Forensic Readiness

John Tan

@stake, Inc.
196 Broadway
Cambridge, MA 02139 USA
tan@atstake.com

July 17, 2001

Abstract

"Forensic Readiness" has two objectives:

1. Maximizing an environment's ability to collect credible digital evidence, and;
2. Minimizing the cost of forensics in an incident response.

This paper will identify measures which may be incorporated into existing procedures for designing networks and deploying systems to increase forensic readiness.

Keywords: security, forensic, methodologies, procedures, host hardening, network design, audit, logging, accountability

1 Introduction

"Forensic Readiness" has two objectives:

- Maximizing the usefulness of incident evidence data
- Minimizing the cost of forensics during an incident response

1.1 Usefulness of Incident Data

Data from an intrusion has multiple, potential uses. It can be used as leverage in an internal incident or evidence in court. It can be used to formulate plans during an incident response or to look for additional vulnerability or compromise. It could even be used against you[1]. There are four potential sources for incident data:

[1] Unless evidence is acquired under instruction from legal council, it is subject to "discovery" in court

- The victim system(s) RAM, registers and raw disk
- The attacking system(s) RAM, registers and raw disk
- Logs (from the victim and attacking systems as well as intermediary systems)
- Physical security at the attacking system (eg., camera monitoring, etc)

A procedure for evidence acquisition and preservation can be simple, rapid and effective, saving time and money. The complexity of your environment however, demands that you define the details ahead of time. Failing to preserve the data on victim or attacking systems in a timely manner will decrease its usefulness as legitimate system usage may claim the resources in which evidence is held. The remainder of the evidence will reside in log files owned by the victim company and/or a third party. In some sense, we may think of physical security records and log files as being similar. It is not within the scope of this paper to discuss where they differ however this source of evidence should not be overlooked during an investigation.

When designing a network and deploying systems, the importance of multitiered logging can not be overlooked either. Data on the attacking and compromised systems is subject to modification by the perpetrator of the intrusion, particularly if said intrusion was successful.

Multi-tiered logging not only provides supporting evidence to what is found on a compromised system but hopefully, it provides direction in terms of incident response and how costly forensic services are utilized.

1.2 The Cost of an Incident

As part of the Honeynet Project (`http://project.honeynet.org`),I had the unique experience to be a judge in a contest where 13 participants were given disk images of a compromised Honeynet system for which they produced a report on the findings from their forensic analysis. While this provided much insight about how to report findings for evaluation as evidence, the most remarkable finding was about the cost of the incident.

In an e-mail dated Tue, 27 Feb 2001, Dave Dittrich, head of the Honeynet Project states, “I already see some interesting things, ... [including] the difference between the time spent by the intruders (about 2 hours) and the time spent to clean up after them (varying widely from a few hours to over 80!)...”. On average, 2 hours of intruder time turned out to mean 40 billable hours of forensic identification. This did not include:

- Intrusion Detection (human element)

- Forensic acquisition of disk images
- Restoration of compromised system
- Hardening of compromised system
- Network scanning for other vulnerable systems
- Communications with stakeholders

You have to be careful with all of these but the faster you can do them, the cheaper you can do them. Proper incident response planning can address these issues. This paper will identify measures which may be incorporated into existing procedures for designing networks and deploying systems to increase forensic readiness specifically.

1.3 Elements of Forensic Readiness

Evidence preservation and time to execute are affected by technical and non-technical factors including:

- How Logging is Done
- What is Logged
- Intrusion Detection Systems (IDS)
- Forensic Acquisition
- Evidence Handling

2 How Logging is Done

From a forensic perspective, the strength of the evidence collected will improve as findings are “validated” by multiple data points. Logging data from various sources have different roles to play in an incident response. Data from an IDS for instance, can act like a “magnet” when searching for the proverbial “needle in the haystack” that a system disk image presents. What would be a 80 hour investigation of poking around in the dark can be cut in half with a lead such as a timestamp from an IDS alarm.
The primary concern in a multi-tiered logging environment will be collection of and reporting on the log data.

2.1 Mechanisms

Centralized logging is the key to efficient IDS and forensic strategies. By collecting log data to a system other than the one that was compromised, the integrity of the log data is better protected. Centralized logging also allows a specific tools to be applied across all log data from multiple platforms. One centralized point of storage for log data is easier to secure, easier to backup and easier to acquire for analysis. Formatting data in a single format, such as syslog, also facilitates easier analysis of log data. While a number of logging mechanisms exist for various platforms, the objective of a centralized logging mechanism is to support the most platforms. For IP based networks, the syslog protocol[1] is the most successful at this. Unix, and most IP devices, support the syslog protocol natively. The proprietary NT Event Log mechanism may be sent via syslog to a centralized server through third party products.

- NTsyslog (http://www.sabernet.net/software/ntsyslog.html)
- WINSYSLOG (http://www.winsyslog.com/en/)
- cls syslogdaemon (http://www.cls.de/syslog/eindex.htm)
- Backlog (http://www.intersectalliance.com/projects/)
- SysLog4NT (http://www.wtcs.org/snmp4tpc/syslog4n.htm)

The recommendation would therefore be to fully exploit syslog functionality present in existing assets and to consider syslog support as a criteria for future system and device purchases. Also, establish a secured syslog server.

