die Partitionsparameter Ihrer Festplatte als mountlist-Eintrag abspeichern und auch den kompletten 'Rigid-Disk-Block', der durchaus über 100 KB lang sein kann, sichern. Beide Dateien sollten Sie auf einer separaten Diskette abspeichern. Sollte

nun ein Virus den 'Rigid-Disk-Block' zerstören, wodurch also die Platte nicht mehr erkannt wird, dann können Sie mittels des mountlist-Eintrages Ihre Festplatte von Hand mounten, das heißt anmelden und anschließend, da nun die Festplatte wieder ansprechbar ist, den vorher abgespeicherten Rigid-Disk-Block auf die Festplatte schreiben, wodurch die Festplatte nun wieder automatisch bootet. Normalerweise braucht man aber diese 'Rigid-Disk-Daten'-Funktionen eher sehr selten.

Oftmals kommt es aber vor, daß der Virus nicht nur auf dem Datenträger steht, sondern daß der Virus auch schon aktiviert wurde, indem z.B. von einer mit einem Bootblockvirus infizierten Diskette gebootet wurde oder weil z.B. in der startup-sequence ein Virusfile aufgerufen wurde oder aber man hat selber ein Virusprogramm (unabsichtlich) gestartet.

VIRUS CONTROL überprüft deshalb z.B. sekundlich den Amiga auf eventuelle Systemveränderungen. VIRUS CONTROL prüft sehr streng. Bei vielen Veränderungen kann VIRUS CONTROL selber erkennen, daß die Veränderungen nicht durch einen Virus, sondern z.B. durch den setpatch-Befehl verursacht sind, also harmlos sind. Es gibt aber auch Veränderungen, die VIRUS CONTROL als verdächtig meldet, wo aber dennoch kein Virus im Spiel ist. Meistens sind die Systemveränderungen also durch harmlose Programme bedingt.

Aber insbesondere bei Änderungen an COLD, COOL, DOIIO, TD-BeginIO, PutMsg, LoadSeg, KickTag, KickChecksum muß aber an einen Virus gedacht werden!! Die Warnmeldungen werden z.B. nach 3 Sekunden wieder automatisch ausgeblendet. Sie können aber jederzeit mit Systemübersicht eine Anzeige der Systemveränderungen anfordern. Wenn sicher ist, daß die Warnmeldungen durch ein harmloses Programm bedingt sind, dann können Sie ganz einfach mit dem Lern-Modus VIRUS CONTROL davon in Kenntnis setzen und Sie werden in Zukunft nicht mehr durch diese Warnmeldungen gestört.

1.5 installation

Installation

VIRUS CONTROL können Sie einfach durch Anklicken des Installierungsspiktogramms installieren, wobei der Commodore-Installer zum Einsatz kommt. Aber auch von Hand ist VIRUS CONTROL völlig problemlos zu installieren, da VIRUS CONTROL nur aus einem File besteht. Dieses kann entweder über die Workbench oder das CLI in das gewünschte Directory kopiert werden. Um auch die OnLine-Hilfe und Vireninformationen abrufen zu können, sollten Sie noch die Datei 'VIRUS CONTROL.guide' in das logische 'S:'-Verzeichnis kopieren. ←

Beim Betrieb von VIRUS CONTROL werden dann u.U. weitere Dateien im logischen 'S:'-Verzeichnis angelegt.

S:VCprotokoll(Datum)	zusätzliches Protokoll bei 'Viren suchen+entf.' und bei 'Dateiveränderungen'
----------------------	--

S:NoWarning, S:Virusname	in diesen beiden Verzeichnissen können Bootblöcke archiviert werden
--------------------------	---

S:VCstartup enthält die VIRUS CONTROL-Einstellungen

Die VIRUS CONTROL-Einstellungen werden neben S:VCstartup auch in der zu VIRUS CONTROL gehörenden .info-Datei abgespeichert und auch in ENVARC:VIRUS CONTROL/VIRUS CONTROL.prefs.

Die VIRUS CONTROL-Einstellungen werden über die Menüleiste des VIRUS CONTROL-Arbeitsfensters erreicht und mit Einstellungen speichern dauerhaft gesichert.

Man kann VIRUS CONTROL auf alle üblichen Arten starten, z.B. im CLI oder in der startup-sequence oder in user-startup mit oder ohne run. Auch ein Workbench-Start durch einen Doppelklick wird unterstützt. Ab Kickstart 2.0 kann man VIRUS CONTROL auch in das WBStartup-Verzeichnis legen.

Wenn sie zusätzlich zu VIRUS CONTROL ein weiteres Antivirusprogramm starten, dann wird dieses Programm womöglich VIRUS CONTROL für einen Virus halten, da COLD, COOL, KickTag und KickChecksum durch VIRUS CONTROL verändert wurden. VIRUS CONTROL muß jedoch diese Vektoren verbiegen, um sich resetfest zu machen. VIRUS CONTROL ist sogar auf 1 MB-Chip-Speicher-Amigas unter Kickstart 1.2/1.3 ohne 'setpatch r' resetfest. Durch die Resetfestigkeit bleibt VIRUS CONTROL normalerweise bis zum Ausschalten des Rechners aktiv, leider gibt es einige wenige Programme (z.B. manche Intros, Spiele), die völlig willkürlich auch eventuell schon belegten Speicher beschreiben, und dadurch womöglich VIRUS CONTROL überschreiben, wodurch VIRUS CONTROL nach Spielende und Reset nicht mehr aktiv ist.

Sollten sie Fast-Speicher von Hand mittels z.B. addmem einbinden, dann muß VIRUS CONTROL vor diesem addmem gestartet werden, da dieses nicht auto-konfigurierende Fast-Speicher nach einem Reset noch nicht vorhanden ist, und VIRUS CONTROL somit ins Leere greifen würde.

Im Normalfall (autokonfigurierender Fast-Speicher) werden Sie aber nie auf Probleme stoßen. Notfalls können Sie vor dem Starten von VIRUS CONTROL run noFast-Speicher aufrufen, wodurch dann VIRUS CONTROL vollständig im unkritischen Chip-Speicher angelegt wird.

VIRUS CONTROL installiert ferner einen Task, der sekundlich alle wichtigen Systemadressen auf Veränderungen überprüft. Dieser Task wiederum installiert einen Prozeß, der die Disketten überprüft. Weiterhin wird ein Inputhandler und Keyboardresethandler installiert.

1.6 selbsttest

VIRUS CONTROL-Selbsttest

VIRUS CONTROL überprüft sein Programm beim Start immer automatisch auf Veränderungen oder einen eventuellen Linkvirusbefall. Eine Veränderung wird mittels Dialogfenster angezeigt. Man kann das Erscheinen des Warn-Dialogfensters verhindern, indem man in der Menüleiste 'VC-Selbsttest bei Aufruf' abschaltet. Hierdurch wird die automatische Start-Kontrolle abgeschaltet. Es erscheint dann also trotz verändertem

VIRUS CONTROL-Programm kein Warn-Dialogfenster mehr.

1.7 arbeitsfenster öffnen

VIRUS CONTROL-Arbeitsfenster öffnen

Beim Aufrufen von VIRUS CONTROL wird ein DF0:-Arbeitsfenster geöffnet. Anschließend werden alle weiteren eingelegten Disketten überprüft und nur bei Virus-Verdacht ebenfalls ein Arbeitsfenster ausgegeben. Durch Abwahl von 'VC-Aufruf->Arbeitsfenster' in der Menüleiste können Sie VIRUS CONTROL veranlassen in Zukunft beim Start des Programmes nur noch dann ein Arbeitsfenster zu öffnen, wenn eine verdächtige Diskette eingelegt sein sollte. So verhält sich VIRUS CONTROL auch wenn es automatisch durch die startup-sequence oder WBstartup aufgerufen wird. Es erscheint dann also nur dann ein Arbeitsfenster, wenn eine verdächtige Diskette gefunden wird. Beim Starten durch die startup-sequence oder WBstartup wird also nicht immer ein DF0:-Arbeitsfenster geöffnet. Man erspart sich dadurch das lästige Wegklicken des DF0:-Arbeitsfensters, wenn von einer normalen Arbeitsdiskette oder Festplatte gebootet wird.

Wenn Sie VIRUS CONTROL bereits beim Booten aktivieren wollen, dann tragen Sie am besten VIRUS CONTROL als zweites Programm nach dem setpatch-Aufruf in s:startup-sequence ein. Unter Kickstart 1.3 genügt das Eintragen von z.B. VIRUS-CONTROL, ab Kickstart 2.0 sollten Sie VIRUS-CONTROL >NIL: eintragen, um ein verfrühtes Öffnen der Workbench zu verhindern. Commodore empfiehlt, ab Kickstart 2.0 die startup-sequence nicht mehr zu verändern, denn ab Kickstart 2.0 stehen die Alternativen s:user-startup oder WBstartup zu Verfügung. Sie können natürlich VIRUS CONTROL auch mit einer dieser beiden Methoden starten, allerdings kann VIRUS CONTROL im Falle eines Aufrufes durch s:user-startup nicht mehr erkennen, daß ein Bootvorgang vorliegt und wird daher immer ein DF0:-Arbeitsfenster öffnen, obwohl womöglich keine verdächtigen Disketten eingelegt sind. Sie können dies verhindern, indem Sie 'VC-Aufruf->Arbeitsfenster' abwählen.

Wenn VIRUS CONTROL also automatisch während des Bootens aufgerufen wird, dann wird normalerweise kein in dem Fall störendes Arbeitsfenster geöffnet. Wenn Sie aber nach dem Bootvorgang VIRUS CONTROL absichtlich über CLI oder Workbench aufrufen, dann wird immer ein Arbeitsfenster geöffnet, und Sie können sich nun der vielfältigen Funktionen von VIRUS CONTROL bedienen.

Es existieren folgende Möglichkeiten, um das Arbeitsfenster zu öffnen:

VIRUS CONTROL erneut über CLI oder Workbench aufrufen

gewünschtes Laufwerk mit linke <ALT>-Taste + 0,1,2,3(Zehnerblock) anwählen. Beliebige Einstellung in Tastaturbelegung möglich.

Einlegen einer 'verdächtigen' Diskette, d.h. die Diskette besitzt einen Nicht-Standard-Bootblock, es könnte sich also um einen Bootblockvirus handeln.

Einlegen einer beliebigen Diskette bei gedrückter linker <ALT>-Taste

commodity 'Anzeige sichtbar' anklicken

das Schließsymbol der Fensterleiste anklicken

im Arbeitsfenster entsprechendes Laufwerks-Symbol anklicken

Wird während der Arbeit mit dem Amiga eine verdächtige Diskette eingelegt, dann wird dieses durch das Erscheinen eines Arbeitsfensters deutlich gemacht. Hält man während des Disk-Einlegens die rechte <ALT>-TASTE gedrückt, so wird die Diskette nicht durch VIRUS CONTROL überprüft. Hält man während des Disk-Einlegens die linke <ALT>-TASTE gedrückt, so erscheint auch bei nicht verdächtigen Disketten das Arbeitsfenster.

Das VIRUS CONTROL-Arbeitsfenster bietet Ihnen eine Vielzahl sehr nützlicher Funktionen zur Virenbekämpfung an. Für die wichtigsten Symbole besteht auch die Möglichkeit die betreffende Funktion alternativ über die Tastatur anzuwählen. Diese Funktionen sind durch einen unterstrichenen Buchstaben gekennzeichnet. Weiterhin finden sich in der Menüleiste eine große Anzahl von Einstellungsparametern. In die Menüleiste gelangen Sie, indem Sie bei aktivem Arbeitsfenster die rechte Maustaste drücken. Alle VIRUS CONTROL-Einstellungen sind resetfest.

Bei der Arbeit mit VIRUS CONTROL erscheinen auch des öfteren sogenannte Dialogfenster, die immer zwei Möglichkeiten zur Auswahl stellen. Wenn Sie das linke Symbol anklicken, dann leiten Sie damit in der Regel einen Vorgang ein. Alternativ können Sie auch die j-Taste wie JA drücken. In Tastaturbelegung können Sie bestimmen, ob auch das Drücken von <Return> oder <Enter> als JA gewertet werden soll. Wenn Sie das rechte Symbol anklicken, dann brechen Sie damit in der Regel ein Vorgang ab. Wenn Sie also nicht sicher sind, was Sie anwählen sollen, dann sollten Sie sicherheitshalber das rechte Symbol anklicken, da dadurch nichts unternommen wird. Anstatt das rechte Symbol anzuklicken können Sie auch jeden Bereich des Dialogfensters außer dem Bereich des linken Symbols anklicken oder auch eine beliebige Taste außer der j-Taste drücken.

1.8 schließsymbol

Schließsymbol

schließt das Arbeitsfenster, VIRUS CONTROL ist aber weiterhin im Hintergrund aktiv. Sie können das Arbeitsfenster auch durch Anklicken von 'Abbruch' oder der schwarzen Tafel oder durch Drücken der <Leertaste>, <ESC>-Taste oder <CTRL> + <c>-Taste schließen.

1.9 eigener schirm

eigener Schirm

Normalerweise arbeitet VIRUS CONTROL auf der Workbench, durch Anklicken von 'Schirm' können Sie VIRUS CONTROL jedoch anweisen, einen eigenen Schirm zu öffnen. Durch erneutes Anklicken von 'Schirm' wird der eigene Schirm geschlossen und wieder der Workbenchschirm benutzt.

Ab Kickstart 3.0 können Sie mit 'Schirm wählen ...' einen beliebigen Schirm für VIRUS CONTROL auswählen.

1.10 reset

Reset

Sie können zwischen 3 Reset-Arten wählen.

NORMAL-RESET

Es wird ein normaler Reset ausgelöst, der der Tastenkombination <CTRL> + <L-Amiga> + <R-Amiga> entspricht. Hierbei bleiben resetfeste Programme wie z.B. VIRUS CONTROL erhalten. Weiterhin werden eventuelle auto-konfigurierende Erweiterungskarten eingebunden.

SOFT-RESET

Es wird ein sogenannter 'weicher' Reset ausgelöst. Hierbei werden keine auto-konfigurierenden Erweiterungskarten eingebunden. Dadurch werden also viele Fast-Speicher-Karten und Festplatten nicht sichtbar.

FULL-RESET

Es wird ein 'Kalt'-Start ausgelöst. Hierbei wird das Betriebssystem komplett neu aufgebaut. Dadurch werden alle resetfesten Programme, also auch VIRUS CONTROL selber, entfernt. Bei diesem Reset werden wie beim NORMAL-RESET eventuelle Erweiterungskarten eingebunden. Der FULL-RESET entspricht also weitgehend einem Aus- und Einschalten des Computers. Dennoch ist manchmal ein Aus- und Ein-Schalten des Computers unerlässlich, weil nur dadurch z.B. der Inhalt des Fast-Speichers gelöscht wird. Zwischen dem Aus- und Einschalten des Computers sollte eine Pause von mindestens 20 Sekunden liegen, da sich die Speicher-Kondensatoren erst nach einigen Sekunden vollständig entladen.

1.11 kill

Kill

Es wird versucht, einen aktiven Virus aus dem Speicher zu entfernen, siehe Virusentfern-Dialogfenster .

1.12 df0df1df2df3

DF0: DF1: DF2: DF3:

Durch Anklicken dieser Symbole können sie das entsprechende Laufwerk überprüfen. Nicht vorhandene Laufwerke werden 'ghosted' dargestellt und können nicht ausgewählt werden. Das momentan ausgewählte Laufwerk wird durch einen gedrückten Knopf gekennzeichnet. Ferner zeigt ein Diskettensymbol an, ob eine Diskette eingelegt ist. Sollte das Diskettensymbol rechts oben ein blaues Viereck aufweisen, dann wird damit die Beschreibbarkeit der Diskette angezeigt.

1.13 zeige dfx

Zeige DFX:

zeigt den Bootblock (1024 Bytes) als ASCII-Dump oder HEX-Dump an. Diese Funktion ist sehr nützlich, da sich viele Bootblockviren durch gewisse Strukturen oder gar Texte zu erkennen geben. Sie können mit den Tasten oder dem Schieberegler über die ganze Diskette wandern, also auch Daten außerhalb des Bootblockbereiches anzeigen lassen. Sie können beliebig viele Bootblockfenster gleichzeitig bedienen, da für jedes Fenster ein völlig eigenständiger Prozess gestartet wird. Durch Anklicken von Disass wird ebenfalls ein völlig eigenständiges Fenster geöffnet, in welchem die Daten disassembliert und kommentiert angezeigt werden.

1.14 installiere dfx

Installiere DFX:

schreibt den Original-Amiga-Bootblock auf Diskette (entspricht dem CLI-Befehl 'install', der jedoch bei einem aktiven Bootblockvirus versagt). Hierbei werden auch FFS oder international formatierte Disketten berücksichtigt. Durch diese Funktion kann also ein Bootblockvirus von der Diskette gelöscht werden. Siehe auch Schnelleinstieg .

mit Kickstart 1.2/1.3- oder Kickstart 2.0-Bootblock-Code

Die neueren Workbench 2.0 - Versionen verwenden einen geringfügig längeren Bootblock-Code. VIRUS CONTROL benutzt normalerweise ebenfalls diesen neuen Code. Man kann jedoch VIRUS CONTROL anweisen, den alten Kickstart1.3-Bootblock-Code zu verwenden, indem man den Menüpunkt 'Installieren mit Kick2-BB' abwählt.

Unter Kickstart1.3 verhalten ist beide Bootblöcke völlig gleich. Es ist also egal, welchen Bootblock-Code Sie unter Kickstart 1.3 verwenden. Lediglich ab Kickstart 2.0 verhalten sich die Bootblöcke verschiedenartig. Wenn man ab Kickstart 2.0 den alten Bootblock-Code verwendet, dann wird vor Beginn des Abarbeitens der startup-sequence die Workbench mit dem Amiga-DOS-Fenster geöffnet, so wie es auch unter Kickstart1.3 gemacht wird. Dies geschieht aus Kompatibilitätsgründen mit einigen unsauber programmierten Spielen. Wenn man aber ab Kickstart 2.0 den neuen Kickstart2.0-Bootblock-Code verwendet, dann wird das sogenannte silent-startup-Verhalten unterstützt. Das heißt, die Workbench wird erst bei der ersten Textausgabe geöffnet. Dadurch können also bereits Befehle der startup-sequence abgearbeitet werden, die z.B. die Workbenchgröße und Auflösung festlegen. Wäre zu diesem Zeitpunkt bereits die Workbench geöffnet, dann könnte nicht mehr automatisch in eine andere Workbench-Auflösung gewechselt werden, sondern es müßten zuvor alle Fensters einschließlich dem Amiga-DOS-Fenster geschlossen werden. Man kann also auch ab Kickstart 2.0 den alten Bootblock-Code verwenden. Da daurch aber das silent-startup-Verhalten ausgeschaltet wird, muß man unter Umständen von Hand alle Fensters schließen, wenn man in eine andere als die Standard-Workbench-Auflösung wechseln will.

1.15 bootblock->datei

Bootblock->Datei

In dem erscheinenden Dateiauswahlfenster geben Sie die Datei an, in welche die Bootblockdaten abgespeichert werden sollen. Danach werden Sie gefragt, wieviele 512-Byte-Blöcke abgespeichert werden sollen.

Da ein Bootblock aus 1024 Bytes, also zwei 512-Byte-Blöcken besteht, sollten Sie die empfohlene Vorgabe von zwei 512-Byte-Blöcken übernehmen. Es gibt einige wenige Bootblockviren welche z.B. vier 512-Byte-Blöcke belegen, in diesem Fall können Sie dann vier 512-Byte-Blöcke anwählen. Durch die Angabe von 1760 könnten Sie eine komplette DD-Disk in einer Datei ablegen.

1.16 bootblock->puffer

Bootblock->Puffer

legt den 1024 Byte langen Bootblock in einem programminternen Puffer ab.

1.17 puffer ->bootblock

Puffer ->Bootblock

Der programminterne Puffer wird als Bootblock auf Diskette geschrieben.

1.18 datei ->bootblock

Datei ->Bootblock

In dem erscheinenden Dateiauswahlfenster geben Sie die Datei an, deren Inhalt auf den Bootblock geschrieben werden soll. Sollte die Länge der angewählten Datei größer 1024 Bytes sein, dann werden Sie gefragt, wieviele 512-Byte-Blöcke abgespeichert werden sollen. Da ein Bootblock aus 1024 Bytes, also zwei 512-Byte-Blöcken besteht, sollten Sie die empfohlene Vorgabe von zwei 512-Byte-Blöcken übernehmen. Es gibt einige wenige Bootblockviren oder Boot-Utilities, welche z.B. vier 512-Byte-Blöcke belegen, in diesem Fall können Sie dann z.B. vier 512-Byte-Blöcke anwählen, und dadurch z.B. den kompletten Virus auf die Diskette schreiben. Im Normalfall sollten Sie also immer mit zwei 512-Byte-Blöcken arbeiten, denn nur diese Einstellung wird vom Betriebssystem unterstützt. Wenn mehr wie zwei 512-Byte-Blöcke geschrieben werden, werden insbesondere bei vollen Disketten bereits Daten überschrieben. Lediglich die unteren zwei 512-Byte-Blöcke sind als Bootblock frei verfügbar.

Mit dieser Möglichkeit, den Bootblock in ein File abspeichern zu können, und dieses File auch wieder auf den Bootblock schreiben zu können, kann man sehr einfach von allen wichtigen Disketten eine Bootblock-Kopie anfertigen, und somit notfalls den Original-Bootblock wiederherstellen. Noch sicherer ist

es natürlich die Originaldiskette komplett zu kopieren und nur mit der Kopie zu arbeiten.

1.19 f.prüfsum->noboot

f.Prüfsum->NoBoot

es wird eine falsche Bootblock-Checksumme auf die Diskette geschrieben, wodurch von der Diskette nicht mehr gebootet werden kann. Ein eventueller Bootblockvirus ist dadurch (vorübergehend) unschädlich gemacht.

1.20 r.prüfsum->boot

r.Prüfsum->Boot

Es wird eine richtige Bootblock-Checksumme geschrieben, wodurch von der Diskette wieder gebootet werden, und somit auch ein eventueller Virus wieder aktiviert werden kann.

1.21 personal bootblock

Personal Bootblock

erstellt einen individuellen Bootblock.

Mit WB-Breite und WB-Höhe können Sie eine sogenannte Overscan-Workbench erstellen. Normalerweise ist die Workbench 640 Pixel breit und 256 Pixel hoch. Wenn Sie eine breitere und oder höhere Workbench wünschen, dann können Sie hier Werte zwischen 0 und 127 eintragen.

Wenn Sie 'Fast-Speicher AN' auf 'Fast-Speicher AUS' klicken, dann wird beim ↵ nächsten Booten das komplette Fast-Speicher belegt, wodurch dann nur noch Chip-Speicher verfügbar ist.

Normalerweise wird das AmigaDOS-CLI-Fenster in 640/200 geöffnet. 'BIG-CLI AN' öffnet nun das AmigaDOS-CLI-Fenster in der größtmöglichen Ausdehnung geöffnet. Ein eventueller Overscan wird hierbei berücksichtigt.

Mit PAL/NTSC-Auflösung AN können Sie bestimmen, in welcher Auflösung der Amiga booten soll. Wirklich sinnvoll ist diese Option allerdings nur bei einem neuen 1MB-Big-Agnus.

Normalerweise ist der Audio-Filter eingeschaltet. Bei manchen Musikstücken erreicht man aber einen volleren Klang, wenn man den Filter ausschaltet. Dies können Sie durch Filter AN/AUS einstellen. Bei eingeschaltetem Filter leuchtet die LED hell. Bei ausgeschaltetem Filter leuchtet die LED schwächer.

Wenn Sie ext.Laufwerke auf AUS klicken, dann werden eventuelle externe

Laufwerke vom Amiga-DOS nicht eingebunden. Sie sparen dadurch jeweils circa 20 KB ein.

Wenn Sie aber auch Laufwerke vertauschen wollen, dann benutzen Sie hierzu die 'Laufwerke ändern' - Möglichkeit von VIRUS CONTROL. Hiermit können sie Laufwerke nach Belieben vertauschen und ausschalten.

Unter Kickstart 2.0 bleibt Overscan, BIG-CLI und PAL/NTSC ohne Wirkung, da unter Kickstart 2.0 diese Dinge im Gegensatz zu Kickstart 1.3 bereits vom Betriebssystem unterstützt werden.

Ich habe mit Absicht keine sogenannte Antivirusoption eingebaut, denn ein lediglich 1024-Byte langer Bootblock kann niemals einen umfassenden Virus-Schutz sicherstellen.

So machen sich zum Beispiel die neuen gefährlichen Linkviren wie z.B. Xeno-Virus und TheTravelingJack nicht resetfest, und können daher von AntivirusBootblöcken auch nicht gefunden werden. Ein primitives Testen der Reset-Vektoren kann also niemals sicher vor Viren schützen.

Wenn man nun eine Diskette mit einem Personal-Bootblock versehen hat, dann wird beim Booten kurz die gewählte Einstellung angezeigt. Sollte diese Anzeige nicht mehr erscheinen, dann befindet sich der Personal-Bootblock nicht mehr auf der Diskette, womöglich wurde er von einem Virus überschrieben.

Leider erscheinen immer wieder neue Bootblockbearbeitungsprogramme, mit deren Hilfe man spezielle Bootblöcke erstellen kann. Dadurch wächst die Anzahl der Nicht-Standard-Bootblöcke weiter an und die echten Bootblockviren haben es leichter, sich unter der Vielzahl dieser Nicht-Standard-Bootblöcke zu verstecken.

Benutzen Sie daher den Personal-Bootblock nur bei echtem Bedarf, ab Kickstart 2.0 werden Sie den Personal-Bootblock wohl nicht mehr benötigen.

1.22 vergleichen

Vergleichen

Sie können Dateien mit Dateien und Disketten mit Dateien und Disketten mit Disketten vergleichen. Beim Vergleichen von Dateien können Sie Offsets angeben, mit denen Sie bestimmen können, ab welcher Position die Datei verglichen werden soll.

Es erscheinen zwei Fenster, in welchen optisch die Unterschiede angezeigt werden, ferner wird in der Titelzeile die Gesamtanzahl der gefundenen Unterschiede angezeigt. Sie können nun bei Bedarf durch ganze Dateien oder über die ganze Diskette wandern, die Daten werden automatisch in beiden Fenstern aktualisiert.

Mit 'Vergleichen' können Sie z.B. sehr einfach erkennen, ob z.B. ein vermeintlich neuer Bootblockvirus lediglich ein z.B. im Text veränderter Abkömmling eines bereits bekannten Virus ist.

1.23 zeige datei

Zeige Datei

Ein Anklicken von 'Zeige Datei' zeigt eine Datei als ASCII-Dump oder HEX-Dump an. Die Datei, die angezeigt werden soll, wird mittels des Dateiauswahlfensters ausgewählt. Hierbei wird bereits DFX:s/startup-sequence vorgegeben, da man mit dieser Funktion meist die erste Zeile der startup-sequence auf einen eventuellen Filevirusbefall überprüfen möchte, denn aufgrund der zusätzlichen Hexadezimal-Darstellung im HEX-Modus kann man erkennen, ob z.B. ein File namens \$a0a0a0 (BUTONIC-JEFF-Virus-V3.00) oder \$a0a0a0a0a0 (RevengeOfTheLamer) durch die startup-sequence aufgerufen wird. Normalerweise erscheinen diese 'unsichtbaren' Zeichen in den Editoren und Textverarbeitungen einfach als Leerzeichen und werden somit also nicht erkannt.

1.24 dateiveränderungen

Dateiveränderungen erkennen -> Linkvirusschutz

Bisher gibt es erst zwei Methoden, sich gegen Linkviren zu schützen.

Das eine sind Programme, die gezielt einen Linkvirus entfernen können. Positiv ist, daß man meist wieder das Originalfile zurückerhält. Der große Nachteil dieser Programme ist, daß sie nur diesen einen Linkvirus entfernen können. Bei zukünftigen Linkviren werden diese Programme meist unwirksam sein.

Das andere sind Programme, die die Filelänge und eine File-Prüfsumme in einer separaten Datei abspeichern. Mit Hilfe dieser Datei kann man nun überprüfen, ob die Files noch unverändert sind. Diese Methode ist durchaus wirksam, aber sie erfordert sorgfältiges Vorgehen und ist recht umständlich.

Selbstverständlich beherrscht VIRUS CONTROL die erstgenannte Methode, also das gezielte Ausbauen von bekannten Linkviren, auch die zweitgenannte Methode beherrscht VIRUS CONTROL, allerdings in einer sehr komfortablen und neuartigen Form.

Wie schon gerade erwähnt bestand bisher der Schutz vor Linkviren darin, die Filelänge und eine Prüfsumme über den Fileinhalt in einer separaten Datei abzuspeichern. Man kann nun regelmäßig mit Hilfe der Datei überprüfen, ob die Files noch unverändert sind. Dieses Verfahren ist zwar recht wirksam, aber leider sehr unkomfortabel. Man muß immer mit einer zusätzlichen Diskette hantieren, auf der man die Prüfdatei abgespeichert hat. Meist muß man sogar mit mehreren Prüfdateien arbeiten, z.B. eine Prüfdatei für die Festplatte, eine Prüfdatei für die Startup-Diskette, eine weitere Prüfdatei für die Extras-Diskette, und noch weitere Prüfdateien für weitere wichtige Disketten. Dieses Verfahren wird dadurch leider etwas umständlich und unübersichtlich.

Ich löste dieses Problem dadurch, indem ich die Originalfilelänge und Originalprüfsumme nicht in eine extra Prüfdatei schreibe, sondern indem ich diese Werte direkt an das File anhänge. Das umständliche Hantieren mit verschiedenen Prüfdateien entfällt vollständig.

Das VIRUS CONTROL-File-Schutz-System hat einen weiteren großen Vorteil. Angenommen Sie erstellen ein gutes Programm, schützen es mit VIRUS CONTROL,

und lassen es dann als PublicDomain vertreiben. Das Programm verbreitet sich nun über die ganze Welt und geht somit durch tausende Hände. Hierbei ist ein hohes Infektionsrisiko gegeben. Schließlich erhält ein Amigafan in z.B. Japan das Programm. Er kann nun einfach mittels VIRUS CONTROL prüfen, ob sich das Programm noch im Originalzustand befindet. Das VIRUS CONTROL-File-Schutz-System kann also weltweit wirksam werden, eine Prüfdatei hingegen kann nur von privatem Nutzen sein.

Das File-Schutz-System wird mit dem Symbol 'Dateiveränderungen' aufgerufen. Es erscheint nun das Dateiauswahlfenster, mit dessen Hilfe Sie ein File oder Verzeichnis anwählen können. Das z.B. angegebene File wird nach der VIRUS CONTROL-Schutz-Kennung durchsucht. Weist das File noch keine Schutz-Kennung auf, dann wird, sofern es sich um ein load-file handelt, automatisch eine 24 Byte lange File-Schutz-Kennung an das Fileende angebracht, in welcher die aktuelle Filelänge und eine Dateiprüfsumme abgespeichert ist. Wenn man 'ungeschützte Dateien schützen' in der Menüleiste abwählt, dann wird ein File erst nach Rückfrage geschützt. Nicht-Load-Files wie z.B. Texte usw. werden nur beachtet, wenn man in der Menüleiste 'alle Dateien schützen' anwählt. In dem Fall kann man auch Nicht-Load-Files (nach Rückfrage) schützen. Im Normalfall werden NichtLoadFiles aber ignoriert, da es oftmals nicht sinnvoll ist, z.B. ein Text-File zu schützen. Anstatt eines einzelnen Filenamens sind auch Verzeichnisangaben erlaubt. Hierbei werden auch eventuell enthaltene Unterverzeichnisse bearbeitet. Dieses bequeme automatische Durcharbeiten eines kompletten Verzeichnisses kann jederzeit durch Anklicken des Schließsymbols abgebrochen werden.

Weist ein File bereits die Schutz-Kennung auf, dann wird nun geprüft, ob die aktuelle Filelänge noch der in der File-Schutz-Kennung abgespeicherten Original-Filelänge entspricht. Weiterhin wird mittels einer File-Check-Summe geprüft, ob sich der Fileinhalt verändert hat. Sollte sich die Filelänge oder der Fileinhalt verändert haben, dann erscheint ein Dialogfenster, das auf den vermutlichen Linkvirusbefall hinweist. Mittels des Dialogfensters ist es nun möglich das File wieder zu entschützen. Beim nächsten File-Schutz-Durchgang würde dieses File dann wieder mit einer neuen File-Schutz-Kennung versehen werden. Mit File-Entschützen kann jedoch kein Linkvirus entfernt werden, denn da Amiga-DOS für load-files einen sehr flexiblen Aufbau gestattet, sind sehr viele Linkvirentypen denkbar. Es ist also völlig unmöglich ein generelles Linkvirusentfernprogramm zu schreiben. Vielmehr muß für jeden Linkvirus gezielt ein spezielles Entfernenprogramm erstellt werden.

Es empfiehlt sich also folgende Vorgehensweise, um sich vor Linkviren zu schützen.

Sicherheitskopien von den Original-Files anfertigen!!

Original-Files schützen durch Aufruf von 'Dateiveränderungen'

regelmäßig überprüfen ob Files noch unverändert sind durch Aufruf von 'Dateiveränderungen'

Wird jedoch eine Fileveränderung festgestellt, dann deutet dies auf einen Linkvirus hin. Man entfernt nun diesen Linkvirus einfach dadurch, indem man das verdächtige Programm mit dem Original-Programm überschreibt. Hat man jedoch das Original-Programm nicht zur Hand, dann kann man mittels 'Viren suchen+entf.' oder UnLink versuchen, das Original-File

wiederherzustellen. Das neue, nicht infizierte Programm sollte nun sofort wieder geschützt werden usw.

Durch das Anklicken von 'Dateiveränderungen' werden also noch ungeschützte Files geschützt und bereits geschützte Files auf Veränderung überprüft. Dieses geschieht weitgehend automatisch.

Ist jedoch 'ungeschützte Dateien schützen' nicht gesetzt, dann muß das 'File schützen' extra bestätigt werden. Es empfiehlt sich, beim ersten Mal die Verzeichnisse automatisch schützen zu lassen. Dannach sollte man aber 'ungeschützte Files schützen' nicht setzen, denn dadurch wird man in Zukunft extra mittels Dialogfenster darauf aufmerksam gemacht, wenn ein neues File im Verzeichniss vorliegt. Ein Virus könnte durchaus ein gefährliches Programm erstellen, bzw. ein vorliegendes Programm einfach überschreiben. Dieses kann nun nicht mehr übersehen werden. Wenn eine File-Veränderung festgestellt wird, dann muß auch diese wegen eines womöglichen Linkvirusbefalls extra bestätigt werden. Wenn man 'geschützte Dateien entschützen' anwählt, dann bietet VIRUS CONTROL die Möglichkeit an, das File zu entschützen, die File-Schutz-Kennung also wieder zu entfernen, wodurch das Original-Programm wieder erhalten wird. In der Regel ist dieses aber nicht sinnvoll.

Man sollte die C:, L:, LIBS:, und DEVS: -Verzeichnisse schützen, da es sich hier um load-files(=ausführbare Programme) handelt, die entweder vom Benutzer selber (C:), oder vom Betriebssystem (L:,LIBS:,DEVS:) aufgerufen werden. Bei diesem Aufruf könnte dann ein Linkvirus aktiviert werden. Info-Files und ein eventuelles Font-Verzeichnis werden generell nicht geschützt, da es angeblich Programme geben soll, welche geschützte Fonts nicht mehr richtig verarbeiten. Nicht-load-files werden erst nach Rückfrage geschützt, da es normalerweise nicht sehr sinnvoll ist, nicht ausführbare Programme zu schützen.

Meine File-Schutz-Kennung läßt sich aufgrund des 'hunk'-Aufbaus völlig problemlos an load-files anhängen. Für Nicht-load-files ist jedoch eine universelle File-Schutz-Kennung völlig undenkbar, da diese Files ja völlig willkürlich aufgebaut sind. Man muß also ausprobieren, ob ein Nicht-load-file auch nach Schützen noch voll funktionsfähig ist. Wenn nicht, kann es wieder entschützt werden. Lediglich im folgenden Fall erscheint es sinnvoll, Nicht-load-files zu schützen. Es ist denkbar, daß sich ein Virus-Programm in die startup-sequence oder mountlist schreibt. Beim Abarbeiten dieser Files beim Booten könnte dann der Virus aktiviert werden. Ein Schützen dieser Files ist durchaus möglich, da die angehängte Fileschutzkennung lediglich als nicht ausführbarer Befehl gewertet wird. Wenn man nun aber selber diese Files abändert, dann erhält man logischerweise beim Aufruf von 'Dateiveränderungen' eine Linkviruswarnung. Man sollte nun das File entschützen, und durch einen erneuten Aufruf von 'Dateiveränderungen' neu schützen lassen.

1.25 Linkviren

File/Linkviren suchen und entfernen

Nach Anklicken von 'Viren suchen+entf.' erscheint das Dateiauswahlfenster, mit dessen Hilfe Sie ein Verzeichnis oder File auswählen können, das Sie nach File-, Link-, Mailbox-, Trojanischen Pferden und Disk-Validator-Viren durchsuchen möchten. Es werden auch eventuelle Unterverzeichnisse durchsucht, wodurch

also auch komplette Festplatten oder Disketten durchsucht werden können.

VIRUS CONTROL zeigt in einem Fenster alle Files an, die überprüft werden. Sollte ein Virus gefunden werden, dann wird gezielt mittels Dialogfenster darauf hingewiesen. Die Entfernung des Virus ist nun ebenfalls bequem mittels Dialogfenster möglich. Wird ein Entfernen des Virus nicht gewünscht, dann wird zumindest eine Viruswarnnotiz an die betreffende Datei angebracht, aber nur dann, wenn noch keine Dateinotiz vorhanden war.

Normalerweise sollten Sie aus Virenschutzgründen immer die Disketten möglichst schreibgeschützt lassen. Sollte allerdings bereits ein Virus auf der Diskette vorhanden sein, dann müssen Sie die Diskette kurzzeitig beschreibbar machen, damit VIRUS CONTROL den Virus wieder löschen kann. Wenn VIRUS CONTROL einen Virus nicht löschen kann, weil die Diskette schreibgeschützt ist, dann erhalten Sie eine Fehlermeldung mit der Nummer 214.

1.26 fileviren, mailboxviren, trojanische pferde

Fileviren, Mailboxviren, Trojanische Pferde

Sollte ein Filevirus, Mailboxvirus oder Trojanisches Pferd gefunden werden, dann werden Sie mittels Dialogfenster gefragt, ob dieser Virus gelöscht werden soll. Sollte sich dieser Filevirus auch in die startup-sequence schreiben, dann wird dieser Filevirusaufruf in der startup-sequence mit Leerzeichen überschrieben, womit der Filevirus vollständig und fehlerfrei entfernt ist.

1.27 Linkviren

Linkviren

Sollte ein Linkvirus gefunden werden, dann wird der angehängte Viruscode von der Datei abgetrennt, wodurch wieder das Original-File erhalten wird. In seltenen Fällen kann das Original-File jedoch nicht mehr wiederhergestellt werden. Der Grund hierfür ist folgender: Der File-Aufbau der AmigaDOS-Load-Files ist sehr komplex. Ein Linkvirus müßte sehr viele Sonderfälle bedenken, wenn er sich völlig sauber an das Originalfile linken wollte. Oftmals aber berücksichtigen die Linkviren diese selteneren Sonderfälle nicht. Dadurch kommt es dann manchmal beim Infizieren des Files zu unwiderbringlichem Datenverlust. Diese Daten kann dann natürlich auch VIRUS CONTROL nicht mehr herbeizaubern.

Angenommen es liegt ein infiziertes File namens 'Linkvirusfile' vor, dann fragt Sie VIRUS CONTROL wenn 'Linkviruskopie' angewählt ist, ob sicherheitshalber eine Kopie dieses Virusfiles unter dem Namen 'Linkvirusfile.XXX' erstellt werden soll. Unter 'LinkVirusFile' wird dann das Original-File wiederhergestellt. Sollte das rekonstruierte File nun aber nicht laufen, dann haben Sie im Falle der 'Linkviruskopie' noch nach wie vor das infizierte File. Im Normalfall aber wird das Originalfile erfolgreich wiederhergestellt und Sie können das Virusfile mit der Endung .XXX löschen.

1.28 Disk-Validatorviren

Disk-Validatorviren

Wenn Sie eine Diskette einlegen, die mit dem ReturnOfTheLamer oder SADDAM-HUSSEIN-Disk-Validator-Virus usw. infiziert ist, dann erkennt VIRUS CONTROL dieses und verhindert die automatische Aktivierung des Virus. Es erscheint ein Dialogfenster, der Sie über diesen Umstand informiert. Anschließend erscheint das Dateiauswahlfenster, in welchem bereits z.B. DF0:1/Disk-Validator vorbelegt ist. Klicken Sie einfach auf 'OK'. Nun korrigiert VIRUS CONTROL die Root-Spur, wodurch die Diskette nicht mehr als fehlerhaft betrachtet wird. Weiterhin werden eventuelle DatenBlöcke wieder entschlüsselt, so daß auf diese Daten nun auch ohne aktiven SADDAM-HUSSEIN-Virus wieder zugegriffen werden kann. Als letztes wird dann auch noch das eigentliche Disk-Validator-Virusfile (1/Disk-Validator) gelöscht.

Unter Kickstart 1.2/1.3 kann VIRUS CONTROL in seltenen Fällen die Diskette nicht allein reparieren und braucht die Hilfe des Original-Disk-Validators. Es erscheint dann ein Dialogfenster, das Sie anweist, von einer sauberen Workbench-Disk zu booten und danach wieder die VIRUS-Disk einzulegen. Da VIRUS CONTROL zuvor 'Disk-validator' in 'Disk-Validata0' umbenannt hat, ist die Diskette nicht mehr infiziös. Es kann kein Disk-Validator von der Diskette geladen werden. Also versucht das Betriebssystem den Disk-Validator aus L: zu laden. Hierzu wird dann auf die Workbench-Boot-Diskette zugegriffen. Der Workbench-Disk-Validator korrigiert nun die Diskette weiter, so daß sie nun auch wieder beschrieben werden kann. Anschließend wird automatisch 'Viren suchen+entf.' aufgerufen, wobei dann nochmal die ganze Diskette überprüft wird und auch das disk-validat0-File gelöscht wird. Anschließend ist die Diskette sauber.

Wenn Sie eben nicht alles verstanden haben, dann ist das nicht weiter schlimm. Befolgen Sie einfach die Anweisungen von VIRUS CONTROL und die Diskette wird praktisch automatisch gesäubert.

In der Menüleiste finden Sie nun aber noch folgende zwei Menüpunkte, die Sie normalerweise ausgeschaltet lassen.

verschlüsselte Daten retten

Sollte eine mit dem Saddam-Hussein-Disk-Validator infizierte Diskette eingelegt werden, dann wird dies automatisch erkannt und die Entfernung des Saddam-Hussein-Disk-Validator-Files und die Wiederherstellung der verschlüsselten Daten wird vorgenommen. Es kann nun aber in seltenen Fällen vorkommen, daß das Saddam-Hussein-Disk-Validator-File bereits gelöscht ist, die Diskette also nicht mehr infektiös ist, und daß aber dennoch auf der Diskette noch von dem Virus verschlüsselte Daten stehen. In diesem Fall würde 'Viren suchen+entf.' bei den betroffenen Files einen Fehler melden und könnte die Files nicht zur Überprüfung einlesen, weil diese Files aufgrund der Virus-Verschlüsselung nicht lesbar sind. Wenn Sie also viele Error-Medungen bei 'Viren suchen+entf.' bekommen, dann kann der Grund in vom Saddam-Hussein-Virus verschlüsselten Daten liegen. Wählen Sie in dem Fall den Menüpunkt 'verschlüsselte Daten retten' an. Bei 'Viren suchen+entf.' werden dann zuerst eventuelle Little Sven oder Saddam-Hussein-kodierte Daten wieder restauriert, wodurch Sie dann anschließend wieder voll nutzbar sind und auch überprüft werden können.

Es kann vorkommen, daß Disketten aufgrund mechanischer Beschädigung oder allgemeiner Abnutzung fehlerhaft werden, so daß bei 'Viren suchen+entf.' einige Programme nicht geprüft werden können und Fehler gemeldet wird. Ein Anwählen von 'verschlüsselte Daten retten' hilft hier natürlich auch nicht weiter, weil die Lese-Fehler ja nicht durch die Virus-Verschlüsselung bedingt sind. Man kann bei einer defekten Diskette nur folgendes empfehlen. Benutzen Sie z.B. das Daten-Rettungs-Programm disksalv. Dieses Programm versucht nun noch so viele Daten wie möglich von der defekten Diskette zu lesen. Das Commodore-diskdoctor-Programm sollten Sie möglichst nicht einsetzen, weil es weniger effektiv arbeitet.

Disk-Validator immer umbenennen

Wenn dieser Menüpunkt angewählt ist, dann wird beim Aufruf von 'Viren suchen+entf.' zuvor ein eventuelles Disk-Vaildator-File in Disk-Vaildata0 umbenannt. Normalerweise brauchen Sie diesen Menüpunkt nicht anzuwählen, da im Falle der bisher bekannten Disk-Validator-Viren VIRUS CONTROL automatisch alle erforderlichen Schritte unternimmt. Diese Funktion ist dafür gedacht, um auch alle zukünftigen noch unbekannten Disk-Validator-Viren unschädlich zumachen, denn dadurch daß das Disk-Validator-File in Disk-Validata0 umbenannt wird, kann der Virus nicht mehr aktiviert werden, da hierzu unbedingt der Name Disk-Validator erforderlich ist. Die delete-Funktion des Betriebssystems hingegen wird normalerweise ein Disk-Validator-File nicht löschen oder umbenennen können, da dies der Virus selber verhindern wird. VIRUS CONTROL benutzt spezielle Routinen, um auch bei einem aktiven Disk-Validator-Virus das Virusfile umzubenennen.

Ab Kickstart 2.0 brauchen Sie keine Angst mehr vor Disk-Validatorviren zu haben, da sich nun die entsprechenden Routinen bereits im Betriebssystem-ROM befinden, und daher kein Disk-Validator-File mehr von Diskette geladen wird, wodurch auch kein Disk-Validator-Virus mehr aktiviert werden kann.

1.29 automatisches entpacken von programmen

automatisches Entpacken und Untersuchen von Programmen

Angenommen Sie erhalten immer die Meldung, daß ein gewisser Filevirus oder Linkvirus aktiv sei. Mit Viren suchen+entf. können Sie aber kein entsprechendes Virusfile ausmachen. Vermutlich liegt der Virus dann in einem gepackten File vor. Dieses ist der große Nachteil der Packer, denn nachdem ein Virusfile gepackt wurde, sieht es ganz anders aus und kann nicht mehr als Virusfile identifiziert werden. Normalerweise liegen Virenfiles aber ungepackt vor, denn z.B. ein Linkvirus hängt sich zwar an ein File, packen kann er das File aber anschließend in der Regel nicht, da hierzu ein sehr großer Programmieraufwand nötig wäre. Deshalb kann man bisher Files nur mit Hilfe von speziellen eigenständigen Pack-Programmen wie Imploder oder PowerPacker packen. Es kann nun also ein Virusfile im Nachhinein absichtlich oder unabsichtlich von Hand gepackt werden, wodurch dann die eben geschilderte Problematik des Nicht-Mehr-Erkennens des Virus entsteht. Setzen Sie also Packer und gepackte Programme nicht unkontrolliert ein. Aber auch bei gepackten Programmen brauchen Sie nicht zu resignieren, denn Sie können z.B. gezielt immer nur ein Programm starten und auf eventuelle Warn-Meldungen von VIRUS CONTROL warten.

Ab VIRUS CONTROL 4.0 ist allerdings auch dieses Packer-Problem deutlich entschärft worden, denn wenn Sie 'Entpackversuch ...' anwählen, dann entpackt VIRUS CONTROL automatisch eventuell gepackte Files, und kann dann auch einen eventuellen Virusbefall erkennen. Wenn VIRUS CONTROL in einem gepackten File einen Virus findet, dann kann dieses File nur gelöscht werden. Ein eventueller Linkvirusausbau ist aus dem gepackten File programmtechnisch nicht möglich. Wenn Sie das gepackte Virusfile nicht löschen wollen, sondern möglichst nur den Linkvirus abtrennen wollen, dann müssen Sie zuvor das Virusfile wieder entpacken.

VIRUS CONTROL zeigt Ihnen an, ob ein ausführbares Programm mit einem der folgenden Packern gepackt wurde.

```
Imploder      (auch library, protect usw.)
PowerPacker   (auch library, mastermode usw.)
CrunchMania   (auch library)
DragPack1.0
HQC-Cruncher
Relokit-V1.0
MastercruncherV1.1-TNM
Mastercruncher 3.0
TitanCrunch1.1
TurboSqueezerV6.1,V8.0
XPK-handler
```

Es werden also nur Packer berücksichtigt, die systemkonforme, relocatible Programme erzeugen. Es werden also keine Adress-Cruncher berücksichtigt, da solche Programme extrem multitasking-unfreundlich sind, da sie völlig willkürlich Speicher überschreiben und damit sehr schnell den Amiga zum Absturz bringen. Adress-Cruncher werden daher normalerweise nicht benutzt, da sie nur sehr eingeschränkt lauffähig sind.

In circa 95% der Fälle wird ein Programm mit dem Imploder oder PowerPacker gepackt, denn diese beiden Packer sind weit verbreitet. Sie sind komfortabel zu bedienen und arbeiten sehr effizient, schnell und sauber.

Neben gepackten Programmen werden auch gepackte Dateiarhive entpackt.

1.30 automatisches entpacken von dateiarchiven

automatisches Entpacken und Untersuchen von Dateiarhiven

Wenn man in der Menüleiste 'Entpackversuch ...' anwählt, dann entpackt VIRUS CONTROL automatisch gepackte startbare Dateien, und kann somit auch in gepackten Dateien Viren aufspüren.

Darüber hinaus kann VIRUS CONTROL aber auch in nicht startbaren Dateiarhiven Viren aufspüren. Hierzu wird allerdings Kickstart 2.0 benötigt. Folgende Dateiarhive können bearbeitet werden, wobei zum Entpacken dieser Archive das betreffende Archivierprogramm im C: oder SYS: - Verzeichnis stehen muß. Es werden alle gängigen Archivierprogramme unterstützt. Da beim Entpacken von Archiven die entsprechenden Archivierprogramme gestartet werden, könnte hierbei ein eventueller Linkvirus gestartet werden. VIRUS CONTROL prüft deshalb vor dem Starten die Archivierprogramme auf Linkvirusbefall. Zukünftige Linkviren können hierbei aber leider nicht erkannt werden.

arc, arj, decrunch, lz, lzx, lharc, lha, lhasfx, shrink, zip, zoo

Die in diesen Archiven enthaltene Dateien werden kurzzeitig und vollkommen automatisch in die Ram-Disk entpackt und dort überprüft. Es werden auch verschachtelte Archive aufgelöst, das heißt z.B. 'lha'-Archive in 'lha'-Archiven werden komplett aufgelöst. Es wird von Ihnen keinerlei Mitarbeit verlangt, ein eventueller Virusbefund wird Ihnen natürlich angezeigt. Ferner werden auch selbstextrahierende lhasfx-Dateien automatisch untersucht. Da beim Entpacken in die Ram-Disk mitunter sehr viel Speicher benötigt wird, sollten Sie anstatt RAM: wenn möglich z.B. eine Festplatte mit ausreichend freiem Speicherplatz zum Entpacken der Dateiarchive anwählen. VIRUS CONTROL gibt natürlich diesen Speicherplatz anschließend wieder frei.

Neben diesen Dateiarchivierprogrammen werden auch alle gängigen Diskettenarchivierprogramme unterstützt.

dimp, dms, lhwarp, warp, zoom

Es handelt sich hierbei um Dateien in denen der Inhalt einer ganzen Diskette komprimiert abgespeichert ist. Diese Daten werden normalerweise auf DF0: entpackt, da dieses jedem zur Verfügung steht. Sie können jedoch aus Geschwindigkeitsgründen auch andere Speichermedien anwählen, welche aber im Aufbau einer DD-Diskette entsprechen müssen, wichtig sind

```
Surfaces      = 2
BlocksPerTrack = 11
LowCyl        = 0
HighCyl       = 79
```

Weiterhin muß der Devicename 3 Buchstaben lang sein. Sollten Sie viel Speicher besitzen, empfiehlt es sich z.B. die RAD: einzubinden, oder wenn Sie eine Festplatte besitzen, können Sie auch mit dem FmsDisk-Paket von Fish-Disk 294 eine Diskette auf Festplatte simulieren. Die selten anzutreffenden lhwarp- und warp-Archive können jedoch nur auf DF0: entpacken.

Wenn Sie eine komplette CD auf Viren durchsuchen wollen, dann sollten Sie ein gut ausgestattetes Computersystem besitzen und folgendermaßen vorgehen. Wählen Sie bei 'Entpackversuch ...' zum Entpacken der Dateiarchive eine Festplatte mit noch mindestens 20 MB freiem Speicher, wählen Sie zum Entpacken der Diskarchive z.B. ein passendes fmsDisk- oder RAD:-Device, legen Sie ein beschreibbare und formatierte Diskette in DF0: ein, da manche Diskarchive nur auf DF0: entpackt werden können, wenn eine Protokolldatei erstellt werden soll, dann legen Sie diese auch auf einer Festplatte mit circa 20 MB freiem Speicher an, denn die Länge der Protokolldatei erreicht oftmals über 10 MB, wählen Sie 'automatisch' an, damit die Virensuche nicht unnötigerweise durch Dialogfenster angehalten wird, wählen Sie aus Geschwindigkeitsgründen 'alle Dateien prüfen' nicht an, denn es reicht aus, wenn Sie startbare Dateien und Archive durchsuchen. Bei den restlichen Dateien könnte höchstens noch eine Beschädigung durch einen Virus erkannt werden, ein Virus selber kann aber in solchen nicht startbaren Dateien nicht enthalten sein.

Ich rate aber vom Durchsuchen einer kompletten CD ab, da durch das hundert- bis tausendfache Aufrufen der verschiedenen Archivierprogramme

der RAM-Speicher sehr schnell völlig fragmentiert wird, das heißt Sie haben vielleicht immer noch insgesamt 10 MB freier Speicher, aber es kann durchaus sein, das der größte zusammenhängende Speicherbereich nur noch 100 KB beträgt, wodurch längere Programme nicht überprüft werden können.

1.31 unsichtbare zeichen in filenames anzeigen

unsichtbare Zeichen in Filenamen anzeigen

Weiterhin wird bei 'Viren suchen+entf.' jedes File darauf überprüft, ob unsichtbare Zeichen in seinem Filenamen enthalten sind. Wenn ja, dann wird mittels Dialogfenster darauf hingewiesen. Da unsichtbare Zeichen normalerweise nicht auftreten, deutet Ihr Vorhandensein oftmals auf einen Filevirus hin. VIRUS CONTROL überprüft auch bei jeder eingelegten Disk, ob in der startup-sequence unsichtbare Zeichen enthalten sind, denn auf diese Art könnte sich ein Filevirus unauffällig aufrufen lassen.

1.32 beschädigte .info-files anzeigen

beschädigte .info-files anzeigen

Unter Kickstart 1.2/1.3 können fehlerhafte .info-files bei aktiver Workbench einen Absturz hervorrufen. VIRUS CONTROL überprüft deshalb die .info-files und gibt eine Warnung aus, wenn die .info-files fehlerhaft zu sein scheinen.

1.33 beschädigte programme anzeigen

beschädigte Programme anzeigen

Einige Bootblockviren beschränken sich nicht nur auf das Überschreiben des Bootblocks, sondern überschreiben auch sonstige zufällige Diskettenblöcke. Dadurch werden einige Files auf der Diskette unwiderbringlich zerstört. Diese Programme lassen sich unter Umständen (insbesondere ab Kickstart 2.0) noch starten, aber ein Absturz wird wohl recht bald eintreten, da ja gewisse Programmteile nicht mehr vorhanden sind. Deshalb prüft VIRUS CONTROL bei 'Viren suchen+entf.' auch darauf, ob load-files womöglich durch einen Virus beschädigt wurden.

1.34 alle dateien prüfen

alle Dateien prüfen

Normalerweise werden aus Geschwindigkeitsgründen nur load-files, also startbare Programme, komplett eingeladen und auf Virenbefall und Beschädigungen durch Viren überprüft. Nicht-Load-Files können zwar normalerweise nicht von Viren infiziert werden, eine Beschädigung solcher Dateien durch Viren ist aber denkbar. Wenn Sie diesen Menüpunkt anwählen

werden alle Files, also auch Nicht-Load-Files komplett eingeladen und auf Beschädigungen durch Viren überprüft.

1.35 automatisch

automatisch

Wenn Sie 'automatisch' anwählen, werden alle weniger wichtigen Dialogfenster nach der circa 3 Sek. dauernden Warn-Dialogfenster-Dauer automatisch mit 'NEIN' beantwortet und VIRUS CONTROL fährt mit der Virensuche fort. Dieses Verhalten kann z.B. nützlich sein, wenn Sie über Nacht eine komplette CD überprüfen wollen. Bei einem gefundenen Virus wird allerdings aus Sicherheitsgründen immer eine Eingabe von Ihnen verlangt.

1.36 dateiauswahlfenster wählen und vorbelegen

Dateiauswahlfenster wählen und vorbelegen

Unter Workbench 2.0 können Sie VIRUS CONTROL anweisen, anstatt des eigenen Dateiauswahlfensters, das komfortablere ASL-Dateiauswahlfenster zu benutzen. Sie können auch einen Pfad eingeben, den VIRUS CONTROL im Dateiauswahlfenster-Pfad vorlegen soll.

1.37 unterverzeichnisse

Unterverzeichnisse

Bei 'Dateiveränderungen' und 'Viren suchen +entf.' können Sie einstellen, ob auch eventuelle Unterverzeichnisse bearbeitet werden sollen.

1.38 protokoll

Protokoll

Bei 'Dateiveränderungen' und 'Viren suchen +entf.' können die Bildschirmausgaben in eine Datei mitprotokolliert werden. Vorgeschlagen wird:

s:VCprotokoll-Datum-Uhrzeit

1.39 filenamenpuffer

Filenamenpuffer

VIRUS CONTROL liest zuerst alle File-Namen in einen Puffer ein. Erst danach werden dann anhand dieser Filenamen die entsprechenden Files bearbeitet.

Der Grund für diesen 'Umweg' liegt darin, daß VIRUS CONTROL weniger auf maximale Schnelligkeit, als auf höchstmögliche Datensicherheit ausgelegt ist. Denn wenn während des Durchsuchens eines Verzeichnisses, Schreibzugriffe in dieses Verzeichnis ausgeführt werden, dann kann dies die Diskettenordnung durcheinanderbringen und zum Absturz führen. Dies passiert zwar eher selten, aber z.B. die 'Ram Disk' hatte lange Zeit hiermit Probleme.

1.40 statistik

Statistik

Am Ende von 'Dateiveränderungen' oder 'Viren suchen +entf.' wird eine informative Statistik ausgegeben.

1.41 linkvirus von file abtrennen

Linkvirus von File abtrennen

Hiermit kann man versuchen auch zukünftige Linkviren von Files abzutrennen. Es stehen Ihnen 3 Methoden zur Auswahl. Bei Methode2 und Methode3 müssen Sie zuvor noch die Virusverlängerung eingeben, d.h. die Zahl, um die das infizierte File länger wie das Original-File ist.

Methode 1 entfernt den ersten Code-Hunk.

Methode 2 kürzt am Anfang des ersten Code-Hunks um die Virusverlängerung.

Methode 3 kürzt am Ende des ersten Code-Hunks um die Virusverlängerung.

VIRUS CONTROL versucht nun das Original-File wiederherzustellen. Das frühere Virusfile wird aber sicherheitshalber nicht gelöscht, sondern mit der Endung .LV1, .LV2 oder .LV3 versehen, entsprechend Methode 1, 2 oder 3.

Diese Funktionen sind wenn überhaupt nur für Amigaprofis von Nutzen. Der Normaluser sollte das Entfernen von Linkviren den Antivirusprogrammen überlassen, denn es sind über diese 3 Methoden hinaus noch weitere Linkvirustypen denkbar, die dann also nicht korrekt behandelt werden können. Ein Antivirusprogramm kann aber automatisch eine auf den jeweiligen Linkvirus zugeschnittene Entfernungsmethode anwenden. Auch VIRUS CONTROL trennt bei 'Viren suchen+entf.' völlig automatisch die bisher bekannten Linkviren ab, ohne daß der Anwender irgendwelche fehlerträchtige Eingaben machen müßte.

1.42 fensterleiste

Fensterleiste

Wenn Sie im Arbeitsfenster das 'Leiste'-Symbol anklicken, dann wird das Arbeitsfenster geschlossen und stattdessen eine Fensterleiste geöffnet, in welcher die Speicherverhältnisse und das Datum mit Uhrzeit angezeigt werden. Durch Anklicken des Schließsymbols wird die Fensterleiste

geschlossen und wieder das Arbeitsfenster geöffnet.
Die Fensterleiste kann also zu den Zeiten angezeigt werden, in denen kein Arbeitsfenster geöffnet ist.
Wenn Sie z.B. mit einem Malprogramm arbeiten, welches auch den Speicher benötigt, der durch Schließen der Workbench erhalten werden kann, dann sollten Sie die Fensterleiste nicht benutzen, da dann wegen des offenen Fensters die Workbench nicht geschlossen werden kann. Wenn Sie aber das Arbeitsfenster durch Anklicken des Schließsymbols schließen, kann die Workbench geschlossen werden, obwohl VIRUS CONTROL weiterhin unsichtbar aktiv ist. Hierin liegt ein großer Vorteil von VIRUS CONTROL, denn die meisten anderen Antivirusprogramme belegen permanent wertvollen Speicher aufgrund offener Fenster oder Schirme.

1.43 diskeinlegen->schnelltest

Diskeinlegen->Schnelltest

Wenn 'Diskeinlegen->Schnelltest' eingestellt ist, dann wird beim Einlegen einer Diskette und auch bei den sonstigen Arbeitsfensteraufrufmöglichkeiten die Diskette zusätzlich zu dem immer stattfindenden Test auf Bootblockviren auch einem Schnelltest auf File- und Linkviren usw. unterzogen.

Es werden hierbei wichtige Dateien wie l/disk-validator, libs/icon.library, system/setmap, c/dir, c/loadwb, c/setclock, c/setpatch, c/version, c/mount, die erste Datei auf der Diskette und die erste Datei der startup-sequence auf Viren untersucht. Manche Viren tarnen sich mit ungewöhnlichen bzw. unsichtbaren Dateinamen. VIRUS CONTROL testet daher auch auf das Vorhandensein solcher ungewöhnlichen Dateinamen.

Weiterhin wird überprüft, ob in der startup-sequence der eingelegten Diskette unsichtbare Zeichen enthalten sind, denn auf diese Art könnte sich ein Filevirus verstecken. Wenn VIRUS CONTROL solche unsichtbare Zeichen findet, dann erscheint ein Dialogfenster mit Angabe der entsprechenden Zeilennummer. Sehen Sie sich nun also bitte die entsprechende Zeile Ihrer startup-sequence an. Am besten löschen Sie diese Zeile. Da unsichtbare Zeichen normalerweise nicht auftreten, deutet Ihr Vorhandensein oftmals auf einen Filevirus hin.

'Diskeinlegen->Schnelltest' führt also auch einen Virenschnelltest durch. Wenn Anzeichen für einen Virusbefall gefunden werden, dann wird automatisch 'Viren suchen+entf.' aufgerufen. Hier können Sie dann die komplette Diskette ganz genau nach Viren-Files durchsuchen lassen. Hierbei können dann auch die Viren-Files restauriert bzw. entfernt werden.

Die 'Diskeinlegen->Schnelltest'-Funktion ist sehr wirksam programmiert und erkennt zu circa 75 % einen Virusbefall. Man kommt also nicht umhin ab und zu die komplette Diskette (oder Festplatte) mittels des zeitintensiveren 'Viren suchen+entf.' vollständig durchsuchen zu lassen.

1.44 diskeinlegen->komplettest

Diskeinlegen->Komplettest

Anstelle von 'Diskeinlegen->Schnelltest' kann man auch 'Diskeinlegen->Komplettest' anwählen, wodurch nun jede neu eingelegte Diskette komplett mittels 'Viren suchen+entf.' nach Viren durchsucht wird.

Diese Funktion ist sehr nützlich, wenn Sie viele Disketten nacheinander auf Virenbefall durchsuchen wollen, denn Sie müssen nun nicht 'Viren suchen+entf' extra anklicken, denn die Virensuchaktionen laufen nun vollkommen automatisch ab.

1.45 boot-menü

Boot-Menü

VIRUS CONTROL besteht hauptsächlich aus 3 Bestandteilen.

Erstens das VIRUS CONTROL- Arbeitsfenster . Dieses erscheint z.B. beim Einlegen von verdächtigen Disketten und bietet eine riesige Auswahl an wirksamen Virenbekämpfungsmaßnahmen an.

Zweitens überprüft ein eigenständiger Kontrolltask z.B. sekundlich alle wichtigen Systembereiche auf einen eventuellen Virusbefall.

Und drittens das VIRUS CONTROL-Boot-Menü. Dieses verhindert das Booten und somit Aktivieren von Bootblockviren.

Der Amiga bootet nur dann von einer Diskette, wenn die Diskette die DOS-Kennung aufweist und wenn die Bootblock-Checksumme korrekt ist. Ist nun der Bootcode der Diskette ungleich dem Standard-Bootcode, so handelt es sich um eine 'verdächtige Disk', da es sich ja um einen Bootblockvirus handeln könnte. Es kann sich aber auch um ein harmloses Boot-Intro oder ein selbststartendes Spiel usw. handeln.

Wenn versucht wird, von einer verdächtigen Diskette zu booten, dann erscheint das sogenannte VIRUS CONTROL-BootMenü. Das Boot-Menü hat also den Sinn, das Aktivieren von Bootblockviren durch Booten von einer Bootblockvirus-Disk zu verhindern. Durch Drücken der rechten <ALT>-Taste kann man das Erscheinen des Boot-Menüs verhindern und es wird sofort auch von einer verdächtigen Disk gebootet. Durch Drücken der linken <ALT>-Taste kann man das Boot-Menü auch bei nicht verdächtigen Disketten erzwingen.

Installiere DFX: r.Checksum->Boot Fast-Speicher AN/AUS Laufwerke ändern
Personal-Bootblock f.Checksum->NoBoot Chip-SpeicherFirstAN/AUS VIRUS CONTROL-ENDE

SOFT-BOOT

ASCII-Anzeige des Bootblock

DISK-BOOT

Ein Anklicken von 'Fast-Speicher' schaltet ein eventuelles Fast-Speicher ↔
abwechselnd
resetfest AN bzw. AUS.

Wenn 'Chip-SpeicherFirst' auf AN geklickt wird, dann wird resetfest, wenn kein spezieller Speicher angefordert wird, Chip-Speicher an Stelle von eventuellem Fast-Speicher verwandt (12.19).

'Laufwerke ändern' erlaubt es softwaremäßig Laufwerke zu vertauschen.

'VIRUS CONTROL-ENDE' beendet VIRUS CONTROL.

Es erfolgt eine ASCII-Anzeige des Bootblock im Boot-Laufwerk.
Ein identifizierter Bootblockvirus wird namentlich angezeigt.

Mit 'Installiere DFX:' kann man diesen Bootblock mit dem
Normal-Bootblock (=Install) überschreiben.

Mit 'Personal BB' kann man einen speziellen Bootblock schreiben.

'f.Checksum->NoBoot' macht eine Boot-Diskette nicht bootfähig.
'r.Checksum-> Boot' macht eine Diskette bootfähig.

Mit 'SOFT-BOOT' erfolgt eine völlig ungefährliche Boot-Simulation!
Es wird also an Stelle des Disketten-Bootblock eine ungefährliche
Boot-Routine ausgeführt.

Durch Anklicken von 'DISK-BOOT' erscheint ein weiteres Dialogfenster,
das auf die Gefahr des Aktivierens von Bootblockviren durch das echte
Disk-Boot hinweist.

Einige wenige Boot-Intros oder Spiele gehen starr von gewissen
Custom-Chip-Zuständen und oder Speicheranordnungen nach einem Reset
aus und laufen daher nicht mit Disk-Boot bzw. mit gewissen Grafikfehlern.
Wenn man durch Drücken von R-ALT das Erscheinen des Boot-Menüs
verhindert, dann laufen meist auch diese seltenen Programme.

Es sind in dem Boot-Menü keine Möglichkeiten vorhanden, Bootblöcke
z.B. als File abzuspeichern usw. Der Grund ist der, daß erst nach dem
Bootvorgang die dos.library eingerichtet ist. Zum Zeitpunkt des Bootens
ist also ein Zugriff auf Files noch nicht möglich. Der Hauptzweck des
Boot-Menüs liegt also in der Verhinderung einer Bootblockvirusaktivierung.
Für alle anderen Aufgaben gibt es ja das Arbeitsfenster, welches nach dem
Booten des Rechners verfügbar ist. Man kann das Boot-Menü auch als eine
Untermenge des Arbeitsfensters betrachten.

1.46 disk-boot verhindern

Disk-Boot verhindern

Bei manchen Amigas kann man auch mit Hilfe von z.B. 'noclick' oder
'tracksalve' das interne Laufwerk nicht davon abhalten, im Leerzustand
permanent störend zu klicken. Man kann nun das Klicken durch Einlegen einer
Diskette abstellen. Wenn man nun aber einen Reset ausführt, dann muß man
die eingelegte Diskette entfernen, damit man schnell von der Festplatte
booten kann. Später muß man wieder die Diskette einlegen, um das Klicken
abzustellen. Dieses lästige Disketten-Einlegen und Entfernen kann man sich
nun sparen, wenn man während des Reset/Bootvorgangs die rechte Maustaste
gedrückt hält. Es wird dann die eingelegte Boot-Diskette ignoriert. Es wird
also trotz eingelegter Boot-Diskette von der Festplatte gebootet. Die
Abfrage der rechten Maustaste kann in 'Tastaturbelegung' deaktiviert werden.

1.47 beenden

VIRUS CONTROL beenden

Um VIRUS CONTROL zu beenden und vollständig aus dem Speicher zu entfernen, stehen Ihnen folgende Möglichkeiten zur Verfügung:

Im Arbeitsfenster 'beenden ...' anwählen oder 'Abbruch' anklicken

commodity 'Entfernen' anklicken

Im VIRUS CONTROL-Boot-Menü 'VIRUS CONTROL Ende' anklicken

Tastenkombination <CTRL> + <L-AMIGA> + <ENTER> oder <RETURN>

während Reset linke Maustaste ohne rechte Maustaste drücken

Durch Drücken der linken Maustaste kann man während des Resets VIRUS CONTROL entfernen. Dieses Verhalten kann man abschalten, da auch z.B. einige Festplattentreiber oder Turbokarten die linke Maustaste benutzen. Ab Kickstart 2.0 muß man die linke + rechte Maustaste drücken, um in das sogenannte Boot-Menü zu gelangen. Hierbei würde dann auch VIRUS CONTROL unabsichtlich entfernt werden. Deshalb betrachtet VIRUS CONTROL die linke Maustaste als nicht gedrückt, wenn gleichzeitig die rechte Maustaste gedrückt wird.

Die Tastenkombination und die Maustastenabfrage kann nach Belieben in Tastaturbelegung eingestellt werden.

1.48 laufwerke ändern

Laufwerke ändern

Es erscheint ein Fenster, in dem Sie komfortabel durch Anklicken mit der Maus Ihre Laufwerke beliebig softwaremäßig vertauschen und oder abschalten können.

Sie müssen jedoch immer DF0: definieren, denn das Betriebssystem geht fest von einem existierenden DF0:, was ja auch bei jedem Amiga der Fall ist. Das heißt aber noch lange nicht, daß auch von DF0: gebootet wird.

Ansonsten sind alle denkbaren Kombinationen erlaubt!

Um diese neue Laufwerk-Einstellung zu aktivieren, müssen Sie einen Reset auslösen, was z.B. durch Anklicken des Reset-Symbols möglich ist.

Der Amiga fährt nun mit der neuen Laufwerk-Einstellung hoch.

Hierfür wird ein nur 288 Bytes verbrauchender 'Handler' resetfest über den COLD- und COOL-Vektor installiert. Das eigentliche VIRUS CONTROL-Programm ist nun aber nicht mehr vorhanden und kann auch nicht zusammen mit dem Laufwerk-Handler betrieben werden. Diese softwaremäßige Vertauschung oder Ausschaltung der Laufwerke sollte also nur in Sonderfällen benutzt werden, denn es handelt sich hierbei um eine zwangsläufig unsaubere Programmierung, die auch nicht immer funktionieren kann. So ignorieren viele Kopierprogramme die Laufwerkvertauschung und oder Abschaltung. Da sich diese Programme nicht der üblichen Betriebssystemfunktionen bedienen, sondern direkt auf die Hardware zugreifen, wird hier natürlich eine softwaremäßige Abschaltung bzw. Vertauschung wirkungslos.

Wenn Sie auf zuverlässige Art Laufwerke vertauschen wollen, dann sollten Sie sich einen der zahlreichen und sehr preiswerten

Hardware-Selectoren einbauen und sich nicht mit Software-Lösungen herumplagen. Ferner unterstützt auch Kickstart 2.0 offiziell das Booten von externen Laufwerken.

Der Laufwerk-Handler wird zwar durch das VIRUS CONTROL-Programm gestartet, nach dem folgenden Reset aber, ist nur noch der Laufwerk-Handler aktiv. Es kann also das eigentliche VIRUS CONTROL-Anti-Virus-Programm und der Laufwerk-Handler nicht zusammenbetrieben werden. Wenn Sie dennoch versuchen VIRUS CONTROL erneut zu starten, erhalten Sie folgende Meldung:

Die neue Laufwerkszuordnung ist resetfest,
erkennbar an einem gelben Resetfarbsignal,
entfernbar durch Reset + linke Maustaste!

Sie können den Laufwerk-Handler also durch einen Reset + linke Maustaste entfernen. Danach läßt sich VIRUS CONTROL wieder installieren.

1.49 speichermedien

Speichermedien, RigidDiskBlock-Verwaltung

Durch Anklicken des Symbols 'Speichermedien' wird ein Fenster geöffnet, in welchem alle Speichermedien als anklickbare Symbole angezeigt werden. Durch Anklicken eines Speichermediums werden dessen Partitionsdaten angezeigt. Diese Partitionsdaten stehen normalerweise auf den untersten beiden Zylindern in dem sogenannten RigidDiskBlock. Mit Hilfe dieses RigidDiskBlock ist also ein automatisches Booten möglich. Man spricht auch von Automount bzw. Autoboot. Bei Speichermedien, die nicht über einen RigidDiskBlock mit darin befindlichen Partitionsdaten verfügen, oder wenn ein beschädigter, ungültiger RigidDiskBlock vorliegt, muß das Speichermedium von Hand mit dem 'mount'-Befehl angemeldet werden, wobei der 'mount'-Befehl die Partitionsdaten aus einer Textdatei namens z.B. mountlist entnimmt. Speichern Sie also von Ihrer Festplatte die Partitionsdaten und den RigidDiskBlock auf Diskette ab. Die Partitionsdaten werden im mountlist-Format abgespeichert und im Falle einer bereits vorhandenen mountlist-Datei an deren Ende angehängt. Ein Abspeichern auf Festplatte macht wenig Sinn, weil im Falle einer nicht mehr ansprechbaren Festplatte die Dateien nicht mehr gelesen werden können. Sollte nun ein Bootblockvirus den RigidDiskBlock ab Zylinder 0 auf Ihrer Autobootfestplatte teilweise überschreiben, so kann die Festplatte nicht mehr automatisch starten. Um die Festplatte wieder ansprechen zu können, müssen Sie nun zuerst mittels des 'mount'-Befehls unter Angabe der Datei, in welche Sie die Partitionsdaten abgespeichert haben, die Festplatte wieder anmelden. Die Festplatte ist jetzt wieder benutzbar, aber nur bis zum nächsten Reset, denn aufgrund des nach wie vor ungültigen RigidDiskBlocks ist kein Automount bzw. Autoboot möglich. Schreiben Sie also auch noch die RigidDiskdaten auf die Festplatte, danach sollte Ihre Platte wieder voll funktionsfähig sein.

