11

Where the Wild Things Are: A Look at the Logs

When we first turned on extensive logging on our gateway host, it was much like moving into a new house with a tin roof in a hail-prone neighborhood under several large oak trees during acorn season. At first, the din was deafening and distracting. We diligently chased down evil-doers and helped secure numerous open systems. Many students got their hands slapped, and a few visited their friendly local deans. Some corporate machines were secured as well.

After a couple years of this, we no longer hear the cacophony. The attacks are routine and mostly boring: the noise rarely contains an interesting signal. Most attacks have only statistical interest at this point.

We do get a daily briefing of the previous day's attacks by email. A sample of our first briefing style is shown in Figure 11.1. This report was wordy and clumsy, and we have fixed that in the latest gateway installation. It has a lot of useless information and lacks something important: the source host for the `/etc/passwd` retrieval. We've learned that single-line entries facilitate further analysis by standard UNIX tools:

```
Sep  6 05:26:21 ftpd[15391]: Sent /etc/passwd (629) to iiufmac01.unifr.ch
```

Our logs generate megabytes of information daily. We use shell scripts and simple UNIX tools to seek out Evil behavior. A single daily report usually suffices. Even diligent examination of these daily summaries can miss slow, carefully paced attacks over weeks or months. Most hackers don't appear to have the patience to wait this long. Why should they? There is no personal risk of detection when the probes are routed properly. Of course, he or she may spook the target's administrator.

Some have suggested that this is a job for an artificial intelligence system. These systems gather reports from many hosts and attempt to detect inappropriate behavior across a number of hosts. We think these tend to be more trouble than they are worth.

```
*** Last two day's warning log:
Sep  6 03:09:47 named[120]: No root nameservers for class 100
Sep  5 20:34:19 named[120]: No root nameservers for class 109

*** Last two day's unsuccessful logins:
12:54:24 Failed login  ajgˆHˆHˆHguard, CALLER=phoenix.Princeton.EDU
20:50:15 Failed login  guargˆHd, CALLER=quasar.ctr.columbia.edu
22:45:10 Failed login  gurad, CALLER=age.cs.columbia.edu

*** FTP: passwd/group fetches:
05:26:17 -------> RETR groupˆM
05:26:17 <--- 150 Opening ASCII mode data connection for group (49 bytes).
05:26:17 Sent file /etc/group
05:26:21-------> RETR passwdˆM
05:26:21<--- 150 Opening ASCII mode data connection for passwd (629 bytes).
05:26:21Sent file /etc/passwd

*** FTP: incoming files for the last two days.
*** FTP: directory changes last two days:
*** FTP: user names last day:
ftpd.log:Sep  6 04:24:02 ftpd[14559]: -------> USER lsˆM
ftpd.log:Sep  6 04:36:48 ftpd[14624]: -------> USER eastockeˆM
ftpd.log:Sep  6 04:38:20 ftpd[14633]: -------> USER eastockeˆM
ftpd.log:Sep  6 04:39:03 ftpd[14638]: -------> USER eastockeˆM
ftpd.log:Sep  6 04:40:33 ftpd[14644]: -------> USER ebzimmerˆM
ftpd.log:Sep  6 04:41:30 ftpd[14646]: -------> USER zbzimmer@ztl.chˆM
ftpd.log:Sep  6 04:43:08 ftpd[14651]: -------> USER zbzimmer @ ztl.chˆM
ftpd.log:Sep  6 05:00:41 ftpd[14773]: -------> USER masteinm@ztl.chˆM
ftpd.log.Sun:Sep  5 11:43:04 ftpd[6641]: -------> USER anynomousˆM
ftpd.log.Sun:Sep  5 14:42:35 ftpd[8109]: -------> USER kxa4244ˆM
ftpd.log.Sun:Sep  5 15:59:15 ftpd[8904]: -------> USER guestˆM
ftpd.log.Sun:Sep  5 15:59:36 ftpd[8919]: -------> USER anonynousˆM
ftpd.log.Sun:Sep  5 16:00:11 ftpd[8927]: -------> USER carlˆM
ftpd.log.Sun:Sep  5 17:51:13 ftpd[9746]: -------> USER research.att.comˆM

*** SMTP DEBUG attempts last two days

*** changes to the list of files checked:
218d217
< /n/inet/etc/named.d/ed.hup

*** changes in files:
filecheck: /n/inet/etc/named.d/ed.hup: No such file or directory
```