While syslog is the best centralized logging mechanism, it is not necessarily secure[2]. There are four distinct efforts to change the syslog protocol. As “RFC status” is the most likely route to implementation in commercial products, the IETF’s own initiative appears to hold the most promise. Efforts to make a more secure syslog include:

- Syslog-ng (http://www.balabit.hu/en/products/syslog-ng/)
- Msyslog (http://www.core-sdi.com/english/freesoft.html)
- Ssyslog (http://www.core-sdi.com/english/slogging/ssyslog.html)
- IETF syslog Working Group (http://www.ietf.org/html.charters/syslog-charter.html)

[2] The IETF syslog Working Group’s own mission[1] statement recognizes that the current version of the protocol is flawed from a security standpoint

Ideally, the syslog protocol would provide client/server authentication, guaranteed message delivery, message privacy, client/server heart-beating to assure all clients are reporting, and nonrepudiation of messages. Keep your eye out for implementation of these protocols in the future as they represent potential improvement to the integrity of evidence data collected.

2.2 Time

When logging is directed from many devices on the network into a single repository, time synchronization becomes an issue. The more devices on the network, the less possible it is to keep them all in sync. Without synchronized times, reporting will be confusing. While the central logging server can time-stamp all records received with one uniform time, evidence is more convincing when timestamps on the compromised host, IDS, and centralized logging server all say the same thing happened at the same time. While a number of time synchronization mechanisms exist for various platforms, the objective of a centralized time synchronization mechanism is to support the most platforms. For IP based networks, the Network Time Protocol (NTP - RFC 0958) protocol is the most successful at this. Unix and most IP devices support the NTP protocol natively. Windows can use NTP through third party products.

- NTP 4.x for Windows NT
  (http://www.eecis.udel.edu/ñtp/ntp_spool/html/hints/winnt.htm)
- NTPTime Client
  (http://home.att.net/T̃om.Horsley/ntptime.html)
- Listing of many Windows NTP clients and servers
  (http://www.eecis.udel.edu/ñtp/software.html#NT)

The recommendation would therefore be to fully exploit NTP functionality present in existing assets and to consider NTP support as a criteria for future system and device purchases.

While NTP is the best centralized time synchronization mechanism, it is not necessarily secure. There are three distinct efforts to change the NTP protocol. As "RFC status" is the most likely route to implementation in commercial products, the IETF's own initiative is again the most promising. Efforts to make a more secure NTP include:

- DARPA Time Synchronization Project
  (http://www.eecis.udel.edu/m̃ills/ntp.htm)
- DARPA Autonomous Authentication
  (http://www.eecis.udel.edu/m̃ills/autokey.htm)
- STIME (http://www.ietf.org/html.charters/stime-charter.html)

Even for companies operating only in one time-zone, an incident will often produce data points spanning multiple time-zones. An intrusion involves at least two parties - an attacker and a victim. Even when the victim and the intruder are in the same building, an intruder may "hide their tracks" by "island hopping" through other systems, potentially located in other time zones.

Systems should be configured to report time in Greenwich Mean Time (GMT) which accounts for time-zones through an offset value (eg., EST is GMT - 05:00 whereas PST is GMT-08:00). This is a standard convention for reporting time within an incident response context. An additional consideration for time synchronization would be accuracy of the time to which devices are synchronized. NTP provides synchronization of time but does not even consider its accuracy. An accurate time signal can easily be received with a standard Global Positioning System (GPS) receiver. Attaching a GPS receiver and the appropriate software to NTP servers in each location can be a cost effective way to synchronize time between offices or even between business partners. The integrity of the GPS time signal is also not something easily challenged in court. Keep your eye out for implementation of these protocols in the future as they represent potential improvement to the integrity of evidence data collected.

2.3 Time-Stamping

"Electronic documents will only stand up in court if the who, what, and when they represent are unassailable[2]." The most obvious implementation of time-stamping within the context of digital forensics is to digitally sign messages (syslog or other) for nonrepudiation purposes. A Digital Notary[3] can meet this need by storing a hash of the message with a trusted date-stamp, and digitally signing the whole package. Time-stamping solutions should also consider the privacy of the messages and the authenticity of client/server communications (which should also be incorporated into the digitally signed package).

Consider using digital notaries on particularly important data collection points.

2.4 Permissions

Log files on hosts and secure log servers need only be writable by the process(es) which will generate log data. Under syslog, messages are delivered to the syslog daemon which will write to the actual log file so users running daemons that utilize the syslog() function need not have write permission to any log files under syslog's control. Write permission to log files should be minimized. Read permission for log files is not typically required by the daemons which generate log

[3] Digital Notary is another term for a trusted, (ideally) impartial third party who will perform the function of digital time-stamping. Surity (http://www.surety.com/index-nn.html) and ValiCert (http://www.valicert.com//partner/html/partner_digitalreceipt.html) are two such companies. Datum (http://www.datum.com/tt/trustedtime/index.html) provides an interesting alternative NOT sold as a product/service combination but as a hardware-only solution.

data and should typically only be granted to a group of administrators responsible for maintaining the application. Log files often store sensitive information or provide attackers with feedback so even read access should be guarded.

