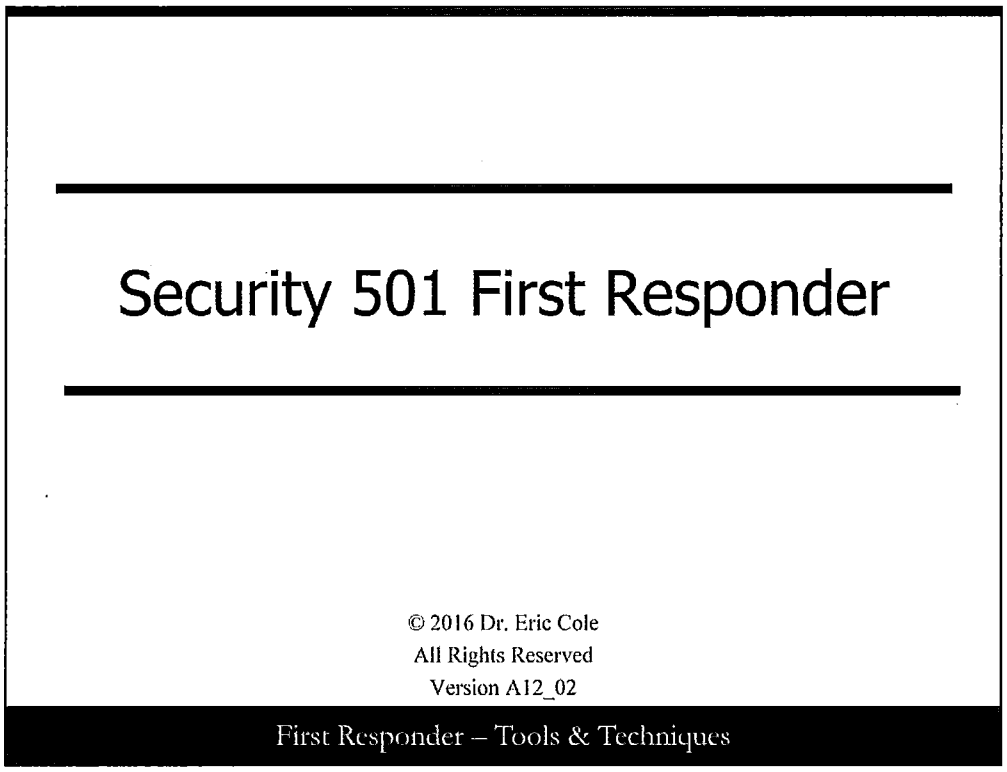
501.4
First Responder

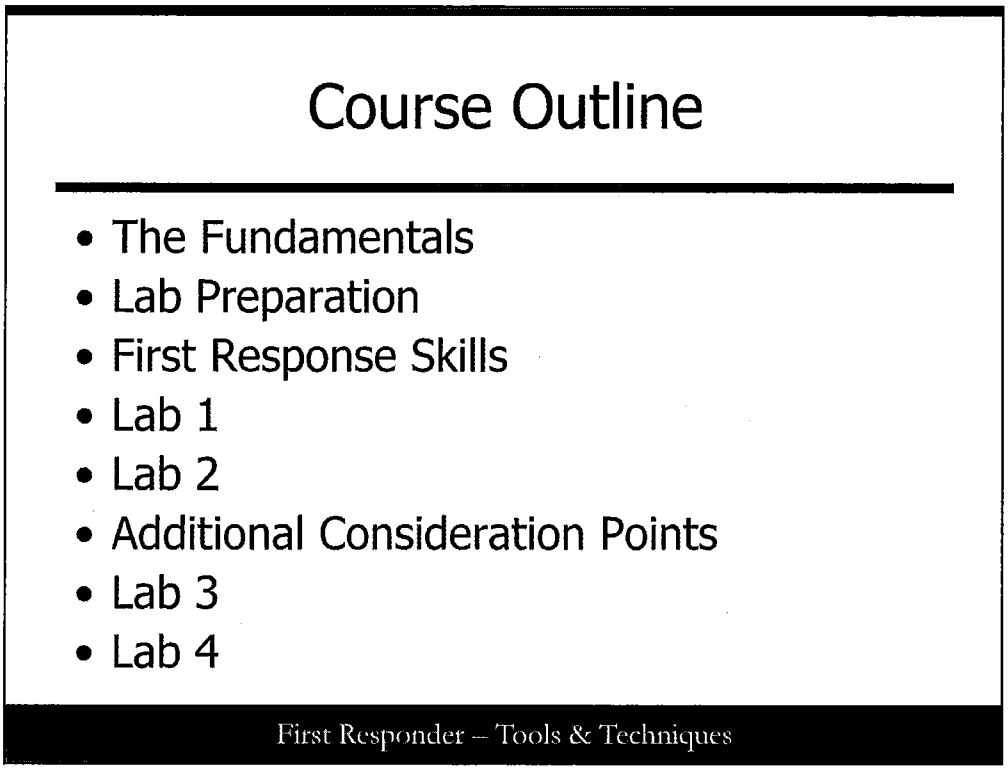
SANS

THE MOST TRUSTED SOURCE FOR INFORMATION SECURITY TRAINING, CERTIFICATION, AND RESEARCH | sans.org



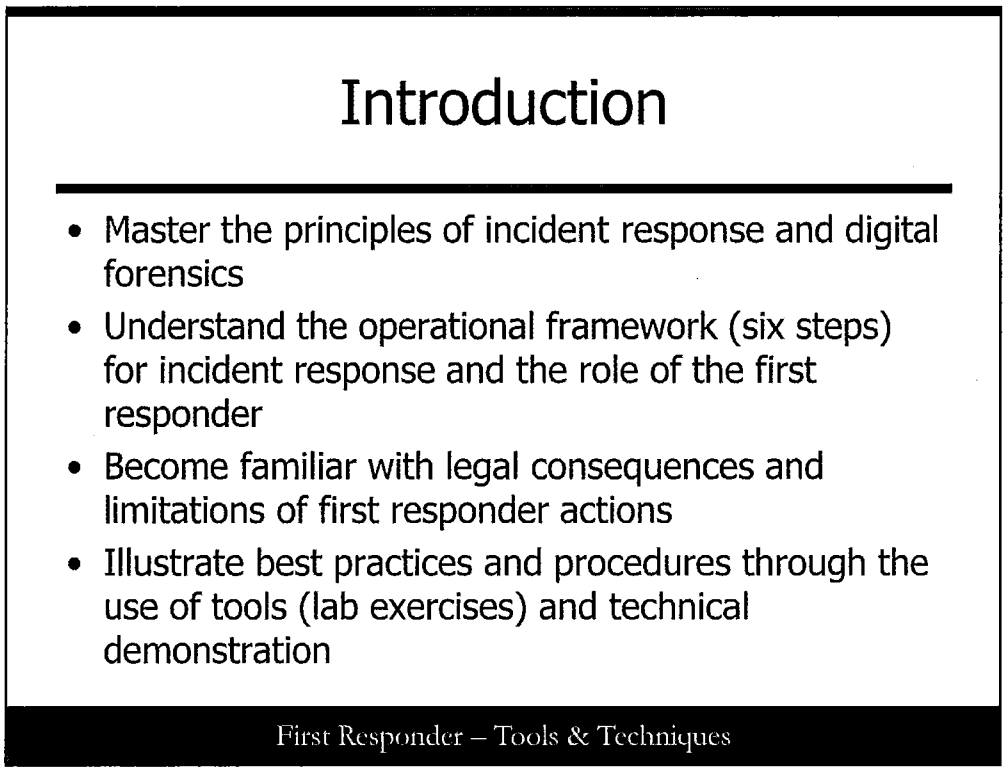
Security 501 First Responder

This is the SANS First Responder course. This course is written for anyone who may be interested in this topic, but it is primarily geared toward security personnel that are members of a Computer Security Incident Response Team (CSIRT), as well as consultants, system administrators, and law enforcement personnel. This is an introductory course in incident handling and the basics of digital forensics that is designed to help participants function as first responders—people who are the first on the scene of a computer security-related incident.



Course Outline

This course is divided up in two main sections. The first module discusses essential information and theoretical concepts, including the incident response framework and the practical limitations/legal implications of digital forensics. The second module focuses on the technical concepts associated with performing incident response in both the UNIX/Linux and Windows environment.