Figure 11.1: A sample daily briefing. People have trouble spelling "guard" (and "research"). In the early morning someone fetched our FTP `/etc/group` and `/etc/passwd` files. No one has installed or removed a directory in our FTP area for awhile. People tried a number of user names when logging into FTP. None are suspicious. There were no SMTP DEBUG attempts (there never are). The file systems have plenty of space. Someone deleted an `ed.hup` file from our `/etc/named.d` directory.

11.1 A Year of Hacking

It can be illuminating to scan long-term logs once in a while. In the next few pages we have gathered plots and summaries of the activity on INET.RESEARCH.ATT.COM, a.k.a. RESEARCH.ATT.COM during 1992, which was the full first year we gathered the more sophisticated logs. In January, Cheswick first gave his "Berferd" talk at Usenix, and the resulting bombardment of hacking probes is evident. There were also two significant failures of the long-term logging mechanism storage during the year; obviously these gaps show up in the logs as well.

We probably get more probes than most sites for several reasons:

- Our host appears in `hosts.txt`, which is an easy list for hackers to use; on machines that don't run the domain name server, it is the only list.
- Our company is a highly visible and popular target. Many hackers still consider AT&T to be The Phone Company, and hence The Target.
- Our gateway work has been published and is well known among the more organized hackers. They view our gateway as a special challenge.
- The gateway machine has high visibility. It contains the *netlib* software library and is used to distribute a variety of research papers and programs. Also, when an internal AT&T user chooses to access an outside host through our gateway, the access appears to come from our gateway, not the internal host.

In the following sections we try to label probes to our system as "evil" or as doorknob-twisting. This judgment is mostly statistical. We are not mind readers. As system administrators we are entitled to judge some probes overtly. For example, it may be a violation of U.S. federal law even to try to log into one of our machines as *root* without our permission. (At least one federal prosecutor said so, although he didn't limit it to *root*.)

The point is that probes come in all flavors and intensities. A single successful one can be literally a federal case. That your host may not be probed this intensively and constantly should not be a reason for complacency. These people keep us very honest. We believe that our detection systems give us a pretty clear view of the actual probes, something most systems lack. In fact, we believe that—

> *we have never had an undetected break-in. . . .*

11.1.1 Login Logs

As anyone who has tried it will tell you, it is dangerous to log attempted logins [Grampp and Morris, 1984]. The log will invariably show some passwords as people get out of sync with the login process. But our gateway machine has provided a good environment for logging these attempts. First, we have very few legitimate users to compromise. Second, they nearly always connect from the inside somehow. If they are on the outside, they first connect through an internal machine via the guard service, then connect to the gateway by some password-free mechanism, similar to *rlogin*, but more secure. Thus, an attempted login as *ches* from an

external source means either the system administrator was badly confused or someone is trying to break in.

The Outside users provide us with three sources of desired account names: *login*, *rlogin*, and FTP. In 1992 there were 16,709 attempted logins (of which 6,319 failed), 102,842 FTP sessions, and 359 *rlogin* attempts.

A summary of the attempted *login* list is quite interesting. Here are the top entries:

```
1831 netlib      41 ^c^c        25 anon         18 guard
1256 anonymous   41 anwar       23 e            18 sync
 448 guest       40 archie      22 help         18 c
 238 ^c          38 root        22 q            18 research
  97 ^z          36 inet        21 logout       17 walk
  89 ftp         33 alur        21 public       17 asdf
  75 quit        32 gaurd       21 s            16 ?
  51 exit        30 ~.          21 ^]           16 gur^hard
  46 d           27 gurad       19 bye          16 ^z^z
  46 a           25 johnm       18 ^[           16 ^c^c^c
```

Netlib and *anonymous* are two FTP logins: people often type *telnet* instead of *ftp*. Most of the rest are typos of our guard service, or various control characters demonstrating regret and error. We don't run an *archie* server [Emtage and Deutsch, 1992], but it was reasonable for outsiders to try. The *root* login attempts are almost certainly people with evil intent, as are the *sync* and probably the FTP attempts.