2.5 Reporting

In the short-term, no messages should be thrown away. Even messages that are "known meaningless" have potential to play a role in some future incident response. The key is to filter these messages through reporting, then move them off-line leaving online only (derived) aggregate data. The period over which this rotation will transpire will depend on how often reporting is performed and how long this information would most probably serve any incident response that required it. Just as in any data mining operation, successful use of the data is dependent upon scalable database design, user-friendly reporting tools, and a good set of canned reports that can be run "out of the box". Just as in most data mining operations, user-friendly reporting tools are only as good as the users behind them and canned reports don't satisfy everyone's needs.

One solution which seems to make sense would be to leverage scales of economy and outsource log data retention and reporting. Such a service[4] would be provided by what is referred to as a Managed Security Provider (MSP). MSPs represent an (ideally) independent third party holding copies of your log data which does provide a very strong second data-point for log data found on a compromised host. MSPs can also (ideally) put more time and resources into reporting on and properly storing log data for use in court.

Network architects should consider the privacy and monetary trade-offs involved with IDS monitoring and reporting, as they apply to using (or not using) MSPs. MSPs offer economies of scale in terms of staffing a skilled, 24x7 team to interpret and respond to IDS data. MSP infrastructure could offer best practices not financially feasible for smaller firms (eg., continuous human monitoring, secure log storage, digital time-stamping, etc). The down-side is that sensitive logging data is retained by an outside party who may also be a juicy target for attackers or opposing legal council. Perhaps syslog, firewall and generic IDS data could be sent to an MSP while internal network monitoring data, select application logs, proxy logs and "custom" IDS data be retained internally "under the advise of legal council" [5]. This way internal incidents remain internal.

Assess how current incident data is collected and reported on. Consider cost and liability when deciding which pieces to outsource to an MSP.

[4] Counterpane was the first and provides the most notable example in this area. http://www.counterpane.com/.

[5] This author is not in any way qualified legal council. This author's opinion was developed after an interview with Jennifer Grannick, criminal defense attorney and Clinical Director of the Stanford University Center for Internet and Society.

2.6 Retention

From an incident response, decision support stand-point, retention of log data should be maximized. Data should be kept online for some period of time that is found to support reporting needs. After that, data should be moved off-line via backups. XML representation and statistical aggregation can also be performed to provide "meta-data" where said "meta-data" could be structured to support questions typically asked during incident responses about long term trends.

From a liability standpoint, the activity that transpires on your network is what is being logged. Ultimately, this data could be used against you. Each organization will have its own level of tolerance for this particular risk. The appropriate time-frame could be no time at all, 2 weeks, 2 months, 6 months, or even 7 years. Regulations, laws, policies or agreements could all be factors in determining the appropriate time-frame for each data element that is logged.

Consider XML to create meta data which will support the incident response effort. Weigh liability vs. obligation and utility when determining retention periods for data elements.

3 What is Logged

What is not logged is lost. Every application on every system or device on your network represents a logging opportunity. Disk space and man-hours are limited however, so when taken to an extreme, this philosophy can result in an unmanageable mess. For this reason, how a system is deployed and what function it serves should influence how much is logged. Host The most useful host level logging includes process, file-system, network and security accounting. Process accounting data from Windows systems is readily usable by forensic tools as is file-system accounting data for just about all platforms . Network and security accounting can also provide useful information but is a less defined realm.

3.1 Host

The most useful host level logging includes process, file-system, network and security accounting. Process accounting data from Windows systems is readily usable by forensic tools[6] as is file-system accounting data for just about all platforms[7]. Network and security accounting can also provide useful information but is a less defined realm.

[6] The NT Event Viewer can be used to view audit records. Process creation (Event 592) is logged with the user, their domain, the program run, time, date and various IDs.

[7] The mactimes(1) tool from The Coroners Toolkit (TCT) may be run from Linux against image files mounted (read-only) via the loopback interface. In that Linux supports a wide range of Windows, Unix and Apple file-systems, MAC times may be easily acquired for a wide range of Operating Systems. http://www.fish.com/tct/index.html

Element	Unix	Windows
Process	May be turned on using accton(1), and in some cases, enabled in the kernel. Provides limited billing oriented accounting (does not include PIDs). Most useful when insecure services (plain-text authenticated services such as telnet(1) or ftp(1)) are used. .history files can provide (spotty) process accounting data and should be collected.	The Audit features of the Security Policy Administrative Tool provided by Microsoft provide Audit Process Tracking which should be enabled for both Success and Failure.
File System	Unless mounted with special options, file-systems will track the (last) Modified, Accessed and Changed (MAC) times for a file. While limited, this data provides useful information for reconstructing a time-line of the intrusion from a disk image.	File-systems will track the (last) Modified, Accessed and Created (MAC) times for a file. While limited, this data provides useful information for reconstructing a time-line of the intrusion from a disk image.
Network	Third party host firewall: Ipfilter, IpChains, etc	Third party host firewall: Zone-Alarm, BlackICE, etc
Security	(Typically) third party security package: BSM, etc This data can be useful but like finding a needle in a haystack, especially without additional package specific tools.	The Audit features of the Security Policy Administrative Tool provided by Microsoft is quite useful.