Ich wiederhole:

Partitionsdaten und RigidDiskBlock auf Diskette sichern, damit im Ernstfall mit Hilfe der Partitionsdaten die Platte wieder angemeldet werden kann und dann die RigidDiskdaten wieder auf Platte geschrieben werden können.

Mit VIRUS CONTROL können Sie also Ihre Festplatte vor Schaden durch Bootblockviren bewahren. Es liegt folgender Sachverhalt vor:
Nur bei Disketten werden die ersten zwei Sektoren als Bootblock benutzt und ausgeführt. Die sogenannten Bootblockviren machen sich diesen Umstand zunutze. Sie kopieren sich auf die ersten beiden Sektoren der Diskette. Beim Booten von der Diskette wird dann der Virus automatisch aktiviert und kann nun weitere Disketten infizieren. Bei den Festplatten gibt es zwar auch je Partition einen Bootblock, dieser wird aber nicht als solcher benutzt, das heißt, er wird nie ausgeführt. Angenommen Sie haben Ihre Festplatte in 3 Partitionen aufgeteilt, dann haben Sie 3 Bootblöcke, also zu Beginn jeder Partition einen Bootblock. Die untersten zwei Zylindern sind bei Auto-Boot-Festplatten meistens nicht für Partitionen nutzbar, da auf diesen untersten zwei Zylindern oftmals Verwaltungsdaten der Festplatte abgelegt werden. So steht hier oftmals eine eventuelle Blockfehlerliste, der Festplattentreiber, die Partitionsdaten der Festplatte usw. In der Regel werden diese Daten gemäß dem von Commodore empfohlenen Rigid-Disk-Standard abgelegt. Dies ist aber nicht immer der Fall, denn der Rigid-Disk-Standard ist lediglich eine Empfehlung, letztendlich kann also jeder Festplatten-Treiber die Festplatte unterschiedlich verwalten. Bei der Diskette hingegen ist durch das Betriebssystem ganz genau festgelegt, wie die Diskette verwaltet wird. Es ist also 100% sicher, daß bei einer Diskette auf Zylinder 0 in Sektor 0+1 immer ein richtiger Bootblock steht, der auch ausgeführt wird. Bootblockviren dürften also nur Sektor 0+1 auf einer Diskette überschreiben, da nur hier ein ausführbarer Bootblock steht. Das Überschreiben von Sektor 0+1 auf einer Festplatte hingegen ist unsinnig, da hier meist gar kein Bootblock steht. Und selbst wenn er hier stünde (z.B. weil bei einer Nicht-Auto-Boot-Festplatte die Partitionen meist schon bei Zylinder 0 beginnen können), dann macht ein Überschreiben dieses Bootblock dennoch keinen Sinn, da ja dieser Bootblock nie ausgeführt wird. Insbesondere bei AutoBoot-Festplatten sind auf den untersten zwei Zylindern und somit auch auf Sektor 0+1 Verwaltungsdaten usw. abgelegt, die ein Auto-Mount und Auto-Boot ermöglichen. Ein Überschreiben dieser wichtigen Daten bewirkt, daß die Festplatte nicht mehr erkannt wird. Es erscheint z.B. folgendes Error-Dialogfenster:

Please insert volume DH0: in any drive.

Circa 1/3 aller Bootblockviren können aufgrund nachlässiger Programmierung eine Festplatte unbrauchbar machen, z.B. ALIEN NEW BEAT, BlackFlash, CODER, DASA(ByteWarrior), Gadaffi, GREMLIN, GX.TEAM, Kauki, Termigator, TURK usw. Diese Viren verbiegen nämlich den DOIO-Vektor der exec.library und prüfen nicht ob sich der DOIO-Zugriff auch wirklich auf das trackdisk.device bezieht. Der Virus versucht sich vielmehr bei jeden 512 oder 1024-Lese oder Schreib-DOIO-Zugriff mit Offset 0 auf das jeweilige Device zu kopieren (=infizieren). Bei einer Festplatte würde dies aber sowieso keinen Sinn machen, da diese ja fest installiert ist und somit nicht verbreitet werden kann. Diese Viren können die Festplatte allerdings nicht physikalisch beschädigen. Der drohende Datenverlust ist aber wohl schlimm genug. Um die Festplatte wieder ansprechbar zu machen, kann man z.B. versuchen, die Platte von Hand zu mounten, um dann den Rigididksblock wieder neu zu schreiben. Auch Programme wie disksalv oder quaterbacktools können unter Umständen weiterhelfen.

Damit Sie auf möglichst bequeme Weise, Ihre Festplatte vor Schaden durch z.B. Bootblockviren bewahren können, bietet Ihnen VIRUS CONTROL folgende Funktionen an.

Partitionsparameter abspeichern
Rigididksblock in Datei abspeichern

erste 1024 Bytes Zylinder 0 zeigen
Datei auf Rigididksblock schreiben

Sollten Sie Ihre Festplatte in mehrere Partitionen aufgeteilt haben, dann ist es egal, welchen Device-Namen sie anwählen. Die Routinen von VIRUS CONTROL sind sehr flexibel programmiert und können alle blockorientierte Speichermedien ansprechen.

Wenn VIRUS CONTROL erkennt, daß auf den in der Regel untersten zwei Festplattenzylindern Rigid-Disk-Daten (=Festplattenverwaltungsdaten) vorhanden sind, dann werden aus Sicherheitsgründen die kompletten Rigid-Disk-Daten verwaltet, wobei durchaus eine Datenmenge von 100 KB und mehr auftreten kann.

Wenn VIRUS CONTROL erkennt, daß das Speichermedium keinen RigidDiskBlock besitzt, dann werden nur die untersten 1024 Bytes verwaltet.

Auf z.B. PAR: oder RAM: ist kein Zugriff möglich, da dieses keine blockorientierten Speichermedien sind.

Nun zu den 4 Funktionen im einzelnen:

Partitionsparameter abspeichern

Hiermit können Sie die auch im Fenster angezeigten Partitionsdaten als gültiger mountlist-Eintrag abspeichern.

erste 1024 Bytes Zylinder 0 zeigen

Es wird der Inhalt von Sektor 0+1 angezeigt. Da es aber keinen genormten Inhalt für Sektor 0+1 gibt, ist auch die Aussagekraft dieser Funktion nicht sehr hoch. Oftmals kann man hier jedoch den verwandten Festplattentyp lesen.

Rigiddiskblock in Datei abspeichern

kompletten Rigiddiskblock abspeichern, aus Sicherheitsgründen möglichst auf eine separate Diskette.

Datei auf Rigiddiskblock schreiben

Diese Funktion ist höchst gefährlich für Ihre Festplattendaten. Denn Sie können damit bei unsachgemäßer Vorgehensweise ebenso wie ein Bootblockvirus Ihre Festplattendaten beschädigen. Benutzen Sie diese Möglichkeit also nur im Notfall, wenn Ihre Platte also z.B. nicht mehr erkannt wird und wenn Sie ein aktuelles Rigiddiskfile zur Hand haben. Sehen Sie auch genau in Ihren Festplattentreiberunterlagen nach, ob hier nicht vielleicht ein ähnliches Programm vorhanden ist, das dann vorzuziehen ist, da es genau auf Ihren Treiber abgestimmt ist.

Um die Gefährlichkeit dieser Funktion etwas zu mildern, speichert VIRUS CONTROL zunächst die aktuellen Rigiddiskdaten nach S:VCsector0+1 ab. Erst danach wird das angewählte File ab Zylinder 0 geschrieben. Dann aber kann diese Funktion wahre Wunder bewirken!! Eine totgeglaubte Festplatte ist nach dem nächsten Reset oftmals wieder 100% okay!!

1.50 bootblock-archivierung

Bootblock-Archivierung

Das größte Problem von Antivirusprogrammen ist die mangelnde Aktualität. Oftmals werden neue Bootblockviren nicht erkannt.

Manche Antivirusprogramme versuchen nun mittels sogenannter brain-files auch zukünftige Bootblockviren zu erkennen.

In dem brain-file werden charakteristische Bytefolgen abgespeichert, aufgrund derer ein zukünftiger Virus identifiziert werden kann.

Das VIRUS CONTROL-Bootblock-Archivierungssystem ist ein sehr einfach zu bedienendes und dennoch höchst leistungsfähiges brain-file-system. So können zum Beispiel sogar selbstmodifizierende Viren automatisch archiviert werden. Andererseits ist das VIRUS CONTROL-Bootblock-Archivierungssystem aber auch sehr flexibel, so daß man es auch zum Backup von Bootblöcken usw. verwenden kann.

VIRUS CONTROL bietet Ihnen die Möglichkeit an, Bootblöcke auf zweierlei Arten zu archivieren.

1. nach S:NoWarning

Ungefährliche Nicht-Standard-Bootblöcke können durch Anklicken von 'nach S:NoWarning' in dem Verzeichnis S:NoWarning unter dem im Texteingabefeld angegebenen Namen abgespeichert werden.

Sollte der abzuspeichernde Bootblock bereits im S:NoWarning-Verzeichnis vorhanden sein, dann kann man diesen Bootblock nun löschen oder umbenennen.

Wird nun in Zukunft eine Diskette eingelegt, deren Bootblock im S:NoWarning-Verzeichnis abgespeichert ist, dann wird anstatt des Arbeitsfensters lediglich ein kurzes rotes Farb-Signal ausgegeben.

Diese Option empfiehlt sich für Disketten, die zwar einen Nicht-Standard-Bootblock besitzen, aber dennoch harmlos sind.

(z.B Intros, Spiele).

Sie können jedoch das Erscheinen des Arbeitsfensters mit Anzeige des Filenamens erzwingen, indem Sie beim Einlegen der Diskette die linke <ALT>-Taste gedrückt halten. Dies ist manchmal sinnvoll, wenn Sie erfahren möchten, wieso diese Diskette als ungefährlich gewertet wird. Wenn Sie also den Archivierungs bzw. Filenamens des Bootblocks erfahren möchten, unter dem er in S:NoWarning abgespeichert ist. Wenn das Arbeitsfenster allerdings bereits offen ist, dann brauchen Sie natürlich die linke <ALT>-Taste nicht zu drücken.

Aus Sicherheitsgründen wird auf 100%-Gleichheit mit den Files in S:NoWarning geprüft, da harmlose Nicht-Standard-Bootblöcke in der Regel immer gleich sind. Dennoch kann man VIRUS CONTROL veranlassen, nur einen bestimmten Bereich des Bootblock zu vergleichen. Dies geschieht mit Hilfe von 'Auto-Archiv'. 'Auto-Archiv' wird weiter unten erklärt. Dies ist zum Beispiel bei manchen Boot-Intro-Generator-Programmen sinnvoll, da sich hier die erstellten Bootblöcke aufgrund frei zu wählender Texte und Farben geringfügig unterscheiden. Wenn also der mittels Auto-Archiv ermittelte Bereich gleich ist, dann wird der Bootblock als gefunden betrachtet und es wird kein Arbeitsfenster ausgegeben.

2. nach S:Virusname

Durch Anklicken von 'nach S:Virusname' wird der Bootblock in dem Verzeichnis S:VirusName unter dem im Texteingabefeld angegebenen Namen abgespeichert.

Sollte der abzuspeichernde Bootblock bereits im S:VirusName-Verzeichnis vorhanden sein, dann kann man diesen Bootblock nun löschen oder umbenennen.

Wird nun in Zukunft eine Diskette eingelegt, deren Bootblock im S:VirusName-Verzeichnis abgespeichert ist, dann wird im Arbeitsfenster zusätzlich der Name angezeigt, unter dem der Bootblock archiviert wurde.

Mit dieser Möglichkeit kann man also auch zukünftige Bootblockviren, (oder Nicht-Standard-Bootblöcke) VIRUS CONTROL bekannt machen.

VIRUS CONTROL kann dadurch also nicht veraltern. VIRUS CONTROL speichert die Bootblöcke als 1024 Byte lange Einzelfiles in dem S:VirusName- oder S:NoWarning-Verzeichnis ab. Man kann die in S:NoWarning oder S:Virusname abgespeicherten Bootblockfiles beliebig weiterverarbeiten und auch anderen Antivirusprogrammen zuführen. Auch kann man die Files z.B. zum Speichern oder Wiederherstellen von Bootblöcken benutzen.

Wenn eine verdächtige Disk eingelegt wird, dann wird zuerst das S:NoWarning-Verzeichnis durchsucht. Wird ein 100%-gleicher Bootblock oder ein 100%-gleicher Auto-Archiv-Bereich gefunden, dann wird lediglich ein rotes Farb-Signal ausgegeben und es erscheint kein Arbeitsfenster.

Bleibt die Suche in S:NoWarning erfolglos, dann wird in S:Virusname nach einem zumindest 92%-gleichen Bootblock oder 100%-gleichen Auto-Archiv-Bereich gesucht. Es erscheint nun immer das Arbeitsfenster. Sollte die Suche erfolgreich gewesen sein, dann wird der entsprechende Archivierungs- bzw. Filename zusätzlich im Arbeitsfenster angezeigt.

Hält man beim Disketten-Einlegen <L-ALT> gedrückt, dann wird immer ein Arbeitsfenster ausgegeben. Sollte die Suche in S:NoWarning nach einem 100%-gleichen Bootblock-File oder einem 100%-gleichen Auto-Archiv-Bereich erfolgreich gewesen sein, dann wird dieser Filename im Arbeitsfenster angezeigt. Wenn nicht, dann wird in S:VirusName weitergesucht und ein eventuell gefundenes Bootblock-File (mindestens 92%-gleiches Bootblockfile oder 100%-gleicher Auto-Archiv-Bereich) im Arbeitsfenster angezeigt. Das Drücken von <L-ALT> dient also dazu, daß bei erfolgreicher Suche in S:NoWarning anstatt eines roten Farb-Signals das Arbeitsfenster mit dem entsprechenden Filenamen ausgegeben wird.

Beim Vergleichen des Bootblock der eingelegten Diskette mit den Bootblock-Files im S:NoWarning-Verzeichnis wird ein Bootblock nur dann als gefunden betrachtet, wenn er wirklich 100% gleich ist oder wenn ein 100%-gleicher Auto-Archiv-Bereich vorhanden ist. Beim Vergleichen des Bootblock der eingelegten Diskette mit den Bootblock-Files im S:VirusName-Verzeichnis wird ein Bootblock dann als gefunden betrachtet, wenn er mindestens zu 92% gleich ist oder wenn ein 100%-gleicher Auto-Archiv-Bereich vorhanden ist. Da die meisten Bootblockviren Infektionszähler und auch andere veränderliche Daten beinhalten kann man nicht auf 100%-Gleichheit prüfen.

Ein Durchsuchen der Archiv-Files unterbleibt, wenn man den Menüpunkt 'mit Archiv-BB vrgleichen' abwählt).

Auf die Archiv-Files kann erst dann zugegriffen werden, nachdem die dos.library eingerichtet wurde, zum Zeitpunkt des Bootmenüs können die Archiv-Files also noch nicht ausgewertet werden.

3. Auto-Archiv

Es gibt auch selbstmodifizierende Bootblockviren. Bei diesen Viren übereinstimmen verschiedene Bootblöcke meist nur bis zu 2%. Eine Erkennung des Virus, indem auf 92%-Gleichheit geprüft wird, schlägt also fehl. Ich habe daher folgende Möglichkeit eingebaut: Sie können bestimmen, ob nur ein bestimmter Teil des Bootblock vergleicht werden soll, denn zumindest die Dekodierroutine ist auch bei einem selbstmodifizierenden Virus immer gleich. Normalerweise vergleicht VIRUS CONTROL immer den gesamten Bootblock. Wenn Sie den Bootblock-Namen aber mit z.B. {18-49}beginnen, dann wird

nur der Bereich ab einschließlich Byte 18 bis einschließlich Byte 49 verglichen, wobei man ab 0 zählt (das erste Byte ist also das Byte 0). Wenn dieser Bereich 100% gleich ist, wird das Bootblockfile als gefunden betrachtet. Durch diese Möglichkeit kann VIRUS CONTROL nicht veraltern, da es sich sogar an selbstmodifizierende Viren anpassen läßt. Es ist sinnvoll immer erst ab Byte 12 suchen zu lassen, da dadurch die Checksumme übersprungen wird, denn diese ist in der Regel meist verschieden. Wenn Sie also einen selbstmodifizierenden Bootblockvirus haben, dann können Sie absichtlich eine weitere Diskette infizieren, und dann diese zwei verschiedenen Bootblöcke auf gleiche Datenbereiche untersuchen. Den größten gleichen Bereich können Sie dann in {} am Anfang des Filenamens angeben. VIRUS CONTROL bietet Ihnen mit 'Auto-Archiv' die Automatisierung dieses Vorgehens an. Hierzu gehen sie folgendermaßen vor. Sie klicken 'Auto-Archiv' an. Geben Sie den Namen ein, unter dem der Bootblockvirus archiviert werden soll. Danach erscheint ein Dialogfenster, mit dessen Hilfe Sie das Bootblockfile entweder nach S:NoWarning oder S:VirusName speichern können. Danach erscheint ein Dialogfenster, das Sie auffordert, eine Diskette einzulegen. Nachdem Sie dies getan haben, klicken Sie 'JA' an. Danach erscheint wieder der Dialogfenster und Sie legen die zweite Diskette ein. Sie klicken wieder 'JA' an. VIRUS CONTROL ermittelt nun den größten gleichen Daten-Bereich und modifiziert den eingegebenen Filenamens entsprechend.

Die eben besprochene Möglichkeit, auch selbstmodifizierende Bootblockviren zu archivieren ist leider doch recht gefährlich und aufwendig. Anfänger sollten daher diese Möglichkeit nicht benutzen!!

Noch ein Hinweis für insbesondere Amiga-Anwender mit nur einem Diskettenlaufwerk: Angenommen Sie booten mit Ihrer normalen Workbench-Diskette. Danach legen Sie eine neue Diskette mit einem Nicht-Standard-Bootblock ein. Sie wollen nun diesen Nicht-Standard-Bootblock archivieren. Geben Sie den gewünschten Namen ein. Bevor Sie diese Eingabe beenden, sollten Sie wieder Ihre Workbench-Diskette einlegen, da sich das Archivierungssystem auf S: bezieht, und S: ist durch den Bootvorgang der Workbench-Diskette zugewiesen. Sie können aber mit Hilfe des assign-Befehls S: beliebig neu zuweisen, z.B. auf die 'Ram Disk' oder was natürlich viel bequemer wäre, auf eine Diskette in einem Zweitlaufwerk, oder gar auf eine Festplatte.

Namentliche Identifizierung von Bootblockviren

Man kann also Bootblockviren mittels des Archivierungssystems erkennen. Der Vorteil dieser Methode ist der, daß auch zukünftige Bootblockviren erkannt werden können, indem diese neue Viren einfach archiviert werden. Der Nachteil des Archivierungssystems ist der, daß selbstmodifizierende Bootblockviren etwas schwieriger in der Handhabung sind. Aus diesem Grund erkennt VIRUS CONTROL programmintern alle mir im Moment bekannten (auch selbstmodifizierenden) Viren. Man ist also nicht auf die Archivierungs-Files angewiesen. Nähere Informationen zu den namentlich erkannten Bootblockviren können Sie der Vireninformationsdatei entnehmen.

VIRUS CONTROL erkennt automatisch die folgenden 4 harmlosen Nicht-Standard-Bootblöcke und öffnet kein Bootmenü oder Arbeitsfenster

AmigaPlus, BootGirl, BootIntro, X-Copy

Ich habe mit Absicht nur diese 4 doch recht häufig auftretenden Bootblöcke

programmintern berücksichtigt. Bei allen anderen Nicht-Standard-Bootblöcken wird immer das Bootmenü bzw. Arbeitsfenster geöffnet. Sicher ist sicher. Denn wenn man zuviele Bootblöcke toleriert, dann wächst die Gefahr, daß man versehentlich einen Virus als harmlosen Bootblock identifiziert. Mit dem Archivieren nach S:NoWarning können Sie jedoch jederzeit einen Nicht-Standard-Bootblock als harmlos kennzeichnen.

1.51 bootblock-analyse

Automatische Bootblock-Analyse

VIRUS CONTROL führt beim Diskeinlegen automatisch eine Bootblock-Analyse durch, deren Ergebnis in der schwarzen Tafel des Arbeitsfensters angezeigt wird.

Fall 1

Es kann ein bestimmter Virus identifiziert werden. Es wird dann auf der Info-Tafel 'Bootblockvirus!' und der 'Virus-Name' ausgegeben.

Fall 2

Es kann kein bestimmter Virus identifiziert werden, dennoch deuten gewisse virenspezifische Programmstrukturen auf einen möglichen Virus hin. In diesem Fall wird auf der Info-Tafel 'Bootblockvirus?' ausgegeben. Es könnte sich also um einen Bootblockvirus handeln, muß aber nicht.

Fall 3

Es kann kein bestimmter Virus identifiziert werden. Es werden auch keine virenspezifische Programmstrukturen gefunden. In diesem Fall wird auf der Info-Tafel 'unbekannter Bootblock' ausgegeben. Die Wahrscheinlichkeit, daß es sich um einen Bootblockvirus handelt, ist also geringer wie in Fall2.

Eine tiefergehende Analyse erhalten Sie, wenn Sie 'Zeige DFX:' anklicken und in dem dann erscheinenden Bootblockfenster Disass anklicken.

1.52 disassembler, analyser

Disassembler, Analyser

Durch Anklicken von 'Disass' werden jeweils 1 KB-Disketten-Daten oder Speicherdaten in disassemblierter Form angezeigt. Bei Bootblockdaten beginnt der eigentliche ausführbare Code erst ab Position 12. Diese Zeile wird optisch hervorgehoben. VIRUS CONTROL ist aber nicht nur ein Disassembler, sondern es erfolgt gleichzeitig auch eine Analyse der Daten, wobei die Analyseergebnisse direkt in der entsprechenden Disassemblierzeile als Kommentar mitangezeigt werden. Hierdurch können Sie also die Vorgehensweise des Virus oftmals einfach wie in einem Buch Zeile für Zeile verfolgen. VIRUS CONTROL ist das einzige Analysesystem, welches direkte Assemblerbefehle kommentieren kann und somit auf vorbildliche Weise Disassembler und Analyser verknüpft.

Für dieses analysierende Disassemblieren ist allerdings etwas Rechenaufwand nötig, aber selbst auf einem 68000-System stehen spätestens nach 5 Sekunden die Ergebnisse fest.

Folgende Analysekommentare sind möglich, wobei die verdächtigen Analyseergebnisse optisch hervorgehoben werden, denn diese deuten verstärkt auf einen Virus hin.

harmlose Analyseergebnisse

die folgenden eher unspezifischen Meldungen findet man z.B. bei Bootintros, aber auch bei Viren und Antivirusprogrammen. Die meisten dieser Analysemeldungen zeigen eine direkte Hardwareprogrammierung an, wodurch solche Programme oftmals nicht auf allen Amigamodellen lauffähig sind.

- Daten (de)kodieren
- Kopierschleife
- linke Maustaste
- rechte Maustaste
- Feuertaste
- schwach programmiert
- OVL-Speicher-Bit
- LED-Soundfilter-Bit
- R/W-Kopf auf Track0?
- Laufwerk bereit?
- R/W-Kopf bewegen
- R/W-Kopf-Richtung
- welche Diskettenseite
- Laufwerk 0,1,2,3
- Laufwerksmotor an/aus
- benutzt Blitter
- copperliste starten
- neue copperliste
- DMA an/aus
- Interrupts an/aus
- Tonausgabe Kanal 0,1,2,3
- neue Grafikdaten
- neue Spritedaten
- verändert Farben
- Drive-Status-Register
- Drive-Select-Register
- Customchips

verdächtige Analyseergebnisse

die folgenden Meldungen deuten auf einen Virus oder ein Antivirusprogramm hin

- löscht COLD
- löscht COOL
- löscht KickMem
- löscht KickTag
- löscht KickChckSum
- prüft COLD
- prüft COOL

```
prüft KickMem
prüft KickTag
prüft KickChckSum
prüft V-Blank-IRQ
prüft Interrupt
von Diskette lesen
direkter RAM-Zugriff
```

die folgenden Meldungen deuten verstärkt auf einen Virus hin, es kann sich aber auch um ein Antivirusprogramm oder ein sonstiges resetfestes Nutzprogramm handeln

```
verändert DOIO
TD-BeginIO verändern
Diskette beschreiben
Disk formatieren
Disk eingelegt?
Disk geschützt?
Diskettenrootspur
prüft ob DOS-Disk
verbiegt Vektor
resetfest mit COLD
resetfest mit COOL
resetfest mit KickMem
resetfest mit KickTag
verändert KickChckSum
verändert V-Blank-IRQ
verändert Interrupt
Supervisorstack
Strahlenposition
```

1.53 systemtest

sekündlicher Systemtest kontrolliert Ihr System auf Veränderungen

VIRUS CONTROL kann also mit Sicherheit das Eindringen von Bootblockviren verhindern, da es verdächtige Disketten anzeigt, und eine gefahrlose Bootssimulation anbietet. Neben diesen Bootblockviren gibt es nun leider aber auch sogenannte Fileviren und Linkviren. Diese können sich z.B. als harmlose CLI-Befehle oder interessante Programme tarnen, die dann beim Aufruf den Virus installieren. Aber auch solche Viren werden von VIRUS CONTROL erkannt, da VIRUS CONTROL extra einen SEPARATEN TASK ZUR SYSTEM-VEKTOREN-KONTROLLE installiert. Dieser Task überprüft sekündlich:

```
ob ein namentlich bekannter Virus aktiv ist
COLD-Vektor, COOL-Vektor
KickMem, KickTag, KickCheckSum-Vektoren
TD-BeginIO-Vektor des trackdisk.device
exec.library-Funktions-Vektoren
intuition.library-Funktions-Vektoren
dos.library-Funktions-Vektoren
die 7 Auto-Interrupt-Vektoren
die 16 exec.library-Interrupt-Vektoren
keyboard-reset-handler-liste
Supervisor-Stack
```

Es erfolgt also eine sehr wirksame und umfassende System-Kontrolle. Wird eine Veränderung festgestellt, dann wird mittels Dialogfenster angezeigt, welcher Vektor verbogen wurde, und auf welche Adresse er verbogen wurde. Den Zahlenwert der Adresse können Sie normalerweise ignorieren, da durch das Multitasking bedingt, für das gleiche Programm je nach Zeitpunkt des Programmstarts durchaus verschiedene Adressen resultieren können.

Diese sekundliche und dennoch unauffällige Kontrolle ist einer der großen Plus-Punkte von VIRUS CONTROL, denn es gibt viele Anti-Virus-Programme, die man in die startup-sequence einbinden kann, und die bei der Abarbeitung der startup-sequence lediglich 1 * auf Viren testen. Womöglich wird nun gemeldet, daß alles okay sei. Es ist nun aber durchaus denkbar, daß man später irgendwann einen Link- oder Filevirus aufruft. VIRUS CONTROL schlägt nun sofort Alarm. Ein Antivirusprogramm, welches aber nur 1 * während der startup-sequence testet, kann diesen Virus erst beim nächsten Boot-Vorgang erkennen und bis dahin kann der Virus nach Lust und Laune walten.

Systemtest-Intervall

Normalerweise wird Ihr Amigasystem sekundlich überprüft. Dadurch erhält man sehr schnell eine Warnmeldung, wenn sich ein Programm an wichtigen Systembereichen zu schaffen macht, andererseits wird dennoch nur sehr wenig Rechenzeit verbraucht (maximal 3%). Ich empfehle also, das Systemtest-Intervall möglichst auf dem Wert 1 zu belassen. Man kann aber VIRUS CONTROL anweisen, eine längere Wartepause zwischen den Systemtests einzulegen. Ein hoher Systemtestwartewert entspricht praktisch einem Abschalten der Systemvektorenkontrolle.

Ein Abschalten der Systemkontrolle erreicht man auch durch Drücken von <L-ALT> und dem <Minus-Zeichen> aus dem rechten Zehnerblock. Mit <L-ALT> und <Enter(Zehnerblock)> schalten Sie die normale sekundliche System-Vektoren-Kontrolle ein. Siehe Tastaturbelegung .

Systemtest-Taskpriorität

Der Kontrolltask wird aus Sicherheitsgründen bevorzugt mit einer Priorität = 1 abgearbeitet. Da wenig Rechenzeit benötigt wird, sollte die Taskpriorität relativ unwichtig sein. Sie können aber dennoch z.B. 0 oder -1 als Taskpriorität eintragen. Mit dem Wert -1 werden immer zuerst fremde Programme vor dem VIRUS CONTROL Task abgearbeitet. Nur im Falle freier Rechenzeit wird der VIRUS CONTROL Task abgearbeitet.

SnoopDos-Task umbenennen

Es gibt Viren, die Dateien nur dann verändern, wenn kein 'SnoopDos' aktiviert ist, denn 'SnoopDos' zeigt Dateizugriffe an, wodurch sich der Virus verraten würde. VIRUS CONTROL verändert nun zufallsgesteuert den letzten Buchstaben, so daß der Virus kein 'SnoopDos'-Task mehr finden kann, und sich dann mit seinen Dateizugriffen verrät. Es gibt mittlerweile allerdings eine neue SnoopDos-3-Version, welche nicht mehr 'SnoopDos' als Tasknamen verwendet, also auch nicht mehr von Viren gefunden wird. Außerdem weist auch VIRUS CONTROL sehr mächtige SnnopDos-artige Funktionen auf, mit denen Sie sehr bequem Dateizugriffe und vieles mehr kontrollieren können. Siehe Benutzung von Betriebssystemfunktionen kontrollieren .

1.54 virusentfern-dialogfenster

Virusentfern-Dialogfenster

Das Erscheinen des Virusentfern-Dialogfensters kann folgende Ursachen haben:

- der sekundlich Ihr System überprüfende Systemtest-Task meldet eine Systemveränderung oder einen aktiven Virus
- Sie klicken im Arbeitsfenster das 'Kill'-Symbol an
- Sie drücken gleichzeitig linke <ALT> und rechte <ALT>-Taste

Nach Anklicken von 'Ende' wird normal weitergearbeitet.

Ein Anklicken von 'eventuellen Virus aus Speicher entfernen' bewirkt das Erscheinen eines weiteren Dialogfensters. Man kann nun auswählen, ob man den eventuell im Speicher aktiven Virus mit oder ohne Reset zu deaktivieren versucht. Wählt man Virusentfernversuch ohne Reset an, dann kann man in der Regel anschließend normal weiterarbeiten, manche Viren können aber mit dieser sanften Virusentfernmethode nicht beseitigt werden. Viel sicherer und empfehlenswerter ist daher der 'Virusentfernversuch mit Reset', da hierbei auch z.B. Tasks oder Interruptserver entfernt werden, die permanent die Vektoren verbiegen. Wenn Sie also nach 'Virusentfernversuch ohne Reset' erneut Warnmeldungen erhalten, dann ist weiterhin womöglich ein Virus aktiv. Sie können diesen Virus dann nur durch 'Virusentfernversuch mit Reset' oder noch sicherer durch Ausschalten des Amigas entfernen.

Bedenken Sie, daß oftmals das pure Auslösen eines Resets einen Virus nicht entfernt. Wenn Sie also z.B. <CTRL> + <L-Amiga> + <R-Amiga> drücken, dann bleibt der Virus aufgrund der meist vorhandenen Resetfestigkeit aktiv. Anders liegt der Fall jedoch bei dem 'Virusentfernversuch mit Reset'. Hierbei werden bei abgeschaltetem Multitasking und abgeschalteten Interrupts die Resetvektoren gelöscht, so daß bei dem nun folgenden softwaremäßig ausgelösten Reset, kein sonstiges resetfestes Programm (=Virusprogramm) außer VIRUS CONTROL mehr eingebunden wird. Der Virus ist also nicht mehr aktiv. Natürlich könnte der Virus nun aber wieder beim Booten von einer verseuchten Diskette aktiviert werden.

Bei dem Virusentfernversuch wird unter anderem der COLD- und COOL-Vektor auf VIRUS CONTROL gesetzt, weiterhin macht sich VIRUS CONTROL auch noch als alleiniges Programm über die Kick-Vektoren resetfest, wodurch zwangsläufig auch die resetfeste Ram-Disk RAD: gelöscht wird.

Abgesehen von der Virusentfernfunktion, bei der aus Sicherheitsgründen alle resetfesten Programm entfernt werden, arbeitet VIRUS CONTROL völlig problemlos mit weiteren Kick-resetfesten Programmen wie RAD: oder turboprint usw. zusammen.

Warndialogfenster-Dauer

Man kann sich das Wegklicken des Warn-Dialogfensters sparen, da nach 3 Sekunden das Dialogfenster automatisch geschlossen wird. Man kann für diese Zeitspanne einen neuen Wert eintragen, 0 bewirkt, daß das Dialogfenster sofort verschwindet. 10 bewirkt, daß der Dialogfenster nach 10 Sekunden verschwindet. 50000 bewirkt, daß das Dialogfenster praktisch nicht mehr automatisch

verschwindet. Man kann sich also ohne Zeitdruck die Vektorveränderung ansehen. Die größtmögliche Sekundenangabe ist 65535, was wohl in jedem Fall ausreichend ist, da dies immerhin 18 Stunden entspricht. Durch Anklicken von 'Ende' ist das Dialogfenster natürlich jederzeit wegklickbar. Man kann aber auch ganz einfach durch Drücken der rechten Maustaste das Verschwinden des Warn-Dialogfensters beliebig lange hinauszögern. Sollte aber dennoch einmal ein Warn-Dialogfenster zu früh verschwinden, dann ist dies nicht weiter schlimm, da Sie sich jederzeit mittels 'Systemübersicht' eine Übersicht über alle veränderten Systemvektoren ausgeben lassen können. Auch durch ein erneutes Starten von VIRUS CONTROL werden die veränderten Systemvektoren wieder angezeigt.

Dialogfenster aktiv öffnen

Damit das Dialogfenster z.B. sofort Tastatureingaben entgegennimmt, können Sie das Dialogfenster auch aktiv öffnen lassen.

Dialogfenster unter Maus öffnen

Normalerweise wird versucht, das Dialogfenster so zu positionieren, daß das 'Nein'-Symbol unter dem Mauszeiger liegt. Andernfalls wird das Dialogfenster an der letzten Bildschirmposition geöffnet.

1.55 systemübersicht

Systemübersicht

Durch Anklicken dieses Symbols kann man sich jederzeit einen umfassenden Überblick über alle wichtigen veränderten System-Vektoren verschaffen, welche zuvor auch schon von dem Systemtest-Task gemeldet wurden.

Oftmals werden System-Vektoren auch durch harmlose Programme verändert:

setpatch	-> Alert, UserState, AllocEntry, Execute, Forbid, Flush, ReleaseSem, ObtSemaphShd, Open, ExAll, AddIntServer, SetIntVector, AddTask, AddLibrary, AddDevice, AddResource, FreePooled,
Iprefs	-> RawDoFmt, CloseWB, DisplayBeep, DateToString, StringToDate, DosGetString, OpenScreenTagList
enforcer	-> Exception1, Exception2, Interrupt1, ColdReboot, Alert
sushi	-> RawPutChar, RawIOinit, RawMayGetChar
munwall	-> Allocmem, FreeMem, Availmem
segtacker	-> Loadseg, NewLoadseg
loadwb	-> SetFensterTitles
exploder	-> Loadseg, NewLoadseg
powerpacker	-> FreeMem
xoper	-> Switch

Da die eben genannten Programme sehr häufig eingesetzt werden, versucht VIRUS CONTROL zu erkennen, ob die Veränderung durch diese harmlosen Programme erfolgt ist. Wenn ja, wird kein Warn-Dialogfenster ausgegeben. Bei neuen Versionen der genannten Programme kann die Erkennung fehlschlagen. Weiterhin verbiegen manche Sound-Player Interrupts, um ein exaktes Timing beim Abspielen der Musik zu erreichen.

VIRUS CONTROL selber muß den COLD und COOL-Vektor und den KickTag-Vektor und

die KickChecksum verändern, um auf allen Amiga-Modellen resetfest zu sein. Durch Abwählen von 'resetfest' kann man allerdings VIRUS CONTROL anweisen, sich nicht resetfest zu machen.

Weiterhin wird der AllocAbs-Vektor verbogen, wodurch VIRUS CONTROL z.B. unter Kickstart 1.2/1.3 das automatische Aktivieren von Disk-Validatorviren verhindern kann. Der PutMsg-Vektor wird von VIRUS CONTROL verbogen, damit eine maximale Systemkontrolle möglich ist, denn über diesen Vektor wird letztendlich die ganze Multitasking-Kommunikation des Betriebssystems abgewickelt.

'Systemüberprüfung' zeigt also im Normalfall zumindest z.B. folgende Übersicht an, wobei weiße Zeilen für Überschriften stehen und schwarze Zeilen für harmlose Zustände. Verdächtige Veränderungen werden durch eine revers-blau-weiße Zeile und dem Zusatz -> ein VIRUS ? kenntlich gemacht.

Reset-Vektoren

```
COLD          -Vektor verändert auf $0001A080 -> VIRUS CONTROL !
COOL          -Vektor verändert auf $0001A000 -> VIRUS CONTROL !
KickTagPtr    -Vektor verändert auf $0001A35A -> VIRUS CONTROL !
***  VIRUS-CONTROL 5.0  ***
KickChecksum-Vektor verändert auf $0001A361 -> VIRUS CONTROL !
trackdisk.device      $07C0C5FC
exec.library          $07C00810
AllocAbs      -Vektor verändert auf $0021A328 -> VIRUS CONTROL !
PutMsg        -Vektor verändert auf $0021A388 -> VIRUS CONTROL !
intuition.library      $07C0E624
dos.library           $07C13304
Auto-Interrupts
Exec-Interrupts
```

Ich zeige also der Vollständigkeit halber auch die auf VIRUS CONTROL selber verbogenen Vektoren an. Ich habe diese Anzeige aus logischen Gründen aufgenommen, da die meisten Antivirusprogramme auf veränderte COLD, COOL und Kick-Vektoren hinweisen.

Oftmals erhält man aber auch z.B. die folgende Übersicht:

Reset-Vektoren

```
COLD          -Vektor verändert auf $0001A080 -> VIRUS CONTROL !
COOL          -Vektor verändert auf $0001A000 -> VIRUS CONTROL !
KickTagPtr    -Vektor verändert auf $0001A35A -> VIRUS CONTROL !
***  VIRUS-CONTROL 5.0  ***
KickChecksum-Vektor verändert auf $0001A361 -> VIRUS CONTROL !
trackdisk.device      $07C0C5FC
exec.library          $07C00810
Alert            -Vektor verändert auf $0021A328 ->          setpatch !
UserState        -Vektor verändert auf $0021A388 ->          setpatch !
AllocEntry       -Vektor verändert auf $0021A3CE ->          setpatch !
intuition.library      $07C0E624
SetWindowTit-Vektor verändert auf $0021F5A4 ->          loadwb !
dos.library           $07C13304
Execute          -Vektor verändert auf $0021A7EC ->          setpatch !
Auto-Interrupts
Exec-Interrupts
```

Die Original-Commodore-Befehle loadwb und setpatch verändern also auch Vektoren. Eine Virus-Gefahr liegt in diesen Fällen natürlich nicht vor.

Den Zahlenwert der Adresse können Sie normalerweise ignorieren, da durch das Multitasking bedingt, für das gleiche Programm je nach Zeitpunkt des Programmstarts durchaus verschiedene Adressen resultieren können.

Zwischen der KickTagPtr-Zeile und KickChecksum-Zeile stehen entsprechend der Anzahl der aktiven Kick-resetfesten-Programme eine entsprechende Anzahl Textzeilen. Normalerweise steht zwischen KickTagPtr und KickChecksum nur eine Zeile, die `*** VIRUS-CONTROL 5.0 ***` lautet, das heißt im Moment ist VIRUS CONTROL das einzige Kick-resetfeste Programm.

Je nach Anzahl weiterer Kick-resetfester Programme kommen weitere Text-Zeilen hinzu. Sie können also anhand der String-Texte erkennen, ob sich ein weiteres Programm resetfest gemacht hat. Sollte VIRUS CONTROL ein Kick-resetfestes Programm finden, welches sich nicht mit einem Text identifizieren will, dann wird folgender Text ausgegeben:

Kein Identifikationsstring, sehr verdächtig -> ein VIRUS ?
denn alle seriösen Kick-resetfeste Programme verwenden einen Identifikationsstring. Wenn ein solcher fehlt, dann deutet dies auf einen Virus hin, der möglichst unentdeckt bleiben will.

Vektorveränderungen, die VIRUS CONTROL programmintern oder mit Hilfe des Lern-Modus als harmlos erkannt hat, werden der Vollständigkeit halber auch angezeigt.

Vektorveränderungen, die VIRUS CONTROL nicht als harmlos erkannt hat, werden optisch durch eine revers-blau-weiße Zeile und dem Zusatz -> ein VIRUS ? hervorgehoben. Dadurch soll verstärkt auf möglicherweise gefährliche Vektorveränderungen hingewiesen werden. Meist wird die Veränderung aber durch ein harmloses Programm und nicht durch ein Virus bedingt sein.

```
*****
*
*   Insbesondere Änderungen an COLD, COOL, DOIO, TD-BeginIO, PutMsg,
*   LoadSeg, KickTag, KickCheckSum deuten auf einen aktiven Virus !!
*
*****
```

Denn erstens ist ein Virus meist resetfest, da sich dadurch seine Verbreitungsgeschwindigkeit stark erhöht. Hierzu muß er COLD, COOL oder die Kick-Vektoren verbiegen. Ein veränderter COLD oder COOL-Vektor ist zu circa 90% durch einen Virus bedingt. Veränderte Kick-Vektoren sind eher durch harmlose resetfeste Programme bedingt, weil dies der offizielle Weg ist, resetfeste Programme zu schreiben. Dennoch machen sich auch viele Viren über die Kick-Vektoren resetfest.

Zweitens will ein Bootblockvirus Disketten infizieren, hierzu kann er z.B. DOIO, PutMsg oder TD-BeginIO verbiegen oder ein Linkvirus will sich an neue Files hängen, hierzu kann er sich z.B. in die LoadSeg-Routine einschleifen. Aber auch harmlose Programme wie z.B. segtracker können den LoadSeg-Vektor verändern.

Sonstige System-Veränderungen sind oft, aber nicht immer, durch harmlose Programme bedingt. So verbiegen z.B. neue Versionen des setpatch-Befehls meist noch mehr Vektoren wie die Vorgängerversionen.

Durch Anklicken des rechten Drittels der betreffenden Zeile können Sie

sich den entsprechenden Speicherbereich ansehen .

erweiterte Systemübersicht

Systemübersicht zeigt in einer Übersicht die System-Veränderungen an, auf die der sekundliche System-Kontroll-Task prüft. Sie können die normale Übersicht in eine erweiterte Übersicht umschalten, indem Sie 'normal/alles' anklicken. Zusätzlich zu der normalen Systemübersicht werden bei der erweiterten Systemübersicht alle ROM-libraries,-devices und -resources vollständig auf Vektorveränderungen untersucht. RAM-libraries,-devices und -resources werden nicht überprüft, da diese von Diskette nachgeladen werden müssen, und deshalb für Viren uninteressant sind, da Sie nicht immer auf Diskette vorhanden sind und auch zum Boot-Zeitpunkt nicht zur Verfügung stehen, also nicht manipuliert werden können. Viren verändern deshalb bisher nur Vektoren von libraries,-devices und -resources, die aus dem ROM aktiviert werden, denn diese sind mit Sicherheit immer vorhanden und können somit bei Bedarf von Viren manipuliert werden. Eine Ausgabe der von Diskette nachgeladenen libraries,-devices und -resources würde Sie also nur unnötig verwirren, da deren Vektoren sowieso immer zwangsläufig auf RAM 'verbogen' sind. Es ist daher auch recht schwierig, RAM-libraries, -devices und -resources auf nachträgliche Veränderungen zu überprüfen.

Bei der erweiterten Systemübersicht werden zusätzlich zu den Auto-Interrupts auch alle eventuell veränderten Exceptions von 2 bis einschließlich 47 ausgegeben. So wird z.B. Exception 8 von Privilegsverletzungshandlern und Exception 2 von dem Enforcer-Programm verändert.

Bei der erweiterten Systemübersicht werden bei den Exec-Interrupts zusätzlich alle Handler und Server-einträge angezeigt. Veränderungen werden zusätzlich optisch kenntlich gemacht.

Bei der erweiterten Systemübersicht werden neben dem Überprüfen der exec.library-Interrupt-Vektoren auch alle vorhandenen Interruptserverlisten auf Nicht-ROM-Einträge überprüft. Solche Nicht-ROM-Einträge weisen aber meist nicht auf einen Virus hin, denn so ist z.B. das Eintragen in die Rasterstrahl-Server-Liste die einzig wirklich saubere Möglichkeit einen Sound-Player oder Scroll-Routine zu realisieren. Das Pack-Programm Imploder läßt auf diese Weise seine Musik abspielen. Es gibt aber auch einige Viren (Incognito,Joshua), die auf diese Weise z.B. 50 mal pro Sekunde ihre Resetfestigkeit sicherstellen.

Bei der erweiterten Systemübersicht werden eventuelle SoftIntListen-Einträge angezeigt. Normalerweise sollten diese Listen leer sein.

Die Veränderungen, die bei der normalen Systemübersicht angezeigt werden, entsprechen den Systembereichen, die der Kontrolltask sekundlich überprüft. Ein Umschalten auf die erweiterte Systemübersicht wirkt sich nicht auf den Kontrolltask aus, denn es macht wenig Sinn, wenn der Systemkontrolltask alle ROM-libraries,-devices,-resources, interruptserver usw. durchtesten würde, denn dadurch würden sehr viele irritierende und unbegründete Warn-Meldungen resultieren. Die erweiterte Systemübersicht-Übersicht ist eher als Informationsquelle für den fortgeschrittenen Anwender gedacht.

Es folgt eine Gegenüberstellung der beiden Systemübersichten

normale Systemübersicht

namentlich bekannter Virus aktiv
Supervisorstack verändert

erweiterte Systemübersicht

namentlich bekannter Virus aktiv
Supervisorstack verändert

Eintrag in keyboard-handlerliste
COLD,COOL,KICK-Vektoren verändert
TD-BeginIO des tackdisk.device
exec,intuition,dos.library
Auto-Interrupts
Exec-Interrupts

Eintrag in keyboard-handlerliste
COLD,COOL,KICK-Vektoren verändert
alle ROM-devices, alle ROM-resources
alle ROM-libraries
Auto-Interrupts + weitere Exceptions
Exec-Interrupts + serverlisteneinträge
SoftIntListen

1.56 lern-modus

Lern-Modus

Wie bereits erwähnt verändern auch einige harmlose Programme Systemvektoren. VIRUS CONTROL erkennt bereits programmintern automatisch die bekanntesten dieser Programme, und öffnet dann kein Warndialogfenster, in der 'Systemübersicht' erhalten Sie auch darüber genaue Angaben. Dennoch wird es immer wieder neue harmlose Programme geben, die Systemvektoren verbiegen. Aus diesem Grunde ist in VIRUS CONTROL ein Lern-Modus eingebaut. Sie klicken 'Systemübersicht' an. In dem nun erscheinenden Systemvektorenübersichtsfenster klicken Sie das linke Drittel der betreffenden Zeile an. Sie werden nun nach dem Namen gefragt, unter dem VIRUS CONTROL die Vektor-Veränderung lernen soll. Anschließend speichert VIRUS CONTROL die Angaben automatisch in seinem .info-File, in S:VCstartup und ENVARC: ab. In Zukunft erscheint nun keine Warn-Meldung mehr, wenn der Vektor von diesem bestimmten Programm verändert wird. Sollte sich aber ein anderes Programm, womöglich ein Virus, an dem Vektor zu schaffen machen, dann erfolgt natürlich auch weiterhin eine Warn-Meldung mittels Warndialogfenster.

Angenommen Sie klicken einen Vektor an, der bereits gelernt wurde, dann fragt Sie VIRUS CONTROL, ob dieser Lern-Eintrag wieder gelöscht werden soll, oder ob Sie den Vektor unter einem neuen Namen lernen wollen.

Wenn Sie ein Kick-resetfestes Programm VIRUS CONTROL bekannt machen wollen, dann klicken Sie die entsprechende Text-Zeile an. In diesem Falle schlägt Ihnen VIRUS CONTROL den Kick-Text als Lern-Namen vor. Auch bei einem zu lernenden keyboard-reset-handler wird Ihnen der entsprechende Identifikationstext vorgeschlagen.

1.57 speicheranzeige

Speicheranzeige

Manchmal ist es aufschlußreich, sich den Speicherbereich anzusehen, auf den ein Vektor zeigt. Klicken Sie hierzu das rechte Drittel der betreffenden Zeile an. Die Zeile auf welche der Vektor zeigt wird optisch hervorgehoben. Vor dieser Zeile werden noch 12 Hexadezimal- bzw. 3 ASCII-Zeilen angezeigt. Dieses ist sinnvoll, weil ein Vektor eher in die Mitte als an den Anfang eines Programmes zeigt. Im Speicher kann man oftmals verräterische Texte erkennen, denn im Speicher liegt ein Virus meist dekodiert und z.T. lesbar vor, auf der Diskette hingegen versteckt sich ein Virus oftmals durch Dekodierung eventueller verräterischer Texte. Auch das Anklicken von

'Disass' kann sehr aufschlußreich sein.

Am Boden des Systemübersichtsfensters befindet sich ein Texteingabefeld, mit dessen Hilfe Sie einen beliebigen Speicherbereich anwählen können. Die Eingabe muß in Hexadezimalschreibweise erfolgen.

Wenn Sie die KickChecksum-Zeile anklicken, erhalten Sie keine Speicheranzeige, da es sich hierbei lediglich um eine Prüfsumme handelt, die nicht auf einen bestimmten Speicherbereich zeigt.

Wenn Sie sich Kick-Resetfeste-Programm im Speicher ansehen wollen, dann sollten Sie nicht KickTagPtr anklicken, sondern den jeweiligen Text, der zwischen KickTagPtr und KickChecksum steht. Nur so gelangen Sie an das jeweilige Kick-resetfeste Programm, wobei Ihnen der Speicherbereich ab rt-init angezeigt wird.

komfortable Fensterbedienung über Schieberegler und Tastatur

Sollten bei 'Zeige DFX:', 'Zeige Datei', 'Vergleichen', 'Systemübersicht' und 'Speicheranzeige' nicht alle Informationen gleichzeitig in dem Fenster sichtbar sind, dann können Sie mit Hilfe des am rechten Fensterrand angebrachten Schiebereglers oder der darunter befindlichen Pfeilsymbole die Bildschirmausgabe beeinflussen. Anstatt die Pfeilsymbole anzuklicken, können Sie auch die <Cursor-Hoch>- bzw. <Cursor-Runter>-Taste betätigen. Durch Drücken von <Home> (Zehnerblock Taste <7>) wird der Anfangsbereich der Daten ausgegeben, durch Drücken von <End> (Zehnerblock Taste <1>) wird der Enbereich angezeigt.

Es werden immer 1 KB Daten für das Fenster bereitgestellt. Sie können aber auch auf Daten außerhalb dieses 1 KB-Bereiches zugreifen, durch Anklicken von 'PgUp' blättern Sie gewissermaßen eine Seite im Speicher zurück und durch Anklicken von 'PgDn' blättern Sie eine Seite im Speicher weiter. Selbiges erreichen Sie durch Drücken von <PgUp> (Zehnerblock Taste <9>) bzw. <PgDn> (Zehnerblock Taste <3>) oder durch Drücken einer <Shift>-Taste und gleichzeitiges Drücken von <Cursor-Hoch> bzw. <Cursor-Runter> oder durch Drücken einer <Shift>-Taste und Anklicken der Pfeilsymbole. Neben diesem seitenweisen Blättern im Speicher ist auch ein zeilenweises Durcharbeiten des Speichers durch Betätigen der <Cursor>-Tasten ohne <Shift> möglich. So wie Sie durch den Speicher wandern können, können Sie auch über eine Diskette oder durch eine Datei wandern.

1.58 benutzung von betriebssystemfunktionen kontrollieren

Benutzung von Betriebssystemfunktionen kontrollieren

Sie können VIRUS CONTROL anweisen, verschiedene Betriebssystemfunktionen zu kontrollieren, es wird Ihnen hierbei auch der Name des aufrufenden Programms, Task oder Prozess, und die Rückkehradresse des Programms angezeigt, das heißt VIRUS CONTROL zeigt Ihnen genau den Speicherbereich an, von wo aus der Funktionsaufruf erfolgte, wodurch also ein Virus genau lokalisiert werden kann. Weiterhin können Sie den Aufruf abbrechen und auch alle Vorgänge in einem Fenster mitprotokollieren lassen. Es wird Ihnen auch der Returncode der Funktionen angezeigt. Die Parameter die an die Funktionen übergeben werden, werden hexadezimal dargestellt, wodurch auch nicht lesbare Zeichen, welche z.B. von manchen Viren verwandt werden,

erkannt werden können. Sie sehen, VIRUS CONTROL bietet teilweise sogar mehr Möglichkeiten und Informationen wie z.B. das populäre SnoopDos. Diese Kontrollmöglichkeiten sind z.B. sehr nützlich, wenn Sie sich über die Gefährlichkeit oder Vorgehensweise eines neuen Programms informieren wollen, auch können Sie hiermit oftmals sehr schnell erkennen, warum ein Programm nicht korrekt arbeitet, z.B. weil es bestimmte weitere Dateien nicht finden kann.

1.59 schreibzugriff auf device melden

Schreibzugriff auf Device melden

In dem Texteingabefeld können Sie z.B. den Device-Namen Ihrer Festplatte mit Doppelpunkt eingeben. Sie können den Device-Namen Ihrer Festplatte z.B. auf folgende Art wermitteln:

1. den CLI-Info-Befehl ausführen:

Sie erhalten z.B. folgende Ausgabe:

Mounted disks:

Unit

VD0:

DH0:

RAM:

DF2:

DF0:

Unter Unit stehen also die Device-Namen

2. den CLI-assign-Befehl ausführen:

Sie erhalten z.B. folgende Ausgabe:

Devices:

CON DF0 DF2 DH0 PAR

PRT RAM RAW SER VD0

am Ende werden also die Device-Namen angezeigt.

Nachdem Sie Ihre Eingabe getätigt haben, erscheint noch eine Sicherheitsabfrage, ob wirklich jeder Schreib-Zugriff auf dieses Speichermedium durch ein oranges Farb-Signal angezeigt werden soll, denn womöglich erlauben es manche Treiber nicht, daß man Sie während Schreibzugriffen stört. Meine bisherigen Tests verliefen alle problemlos. Sie sollten dennoch unbedingt vor Einsatz dieser Funktion ein Backup anfertigen, denn es kann keine Garantie für alle möglichen Speichermedien gegeben werden. Außerdem werden durch einen Absturz während eines Schreibzugriffes meist wichtige Organisationsstrukturen zerstört, wodurch oftmals ein ungültiger Datenträger resultieren könnte. Setzen Sie diese Funktion also nicht unkontrolliert ein, sondern z.B. nur zum gezielten Suchen nach Viren. Sie beenden die Schreibzugriffanzeige, indem Sie das Texteingabefeld wieder löschen.

1.60 startup-sequence-kontrolle

Startup-Sequence-Kontrolle

Ab Kickstart 2.0 kann man sich mittels 'Notify' über Veränderungen an Files informieren lassen. Sollte ab Kickstart 2.0 die s:startup-sequence verändert werden, dann wird Ihnen dieses von VIRUS CONTROL angezeigt.

VIRUS CONTROL überprüft weiterhin die startup-sequence einer neu eingelegten Diskette auf unsichtbare Zeichen und testet das erste File der startup-sequence auf Virenbefall. Siehe Diskeinlegen->Schnelltest .

1.61 bootblock-schreibzugriffkontrolle

Bootblock-Schreibzugriffkontrolle

VIRUS CONTROL zeigt vor jedem Schreibzugriff auf den Bootblock die zu schreibenden Daten an. Dies kann sehr aufschlußreich sein. Durch Drücken der 'j'-Taste oder durch Anklicken des 'JA'-Symbols wird der Bootblock beschrieben. Durch Drücken der 'n'-Taste oder durch Anklicken des 'NEIN'-Symbols wird ein Beschreiben des Bootblock verhindert. Wenn Sie eine Diskette formatieren oder kopieren, müssen Sie natürlich das Beschreiben des Bootblock erlauben. Wenn Sie die <CTRL>-Taste oder <L-ALT> drücken, dann erfolgt keine Bootblock-Schreibzugriff-Kontrolle. Sie können in der Menüleiste die 'BB-Beschreibkontrolle' abschalten.

1.62 zero-location-bug

Zero-Location-Bug

VIRUS CONTROL behebt auch den sogenannten Zero-Location-Bug. Speicherstelle 0 sollte immer den Inhalt 0 aufweisen. Wenn dies nicht der Fall ist, stürzen einige Programme ab. Leider beschreibt nun z.B. der Commadore-Festplatten-Controller A2090 manchmal versehentlich Speicherstelle 0. VIRUS CONTROL überprüft sekundlich Speicherstelle 0. Sollte in Speicherstelle 0 nicht mehr 0 stehen, dann setzt VIRUS CONTROL Speicherstelle 0 wieder auf 0. Sollte ein Programm verbotenerweise Speicherstelle 0 beschreiben, dann können Sie durch Einschalten der Warntonfunktion dieses sichtbar machen, denn dann zeigt VIRUS CONTROL diesen verbotenen Schreibzugriff mit einem roten Farbsignal an.

1.63 move sr,<ea> handler

move sr,<ea> - Privilegs-Verletzungs-Handler

Es gibt einige wenige Programme, die den move sr,<ea> Befehl benutzen. Dieser Befehl ist aber ab 68010 aufwärts privilegiert. Da der move sr,<ea> Befehl problemlos auf dem 68000 Prozessor verwandt werden konnte, gibt es Programme, die diesen Befehl benützen.

Wenn man nun ein solches Programm auf einem 68010,20,30-Prozessor einsetzt, führt dies zu einem 'TaskHeld' (Exception 8 = Privileges-Verletzung). Sie können nun durch Anwählen von 'move sr,<ea> abfangen' in der Menüleiste einen resetfesten Exception-Handler installieren. Hierbei wird auch das VBR-Register ausgewertet. Normalerweise brauchen Sie aber diesen Handler nicht einzustellen, da Programme welche den move sr,<ea> Befehl benutzen, doch sehr selten sind.

1.64 uhr aktivieren

Uhr aktivieren

Jeder Amiga-User, der eine akkugepufferte Echtzeit-Uhr besitzt, kann unter Kickstart 1.3 Probleme mit seiner Uhr bekommen. Es liegt folgender Sachverhalt vor:

Durch das Setzen gewisser Register kann man die Echtzeituhr anhalten. Manche preferences- und setclock-Versionen nehmen nun fälschlicherweise an, daß gar keine Uhr vorhanden sei, da sie kein Ticken der Uhr erkennen können, da sie ja angehalten ist. Man erhält z.B. folgende Meldung: 'Battery Backed up Clock not found'

Zum Setzen dieser Uhr-Register kann es rein zufällig bei einem schweren Amiga-Absturz kommen. Natürlich kann auch ein Virusprogramm mit Absicht diese Register beschreiben. Das war es dann aber auch schon, was ein Virus mit der Uhr anstellen kann. Ein Virus, der in der batteriegepufferten Uhr überlebt, kann es niemals geben, denn die Uhr besteht lediglich aus 16 HalfByte-Registern. Der Uhren-Virus ist also ebenso ein Gerücht, wie die Behauptung, Viren könnten sich auf schreibgeschützte Disketten schreiben. Auch dieses ist zumindest bei einem nicht defekten Amiga-Laufwerk niemals möglich. Arbeiten Sie also möglichst immer mit Schreibschutz!

Die WB1.2-setclock-Versionen und die neueren WB1.3-setclock-Versionen lassen sich nicht durch eine angehaltene Uhr täuschen. Die älteren Workbench1.3-setclock-Versionen behaupten leider hartnäckig, daß keine Uhr vorhanden sei.

Der Ärger mit der angeblich verschwundenen Uhr hat nun ein Ende, denn wenn Sie im Arbeitsfenster 'aktiviere Uhr' anklicken, dann wird eine eventuell angehaltene Uhr wieder aktiviert und kann nun wieder neu gesetzt werden.

Ab Kickstart 2.0 sollten keine Uhrprobleme mehr auftreten, da nun stark verbesserte Uhrrouinen in das Betriebssystem-ROM eingearbeitet wurden.

1.65 palntsc

PAL/NTSC

Kickstart 1.2/1.3

VIRUS CONTROL zeigt in der Menüleiste an, ob Sie einen PAL- oder NTSC-Amiga besitzen. PAL bedeutet, daß Sie normalerweise 256 Linien

benutzen können, die 50 Mal in der Sekunde angezeigt werden. NTSC bedeutet, daß Sie normalerweise 200 Linien benutzen können, die 60 Mal in der Sekunde angezeigt werden. PAL kann also mehr Informationen darstellen, da 56 mehr Zeilen vorhanden sind. Allerdings flackert das PAL-Bild stärker, da dieses Mehr an Zeilen nur mit einer geringeren Bildwiederholfrequenz zu bewältigen ist. Normalerweise werden Sie Ihren Amiga in der PAL-Auflösung betreiben. Leider erkennt das Betriebssystem nicht immer ein vorhandenes PAL-System. Es kommt also manchmal zu einem NTSC-Boot trotz PAL-Amiga. Sie merken dies daran, daß anstatt 256 Zeilen nur 200 Zeilen zur Verfügung stehen. In der Menüleiste sehen Sie welche Grafikauflösung momentan aktiv ist. Sie können nun diese Grafikauflösung auf NTSC oder PAL umschalten. Diese gewählte Grafikauflösung bleibt resetfest eingestellt. Wenn Sie auf einem PAL-Amiga in die NTSC-Auflösung wechseln, dann wird wegen der nun verminderten Zeilenanzahl stellenweise Müll angezeigt, da die ganzen Grafik-Strukturen ja nach wie vor auf den größeren PAL-Bereich ausgelegt sind. Wenn Sie umgekehrt auf einem NTSC-Amiga in die PAL-Auflösung wechseln, dann wird zwar von 60 Hz auf 50 Hz umgeschaltet, was eine Stauchung des Bildschirms zur Folge hat, aber Sie können nun dennoch nicht 256 anstatt 200 Zeilen ausnutzen, da beim Booten für die Workbench lediglich Speicher für 200 Zeilen angelegt wurde. Sie sollten also nach einer Änderung der Grafikauflösung einen Reset auszulösen, damit die Workbench auch optisch korrekt in der neuen Auflösung aufgebaut wird.

Der alte Fat-Agnus, der bis circa 1990 verwandt wurde, konnte nur 512 KB-Chip-Speicher ansprechen konnte. Es existierte eine PAL und einer NTSC-Version. Ein richtiges Umschalten zwischen PAL und NTSC war also nicht möglich. Man kann zwar das Betriebssystem veranlassen, nur 200 Linien zu benutzen, aber der Hauptnachteil von PAL, nämlich die geringe Bildwiederholfrequenz von 50 Hz bleibt erhalten. Die neueren Big-Agni, der auch 1 bzw. sogar 2 MB-Chip-Speicher ansprechen kann, gibt es nur noch in einer Version, da diese neue Big-Agni sowohl in der PAL- als auch NTSC-Auflösung betrieben werden kann. Vorausgesetzt Sie besitzen einen Big-Agnus, dann können Sie nun unter Kickstart 1.3 ganz nach Belieben im NTSC oder PAL-Modus arbeiten. Es empfiehlt sich unter Kickstart 1.3 die PAL-Auflösung anzuwählen, denn dadurch kann es niemals mehr zu versehentlichen NTSC-Boots kommen.

Kickstart 2.0 und höher

Unter Kickstart 1.2/1.3 war lediglich eine Grafikauflösung fest durch das Betriebssystem vorgegeben, nämlich PAL oder NTSC. Ab Kickstart 2.0 werden nun erfreulicherweise alle Grafikauflösungen direkt vom Betriebssystem unterstützt. Benutzen Sie also die entsprechenden Workbench 2.0-Programme um die gewünschte Auflösung einzustellen. Dennoch kann die VIRUS CONTROL-PAL/NTSC-Umschaltung auch ab Kickstart 2.0 sehr hilfreich sein, und zwar aus folgendem Grund: Schirme, die mit OpenScreen() geöffnet werden, benutzen als Auflösung PAL oder NTSC, je nachdem wie die entsprechende Brücke auf der Mutterplatine gesetzt ist. Angenommen Sie haben Ihren Amiga auf NTSC gejumpert, um die angenehmere 60 Hz-Bildwiederholfrequenz zu nutzen. Dies hat allerdings den Nachteil, daß Programme, die von einem PAL-Bildschirm ausgehen, nur eingeschränkt benutzt werden können, da der untere Teil des Schirms abgeschnitten wird. Wenn Sie nun vor Aufruf dieses Programmes mit Hilfe von VIRUS CONTROL auf PAL wechseln, dann wird auch ein über OpenScreen() geöffneter Schirm in PAL geöffnet werden, da VIRUS CONTROL dem Amiga-Betriebssystem sagt, daß auf der Mutterplatine PAL gejumpert wäre. Die VIRUS CONTROL-PAL/NTSC-Umschaltung schaltet also ab Kickstart 2.0 den Default-Monitor um. Aber auch insbesondere der umgekehrte Fall ist denkbar.

Angenommen Sie besitzen wie in Europa üblich einen PAL-Amiga, und haben aber wegen dem ruhigeren Bild eine NTSC-Workbench eingestellt, dann ist es sehr störend wenn weiterhin Schirme in PAL geöffnet werden. Mit VIRUS CONTROL können Sie Ihren Amiga auf NTSC umschalten, hierbei wird auch ein eventueller DOUBLESCAN-Modus erkannt und übernommen.

Ich fasse zusammen:

Unter Kickstart 1.2/1.3 können Sie Ihren Amiga auf PAL oder NTSC einstellen. Nach einem Reset verhält sich Ihr Amiga bis zum Ausschalten 100% wie ein PAL- oder NTSC-Amiga.

Ab Kickstart 2.0 können Sie ebenfalls Ihren Amiga auf PAL oder NTSC einstellen. Oftmals wird diese Einstellung aber durch das Betriebssystem wieder aufgehoben. Sehr nützlich ist es jedoch, daß Sie bestimmen können, ob neu geöffnete Schirme, wenn nichts näheres angegeben ist, in PAL oder NTSC geöffnet werden sollen. Hierbei wird auch ein eventueller DOUBLESCAN-Modus der Workbench für die neuen Schirme übernommen. Dies ist insbesondere für VGA-Multiscan-Monitor-Besitzer sinnvoll, da diese Monitore keine 15 kHz Horizontalfrequenz beherrschen. Durch DOUBLESCAN wird eine 31 kHz-Horizontalfrequenz erzeugt, welche auch durch VGA-Monitore verarbeitet werden kann. Am A3000 konnten meist problemlos VGA-Monitore angeschlossen werden, da der eingebaute Flickerfixer 31 kHz erzeugte. Der A4000 besitzt leider keinen Flickerfixer. Anstelle der normalen PAL oder NTSC-Modi müssen zum Ansteuern eines VGA-Monitors die 29 kHz DOUBLESCAN-Modi benutzt werden.

1.66 chip-speicher bevorzugen

Chip-Speicher bevorzugen

Diese Option ist z.B. bei Intros usw. empfehlenswert, die nicht gezielt Chip-Speicher anfordern. Hierbei wurde nicht bedacht, daß es auch Amigas gibt, die nicht nur Chip-Speicher, sondern auch zusätzlich Fast-Speicher besitzen. Dieser Fast-Speicher wird, wenn nicht gezielt Chip-Speicher angefordert wird, vor Chip-Speicher belegt. Dies hat zur Folge, daß z.B. Grafik- oder Musik-Daten im Fast-Speicher abgelegt werden, wodurch sie von den Custom-Chips nicht mehr erreicht werden können. Es kommt daher zu Grafik- und Soundfehlern. Das Amigabetriebssystem belegt also möglichst Fast-Speicher, wenn der zu belegende Speicher nicht näher beschrieben wird. Dies geschieht aus dem Grund, weil Chip-Speicher kostbarer und langsamer als Fast-Speicher ist (maximal 2MB Chip-Speicher sind möglich). Nachteilig ist dieses Verfahren aber bei dem oben beschriebenen unsauberen Programmierstil.

1.67 fast-speicher-zugriff erlauben

Fast-Speicher-Zugriff erlauben

Es geht um das bereits bei 'Chip-Speicher bevorzugen' beschriebene Problem. Anstatt das Problem mittels 'Chip-Speicher bevorzugt belegen' zu lösen, kann man auch den radikaleren Weg mit 'Fast-Speicher-Zugriff verbieten' gehen.

1.68 arbeitsfensterfarben

Arbeitsfensterfarben

Um den 3D-Effekt der Arbeitsfenstergrafik besser zur Geltung zu bringen, werden unter Kickstart 1.3 vorübergehend graue Workbench-Farben gesetzt. Da ab Kickstart 2.0 bereits vom Betriebssystem ein 3D-Effekt benutzt wird, werden ab Kickstart 2.0 die Workbench-Farben im Normalfall nicht verändert.

Möchten Sie für das VIRUS CONTROL-Fenster spezielle Farben benutzen, dann rufen Sie VIRUS CONTROL mit z.B. -c09be00080fff0000 auf. Sie müssen also direkt hinter -c die 4 Farben in 4-stelliger Hexadezimal-Schreibweise ohne Leerzeichen angeben.

1.69 farbsignale

Farbsignale

In den früheren VIRUS CONTROL-Versionen zeigte VIRUS CONTROL durch ein weiß-blaues Farbsignal an, daß die Diskette überprüft wurde und soweit in Ordnung war. Auch wurde z.B. beim Reset durch dieses weiß-blaue Farbsignal auf ein aktives VIRUS CONTROL hingewiesen. Ab VIRUS CONTROL 5.0 werden normalerweise keine Farbsignale mehr ausgegeben, da diese eher störend wirken und in Bezug auf Fremdgrafikkarten Probleme verursachen könnten. Sie können aber die Ausgabe dieser Farbsignale wieder anschalten, indem Sie den Warnton einschalten. Mit 'Warnton' schalten Sie also Warnton und Farbsignale an oder aus. Während aber der Warnton auf verdächtige Veränderungen aufmerksam macht, zeigen die Farbsignale lediglich ein aktives VIRUS CONTROL an.

dunkelblau-hellblau-weiß: zeigt an, daß VIRUS CONTROL aktiv ist,
tritt z.B. bei Reset und Disk-Einlegen auf,
aber nur wenn 'Warnton' angewählt

rot: 'verdächtiger' Bootblock in S:NoWarning enthalten
oder es hat jemand Speicherstelle 0 beschrieben,
aber nur wenn 'Warnton' angewählt

schwarz: kann Arbeitsfenster wegen Speichermangel nicht öffnen,
aber nur wenn 'Warnton' angewählt

gelb: während Reset wenn Laufwerk-Vertauscher(+Ausschalter) aktiv

orange: Schreibzugriff auf ein Speichergerät anzeigen

1.70 warnton

Warnton

Wenn Sie über die Menüleiste 'Warnton' anwählen, dann wird in den folgenden 4 Fällen zusätzlich zu den üblichen Warnmeldungen über Dialogfenster noch

ein Warnton ausgegeben.

Es wurde eine Diskette mit einem nicht normalen Bootblock eingelegt.

Beim Suchen nach File/Linkviren wurde ein Virus gefunden.

Beim Suchen nach Dateiveränderungen wurde eine Dateiveränderung erkannt.

Der Systemtest-Task hat eine Systemveränderung oder Virusbefall erkannt.

1.71 tastaturbelegung

Tastaturbelegung

VIRUS CONTROL bietet Ihnen die Möglichkeit mit Hilfe von Tastenkombinationen gewisse Funktionen bequem und schnell auszulösen. Nachdem Sie in der Menüleiste den Menüpunkt 'Tastaturbelegung ...' angewählt haben, erscheint ein Fenster in dem Sie durch Anklicken Ihre Tastenkombinationen anwählen können. Sie schließen das Fenster durch Anklicken des Schließsymbols und können dann mit 'Einstellungen speichern' u.a. auch die neue Tastaturbelegung dauerhaft abspeichern. Ich erkläre nun die Standardeinstellungen, die Sie aber nach Belieben abändern können.

L-ALT + 0,1,2,3 (Zehnerblock) -> DF0: DF1: DF2: DF3: Arbeitsfenster aufrufen

Das VIRUS CONTROL-Arbeitsfenster kann man normalerweise durch Gedrückthalten der linken <ALT>-Taste und dann durch Drücken und Loslassen z.B. der 0-Taste aus dem Zehnerblock aufrufen. Anstelle von <L-ALT> können Sie weitere oder andere Sondertasten benutzen. Weiterhin können Sie anwählen ob das Diskettenlaufwerk über die normalen 0,1,2,3-Tasten oder und über die Zehnerblock-0,1,2,3-Tasten ausgewählt werden soll.

CTRL + L-AMIGA + RETURN (o. ENTER) -> VIRUS CONTROL entfernen

Durch gleichzeitiges Gedrückthalten der <CTRL>-Taste und linken AMIGA-Taste und durch Drücken der Return oder Enter-Taste kann man VIRUS CONTROL bequem ohne Rückfrage beenden.

L-ALT + - (Zehnerblock) -> System-Vektoren-Kontrolle AUS.

Durch Gedrückthalten der linken <ALT>-Taste und Drücken der Minustaste aus dem Zehnerblock kann man die System-Vektoren-Kontrolle ausschalten.

L-ALT + Enter -> System-Vektoren-Kontrolle AN.

Durch Gedrückthalten der linken <ALT>-Taste und Drücken der Entertaste kann man die System-Vektoren-Kontrolle wieder anschalten.

CTRL od. R-ALT -> keine Funktionsbenutzungskontrolle
-> keine Bootblock-Schreibzugriffkontrolle

Durch Gedrückthalten der <CTRL>-Taste oder rechten <ALT>-Taste erfolgt sofort ohne Rückfrage z.B. ein Dateiöffnungs- oder Schreibzugriff, trotz eventuell eingestellter Zugriffskontrolle.

L-ALT + R-ALT -> Virusentfern-Dialogfenster

Durch gleichzeitiges Drücken der linken und rechten <ALT>-Tasten wird das Virusentfern-Dialogfenster aufgerufen.

L-ALT + Disk einlegen -> Arbeitsfenster (also auch wenn Disk n.verd.)

Wenn während dem Einlegen einer Diskette die linke <ALT>-Taste niedergedrückt wird, dann erscheint das Arbeitsfenster, auch wenn es sich um eine harmlose Diskette handelt. Dieses Verhalten der linken <ALT>-Taste können Sie nun an- oder ausschalten.

R-ALT + Disk einlegen -> kein Arbeitsfenster (egal ob Disk verd.)

Wenn während dem Einlegen einer Disk die rechte <ALT>-Taste niedergedrückt wird, dann erscheint kein Arbeitsfenster, auch wenn es sich um eine verdächtige Diskette handeln sollte. Dieses Verhalten der rechten <ALT>-Taste können Sie nun an- oder ausschalten.

L-ALT + Reset -> Boot-Menü (also auch wenn Disk n.verd.)

Wenn während des Resets die linke <ALT>-Taste niedergedrückt wird, dann erscheint immer das Boot-Menü, insofern eine Bootdiskette eingelegt ist. Dieses Verhalten der linken <ALT>-Taste können Sie nun an- oder ausschalten.

R-ALT + Reset -> kein Boot-Menü (egal ob Disk verd.)

Wenn während des Resets die rechte <ALT>-Taste niedergedrückt wird, dann erscheint auch bei verdächtigen Bootdisketten kein Boot-Menü. Dieses Verhalten der rechten <ALT>-Taste können Sie nun an- oder ausschalten.

Weiterhin können sie anwählen ob die Maustaten während des Resets abgefragt werden sollen, wenn ja, wird mit der linken Maustaste VIRUS CONTROL beendet und mit der rechten Maustate ein Booten von Diskette verhindert.