A manual cull of this list revealed the following:

```
38 root          13 who          5 nobody         2 sys
18 sync          11 uucp         3 ingress        2 sysop
16 td            11 berferd      3 sysadm         2 field
14 ches          8 bart          3 system         1 adm
14 adb           8 rtm           2 log
14 dmr           5 bin           2 authenticator
```

Most of these are well-known accounts, most of the rest appear in the bogus FTP `/etc/passwd` file shown in Figure 1.2 on page12.

The *rlogin* attempts show a similar pattern. In most cases, the callers apparently typed `rlogin` instead of `telnet`. Nevertheless, we counted the 41 *guest* attempts as knob-twisting, and the 6 *root* and *sys* attempts as blatantly evil. The FTP connections again show a similar pattern.

Figure 11.2 plots the knob-twisting probes at the top, and the evil ones at the bottom. The lower line in the upper graph is the `/etc/passwd` fetch rate via FTP.

11.1.2 Finger Attempts

Originally, we considered all *finger* attempts to be nosy or worse. It soon became clear that this was a mistake. Many people use *finger* to locate a person and get their email address and phone number. We came to this conclusion from the frequency and specific nature of most of the calls.

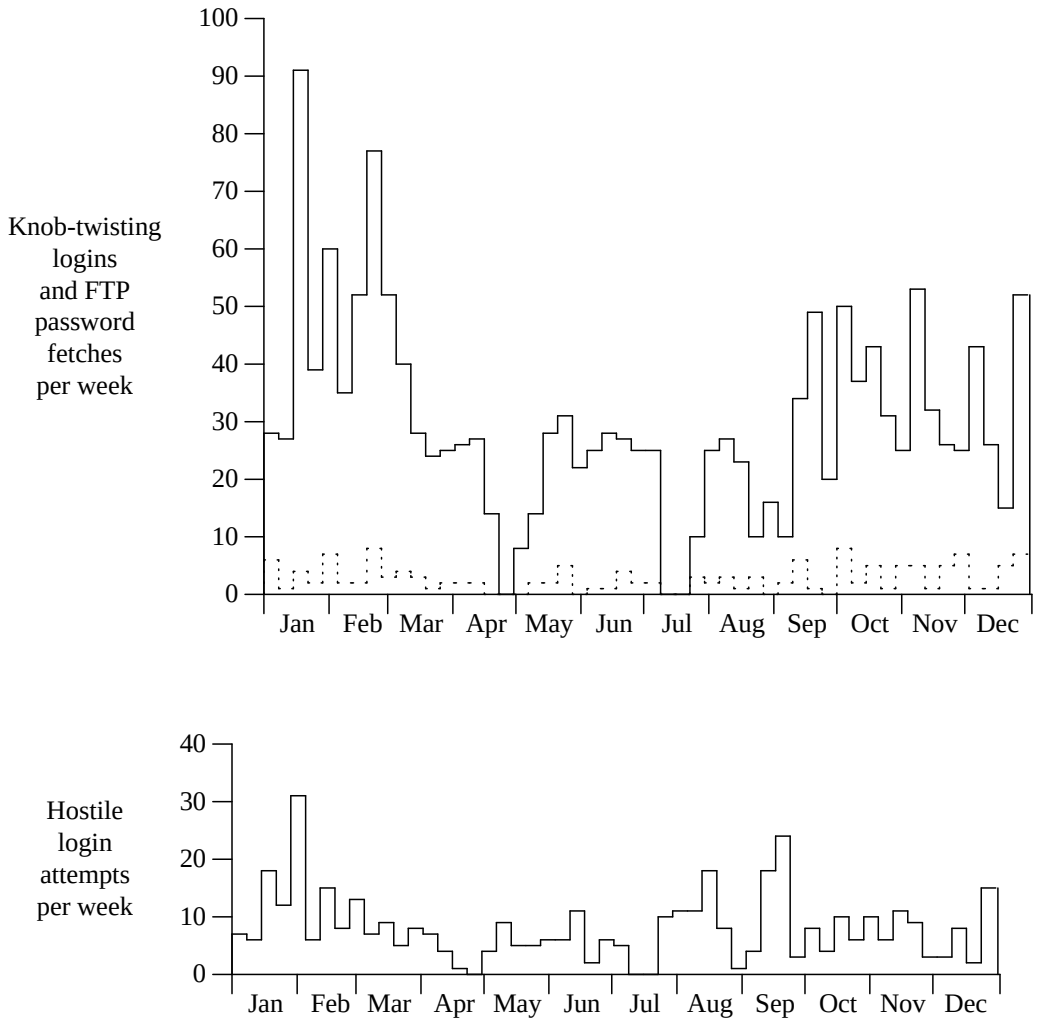

Figure 11.2: Summary of knob-twisting events in 1992. The upper graph shows FTP `/etc/passwd` fetches (dotted line) plus *guest* and similar logins (solid line). The lower graph shows hostile login attempts.

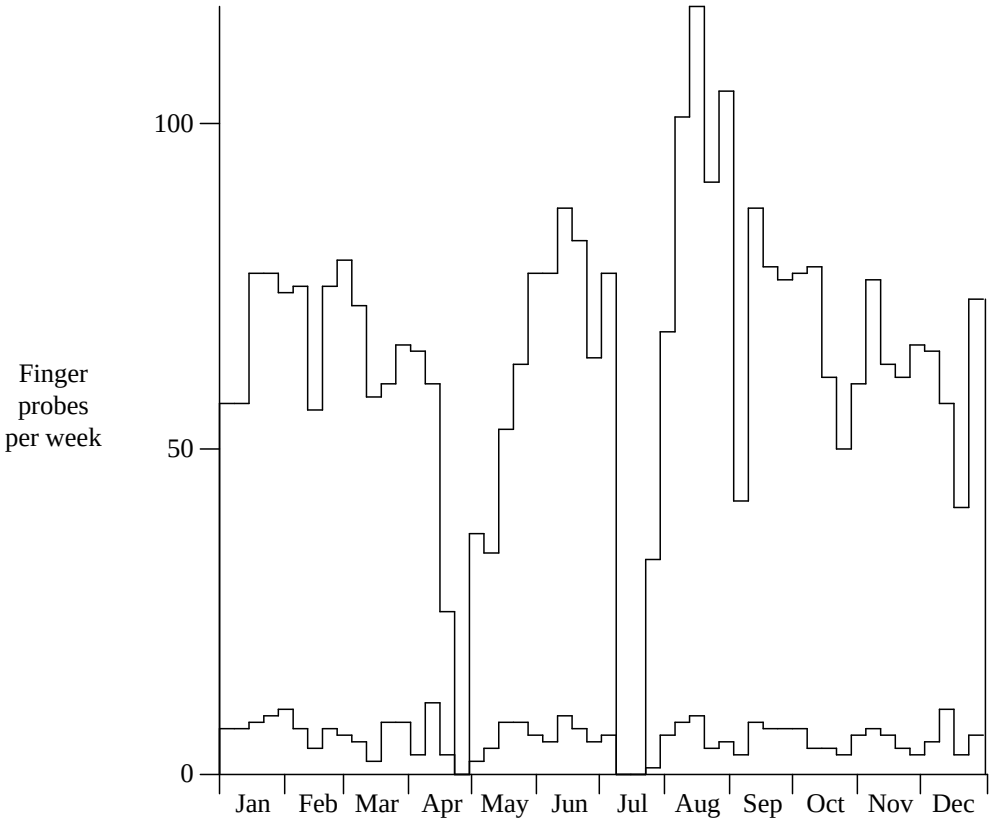

Figure 11.3: Distribution of evil *finger* requests (lower) plus other *finger* requests (upper).