The recommendation is therefore to use default file system accounting and weigh performance vs. risk when considering using process accounting. Consider the impact of backup software on file system accounting as well.

3.1.1 Unix

Unix logging is configured for each application and the kernel. Logging is largely done to ASCII text files under /var however some applications may behave differently. Unix offers /etc/syslog.conf as a central configuration file for distribution of messages based on a predefined (by the syslog protocol) set of priorities. Applications designed to utilize the syslog() function to send messages to syslogd(1M). The syslog facility captures, a facility indicator, a severity level, a timestamp, a tag string and optionally the process ID[3].
Process accounting may be useful when user credentials represent a problematic

area of risk. Systems on hostile network (eg., DMZ), shared by multiple interactive users (eg., a shell server), or running weakly authenticated services might consider taking the performance hit and extra disk space requirements. Other systems may find that simply editing /etc/syslog.conf will suffice. Ideally all messages would be captured to non-volatile local locations (eg., not /dev/null or /dev/console) and duplicated to a (secure) remote log host.

Unix process accounting tracks command name, user and group IDs, controlling tty, process exit status, memory usage, characters transferred and blocks read or written. This is slightly more performance oriented than Windows process accounting and it noticeably lacks PID of the process and its parent. Data is stored in a binary format as opposed to ASCII so process accounting programs (such as acctcom(1)) must be used to report on said data.

3.1.2 Windows

Windows logging is configured for each application and the Operating System itself. Logging is largely done to the Windows Event Log facility though some applications may log data to ASCII text files anywhere on the system. Windows also offers the Windows Registry facility which is used to store configuration and sometimes other data. The Windows Registry maintains MAC times for registry keys and their values similarly to that described above under File-system Accounting.

The Windows 2000 Security Options section of the Local Security Settings Administrative Tool provides a number of logging opportunities including access of global system objects as well as use of Backup and Restore privilege. Additionally, control over the number of logins cached, clearing of the system pagefile (swap) on shutdown, and the ability to shutdown if unable to log security audits to name a few. These more subtle settings are subject to each organizations level of risk tolerance.

The Audit Policy section of the Local Security Settings Administrative Tool (NT User Managers Audit Policy equivalent) provides the most obvious logging opportunities. In general, all failures should be audited. Tracking of successes is subjective to each organizations risk tolerance.

3.2 Network

Sources for forensic evidence on the network can be abundant if one considers where those may lie. The most obvious places on the network for finding logging opportunities of interest to a forensic acquisition would include:

- Firewall
- IDS
- DNS

- Router (eg., Cisco NetFlow, etc)
- Proxy Servers
- DHCP Servers
- Dial-up Servers
- VPN

In addition to the logs of network services an intruder may attack or otherwise utilize, the network provides the additional forensic opportunity of network monitoring. Some incidents require network monitoring simply to obtain IP address information, perhaps to augment an application log which lacks this information but could be correlated through a timestamp. For more sophisticated, insider incidents involving abuse of legitimate access, sophisticated commercial products[4] are available which present captured network traffic in such a way as to enable analysis that would otherwise be infeasible in terms of man-hours and cost.

All network services providing applications should be scrutinized for logging opportunities. Logging should be done by IP address as opposed to resolving names for those IP addresses. Ideally, both would be stored however, due to concerns with DNS poisoning, post-processing logs to populate DNS names would be recommended. This processing could be more sensitive to the integrity of the name resolved and offload considerable overhead from the production system.

4 Intrusion Detection Systems (IDS)

At one time the argument was whether HIDS or NIDS was better. Today, as evidenced by the merging of HIDS and NIDS in the market place, a mixed solution is necessary. This is complimentary to the forensics oriented desire for multi-tiered and centralized logging. More importantly however, HIDS and NIDS alarms from mixed or stand-alone solutions can provide direction to needle-in-the-haystack searches through acquired disk images, thus minimizing billable forensic lab hours. Ideally, a mix of these alarms should accompany any disk image that is sent to a forensic lab.

4.1 NIDS

NIDS come in two flavors, signature based and anomaly based. Some will argue a mix of these is the best solution, others will pick one over the other. Of note is that the forensic value of anomaly based NIDS (or HIDS for that matter) is subject to computing policies with regard to expectation of privacy at least in terms of their use as leverage in an incident response.

The most significant attribute of a NIDS from a forensic standpoint will be the integrity of its log data. NIDS should not be subject to message spoofing and should maintain a heart-beat between sensors and controllers to assure full operation at all times. Regardless of how alarms are managed and prioritized, all alarms and as much NIDS data as possible should be retained.