Return/Enter -> JA bei Dialogfenster

Hiermit können Sie bestimmen, ob die Return und Enter als JA-Eingabe bei dem Dialogfenster gewertet werden sollen.

1.72 einstellungen speichern

Einstellungen speichern

Nehmen Sie die gewünschten Einstellungen im VIRUS CONTROL-Arbeitsfenster vor und speichern Sie dann diese Einstellungen mit 'Einstellungen speichern'. Die VIRUS CONTROL-Einstellungen werden neben S:VCstartup auch in der zu VIRUS CONTROL gehörenden .info-Datei abgespeichert und auch in ENVARC:VIRUS CONTROL/VIRUS CONTROL.prefs. Egal ob Sie nun VIRUS CONTROL über die Shell oder Workbench starten, es erfolgt immer eine optimale Auswertung der Einstellungen, denn es werden sowohl eventuelle CLI-Parameter als auch eine eventuelle .info-Datei oder eine eventuelle s:VCstartup-Datei gemeinsam ausgewertet.

1.73 befehlsdateimodus

Einsatz von VIRUS CONTROL in Befehlsdateien

Wenn Sie in der Shell oder startup-sequence VIRUS CONTROL gefolgt von einer kompletten Pfadangabe aufrufen, dann wird das angegebene Verzeichnis nach Viren durchsucht, wenn keine Viren gefunden wurden, dann kehrt VIRUS CONTROL mit dem Returncode 0 zurück, sollten aber Viren gefunden worden sein, dann wird das 'Viren suchen+entf.'-Fenster nicht automatisch nach 3 Sekunden geschlossen und nachdem Sie selber das Fenster geschlossen haben wird ein Returncode = 5, also WARN zurückgeliefert.

1.74 commodity

commodity

Wenn VIRUS CONTROL ab Kickstart 2.0 betrieben wird, dann bindet sich VIRUS CONTROL auch als commodity ein, aber nur dann wenn bereits eine andere commodity eingebunden ist. Dadurch wird Speicherplatz eingespart, denn ein Benutzer der keine commodities benutzt, wird auch VIRUS CONTROL nicht als commodity bedienen. Sollte allerdings bereits die 'commodities.library' durch andere commodities geladen sein, dann installiert auch VIRUS CONTROL eine commodity, da dies nun praktisch keinen weiteren Speicherplatz mehr verbraucht. Wenn Sie in der Menüleiste den Menüpunkt commodity abwählen, dann wird nie eine commodity installiert.

Durch das Anklicken von 'Anzeige sichtbar' rufen Sie das Arbeitsfenster auf. Sie können nun z.B. 'Tastaturbelegung ...' anwählen und die VIRUS CONTROL-Tastaturkombinationen verändern. Eine dauerhafte Abspeicherung erreichen Sie dann durch Anwählen des Menüpunktes 'Einstellungen speichern'.

Durch das Anklicken von 'Entfernen' beenden Sie VIRUS CONTROL. Das Anklicken von 'Anzeige verborgen' schließt das Arbeitsfenster, Ein Anklicken von 'Inaktiv' und 'Aktiv' wird ignoriert.

Bedenken Sie bitte, daß VIRUS CONTROL kein reinrassiges commodity-Programm ist, so werden z.B. keine TOOL TYPES unterstützt. Die commodity-Funktionen von VIRUS CONTROL sind lediglich eine Zugabe, denn die vielfältigen Anti-Virus-Möglichkeiten von VIRUS CONTROL können nicht über eine commodity

gesteuert werden, weiterhin ist VIRUS CONTROL auch unter Kick1.2/1.3 lauffähig, wo prinzipiell keine commodities möglich sind.

1.75 appicon, drag and drop

AppIcon, Drag and Drop

Ab Kickstart 2.0 zeigt VIRUS CONTROL ein sogenannten AppIcon auf der Workbench an. Auch ist das Arbeitsfenster von VIRUS CONTROL selber als AppFenster ausgelegt, das heißt Sie können eine Datei oder ein Verzeichnis oder ein Diskettensymbol über dem AppIcon oder über dem Arbeitsfenster fallenlassen. Es wird dann das entsprechende Verzeichnis nach Viren durchsucht.

1.76 multiselect

Multiselect

Ähnlich verhält sich VIRUS CONTROL, wenn Sie den Aufruf über die Workbench vornehmen und hierbei ein weiteres Icon über Multiselect mitangeben. Es wird dann ebenfalls das entsprechende Verzeichnis auf Viren durchsucht.

1.77 amigaguide-hilfe

AmigaGuide-Hilfe

VIRUS CONTROL verfügt über eine integrierte AmigaGuide-Hilfe. Sie rufen die AmigaGuide-Hilfe auf, indem Sie in dem Arbeitsfenster von VIRUS CONTROL das unten rechts befindliche 'Hilfe'-Symbol anklicken oder aber indem Sie in der Menüleiste den Menüpunkt 'Hilfe ...' mit der rechten Maustaste anwählen. Alternativ können Sie auch die <Help>-Taste oder <h>-Taste drücken. Hierbei wertet VIRUS CONTROL die aktuelle Mausposition aus und zeigt direkt den dem Symbol entsprechenden Hilfstext an. Ab Kickstart 2.0 können Sie auch durch Drücken der <Help>-Taste Hilfe zu den Menüpunkten anfordern. Darüberhinaus zeigt VIRUS CONTROL automatisch bei einem gefundenen Virus die entsprechenden Virusinformationen an, wenn in der Menüleiste 'Vireninformationen' angewählt ist. Die Anzeige der Hilfstexte erfolgt immer als völlig eigenständiges Programm, so daß Sie völlig ungestört mit VIRUS CONTROL weiterarbeiten können.

1.78 zukünftige viren

zukünftige Viren

Das größte Problem von Antivirusprogrammen ist die mangelnde Aktualität. Oftmals werden neue Viren nicht erkannt. Bei der Programmierung von VIRUS CONTROL wurde deshalb sehr viel Wert darauf gelegt, daß es auch

bei den noch kommenden Viren möglichst wirksam ist. Folgende Methoden können auch bei zukünftigen Viren Schutz bieten:

- sekündlicher und umfassender Systemtest
- Benutzung von Betriebssystemfunktionen kontrollieren
- Startup-Sequence-Kontrolle
- Schreibzugriff auf Device melden
- Bootblock-Schreibzugriffkontrolle
- Bootblock-Archivierung
- Bootblock- und Speicheranalyse
- ungefährliches Booten von unbekannten Disketten
- Festplatten-Rigiddiskblock-Verwaltung
- unsichtbare Zeichen in startup-sequence melden
- unsichtbare Zeichen in Filenamen melden
- zukünftige Linkviren abtrennen
- Dateiveränderungen erkennen

1.79 warnung

Warnung

Ich bin davon überzeugt, daß VIRUS CONTROL das zumindest momentan leistungsfähigste Antivirusprogramm ist. Ich kann jedoch aufgrund der Komplexität des Programmes nicht für Fehlerfreiheit garantieren.

Ich wiederhole:

VIRUS CONTROL wird mit viel Sorgfalt erstellt. Dennoch kann keine Garantie für Fehlerfreiheit übernommen werden. Für eventuelle Schäden wird nicht gehaftet. VIRUS CONTROL sollte dennoch problemlos mit allen Amigas, Prozessoren, Kickstart 1.2, 1.3, 2.x, 3.x, Speichererweiterungen usw. zusammenarbeiten.

Ich möchte an dieser Stelle ganz deutlich auf folgendes hinweisen:

EIN ABSOLUTER SCHUTZ VOR VIREN IST UNMÖGLICH!!!!

Fertigen Sie möglichst immer Sicherheitskopien an!

Arbeiten Sie wenn möglich immer mit Schreibschutz!

1.80 allgemeine einführung in die virenproblematik

Allgemeine Einführung in die Virenproblematik

Geschichte

Verbreitung

Bootblockviren

Beseitigung

Rigiddiskblock beschädigen

Fileviren

Disk-Validatorviren

Linkviren

Beseitigung

Virusanzeichen

Packerproblematik

Uhrvirus

Lauffähigkeit von Viren
Zukunftsansichten
Autoradresse

1.81 geschichte

Geschichte

Mit der zunehmenden Verbreitung von mittlerweile recht preiswerten Computern wächst auch die Gefahr, die von sogenannten Computerviren ausgeht. Es handelt sich hierbei um Programme, die sich meist unauffällig und automatisch weiterverbreiten, das heißt sich auf z.B. weitere Disketten kopieren. Bevor ein Virus solche Neuinfektionen oder gar Datenzerstörungen vornehmen kann, muß das Virusprogramm zuerst gestartet werden. Die Programmierer von Viren sind also bestrebt, möglichst sichere und dennoch unauffällige Aktivierungsmöglichkeiten für Ihre Virenprogramme zu finden. Entsprechend der verschiedenen Virusstartmethoden kann man vier Hauptvirenarten auf dem Amiga unterscheiden. Die ersten Amiga-Viren waren die Bootblockviren, und zwar der SCA -Virus, der sich mit dem Infizieren der Bootdiskette begnügte, gefolgt vom ByteBandit -Virus, der bereits jede eingelegte Diskette infizierte. Kurz darauf erschien der DASA-ByteWarrior -Virus, welcher zusätzlich, wenn auch wohl unabsichtlich, Festplatten unbrauchbar machte. Anschließend brach die Zeit der Lamer-Bootblockviren an, welche einen neuen Qualitätsstandard für Viren setzten, indem sie die Bootblöcke aus Tarngründen immer verschieden zufallsgesteuert verschlüsselten und außerdem den Antivirusprogrammen einen harmlosen Bootblock vortäuschten. Nach den Bootblockviren kamen dann auch die Fileviren auf, kurz darauf die Linkviren und letztlich noch die Disk-Validatorviren. In letzter Zeit tauchen vermehrt sogenannte Trojanische Pferde auf, das heißt vermeintlich harmlose oder interessante Programme, die allerdings in Wirklichkeit meist unbemerkt einen Bootblock-, File-, Disk-Validator-, oder Linkvirus installieren. Viele der neueren Viren zeichnen sich weniger durch intelligente Programmierung als durch böswillige und plumpe Datenzerstörungen aus. Vermehrt tauchen auch Programme auf, die versuchen, in Mailboxrechner einzudringen, bzw. Sie zu schädigen.

1.82 verbreitung

Verbreitung

Trotz der zunehmenden Anzahl von Viren gibt es dennoch viele Leute, die noch nie Kontakt mit einem Computervirus hatten. Voraussetzung hierfür ist allerdings, daß man z.B. auf Raubkopien verzichtet. Gerade Raubkopien stellen einen idealen Nährboden für Viren dar. Die Kopien gehen durch viele Hände. Hierbei ist die Gefahr sehr groß, daß die Diskette mit einem Virus infiziert wird, denn bei Raubkopierern sind sehr viele Disketten im Umlauf, so daß dann und wann auch ein Virus auftaucht und aktiv wird. Beim Kopieren muß nun der Schreibschutz der Diskette entfernt werden. Aus Bequemlichkeit wird die Diskette oftmals nicht mehr schreibgeschützt und ein Virus kann

sich auf die Diskette kopieren. Hinzu kommt, daß hauptsächlich Spiele raubkopiert werden, da hier oftmals auf eine ausführliche Anleitung verzichtet werden kann. An Spielen wiederum sind oftmals jüngere Amiga-User interessiert, welche aus Unerfahrenheit auch deutliche Hinweise auf einen Virusbefall Ihrer Disketten nicht erkennen.

Auch bei Public-Domain-Disketten sollte man vorsichtig sein, insbesondere wenn man sie von Freunden bezieht. Es empfiehlt sich die Public-Domain-Disketten bei einem bekannten und günstigen PD-Händler zu bestellen.

Da die PD-Händler sehr viele Disketten umsetzen, besteht natürlich die Gefahr, daß sich hierbei Viren einschleichen könnten. In der Regel aber überprüfen die PD-Händler die Disketten sorgfältig auf einen eventuellen Virenbefall. In letzter Zeit setzen sich zunehmend CD-Laufwerke durch, da allerdings auf eine CD an die 600 MB Daten passen, steigt doch deutlich die Gefahr, daß sich auf der CD womöglich Viren befinden.

Beim Kauf von kommerzieller Software ist das Virenrisiko am geringsten, denn es gibt nichts rufschädigeres für eine Firma, als ein Virus auf einem ihrer Produkte.

1.83 bootblockviren

Bootblockviren

Hierbei handelt es sich um die ältesten und verbreitetsten Amiga-Viren. Diese Viren befinden sich auf dem Bootblock der Diskette und werden beim Booten von der Diskette aktiviert, da das Betriebssystem beim Booten von Diskette ein auf dem Bootblock vorhandenes Programm automatisch ausführt. Sollte hiermit also ein Virusprogramm zur Ausführung gebracht werden, dann installiert sich dieser Virus im System, das heißt, er verbiegt gewisse Vektoren wie z.B. DoIO() oder PutMsg() oder TD_BeginIO(), um die Zugriffe auf den Bootblock kontrollieren zu können. Hierdurch kann dann bereits beim Einlegen einer Diskette deren Bootblock mit dem Virus-Programm überschrieben werden, wodurch nun auch diese Diskette infiziert wurde. Der Virus hat sich also weiterverbreitet. In der Regel macht sich ein Virus auch resetfest. Hierzu kommen zwei Möglichkeiten in Betracht, der COLD- oder COOL-Vektor oder die Kick-Vektoren. Dadurch bleibt der Virus auch noch nach einem Reset und Booten von einer noch nicht mit dem Virus infizierten Diskette aktiv und es können weiterhin Disketten infiziert werden. Der Virus kann dann nur durch Ausschalten des Computers sicher deaktiviert werden. Ein Virus muß nicht unbedingt resetfest sein, denn es würde genügen, wenn er die neu eingelegten Disketten infiziert. Dennoch sind die meisten Viren resetfest, da sich dadurch die Verbreitungsgeschwindigkeit und Effektivität des Virus stark erhöht.

Leider existieren mittlerweile Hilfsprogramme wie z.B. bootjob, die es ermöglichen, Bootblöcke in ausführbare Programme zu überführen. Dadurch können dann die Bootblockviren problemlos kopiert und weitergegeben werden. Beim Aufruf eines solchen Programmes wird dann der Bootblockvirus im Speicher installiert. Die eigentliche automatische Verbreitung des Virus erfolgt allerdings weiterhin über den Bootblock, allerdings entsteht bei dem Versuch eine Diskette zu infizieren meist eine Not-A-DOS-Disk, da Programme wie bootjob die ersten 12 Bytes eines Bootblocks (DOS-Kennung) verwerfen, wodurch also ein fehlerhafter Bootblock geschrieben wird.

VIRUS CONTROL erkennt auch in Programme überführte Bootblockviren und erlaubt Ihnen die Löschung dieser Programme.

Siehe Bootblockvirenübersicht .

1.84 beseitigung

Beseitigung der Bootblockviren

Wenn man also einen Virus beseitigen will, dann sollte man den Rechner ausschalten und anschließend von einer garantiert sauberen Diskette booten (z.B. Original-Workbenchdiskette). Dadurch ist sichergestellt, daß kein aktiver Virus beim Virus-Entfernen stört. Danach startet man das Anti-Virusprogramm und überprüft die Disketten oder Festplatte auf Virenbefall. Bootblockviren kann man relativ einfach bekämpfen, da man genau weiß, wo sie sich befinden. Durch Neu-Installieren des Bootblocks, also durch Überschreiben des Bootblockvirus mit dem normalen Bootblock ist der Bootblockvirus entfernt.

Bootblockviren werden von VIRUS CONTROL automatisch beim Einlegen einer Diskette erkannt. Wenn es sich um einen Bootblockvirus handelt, dann meldet VIRUS CONTROL den Virus mit Namen. Mit Installiere DFX: entfernen Sie nun den Bootblockvirus. Wenn 'unbekannter BB' oder 'Bootblockvirus?' angezeigt wird, dann muß es sich nicht immer um einen Bootblockvirus handeln, es könnte auch irgendein harmloser z.B. Grafikeffekt-Bootblock vorliegen. Auch besitzen z.B. manche Spieledisketten einen speziellen Spiellade-Bootblock, der zwar auch verdächtig aussieht, aber dennoch nicht mit 'Installiere BB' vernichtet werden darf, weil dann das Spiel nicht mehr geladen werden kann. Normalerweise empfiehlt sich aber bei fast allen Disketten eine 'Installiere BB'. Im Zweifelsfalle sollten Sie vorher den Bootblock mittels 'BB -> Datei' abspeichern oder noch besser eine Diskettenkopie anfertigen.

1.85 rigiddiskblock beschädigen

Rigiddiskblock beschädigen

Obwohl Bootblockviren recht einfach zu erkennen und zu entfernen sind, darf man sie dennoch nicht unterschätzen, denn manche Bootblockviren versuchen irrtümlicherweise eine Festplatte zu infizieren und überschreiben wichtige Festplattensystemdaten auf Zylinder 0, wodurch nun viele Festplatten nicht mehr erkannt werden. Der DASA-ByteWarrior war der erste Virus, der dieses Verhalten zeigte. Aufgrund schlechter Programmierung prüfen manche Bootblockviren nicht, ob sich der DoIO()-Zugriff auf das trackdisk.device bezieht. Der Virus versucht sich vielmehr bei jedem 512- oder 1024-Byte-Lese oder Schreib-DoIO()-Zugriff mit Offset 0 auf das jeweilige Speichergerät zu kopieren(=infizieren). Der Virus überschreibt also Daten auf Zylinder 0. Auf Zylinder 0 befindet sich aber bei Festplatten normalerweise der sogenannte Rigiddiskblock, dieser Datenbereich beinhaltet alle wichtigen Daten zur Festplattenverwaltung, wie z.B. Errorliste, Partitionsdaten, Mountliste usw. Eine auch nur teilweise Beschädigung dieser Daten führt dazu, daß die Festplatte nicht mehr eingebunden werden kann, also nicht mehr ansprechbar ist. Mittlerweile liegen erfreulicherweise manchen Festplattencontrollern Hilfsprogramme bei, mit denen man den Rigiddiskblock in eine Datei abspeichern kann und bei Bedarf wieder auf die Festplatte zurückschreiben kann. Auch VIRUS CONTROL bietet Ihnen solche Funktionen unter Speichermedien an.

Die Hauptfunktion eines Virus liegt in der Vermehrung, so gesehen ist es Unsinn, einen vermeintlichen Festplattenbootblock zu infizieren, zumal beim

Booten von Festplatten auch die Bootblöcke der jeweiligen Partitionen ignoriert werden. Weiterhin sind Festplatten in der Regel fest installiert und somit nicht dazu geeignet, den Virus auf andere Rechner zu übertragen. Es macht also keinen Sinn, eine Festplatte zu infizieren, es sei denn, der Sinn oder vielmehr Wahnsinn des Virus besteht im primitiven Zerstören von Daten. Leider sind Festplatten in der Regel nicht physikalisch schreibschützbar und somit sehr empfindlich. Ein regelmäßiges Backup ist die somit einzige sichere Schutzmethode!!!!

1.86 fileviren

Fileviren

Im Gegensatz zu den Bootblockviren stehen Fileviren in echten startbaren Programmen. Die Disk-Validatorviren und Linkviren werden aufgrund ihres besonderen Funktionsprinzips als eigene Viruskategorie abgetrennt. Alle restlichen startbaren Virusprogramme faßt man dann unter der Bezeichnung Fileviren zusammen. In der Regel fügt der Filevirus in die Startup-Sequence eine Zeile ein, in der er sich selber aufrufen läßt. Damit diese zusätzliche Zeile weniger auffällt, besteht der Virusfilename oftmals aus unsichtbaren Steuerzeichen. Der Filevirus wird nun also beim Booten und Abarbeiten der Startup-Sequence aktiviert. Ein Filevirus läßt sich also relativ leicht entfernen. Man muß die neue Zeile mit dem Aufruf des Filevirus löschen und sollte auch das Filevirusprogramm selber löschen. Neben diesen 'echten' sich verbreitenden Fileviren, gibt es auch Programme, die z.B. lediglich Daten zerstören oder in Mailboxen einbrechen wollen. Siehe Filevirenübersicht .

1.87 disk-validatorviren

Disk-Validatorviren

Es handelt sich hierbei um eine spezielle Art von Fileviren, welche sich anstelle des Original-Disk-Validators auf die Diskette schreiben und somit automatisch beim Einlegen der Diskette gestartet werden, da sie weiterhin die Diskette als fehlerhaft kennzeichnen. Meist sind die Disk-Validatorfiles 1848 Bytes lang und überschreiben das ebenfalls oftmals 1848 Byte lange Original-Disk-Validatorfile. Der Virus tarnt sich also als Disk-Validator. Wenn man eine fehlerhafte Diskette einlegt, dann versucht das Amiga-Betriebssystem den Fehler zu beheben. Hierzu wird das Disk-Validatorfile benötigt. Dieses steht bei Kickstart 1.2/1.3 noch nicht im ROM, sondern muß nachgeladen werden. Zuerst wird versucht von der eben eingelegten fehlerhaften Diskette selber (:L/Disk-Validator) den Disk-Validator zu laden. Wenn dies fehlschlägt, wird versucht den Disk-Validator von dem logischen Gerät L: zu laden (L:Disk-Validator). Wenn auch dieses fehlschlägt, erscheint ein System-Requester mit diesem Text: 'Unable to load disk validator'

Die Disk-Validatorviren verbreiten sich also über fehlerhafte Disketten, wobei sich der Virus als l/Disk-Validator auf der fehlerhaften Diskette befindet. Eine fehlerhafte Diskette erhält man z.B. wenn während eines Schreibzugriffes ein Absturz oder Reset erfolgt. Die Disk-Validatorviren erzwingen das Nachladen des Disk-Validatorvirusfiles, indem sie die Disk

als fehlerhaft kennzeichnen. Wenn nun eine solche Diskette eingelegt wird, dann wird der Virus durch das Betriebssystem durch Laden und Starten von :L/Disk-Validator aktiviert.

Das große Problem an den Disk-Validator-Viren ist, daß sie sofort beim Einlegen der infizierten Diskette aktiviert werden. Es genügt also, eine infizierte Diskette kurzzeitig in ein Laufwerk einzulegen und wieder zu entfernen. Das Betriebssystem übernimmt hierbei die Aktivierung des Virus. Um einen Bootblock-Virus zu aktivieren, muß von der infizierten Diskette gebootet werden und um eine File- oder Linkvirus zu aktivieren, muß das betreffende File gestartet werden. Bei den Disk-Validatorviren genügt bereits das kurzzeitige Einlegen einer infizierten Diskette zum Aktivieren des Virus. Siehe Disk-Validatorvirenübersicht .

1.88 linkviren

Linkviren

Diese Viren sind die wohl gefährlichsten und auch am schwierigsten zu programmierenden Viren. Ein Linkvirus hängt sich vor ein Programm, wodurch dieses also etwas länger wird. Linkviren sind somit relativ schwer auszumachen, da bis auf die größere Programmlänge keine Veränderungen zu erkennen sind. Auch ist es oftmals schwer, den Linkvirus wieder vom Programm abzutrennen. Ein Linkvirus hängt sich auch manchmal an das erste Programm der Startup-Sequence. Dadurch ist sichergestellt, daß der Linkvirus beim Booten aktiviert wird. Linkviren infizieren manchmal auch libraries, devices und handler, also die Programme in den Verzeichnissen libs, devs, oder l, da diese letztendlich auch ausführbare Programme sind, welche bei Bedarf vom Betriebssystem gestartet werden. Gefährlich sind aber auch Linkviren welche nur beim Aufruf von bereits infizierten Programmen weitere infizieren, sich also nicht resetfest machen und keine sonstigen Vektoren verbiegen, sich also nicht dauerhaft installieren und sich auch sonst nicht verraten. Solche Linkviren könnten lange unentdeckt bleiben. Allerdings ist die Verbreitungsgeschwindigkeit solcher Viren auch geringer. VIRUS CONTROL kann mit seinen universellen Schutzmechanismen wie z.B. Dateiveränderungen und Funktionskontrolle auch solche vielleicht in Zukunft vermehrt auftretende Linkviren erkennen (z.B. RedOctober1.7). Siehe Linkvirenübersicht .

1.89 Beseitigung der Fileviren, Disk-Validatorviren und Linkviren

Beseitigung der Fileviren, Disk-Validatorviren und Linkviren

Bootblockviren erkennt VIRUS CONTROL automatisch beim Einlegen einer infizierten Diskette. Mit Installiere DFX: können Sie dann den Bootblockvirus entfernen.

Im Gegensatz zu den Bootblockviren stehen die restlichen Viren wie die Fileviren, Linkviren und Disk-Validatorviren in echten Dateien. Mit Viren suchen+entf. können Sie Disketten oder Festplatten usw. nach diesen Virusprogrammen durchsuchen.

1.90 virusanzeichen

Virusanzeichen

Nachdem ein Virus gestartet wurde hängt er normalerweise seine Infektionsroutinen in dafür geeignete Betriebssystemvektoren ein. Sehr oft werden hierzu der DoIO()- oder TD_BeginIO()-Vektor verändert.

Weiterhin machen sich Viren auch sehr oft resetfest, das heißt, der Virus ist nur durch Ausschalten des Rechners deaktivierbar, wodurch sich die die Verbreitungsgeschwindigkeit des Virus sehr stark erhöht.

Hierzu werden COLD, COOL oder die Kick-Vektoren verändert.

Ein veränderter COLD oder COOL-Vektor ist zu circa 80% durch einen Virus bedingt. Veränderte Kick-Vektoren sind eher durch harmlose resetfeste Programme bedingt, weil dies der offizielle Weg ist, resetfeste Programme einzubinden. Dennoch machen sich auch viele Viren über die Kick-Vektoren resetfest.

```
*****
*
*   Insbesondere Änderungen an COLD, COOL, DOIO, TD-BEGINIO, PutMsg, *
*   LoadSeg, KickTag, KickChecksum deuten auf einen aktiven Virus !! *
*
*****
```

Oftmals verbiegen Viren noch weitere Vektoren, wie z.B. den Autointerrupt 3 oder Systemvektoren wie Wait(), ExitIntr(), Supervisor() oder AllocMem() usw. Diese Vektoren werden mehrmals in der Sekunde aufgerufen, wodurch also auch der Virus mehrmals pro Sekunde z.B. seine Resetfestigkeit usw. sicherstellen kann.

Viren verraten sich oftmals durch solche Vektorveränderungen, aber auch viele harmlose Programme nehmen Vektorveränderungen vor, so z.B. auch der Original-Commodore-setpatch-Befehl, um einige kleinere Betriebssystemfehler auszubügeln. Also oftmals sind Vektorveränderungen auch durch harmlose Programme bedingt. Mit Hilfe der sekundlichen VIRUS CONTROL- Systemkontrolle können sie bestimmte Vektorveränderungen bestimmten Programmen zuordnen.

1.91 packerproblematik

Packerproblematik

Siehe automatisches Entpacken von Programmen .

1.92 uhrvirus

Uhrvirus

Siehe Uhr aktivieren .

1.93 lauffähigkeit von viren

Lauffähigkeit von Viren auf verschiedenen Amiga-Modellen

Viele Viren, insbesondere die Bootblockviren, sind oftmals sehr unsauber und wenig flexibel programmiert, so daß sie des öfteren nur auf den Standard-Amigas laufen, das heißt auf Amigas ohne Fast-RAM, ohne Turbokarte, ohne Festplatte und oftmals auch nur mit einer bestimmten Kickstartversion wie z.B. Kickstart 1.3. Im folgenden erkläre ich die Hauptgründe, warum viele Viren auf den neueren Amigamodellen oft nicht korrekt funktionieren.

- Festplatte
- Speicherausbau
- höhere Prozessoren
- feste Kickstart-Adressen
- Bootdisk-Bug
- Drivebit-Bug
- Schlußfolgerung

1.94 festplatte

Festplatte

Manche Bootblockviren gehen immer von einem Disketten-Boot aus und stürzen bei einem Festplatten-Autoboot ab, oder noch schlimmer, sie versuchen die Festplatte irrtümlich zu infizieren und überschreiben wichtige Festplattensystemdaten auf Zylinder 0, wodurch nun viele Festplatten nicht mehr erkannt werden. siehe Rigidiskblock beschädigen .

1.95 speicherausbau

Speicherausbau

Einige Viren nehmen z.B. feste Adressen für die execbase oder den Supervisorstack an. Je nach Speicherausbau liegen diese Betriebssystemstrukturen aber an anderen Adressen. Auch werden des öfteren die resetfesten Strukturen im gerade verfügbaren Speicher angelegt, was bei Vorhandensein von Fast-RAM den Effekt hat, daß die im Fast-RAM angelegten Strukturen bei einem Reset unter Kickstart 1.2/1.3 nicht gefunden werden. Das Programm ist dann also nicht resetfest. Ebenso sind fast alle Viren auf Amigas mit 1 MB Chip-RAM unter Kickstart 1.2/1.3 nicht resetfest. Mit z.B. 'setpatch r' kann die Resetfestigkeit manchmal wiederhergestellt werden. Viele Viren machen sich über den COOL-Vektor resetfest. Diese Viren sind unter den frühen Kickstart 2.0 - Versionen (36.xxx) nicht mehr resetfest, da hier nur die Kick-Vektoren ausgewertet werden. Auch kann es z.B. vorkommen, daß der Virus Fast-RAM als Pufferspeicher für den trackdisk.device-Zugriff benutzt, was dann unter Kickstart 1.2/1.3 auch nicht funktionieren kann. Oftmals belegen Bootblockviren willkürlich Speicher ab z.B. \$7f000, in der Annahme, daß hier der sichere Supervisor-Stackbereich sei, was aber nur für 512 KB-Chip-RAM-Amigas gilt. Insbesondere bei 1 MB-Chip-RAM-Amigas kann es bei Ausnutzung des Chip-RAM zum Überschreiben des Virus und somit zum Absturz kommen.

Es gibt neuerdings auch Viren, die annehmen, daß jeder Amiga 2 MB ChipMem hätte.

1.96 höhere prozessoren

höhere Prozessoren

Manche Viren benutzen selbstmodifizierenden Code. Aufgrund des Code-Cache kann diese unsaubere Methode aber fehlschlagen. Manche Viren bauen erst beim Betrieb z.B. gewisse Sprungbefehle auf. Meistens werden solche Codemodifizierungen nur an einigen Stellen des Viruscodes vorgenommen und in der Regel befinden sich diese Codebereiche noch nicht im Code-Cache oder werden noch nicht sofort benutzt, wodurch diese Codemodifizierungen dann meist mit Glück funktionieren. Aber eben nur meist. Manchmal ist dieser selbstmodifizierende Code der Grund für das Nichtfunktionieren von Viren auf höheren Prozessoren.

Manche Viren verschlüsseln den gesamten Virus-Code mit einem Zufallswert, wodurch das Erkennen und Analysieren des Virus erschwert werden soll. Manchmal trifft man z.B. folgenden Fall an: Der Virus beginnt mit der kurzen Dekodierroutine, auf diese folgt direkt der zu dekodierende Virusbereich. Meist wurde auch das Ende der Dekodierroutine bereits mitkodiert. Solche Viren werden z.B. auf einem 68030 meist nicht laufen. Der Grund hierfür ist der größere Prefetch bei den höheren Prozessoren. Auf dem 68000 tritt der Prefetch noch nicht in Erscheinung. Auf dem 68030 jedoch beträgt der Prefetch bereits bis zu drei Worte, das heißt, daß bis zu drei Worte bereits in den Prozessor eingelesen werden, obwohl die Ausführung der vorhergehenden Befehle noch gar nicht abgeschlossen ist. Mit Hilfe dieser Pipeline-Struktur kann die Arbeitsgeschwindigkeit des Prozessors erhöht werden. Durch diesen größeren Prefetch wird nun das Ende der Dekodierroutine zwar im Speicher richtig dekodiert. Im Prozessor selber wird aber der fehlerhafte verschlüsselte Code benutzt, wodurch es zum Absturz kommen muß. So können sich auf Turbo-Amigas z.B. die meisten Lamer und Disk-Validatorviren aufgrund vorzeitigen Absturzes meist nicht installieren. Nach Eingabe des Befehls 'cpu nocache' hingegen gelingt die Virus-Installation. Wenn die Dekodierroutine selbst nicht verschlüsselt ist, sondern nur die auf die Dekodierroutine folgenden Daten, dann wird das Kodieren bzw. Dekodieren auch auf den höheren Prozessoren funktionieren. Aber auch nur meist, da doch in einigen Fällen aufgrund des Code-Cache und Prefetch die Daten im Prozessor nicht den Daten im Speicher entsprechen, wodurch dann also ein Absturz resultieren kann. Auf dem 68000 konnte man noch wild drauflos programmieren, da noch keine Besonderheiten wie Code-Cache oder Prefetch zu berücksichtigen waren. Beim 68000 war also der aktuelle Code, welchen der Prozessor gerade eingelesen hat und nun ausführt, immer identisch mit dem wirklichen Code im Speicher, wodurch also selbstmodifizierender Code und das Kodieren ganzer Datenbereiche problemlos möglich war. Auf den höheren Prozessoren ist dieses Vorgehen verboten, da man aufgrund der nun im Prozessor eingebauten Caches und des größeren Prefetches keine exakten Vorhersagen mehr machen kann.

1.97 feste kickstart-adressen

feste Kickstart-Adressen

Insbesondere einige sehr alte Bootblockviren springen direkt in das Betriebssystem ein. Hierbei werden meist die Kickstart 1.2 Adressen benutzt. Unter Kickstart 1.3 bringt ein solcher Virus den Amiga unweigerlich zum Absturz. Auch der umgekehrte Fall kann beobachtet werden, daß also Viren z.B. feste Adressen des Kickstart 1.3 benutzen, wodurch dann ein solcher Virus nur unter Kickstart 1.3 laufen kann.

Neben der Benutzung absoluter Adressen sind der Bootdisk-Bug und der Drivebit-Bug die Hauptgründe dafür, daß viele insbesondere Bootblockviren nicht korrekt mit Kickstart 2.0 laufen.

1.98 bootdisk-bug

Bootdisk-Bug

Viele ältere Bootblock-Viren begnügten sich mit dem Infizieren der eingelegten Bootdiskette. Der Betriebssystem-DoIO()-Boot-Aufruf wurde anhand der Bedingung `cmpa.l 40(A1),A4` ermittelt, was allerdings nur für Kickstart 1.2/1.3 zutrifft. Unter Kickstart 2.0 hingegen ist diese Bedingung nie erfüllt, denn es ist purer Zufall, daß unter Kickstart 1.2/1.3 während dem Betriebssystem-DoIO()-Boot-Aufruf in `a4` der Inhalt von `40(a1)=IO-Puffer` steht. Auf solche Zufälligkeiten darf man sich nicht verlassen. Dieser Bootdisk-Bug ist der Grund warum z.B. der SCA -Virus ab Kickstart 2.0 keine Bootdisketten mehr infiziert.

1.99 drivebit-bug

Drivebit-Bug

Viele zu Kickstart 1.2/1.3-Zeiten entstandene (Bootblock)viren testen den Schreibschutzstatus einer Diskette folgendermaßen:

```
move.l 24(A1),A0      ; IO-Unit, MsgPort des Laufwerks
move.b 65(A0),$bfd100 ; hier standen unter Kickstart 1.3 die DriveBits,
                      ; das heißt die motorbezogenen Laufwerkbits, die
                      ; in das Drive-Select-Register geschrieben wurden
```

Diese Laufwerksporthstruktur ist aber erst ab Kickstart 2.0 teilweise offiziell dokumentiert, wobei aber solch extrem hardwareabhängige Dinge wie Laufwerkbits mit Absicht nicht dokumentiert sind, denn anstatt direkt die Hardware zu quälen, soll man den korrekten Weg über das `trackdisk.device` mittels z.B. `TD_PROSTATUS` gehen, außerdem ist die Kickstart 2.0-Struktur deutlich abweichend von der Kickstart 1.3-Struktur aufgebaut, ab Position 64 ist die Kickstart 2.0-Struktur nicht mehr definiert, so daß der Befehl `move.b 65(A0),$bfd100` unsinnige Werte in den Diskettencontroller schreibt, wodurch ein später folgender `CMD_WRITE` Befehl nicht korrekt funktioniert, wodurch oftmals eine komplette Spur zerstört wird, meist Spur 0 Kopf 0, also Block 0-10. Diese alten Kickstart 1.3-Bootblockviren, welche den Drivebit-Bug aufweisen, können sich also unter Kickstart 2.0 nicht

verbreiten, sondern machen stattdessen Sektor 0-10 unbrauchbar, wodurch eine 'Not A Dos-Disk' resultiert. Mit disksalv können Sie den Großteil der Diskettendaten retten.

1.100 schlußfolgerung

Schlußfolgerung

Insbesondere die Bootblockviren tun sich durch unsaubere Programmierung hervor, so daß nur noch knapp die Hälfte der Bootblockviren auch auf den neueren Amigamodellen wie A600,1200,3000,4000 laufen.

Die File- und Linkviren hingegen sind meist flexibler und sauberer als die Bootblockviren programmiert, so daß annähernd 3/4 der File- und Linkviren auch auf den neueren Amigas funktionieren.

Die Virengefahr bleibt also bestehen, zumal auch ein Virus, der beim Versuch sich zu installieren abstürzt, sehr störend sein kann und hierbei auch zu Datenverlust führen kann. Andererseits werden sich die Virenprogrammierer auch auf die neuen Verhältnisse einstellen.

Mittlerweile gibt es auch schon viele Viren die erst ab Kickstart 2.0 laufen, da sie gewisse Funktionen oder Zustände benötigen, die erst ab Kickstart 2.0 gegeben sind, so ist es z.B. erst ab Kickstart 2.0 sehr einfach, die dos.library-Funktionen zu verändern.

Siehe z.B. Polyzygotronifikator oder DEBUGGER .

1.101 zukunftsauaussichten

Zukunftsauaussichten

Die Virensituation auf dem Amiga ist in der Tat sehr ärgerlich und wird es auch bleiben. Insbesondere den noch unerfahrenen Computeranfängern wird dadurch der Einstieg in Ihr neues Hobby unnötig erschwert.

Dennoch besteht kein Grund zur Resignation, denn VIRUS CONTROL ist sehr vorausschauend programmiert und gibt dem Anwender viele nützliche Analysehilfsmittel an die Hand, um auch zukünftige Viren zu erkennen.

Bedenken Sie aber immer, daß es niemals einen wirklich hundertprozentigen Virusschutz geben kann. Man kann ja auch durch einen versehentlichen format oder delete-Befehl bereits wichtige Daten verlieren. Fertigen Sie daher regelmäßig Sicherheitskopien an und lassen Sie Ihre Disketten möglichst immer schreibgeschützt.

Programmtechnisch wurden bereits alle typischen Virenarten auf den Amiga umgesetzt, das heißt, es gibt Bootblockviren, Fileviren und Linkviren. Dennoch wird die Virenentwicklung leider weitergehen. Auf der einen Seite werden viele Abwandlungen von bereits bekannten Viren auftauchen. Auf der anderen Seite werden aber auch Viren erscheinen, welche neue Methoden für ihre Infektionen und sonstigen Taten benutzen werden. Oftmals stellen sich aber angeblich 'neue' Viren als Programme heraus, die lediglich eine Schadensfunktion beinhalten. Wenn man ein solches Programm aufruft, dann wird z.B. die Diskette formatiert und Ende. Das Programm kann sich also z.B. nicht weiterverbreiten. Der Schaden ist also auf eine Diskette beschränkt. Man sollte ein Programm, dessen Sinn und Herkunft unklar ist,

nur mit besonderer Sorgfalt starten, also alle Disketten schreibschützen und eine eventuelle Festplatte ausschalten. Leider ist letzteres nicht immer möglich, womit wieder einmal der Sinn regelmäßiger Sicherheitskopien deutlich wird.

Das Amiga-Betriebssystem (aber auch andere Betriebssysteme) sind mittlerweile dermaßen flexibel und leistungsfähig, daß sehr viele Möglichkeiten für ein Virus vorhanden sind, sich irgendwo einzuhängen und irgendwelche Manipulationen vorzunehmen. Je komplexer ein Betriebssystem, desto mehr Möglichkeiten tun sich leider auch für ein Virusprogramm auf. Die weitreichenden Möglichkeiten der neuen leistungsfähigen Betriebssysteme kann man also sinnvoll für sehr brauchbare Programme verwenden. Aber leider kann man in einem solchen komplexen Betriebssystem auch einen Virus einbinden. Doch dieser Sachverhalt ist ein Grundproblem unseres Lebens: Man kann viele Dinge sinnvoll einsetzen oder aber auch mißbrauchen. Es hängt nun mal auch vom Menschen selbst ab, wie er mit seinen Möglichkeiten umgeht. Das Virenproblem ist also oftmals auf Probleme der Virenprogrammierer selbst zurückzuführen. Manchmal sind es junge Menschen, die hier erstmals die Möglichkeit sehen, etwas 'Besonderes' zu leisten. Über die Konsequenzen Ihres Handelns sind sie sich in der Regel nicht bewußt, da sie auch gar nicht weiter darüber nachdenken.

Ich möchte an dieser Stelle ganz ausdrücklich an alle 'Viren-Programmierer' appellieren, das Programmieren von Viren einzustellen, da Viren niemandem nützen, sondern vielmehr unübersehbaren Schaden anrichten können. Steckt bitte Eure Programmierenergien in das Erstellen von sinnvollen Programmen. Für diese Programme kann man dann auch öffentlich Lob ernten, was bei einem Virus natürlich nicht möglich ist. Das Erstellen und Verbreiten von Viren ist keineswegs ein lustiges Kavaliersdelikt, sondern wird vielmehr schwerwiegende Strafen, zumindest finanzieller Art, nach sich ziehen. Die Gesetzeslage ist hier mittlerweile eindeutig. Also setzt bitte nicht Eure Zukunft auf's Spiel, indem Ihr Euch mit dem völlig unnötigen Programmieren von Viren beschäftigt.

1.102 autoradresse

Autoradresse

Pius Nippgen
Bergstr.12
66453 Gersheim
Germany

1.103 konkrete virenbeschreibungen

Konkrete Virenbeschreibungen

Leider ist es recht schwierig, eine gewisse Ordnung in den Virusdschungel zu bringen, denn nicht immer lassen sich Viren eindeutig einer bestimmten Virusgruppe zuordnen bzw. zeigen Merkmale verschiedener Virusarten. Oft werden auch Viren nur geringfügig abgeändert und als angeblich neue

Viren verbreitet. Auch kursieren oftmals verschiedene Namen für letztendlich den gleichen Virus. Wenn bei den nun folgenden Virenbeschreibungen bei einem Virus lediglich z.B. SCA-Abkömmling steht, dann bedeutet dies, daß der Virus mit dem Originalvirus bis auf einige unwesentliche meist Textänderungen identisch ist.

Bootblockviren

Fileviren

Disk-Validatorviren

Linkviren

1.104 Bootblockviren

Bootblockviren

16Bit Crew
6ULDV8
A.H.C.
AEK
AIDS
AIDS-HIV
ALIEN NEW BEAT
AmigaFreak
Amiga-Master
Angel
Australian-Parasite
BamigaSectorOne
BlackFlash
BLACK-KNIGHT
BladeRunners
BLF
BlowJob
BUTONIC's VIRUS 1.1
ByteBandit
ByteBandit 1
ByteBandit 2 usw.
ByteBanditImitation
ByteBanditVIPHS
ByteVoyager I
ByteVoyager II
CCCP
Chaos-TaiPan (Chaos)
Claas Abraham
CLIST-Lamer (UK-Lamerstyle)
CLONK
COBRA
CODER
Copylock
CRACKER-Exterminator
Crackright (DiskDoktors)
CREEPING-EEL
DAFGderFEHY

DASA-ByteWarrior
DATA CRIME
DAT-89
DATALOCK
DERK-MALLANDER
Destructor
DETLEF
DIGITAL DREAM
DigitalEmotions
DiskHerpes (Phantasmumble, Phantastograph)
DIVINA EXTERMINATOR I
Dotty
DUMDUM
DUM2DUM
ELECTRO-VISION
ELENI!
ELENI-CLOCK
EXCREMENT
Exterminator II
EXTREME
FAST
FAST 1
FastEddie
FastEddie-Infecter
F.I.C.A
Forpib
FrenchKiss
FRESHMAKER
Frity
FUCK
FUCK-Lamer (INGO 'S RETURN)
fuck.device
Future-Disaster
Gadaffi
Gandalf
GENESTEALER
Glasnost
Graffiti
GREMLIN
GuardiansBootAids
GX.TEAM
GYROS
HCS
HEIL
HILLY
HODEN V33.17
HULKSTERS
ICE
Incognito
Inger.IQ.Virus
JINX
JITR
Joshua
Joshua 1
JulieTick-PREDATOR
KaKo
Kauki

Killed
L.A.D.S
L.A.D.S - A.I.D.S
LameBlame-TaiPan (LameBlame, CHEATER-HIJACKER, POLISH)
Lamer-Bootblockviren (10)
Laureline V1.0
LEVIATHAN
Little Sven
Loverboy&Sexmachine
LSD
MAD
MAD II
MAD III
MAD IV
MEGAMASTER
METAMORPHOSISV1.0
MEXX
MG's Virus V1.0
MicroMaster
MICROSYSTEMS
Morbid.Angel.Virus
MOSH
MOSH 2
Mount-ELENI-WIRUS
MUTILATOR
Nasty
No.Bandit.any.More
Obelisk
Obelisk II
OPAPA
Overkill
PARADOX I
PARADOX II
PARAMOUNT
PARATAX I
PARATAX II
PARATAX III
PentagonCircleVirusSlayer
PERVERSE I
PowerBomb
PowerTeam
PVL
Revenge Bootloader
Revenge
Ripper
Riska
SACHSEN NO.1
SACHSEN NO.3
SaddamHussein
SAO PAULO
SATAN
SCA
SCA-2001
SCA-AIDS
SCA-DAG
SCA-Kefrens
SCA-MAX

SCARFACE
Sendarian
Sentinel-USSR492
SHIT
Sonja
SS
STARCOM
Starfire-Northstar (4)
Starfire-EastStar
Suicide
SuperBoy
SystemZ 3.0,4.0,5.0,5.1,5.3,5.4
SystemZ 6.1,6.3,6.4,6.5
TAI
Target
TELSTAR(SystemZ-V6.0)
Termigator
T.F.C. Revenge Virus
TIME-BOMB-V1.0
TomatesGentechnicService
Traveller1.0
TRIPLEX
TRISECTOR 911
TURK VIRUS 1.3
TWINZ SANTA CLAUS
Uhr
ULTRA-FOX
Umyj Dupe
VCCofTNT
VERMIN
VirusConstructionI
VirusConstructionII
VIRUS FIGHTER V1.0
VirusV1
Vkill 1.0
WAFT
WAHNFRIED
WARHAWK
Warsaw Avenger
ZACCESS V1.0
ZACCESS V2.0
ZACCESS V3.0
Z.E.S.T
ZENKER
Zombi I

1.105 16bit crew

16Bit Crew

Es handelt sich um einen unsauber programmierten Bootblockvirus, welcher ab \$7ec00 im Speicher steht. Der 16BitCrew-Virus macht sich über den COOL-Vektor resetfest. Er infiziert nur die Bootdiskette. Dieser Virus gleicht programmtechnisch sehr stark dem Graffiti -Virus. Der Hauptunterschied besteht in der fehlenden Grafikmeldung. Man kann am Ende

des 16BitCrew-Bootblocks folgenden Text erkennen: The 16Bit Crew 1988

1.106 6uldv8

6ULDV8

Es handelt sich um einen ByteBandit -Abkömmling, der sich neben einigen unwesentlichen Code-Optimierungen lediglich in der Vertikal-Blank-Routine unterscheidet.

Beim Original-ByteBandit-Virus wird der Rechner nach 7 Minuten blockiert. Auf dem Bildschirm wird dann nur noch die Hintergrundfarbe angezeigt. Diese Blockade tritt aber nur dann auf, wenn der Rechner mindestens 2 * resettet wurde UND mindestens 6 Disketten infiziert wurden. Man kann die Blockade durch folgende Tastenkombination aufheben: L-ALT L-AMIGA SPACE R-AMIGA R-ALT
Die Tasten müssen nacheinander gedrückt werden, wobei die vorherigen Tasten weiterhin gedrückt bleiben müssen.

Beim 6ULDV8-Bootblockvirus wird nun immer nach circa 18 Minuten der Bildschirm zum 'Durchlaufen' gebracht. Das Bild sieht also so aus, als ob es nicht mehr synchronisiert werden könnte. Eine Weiterarbeit am Rechner ist nur durch Drücken von L-ALT + s + f6 möglich.

1.107 a.h.c.

A.H.C.

Es handelt sich um einen unsauber programmierten Bootblockvirus, der ab \$7FA00 im Speicher steht. Aufgrund absoluter ROM-Adressenbenutzung läuft der Virus nicht mit Kickstart 2.0,3.0. Der Virus ist nur für Kickstart1.2 und 1.3 vorbereitet, funktioniert aber aufgrund eines Flüchtigkeitsfehlers nur mit Kickstart1.3.

Der Virus macht sich über den COOL-Vektor resetfest und verbiegt den DOIO-Vektor, um Disketten bereits beim Einlegen infizieren zu können. Vor der Infektion erscheinen allerdings noch verschiedene Alerts, wodurch man also gewissermaßen vorgewarnt wird.

Hello, A.H.C. speaking here!!!

PRESS RIGHT BUTTON

For a good Fuck... A.H.C.!

I'm the A.H.C.-VIRUS

I control your computer!!!

Ein dreifaches Mitleid für AtariST

A.H.C. (hahaha)

Da der Virus nicht gezielt Disketten infiziert, sondern prinzipiell auf jeden Zylinder 0 1024 Bytes schreibt, können Festplatten beschädigt werden.

1.108 aek

AEK

SCA -Abkömmling.

1.109 aids

AIDS

Es handelt sich nicht um einen Lamer-Bootblockvirus .

1.110 aids-hiv

AIDS-HIV

SCA -Abkömmling.

1.111 alien new beat

ALIEN NEW BEAT

Es handelt sich um einen sehr unsauber programmierten Bootblockvirus, welcher nur unter Kickstart 1.2 läuft. Der ALIEN NEW BEAT - Virus löscht den KickTag-Pointer und macht sich über Cold und Cool resetfest. Er ist nun also das einzige resetfeste Programm. Ferner wird der findresident()-Vektor und DoIO()-Vektor verbogen, um jede eingelegte Diskette zu infizieren. Man kann den folgenden Text im Bootblock erkennen:

```
THIS IS THE ALIEN NEW BEAT BOOT! THE BOOT WHICH CREATES A NEW DIMENSION IN  
MEMORY. THIS IS A NEW STYLE OF VIRUS HUNTING!!! 179092 V1.0 Ir 04/01/1989
```

```
You won't believe it, but this thing kills the SCA, ByteBandit,  
Dasa(ByteWarrior), AIDS AND NorthStar virus!!!!
```

Es wird hier also behauptet, daß es sich um einen ganz tollen Anti-Virus-Bootblock handelt. Diese Aussagen dienen aber nur dazu, um den Virus zu tarnen. Es handelt sich um einen 100% Bootblockvirus, welcher jede eingelegte Diskette infiziert. Darüber hinaus besteht auch Gefahr für Festplatten, da der Virus aufgrund nachlässiger Programmierung auch Zylinder 0 der Festplatte beschreiben kann. siehe Rigidiskblock beschädigen .

1.112 amigafreak

AmigaFreak

ByteBandit -Abkömmling.

1.113 amiga-master

Amiga-Master

SCA -Abkömmling.

1.114 angel

Angel

Es handelt sich um einen Bootblockvirus, der einen über die Strahlenposition zufällig verschlüsselten Bootblock schreibt. Der Virus kann sich nur unter Kickstart 1.2/1.3 erfolgreich verbreiten. Ab Kickstart 2.0 werden bei einem Infektionsversuch die ersten 11 Sektoren (Spur0,Kopf0) einer Diskette unbrauchbar gemacht. siehe Drivebit-Bug . Unter Kickstart 1.2/1.3 arbeitet der Virus nun folgendermaßen. Er speichert programmintern die Anzahl der erfolgreich durchgeführten Disketteninfektionen ab.

Nach 201 Disketteninfektionen

wird der Titel des aktiven Fensters abgeändert auf

'The Travel of the Angelvirus Generation Nr. 201 Hi Butonic & Gandalf'

Nach 231, 261 und 291 erfolgreichen Disketteninfektionen wird jeweils ebenfalls diese Titeländerung vorgenommen.

Nach 321 Disketteninfektionen

wird der Titel des aktiven Fensters abgeändert auf

'Laufwerk DF0: ist leider beschädigt...'

Die Glaubwürdigkeit dieser Meldung wird dadurch unterstrichen, daß der Schreib-Lesekopf 7 mal mit viel Lärm die Diskette hoch und runter gefahren wird. Die Diskette nimmt hierbei allerdings keinen Schaden.

Nach 351, 381, 411 usw., also nach jeweils 30 weiteren erfolgreichen Disketteninfektionen wird dieselbe Prozedur, also Titeländerung auf

'Laufwerk DF0: ist leider beschädigt...'

und Schreib-Lesekopf-Lärm durchgeführt.

Damit sich diese Titeländerungen auch optisch zeigen, muß das entsprechende Fenster manipuliert werden.

Der Angel-Virus verbiegt den PutMsg()-Vektor und filtert hierbei eventuelle Schreib- oder Lese-messages an den trackdisk.device-task heraus, um sich dann mit der Disketteninfektion dazwischenzuschieben.

Weiterhin wird der Wait()-Vektor verbogen, um dadurch die PutMsg()-Viruseinschleifung sicherzustellen.

Der Angel-Virus macht sich über den COOL-Vektor resetfest. Sollten jedoch bereits andere resetfeste Programme vorhanden sein, so verhält sich der Angel-Virus recht intelligent, denn er verändert NICHT die bereits benutzten COLD-, COOL- oder KickTag-Vektoren, sondern er ändert den Anfang der entsprechenden resetfesten Programme auf einen Sprung in das Angel-Virusprogramm ab. Das heißt, der Virus verändert keine eventuell

bereits benutzten Reset-Vektoren, wodurch er länger unentdeckt bleiben kann.

Es existiert auch eine im Nachhinein beschädigte Angel-Virus-Variante, bei der ein völlig unsinniger Befehl in den Viruscode reingeschmiert wurde, wodurch der Virus nicht mehr resetfest ist. Vor diesem Reset ist der Virus allerdings voll infektiös.

1.115 austral.parasite

Australian-Parasite

Es handelt sich um einen unsauber programmierten Bootblockvirus, der sich an den Supervisor-Stack-Boden kopiert. Er macht sich über den COOL-Vektor resetfest und verbiegt den DoIO und TD_BeginIO()-Vektor, um zukünftig eingelegte Disketten zu infizieren. Manchmal wird der Bildschirminhalt auf den Kopf gestellt. Im Bootblock kann man folgenden Text lesen:

```
The Australian Parasite! By Gremlin 18/5/88!
Will NOT destroy game bootsectors or corrupt disks,
and kills other viruses!
```

Der Australian-Parasite-Virus versucht sich hiermit als Antivirus zu tarnen. Dies ist völlig falsch, denn der Australian-Parasite-Virus ist ein ganz normaler Bootblockvirus, welcher generell jeden Bootblock überschreibt, es werden also auch Spiele-Bootblöcke überschrieben.

Das mittlerweile etwas veraltete Virus-Schutz-Programm VirusX (letzte offizielle Version war VirusX 4.01) kennt noch nicht den Saddam-Hussein Disk-Validatorvirus. Sollte dieser Saddam-Hussein-Disk-Validatorvirus aktiv sein, dann meldet VirusX manchmal fälschlicherweise einen aktiven Australian-Parasite-Virus.

1.116 bamigasectorone

BamigaSectorOne

SCA -Abkömmling.

1.117 blackflash

BlackFlash

Es handelt sich um einen unsauber programmierten Bootblockvirus, welcher im Speicher ab \$7f000 steht. Er macht sich über den COOL-Vektor resetfest und verbiegt den DoIO()-Vektor, um jede eingelegte Diskette zu infizieren. Festplatten können durch Überschreiben von Zylinder 0 unbrauchbar werden. siehe Rigiddiskblock beschädigen . Manchmal gibt sich der Virus durch folgende Meldung zu erkennen: Auf einem schwarzen Bildschirm werden folgende rote Texte ausgegeben:

```
HELLO, I AM AMIGA !  
PLEASE HELP ME !  
I FEEL STICK !  
I HAVE A VIRUS !  
; BY BLACKFLASH !
```

Diese Texte kann man auch am Ende des Bootblocks lesen.

1.118 black-knight

BLACK-KNIGHT

Es handelt sich um einen Bootblockvirus, der fest ab \$7F300 im Speicher steht. Er macht sich über den COOL-Vektor resetfest und verbiegt den DoIO()-Vektor, um Disketten bereits beim Einlegen infizieren zu können. Allerdings kann hierbei auch der wichtige Zylinder 0 einer Festplatte mit den meist darauf enthaltenen Rigid-Disk-Daten beschrieben werden, wodurch die Festplatte nicht mehr ansprechbar ist, weil die Verwaltungsdaten beschädigt wurden. siehe Rigidiskblock beschädigen .

In dem Viruscode ist folgender kodierter Text versteckt:

```
BLACK KNIGHT (12/11/91)
```

1.119 bladerunners

BladeRunners

```
SCA -Abkömmling.
```

1.120 blf

BLF

Es handelt sich um einen typischen Bootblockvirus, welcher sich über den COOL-Vektor resetfest macht. Der Cold und die Kick-Vektoren werden gelöscht. Es wird der DoIO()- und BeginIO()-Vektor verbogen, um eingelegte Disketten zu infizieren. In dem Bootblock ist der folgende Text nicht lesbar verschlüsselt:

```
you have found the routine !! This is the new virus by BLF
```

Dieser Text wird aber niemals angezeigt.

1.121 blowjob

BlowJob

Es handelt sich um einen unsauber programmierten Bootblockvirus, welcher ab \$7f000 im Speicher steht. Zu Beginn des Bootblocks kann man folgenden Text lesen:

[0.5] [1] [1.8] MB Memory Allocator 3.01

Durch diesen Text will sich der Bootblockvirus als angeblich nützlicher Bootblock tarnen. Es handelt sich aber um einen typischen Bootblockvirus. Beim Booten von einer mit dem BlowJob-Bootblockvirus infizierten Diskette macht sich der Virus direkt über die Kick-Vektoren resetfest. Andere Kick-resetfeste Programme wie z.B. RAD: oder turboprint werden hierbei entfernt. Beim nächsten Reset wird nun der DoIO()-Vektor verbogen, um die erste Diskette zu infizieren, welches meist die Bootdiskette ist. In seltenen Fällen kann die Festplatte unbrauchbar werden. Ferner wird der Vertikal-Blank-Interrupt verbogen, um nach 10 Minuten einen Alert mit folgender Meldung auszugeben:

ONCE AGAIN SOMETHING WONDERFUL HAPPENED (HE HE HE)..
PLEASE POWER OFF - PLEASE POWER OFF - PLEASE POWER OFF

Diesen Text kann man im Bootblock nicht lesen, da er kodiert vorliegt. Nach Beenden des Alerts stürzt der Amiga ab. Es gibt keinen ersichtlichen Grund, warum dieser Virus BlowJob genannt wird.

1.122 butonic's 1.1

BUTONIC's VIRUS 1.1

Der BUTONIC's VIRUS 1.1 ist ein etwas unsauber programmierter Bootblockvirus, da er von der Speicheranordnung eines 512 KB-Amigas ausgeht. Deshalb kann es auf den anderen Amigas unter Umständen zu Problemen kommen. Nach jeweils 15 Resets wird folgende Bildschirmmeldung ausgegeben:

BUTONIC's VIRUS 1.1
GREETINGS TO HACKMACK

Wenn man beim Booten die rechte Maustaste und die s-Taste drückt, dann erscheint eine Meldung, wieviele Disketten bereits infiziert wurden.

<GENERATION NR. #####>

Diese Textmeldungen liegen verschlüsselt vor und sind daher z.B. mit einem Diskettenmonitor nicht zu erkennen. Manchmal wird der BUTONIC's VIRUS 1.1 auch als BAHAN-Virus bezeichnet, weil man diese Buchstabenfolge erkennen kann. Diese Buchstabenfolge hat aber keinen tieferen Sinn. Der BUTONIC's VIRUS 1.1 macht sich über den COOL-Vektor resetfest. Bei dem nächsten Reset wird der DoIO()-Vektor verbogen, um beim gleichfolgenden Booten die Bootdiskette zu infizieren. Danach wird der DoIO()-Vektor wiederhergestellt, es wird also nur die Bootdiskette infiziert. Wenn man beim Booten die linke Maustaste und die Enter-Taste drückt, dann wird zwar die aktuelle Bootdiskette noch infiziert, danach aber wird der Virus entfernt.

1.123 bytebandit

ByteBandit

Es handelt sich um einen Bootblockvirus. Der Name ByteBandit-Virus beruht darauf, daß man folgenden Text im Bootblock lesen kann:

Virus by Byte Bandit in 9.87.Number of copys :

Der ByteBandit-Virus infiziert nicht nur die Bootdiskette, sondern auch jede eingelegte Diskette. Die Besonderheit des ByteBandit-Virus ist die Blockierung des Rechners nach 7 Minuten. Auf dem Bildschirm wird dann nur noch die Hintergrundfarbe angezeigt. Diese Blockade tritt aber nur dann auf, wenn der Rechner mindestens 2 * resettet wurde UND mindestens 6 Disketten infiziert wurden. Man kann die Blockade durch folgende Tastenkombination aufheben:

L-ALT L-AMIGA SPACE R-AMIGA R-ALT

Die Tasten müssen nacheinander gedrückt werden, wobei die vorherigen Tasten weiterhin gedrückt bleiben müssen. Der ByteBandit kopiert sich an den Supervisor-Stackboden und macht sich über Kick-Strukturen resetfest. Er infiziert Disketten bereits beim Einlegen. Hierzu verbiegt er den BeginIO()-Vektor des trackdisk.devices. Ferner wird der exec.library Vertikal-Blank-Vektor verbogen, um nach sieben Minuten den Rechner zu blockieren. Wohingegen der SCA -Virus nur die Bootdiskette infizierte, stellt nun der ByteBandit-Virus eine neue Virus-Qualität dar, da er Disketten sofort beim Einlegen infiziert. Trotz Baujahr 87 ist der ByteBandit-Virus sehr interessant programmiert. Dennoch kann es zum Beispiel mit Speichererweiterungen zum Absturz kommen, da in diesem Fall der Supervisor-Stack nicht immer richtig ermittelt wird.

1.124 bytebandit 1

ByteBandit 1

Es handelt sich um eine 'Vergewaltigung' des Original- ByteBandit -Virus. Es bestehen folgende Unterschiede: Während der Original-ByteBandit-Virus auch mit Kickstart 1.3 arbeitet, funktioniert der ByteBandit1-Virus nur noch mit Kickstart 1.2. Der Grund ist folgender: Der ByteBandit-Virus verändert das trackdisk.device, damit Disketten bereits beim Einlegen infiziert werden. Der Original-ByteBandit-Virus ermittelt die Adresse des trackdisk.device mit Hilfe der FindName()-Funktion, welcher der Text 'trackdisk.device' übergeben werden muß. Diesen Text 'trackdisk.device' kann man auch am Ende des Original-ByteBandit-Virus lesen. Im ByteBandit1-Virus ist dieser Text nicht mehr vorhanden, anstatt dessen übergibt der ByteBandit1-Virus eine ROM-Adresse, an der der Text 'trackdisk.device' steht. Allerdings ist diese Adresse nur für Kickstart 1.2 zutreffend. In Kickstart 1.3 steht dieser Text an einer anderen Adresse. Im Original-ByteBandit ist zu Beginn des Bootblocks folgender Text zu lesen:

Virus by Byte Bandit in 9.87.Number of copys :

Anstatt dieses Textes ist beim ByteBandit1 eine Routine vorhanden, welche nach dem nächsten Reset das komplette Chip-RAM bis auf 86016 Bytes belegt. Weiterhin kann man eine mit dem ByteBandit1 infizierte Diskette daran

erkennen, daß sich der Bildschirm pink färbt, wenn man beim Booten F10 drückt.

1.125 bytebandit 2

ByteBandit 2

Es handelt sich um einen Bootblockvirus, welcher auf dem ByteBandit -Virus basiert. Es bestehen lediglich folgende zwei Unterschiede: Man kann keinen typischen Text im Bootblock mehr erkennen. Beim ByteBandit konnte man Virus by Byte Bandit in 9.87.Number of copys : am Anfang des Bootblocks lesen. Beim Original-ByteBandit wird der Rechner nach 7 Minuten blockiert. Auf dem Bildschirm wird dann nur noch die Hintergrundfarbe angezeigt. Diese Blockade tritt aber nur dann auf, wenn der Rechner mindestens 2 * resettet wurde UND mindestens 6 Disketten infiziert wurden. Man kann die Blockade durch folgende Tastenkombination aufheben: L-ALT L-AMIGA SPACE R-AMIGA R-ALT. Beim ByteBandit2 wird nach 70 Sekunden und 5 Resets und 5 Disketteninfektionen ebenfalls der Rechner angehalten. Nach Drücken der '['-Taste aus dem Zehnerblock kann weitergearbeitet werden.

1.126 bytebanditimitation

ByteBanditImitation

Es handelt sich auch hierbei um eine schlechte und fehlerhafte Nachprogrammierung des Original- ByteBandit -Virus. Der anfängliche ByteBandit-Text ist nicht mehr vorhanden und die Vertikal-Blank-Routine wurde entfernt.

1.127 bytebanditviphps

ByteBanditVIPHS

ByteBandit -Abkömmling, aufgrund weniger Textes konnte der Virus-Code nach vorne verschoben werden. Anstatt Bitplane DMA aus erfolgt Anbsturz.

Insbesondere der SCA - und ByteBandit -Virus dienten als Vorlage für eine Vielzahl weiterer Bootblockviren. Meistens wurde lediglich der Text abgeändert, manchmal aber wird auch versucht, den eigentlichen Virus-Code abzuändern, wodurch oftmals recht fehlerhafte Viren in Umlauf gelangen.

1.128 bytevoyager i

ByteVoyager I

Es handelt sich um eine Weiterentwicklung des BlowJob -Bootblockvirus. Es liegt also auch ein unsauber programmierter Bootblockvirus vor, welcher ab

\$7f000 im Speicher steht. Der ByteVoyager arbeitet nun aber ähnlich den Lamerviren mit zufälligen Kodiererroutinen, wodurch jeder Bootblock ein anderes Aussehen erhält. Beim Booten von einer mit dem ByteVoyagerI Bootblockvirus infizierten Diskette macht sich der Virus direkt über die Kick-Vektoren resetfest. Andere Kick-resetfeste Programme wie z.B. RAD: oder turboprint werden hierbei entfernt. Beim nächsten Reset wird der DoIO()-Vektor verbogen, um nun alle weiteren Disketten infizieren zu können. Weiterhin wird hierbei der Diskettenname in 'Infected by BYTE VOYAGER !!!!!' umbenannt. In seltenen Fällen kann die Festplatte unbrauchbar werden. Ferner wird der Vertikal-Blank-Interrupt \$6c verbogen, um nach 15 Minuten den Bildschirm für 1 Minute lang 25 mal in der Sekunde AUS und AN zu schalten. Nach 16 Minuten wird der Amiga dann zum Abstürzen gebracht.

1.129 bytevoyager ii

ByteVoyager II

siehe ByteVoyager I . Es gibt nur einen wesentlichen Unterschied. Der Diskettenname wird in 'Another Virus by Byte Voyager' umbenannt. Bei ByteVoyagerI wurde der Diskettenname in 'Infected by BYTE VOYAGER !!!!!' umbenannt.

1.130 cccp

CCCP

siehe CCCP-Bootblock+Linkvirus .

1.131 chaos-taipan

Chaos-TaiPan (Chaos)

Dieser Bootblockvirus baut sehr stark auf dem
LameBlame-TaiPan(LameBlame,CHEATER-HIJACKER,POLISH) auf.
Es handelt sich also um einen Bootblockvirus, welcher sich über den COOL-Vektor resetfest macht. Weiterhin wird der DoIO()-Vektor verbogen, um Disketten beim Einlegen zu infizieren. Die Bootblöcke sehen immer verschieden aus, da sie mit einem Zufallswert (Rasterstrahl) kodiert werden. Beim LameBlame-TaiPan wurde nach 8 Disketteninfektionen lediglich ein Alert ausgegeben. Der Chaos-TaiPan ist nun aggressiver programmiert und überschreibt nach 8 Disketteninfektionen die komplette Disketten mit unsinnigen Daten. Danach wird ein Alert mit z.B. folgendem Text ausgegeben:

Chaos! by Tai-Pan Number of Diskformats : 001

Danach wird ein Reset bzw. Absturz ausgelöst.

Der Chaos-TaiPan-Bootblockvirus wird auch durch das Programm
VIRUS-INSTALL v2.0 installiert.

1.132 claas abraham

Claas Abraham

Es wird der Auto-Interrupt 2 verbogen. Dieser wird z.B. beim Drücken oder Loslassen einer Taste ausgelöst. In diesem Interrupt wird dann immer der Virus über Cold, Cool und die Kick-Vektoren alleinig resetfest gemacht. Die Kickstruktur weist folgenden Identifizierungsstring auf:

>>> Claas Abraham Virus !!! <<<

Der Virus arbeitet mit selbstmodifizierendem Code, so daß die Virusbootblöcke immer verschieden aussehen, obwohl letztendlich immer der gleiche Virus vorliegt. Nach dem nächsten Reset wird beim Abarbeiten der Kickstruktur der BeginIO()-Vektor des trackdisk.devices verbogen, wodurch nun Disketten bereits beim Einlegen infiziert werden.

Nach 15 Disketteninfektionen wird die Diskette ab Block 880 (Root-Spur) formatiert.

1.133 clist-lamer(uk-lamerstyle)

CLIST-Lamer(UK-Lamerstyle)

siehe Lamer-Bootblockviren .

1.134 clonk

CLONK

Es handelt sich einen sogenannten Antivirus-Bootblock, das heißt, der Anwender wird zuerst gefragt ob ein verdächtiger Bootblock überschrieben werden soll. CLONK macht sich über die Kick-Vektoren unter dem Namen CLONK! resetfest, und verbiegt weiterhin den KickChecksum- und DOIO-Vektor.

1.135 cobra

COBRA

Es handelt sich um einen ByteBandit -Bootblockvirus-Abkömmling, es gelten also die beim ByteBandit-Virus wiedergegebenen Informationen, es besteht jedoch folgender Unterschied, anstatt nach 2 Resets und 6 Disketteninfektionen den Rechner dann nach 7 Minuten zu blockieren, werden nach 2 Resets und 8 Disketteninfektionen alle Diskettenlese- und schreibzugriffe in ein Schreiben des Virusbootblockes umgewandelt, das heißt es wird nicht nur der Bootblock mit dem Virusbootblock überschrieben, sondern es werden auch zufällige Diskettenbereiche überschrieben, wodurch die Diskette zum Großteil unbrauchbar wird. Der Name des Virus rührt von folgendem Text her, den man am Ende des Bootblocks lesen kann:

COBRA HAW HAW HAW

1.136 coder

CODER

Es handelt sich um einen unsauber programmierten Bootblockvirus, welcher ab \$7f600 im Speicher steht. Der CODER-Virus macht sich über die Kick-Vektoren resetfest, wobei andere Kick-resetfeste Programme rausgeworfen werden. Er verbiegt den DoIO()-Vektor, um Disketten beim Einlegen zu infizieren. Unter Umständen wird aufgrund unsauberer Programmierung auch Zylinder 0 von Festplatten beschrieben, siehe Rigidiskblock beschädigen .

Der CODER-Virus verbiegt den AutoInterrupt 2 (\$68), welcher beim Drücken oder Loslassen von Tasten ausgelöst wird. Nach einer gewissen Anzahl von Tastaturbetätigungen wird der Rechner zum Abstürzen gebracht. Am Anfang des CODER-Bootblocks kann man folgenden Text lesen:

Bootblock installed with 'CODER' - The Ultimate Viruskiller!!

Mit diesem Text will sich der Virus als Antivirus tarnen. Ab \$7fa00 kann man einen Text lesen, welcher unmißverständlich auf den Virus hinweist:

Something WONDERFUL has happened!! Your Amiga is alive, and it is infected with the 'Coders Nightmare Virus'. - The ultimate key-killer, masterminded by the megamighty Mr. N of The PowerBomb Systems!!

1.137 copylock

Copylock

Es handelt sich um einen Bootblockvirus, der sich über den COOL-Vektor resetfest macht. Nach dem nächsten Reset wird dann der DoIO()-Vektor verbogen, um Disketten bereits beim Einlegen infizieren zu können. Der Virus belegt 2048 Bytes (Block 0,1,2,3) auf Diskette, wodurch also bei vollen Disketten die Dateien beschädigt werden, die Daten auf Block 2,3 belegen, denn nur Block 0,1 ist automatisch für den Bootblock reserviert. Es ist auch umgekehrt denkbar, daß bereits eine infizierte Diskette vorliegt und daß später weitere Dateien auf die Diskette abgespeichert werden, wobei der Viruscode auf Block 2,3 überschrieben werden kann, wodurch es dann beim Booten von dieser Diskette zum Absturz kommt, denn beim Booten von einer mit dem Copylock-Virus infizierten Diskette lädt der Virus 2048 Bytes nach \$7f000 nach. Die ersten 1024 Bytes ähneln aus Tarngründen sehr einem normalen Bootblock, der Virus hat lediglich 44 Bytes des Original-Bootblocks mit einer 44-Byte langen Virusnachladeroutine ausgetauscht, die restlichen 1024 Bytes von Block 2,3 stellen den eigentlichen Viruscode dar.

Beim Infizieren einer neuen Disketten werden die zu schreibenden 2048 Bytes ab \$7f800 angelegt und mit der Strahlenposition aus \$dff006 zufällig verschlüsselt.

Im Speicher kann ab \$7f420 folgenden Text lesen:

Copylock Amiga (c) Rob Northern. All rights reserved.

und ab \$7f6c0 folgenden Text:

```
* YEP! ROB NORTHERN ON THE BOARD ! MY COPYLOCKS ARE FUCK.  
THE CRACKERS ARE BETTER THAN ME. THAT'S WHY I'M WRITING  
VIRUSES !!! (IN THE HOPE THAT THEY ARE BETTER AS MY COPYLOCKS!) *
```

1.138 cracker-extermin.

CRACKER-Exterminator

Es handelt sich um einen Bootblockvirus, der fest ab \$7ae00 im Speicher steht, weiterhin wird auch Speicher ab \$70000 beschrieben. Der Virus macht sich über den COOL-Vektor resetfest und verbiegt den DoIO()-Vektor, um Disketten bereits beim Einlegen zu infizieren. Nachdem 3 Disketten infiziert wurden, wird eine vierte eingelegte Diskette nicht infiziert, sondern es wird der Schreib-Lesekopf des betroffenen Diskettenlaufwerks endlos zu andauernden Richtungswechseln gezwungen, eine Tortur, die die Diskettenlaufwerksmechanik nicht lange mitmachen wird, aber Sie werden sicherlich möglichst bald von Hand einen Reset auslösen, denn die Töne, die das Laufwerk von sich gibt, lassen einen eindeutig um die Gesundheit des Laufwerks fürchten.