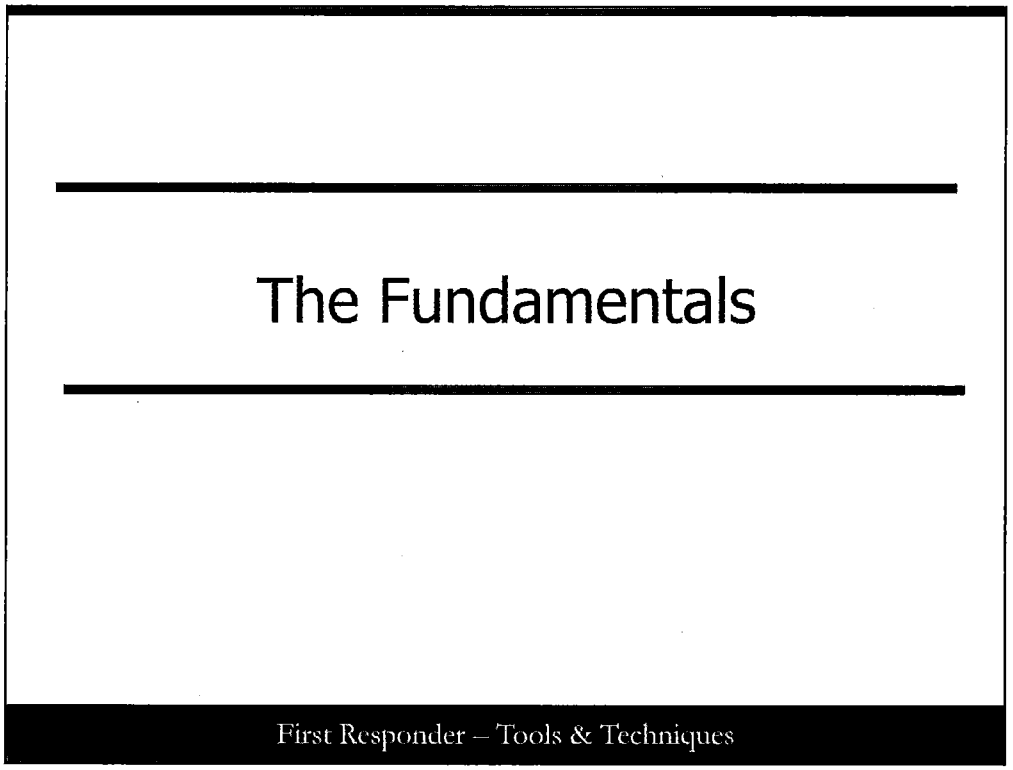


Introduction

In this course, we provide you with a solid understanding of the principles of incident response and digital forensics. This enables you to apply these principles when responding to an incident, regardless of the nature of the incident. Understanding these principles, you will know what to do and what not to do during an incident, and you will be able to take reasonable action within your technical capabilities to preserve data and provide information to management.

We discuss the © SANS Six Steps of Incident Handling. One of the goals for this class is to ensure that the role of first responder is clear. Although each case is different, an organization may want to expand on the traditional role of the first responder; typically, it is the responsibility of the first responder to triage the situation.

During triage, the first responder needs to understand the nature, timing, and extent of the incident to report the situational facts to management so that tactical decisions can be made. This information is typically collected through inquiry, observation, and real-time analysis. This course demonstrates best practices during incident response through the use of tools such that the first responder's actions will not negatively impact the organization.



Part One: The Fundamentals

Outcome Statement

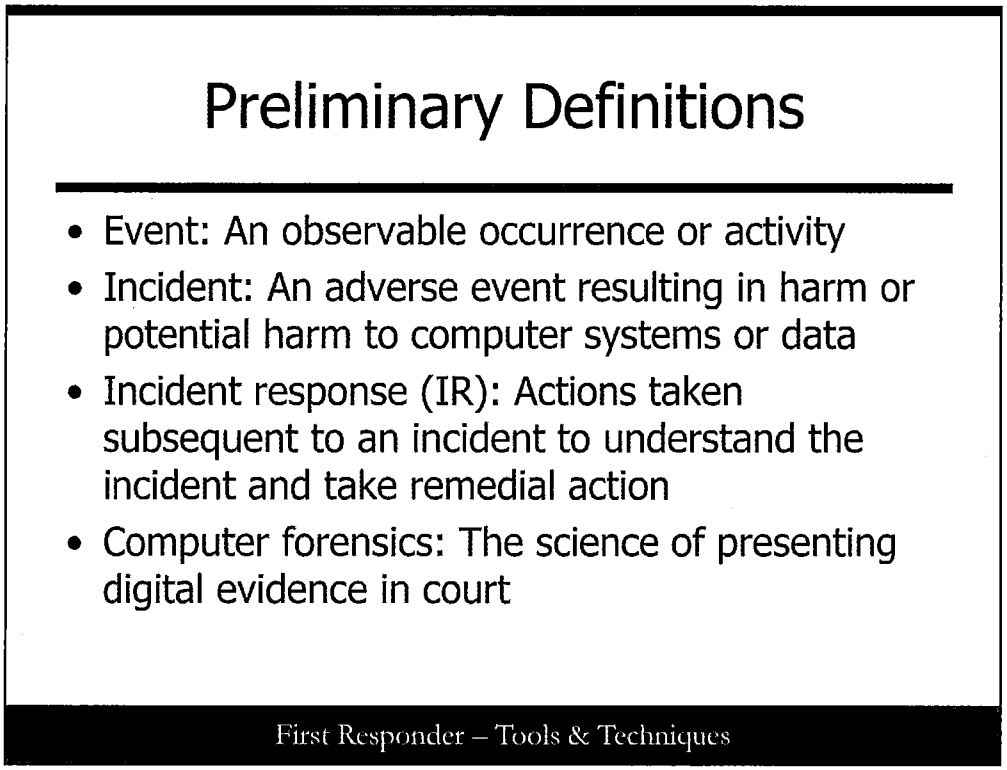
Let's start off by discussing and explaining the fundamentals of incident response and digital forensics. In this course, you are introduced to the principles of incident response and digital forensics as well as the © SANS Methodologies, and you will be able to describe the actions that occur during each step with a precision focus on the actions that occur during triage and the impact of their actions on a system.