4.2 HIDS

The most significant attribute of a HIDS from a forensic standpoint will be where it sits relative to an intruder. The most recent trend in intrusion attack kits (root-kits) is to move from user-land to the kernel[8]. In the past, kernel level compromise of a system meant recompiling the kernel from source on the target system or, in extreme cases, patching a running kernel. This was considerably difficult however, the recent advent of Loadable Kernel Modules (LKM) has made kernel compromise scriptable.
HIDS currently:

- Audit file integrity
- Perform signature matching
- Seek to detect buffer overflows

4.3 The Kernel

Though no commercial HIDS vendors (including hybrid IDS vendors) have expressed intent to move into the kernel, we do see researchers moving user-land HIDS techniques into the kernel[9]. This however, is just the start. New, kernel-level techniques are needed to detect kernel-level root-kits. Those techniques are being explored by the underground[10] and research communities.
As with the anti-virus game, the Kernel IDS game is won by being there first. Unlike the anti-virus game, this does not necessarily mean loading software on every system. In addition to moving new and existing IDS techniques into the kernel, performance monitoring offers promise for detection of compromised kernels.
To further define the use of performance monitoring as an IDS technique, consider that for the most part, kernel modifications will try to do similar things; log input, provide a backdoor or hide their presence. They typically do this by trapping syscalls to facilitate modified behaviors. In doing so, they utilize more (or in some cases, potentially less) resources than the original system call. To

[8] A number of Loadable Kernel Module (LKM) root-kits have appeared since 1999 including SLKM http://www.pimmel.com/articles/slkm-1.0.html, and CaRoGNa http://s0ftpj.org/tools/carogna.c though this concept was published as early as September 01, 1997 in Phrack Magazine Volume 7, Issue 51.

[9] Foundstone R&D has published a Intrusion Detection tool (Carbonite) which provides, lsof and ps at the kernel level. http://www.foundstone.com/rdlabs/proddesc/carbonite.html

[10] The Italian S0ft Project group has been working on techniques under Linux for compromised kernel operation and as such, detection. http://www.s0ftpj.org/docs/lkm.htm

some extent, the data collected by Unix process accounting may be useful to these ends as it tracks process exit status, memory usage, blocks read or written, and user, system and elapsed time in clock ticks.

For instance, installing a root-kit will change the heart-beat of your machine. If the root-kit uses crypto, the processor usage should jump. Alternately, we can push the limits of the system to flush out similar resource utilization anomalies. While supplying an intense amount of keyboard activity and monitoring memory and disk activity, keystroke loggers might be detected. Similarly, performing a huge amount of BASIC_AUTH to your web server may reveal processor, disk space and network activity baseline and anomaly thresholds. This is akin to driving a pheasant out of a gorse bush by beating the bush with a stick[5].

Forensic acquisition with a compromised kernel needs to be different than forensic acquisition without a compromised kernel so detection of kernel compromise is essential to an incident response. Using standard acquisition techniques on a system with a compromised kernel will likely lead to less than adequate data while using advanced techniques on a system with a standard compromise could double or triple the man-hours required for forensic identification.

5 Forensic Acquisition

Forensic acquisition should follow intrusion detection in a timely manner. As such, much of the forensic readiness effort should be put toward deciding how evidence will be acquired from any computer or other device used on the network. In many cases, standard systems with standard disk capacity will fall under common procedures. The more esoteric the system or the higher its capacity, the more unique its requirements may be.

Forensic acquisition typically amounts to collection of volatile data (RAM, register state, network state, etc) and imaging (see below) of the systems disks. In some cases, these techniques may be appropriate. In other cases, use of backups may be adequate. A number of acquisition scenarios are presented below to highlight what needs be considered during forensic acquisition planning.

5.1 Standard Volatile Data Acquisition

In the case of a standard compromise, involving no more than a user-land rootkit, a forensic acquisition could be performed using a statically linked set of programs and libraries (perhaps on CDROM) to collect evidence data. This process needs to observe the widely accepted Order Of Volatility[6] (OOV) which implies that collecting some data impacts other data. The OOV identifies an order which maximizes the usefulness of the data collected based on its relative volatility.

Optimally, collection of volatile data would be done via a forensic shell which would not only redirect output via an encrypted network transport to an evidence server but would also do the same for script(1) output. Capturing script(1) output would effectively log all actions taken by the individual using the shell to capture the data which would represent very authoritative investigator note-taking.

5.1.1 Unix

Options for collection of volatile data under Unix include grave-robber from The Coroners Toolkit, and doing it by hand. The primary problem with either of these options is where to put the data. Simple elements, captured by hand can be captured to floppy disk in many cases but in many cases may require a network transport utility such as nc(1). Unfortunately, while grave-robber does an excellent job collecting a wide range of data in an automated fashion which observes the OOV, it does so by writing its output to the file-system. For this reason, grave-robber is best suited to be used for volatile data collection in situations where high-capacity removable media is available or only live disk images will be taken. If grave-robber must output to a fixed disk associated with the compromised system, that disk image should be taken prior to volatile data collection, actually violating the OOV. This is especially true if that disk is a system disk (providing /var, /tmp, /, etc).
Keeping a CDROM based tool kit to provide trusted versions of TCT and the Unix commands it uses (as documented in paths.pl of TCT), and a network transport utility, will help assure incident data is collected in a timely manner and with integrity.

