Security Logging, Intrusion Detection, and Fraud

This article discusses some issues in identifying security violations, utility of computer logs, methods of faking security violations, and best practices in maintaining security.

We need to know what the plaintiffs are going to be using to substantiate their claims that Equipment Hall cracked into the plaintiff’s computer systems, and what Equipment Hall allegedly did to their systems.

As we discussed in our meeting yesterday, 17th October 2007, the plaintiff will have a very difficult time proving anything based on computer logs unless the plaintiff made copies of all logs on write-once, immutable media (e.g. CD-R) immediately, gave copies of these logs to an independent party at that time, and can prove that they have not been manipulated after the fact.

The document I saw yesterday stated that the plaintiff didn’t know how or what Equipment Hall did in accessing the plaintiff’s system.

System Log Files

Computer systems generally maintain log files which track significant events on the system. These logs are usually simple ASCII text files with a line for each event showing the time, the program that created the log entry, and information that program deems interesting (e.g. IP addresses, commands, and other information).

These system logs are primarily used by system administrators to monitor system performance and detect problems such as attempts by outsiders to gain unauthorized access to their systems.

These log files can grow very large, and are usually “rotated” daily to prevent their consuming all the free disk space on the systems. This log rotation process creates new log files daily, and keeps old log files on the system, compressed to save space. Old log files are kept for limited periods of time which may be determined by the system administrators. Often this period is a week or less.

Unless these log entries are written to immutable, write-only, media such as a hard copy printer, they are easily modified using standard text editors such as “edlin”, “notepad”, with no way of telling that the log files were changed.

Many crackers (people who gain access to other people’s systems with intent to get unauthorized access and/or do damage to the system) are well aware of the dangers of detection via the system’s log files, so they have tools that remove any traces of their presence from the system’s logs.

When any file is written or modified on a computer system, the system updates the modification time of the file. This time is not reliable as evidence as it is easily set to any desired time with a single command in common scripting languages such as python or perl which are commonly found on any Linux or Unix system.

The only way one could be sure that log files hadn’t been changed would be (a) if they written to write-once media as the logged events occurred, and (b) this write-once media were controlled by a dependable third party.

Accounting System Logs

Many accounting systems are written using general purpose SQL database programs to store all their data. While the accounting software may have strong procedures to insure data integrity, it is usually possible for somebody to bypass the accounting software by going directly to the underlying SQL program to make changes.
Faking activity by IP address

If someone wanted to make it appear that their system had been accessed by an outsider on any IP, it would be very easy unless stringent controls of the system logs were in place, and even then it wouldn’t be difficult to fake access.

The easiest way to do this would be to simply edit the system’s log files using standard text processing tools, inserting lines that made it appear that the system was accessed by somebody from their target’s IP address.

If the system and accounting logs were secure, it would be easy to simulate access by an machine with an external IP address by (a) adding a second network interface card (NIC) to an internal machine with an IP address in the same block as the external IP one wanted to use, and attach another machine to that second NIC, assigning the external IP to the second machine. After doing this, any log entries of activity from the second machine would show the external IP address one wanted to fake.

One could configure a bogus user account on the system and in the accounting software, then access the system(s) from the second machine with the faked external IP address.

The cost for the second NIC could be as little as $5.00 for a 3COM 3C905b from RE*PC. The second machine could be any laptop connected to this NIC.

Information Needed in Discovery

At this time I think the major things needed in discovery concerns what evidence they have regarding the alleged cracking attempts, what is their network configuration, and are their personnel capable of manipulating the data to make it appear that their systems were cracked.

What is their Evidence?

- Is their evidence based on logs?
- If so, how were they kept to insure that they’re accurate?
- Who is vouching for the log’s validity?
- What other evidence do they have to back up their claims?
- What systems were allegedly accessed?

What are Their Security Policies and Procedures?

1. Do they invalidate user’s accounts when employees or contractors are terminated?
2. Do they change all administrative passwords when employees or contractors are terminated?
3. How do they limit access to their systems from the Internet?
4. How much connection data are they logging?
5. To they log to write-once media?
6. How frequently do they rotate their logs?
7. How long do they keep their logs?
8. What type of software do they use to insure the integrity of their systems including anti-virus, detection of key loggers, detection of Trojan horses, etc.?
What is Their Network Configuration?

1. Do they have a current network diagram?
2. Do they have a firewall?
3. Is the firewall an appliance (e.g. Cisco, Linksys, Netgear) or a computer system?
4. Do they have a private internal LAN?
5. Is their accounting system directly accessible from the Internet?
6. If the accounting system is accessible from the Internet, how is it protected?