Zu Beginn des Bootblocks kann man folgenden Text lesen:

The CRACKER Exterminator

Weiterhin ist in dem Virus folgender allerdings verschlüsselter und somit nicht lesbarer Text enthalten:

```
Hi hacker! your amiga is infected with a new generation of virus called:  
The CRACKER Exterminator ! made by: A:C:C:W [Anti-Cracker-Club-West]  
have fun, you fucking cracker!
```

1.139 diskdoksors

Crackright (DiskDoksors)

Der Crackright (DiskDoksors)-Virus ist ein sehr heimtückischer Virus. Er offenbart erst nach 2000 Disketten-Infektionen seine volle Bösartigkeit. Der Crackright (DiskDoksors)-Virus macht sich über den COLD- und COOL-Vektor resetfest. Er verbiegt den DoIO()-Vektor, um Disks schon beim Einlegen zu infizieren. Der Crackright (DiskDoksors)-Virus installiert einen Task namens 'clipboard.device', welcher permanent die Vektoren auf den Virus verbiegt. Weiterhin kopiert dieser Task seinen Code im Speicher umher, was aber ohne weitere Bedeutung ist. Es werden auch bei jedem DoIO()-Zugriff die Vektoren verbogen. Ab dem fünften Reset wird bei jedem Reset 10240 Bytes * Resetanzahl Chip-RAM abgezogen. Nach dem zehnten Reset wird also 10*10240=102400 Bytes Chip-RAM belegt. Der Crackright (DiskDoksors)-Virus merkt sich die Anzahl aller bisherigen Disk-Infektionen. Wenn die Anzahl der Disk-Infektionen größer als 2000 wird, ändert der Crackright (DiskDoksors) Virus sein bisheriges 'relativ friedliches' Verhalten. Er verbiegt den exec.library-Vertikal-Blank-Vektor, um nach 30 Minuten den Rechner zu resetten. Zuvor wird der Rechner ab circa der 26ten Minute zunehmend

langsamer. Bei jedem fünften Disketten-Einlegen wird die obere Diskhälfte einschließlich der Root-Spur FORMATIERT!!!! Der Virus läuft nur mit Kickstart 1.2. Mit Fast-RAM usw. arbeitet er problemlos zusammen. Der Crackright(DiskDoktors)-Virus ist recht interessant programmiert. Verwerflich ist jedoch seine primitive Bösartigkeit.

1.140 creeping-eel

CREEPING-EEL

Es handelt sich um einen DiskHerpes(Phantasmumble,Phantastograph) -Abkömmling, der sich nur durch einen am Ende des Bootblocks abgeänderten Text vom Original unterscheidet.

```
--- Hello Computerfreak ---  
You've got now your first VIRUS  
**** THE CREEPING EEL ****  
Many Disks are infected !!
```

Written by >MAX OF STARLIGHT<
© 29.04.1992 <<MAX>>

1.141 dafgderfehy

DAFGderFEHY

MAD II -Abkömmling.

1.142 dasa-bytewarrior

DASA-ByteWarrior

Der Virus löscht den COLD- und COOL-Vektor und macht sich über die Kick-Vektoren resetfest. Er verbiegt den DoIO()-Vektor, um Disks schon beim Einlegen zu infizieren. Der Virus nistet sich im vermeintlich sicheren Supervisorstack \$7f800-\$80000 ein, was jedoch nur für 512 KB-Amiga gilt. Ferner läuft er nur mit Kickstart 1.2, da er direkt ins ROM einspringt. Es wird gelegentlich die Meinung vertreten, der DASA-ByteWarrior-Virus sei als Antivirus geplant gewesen, dem ist nicht so, denn der DASA-ByteWarrior ist geradezu der Prototyp eines Bootblockvirus. Leider weist der DASA-ByteWarrior-Virus auch noch einen sehr schwerwiegenden Programmierfehler auf, siehe Rigiddiskblock beschädigen .

1.143 data crime

DATA CRIME

CCCP -Abkömmling.

1.144 dat-89

DAT-89

Der DAT-89-Bootblockvirus ist sehr stark mit dem DASA-ByteWarrior -Virus verwandt. So steht der Virus ebenfalls ab \$7f800 im Speicher, macht sich über KickTag und KickChecksum resetfest, und funktioniert nur unter Kickstart 1.2. Es gibt lediglich folgende Unterschiede:
Der DAT-89 ist bösartiger wie der DAS-ByteWarrior-Virus, da er bei jeder eingelegten Disk nicht nur den Bootblock, sondern auch den Rootblock überschreibt, wodurch die Diskette unlesbar wird. Mit z.B. disksalv können aber in der Regel die Daten wiederhergestellt werden.
Weiterhin wird nach circa 15 Resets ein Alert ausgegeben.
Der DAT-89-Virus versucht sich mit folgenden Texten als ein harmloses Antivirusprogramm zu tarnen.

```
DAT ANTIVIRUS V1.25
DAT '89!!!
THIS BOOT RESETS ALL VECTORS SO THAT NO VIRUS
CAN TAKE CONTROL OVER THE COMPUTER! SIGNED: DAT '89!
```

1.145 datalock

DATALOCK

1.1

Es handelt sich um ein unsauber programmierter Bootblock-Virus, der aufgrund des Drivebit-Bug ab Kickstart 2.0 nicht richtig funktioniert.
Der Virus löscht den COLD und COOL-Vektor und macht sich über die Kick-Vektoren als alleiniges Programm resetfest, wobei kein Identifikationsstring verwandt wird. Weiterhin wird der DOIO-Vektor verbogen. Da hierbei nicht gezielt auch Diskettenzugriffe geprüft wird, kann der Rigiddiskblock z.B. einer Festplatte beschädigt werden. Der Virus steht ab \$7F700 im Speicher. Bei einem Lesezugriff auf den Bootblock schiebt der Virus ein Beschreiben des Bootblocks mit dem Viruscode dazwischen. Bei einem Schreibzugriff auf den Bootblock überschreibt der Virus 8 Blöcke in der Rootspur ab 890. Die Daten beginnen mit
DATALOCK 1.1 (c) '94!..ALL (?) CODE BY DEATHCORE
und enden mit zufälligen Daten.

1.2

Im Unterschied zu DATALOCK 1.1 werden anstatt 1024 nun 2048 Bytes ab Block 0 geschrieben, wodurch also Daten auf Block 2,3 überschrieben werden können. Es werden nun 4 Blöcke in der Rootspur ab 880 überschrieben und weiterhin zufallsgesteuert 4 weitere Diskettenblöcke. Die Daten beginnen mit
>>> DATALOCK 1.2 <<< '94! (c) by DEATHCORE.

1.146 derk-mallander

DERK-MALLANDER V1.0

Es handelt sich um einen Bootblockvirus, der ab \$7F800 im Speicher steht. Der Virus läuft auf Rechnern, welche 1 oder 2 MB Chip-RAM besitzen oder auf Rechnern mit Fast-RAM und Kickstart 2.0. Auf z.B. einem 512KB-Chip-RAM-Amiga unter Kickstart 1.3 erfolgt immer sofort ein Reset, da der Supervisorstack in dem Fall mit dem Virus-Speicherbereich kollidieren würde. Der Virus macht sich über KickTag, KickChecksum resetfest, wobei keine Identifikationsstrings verwendet werden. Andere Kick-resetfeste Programme wie z.B. RAD: werden gelöscht. Weiterhin wird auch der Cold, Cool und Warm-Vektor gelöscht. Nach dem nächsten Reset wird der DoIO()-Vektor verbogen, um Disketten bereits beim Einlegen zu infizieren. Achtung, es können auch die Systeminformationen von Auto-Boot-Festplatten beschädigt werden, wodurch die Festplatte nicht mehr erkannt wird. Der Virus belegt 4 Blöcke auf Diskette. In der ersten 2 Blöcken, dem eigentlichen Bootblock, steht der Virus-Code. Der Virus legt aber in den nächsten zwei Blöcken noch den normalen Bootblock ab, wodurch bei vollen Disketten Dateien unwiderbringlich beschädigt werden können. Der Virus belegt nach und nach das ganze Chip-RAM an, da er mehr oder weniger schnell immer 4 KB anfordert. Wenn kein Chip-RAM mehr vorhanden ist, wird ein Alert mit folgendem Text ausgegeben. Danach erfolgt ein Reset:

J.D. MALLANDER VIRUS V. 1.0
I need lots of money - buy my Cool pd serie 'action power'

Man kann diesen Text im Bootblock nicht lesen, da er kodiert vorliegt. Manchmal wird der MALLANDER-Virus auch DERK-Virus genannt, weil man 2 * DERK in dem Virus-Bootblock lesen kann.

1.147 destructor

Destructor

Es handelt sich um einen Bootblockvirus, welcher sich über den COLD-Vektor resetfest macht. Nach einem Reset wird die eingelegte Bootdiskette formatiert. Es handelt sich nicht direkt um einen Virus, da keine neuen Disketten infiziert werden.

1.148 detlef

DETLEF

Es handelt sich um einen Bootblockvirus, der sich über die Kick-Vektoren alleinig resetfest macht, hierbei gehen also andere resetfeste Programme wie z.B. die resetfeste RAM-Disk RAD: verloren. Weiterhin wird der DoIO()-Vektor verbogen, um Disketten bereits beim Einlegen zu infizieren. Hierbei werden immer 2048 Bytes ab \$70000 mit den Original-IO-Request-Daten und dem zu schreibenden Bootblock überschrieben. Der Bootblock wird dann noch mit der zufälligen Mausposition aus \$dff00a verschlüsselt. Während eines Resets gibt sich der Virus manchmal mit folgendem Alert zu erkennen:

Guten Tag.
» Ich heie DETLEF «
Ich werde Sie in der nchsten Zeit etwas nerven.
Gemacht wurde ich von . M A X .
gefolgt von etwas Bildschirmmll, weil der Alertdaten nicht
korrekt beendet werden.

Den Text kann man im Bootblock nicht erkennen, da er verschlsselt vorliegt.

1.149 digitalemotions

DigitalEmotions

Es handelt sich um einen Bootblockvirus, welcher sich ber den COOL-Vektor
resetfest macht. Er infiziert nur die Bootdiskette und gibt sich manchmal
beim Booten durch einen Alert mit folgendem Text zu erkennen:

WJSVT
*** DIGITALEMOTIONS ***

1.150 digital dream

DIGITAL DREAM

Es handelt sich um einen unsauber programmierten Bootblockvirus,
der ab \$70000 bis \$80000 Speicher belegt. Der Virus funktioniert
prinzipiell auch unter Kickstart 2.0, er strzt aber auf hheren
Prozessoren ab, da die Dekodieroutine des Virus mit den Caches
kollidiert. Weiterhin strzt der Virus auf Systemen mit Speicher
oberhalb der unteren 16 MB ab, da der Virus Speicherstelle \$400
lscht, \$400 aber Teil der System-Speicherliste ist und bei Speicher
oberhalb 16 MB ungleich Null ist.
Der Virus macht sich alleinig ber die KICK-Vektoren unter dem Namen
>>DIGITAL DREAM<< resetfest und lscht den COLD- und COOL-Vektor.
Weiterhin wird der Supervisor-Vektor verbogen um permanent die
alleinige Resetfestigkeit und DOIO-Vektor-Verbiegung sicherzustellen.
Der DOIO-Vektor wird verbogen, um Disketten bereits beim Einlegen
infizieren zu knnen. Hierbei wird auf Block 0,1 der Diskette der
Viruscode und auch Block 2,3 der ehemalige Original-Bootblock
geschrieben, wodurch Daten berschrieben werden knnen.
In dem entschlsselten Viruscode kann man lesen

>>DIGITAL DREAM<<
by Max of StarLight

1.151 diskherpes

DiskHerpes (Phantasmumble, Phantastograph)

Der DiskHerpes-Virus wird auch Phantasmumble genannt, obwohl es für diese Bezeichnung keinerlei Gründe gibt. Es handelt sich um einen sehr bösartigen Bootblockvirus, der immer ab \$7EC00 im Speicher steht. Wenn man von einer mit dem Herpes-Virus infizierten Diskette bootet, so wird der Virus aktiviert und macht sich über den COOL-Vektor resetfest. Weiterhin wird der DoIO()-Vektor verbogen, um bei jedem Reset die Bootdiskette zu infizieren, das heißt es wird der Bootblock der Bootdiskette mit dem Disk-Herpes-Bootblock überschrieben. Leider werden aber auch 40 Blöcke ab Block 880 überschrieben, hierdurch wird der Diskette unwiederbringlichen Schaden zugefügt. Da die Root-Spur überschrieben wurde, wird nun die Diskette nicht mehr als DOS-Diskette anerkannt. Mit disksalv kann man in der Regel die meisten Dateien retten. Nach 20 Resets gibt sich der Virus namentlich zu erkennen:

--- Hello Computerfreak ---

You've got now your first VIRUS
** D i s k - H e r p e s **
Many Disks are infected !!

Written by >tshteopghraanptha<
© 22.07.1987 in Berlin

Dieser Text ist mit einem schwarz-rot-gelbem(=Deutschlandfarben) Bildschirm unterlegt. Wenn man während des Resets die linke Maustaste und die Joystick-Feuertaste gedrückt hält, dann wird der Virus entfernt. Als Bestätigung färbt sich der Bildschirm kurzzeitig schwaz-rot-gelb.

1.152 divina extermin.

DIVINA EXTERMINATOR I

Der Virus fordert mit AllocMem() 1024 Bytes für seinen Virus-Code an. In diesen Speicherbereich wird dann der Virus-Code kopiert, wobei zuvor eine Dekodierung erfolgt. In dem Speicherbereich kann man dann folgenden Text lesen:

VIRGO PRESENTS DIVINA EXTERMINATOR I COPIES:

Im Bootblock selber kann man diesen Text wegen der Kodierung nicht erkennen. Da der Virus nicht gezielt Chip-RAM anfordert, wird er auf Amigas mit Fast-RAM meist nicht resetfest sein, da der Virus sich dann meist in das Fast-RAM kopiert, wo er dann bei einem Reset nicht gefunden wird, da das Fast-RAM erst später konfiguriert wird. Wenn von einer mit dem DIVINA EXTERMINATOR I infizierten Diskette gebootet wird, dann wird für 1 Sekunde der Interrupt-Vektor \$6c verbogen. Anschließend macht sich der Virus über den COOL-Vektor resetfest und verbiegt den Exec-Vertikal-Blank-Interrupt. Weiterhin wird der DoIO()-Vektor verbogen, um jede eingelegte Diskette zu infizieren, wobei der Virus-Bootblock mit einem Strahlenpositionszufallswert

(\$dff006) verschlüsselt wird. Außerdem wird noch der Interrupt-Vektor \$64 verbogen, um andauernd den DoIO()-Vektor auf den Virus zu verbiegen. Nach 3 Disk-Infektionen und 2 Minuten wird der Exec-Vertikal-Blank-Interrupt wiederhergestellt und dafür der Interrupt-\$68 verbogen. Sobald nun 10 Mal die k-Taste gedrückt wurde, stürzt der Amiga ab.

Der Virus versucht anstatt seines Virus-Bootblocks einen Standard-Bootblock vorzutauschen.

Da der Virus nicht gezielt auf Disketten-Zugriffe prüft, kann es sein, daß eine Festplatte unbrauchbar wird.

1.153 dotty

Dotty

Der Dotty-Bootblockvirus steht ab \$7f000 im Speicher. Er löscht den COOL-Vektor und macht sich über KickTag, KickChecksum resetfest. Hierbei werden andere Kick-resetfeste Preogramm wie z.B. RAD: entfernt. Nach dem nächsten Reset wird dann der DoIO()-Vektor verbogen, um Disketten bereits beim Einlegen infizieren zu können. Nachdem eine Diskette infiziert wurde, wird der EXEC-IRQ-3 verbogen, um dadurch 7 Minuten zu warten. Danach erfolgt ein Rechnerabsturz. Angeblich soll der Mauspointer oder Screen manipuliert werden, aber die betreffende Routine ist völlig kaputt, so daß nach einem kurzen Herumstochern in der Intuition-Base ein Absturz erfolgt. Folgender Text kann in dem Bootblock gelesen werden:

```
Made by the mysterious RT for P.A.L (People Against Lameness)
Watch for your mousepointer. It might go nuts or fuck up your screen.
Greetz only to the BEST OF THE BEST:
Vision. Skid Row. Kefrens. Flash. Scoopex. Fica.
```

1.154 dumdum

DUMDUM

Der DUMDUM ist ein typischer Bootblockvirus, welcher sich über den COOL-Vektor resetfest macht. Nach dem nächsten Reset wird dann der DoIO()-Vektor verbogen, um Disketten bereits beim Einlegen zu infizieren. Der Virus steht ab \$7FA00 im Speicher. In der neuen DoIO()-Routine wird der Auto-Interrupt 1 (\$64) verbogen, um die Resetfestigkeit sicherzustellen.

1.155 dum2dum

DUM2DUM

Es handelt sich um einen Bootblockvirus, der sich über den COOL-Vektor resetfest macht und den DOIO-Vektor verbiegt, um Disketten bereits

beim Einlegen zu infizieren, wobei die ersten 6 Blöcke der Diskette überschrieben werden, wodurch also eventuelle Daten auf den Blöcken 2 bis 5 verloren gehen, wodurch Lesefehler resultieren können. Weiterhin manipuliert der Virus das BitMap-Flag und die Bitmap in Rootblock 880. Der Virus zählt in der DOIO-Routine einen Zähler hoch, wenn dieser bei 80 angelangt ist, nimmt der Virus an, daß nun auch die dos.library installiert ist und verbiegt den Open,Write und Read-Vektor, allerdings nur unter Kickstart 1.3 und auch nur dann, wenn diese Vektoren noch nicht anderweitig verbogen wurden. Diese neuen Virusroutinen bewirken nun, daß in Abhängigkeit von der Strahlenposition verfälschte Daten geschrieben werden, es resultieren also meist unbrauchbare Dateien. Der Name des Virus rührt daher, daß man in dem Viruscode DUM<II>DUM lesen kann.

1.156 electro-vision

ELECTRO-VISION

Es handelt sich um einen ByteBandit -Abkoemmling, der sich lediglich durch Textänderungen vom Original unterscheidet, so soll z.B. der Text 'ANTI-VIRUS' einen harmlosen Bootblock vortäuschen. Weiterhin wird anstelle der Rechnerblockade nach 7 Minuten lediglich die LED-Helligkeit/Soundfilter umgeschaltet.

1.157 eleni!

ELENI!

Es handelt sich um einen unsauber und fehlerhaften programmierten Bootblockvirus, der sich über den COOL-Vektor resetfest macht und den DOIO-Vektor verbiegt. Bei einem DOIO-READ-Befehl wird der Virusbootblock geschrieben und anschließend unsinnigerweise ein RAWWRITE-Kommando abgesetzt, wodurch des öfteren Lesefehler auf Spur0 resultieren. Da keine spezielle Prüfung auf Disketten unternommen wird, können durch diese Virusinfektion Festplatten unbrauchbar werden. Weiterhin wird bei jeder Disketteninfektion versucht, das Zehner-Jahres-Register bei einer Hardwareuhr ab \$DC0000 um eins zu erniedrigen, aber nicht alle Amigas besitzen ab \$DC0000 eine Hardwareuhr. Bei einem DOIO-WRITE-Befehl wird der loadseg-Vektor verbogen. In diesem Virus-loadseg-Code wird zuerst geprüft ob das Zehner-Jahres-Register = 1 ist, in diesem Fall werden die ersten 6 Zeichen des Namens des zu ladenden Files mit ELENI! überschrieben, es wird jedoch vergessen den Text mit 0 abzuschließen, wenn dies der Fall wäre, könnte auf diese Weise ein Programm namens ELENI! geladen werden. Wenn das Zehner-Jahres-Register <> 1 ist, was meist der Fall ist, dann wird das zu ladende File, wenn es kleiner 100000 Bytes ist, nach \$70000 eingelesen, was allerdings 2 MB ChipMem voraussetzt. Da nur 26 Zeichen des Filenamens berücksichtigt werden, kann das Öffnen des Files fehlschlagen, in dem File werden dann circa die ersten 5 KB nach \$4EAEFDD8 durchsucht und durch \$4EAEEC00 ersetzt, also üblicherweise jsr -552(a6) durch jsr -5120(a6), wodurch diese Files in der Regel später

nicht mehr funktionieren können, lediglich wenn der Virus aktiv ist, könnten in seltenen Fällen die Programme noch arbeiten, denn der Virus schreibt `jmp -552(a6)` nach `-5120(a6=execbase)`, was allerdings nur dann Erfolg haben kann, wenn sich an 2 MB ChipMem direkt FastMem anschließt, in welchem die `execbase` aufgebaut wurde.

Unter Kickstart1.2/1.3 führt der Aufruf der nicht existierenden `ColdReboot`-Routine zum Absturz, der meist nur durch Ausschalten des Amigas beendet werden kann. Weiterhin funktioniert das Verbiegen des `loadseg`-Vektors auch nicht unter Kickstart1.2/1.3.

Der `ELENI!`-Bootblockvirus wird auch durch ein Programm namens `MessAngel Killer` installiert.

1.158 eleni-clock

ELENI-CLOCK

Es handelt sich um eine Bootblockvirus, der immer ab `$07f144` im Speicher steht und sich über den `COOL`-Vektor resetfest macht. Weiterhin wird der `SumKickData`-Vektor und `DOIO`-Vektor verbogen. Bevor der Virus einen Bootblock infiziert, prüft er ob in dem Bootblock der Befehl `jsr -456(a6)` vorkommt, wenn ja, dann handelt es sich zumindest nicht um einen normalen Standardbootblock und der Virus sichert den Bootblock nach Block 1738-1739, was natürlich insbesondere bei vollen Disketten zu Datenverlust führen wird. Beim Booten von der Diskette wird dann dieser Originalbootblock aus Tarngründen ausgeführt. Da der Virus nicht explizit auf das `trackdisk.device` prüft, kann der Virus auch versehentlich Festplatten infizieren und somit unbrauchbar machen. Der Virus manipuliert eine eventuell ab `$DC0000` vorhandene Echtzeituhr und in Abhängigkeit von den Uhrwerten und dem `Netzteiltickcounter` wird des öfteren eine Endlosschleife aufgerufen, in welcher die Rechner-LED und die Diskettenlaufwerke-LED's blinken und die Stepmotoren bewegt werden. Weiterhin wird die Tastatur blockiert. Ein Ende ist oftmals nur noch durch Ausschalten des Rechners möglich. Gegen Ende des Bootblockes kann man `*ELENI*` lesen.

1.159 excrement

EXCREMENT

Es handelt sich um einen Bootblockvirus, der immer ab `$7F400` im Speicher steht. Er weist deutliche Ähnlichkeit mit dem `SCA`-Virus auf und infiziert wie dieser nur die Bootdiskette, indem der `DoIO()`-Vektor beim Booten kurzzeitig verbogen wird. Resetfestigkeit wird über den `COOL`-Vektor vorgenommen. Wird während dem Reset die linke Maustaste gedrückt, dann wird der Virus entfernt. Bei jedem Reset wird für kurze Zeit die LED zum Blinken gebracht. Nach jeweils 16 Bootdisketteninfektionen wird ebenfalls die LED zum Blinken gebracht, nun aber deutlich länger wie während des Resets.

Zu Beginn des Botblocks kann man lesen: `EXCREMENT`

Der `EXCREMENT`-Bootblockvirus ist wie der `SCA`-Virus nicht 100% unter Kickstart 2.0 lauffähig, weil er das Vorliegen einer Bootdiskette

ebenfalls über die nur unter Kickstart 1.2/1.3 erfüllte Bedingung `cmpa.l 40(A1),A4` ermittelt. Unter Kickstart 2.0 wird also der `DoIO()`-Vektor nicht mehr restauriert, es wird also weiterhin die `Virus-DoIO()`-Verbiegung benutzt, welche aber in einem Absturz endet, weil der Bootblockviruscode 1028 Bytes kopiert und somit den `jmp ORG-DoIO()`-Befehl zerstört.

1.160 exterminator ii

Exterminator II

macht sich über `Cool` resetfest, nach nächstem Reset wird dann nach `SCA` -Manier kurzzeitig der `DoIO()`-Vektor verbogen, um die Bootdiskette zu checken, wenn ein bekannter Virus gefunden wird, dann wird automatisch der Antivirusbootblock geschrieben und danach erst der Alert, danach Softreset.

1.161 extreme

EXTREME

Es handelt sich um einen Bootblockvirus, welcher im Supervisorstack (`Boden+$1000`) steht.
Der EXTREME-Virus macht sich über `KickTagPtr` und `KickChecksum` resetfest. Diese Resetfestigkeit wird dadurch sichergestellt, daß bei jedem Vertikal-Blank-Interrupt diese Vektoren erneut auf den EXTREME-Virus gesetzt werden.
Als 'Kick-String' wird 'THE EXTREME ANTI-VIRUS HA HA !!!' verwendet. Weiterhin verbiegt der Virus den `DoIO()`-Vektor, wodurch Disketten bereits beim Einlegen infiziert werden können. Nach 3 Infektionen werden alle nicht schreibgeschuetzten Disketten schnellformatiert und danach ein Alert mit folgendem Text ausgegeben:

```
THE EXTREME ANTI-VIRUS  HA HA !!!  
BACK TO LIVE BACK TO REALITY  
SICO DE MOEL  BERGERWEG 100  
CALL 072-114816
```

Nach Betätigen einer Maustaste stürzt der Amiga ab.
Den obigen Text kann man auch im Bootblock selber lesen.

1.162 fast

FAST

Der FAST-Virus macht sich über den `COOL`-Vektor resetfest. Weiterhin wird der `DoIO()`- und `FreeMem()`-Vektor verbogen. Die Disketteninfektion erfolgt bei einem Format- oder Schreibzugriff auf den Bootblock. Beim Disketteneinlegen erfolgt also keine Infektion. Nach 16 Infektionen wird ein Alert ausgegeben.

1.163 fast 1

FAST 1

Der FAST1-Virus entspricht programmtechnisch dem FAST -Virus.

1.164 fasteddie

FastEddie

Es handelt sich um einen Bootblockvirus, welcher ab \$7F000 im Speicher steht. Er macht sich über KickTag, KickChecksum resetfest, wobei andere Kick-resetfeste Programme wie z.B. RAD: verloren gehen. Es werden kein Kick-Identifikationsstrings verwandt. Nach dem nächsten Reset wird der DoIO()-Vektor verbogen, um Disketten bereits beim Einlegen zu infizieren. Hierbei wird auch der Diskname in 'This disk is infected (HE-HE)' umbenannt. Weiterhin wird in einem zufälligen Diskettenblock ab Position 256 'Fast Eddie' geschrieben, wodurch Programme oder Daten unwiderbringlich beschädigt werden. Nach der ersten Disk-Infektion wird auch der Interrupt 3 verbogen, um nach 15 Minuten die Tastatur zu sperren und um nach 20 Minuten den Rechner endgültig zum Absturz zu bringen. Da der Virus mit einem Zufallswert verschlüsselt ist, kann man keine verdächtigen Texte im Bootblock lesen. Im Speicher steht der entschlüsselte Virus. Hier kann man Call 43-444304 and ask for HENRIK HANSEN (FAST EDDIE) lesen.

1.165 fasteddie-infector

FastEddie-Infector

Es handelt sich um einen FastEddie -Abkömmling. Es gibt lediglich formale Unterschiede, so wird nun als neuer Diskettenname '>INFECTOR BY DARK<' benutzt und Datenblöcke werden nun mit 'INFECTOR GO!' beschädigt. Nach Dekodierung des Virus ist im Speicher lesbar:

THIS IS THE FIRST VIRUS WRITTEN BY THE DARK AVENGER !!!

1.166 f.i.c.a

F.I.C.A

Der F.I.C.A - Virus kopiert sich an den Supervisorstack-Boden und macht sich Kick-kompatibel resetfest. Es wird der SumKickData- und TD_BeginIO()-Vektor verbogen. Es wird jede eingelegte Diskette infiziert, wobei immer ein normaler Bootblock vorgetäuscht wird.

1.167 forpib

Forpib

Es handelt sich um eine schlecht programmierte ByteBandit -Nachahmung.

1.168 frenchkiss

FrenchKiss

Es handelt sich um einen Bootblockvirus, welcher ab \$7f0d0 im Speicher steht. Zu den 1024-Bytes Bootblock-Code lädt der Virus noch die nächsten drei 256-Byte-Blöcke nach, so daß der FrenchKiss-Virus insgesamt aus 5 * 256-Byte-Blöcken besteht. Der Virus macht sich über den COOL-Vektor resetfest und verbiegt den DoIO()-Vektor, um Disketten direkt beim Einlegen zu infizieren. Weiterhin wird der Interrupt 3 und Exec-Interrupt 3 verändert. Der Virus kann auch die Root-Spur einer Diskette überschreiben.

1.169 freshmaker

FRESHMAKER

Es handelt sich um einen sehr unsauber programmierten Bootblockvirus, der wegen direkter ROM-Einsprünge nur unter Kick 1.3 läuft. Der FRESHMAKER-Bootblockvirus weist starke Ähnlichkeit mit dem MUTILATOR -Bootblockvirus auf und ist als dessen Vorläufer zu betrachten. Der FRESHMAKER steht immer ab \$78000 im Speicher und macht sich über den COOL-Vektor resetfest. Nach dem nächsten Reset wird ein eventuell in \$c000b0 stehender DOIO-Vektor verbogen, um Disketten bereits beim Einlegen infizieren zu können. Weiterhin wird auch der Supervisor-Vektor verbogen, um circa 50 * pro Sekunde die COOL-Resetfestigkeit und DOIO-Einschleifung des Virus sicherzustellen. Weiterhin wird der findresident()-Vektor verbogen, um nach 10 findresident()-Aufrufen den Rechner in eine endlose LED-Blinkroutine zu schicken. Nach 10 Disketteninfektionen gibt der Virus einen Alert mit folgender Meldung aus:

```
ES IST WIRKLICH NICHT ZU GLAUBEN
DU BOOTEST MIT EINER UNGESCHÜTZTEN DISK !
ES IST WIRKLICH SCHADE(!), DAß ES SOLCHE
LAMER (!) NOCH GIBT. DU HAST WOHL KEINE
ANGST VOR VIREN ??? ICH WÜNSCHE DIR
NOCH VIEL SPAß (!) MIT DEINEM AMIGA
UNTERZEICHNET: THE FRESHMAKER IN 1991 !
```

Nach Beenden dieses Alert wird ein Softreset ausgelöst. Den Text kann man im Bootblock nicht lesen, da er kodiert vorliegt.

Der FRESHMAKER geht davon auf, daß sogenanntes Ranger-memory ab \$C00000 vorhanden ist, denn der Virus schreibt starr die neue DOIO-Virusadresse, findresident-Adresse, supervisor-Adresse und COOL-Adresse in das Rangermemory, aber das hat nur dann einen Effekt

wenn Kickstart1.3 und Speicher ab \$C000000 vorhanden ist.

Da der FRESHMAKER nicht gezielt auf Disketten prüft, kann auch Zylinder 0 von Festplatten beschrieben werden, und die Festplatte dadurch nicht mehr ansprechbar sein.

1.170 frity

Frity

ByteBandit -Abkömmling.

1.171 fuck.device

fuck.device

Es handelt sich um einen unsauber programmierten Bootblockvirus, der bei Diskettenmanipulationen immer 1024 Bytes ab \$70000 überschreibt. Der Virus macht sich nur unter Kickstart 1.3 über den COOL-Vektor resetfest. Weiterhin wird auch unter Kickstart 1.2 oder 2.0 der DoIO()-Vektor verbogen, um Diskettenmanipulationen vorzunehmen. Es wird dann abwechselnd bei einer eingelegten Diskette entweder der Virusbootblock auf den Bootblock(Block0+1) geschrieben oder aber es werden 2048 Bytes auf Block 0,1,2,3 geschrieben. Diese 2048 Bytes bestehen meist aus 793 Bytes 'fuck.device', 0,0 und dann noch 1255 zufällige Bytes. Es resultiert eine 'Not A Dos-Disk' und dadurch daß auch Block 2,3 überschrieben wird, können Programme irreversibel beschädigt werden. Man kann in der Mitte des Bootblocks noch graphics.library, intuition.library, layers.library, timer.device, trackdisk.device, console.device und input.device lesen. Der Virus macht hiervon aber keinen Gebrauch.

1.172 fuck-lamer(ingo`s return)

FUCK-Lamer(INGO`S RETURN)

siehe Lamer-Bootblockviren .

1.173 future-disaster

Future-Disaster

Es handelt sich um einen nur unter Kickstart 1.2 lauffähigen Bootblockvirus. Für den Namen 'Future-Disaster' gibt es keinen ersichtlichen Grund. Der Virus steht ab \$7fb00 im Speicher und verbiegt vorübergehend den BeginIO()-Vektor des trackdisk.devices. Der Virus macht sich über den COOL-Vektor resetfest und infiziert Bootdisketten durch Verbiegen des DoIO()-Vektors.

Festplatten können unbrauchbar werden.
Nachdem 7 Bootdisketten infiziert wurden, wird die nächste Bootdiskette unbrauchbar gemacht, indem 5 Spuren einschließlich der Root-Spur überschrieben werden. Weiterhin wird der Bootblock mit zufälligen unsinnigen Daten überschrieben, wodurch eine NON-DOS-Disk resultiert. Den Großteil der Daten kann man jedoch meist mit z.B. disksalv retten.

1.174 gadaffi

Gadaffi

Der Gadaffi-Virus ist sehr stark dem DASA-ByteWarrior -Virus nachempfunden. Der Gadaffi-Virus unterscheidet sich hauptsächlich nur dadurch, daß er nach einigen Resets ein 'schwirrendes' StepperMotor-Geräusch erzeugt. Der schwerwiegende Programmierfehler des DASA-ByteWarrior-Virus ist also genauso im Gadaffi-Virus enthalten. Wie der DASA-ByteWarrior-Virus ist also auch der Gadaffi-Virus höchst gefährlich, da er z.B. Festplatten unbrauchbar machen kann. Der Gadaffi-Virus löscht den COLD-Vektor und macht sich über Cool und Kick-Vektoren resetfest. Er verbiegt den DoIO()-Vektor, um Disks schon beim Einlegen zu infizieren. Der Gadaffi-Virus nistet sich im vermeintlich sicheren Supervisorstack \$7e800-\$80000 ein, was jedoch nur für 512 KB-Amiga gilt. Ferner läuft er nur mit Kickstart 1.2, da er direkt ins ROM einspringt. Der größte Programmierfehler ist jedoch die fehlende Prüfung auf 'trackdisk.device', wodurch Festplatten sehr gefährdet sind. siehe Rigidiskblock beschädigen .

1.175 gandalf

Gandalf

Der Gandalf-Bootblockvirus weist eine sehr starke Verwandtschaft zu dem Angel -Virus auf, was sich auch in den gegenseitigen Grußbotschaften zeigt. Im Gegensatz zum Angel-Virus ist der Gandalf-Virus aber auch unter Kickstart 2.0 lauffähig, denn der Gandalf-Virus ist sauberer programmiert. Er spricht das Laufwerk nicht mehr direkt an, sondern benutzt hierzu die entsprechenden korrekten trackdisk.device-Befehle wie z.B. TD_PROSTATUS.

Nach 128 Disketteninfektionen

wird der Titel des aktiven Fensters und Schirms abgeändert auf
'Gandalf's Rache 1.5.90 - Ser.Nr. B00128 - Hi Butonic & Angel!'

Nach jeweils 512 Disketteninfektionen

wird der Titel des aktiven Fensters abgeändert auf
'Laufwerk DF0: IST ZERRSTÖRT, Tod für ALLE LAMER!'

Die Glaubwürdigkeit dieser Meldung wird dadurch unterstrichen, daß der Schreib-Lesekopf 5 mal mit viel Lärm die Diskette hoch undrunter gefahren wird. Die Diskette nimmt hierbei allerdings keinen Schaden, dennoch sieht es zunächst danach aus, da der Virus während der nächsten 6 Diskettenwechsel eine Benutzung der eingelegten Diskette verhindert. Das Betriebssystem zeigt Read/Write-Error-Requester an. Danach aber läßt der Virus wieder eine Benutzung des

Diskettenlaufwerkes zu und zeigt dies auch mit einer entsprechenden Fenstertiteländerung an:
'Glück gehapt LAMER!'

Die Titeländerungen werden von einem circa 3 Sekunden langen Bildschirmflackern(=DisplayBeep) begleitet.
Damit sich die Titeländerungen auch optisch zeigen, muß das entsprechende Fenster manipuliert werden.

Unter Kickstart 1.2/1.3 wird erst dann eine eingelegte Diskette infiziert, wenn ein weiterer Zugriff auf die Diskette erfolgt, wenn also z.B. ein Programm von der Diskette gestartet wird. Unter Kickstart 2.0 löst bereits das alleinige Disk-Einlegen die Infektion aus.

Der Gandalf-Virus verbiegt wie der Angel-Virus den PutMsg()-Vektor, um die Disketteninfektionen vorzunehmen.
Weiterhin verbiegt der Gandalf-Virus den ExitIntr()-Vektor, um dadurch die PutMsg()-Virus-einschleifung sicherzustellen. Der Angel-Virus benutzte hierzu den Wait()-Vektor.

Der Gandalf-Virus löscht den COLD-Vektor und macht sich über den COOL-Vektor resetfest. Sollten jedoch bereits COOL- oder KickTag-resetfeste Programme vorhanden sein, dann verändert der Gandalf-Virus NICHT die bereits benutzten Cool oder KickTag-Vektoren, sondern er ändert den Anfang der entsprechenden resetfesten Programme auf einen Sprung in das Gandalf-Virusprogramm ab. Dieses recht intelligente Reset-Verhalten des Gandalf-Virus gleicht ebenfalls dem Angel-Virus.

1.176 genestealer

GENESTEALER

Es handelt sich um einen Bootblockvirus, welcher ab \$7EC00 im Speicher steht und sich über den COOL-Vektor resetfest macht. Weiterhin wird der DoIO()-Vektor verbogen, um Disketten bereits beim Einlegen zu infizieren. Der aktive Virus versucht einen normalen Bootblock vorzutäuschen. Am Ende des Bootblocks kann man GENESTEALER VIRUS!!! by someone lesen. Wenn keine PAL-Auflösung vorliegt, wird die Root-Spur der Diskette beschädigt. Weiterhin können insbesondere die Systemdaten einer Festplatte beschädigt werden.

1.177 glasnost

Glasnost

Es handelt sich um einen Bootblockvirus. Er unterscheidet sich von den üblichen Bootblockviren dadurch, daß der eigentliche VirusCode in Block 2+3 der Diskette steht. Der Bootblock (Block 0+1) beinhaltet lediglich die Nachladeroutine der Daten von Block 2+3.
Der Bootblock (Block 0+1) sieht also relativ normal aus, wodurch sich der Virus also verstecken will. Bei vollen Disketten ist es durchaus möglich, daß auch Daten nach Block 2+3 geschrieben werden, lediglich auf Block 0+1

werden keine Datei-Daten geschrieben. Der Glasnost VIRUS kann also durch das Überschreiben von Block 2+3 Daten zerstören. Der Virus steht ab \$7f000 im Speicher und macht sich über KickTag, KickChecksum resetfest, wobei andere Kick-resetfeste Programme rausgeworfen werden. Nach einem Reset wird dann der DoIO()-Vektor verbogen, um jede eingelegte Diskette zu infizieren, das heißt Block 0,1,2,3 zu überschreiben. Danach wird ein zufälliger Disketten-Block ab Position 256 mit folgenden 4 Langworten beschrieben: \$11111111,\$22222222,\$44444444,\$88888888. Weiterhin wird nun auch der \$6c-Auto-Interrupt-Vektor verbogen, um nach 15 Minuten die Tastatur zu blockieren und um nach 20 Minuten den Rechner anzuhalten. Eine Festplatte kann, wenn auch eher selten, unbrauchbar werden. Im dritten Block einer infizierten Diskette kann man folgenden Text lesen: Glasnost VIRUS by Gorba!! First release

1.178 graffiti

Graffiti

Es handelt sich um einen unsauber programmierten Bootblockvirus, welcher ab \$7ec00 im Speicher steht. Der Graffiti-Virus macht sich über den COOL-Vektor resetfest. Er infiziert nur die Bootdiskette. Manchmal wird beim Booten der Bildschirm schwarz und es erscheint der folgende rote Text:

VIRUS! written by Graffiti

Diesen Text kann man auch am Ende des Bootblocks lesen. Über diesem Text dreht sich eine Vektorgrafik, welche aus drei großen roten Buchstaben besteht:

AMA

1.179 gremlin

GREMLIN

Es handelt sich um einen unsauber programmierten Bootblockvirus, welcher ab \$7f400 im Speicher steht. Der GREMLIN-Virus macht sich über den COOL-Vektor resetfest und verbiegt den DoIO()-Vektor, um jede eingelegte Diskette zu infizieren. Nach 16 Disk-Infektionen gibt sich der GREMLIN-Virus zu erkennen. Es wird für kurze Zeit das Wort GREMLIN in weißer Schrift in der Mitte eines roten Bildschirms angezeigt. Eine Festplatte kann durch den GREMLIN-Virus unbrauchbar werden. Ferner wird auch der SumKickData()-Vektor verbogen. Wenn man während des Resets die linke Maustaste drückt, wird der Virus entfernt. Nach spätestens circa 2 Stunden wird der Bildschirm schwarz geschaltet. Man kann nun zwar noch blind weiterarbeiten, ein Reset wird aber unausweichlich.

1.180 guardiansbootaids

GuardiansBootAids

Warsaw Avenger -Abkömmling.

1.181 gx.team

GX.TEAM

Es handelt sich hierbei um einen sehr unsauber programmierten Bootblockvirus. Er läuft nur mit Kickstart 1.2. Er macht sich über den COOL-Vektor resetfest. Außerdem verändert er die Kick-Vektoren, wodurch resetfeste Programme wie RAD: oder turboprint rausgeworfen werden. Weiterhin verbiegt der GX.TEAM-Virus den DoIO()-Vektor, um Disketten bereits beim Einlegen zu infizieren. Allerdings prüft der GX.TEAM-Virus nicht, ob sich der DoIO()-Zugriff auch wirklich auf das trackdisk.device bezieht, deshalb sind Festplatten sehr gefährdet. siehe Rigidiskblock beschädigen . Nach einem Reset wird auch der Interrupt 3 verbogen, wodurch dann circa 3 * in der Sekunde der Cool,Kick und DoIO()-Vektor auf den GX.TEAM-Virus verbogen werden. Nach jeweils 5 Resets und 4432 DoIO()-Aufrufen wird folgender Alert ausgegeben: (Die Wahrscheinlichkeit, daß der Alert erscheint, ist also recht gering)

Mais qui voilà ??C'est le nouveau VIRUS de GX.TEAM !!

WAAAHH! Les salauds! Les ... (Insultes diverses)

He!He! SILENCE :

GX.TEAM entre enfin dans la légende ...

BYE!!!

1.182 gyros

GYROS

Es handelt sich um einen Bootblockvirus, welcher ab \$7ec00 im Speicher steht. Er macht sich über den COOL-Vektor resetfest. Beim Reset wird dann der DoIO()-Vektor verbogen, um die Bootdiskette zu infizieren. Nach 10 Infektionen wird der Bildschirm schwarz, und der Rechner muß von Hand resettet werden, dabei wird dann eine GURU-Nummer angezeigt. Anscheinend ist der Virus noch nicht fertig programmiert, denn der Virus-Code läßt darauf schließen, daß anstatt des Rechnerabsturzes eine COPPER-Grafik geplant war. Im Bootblock kann man folgenden Text lesen:

Dear Arnd! Your Amiga is fucked from a nice GYROS.

Many greetings to you from Goebloidiel!!

1.183 hcs

HCS

Es handelt sich um einen Bootblockvirus, welcher ab \$7ec00 im Speicher

steht. Der HCS4200-Virus macht sich über den COOL-Vektor resetfest und überprüft beim Booten die Bootdiskette. Wenn ein Virus gefunden wird, dann wird dieser durch den HCS4220-Bootblock überschrieben. Da dieses automatisch erfolgt, muß man den HCS4220 als Virus betrachten. Sollte die Diskette schreibgeschützt sein, dann weist ein Alert auf den gefundenen Virus hin.

1.184 heil

HEIL

siehe SS -Bootblock. Es handelt sich um einen geringfügig 'weiterentwickelten' SS-Bootblock. So ist der Virus-Code stellenweise etwas optimiert, weiterhin wird der Viruscode mit einem AllocAbs() gesichert. Anstatt SS.greetings wird SS.greets verwandt. Der Programmierfehler am Ende der Copper-grafik ist behoben, dies ist aber ohne Belang, da die nun verwandte Endlosschleife ebenfalls nur mit einem Reset beendet werden kann. Der Bootblock ist anstatt mit !SS! mit HEIL kodiert.

1.185 hilly

HILLY

Der HILLY-Virus ist ein sehr bösartiger Bootblockvirus!! Er ist sehr unsauber programmiert. Deswegen läuft er auch nur unter Kickstart 1.2. Der Virus gibt nie eine Meldung oder ähnliches aus. Auch ansonsten gibt es keinen Grund für einen bestimmten Namen. Woher also der Name HILLY-Virus stammt, ist nicht zu ergründen. Wenn man erstmals von einer Diskette bootet, welche mit dem HILLY-Virus infiziert ist, dann macht sich der Virus über die Kick-Vektoren resetfest. Hierbei werden andere Kick-resetfeste Programme wie RAD: oder turboprint rausgeworfen. Beim nächsten Reset wird der Virus dann voll aktiviert. Es wird der DoIO()-Vektor verbogen und die Bootdiskette infiziert. Weiterhin wird der Exec-Vertikal-Blank-Interrupt verbogen, allerdings wird in der neuen Vertikal-Blank-Routine sofort zu der Original-Routine weitergesprungen. Wenn man eine Diskette einlegt, dann wird sie infiziert, das heißt, der Bootblock der eingelegten Diskette wird mit dem Virus-Bootblock überschrieben. Die große Bösartigkeit des HILLY-Virus liegt nun darin, daß bei der nächsten eingelegten Diskette nicht der Bootblock überschrieben wird, sondern es wird ein zufällig ermittelter 512 Bytes großer Block auf der Diskette überschrieben. Die nächste eingelegte Diskette wird dann wieder infiziert usw.

1.186 hoden v33.17

HODEN V33.17

Es handelt sich um einen Bootblockvirus, welcher ab \$7f000 im Speicher steht. Er macht sich über KickTag, KickChecksum resetfest. Hierbei gehen andere Kick-resetfeste Programme verloren. Nach dem nächsten Reset wird dann der DoIO()-Vektor verbogen, um jede eingelegte Diskette zu infizieren, das heißt es wird

der Bootblock mit dem Virus-Bootblock überschrieben.
Nach 5 Infektionen huscht ein kleiner lachender gelber Kopf mit Brille von links nach rechts über den Bildschirm.
Der Name des Virus rührt daher, daß man am Ende des Virus-Bootblocks HODEN V33.17 lesen kann. Der Virus funktioniert aufgrund direkter ROM-Einsprünge nur mit Kickstart 1.2.
Der Virus kann zum Verlust von Festplatten-Daten führen, da er nicht gezielt auf das trackdisk.device zugreift.

1.187 hulksters

HULKSTERS

PentagonCircleVirusSlayer -Abkömmling.

1.188 ice

ICE

SCA -Abkömmling.

1.189 incognito

Incognito

Es handelt sich um einen neuartig programmierten Bootblockvirus, welcher nur mit Kickstart 1.2 läuft. Der Name Incognito-Virus rührt daher, daß der Virus sich nicht durch irgendwelche auffälligen Erscheinungen zu erkennen gibt. Der Incognito-Virus macht sich über die Kick-Vektoren resetfest, hierbei werden andere Kick-resetfeste Programme wie z.B. RAD: entfernt. Als Name der Kick-Struktur wird ein zufälliges Zeichen benutzt. Weiterhin hängt der Incognito-Virus eine Routine in den Vertikal-Blank-Server. Dadurch werden 50 * in der Sekunde die Kick-Vektoren und der DoIO()-Vektor auf den Incognito-Virus gesetzt. Außerdem wird 50 * in der Sekunde geprüft, ob ein Laufwerk angelaufen ist. Wenn ja, wird der Bootblock der eingelegten Diskette überschrieben.

1.190 inger.iq.virus

Inger.IQ.Virus

ByteBandit -Abkömmling.

1.191 jinx

JINX

Es handelt sich um einen Bootblockvirus, der sich zufallsgesteuert über die Strahlenposition verschlüsselt und sich ebenfalls zufallsgesteuert an einen zufälligen Bereich im Supervisorstack kopiert. Der Virus macht sich unter dem Namen JINX kompatibel über die Kick-Vektoren resetfest und verbiegt weiterhin den SumKickData()-Vektor und hängt sich in den Exec-Interrupt-3-Vertikal-Blank-Server, um immer aktiv zu bleiben. Durch Verbiegen des TD-BeginIO-Vektors kann der Virus Disketten bereits beim Einlegen infizieren.

1.192 jitr

JITR

Es handelt sich um einen Bootblockvirus, welcher sich über den COOL-Vektor resetfest macht und kurzfristig während des Resets den DoIO()-Vektor verbiegt, um die Bootdiskette zu infizieren. Erfreulicherweise infiziert der JITR-Virus nur Disketten mit einem normalen Bootblock. Dadurch gehen also keine wichtigen Bootblöcke verloren. Man kann im JITR-Virus-Bootblock den folgenden typischen Text erkennen:

```
I'm a safe virus! Don't kill me! I want to travel!
And now a joke : ATARI ST
This virus is a product of JITR
```

1.193 joshua

JOSHUA

Es handelt sich um einen Bootblockvirus, welcher sich über den COLD-Vektor resetfest macht. Der JOSHUA verbiegt den BeginIO()-Vektor des trackdisk.devices, um Disketten bereits beim Einlegen zu infizieren. Der JOSHUA spiegelt – wie die Lamerviren – immer einen normalen Bootblock vor. Im Gegensatz zu den Lamerviren begnügt sich der JOSHUA – Virus aber mit dem Überschreiben des Bootblocks. Es werden also keine sonstigen Daten auf der Diskette überschrieben, wie dies bei den Lamerviren der Fall ist. Der JOSHUA hängt eine Routine in den Vertikal-Blank-Interrupt-Server ein. Diese trägt den Namen 'k.device' (kann man sich mit xoper oder artm mittels interrupts ansehen). Diese Routine setzt 50 * pro Sekunde den DoIO()-Vektor auf den Original-ROM-Wert. Nach 6 Resets und 6 Disketteninfektionen wird mit Hilfe dieser Routine 10 Minuten gewartet und dann ein Sprite diagonal über den Bildschirm bewegt. In dem Sprite kann man JOSHUA lesen. Der JOSHUA läuft nur mit Kickstart 1.2. Der JOSHUA ähnelt etwas den Lamer-Bootblockviren .

1.194 joshua 1

JOSHUA 1

Es bestehen lediglich folgende 2 Unterschiede zum ByteBandit -Virus:

1. Der JOSHUA1-Virus arbeitet ein wenig mit Kodiererroutinen um z.B. den 'trackdisk.device'-String zu verstecken.
2. beim ByteBandit-Virus erfolgt nach 7 min. eine Rechnerblockade, beim JOSHUA1 erscheint stattdessen nach 10 min für 50 sec. ein Sprite, in welchem man JOSHUA lesen kann.

Der JOSHUA1-Virus ist sehr stark dem ByteBandit-Virus nachprogrammiert.

1.195 julietick

JulieTick-PREDATOR

Es handelt sich um einen Bootblockvirus, das heißt, der Virus wird durch Booten von einer mit dem Julie-Tick-Virus infizierten Diskette aktiviert. Als erstes prüft der Julie-Tick-Virus, ob sich ein Programm über den KickTag-Vektor resetfest gemacht hat. Wenn ja, so wird der KickTag-Vektor gelöscht und ein Software-Reset ausgelöst. Manche Festplatten-Controller, wie z.B. ALF2, verändern allerdings immer völlig automatisch während eines Resets die Kick-Vektoren. Der Julie-Tick-Virus findet nun also immer wieder einen veränderten Kick-Vektor vor und löst daher wieder einen Software-Reset aus. Der Rechner ist also in einer Endlosschleife gefangen. Angenommen der KickTag-Vektor war nicht gesetzt bzw. konnte erfolgreich gelöscht werden, dann wird anschließend der DoIO()-Vektor verbogen und auch die Privilegsverletzung-Exception (\$20) wird verbogen. Diese Privilegsverletzung wird mindestens 50 mal in der Sekunde durch das Betriebssystem selber ausgelöst. Der Supervisor()-Befehl benutzt diese Exception, um in den Supervisor-State zu gelangen. Hier erfolgt dann das Umschalten der Tasks. Es wird also mindestens 50 mal in der Sekunde in das Virusprogramm gesprungen. Hier wird dann immer der COOL- und DoIO()-Vektor verbogen. Auch wird geprüft, ob die Bedingungen für das Abspielen einer kurzen Melodie erfüllt sind. Dies ist meist nach 6 Resets und 2 Disketten-Infektionen der Fall. Da der DoIO()-Vektor verbogen wurde, wird nun jede eingelegte Diskette infiziert. Hierbei wird auch noch der BeginIO()-Vektor des trackdisk.devices verbogen. Der Virus sollte eigentlich PREDATOR heißen, da am Ende des Virus folgender mit not.b verschlüsselter Text zu finden ist:

VIRUS PREDATOR (4-88-SPAIN) ID: 027798336

1.196 kako

KaKo

Der Name KaKo rührt daher, daß man zu Beginn des Bootblockes KaKo lesen kann. Es handelt sich um einen EXTREME -Abkömmling, der lediglich einen anderen Alert ausgibt, und zwar einen 245 Zeilen hohen Alert mit einigen nichtssagenden Punkten darin.

1.197 kauki

Kauki

Es handelt sich um einen unsauber programmierten Bootblockvirus, der nur unter Kickstart 1.2 läuft. Er macht sich über den COOL-Vektor resetfest. Während des Resets verbiegt er kurzzeitig den DoIO()-Vektor, um die Bootdiskette zu infizieren. Ansonsten werden keine Disketten infiziert. Wenn man von einer mit dem Kauki-Virus infizierten Diskette bootet, dann erscheint ein scrollendes rosafarbenes Gittermuster, über welchem sich der Schriftzug Kauki bewegt. Dieser Grafik-Effekt ist recht ansehnlich, ich rate aber insbesondere Festplattenbesitzern vom Experimentieren mit diesem Virus ab, da er aufgrund sehr nachlässiger Programmierung versehentlich auch Zylinder 0 der Festplatte überschreiben kann, wodurch dann die Festplatte meist unbrauchbar ist. siehe Rigiddiskblock beschädigen .

1.198 killed

Killed

Es handelt sich um einen Bootblockvirus, der ab \$7EC00 im Speicher steht. Der Virus macht sich über den COOL-Vektor restfest und verbiegt den DoIO()-Vektor, um Disketten zu infizieren. Wenn man während des Resets die linke Maustaste drückt, dann gibt sich der Virus zu erkennen, indem er in schwarzer Schrift den aktuellen Infektcounter auf einem weißen Hintergrund anzeigt. Nach erneutem Drucken der linken Maustaste wird mit dem Bootvorgang fortgefahren. Drückt man aber die rechte Maustaste, dann wird zuerst der Virus deaktiviert, was durch die Textausgabe Killed angezeigt wird und dann mit dem Booten fortgefahren. Am Ende des Bootblocks kann man folgenden Text lesen:

```
graphics.library dos.library Killed    Copy:029
```

1.199 l.a.d.s

L.A.D.S

Der Name rührt daher, daß im dritten Long-Word LADS steht. Dieser Bootblockvirus tarnt sich als Antivirus, indem er bei jedem Booten von einer infizierten Diskette folgende Alert-Meldung ausgibt:

```
L.A.D.S Virus Hunter
No virus in memory
Press any mouse button
```

Der L.A.D.S Virus macht sich über die Kick-Vektoren resetfest, wobei allerdings alle anderen Kick-resetfesten Programme wie z.B. RAD:, ALF oder turboprint usw. verlorengehen. Der Virus steht ab \$07f400 im Speicher. Beim Reset wird der DoIO()-Vektor verbogen, um die Bootdiskette und auch jede andere neu eingelegte Diskette zu infizieren. Festplatten sind nicht gefährdet, da sich der Virus nur in DoIO()-Zugriffe auf das trackdisk.device reinhängt. Der L.A.D.S Virus weist eine Besonderheit auf. Nachdem fünf

Disketten infiziert wurden, wird ein Inputhandler installiert, welcher die X/Y-Maus-Koordinaten invertiert. Die Maus verhält sich nun also gerade anders herum wie gewohnt. Nach acht Disketten-Infektionen gibt sich der Virus mit einem Alert zu erkennen:

AMIGA COMPUTING Presents:
The GREMLIN Virus
All Code (c) 1989 By Simon Rockman

Dieser Text liegt verschlüsselt im Bootblock vor und kann daher normalerweise nicht gelesen werden. Anstatt LADS-Virus sollte man den Virus also eher Gremlin-Virus nennen. Aber der Name Gremlin ist bei Viren-Programmierern recht beliebt, und ist gewissermaßen schon für andere Viren 'vergeben'. Dieser Bootblockvirus ist 'relativ' sauber programmiert und läuft daher auch unter Kickstart 2.0 auf dem Amiga 3000. Wie fast jeder Virus weist auch dieser Virus Programmierfehler auf. Angenommen der Maus-Koordinaten-Vertausch-Handler ist bereits aktiv. Wenn nun eine weitere Diskette infiziert wird, dann wird hierbei versehentlich noch einmal der Handler installiert. Da hierzu nochmals die gleiche Interrupt-Struktur benutzt wird, ist ein sofortiger Absturz unausweichlich.

1.200 l.a.d.s - a.i.d.s

L.A.D.S - A.I.D.S

L.A.D.S -Abkömmling.

1.201 lameblame-taipan(lameblame,cheater-hijacker,polish)

LameBlame-TaiPan (LameBlame,CHEATER-HIJACKER,POLISH)

Es handelt sich um einen Bootblockvirus, welcher sich über den COOL-Vektor resetfest macht. Weiterhin wird der DoIO()-Vektor verbogen, um Disketten beim Einlegen zu infizieren. Die Bootblöcke sehen immer verschieden aus, da sie mit einem Zufallswert (Rasterstrahl) kodiert werden.

Nach 8 Disketteninfektionen erscheint ein Alert mit z.B. folgendem Text:

-+= CHEATER HIJACKER =+- GENERATION 0004

oder es gibt auch Virus-Bootblöcke mit folgendem Text:

LameBlame! by Tai-Pan - Number of Copys : 0002

oder

POLISH P-1B IS RUNNING.. GENERATION : 0022

Der LameBlame-TaiPan-Bootblockvirus wird auch durch das Programm
VIRUS TERMINATORV6.0 installiert.

1.202 lamer-bootblockviren

Lamer-Bootblockviren

Alle Lamerviren sind sehr bösartige Viren, da sie mit voller Absicht Daten unwiederbringlich zerstören. Neben dem Überschreiben des Bootblocks beim Einlegen einer Diskette werden unter gewissen Umständen auch zufällige Disketten-Datenblöcke mit dem Wort 'LAMER!' oder 'Lamer!' überschrieben, wodurch Programme zerstört werden. Die Lamerviren arbeiten mit zufälligen KodierROUTINEN, wodurch jeder Bootblock ein anderes Aussehen erhält. Die Lamerviren machen sich über eine Kick-Struktur resetfest. Alle Lamer-Bootblockviren weisen den String 'The LAMER Exterminator !!!' in der Kick-Struktur auf. Dieser String ist jedoch nicht immer mit 0 abgeschlossen. Die Lamerviren machen sich 100% nur über die Kick-Vektoren resetfest. In den Kick-Resident-Strukturen ist manchmal kein MatchWord enthalten. Dies ist auch nicht nötig, da das Betriebssystem diese Strukturen sofort übernehmen kann und nicht erst zusammensuchen muß. Verschiedentlich wird behauptet, Lamerviren würden sich auf neuartige Weise über SumKickData() resetfest machen. Dies ist unmöglich!!! Lamerviren verbiegen den SumKickData()-Vektor nur deshalb, damit sie nicht bei der Installation eines anderen resetfesten Programms abgehängt werden. Bis jetzt sind nur Disketten gefährdet, da sich alle bisherigen Lamerviren in den BeginIO()-Vektor des trackdisk.devices einhängen. Hierdurch kann dann eine Diskette bereits beim Einlegen infiziert oder beschädigt werden. Für z.B. Festplatten besteht noch keine Gefahr. Lamerviren laufen auch mit Kickstart 1.3 und Fast-RAM. Alle Lamer-Bootblockviren täuschen anstelle des Lamer-Bootblocks einen Standard-Bootblock vor. Bei einem aktiven Lamer-Virus versagen also viele herkömmlichen Antivirusprogramme. (jedoch nicht VIRUS CONTROL).

```
Lamer!I      nicht Kick-kompatibel, Kennzeichen: dc.w $b118 ab Byte 72

Lamer!II.1   Kick-kompatibel, Kennzeichen: dc.w $d310 ab Byte 46
             überschreibt Datenblöcke mit Lamer! nach 2 Resets + 3 Infects

             Kick-kompatibel, Kennzeichen: dc.w $d310 ab Byte 70
             überschreibt Datenblöcke mit Lamer! nach 2 Resets + 3 Infects

             Kick-kompatibel, Kennzeichen: dc.w $b118 ab Byte 72
             überschreibt Datenblöcke mit LAMER! nach 6 Resets + 3 Infects

Lamer!II.2   Kick-kompatibel, Kennzeichen dc.l $b3beb2bb ab Byte 50
             FORMATIERT!! Disks nach 6 Resets+4 Infects,

             LamerII.2 auch von Hand an Programme gelinkt,
             LAMER-Trojan-Horse(endcli,loadwb,virusx)

Lamer!III    Kick-kompatibel, Kennzeichen: ab Byte 1012 $abcd
             überschreibt Datenblöcke mit Zufallswerten nach
             2 Resets+4 Infects, verschiebt Original-Bootblock von Block 0,1
             nach Block 2,3 wodurch der Virus lange unerkant bleibt, da
             weiterhin auch der Original-Bootblock ausgeführt wird.

Lamer!IV     Kick-kompatibel; neu ist, daß der BeginIO()-Vektor des
             timer.device verbogen wird, wodurch der Virus immer wieder
             installiert wird. Überschreibt Datenblöcke mit Lamer! nach
             2 Resets+3 Infects
```

CLIST-LAMER Der CLIST-LAMER-Virus gleicht sehr stark dem Lamer!IV-Virus.
Es werden allerdings keine zufälligen Datenblöcke
überschrieben. Als Kick-Identifikationsstring wird anstelle
des üblichen 'The LAMER Exterminator !!!' 'clist.library'
verwendet, so wie es auch beim Revenge of the Lamer
Exterminator-Filevirus der Fall ist.

FUCK-Lamer = INGO'S RETURN
Der FUCK-Lamer gleicht sehr stark dem Lamer!II.1-Virus.
Kennzeichen: dc.w \$abcd ab Byte 918
Er überschreibt allerdings nach 3 Resets + 1 Infect
die DatenBlöcke anstatt mit Lamer! oder LAMER! mit FUCK!!
und anstelle des typischen Kick-Identifikationsstrings
'The LAMER Exterminator !!!' wird '>>INGO'S RETURN << suffer!'
verwandt.

Neben diesen Lamer-Bootblockviren gibt es auch noch einen Lamer-Filevirus
namens RevengeOfTheLamerExterminator
und den ReturnOfTheLamer-Disk-Validatorvirus .

1.203 laureline v1.0

Laureline V1.0

Es handelt sich um einen Bootblockvirus, der ab \$6E800 im Speicher steht,
und den COLD und KickTag-Vektor löscht und sich als alleiniges Programm
über den COOL-Vektor resetfest macht. Nach dem nächsten Reset wird dann
der DOIO-Vektor verbogen, um die Bootdiskette zu infizieren.
Da der Virus die Bootdiskette ähnlich dem SCA-Virus zu infizieren versucht,
weist er also auch den Bootdisk-Bug auf, wodurch der Virus zwar unter
Kickstart2.0 auch resetfest ist und den DOIO-Vektor verbiegt,
eine Disketteninfektion bleibt aber ab Kickstart 2.0 in der Regel aus.
Nach mindestens 35 Disketteninfektionen und Drücken der rechten Maustaste
wird ein Alert mit z.B. folgendem vorher dekodierten Text ausgegeben.

The Laureline Virus V1.0

Code by Cat Lord

Report: 30.05.93

Sex: Male
Number of copy: 0002
Laureline Male Found: 0000
Laureline Female Found: 0000
Girl Maked: 0000
Disk Found: 0002
Dos Boot Found: 0000
Other Virus Founf: 0000
Amiga V1.2: 0000
Amiga V1.3: 0002
Amiga V2.0: 0000
Amiga V3.0: 0000

1.204 leviathan

LEVIATHAN

siehe LEVIATHAN-Bootblock+Filevirus .

1.205 little sven

Little Sven

Es handelt sich um einen Bootblockvirus, welcher auch mit Hilfe eines Trojanischen Pferdes (XCopyPro6.5, Filelänge 28336) verbreitet wird. Der Virus macht sich über den COOL-Vektor resetfest. Weiterhin wird des DisplayAlert()-Vektor, Supervisor()-Vektor und TD_BeginIO()-Vektor verändert, wodurch dann Disketten-Manipulationen möglich werden. Um eine neue Diskette zu infizieren verschiebt der Virus den Originalbootblock (1024 Bytes) von Block 0,1 nach Block 2,3. Bei vollen Disketten kann dadurch eine Datei unwiederbringlich beschädigt werden, weil ab einschließlich Block 2 Daten abgespeichert werden können. Der eigentliche Virus-Code wird dann auf den Bootblock (Block 0,1) geschrieben. Neben diesen Disk-Infektionen hängt sich der Virus auch in alle Disketten-Schreib- und Lese-Zugriffe und verschlüsselt hierbei die aktuellen Datenblöcke. Als Kennzeichen für einen verschlüsselten Datenblock ändert der Virus die Datenblockkennung \$00000008 in \$ABCD0008. Dadurch sind diese Daten nur noch bei aktivem Virus benutzbar. Der Name Little Sven rührt daher, daß man im Speicher, wo der Virus entschlüsselt vorliegt, 'The Curse of Little Sven!' lesen kann. VIRUS CONTROL kann die verschlüsselten Daten wiederherstellen.

1.206 loverboy&sexmachine

Loverboy&Sexmachine

16Bit Crew -Abkömmling.

1.207 lsd

LSD

SCA -Abkömmling.

1.208 mad

MAD

ByteBandit -Abkömmling.

1.209 mad ii

MAD II

Es handelt sich um einen unsauber programmierten Bootblockvirus, welcher aufgrund nachlässiger Programmierung auch Festplatten schädigen kann. Weiterhin funktioniert er nur unter Kickstart 1.2. Der MAD-Virus steht ab \$07fb00 im Speicher und verbiegt den DoIO()-Vektor, um jede eingelegte Diskette zu infizieren. Er löscht den COLD-Vektor und macht sich über Cool,Kick resetfest. Der MAD-Virus ist ein Gadaffi -Abkömmling, allerdings sind hier die NOPS in der Stepper-Motor-Geräusch-Routine wohl aus Versehen (???) von \$4e71 auf \$4d71 (gibts nicht) abgeändert, wodurch es zum Absturz kommt, wenn diese Routine aufgerufen wird. Er erscheint dann lediglich ein schwarzer Bildschirm.

1.210 mad iii

MAD III

fehlerhafter DASA-ByteWarrior -Abkömmling, da DASA0.2 in MADIII geändert wurde. DASA0.2 ist aber sinnvoller Code, der nicht verändert werden darf.

1.211 mad iv

MAD IV

Es handelt sich nicht um einen Lamer-Bootblockvirus .

1.212 megamaster

MEGAMASTER

Es handelt sich um einen unsauber programmierten Bootblockvirus, welcher ab \$7e300 im Speicher steht. Der MEGAMASTER-Virus macht sich über den COOL-Vektor resetfest und verbiegt den DoIO()-Vektor, um die Bootdiskette zu infizieren. Weiterhin wird manchmal beim Reset folgende Meldung ausgegeben:

Surprise!!! Your Amiga is controlled by MEGAMASTER

1.213 metamorphosis1.0

METAMORPHOSISV1.0

siehe METAMORPHOSISV1.0-Bootblock+Linkvirus .

1.214 mexx

MEXX

SCA -Abkömmling.

1.215 mg's virus v1.0

MG's Virus V1.0

Es handelt sich um einen Bootblockvirus, welcher sich selbst verschlüsselt. Deshalb funktioniert der Virus auch nicht auf höheren Prozessoren, da die Virus-Dekodierroutine sich nicht mit einem größeren Prozessor-Prefetch verträgt. Der Virus macht sich alleinig über KickTag, KickChecksum resetfest. Als Identifikationsstring wird MG's Virus V1.0 verwendet. Der COOL-Vektor wird gelöscht. Weiterhin wird in den Exec-Vertikal-Blank-Interruptserver eine Routine eingehängt, welche permanent die Aktivität des Virus sicherstellt. Auch durch Verbiegen des SumKickData()-Vektors und GetMsg()-Vektors stellt der Virus sein Aktivität sicher. Die eingehängte Interruptservernode trägt keinen Namen und hat die Priorität -100. Durch Verbiegen des BeginIO()-Vektors des trackdisk.devices können Disketten bereits beim Einlegen infiziert werden. Der Virus versucht anstelle des Virus-Bootblocks einen normalen Bootblock vorzutäuschen.

1.216 micromaster

MicroMaster

SCA -Abkömmling.

1.217 microsystems

MICROSYSTEMS

Es handelt sich um einen Bootblockvirus, welcher ab \$7f400 im Speicher steht. Der Virus arbeitet nur mit Kickstart 1.2 und verbiegt Cold, Cool, AddTask, RemTask und DoIO. Im Bootblock kann man folgenden Text lesen:

YOUR AMIGA IS INFECTED BY A NEW GENERATION OF VIRUS
CREATED IN SWEDEN BY MICROSYSTEMS

1.218 morbid.angel.virus

Morbid.Angel.Virus

ByteBandit -Abkömmling.

1.219 mosh

MOSH

Es handelt sich um einen Vorläufer des SS und HEIL - Bootblocks. Der MOSH-Bootblock ist mit dem Wert 'MOSH' verschlüsselt, wodurch folgender Text im Bootblock nicht mehr lesbar ist:

```
MAFIA dos.library graphics.library
This is a new DR.MOSH (MBI) production
Contact us at PLK 098107 A, 2380 Schleswig,
Deutschland
Ah, one question! Why can you read this?
My decode routine is simple, isn't it?
spread the word and the program (hehe)!
```

Der MOSH-Bootblock steht ab \$7c000 im Speicher und macht sich über den COOL-Vektor resetfest. Nach dem nächsten Reset wird der Autointerrupt2(\$68) verbogen, um nach circa 200 Tastaturbetätigungen mit gleichzeitig gedrückter linker Maustaste eine Coppergrafik auszugeben, welche nur mit einem Reset beendet werden kann.

Es erscheint ein gelber Text auf schwarzem Hintergrund:

MAFIA Contact us at PLK 098107 A, 2380 Schleswig, MOSH

Da keine neuen Disketten infiziert werden, kann man nicht direkt von einem Virus sprechen, aber das Erzwingen eines Resets nach einer gewissen Zeit ist ärgerlich genug.

1.220 mosh 2

MOSH 2

Wie MOSH, nur daß auch OldOpenLibrary() verbogen wird.

Bei jedem Aufruf von OldOpenLibrary() wird nun ein Alert mit folgendem Text ausgegeben:

```
Hey you old lame! Are you sure, what you are doing?
```

Nach Drücken der rechten oder linken Maustaste kann normal weitergearbeitet werden.

1.221 mount-eleni

Mount-ELENI-WIRUS

Es handelt sich um einen Bootblockvirus, der starr ab \$7F400 im Speicher steht, bzw. ab 1 MB Chip-Mem starr ab \$FE000. Der Virus macht sich über den COOL-Vektor resetfest und verbiegt den SumKickData und DOIO-Vektor um Disketten infizieren zu können, was bei einem DOIO-Lese-Zugriff ab Block 0 erfolgt, da aber nicht explizit auf Disketten gepüft wird, kann der RigidDiskblock von Festplatten Schaden nehmen.

Wenn ein DOIO-Schreibzugriff erfolgt, dann wird geprüft ob in \$f80000 \$1111 steht, was bei 256 KB großen Kickstarts 1.2,1.3 der Fall ist, wo

\$fc0000 nach \$f80000 gespiegelt wird, bei 512 KB-Kickstart, welche ab \$f80000 beginnen ist in \$f80000 \$1114 zu lesen. Unter Kickstart 1.2/1.3 wird also nichts weiter unternommen, weiterhin wird geprüft ab in \$08193Fd8 \$99 steht, denn der Virus versucht zuvor \$99 nach \$08193Fd8 zu schreiben, was aber nur bei wenigen Amigas der Fall sein wird, auf 68000-Amigas oder nur Chip-Mem-Amigas kann diese Fast-Mem-Adresse nie gültig sein. Aber angenommen diese Fast-Mem-Adresse existiert und mindestens Kick2.0 liegt vor, dann wird versucht innerhalb des DOIIO-Schreibzugriffs ein neues nur 208 Byte langes C/Mount-File zu schreiben, und ein 1024 Byte langes C/D-File, in welchem die 1024 Bytes des Bootblockvirus abgespeichert werden. Das C/Mount-File wird meistens beim Abarbeiten der startup-sequence aufgerufen und versucht dann aus SYS:C/D 1024 Bytes nach \$7f000 zu lesen. Im Viruscode ist FMFOJ XJSVT zu lesen, was nach Subtraktion mit 1 ELENI WIRUS ergibt. Programmtechnisch sind gewisse Ähnlichkeiten mit dem ELENI!-Bootblockvirus zu erkennen.

1.222 mutilator

MUTILATOR

Es handelt sich um einen sehr unsauber programmierten Bootblockvirus, der wegen direkter ROM-Einsprünge nur unter Kickstart 1.3 läuft. Weiterhin geht der Virus davon auf, daß sogenanntes Ranger-RAM ab \$C00000 vorhanden ist, denn der Virus schreibt starr eine neue DoIO()-Virusadresse nach \$c000b0 und eine neue Supervisor()-Virusadresse nach \$c0025a, aber nur wenn Speicher ab \$C00000 vorhanden ist, wird die execbase hier angelegt, so daß an den Adressen \$c0025a und \$c000b0 in der Tat der DoIO()-Vektor und Supervisor()-Vektor steht, so daß also nur bei vorhandenem Speicher ab \$C00000 der Virus erfolgreich den DoIO()-Vektor verbiegen und Disketten infizieren kann. Sollte, wie es meist der Fall ist, kein Speicher ab \$C00000 vorhanden sein, dann kann der Virus den DoIO()-Vektor nicht verbiegen, da der Virus ins Leere schreibt.

Der Virus steht immer ab \$07cad0 im Speicher und macht sich über den COOL-Vektor resetfest. Der COLD-Vektor und die Kick-Vektoren werden gelöscht. Nach dem nächsten Reset wird ein eventuell in \$c000b0 stehender DoIO()-Vektor verbogen, um Disketten bereits beim Einlegen infizieren zu können. Weiterhin wird auch der Supervisor()-Vektor verbogen, um circa 50 * pro Sekunde die COOL-Resetfestigkeit des Virus sicherzustellen. Nach circa 18 Minuten hält der Virus den Rechner an. Es erscheint eine einfache Coppergrafik (blaugrüner Balken auf schwarzem Hintergrund) und die Recher-LED wird permanent an und aus geschaltet. Ein Ende ist nur über Reset möglich.

Nach 3 Disketteninfektionen gibt der Virus einen Alert mit folgender Meldung aus:

```
THIS IS THE NEW MUTILATOR-VIRUS !  
BY MAX OF STARLIGHT
```

Nach Beenden dieses Alert wird ein Reset ausgelöst. Den Text kann man im Bootblock nicht lesen, da er kodiert vorliegt. Desweiteren steht noch folgender Text in dem Viruscode, der aber nie angezeigt wird:

Thanx to The Executors for Spreading this GREAT code ! done: - 1992 -

1.223 nasty

Nasty

Es handelt sich um einen Bootblockvirus, welcher wegen der Benutzung absoluter ROM-Adressen nur unter Kickstart 1.2 läuft. Es steht ab \$7F000 im Speicher und macht sich über den COOL-Vektor resetfest. Weiterhin wird der DoIO()-Vektor verbogen, um Disketten bereits beim Einlegen zu infizieren. Der Virus kann auch insbesondere die Verwaltungsdaten einer Festplatte überschreiben. Der Virus verbiegt auch den Alert()-, Superstate()- und Userstate()-Vektor, um dadurch permanent die Vektoren zu verbiegen. Am Ende des Bootblocks kann man Nasty-Nasty! lesen. Nach 5 Disketteninfektionen wird die jeweilige Diskette formatiert.

1.224 no.bandit.any.more

No.Bandit.any.More

ByteBandit -Abkömmling.

