RWatcher

A Configurable Resource Watching INIT that Provides

Partial Virus Protection for Macintosh Programmers

Version 1.0. November, 1988.

John Norstad
Academic Computing and Network Services
Northwestern University
2129 Sheridan Road
Evanston, IL 60208

Bitnet: jln@nuacc
Internet: jln@nuacc.acns.nwu.edu

Copyright Notice and Disclaimer

Introduction

CE Software's Vaccine by Donald Brown is currently the best general-purpose protection against Macintosh viruses. Vaccine contains a number of specific and general checks to detect and prevent suspicious activities typical of viruses. It is effective against Scores and nVIR, and has enough generality so that it will likely be effective against other viruses that may appear.

Among other checks, Vaccine complains whenever an attempt is made to create a CODE resource. This is good for the general public, but some programmers find it very annoying. Whenever a program is linked the linker creates CODE resources, and Vaccine complains about each one.

For MPW programmers Vaccine provides an option to “always compile MPW INITs”. This option appears to be misnamed, since it actually permits the creation of CODE resources as well as INITS in the MPW environment. Enabling this option permits unobstructed use of MPW, but does not help those programmers who use non-MPW development environments.

RWatcher was written for those non-MPW programmers who would like some virus protection, especially against Scores and nVIR, but are not willing to use Vaccine.

RWatcher is a very simple INIT that patches the AddResource and ChangedResource system traps. The patches watch for attempts to create or modify any of a list of resources. If such an attempt is made the patches beep 10 times and exit to shell (quit the offending application).

The list of resources to be monitored is specified in the RLIS 128 resource on the INIT file. This list can be easily modified with ResEdit.

As with Vaccine, RWatcher provides only partial protection against viruses. In fact, RWatcher is much weaker protection than Vaccine. RWatcher will only catch those viruses that are explicitly listed in its RLIS 128 resource. The default configuration is currently set to catch Scores and nVIR, but it's almost certain that any new virus will get around RWatcher's checks. If a new virus appears you'll have to teach RWatcher to recognize it, by adding appropriate entries to RLIS 128. Vaccine, on the other hand, contains some general-purpose checks (like prohibiting the creation of any CODE or INIT resource) that are likely to be effective against future viruses.

For this reason, use Vaccine if it's at all possible—it provides much stronger protection than RWatcher. Use RWatcher only if you are a non-MPW programmer who refuses to use Vaccine and would otherwise have no protection at all.

If you do decide to use RWatcher, please read the following sections carefully. They will show how you can significantly strengthen its checks by modifying the RLIS 128 resource. Make your checks as strong as possible to provide the best possible protection.

Unlike Vaccine, RWatcher does not attempt to put up a dialog box when it detects an attack. In the cases of both Scores and nVIR, neither QuickDraw, the Window Manager, nor the Dialog Manager have been initialized at the time of attack, so Vaccine bombs when it tries to put up the dialog. RWatcher instead beeps 10 times and exits to shell. If you hear these 10 beeps when you try to launch a new program, then it probably has a virus.

RWatcher also does not have any CDEV to set options. You must use ResEdit if you want to change its default settings. Since RWatcher was designed for use by programmers, I don't feel this is a major drawback.

Another difference between RWatcher and Vaccine is that you can tell RWatcher exactly what you want it to look for by modifying the RLIS 128 resource with ResEdit. The only option Vaccine has is the “always compile MPW INITS” option.

How to Install RWatcher

To install RWatcher just drag a copy into your system folder and reboot. If you'd like to verify that it's active, get into ResEdit and try to create any nVIR resource. The system should beep 10 times and quit ResEdit.

Warning: RWatcher **will not** protect you if you install it on a system that is already infected. It **will** protect a clean system against future infection by Scores and/or nVIR. Check your system first with Interferon or some other virus detection tool before installing RWatcher.

RWatcher's Default Configuration

I currently distribute RWatcher with the following resources specified in the RLIS 128 resource list:

Type	ID	Size	Virus
INIT	6	772	Scores
INIT	10	1020	Scores
INIT	17	480	Scores
atpl	128	2410	Scores
DATA	-4001	7026	Scores
INIT	32	366	nVIR
INIT	32	416	nVIR
nVIR	any	any	nVIR

The last entry above matches any ID or size. Any nVIR resource is considered suspect.

How to Change the RWatcher Configuration

You use ResEdit to edit the RLIS 128 resource to modify the list of resources that RWatcher considers suspect. Since you're a programmer, I don't have to tell you how to use ResEdit!

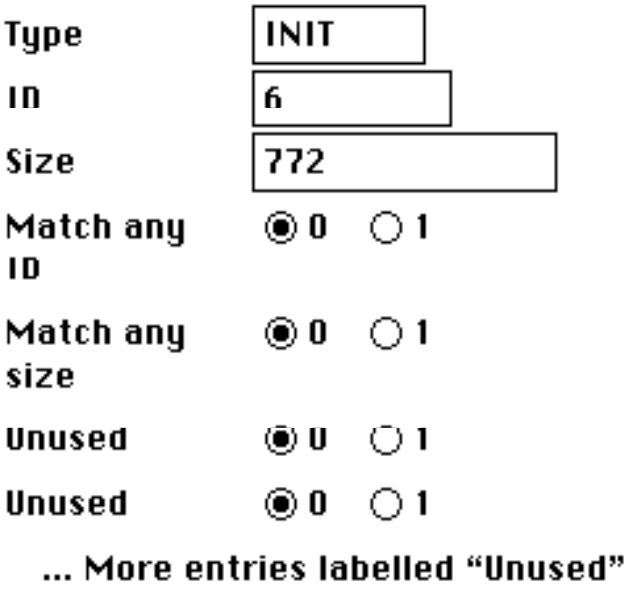
After using ResEdit to modify RLIS 128 you must reboot before the changes will take effect.