Accounting Software

1. What is the name of the accounting software in use?
2. Is this software currently licensed and all fees paid?
3. Does this accounting software maintain an audit trail of all activity including user name, IP address, time, and action?
4. Does this accounting software use an SQL system for data storage?
5. What SQL system is being used?
6. Does that SQL system support stored procedures?
7. Does that SQL system support triggers?
8. Is the accounting data protected by stored procedures and/or triggers that would prevent modification of transaction data, logs, or other modifications not under the control of the accounting software?

Personnel

1. What is the Skill Level of their Personnel?
2. Who has administrative access to the systems?
3. Who has access to the accounting software?
4. Who has administrative access to the accounting software?

Security Policies, Intrusion Detection, and Response

Protecting a computer system or network of systems against data loss either due to hardware failure or compromise by outsiders requires comprehensive policies and procedures which covers:

- System and data backup with off-site copies.
- Controlling access to the systems, limiting access to the people necessary to accomplish specified tasks.
- Procedures to insure the integrity of systems. These include firewalls, intrusion detection, anti-virus, and logging all system activity.
- Audit trails showing who did what, with which, and to whom.
- Defined procedures to deal with hardware failure, unauthorized access, and recovery in cases of human error.

**Backups and Recovery**

It is critical to have backups of all systems and data done often enough to allow recovery in case of system failure or compromise. Daily backups are sufficient for many businesses.

Recovery procedures are also critical, and must be understood and tested in advance of failures.

There should always be off-site backups in case of fire, theft, or other events.

**System Access Control**

Access to systems should be controlled and limited to those people who need to use or maintain different parts of the system.

Access should require strong passwords or other methods of authorization depending on the value of the systems being protected.

When an employee or contractor is terminated, their access needs to be disabled immediately, and any passwords or authorization keys they have changes or authorizations revoked in the case of SSL certificates.

**System Logs and Monitoring**

Most computer systems maintain logs of significant activities including log in transactions (including failed attempts), data transfers with ftp, etc.

**Manual Log Monitoring**

Computer logs tend to be very large on active systems so it’s critical to have procedures that watch for significant events. One may be able to pick up anomalies by examining the files manually, but this doesn’t work well if the logs are large.

**Automatic Log Watch**

We use automated procedures that watch log files for significant events including log in attempts, successful and unsuccessful, unauthorized access to system services, and changes in network interfaces (NICs) which might indicate sniffing activities. When any of these significant events appear in the system logs, the log watch program immediately sends an e-mail notice to the security address on the system. We monitor these notices continually, prepared to take immediate action if necessary (I just got 35 notices of unsuccessful attempts to log in to several of our systems while typing this paragraph and added the IP address to our block list).

**Intrusion Detection**

In addition to watching the system logs for evidence of intrusion, we also have tools to detect changes in the system which might have been made in ways that don’t appear in the logs.
This is done with software that makes and maintains a database of all critical files on the system, and sends e-mail reports of any changes found during the daily maintenance runs. The database includes information necessary to detect any changes in critical files including the program’s ownership and modes, and MD5 and SHA1 cryptographic digests of the file. This is in addition to anything the underlying vendor’s software provides that can be used to verify the integrity of system software.

The first time the intrusion detection software is run, it creates the initial database for comparison during the daily processing. Ideally, this should be done before a system is put on-line so one knows that it’s starting with a clean system.

When the intrusion detection software is run after the first time, it scans the entire system looking for new files that aren’t in the database, existing files that have changed in any way, and missing files that have been deleted from the system. If any changes are found, it sends a report via e-mail to the security address on the system.

We will often run the intrusion detection software immediately after doing software updates on a system to update the database with the new file data.

**Incident Response**

The response to e-mail notification of security events depends on the type of notice, and the threat assessment of the notice.

**Login Attempts, Failed and Successful**

If the notice is from the log watch programs indicating a failed login attempt, we look at the IP address of the system from which the attempt was made to determine if it’s a system that normally has legitimate access to the system, or an external system that probably shouldn’t be attempting to log in.

If it’s a local IP address, with very few failures, we assume that somebody fumbled their user name or password. If it’s local, with multiple failed attempts to log in to administrative accounts, we will notify the responsible people at the site to let them know that somebody on site may be trying to gain unauthorized access to the system.

If the IP address is not local to the target system, our response varies depending on the location of the system attempting to make the connection, and to the number and frequency of connection attempts.

If the connecting IP address is outside the U.S., or there is a high number of connection attempts on one or more systems, we immediately add that IP address to a DNSRBL (Domain Name Service Real-Time Block List) that will block access from that IP address to any systems here or at our customer’s sites. This prevents further log in attempts, blocks access to any other TCP based services, and blocks e-mail from the IP except messages addressed to our abuse, postmaster, security, or support addresses.

As an example of this, I just received notices of 16 attempts to connect to systems in Walla Walla from an IP that had made a single connection to a system in Kansas City this morning. The IP is U.S. based so I hadn’t done anything with it on the first notice, but just black listed it when I saw additional connections to geographically separated customer systems.