1.225 obelisk

Obelisk

Kopiert sich an Supervisorstackboden. Macht sich über COOL-Vektor resetfest. Infiziert nur BootDisk. Es wird vor dem Booten eine Grafikmeldung angezeigt: deutsche Flagge mit dem Text:
OBELISK CRACKING CREW Kickstart 1.3 + Fast-RAM kompatibel.

1.226 obelisk ii

Obelisk II

Lediglich der im Virus-Code enthaltene `move.l #'GURU',$0060` und der abgeänderte Text deuten auf den OBELISK CREW VIRUS. Aber ansonsten entspricht der Virus vollkommen dem OPAPA-Virus. Der Obelisk II ist also programmtechnisch gesehen ein OPAPA -Virus.

1.227 opapa

OPAPA

Als Vorlage wurde der ByteBandit -Virus verwendet. Der OPAPA-Virus unterscheidet sich von seinem Vorbild hauptsächlich in folgenden Punkten:

Der ByteBandit-Virus gibt sich durch folgenden Text zu Beginn des Bootblocks zu erkennen:

```
Virus by Byte   Bandit in  9.87.Number of          copys :
```

Der OPAPA-Virus gibt sich durch folgenden Text mehr am Ende des Bootblocks zu erkennen:

```
I'M THE OPAPA-VIRUS!..d.READY...STEADYx..FORMAT!
```

Beim ByteBandit-Virus wurde nach 7 Minuten der Rechner blockiert. Auf dem Bildschirm wurde dann nur noch die Hintergrundfarbe angezeigt. Diese Blockade tritt aber nur dann auf, wenn der Rechner mindestens 2 * resettet wurde UND mindestens 6 Disketten infiziert wurden. Man kann die Blockade durch folgende Tastenkombination aufheben: L-ALT L-AMIGA SPACE R-AMIGA R-ALT

Beim OPAPA-Virus wird nach 8 Minuten der Bildschirm schwarz und es wird die folgende Meldung in gelber Schrift ausgegeben:

```
I'M THE OPAPA-VIRUS!
```

```
READY
STEADY
FORMAT!
```

Weiterhin wird nun der Steppermotor des Diskettenlaufwerks bewegt. Dies hört sich dann so an, als ob die Diskette formatiert würde, zusammen mit der Bildschirmmeldung, welche ja auf das Formatieren hinweist, kann einem dieses einen ordentlichen Schrecken versetzen, aber erfreulicherweise wird nur der Steppermotor bewegt, es werden keine Daten geschrieben, die Diskette nimmt also keinen Schaden. Man kommt aus diesem schwarzen Bildschirm ohne Reset nicht mehr heraus. Diese OPAPA-Virusmeldung tritt aber nur auf, wenn der OPAPA-Virus mindestens 9 * installiert wurde UND wenn mindestens 8 Disketten infiziert wurden.

1.228 overkill

Overkill

Es handelt sich um einen Bootblockvirus, der ab \$7f000 Speicher belegt. Der Virus macht sich über den COOL-Vektor resetfest und verbiegt den DoIO()-Vektor, um Disketten bereits beim Einlegen infizieren zu können. Der Virus belegt 2048 Bytes (Block 0,1,2,3) auf Diskette, wodurch also bei vollen Disketten die Dateien beschädigt werden, die Daten auf Block 2,3 belegen, denn nur Block 0,1 ist automatisch für den Bootblock reserviert. Es ist auch umgekehrt denkbar, daß bereits eine infizierte Diskette vorliegt und daß später weitere Dateien auf die Diskette abgespeichert werden, wobei der ursprüngliche Original-Bootblock, der auf Block 2,3 verschoben wurde, überschrieben

werden kann, wodurch es dann beim Booten von dieser Diskette zum Absturz kommt, da diese Original-Bootblock-Daten zum erfolgreichen Booten benötigt werden.

Da die Virusinfektionsroutine etwas nachlässig programmiert ist, wird irrtümlicherweise auch versucht Festplatten beim Bootversuch zu infizieren, wodurch die Rigid-Disk-Daten auf Zyinder 0 beschädigt werden. siehe Rigidiskblock beschädigen .

Der übliche Disketteninfektionsablauf sieht folgendermaßen aus: Nachdem eine beschreibbare Disk eingelegt wurde, werden ab einer zufälligen Diskettenposition (abgeleitet von Strahlenposition) 1024 nicht verschlüsselte Virusdatenbytes (ab Speicherstelle \$07f700) geschrieben. Ab Blockposition 34 kann man dann wie auch im Speicher ab \$7f722 z.B. folgenden Text erkennen:

Overkill by the ENEMY !

Diese teilweise und zufällige Diskettendatenbeschädigung erfolgt bei jeder eingelegten Diskette. Die eigentliche Disketteninfektion, also das Kopieren des Viruscodes auf den Bootblock, wird anschließend vorgenommen, allerdings nur, wenn die Diskette nicht schon bereits infiziert ist. Es wird dann der 1024 Byte lange Original-BB von Block 0,1 nach \$7f000 eingelesen und danach unverändert auf Block 2,3 der Diskette geschrieben. Danach wird der Viruscode von \$07f700 nach \$07fb40 kopiert, dort mit der zufälligen Strahlenposition verschlüsselt, und dann als neuer Bootblock von \$07fb00 nach Block 0,1 geschrieben.

Der Overkill-Virus löscht den COLD-Vektor und verbiegt den SumKickData()-Vektor um hierbei die COOL-Resetfestigkeit sicherzustellen. Geplant war wohl auch eine zusätzliche Kick-Resetfestigkeit, denn die entsprechenden Strukturen sind in dem Virus-Code ab \$07f700 vorhanden, aber außer überflüssigen und harmlosen Neuberechnungen der KickCheckSumme wird nichts unternommen.

Ab Kickstart 2.0 werden beim Disketteninfizierungsversuch aufgrund des Drivebit-Bug komplette Diskettenspuren (bei DD-Disks 11 Sektoren) unbrauchbar gemacht (No Sector Header).

1.229 paradox i

PARADOX I

Dieser Bootblockvirus ist stark mit dem BlowJob -Bootblockvirus verwandt. Es bestehen folgende Unterschiede:

Beim BlowJob-Virus kann man zu Beginn des Bootblocks folgenden Text lesen:

[0.5] [1] [1.8] MB Memory Allocator 3.01

Beim PARADOX I steht statt dessen in der Mitte des Bootblocks:

A new age of virus-production has begun
This time PARADOX brings you the 'LOGIC BOMB' Virus !!!

Man erkennt also schon an dem Text, daß ein Virus vorliegt, es handelt sich aber um einen normalen Bootblockvirus, die Bezeichnung 'LOGIC BOMB' soll

wohl nur den Anwender erschrecken. Der BlowJob-Virus gibt sich nach 10 Minuten mit folgender Alert-Meldung zu erkennen:

```
ONCE AGAIN SOMETHING WONDERFUL HAPPENED  (HE HE HE)
PLEASE POWER OFF - PLEASE POWER OFF - PLEASE POWER OFF
```

Der PARADOX I - Virus hingegen gibt keine Meldung aus, sondern hält den Rechner direkt nach 10 Minuten an.

1.230 paradox ii

PARADOX II

Dieser Bootblockvirus steht programmtechnisch zwischen dem BlowJob und dem ByteVoyager . An den weiterentwickelten ByteVoyager erinnert die komplette Bootblock-Verschlüsselung. Es erfolgt aber noch keine Umbenennung des Diskettennamens. Im Gegensatz zum BlowJob erfolgt aber kein Alert, sondern der Rechner wird einfach nach 40 Minuten angehalten. Im dekodierten Bootblock kann man folgenden Text lesen:

```
This is the second VIRUS by PARADOX
- For swapping call: 42-455416 -
  ask for Henrik Hansen  ..
```

1.231 paramount

PARAMOUNT

Es handelt sich um einen 'defekten' DASA-ByteWarrior -Bootblockvirus. Es wurde lediglich zusätzlich der folgende Text an das Ende des Bootblocks geschrieben:

```
PARAMOUNT SOFTWARES CREW 1988 :
Greetings to : Napoleon ,Obelisk, Idefix, Asterix Hamburg ...
```

Weiterhin wurde willkürlich 'PSCW !!',10,0 in den Bootblock-Code geschrieben. Dadurch ist der Virus nicht mehr lauffähig. Ein Booten von der Diskette ist nicht mehr möglich. Es erfolgt immer eine GURU-Meldung.

Der PARAMOUNT-Virus kam also nicht als eigenständiger Virus betrachtet werden, denn hier hat lediglich jemand ein wenig herumgepfuscht.

1.232 paratax i

PARATAX I

SCA -Abkömmling.

1.233 paratax ii

PARATAX II

Dieser Bootblockvirus ist praktisch mit dem Crackright (DiskDoktors) identisch. Es bestehen lediglich Text-Unterschiede. So wurde z.B. 'clipboard.device' durch 'dos.library' ersetzt.

1.234 paratax iii

PARATAX III

16Bit Crew -Abkömmling.

1.235 pentagonslayer

PentagonCircleVirusSlayer

als Antivirus gedacht. schlecht programmiert, da feste Adressen benutzt. überschreibt nur Bootdiskette und zwar nur dann, wenn ein Virus darauf identifiziert wurde.

1.236 perverse i

PERVERSE I

Es handelt sich um einen neuartigen Bootblockvirus, welcher aufgrund recht sauberer Programmierung auch z.B. unter Kickstart 2.0 läuft.

Der PERVERSE I - Virus macht sich Kick-kompatibel resetfest.

Als Identifikationsstring wird der folgende Text verwandt:

BootX-Viruskiller by P.Stuer

Damit soll der User getäuscht werden, denn es handelt sich keineswegs um einen Viruskiller, sondern um einen Virus.

Es wird ein Inputhandler installiert. Dieser hat zwei Aufgaben.

Erstens: wenn eine Diskette eingelegt wurde, dann schreibt sich der

Virus immer auf den Bootblock von DF0:

Zweitens: Nach 10 Minuten wird die Tastatureingabe zu einer Textausgabe umfunktioniert. Es erscheint der folgende Text:

```
SOFTWARE_PIRATES RUINED MY EXCELLENT PROFESSIONAL DTP_PROGRAM
NOW I REVENGE MYSELF ON THESE IDIOTS BY PROGRAMMING VIRUSES
THIS IS PERVERSE I, BECAUSE I LIKE ASSHOLE_FUCKING
I PROGRAM VIRUSES FOR MS_DOS TOO
```

Der Virusprogrammierer scheint ein frustrierter kommerzieller Programmierer zu sein, der verärgert darüber ist, daß sich sein Programm aufgrund Raubkopiererei schlecht verkauft hat.

Deshalb will er sich nun anscheinend mit der Programmierung von Viren rächen.

1.237 powerbomb

PowerBomb

ByteBandit -Abkömmling.

1.238 powerteam

PowerTeam

Programmtechnisch ähnelt der PowerTeam-Virus etwas dem Sonja -Virus. So wie dieser kommuniziert auch der PowerTeam-Virus nicht korrekt mit dem trackdisk.device, so daß ein Virusinstallationsversuch nur unter Kickstart 1.2/1.3 gelingt. Der Virus macht sich kompatibel über die Kick-Vektoren resetfest. Als Kick-Identifikationsstring wird 'PowerTeam' verwandt. Eventuelle sonstige resetfeste Programme bleiben erhalten. Zum Infizieren von Disketten verbiegt der Virus den TD_BeginIO()-Vektor des trackdisk.devices. Ein aktiver PowerTeam-Virus versucht anderen Programmen einen normalen Bootblock anstelle des Virusbootblocks vorzugaukeln. Nach circa 10 Disketteninfektionen gibt sich der Virus durch folgenden optischen Effekt zu erkennen: Es wird ein neuer Schirm geöffnet und die Workbench-Grafikdaten in diesen Schirm kopiert und zwar so, daß der Workbench-Schirminhalt nach links heruntergekippt aussieht. Der eigentliche Workbench-Schirm hingegen wird unsichtbar nach hinten geschaltet. Anschließend wird noch ein Alert mit folgendem Text angezeigt:

Virus Meditation #0026051990.PowerTeam

der Virus wurde also am 26.05.1990 fertiggestellt. Man kann diese Textmeldung nicht im Bootblock erkennen, da sie kodiert vorliegt. Nach Betätigen einer Maustaste wird wieder der Original-Workbenchschirm nach vorne geholt. Optisch sieht das dann so aus, als ob sich der links abgerutschte Workbench-Schirm wieder korrekt aufrichten würde. Im Endeffekt wird also für die Dauer des Alerts der Anschein erweckt, als ob der Workbench-Schirm durch Viruseinwirkung etwas nach links unten asymmetrisch verkantet worden wäre.

1.239 pvl

PVL

Es handelt sich um einen System3.0 -Bootblock, bei welchem jedoch die Viruswarnmeldungen in Okay-Meldungen verändert wurden, wodurch also der User trotz Virenbefall in Sicherheit gewogen werden soll. Aber soweit kommt es erst gar nicht, denn das Abarbeiten des Bootblocks wird sofort durch einen RTS-Befehl beendet, wodurch erstens der Virus nicht aktiviert wird und zweitens der Amiga abstürzen muß, weil auch die Vorbereitungen zum Öffnen der dos.library nicht unternommen werden.

Warning: This disk is infected with the ByteBandit-Virus!
Warning: This disk is infected by the SCA-Virus

wurde verändert in

This Disk Is not Infected By Any Kind Of Virus Its Okay !
This disk is okay continue with your working

1.240 revenge bootloader

Revenge Bootloader

Es handelt sich praktisch um den ByteBandit -Bootblockvirus. Es wurde lediglich der typische ByteBandit-Text abgeändert. Im Original-ByteBandit ist zu Beginn des Bootblocks folgender Text zu lesen:

Virus by Byte Bandit in 9.87.Number of copys :

Anstatt dieses Textes ist beim Revenge Bootloader folgender Text zu lesen:

Revenge Bootloader!

Der Name Revenge Bootloader ist wohl mit Absicht irreführend gewählt, denn es handelt sich keineswegs um einen harmlosen Bootloader, sondern es liegt vielmehr ein gefährlicher Virus vor.

1.241 revenge

Revenge

Der Revenge-Virus ist ein eigenständiger Virus und weist keine Ähnlichkeiten mit dem ByteBandit -Virus auf, wie manchmal behauptet wird. Der Revenge-Virus ist deutlich schlechter wie der ByteBandit-Virus programmiert, da er z.B. absolute Adressen benutzt. Der Revenge-Virus macht sich über den COOL-Vektor resetfest und infiziert Disketten durch Verbiegen des DoIO()-Vektors. Ferner wird auch der Exec-IRQ-3-Interrupt verbogen, um nach einer gewissen Zeit den Mauszeiger in ein Phallus-Symbol zu verwandeln, vorausgesetzt der Virus wurde bereits 5 * installiert. Im unteren Teil des Bootblocks kann man Revenge V1.2 lesen. Es sind nun auch einige Revenge-Virus-Abkömmlinge im Umlauf, bei denen lediglich dieser Text abgeändert wurde. So z.B. beim Sendarian -Virus. Dieser ist also bis auf den Text identisch mit dem Revenge-Virus.

1.242 ripper

Ripper

Starfire-Northstar -Abkömmling

1.243 riska

Riska

ByteBandit -Abkömmling.

1.244 sachsen no.1

SACHSEN VIRUS NO.1

Es handelt sich um einen Bootblockvirus, der nur unter Kickstart 1.3 läuft, da eine feste ROM-Adresse für den CloseDevice()-Vektor gesetzt wird. In dem Bootblock ist der folgende Text kodiert, also nicht lesbar vorhanden:

```
HI USER!!! I CONTROL YOUR SYSTEM!!! WHY ???  
** SACHSEN VIRUS NO.1 ** IS IN MEMORY -HAVE FUN...  
SHORT CODE BY: S. OF ALPHABIT (DATE: 14/07/91)
```

Der Virus steht ab \$78000 im Speicher und macht sich über den COOL-Vektor resetfest. Bei jedem Reset werden weitere 40960 Byte Chip-RAM belegt, jedoch nur wenn eine neue Diskette infiziert wurde. Wenn also die Bootdisk bereits infiziert war, dann wird nur 40960 Byte Chip-RAM vergeudet. Der Virus infiziert nur die Bootdiskette und zwar unter Zuhilfenahme des CloseDevice()-Vektors. Festplatten sind nicht gefährdet.

1.245 sachsen no.3

SACHSEN VIRUS NO.3

Es handelt sich um einen Bootblockvirus, welcher ab \$78000 in Speicher steht. Er macht sich über den COOL-Vektor resetfest und löscht den Cold, KickMem, KickTag und KickChecksum()-Vektor. Sollten diese Vektoren nicht Null gewesen sein, dann wird ein COLD-Reset ausgelöst, das heißt ein Reset, bei welchem alle resetfesten Programme verloren gehen. Dadurch wird z.B. die resetfeste RAM-Disk RAD: zerstört. Der Virus verbiegt weiterhin den Wait()-Vektor, um permanent seine Aktivierung sicherzustellen. Außerdem wird der DoIO()-Vektor verbogen, um Disketten bereits beim Einlegen zu infizieren. Hierbei werden die ersten vier 512-Byte-Blöcke der Diskette überschrieben. Da sich der Virus also nicht auf das Überschreiben des Bootblocks (die ersten zwei 512-Byte-Blöcke) beschränkt, sondern auch noch die nächsten zwei 512-Byte-Blöcke überschreibt, können dadurch insbesondere bei recht vollen Disketten Dateien beschädigt werden. Noch größer ist aber der Schaden, den der Virus durch Überschreiben eines zufälligen 512-Byte-Blockes mit SACHSEN3 verursacht. Weiterhin benennt der Virus Disketten in 'SACHSEN NO.3 ON DISK !!!' um. Außerdem wird nach einer gewissen Anzahl von Disketteninfektionen ein Alert mit anschließendem COLD-Reset ausgegeben, wobei folgender Text angezeigt wird:

```
SACHSEN VIRUS NO.3 in Generation : xxxx is running
```

In seltenen Fällen kann auch eine Festplatte durch teilweises

Überschreiben des Rigidiskblocks unbrauchbar werden.
Man kann in dem Bootblock keine verräterischen Text erkennen,
da diese dekodiert vorliegen.

1.246 saddamhussein

SaddamHussein

Dieser Bootblockvirus ist programmtechnisch identisch mit dem
BlowJob -Virus. Es bestehen nur zwei Unterschiede:

Beim BlowJob-Virus kann man zu Beginn des Bootblocks folgenden Text lesen:

[0.5] [1] [1.8] MB Memory Allocator 3.01

Beim SaddamHussein-Virus steht zu Beginn des Bootblocks folgender Text:

A2000 MB Memory Controller V2

Der BlowJob-Virus gibt sich nach 10 Minuten mit folgender Alert-Meldung zu
erkennen:

ONCE AGAIN SOMETHING WONDERFUL HAPPENED (HE HE HE)
PLEASE POWER OFF - PLEASE POWER OFF - PLEASE POWER OFF

Der SaddamHussein-Virus hingegen gibt hierbei folgenden Text aus:

TOO BAD BROTHER SADDAM HUSSEIN STRIKES BACK !!!
THE ONLY ESCAPE IS TO TURN THE POWER OFF !!!

1.247 sao paulo

SAO PAULO

Es handelt sich um einen Bootblockvirus, der ab \$7FC00 im Speicher steht.
Er macht sich über den COOL-Vektor resetfest und verbiegt nach dem nächsten
Reset den DoIO()-Vektor um die Bootdiskette und zukünftig eingelegte
Disketten zu infizieren. Der eigentliche Viruscode steht erst im letzten
Drittel des Bootblocks, damit der Bootblock auf den ersten Blick möglichst
weiterhin wie ein normaler Bootblock aussieht. Im letzten Drittel des
Bootblocks kann man auch den Namen des Virus 'SAO PAULO!' lesen.

1.248 satan

SATAN

Es handelt sich um einen Bootblockvirus, der ab \$7CD30 im Speicher steht.
Er macht sich über den COOL-Vektor resetfest und verbiegt beim nächsten
Reset den DoIO()-Vektor um die Bootdiskette und zukünftig eingelegte
Disketten zu infizieren. Die Virusroutinen machen circa 40 % des Bootblocks

aus. Nach diesen 40 % Viruscode folgt noch 60 % reiner lesbarer Text, u.a. Hier spricht der SATAN !!!!

1.249 sca

SCA

Es wird beim Booten die Bootdiskette mit dem SCA-Virus infiziert. Ansonsten werden keine Disketten infiziert. Es handelt sich um einen Bootblockvirus, welcher sich über den COOL-Vektor resetfest macht. In dieser COOL-Routine wird der DoIO()-Vektor verbogen, damit beim gleich folgenden Diskbootversuch die Bootdiskette infiziert werden kann. Danach wird der DoIO()-Vektor wieder rerestauriert. Wird während des Resets die linke Maustaste gedrückt, so wird der SCA-Virus entfernt. Die Besonderheit des SCA-Virus liegt darin, daß nach 15 Infektionsversuchen (entspricht meist 15 Resets) eine längere Virusmeldung ausgegeben wird. Wenn man bedenkt, daß es sich bei dem SCA-Virus um den ersten funktionsfähigen Amiga-Virus handelt, so muß man die Programmierung durchaus als bemerkenswert, wenn auch keinesfalls als lobenswert betrachten. Die willkürliche Speicherbenutzung ab \$7c8fc muß als Programmierfehler betrachtet werden. Es gibt mittlerweile sehr viele SCA-Viruskopien, welche sich lediglich durch einen abgeänderten Text von dem Original-SCA-Virus unterscheiden. Wenn VIRUS CONTROL einen Original-SCA-Virus auf Diskette identifiziert hat, dann wird SCA-SCA! als Virusname gemeldet. Sollte es sich um einen SCA-Abkömmling handeln, dann können die letzten 4 Buchstaben anders lauten. Es wird also in solchen Fällen z.B. SCA-INC! oder SCA-BS! gemeldet.

Ab Kickstart 2.0 läßt sich der SCA-Virus zwar im Speicher installieren, wenn man dann aber bei aktivem Virus erneut von einer bereits infizierten Diskette bootet, dann erfolgt ein Absturz. Der Grund ist folgender. Der SCA-Virus kopiert immer den kompletten Bootblock starr nach \$7ec00, ohne vorher zu prüfen, ob nicht bereits der Virus aktiv ist, wodurch also ein bereits aktiver SCA-Virus mit nicht initialisierten Daten überschrieben wird. Das heißt, unter anderem wird ab \$7efdc der jmp Original-DoIO()-Befehl mit jmp \$00000000 überschrieben. Unter Kickstart 1.2/1.3 wird dieser nun falsche Befehl nicht benutzt, da vor dem Ausführen des Virus-Bootblockes der Bootdiskette bereits der aktive Virus den DoIO()-Vektor wieder auf den Original-ROM-Wert restauriert hat, denn der SCA-Virus will nur die Bootdiskette infizieren, und nachdem er das Vorliegen des Boot-DoIO auf die Bootdiskette mittels cmpa.l 40(A1),A4 erkannt hat, restauriert er den DoIO()-Vektor und infiziert die Bootdiskette, weitere Diskinfektionen sind dann nicht mehr möglich, anschließend kann dann ruhig der Viruscode der Bootdiskette einen unsinnigen jmp \$000000 kopieren, denn dieser wird nun nicht mehr angesprungen.

Ab Kickstart 2.0 wird aber der DoIO()-Vektor nicht mehr auf den ROM-Wert zurückgesetzt, da die Bedingung cmpa.l 40(A1),A4 nur unter Kickstart 1.2/1.3 als Betriebssystem-DoIO()-Boot-Aufruf interpretiert werden konnte. Das heißt ab Kickstart 2.0 wird recht bald dieser unsinnige jmp \$00000000 benutzt, wodurch es zum Absturz kommt. Der SCA-Virus ist aber auch unter Kickstart 2.0 resetfest, wodurch es dann beim Booten von einer noch nicht infizierten Diskette nicht zum Absturz kommt. Es kommt allerdings auch nicht zu einer Infektion, eben weil ab Kickstart 2.0 die cmpa.l 40(A1),A4 Bedingung nicht mehr erfüllt ist.

Der SCA-Virus würde also auch unter Kickstart 2.0 korrekt laufen, wenn nur einer der folgenden 3 Punkte erfüllt wäre.

1. nicht immer den gleichen starren Adressbereich benutzen
2. vor dem Kopieren der Virusdaten auf bereits aktiven Virus testen
3. nicht starr `cmp.l 40(A1),A4` als `Boot-DoIO()`-Zugriff-Bedingung annehmen

Unter Kickstart 1.3 funktioniert der SCA-Virus nur deswegen 100%, weil Punkt 3 erfüllt ist. siehe `Bootdisk-Bug` .

Es existieren mittlerweile sehr viele Abkömmlinge des SCA-Virus, bei denen lediglich unwesentliche Textänderungen vorgenommen wurden.

1.250 sca-2001

SCA-2001

SCA -Abkömmling.

1.251 sca-aids

SCA-AIDS

SCA -Abkömmling

1.252 sca-dag

SCA-DAG

SCA -Abkömmling, wird auch durch `DAG-Virus-Infector` installiert

1.253 sca-kefrens

SCA-Kefrens

SCA -Abkömmling.

1.254 sca-max

SCA-MAX

SCA -Abkömmling.

1.255 scarfce

SCARFACE

Der SCARFACE-Virus ist eine billige Kopie des ByteBandit -Virus. Lesen Sie daher bitte auch die Informationen zum ByteBandit-Virus durch. Der SCARFACE-Virus unterscheidet sich nun lediglich in folgenden Punkten vom ByteBandit-Virus: Die Besonderheit des ByteBandit-Virus ist die Blockierung des Rechners nach 7 Minuten. Auf dem Bildschirm wird dann nur noch die Hintergrundfarbe angezeigt. Diese Blockade tritt aber nur dann auf, wenn der Rechner mindestens 2 * resettet wurde UND mindestens 6 Disketten infiziert wurden. Man kann die Blockade durch folgende Tastenkombination aufheben: L-ALT L-AMIGA SPACE R-AMIGA R-ALT. Der SCARFACE-Virus geht hier viel primitiver zu Werke. Er bringt den Rechner nach knapp 3 1/2 Minuten zum Abstürzen, vorausgesetzt es wurden 10 Disketten infiziert und 2 Resets ausgelöst. Der ByteBandit-Virus gibt sich durch folgenden Text am Bootblock-Anfang zu erkennen:

Virus by Byte Bandit in 9.87.Number of copys :

Beim SCARFACE-Virus kann man am Bootblockende folgenden Text lesen:

SCARFACE trackdisk.device dos.library

1.256 sendarian

Sendarian

Revenge -Abkömmling.

1.257 sentinel-ussr492

Sentinel-USSR492

Es handelt sich um einen EXCREMENT -Abkömmling.
Der Text EXCREMENT wurde durch Sentinel! ersetzt und gegen Ende des Bootblocks ist folgender zusätzlicher Text zu lesen:

The USSR 492 is proud to present >>> The Sentinel <<< written in
deep space by Monxla-B ! Published by Voronezh ! Thanx to Phantom
for the menthal help ! ... USSR 492 forever

Allerdings hat sich bei dieser primitiven Textänderung auch noch ein kleiner Fehler eingeschlichen, denn der Sentinel-USSR492 kann aufgrund einer fehlerhaften Adressierung nicht mehr erkennen, ob schon eine mit dem Sentinel-USSR492-Virus infizierte Disk vorliegt. Das ist aber nicht weiter schlimm, es werden nun also überflüssigerweise bereits infizierte Disketten noch mal infiziert.

Bei Bootblockviren hat eine mehrmalige Infizierung meist keine Konsequenzen, denn es wird ja lediglich der Bootblock erneut überschrieben, also immer das gleiche Ergebnis erhalten.

Anders sieht es allerdings bei mehrmaligen Infizierungen durch Linkviren aus, denn hierbei wird das infizierte File immer länger. Aber da eine mehrmalige Fileinfizierung keinen Sinn macht, belassen es Linkviren meistens bei einer einmaligen Infizierung.

1.258 shit

SHIT

Da man zu Beginn des sich selbst verschlüsselnden Bootblockvirus immer Nuked007 lesen kann, wird der Virus manchmal Nuked007 genannt. Der Shit-Virus ist entfernt dem ByteBandit -Virus nachprogrammiert. War der ByteBandit ein meist funktionierender Bootblockvirus, so stürzt der SHIT-Virus allerdings nach kurzer Zeit ab. Vorher werden jedoch die Dateien auf der Diskette mehr oder weniger beschädigt. Unter anderem werden auch \$1400 Bytes ab Block 0 geschrieben. Der Virus macht sich über die Kick-Vektoren resetfest, wobei kein Identifikationstring verwandt wird. Es werden die Auto-Interrupts 1,2,3 und der Exec-Interrupt 3 (Vertikal Blank) verbogen. Die Disk-Infektionen werden durch einen verbogenen TD_BeginIO()-Vektor ermöglicht.

1.259 sonja

Sonja

Der Virus macht sich kompatibel über die Kick-Vektoren resetfest. Als Kick-Identifikationsstring wird 'It's Sonja VIRUS!!!' verwandt. Eventuelle sonstige resetfeste Programme bleiben erhalten. Zum Infizieren von Disketten verbiegt der Virus den TD_BeginIO()-Vektor des trackdisk.devices. Allerdings kommuniziert der Virus nicht korrekt mit dem trackdisk.device, so daß ein Virusinstallationsversuch nur unter Kickstart 1.2/1.3 gelingt. Der Virus schreibt über die Strahlenposition zufällig verschlüsselte Virus-Bootblöcke. Ein aktiver Sonja-Virus versucht anderen Programmen einen normalen Bootblock anstelle des Virusbootblocks vorzugaukeln. Nach jeweils 20 Disketteninfektionen begnügt sich der Virus nicht mehr mit dem Schreiben eines Virus-Bootblockes, sondern er schreibt zuvor jede zweite Diskettenspur mit Nullen voll, wodurch also praktisch alle Dateien beschädigt werden oder verloren gehen. Weiterhin ändert der Virus den Diskettennamen auf 'It's Sonja VIRUS!!!',0 indem ein neuer Rootblock geschrieben wird. Nach dem Formatieren der Diskette und erneutem Schreiben des Virus-Bootblockes wird ein Soft-Reset versucht. Eine vom Sonja-Virus formatierte Diskette ist also leer und auch z.B. disksalv wird nicht mehr viel retten können, da zuviele Daten überschrieben wurden.

Im Rootblock (Block 880) kann man folgendes lesen:

```
$14,'It's Sonja VIRUS!!!',$0,'(c)91 by MC',0
```

Da das Betriebssystem den Diskettenname als BCPL-String verwaltet, wird die Länge des Strings nicht durch \$0 definiert, sondern durch das erste Byte des BCPL-String, wodurch also das \$0-Byte als letztes Zeichen des Diskettennamen interpretiert wird. Dieser unzulässige Diskettenname und auch der Umstand, daß kein korrekter BitMap-Block erstellt wird, führen dazu, daß beim Versuch Daten auf diese vom Sonja-Virus formatierte Diskette zu schreiben, meist ein Absturz erfolgt.

1.260 ss

SS

--

Es handelt sich strenggenommen nicht um eine Virus, da keine neuen Disketten infiziert werden. Dennoch ist der SS störend, da er den Rechner nach insgesamt 25 Sekunden Drücken der linken Maustaste zum Absturz bringt.

Da der SS-Bootblock mit eor.l #'!SS!' kodiert, kann man z.B. folgenden verräterischen Text nicht erkennen:

Program by Adolf Hitler - Text by Göbbels

Der SS kopiert sich nach \$7c000 und macht sich über den COOL-Vektor resetfest. Nach dem nächsten Reset wird ein CIA-A Interrupt eingehängt, wodurch bei jedem Tastendruck der COOL-Vektor auf den SS verbogen wird, Damit soll die Resetfestigkeit sichergestellt werden. Der Name der Interrupt-Node ist SS.install. Weiterhin wird ein Vertikal-Blank-Interrupt mit Namen SS.greetings eingehängt, wodurch 50 mal in der Sekunde geprüft wird, ob die linke Maustaste gedrückt ist. Wenn ja wird ein Zähler um 1 erhöht. Wenn der Zähler auf 1280 steht, wird eine Copper-Geafik ausgegeben:

Auf einem schwarzen Hintergrund wird mit weißer Schrift ein Hakenkreuz, eine SS-Rune, ein Hakenkreuz und der Text your computer is infected by SSvirus! ausgegeben.

Aufgrund eines Programmierfehlers (jmp \$219908) muß anschließend der Rechner resettet werden.

1.261 starcom

STARCOM

Bei der STARCOM-Familie handelt es sich lediglich um Abkömmlinge bereits bekannter Bootblockviren

ByteBandit
ByteVoyager I
CCCP
Lamer-Bootblockviren
Warsaw Avenger

1.262 starfire-north

Starfire-Northstar

Es gibt mehrere Versionen (z.B. OldNorthstar und NewNorthstar) und auch Nachahmer (wie z.B. Ripper-Virus). Es handelt sich hierbei um unsauber programmierte Bootblockviren, welche ab \$7ec00 im Speicher stehen. Der Starfire-Northstar-Virus macht sich über den COOL-Vektor resetfest und überschreibt die Bootdiskette mit dem Starfire-Northstar-Virus, falls sich ein älterer Virus, wie SCA oder ByteBandit auf der Diskette befindet. Der Starfire-Northstar soll also ein Antivirus sein. Er sollte dennoch gelöscht werden, da er resetfest ist und Bootblöcke überschreiben kann und somit also doch als VIRUS zu betrachten ist.

1.263 starfire-eaststar

Starfire-EastStar

Starfire-NorthStar -Abkoemmling, lediglich Textänderungen, wie z.B. Noth-Star in East-Star und gegen Ende ist z.B.

VIRUS Detected on Disk! durch

No Virus is on the Disk RED BURNING BIGSTAR

ersetzt, wodurch der Anwender getäuscht werden soll.

Der Starfire-EastStar-Bootblockvirus ist mit der Hunklab -Methode vor ein 6624 Byte langes Programm namens MComm gelinkt, wodurch ein 8340 Byte langes Programm entsteht, welches beim Start im Speicher den Starfire-EastStar installiert, allerdings kann der Starfire-EastStar Disketten nicht erfolgreich infizieren, da er keine DOS-Kennung schreibt, wodurch eine 'Not-A-Dos-Disk' entsteht.

1.264 suicide

Suicide

Es handelt sich um einen Bootblockvirus, der ab \$7E120 dekodiert im Speicher steht, hier kann man dann u.a. auch folgenden Text lesen:

>> Suicide Machine by MAX in 24.09.1992 <<

Der Virus macht sich als alleiniges Programm über KickTag resetfest und verbiegt den Supervisor()-Vektor, wodurch in der neuen Virus-Supervisor-Einschleifung 50 Mal pro Sekunde die alleinige Resetfestigkeit des Virus sichergestellt wird, indem Cold/Cool gelöscht wird und KickTag auf den Virus gesetzt wird.

Weiterhin wird der DoIO()-Vektor verbogen, um die Bootdiskette und auch zukünftig eingelegte Disketten zu infizieren. Aufgrund unsauberer Infektionsroutinen kann auch Zylinder 0 von Autobootfestplatten beschrieben werden, wodurch die Rigid-Disk-Daten beschädigt werden, wodurch dann die Festplatte nicht mehr bootet. siehe Rigiddiskblock beschädigen .

Der Virus schreibt unter Benutzung der aktuellen Strahlenposition (\$dff006) einen somit zufällig kodierte Bootblock auf die Diskette. Hierzu wird eine kodierte Kopie des Virus ab \$7e620 angelegt.

1.265 superboy

SuperBoy

steht ab \$7ec00 im Speicher, macht sich über Cool resetfest, infiziert nur Bootdiskette, nach 7 Disk-Infektionen Alert

1.266 systemz

SystemZ 3.0,4.0,5.0,5.1,5.3,5.4

Es handelt sich um einen sogenannten Antivirus-Bootblock. Wenn die Bootdiskette z.B. mit dem SCA oder ByteBandit -Virus infiziert ist, so wird mittels Alert ein Überschreiben des Bootblocks mit dem SystemZ-Virus angeboten. Wenn man die linke Maustaste beim Booten gedrückt hält, so wird ohne Rückfrage generell der Bootblock der Diskette mit dem SystemZ-Virus überschrieben. Hierin liegt die größte Gefahr des SystemZ-Virus. Es handelt sich um ein Bootblockvirus, der etwas dem SCA -Virus gleicht. Er löscht den COOL-Vektor und macht sich über die Kick-Vektoren resetfest. Die residente Kick-Routine verbiegt den DoIO()-Vektor auf SystemZ, um beim gleich folgenden Diskbootversuch die Bootdiskette auf SCA und ByteBandit zu prüfen. Wenn positiv, kann mittels Alert ein Überschreiben des Bootblocks mit dem SystemZ-Virus angewählt werden (Left MouseButton: Kill the Virus). Nach der Prüfung der Bootdiskette wird der DoIO()-Vektor wieder rerestauriert. Es werden somit später beim Disketten-Einlegen keine Disketten geprüft bzw. infiziert. Der SystemZ-Virus ersetzt nach Rückfrage einen z.B. SCA oder ByteBandit-Virus-Bootblock durch den SystemZ-Bootblock. Man kann daher den SystemZ-Virus als einen Antivirus betrachten. Wenn man jedoch beim Booten die linke Maustaste gedrückt hält, so wird generell jeder Bootblock überschrieben. Aufgrund dieser Eigenschaft muß man SystemZ als Virus betrachten. Ein weiterer Grund, den SystemZ-Virus zu entfernen liegt darin, daß er nur auf 512KB-Amigas 'korrekt arbeitet', da er sich im vermeintlichen Supervisorstack \$7e800-\$80000 einnistet. Drückt man während des Resets die rechte Maustaste, so wird der SystemZ-Virus entfernt. Eine kurze Tonfolge und Grafik zeigt den aktiven SystemZ-Virus an. Es gibt mittlerweile eine ganze Menge SystemZviren, welche aber im Prinzip immer gleich arbeiten. SystemZ 3.0, 4.0, 5.0, 5.1, 5.3, 5.4

1.267 systemz 6.1,6.3,6.4,6.5

SystemZ 6.1,6.3,6.4,6.5

Bei SystemZ-6.1, 6.3, 6.4, 6.5 handelt es sich wieder um echte SystemZ-Bootblöcke. Diese neueren SystemZ-Varianten unterscheiden sich etwas von den früheren SystemZ-Varianten, denn wenn diese neueren

SystemZ-Varianten einen Virus auf Diskette finden, dann wird nun folgende Meldung ausgegeben:

```
Warning: Disk contains a Virus!  
Use install or an other program to remove the virus
```

Diese neueren SystemZ-Varianten beschreiben also nie die Diskette, auch nicht automatisch, wenn man die linke Maustaste drückt. Diese neueren SystemZ-Varianten sind also ungefährlich. Die neuesten SystemZ-6.3, 6.4, 6.5 benutzen nun den Supervisorstack. Die früheren SystemZ benutzen willkürlich Speicher ab z.B. \$7e800.

1.268 tai

TAI

Bei der TAI-Familie handelt es sich lediglich um Abkömmlinge bereits bekannter Viren wie z.B.

```
ByteBandit  
CCCP  
Gadaffi  
SCA
```

1.269 target

Target

Es handelt sich um einen unsauber programmierten Bootblockvirus, welcher ab \$70000 Speicher benutzt. Der Target-Virus löscht den COLD-Vektor und macht sich über den COOL-Vektor resetfest. Der Target-Virus prüft, ob von einer ganz bestimmten Diskette gebootet wird. Wenn ja, dann wird diese Bootdiskette zum Großteil mit unsinnigen Daten überschrieben, also unbrauchbar gemacht. Normalerweise ist der Target-Virus also ungefährlich. Mit dem Target-Virus soll also gezielt jemandem Schaden zugefügt werden.

1.270 telstar(systemz-v6.0)

TELSTAR (SystemZ-V6.0)

Die Texte, die man in diesem Bootblock lesen kann, lassen vermuten, daß ein SystemZ-V6.0-Bootblock vorliegt. Programmtechnisch besteht allerdings absolut keine Gemeinsamkeit mit den SystemZ -Bootblöcken. Der TELSTAR macht sich über den Cold und COOL-Vektor resetfest. Ein Teil der Daten sind kodiert. Aber auch nach der Dekodierung kann man die Daten nicht lesen, da sie der direkten Copper-Programmierung dienen. Der TELSTAR gibt nach 4 bzw. 2 Resets (Zähler in \$c0) eine Copper-Grafik aus, in welcher man die holländische Flagge mit folgendem Text erkennen kann:

```
TELSTAR Spreading is our business
```

and business is good

Ansonsten macht der TELSTAR nichts. Es handelt sich also nicht um einen Virus und auch nicht um einen Antivirus. Angenommen man versucht von einer mit dem ByteBandit infizierten Diskette zu booten. Die normalen SystemZ-Varianten weisen mittels Alert auf den Virus hin. Der TELSTAR hingegen überprüft den Bootblock nicht. Der Virus wird also installiert.

1.271 termigator

Termigator

Es handelt sich um einen Bootblockvirus, welcher nur mit Kickstart 1.2 läuft. Der Termigator-Virus macht sich über den COOL-Vektor resetfest und verbiegt den DoIO()-Vektor, um jede eingelegte Diskette zu infizieren. Eine Festplatte kann unbrauchbar werden. Manchmal erscheint ein Alert mit folgendem Text:

Only the TERMIGATOR'VIRUS makes it possible! Bye!

Dieser Alert endet normalerweise in einem Reset. Wenn man aber R-ALT drückt und dann den Alert durch einen Mausklick beendet, und dann nacheinander die Tasten i l o v e g x v i r u s drückt, dann kann man normal weiterarbeiten.

1.272 t.f.c. revenge virus

T.F.C. Revenge Virus

EXTREME -Abkömmling.

1.273 time-bomb-v1.0

TIME-BOMB-V1.0

Da der TIME-BOMB-Virus nicht resetfest ist, und auch keine Disketten beim Einlegen infiziert, kann er sich also praktisch nicht unbewußt verbreiten. Der Name TIME-BOMB ist also unsinnig. Eine Gefahr droht nur dann, wenn der TIME-BOMB-Bootblock schon auf Disk drauf ist. Wenn man dann von dieser Disk bootet, wird nach dem zweitem Reset die Root-Spur der Bootdiskette überschrieben, wodurch die Diskette unbrauchbar wird, (mit disksalv zum Großteil restaurierbar). Der TIME-BOMB-Virus ist mehr ein Bootblockvirusversuch. Er ist insgesamt mäßig, unlogisch und wenig wirksam programmiert. Ferner benutzt er willkürlich Speicher ab \$70000. Er kann zwar durch Überschreiben der Root-Spur der Bootdiskette durchaus datenvernichtend wirken, aber dieser Fall wird eher selten auftreten, da sich der TIME-BOMB-Virus nicht selbständig auf andere Disketten verbreiten kann.

1.274 tomatesgentechnicservice

TomatesGentechnicService

TIME-BOMB-V1.0 -Abkömmling.

1.275 traveller1.0

Traveller1.0

Es handelt sich um einen unsauber programmierten Bootblockvirus, welcher ab \$7f000 im Speicher steht. Am Ende des Bootblocks kann man den Virus-Namen lesen. Der Traveller 1.0 - Virus ist programmtechnisch nahezu identisch mit dem BlowJob -Virus. Es besteht folgender Hauptunterschied:
Der BlowJob-Virus gibt sich nach 10 Minuten mit folgender Alert-Meldung zu erkennen:

ONCE AGAIN SOMETHING WONDERFUL HAPPENED (HE HE HE)..
PLEASE POWER OFF - PLEASE POWER OFF - PLEASE POWER OFF

Der Traveller-Virus gibt anstatt eines Alert eine Grafik aus, und zwar einen rot-grün-blauen Balken auf schwarzem Hintergrund, in welchem folgender Text zu lesen ist:

NEVER HEARD OF VIRUSPROTECTION ???? - LAMER !!!!

Dieser Text kann im Bootblock nicht gelesen werden, da im Bootblock lediglich die entsprechenden Copperlisten vorliegen.

1.276 triplex

TRIPLEX

Es handelt sich um einen Bootblockvirus, welcher sich über den COOL-Vektor resetfest macht. Weiterhin wird der DoIO()-Vektor verbogen, um Disketten bereits beim Einlegen zu infizieren.
Man kann folgenden Text im Bootblock lesen:

This nice little Virus was written in 1990 by Darth Vader of TRIPLEX!!

1.277 trisector 911

TRISECTOR 911

Es handelt sich um einen Bootblockvirus, welcher sich mit folgendem lesbarem Text im Bootblock als Antivirusprogramm tarnen will.

This disk was installed with the TRISECTOR 911 virus-slayer!

NO virus is safe for our killer... Signed: TRISECTOR 911

Virus problems??? Call 911 (Collect call)

Der Virus steht ab \$7F000 im Speicher und macht sich über KickTag, KickChecksum resetfest, wobei andere Kick-resetfeste Programme wie z.B. RAD: verloren gehen. Es werden keine Kick-Identifikationsstrings verwandt. Nach dem nächsten Reset wird der DoIO()-Vektor verbogen, um Disks bereits beim Einlegen zu infizieren. Nach der ersten Disk-Infektion wird auch der Exec-Interrupt-3 verbogen, um nach 15 Minuten die Tastatur zu sperren.

1.278 turk virus 1.3

TURK VIRUS 1.3

Es handelt sich um einen unsauber programmierten und sehr böartigen Bootblockvirus. Er verbiegt den COOL-Vektor, um sich resetfest zu machen. Weiterhin verbiegt er den DoIO()-Vektor, um Disketten bereits beim Einlegen zu infizieren. Man kann den Virus entfernen, indem man während des Resets beide Maustasten und die F10-Taste drückt. Die erfolgreiche Virusentfernung wird durch einen Regenbogen-Effekt signalisiert. Nach dem Reset versucht der Virus 851968 Bytes zu belegen, diese Zahl erhöht sich bei jedem weiteren Reset um 425984. Nach 5 Resets oder 10 Disketten-Infektionen wird die Hälfte der Diskette formatiert. Nach dem Formatieren wird folgender Alert ausgegeben:

Amiga Failure... Cause: TURK VIRUS Version 1.3!

ACHTUNG: höchste Gefahr für Festplatten-Besitzer: Der TURK-Virus prüft nicht, ob sich der DoIO()-Zugriff auch wirklich auf das trackdisk.device bezieht. Vielmehr versucht sich der Virus bei jedem 512 oder 1024-Lese oder Schreib-DoIO()-Zugriff mit Offset 0 auf das jeweilige Device zu kopieren(=infizieren) oder gar eine Formatierung durchzuführen. Viele Festplattentreiber legen wichtige Festplattenverwaltungsdaten wie z.B. Errorliste, Partitionsdaten, Mountliste usw. auf den unteren Zylindern ab. Diese werden hierbei überschrieben. Die Festplatte wird unbenutzbar!! siehe Rigiddiskblock beschädigen .

1.279 twinz santa claus

TWINZ SANTA CLAUS

CODER -Abkömmling.

1.280 uhr

Uhr

Es handelt sich um einen Bootblockvirus, welcher meist dekodiert ab \$7f800 im Speicher steht, ab \$7f000 steht immer eine kodierte

Kopie des Virus, die bei Disketteninfektionen auf den Bootblock geschrieben wird. Der Bootblock wird unter Benutzung eines CIA-Timers zufallsgesteuert verschlüsselt. Der Virus löscht den COLD- und KickTag-Vektor und macht sich als alleiniges Programm über den COOL-Vektor resetfest. Weiterhin wird der DoIO()-Vektor verbogen, um die Bootdiskette und alle weiteren eingelegten Disketten zu infizieren. Darüberhinaus wird auch der Autointerrupt 3 verbogen, um 50 Mal pro Sekunde (Vertikal-Blank-Interrupt) den COOL- und DoIO()-Vektor auf den Virus zu verbiegen. Der Virus versucht einen normalen Bootblock vorzutäuschen, was aber nur bei schreibgeschützten Disketten gelingt. Nach 4 Resets versucht der Virus zufallsgesteuert in der Virus-Vertikal-Blank-Routine diverse Effekte zu erzielen, so wird versucht, eine eventuelle Echtzeituhr der alten A2000A-Modelle schneller laufen zu lassen, indem zusätzliche Werte auf das Sekundenregister addiert werden oder es wird Timer B der CIA-A manipuliert, wodurch das timer.device des Betriebssystems verwirrt wird oder es wird die Strahlenposition beeinflusst, was sich in einem kurzen Bildschirmflackern äußert oder es wird die Mausposition verändert oder es wird der Rechner durch Warteschleifen abgebremst.

1.281 ultra-fox

ULTRA-FOX

Der Virus macht sich über den COOL-Vektor resetfest. Es wird nur die Bootdiskette infiziert. Nach 16 Resets gibt sich der Virus durch eine Copper-Grafik mit dem Text 'UltraFox' zu erkennen

1.282 umyj dupe

Umyj Dupe

Es handelt sich um eine Weiterentwicklung des DASA-ByteWarrior -Bootblockvirus. Es wird nun auch der Diskettenname in 'Umyj Dupe - Wash Ur Ass' umbenannt und beim Reset erscheint ein Alert mit folgendem Text:

Umyj Dupe - Wash Your Ass

Wie der DASA-ByteWarrior-Bootblockvirus läuft auch dieser Virus nur mit Kickstart 1.2 und kann (vesehentlich) wichtige Festplattendaten auf Zylinder 0 überschreiben. siehe Rigidiskblock beschädigen .

1.283 vccoftnt

VCCofTNT - ACCESS FORBIDEN

Es handelt sich nicht um einen wirklichen Bootblockvirus, da sich der

Bootblock nicht automatisch kopieren kann. Es erfolgt also keine Neuinfektion weiterer Disketten.
Wenn man von einer Diskette bootet, auf welcher sich der VCCofTNT-Bootblock befindet, dann wird eine rote Grafik auf schwarzem Hintergrund ausgegeben, in welcher man folgendes lesen kann:

```
VCC of TNT
ACCESS
FORBIDEN
```

Man kann diesen Text nicht im Bootblock lesen, da er durch direkte Programmierung der Customchipregister erzeugt wird.
Wenn man nun die rechte oder linke Maustaste drückt, dann werden willkürliche Speicherdaten auf Block 0+1 und Block 880+881 geschrieben. Hierdurch resultiert eine unbrauchbare 'Not A DOS-Disk'. Mit disksalv können Sie womöglich den Großteil der Daten wieder retten.

Wenn man anstatt der Maustasten eine Tastaturtaste drückt, dann passiert nichts, es sei denn, man drückt die CTRL-Taste. In dem Fall wird anschließend solange gewartet bis die rechte Maustaste gedrückt wird, um dann normal weiter zu booten. In diesem Ausnahmefall wird die Diskette also nicht beschrieben.

1.284 vermin

```
VERMIN
-----
```

Der VERMIN-Virus basiert auf dem SCA -Bootblockvirus, Es bestehen lediglich folgende Unterschiede: Der Virus steht ab \$7eb10 im Speicher, der SCA ab \$7ec00. Beim SCA-Virus wird nach nach 15 infizierten Bootdisketten eine Meldung ausgegeben. Diese fehlt beim VERMIN-Virus. Anstatt dessen besteht die zweite Hälfte des VERMIN-Virus-Bootblockes aus zufälligen Strahlenpositionswerten. Da diese immer verschieden sind, muß für jeden Bootblock eine neue Checksum berechnet werden. Man findet also ab Byteposition 12 des Bootblocks nicht mehr immer 'CHW!', wie dies beim SCA fest der Fall war.

1.285 virusconstr.i

```
VirusConstructionI
-----
```

siehe The Virus Construction Set von STR .

1.286 virusconstr.ii

```
VirusConstructionII
-----
```

siehe The Virus Construction Set von STR V2.0 .

1.287 virus fighter v1.0

VIRUS FIGHTER V1.0

VKill 1.0 -Abkömmling.

1.288 virusv1

VirusV1

Es handelt sich um einen Bootblockvirus, welcher ab \$7EC00 im Speicher steht. Er macht sich über den COOL-Vektor resetfest und verbiegt den DoIO()-Vektor, um Disketten bereits beim Einlegen zu infizieren. Alle 16 Disk-Infektionen erfolgt eine Copper-Grafik. Auf einem schwarzen Hintergrund erscheint folgender grüner Text:

VIRUS Wir sind wieder daaahaaa..

Den Text 'Wir sind wieder daaahaaa' kann man auch im Bootblock lesen.

1.289 vkill 1.0

VKill 1.0

VKill 1.0 versteht sich als ein Virus-Killer, der im Bootblock selber steht. VKill wird also aktiviert, indem man von einer Diskette bootet, auf deren Bootblock sich VKill befindet. Legt man nun eine mit dem SCA -Bootblockvirus oder ByteBandit -Bootblockvirus infizierte Diskette ein, so weist VKill 1.0 mittels Requester daraufhin, und bietet hierbei die Möglichkeit an, den Virus-Bootblock durch den VKill-Bootblock zu ersetzen. Legt man eine Diskette mit einem anderen nicht normalen Bootblock ein, dann weist VKill ebenfalls mittels Requester daraufhin. Folgende Texte können in den von VKill ausgegebenen Requestern erscheinen:

VKill 1.0 infection control
One less virus to kill
Bootblock neutralized
SCA virus detected!
Byte Bandit virus detected!
Bootblock code not normal
Disk write protected
Destroy
Ignore
Thanks

Legt man eine Diskette mit einem normalen Bootblock ein, dann wird automatisch diese Diskette mit VKill infiziert. Es wird also ohne Rückfrage ein VKill-Bootblock geschrieben. Dies ist auch der Grund, warum man VKill als Virus betrachten muß. Erfreulicherweise infiziert VKill nur Disketten mit einem normalen Bootblock. Dadurch gehen also keine wichtigen Bootblöcke verloren. VKill macht sich über den COOL-Vektor resetfest. Der Virus-Code wird in den Supervisorstack kopiert.

Es gibt bisher drei Möglichkeiten, wie sich Bootblockviren verbreiten:

1. Es wird der DoIO()-Vektor in der exec.library verbogen oder
2. Es wird der BeginIO-Vektor im trackdisk.device verbogen.
VKill benutzt nun eine neue bisher noch nicht verwendete Möglichkeit:
3. Es wird der PutMsg()-Vektor in der exec.library verbogen.

Alle drei Methoden führen zum 'Erfolg', da das Betriebssystem beim Disketteneinlegen einen DoIO()-Zugriff auf den Bootblock ausführt. Dieser DoIO()-Zugriff geht dann im Betriebssystem in einen BeginIO-Zugriff weiter, dieser wiederum mündet letztendlich in einem PutMsg()-Aufruf. VKill ist also sehr interessant programmiert. Auch ist der Code recht sauber und optimiert programmiert, was ja bei Virus- oder Antivirusprogrammen oft nicht der Fall ist. Dennoch ist auch VKill nicht fehlerfrei, so wird z.B. in einer Schleife 1 LongWord über den reservierten 1024-Bytes-Speicherbereich hinaus geschrieben. Wenn VKill aktiv ist, dann spiegelt es anderen Programmen anstatt eines VKill-Bootblocks einen normalen Bootblock vor. Auch wegen dieser Verschleierungstaktik muß man VKill als Virus betrachten.

1.290 waft

WAFT

Es handelt sich um einen Bootblockvirus, welcher sich über den COOL-Vektor restfest macht und den DoIO()-Vektor verbiegt, um Disketten bereits beim Einlegen zu infizieren. Der aktive WAFT-Virus täuscht anstelle seines Virus-Bootblocks einen normalen Bootblock vor. Nach einer gewissen Anzahl Disk-Infektionen stellt der Virus z.B. die Sprites ab oder läßt einen Alert mit folgendem Text erscheinen:

```
    ! !  W A F T ! !  
    Quality made in West-Germany
```

Der Text ist nur im Speicher zu lesen, da er im Bootblock kodiert vorliegt.

1.291 wahnfried

WAHNFRIED

Es handelt sich um einen Bootblockvirus, der ab \$07EF50 im Speicher steht. Der Virus macht sich über den COOL-Vektor resetfest und verbiegt den PutMsg()-Vektor, um Disketten bereits beim Einlegen infizieren zu können. Nach 20 Disketteninfektionen, und danach immer nach 10 Disketteninfektionen, erscheint folgender Alert:

```
    Hardware Failure.   Press left mouse button to continue  
    Guru Meditation    #00000015.00C03L12
```

```
    Hooligen-Bits randalieren im Datenbus!
```

```
    Gruß   Erich!
```

Diesen Text liegt kodiert im Bootblock vor und kann somit nicht gelesen werden. Folgender Text ist aber immer auffällig gegen Ende des Bootblocks zu erkennen:

WAHNFRIED STRIKES AGAIN !!

Der WAHNFRIED-Virus weist einen Programmierfehler auf, denn er benutzt die Bootblock-Checksumme als Kennzeichen für eine bereits infizierte Diskette, das heißt der WAHNFRIED-Virus schreibt immer die gleiche Bootblock-Checksumme. Da der WAHNFRIED-Virus aber Daten verändert, z.B. den Disketteninfektionzähler, muß auch die Bootblock-Checksumme neu berechnet werden. Da die Bootblockchecksumme aber fälschlicherweise als starre Identifikationskennung mißbraucht wird, sind die infizierten Disketten meist nicht bootfähig.

1.292 warhawk

WARHAWK

Der WARHAWK-Virus ist ein unsauber programmierter Bootblockvirus. Er macht sich über den COOL-Vektor resetfest. Bei dem nächsten Reset wird der DoIO()-Vektor verbogen, um beim gleich folgenden Booten die Bootdiskette zu infizieren. Danach wird der DoIO()-Vektor wiederhergestellt, es wird also nur die Bootdiskette infiziert. Nach jeweils 4 Resets meldet sich der Virus: Es erscheint eine nette Copper-Grafik-Spielerei, wobei folgender Scroll-Text ausgegeben wird:

WARHAWK SAYS : KILLING YOUR DISKS WITH OUR VIRUS IS A WONDERFUL THING !
CONTACT : UCS, PLK 000257-A, 3457 STADTOLDENDORF ! HEY BAD ! FUCK OFF

1.293 warsaw avenger

Warsaw Avenger

Der Warsaw Avenger ähnelt sehr stark den Lamer-Bootblockviren . Der Hauptunterschied liegt in der fehlenden Bootblock-Verschlüsselung. Der Virus steht in einem zufälligen Speicherbereich im Supervisorstack. Er macht sich über KickTag, KickCheckSum komaptibel resetfest. Als Name der Kick-Struktur wird 'Warsaw Avenger presents!!!' verwendet. Diesen Text kann man auch am Ende des Bootblocks lesen. Um die Kick-Resetfestigkeit zu erhöhen wird SumKickData() verbogen. Weiterhin wird der BeginIO()-Vektor des trackdisk.devices verbogen, um Disketten bereits beim Einlegen infizieren zu können. Neben dem Überschreiben des Bootblockes werden auch zufällige Blöcke mit 'Warsaw' überschrieben.

1.294 zaccess v1.0

ZACCESS V1.0

16Bit Crew -Abkömmling.

1.295 zaccess v2.0

ZACCESS V2.0

ByteBandit -Abkömmling.

1.296 zaccess v3.0

ZACCESS V3.0

EXTREME -Abkömmling.

1.297 z.e.s.t

Z.E.S.T

Es handelt sich um einen L.A.D.S -Abkömmling.

Es bestehen lediglich folgende Unterschiede:

Anstatt nach 5 Disketten-Infektionen die X/Y-Maus-Koordinaten zu invertieren, kann nun die Maus nur noch vertikal verschoben werden. In dem Bootblock kann folgender Text gelesen werden, mit welchem sich der Virus als Antivirus-Bootblock tarnen will:

Z.E.S.T is the B.E.S.T Virus-Killer ever made!!!
finds 211 Viruses including the dangerous Nastyvirus !!!

1.298 zenker

ZENKER

Es handelt sich um eine Bootblockvirus, welcher ab \$7f400 im Speicher steht. Der Bootblock einer infizierten Diskette ist mit Absicht möglichst unauffällig gestaltet, das heißt er ist meist wie der Original-Commodore-Bootblock weitgehend leer und darüber hinaus soll der lesbare Text

Commodore Bootloader (20 Oct 1987)

den Anwender in Sicherheit wiegen. Es handelt sich bei diesem unscheinbaren Bootblock jedoch um eine Laderoutine, welche ab Block 896 von Spur 40 2048 Bytes nach \$7f800 nachlädt und dann in diesen eigentlichen Viruscode einspringt, wobei als erstes der frühere Original-Bootblock, der ab \$7fc00 steht, aus Tarngründen ausgeführt wird. Danach werden alle restfesten

Programme ausgetragen und nur noch der ZENKER-Virus macht sich über den COOL-Vektor resetfest. Nach dem nächsten Reset wird dann der DoIO()-Vektor verbogen, um die eingelegte Bootdiskette und auch weitere in Zukunft eingelegte Disketten zu infizieren. Beim Infizieren einer Diskette wird zuerst der Original-Bootblock der eingelegten Diskette nach \$7fc00 eingelesen, danach werden die eigentlichen 1024 Bytes Viruscode, welche ab \$7f800 stehen und die darauf folgenden 1024 Original-Bootblock-Bytes ab Block 896 (Spur 40) auf der Diskette abgespeichert, zuvor wurde noch der Text == ZENKER == in diese 2048 Bytes geschrieben und in folgendem Text, welcher ebenfalls in diesen 2048 Bytes steht, die Zahl um eins erhöht:

```
NOW I'M IN THE 24 GENERATION ONLY THE ZENKER CAN COPY IT !
```

Nachdem also der der Viruscode und der Original-BB auf der Diskette ab Block 896 abgespeichert wurden, wird zum Abschluß noch dieser unscheinbare Virusloader-Bootblock auf den Bootblock geschrieben. Hierbei wird willkürlich Speicher ab \$7f400 belegt, weiterhin wird auch willkürlich der Speicherbereich ab \$75000 zum Zwischenspeichern der Original-IO-Request-Daten mißbraucht, insofern ist der AllocAbs()-Aufruf des Virus für den Speicherbereich \$7f800 bis \$7fc00 nicht ausreichend.

Der ZENKER-Virus wird über kurz oder lang zu Schwierigkeiten mit der befallenen Diskette führen, denn der ZENKER-Virus schreibt seine 2048 Bytes auf Spur 40 und gerade Spur 40 ist die am meisten benutzte Diskettenspur, denn das Amigabetriebssystem legt aus Geschwindigkeitsgründen Daten zuerst auf den mittleren Spuren auf und um Spur 40 herum ab. Je voller die Diskette dann wird, umso mehr werden dann auch die äußeren Spuren belegt, da aber zum Anfahren der äußeren Spuren der Schreib-Lese-Kopf längere Wege zurücklegen muß, und da auch zwischenzeitlich immer wieder Informationen von der mittleren Rootspur 40 gelesen werden müssen, werden innere Spuren vor äußeren Spuren bevorzugt. Das heißt, daß in der Regel eine z.B. nur 10% volle Diskette bereits Spur 40 komplett belegt hat, so daß der ZENKER-Virus auch bei weniger vollen Disketten beim Schreiben seiner 2048 Bytes meist bereits wichtige Daten überschreibt. Meist kann man mit z.B. disksalv den Großteil der Diskettendaten retten, zumindest die überschriebenen 2048 Bytes sind allerdings verloren.

1.299 zombi i

Zombi I

Es handelt sich um einen Bootblockvirus welcher ab \$7a000 im Speicher steht und sich über den COOL-Vektor resetfest macht. Es wird nur die Bootdiskette infiziert. Hierzu wird kurzzeitig der DoIO()-Vektor verbogen. Festplatten sind normalerweise nicht gefährdet. Nach 15 Bootdisketten-Infektionen gibt sich der Zombi I - Bootblockvirus zu erkennen. Es erscheint der folgende Alert:

```
      Hello AMIGA User !!!!!
      HERE IS ZOMBI I
If you want to clean your Disks
  use Zombie I without risks !
```

Dieser Text ist im Bootblock nicht lesbar, da er kodiert vorliegt.

Vor dem Erscheinen dieses Alerts wurde der Diskettenname in Zombi I umgeändert, indem ein neuer RootBlock und BitMapBlock geschrieben wurde. Hierbei wird allerdings recht grob zu Werke gegangen, so daß meist eine fehlerhafte Diskette resultiert. Mit z.B. disksalv sollten sich jedoch die Daten retten lassen.

1.300 Fileviren

Fileviren

echte Viren mit Infektionsroutine, um neue Datenträger zu infizieren

Die 'echten' sich verbreitenden Fileviren tragen meist in die Startup-Sequenz eine Zeile ein, in der er sich selber aufrufen lassen. Damit diese zusätzliche Zeile weniger auffällt, besteht der Virusfilename oftmals aus unsichtbaren Steuerzeichen. Der Filevirus wird nun also beim Booten und Abarbeiten der Startup-Sequenz aktiviert. Ein Filevirus läßt sich also relativ leicht entfernen. Man muß die neue Zeile mit dem Aufruf des Filevirus löschen und sollte auch das Filevirusprogramm selber löschen.

AMIGAKNIGHTS
Anti-EuroMail-Virus
BGS9 I+II+III
BLUEBOX-icon.library
BRET-HAWNES
BUTONIC-JEFF-VirusV1.31(4.55)+V3.00(3.10)
CompuPhagozyte4
CompuPhagozyte5
CompuPhagozyte6
CompuPhagozyte8
CompuPhagozyte3
CompuPhagozyte7
DARTH VADER V1.1
DISASTER-MASTER V2
LEVIATHAN-Bootblock+Filevirus
Liberator1.21-MemCheck
Liberator3.0-cv
Liberator5.01-pv
LUPO
NANO1
NANO2
NaST
NoVi
Purge
RevengeOfTheLamerExterminator
scsi
SEPULTURA
Sepultura (V2.26)
TeleCom
Terrorists
VT-Faster

Trojanische Pferde (=Programme, die unauffällig Viren installieren)

Neben dieses 'echten' sich verbreitenden Fileviren zählt man auch die sogenannten 'Trojanischen Pferde' zu den Fileviren. Es handelt sich hierbei um Programme, welche unauffällig Viren aller Art installieren. Alle Viren sind beim allerersten Mal darauf angewiesen, daß sie gestartet werden. Die 'Trojanischen Pferde' erreichen das nun dadurch, indem sie sich als harmlose oder interessante Programme tarnen. Wenn der Computeruser diese Programme nun startet, wird unbemerkt der eigentliche Bootblock-, File-, oder Linkvirus installiert. Der Name 'Trojanisches Pferd' entstammt folgender griechischer Sage: Die Stadt Troja war uneinnehmbar. Die Griechen besannen sich daher auf eine List: Sie versteckten Ihre Soldaten in einem großen hölzernen Pferd. Die Trojaner hielten das Pferd für ein Geschenk der Götter und brachten es in Ihre 'uneinnehmbare Stadt'. Nachts kamen die Griechen aus dem Pferd und nahmen Troja ein.

Challenger-Fish622
CLONK-Installer
ColorsVirusCarrierTurkBB
DriveInfo V0.91
EXCREMENT-Installer
GENERALHUNTER V3.2
Intro-Maker V1.00 by TCR
JEFF-Viruskiller
LADS-MVK
LAMER-bomb (Gotcha LAMER)
LAMER-Trojan-Horse (endcli,loadwb,virusx)
MessAngel
MODEMCHECK-FUCK-Virus
scan.x
T.F.C.-loadwb
THE SMILY CANCER II
VIRUS TERMINATORV6.0
VIRUS-INSTALL v2.0
XCopyPro6.5

Trojanische Pferde, die durch einmaliges Virus-Anlinken entstehen

Es werden auch Bootblock-, File-, Link- oder Disk-Validatorviren von Hand an harmlose Programme gelinkt, wodurch ebenfalls 'Trojanische Pferde' entstehen. Es handelt sich hierbei nicht um echte Linkviren, denn diese können sich ja völlig automatisch an weitere Programme linken. Im Moment kann man rein formal vier verschiedene Varianten unterscheiden, die allerdings praktisch betrachtet den gleichen Zweck erfüllen. Wenn VIRUS CONTROL eine Datei als ??? Hunklab-Virus ??? oder ??? XLINK-Virus ??? oder ??? \$4EB9-Virus ??? oder ??? \$4EB9-4EF9-Virus ??? erkennt, dann muß nicht immer ein angelinkter Virus vorliegen, denn VIRUS CONTROL kann nur erkennen, daß von Hand ein Programm an ein anderes Programm gelinkt wurde, das kann ein harmloser Spieltrainer sein, oder aber doch oftmals auch ein Virus, entweder ein bereits bekannter Virus oder aber eine meist gegen Mailboxen gerichtete Virusneuentwicklung.

\$4EB9-Link
\$4EB9-4EF9-Link
Hunklab-Link
XLINK-Link

Programme, die eindeutig erkennbar Viren installieren

Obwohl man diese Programme eigentlich nicht als Viren bezeichnen kann, weil sie keine automatische Schädigung von Computerdaten vornehmen, werden sie dennoch der Vollständigkeit halber erwähnt.

- Bootblock-Massacre
- BootShop
- DAG-Virus-Infecter
- VirusMaker V1.0
- The Virus Construction Set von STR
- The Virus Construction Set von STR V2.0

Daten-Zerstörungsprogramme

Es handelt sich meist nicht um echte sich automatisch verbreitende Viren, sondern eher um oftmals mäßig programmierte Daten-Zerstörungsprogramme. In der Regel wird nach außen hin der Anschein eines harmlosen oder interessanten Programmes erweckt um so ein Starten des Virusprogrammes zu veranlassen. Es handelt sich also meist auch um Trojanische Pferde. Oftmals verraten sich diese Programme auch durch System-Requester, wenn z.B. ein Schreibzugriff auf eine schreibgeschützte Diskette versucht wird, bei Festplatten hingegen wird man weniger gewarnt, da diese leider normalerweise immer beschreibbar sind.

- AAA-Enhancer
- A.I.S.F. INTERLAMER
- Aibon
- Aibon2
- BootX-Updater
- ByteParasiteI
- ByteParasiteII
- ByteParasiteIII
- CHAOS-MASTER V0.5
- Commodore-Virus
- CompuPhagozytel
- CompuPhagozyte2
- CONMAN-TROJAN
- D&A
- Decompiler
- Degrad
- Descriptor V3.0
- DISK-KILLER V1.0
- DiskSpeedCheckV1.01B
- Disktroyer
- D-Structure (A, B, C)
- Elieen
- Excreminator V1.0
- FCheck
- Freedom
- TimeBomb V0.9
- TimeBomber (VIRUSTEST)
- VirusBlaster
- VMK V3.00

Mailbox schädigende Programme

Es handelt sich meist nicht um echte sich automatisch verbreitende Viren, sondern eher um meist mäßig programmierte Daten-Zerstörungsprogramme, die mehr oder weniger zielgerichtet gegen Mailboxsysteme vorgehen, das heißt oftmals werden z.B. lediglich Daten auf einem eventuellen Datenträger namens BBS: gelöscht oder Dateien wie z.B. BBS:User.Data verändert. Oftmals wird von Hand die Virusroutine an ein bekanntes Programm gelinkt wodurch also ein Trojanisches Pferd entsatanden ist, welches dann z.B. Sysops zum Starten des Programmes animieren soll. Angenommen ein Sysop startet so ein Mailbox-schädigendes Programm, dann sieht zunächst alles nach einem harmlosen Programm aus, denn der Sysop soll keinen Verdacht schöpfen, derweil trägt aber das Programm unauffällig einen neuen User in die Systemdateien ein. Der Virusprogrammierer versucht dann einige Wochen später mit diesen Namen und Passwörtern Zugang in verschiedene Mailboxen zu finden, in der Hoffnung, daß einige Sysops das Programm mal ausprobiert haben.

```
AE-Registrator
AmiPatch V1.0a
Dialer V2.8g
dm-trash
DOOM
DOOR_BELLS
DOpusrt
EASY-E
LHACHECK 1.1
LOOK-BBS
M_CHAT V2.3
ModemSpeederV2.1
Mongo
NoGuruV2.0
showsysops
SwiftWare-DevilDoor8
SysinfoV2.2
timer
Top util V1.0
TROJAN KILLER V3.0
Viewtek
XPR-SpeederV3.2
```

1.301 amigaknights

AMIGAKNIGHTS

Es handelt sich um ein 6048 Byte langes Programm. Es bestehen sehr starke Ähnlichkeiten mit dem Butonic-Filevirus. Der AMIGAKNIGHTS-Filevirus macht sich über die Kick-Vektoren resetfest, wobei aber der Virus auf Amigas mit Fast-RAM meist nicht resetfest ist. Als Kick-Identifikationsstring wird Daten-Müll benutzt. Der AMIGAKNIGHTS-Filevirus verbiegt weiterhin den DoIO()-Vektor, um Disketten beim Einlegen zu infizieren. Hierbei wird das Virusfile unter dem Namen initial_cli in das Basisverzeichnis der eingelegten Diskette geschrieben. Weiterhin wird initial_cli an den Anfang der Startup-Sequence geschrieben, damit das Virusfile bei jedem Booten

aufgerufen wird. Nach 5 Resets gibt sich der Virus zu erkennen.
Es erscheint der folgende rosafarbene Text auf schwarzem Hintergrund:

```
YEAH, THE INVASION HAS STARTED! YOUR  
TIME HAS RUN OUT, AND SOON WE WILL BE  
EVERYWHERE!
```

```
THIS IS GENERATION 0039 OF THE EVIL  
AMIGAKNIGHTSVIRUS  
GREETINGS TO DUFTY, DWARF, ASID CUCUMBER  
ASTERIX, ANDY, AND ALL AMIGIANS I KNOW
```

Zwischen diesen beiden Textblöcken wird eine mittelmäßige Vektor-Grafik angezeigt, wobei folgender Text erscheint: Toco of THE AMIGAKNIGHTS

Sollte ein AMIGAKNIGHTS-Virusfile gefunden werden,
dann können Sie dieses Virusfile löschen lassen.
Sollte in der ersten Zeile der Startup-Sequence `initial_cli` stehen,
dann werden diese Zeichen mit Leerzeichen überschrieben,
wodurch das AMIGAKNIGHTS-Virusfile beim Booten nicht mehr
aufgerufen wird.

1.302 anti-euromail-virus

Anti-EuroMail-Virus

Es handelt sich um einen Filevirus, das heißt, der Virus steht in einem Programm. Der Anti-EuroMail-Filevirus wurde unter dem Namen QuickInt in das Zerberus-Netz gesetzt. QuickInt kann man also auch als Trojanisches Pferd betrachten. Beim Aufrufen des Programms wird der Virus aktiviert. Hierbei gibt der Virus folgende Meldung aus:
Unable to load : object not of required type
Damit soll der User in Sicherheit gewogen werden, da dieser nun glaubt, daß das Programm vom Amiga-Betriebssystem nicht ausgeführt werden konnte, und daß somit auch keinerlei Gefahr bezüglich einer eventuellen Virus-Aktivierung bestehen kann. Dennoch ist der Virus jetzt aktiv und erstellt einen Prozess namens 'clipboard.device'. Weiterhin schreibt der Virus nun an den Anfang der Startup-Sequence \$a0\$0a und erstellt im C-Verzeichnis ein 3196 Byte langes Virusfile unter dem Namen \$a0, allerdings nur wenn als Name EM, EUROMAIL oder EUROSYS vorliegt. Dadurch wird der Virus also trotz fehlender Resetfestigkeit bei jedem Booten wieder aktiviert. Zumindest war dies so geplant, denn der mir vorliegende Anti-EuroMail-Filevirus weist einen Fehler auf, denn er setzt bei seinem C:\$a0-Virusfile mittels protect nur das w-Flag und d-Flag. Da das r-flag gelöscht ist, kann das Virusfile beim Booten nicht eingelesen und auch nicht ausgeführt werden. Die Abarbeitung der Startup-Sequence bricht mit einer Fehlermeldung ab. Dies gilt jedoch nur unter Kickstart 2.0 oder beim Booten von einer FFS-Festplatten-Partition. Wenn unter Kickstart 1.3 von einer Diskette gebootet wird, dann werden die protection-bits ignoriert und der Virus ist funktionsfähig. Der Anti-EuroMail-Filevirus liegt als ein im MasterMode des PowerPacker gepacktes file vor. Hierdurch sollen verräterische Texte versteckt werden. Die Filelänge beträgt 3196 Bytes, entpackt 3888 Bytes. Der File-Inhalt von \$a0 und QuickInt ist 100% identisch. Das wirklich Bösertige an dem Anti-EuroMail-Filevirus ist, daß er nach einigen Minuten alle Dateien in einem eventuellen EM: oder EUROMAIL:

oder EUROSYS: - Verzeichnis mit unsinnigen Daten überschreibt. Der Virus schädigt also hauptsächlich EuroMail-Besitzer. Überprüfen Sie also die erste Zeile Ihrer Startup-Sequence auf das Vorhandensein einer eventuellen Leerzeile und löschen Sie diese Zeile.

