This column focuses on secure computing, providing tools and tips for those in the information security trenches. Each issue, we'll evaluate new technologies (primarily in the open source space) and discuss ways to integrate them into your organization.

We want to hear from you. Got a great utility or "magic" script that's saved you hours of tedious keyboard pounding? Something new we haven't heard about? Let us know at ciocorner@sandstorm.net.

Incident Response Toolkits

In this month's column, we're going to lay the groundwork for developing a tool to help mitigate a security breach in your IT infrastructure: an incident response toolkit. Also, we'll discuss a procedure for responding to an incident.

One of the reasons that computers have become so pervasive in our society is because electronic data is easily created, modified and deleted. However, in a criminal or civil "cyber crime" proceeding, computer data is evidence. Therefore, it can be difficult to establish the integrity of computer data. Lengthy investigations or court cases may directly impact availability — or the public trust. Additionally, the sensationalism that accompanies computer crime stories often distorts the truth in the minds of the public. If your information infrastructure is perceived as "insecure," you may be unable to sustain your business.

An incident response toolkit is vital for evidence preservation. Therefore, it's a good idea to develop one for each production server (those used for daily operation). As you may use many types of servers (different platform, with varying binary executable formats), a specific and customized toolkit must be created for each machine. This toolkit will consist of software and data files useful during the verification and investigation stages of an incident response.

The software and data should be stored on a write-protected medium, such as a CD-R disc. In addition the toolkit should be physically attached to its corresponding machine, along with traditional disaster recovery materials already in place (taped to the side of the machine in an envelope or CD sleeve, for example).

Although the specific software applications that should be included will change over time (as dictated by market and industry forces), the conceptual model of an incident toolkit should remain relatively stable. To be effective, the toolkits should conform to the following standards:

The toolkit must contain (with rare exception) all software necessary to conduct the preliminary investigation of an intrusion incident. This includes software capable of comparing cryptographic hashes, recording the contents of random access memory (RAM or core) to disk, and displaying and recording vital system information (such as open network connections, disk utilization and process information).

Cryptographic hashes of vital system components, such as binary executables and libraries, must be generated and stored on the toolkit.

It is common practice for hackers to modify system tools and applications to conceal their activities. For example, the Unix application "netstat" (used to display network connections) is often replaced with a modified version that masks the intruder's presence (at which point the system would appear to be running normally). Cryptographic hashes can be used to detect this type of activity and the authenticity of files.

A cryptographic hash or checksum is like an ID card for a file, represented by a small string of characters. It is generated using the contents of the file in a series of mathematical formulas. The cryptographic hash is unique to the sequence of bytes (the file) it was generated from. Consequently, it is possible to detect the modification of a file by recalculating and comparing the hash values. In order for checksums to be useful, you must have the checksums of your files before someone breaks into your system.

The toolkit must contain an operating system capable of booting the target machine, with all drivers and kernel modules necessary to operate the installed hardware.

Whenever possible, the software applications should be statically linked with libraries stored on the toolkit itself.

Since many software applications perform common tasks, modern operating systems package code into files called libraries. Programmers can then use these libraries to execute tasks, rather than having to write their own similar procedure. Because the libraries on a suspect machine may have been tampered with, the software on the toolkit should only interact with its own uncompromised library files.

The "incident response" will always be initiated by an initial report: a firewall or IDS may trigger an alert, log files might indicate an unauthorized access, or an employee may simply complain about a server "acting weird." Regardless of the source, the report of a suspected intrusion incident must be followed by an incident

response procedure. Here's an example:

1. Appropriate personnel are notified that an intrusion has occurred, and an investigation is underway.
2. Record all current network connections.
3. Record active users currently logged on.
4. Remove the system from the production network.
5. The Incident Toolkit is used to record system information and contents of RAM to a removable volume. This includes:
   - all current processes
   - all open files (files may be deleted if a process exits when the network is disconnected)
   - any other volatile data that would be lost, such as memory or cache (Kossakowski, K. P., & Allen, J., & Alberts, C., & Cohen, C. (1999). Responding to Intrusions. Pittsburgh: Carnegie Mellon.)
6. A separate host is used to conduct a network scan for open ports on the compromised system.
7. The system is powered down, and an evidentiary backup of the hard drive is created.
8. The original drive is labeled and sealed.
9. Second stage investigation and analysis are conducted using the evidentiary backup.
10. Solution is developed based upon conclusions drawn from the second stage investigation. A report detailing the incident response and solution is developed and delivered to senior management.

Having a detailed incident response plan and the staff capable of implementing and executing the plan can make the difference between collecting the necessary evidence or not. Being able to clearly document the steps that were taken after an incident and showing chain-of-custody with the collected data can dramatically increase the chances of prosecuting the perpetrator.

As with any new process that is deployed, starting with the most public and critical servers within the organization will get your team familiar with the requirements of assembling the tool-kit. Then, once your core servers are under the plan, the tool-kits will need to be updated when core binaries or libraries are updated. It's an arduous task at best. Is it worthwhile? We think so. CDM

Walker Whitehouse *is CIO and* **Mike Yamamoto** *is a Network Systems Engineer at Sandstorm Enterprises, which develops aggressive software products for network monitoring, network forensics analysis, and security auditing including telephone scanning, penetration testing, and vulnerability assessment.*