Of the 11366 *finger* calls logged, 73% were for a specific login name, and the list is a who's-who of AT&T Bell Laboratories.

Of course, there was certainly monkey business as well. We have no idea how many of the *finger*s for "all" were innocent. A hand-culled list of provocative or dangerous inquiries include:

```
128 berferd          1 hello-ches       1 adm
 10 fred.berferd     1 hi_ches          1 admin
  8 root             1 berferd.         1 sys
  7 log              1 berferd.fred
  3 Fred.berferd     1 berferd.fred.
```

In Figure 11.3 we show the distribution of the fingers of "all" plus the excessively curious requests from this list.

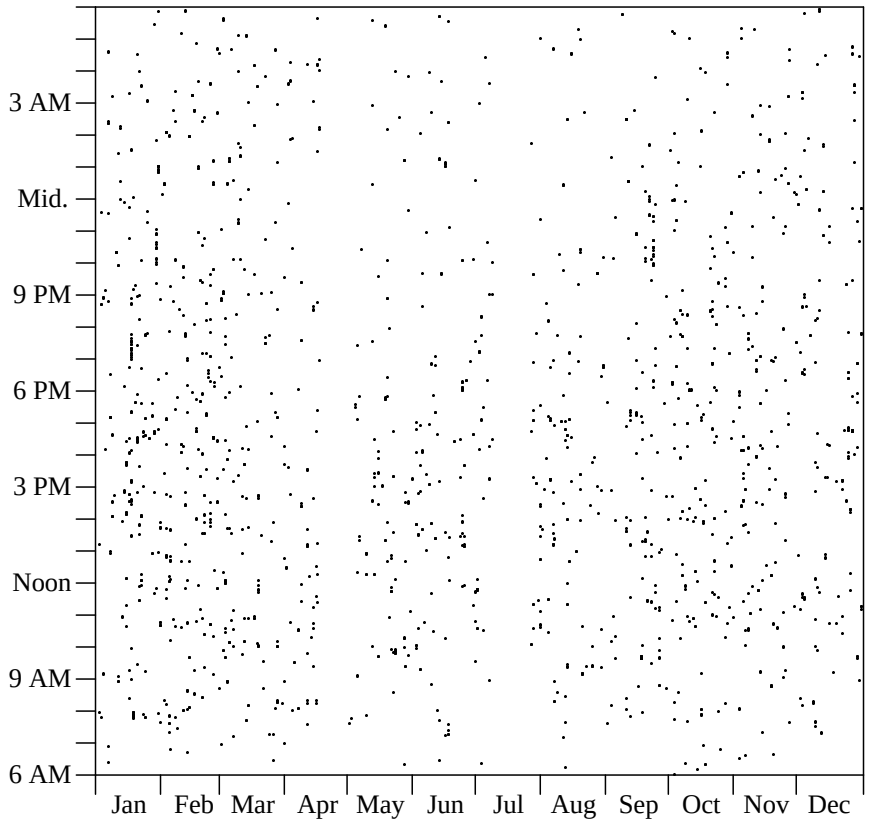

Figure 11.4: Hacker's hours: all probes for a year, logged by the time-of-day (Eastern Time). We start the hacker's day at 6AM, the slowest time of the day. Note the two monitor failures, in April and July.

11.1.3 Hacker's Hours

A couple of gigabytes of logs can be a rich source of slightly interesting statistics. Figure 11.4 shows all the mild and earnest probes of our gateway through 1992, plotted against time-of-day.

This figure shows several interesting patterns. The hours tend to correspond with an average work day in North America. With the growing number of networks abroad (see Figure 1.1 on page 5) this will probably change. The average prober does not keep the midnight hours some programmers are famous for. Apparently the average hacker is occupied elsewhere late at night during the Northern Hemisphere's warmer months.