1.303 bgs9 i+ii+iii

BGS9 I+II+III

Der BGS9-Virus ist ein relativ ungefährlicher und neuartig programmierter Virus. Es handelt sich um einen sogenannten Filevirus. Das heißt, der Virus verbreitet sich direkt als File. Dieses Virusfile ist 2608 Bytes lang und hat den Namen des ersten Programms der Startup-Sequence. Das frühere Original-File steht nun unter einem unlesbaren Namen in devs: (devs:a0a0a0202020a0202020a0). Der BGS9-Virus wird also in der Regel automatisch beim Abarbeiten der Startup-Sequence aktiviert. Hierbei macht sich der Virus, wenn noch nicht geschehen, über Kick-Mem, Kick-Tag resetfest. Hierbei werden andere Kick-resetfeste Programmen, wie z.B. RAD: nicht gelöscht. Der BGS9-Virus ist auch in anderer Hinsicht flexibel programmiert, daß heißt, er arbeitet auch mit Fast-RAM oder Kickstart 1.2 und 1.3. Nachdem sich der Virus also resetfest gemacht hat, wird nun, um unauffällig zu bleiben, das eventuelle frühere Original-File in devs: ausgeführt. Beim Ausführen des Virusfiles passiert also außer dem Resetfestmachen und der Ausführung einen eventuellen devs:Original-Files nichts!! Die Verbreitung des Virus erfolgt beim Bootvorgang. In der resetfesten Kick-Routine wird der OpenWindow()-Vektor verbogen. Beim nächsten OpenWindow()-Aufruf wird nun zuerst versucht, das File devs:a0a0a0202020a0202020a0 zu öffnen. Wenn dies fehlschlägt, ist die vorliegende Diskette noch nicht infiziert. In diesem Falle wird nun aus sys:s/Startup-Sequence der erste Filenamen geholt und dieses File, notfalls unter Hinzufügung von c/, in devs:a0a0a0202020a0202020a0 umbenannt. Anschließend wird unter dem Original-Filenamen das 2608-Byte lange Virusfile erstellt. Bei einer schreibgeschützten Diskette erscheinen keine verräterische System-Requester, da diese zuvor durch Setzen von WindowPrt auf -1 verboten wurden. Es wird nun wieder der OpenWindow()-Vektor restauriert und bei jedem 4 Reset eine Bildschirmmeldung ausgegeben. Bis zum nächsten Reset passiert nun absolut nichts mehr!! Der BGS9-Virus ist also recht einfach entfernbar, indem man das erste Programm der Startup-Sequence, das 2608 Bytes lang ist, löscht. Danach kopiert man das Original-File aus devs: mittels eines Disketten-Hilfsprogramms unter den Original-Namen zurück. Wenn eine Diskette kein devs-Verzeichnis besitzt, dann ordnet das Betriebssystem devs: dem Basisverzeichnis zu. In diesem Fall wird mit devs: also das Basisverzeichnis angesprochen, und das Originalfile wird auch hier abgelegt.

Es gibt eine Abart des BGS9-Filevirus (BGS9-II), welcher anstelle devs:a0a0a0202020a0202020a0 devs:a0e0a0202020a0202020a0 verwendet. e0 ist als einziges Zeichen als à sichtbar.

Ein weiterer BGS9-Abkömmling (BGS9-III), verwendet anstelle devs:a0a0a0202020a0202020a0 devs:a0

Noch ein Wort zu VirusX4.01 usw.:

VirusX4.01 geht nach folgendem Prinzip vor. Wenn in devs: ein vom BGS9-Virus verstecktes Programm gefunden wird, dann nimmt Virusx4.01 starr an, daß nun

unter dem ersten Filenamen der Startup-Sequence das BGS9-Virusfile steht. Dies muß aber nicht so sein, denn oftmals wird ja die Startup-Sequence abgeändert usw. Wenn nun mittlerweile ein harmloses Programm als erstes File in der Startup-Sequence steht, dann kann VirusX4.01 natürlich den BGS9-Virus hier nicht finden, stattdessen behauptet VirusX, es läge ein BGS9-ähnlicher Virus vor. Dadurch werden völlig saubere Programme in Mißkredit gebracht. Bei einer frisch infizierten Diskette stimmen die Aussagen von VirusX, bei einer älteren und abgeänderten Diskette aber steht womöglich nur noch das unsichtbare Original-File im devs-Verzeichnis, die ganze Diskette aber ist sauber, dennoch weist VirusX auf einen Virus hin. Das Ganze ist durchaus gut gemeint von VirusX, führt aber oftmals zu Fehlschlüssen.

1.304 bluebox-icon.library

BLUEBOX-icon.library

Wenn man das 5608 Byte lange, PowerPacker-artig gepackte, Bluebox-Programm startet, dann wird eine bestehende libs:icon.library durch eine 6680 Byte lange gleichnamige Virus-library ersetzt. Danach erscheint das eigentliche BLUEBOX-Programm-Fenster.

Der eigentliche Virus befindet sich in der icon.library.

Beim nächsten Booten wird beim Ausführen des loadwb-Befehls die icon.library vom Betriebssystem automatisch aufgerufen, wodurch dann auch der Virus aktiviert wird.

Unter Kickstart 2.0 befindet sich die icon.library im ROM, so daß eine etwaige icon.library auf Diskette ignoriert wird.

Unter Kickstart 2.0 wird der BLUEBOX-Virus also nie aktiv.

Unter Kickstart 1.3 wird der Virus allerdings aktiviert.

Hierbei wird ein Prozess mit dem unauffälligen Namen 'input.device' erstellt und es wird weiterhin der Interrupt-5

(Eingabepuffer des seriellen Ports voll) verbogen.

Der BLUEBOX-Virus scheint also nicht direkt schädlich zu sein.

Allerdings kann es zu Problemen bei der Benutzung von DFÜ-Programmen kommen, da der BLUEBOX-Virus die serielle Datenübertragung stören kann, da er sich nicht an die üblichen Programmierrichtlinien für das serielle Device hält, sondern die Hardware direkt anspricht.

Ein BLUEBOX-Programm soll ein kostenloses Telefonieren ermöglichen.

Hierzu werden unter anderem gewisse Töne generiert.

Nur wenige Leute werden sich dieser kriminellen und unzuverlässigen Methode bedienen, zumal sich die Details diese Methode ändern.

Es werden also des öfteren neue BLUEBOX-Programme erforderlich, womit also das vorliegende BLUEBOX-Programm wohl eh veraltet ist.

1.305 bret-hawnes

BRET-HAWNES

Es handelt sich um einen Filevirus, das heißt, der Virus steht in einem eigenen File. Der Virusname BRET HAWNES rührt daher, daß man am Ende des 2608 Byte langen Programms folgenden Text lesen kann:

U LIKE MY FIRST LINKVIRUS ? DONE BY BRET HAWNES 210290

Entgegen dieser Meldung handelt es sich aber bei dem BRET-HAWNES-Virus nicht um einen Linkvirus, sondern um einen Filevirus. Wenn man das BRET-HAWNES-Virusfile aufruft, dann kopiert sich der Virus nach \$7f000 und macht sich als alleiniges Programm über die Kick-Vektoren resetfest. Der COOL-Vektor wird gelöscht. Interessant wird es nach dem nächsten Reset. Hierbei wird der OldOpenLib()-Vektor verbogen, um auf einen OldOpenLib('intuition')-Zugriff zu prüfen. Wenn dies der Fall ist, wird der OpenWindow()-Vektor der intuition.library verbogen. Beim ersten Aufruf der OpenWindow()-Funktion wird nun eine Infektion versucht. Anschließend wird der OpenWindow()-Vektor wieder restauriert. Der OldOpenLib()-Vektor bleibt verbogen, was nun aber keine Konsequenzen mehr hat, da nun auf eine unmögliche Library geprüft wird. Der langen Rede kurzer Sinn, nach jedem Reset wird ein Infektionsversuch unternommen. Dieser sieht folgendermaßen aus: Der Virus trägt den Aufruf seines Virusfiles in s/Startup-Sequence ein. Sollte die Startup-Sequence länger wie 1024 Bytes sein, dann werden die restlichen Bytes einfach abgeschnitten. Die Startup-Sequence beginnt nun mit \$c0a0e0a0c00a. Das Virusfile mit Namen \$c0a0e0a0c0 wird nun also nach jedem Reset durch diese 6 Bytes (5 Bytes Virusfilename + 1 Linefeed) zu Beginn der Startup-Sequence aufgerufen. Nachdem also die Startup-Sequence abgeändert wurde, wird nun noch das eigentliche 2608 Bytes lange Virusfile \$c0a0e0a0c0 im Basisverzeichnis erstellt. Weiterhin verbiegt der Virus den Autointerrupt 3 (\$6c), um nach 20 Minuten folgende Meldung in weißer Schrift auf blauem Hintergrund auszugeben:

```
GUESS WHO'S BACK ??? YEP. BRET HAWNES BLOPS YOUR SCREEN
  I'VE TAKEN THE CONTROL OVER YOUR AMIGA !!!
THERE'S ONLY ONE CURE: POWER OFF AND REBOOT ! ! ! ! !
```

Diesen Text kann man in dem Virusfile nicht lesen, da es sich um direkte BitMap-Daten handelt, welche mittels direkter Copper-Programmierung sichtbar werden. Ein Weiterarbeiten ist nun nicht mehr möglich. Ein Auschalten des Computers wird also nötig. Wenn die zehnte Diskette infiziert wurde, dann wird diese anschließend teilweise formatiert und es erscheint die eben erwähnte Meldung. Dieses Formatieren ist das eigentlich Bösertige des BRET-HAWNES-Filevirus.

1.306 butonic-jeff

BUTONIC-JEFF-VirusV1.31(4.55)+V3.00(3.10)

Der BUTONIC-JEFF-Virus trägt diesen Namen, weil bei einem Reset manchmal folgende Alert-Meldung ausgegeben wird:

```
HI.
JEFF's speaking here ...
(w) by the genius BUTONIC
V 3.00/9.2.89-Gen.00025
```

```
Greetings to *Hackmack*,*Atlantic*,
& Alex, Frank, Wolfram, Gerlach, Miguel, Klaus, Snoopy-Data!
```

Das Erscheinen des Alert ist unregelmäßig, da das Erscheinen auch von einem

Zufallswert (Strahlenposition) abhängig ist. Durch gleichzeitiges Drücken beider Maustasten und der y-Taste kann man jedoch das Erscheinen des Alerts erzwingen. Bei jedem zweiten Einlegen einer nicht schreibgeschützten Diskette wird der Titels des aktuellen Windows verändert. Diese Änderung wird allerdings erst nach einer Veränderung des Windows sichtbar (also z.B. wenn man das Sizing oder Window-Depth-Gadget betätigt). Folgende WindowTitel können erscheinen:

```
Ich brauch jetzt Alk'!  
Bitte keinen Wodka!  
Stau auf Datenbus bei Speicherkilometer 128!  
Mehr Buszyklen für den Prozessor!  
Ein dreifach MITLEID für Atari ST!  
©89 by BUTONIC  
PC/XT: Spendenkonto 004...  
Freiheit für den Tastaturprozessor!  
C für Looser  
Paula meint, Agnus sei zu dick.  
Die CPU braucht etwas Schmieröl  
C64 - jetzt mit Pampers im 3erPack  
JEFF=ungefährlich+schützt vor Viren
```

Sowohl der Alert-Text wie auch die Window-Titel liegen kodiert vor und sind dadurch mit einem Filemonitor oder Diskettenmonitor nicht zu erkennen. Der BUTONIC-JEFF-Virus-V3.00 hat absolut nichts mit dem DASA-Bootblockvirus gemein. Es ist vielmehr purer Zufall, daß DASA im Virusprogramm steht. Zufälligerweise erscheint eine gewisse Programmstruktur in ASCII-Darstellung als DASA. Der BUTONIC-JEFF-Virus-V3.00 ist nicht direkt bösartig. Es handelt sich um einen sogenannten Filevirus. Das heißt, der Virus steht konkret in einem Programm und wird durch den Aufruf dieses Programms aktiviert. Der Aufruf des Virusprogramms wird dadurch sichergestellt, indem dieser Aufruf an den Anfang der Startup-Sequence geschrieben wird. Der Filename des BUTONIC-JEFF-Virus-V3.00 lautet \$a0a0a0, er besteht also aus unsichtbaren Leer-Zeichen. Hierdurch wird oftmals das Vorhandensein des zusätzlichen Virusfiles übersehen. In die startup-sequence wird \$a0a0a0209b41 eingetragen, also zusätzlich zu dem Filenamen noch eine Cursor-Up-Sequenz, wodurch z.B. beim type-Befehl der Virusfilename versteckt wird, das heißt es erscheint auch keine verräterische Leerzeile mehr, da diese durch Cursor-Up und die nächste Text-Ausgabe überschrieben wird.

Beim Aufruf des Virusfiles, welches 2916 Bytes lang ist, werden DoIO, KickTag und KickChecksum verändert. DoIO wird verbogen, um soeben eingelegte Disketten infizieren zu können. KickTag und KickChecksum wird verändert, um sich resetfest zu machen. Leider werden hierbei andere Kick-resetfeste Programme wie RAD: oder turboprint herausgeworfen. Beim nächsten Disketten-Einlegen wird dann noch Cold und Cool gelöscht. Der BUTONIC-JEFF-Virus ist nun also das einzige resetfeste Programm. Man kann den Virus aus dem Speicher entfernen, indem man beim Disketteneinlegen beide Maustasten und L-ALT und L-AMIGA drückt. Der Virus infiziert nur Disketten!! Weiterhin werden nur Disketten mit einer Startup-Sequence infiziert, welche kürzer als 480 Bytes ist. Es wird immer versucht beim Booten die Bootdisk zu infizieren. Weiterhin wird bei jeder zweiten eingelegten Diskette ein Infektionsversuch unternommen. Wird beim Disk-Einlegen L-Amiga gedrückt, so wird jedes Mal ein Infektionsversuch unternommen. Wird beim Disk-Einlegen L-Shift gedrückt, so wird nie ein Infektionsversuch unternommen. Der BUTONIC-JEFF-Virus greift direkt mittels des trackdisk.devices auf die Diskette zu. Es werden also keine Open, Write usw. Befehle der dos.library

benutzt. Diese Methode ist recht schwierig zu programmieren, da man vollkommen von Hand die komplizierte Amiga-DOS-Disketten-Verwaltung nachbilden muß. Deswegen auch die doch recht lange Virusfilelänge von 2916 Bytes. Man muß dem BUTONIC-JEFF-Virusprogrammierer also durchaus beachtliche Programmierkenntnisse bescheinigen. Bedauerlich, daß sie für Virenprogrammierung verschwendet werden. Allerdings kann es in seltenen Fällen auch zu Disketten-Fehlern kommen, wenn das Amiga-DOS gleichzeitig mit dem Virus die Diskette beschreibt. Es wäre denkbar, daß beide die gleichen Blöcke beschreiben, da beide ja nicht zusammenarbeiten. Trotz der recht sauberen Programmierung ist dennoch ein schwerwiegender Fehler vorhanden. Auf Rechnern mit echtem FastRam wird der Virus nämlich meist nicht resetfest sein, da die Kick-Struktur im am höchsten priorisierten Speicher angelegt wird. Bei einem Reset werden aber nur Kick-Strukturen im Chip- und Ranger-RAM gefunden. Die Kick-Struktur kann also nicht gefunden werden, und der Rechner bleibt hängen. Aber man darf diesen Fehler wohl eher als einen Vorteil betrachten, da dadurch der Virus nicht mehr resetfest ist.

Ein Vorgänger des Butonic V3.00 war der Butonic V1.31. Im Prinzip arbeiten beide Fileviren gleich. Es existieren folgende Unterschiede:
Der Butonic V1.31 verbiegt auch noch Auto-Interrupt 2 und speichert das Virusfile unter mehreren verschiedenen Namen ab: AddBuffers, Add21K, Fault, break, changetaskpri, wait, \$a0, \$a0\$a0\$a0, Arthus, Helmar, Aloisius.
Der Butonic V3.00 ist besser programmiert und auch die Textmeldungen lauten geringfügig anders. Es wird nicht mehr der Auto-Interrupt 2 verbogen und es wird das Virusfile immer nur unter dem Namen \$a0\$a0\$a0 erstellt.

Mittlerweile ist noch ein V1.31-Abkömmling, welcher auch über ein Trojanisches Pferd namens snoopdos1.9 verbreitet wird, aufgetaucht, bei welchem lediglich die Texte abgeändert wurden.

Alertmeldung:

Hallo hoffentlich stoere ich sehr !

* I am JEFF - the old Virus family for an Amiga *
(w) by the nicely BUTONIC.
V 4.55/29.02.93 - Generation Nr.00037

Killings goto* BootX *,* VirusZ *, Virus_Checker ,
Viruscope, Maus , Virus-Checker , Virus Control and big VT !!

Fenstertiteländerungen:

Hallo gib die Cola her !
Lass die Chips roesten und nicht rosten !!!
Nimm die Birne weg sonst krachts!
Wenn Du nicht spurst dann gibts \$!
BoTiNuC!
Schaem Dich Du Banause lass es sause Junge ...aber nicht schlappi
Willst Du Nachhilfe oder was is los ?
Gib es auf Du lahmer socke
Wer andern eine Grube graebt faelltselfst in dieselbige !!!
Wo willst den jetzt wieder hin
Kannst Du mal Ruhe geben Du alter Knochen-Kerl ...
Liebst Du Viren, dann weiss ich auch, wer Dich am meisten hasst

Das Virusfile wird unter verschiedenen Namen abgespeichert, wodurch auch

verschiedene startup-sequence-Eintragungen vorgenommen werden:

```
'LoadWB      '$0a
'Mount      ', $0a
'Cls        ', $0a
'VirusY      ', $0a
'setclock opt i ', $0a
'info      ', $0a
$A020, $0A
$A0A0A020, $0a
'Obelix     ', $0a
'Idefix     ', $0a
'Asterix     ', $0a
```

1.307 compuphagozyte4

CompuPhagozyte4

Es handelt sich um ein 916 oder 952 Byte langes Programm, welches Speicher ab \$7C000 belegt und sich über den COOL-Vektor resetfest zu machen versucht. Es wird der OldOpenLibrary()-Vektor verbogen, um bei jedem OldOpenLibrary()-Aufruf eine Disketteninfektion zu versuchen, das heißt es wird das 916 oder 952 Byte lange Virusfile unter dem Namen DF0:a0,a0,a0,a0 neu geschrieben und in DF0:s/startup-sequence die ersten 5 Bytes mit dem Aufruf des Virusprogrammes, nämlich a0,a0,a0,a0 und 0a als Linefeed überschrieben. Vermutlich wird dadurch die Abarbeitung der startup-sequence nach dem Aufruf des Virusprogrammes abbrechen, weil die früheren Original-Programm-Aufrufe meist teilweise überschrieben sind und somit nun syntaktisch falsch sind. Da nur unter Kickstart 1.2 der OldOpenLibrary()-Vektor verbogen wird, erfolgt nur unter Kickstart 1.2 eine Virusverbreitung. Gegen Ende des Virusfiles kann man lesen:

The might of The Emperor is unlimited!!! COMPUPhagozyte !!!

Es existiert ein CompuPhagozyte4-Abkoemmling namens VT-Faster, in welchem folgender Täuschungstext zu lesen ist:

Use VT the best viruskiller of the world! Max of Starlight

Anstatt DF0:a0,a0,a0,a0 in DF0:s/startup-sequence einzutragen, wird nun HD0:c/VT in HD0:s/startup-sequence eingetragen.

1.308 compuphagozyte5

CompuPhagozyte5

Es handelt sich um ein 892, 900 oder 936 Byte langes File. Es bestehen lediglich folgende Unterschiede zu CompuPhagozyte4, anstatt starr 'DF0:' zu infizieren wird nun universell ':' verwandt und der Text 'COMPUPhagozyte' ist vom Fileende an den Fileanfang verschoben worden. Außerdem ist nun die Virusinfektion nicht mehr auf Kickstart 1.2

beschränkt.

1.309 compuphagozyte6

CompuPhagozyte6

Es handelt sich um ein 1008 oder 1048 Byte langes File, welches gegenüber CompuPhagozyte5 insbesondere folgende Unterschiede aufweist. Am Fileanfang ist folgender Text zu lesen:

```
>COMPUPhagozyte Protection File Wjsvt
```

Gegen Ende des Files befindet sich kodierter Text, der nach Subtraktion mit 1 folgendes ergibt:

```
The Return of The Emperor Of Trillion Bytes
```

Es wird zusätzlich der SumKickData()-Vektor verbogen, um die COOL-Resetfestigkeit zu erhöhen und um COLD- und Kick-Vektoren zu löschen.

Es werden nun nicht mehr die ersten 5 Bytes der startup-sequence mit a0,a0,a0,a0, 0a überschrieben, denn dadurch bricht die startup-sequence oftmals nach dem Virusaufruf vorzeitig und somit verräterisch wegen Syntax-fehler ab. Nein, jetzt werden 6 Bytes a0,a0,a0,a0,20, 0a als zusätzliche erste Zeile vor die bereits bestehenden Zeilen eingefügt. Da aber immer 1006 Bytes startup-sequence geschrieben werden, wird im Falle einer kürzeren startup-sequence Müll angehängt, was dann auch zu Fehlern führt, was aber oftmals nicht auffällt, weil ja vorher alle Programme korrekt aufgerufen werden konnten.

Wenn die startup-sequence allerdings größer 1000 Bytes ist, dann tritt ein anderer Fehler auf, denn es werden dann die Bytes 1000 bis 1005 der startup-sequence gewissermaßen verschluckt, weil 1006 Bytes (6 Virus-Bytes + 1000 Original-Bytes) über die startup-sequence geschrieben werden. Ab Position 1006 bleibt die startup-sequence aber unverändert. Diese neue startup-sequence-Methode ist also auch nicht viel besser wie die alte Methode.

CompuPhagozyte6 prüft nicht ob bereits eine Infektion vorliegt und schreibt somit immer wieder eine weitere neue Virusprogrammaufrufzeile an den Anfang der startup-sequence.

1.310 compuphagozyte8

CompuPhagozyte8

Es handelt sich um ein 1952 Byte langes File, welches auf CompuPhagozyte6 aufbaut. Folgende Unterschiede sind vorhanden. Am Fileanfang ist insbesondere aus Tarngründen folgender Text zu lesen:

```
      :AmigaDOS Datafile @ 1988 by CBM  
This file contains important disk data for Block Allocation !
```

```
>>> WARNING: Deletion of this file could destroy all disk datas !!! <<<
```

Gegen Ende des Files befindet sich kodierter Text, der nach Subtraktion mit 1 folgendes ergibt:

```
The Emperor Of Trillion Bytes presents:...  
a new CompuPhagozyte !!!
```

Während bei CompuPhagozyte4,5,6 die Virusverbreitung in den OpenOldLibrar()-Vektor eingeschleift war, wird nun stattdessen der Open()-, Lock()- und loadseg()-Vektor der dos.library zu diesem Zweck verbogen. Da diese Vektoren sehr häufig benutzt werden, kommt es sehr schnell zur Virusverbreitung. Weiterhin wird noch in den DoIO()-Vektor eine Routine eingeschleift, welche darauf achtet, daß der Open()-, Lock()- und loadseg()-Vektor auf den Virus verbogen bleibt. Neben dem SumKickData()-Vektor wird nun auch noch der Autointerrupt3-\$6c verbogen, um die alleinige Resetfestigkeit des Virus sicherzustellen.

Das Schreiben des Virusfiles und das Verändern der startup-sequence erfolgt wie bei CompuPhagozyte6 beschrieben, neu ist allerdings, daß noch zusätzlich die protectionbits verändert werden, das heißt bei dem Virusfile wir nur noch E(execute,ausführbar) erlaubt und bei der startup-sequence E(execute,ausführbar) und R(read,lesbar), da bei dem Virusfile R(read,lesbar) nicht gesetzt ist, lehnt Kickstart 2.0 ein lesen und somit auch starten des Programmes ab, Kickstart 1.3 ist hier toleranter.

Im Gegensatz zu CompuPhagozyte6 wird keine Mehrfachinfektion vorgenommen.

Die unterschiedlichen Längen funktionell identischer Compuphagozyten-Files sind durch eine zusätzliche (überflüssige) Exechecksum-Berechnung zur Sicherstellung der Resetfstigkeit bedingt. Manchmal ist auch eine move-Kopierschleife durch ein copymen()-Aufruf ersetzt worden.

Neben den 'lediglich' Daten-zerstörenden CompuPhagozyte1,2 und den Fileviren CompuPhagozyte4,5,6,8 gibt es der Vollständigkeit halber noch zwei weitere CompuPhagozyte-Exemplare zu nennen: Die harmlose CompuPhagozyte3 und der mißglückte Linkvirusversuch CompuPhagozyte7 .

1.311 compuphagozyte3

CompuPhagozyte3

Dieses 568 oder 592 Byte lange File zählt auch zur CompuPhagozyten-Familie, denn es ist folgender Text enthalten:

```
The COMPUPhagozyte in 9.91 !!!  
The Emperor Of Trillion Bytes strikes back !!
```

Das Programm löscht das CLI-Fenster, indem 30 Linefeeds ausgegeben werden und verbiegt den COOL-Vektor auf \$7C000, wo eine resetfeste Routine angelegt wird, die lediglich die COLD-, Warm- und Kick-Vektoren

löscht. Es handelt sich also nicht um einen Virus.

1.312 compuphagozyte7

CompuPhagozyte7

Man kann gegen Ende der 2300 Byte langen Datei folgenden Text lesen:

CompuPhagoLink by The Emperor Of Trillion Byte

Es handelt sich also anscheinend um den Versuch einen Linkvirus zu programmieren, der allerdings im vorliegenden Fall völlig gescheitert ist, denn das vorliegende File kann nicht gestartet werden, denn das Betriebssystem kann das File nicht laden, weil absolut wichtige Angaben, wie z.B \$3F2-Endekennungen völlig fehlen.

1.313 darth vader v1.1

DARTH VADER V1.1

Dieser 784 Byte lange Filevirus tritt unter dem unlesbaren Namen \$A0 auf. Sollte man dieses Programm umbenennen, dann stürzt dieses umbenannte Programm beim Starten ab, da der DARTH VADER-Filevirus fest von einem Dateinamen \$A0 ausgeht. Den Inhalt dieses Virusprogramms legt er in einem Puffer ab. Weiterhin macht sich der Virus über den COOL-Vektor resetfest, was aber in Gegenwart von Fast-RAM versagt. Nach dem Reset wird der OldOpenLibrary()-Vektor verbogen. Wenn nun später ein Programm diesen OldOpenLibrary()-Vektor aufruft, dann schreibt sich der Virus unter dem Filenamen \$A0 auf die Diskette. Weiterhin wird in die erste Zeile der Startup-Sequence \$A00A eingefügt, womit der Aufruf des Virus sichergestellt wird. Nach sechs Infektionen gibt sich der Virus zu erkennen, indem folgender Text ausgegeben wird.

VIRUS(V1.1) BY DARTH VADER

1.314 disaster-master v2

DISASTER-MASTER V2

Es handelt sich um einen Filevirus, das heißt, der Virus steht in einem Programm und wird durch das Starten dieses Programms aktiviert. Man erkennt den Virus durch das Vorhandensein eines 1740 Byte langen Programms namens cls im c-Verzeichnis, welches automatisch durch den ersten Befehl der Startup-Sequence cls * aufgerufen wird. Erfreulicherweise ist dieser Virus nicht direkt bösartig. Wenn man das cls-Programm aufruft, macht sich der Virus durch Verändern von KickTagPtr und KickChecksum resetfest. Hierbei werden allerdings andere Kick-resetfeste Programme wie z.B. RAD: oder turboprint entfernt. Solche Programme sind also nach dem nächsten Reset nicht mehr vorhanden. Beim Aufrufen des Programms wird gemäß des Basic-Befehls CLS der Bildschirm gelöscht. Durch dieses Verhalten tarnt sich

also der Virus. Ruft man `cls` jedoch mit `*` auf, dann wird der Bildschirm nicht gelöscht. Hiervon macht der Virus selber Gebrauch, denn er ändert, wie weiter unten beschrieben, die Startup-Sequence dahingehend ab, daß er als ersten Programmaufruf `cls *` hineinschreibt. Würde er nur `cls` schreiben, dann würde sich der Virus doch sehr schnell durch das Löschen des Bildschirms verraten. Beim Aufrufen des Virusprogramms werden also lediglich die beiden Kick-Vektoren verändert. Sonst passiert nichts! Interessant wird es erst beim nächsten Reset. Beim Abarbeiten der `rt_init`-Routine der resetfesten Kick-struktur wird der `DoIO()`-Vektor verbogen, damit der Virus beim gleichfolgenden Disketten-Bootversuch wieder angesprungen wird. Hier wird nun der Cold und COOL-Vektor gelöscht und der Virus macht sich sicherheitshalber noch einmal resetfest. Dann wird der `DoIO()`-Vektor wieder restauriert, dafür wird nun aber der `OpenWindow()`-Vektor verbogen, damit der Virus vor dem gleich folgenden Öffnen des AmigaDOS-CLI-Windows noch einmal angesprungen wird. Hier nun findet die Disketten-Infektion bzw. die Weiterverbreitung des Virus statt. Ein früherer Zeitpunkt ist auch nicht möglich, da dann die `dos.library` noch nicht eingerichtet wäre. Es wird nun zuerst der `OpenWindow()`-Vektor wieder restauriert und dann ein Infektionsversuch unternommen. Erfreulicherweise geht der Virus hierbei recht behutsam vor. So wird geprüft, ob die Diskette validiert und beschreibbar ist, und ob auch noch genügend Platz vorhanden ist, um das `cls`-Programm aufzunehmen. Diese Prüfungen werden natürlich auch aus dem Grunde unternommen, um dem Erscheinen von verräterischen System-Requestern vorzubeugen. Wenn keine `:s/Startup-Sequence` oder kein `c`-Verzeichnis vorhanden ist, erfolgt keine Infektion. Andernfalls wird das eigentliche Virusprogramm unter dem Namen `cls` im Verzeichins `:c` erstellt. Danach wird der Virusfile-Aufruf `cls *` als erste Zeile in die Startup-Sequence geschrieben. Die ursprüngliche Startup-Sequence wird dann angehängt. Es wird nun bei jedem Booten beim Abarbeiten der Startup-Sequence automatisch der Virus aktiviert. Der Virus kann sich deswegen verbreiten, weil er resetfest ist. Hierdurch kann er sich also auf noch nicht infizierte Bootdisketten kopieren. Der Virus infiziert auch eine Festplatte. Man entfernt den Virus einfach durch Löschen des Virusfiles `:c/cls` und durch Löschen des Aufrufs in der Startup-Sequence `cls *` Wenn die Infektion erfolgreich verlief, wird ein Zähler um 1 erhöht. Nach 20 erfolgreichen Infektionen gibt sich der Virus dann offen zu erkennen. Es wird ein Alert ausgegeben.

```
Software Failure.  Press left mouse button to continue
Guru Meditation #00000002.06001989
```

Nach Drücken der rechten oder linken Maustaste erscheint ein zweiter Alert

```
Incoming special-message
Your Amiga is infected by DISASTER-MASTER V2 !!!
probably the best virus ever created by mankind
Left = continue  Right = self-destruction...
```

Nach Drücken der linken Maustaste wird normal weiter gebootet. Nach Drücken der rechten Maustaste erscheint ein Farbmuster auf dem Schirm. Man kann die Anzeige dieses Farbmuster nur durch Reset verlassen. Man ist nun aber in einer Reset-Endlosschleife gefangen und kommt nicht umhin, den Amiga ganz auszuschalten. Die Alerts erscheinen also recht selten. Der Virus gibt sich nun aber auch noch durch folgendes Verhalten zu erkennen. Er manipuliert zufallsgesteuert (Rasterzeile) das gleich erscheinende AmigaDOS-CLI-Window, indem er die `newWindow`-Struktur verändert. So wird entweder die Titelzeile verändert oder die Windowgröße auf 320,32 gesetzt oder aber der BlockPen auf 3 gesetzt oder als letzte Möglichkeit ein rahmenloses Window definiert. Bis

zum nächsten Reset passiert nun nichts mehr. Der DISASTER-MASTER V2 - Virus ist für ein Virusprogramm bemerkenswert sauber programmiert. Der Code ist allerdings nicht sonderlich optimiert, da z.B. short-branches und moveq-Befehle nicht konsequent genutzt werden. Wenn Ihr Amiga über echtes Fast-RAM (z.B. \$200000-\$400000) verfügt, dann ist der DISASTER-MASTER V2 Virus meist nicht resetfest. Vielmehr bleibt Ihr Amiga dann meist in der Resetroutine hängen. Der Grund hierfür liegt in einem Programmierfehler. Das Betriebssystem sucht nach Resident-Strukturen nur in \$0-\$200000 und \$c00000-\$dc0000. Da der DISASTER-Master V2 bei seiner Installation keinen speziellen Speicher anfordert, wird die Kick-Struktur meist im höher priorisierten Fast-RAM angelegt. Hier kann sie bei einem Reset aber nicht gefunden werden. Negativ anzumerken ist auch, daß alle sonstigen resetfesten Programme gelöscht werden. Aber dieses geschieht wohl eher absichtlich. Dieser Virus gibt sich durch Ansehen des Programms mit einem Filemonitor nicht zu erkennen, da die Texte kodiert vorliegen. Der DISASTER-MASTER V2 - Virus wurde auch über ein Trojanisches Pferd namen Intro-Maker V1.00 by TCR verbreitet.

1.315 leviathan-bootblock+filevirus

LEVIATHAN-Bootblock+Filevirus

Es handelt sich um einen kombinierten Bootblock/Filevirus, das heißt der Virus verbreitet sich sowohl als Bootblockvirus als auch über ein startbares Virusfile und sowohl das Booten des Bootblocks oder das Starten des Virusfiles installiert den identischen Virus im Speicher ab \$7f000, der wie gesagt Bootblöcke als auch startbare Files schreiben kann. Der Virus macht sich über den COOL-Vektor resetfest. Während des nächsten Resets wird dann der DoIO()- und OldOpenLib()-Vektor verbogen. Die Virus-DoIO()-Routine dient zum Schreiben des Virus als Bootblock und die Virus-OldOpenLib()-Routine wird zum Infizieren des Bootmediums sys: verwandt. Es wird zu Beginn von sys:s/startup-sequence s/\$C0,\$0A als Aufruf des Virusfiles eingefügt und dann noch das 1056 Byte lange Virusfile als sys:s/\$C0 geschrieben. Im Gegensatz zu \$A0 (Shift-Space) oder \$20 (Space) ist \$C0 sichtbar, und zwar als Å

Der Grund warum der LEVIATHAN-Virus nicht ab Kickstart 2.0 arbeitet ist folgender, der OldOpenLib()-Vektor wird in der COOL-Routine, also bereits vor dem eigentlichen Booten verbogen, das heißt die Virus-OldOpenLib()-Routine wird bereits durchlaufen, bevor die dos.library eingerichtet ist, der Virus geht aber fest von einer bereits installierten dos.library aus. Ein simpler Test, ob OpenLibrary('dos.library') Erfolg hatte, würde genügen, und der Virus wäre auch unter Kick2.0 lauffähig. Unter Kickstart1,3 läuft der Virus nur deswegen, weil zufälligerweise der OldOpenLib()-Vektor vom Betriebssystem nicht so früh benutzt wird.

Zu Beginn des Virus kann man lesen:

```
-- LEVIATHAN --
```

und am Ende des Virus ist folgender kodierter Text versteckt:

```
YOU ARE THE OWNER OF A NEW GENERATION OF VIRUS!
IT FUCKS YOUR STARTUP-SEQUENCE! HAVE FUN....
```

1.316 liberator1.21-memcheck

Liberator1.21-MemCheck

Wenn man das 10936 Byte lange normalerweise MemCheck genannte File startet wird folgender Text ausgegeben:

```
<<<<<<<< MemCheck v8.1 - August 1991 >>>>>>>>
THIS PROGRAM IN THE STARTUP-SEQUENCE WILL KILL ALL VIRUSES
      <<CODED BY MARC OF SLIPSTREAM >>  ^^^^^
MEMORY CLEAR ----- NO VIRUS ----- MEMORY CLEAR.
DISK-VALIDATORS CLEAN ----- NO VIRUS ----- DISK-VALIDATORS CLEAN
```

Wenn man das Programm mit dem Parameter s startet, dann unterbleibt die Textausgabe. Der Virus schreibt 'MemCheck s' an den Anfang von z.B. DF0:s/startup-sequence, damit er nach jedem Booten automatisch aufgerufen wird, denn der Virus macht sich nicht resetfest. Auch kann sich der Virus nicht verbreiten, also kein neues MemCheck-File schreiben.

In einer Datei namens 'Sys:.FastDir,\$20,\$20' wird ein Zähler angelegt, um bei Erreichen des Wertes 15 folgenden Text auszugeben:

```
Hello from the Liberator virus v1.21
Lets play trash the harddisk
I'm outta here, kiss mine you lamer!
```

In dem Virusfile kann noch folgenden Text lesen:

```
Congratulations your hard disk has been liberated of virus protection!!
Hello from the Liberator virus v1.21. The anti-antivirus is born!
```

Angeblich soll es sich also um ein Anti-Antivirusprogramm halten, welches Antivirusprogramme entfernt. Darauf deutet auch der folgende Text, welcher ebenfalls im dem Virusfile steht:

```
ZeroVirus VIRUSEXPERT VirusKiller(PvL) ZeroVirusIII Virus_Checker
Master_Virus_Killer_v2.1 BLVC Berserker BerserkerV5.0
```

1.317 liberator3.0-cv

Liberator3.0-cv

Wenn man das 10712 Byte lange normalerweise cv genannte File startet wird folgender Text ausgegeben:

```
Check Vectors rev 5.1 All Rights Reserved more TUPperware ©by Mike Hansell
Reset vectors ok, Nothing resident, Trackdisk.device not intercepted,
DoIO ok, VBlank ok, dos.library not intercepted.
System appears to be free of viruses and trojans!
```

Im Gegensatz zum Liberator1.21-MemCheck-Virus, der 'MemCheck s' immer starr an den Anfang der startup-sequence schrieb, geht der Liberator3.0-cv-Virus intelligenter zu Werke und fügt den Virusprogrammaufruf 'cv >NIL:' meist als unauffällige vorletzte Zeile in die startup-sequence ein.

Wie auch schon der Liberator1.21-MemCheck kann sich auch der Liberator3.0-cv nicht verbreiten, also kein neues cv-File schreiben.

Es wird ebenfalls ein Zähler in einer Datei namens 'Sys:.fastdir,\$20,\$20' angelegt, und zwar mit dem Startwert \$310a, also eine lesbare 1. Bei jedem Virusaufruf wird der Zähler asciimäßig um eins erhöht, um beim Erreichen von 100 (= \$3130300a) folgenden Text auszugeben:

```

    Congratulations your hard disk has been liberated of virus protection!!
      Hello from the Liberator virus v3.0 - Digital Deviant
        The anti-antivirus is here again!
          Lets play trash the hard disk
            and ram the disk heads
          Only hardcore belgian rave can
            truely liberate the mind!
          The liberator 15/01/92

```

Wie bei Liberator1.21-MemCheck kann man die Namen von Antivirusprogrammen gegen Ende des Virusfiles erkennen.

1.318 liberator5.01-pv

Liberator5.01-pv

Wenn man das 16924 Byte lange normalerweise pv genannte File startet wird folgender Text ausgegeben:

```

PV(Protect Vectors) v1.02 by Peter Stuer
July 22, 1992 FREeware
Reset vectors ok, Nothing resident, Trackdisk.device not intercepted,
DoIO ok, VBlank ok, low interrupts ok, dos.library not intercepted.
monitoring vectors...
Fully Kickstartv2.xx compatible, stops all viruses,checks disk-validators,
Use run to push this program into the background.

```

Es werden dann aus dem aktuellen c-Verzeichnis die Dateien c/pv c/run c/br in die c-Verzeichnisse der vorhandenen Devices wie z.B. DF0: usw. kopiert, wobei der zweite Buchstabe des Virusfiles zufallsgesteuert ausgewählt wird, d.h. es wird also z.B. der Name pa oder py usw. für die Viruskopie benutzt. Dementsprechend wird auch meist als vorletzte Zeile in die startup-sequence z.B. folgendes eingefügt: br c:pa wobei es sich bei br normalerweise um das Programm runback handelt, denn im Gegensatz zu den früheren Liberatorviren bleibt der Liberator5.01-pv aktiv (alle 4 Sekunden Lock-Zugriffe), kehrt also nicht in die Shell zurück.

Es werden die Dateien 's/.info,\$20' und '.fastdir,\$a0,\$20' als Zähler benutzt um wohl irgendwann folgenden Text auszugeben:

```

    Congratulations this disk has been liberated of virus protection!!
      Hello from the Liberator virus v5.01 - Random Disaster
        The anti-antivirus is here again!
          Lets play trash the hard disk
            and ram the disk heads
          The piracy curse
          Liberator V - The future is near.
          Look out for Liberator VI - The final nightmare ...

```

```

        coming soon from a lame swapper near you!
    Respect to the virus masters Lamer Exterminator,crime & Contrast
        And remember - be excellent to each other!
        The liberator 27/07/92

```

Das Gefährlichste am Liberator5.01-pv ist aber, daß er auch noch die s/shell-startup-Datei der Datenträger verändert, indem er folgende Zeilen anhängt:

```

;liberatorV - controlling me!
alias copy delete
alias delete 'echo *'No file to delete, can't find*'

```

Eine eventuelle s/shell-startup-Datei wird vom Betriebssystem automatisch beim Öffnen einer neuen Shell abgearbeitet. Durch diese vom Virus angefügten Zeilen wird nun der copy-Befehl in den delete-Befehl umdefiniert.

Wenn Sie nun also in der Shell mittels des copy-Befehl eine Datei kopieren wollen, dann wird statt dessen diese Datei gelöscht, denn für copy wird die Zuordnung delete gefunden und Ende, daß für delete noch die Zuordnung echo definiert wurde, wird dann nicht mehr wirksam.

Wäre gleich alias copy 'echo *'No file to delete, can't find*' eingetragen worden, dann würde anstatt copy lediglich eine ungefährliche echo-Meldung erfolgen, zwar auch sehr irritierend und ärgerlich, aber doch nicht mit Datenverlust verbunden. Anstatt des delete-Befehl wird jedoch der echo-Befehl ausgeführt.

Überprüfen Sie also regelmäßig Ihre s/startup-sequence, s/user-startup und s/shell-startup auf verdächtige Eintragungen.

1.319 lupo

LUPO

Es handelt sich lediglich um ein verändertes NANO1 -File.
Der Alerttext wurde verändert.

```

...a new virus runner from L U P O !

```

```

Cave Killer:
  BootX killed,VirusX killed,Virus Detector kill
  Virus Terminator lives,Virus Control lives enough ,
  these two must die,then paradise of viruses begin ! ...

```

Und als Virusfile wird anstatt a0a0a0a0a0a0 200000000000 verwandt und daher auch anstatt a0a0a0a0a0a0 20 0a 200000000000 20 0A an den Anfang der startup-sequence geschrieben.

Allerdings bringt diese Aktion nicht den gewünschten Erfolg, man kann zwar erfolgreich ein Virusfile mit dem Namen \$20 erstellen, es ist aber nicht möglich ein nur aus \$20 bestehendes Virusfile durch die startup-sequence aufrufen zu lassen, denn Spaces werden und müssen bei der Abarbeitung der startup-sequence überlesen werden.

1.320 nano1

NANO1

Es handelt sich um ein 1484 Byte langes File, welches auf Compuphagozyte6 basiert. Es bestehen folgende Unterschiede:

Als Virusfile wird anstatt a0a0a0a0 a0a0a0a0a0a0 verwandt und daher auch anstatt a0a0a0a0 20 0a a0a0a0a0a0a0 20 0a an den Anfang der startup-sequence geschrieben.

Nach 5 Resets wird mittels Copperprogrammierung die Deutschlandfahne angezeigt. Nach Drücken der linken Maustaste stürzt der Amiga ab.

Nach 5 Disksetteninfektionen (=5 OpenOldLibrary-Aufrufen) wird folgender Alert ausgegeben.

...another masterpiece by N A N O !!!

GREETINGS TO:

Byte Bandit, Byte Warrior, DEF JAM, DiskDoktors
FANTASY, Foundation For The Extermination Of Lamers,
I.R.Q. Team, Obelisk Softworks Crew, S.C.A., UNIT A ...

1.321 nano2

NANO2

Während NANO1 ein mit Alert und Coppergrafik aufgepeppten CompuPhagozyte6 darstellte und anstatt a0a0a0a0 a0a0a0a0a0a0 verwandte, so ist der 1472 Byte lange NANO2 nahezu identisch mit CompuPhagozyte8 .
Zu Beginn des NANO2-Files ist ein anderer Text zu lesen

>Virus in 9.91 by Nano

und programmtechnisch wurde der CompuPhagozyte8-Code durch die Verwendung von Unterrouinen anstatt mehrfach vorhandener Routinen etwas optimiert.

1.322 nast

NaST

ein BGS9 -Abkömmling. Es wird allerdings auch FindTask(), OldOpenLibrary(), OpenLibrary() und Interrupt3 verbogen. Das Originalfile wird nach c/A020A020A0202020A0202020A0 verschoben.

1.323 novi

NoVi

Terrorists -Abkömmling, statt TTV1 NoVi. Original-File wird nach c/.fastdir\$a0 verschoben.

1.324 purge

Purge

Das 9812 Byte lange, imploder gepackte, entpackt 14864 Byte lange Purge-Virus-Installer-Programm versucht nach dem Start das 5300 Byte lange, imploder gepackte, entpackt 14776 Byte lange Purge-Virus-Programm auf DH(0,1,2,3): HD(0,1,2,3): DF(0,1,2,3): A: B: abzuspeichern, und zwar unter dem Namen Purge im C und WBstartup-Verzeichnis. Anschließend wird durch Eintragen von Run >NIL: Purge in S/user-startup und S/startup-sequence der automatische Aufruf des Purge-Programmes beim Booten sichergestellt. Damit dieser Eintrag weniger auffällt, wird er möglichst erst nach zwei Originalzeilen eingefügt. Danach wartet das Purge-Virus-Installer-Programm 20 Minuten mittels DosDelay um dann alle .info-Dateien mit einer neuen 2164 Byte langen .info-Datei zu überschreiben, in welcher

```
FUCKING
SOFTWARE
PIRAT
```

zu lesen ist, es wird allersdings immer starr die gleiche .info-Datei geschrieben, so daß alle icons gleich aussehen und übereinanderliegen und auch vom gleichen Typ, nämlich Projekt sind, so daß eventuelle Programme nicht mehr über die Workbench gestartet werden können, die eigentlichen Programme sind aber noch unversehrt vorhanden, nur die dazugehörigen icons wurden überschrieben. Nach dieser .info-Überschreib-Aktion wird 1 Minute gewartet und dann ein CON:-Fenster mit folgendem Text geöffnet:

```
Friend of Terminator is there !!!
```

```
ANTIPIRAT
Power of Destroying !!!
```

```
My ultimate answer against all the fucking softwarepirats !
```

```
Hi Anatol,Cycledom,Primitive,Björn,Dead Homer,Brian,
Gigant,Termination 8,Hardball & Slimeck.
```

```
Worked on all available devices...!
```

```
Ready....
```

Wenn man das Purge-Virus-Programm startet, dann überschreibt es sich mit einem 5212 Byte langen ungepackten File, und wartet dann wie bei dem Purge-Virus-Installer-Programm beschrieben 20 Minuten, um dann die .info-Überschreib-Aktion durchzuführen und das Fenster auszugeben.

1.325 revengeofthelamerexterm.

RevengeOfTheLamerExterminator

Neben den bekannten Lamer-Bootblockviren gibt es neuerdings auch einen Lamer-Filevirus. Sein Name ist 'RevengeOfTheLamerExterminator', gemäß dem

3-seitigen Alert, welcher nach 8 Minuten erscheint. Hierbei werden auch alle eingelegten und nicht schreibgeschützten Disketten formatiert. Der 'RevengeOfTheLamerExterminator' ist kein Bootblockvirus, denn er verbreitet sich nicht über den Bootblock, sondern er verbreitet sich mit Hilfe der Startup-Sequence. Hierzu schreibt er sich als normales Programm unter dem Namen \$a0a0a0a0a0 auf Diskette und schreibt zusätzlich in die Startup-Sequence \$a0a0a0a0a0. Der Virus wird nun also beim Abarbeiten der Startup-Sequence aktiviert. \$a0a0a0a0a0 erscheint lediglich als Leerzeile und wird dadurch leicht übersehen. Der 'RevengeOfTheLamerExterminator' tritt in mindestens zwei, allerdings nur unwesentlich verschiedenen, Versionen auf. Beide Versionen sind nicht fehlerfrei programmiert, so daß manchmal beim Infizieren der Diskette das Inhaltsverzeichnis zerstört wird. Hierbei wird oftmals ein Endlosverzeichnis erstellt. Als Folge davon kann es nun vorkommen, daß sich der Rechner in dieser Endlosschleife verfängt, bevor er auf das Virusfile stößt. Man kann in diesem Fall den Virus also nicht erkennen. Auch wird manchmal ein Virusfile erstellt, auf welches unter Kickstart 1.2/1.3 nicht mit der üblichen ExNext-DOS-Funktion zugegriffen werden kann. Das Programm wird also nicht erkannt. Dennoch ist es aber vorhanden und kann auch gestartet werden. Da aber VIRUS CONTROL bei jeder eingelegten Diskette die Startup-Sequence nach unsichtbaren Sonderzeichen überprüft, werden Sie auch in diesem Fall auf die Virus-Infektion hingewiesen.

1.326 scsi

scsi

Beim Start dieses 1560 Byte langen Programms wird geprüft, ob eine Datei namens s:\$ç\$; (\$733a24e724a1) vorhanden ist, wenn ja wird das Virusprogramm sofort beendet, es handelt sich hierbei also um eine Hintertür für den Virusprogrammierer selber. Wenn diese s:\$ç\$;-Datei nicht vorhanden ist, was wohl immer der Fall ist, dann wird geprüft, ob die aktuelle Systemzeit bereits den 27.02.93 erreicht hat. Wenn nein, dann wird versucht eine Datei mit dem nicht lesbaren Namen c:\$20 zu laden und auszuführen. Danach erfolgt das Virusprogrammende. Sollte allerdings bereits der 27.02.93 erreicht oder verstrichen sein, dann wird der Virus sehr bösartig, man kann also von einer Zeitbombe sprechen, der Virus überschreibt dann den Rootblock aller Speicherpartitionen, deren Devicename mit scsi beginnt, mit Nullen, wodurch eine 'Not a DOS disk' resultiert. Mit z.B. disksalv können Sie den Großteil der Daten wieder retten.

Der scsi-Virus ist lediglich das Produkt eines bisher unbekannten Installierungsprogramms, denn in dem scsi-Virus selber sind keine Routinen vorhanden, die den Virus verbreiten, die also z.B. den Original c:version-Befehl in c:\$20 umbenennen und dann den Virus unter dem früheren Original-Namen c:version abspeichern.

Womöglich gibt es auch gar kein Installierungsprogramm und diese Zeitbombe wurde von Hand auf gewisse Leute angesetzt, indem man ihnen also eine bereits infizierte Diskette (oder dms, zoom-file) zukommen lies.

Der scsi-Virus funktioniert also ähnlich wie der BGS9-Virus, das heißt ein Original-Programm wird in ein unsichtbares File umbenannt und der Virus wird unter dem Original-Namen abgespeichert,

wobei der Virus das ehemalige Original-Programm zur Ausführung bringt.

Wohingegen der BGS9-Virus allerdings diese Infektionsvorgänge selbstständig ausführt, also ein vollständiger Virus ist, so fehlen in dem scsi-Virusprogramm solche Verbreitungsroutinen, dafür ist nun aber die schädliche Wirkung (Rootblöcke überschreiben) hinzugekommen.

Der scsi-Virus ist recht sauber programmiert und läuft daher mit allen Kickstartversionen.

1.327 sepultura

SEPULTURA

Nach dem Start dieses 1876 Byte langen Virusfiles macht sich der Virus über die Kick-Vektoren resistent, was aber nur bei Abwesenheit von Fast-RAM gelingt. Eventuelle weitere Kick-resistente Programme gehen immer verloren. Weiterhin werden der Open(), Lock(), Loadseg(), Rename() und Delete()-Vektor der dos.library verbogen, hierbei wird aber starr von dem besonderen Kickstart 1.3 - dos.library-Aufbau ausgegangen, so daß der Virus unter Kickstart 2.0 abstürzt. Während eines Resets wird zufallsgesteuert manchmal folgender Alert ausgegeben:

```
                Hi Guyz !!  
        SEPULTURA strikes back with their new  
                VIRUS!!!  
        Look out for the SEPULTURA Boot-Virus !  
Greets to: MAX C. - JAIRO T. - IGOR C. - ANDREAS K. - CHUCK  
        (made in Belo Horizonte/Brasil 1988)
```

Weiterhin ist in dem Virus noch folgender verschlüsselter Text enthalten:

```
        Thanx to Max of StarLight for spreading the Virus to Germany
```

Während des Resets wird immer kurzzeitig der FindResident()-Vektor verbogen, um dann für circa 6 Sekunden lang den \$6c-Interrupt-Vektor 3 zu verbiegen. Nach diesen 6 Sekunden ist in der Regel der Bootvorgang so weit fortgeschritten, daß nun auch die dos.library eingerichtet ist, so daß der Virus nach diesen 6 Sekunden wie oben bereits beschrieben Open(), Lock(), Loadseg(), Rename() und Delete()-Vektor der dos.library verbiegen kann. Wenn nun einer der erwähnten dos.library-Vektoren aufgerufen wird, was sehr oft der Fall ist, dann prüft der Virus ob der betroffene Datenträger, (unterstützt wird nur df0,df1,df2) schon infiziert ist, wenn nein, dann wird \$A0A0A0A0 0A an den Anfang der startup-sequence eingefügt und eine maximal 1000 Byte lange df(0,1,2):s/startup-sequence neu geschrieben. Danach wird noch das Virusfile unter dem unsichtbaren Namen df(0,1,2):A0A0A0A0 auf die betreffende Diskette geschrieben.

1.328 sepultura (v2.26)

Sepultura (V2.26)

Der Sepultura (V2.26) (1980 Bytes) baut auf dem SEPULTURA -Filevirus auf. Er macht sich jedoch nicht mehr über die Kick-Vektoren resetfest und läuft nun auch unter Kickstart 2.0. Im Gegensatz zum SEPULTURA-Virus liegt der Sepultura (V2.26)-Virus allerdings komplett kodiert vor. Beim Start des Virus wird zuerst der Viruscode dekodiert, was bei höheren Prozessoren in einem Absturz endet, weil aufgrund des größeren Prefetch noch verschlüsselte Daten im Code-Cache stehen. Daß diese Daten mittlerweile im Speicher und Data-Cache dekodiert vorliegen erkennt der Code-Cache nicht. Selbstmodifizierender Code führt daher auf höheren Prozessoren meist zum Absturz.

1.329 telecom

TeleCom

Nach dem Start des 756 Byte langen Programms wird der Viruscode nach \$c71000, also in das sogenannte Ranger-RAM kopiert, hier ist allerdings nur bei eher wenigen Amigas Speicher vorhanden, so daß der Virus auf vielen Amigas nicht funktionieren wird. Außerdem arbeitet der Virus aufgrund direkter ROM-Einsprünge nur mit Kickstart 1.3 und benutzt auch noch fest Speicher ab \$70000. Weiterhin wird der COLD-Vektor gelöscht und der Virus macht sich über den COOL-Vektor resetfest. Nach dem nächsten Reset wird dann kurzzeitig der Findresident()-Vektor verbogen, um dann, nachdem die intuition.library eingerichtet wurde, in dieser den OpenWindow()-Vektor zu verbiegen. Der DoIO()-Vektor wird verbogen, um zu erkennen, wann der Bootzugriff erfolgt ist, um dann der Virus-OpenWindow-Routine anzuzeigen, daß sie nun vor dem Öffnen des CLI-Windows die Virusinfektion vornehmen kann, denn nun ist die dos.library benutzbar. Es wird a0 0a an den Anfang der startup-sequence eingefügt, jedoch wird nur maximal 1024 Byte startup-sequence neu geschrieben, so daß eine längere startup-sequence gekappt wird. Danach wird noch das Virusfile unter dem Namen a0 auf die Bootdiskette geschrieben. Am Ende des Virusfiles ist dekodiert der folgende Text versteckt: s/startup-sequence TeleCom

1.330 terrorists

Terrorists

Der Terrorists-Virus ähnelt sehr dem BGS9 -Virus. Lediglich die folgenden Unterscheide sind vorhanden:

Terrorists

BGS9

nach 12 Resets folgende Meldung:

the names have been changed
to protect the innocent...
the terrorists have you under control

nach 4 Resets folgende Meldung:

a computer virus is a disease
terrorism is a transgression
software piracy is a crime

everything is destroyed	this is the cure
your system is infected	BSG 9 Bundesgrenzschutz Sektion9
there is no hope for better times	Sonderkommando 'EDV'
the first terrorists Virus !!!	
 das Original-File wird versteckt	 das Original-File wird versteckt
als sys:a0202020a02020a020a0a0	als devs:a0a0a0202020a0202020a0
	od. devs:a0e0a0202020a0202020a0
 es wird nichts kodiert	 liegt kodiert vor
 Fileviruslänge 1612 Bytes	 Fileviruslänge 2608 Bytes

1.331 challenger-fish622

Challenger-Fish622

Auf der Fish-Diskette 622 fungiert die deutsche Version des Frage-Spieles Challenger als Trojanisches Pferd. Beim Start des Programms wird in sys:devs/keymaps/a ein Original-Workbench-1.3-setclock-Befehl erstellt. Der Code des Original-Workbench-1.3-setclock-Befehls ist in dem Challenger Programm selber enthalten. Deshalb versucht VIRUS CONTROL auch nicht devs/keymaps/a nach c:setclock umzubenennen, weil nicht gewährleistet ist, ob der Virus-interne-setclock-Befehl immer der richtige ist, denn obwohl es sich hierbei um einen Original-Commodore-setclock-Befehl handelt, so gibt es doch verschiedene Commodore-setclock-Befehle. Kopieren Sie sich also Ihren normalen setclock-Befehl bei Bedarf nach c: zurück. Danach wird ein neuer Virus-setclock-Befehl in sys:c/setclock erstellt, welcher aus Tarngründen die gleiche Länge wie der Original-Workbench-1.3 setclock-Befehl aufweist (4884 Bytes). Weiterhin wird eine Datei namens sys:devs/keymaps/rca angelegt, in welchem unter anderem die Texte stehen, welche am 24. Juli angezeigt werden. Die Dateien sys:c/setclock und sys:devs/keymaps/rca sind mit dem Imploder gepackt und benötigen zum Entpacken die explode.library. Deshalb bleibt die Abarbeitung der Startup-Sequence auch hängen, wenn keine explode.library vorhanden ist. Das Challenger-Programm ist also ein Trojanisches Pferd, welches zum Installieren eines Virus dient. Anstelle des Original-setclock-Befehls tritt das Virus-setclock-File. Beim Abarbeiten der Startup-Sequence wird normalerweise immer der setclock-Befehl aufgerufen, um die Uhrzeit aus der Echtzeit-Uhr auszulesen. Unter Kickstart 2.0 ist der setclock-Befehl überflüssig, da hier die Uhrzeit automatisch vom Betriebssystem (battclock.resource) ausgelesen wird. Der Virus macht sich nicht resetfest und verändert auch keine Vektoren. Die loadseg()-Veränderung ist durch die explode.library bedingt, welche beim Entpacken des Virus-setclock-Befehls aktiviert wird. Der Virus-setclock-Befehl ruft als erstes den Original setclock-Befehl in sys:devs/keymaps/a auf, um die Uhrzeit zu setzen. Sollte nun der 24. Juli erkannt werden, dann öffnet der Virus einen Screen mit folgendem Text:

```
Guten Tag, hier ist der Guru Ihres Amiga-Computers
Laut Arbeitsvertrag habe ich das Recht auf einen Medita-
tionstag pro Jahr. In meinem Fall ist das der 24 Juli
jeden Jahres. Da wir heute dieses Datum schreiben, stehe
ich Ihnen erst morgen wieder zur Verfügung. Bitte haben
Sie Verständnis dafür, denn auch wir Gurus müssen einmal
```

ausruhen.

Der Virus wird nur am 24. Juli aktiv. Man kann aber eigentlich nicht von einem Virus sprechen, da er sich nicht selber verbreiten kann. Leidiglich das Challenger-Programm kann den Virus installieren.

1.332 clonk-installer

CLONK-Installer

Es wird ein enforcer-Archiv verbreitet, in welchem ein 12952 langes enforcer-Programm enthalten ist, wobei vor das eigentliche enforcer-Programm ein Hunk gehängt wurde, welcher den CLONK-Antivirus im Speicher installieren soll. Da jedoch mit nicht initialisierten absoluten Adressen gearbeitet wird, gelingt die Installation normalerweise nicht.

1.333 colorsviruscarrierturkbb

ColorsVirusCarrierTurkBB

Es handelt sich um ein anscheinend harmloses kleines Programm (2196), welches beim Starten nach einem kurzen Farb-Flackern ein CON-Window mit dem Titel Color Demo! öffnet. In dem Window kann man folgenden Text lesen:

```
Hope you enjoy this proggy!
It was put together in ten minutes ....
Press Left Mouse Button for the demo ...
** Press Right Mouse Button to end **
```

Nach Drücken der linken Maustaste erscheint eine einfache Copper-Grafik, welche nach Drücken der rechten Maustaste wieder verschwindet.

Leider ist dies nicht alles was das Programm macht, denn es verbiegt als allererstes den COOL-Vektor und den DoIO()-Vektor. Der DoIO()-Vektor wird immer auf \$70000 verbogen. Weiterhin wird der TURK-Bootblockvirus nach \$7f000 kopiert. Jede eingelegte Diskette wird nun mit dem TURK-Bootblockvirus überschrieben.

Das ColorsVirusCarrierTurkBB-File verbreitet sich nicht von alleine weiter, es dient also 'nur' der Verbreitung des TURK-Bootblockvirus. Es handelt sich also um ein Trojanisches Pferd.

1.334 crime!++-trojan.pferd

DriveInfo V0.91

Der Crime!++ -Linkvirus wurde mittels eines Trojanischen Pferdes namens DriveInfo (1704 Bytes) in Umlauf gebracht. Wenn man dieses

Programm startet erscheint z.B. folgende unscheinbare Ausgabe:

```
DriveInfo V0.91 by ESP
```

```
Drive      : DF0
Volume     : aaa
Free Bytes: 18944
```

Weiterhin wird aber auch unsichtbar ein Interrupthandler mit dem Namen 'Install yeah!' installiert, mit dessen Hilfe nach circa einer Minute der eigentliche Crime!++ -Linkvirus installiert wird. Dieses Verhalten kann für viele Leute heimtückisch sein, denn wenn man es sich angewöhnt hat, während des Startens neuer Programme gleichzeitig mittels eines Antivirusprogrammes auf Systemveränderungen zu testen, dann schlägt dieses Vorgehen hier fehl. Einmal mehr zeigt sich hier die Überlegenheit von VIRUS CONTROL, dessen permanenter und resetfester System-Check andauernde Sicherheit gibt und Sie deshalb auch auf die späteren Systemveränderungen hinweist.

1.335 excrement-installer

EXCREMENT-Installer

Es handelt sich bei diesem 1180 Byte langen File um ein Trojanisches Pferd, das lediglich den EXCREMENT -Bootblockvirus in den Speicher kopiert und den COOL-Vektor auf diesen Viruscode verbiegt. Nach dem nächsten Reset wird dann dieser EXCREMENT-Bootblockvirus aktiviert und kann die eingelegte Bootdiskette infizieren.

1.336 generalhunter v3.2

GENERALHUNTER V3.2

Das 26112 Byte lange File enthält verdächtigen Text, der auf eine womöglich versteckte Virusfunktion hindeutet

```
SIGNUM Ich bin der Bobble-Signum-Virus (3.Generation)
Du brauchst mich nicht mehr zu kopieren, denn das habe
ich jetzt auch schon selber drauf!
```

1.337 intro-maker v1.00 by tcr

Intro-Maker V1.00 by TCR

Tarnt sich als ein Hilfsprogramm zum Erstellen von Boot-Intros. Hierbei wird allerdings auch der DISASTER-MASTER V2 -Virus installiert.

1.338 jeff3.10-trojan.pferd

JEFF-Viruskiller

Das 9064 Byte lange Programm täuscht vor, den BUTONIC-JEFF-VirusV1.31(4.55)+V3.00(3.10) zu suchen und zu beseitigen, aber gerade das Gegenteil ist der Fall, der Virus wird installiert.

1.339 lads-mvk

LADS-MVK

Dieses Programm wird unter dem Namen MVK.exe als neuer Mini-Viruskiller angepriesen. Es handelt sich allerdings bei diesem 1052 Byte langem Programm lediglich um den mißglückten Versuch den LADS-Bootblockvirus in ein startbares Programm umzuwandeln. Das Programm startet zwar und installiert auch den L.A.D.S -Bootblockvirus im Speicher, allerdings kann dieser Virus keine erfolgreichen Disketteninfektionen vornehmen, da bei der Erstellung des startbaren Programms die ersten 3 Bootblock-Langworte lediglich durch einen move.l \$4.w,a6-Befehl ersetzt wurden. Das Fehlen des zweiten und dritten Langwortes hätte noch keine Konsequenzen, da die Boot-Checksumme (zweites LW) von dem LADS-Bootblockvirus jeweils neu berechnet wird und der Wert des dritten LW ist eh immer egal, aber das Fehlen des ersten Langwortes, nämlich der DOS-Kennung hat Konsequenzen, denn dadurch wird eine unbenutzbare Not-A-Dos-Disk geschrieben. Dieser fehlerhafte LADS-Bootblockvirus kann sich also nicht automatisch über das Neuinfizieren von Disketten erfolgreich verbreiten. Weiterhin wurden unwichtige Textänderungen, wie z.B. L.A.D.S Virus Hunter in A.I.D.S VIRUS-HUNTER, vorgenommen.

1.340 lamer-bomb(gotcha lamer)

LAMER-bomb (Gotcha LAMER)

Wenn man das 776 Byte lange Programm minidemo.exe startet, dann versucht dieses Programm einen Virus-Hunk vor die Programme DH0:c/dir, DH0:c/run, DH0:c/cd und DH0:c/execute zu linken. Dadurch werden diese Programme um 372 Bytes länger. Dies ist alles, was minidemo.exe macht, es kann sich also z.B. nicht von alleine weiterverbreiten und ist auch nicht resetfest. Man kann minidemo.exe als ein Trojanisches Pferd betrachten.

Die infizierten Programme DH0:c/dir, DH0:c/run, DH0:c/cd oder DH0:c/execute verbiegen beim Aufruf den DoIO()-Vektor, um dann beim nächsten Disketten-Einlegen diese Diskette schnellzuformatieren. Hierbei gehen die meisten Disketten-Daten verloren. Danach wird folgender Alert ausgegeben:

HABAHE... Gotcha LAMER!!!

Nach Betätigen einer Maustaste wird dann ein Software-Reset ausgelöst. Die Programme sind nicht resetfest.

1.341 lamerbb-trojan.pferd

LAMER-Trojan-Horse (endcli, loadwb, virusx)

Es gibt ein verändertes loadwb-Programm (4172 Bytes) und virusx-Programm (13192 Bytes), welche sich beim Start wie loadwb bzw. virusx verhalten, gleichzeitig wird aber ein LAMER-Bootblockvirus installiert. Die Verbreitung des Virus erfolgt 100% als LAMER-Bootblockvirus über den Bootblock. siehe Lamer-Bootblockviren . Diese veränderten loadwb und virusx-Programme sind also Trojanische Pferde.

1.342 messangel

MessAngelKiller

Es handelt sich um 1808 Byte langes gepacktes bzw. 7100 Byte langes ungepacktes Programm, welches im CLI vorgibt ein MessAngelKiller zu sein, in Wirklichkeit wird jedoch an den Anfang von sys:s/startup-sequence "B", \$a0 geschrieben. Nach diesen 2 Bytes werden immer starr 4998 Bytes der Original-startup-sequence geschrieben, sollte sie länger gewesen sein, so ist der Rest verloren, sollte sie kürzer sein, wird der Rest mit Nullen aufgefüllt. Weiterhin wird das entsprechende 1504 Byte lange File namens sys:B erstellt. Anschließend wird auf das Drücken der rechten Maustaste gewartet und dann unsauber ins ROM gesprungen, was meist zu einem Reset führt.

MessAngel-B

Beim Abarbeiten der startup-sequence wird nun das Programm namens B ausgeführt, welches praktisch nur aus den
ELENI! -Bootblockvirusdaten besteht, es erfolgt ein Sprung an den Viruscodeanfang, wobei dann also der Virus installiert und wie beim ELENI!-Bootblockvirus üblich anschließend ein Aufruf der Kickstart2.0-ColdReboot()-Routine versucht wird.

1.343 modemcheck-fuck-virus

MODEMCHECK-FUCK-Virus

MODEMCHECKV1.1-Trojanisches Pferd

Der eigentliche Virus, genannt FUCK-Virus, ein 3604 Byte langer c:loadwb-Befehl, wird durch das Starten des 15516 Byte langen, CrunchMania gepackten Programmes installiert. Entpackt ist dieses 'Trojanische Pferd' 22252 Bytes lang.

Das Installierungs-Programm beendet zuerst ein eventuell laufendes snoopdos-Programm, damit das Anlegen des Virus als neuer c:loadwb-Befehl nicht bemerkt werden soll. Diejenigen Leute, die allerdings snoopdos einsetzen, werden wohl schon das plötzliche Verschwinden des

snoopdos-Fensters bemerken und als Alarmzeichen betrachten.
Zum Schluß wird dann noch zur Tarnung folgender Text ausgegeben:

```
MODEMCHECK V1.1 (07.05.93)
Copyright © 1993 by Mario Zeltik. All Rights Reserved
```

```
Checking CTS Line.....Ok!
Checking CD  Line.....Ok!
Checking DTR Line.....Ok!
Checking RI  Line.....Ok!
Checking TXD Line.....Ok!
Checking RXD Line.....Ok!
Checking RTS Line.....Ok!
```

Verdächtigerweise erscheint dieser Text jedoch auch dann,
wenn gar kein Modem angeschlossen ist.

FUCK-Virus (Festplattenvirus)

Der Virus funktioniert auf allen Amigas und allen Betriebssystemen.

Dadurch daß sich das Virusprogramm 'loadwb' nennt, wird es beim Booten
von einer normalen Workbench-Diskette durch die startup-sequence aufgerufen.

Die Original-Commodore-loadwb-Befehle sind immer kleiner 3000 Bytes,
ein größerer loadwb-Befehl sollte also immer ersetzt werden.

Das Virusprogramm besteht aus zwei Teilen, dem Original-loadwb-38.9-Teil,
der unter Kickstart 1.2/1.3 keine Workbench hochfährt und einem Virus-Teil,
welcher mittels CreateProc() als eigener Prozess namens Diskdriver.proc
abgekoppelt wird. Dieser Virusteil wartet dann 10 Minuten, bevor er seine
zerstörerische Arbeit aufnimmt. Zuerst wird dann geprüft ob eine Datei
namens S:HORSE vorhanden ist, wenn ja, dann unternimmt der Virus nichts,
dies ist also als Hintertür gedacht. Im Gegensatz zum Installierungsprogramm
schaltet das Virusprogramm ein eventuell aktives snoopdos-Programm
allerdings nicht ab, so daß man diesen Open oldmode-Zugriff mitverfolgen
kann. Den Text S:HORSE kann man in dem Virusfile jedoch nicht lesen,
da er kodiert vorliegt.

Anschließend hangelt sich der Virus über dosbase,root und info zu der
Device-Liste mit eventuellen startup- und environment-Einträgen.
Der Virus interessiert sich aber nur für solche Devices, die entweder mehr
als 2 Köpfe oder mehr als 89 Zylinder oder mehr als 22 Sektoren aufweisen,
das heißt Disketten-Laufwerke werden mit Absicht nicht erfaßt, man kann
also von einem Festplattenvirus sprechen. Da je nach Festplattentreiber die
Werte für Sektorenanzahl, Kopfanzahl, Zylinderanzahl usw. verschieden sind,
ergibt sich aus diesen verschiedenen Werten auch ein unterschiedliches
Datenzerstörungsverhalten des Virus. Mittels CMD_WRITE werden u.U. alle
tracks mit FUCKFUCKFUCKFUCKFUCK usw. vollgeschrieben, oder aber auch z.B.
jeder zweite Track übersprungen. Letztendlich wird aber die ganze Partition
durchgearbeitet. Da sich aber Dateien meist über mehrere tracks erstrecken,
sind somit dennoch fast alle Programme verloren. Da der unterste track der
Partition, welcher die DOS-Kennung in den beiden ersten reservierten
Blöcken enthält immer überschrieben wird, resultiert eine 'Not a DOS Disk'.

Der Virus stellt also eine neue Virus-Qualität dar, denn es handelt sich

um den ersten Virus, der sich gezielt Speichergeräte vornimmt und die Daten zum Großteil unwiderbringlich überschreibt.

Mittlerweile sind Abkoemmlinge aufgetaucht, welche anstatt "FUCK" z.B. "LAME", "DLT " oder " TAI" schreiben.

1.344 scan.x

scan.x

Es wird ein bossnukel.5-Archiv weitergegeben, welches Zusatzprogramme für den Betrieb von Mailboxen enthält. Neben einem harmlosen 35308 Byte langen Programm namens bossnuke.x ist auch ein 18560 Byte langes Programm namens ULOG.X enthalten. An den letzten Hunk von ULOG.X ist eine verschlüsselte Virusinstallierungsroutine angehängt, welche beim Starten von ULOG.X aufgerufen wird. Diese Virusinstallierungsroutine versucht ein 1060 Byte langes info-File namens L.info nach BBS:COMMANDS/BBSCMD und ein 712 Byte langes Programm namens scan.x nach doors: zu schreiben. Das Schreiben der Virenfiles gelingt nur wenn die entsprechenden Verzeichnisse bereits existieren.

Die L.info-Datei zeigt eine kleine Platine und beinhaltet folgende ToolTypes:

```
ACCESS=001
LOCATION=doors:scan.x
MULTINODE=NO
PRIORITY=0
STACK=4096
TYPE=XIM
```

Es wird also auf doors:scan.x verwiesen. Und dieses scan.x ist nun der eigentliche Virus, welcher auf allen Rechnern funktioniert. Es handelt sich um einen Virus, der ähnlich dem MODEMCHECK-FUCK-Virus gezielt Festplatten und ähnliche große Speichermedien mit Daten überschreibt und somit die Originaldaten zerstört. Disketten werden auch bei scan.x verschont. Leider subtrahiert der scan.x-Virus von dem untersten Zylinder nochmals zwei Zylinder, wodurch somit auch der sich normalerweise darunter befindliche zwei Zylinder große RigidDiskBlock zerstört wird. Der scan.x-Virus überschreibt normalerweise jeden vierten Zylinder mit DOS3DOS3DOS3DOS3DOS3DOS3DOS3DOS3DOS3 usw. wobei mit dem ersten Zylinder der Partition minus 2 begonnen wird, wodurch also in der Regel der RigidDiskBlock überschrieben wird und somit die Festplatte nicht mehr bootbar ist. Weiterhin sind auch mit Datenrettungsprogrammen nur noch sehr wenig Daten zu retten, da Dateien meist über mehrere Zylinder verteilt sind, und somit zumindest teilweise beschädigt sind, wodurch also z.B. startbare Programme nicht mehr geladen werden können. Der scan.x-Virus durchsucht über rootnode, dosinfo usw. die Device-Liste der dos.library nach Devices. Hierbei werden nur Devices mit lowcyl<>0 und blockpertrack >11 berücksichtigt, was z.B. für Festplatten meist typisch ist, da hier meist lowcyl=2 ist, da Zylinder 0,1 meist von dem RigidDiskBlock belegt wird. Der scan.x-Virus bearbeitet maximal 15 Devices. Danach wird ein Reset ausgelöst.