5.1.2 Windows

There is only one option for volatile data collection under Windows and that is to do it by hand. Again, the problem of where to put the data exists and the answers are the same as for Unix. Under Windows, pagefile.sys should be considered as a fairly volatile item and it should be assured that the Local Security Policy Administrative Tools "Clear virtual memory pagefile when system shuts down" policy is disabled though changes to the system are highly discouraged in general.

5.2 Volatile Data Acquisition with a Compromised Kernel

In the case of a kernel compromise, forensic acquisition needs to happen from outside the compromised area. This changes the game considerably. The universally uncomplicated method of acquisition in this case is to lose volatile data, power the system off and image the disks from a trusted system. In most cases, some limits could be pushed to change this. Analysis of data collected at this low level is not something supported by current tool-sets however, so additional work would also be required to actually use data collected in such a manner.

5.2.1 Sparc

Sun Microsystems Sparc and UltraSparc keyboards come standard issue with an L1-A key which signals an interrupt in the kernel, freezes system state and invokes the OpenBoot command line interface. This interface provides direct access into the system at an assembly level as well as a FORTH interpreter and a number of commands implemented as FORTH words. The .registers word for instance, will display register states and the sync word will dump memory to the partition defined as swap then reboot the system.

A forensics oriented re-write of the assembly code called by the sync FORTH word in OpenBoot would greatly benefit acquisition with a compromised kernel. Such modifications to sync would include a user specified dump location, possibly including an option to use tftp as an output. The modified sync would also then exit back to the OpenBoot command line allowing the user to boot, power-off or optimally, go (return to a running state).

5.2.2 Laptops

Laptops typically feature a forensic god-send hibernation. Power management features designed to preserve state while minimizing energy consumption provide a highly promising option for collection of volatile incident data. Laptop disks are typically IDE and cables are available[11] to adapt a 44 pin laptop hard drive to use like a regular 40 pin IDE hard drive. Acquiring one up front can save time during an incident as it will enable evidence to be captured to a file which may be sent electronically.

5.2.3 Intel

Intel presents no ready options for stepping outside the area of compromise. In the pre-windows days of the 384 and 486, products like the Periscope board allowed one computer to debug the other via direct hardware access. Such a setup would be an ideal proto-type for an Intel forensic acquisition alternative. Such products served the computer programmer market at one point but have gone the way of the dinosaur these days.

Another less than optimal alternative is to run servers as Virtual Machines (VMs). Products such as VMWare have descent performance and ease of backup/restore working in their favor. Under a VM scenario, the host OS could provide an access point to the VMs memory and disk which is outside the area of compromise. Few organizations would be willing to run their production

[11] Cables Online sells a 2.5 Laptop IDE Hard Drive Adapter for $7.99 (as of 06/06/2001). http://www.cablesonline.net/25hdmounkitw.html

systems as VMs however and VMWare does have its limits.

5.3 Imaging

Imaging is similar but not the same as taking a backup. Backup software operates at the file-system level and, in some cases, will update file-system accounting data (Access time a.k.a. atime). In that MAC timestamps are only maintained by the system for the last Modification, Access, and Creation operations on a given file, a backup can effectively "zero-out" some of the most valuable forensic data. Imaging operates below the file-system level and as such, does not update file-system accounting times. Imaging software also captures deleted data which still resides on the disk but is not presented to the file-system and therefore absent from a backup.

In some cases, database Management Systems (DBMS) for example, MAC data may not be relevant and a database backup will suffice. Even the unallocated section of a disk is not relevant if the DBMS uses cooked files on a dedicated disk since all DBMS data would reside within that file. Of course, the DBMS still resides on a system with an Operating System and deleted files from those disks may very well be of interest.

Thus the requirement to consider in advance how evidence will be acquired, particularly from complicated, high capacity or esoteric systems.

5.3.1 Live Disk Acquisition

Sometimes downtime for a system is not an option. In such a case, it should be made known to the decision maker on this issue that it will not be possible to cryptographically prove that images taken came from the physical disks of the compromised system. The cryptographic checksum of the live disks will constantly change and will never generate another image cryptographically matching the first. A Digital Notary or additional evidence can help with this issue.

With that said, imaging of a live systems disk(s), where compromised is not at the kernel level, will require a trusted version of dd(1) for Unix or some open source equivalent for Windows[12]. Use of closed source products such as Nortons Ghost is risky as forensic methods must either be sanctioned by the court (eg., I did the right thing because I used program X and program X is a method the court accepts) or defendable by expert witness. As an expert witness, closed source is an undesirable platform to provide this defense from. With all imaging software, special attention should be paid to command line options that might impact the integrity of the image.

Once an imaging program has been selected, the primary concern will be where to put the data. A number of alternatives exist when using dd(1) which is capable of sending the image data to stdout. A number of network transport

[12] If such an animal exists.

utilities including exist[13], many of which provide encrypted network transport which can be a plus for data privacy and authentication of sender and recipient. The dd(1) command can be piped to such utilities to a remote evidence server, ideally directly connected to the victim system via cross-over cable. As the images are captured on the evidence server, the opportunity presents itself to take cryptographic hashes and possibly involve a Digital Notary.

At a minimum, cryptographic hashes should be taken and stored separately and securely to assure their integrity. This way, the integrity of a given copy of the disk image is provable via comparison of this baseline signature versus that given copys cryptographic signature.