© SANS Incident Handling Methodology:

- Preparation
- Identification
- Containment
- Eradication
- Recovery
- Lessons Learned

© SANS Forensic Methodology:

- Verify the incident
- Gather a system description
- Collect evidence, logs, reports
- Timeline creation
- Media Analysis
- Recover data
- Keyword search
- Report



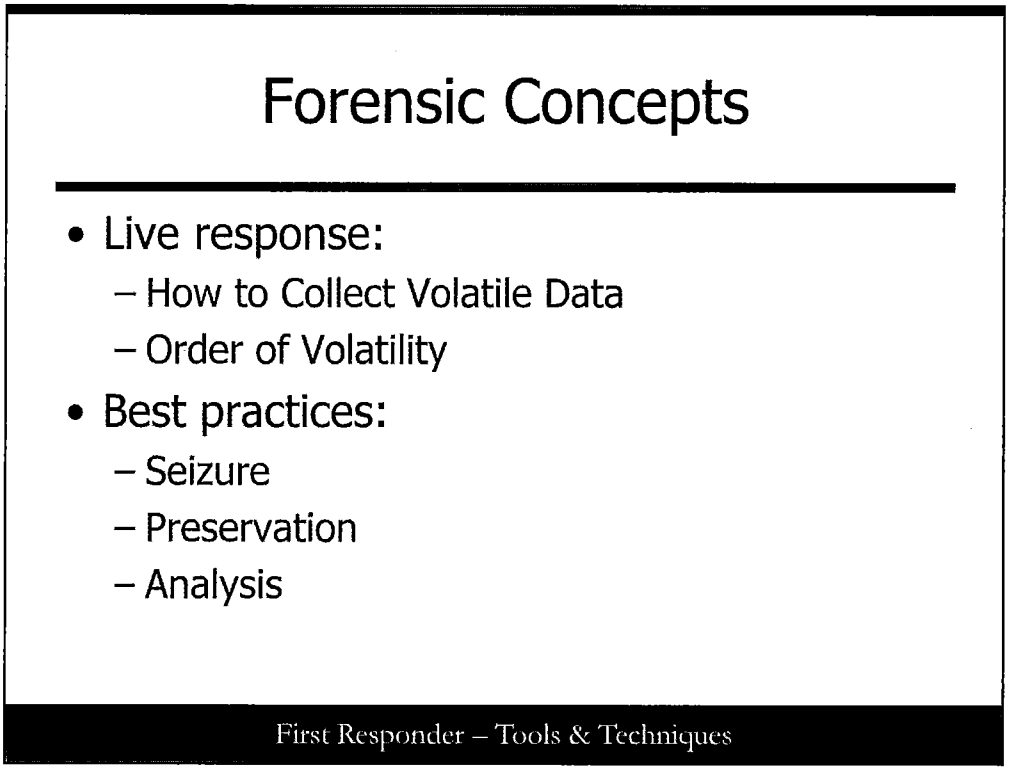
Preliminary Definitions

A computer security-related incident can be defined in a variety of ways. This course uses the same definition as the © SANS *Incident Handling and Hacker Exploits* course: "An Incident is an action that results in harm or potential harm to your computer systems or the data on them." Incidents manifest themselves as events or activities that are reported or noticed. These events are often initially noticed due to the anomalous or malicious behavior associated with the occurrence. Notification of an event may come from a variety of sources and can be reported into a number of different organizational units. For example:

- A network outage may prompt investigation by the network team resulting in the discovery of a denial of service situation that gets reported to the information systems department manager.
- Unusual occurrences noted by a system administrator may indicate an unknown user is actively on the system, which is subsequently reported to the business process owner.
- An automated security tool may detect and alert the information security team that an attack on a known vulnerability has been attempted.
- The marketing department may notice that a web page has been defaced.
- A customer may call the fraud department and report that his account information has been fraudulently used.

It is often the job of the first responder to determine whether an event is indeed an incident. This is known as confirmation or verification. During confirmation it is important to use tools and techniques that do not potentially invalidate the integrity or availability of the data that may later be needed for analysis. This is where digital forensics comes into play.

Digital forensics involves appropriately preserving the integrity of data and analyzing the data in a scientifically unbiased manner. Digital forensics applies to both host-related and network-related data; this includes anything involved in the input, processing, output, transmission, or storage of data. Digital forensic procedures must be performed in a sound manner using tested and accepted tools so that the results can be reproduced for legal proceedings if necessary. Effective incident response involves the use of digital forensic principles and techniques to appropriately respond, should the need arise for subsequent legal action.



Forensic Concepts

As a first responder, your preliminary efforts are usually focused on live systems and data collection. As such, it is important to understand that your actions can have a significant impact on operations and the ability to uncover the details of the event. Although live systems are inherently changing, when performing procedures on a live system, we are introducing processes. If, as a first responder, you find it necessary to actually seize a computer, then be aware of the organization's policy and position on such activity and/or your legal right to do so.

