Logging

Shadow Hawk

Shadow Hawk Busted Again

As many of you know, Shadow Hawk (a/k/a Shadow Hawk 1) had his home searched by agents of the FBI. . .

When he was tagged by the feds, he had been downloading software (in the form of C sources) from various AT&T systems. According to reports, these included the Bell Labs installations at Naperville, Illinois and Murray Hill, New Jersey.

—Phrack Issue 16, File 11, November 1987

How was Shadow Hawk Detected?

- He had broken into some Bell Labs machines
- He tried to use `uucp`—a dial-up file transfer/email/remote execution package that came with Unix—to grab `/etc/passwd` files from other machines
- Uucp logged all file transfer requests
- Several people at Murray Hill had automated jobs that scanned the log files for anything suspicious
- (What did we look for?)

What Did We Look For?

- Requests denied for security reasons
- Anything mentioning `/etc/`
- Attempts to execute commands for functions other than email and netnews

Stalking the Wily Hacker

- An accounting file didn't balance—a username had been added without the proper bookkeeping entries
- Cliff Stoll noticed and tried to figure out what was going on
- Ultimately, it led to a KGB-controlled operation aimed at military secrets. . .
- (It also led to a *Communications of the ACM* paper, a book, and an episode of *Nova* for Cliff. . .)

An Evening with Berferd

```
19:43:10 smtpd: <--- 220 inet.att.com SMTP
19:43:14 smtpd: -------> debug
19:43:14 smtpd: DEBUG attempt
19:43:14 smtpd: <--- 200 OK
19:43:25 smtpd: -------> mail from:</dev/null>
19:43:25 smtpd: <--- 503 Expecting HELO
19:43:34 smtpd: -------> helo
19:43:34 smtpd: HELO from
19:43:34 smtpd: <--- 250 inet.att.com
19:43:42 smtpd: -------> mail from: </dev/null>
19:43:42 smtpd: <--- 250 OK
19:43:59 smtpd: -------> rcpt to:</dev/^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H
19:43:59 smtpd: <--- 501 Syntax error in recipient name
19:44:44 smtpd: -------> rcpt to:<|sed -e ’1,/^$/’d | /bin/sh ; exit 0">
19:44:44 smtpd: shell characters: |sed -e ’1,/^$/’d | /bin/sh ; exit 0"
19:44:45 smtpd: <--- 250 OK
19:44:48 smtpd: -------> data
19:44:48 smtpd: <--- 354 Start mail input; end with <CRLF>.<CRLF>
19:45:04 smtpd: <--- 250 OK
19:45:04 smtpd: /dev/null  sent 48 bytes to  upas.security
```

What was the Common Thread?

- Log files of various sorts
- “Extraneous” information
- Log files can prevent problems, help you figure out how the system was penetrated, what was affected, and—if you’re lucky and persistent—who did it

Where Do Log Files Come From?

- Many different system components can produce logs
- Often, these aren’t enabled by default
- Should they be?

Web Logs

- Here’s an entry from one of my web server logfiles:

```
aaa.bbb.ccc.ddd - - [20/Nov/2005:22:32:15 -0500] "GET
/1e/chap02.pdf HTTP/1.1" 200 215350
"http://www.wilyhacker.com/1e/" "Mozilla/4.0 (compatible;
MSIE 6.0; Windows NT 5.1; SV1; Tablet PC 1.7;
.NET CLR 1.0.3705; .NET CLR 1.1.4322; FDM)"
```

- Let’s look at it piece by piece

Web Logs

- IP address and timestamp
- Hmm—what time zone?
- The actual command transmitted, plus the response
- Lots of information about the client
- Hmm—web clients tell a lot about themselves. . .

Aside: Web Browsers Disclose a Lot

```
GET / HTTP/1.1
Host: localhost:8000
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.8;
    rv:25.0) Gecko/20100101 Firefox/25.0
Accept: text/html,application/xhtml+xml,application/xml;
    q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
```

But They Can Lie

```
GET / HTTP/1.1
Host: localhost:8000
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0;
    Windows NT 5.1)
Accept: text/html,application/xhtml+xml,application/xml;
    q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
```

From a security perspective, a server can't trust anything a browser (or any other client) says…

Detecting Problems Via Logfiles

The ”Code Red” worm activity can be identified on a machine by the presence of the following string in a web server log files:

```
/default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNN%u9090%u6858%ucbd3%u7801%u9090%u68
u7801%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u
b%u53ff%u0078%u0000%u00=a
```