5.3.2 Physical Disk Acquisition

From a business perspective, some systems can just go away while others cant be taken offline. Fortunately, there are also systems which can be taken down for a minimum amount of time necessary to properly preserve the incident data (and potential evidence). Minimization of the time and cost to physically acquire evidence from a system can provide strong incident response leverage efficiently.

The most efficient physical acquisition starts with a new set of disks with the same or greater capacity than those to be imaged on the target system. Once any volatile data has been collected, the system should be powered off. Typically, avoiding Operating System shutdown routines is advantageous so no temporary files are cleared. The disks should then be attached to a system which may be trusted. If youre prepared, this might simply mean slipping a bootable CDROM with Unix and statically linked copies of dd(1) and nc(1) or one of netcats cryptographic alternatives.

Once the disks are physically attached to a system which can be booted to provided a trusted kernel and tool-set, dd(1) should be used to create an image, either to a disk mounted on the imaging workstation or over the network to an evidence server.

Once disk images have been captured to a file on an evidence (or imaging) server, they should be captured again. Cryptographic checksums should be taken and compared for both copies of each image to assure the integrity of the image taken.

Once integrity of the captured image(s) has been established, the original disk(s) should be removed and put into static a proof bag. A Chain of Custody should be started for the physical disk(s) and it, along with the disk(s) itself should be stored securely.

Assuming the system must return to service as quickly as possibly, the new disk(s) should be placed in the imaging server and the captured disk image(s) transferred to the new disk(s). The new disks may then be put back into the original system which may then be brought back online. This system will be an

[13] A number of encryption-enabled network transport utilities exist. Some include: zeedeebee (http://www.winton.org.uk/zebedee), cryptcat (http://www.farm9.com) and netcat (http://www.l0pht.com/ weld/netcat.html).

exact image of the compromised system and, as such, will still be compromised and just as vulnerable as the original system. The evidence will however, be preserved and business continuity may be established or at least pursued. For instance, a new system could be built and hardened while the original system performs any functionality it is capable of in its compromised state.

All these variables underscore the importance of planning ahead of time for how forensic acquisitions will happen on each distinct platform.

6 Evidence Handling

Evidence handling represents the "rest of the equation" after evidence has been acquired.

6.1 Chain of Custody

The objective of a Chain of Custody document is to track who had access to a given piece of evidence, when and optionally for what purpose. Immediately upon acquisition, the responder to an incident should start tracking who has custody of what evidence. This custody needs to be tracked as it is changed and, in the case of digital evidence which may be copied, replicated.

Chain of Custody forms should be readily available to those who respond to incidents. The form should include an area for recording physical attributes of the evidence (either where it came from or serial numbers, model numbers, etc from its labels). Capacity, block sizes, format and other information is also useful. For example, for DLT copies of evidence data being archived, I include the command used to get the data onto the DLT. The rest of the form should include a sign-out sheet style chart which tracks names, signatures, dates and optionally purposes.

The life of a Chain of Custody document should start when the data is first considered as potential evidence and should continue through presentation of the item as evidence in court.

6.2 Network Transport

The objective of digital evidence transport is to provide a provable means of restricted access to evidence. Use of cryptographic network transport utilities such as zeebeedee(1) or cryptcat(1) offer privacy and authentication of sender and receiver. Netcat (nc), which does not use encryption for either purpose, is still a perfectly acceptable mechanism. Integrity of data which has undergone network transport may be proved via cryptographic hashing prior to sending and after receiving, then comparing the results, which should match.

Incident response teams should familiarize themselves with tools to perform these functions.

6.3 Physical Transport

The objective of physical evidence storage is again to provide a provable means of restricted access to evidence. Physical transport of evidence should be handled by a federal courier such as U.S.P.S., U.P.S. or Federal Express. Evidence should be sent and Chain of Custody documentation should be updated to reflect tracking numbers and any other information available. Evidence should be packed to meet at a minimum, the standards established by the F.B.I.[14]. Incident response teams should familiarize themselves with these guidelines.

6.4 Physical Storage

The objective of physical evidence storage is again to provide a provable means of restricted access to evidence. While it may be desirable under various circumstances, selecting an evidence safe is less about prevention than it is about tamper evidence. If it can be proved that surreptitious access to the evidence was not gained, and record keeping with regard to legitimate access is in order, it can be argued that access to the evidence was successfully restricted. Depending on the environment, it may be necessary to take measures to prevent the situation where tampering actually was evident as access was not properly restricted. Card based access control systems can provide protection and automatically handle record keeping.

The container in which evidence will be stored should consider the difference between computer media and paper with regard to fire protection. Where paper chars at 460oF, data will start disappearing around 120 oF. Data safes may be purchased or tamper-evident document containers may be used in conjunction with media coolers[7]. Media coolers also increase the time for physical intrusion as larger shapes must be drilled or torched out of the safe to remove a 12.25x10x10 cooler than a CDROM or DLT cassette.

Ultimately, a secure container within an audited access controlled room with camera monitoring and limited traffic would be provide a foundation for secure physical storage of evidence.

