;login:
THE MAGAZINE OF USENIX & SAGE
November 2001 • Volume 26 • Number 7

Special Focus Issue: Security
Guest Editor: Rik Farrow

inside:

USENIX & SAGE
The Advanced Computing Systems Association &
The System Administrators Guild

incident response:

Performing Investigations on a Live Host

by Keith J. Jones

Keith Jones is a computer forensic consultant for Foundstone. His primary area of concentration is incident response program development and computer forensics.

Keith.jones@foundstone.com

Corporate IT staffs are investigating computer security incidents and computer crime more than ever before. Who would have thought the IT staff would become the "network cops" of the company? But that is exactly what they have become. Therefore, your Incident Response (IR) staff needs to be armed and prepared to support the decisions and investigations to protect corporate assets, protect employee privacy, and enforce the policies that general counsel and senior management endorse. A methodology and formal investigative process needs to be implemented.

This article will describe the process of performing a successful live incident response on a UNIX operating system and will discuss the mechanisms used to preserve the evidence. It is assumed the reader has basic system administration skills and little or no experience with investigations. Therefore, this article will provide deeper focus on the investigative aspects of the live response and methods used to collect the evidence in a forensically sound manner rather than the technical usage of the tools. Since no investigation is the same, a step-by-step process that will encompass every aspect you may encounter is difficult to provide. The information in this article will provide a solid base to executing and transferring most of the information needed for a successful investigation in a forensically sound manner.

Choosing the Toolkit for the Investigation

In order for the investigator to examine the victim machine, the correct tools must be used. To select the tools you use, keep the "big picture" in your mind. What log files do you want to retrieve? What UNIX tools help you determine the state of the system? What configuration files do you want to review? Before you create your trusted toolkit, you must determine the information you wish to acquire. The tools used in the live investigation of computer crimes are not specialized. They resemble a handful of items in the typical system administrator's toolbox. Minimally, the following areas of a suspect machine need to be examined during an investigation:

1. The current date and time of the victim server are recorded. This information will provide a baseline if the system time has been skewed from other servers present in the investigation.

   - Tool: *date*

2. All information about each network interface card, such as network addresses and states, is recorded. If the suspect altered the IP address or started a network monitoring tool (e.g., sniffer) on the victim machine, this data would display the mischievous action.

   - Tool: ifconfig
   - Typical Switches: -a

3. Each name, command line argument, length of execution time, and the user who executed the process are recorded. Rogue processes are typically initiated by intruders, and the currently running process will be part of the evidence used to prove that the rogue process was executed.

- Tool: ps
  - Typical Switches: -aux (or –ef for Solaris)

4. The open network sockets and files along with the process number that opened them are recorded. Typically, intruders leave back doors for future external access into compromised systems. External access will require network access, and open sockets are the indicator of a back door's presence. Furthermore, unknown open files usually indicate a monitoring tool that is either reading or writing to a file. As an example, the most common type of monitoring tool is a sniffer, and it writes to a log file.

- Tool: netstat
  - Typical Switches: -an
- Tool: lsof

5. Since an executable file can be deleted from the file system and currently be executing, your initial response may be the only chance to capture a copy and perform offline tool analysis on a rogue process. All suspect processes are archived in a forensic manner as an executable file.

- Tool: the proc file system
  - Executable Image Location: /proc/<PID>/exe
- Tool: carbonite (for Linux)
  - Location: *http://www.foundstone.com*

6. The IP routing table is documented. Unauthorized routing of the victim machine's network packets will be evident in the routing table, and it may indicate possible man-in-the-middle network monitoring and attacks.

- Tool: netstat
  - Typical Switches: -nr

7. The currently logged in users and their initiating place of connection are documented.

- Tool: w

8. The loaded kernel modules (if your version of UNIX allows for this) are documented. Kernel modules literally alter the operating system and are in the intruder's benefit to load them, obviously an area that needs to be examined by the investigator.

- Tool: lsmod

9. The last accessed, modified, and created timestamp information for each file on the victim machine are recorded. Correlating the timestamps for significant events provides information such as last execution, unauthorized modifications, and other mischief. Furthermore, file permissions play a significant role in most investigations and are easily captured during the same step as timestamps.