The part of this plot that surprised us was how clearly an attack shows up as a vertical line of dots. Here again the human eye shows a fine ability to pick out patterns. It is possible that

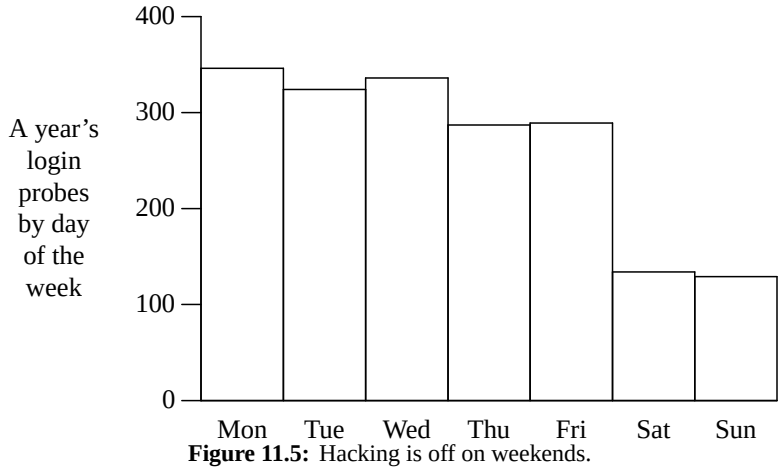

Figure 11.5: Hacking is off on weekends.

crontab-related automated probes might show up as a horizontal line, but none are evident here.

We have always suspected a link between the attacks and the academic school year. We've thought we noticed increased probing during exam week and early in the school year. This plot doesn't seem to support this observation.

Figure 11.5 clearly shows that the average hacker takes the weekend off.

11.1.4 Other Probes

Our logs show numerous other probes. Below are the commands various folks tried to execute via *rsh*. We've elided simple misguided attempts to use our gateway services such as *netlib*, and the huge environment variable list emitted for X11 commands.

```
4  who
4  xauth merge - ;
        exec xterm -ls -n research.att.com
2  cat /etc/passwd
2  finger
2  ls
1  csh -bif
1  domainname
1  ls /tmp
1  ps
1  rcp -t /usr/chou
1  whoami
```

For the year we logged 272 accesses from 171 sites to various TCP and UDP ports, plus the *portmopper*. The service list includes:

Table 11.1: Proxy Usage Summary for 1992.

Service	Connects	Outgoing Volume	Incoming Volume
ftp/data	190,367	2,182,110,906	15,426,729,499
telnet	51,135	48,444,067	1,826,136,513
ftp	46,623	10,928,129	39,620,642
uucp	18,018	528,119,019	5,615,245,849
finger	10,426	58,643	7,175,446
netb	7,361	62,292,041	122,704,920
weather	2,956	6,955,415	135,584,335
dns	1,066	42,625	885,984
3001	852	463,257	6,876,591
hostnames	635	4,396	412,673,457
whois	530	5,019	446,565
xnetlib	488	145,800	1,100,667
2592	292	1,072,032	117,906,012
dragonmud	209	117,574	20,067,071
smtp	161	5,932	118,539
callsigns	140	262,667	9,971,001
webster	111	6,623	186,115
login	58	199	90
X	47	4,169,752	1,513,048

```
31 ?              11 gopher(70)      1 1669
31 whois(43)       4 uucp(540)       1 conference(531)
26 nntp(119)       4 nfs(2049)       1 supdup(95)
24 rstatd(RPC)     3 systat(11)      1 nameserver(42)
23 rusersd(RPC)    3 unknown         1 auth(113)
23 ypserv(RPC)     3 xdmcp(177)      1 bootp(67)
22 x11(6000)       2 tcpmux(1)       1 hostnames(101)
21 mountd(RPC)     2 testing
19 tftp(69)        2 sprayd(RPC)
10 snmp(161)
```

11.2 Proxy Use

The proxy logs from our circuit-level service give us a bit more information than the IP accounting available from a router. The circuit log gives us the number of bytes transferred for each connection.

A summary of the most common proxy connections is shown in Table 11.1. The `weather`, `dragonmud`, `callsigns`, and `webster` services are supplied by a few specific machines and are not general or registered services. We were unable to classify connections to ports 2592 and 3001: we suspect they were games that are no longer available.

Table 11.2: Some Probe Sources by Domain in 1992.