From `http://www.cert.org/advisories/CA-2001-19.html`

An Attempted Intrusion?

```
[Sun Nov 20 23:17:18 2005] [error] [client www.xxx.yyy.zzz]
File does not exist: /usr/pkg/share/httpd/htdocs/xmlrpc
[Sun Nov 20 23:17:28 2005] [error] [client www.xxx.yyy.zzz]
File does not exist: /usr/pkg/share/httpd/htdocs/phpgroupware
```

(There were many more attempts from that IP address.) Both of these represent services with known security holes

It Still Happens

- Logs from the last few days show different attempts to find web holes
- Probes came from China, Germany, Indonesia, Japan, UK, Jamaica, and the US
- Targets probed include several PHP scripts, some mysql content, and what appear to be WordPress scripts

```
[error] [client aaa.bbb.ccc.ddd]
File does not exist:
/usr/local/www/fw-insec2/wp-admin
```

I Spotted Several Variants of This. . .

```
/cgi-bin/php4?%2D%64+%61%6C%6C%6F%77%5F%75%72%6C%5F%69
  %6E%63%6C%75%64%65%3D%6F%6E+%2D%64+%73%61%66%65%5F
  %6D%6F%64%65%3D%6F%66%66+%2D%64+%73%75%68%6F%73%69
  %6E%2E%73%69%6D%75%6C%61%74%69%6F%6E%3D%6F%6E+%2D
  %64+%64%69%73%61%62%6C%65%5F%66%75%6E%63%74%69%6F
  %6E%73%3D%22%22+%2D%64+%6F%70%65%6E%5F%62%61%73%65
  %64%69%72%3D%6E%6F%6E%65+%2D%64+%61%75%74%6F%5F%70
  %72%65%70%65%6E%64%5F%66%69%6C%65%3D%70%68%70%3A%2F
  %2F%69%6E%70%75%74+%2D%64+%63%67%69%2E%66%6F%72%63
  %65%5F%72%65%64%69%72%65%63%74%3D%30+%2D%64+%63%67
  %69%2E%72%65%64%69%72%65%63%74%5F%73%74%61%74%75%73
  %5F%65%6E%76%3D%30+%2D%6E
```

That decodes to

```
/cgi-bin/php4?-d+allow_url_include=on+-d+safe_mode=off
    +-d+suhosin.simulation=on+-d+disable_functions=""+
    -d+open_basedir=none+-d+auto_prepend_file=php://input+
    -d+cgi.force_redirect=0+-d+cgi.redirect_status_env=0+-n
```

They're trying to run some nasty PHP in non-safe mode. . .

Problems with Log Files

- How did I spot those probes?
- Manual search through error_log
- Not very scalable...

Log File Scanners

- Need to automate scans
- Pick out “interesting” events
- Hmm—what’s interesting?

Log Files and Intrusion Detection

- Analyzing log files like that is a form of intrusion detection
- Can look for specific signatures, such as examples above
- Or—can look for anomalous patterns, such as too many misses or too-long URLs

Correlating Log Files

- Sometimes, the interesting information is spread among several log files
- Need accurate timestamps for correlation between machines
- Timestamps should generally be in UTC, rather than the local timezone

Types of Correlation

- Intra-machine—different forms of logfile
- Intra-site
- Inter-site
- Watch out for privacy issues!

Types of Logs

- Application-specific: web, email, etc.
- Generic OS logs
- Network element logs

Sliced Another Way

- Routine processing
- Error messages
- Authentication events and/or errors
- Access control events and/or errors

Processing Logs

- Primary rule: retain raw data as long as possible
- Be suspicious—log files often contain enemy-supplied data
- Be *especially* careful if you use a web browser to look at log file data: loggable data has included things like Javascript-based pop-ups
- Crunch log down to manageable size
- Pick out interesting items

Storing Logs

- Flat files work for small sites
- Databases scale better—and allow for sophisticated queries

☞ Probably the best answer for most places

- But—watch out for “Little Bobby Tables”

Network Logs

- Can you do full-traffic logging?
- Suppose your Internet link is 100M bits/sec, or 12.5M bytes/sec
- Assume you run the link at 40% capacity == 5M bytes/sec
- On a 2TB drive, you can store 400,000 seconds, more than 4.5 days
- An 8TB drive costs $< \$300$, quantity 1—this is affordable even for a 1G bps link
- That looks feasible
- But—can you extract any meaning from that much data?