When you perform live response tasks, much of the data collected helps to identify things, such as running processes, open ports, established connections, and users logged in. All of these help you understand the current state of the system, but also can be used later during the forensic process to substantiate findings and correlate activities. After data is successfully preserved in a live response scenario, responsibilities may include conducting cursory analysis of the volatile output or even cursory media analysis.

Best practices suggest that a live response should follow the order of volatility, which means that you want to collect data that is changing the most rapidly to the least rapidly. Data should always be collected onto a removable media or across the network, and you should avoid writing information to the internal disk wherever possible. The order of volatility is

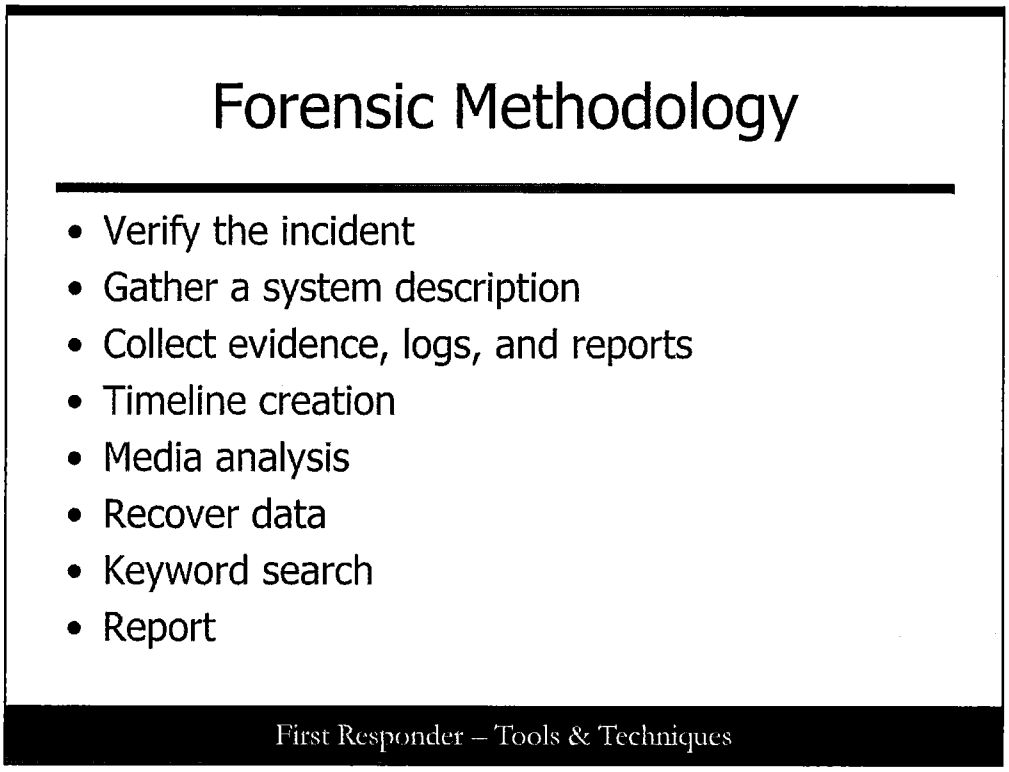
- Memory
- Swap or page file
- Network status and current/recent network connections
- Running processes
- Open files

All data that is preserved must be handled appropriately, which includes documentation of the method and software used, commands issued, and the time the commands were issued. The media should be secured in a lab environment when being analyzed or a safe when not in use. The documentation should also include custody forms and other methods to show the validity and integrity of the image, such as hash values. Later in the course, various tools will be shown that can be used to collect and analyze system data. Live response examples include:

Network data: Information that is presented from a network device such as an intrusion detection system (host-based or network-based) or perhaps a real-time capture of an attempted or successful compromise, or unusual traffic.

Volatile data: Information that is consistently changing such as network state, memory, and processes.

Persistent data: Files (especially logs) and other artifacts that are not currently changing. (Although, if the system is running, then certain files could be changing.) A full bit-for-bit disk image can be performed in a live manner.



© SANS Forensic Methodology

System and network forensic response should follow a sound, well-thought-out methodology. As a first responder, you need to be aware of the time value of information and accurate collection of information. Forensic analysis is a lengthy process designed to dig through the gigabytes of data that is on a disk drive. The first responder typically does not perform forensic analysis; however, it is important to understand the methodology so that the actions taken do not conflict with the forensic efforts that will take place when the incident has been contained. The following describes the forensic methodology in greater detail:

- **Verify the incident:** We want to make sure we spend time appropriately. Some additional questions here are usually warranted to make sure that you are actually dealing with an incident.
- **Gather a system description:** First responders need to record the location of the system, assigned user, serial number of the system, serial number of the hard drives, and general condition of the equipment. As part of the system description, the system name, MAC address, and IP address may also be recorded at this time if known.
- **Collect evidence, logs, reports:** As a matter of good forensic practice and best practices in the industry, analysis should be done only on a bit-for-bit copy of the suspected media. In addition, it is best practice never to work on the original image/copy but to create a working copy. In case something goes wrong, you can always make another image/copy of the original image/copy provided it is still valid.) The image should be validated (preview the image to ensure it contains a good structure) and verified (compare the acquisition hash to the image hash to ensure they match).