- Tool: ls
  - Typical Switches for Last Accessed Time: -alR –time=atime /
  - Typical Switches for Last Modified Time: -alR /
  - Typical Switches for Created Time: -alR –time=ctime /
- Tool: find

. . . your initial response may be the only chance to capture a copy and perform offline tool analysis on a rogue process

The simplest back door installed by an intruder is a shell, such as bash, listening on a random port.

- Tool: The Coroner's Toolkit (mactime)
- Location: *http://www.porcupine.org*

10. All auditing files are archived into evidence. UNIX auditing is typically performed by the syslog facility, and each log file is determined from the /etc/syslog.conf file. In addition to these logs, the wtmp, utmp, and lastlog files are archived into evidence so that the login history is available for the investigation.

- Tool: cat
- Tool: last

11. The /etc/passwd file is submitted into evidence. This file will be examined for possible back doors into the system. It is well known that valid, but not necessarily authorized, user credentials are the easiest way to avoid intrusion detection systems and simple access.

- Tool: cat

12. The /etc/inetd.conf file is submitted into evidence. The simplest back door installed by an intruder is a shell, such as bash, listening on a random port. This back door will be easily observed in this configuration file.

- Tool: cat

13. All suspicious files on the victim machine are submitted into evidence for offline tool analysis. The files can be transferred by using cat and redirecting the output to a netcat TCP session, which is explained in the next session.

- Tool: cat
- Tool: strings
- Tool: strace
- Tool: file

It has to be assumed that any tool resident on the victim machine may be compromised. If the intruder used a publicly available rootkit, they can trojan any system tool such as ps, netstat, ls, and even bash to provide false results to the user executing them. The tools listed above must be executed within a trusted shell, which is compiled on a system without a history of incidents. Therefore, it is necessary to execute a trusted shell before your initial response begins. Furthermore, the tools you use must be compiled statically on a forensic workstation and transferred to the victim machine. By statically compiling your tool set, you avoid running untrusted, potentially damaging processes on the victim machine. The tools can be accessed by writing them either to floppy or CD-ROM. This toolkit media is inserted into the victim machine and mounted. Additionally, it is highly recommended to perform the response from the victim console and not an X-Windows session. There are security considerations such as session spying inherent with X-Windows usage. Furthermore, a response should never be performed across a network connection such as Telnet. Once the media is mounted, the trusted shell is run by executing it on the command line. The response can begin once the investigator has completed the proper documentation and planning steps, explained in an upcoming section.

Storage of the Digital Evidence

There are several options for storing the data produced in the last section. The first and most intrusive method is to save the data on the victim machine's hard disk for

analysis. The second method, less obtrusive but also less practical, is to store the data on external media such as a floppy disk. The third and least obtrusive method involves transmitting the data to a forensic analysis machine and saving on its hard disk drive. This is the method that is recommended and is used in the rest of this article.

The method of saving the data directly to the forensic machine uses a TCP/IP network as the transmission medium. The tool that will easily establish a TCP session is named netcat. netcat is used in two modes: connection mode or server mode. The victim machine will utilize the connection mode while the forensic analysis machine will use the server mode. Since netcat establishes a TCP session, information can be sent through the connection to a server by using the command line pipe. It is possible to transfer whole files from one machine to another by simply redirecting the data on the forensic machine that was received through netcat. The complete transmission process can be summed up with the following commands:

Victim Machine: <command line> | nc <IP of the forensic workstation> <port number>

Forensic Workstation: nc –l –p <port number> > <command line>.txt

Not every network used to transfer the data from the response will be trusted. To overcome this hurdle, another tool named cryptcat can be used in the same manner as netcat. The difference between the tools is that cryptcat encrypts the TCP channel. The encryption provides two aspects: authenticity and secrecy. If a bit were changed in transmission, the data would be invalid when received on the forensic machine. Additionally, if an intruder is listening with a network monitoring tool, he or she will observe garbled data due to the encryption.

There are two realistic choices for the network transfer of incident data: create a temporary network using a crossover cable between the forensic workstation and the victim machine or connect the forensic machine directly to the untrusted network. A topic of debate is whether the responder chooses to remove the victim machine from the live, untrusted network at the time of detection. One logical approach to this question is to perform the smallest subset of investigative steps to find out if removing it from the live, untrusted network will trigger malicious code the intruder left behind. Additionally, leaving the victim machine on the network will give the investigator the chance to collect evidence if the intruder were to return by passively monitoring the situation with a network monitoring tool.