1.345 t.f.c.-loadwb

T.F.C.-loadwb

Es handelt sich um ein 2804 Byte langes File, welches sich als loadwb-Befehl tarnt. Es liegt ein Trojanisches Pferd vor, da nach Ausführen der loadwb Funktionen der T.F.C. Revenge Virus -Bootblockvirus installiert wird. Der Virus verbreitet sich dann nur als T.F.C.-Bootblockvirus.

1.346 the smily cancer ii

THE SMILY CANCER II

Dieses 4676 Byte lange Programm dient lediglich zum Installieren des THE SMILY CANCER I von CENTURIONS -Linkvirus.
Das THE SMILY CANCER II - Programm selber verbreitet sich nicht von alleine. Die Verbreitung erfolgt als THE SMILY CANCER I von CENTURIONS -Linkvirus. Das THE SMILY CANCER II - Programm versucht sich als loadwb-Befehl zu tarnen, man kann also von einem Trojanischen Pferd sprechen.

1.347 virus terminatorv6.0

VIRUS TERMINATORV6.0

Wenn man dieses 1880 Byte lange Programm startet wird folgender Text ausgegeben:

```
VIRUS TERMINATOR V6.0 by Rudolf Meiler

--> This Killer kills about 240 viruses <--

Memory Check : No Virus found !!!
File   Check : No Virus found !!!
Boot   Check : No Virus found !!!
```

Das Programm versucht sich also als ein tolles Antivirusprogramm zu tarnen, gleichzeitig wird allerdings ein Bootblockvirus installiert. Es handelt sich also um ein trojanisches Pferd. Das 1880 Byte lange Programm kann sich nicht selber verbreiten. Von Interesse ist also der installierte Bootblockvirus, welcher sich über den COOL-Vektor resetfest macht und den DoIO()-Vektor verbiegt, um zukünftig eingelegte Disketten zu infizieren. Nach acht Bootblock-Lese-Zugriffen offenbart sich der Virus mit einem Alert, wobei folgender Text angezeigt wird:

```
-+= CHEATER HIJACKER   GENERATION 0004 =+-
```

Da der Bootblock mit einer zufälligen Strahlenposition verschlüsselt wird, kann man keine verräterischen Texte erkennen.

Das Trojanische Pferd 'VIRUS TERMINATORV6.0' installiert also einen Bootblockvirus. siehe LameBlame-TaiPan (LameBlame,CHEATER-HIJACKER,POLISH) .

1.348 virus-install v2.0

VIRUS-INSTALL v2.0

Wenn man dieses Programm startet, dann wird eindeutig der Zweck des Programms erläutert, nämlich das Schreiben eines Bootblockvirus auf Diskette. Startet man das Programm mit dem Parameter 1, dann wird ein LameBlame-TaiPan (LameBlame, CHEATER-HIJACKER) - Bootblock geschrieben, startet man das Programm mit dem Parameter 2, dann wird ein Chaos-TaiPan(Chaos)-Bootblock geschrieben.

Siehe LameBlame-TaiPan(LameBlame,CHEATER-HIJACKER,POLISH)
und Chaos-TaiPan(Chaos) .

VIRUS TERMINATORV6.0 aktiviert also den CHEATER-HIJACKER-Bootblock im Speicher. VIRUS-INSTALL v2.0 schreibt den LameBlame-Bootblock auf Diskette. Zwischen den beiden Bootblöcken gibt es nur minimale Textunterschiede. VIRUS-INSTALL v2.0 schreibt noch auf Wunsch den Chaos-TaiPan(Chaos) auf Diskette.

1.349 little sven-trojan.pferd

XCOPYPro6.5

Diese inoffizielle 28336 Byte lange Version des XCOPY-Diskettenkopierprogramms installiert den Little Sven -Bootblockvirus im Speicher und schreibt auch den Little Sven auf den Diskettenbootblock.

1.350 ??? \$4eb9-virus ???

\$4EB9-Link

Die \$4EB9-Link-Methode wird insbesondere zum unauffälligen Verbreiten von neu geschriebenen meist nur Mailbox schädigenden Programmen verwandt. Zu Beginn eines solchen \$4EB9-Link-Files stehen zwei jsr(=\$4EB9)-Unterprogrammaufrufe und danach erfolgt das Programmende. Es kann nun der erste oder zweite Unterprogrammaufruf zum Starten des Virus benutzt werden, der andere Unterprogrammaufruf ruft dann aus Tarngründen das harmlose Originalprogramm auf. Meist liegt der Virusprogrammteil noch mit z.B. imploder gepackt vor, um somit eventuell verräterischen Text zu kodieren. Weiterhin prüfen diese Viren meist auf die Anwesenheit von snoopdos, sollte dieses Analyseprogramm aktiv sein, dann unternehmen die Viren nichts, denn ein aktives snoopdos würde die Aktivität der Viren, sprich Filezugriffe, anzeigen und den Virus somit verraten.

diskrepairV1.20 (49336 Bytes)

Der erste Unterprogrammaufruf startet DiskRepair V1.20, nach dessen Beendigung ruft der zweite Unterprogrammaufruf den Virus auf, welcher nach folgenden Verzeichnissen sucht:

utils
bbs
bbs/utils

PowerPacker3.2-Bomb

Es handelt sich um eine modifizierte PowerPacker3.0b-Version mit einer Filelänge von 71308 Bytes. Man kann also auch von einem Trojanischen Pferd sprechen. Diese 'neue' Version soll anscheinend jemandem gezielt Schaden zufügen, da beim Start des Programms in DH0: und DH0:BBS/ (und auch in DH1: und DH1:BBS/ ??) nach einem 120120, 106140 oder 104044 Byte lange File gesucht wird. Wird ein solches gefunden (angeblich soll insbesondere AmiExpress betroffen sein), dann wird dieses Programm beschädigt. Ich habe testhalber ein 120120 langes Programm erstellt. In diesem Programm wurden dann 3 Bytes überschrieben, und zwar Byte 22145 mit \$53, Byte 22163 mit \$02 und Byte 48286 mit \$2e.

Weiterhin wird ein eventuelles C:why gelöscht und mit einer leeren C:why-Datei ersetzt.

Es handelt sich eigentlich nicht um einen Virus, da keine Vermehrung erfolgt. Es werden auch keine Vektoren verbogen.

Bevor der PowerPacker3.2-Bomb obige File-Manipulationen vornimmt, prüft er zuerst auf die Anwesenheit von SnoopDos. Sollte SnoopDos aktiv sein, dann werden keine File-Manipulationen vorgenommen. Der PowerPacker3.2-Bomb will sich also nicht durch SnoopDos enttarnen lassen.

Dennoch macht der PowerPacker3.2-Bomb einen sehr schlecht programmierten Eindruck, denn wenn z.B. kein DH0: vorhanden ist, dann erscheint ein System-Requester, welcher den User anweist, DH0: einzulegen.

Dieses ist wohl sehr verdächtig. Auch die Abänderung des ausführbaren why-Befehls in eine nicht mehr ausführbare Datei erscheint mir eher unsinnig.

Also man sollte über den PowerPacker3.2-Bomb nicht viel Worte verlieren, da er sich eh nicht von alleine verbreiten kann, und wenn Sie dennoch auf Ihn stoßen sollten, dann wird im Normalfall auch nicht viel passieren. Selbst wenn man unwissend den PowerPacker3.2-Bomb benutzt, so wird man dies nicht lange tun, da diese Version recht bald an den verschiedensten Stellen abstürzt, und außerdem gibt es ja auch bereits z.B. die neue, bessere und offizielle PowerPacker4.0-Version.

Man sollte also dieses Schrott-Programm löschen und anstatt dessen eine aktuelle PowerPacker-Version benutzen.

Der PowerPacker3.2-Bomb wurde über Mailboxen in dem Archiv qtx_pow.lzh mit der Filelänge von 139670 Bytes verbreitet.

PP-Snap1.61, PP-Died2.8, PP-MegaMon

Ebenso wie PowerPacker3.2-Bomb versucht auch das 44260 Byte lange Programm PP-Snap1.61 gewisse Mailboxmanipulationen. Ähnlich verhält sich auch das 67028 Byte lange PP-Died2.8 und 26856 Byte lange PP-MegaMon-Programm.

Bei all diesen Programmen wurde eine Mailboxmanipulationsroutine mittels der \$4EB9-Link-Methode von Hand an bekannte Programme wie PowerPacker, Snap, Died oder MegaMon gelinkt.

SnoopEx-DLog-DevilDoor11 (23452 Bytes)

aufgrund fehlerhafter Hunkstruktur nur unter Kick1.2/1.3 ladbar, manipuliert BBS:-Dateien, um höhere Zugriffsrechte zu erhalten.

WhiteBoxV8.0 (34896 Bytes)

 Der erste Unterprogrammaufruf startet den Virus und danach ruft der zweite Unterprogrammaufruf das WhiteBoxV8.0-Programm auf. Der Virus sucht wiederum unter anderem nach bbs:

1.351 ??? \$4eb9-4ef9-virus ???

\$4EB9-4EF9-Link

Wie die \$4EB9-Link-Methode dient auch die \$4EB9-4EF9-Link-Methode zum Anhängen eines meist Mailbox-schädigenden Virusteiles an ein sauberes Programm. Programmtechnisch wird einmal mit einem jsr(\$4EB9)-Befehl ein Programmteil aufgerufen und dann mit einem jmp(\$4EF9)-Befehl ein weiterer Programmteil. Diese jsr/jmp-Kombination kann auch mehrfach auftreten. Vertreter dieser Linkmethode sind z.B. ACP, CLP, PHA und merry. Es handelt sich hierbei um meist nicht lauffähige Virusversuche, welche Dateien im S:-Verzeichnis zu überschreiben versuchen und auch Zugriffe auf BBS: vornehmen.

1.352 ??? hunklab-virus ???

Hunklab-Link

Es existieren mittlerweile sogenannte 'Hunklab'-Programme, mit denen es möglich ist, ein startbares File vor ein anderes startbares File zu linken. Angeblich soll auch eine Version des Diskettenkopierprogrammes XCopy eine solche 'Hunklab'-Funktion anbieten. Kein Programm sollte solch eine Funktion anbieten, da damit jeder Laie ein beliebiges Virusprogramm vor ein harmloses Programm linken könnte, und somit auf einfachste Weise 'Trojanische Pferde' erstellen könnte. Die Verbreitung von Viren würde durch solch eine Funktion sehr gefördert werden, also insbesondere bei kommerziellen Programmen wäre eine solche Funktion höchst unverständlich und sollte schnellstmöglich wieder entfernt werden. Es sind nun leider auch schon mit 'hunklab' erstellte Trojanische Pferde im Umlauf, womit insbesondere bereits bekannte Viren verbreitet werden:

2.MZaus	(1952)	installiert	GYROS -Bootblockvirus.
6Vekaus	(1916)	installiert	DUMDUM -Bootblockvirus.
ALFaus	(1928)	installiert	Dotty -Bootblockvirus.
DWEdit(1.62)	(43700)	installiert	Aibon2 -Virus.
ToolsDaemon2.2	(7128)	installiert	Aibon2-mount -Virus.
Jeffkiller2.67	(9960)	installiert	DIGITAL DREAM -Bootblockvirus.
MComm	(8340)	installiert	Starfire-EastStar -Bootblockvirus.
SeekSpeed	(33704)	installiert	butonic3.00 -Virus.
mount	(9396)	installiert	butonic1.31 -Virus.
snoopdos(1.9)	(15336)	installiert	butonic1.31 -Virus.
snoopdos(2.1)	(13776)	installiert	Saddam-4711-Disk-Validator .

Die hohen Versionsnummer 1.9 und 2.1 bei snoopdos sollen lediglich das Interesse an diesen Trojanischen Pferden erhöhen, in Wirklichkeit handelt es sich nur um geringfügig abgewandelte snoopdos1.7-Versionen.

Am Anfang von mit 'Hunklab' erstellten Files kann man des öfteren lesen:

United.ForceS

1.353 ??? xlink-virus ???

XLINK-Link

Wie die \$4EB9-Link -Methode wird auch die XLINK-Methode insbesondere zum unauffälligen Verbreiten von neu geschriebenen meist nur Mailbox schädigenden Programmen verwandt.

Dialer V2.8g

Dieses mit CrunchMania gepackte 11992 Byte lange Programm (entpackt 33908 Bytes) enthält einen mit der XLINK-Methode angelinkten Virusteil, der Daten aus bbs:user.data liest und Daten nach bbs:nodel/NOCALLERSAT300 schreibt.

1.354 bootblock-massacre

Bootblock-Massacre

Das 9592 Byte lange Programm ist mit einem sehr schmutzigen Packer komprimiert worden, welcher das Programm starr nach \$31000 entpackt, wodurch oftmals der Computer abstürzt, falls in diesem Speicherbereich bereits Programme laufen. Andernfalls wird im CLI die Bedienungsanleitung ausgegeben. Durch Drücken verschiedener Funktionstasten kann man u.a. verschiedene Bootblockviren (Stand 1988) auf Diskette schreiben lassen.

1.355 bootshop

BootShop

Nach dem Start der 108008 Byte langen mit DefJam3.2-gepackten Programmes wird einem die Installierung verschiedener Bootblöcke, darunter auch Virusbootblöcke, angeboten.

1.356 dag-virus-infector

DAG-Virus-Infector

Nachdem dem Start des 7360 Byte langen entpackten Programmes wird der folgende eindeutige Text im CLI-Fenster ausgegeben:

```
Virus Infector BY DAG's COPYKLAU2.0    V
      Mod by DAG  VIRUS Version! BEWARE !
      GIZMO! SAYS: LET THIS SHIT FLOW !!!!!
```

```
GIZMO! SAYS: LET THIS SHIT FLOW !!!!!
```

```
Insert Disk to be Patched in Drive DF1:
RETURN writes new Sectors!!
```

```
Boot-Sectors now modified.....Enjoy!!
```

Es wird der SCA -Bootblockvirus auf den Bootblock der Diskette in DF1: geschrieben. Es besteht lediglich folgender Textunterschied:

```
Try ANTIVIRUS from DAG
```

Sollte kein DF1: vorhanden sein erfolgt ein Task-Held.

1.357 virusmaker v1.0

VirusMaker V1.0

Nach dem Start dieses 84216 Byte langen entpackten Programmes erscheint ein optisch ansprechender Schirm, der das Installieren folgender Bootblockviren anbietet:

```
ByteBandit,DASA-ByteWarrior,SCA,Gadaffi,
ASS(ist kein Bootblockvirus),NorthStar1+2,SystemZ
```

1.358 virusconstructionseti

The Virus Construction Set von STR

Es handelt sich um ein 10192 Byte langes mit PowerPacker gepacktes File, entpackt 19452 Bytes lang, mit welchem man einen Bootblockvirus auf DF0: schreiben kann. Nach dem Start des Virus Construction Set wird der Bootblockvirus ab \$7f000 abgelegt. Sie haben nun die Möglichkeit, einen bis zu 60 Buchstaben langen Text einzugeben, welcher dann nach 5 Reset als Alertmeldung verwendet wird. Ansonsten können Sie nichts beeinflussen. Der Bootblockvirus arbeitet folgendermaßen:

Es handelt sich um einen unsauber programmierten Bootblockvirus, welcher im Speicher ab \$7f000 steht. Er macht sich über den COOL-Vektor(\$7F0BE) resistent und verbiegt nach dem nächsten Reset den DoIO()-Vektor(\$7F0D2), um jede eingelegte beschreibbare Diskette zu infizieren. Nach 5 Resets wird ein Alert mit dem im Virus Construction Set eingegebenen Text angezeigt. Das Anzeigen des Alerts klappt aber nur dann, wenn mindestens 512 KB sogenanntes Ranger-RAM ab \$c00000 vorhanden ist, da der Virus die intuition.library-base in \$0c3af7 abzuspeichern versucht.

1.359 virusconstructionsetii

The Virus Construction Set von STR V2.0

The Virus Construction Set von STR V2.0 ist die Weiterentwicklung des The Virus Construction Set von STR . Es handelt sich um ein 32360 Byte langes mit PowerPacker gepacktes File, entpackt 47944 Bytes lang, mit welchem man einen Bootblockvirus auf DF0: schreiben kann. Nach dem Start des Virus Construction Set V2.0 haben Sie die Möglichkeit, einen bis zu 60 Buchstaben langen Virus-Text einzugeben, danach können Sie einen bis zu 20 Buchstaben langen Virusnamen eingeben, und danach werden Sie gefragt ob ein verschlüsselter oder unverschlüsselter Bootblockvirus geschrieben werden soll.

Nun zu dem Bootblockvirus. Im Gegensatz zu seinem Vorgänger steht dieser Bootblockvirus nicht an einer bestimmten Speicherstelle, beschädigt aber dennoch Speicher ab \$70000. Der kodierte und dekodierte Bootblock sind praktisch identisch, anhand eines Flags erkennt der Bootblock, ob eine eventuelle Dekodierung nötig ist. Das Funktionsprinzip ist ähnlich wie bei dem Vorgänger, das heißt der Virus macht sich über den COOL-Vektor resetfest, um nach dem nächsten Reset den DoIO()-Vektor zu verbiegen, um dann beschreibbare Disketten bereits beim Einlegen zu infizieren. Nach 5 Resets erscheint eine Alertmeldung mit dem früher eingegebenen Text. Der Intuitionbase-Fehler der früheren Version wurde korrigiert. Der kodierte Bootblock stürzt jedoch auf höheren Prozessoren ab, weil er im Endeffekt mit selbstmodifizierendem Code arbeitet.

1.360 aaa-enhancer

AAA-Enhancer

Nach dem Start des 3984 Byte langen Programmes, das erst ab Kickstart 2.0 läuft, wird in einem Fenster Text ausgegeben, welcher besagt, daß dieses Programm auf neueren A1200 und A4000 die bereits versteckt vorhandenen AAA-Grafikeigenschaften aktivieren würde. In Wirklichkeit aber wird der Write()-Vektor verbogen, um folgende Wörter bzw. Zahlen gegeneinander auszutauschen:

perverse	<->	reliable
Computer	<->	vibrator
sexual	<->	actual
friend	<->	bugger
pocket	<->	vagina
follow	<->	stroke
randy	<->	ready
blood	<->	sperm
bitch	<->	woman
head	<->	hole
rich	<->	poor
warm	<->	cold
open	<->	lock
love	<->	hate
meet	<->	fuck
lift	<->	drop

girl	<->	wife
kill	<->	kiss
look	<->	piss
nice	<->	shit
soft	<->	hard
ball	<->	hand
cock	<->	nose
dear	<->	dead
skin	<->	cunt
egg	<->	lip
car	<->	ass

0	<->	9
1	<->	8
2	<->	7
3	<->	6
4	<->	5

Hierdurch wird oftmals der Sinn eines Textes entstellt oder ins Gegenteil verkehrt. Es sind aber nicht nur Texte betroffen, da beim Kopieren von startbaren Programmen meist nicht mehr lauffähige Kopien entstehen, da auch die Assemblerbefehle des öfteren rein zufällig teilweise als '0', '1' usw. erscheinen. Die Wörter bzw. Zahlenveränderungen sind allerdings umkehrbar, da z.B. '0' immer in '9' und '9' immer in '0' geändert wird. Durch erneutes Abspeichern der veränderten Daten erhält man wieder die ursprünglichen Daten zurück.

In dem Programm kann man weiterhin negativen Text bezüglich der SHI (Safe Hex International) Gruppe lesen.

1.361 a.i.s.f. interlamer

A.I.S.F. INTERLAMER

Nach dem Start des 8708 Byte langen Programmes wird der Viruscode nach \$7F000 kopiert und zur Tarnung eine Fensterleiste mit dem Titel 'Virus-Checker V6.72' geöffnet. Weiterhin wird der Vertikal-Blank-Interrupt \$6c verbogen, um nach 106 Minuten einen Alert mit folgendem Text auszugeben:

```
    !!! CRIME DO NOT PAY !!!
WHY ARE YOU SWAPPING ILLEGAL SOFT ?
    BECAUSE YOU ARE A CRIMINAL !!!!
        AND BE SURE:
WE (A.I.S.F.) WILL GET YOU !
(A)NTI
(I)LLEGAL
(S)WAPPING
    (F)OUNDATION
    -PRESS MOUSE TO CONTINUE-
```

Nach dem Drücken einer Maustaste wird durch entsprechendes Steppen des Diskettenlaufwerkschreiblesekopfes ein schwirrendes Geräusch erzeugt. Die Diskettendaten bleiben hierbei jedoch meist erhalten. Um einen dauerhaften Schaden des Diskettenlaufwerkes zu vermeiden, sollten Sie möglichst schnell von Hand einen Reset auslösen.

Wenn man das Programm mit dem Parameter * startet, dann installiert sich der Virus nicht und gibt lediglich folgende Textmeldung in der Shell aus:

```
WOW! YOU GUY MUST BE ELITE !!!!!
```

Die Texte liegen in dem Programm verschlüsselt vor und sind daher nicht lesbar. Gegen Ende des Programmes kann man jedoch mehrmals lesen:

```
THE A.I.S.F. INTERLAMER-VIRUS
```

1.362 aibon

Aibon

Nach dem Start von Express2.20 (194064 Bytes) oder acp.ctrl (56016) wird mit jmp an das Ende des ersten Code-Hunks gesprungen, um sofort mit schädlichen Aktionen zu beginnen. Als erstes wird der serielle Port des CIA-A von Eingabe auf Ausgabe geschaltet, wodurch der serielle Bit-Strom von der Tastatur nicht mehr verarbeitet wird. Die Tastatur ist also praktisch tot, lediglich die Resettastenkombination, welche ja nicht über den seriellen Port des CIA-A abgearbeitet wird, sondern eine eigene Hardwareschaltung besitzt, zeigt noch Wirkung, allerdings unter Umständen erst nach circa 10 Sekunden, und zwar dann wenn Ihr Amiga sogenannte Keyboardresethandler unterstützt, was bei neueren A2000 und allen A3000, A4000 der Fall ist. Während dieser 10 Sekunden arbeitet der Amiga und somit auch der Virus normal weiter. Als erstes wird ein 776 Byte langes File namens s:aibon erstellt und dann eine neue s:startup-sequence geschrieben mit nur s:aibon,\$0a,\$0a als Inhalt. Danach wird ein Datenträger namens bbs: angefordert, wenn dies nicht gelingt, macht sich der Virus über sys: her, das heißt er überschreibt die Mehrzahl der vorhandenen Dateien der Bootdiskette oder Autobootfestplatte mit 42 Bytes ab Speicherstelle 0. Es werden unter Umständen mehrmals die gleichen Dateien überschrieben oder auch manchmal ein Reset ausgelöst, es kommt also nie zur Ausführung des ursprünglichen z.B. Express2.20-Programmes. Nach dem nächsten Reset wird das eigentliche s:aibon-Virusfile durch die startup-sequence aufgerufen, als Erstes wird wiederum die Tastatur blockiert, ein Erstellen von s:aibon und Neuschreiben der s:startup-sequence unterbleibt allerdings, denn in der Regel wurde dies ja bereits beim Installieren durch Express2.20 oder acp.ctrl vorgenommen, es wird also gleich zum Anfordern von bbs bzw. sys und Überschreiben der Dateien übergegangen. Bei schreibgeschützten Disketten erscheinen entsprechende Systemrequester.

Express2.20 (194064 Bytes) und acp.ctrl (56016) sind also als Trojanische Pferde zu betrachten, welche den eigentlichen 776 Byte langen aibon-Virus installieren. Am Ende der Files kann man lesen: bbs: sys: ram: dos.library

1.363 aibon2

Aibon2

Es gibt mittlerweile eine Weiterentwicklung des Aibon -Virus, nennen wir ihn aibon2. Er wird durch ein 1872 Byte langes File installiert. Eben dieses Installllierungsprogramm wurde auch mittels Hunklab vor das 41468 Byte lange DWEdit1.62-Programm (siehe dirwork auf fishdisk 721) gelinkt, wodurch also ein 43700 Byte langes Trojanisches Pferd entstanden ist. Wohingegen bei Installierungen des aibon sofort zur Sache gegangen wird, wird der aibon2 auf viel raffiniertere Weise installiert. Es wird beim Start ein eigener Prozess mit dem unauffälligen Namen 'BackGround_Process' abgesetzt, der zunächst mal nur 57 Minuten lang wartet. Das ursprüngliche DWEdit1.62 gelangt also sofort zur Ausführung und dem User kann nichts Ungewöhnliches auffallen. Nach 57 Minuten prüft der Virus dann auf das Vorhandensein eines Ports namens 'ser.read'. Nur in dem eher seltenen Fall, daß ein solcher Portname existiert, werden die bereits von aibon bekannten Virusinstallierungs- und Lösch-Aktionen unternommen. Am Ende des 784 Byte langen aibon2-Files kann man lesen: sys: bbs: df0: df1: dos.library. ram: wurde also durch df0: df1: ersetzt. Es existiert eine Aibon2-Variante, die mittels Hunklab an ToolsDaemon2.2 gelinkt wurde. Als Virusfilename wird das weniger verdächtig erscheinende mount anstatt aibon verwandt und anstatt sys: bbs: df0: df1: werden nun Dateien in hd0: sys: ram: zerstört.

1.364 bootx-updater

BootX-Updater

Nach dem Start dieses 2052 Byte langen Programmes wird ein Fenster geöffnet, in welchem Text ausgegeben wird, welcher besagt, daß dieses Programm die BootX-Recognitionfiles updaten würde, wodurch also BootX 87 neue Viren kennen würde, in Wirklichkeit wird aber mittels Execute() eine Schnellformatierung von WORK: DH0: und DF0: versucht, hierbei wird vorher auf die Existenz des format-Befehls und der Partitionsnamen geprüft und die Systemrequester abgeschaltet, damit sich das Programm nicht sofort durch Errormeldungen verrät. In der Regel lassen sich die Daten mit z.B. disksalv restaurieren. In dem Programm kann man weiterhin negative Äußerungen über SHI (Safe Hex International) lesen.

1.365 byteparasitei

ByteParasiteI

Es handelt sich nicht um einen Virus. Das Programm ist nicht resetfest und infiziert auch keine neuen Disketten. Die einzige Funktion des ByteParasite-Programms ist das primitive Zerstören von Daten. Wenn man das 2108 Byte lange Programm aufruft, dann wird z.B. die Startup-Sequence und der dir-Befehl mit dem cd-Befehl überschrieben. Danach wird noch der Interrupt 3 (\$6c) verbogen. In dieser Interruptroutine wird geprüft, ob soeben eine Diskette eingelegt wurde. Wenn ja, erfolgt ein Absturz. Aufgrund schlechter Programmierung erfolgt z.B. auch beim

Versuch eine schreibgeschützte Diskette zu schädigen ein Absturz.
Es macht wenig Sinn, die Startup-Sequence mit einem Programm zu überschreiben, da dadurch beim nächsten Booten die Startup-Sequence nicht mehr abgearbeitet wird.

Das ByteParasite-Programm ist also ein sehr plumpes und lediglich auf Zerstörung ausgelegtes Programm. Erfreulicherweise hält sich der Schaden in Grenzen, da meist nur die Startup-Sequence verlorenggeht.

Da sich das Programm nicht automatisch verbreiten kann, werden Sie wohl hoffentlich niemals diesem Programm begegnen.
Am Ende des ByteParasite-Files kann man folgenden Text lesen:

```
dos.library cd dir s/Startup-Sequence cd
by ByteParasite in 9.90 from Hacker & Cracker GmbH GERMANY
```

1.366 byteparasiteii

ByteParasiteII

Es handelt sich um einen relativ harmlosen Filevirus.
Wenn man das 908 Byte lange Programm startet, dann erscheint eine Fensterleiste mit folgendem Text: VirusX: Checking Device DF0:
Damit will sich das Programm als das Antivirusprogramm VirusX tarnen.
Nach Anklicken des Closegadgets wird versucht DF0:c/VirusX nach DF1:c/VirusX zu kopieren. Auf diese ziemlich uneffektive Weise versucht sich der Virus weiterzuverbreiten. Voraussetzung wäre also, daß das Virusfile als DF0:c/VirusX vorliegt und daß ein Schreibzugriff auf DF1:c möglich ist.
Außer diesem relativ harmlosen Kopierversuch passiert nichts.
Am Ende des ByteParasite-Files kann man folgenden Text lesen:

```
Now you had have your fun with BYTEPARASITE II !
by Hacker & Cracker GmbH GERMANY in 9.91
```

1.367 byteparasiteiii

ByteParasiteIII

Wenn man dieses 2160 Byte lange Virusfile startet, dann werden eine ganze Menge Vektoren aufgrund fehlerhafter Programmierung auf zum Teil unsinnige Werte gesetzt, wodurch recht bald ein Absturz resultiert. z.B. beim nächsten Disk-Einlegen. Der Virus versucht sich als c/Virus-Checker weiterzukopieren. Aufgrund fehlerhafter Programmierung bleibt es aber meist beim Versuch.
Es wird KickTag, KickMem, KickChecksum, Cold, Cool und Interrupt3(\$6c) verändert. Das Programm versucht sich durch folgende Titelzeile als Antivirusprogramm zu tarnen:

```
Virus-Checker V3.0    Checking DF0: For Viruses
```

Am Ende des ByteParasite-Files kann man z.B. folgenden Text lesen:

```
BYTEPARASITE III in 9.91 by Hacker & Cracker GmbH.  GERMANY
```

Es existiert ein ByteParasiteIII-Abkoemmling namens FCheck, der

lediglich Textunterschiede aufweist.

Bulgarian Antilamer and Virihacker & Cracker GmbH. BULGARIA

In ByteParasiteII+III wurde versucht, eine Infektionsroutine einzuprogrammieren, dennoch führe ich diese beiden Viren unter Punkt b. auf, da die Infektionsroutine normalerweise nicht funktioniert, und in der Regel lediglich zum Zerstören von Daten führt. Auch müßte ein funktionsfähiger Filevirus z.B. die Startup-Sequence manipulieren, um möglichst bei jedem Booten aktiv zu werden.

1.368 chaos-master v0.5

CHAOS-MASTER V0.5

Der CHAOS-MASTER tarnt sich als normaler dir-Befehl.
Darüber hinaus geschieht allerdings noch folgendes:

Nach Start des 12972 Byte langen mit PowerPacker gepackten Programms wird versucht das Virusfile unter dem aktuellen Namen nach SYS:c zu kopieren. Hierbei werden jedoch immer nur 12972 Bytes kopiert, wodurch ein nicht startbares File resultiert, wenn das Virusfile zuvor entpackt wurde.

Sollte jedoch ein Kopieren nach SYS:c nicht möglich sein, dann wird ein fehlerhaftes 370 Byte langes disk.info-File erstellt. Die Workbench wertet das disk.info-File für das Aussehen des Disketten-Icons aus. Dadurch kommt es bei loadwb und eingelegter Diskette mit fehlerhaftem disk.info-file oder beim späteren Einlegen einer Diskette mit einem fehlerhaften disk.info-File und aktiver Workbench zum Absturz. Sollte jedoch der loadwb-Befehl noch nicht ausgeführt worden sein, dann kann man die Diskette problemlos einlegen, da das disk.info-File nur bei aktiver WorkBench ausgewertet wird. Unter Kickstart 2.0 führt ein fehlerhaftes disk.info-File nicht mehr zum Absturz.

Am Ende des CHAOS-MASTER-Files und in dem disk.info-File kann man folgenden Text lesen:

Sorry an alle User, die sich jetzt mit CHAOS-MASTER V0.5 beschäftigen dürfen!!!!!! Diese Virus ist nicht für 'LAMER', sondern die Saftsäcke, die ins ZERBERUS Mails im Sinne von Wer sich mit Virenkillern nicht auskennt ist selbst Schuld geschrieben!!!

Der CHAOS-MASTER V0.5 führe ich auch bei Unterpunkt b. auf, da es sich hierbei nicht um einen wirklich funktionsfähiger Filevirus handelt, denn hierfür bedarf es mehr als dem bloßen Kopieren eines Files. So müßte z.B. durch Manipulation der Startup-Sequence eine Aktivierung des Virus beim Booten sichergestellt werden. Oder aber der Virus sollte resetfest sein.

1.369 commodore-virus

Commodore-Virus

Das 1752 Byte lange Programm weist programmtechnische Ähnlichkeiten mit dem DISK-KILLER V1.0 -Virus auf, so wird z.B. auch das Programm sofort beendet, wenn man beim Start einen nicht existierenden Filenamen als Paramter angibt. Allerdings ist der Commodore-Virus schlechter programmiert, so wird z.B. der COOL-Vektor auf eine Textpassage verbogen und die execchecksumme nicht neu berechnet. Das Programm kann also nicht resetfest sein und beschränkt sich somit also auf folgende schädliche Aktionen. Das Programm zählt bei jedem Aufruf die Speicherstelle \$66666 um eins hoch. Wenn der Wert 2 vorliegt wird s/Startup-Sequence gelöscht und ein neues Verzeichnis namens 'Commodore war hier !!' erstellt. Danach wird eine Alert mit folgendem Text ausgegeben:

```
Ihr Computer ist überhitzt !!!  
Wenn es nach dem Reset ein absturz gibt  
SCHALTEN SIE BITTE AUS  
Commodore 1987
```

Wenn der Wert 3 vorliegt wird ebenfalls versucht s/Startup-Sequence zu löschen und ein neues Verzeichnis namens 'Commodore war hier !!' zu erstellen. Sollte die Diskette schreibgeschützt sein, wird nun aber extra folgender Text im CLI ausgegeben:

```
USER-REQUEST  
Please remove the Write-Protection  
And Press Mouse-Button to Continue
```

Nachdem man die Diskette beschreibbar gemacht hat und der Virus die Diskette bearbeitet hat wird zur Tarnung ein CLI-Fenster mit folgender Text-Ausgabe aufgemacht:

```
KEIN VIRUS IN DRIVE DF0: GEFUNDEN !!  
Commodore 1987
```

Weiterhin kann man in dem Virusfile noch folgenden Text lesen:

```
You have found the Routine !  
This is the new Commodore-Virus !  
BY STARLIGHT ENTERPRISES 1992
```

1.370 compuphagozyte1

CompuPhagozytel

Wenn man das ungepackte 1452 Byte lange Programm startet, dann wird ein Fenster mit nur Titelhöhe geöffnet, wobei der Titel lautet:

```
Virus-Checker V4.0 by Michael Ortmanns
```

Danach werden 1452 Bytes aus einem eventuellen ':c/Virus-Checker'-File nach \$7c000 gelesen. Auf diese 1452 Bytes wird dann bei jeder eingelegten Diskette ein eventuelles ':c/Virus-Checker'-File gekürzt. Wenn man das Virusprogramm mit der Angabe '--che' startet, dann erscheint als Fenstertitel

```
Checking DF0: For Viruses
```

und nach dem Versuch 1452 Bytes aus einem eventuellen
' :c/Virus-Checker'-File nach \$7c000 zu lesen, wird das
Programm sofort wieder beendet. Anderfalls muß man das
Programm von Hand durch Anklicken des Close-Gadgets beenden.
Man kann in dem CompuPhagozytel-File folgenden Text lesen:

```
The CompuPhagozyte has attached to your system !  
Wait for much more better virus on other systems, too!  
The CompuPhagozyte in 9.91 by The Emperor Of Trillion Bytes !
```

CompuPhagozytel ist also ein Programm, welches dem Ruf des
Antivirusprogrammes 'Virus-Checker' und dessen Programmierer
böswillig Schaden zufügt.

In dem CompuPhagozytel-File ist auch eine Routine enthalten, die mittels
direkter Hardwareprogrammierung einige Sektoren der Diskette beschädigen
kann. Diese Routine wird aber nur angesprungen, wenn die Diskette
schreibgeschützt ist, wodurch sie also unwirksam bleibt und außerdem
wird diese Routine um 2 Bytes zu früh angesprungen, wodurch sie eh nicht
richtig funktioniert.

1.371 compuphagozyte2

CompuPhagozyte2

Wenn man das ungepackte 1148 Byte lange Programm startet, dann wird
ein Fenster mit nur Titelhöhe geöffnet, wobei der Titel lautet:

```
VirusX 5.00 by Steve Tibbett
```

Danach werden 1148 Bytes aus einem eventuellen ' :c/VirusX'-File
nach \$7c000 gelesen. Auf diese 1148 Bytes wird dann bei jeder eingelegten
Diskette ein eventuelles ' :c/VirusX'-File gekürzt.

Man kann in dem CompuPhagozyte2-File folgenden Text lesen:

```
The CompuPhagozyte has attached to your system !  
Wait for new virus in other computer-systems !  
The CompuPhagozyte in 9.91 by The Emperor Of Trillion Bytes !
```

CompuPhagozyte2 ist also weitgehend mit CompuPhagozytel identisch,
anstatt Virus-Checker wird VirusX verwandt und die kürzere Filelänge
ist dadurch bedingt, daß keine Parameterangabe wie bei CompuPhagozytel
('-che') unterstützt wird.

1.372 conman-trojan

CONMAN-TROJAN VIRUS

Das 4004 Byte lange gepackte Programm besteht aus einem dir-Befehl
mit von Hand angehängtem Viruscode. Beim Start des Programmes werden

4004 Bytes aus C:DIR nach \$70000 kopiert, weil der Virus annimmt, daß er als C:DIR verbreitet wird. Weiterhin wird ein Virusprozess namens "Workbench " kreiert, welcher ein vorhandenes SnoopDos zu beenden versucht, damit die Virusdateizugriffe möglichst nicht auffallen sollen. Wenn der Virusprozess einen Diskettenwechsel feststellt, schreibt er die 4004 Bytes ab \$70000 als neues DF0:C/DIR-File. Nach mehreren Disketteninfektionen wird bei der eingelegten Diskette DEVS:SYSTEM-CONFIGURATION C:LOADWB L:RAM-HANDLER gelöscht und anschließend folgender Alert ausgegeben:

```
THIS IS NOT A SYSTEM ALERT! THIS IS THE NEW CONMAN-TROJAN VIRUS!
ALL DISK ACTIVITIES WILL BE DISABLED!
GREETINGS TO JOE/DEFJAM BRUCE/DEFJAM NATAS/DEFJAM ALEX/DEFJAM AND DOC!
CONTACT ME 030-615-89-92 USR 1.4. NO STUFF! ONLY VIRUS-PROGRAMMER AREA!
```

Anschließend ist keine normale Weiterarbeit mehr möglich, da der Mauszeiger und die Diskettenlaufwerke blockiert sind.

1.373 d&a

D&A

Dieses 1052 Byte lange Programm installiert einen SCA-artigen Virus, welcher sich über den COOL-Vektor resetfest macht und beim Booten die Bootdiskette mit Hilfe des kurzzeitig verbogenen DoIO()-Vektors zu infizieren versucht. Anstatt der DOS-Kennung wird jedoch \$00000400 geschrieben, wodurch diese Diskette nicht mehr als DOS-Diskette erkannt wird. Ansonsten entspricht der Bootblock weitgehend dem SCA-Bootblock, lediglich einige Textveränderungen von SCA in D&A wurden vorgenommen. Die Ursache für diesen fehlerhaften D&A-Bootblock liegt in einer fehlerhaften Konvertierung des Original-D&A-Bootblocks in ein startbares File.

1.374 decompiler

Decompiler

Es handelt sich um ein 53992 Byte langes Amos-Programm, welches folgende Verzeichnisse umbenennt

```
'sys:libs' in 'sys:libs '
'sys:l'    in 'sys:l '
'sys:Devs' in 'sys:Devs '
'sys:fonts' in 'sys:fonts '
```

Durch das Anhängen des Leerzeichens können Ihre Programm meist nicht mehr richtig arbeiten, da sie wichtige Systemdaten nicht mehr finden können.

1.375 degrad

Degrad

Es handelt sich um ein 5612 Byte langes File, welches zum Großteil aus Null-Bytes besteht. Wenn man das Programm startet, wird versucht das scsi.device mit Unit=0 und flags=0 zu öffnen, um dann ab Block 0 5120 Null-Bytes zu schreiben, wodurch der Rigidiskblock der entsprechenden Festplatte beschädigt wird, wodurch die Festplatte nicht mehr eingebunden wird.

1.376 descriptor v3.0

Descriptor V3.0

Es handelt sich um ein 7016 Byte langes schädliches Programm, welches unter Kickstart 1.2/1.3 unter Ausgabe folgenden Textes im CLI sofort wieder beendet wird: Hey yo lame bastard! What about kick 2.0?

Ab Kickstart 2.0 nimmt das Programm seine Arbeit auf und öffnet zwei untereinander angeordnete Fenster mit folgendem Aussehen:

Descriptor V3.0 by Colorboy of Submission on 29-JUN-1993

```

.. /\ ..... /\ ..... /\ .....
:/  - - - - | /  - - - - | /  - - - - | SUBMISSION `:
/    |  _  |    |  _  |    |  _  | : .....
\ /\_   \      \ /\_   \ /\_   \ : THE:SIGN `::
/    |  /    |  /    |  /    | / : ...OF.....
\   _ _ _ / \ _ _ _ / \ _ _ _ / : ::STYLE`:::
` \ /      DESCRIPTOR   \ / B-S: .....

```

Credits
-^o^- - ^o^-
CODING BY COLORBOY
DEZIGN &IDEA BY MCI
TYPE IN A GROUP NAME NOW!
(C)OPYRIGHT SUBMISSION 93

-\$\div\$!\\$ \div\$- Your HD is deleted ... Happy Birthday -MCI-/DCS AHHAHAHAHAHAHA -\$ ←
 \div\$!\\$ \div\$-

Descriptor V3.0 User Interface

```

Search for Group Name      ^ø^      Update by MCI      ^ø^

List all Names  Pop Screen back About
View next Logo  Reload Descriptions ^Submission 93^ Window backpop Quit

```

Anschließend wird versucht eine bereits existierende Datei namens S:Descriptions.TXT zu öffnen, nur wenn dies gelingt arbeitet der Virus weiter, zunächst wird die Zeile

-\$\div\$!\\$ \div\$- Your HD is deleted ... Happy Birthday -MCI-/DCS AHHAHAHAHAHAHA -\$ ←
 \div\$!\\$ \div\$-

in -\$ \div\$!\\$ \div\$- Snap is not in memory. I try to load it from C-Dir. -\$ ←
 \div\$!\\$ \div\$-

geändert. Danach wird über die dos.library-execute-Funktion folgender CLI-Befehl zur Ausführung gebracht:

```
delete :#? all
```

Allerdings werden die Meldungen des delete-Befehls nicht umgeleitet, so daß man also anhand der in der shell erscheinenden delete-meldungen doch deutlich auf die schädlichen Aktionen des Programmes hingewiesen wird.

Nachdem der delete-Befehl abgearbeitet ist, wird die oben erwähnte Zeile erneut verändert und zwar erscheint folgender Text:

```
-$\div$!\$\div$- Use underlined characters as hotkeys and follow the directions. - ←  
$\div$!\$\div$-
```

1.377 disk-killer v1.0

DISK-KILLER V1.0

Es handelt sich um ein 1368 Byte langes schädliches Programm, welches nach dem Start etwas Text im CLI ausgibt und danach sofort alle Disketten durch direkte Hardwareprogrammierung schnellformatiert. Die Daten können meist nicht mehr rekonstruiert werden. Wenn man aber das DISK-KILLER-Programm mit einem nicht existierenden Filenamen als Paramterangabe aufruft, dann wird das Programm sofort beendet, ohne Disketten zu formatieren.

1.378 diskspeedcheckv1.01ß

DiskSpeedCheckV1.01ß

Nach dem Start dieses 33152 Byte langen Files wird folgender Text im CLI ausgegeben:

```
Disk Speed Check V1.01ß - © Micro-Tech Softwares® 1992  
Programming by Alan Forslake. (1.10.1992)  
Usage : DSC -mode <drive>, Where mode is :  
1=disk speed check 2=scsi speed test
```

Das Programm täuscht also vor, (SCSI-)Festplatten auf ihre Geschwindigkeit zu testen. Damit man dieses dem Programm auch zutraut, wurde die Filelänge mit Nullbytes von circa 500 Bytes auf 33152 Bytes aufgeblasen.

Dieser Text dient aber lediglich der Tarnung, in Wirklichkeit werden folgende eventuelle Dateien gelöscht:

```
sys:s/startup-sequence  
bbs:user.keys  
bbs:user.data  
s:acp.startup
```

Danach wird noch folgender Text ausgegeben, um den Anwender zu beruhigen, warum kein Speedtest durchgeführt wurde.

```
Didnt` found a supported SCSI drive, sorry!  
Try to contact Alan Forslake on :223/22/32 in@sf@com
```

1.379 diskroyer

Diskroyer V1.0

Es handelt sich um ein 804 Byte langes File, welches AllocMem() verbiegt. Beim Start des Programms wird das betreffende CLI-Fenster in Art eines cls-Befehls gelöscht. Der Virus tarnt sich also als cls-Befehl. Wenn 150 Mal allocmen aufgerufen wurde, was je nach Systemaktivität recht bald der Fall ist, werden nicht schreibgeschützte Disketten schnellformatiert. Danach wird ein Alert mit folgendem Text ausgegeben, welchen man auch am Ende des Diskroyer-Files lesen kann.

```
Diskroyer V1.0  
(w) and © 1991 by the powerful  
The Fanatic Crew  
Switch off and reboot (ha, ha, ha, ha ...)  
released 09.09.1991
```

Diskroyer V2.0

Es existiert noch ein 812 Byte langes Diskroyer2.0-File, welches praktisch dem Diskroyer 1.0 entspricht, anstatt AllocMem() wird GetMessage() verbogen und als Zähler anstatt \$96 \$222222 verwandt und im Alert ist anstatt 09.09.1991 30.11.1991 zu lesen.

1.380 d-structure(a,b,c)

D-Structure (A,B,C)

Es handelt sich um ein schädliches Programm, von dem 3 geringfügig verschiedene Versionen bekannt sind. Die früher programmierten 352 und 428 Bytes lange Versionen, auch Typ A und B genannt, sind noch recht fehlerhaft programmiert, gegen Ende dieser beiden Versionen kann man lesen:

```
'D-Structure'
```

Lediglich die neuere 464 Byte lange Version, auch Typ C genannt, arbeitet recht stabil, gegen Ende dieser Version kann man lesen:

```
'D-Structure '
```

Wenn man ein D-Structure-Programm startet, dann wird der Viruscode fest nach \$7C000 kopiert und der OldOpenLibrary()-Vektor auf den Viruscode verbogen. Hier wird nun geprüft, ob ein OldOpenLibrary-Aufruf für die

'dos.library' anliegt, wenn ja führt der Virus diesen Aufruf über OpenLibrary() aus und verbiegt in der nun erhaltenen dos.library-Adresse den Write()-Vektor, der OldOpenLibrary()-Vektor hingegen wird wieder restauriert. Der Virus schreibt nun bei jedem fünften Write()-Aufruf willkürlich das komplette D-Structure-file (z.B. in ein shell-Fenster oder eine Datei). Die ursprünglich zu schreibenden Daten gehen verloren.

1.381 elien

Elien

In dem 1016 Byte langen, PowerPacker gepackten, ungepackt 596 Byte langen File kann man u.a. lesen

Elien_virus_checker v0.1 by zupa/T.L.X.

Dieser Text wird auch bei der version-Abfrage ausgegeben.

Mit einem VirusChecker hat das Programm allerdings nichts gemeinsam, denn das Elien-Programm versucht lediglich eine 900000 Byte lange Datei namens sys:MeGaSUXX.TXT zu erstellen, in welcher immer wieder 9 kleine a-Buchstaben gefolgt von einer Null stehen.

1.382 excreminator v1.0

Excreminator V1.0

Es handelt sich um ein 2392 Byte langes schädliches Programm, welches allerdings zum Großteil aus Null-Bytes besteht.

In dem Programm kann man verräterische Texte erkennen.

Von einem Virus kann man nicht direkt sprechen, da sich das File nicht selbstständig verbreiten kann. Es werden auch keine Vektoren verbogen. Beim Start des Programms wird lediglich versucht, eine Datei namens df0:libs/Exec.library zu öffnen.

In dieser 4 Byte langen Datei wird lediglich ein Zähler abgelegt.

Bei jedem Start des Programms wird der Zähler um 1 erniedrigt.

Sobald der Zähler von 5 auf 0 herabgezählt wurde, werden alle Disketten schnellformatiert. Anschließend wird der folgende Alert ausgegeben:

FUCKED UP! LAME SUCKER !!!

Use a better Viruskiller next time

e.g. Excreminator II HAHAAHA

Das Programm tarnt sich durch folgende Textausgaben als ein Antivirusprogramm. Man kann also auch von einem Trojanischen Pferd sprechen.

-- Excreminator V1.0 --

Written by 'The Lame Trio (TLT)' in 1991

Memory Check ... OK! No Virus found!

Checking Bootblock for Virus ... OK! No Virus found!

Der Virus ist nicht direkt mit timebomb oder virustest verwandt, aber es liegt das gleiche Funktionsprinzip vor.

1.383 freedom

Freedom

Es handelt sich nicht um einen Virus. Das Programm ist nicht resetfest und infiziert auch keine neuen Disketten. Die einzige Funktion des Freedom-Programms ist das primitive Zerstören von Disketten-Daten. Wenn man das 10876 Byte lange Programm aufruft, dann wird folgende Meldung ausgegeben:

```
Freedom !           by Steve Tibbett
Checking df0: for 126 viruses
```

Sollte die Diskette schreibgeschützt sein, dann wird
Diskette ist schreibgeschützt !! ausgegeben und Ende.

Damit soll der Anschein eines nützlichen Antivirusprogramms erweckt werden. Man kann also von einem Trojanischen Pferd sprechen. Aber anstatt auf Viren zu prüfen wird sofort angefangen, willkürlich Diskettenblöcke zu überschreiben. VIRUS CONTROL meldet sofort einen 512-Byte-Zugriff auf den Bootblock. Hierbei sieht man, daß lediglich unsinnige Daten geschrieben werden. Die Diskette wird dadurch unlesbar. Nach einiger Zeit werden folgende Meldungen ausgegeben:

```
Saddam-Virus removed und kurz darauf nochmal
Saddam-Virus removed
```

und

```
SmilyCancer-Virus removed und kurz darauf nochmal
SmilyCancer-Virus removed
```

Je länger man das Freedom-Programm laufen läßt, umso mehr Daten gehen verloren, da sie mit unsinnigen Daten überschrieben werden.

Das Freedom-Programm ist ein sehr plumpes und nur auf Diskettendatenzerstörung ausgelegtes Programm. Wahrlich keine Programmiermeisterleistung. Da sich das Programm nicht automatisch verbreiten kann, werden Sie wohl erfreulicherweise niemals diesem Programm begegnen. Das Freedom-Programm ähnelt sehr stark dem VirusBlaster -Programm.

1.384 timebomb v0.9

TimeBomb V0.9

Es handelt sich um einen Filevirus, welcher durch das Programm BMassacre erstellt wird. BMassacre erstellt in C: ein 7840 Byte langes Programm namens

.info Weiterhin trägt BMassacre in die erste Zeile der Startup-Sequence .info ein, damit das Virusfile bei jedem Booten aufgerufen wird, denn das Virusfile ist nicht resetfest. Ferner wird eine 1 Byte lange Datei namens 'df0:pic.xx' angelegt, in welchem die Zahl 6 steht. Das Virusprogramm kann sich nicht weiterverbreiten. Der Virus kann also nur von dem Viren-Generator-Programm BMassacre erstellt werden. Das Virusfile selber macht nun folgendes. Es wird bei jedem Booten von der Startup-Sequence aufgerufen. Der Virus öffnet nun die Datei 'df0:pic.xx' und liest das eine Byte ein. Der Wert wird um 1 erniedrigt. Wenn der Wert 0 erreicht hat, dann wird versucht, die Diskette zu formatieren. Danach wird folgende Meldung ausgegeben und anschließend ein TaskHeld ausgelöst.

Hey Looser ! I hate you !

Wenn der Wert noch nicht 0 erreicht hat, dann wird der neue Wert unter dem File 'df0:pic.xx' wieder auf Diskette zurückgeschrieben. Sollte die Diskette aber schreibgeschützt sein, dann wird folgende Text-Meldung im CLI ausgegeben:

User Request : Please remove write Protection and press
left Mouse Button to continue

Diese Meldung wird solange ausgegeben, bis man den Schreibschutz wirklich entfernt hat. Nach erfolgreichem Schreiben von 'df0:pic.xx' wird zum Abschluß noch folgende Meldung im CLI ausgegeben:

RAM CHECKED - NO VIRUS FOUND

Diese Meldung dient also lediglich der Tarnung.

1.385 timebomber(virustest)

TimeBomber(VIRUSTEST)

Es handelt sich um eine verbesserte Version des TimeBomb V0.9 .
Es bestehen lediglich folgende Unterschiede:

TimeBomber(VIRUSTEST)

TimeBomb V0.9

Der TimeBomber-Virus wird von
dem Programm TimeBomber erstellt.

Der TimeBomb V0.9-Virus wird von
dem Programm BMassacre erstellt.

Das Virusfile heißt VIRUSTEST
und ist 936 Bytes lang

das Virusfile heißt .info
und ist 7840 Bytes lang

das 1 Byte lange Zähler-File
heißt df0:VIRUSTEST.DATA

das 1 Byte lange Zähler-File
heißt df0:pic.xx

Wenn df0:VIRUSTEST.DATA nicht gelesen
werden kann, erscheint folgende Meldung:
Access Failure File may not exist.

Bei Nichtvorhandensein
von df0:pic.xx stürzt der
TimeBomb V0.9 ab

Wenn der File-Zähler bei 0 angelangt ist,
wird 100% der Diskette formatiert

Bei Erreichen des Wertes 0
wird 'nur' 95% formatiert

Nach dem Formatieren erscheint Nach dem Formatieren erscheint
Sorry Looser, that's all you're gonna get! Hey Looser ! I hate you !

1.386 virusblaster

VirusBlaster

Es handelt sich nicht um einen Virus. Das Programm ist nicht resetfest und infiziert auch keine neuen Disketten. Die einzige Funktion des VirusBlaster-Programms ist das primitive Zerstören von Disketten-Daten. Wenn man das 9232 Byte lange Programm aufruft, dann wird folgende Meldung ausgegeben:

VirusBlaster V2.3 © by M&T 7/91
Untersuche DF0: auf Boot- und LinkViren...

Sollte die Diskette schreibgeschützt sein, dann wird
Diskette ist schreibgeschützt !! ausgegeben und Ende.

Damit soll der Anschein eines nützlichen Antivirusprogrammes erweckt werden. Man kann also von einem Trojanischen Pferd sprechen. Aber anstatt auf Viren zu prüfen wird sofort angefangen, willkürlich Diskettenblöcke zu überschreiben. VIRUS CONTROL meldet nach einiger Zeit 2 * einen Schreibzugriff auf den Bootblock. Hierbei sieht man, daß lediglich unsinnige Daten geschrieben werden. Die Diskette wird dadurch unlesbar.

Je länger man den VirusBlaster laufen läßt, umso mehr Daten gehen verloren, da sie mit unsinnigen Daten überschrieben werden.

Das VirusBlaster ist ein sehr plumpes und nur auf Diskettendatenzerstörung ausgelegtes Programm. Wahrlich keine Programmiermeisterleistung. Da sich das Programm nicht automatisch verbreiten kann, werden Sie wohl erfreulicherweise niemals diesem Programm begegnen.

Das VirusBlaster-Programm ähnelt sehr stark dem Freedom -Programm. Hauptunterschied ist die deutsche anstatt der englischen Startmeldung.

1.387 unnamed.1

VirusBlaster

Es handelt sich nicht um einen Virus. Das Programm ist nicht resetfest und infiziert auch keine neuen Disketten. Die einzige Funktion des VirusBlaster-Programms ist das primitive Zerstören von Disketten-Daten. Wenn man das 9232 Byte lange Programm aufruft, dann wird folgende Meldung ausgegeben:

VirusBlaster V2.3 © by M&T 7/91
Untersuche DF0: auf Boot- und LinkViren...

Sollte die Diskette schreibgeschützt sein, dann wird

Diskette ist schreibgeschützt !! ausgegeben und Ende.

Damit soll der Anschein eines nützlichen Antivirusprogrammes erweckt werden. Man kann also von einem Trojanischen Pferd sprechen. Aber anstatt auf Viren zu prüfen wird sofort angefangen, willkürlich Diskettenblöcke zu überschreiben. VIRUS CONTROL meldet nach einiger Zeit 2 * einen Schreibzugriff auf den Bootblock. Hierbei sieht man, daß lediglich unsinnige Daten geschrieben werden. Die Diskette wird dadurch unlesbar.

Je länger man den VirusBlaster laufen läßt, umso mehr Daten gehen verloren, da sie mit unsinnigen Daten überschrieben werden.

Das VirusBlaster ist ein sehr plumpes und nur auf Diskettendatenzerstörung ausgelegtes Programm. Wahrlich keine Programmiermeisterleistung. Da sich das Programm nicht automatisch verbreiten kann, werden Sie wohl erfreulicherweise niemals diesem Programm begegnen.

Das VirusBlaster-Programm ähnelt sehr stark dem Freedom -Programm. Hauptunterschied ist die deutsche anstatt der englischen Startmeldung.

1.388 vmk v3.00

VMK V3.00

Es handelt sich um ein 2620 Byte langes File, welches nach dem Start versucht, das scsi.device mit unit=0 und flags=0 zu öffnen. Sollte dies gelingen wird im ersten Block des Rigiddiskblocks geprüft, ob das Byte an Position \$2b = 0 ist, wenn nein wird dieses Byte um eins erniedrigt, da dieser Wert anfänglich meist \$ff(=255) ist, wird nach 255 weiteren Programmstarts der Wert 0 erreicht, sollte man wie empfohlen, VMK in die startup-sequenmce übernehmen, wäre das zwangsläufig nach 255 Bootvorgängen der Fall. Solange noch nicht der Wert Null erreicht ist, tarnt sich das Programm durch die CLI-Ausgabe von VirusMemKill V3.00 © Chris Hames gefolgt von verschiedenen Vektorenangaben. Wenn der Wert 0 erreicht ist, werden 102400 Bytes ab Block 0 geschrieben, wodurch der Rigiddiskblock und womöglich auch noch weitere Daten überschrieben werden, die Festplatte ist nicht mehr ansprechbar.

1.389 ae-registrator

AE-Registrator

Es handelt sich um ein 656 Byte langes imploder-gepacktes File, (entpackt 664 Bytes), welches angeblich eine unregistrierte AmiExpress-Version in eine registrierte Version umwandeln soll. Entsprechende Textpassagen kann man in dem File nachlesen. Beim Start des AE-Registrator-Programmes wird versucht eine Datei namens bbs:user.data neu zu öffnen, wodurch eine eventuelle bereits existierende Datei gelöscht wird. Anschließend erfolgt zwangsläufig ein Absturz, da der Virusprogrammierer erstens aufgrund eines Programmierfehlers immer ein erfolgreiches Öffnen der Datei annimmt und dann zweitens auch noch ein nicht initialisiertes Filehandle

für einen Schreibzugriffsversuch benutzt.

1.390 amipatch v1.0a

AmiPatch V1.0a

Dieses 8288 Byte lange Programm greift auf eine eventuelle Datei BBS:User.Data und l/BBSHelp.Txt zu. Weiterhin wird eine Datei namens bbs:011011 erstellt.

1.391 dm-trash

dm-trash

In der Anleitung zu diesem 4764 Byte langen PowerPacker-gepackten Mailbox-schädigenden Programm wird fälschlicherweise behauptet, es existiere eine neue virusverseuchte DMS-Version, und um diese zu entfernen müsse man das Programm starten. Das Programm entfernt aber keineswegs eine virusverseuchte DMS-version, zumal es eine solche bisher noch gar nicht gibt, die existierende 53576 Byte lange DMSfixed-Version ist sauber, vielmehr verändert das Programm BBS:user.data, BBS:configl und BBS:user, um einen neuen User namens ZAPA einzutragen. Also nicht DMSfixed ist das schädliche Programm, sondern dm-trash.

1.392 doom

DOOM

Nach Start des 406012 Byte langen gepackten Files überschreibt es sich mit einem 32020 Byte langen File. Weiterhin wird ein neues gepacktes copy und assign File geschrieben und ein ungepacktes diskfont.library File. Diese Files verhalten sich wie die Originalfiles, zusätzlich verbiegen sie jedoch unter Berücksichtigung eines eventuellen VBR den Autointerrupt 5 (=\$74), dieser Interrupt wird z.B. beim Empfang von Daten über die serielle Schnittstelle aufgerufen, der Virus überprüft somit die seriellen Daten.

1.393 door_bells

DOOR_BELLS

Dieses 15308 Byte lange Programm verlangt nach einer rexxplslib.library und soll SYS: und dh0: mittels FORMAT QUICK formatieren.

1.394 dopusrt

DOpusrt

Wenn man dieses 6408 Byte lange File startet, dann wird auf das Vorhandensein von bbs:user.data und bbs:user.keys geprüft. Wenn diese Dateien vorhanden sind, dann wird an deren Ende ein User-Eintrag vorgenommen, wodurch nun also eine Hintertür zur Benutzung der Mailbox geschaffen wurde.

1.395 easy-e

EASY-E

Es handelt sich um ein 38860 Byte langes gepacktes Intro, bei welchem an den imploder-decrunchheader von Hand ein 444 Byte langer Viruscode angehängt wurde, welcher Schreib/Lesezugriffe auf dh0:bbs/user.data versucht.

1.396 lhacheck 1.1

LHACHECK 1.1

Es handelt sich um ein 3836 Byte langes Programm, welches lha-Archive auf Korrektheit prüfen soll, in Wirklichkeit aber als Einbruchwerkzeug in Mailboxsysteme gedacht sein soll, so werden z.B. bei Vorhandensein von snoopdos keine weiteren Aktionen unternommen.

1.397 look-bbs

LOOK-BBS

Es handelt sich um ein 1392 Byte langes mit Turbosqueezer gepacktes File, entpackt 1456 Bytes, welches SnoopDos abzuschalten versucht. Weiterhin werden unter Umständen Zugriffe auf AUX: und BBS:USER.DATA versucht, CLI-Ausgaben getätigt, COOL-Vektor auf Endlosfarbroutine verbogen, Alert ausgegeben, soll wohl als Einbruchswerkzeug in Mailboxen gedacht sein.

1.398 m_chat v2.3

M_CHAT V2.3

Es handelt sich um ein 13492 Byte langes Programm, welches nach dem Start überprüft, ob ein aktives AmiExpress-System installiert ist, wenn nicht erfolgt das Programmende unter Ausgabe von:

 Couldn't create reply port

Das Programm soll laut beiliegendem Doc-File MultiChat ermöglichen,

in Wirklichkeit wird aber bei einem aktiven AmiExpress-System mittels Execute()-Aufruf eine Schnellformatierung von dh0: system2.0: work: dh1: bbs: df0: df1: dh2: dh3: dh4: df2: HD: df0: versucht, wobei als Volumenname HAAAAHA verwandt wird. Nach erfolgreichem Schnellformatieren erfolgt noch folgende Ausgabe:

Sorry, the BBS is not registred

Das Programm schädigt also zielgerichtet nur AmiExpress-Systeme. Mit disksalv oder anderen Datenrettungsprogrammen sollten die meisten Daten wiederherstellbar sein.

1.399 modemspeederv2.1

ModemSpeederV2.1

Nach dem Start dieses 12492 Byte langen Programms wird eine eventuelle libs:xprcheatmodem.library mit folgendem Text überschrieben:

AGAINST BULLSHIT !!!
Sei auch Du fuer ehrliches DFUe-Betreiben !!!

1.400 mongo

Mongo

Mongo09 (3368) und Mongo51 (2260) sind Programme, die Mailboxrechner infiltrieren sollen. Sollte snoopdos aktiv sein, unternehmen die Programme nichts, um sich nicht zu verraten.

1.401 noguruv2.0

NoGuruV2.0

Dieses 1124 Byte lange Programm verbiegt nach dem Start den Alert()- und Autorequest()-Vektor. Es wird folgender Text im CLI ausgegeben:

NO-GURU V2.0 Installed!

Any guru's OR alerts will now cause a reset instead of locking your Amiga!
AmigaDOS requesters are now canceled
Copyright (C) 1991, PSEUDO-OPS

This program is NOT public domain, and may only be used with the permission
of a PSEUDO-OPS member!!!

'Making life with AmiExpress just that little bit easier...'

Es wird versucht die Datei bss:user.data einzuladen.
In dieser Datei wird dann gezielt nach den Texten renegade,

jock rockwell und spiral gesucht. Sollten solche Texte gefunden werden, dann werden diese verändert. Danach wird die Datei wieder zurückgeschrieben. Das Programm soll also gezielt einige wenige AmiExpress-Benutzer schädigen.

1.402 showsysops

showsysops

Dieser 7860 Byte lange Programm greift auf BBS:USER.DATA zu.

1.403 swiftware-devildoor8

SwiftWare-DevilDoor8

Es handelt sich um ein 44224 Byte langes File, bei welchem von Hand ein neuer Hunk vor das ursprüngliche Programm gelinkt wurde, um somit das Sysop-Passwort ausspähen und abzuspeichern zu können.

Hierzu erfolgen Zugriffe auf bbs:user.data und bbs:node0/nocallersat300. Diese Dateinamen liegen kodiert vor.

Diese Mailboxeinbruchsprogramme nach Door-Machart hängen Ihre Virusroutine sehr unsauber an das Programm an, so daß solche Programme nur noch unter Kickstart 1.2/1.3 geladen werden können. Selbst ein loadseg()-Ladeversuch eines solchen Door-Programmes aus einem Monitor heraus führt unter Kickstart 2.0 wegen dem fehlerhaften Hunkaufbau zum Absturz.

1.404 sysinfov2.2

SysinfoV2.2

das mit PowerPacker geapckte 3928 Byte lange Programm, ungepackt 5656 Bytes, löscht womöglich Dateien auf einem eventuellen Datenträger namens BBS:, denn in dem Programm ist folgender Text zu finden: delete BBS:#? all Allerdings ist das Programm aufgrund eines fehlerhaften Hunkaufbaus oftmals nicht zu laden bzw. stürzt recht schnell ab.

1.405 timer

timer

Wenn man das 4812 Byte lange timer-Programm startet, dann erscheint rechts unten auf der Workbench ein kleines Fenster mit dem Titel V1.1, danach wird das eigentliche 1712 Byte lange Virusfile nach :c/SetMap und :system/SetMap kopiert. Nachdem der Virus installiert wurde, werden zur Vervollständigung der Tarnung im dem Fenster die aktuellen Speicher- und Zeitwerte angezeigt.

```

RAM .....
Chip .....
-----
Time .....
Date .....

```

Beim nächsten Booten wird dann durch den üblicherweise in der startup-sequence vorkommenden 'setmap'-Befehl das Virusprogramm gestartet. Es wird dann zunächst wie erwartet die gewünschte Tastaturbelegung geladen, danach erfolgt allerdings als Virusaktion das Verbiegen des Autointerrupt 5 (\$74), über den der Interrupt läuft, wenn der Eingabepuffer des seriellen Ports voll ist. Durch Verbiegen dieses Vektors kann der Virus also die eintreffenden bzw. zurückgeechoten Daten (Passwörter usw.) kontrollieren. Weiterhin kann der Virus beim Empfangen einer bestimmten Zeichenfolge gewisse Aktionen auslösen, das heißt, angenommen man hat einer Mailbox diesen Virus untergeschoben, dann kann man von außen diese Mailbox kontrollieren.

1.406 top util v1.0

Top util V1.0

Es handelt sich um ein 2260 Byte langes schädliches Programm, welches beim Start folgenden Text im CLI ausgibt:

```

      Top util By Zacker of EnSoniC V1.0
      Call Zack BBS 16.8 HST 407-232-6324 HST ONLY !!

```

```

USEAGE: Top (num /ALL) <-Hfname> <-Sfname> <-Ttext>
num : Min. megs to get on the list
ALL : Show all users (default to BBS:User.rpt)
-H  : Use the file name after -H as the TOP hdr
      if no -s is used default = BBS:user.rpt
-T  : text to be used with top dog
      : TOPDOG (I just added)
-S  : Use the file name after -S as TOP list

```

Wenn man beim Aufruf des Programmes irgendeine Parameterangabe macht, dann gibt der Virus in der Shell folgende Zeile aus:

```

..... WORKING !!!!

```

und versucht anschließend eine eventuelle Datei namens bbs:user.data mit 66 Bytes zu überschreiben, worin u.a. folgender Text enthalten ist:

```

The Three Musketeers

```

1.407 trojan killer v3.0

TROJAN KILLER V3.0

```

-----

```

Es handelt sich um ein 10536 Byte langes schädliches Programm, welches beim Start die Dateien ConDevice und StrInOut zu löschen versucht und dann folgenden Text im CLI ausgibt:

```
TROJAN KILLER V3.0   (23/8/92)
~~~~~
Please enter the full path U have to your download dir
< eg. BBS:warez/upload >

The directory?:
```

Anschließend werden dann BBS:user.data und BBS:user.keys in das von Ihnen angegebene Verzeichnis unter den Namen demo98.lha und demo99.lha kopiert.

Es werden noch folgende Meldungen im CLI ausgegeben.

```
Report: 0 trojan(s) found'
Checking for known trojans in memory...!
Trojan(s) found on harddisk      : 0!
Trojan(s) found in memory       : 0

Please contact ->NYLONMAN<- for new trojan killers!!!