For your convenience, I've created a ResEdit template for editing the resource. You'll find it on the file named “RLIS Template”, resource type TMPL, resource name RLIS. You'll want to copy this template into your copy of ResEdit. Use a copy of ResEdit to do this—I've had bad experiences trying to use ResEdit to modify itself!

After you've added the template to ResEdit, use ResEdit to open the RLIS 128 resource on the RWatcher INIT file. You should see a list of entries each of which looks like the following:

5

*****



There are only five fields in each entry that are used. The “Unused” fields are there to pad the entry out to a word boundary.

Use the Type, ID, and Size fields to specify the type, id, and size of a suspect resource.

If you'd like the entry to match any id, set the "Match any ID" field to 1. In this case the contents of the ID field are irrelevant—you can leave it blank or specify a 0 or anything else you wish. Similarly, if you'd like the entry to match any size, set the "Match any size" field to 1. In this case the contents of the Size field are irrelevant.

If you only program applications, DAs, and other non-systems programs, and you never write INITs, you can and by all means should significantly strengthen RWatcher's checks by adding an entry to prohibit the creation of INIT resources. Add the following entry to the RLIS 128 list:

Type = INIT
ID = 0
Size = 0
Match any ID = 1
Match any size = 1

With these settings you'll still be able to create CODE resources, but you won't be able to create INIT resources. Most viruses create an INIT.

For even greater protection you can add similar wild card entries for all the other types of resources that you never muck with, like WDEFs, DRVRs, and so on. It's up to you to decide which resource types you want to prohibit.

Using Both Vaccine and RWatcher

You can use Vaccine and RWatcher together if you wish (I do). In this case the order in which the INITs are executed is important. The system executes INIT files at boot time in alphabetical order. The INIT that is loaded **last** is the one whose AddResource and ChangedResource patches will be applied **first**.

I renamed Vaccine by putting a space in front of the name, so it is loaded first. Thus Vaccine is loaded before RWatcher, and hence RWatcher's checks are done before Vaccine's. If I try to run an application that is infected with nVIR or Scores on my machine RWatcher detects the attempt to infect my system, beeps 10 times, and exits to shell. Vaccine never gets a chance to do its thing (which is bomb in this case). Vaccine is there as a backup to catch things that get past RWatcher.

If you don't rename the INITs RWatcher will execute before Vaccine, and hence Vaccine's checks will be done before RWatcher's. In this case if you run an infected application Vaccine will detect it, and you'll get the bomb.

Source Code

I distribute RWatcher with source code (MPW Assembler). It's a very simple piece of code. You can examine it to see exactly how RWatcher works and what it does.

There are arguments for and against distributing source for anti-viral software. For example, if source were available for Vaccine it might make it too easy for potential virus writers to defeat Vaccine's protection mechanisms. On the other hand, there is undisputed value in being able to see exactly how the software works and what it does. In the case of RWatcher I decided that releasing source was more valuable than trying to foil hackers.

The source code I distribute also contains a nice version of the ShowInit procedure that displays the icon in the bottom left corner of your screen at boot time. Feel free to use my version of ShowInit in any INITs that you might write.

Tests

I've tried to do a reasonable job of testing. I've verified that RWatcher does indeed protect clean systems against infection by Scores and nVIR.
A detailed report of my testing is on the file named “Notes”.

More Information on Scores and nVIR

Last spring I figured out Scores and posted three notes on the UseNet news group comp.sys.mac describing what it does to your system, how to detect it, and how to get rid of it. If you haven't seen those notes please feel free to write to me at the address on the cover page and I'll send you copies.

I'm not an expert on nVIR, but I have done a little bit of experimenting with it. More details on nVIR are in the “Notes” file.

A Final Word

Both Scores and nVIR have been around for quite some time now (almost a year). Despite tremendous publicity and the existence of quite a variety of good virus fighting tools, they both continue to spread widely and rapidly. I have a bulging file folder filled with correspondence from people asking for help fighting infections. The problem appears to be especially acute in Universities. I've gotten messages from many people at major Universities telling me that they're continuing to experience both nVIR and Scores infections. My University (Northwestern) has experienced a number of infections in different labs and offices in recent months.

Please take the time to protect yourself. An ounce of prevention really is a worth a ton of cure in this case. It only takes a few minutes to install Vaccine and/or RWatcher, but it can take hours to clean up an infected system, and I recently spent an entire weekend disinfecting lab startup disks, servers, and computing center staff machines.

By protecting your machine(s) you will save yourself much grief, and more importantly you will help slow the spread of these viruses. Install Vaccine and use it religiously. If you're a non-MPW programmer and refuse to use Vaccine, use my RWatcher INIT. Or use both Vaccine and RWatcher.

I also recommend that you get a copy of Interferon 3.1 and run it periodically to make sure that your system is clean. It's a good idea to run Interferon just before you do file backups. Interferon will detect both Scores and nVIR infections. Avoid Apple's Virus Rx detection program—it doesn't properly detect nVIR!

If you do get infected, you can use KillScores and/or Ferret to get rid of Scores, or AntiPan and/or Vaccination to get rid of nVIR.

These virus-fighting tools can be found on CompuServe, AppleLink, Internet Mac archive sites, and most good bulletin boards. If you have access to TCP/IP, use anonymous FTP to access the Mac archives at rascal.ics.texas.edu. Rascal has a very good collection of up-to-date virus-fighting tools and information.

If you manage a University lab or other Mac lab it's even more important that you protect your machines. If you don't install Vaccine on all of your startup disks you're asking for trouble. Also educate your users, and get them to practice “safe hex”.