Suppose Your Site is Penetrated

- A full-traffic log can show you what was done
- Well, maybe it can't—did the bad guys use encryption?
- Such logs may or may not be helpful
- They're a *tremendous* privacy risk

Network Connection Logs

- Many routers can produce “flow logs”
- A flow is (roughly speaking) a TCP connection
- This sort of traffic analysis can reveal many types of attacks

Limits of Full-Traffic Logs

- Capturing every packet is hard
- If you want, say, URLs accessed from your site, it’s easier to use a web proxy log
- Must prevent direct web access—block ports 80 and 443

Security and Logfiles

- What sorts of security do logfiles need?
- Confidentiality? Integrity? Availability?
- All of them!

Confidentiality

- Logfiles can contain sensitive data
- Again, watch out for personal privacy
- Besides, you don’t want to tip off the attacker about what you know

Integrity

- Make sure the enemy can’t tamper with your logs
- Prime target for many hackers!
- Absence of log file entries is not evidence that nothing happened

Availability

- Attack: fill up log area with innocent garbage
- When the log file is full, launch the real attack
- Some systems will lock up if the log files are full—easy denial of service attack against the whole system!
- Which is better—no logs, or no processing?

Storing Log Files

- Local machine—easy, but vulnerable to attackers
- Use a log server
- Note: log server must authenticate log requests

Secure Logs

- Scope: assume no high-bandwidth connection to log server
- Protocols exist for secure logging
- Limit attacker’s ability to read or corrupt log files

Creating Good Logfiles

- What should be in a log file?
- How is functionality divided?
- Remember—logfiles are parts of *systems*

What’s in a Logfile?

- Timestamp—when did it happen?
- What happened? To what resource or resources?
- Who did it?
- What “session” or network connection did it come from?

Logfiles and “Sessions”

- The accounting log on an old desktop of mine says:

  ```
  sh          -S      root     __          0.61 secs
  ```

  (timestamp omitted)
- That is—**`root`** ran a shell, which in turn performed some operation that only root can do
- Was it me, in a root window? A system daemon? An attacker?
- If it was an attacker, was there a network connection involved? From what IP address?
- In this case, the **`--`** is supposed to be the pseudo-tty, so it’s not from a window—but the process can set it to null, so it’s not trustworthy.

Things to Log

- Desired log level varies; tradeoff between comprehensiveness and space or performance
- Always log security events: permission denied, `su`, use of privileges, authentication failures, etc.
- Always log session start/end
- Log accesses? To what? By which users?

What if You Can’t Log Everything?

- Application processes may not know enough to create proper logs
- Example: a Unix application running in remote login window doesn’t, in general, know the IP address of the client
- Solution: provide linkage information
- The remote login program logs the connection information and tty name; the application logs the tty name.
- Correlation done by the log file analyzer

Logfiles and Auditing

- Logfiles can act as a deterrent
- If people know the logfiles show what they did, maybe they won’t do bad things
- Example: the original purpose of a cash register was to produce the manager’s copy of all receipts
- How do we do such audits?

Follow Sample Transactions

- Find all log records pertaining to a transaction
- Note: implies ability to correlate
- May require extra tag fields in log messages: tag actions with “session” information
- Manually check each step
- Supplement with phone calls, phone logs, etc

The Sad Reality

- Lots of folks have very good logs
- Few of these logs are ever examined
- Many warnings are missed

The Wikileaks Data

“Weak servers, weak logging, weak physical security, weak counterintelligence, inattentive signal analysis . . . a perfect storm.’

Pfc. Manning, convicted of the leaks

What Went Wrong?

- Manning brought in CDs—regulations prohibit “personally owned computers and associated media” in secure facilities
- No one objected
- No one checked the logs for how much he was downloading
- Possibly bad personnel security—should he have received a clearance? Were there changes in his behavior that should have been noticed?

The Snowden Files

- The public doesn't know what he took, or even how much
- According to press reports, the NSA doesn't know, either:
  - Inadequate logging?
  - Did he tamper with logs?
  - Did he disable logging on machines for which he was a sysadmin?
  - Did he use some path (e.g., directly mounted file system) that had no logging?
  - It's unclear if we'll ever find out...