200	edu
38	com
6	gov
4	net
3	mil
2	org
19	ca
10	de
7	uk
5	au
4	nl
3	il
3	ch
2	fi
2	fr
2	gr
2	tw
1	at, es, hu, ie, it, jp, kr, no, nz, sg, ve, za

It is not surprising that many more bytes are imported than exported using the proxy mechanism, because data tends to flow toward a user and all these calls have the user on the inside. The packet volume reports from our regional network provider show that we actually export quite a bit more than we import. This is certainly due to the popularity of the *netlib* service and the other files available for anonymous FTP from our gateway.

It occurred to us while looking up some of the unusual services requested through proxy that such an examination of logs might reveal sensitive personal information, even if the contents of the connection are not recorded. For example, one might consider a connection to a special port that provides a suicide hotline. Similarly, our mail logs do not record the contents of messages, but a traffic analysis of sender/receiver pairs could reveal personal information.

11.3 Attack Sources

The distribution of some attack sources in 1992 is shown in Table 11.2. You can see that most probes came from educational sites.

Recently, other sites such as TAMU [Safford *et al.*, 1993b] report that an increasing number of probes are coming from commercial sites. We suspect that this reflects the growing number of companies whose business is providing access to the Internet.

The distribution is highly nonlinear; a few sites account for a high percentage of the misbehavior we see. One should not conclude that the attackers are actually at those sites: the first thing a

Table 11.3: Frequency of Attacks During February and March 1992

Incident	Number
guest/*demo*/*visitor* logins	296
*rlogin*s	62
FTP password file fetches	27
NNTP	16
portmopper	11
whois	10
SNMP	9
X11	8
TFTP	5
ARP checks	4
systat	2
NFS	2
Number of evil sites	95

hacker learns is how to hide his or her trail. One of the main goals of cracking new machines is to attain a new hacking base. Hackers trade the names of open terminal servers, security-lax sites, and conquered machines. A machine with a guest account is a fine base for a hacker.

One persistent offending site also hosts a well-known source archive accessible via NFS. We wonder if there is a connection. We also wonder about the integrity of the code in the archive. Have any Trojan horses been planted?

Universities often have liberal access policies or inexperienced system administrators. It can be hard to track down a few antisocial students among the thousands on campus. Public PC and workstation labs are open to almost anyone.

Most of the remaining attacks come from various places outside of the United States. We can always tell when a new country connects to the Internet when we encounter a domain we haven't seen.

Sometimes attacks come from users at companies, government agencies, or (rarely) the military. We have found these sites are usually acutely interested in hacker activity and their own security. Holes at these sites are quickly plugged.

On rare occasion we have been probed on the outside by callers from within AT&T. The gateway logs have made it easy to investigate these probes and identify the culprit.

Table 11.3 shows the frequency of probes during February and March of 1992. The "ARP checks" indicate an address space probe judged to be suspicious enough to log; the other entries are based on a count of the automated trap messages generated. The FTP and TFTP entries are of particular interest, since they are rarely, if ever, innocent. Other incidents, i.e., the *whois* connections, a few of the *portmopper* traps, and the SNMP messages, turned out to be benign.

```
From: adm@research.att.com
To: trappers
Subject: udpsuck nfs(2049)

UDP packet from host a.non-us.edu (173.46.173.146): port 804, 40 bytes
    0:   2964e5a6 00000000 00000002 000186a3   )d..............
   16:   00000002 00000000 00000000 00000000   ................
   32:   00000000 00000000                     ........
/usr/ucb/finger @173.46.173.146 2>&1
[173.46.173.146]
Login       Name              TTY Idle    When    Where
lu       Lee User              a  8:41 Fri 12:55  direct to room 101
ano      A.N. One              h6   3d Tue 00:49  direct to 719
nsa      Nun Atall             p0   36 Thu 18:56  eqg01:0.0
nsa      Nun Atall             p1   24 Thu 18:57  eqg01:0.0
```

Figure 11.6: A captured NFS request.