6.5 Examination

Evidence is never examined directly. The rule is to collect now and examine later. What is examined later is not the original evidence itself but rather a copy of the original evidence data. The most original version of any digital evidence

[14] The F.B.I. Handbook of Forensic Services outlines a procedure for Packing and Shipping Evidence (http://www.fbi.gov/hq/lab/handbook/submissn.htm#Packaging and Shipping Evidence) in general as well as for computers specifically (http://www.fbi.gov/hq/lab/handbook/examscmp.htm).

(file) should be stored, ultimately on Write-Once-Read-Many (WORM) media, with a cryptographic hash stored offline in a physically secure container.
If disk images in particular are to be examined, the most important point will be the starting point. Once a copy of the image file has been transferred to a system for examination, the disk image(s) should be mounted read-only with the now classic command line:

```
# mount -o ro,loop,nodev,noexec victim.hda8.dd /t
```

Where victim.hda8.dd is a dd image of the victim systems root disk partition. Additional disk partitions from the compromised system can be mounted under the temporary root (/t) in their standard locations (eg., /var as /t/var). From this point, the evidence can be examined without concern for affecting its integrity.
Incident response teams should be aware of and follow this practice of only working from copies of the original evidence and mounting them read-only.

7 Acknowledgements

While the concept of forensic readiness is my own, most of the ideas here are not original.

The work of Dan Farmer has been cited both in reference to the Order of Volatility concept as well as The Coroners Toolkit. Many of these concepts were developed using TCT and reading Dans works. Another major contributor to both of those works was Wietse Venema of IBM.
Peiter Mudge Zatko from @stake was another major contributor to this paper. His understanding of Sun Firmware and FORTH has come in the form of divine guidance to me in wrestling with how to deal with perfect root-kits.
Chris Anley of @stake was one of many contributors with regard to the concept of performance databased IDS for kernel compromise. Brian Carrier of @stake also was of help with regard to the overall direction of the paper as well as offering opinions on the analysis of data from advanced, acquisition with a compromised kernel techniques.
Chris Wysopal and Hobbit of @stake and Lance Spitzner of Sun have all helped in my understanding of forensics in general. The concepts I present here, I acquired or developed while learning from their assistance, questions and examples.
Sebastien Lacoste-Seris and Nicolas Fischbach of Securite.org also should be thanked for putting me on the spot at the October, 2000 Black Hat Briefings in Amsterdam. Their heavily accented queries about what happens when the system calls table has been buggered lead to my exploration of perfect root-kits. This forced finalization of my explorations to control firmware sync behaviors

and added some host configuration considerations to the list. In fact, this paper was inspired by the fact that my Black Hat response was less than earth shattering. The fact that hidden data would have to be accounted for via external sources was not as far as we can go; firmware sync is the (imperfect) next step.

Special thanks to those who reviewed the (very different) first draft of this white paper. The scope of the changes from the first to this version speaks volumes about the value of your input.

Appendix: Windows Local Security Settings Logging Opportunities

Policy	Minimal	Optimal
Audit use of Backup and Restore privilege	Enabled	Enabled
Clear virtual memory pagefile when system shuts down	Disabled	Disabled
Do not display last username in logon screen	Disabled	Enabled
Number of previous logons to cache	10 logons	Site specific (max. is 50)
Shut down system immediately if unable to log security audits	Disabled	Enabled

Table 1: Windows 2000 Local Security Settings: Security Options

Policy	Minimal	Optimal
Audit account logon events	Success,Failure	Success,Failure
Audit account management	Success,Failure	Success,Failure
Audit directory service access	Failure	Success,Failure
Audit logon events	Failure	Success,Failure
Audit object access	Failure	Failure
Audit policy change	Success,Failure	Success,Failure
Audit privilege use	Failure	Success,Failure
Audit process tracking	Failure	Success,Failure
Audit system events	Failure	Success

Table 2: Windows 2000 Local Security Settings: Audit Policy

Policy	Minimal	Optimal
Logon and Logoff	Failure	Success,Failure
File and Object Access	Failure	Success,Failure
Use of User Rights	Failure	Failure
User and Group Management	Success,Failure	Success,Failure
Security Policy Changes	Success,Failure	Success,Failure
Restart, Shutdown and System	Success,Failure	Success,Failure
Process Tracking	Failure	Success,Failure

Table 3: Windows NT Domain Manager: Policy Tab

References

[1] The IETF syslog Working Group's current mission statement - http://www.ietf.org/html.charters/syslog-charter.html.

[2] Merrill, Charles R., "Time is of the Essence", CIO.com, March 15, 2000.

[3] Solaris man(1) pages for syslog(3C)

[4] NetWitness Software Product Description - http://www.forensicexplorers.com/software.asp

[5] Anley, Chris, an interview via email on March 6, 2001.

[6] Farmer, Dan, Wietse venema, "Computer Forensic Analysis", slide presentation for IBM T.J. Watson labs, 8/6/1999 - Section 1: Introduction / Murder on the Internet Express - http://www.fish.com/forensics/intro.pdf

[7] Schwab Media Cooler Product Description - http://www.safes.com/schwabmediacooler.html