Mac Cracking, Part II: Net Watchman 2.1

Part II in an ongoing series

by The Observer, 11/20/95

The Scenario...

The AG Group is a company which makes a variety of network tracking products, such as EtherPeek, TokenPeek, and one called Net Watchman. Net Watchman keeps tabs on each machine on a network and alerts the user when specified conditions arise on a particular machine. These products are all made available free on the net, with one (among others) minor problem: They only work for ten minutes.

Unfortunately, Net Watchman's demo has two other limitations--it won't print, and it deletes its settings files when it quits. After many of my hours spent on this, oleBuzzard was kind enough to ask around and found that no one else had cracked past these, either. This made me stop feeling quite so weak and tiny. After a while, I decided I wasn't getting anywhere with the other two problems, and decided to release what you could fairly call a half-assed crack. Oh well. Hopefully it will still be interesting.

AG Group demos can be found at ftp.aggroup.com. The demo isn't included with this file since it's a bit chunky to include for the hell of it. What is included is a patcher to enact the crack I describe here. I hope most people will do it the "real" way though.

Tools...

Resorcerer, Resorcerer, Resorcerer, MacsBug

Planning...

Readers following the MacCracking series may remember the previous file, on a program called Relax. Relax put up a nagware box which stuck around for a few seconds before letting you get on with your life. All we wanted to do was get rid of this box.

Now things are a bit more complicated, centering around one problem. Net Watchman uses three procedures: DoNoteAlert, DoStopAlert, and DoCautionAlert to draw dialogs similar to the one which comes up telling you to fuck yourself and buy the program. The string to display is passed to these procedures. So the idea of finding a DLOG to kill doesn't work here.

First Shot...

What about looking for the STR# we want to kill instead? Ooh, an idea. The string that shows up to announce your time is up is ID 136, or 88 in hex. Every one of the 14 CODE segments has a million of 88's in it. Time to think of something else.

Next...

Why don't we look for _ExitToShell's. There are actually a little bunch of these, but not so many we can't look at them all. We find two likely

candidates: One in a procedure called Terminate, the other in DoStopAlert.

We flip a coin--terminate. Swap the _ExitToShell in terminate to NOP, and run the program until it's supposed to quit. It quits normally. The only thing terminate does is quit when you do it from the menu, apparently (as this does break the program when you do that).

OK, DoStopAlert. This one looked really really promising. Again, we use the trial-and-error method of setting the _ExitToShell here to NOP and waiting to see what happens. Damn--quits normally.

Big Guns...
OK, I'm sort of floundering here. If the program quits at an _ExitToShell, it's not an obvious one. And it doesn't even have to use _ExitToShell--a better way of quitting anyway is to just have the code end.

So let's go into MacsBug. We put a breakpoint on the DoStopAlert procedure (BR command) and watch what happens when we run out of time. Not a peep from MacsBug.

Try again, but breaking on the DoNoteAlert procedure. Ding! MacsBug breaks when the out of time dialog comes up. The DoNoteAlert procedure is a part, though clearly not all, of the process to force a quit after ten minutes.

Slowly but surely...
CODE 2 has been the focus of our investigations until this point, simply because it contains all three Do____Alert procedures, as well as terminate and other important things. We search for places in CODE 2 that DoNoteAlert is called.

Ooh! There's one! At offset 54, in main.

```
move.w    #$000B,-(sp)
jsr       DoNoteAlert
clr.l     -(sp)
```

Main has a somewhat startupy ring to it, and coincidentally a nag box does come up when you open the demo. Let's NOP the sucker. And it works! No more startup nag box. The time limit is still in effect, though.

Before leaving main, we notice something else--a call to _TickCount. A way to time something if there ever was one. No sooner is _TickCount called, than something's address is placed into register a5:

```
_TickCount           ; TB trap
move.l    (sp)+,-$138E(a5)
jsr       EventLoop
```

We also see the logic then jumps into the EventLoop procedure. Let's follow it!

Mount Elegance...
Something one needs in Mac programming is a loop that keeps asking the

computer what the user's up to. Is the jerk clicking, typing, what? One could call this an event loop, and, as we've noticed, a procedure EventLoop exists here. And it even contains a call to _TickCount.

This time that _TickCount is called, something is moved into d0. The next step is to subtract a5 and d0, compare the results, and branch depending on the result. If we don't branch, two instructions down the line is the evil _DoNoteAlert:

```
_TickCount          ; TB trap
move.l    (sp)+,d0
sub.l     -$138E(a5),d0
cmpi.l    #$00008CA0,d0
bcs.s     EventLoop+$36
move.w    #$000B,-(sp)
jsr       DoNoteAlert
```

So let's tell it to always branch.

Patience...
We wait 10 minutes... 10:01... 10:05... 10:10... 10:30... 11 minutes. Yup, this sucker's cracked. (Incidentally, one of the reasons this was an annoying crack was that I had to wait 10 minutes to test if I had found what I wanted. Sorry to bitch, just wanted to say that.)

That's it.
And that's Basic Mac Cracking number 2. Not fantastic, but something. If anyone finds a demo cracked to get rid of the printing and deleting limitations, or a full copy... Please email me so we can figure out the differences. I'm dyin' to know. Something that makes the deleting so weird is that there are three calls, FSpDelete, PBHDelete, and HDelete, that delete files--and I can't find any of their traps in the code in the code. Spooky. I should take this opportunity to congratulate the folks at AG Group for knowing their shit.

Until next time...

Comments, questions, suggestions, corrections (hey! that rhymes!)--Observer on k0p, or an407599@anon.penet.fi.

Sorta funnyish quote of the issue...
Caller: "A friend of mine gave me your software, and I'm missing one of the manuals..."