Note: This program only works on AmiExpress 1.xx and 2.xx

Press <<ENTER>>
```

1.408 xpr-speederv3.2

XPR-SpeederV3.2

Es handelt sich um ein 9556 Byte langes File, welches Z-Modem-Übertragungen beschleunigen soll. Angenommen jemand betreibt eine Mailbox und ruft nun dieses Programm dauerhaft auf, weil er den Versprechungen glaubt, die in dem beliebigen Dokumentationsfile gemacht werden. Im shell-Fenster erscheint dann zunächst folgende eher beruhigende Meldung:

```
XPR-Speeder V3.2 - ACTIVE -
```

Die böse Überraschung kommt dann nachts um 04:13, denn auf diese 'unmenschliche' Uhrzeit wartet der Virus, um dann aktiv zu werden. Er setzt dann über die dos.library-execute-Funktion folgenden CLI-Befehl ab

```
delete bbs:#? all
```

Damit sich der Virus nicht vorzeitig verrät, werden die delete-Meldungen in die Datei ram:temp umgeleitet.

Nachdem der delete-Aufruf beendet ist, schreibt der Virus nun endlos immer 495616 Müll-Bytes in eine Datei namen bbs:Dip_in_DUDE, welche somit also immer größer wird, und dadurch ein Rekonstruieren der gelöschten Dateien mit z.B. disksalv immer unmöglicher macht. Irgendwann weist das Betriebssystem mittels Systemrequester darauf hin, daß die Diskette oder Festplattepartition voll geworden ist, der Virus versucht aber dennoch

weiterhin endlos Daten in die bbs:Dip_in_DUDE-Datei zu schreiben.

Gegen Ende des XPR-SpeederV3.2-File kann man u.a. folgenden Text lesen:

```
XPR-Speeder V3.2 - ACTIVE -  
04:13  
BBS  
delete bbs:#? all  
bbs:Dip_in_DUDE  
ram:temp  
dos.library
```

1.409 Disk-Validatorviren

Disk-Validatorviren

```
-----  
Orange-Disk-Validatorvirus(=DiskVal1234)  
ReturnOfTheLamerExtreminator-Disk-Validatorvirus  
SADDAM-HUSSEIN-Disk-Validatorvirus und Abkömmlinge
```

Das große Problem an den Disk-Validatorviren ist, daß sie sofort beim Einlegen der infizierten Diskette aktiviert werden, denn diese Disketten weisen einen absichtlichen Fehler auf, wodurch dann beim Disketten-Einlegen automatisch vom Betriebssystem das l/disk-validator-Programm gestartet wird. Dieses Programm ist aber das Virusprogramm. Im Falle des ReturnOfTheLamer-Disk-Validatorvirus kann dieses Programm meistens wegen einem '202 Object in use error' nicht gelöscht werden, und bei dem SADDAM-HUSSEIN-Disk-Validatorvirus werden oftmals viele Datenblöcke auf der Diskette kodiert, die nur bei aktivem Virus wieder fehlerfrei gelesen werden können. Es ist also sehr schwer, diese äußerst hartnäckigen Viren wieder zu entfernen. Es empfiehlt sich daher folgende Vorgehensweise: Man kopiert alle Dateien von der Diskette auf Festplatte oder vd0:. Hierbei korrigiert der SADDAM-HUSSEIN-Virus bei Bedarf die Dateien selber. Sie brauchen auch keine Angst um Ihre Festplatte zu haben, da die bisherigen Disk-Validatorviren nur Disketten schädigen. Nachdem Sie die Dateien auf die Festplatte kopiert haben, schalten Sie den Rechner aus, und booten danach von Festplatte oder von einer garantiert sauberen Diskette. Wenn Sie keine Festplatte besitzen können Sie auch die resetfeste RAM-Disk vd0: von z.B. Fish Disk 58 benutzen. Nachdem Sie die Dateien nach vd0: kopiert haben, müssen Sie einen 'Kill-Virus-Versuch mit Reset' oder einen 'Full-Reset' auslösen. Booten Sie anschließend von einer garantiert virenfreien Diskette. RAD: kann hierzu nicht benutzt werden, da sie den Kill-Reset nicht überlebt. Die infizierten Disketten sollten Sie baldmöglichst formatieren oder eine andere Disk daraufkopieren, allerdings nicht mit den normalen DOS-Befehlen, da hierbei beim Disketten-Einlegen der Virus wieder aktiviert würde. Unter Kickstart 2.0 stellen die Disk-Validatorviren keine Gefahr mehr dar. Angenommen man legt unter Kickstart 2.0 eine mit einem Disk-Validatorvirus infizierte Diskette ein, dann wird nicht mehr das Disk-Validatorvirusfile von der Diskette geladen, sondern es wird der nun bereits im ROM vorhandene Disk-Validator aufgerufen. Dieser korrigiert dann die Diskette, so daß diese Diskette nun auch unter Kickstart 1.3 ohne Aufruf des auf der Diskette vorhandenen Disk-Validatorvirusfiles eingelegt werden kann. Eventuell vom SADDAM-HUSSEIN-Virus verschlüsselte Daten können so aber leider nicht wiederhergestellt werden. Das Disk-Validatorvirusfile sollte man sicherheitshalber sofort löschen. Aber auch wenn Sie unter Kickstart 1.3

arbeiten, brauchen Sie sich keine Sorgen mehr bezüglich der Disk-Validatorviren zu machen, vorausgesetzt, Sie benutzen VIRUS CONTROL, denn VIRUS CONTROL verhindert erstens das automatische Aktivieren der Disk-Validatorviren und zweitens rekonstruiert VIRUS CONTROL alle Daten auf Disketten, welche mit dem SADDAM-HUSSEIN-Virus oder Abkömmlingen infiziert waren. Als drittes wird natürlich auch noch das Disk-Validatorvirusfile selber gelöscht.

1.410 orange-disk-validatorvirus(=diskval1234)

Orange-Disk-Validatorvirus(=DiskVal1234)

Dieser Disk-Validatorvirus basiert zwar auch auf dem SADDAM-HUSSEIN-Disk-Validatorvirus, ist aber viel bösartiger programmiert, da er bei praktisch jedem Disk-Zugriff Daten unwiderbringlich überschreibt. In zufälligen Datenblöcken wird ab Position 90 1*\$1234 (deshalb auch der Name DiskVal1234) und ab Position 100 bis 232 \$4e71 geschrieben. Dadurch werden Dateien zerstört und es kommt andauernd zum Absturz. Bei einem solchen Absturz kann sich der Bildschirm fleischfarben färben, daher der Name. Der Original-Saddam-Hussein verschlüsselte die Daten unter Benutzung eines typischen Langwortes wie z.B. 'IRAK', man konnte die Daten also meist wieder retten, der Orange-Disk-Validatorvirus aber nimmt keine Verschlüsselung der Daten unter Benutzung eines typischen Langwortes vor, sondern überschreibt die Daten unwiderbringlich, eine Rettung der Daten ist nicht möglich. Im Gegensatz zum Saddam-Hussein ist der Orange nicht kodiert. Gegen Ende des Files kann man folgenden Text lesen:
DF1:l DF1:l/Disk-Validator strap mycon write intuition.library
trackdisk.device

1.411 returnofthelamerexterminator-disk-validatorvirus

ReturnOfTheLamerExterminator-Disk-Validatorvirus

Der ReturnOfTheLamerExterminator macht sich Kick-kompatibel über Kick-Tag und Kick-Checksum resetfest. Weiterhin wird der erste Eintrag des Vertikal-Blank-Server verbogen, wodurch die Kick-Resetfestigkeit sichergestellt wird. Ferner wird auch der BeginIO()-Vektor und Close()-Vektor des trackdisk.device verbogen, und auch der BeginIO()-Vektor des keyboard.devices wird verbogen. Beim Reset wird kurzzeitig der OpenWindow()-Vektor verbogen. Der Virus verbreitet sich dadurch, daß er anstelle des Original-Disk-Validatorfiles das Virus-Disk-Validatorfile auf die Diskette schreibt und die Diskette durch Löschen des BitMap-Flags fehlerhaft macht. Leider zerstört der ReturnOfTheLamerExterminator auch unwiderbringlich Daten, indem zufallsgesteuert Daten-Blöcke auf der Disk mit LAMER!!! überschrieben werden. Weiterhin kann der Virus auch alle Disketten schnell-formatieren. Der Virus ist allerdings noch recht fehlerhaft programmiert, so daß es in der Regel zu keiner Neu-Infektion von Disketten kommt, vielmehr erfolgt oftmals ein Absturz beim Einlegen von Disketten. Im Gegensatz zum Original-Commodore-Disk-validatorfile kann man wegen Kodierung des Programmcodes keinerlei lesbaren Text ausmachen.

1.412 saddam-hussein-disk-validatorvirus und abkömmlinge

SADDAM-HUSSEIN-Disk-Validatorvirus und Abkömmlinge

Dieser Virus ist mit dem ReturnOfTheLamerExterminator-Disk-Validatorvirus verwandt. Allerdings ist der SADDAM-HUSSEIN-Disk-Validatorvirus deutlich funktionsfähiger und aggressiver programmiert. Es bestehen insbesondere folgende Unterschiede: Der SADDAM-HUSSEIN-Virus macht sich anstatt über die Kick-Vektoren über den COLD-Vektor resistent, wodurch er auch auf 1 MB-Chip-RAM-Amigas resistent ist. Der SADDAM-HUSSEIN ist der erste Virus, der diese Möglichkeit nutzt.

Der ReturnOfTheLamerExterminator-Disk-Validatorvirus überschreibt manchmal zufallsgesteuert Daten-Blöcke auf der Diskette mit 'LAMER!!!'. Der SADDAM-HUSSEIN-Disk-Validatorvirus überschreibt nur die Daten-Block-Kennung mit 'IRAK' und kodiert den restlichen DatenBlock-Inhalt. Bei nicht aktivem SADDAM-HUSSEIN-Virus kann man also die betroffenen Dateien nicht mehr lesen. Bei aktivem Virus sind die Dateien wieder lesbar, da der Virus die notwendigen Änderungen automatisch beim Lesezugriff vornimmt.

Manchmal versucht der Virus, Disketten schnellzuformatieren.

In dem Original-Commodore-Disk-Validator-File sind mehrere Textpassagen enthalten, gegen Ende des Files kann man z.B. bad header oder seltener bad extension lesen. Im Gegensatz zum Original-Commodore-Disk-Validator-File ist der Großteil des Saddam-Hussein-Files kodiert, lediglich gegen Ende kann etwas Text erkannt werden. Es gibt nun mittlerweile sehr viele Saddam-Hussein-Disk-Validatorviren-Abkömmlinge, welche anstatt 'IRAK' ein anderes Langwort zur Datenverschlüsselung und meist auch den lesbaren Datei-Schlußtext etwas abändern. Rein programmtechnisch sind die Viren identisch, lediglich der SADDAM][besitzt eine geringfügig abgeänderte Dekodieroutine, was aber auch nichts am Grundprinzip ändert. Das Antivirusprogramm VIRUS CONTROL läßt sich durch diese formalen Änderungen nicht irritieren und wird auch eventuelle zukünftige Saddam-Hussein-Abkömmlinge sicher identifizieren und entfernen können.

SADDAM-HUSSEIN-Disk-Validatorvirus und Abkömmlinge

Kodier-LW	lesbarer Text am Dateiende	erst nach Dekodierung lesbar
'IRAK'	BitMap Checksum Error	SADDAM VIRUS
'LAME'	BitMap Checksum Error	SADDAM VIRUS
'LOOM'	BitMap Checksum Error	LOOOOM VIRUS
'IRAK'	BitMap Checksum Error	SADDAM] [
'KICK'	Use Kickstart 1.2/1.3	KICK VIRUS
'NATO'	Greatest Human Error	NATO VIRUS
'AFFE'	!gnihton yas,raeh,eeS -> See,hear,say nothing!	Gorila Virus
'GRAL'	! lleh eht ot yaw eht -> the way to the hell !	HolyGralViri

'IRAN'	nac uoy tahw lla llik -> kill all what you can	Ayatollahvir
'RISC'	kein lesbarer Text	RISC VIRUS
'4711'	kein lesbarer Text	Parfum Virus
'666', \$A0	kein lesbarer Text	Animal Virus
\$A0A0A0A0	kein lesbarer Text	Laurin Virus
'HARD'	kein lesbarer Text	HARDEX VIRUS
Virus nicht lauffähig, da trackdisk.device-string gelöscht, wodurch sich der Virus nicht mehr in das Sytem einklinken kann		

nicht ladbarer Disk-Validatorvirus

Es existiert auch noch ein 14524 Byte langes File, an dessen Anfang der Saddam-Hussein-Disk-Validatorvirus steht. Da aber das File in seiner Gesamtheit nicht korrekt aufgebaut ist, verweigert das Amiga-Betriebssystem auch unter Kickstart 1.2/1.3 das Laden. Der Virus kann also nicht aktiv werden, wodurch das Programm ungefährlich ist.

1.413 Linkviren

Linkviren

IRQ-Typ (Der Virus hängt einen neuen Codehunk vor das Originalprogramm)

Antichrist
BURN
CHRISTMAS VIOLATOR
CCCP-Bootblock+Linkvirus
Hochofen(=Trabbi)
IRQ-Linkvirus I+II
MegaLINK
MENEM'S REVENGE
METAMORPHOSISV1.0-Bootblock+Linkvirus
QRDL
THE SMILY CANCER I von CENTURIONS
The Traveling Jack - Linkvirus I+II
Viewtek

Xeno-Typ (Der Virus schreibt sich an den Anfang des ersten Codehunks)

BESTIAL DEVASTATION
New Age
Xeno-Virus I+II

LZ-Typ (Der Virus hängt sich an das Ende des ersten Codehunks)

Der Virus verändert nur das Ende des ersten Codehunks

Crime!
Crime!++
Crime'92
FileGhost
FileGhost2
GoldenRider
LZ-Linkvirus

Der Virus verändert Anfang und Ende des ersten Codehunks

COMMANDER
DarkAvenger
Infiltrator
Polyzygotronifikator

Red-Typ (Viruscode + Originalprogramm zu einem Codehunk zusammenfassen)

Red October V1.7

DEBUGGER-Typ (Virus in Codehunkanfang und in Debughunk am Fileende)

DEBUGGER

1.414 antichrist

Antichrist

Es bestehen lediglich folgende Unterschiede gegenüber dem
The Traveling Jack - Linkvirus I+II , der Viruscode wird nicht kodiert
und die zufällig geschriebene Textdatei ist nur 26 Byte lang
und beinhaltet nun folgenden Text:

The Antichrist is back

Der Name dieser Textdatei lautet nun Antichrist.X,
wobei X meist ein zufälliger Buchstabe ist.

1.415 burn

BURN

Der BURN-Virus ist sehr stark mit dem MODEMCHECK-FUCK-Virus und
scan.x -Virus verwandt, denn einerseits weist der BURN-Virus die gleiche
Festplattendatenzerstörungsroutine wie der scan.x-Virus auf und andererseits
hat der BURN-Virus mit dem MODEMCHECK-FUCK-Virus die Abkoppelung eines
speziellen Virusprozesses gemeinsam.
BURN, MODEMCHECK-FUCK und scan.x sind sogenannte Festplattenviren, die
gezielt die Daten auf großen Speichermedien überschreiben. Der BURN-Virus

stellt die bislang neueste Entwicklung dar, da hier zu dem datenzerstörenden Verhalten noch eine Linkvirusfunktion hinzugekommen ist.

Der BURN-Virus stürzt unter Kickstart 1.2/1.3 ab, weil in dem Fall irrtümlich als dosbase der Wert 0 verwandt wird.

Der BURN-Virus hängt beim Infizieren eines Programmes einen 204 Byte langen Virushunk vor das Programm und einen 2172 Byte langen Virushunk an das Ende des Programmes. Die Programme werden dadurch um 2412 Bytes verlängert (es werden noch 36 weitere Verwaltungsbytes benötigt). Mehrfachinfektionen sind möglich, ebenso können gleichzeitig mehrere Virusprozesse aktiv sein. Auch beim Infizieren von Programmen legt der BURN-Virus ein festplatten-spezifisches Verhalten an den Tag, den Programme auf DD- oder HD-Disketten werden nicht infiziert, da der Virus nur auf Datenträgern mit mehr als 3520 Blöcken Infizierungen vornimmt. Hierbei wird mittels ExNext() zufällig unterschiedlich tief in C: ein Programm gesucht, später werden auch weitere Devicenamen bearbeitet. Während sich der BURN-Virus an ein Programm hängt, verbiegt er zuvor mittels SetFunction() den Write-Vektor. Damit das Infizieren, also das Beschreiben der Datei auch möglichst gelingt, setzt der Virus zuvor die protectionbits auf rwed. Beim Start eines mit dem BURN-Virus infizierten Programmes wird zuerst der kurze 204 Byte lange Virushunk ausgeführt. Hierbei wird die Adresse des an das Programm angehängten 2172 Byte langen Virushunks gesucht, desweiteren wird aus der TaskWaitList ein zufälliger Prozessname ausgewählt und auch der entsprechende Stack-Wert übernommen, mit diesen Werten wird dann mit CreateProc() der 2172 Byte lange Virushunk als eigenständiger Virusprozess gestartet. Der Virus macht sich nicht resetfest. Vor dem 6.2.1994 versucht der Virus nur Programme zu infizieren. Am 6.2.1994 und dann immer jeweils alle 16 Tage wird jedoch eine Datenzerstörungsroutine ausgeführt. Während der anderen 15 Tage beschränkt sich der Virus auf Infizierungsversuche. Der Virus legt zwischen den einzelnen Infizierungsversuchen eine Pause mit einer zufälligen Dauer von 0 bis circa 82 Sekunden ein. Die Datenzerstörungsroutine ist weitgehend mit der Routine aus bossnukel.5/scan.x identisch, das heißt es werden bei Festplatten und ähnlich großen Speichermedien die Daten und auch der RigidDiskBlock überschrieben. Disketten bleiben verschont.

1.416 christmas violator

CHRISTMAS VIOLATOR

Der Programmierer dieses Linkvirus hat sich an dem IRQ-Linkvirus orientiert. Während sich der IRQ über die Kick-Vektoren resetfest macht, so macht sich der CHRISTMAS nun über den COOL-Vektor resetfest. Zum Infizieren von startbaren Dateien verbiegen beide Viren den OldOpenLibrary()-Vektor. Die eigentlichen Linkroutinen hat der CHRISTMAS weitgehend von dem IRQ übernommen. Dennoch liegen Welten sich dem IRQ und CHRISTMAS Virus, denn während man dem IRQ-Linkvirus eine recht elegante Programmierung zugestehen muß, so zeichnet sich der CHRISTMAS durch eine sehr unsaubere Programmierung aus, wodurch der Virus normalerweise gar nicht lauffähig ist, so ist z.B. unbedingt Kickstart 1.3 und Ranger-RAM ab \$C00000 erforderlich. In dem Virus ist folgender Text kodiert versteckt:

```
>>> CHRISTMAS VIOLATOR by the Dream Team - (HE HE)<<< Have a nice day...
```

1.417 cccp-bootblock+linkvirus

CCCP-Bootblock+Linkvirus

Es handelt sich um den ersten Virus, welcher sowohl ein Bootblockvirus als auch ein Linkvirus ist. Der Name CCCP VIRUS rührt daher, daß man diesen Text am Anfang des Virus lesen kann. Der Virus verbiegt den Autointerrupt 3 (\$6c). Dadurch wird nun 50 * pro Sekunde der KickTag-Pointer und der COLD-Vektor gelöscht und der COOL-Vektor auf den CCCP-Virus gesetzt. Dadurch ist nun der CCCP-Virus das einzige resetfeste Programm. Bei einem Reset wird der DoIO()-Vektor verbogen, wodurch nun die Diskinfektionen möglich werden. Als erstes schreibt sich der CCCP-Virus auf den Bootblock. Als zweites versucht sich der Virus dann an ein Programm zu linken. Pro Booten wird maximal ein Programm infiziert. In der Regel sind nur Disketten gefährdet.

1.418 hochofen(=trabbi)

Hochofen(=Trabbi)

Es handelt sich um einen Linkvirus, welcher nur beim Start eines bereits infizierten Programms weitere Programme infizieren kann. Der Virus selber installiert nur einen Task namens 'Greetings to Hochofen' im Speicher, der nach einiger Zeit durch direkte Copper-Programmierung eine Deutschlandflagge ausgibt. Wiederum nach einiger Zeit erscheint ein Auto-Requester mit dem Text 'Fasten seat-belts'. Dannach beendet sich der Task. Der Virus hängt beim Infizieren einen Virus-Code-Hunk vor die Original-ProgrammeFiles, wodurch die Programme um 3000 Bytes verlängert werden. Da der Virus nicht alle Hunk-Typen korrekt verarbeitet, stürzt der Virus beim Infektionsversuch mancher Programme ab (z.B. cmon) oder aber die infizierten Programme sind aufgrund von Beschädigungen erst gar nicht ladbar (tnt) oder stürzen nach dem Start ab. Meist funktioniert aber die Infektion und die infizierten Programme arbeiten korrekt. Aus beschädigten Programmen kann natürlich nicht mehr das Original-File rekonstruiert werden.

1.419 irq-linkvirus i+ii

IRQ-Linkvirus I+II

Der IRQ-Linkvirus macht sich über die Kick-Vektoren resetfest. Er verbiegt den OldOpenLibrary()-Vektor. Wenn nun ein Programm diese Funktion aufruft, so wird zufallsbedingt entweder :c/dir oder das erste Programm aus :s/Startup-Sequence, evetuell mit vorangestelltem c: infiziert. Hierbei wird das Programm um 1096 Bytes verlängert. Weiterhin wird zufallsbedingt die aktive Windowtitelleiste in

AmigaDOS presents:a new virus by the IRQ-TeamV41.0

geändert. Der IRQ-Virus ist äußerst interessant, da er der erste Amiga-Linkvirus ist. Trotz sehr guter Programmierung, bin ich auf folgende Programmierfehler gestoßen:

- nach Resident-Strukturen wird nur in \$0-\$200000 und \$c00000-\$dc0000

gesucht. Sollte jemand echtes Fast-RAM(z.B.\$200000-\$400000) besitzen, so könnte die Resident-Struktur hier angelegt werden, da IRQ keinen speziellen Speicher anfordert, dies ist der Grund, warum IRQ nicht auf jedem Amiga resetfest ist.

- Programme, die sich über KickTagPointer resetfest gemacht haben, werden abgehängt, da sich der Virus allein in KickTagPointer einbindet, ein eventueller RAD:-Inhalt geht somit verloren.
- Der IRQ-Virus läuft nur aufgrund eines Denkfehlers nicht mit Kickstart 1.3. Der IRQ-Virus kopiert den Viruscode in einen angeforderten Speicherbereich, da ja nach dem Programmende der Speicher automatisch wieder freigegeben wird. Nach dem Kopieren verbiegt der IRQ-Virus den OldOpenLibrary()-Vektor korrekt auf den neuen Speicherbereich, der Original-Vektor wird aber im aktuellen Speicherbereich abgelegt, welcher aber ja gleich freigegeben wird. Im aktuellen Speicher steht somit auch bei Kickstart 1.3 der Kickstart 1.2-Wert, da dieser nicht mit dem 1.3-Wert überschrieben wird.

Der IRQ-Virus ist nicht direkt bösartig, da er sich meist korrekt an die Programme anbindet. Allerdings ist die Hunk-auswertung nicht 100% korrekt, so daß durchaus Probleme denkbar sind. Sie sollten den IRQ-Virus eliminieren, da er zumindest lästig ist.

Es gibt eine Abart des IRQ-LinkVirus, welche mit Absicht nicht prüft, ob das File schon infiziert ist. Dadurch kann es passieren, daß Sie ein z.B. dir-File haben, an welches sich z.B. schon 7 * der IRQ-Virus gelinkt hat. Diese Programme sind dennoch meist voll funktionsfähig.

1.420 megalink

MegaLINK

Der Virus hängt sich als ein neuer Codehunk vor das Originalprogramm, wodurch dieses um 1044 Bytes verlängert wird. Diese Linkroutine wurde von dem CCCP -Bootblock+Linkvirus übernommen. Ansonsten sind die beiden Viren aber verschieden. Der MegaLINK macht sich nicht resetfest und kann somit nur neue Files infizieren, wenn ein bereits infiziertes File aufgerufen wird, pro Aufruf wird maximal ein File infiziert. Files größer 15000 Bytes werden nie infiziert. Der MegaLINK durchsucht zuerst das sys:-Verzeichnis und dann bei Bedarf noch das sys:c-Verzeichnis. Bereits einmal infizierte Files werden nicht nochmal infiziert, sondern es wird mittels exnext() weiter nach noch nicht infizierten Files gesucht. Die Protectionbits der Files werden vor dem Infektionsversuch immer auf RWED gesetzt. Der MegaLINK schaltet auch die Systemrequester aus, um unauffällig agieren zu können. Gegen Ende des Viruscodes liegt folgender kodierter Text vor:

```
sys: sys:c dos.library MegaLINK
```

1.421 menem's revenge

MENEM'S REVENGE

Es handelt um einen Linkvirus, welcher die infizierten Programme um 3076 Bytes verlängert. Hierbei wird ein Code-Hunk und Daten-Hunk vor das Originalprogramm gehängt. Der Virus verbiegt den loadseg()-Vektor. Über diesen Vektor werden z.B. die Programme geladen, wenn man sie über die Workbench startet. Der Virus infiziert immer das vorhergehende Programm. Das gerade aktuell gestartete Programm wird also erst dann infiziert, wenn ein neues Programm gestartet wird. Das es nicht einfach ist, einen Linkvirus zu programmieren, beweist der MENEM-Virus. Nur wenige der infizierten Programme sind lauffähig. Meistens endet der Start von infizierten Programmen in einem Guru. Der Grund warum die meisten infizierten Programme abstürzen, liegt darin, daß der MENEM-Virus die Programme beim Infizieren meist unwiderbringlich beschädigt. Es ist dann also auch nicht mehr möglich das Original-Programm zu restaurieren. Insbesondere, wenn in einem Programm viele Hunks auftreten, muß der MENEM-Virus passen und kreierte kaputtene Files, welche natürlich auch nicht mehr wiederhergestellt werden können. Der Virus kann sich mit folgendem Alert melden:

```
MENEM'S REVENGE HAS ARRIVED !!!  
ARGENTINA STILL ALIVE
```

1.422 metamorphosisv1.0-bootblock+linkvirus

METAMORPHOSISV1.0-Bootblock+Linkvirus

Es handelt sich wie bei dem CCCP-Virus um einen kombinierten Bootblock- und Linkvirus. Wenn der Virus-Bootblock durch Booten oder ein infiziertes Programm gestartet wird, dann installiert sich Virus im Speicher, indem der OldOpenLibrary()-Vektor verändert wird, und wenn nun irgendein Programm diesen Vektor aufruft, dann versucht der Virus ein Programm in c: zu infizieren, indem der Virus-Code nach IRQ-Virusart als eigener Hunk vor das Programm gehängt wird. Die Programme werden hierbei um 1060 Bytes verlängert. Bereits infizierte Programme werden nicht noch einmal infiziert. Der Virus macht sich über den COLD-Vektor resetfest. In der Resetphase wird auch kurzzeitig der COOL-Vektor benutzt, der aber letztendlich dann gelöscht wird. Der Virus steht immer ab \$7FA72 im Speicher. Beim Versuch einen Disketten-Bootblock zu infizieren kann auch der Rigiddiskblock von Autoboot-Festplatten beschädigt werden. siehe Rigiddiskblock beschädigen . Zu Beginn des Virus-Bootblocks oder eines infizierten Programms kann man folgenden Text lesen:

```
-METAMORPHOSIS V1.0- the next Generation from LAMER-EXTERMINATOR !
```

Normalerweise wird beim Disketten-Einlegen diese mit dem Virus-Bootblock infiziert. Manchmal aber werden auch alle im Moment eingelegten Disketten schnellformatiert.

1.423 qrdl

QRDL

Es handelt sich um einen Linkvirus, der aufgrund des Drivebit-Bug und Bootdisk-Bug nur unter Kickstart 1.2/1.3 korrekt funktioniert.

Der Virus speichert sich verschlüsselt ab, im Speicher kann man jedoch folgenden dekodierten Text lesen:

(C)1992-04-16 QRDL. Release 1.1 Born in Poland, Grt to Jack

Nach dem Start eines infizierten Files hängt sich der Virus in die CIA-A-Interruptserverliste ein, das heißt der Viruscode wird bei jedem Tastendruck durchlaufen und hierbei wird dann in circa 50 % der Fälle entweder der COOL-Vektor gelöscht oder aber auf den Virus verbogen.

Während des Resets wird die Bootdiskette in DF0: infiziert, das geht folgendermaßen vor sich, in der COOL-Routine wird der DoIO()-Vektor verbogen, um dann während des Diskettenbootzugriffs sich wieder in die CIA-A-Interruptserverliste einzuhängen und um den OpenLibrary()-Vektor zu verbiegen, der DoIO()-Vektor hingegen wird wieder restauriert. Der OpenLibrary()-Vektor wird nur während des Resets kurzzeitig verbogen, und zwar um dann den OpenWindow()-Vektor zu verbiegen, wohingegen der OpenLibrary()-Vektor wiederum restauriert wird. Wenn nun das Betriebssystem das AmigaDOS-Shell-Fenster öffnen will und hierzu OpenWindow() und damit den Virus aufruft, erfolgt nun die Disketteninfektion, nachdem als erstes wieder der OpenWindow()-Vektor restauriert wird. Da während des Bootens eh sehr viele Diskettenzugriffe erfolgen, fällt zu diesem Zeitpunkt eine Infektion nicht sonderlich auf. Der Virus versucht nun DF0:s/startup-sequence nach \$70000 zu laden, um das erste File der startup-sequence zu ermitteln und zu infizieren. Dies gelingt, wenn das File mit einem vollständigen Pfad eingetragen ist oder wenn das File im c- oder Basisverzeichnis steht. Der Virus hängt dann einen neuen Hunk vor das File, wodurch das File 2320 Byte länger wird. Wenn aber keine DF0:s/startup-sequence gefunden wird, wenn der Virus also keine Infizierung vornehmen kann, dann versucht er statt dessen die BitMap der Diskette in DF0: freizugeben. Dadurch resultiert dann eine scheinbar völlig leere Diskette (used 0 Free 1758), das heißt selbst der Bootblock wurde freigegeben, die Diskette ist noch lesbar und benutzbar, aber spätestens nach einigen Abspeicherversuchen erfolgt ein Absturz, da durch das Abspeichern wichtige noch benötigte aber nun vermeintlich freie Diskettenverwaltungsstrukturen überschrieben wurden.

1.424 smilycancercenturions

THE SMILY CANCER I von CENTURIONS

Es handelt sich um einen richtigen Linkvirus, welcher sich als Code-Hunk vor das eigentliche Programm hängt. Dadurch wird das Programm 3916 Bytes länger. Es wird das erste File der Startup-Sequence infiziert. Man kann den Virus also dadurch entfernen, indem man das erste File der Startup-Sequence durch ein sauberes Originalfile ersetzt. Beim Aufruf eines solchen infizierten Programms wird der BeginIO()-Vektor des trackdisk.devices auf \$7f128 verbogen. Hierdurch kann der Virus eine soeben eingelegte Diskette infizieren. Festplatten sind nicht gefährdet! Weiterhin verbiegt der Virus die Kick-Vektoren um sich resetfest zu machen. (KickTag auf \$7FE00). Hierbei werden andere resetfeste Programme wie RAD:, ALF oder turboprint nicht herausgeworfen. Weiterhin verbiegt der SMILY CANCER den SumKickData()-Vektor, um damit eher seine Reset-Festigkeit sicherzustellen. Der SMILY CANCER belegt völlig willkürlich den Speicherbereich von \$7f000 bis \$7ff30. Auf 512 KB-Amigas liegt hier der

1.425 thetravelingjack

The Traveling Jack - Linkvirus I+II

Es handelt sich um einen völlig neuartig programmierten Linkvirus, denn der Traveling-Jack-Virus ist der erste Virus, welcher sich an die BCPL-Spezialitäten des Amiga-DOS heranwagt. Genauer gesagt, der Traveling-Jack-Virus verbiegt die Startroutine der Global-Vektor-Funktionen. Bisher gibt es noch kein Antivirusprogramm, welches an dieser Stelle auf eine Virus-Infektion prüft, denn dieser Bereich des Amiga-DOS ist recht schwer verständlich. Der Traveling-Jack-Virus ist also für alle bisherigen Antivirusprogramme unsichtbar. Der Traveling-Jack-Virus ist recht aufwendig programmiert. Man erkennt sofort, daß hier keine Anfänger am Werk waren. Der Virus läuft jedoch z.B. nicht auf dem 68030, da der Virus beim Dekodieren seiner eigenen Dekodierroutine seinen Code modifiziert. Selbst ein Ausschalten der Caches kann das Abstürzen nicht verhindern. Wenn ein Open(), Lock(), DeleteFile(), oder Rename() Aufruf erfolgt, dann wird durch Zufall (strahlenpositionsabhängig) eine der 3 folgenden Aktionen ausgeführt:

- Aktion 1. es passiert nichts
- Aktion 2. es wird ein Text-File namens VIRUS.XX geschrieben
- Aktion 3. es wird eine startbares Programm infiziert

Da zumindest die Lock()-Routine sehr häufig aufgerufen wird, kommt es auch sehr schnell zu Punkt 2 und Punkt 3.

Aktion 2

Es wird ein 198 Byte langes Text-File namens VIRUS.XX geschrieben. XX steht für eine zweistellige Hexadezimalzahl. XX wird durch Zufall aus dem Netzfrequenz-Counter gewonnen. Der Inhalt dieser 198 Byte langen Text-Files ist immer gleich:

The Traveling Jack....

I'm traveling from town to town looking for respect,
and all the girls I could lay down make me go erect.

-Jack, 21st of September 1990

Aktion 3

Der 'The Traveling Jack' - Virus hängt sich an ausführbare Programme. Hierzu zählen z.B. auch die Handler und Devices im libs und devs - Verzeichis. Es können alle ausführbaren Programme in allen Verzeichnissen infiziert werden. Da der Virus hierbei aber zufallsgesteuert vorgeht, kann man nicht voraussagen, welche Programme am ehesten infiziert werden. Sicher ist nur, daß Programme kleiner 2000 Bytes nicht infiziert werden.

Der Traveling-Jack-Virus macht sich nicht resetfest. Dadurch fällt er auch weniger auf. Der Virus hat es auch gar nicht nötig, sich resetfest zu machen, denn dadurch, daß er sehr viele Programme infiziert, erwischt er sicherlich auch mal ein Programm, welches immer durch die Startup-Sequence aufgerufen wird. Dadurch wird dann auch der Virus bei jedem Booten aktiviert. Durch die Infektion mit dem 'The Traveling Jack' - Virus verlängern sich die Programme um 2400 oder 2404 Bytes. Der Virus verwendet

zufallsgesteuert vier verschiedene Code-Anfänge, wodurch die Identifizierung des Virus erschwert wird. Mit einem File- oder Diskmonitor kann man keine verräterischen Texte erkennen, da der Virus seinen Code immer zufallsgesteuert verschlüsselt. Es existiert eine weitere Variante des 'The Traveling Jack' - Virus. Dieser Virus verlängert die Programme um 2460 oder 2464 Bytes. Diese Variante arbeitet verstärkt mit KodierROUTINEN, im Endeffekt sind aber beide Viren praktisch identisch. Wie meist bei Linkviren der Fall, infiziert der 'The Traveling Jack' - Virus auch Programme auf der Festplatte. Die Programmierung von Linkviren ist recht anspruchsvoll. Der 'The Traveling Jack' - Virus ist der bisher raffinierteste Linkvirus, da er für die üblichen Antivirusprogramme praktisch unsichtbar ist. Erfreulicherweise wird man durch die Text-Files gewarnt.

1.426 viewtek

Viewtek

Es ist ein 93844 Byte langes Viewtek-Virus-File unterwegs, welches sich gegenüber dem 88944 Byte lange Original-Viewtek-File durch zwei zusätzliche Hunks vor dem Programm unterscheidet.

In dem ersten Hunk wird auf das Vorhandensein einer Datei namens S:HauptPfad geprüft, sollte eine solche nicht zu finden sein, wird mit dem nächsten Hunk weitergearbeitet, sollte allerdings eine solche Datei zu finden sein, dann wird in dieses Mailboxsystem ein User/Kowalsky mit Sysop-Status eingetragen, wozu weitere Dateien wie z.B. BoxDaten/BoxParameter verändert werden. Während dieser erste Hunk auf ein spezielles Mailboxsystem abzielt, wird der zweite Virushunk jedem gefährlich, denn hier wird ein Linkvirus aktiviert, der eben diesen Virushunk vor folgende häufig vorkommende Files zu hängen versucht, wobei diese Files um 4036 Bytes verlängert werden.

```
c:arc c:dms c:ed c:fastgif c:iconx c:iprefs c:lharc c:mount
c:ppshow c:setpatch c:show c:shrink c:version c:vt c:zoo
```

Weiterhin wird ein Prozess namens trackdisk.device erstellt, welcher aus Tarngründen in Stackgröße, Priorität dem normalen pro Diskettenlaufwerk einmal auftretenden trackdisk.device-task nachempfunden wird.

Um noch weniger aufzufallen ändert der Virus-trackdisk.device-Prozess seinen Node-Typ von Prozess auf Task, weil es sich bei dem echten trackdisk.device-task nur um einen task handelt.

Weiterhin setzt der Virus-trackdisk.device-Prozess tc_UserData kurzzeitig auf \$ff8f3826, wodurch eine weitere Installation des Virus verhindert wird, da der Virus alle Tasks durchtestet, ob sie diese Kennung aufweisen.

Außerdem verändert der Virus das normalerweise unbenutzte pad-Listen-Füllbyte der exec-memlist.

In beiden Virushunks liegen große Programmteile kodiert vor, ferner werden alle Dateizugriffe durch direkte Benutzung der Global-Vektor-Tabelle vorgenommen, so daß man mit z.B. snoopdos keinerlei Programmzugriffe erkennen kann.

1.427 bestial devastation

BESTIAL DEVASTATION

Dieser Linkvirus wurde von dem Xeno-Linkvirus abgeleitet, es bestehen folgende Unterschiede, anstatt Open(),loadseg() und Lock() wird nur noch Open() verbogen. Der Virus gibt sich nie mehr durch ein CLI-Meldung zu erkennen, statt dessen steht direkt lesbar im File:

> BESTIAL DEVASTATION <

Da der BESTIAL-Virus genau die gleichen Infektionsroutinen wie der Xeno-Virus benutzt, gilt auch für die vom BESTIAL-Linkvirus befallenen Dateien, daß viele nach der Infektion nicht mehr lauffähig sind.

Normalerweise schafft es der BESTIAL-Virus aber erst gar nicht, Files zu infizieren, denn da der loadseg() und lock()-Vektor nicht mehr verbogen werden, glaubte der Virusprogrammierer darauf verzichten zu können, deren aktuelle Original-Werte in den Virus-Code einzutragen, da aber dennoch zu Beginn des Infektionsversuchs in der verbogenen Open()-Vektor-Routine auch die virusinterne Lock-Routine benutzt wird, was beim XENO-Vorbild auch wegen des Verhinderns einer Endlosschleife dringend nötig war, so führt dies nun in der Regel zum Absturz, da im Viruscode noch unsinnige nicht aktualisierte Ranger-RAM-Werte stehen.

1.428 new age

New Age

Es handelt sich um einen Linkvirus, der den Write-Vektor verbiegt, allerdings nur, wenn dieser als jmp-Sprung vorliegt, was erst ab Kickstart2.0 der Fall ist. Die Files werden nach Xeno -Art infiziert, das heißt der Virus schreibt sich an den Anfang des ersten Codehunks, verlängert also den ersten Codehunk und somit auch das ganze File um 668 Bytes. Wie schon beim Xeno-Linkvirus sind auch die vom NewAge-Virus befallenen Files nach der Infektion oftmals nicht mehr lauffähig, da z.B. die relocunks meist nicht richtig angepaßt werden. Am Ende des Viruscodes kann man lesen

New Age Virus . 1992 ByEvil Jesus

1.429 xeno-virus i+ii

Xeno-Virus I+II

Der Xeno-Virus ist ein echter Linkvirus. Er ist nicht direkt bösartig, dennoch laufen manche infizierte Programme nicht mehr. Er funktioniert auf allen Amigas, da er mit Kickstart 1.2, 1.3, Fast-RAM usw. zusammenarbeitet. Der Xeno-Virus ist der erste Virus, welcher die dos.library manipuliert. Es werden der Open()-, loadseg()- und lock()-Vektor verbogen. Wenn nun diese Vektoren aufgerufen werden, dann werden die entsprechenden Programme infiziert. Da diese Vektoren sehr häufig benutzt werden, kommt es auch sehr schnell zur Infektion sehr vieler Programme. Die Programme vergrößern sich

hierbei um 1124 Bytes, da der erste Code-hunk durch den Viruscode um 1124 Bytes verlängert wird. Programme, welcher kleiner als 384 Bytes sind, werden nicht infiziert. Leider sind circa ein Drittel der Programme nach der Infektion nicht mehr lauffähig. Insbesondere längere Programme stürzen beim Start sofort ab. Auch bei den anderen Linkviren kommt es vor, daß Programme nach der Infektion nicht mehr laufen. Die Rate der nicht mehr lauffähigen Programme liegt aber in der Regel unter 10%, der Xeno-Virus steht hier mit über 30% sehr schlecht da. Der Xeno-Virus infiziert nur Files, welche nicht mit einer Zahl zwischen 0 bis 9 beginnen, und welche im Filenamen selber nur 0-9 und a-z A-Z aufweisen. Ferner werden die Programme im l und devs-Verzeichnis sowie das FastFileSystem-Programm nicht infiziert. Hierdurch soll erreicht werden, daß devices und libraries und fonts nicht infiziert werden, da diese ebenfalls wie starbare Programme aufgebaut sind. Der Xeno-Virus macht sich nicht resetfest. Er muß also nach einem Reset zuerst wieder durch den Aufruf eines bereits infizierten Programms aktiviert werden. Hierbei werden dann die drei obengenannten Vektoren verbogen, wodurch nun wieder Programme infiziert werden können. Dadurch, daß der Xeno-Virus keine Reset-Vektoren verändert und auch ansonsten nur die dos.library manipuliert, erkennen ihn viele Antivirusprogramme nicht. Dann und wann gibt sich der Virus zufallsgesteuert zu erkennen, indem folgende Meldung ausgegeben wird: Greetings Amiga user from the Xeno virus!

Der Xeno-II unterscheidet sich von Xeno-I lediglich durch einige Textänderungen, um dadurch von manchen Antivirusprogrammen nicht mehr erkannt zu werden.

1.430 crime!

Crime!

siehe GolderRider oder LZ . Wie bei diesen beiden Linkviren handelt es sich auch bei dem Crime! um einen nur unter Kickstart 1.3 laufenden Linkvirus, welcher sich an das Ende des ersten Code-Hunks hängt und gegen Ende des Codehunks einen eventuellen RTS-Befehl mit einem Sprungbefehl in den Viruscode überschreibt. Sollte der Code-Hunk mit einem RTS-Befehl enden, so wird dieser mit NOP überschrieben, da ein bra.s mit Offset 0 nicht möglich ist und durch ein NOP ersetzt werden kann. Hierbei wird das File um 1000 Bytes verlängert. Mehrfachinfektionen sind möglich. Der Virus macht sich über den COOL-Vektor resetfest und kann Files infizieren, indem er AllocMem()-, Open()-, LoadSeg()- und die Adresse der Startroutine der Global-Vektor-Funktionen verbiegt. Die Kick-Vektoren und der COLD-Vektor werden gelöscht, wodurch der Virus das einzige resetfeste Programm ist. Bei aktivem Virus kann man im Speicher Crime! lesen, daher der Name. Der Virus kopiert sich unter Zuhilfenahme von MaxLocMem 9216 Bytes unterhalb des Chip-RAM-Endes.

1.431 crime!++

Crime!++

Starke Ähnlichkeiten mit Crime! , jedoch fehlerhafter programmiert. Absturz auf höheren Prozessoren wegen selbstmodifizierendem Code.

Files werden um 872 Bytes verlängert. Statt AllocMem() wird Wait() verbogen. Ansonsten praktisch gleich. Im Speicher kann man Crime!++ lesen. Kopiert sich in Supervisorstack. Der Crime!++ - Linkvirus wurde mittels eines Trojanischen Pferdes namens DriveInfo V0.91 (1704 Bytes) in Umlauf gebracht.

1.432 crime'92

Crime'92

Es handelt sich um einen Linkvirus, der sich wie auch LZ, Goldenrider, Crime und Crime++ an das Ende des ersten Hunks hängt. Es wird nun versucht, den Viruscode durch Änderung eines Original-RTS-Befehls in einen BRA.S zur Ausführung zu bringen. Wenn der letzte Befehl des Original-Hunks ein RTS ist, dann ist keine Änderung in BRA.S möglich, da bei BRA.S kein 0-Offset möglich ist, und ein BRA.W wäre mit 4 Bytes zu lang. Anstatt diesen RTS nun einfach mit NOP zu überschreiben, wie dies der GoldenRider, Crime oder Crime++ machen, überschreibt der Crime'92 den RTS-Befehl bereits mit dem Viruscode. Beide Methoden sind gleichwertig.

Um Antivirusprogrammen das Erkennen des Crime'92-Virus zu erschweren, schreibt der Virus an den Anfang des Viruscodes eine jeweils unterschiedliche Dekodierroutine, um den verschlüsselten Virus zu dekodieren. Zufallsbedingt durch Strahlenposition und Mausposition und Netzteiltickcounter werden unterschiedliche Register und unterschiedliche Dekodierschleifen und unterschiedliche Dekodierbereiche benutzt. Im dekodierten Virus bzw. im Speicher kann man lesen: Crime'92

Der Virus löscht die Kick-Vektoren und macht sich über den COOL-Vektor resetfest. Unter Kickstart 1.2/1,3 wird noch zusätzlich der COLD-Vektor verbogen, um analog 'setpatch r' den Fehler auszubügeln, daß unter Kickstart 1.2/1.3, wenn mehr wie 512 KB Chip-RAM vorhanden ist, keine Programme (und somit auch der Virus) nicht mehr resetfest sind, da der Speicher jeweils neu aufgebaut wird, da zu Zeiten der Programmierung von Kickstart 1,2/1.3 Chip-RAM größer 512 KB noch als fehlerhaft betrachtet wurde, denn anfangs zu Zeiten Kickstart 1.2/1.3 wurde der Amiga lange Zeit nur mit 256 KB bzw. 512 KB Chip-RAM ausgeliefert.

Weiterhin wird der sehr häufig benutzte Wait()-Vektor verbogen, um bei Vorliegen von Kickstart 1.2/1.3 die Adresse der StartRoutine der Global-Vektor-Funktionen und die Adresse der internen Global-Vektor-Open-Funktion und Global-Vektor-loadseg-Funktion auf den Virus zu setzen. Bei höheren Kickstartversionen wird statt dessen der loadseg()- und newloadseg()-Vektor verbogen. Wenn man nun ein Programm startet, dann versucht zuerst der Virus dieses Programm zu infizieren, zuvor werden mit SetProtection() alle Dateimanipulationen erlaubt. Files größer 102400 Bytes werden nie infiziert. Es werden logischerweise nur startbare Files infiziert. Mehrfachinfektionen treten nicht auf. Der Crime'92 beschränkt sich aber leider nicht nur auf das Infizieren von startbaren Dateien, sondern zeigt nach jeweils 32 Fileinfektionen folgendes sehr schädliches Verhalten. Beim Laden eines Files wird ermittelt von welchem Speichermedium das File geladen wurde und dann werden mittels direkter device-programmierung ab Position 1024 des Datenträgers 3072 zufällige Speicherbytes geschrieben. Bei einer Diskette werden also nach dem Bootblock die ersten 6 512-Blöcke überschrieben, wodurch ein meist

doch begrenzter Datenverlust resultiert, schlimmer sind die Auswirkungen bei Festplatten, weil hier in den Rigid-Disk-Datenbereich geschrieben wird, wodurch nach dem nächsten Reset die Festplatte nicht mehr bootet, da zu Recht ein Prüfsummenfehler im Rigid-Disk-Datenbereich festgestellt wird. Abhilfe schafft nur das Mounten der Platte von Hand und anschließendes Neuschreiben des Rigid-Disk-Bereiches. siehe Rigid-Diskblock beschädigen .

1.433 fileghost

FileGhost

Der FileGhost-Linkvirus, der erst ab Kickstart 2.0 lauffähig ist, wird durch ein 8160 Byte langes Trojanisches Pferd namens HardSpeeder in Umlauf gebracht. Dieses Programm verspricht das schnellere Laden von Programmen von Festplatten. Die beiliegende Anleitung erzählt, daß hierzu der loadseg(), newloadseg() und forbid()-Vektor verbogen werden müßte und auf dh0:c/setpatch zugegriffen werden müßte. In Wirklichkeit aber wird der FileGhost-Linkvirus installiert, welcher diese Vektorveränderungen vornimmt, so daß nun jedes zu ladende Programm infiziert werden kann. Programme werden in der Regel über loadseg() oder newloadseg() eingeladen. Der forbid()-Vektor wird verbogen, um bei jedem forbid()-Aufruf einen Zähler hochzuzählen, um dann diesen Zufallswert für das teilweise Verschlüsseln des Viruscodes zu verwenden. In dem entschlüsselten Viruscode im Speicher kann man lesen

```
Hi Friend! Don't worry...
It's only the FileGhost.....
```

Wenn man das HardSpeeder-Programm mit dem Parameter ? aufruft oder kein dh0:c/setpatch vorhanden ist, dann wird kein FileGhost-Linkvirus installiert. Neben dem Installieren des FileGhost-Linkvirus im Speicher infiziert das HardSpeeder-Programm auch das File dh0:c/setpatch, damit der FileGhost-Linkvirus möglichst bei jedem Booten mit dem Aufruf des setpatch-Befehls aktiviert wird, denn der FileGhost selber ist nicht resetfest. Der FileGhost-Linkvirus hängt sich nach LZ-Linkvirus -Manier an das Ende des ersten Code-Hunks und überschreibt gegen Ende des Codehunks einen eventuellen RTS-Befehl mit einem Sprungbefehl in den Viruscode. Sollte der Code-Hunk mit einem RTS-Befehl enden, so wird dieser mit NOP überschrieben, da ein bra.s mit Offset 0 nicht möglich ist und durch ein NOP ersetzt werden kann. Die Files werden durch den Virusbefall um 876 Bytes verlängert.

FileGhost2

Gegenüber dem FileGhost -Linkvirus wird nun nur noch der loadseg-Vektor verbogen und die Files werden statt 876 nun um 796 Byte verlängert. Auch der Text am Ende des Viruscodes hat sich verändert:

```
FileGhost2 - Merry X-Mas and a happy new year
```

1.434 goldenrider

GoldenRider

Er arbeitet nicht mit Kickstart 2.0 und ist immer ab \$7c000 im Speicher zu finden. Er macht sich über den COOL-Vektor resetfest. Weiterhin wird der DoIO()-Vektor verbogen, um dadurch zu erkennen ob eine soeben eingelegte Diskette beschreibbar ist. Diese 'Methode' ist relativ fehlerträchtig, da der DoIO()-Vektor nicht nur für das trackdisk.device benutzt wird. Auch ansonsten ist der GoldenRider-Linkvirus etwas schwächer wie z.B. der programmtechnisch verwandte LZ-Linkvirus programmiert. So werden manchmal Programme beim Infizieren aufgrund eines Programmfehlers unwiderbringlich zerstört. Programme größer 100000 Bytes werden generell nicht infiziert, ebenso werden Programme nicht infiziert, wenn in ihrem Filenamen ASCII-Zeichen kleiner 64 auftreten. Ausgenommen sind '/',':','0' und '1'. Normalerweise bestehen Filenamen aber aus Buchstaben, wodurch diese Programme dann auch infizierbar sind. Der GoldenRider-Linkvirus verbiegt dann auch noch den Open()-Vektor der dos.library. Das Infizieren erfolgt dann also z.B. beim Anschauen oder beim Kopieren von ausführbaren Programmen. Der GoldenRider-Linkvirus unternimmt aber nur in circa 50% der Fälle einen Link-Versuch, da dies zufallsgesteuert wird (Rasterstrahlposition). Bei der Infektion werden die Programme um 868 Bytes länger. In dem infizierten Programm kann man >>> Golden Rider <<< by ABT lesen. Wie der LZ-Linkvirus infiziert auch der GoldenRider-Linkvirus die Programme mehrfach. Im Gegensatz zum LZ-Linkvirus sind jedoch beim GoldenRider auch die mehrfach infizierten Files lauffähig. Der GoldenRider-Linkvirus benutzt im Prinzip dieselbe Link-Logik wie der LZ-Linkvirus, allerdings wird nur ein eventueller RTS durch BRA.S Viruscode ersetzt. Sollte RTS der letzte Befehl des Original-Hunks sein, dann wird er durch NOP ersetzt, denn ein BRA.S mit Offset 0 ist nicht möglich und ein BRA.W mit Offset 0 wäre 4 Bytes lang und somit 2 Bytes länger wie der RTS-Befehl, also zu lang.

1.435 lz-linkvirus

LZ-Linkvirus

Grob gesagt gibt es bisher 3 Virenarten auf dem Amiga: Bootblockviren, Fileviren (mit Sonderfall Disk-Validatorviren) und Linkviren. Programmtechnisch kann man die Linkviren in 3 Arten unterscheiden:
Erstens IRQ-Typ:
Hier wird ein kompletter Hunk an den File-Anfang gehängt.
Zweitens Xeno-Typ:
Hier wird der erste Hunk verlängert, wobei die Virus-Daten an den Anfang des ersten Hunks geschrieben werden.
Diese zwei Linkviren-Arten sind relativ schwierig zu programmieren, da hierbei z.B. die reloc-hunks usw. neu berechnet werden müssen. Etwas leichter zu programmieren ist nun der dritte und neueste Linkvirustyp, welcher nun mit dem LZ-Linkvirus und GoldenRider-Linkvirus funktionsfähig vorliegt. Bei den ersten zwei Linkvirustypen wurde immer zwangsläufig als erstes beim Programmaufruf der Virus-Code durchlaufen. Das heißt, es wurde immer der Linkvirus aktiviert und im Speicher installiert, auf daß nun neue Files infiziert werden konnten. Bei dem neuen dritten Linkvirustyp hingegen wird der Virus-Code nur in circa 50% der Fälle durchlaufen, da der Virus sich in eine Unterroutine am Ende des ersten Code-Hunks

einhängt. Es kann aber durchaus sein, daß diese Unterroutine nie aufgerufen wird, da diese Unterroutine vielleicht nur für eine Fehlermeldung oder einen sonstigen Sonderfall gebraucht wird. Dem 'Nachteil', daß der Virus nicht immer aktiviert wird, steht der 'Vorteil' der relativ leichten Programmierbarkeit gegenüber. Es wird am Ende des ersten Code-Hunks nach einem 'rts'-Befehl (=Ende einer Unterroutine) gesucht. Wenn ein solcher Befehl gefunden wird, dann wird der 'rts'-Befehl in einen Sprung in den Virus-Code umgeändert. Erst am Ende des Virus-Codes erfolgt dann der 'rts'-Befehl. Der Virus hängt sich also in eine eventuelle Unterroutine ein. Da der Virus-Code an das Ende des Code-Hunks angehängt wird, brauchen auch keine Reloc-Hunks usw. neu berechnet zu werden. Der Virus muß lediglich die zwei Code-Hunk-Längen-Angaben um die Virus-Code-Verlängerung erhöhen. Dadurch, daß dieser neue Linkvirustyp ein \$4e75(rts) einfach als Ende einer Unterroutine betrachtet und sich nun willkürlich hier reinhängt, können manchmal nicht mehr funktionsfähige Programme entstehen. Bemerkenswert ist aber an diesem neuen Linkvirustyp die relativ einfache Programmierbarkeit, wodurch in Zukunft leider verstärkt mit Linkviren dieses Typus gerechnet werden muß.

Bei dem LZ-Linkvirus handelt es sich um einen recht sauber programmierten Linkvirus, welcher allerdings nur unter Kickstart 1.2/1.3 arbeitet. Der LZ-Linkvirus verändert die interne Adresse der AmigaDOS-Write-Funktion in der Global-Vektor-Table. Da letztendlich alle Write()-Aufrufe über diese Tabelle laufen, kann sich der Virus in jeden Schreibzugriff einklinken. Die Neuinfektion von Programmen geschieht also in der Regel beim Kopieren von startbaren Programmen. Das Original-File bleibt unverändert und die Kopie wird infiziert. Es werden jedoch nur circa 50% aller ausführbaren Programme infiziert, da für eine Infektion gewisse Bedingungen vorhanden sein müssen. Es muß am Ende des ersten Code-Hunks eine Unterroutine vorhanden sein, an die der Virus sich dann dranhängen kann. Das heißt der Virus sucht nach \$4e75=rts oder \$4eee=jmp offset(a6) und ersetzt diesen 'Beende Unterroutine'-Befehl durch einen Sprung in den Virus-Code. Auch wenn der RTS der letzte Befehl des Original-Hunks ist, kann er durch BRA.S ersetzt werden, da noch 2 Bytes im Viruscode selber übersprungen werden, wodurch also RTS in BRA.S mit Offset 2 umgeändert wird, ein BRA.S mit Offset 0 wäre nämlich unmöglich. Erst am Ende des Virus-Codes erfolgt dann mit einem 'rts'-Befehl das Ende der Unterroutine. Der Virus versucht sich also in eine eventuelle Unterroutine einzuhängen. Es werden nun also circa 50% aller Files infiziert. Aber wenn man diese infizierten Programme startet, dann wird nur wiederum in circa 50% der Fälle der Virus auch gestartet, denn es wird nicht immer die Unterroutine mit dem darangehängten Virus durchlaufen. Es kommt darauf an, wofür oder in welchem Fall diese Unterroutine gebraucht wird. Das kann immer sein, dann wird der Virus immer aktiviert, oder nur in einem Sonderfall, dann wird der Virus meist nicht aktiviert.

Trotz recht sauberer Programmierung weist der LZ-Linkvirus einen schwerwiegenden Fehler auf, denn er infiziert Programme mehrfach, da er sich an seinen eigenen Virus-Code anhängt. Solche mehrfach infizierten Files stürzen jedoch kurz nach dem Start ab.

Der LZ-Linkvirus verlängert bei einer Infektion die Programme um 400 Bytes. Diese erstaunlich kurze Virus-Code-Länge wird durch den Verzicht auf jegliche Text-Meldungen erreicht. Der Virus besteht 'lediglich' aus der Infizierungsroutine. Sollte der erste Code-Hunk des Programms kleiner 1000 Bytes sein, dann erfolgt keine Infektion.

Der Virus hat es nicht nötig, sich auf auffällige Art restfest zu machen, da er ja über kurz oder lang durch ein bereits infiziertes Programm aufgerufen wird. Wenn man 'run' oder 'newcli' aufruft, dann wird der Virus entfernt, da unter Kickstart 1.2/1.3 hierbei die Global-Vektor-Table mit den Original-ROM-Werten refresht wird.

1.436 commander

COMMANDER

Es handelt sich um einen Linkvirus, der Programme um 1664 Bytes verlängert und den Dateikommentar auf \$A0 setzt. Programme die mit v oder V beginnen, werden nicht infiziert. Der Viruscode ist zufallsgesteuert (Strahlenposition) kodiert und endet immer mit \$7419F1A2. Durch diese Kennung verhindert der Virus die Mehrfachinfektion eines Programmes. Der Virus hängt seine Daten an das Ende des ersten Codehunks. Damit dieser Viruscode auch aufgerufen wird, ersetzt der Virus in den Originaldaten einen jsr Offset(a6) \$4eae.... oder bsr.l Offset-Befehl \$6100.... durch einen jsr Viruscode(pc)-Befehl \$4eba.... Wenn nun das infizierte Programm gestartet wird, dann erfolgt ein Sprung in den Viruscode, wo dann der Virus installiert wird, hierzu wird der open-, rename-, lock-, loadseg-, setcomment-Vektor auf den Virus verbogen, wobei dann beim Aufruf dieser Befehle eine Virusinfektion versucht wird. Neben Kickstart2.0 werden auch die Besonderheiten der Kickstart1.3-dos.library berücksichtigt. Nachdem sich der Virus installiert hat, wird noch versucht konkret DH0:C/LoadWB zu infizieren. Der Virus verbiegt weiterhin Examine und Exnext, um hierbei ein nicht infiziertes File vorzutäuschen, indem kein Dateikommentar und eine Filelänge minus Viruscode im Fileinfoblock zurückgemeldet wird. Nach Dekodierung ist am Ende des Viruscodes zu lesen:

```
-<( COMMANDER )>- by Bra!N BlaSTer in 1994 DH0:C/LoadWB
```

1.437 darkavenger

DarkAvenger

Es handelt sich um einen Linkvirus, der sich kodiert an das Ende des ersten Codehunks anhängt und an den Anfang des Codehunks einen Sprungbefehl in den angehängten Viruscode schreibt. Es handelt sich bei dem DarkAvenger um eine Weiterentwicklung des Infiltrator -Linkvirus. Es existieren zwei nur geringfügig unterschiedliche DarkAvenger-Versionen, die eine Version, Typ A genannt, verlängert Files um 1128 Bytes und ändert beim Start zufallsgesteuert die Titelleiste des aktiven Fensters auf

```
-- The Dark Avenger --
```

Bei der anderen DarkAvenger-Version, Typ B genannt, fehlt die Titeländerungsroutine, wodurch die Files nur um 1072 Bytes verlängert werden. Beim Typ B kann man nach der Dekodierung folgenden Text lesen:

```
Reminders of past , fear of the future: SEPTIC SCHIZO.
```

Der DarkAvenger-Linkvirus unterscheidet sich von dem Infiltrator dadurch,

daß anstatt des loadseg()-Vektors nun der Open()-Vektor verbogen wird, und daß der DarkAvenger unter allen Kickstartversionen läuft, weil er den speziellen Aufbau der dos.library unter Kickstart 1.2/1.3 gesondert behandelt. Der Infiltrator-Linkvirus hingegen läuft erst ab Kickstart 2.0. Ansonsten gelten die beim Infiltrator-Linkvirus gemachten Angaben.

1.438 infiltrator

Infiltrator

Dieser Linkvirus macht sich nicht resetfest und arbeitet erst ab Kick2.0. Die ersten 4 Bytes eines startbaren Programms werden mit einem bsr-Befehl überschrieben. Mit diesem bsr-Befehl wird in den Virus-Code gesprungen, der kodiert an das Ende des ersten Code-hunks angehängt wird. In diesem Virus-Code wird der loadseg()-Vektor verbogen, wodurch nun Programme beim Starten infiziert werden können, wobei vorher mittels SetProtection() die Protectionbits auf RWED gesetzt werden. Es sind nicht alle infizierten Programme lauffähig, denn wenn der erste Original-Befehl ein absoluter Befehl war, dann wird beim Starten des infizierten Programms der bsr Virus-Code-Befehl durch die Adressanpassung zerstört und es erfolgt ein Absturz. Der Infiltrator-Virus versucht solche Fälle zu verhindern, indem er Programme mit einem absoluten jmp- oder jsr-Befehl nicht infiziert. Durch diesen Test werden aber nur die häufigsten absoluten Befehle erkannt. Die infizierten Programme werden um 1052 Bytes verlängert. Der Name des Virus rührt von folgendem Text her, der nach Entschlüsselung wie folgt lautet:

```
Howdy hacker! This is The Infiltrator! Smart people with
knowledge about this code can do ALOT of damage, belive me!
```

Der Infiltrator prüft weiterhin auf das Vorhandensein einer Datei namens user.data, welche dann u.U. verändert wird.

Der Infiltrator-Linkvirus hängt also seinen Viruscode wie die LZ-, Goldenrider- und crime-Linkviren an das Ende des ersten Code-Hunks. Während die LZ-, Goldenrider- und crime-Linkviren aber den Einsprung in den Virus-Code durch Veränderungen des Original-Codes vom Ende her zu bewerkstelligen versuchen, verändert der Infiltrator-Virus die ersten 4 Bytes des Original-Codes, wodurch im Gegensatz zu den LZ-, Goldenrider- und crime-Linkviren immer der Virus-Code-Aufruf sichergestellt ist. Nach Infiltrator-Art infizierte Programme haben aber den Nachteil, daß sie abstürzen können, und zwar dann wenn eine Adressanpassung aufgrund eines ursprünglich hier stehenden absoluten Befehls vorgenommen wird.

1.439 polyzygotronifikator

Polyzygotronifikator

Es handelt sich um einen Linkvirus, welcher ab Kickstart V37 den loadseg-Vektor verbiegt, um Files beim Laden zu infizieren, hierbei werden die Files zufallsgesteuert (Strahlenposition) um 1248 bis 1296 Bytes verlängert, wobei der Virus seine Daten an das Ende des ersten

Codehunks anhängt, den Codehunk also verlängert, die variable Viruslänge wird dadurch möglich, daß zu Beginn der Viruscodes verschiedene Sprungbefehle mit jeweils unterschiedlichen Offsets geschrieben werden, wodurch also mehr oder weniger (Müll)Daten übersprungen werden. Auf diesen unterschiedlich langen Virusvorspann folgt dann der kodierte und immer gleich lange Hauptviruscode, der mit \$1994 endet, das heißt der erste Codehunk eines infizierten Files endet mit \$1994, durch diese Kennung vermeidet der Virus Mehrfachinfektionen. Der Virus infiziert nur dann Files, wenn snoopdos nicht vorhanden ist, um sich nicht unnötig zu verraten. Da aber das neue snoopdos3 nicht mehr einfach snoopdos als Tasknamen benutzt, wird die Anwesenheit von snoopdos3 nicht erkannt. Es werden nur Files auf Datenträgern mit einer Kapazität größer 4 KB infiziert, auf Disketten werden also keine Files infiziert. Weiterhin werden generell keine Files infiziert, welche . oder - im Filenamen aufweisen. Der Virus hängt also seinen Viruscode an das Ende des ersten Codehunks, damit aber auch der Aufruf dieses Viruscodes gesichert ist, sucht der Virus nach `move.l $4.w,a6` und ersetzt diese 4 Bytes durch bsr Viruscode, sollte `move.l $4.l,a6` gefunden werden, dann werden diese 6 Bytes durch bsr Viruscode gefolgt von dem 2 Byte-Befehl NOP ersetzt. Wenn ein infiziertes File gestartet wird und dann mit bsr Viruscode der Viruscode ausgeführt wird, dann wird in diesem Viruscode der Anfang des Viruscodes mit `move.l $4.w,a6` überschrieben, um damit möglichst wieder den Originalfilezustand wiederherzustellen, sicherer wäre es allerdings gewesen, direkt den bsr Viruscode-Befehl wieder durch `move.l $4.a6` zu ersetzen.

1.440 red october v1.7

Red October V1.7

Es gibt keine Begründung für diese Bezeichnung, denn der Virus macht niemals eine entsprechende Meldung. Das einzige Kriterium, das man zur Namensfindung heranziehen könnte, wäre der folgende Text, den man ab Position 1178 in infizierten Programmen lesen kann.

```
timer.device          dos.library  ram:  ram:1
```

Dieser Virus nahm seinen Ursprung mit einem 1296 Bytes langem File, in welchem der reine Virus-Code enthalten war. Eben diese Viruscode-Informationen verlängern dann ein infiziertes Programm um ebenfalls 1296 Bytes.

Wenn man das Ursprungsvirusfile oder ein neues infiziertes File startet, dann wird zufallsgesteuert in circa 6% der Fälle ein Reset ausgelöst. In circa 33% der Fälle wird versucht eine Linkvirusinfektion vorzunehmen. Hierbei wird mittels der üblichen DOS-Funktionen die Diskette nach noch nicht infizierten Programmen durchsucht. Wenn man also ein infiziertes Programm startet, dann wird maximal ein weiteres Programm infiziert. Der Virus verbiegt keine Vektoren und ist auch nicht resetfest.

Es handelt sich um einen neuen Linkvirustyp, denn der Virus arbeitet nach folgendem neuartigen Prinzip.

Der Virus hängt seinen Code, welcher dem Originalvirusfile entspricht, direkt vor das zu infizierende File. Es bleiben alle Hunkstrukturen des Original-Files erhalten. Es wird also das Programm direkt von Diskette mittels Read eingelesen und keinerlei Umrechnungen vorgenommen. Der Viruscode und das Original-File werden nun als ein kompletter

Code-Hunk ohne Relok-Hunk abgespeichert. Wenn man nun dieses infizierte File startet, dann wird der Virus-Code zuerst ausgeführt. Hierbei erfolgt wie bereits erwähnt in 6% der Fälle ein Reset, in 33% ein Neuinfektionsversuch und in 60% der Fälle wird einfach nur das Original-File zur Ausführung gebracht. Dieses funktioniert aber nicht immer. Es wird folgendermaßen vorgegangen. Der Virus-Code speichert das Original-File nach RAM:1 ab und lädt dann das Originalfile mittels loadseg() zwecks korrekter Relozierung ein. Dann wird in diesen Original-Code direkt eingesprungen. Als Prozeß-Strukturen usw. werden die beim Starten des infizierten Programms angelegten weiterverwendet.

1.441 debugger

DEBUGGER

Es handelt sich um einen Linkvirus, der unter Kickstart 1.2/1.3 abstürzt, da willkürlich versucht wird, Funktionen aufzurufen, die erst ab Kickstart 2.0 vorhanden sind (GetProgramDir, CacheClearU). Infizierte Files sind um 1088 Bytes verlängert. Es handelt sich um einen neuartigen Linkvirus, welcher die ersten 186 Bytes des ersten Codehunks durch Viruscode ersetzt und an das Ende des Programms einen Debughunk anhängt, in welchem weiterer Viruscode und die 186 geretteten Originaldaten stehen. Am Fileende kann folgender Text gelesen werden.

DEBUGGER(041994)

Wenn man ein infiziertes File startet, dann wird das soeben gestartete Programm mit Open() geöffnet und der Debug-Hunk vom Ende des Programmes eingelesen und in diese Daten eingesprungen. Dieses Einlesen des Debughunks ist nötig, da das Betriebssystem bei loadseg debug-hunks wie auch name-hunks überliest, da sie für die Programmausführung unwichtig sind.

Der Virus kopiert dann die 186 Originalbytes an den Anfang des ersten Codehunks zurück und reloziert anschließend diese Daten, denn der Virus hat bei dem infizierten File die Hunklänge des ersten Hunks solcherart verlängert, daß die reloc-hunk-kennung nicht erkannt wird, damit die 186 Virusbytes nicht fälschlicherweise mit den reloc-informationen für die Originaldaten reloziert werden. Man erkennt, daß dieser Linkvirus von einem Profi programmiert wurde.

Anschließend wird der loadseg- und write-Vektor verbogen, um beim Aufruf der loadseg-Funktion Files zu infizieren.

1.442 lazarus

Lazarus

Mit Kickstart 1.2/1.3 wurde im C-Verzeichnis ein Programm namens 'diskdoctor' mitgeliefert, welches einem vom Betriebssystem zum Reparieren beschädigter Disketten empfohlen wurde. Allerdings war dieses Programm nicht sonderlich leistungstark, vielmehr wurden auch oft unnötigerweise weitere Daten von dem Programm gelöscht. Das Mittel der Wahl bei Problemen mit Speichermedien ist 'disksalv'. Wenn der 'diskdoctor' das Problem nicht

beheben konnte, dann wurde dies durch Umbenennung der Diskette in 'Lazarus' angezeigt. Ein Virus namens 'Lazarus' hingegen ist bisher nicht bekannt.

1.443 myindex

alphabetische Sortierung aller Schlagwörter

\$4EB9-4EF9-Link
\$4EB9-Link
16Bit Crew
6ULDV8
A.H.C.
A.I.S.F. INTERLAMER
AAA-Enhancer
ACP
AE-Registrar
AEK
Aibon
Aibon2
AIDS
AIDS-HIV
ALIEN NEW BEAT
alle Dateien prüfen
Allgemeine Einführung in die Virenproblematik
Amiga-Master
AmigaFreak
AmigaGuide-Hilfe
AMIGAKNIGHTS
AmiPatch V1.0a
Angel
Anti-EuroMail-Virus
Antichrist
AppIcon, Drag and Drop
Arbeitsfenster öffnen
Arbeitsfensterfarben
Austral.Parasite
automatisch
automatisches Entpacken von Dateiarchiven
automatisches Entpacken von Programmen
Autoradresse
BamigaSectorOne
Bedienungsanleitung
beenden
Befehlsdateimodus
Benutzung von Betriebssystemfunktionen kontrollieren
beschädigte .info-files anzeigen
beschädigte Programme anzeigen
Beseitigung der Bootblockviren
Beseitigung der Fileviren, Disk-Validatorviren und Linkviren
BESTIAL DEVASTATION
BGS9 I+III+III
BLACK-KNIGHT
BlackFlash
BladeRunners
BLF

BlowJob
BLUEBOX-icon.library
Boot-Menü
Bootblock->Datei
Bootblock->Puffer
Bootblock-Analyse
Bootblock-Archivierung
Bootblock-Massacre
Bootblock-Schreibzugriffkontrolle
Bootblockviren
Bootblockvirenübersicht
Bootdisk-Bug
BootShop
BootX-Updater
BRET-HAWNES
BURN
BUTONIC's 1.1
BUTONIC-JEFF
ByteBandit
ByteBandit 1
ByteBandit 2
ByteBanditImitation
ByteBanditVIPHS
ByteParasiteI
ByteParasiteII
ByteParasiteIII
ByteVoyager I
ByteVoyager II
CCCP-Bootblock+Linkvirus
Challenger-Fish622
CHAOS-MASTER V0.5
Chaos-TaiPan
Chip-Speicher bevorzugen
CHRISTMAS VIOLATOR
Claas Abraham
CLIST-Lamer (UK-Lamerstyle)
CLONK
CLONK-Installer
CLP
COBRA
CODER
ColorsVirusCarrierTurkBB
COMMANDER
commodity
Commodore-Virus
CompuPhagozytel
CompuPhagozyte2
CompuPhagozyte3
CompuPhagozyte4
CompuPhagozyte5
CompuPhagozyte6
CompuPhagozyte7
CompuPhagozyte8
CONMAN-TROJAN
Copylock
CRACKER-Extermin.
CREEPING-EEL

Crime!
Crime!++
Crime!++-Trojan.Pferd
Crime'92
D&A
D-Structure (A,B,C)
DAFGderFEHY
DAG-Virus-Infecter
DarkAvenger
DARTH VADER V1.1
DASA-ByteWarrior
DAT-89
DATA CRIME
DATA LOCK
Datei ->Bootblock
Dateiauswahlfenster wählen und vorbelegen
Dateiveränderungen erkennen
DEBUGGER
Decompiler
Degrad
DERK-MALLANDER
Descriptor V3.0
Destructor
DETLEF
DF0: DF1: DF2: DF3:
Dialer V2.8g
DIGITAL DREAM
DigitalEmotions
Disassembler, Analyser
DISASTER-MASTER V2
Disk-Boot verhindern
DISK-KILLER V1.0
Disk-Validatorviren
Disk-Validatorviren, (dekodieren, umbenennen)
Disk-Validatorvirenübersicht
DiskDoktors
Diskeinlegen->Komplettest
Diskeinlegen->Schnelltest
DiskHerpes
diskrepairV1.20
DiskSpeedCheckV1.01B
Disktroyer
DIVINA EXTERMIN.
dm-trash
DOOM
DOOR_BELLS
Dopusrt
Dotty
Drivebit-Bug
DUM2DUM
DUMDUM
DWEdit (1.62)
EASY-E
Einstellungen speichern
ELECTRO-VISION
ELENI!
ELENI-CLOCK

Elien
EXCREMENT
EXCREMENT-Installer
Excreminator V1.0
Exterminator II
EXTREME
F.I.C.A
f.Prüfsum->NoBoot
Farbsignale
FAST
FAST 1
Fast-Speicher-Zugriff erlauben
FastEddie
FastEddie-Infector
FCheck
Fensterleiste
feste Kickstart-Adressen
Festplatte
File/Linkviren suchen und entfernen
FileGhost
FileGhost2
Filennamenpuffer
Fileviren
Fileviren, Mailboxviren, Trojanische Pferde
Filevirenübersicht
Forpib
Freedom
FrenchKiss
FRESHMAKER
Frity
FUCK
FUCK-Lamer (INGO 'S RETURN)
fuck.device
Future-Disaster
Gadaffi
Gandalf
GENERALHUNTER V3.2
GENESTEALER
Geschichte
Glasnost
GoldenRider
Graffiti
GREMLIN
GuardiansBootAids
GX.TEAM
GYROS
Hauptmerkmale
HCS
HEIL
HILLY
Hochofen (=Trabbi)
HODEN V33.17
HULKSTERS
Hunklab-Link
höhere Prozessoren
ICE
Incognito

Infiltrator
Inger.IQ.Virus
Installation
Installiere DFX:
Intro-Maker V1.00 by TCR
IRQ-Linkvirus I+II
JEFF3.10-Trojan.Pferd
JINX
JITR
JOSHUA
JOSHUA 1
JulieTick
KaKo
Kauki
Kill
Killed
Konkrete Virenbeschreibungen
L.A.D.S
L.A.D.S - A.I.D.S
LADS-MVK
LameBlame-TaiPan (LameBlame, CHEATER-HIJACKER, POLISH)
LAMER-bomb (Gotcha LAMER)
Lamer-Bootblockviren
LamerBB-Trojan.Pferd
Lauffähigkeit von Viren
Laufwerke ändern
Laureline V1.0
Lazarus
Lern-Modus
LEVIATHAN-Bootblock+Filevirus
LHACHECK 1.1
LOOK-BBS
Liberator1.21-MemCheck
Liberator3.0-cv
Liberator5.01-pv
Linkviren
Linkviren
Linkvirenübersicht
Linkvirus von File abtrennen
Little Sven
Little Sven-Trojan.Pferd
Loverboy&Sexmachine
LSD
LUPO
LZ-Linkvirus
MAD
MAD II
MAD III
MAD IV
MComm
MegaLINK
MEGAMASTER
MENEM'S REVENGE
merry
MessAngel
METAMORPHOSISV1.0-Bootblock+Linkvirus
MEXX

MG's Virus V1.0
MicroMaster
MICROSYSTEMS
MODEMCHECK-FUCK-Virus
ModemSpeederV2.1
Mongo
Morbid.Angel.Virus
MOSH
MOSH 2
mount
Mount-ELENI-WIRUS
move sr,<ea> Handler
Multiselect
MUTILATOR
M_CHAT V2.3
NANO1
NANO2
NaST
Nasty
New Age
No.Bandit.any.More
NoGuruV2.0
Notify
NoVi
Obelisk
Obelisk II
OPAPA
Orange-Disk-Validatorvirus(=DiskVal1234)
Overkill
Packerproblematik
PAL/NTSC
PARADOX I
PARADOX II
PARAMOUNT
PARATAX I
PARATAX II
PARATAX III
PentagonSlayer
Personal Bootblock
PERVERSE I
PHA
Polyzygotronifikator
PowerBomb
PowerPacker3.2-Bomb
PowerTeam
PP-Died2.8
PP-MegaMon
PP-Snap1.61
Protokoll
Puffer ->Bootblock
Purge
PVL
QRDL
r.Prüfsum->Boot
Red October V1.7
Reset
ReturnOfTheLamerExterminator-Disk-Validatorvirus

Revenge
Revenge Bootloader
RevengeOfTheLamerExterm.
Rigiddiskblock beschädigen
Ripper
Riska
SACHSEN NO.1
SACHSEN NO.3
SADDAM-HUSSEIN-Disk-Validatorvirus und Abkömmlinge
SaddamHussein
SAO PAULO
SATAN
SCA
SCA-2001
SCA-AIDS
SCA-DAG
SCA-Kefrens
SCA-MAX
scan.x
SCARFACE
Schirm wählen
Schließsymbol
Schlußfolgerung
Schnelleinstieg
Schreibzugriff auf Device melden
scsi
SeekSpeed
Selbsttest
Sendarian
Sentinel-USSR492
SEPULTURA
Sepultura (V2.26)
SHIT
showsypsops
SmilyCancerCenturions
snoopdos (1.9)
snoopdos (2.1)
SnoopEx-DLog-DevilDoor11
Sonja
Speicheranzeige
Speicherausbau
Speichermedien, RigidDiskBlock-Verwaltung
SS
STARCOM
Starfire-EastStar
Starfire-Northstar
Startup-Sequence-Kontrolle
Statistik
Suicide
SuperBoy
SwiftWare-DevilDoor8
SysinfoV2.2
Systemveränderungen kontrollieren
SystemZ
SystemZ 6.1,6.3,6.4,6.5
Systemübersicht
T.F.C. Revenge Virus

T.F.C.-loadwb
TAI
Target
Tastaturbelegung
TeleCom
TELSTAR(SystemZ-V6.0)
Termigator
Terrorists
THE SMILY CANCER II
TheTravelingJack
TIME-BOMB-V1.0
TimeBomb V0.9
TimeBomber(VIRUSTEST)
timer
TomatesGentechnicService
ToolsDaemon2.2
Top util V1.0
Traveller1.0
TRIPLEX
TRISECTOR 911
TROJAN KILLER V3.0
TURK VIRUS 1.3
TWINZ SANTA CLAUS
Uhr
Uhr aktivieren
Uhrvirus
ULTRA-FOX
Umyj Dupe
unsichtbare Zeichen in Filenamen anzeigen
Unterverzeichnisse
VCCofTNT
Verbreitung
Vergleichen
VERMIN
Viewtek
VIRUS FIGHTER V1.0
VIRUS TERMINATORV6.0
VIRUS-INSTALL v2.0
Virusanzeichen
VirusBlaster
VirusConstr.I
VirusConstr.II
VirusConstructionSetI
VirusConstructionSetII
Virusentfern-Dialogfenster
VirusMaker V1.0
VirusV1
VKill 1.0
VMK V3.00
VT-Faster
WAFT
WAHNFRIED
WARHAWK
Warnton
Warnung
Warsaw Avenger
WhiteBoxV8.0

Xeno-Virus I+II
XLINK-Link
XPR-SpeederV3.2
Z.E.S.T
ZACCESS V1.0
ZACCESS V2.0
ZACCESS V3.0
Zeige Datei
Zeige DFX:
ZENKER
Zero-Location-Bug
Zombi I
Zukunftsansichten
zukünftige Viren