If the connecting e-mail address is inside the U.S. and/or comes from a network where I think they might actually care that their system is being used in cracking attempts, I will forward copies of the e-mail notices to the abuse address at that network telling them that they may well have a cracked system that’s being used to attack others. Typically we send notices to corporations and educational institutions.

If the log in attempt is successful, and comes from a suspicious IP address, we take immediate action to determine whether this is a legitimate connection, and if not,
we escalate to a full check of the system which will be described in the next section where the intrusion detection software had detected unauthorized changes.

**Intrusion Detection Checking**

The daily security checking software e-mails a report any time changes are found on the system that may indicate a system compromise. Some changes are normal, say when new users are added to a system or software is updated.

If there are changes in the system’s user and password files, we check to see if the changes appear to be routine changes or if they indicate cracking activity by doing things like adding accounts with administrative privileges. If there are suspicious changes, we will go into further analysis as described below.

If the report shows new files on the system, particularly those that run with elevated, setuid (Set User ID) privileges, we have to look at these more closely to see if they’re legitimate additions or something created by a cracker.

If the report shows changes in critical files, particularly those that are used to monitor system processes, files, etc., this will trigger an in-depth analysis.

If the report shows missing files, those deleted from the system but in the database, we will determine whether these deletions are normal, or indications of lost programs or data. This can be very valuable to know quickly in case someone has accidentally deleted a file, particularly one that’s used infrequently, as it allows one to restore the file from current backups.

At this point, if no serious questions are raised by the reports, we will update the critical files database with the current data, and be done until the next incident.

**Dealing with Intrusions**

When we have evidence that a system has been cracked, we take immediate action to notify the system owner that their system has been cracked, and determine if there are active processes on the system being used by the crackers, identify where they’re coming from, block access from any external IPs they’re using, kick them off the system, then clean up anything they’ve changed restoring the system to clean operation.

Depending on the nature and source of the cracking we may take further steps to track future attempts by the same people and gather evidence for later legal action.

**Locate and Kill Active Processes**

The first step is to identify on-going activity by the crackers. This is often done by examining processes using new files identified by the intrusion detection software. We will look at all resources used by those new files to determine what they’re doing, and any connections to external systems they may have open. We will also look at all open connections on the network to see if there are suspicious connections (e.g. to anything in undernet.org or other known sites frequented by crackers). If we find open connections to these systems, we will then look at the process with the connection to determine what programs and files they’re using.

Assuming that we have found connections to outside systems, we will then kill these processes, and block further access to/from the IP addresses in use by these processes.

We will then kill any other process associated with the new files. Occasionally we will find that the cracker has inserted code to automatically restart their programs, and we need to identify and disable that code.
Copy and Remove Cracker’s Files

After we have killed any active processes found, we will make a private directory on the system, and copy all programs we find related to this cracking incident to that directory for later analysis, then remove the crackers files and directories from their original location.

Check System for Changed Files

We will first check for files commonly changed by crackers that may not have appeared in the previous day’s intrusion detection report. The programs that display running processes, find files, list directories, and similar functions are frequently replaced with ones that hide the cracker’s programs.

If we find changes in these programs, we will replace them from the original installation files before doing further checking.

We will then run the vendor’s file integrity checking software to see if it detects changes in other system files than the ones we frequently find changed, replacing them as necessary.

We will then run the daily intrusion detection software manually to find any changes that have been made since the previous run (wash, rinse, repeat).

Determine How Cracker Got In

After we have cleaned the system, and are sure that all the cracker’s processes have been exorcised from the system, we then will do further checking if necessary to determine how the cracker gained access to the system (assuming that we learned of the cracking from the intrusion detection software, and not from a real-time e-mail showing the access). At this point we will have a directory dedicated to this cracking incident with all the cracker’s files we found. We will also have the output of the various commands used to identify the connections and processes used.

We will then examine ALL the recent log files on the system looking for IP addresses and file names associated with this cracking attempt, making copies of any log files where this evidence is found. This often will disclose previously unknown avenues of attack, and can be used to refine the log watching procedures to notify of future attempts via this vector.

Further Processing and Decisions

After the system is clean, and we have copies of the relevant files and logs, we have to decide whether there’s any reason to fear a future cracking attempt, and whether legal action is appropriate. The vast majority of cracking attempts are done by crackers unknown to the victim, with the primary goal of using their system to mount further attacks. There’s little chance of useful legal action against a random attack from Russia, China, or similar locations.

If the cracking/unauthorized access warrants legal action, say against a former or current employee, it will be necessary to gather and preserve evidence in a form acceptable in court. This requires taking steps to insure that any and all logs are kept in such a manner that it can be established that they have not been manipulated.