Since every investigation should be performed under the pretense that it is the "big one," it is assumed that one day the investigator will be called to the stand to swear under oath that the data is unaltered. There will be a mechanism in place to validate the authenticity of the data at any point if it is questioned. The mechanism generally accepted by the industry is an MD5 checksum. The MD5 checksum is a 128-bit length string computed from a file's contents and it is highly unlikely that two files will have the same value. To create an MD5 checksum list of several files, the following command will work well:

Forensic Workstation: md5sum –b <filenames> > md5sums.txt

The MD5 checksum will be computed for every file transferred from the victim machine to the forensic workstation. The checksum will be computed and saved on the forensic server itself. Lastly, the evidence data and the MD5 checksum file will be copied to unalterable media such as CD-ROM with the disc closed after the write.

Not every network used to transfer the data from the response will be trusted.

Documentation and Planning

Typically, documentation does not come naturally to technical individuals. However, documenting the steps taken during an incident response is paramount. Records of a response performed months or years prior have a longer shelf life than an individual's memory. Planning is also very important to the response because sometimes the investigator may only have one chance to respond correctly. Planning the commands, the order, and what switches will be used on the victim machine will follow hand-in-hand with the documentation. A simple spreadsheet is used to document what commands executed on the victim machine can be created before the response is performed, therefore allowing the investigator to plan and research the tools before they are run live during the response. An example of this spreadsheet is viewed below.

Start Time	Command Line	Trusted Execution	Untrusted Execution	md5sum	Comments
4/3/2001 10:37:47	date | nc 192.168.69.2 2222	X		e913412389f3430c5662a3ee54aef082	daylight savings time in effect.
4/3/2001 10:42:15	netstat –an | nc 192.168.69.2 2222	X		37cfdab36f8e42369f099d39af36b275	

The columns "trusted execution" and "untrusted execution" indicate how the tool was executed and are the proper place for this documentation. In these columns it will be considered an untrusted execution if any code contained on the victim machine is run, such as dynamically loaded libraries and other tools.

Another aspect that must be documented is the transfer of evidence. The chain of custody is the record of when, to whom, and where a piece of evidence is transferred. A completed chain of custody form will provide an extra level of authenticity of the evidence, if it is ever questioned, and is standard for any law enforcement investigation. A sample chain-of-custody form is observed here.

Case Number: FS-010101 Evidence Tag: 001		**EVIDENCE** CHAIN OF CUSTODY			
Evidence Description:	CD-ROM containing live response data files				
Source Location:	X	Source Name:	X	Date:	06/01/2001 12:10
Destination Location:	Onsite Investigation Miami, FL.	Destination Name:	Keith J. Jones	<Keith's Signature>	
Source Location:	Washington, DC	Source Name:	Keith J. Jones	Date:	06/02/2001 14:43
Destination Location:	Evidence Safe, Washington, DC	Destination Name:	Keith J. Jones	<Keith's Signature>	
Source Location:	Washington, DC	Source Name:	Keith J. Jones	Date:	06/04/2001 15:33
Destination Location:	Washington, DC	Destination Name:	Kevin Mandia	<Keith's Signature>	<Kevin's Signature>
Source Location:		Source Name:		Date:	
Destination Location:		Destination Name:			
Source Location:		Source Name:		Date:	
Destination Location:		Destination Name:			

If the proper documentation is created throughout the investigation, a final report is simple to compile. The information can be summarized from the various documentation sources easily. Simple generation of the final report can be the greatest motivation to document properly.

Conclusion

This article provides a summary of tools and techniques used for a successful live incident response on a UNIX operating system. Proper documentation and planning are needed in order to keep mistakes to a minimum. Documentation includes documenting the chain of custody to uphold the authenticity of the evidence should legal action follow an investigation. A toolkit of trusted binary files must be used during the response. The toolkit should be compiled statically and transferred to the victim machine before the response is executed. Lastly, steps such as calculating an MD5 checksum on the evidence and archiving it to a read-only media will also be used to prove authenticity.

These principles can be easily applied to a Microsoft Windows NT/2000 machine. Substitution of the UNIX tools with ones that are similar for NT/2000 will need to be performed, but that is not difficult. The documentation and planning stages will be exactly the same.

Proper documentation and planning are needed in order to keep mistakes to a minimum.