The essential fact, though, is that the Internet can be a dangerous place. Individuals attempted to grab our password file at a rate exceeding once every other day. Suspicious RPC requests, which are difficult to filter via external mechanisms, arrived at least weekly. Attempts to connect to nonexistent bait machines occurred at least every two weeks. It is worth noting that during the Berferd incident, we attempted, without success, to lure the intruders to that machine, which actually existed at the time. Now, connection requests have become commonplace. We do not know if there are that many more hackers or if they have simply gotten more sophisticated in their targeting.

11.4 Noise on the Line

Of late, it has become more difficult to distinguish sophisticated legitimate programs from attacks. Consider the log message shown in Figure 11.6. It shows an NFS `NOP` packet. When we first saw such a packet, we were extremely suspicious. Normal NFS operations must be prefaced by a negotiation with the remote mount daemon; since we do not run a real version of the latter, occurrences of the former seemed to be quite out of line. We suspected an attack via forged NFS requests, a scenario we believe to be quite possible. Reality was rather more complex.

It turns out that a number of Internet public archive sites are accessible via NFS. The convenience of such an approach is undeniable, though the security of it is debatable. Our problem is with the implementation, coupled with inadequacies in our current monitoring tools. It turns out that the *amd* automounter [Pendry, 1989] will generate a `NOP` packet before attempting to contact the mount daemon. If NFS is not running, there is no reason to try to mount the file system. These are the requests that we have been seeing.

Their frequency has become worrisome. Given the existence of public NFS archives, checking to see if we offer such a service cannot be considered a hostile act. On the other hand, what we

see with our current tools—NFS `NOP`s and queries to the mount daemon—are not distinguishable from a genuine attack. Our choices are either to ignore all such requests or to emulate more of the protocol, so we can see what is really intended. Neither alternative is appealing.

Other forms of noise are more annoying than confusing. A remarkable number of people seem to type *rlogin* when they mean *ftp* or *telnet*. We see several attempts—and log messages—when users try to connect to our machine using their own logins.

Similarly, a number of people think that if we support anonymous FTP, we should also support anonymous *rcp*. But *rcp* is implemented using the *rsh* protocol, and we do not wish to expose our system to that extent. Nevertheless, the requests are not worrisome; the originator's benign intent is fairly obvious.

We should add a caveat here. A clever intruder, knowing of this strategy, could launch innocent-appearing probes that would be ignored either by the monitoring software or by the administrators who use it. Only after a successful response, and no follow-ups, would an actual penetration be attempted. For example, a hacker who knew of an NFS hole could send a `NOP` first. If the system responded, and if there were no signs of counterintelligence probes or administrative messages, the full-scale attack could be launched. If someone did object to the `NOP`, the hacker would have a perfect cover story.

Most of the probes we have recorded by our SNMP monitor have been similarly innocuous. It seems that certain popular network management packages want to manage, or at least know something about, all of the other hosts to which their own machine talks. They do this by querying certain services, including SNMP, on the remote host, and present the information to the humans using the package. Unfortunately, these tend to ring several of our alarms. Thus far, at least, the spoor of such occurrences has been quite distinctive.

The same is true for most of the X11 connections we have seen. Although potentially quite worrisome—an attacker wielding an X11 hacking tool could dump windows, monitor keystrokes, and possibly even inject synthetic input—very few of the occurrences thus far have been suspicious. The usual situation is that someone from behind our firewall has dialed out to a remote machine and fired up the *emacs* text editor; it in turn decides to open up an X11 window on the user's home machine. But that machine is hidden behind the firewall, and our gateway machine—the apparent source of the *emacs* session—logs the connection attempt.

Another source of noise is quite different: it is the legions of people around the Internet who try to log in to our gateway as *guest*. As noted, this does not work, but it does generate an annoying mail message. We rarely bother to ask that the attempts stop; some users, however, are quite persistent and will retry the *guest* account ad infinitum.

Finally, some of our monitors pick up attempts to connect to harmless, and even useless, services. For example, we know of no reason for anyone to connect to the *tcpmux* port [Lottor, 1988]—vendors rarely even support it. But we have seen attempts to connect to port 1 on our system. Other such incidents are discussed in [Bellovin, 1993].