By imaging the media with tools such as dd or Ghost and analyzing the copy, you preserve the original media for later analysis so that the results can be recreated by another competent examiner if necessary. Information that is gathered should always be placed on a removable media. Avoid putting information on the system disk whenever possible. During this phase, we preserve and collect volatile data first starting with the system date/time date information, and proceeding in the order of volatility. One of the advanced forensic analyst skills is constructing digital signatures of the data you collect. You should be aware that another person may question the integrity of the information that you collect, so each piece of evidence must have a detailed record of where it has been and who has examined it and must be physically secured in a locked container when it is not used. By not having sound evidence handling practices you can often cause the evidence to come into question and risk it being inadmissible in court. The following are some of the key steps in performing these practices:

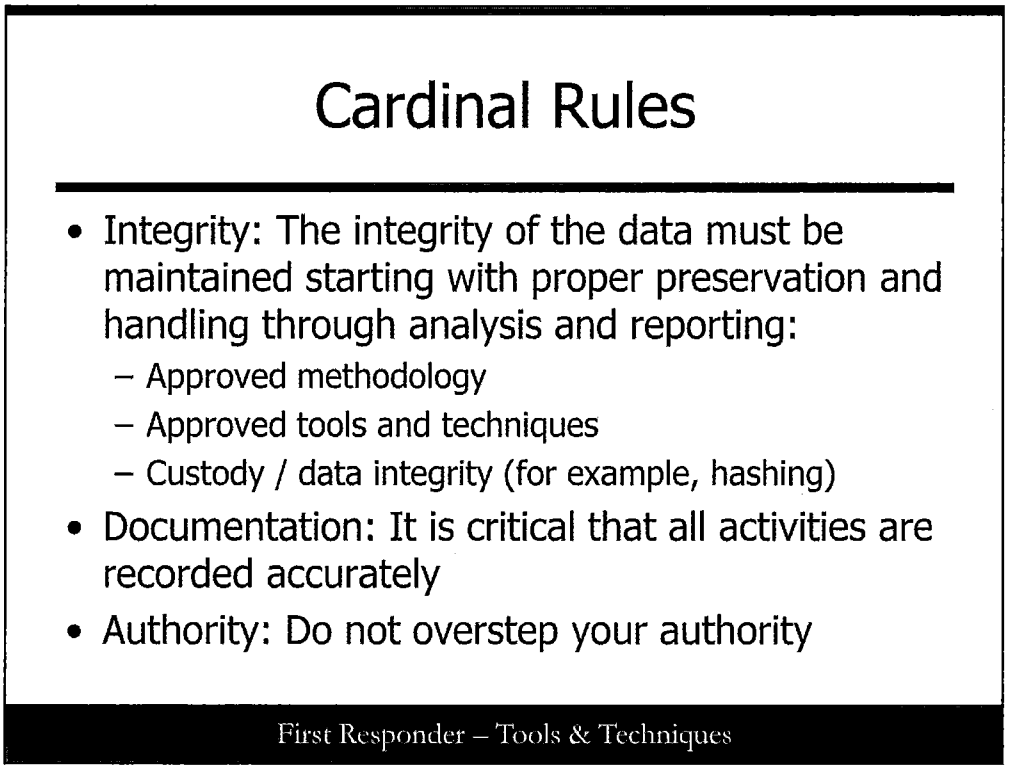
- **Timeline creation:** A full forensic analysis is usually performed with tools designed to analyze files on the media that show when files were created, accessed, and modified. By going through the painstaking process of putting files and their times in order on a system, you can get a sense of what someone has done to the system. Note that there are commands that can change these times, so they are not always as reliable as we want them to be. There are three main times we are interested in: the modified, accessed, and created times of files on a system.
- **Modified:** This is the last time that a file was written to and is normally used to see what files have had their contents changed over time.
- **Accessed:** This is the last time a file was opened and read. You may see this time change when certain programs access a given file, such as an antivirus scan or the UNIX find command working over a file or directory.
- **Created / Changed:** This is the last time that the directory contents for the file changed or the time a file was initially created. This time would be updated if, say, the permissions on a file changed or if it were compressed/uncompressed. Normally, this time should be updated when files first appear on a system.
- **Media analysis:** This is the process of searching out detailed information on the hard disk of the computer to determine how it was used, when it was used, what was written to the disk, whether information was erased, where malware is hiding, and so on. Media analysis is an involved process and is operating system-specific. You need to have a good, sound understanding of the operating system files and directory layout and how the operating system uses disk resources. Each case is different and will have different relevant files and directories that need to be checked on. For instance, a virus or other malware might want to hide in an operating system directory, whereas a user who is storing copyrighted MP3 files might organize them in some sensible manner (such as genre, artist, and album). Example searches here look for files that are particularly large such as sniffer data or movie files.

One of the most valuable tasks you can do is to verify the integrity of operating system files and startup/running environment (/etc or the Registry). By looking over the operating system directories, you can often find malware that is hidden on the computer. Note that to make effective use of this technique, you need to have an idea of what is supposed to be on the system in the first place, so it pays to examine "known good" systems.

Another form of analysis here is analyzing the system against a known hash set. Determining usage patterns such as browser history, Recycle Bin, or temp areas is another way of seeing how the computer was used. There are tools to help reconstruct disk areas of the more popular applications on the market.

When performing media analysis, forensic best practices suggest that the examiner never work on the original media (unless there is absolutely no other alternative—very unlikely). In addition, forensic best practices also suggest that the examiner does not trust potentially compromised systems (which includes binaries and logs). The following are the steps that would be performed:

- **Recover data:** Often people like to try and delete data from the system in an attempt to cover their tracks. We know that in Windows if you use Explorer, data gets sent to the Recycle Bin, and it needs to be deleted from there. But there are still traces on the system for some period of time; after all, companies do sell data recovery software, don't they?
- **Keyword search:** This is the process of searching the media to find more clues. Often those clues are file entries that, when coupled with a growing timeline, allow the forensic analyst to reconstruct the events. Using a solid keyword list, it may be possible to find files of interest on the disk media. For example, a remote BotNet / Remote Access tool often has IP addresses or other keywords that can be used to identify the malware. As a first responder, you should work with the Incident Handling team and forensic analyst to help provide the best keyword list that you can.
- **Reporting**: Writing the report is one of the most difficult tasks. The written report needs to be factually accurate and free from speculation or bias. A professional forensic report should include an executive summary, including a description of the incident and the overall findings. In addition, the report should include a description of the data preserved, a detailed explanation of the procedures performed, and a summary of the facts.



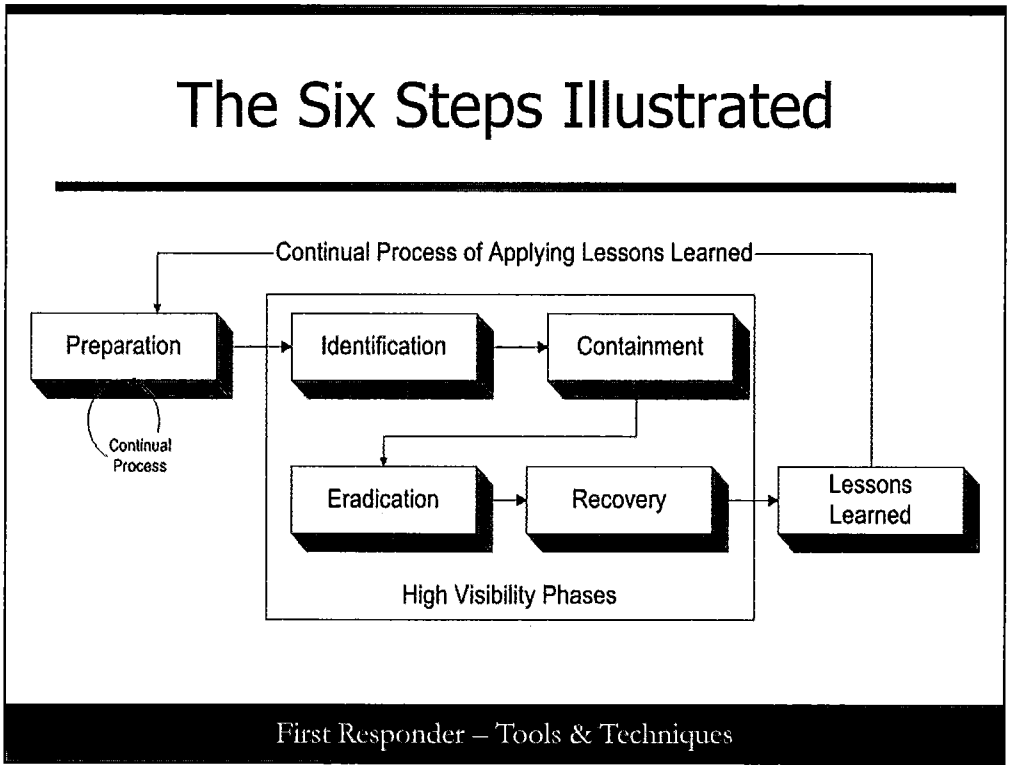
Cardinal Rules

When performing incident response and digital forensics, it is imperative to consider the following cardinal rules:

Never work on original evidence unless there is absolutely no alternative or it is a matter of life or death. Although it is desirable to have quick results, this is the biggest mistake made by system administrators and security analysts that are not trained in first response techniques and forensic principles. The integrity of the original data should be preserved in a manner that is forensically sound. For best evidence, the original media is generally considered best evidence. For example, if the system is in a post mortem state, the original hard disk drives (HDD) would be considered best evidence. A bit-for-bit copy or image that can be verified and validated, and that is properly handled with respect to custody, is generally considered equivalent. If the original HDD is placed back into the system for continuing operations, the copy or image then becomes the best evidence for that particular point in time.

When you arrive on scene, you must articulate that any actions taken by the responder should be done in a manner that preserves the integrity of the data whenever practical. This includes the use of tools and commands that are tested and trusted. Running tools from a subject operating system can potentially provide incorrect results or have adverse effects such as in the case of a rootkit or Trojan. In addition, a hashing algorithm such as MD5 or SHA256 should be used to validate that the data preserved is the same as the original data, that the tools used are the actual tools that have been approved for use in incident response, and that the output from any command or tool is recorded so that the results being analyzed can be verified subsequent to the onsite analysis. Finally, a proper chain of custody must be maintained on the original/best evidence, tools used, and output analyzed.

Along with the preceding rules, it is also imperative that every action is properly documented. This means recording the date/time of the actions taken, the syntax of the tools or commands used, the specifics of the associated systems(s), and the results of your actions. Keep in mind that when electronic tools and commands are issued, a hash of the tools and commands, as well the output can serve as excellent documentation. This is not an option when activities are observed on the screen or when a description of the event is given. Not recording these items is the second biggest mistake made by people not properly trained in first response.



The Six Steps Illustrated

Here is an illustration of the © SANS Six Step Incident Handling Methodology that provides the framework for effective incident response. You can see that there are two feedback loops revealing that incident response is not a single event; rather, it is a process of continual preparation and improvement based on best practices and applying the results of an incident to your environment. We should always be preparing for incident response. On the same line of reasoning, we should also always be improving after an incident.

Incident handling is the action plan in dealing with intrusions, cyber-theft, malware, and other security-related events. The first responder role is critical and should be considered as such. By having both foundational skills (this course) and steps/procedures in place, you will be better equipped to deal with these events as a first responder. It is important to note that a fully trained and qualified incident handler should perform any first responder-related activities; however, this does not mean that a first responder can necessarily perform all incident handling tasks that might be required during an incident. The purpose of outlining all the tasks is to familiarize the first responder with the associated tasks within the overall methodology.

A key concept associated with incident response is the concept of determined action. This is an action that is designed to produce some result or information that guides the next step in the process. Although identification of an event is important, verification that the event is an incident is perhaps more important. When determining if an event is an incident, the first responder needs to take actions designed to provide information and guide subsequent activities. These actions must be taken in the best suitable manner, given the business constraints and regulatory environment involved.

Therefore, it is important that the overall goals are determined as soon as possible. Essentially, this means that management must decide if they want to collect or preserve data at the risk of allowing the incident to remain pervasive until this process is completed, or if the primary goal is to limit exposure, risk, and liability. Keep in mind that there can be compromises between these two diametrically opposed objectives.

This course focuses on the identification phase of incident response because this is the primary duty of the first responder. As we progress through this course, the specific steps in this methodology will be highlighted as they relate to the first responder framework so that the student has a better understanding of the overall methodology.

The Six-Step Process for Incident Handling

Based on the importance of incident response across the industry, it is important that a clear and standard process be followed. To create a starting point, the U.S. Department of Energy (DOE) led an initiative to build a six-step process in the early 1990s. The six-step process used in this course and throughout the industry is based on the original process developed as part of a joint effort led by DOE.

The six steps listed here can help serve as a roadmap or a compass, if you will, to develop a phased approach to incident handling. Keep in mind that for this process to be successful, each step must be followed. The six steps are:

1. Preparation
2. Identification
3. Containment
4. Eradication
5. Recovery
6. Lessons Learned

Preparation (1)

The preparation step is the first and most critical step of the incident-handling process. The tasks associated with this step must be performed in advance, before the incident has occurred. This is the reason why it is often overlooked—or even skipped. SANS recommends that you spend enough time preparing all the elements that are required during an incident, with the goal of increasing the efficiency and success of your incident-handling efforts.

When it comes to incident handling, planning is everything, and preparation plays a vital role. It is important to have a policy in place that covers an organization's approach to dealing with an incident. One item that a security policy needs to cover is whether a company is going to notify law enforcement officials or remain silent when an incident occurs. The answer to that might depend on the severity of the incident; if so, what guidelines should the responder use to decide whether to call? If you are going to contact law enforcement, have a list of phone numbers for each agency you might need to involve.

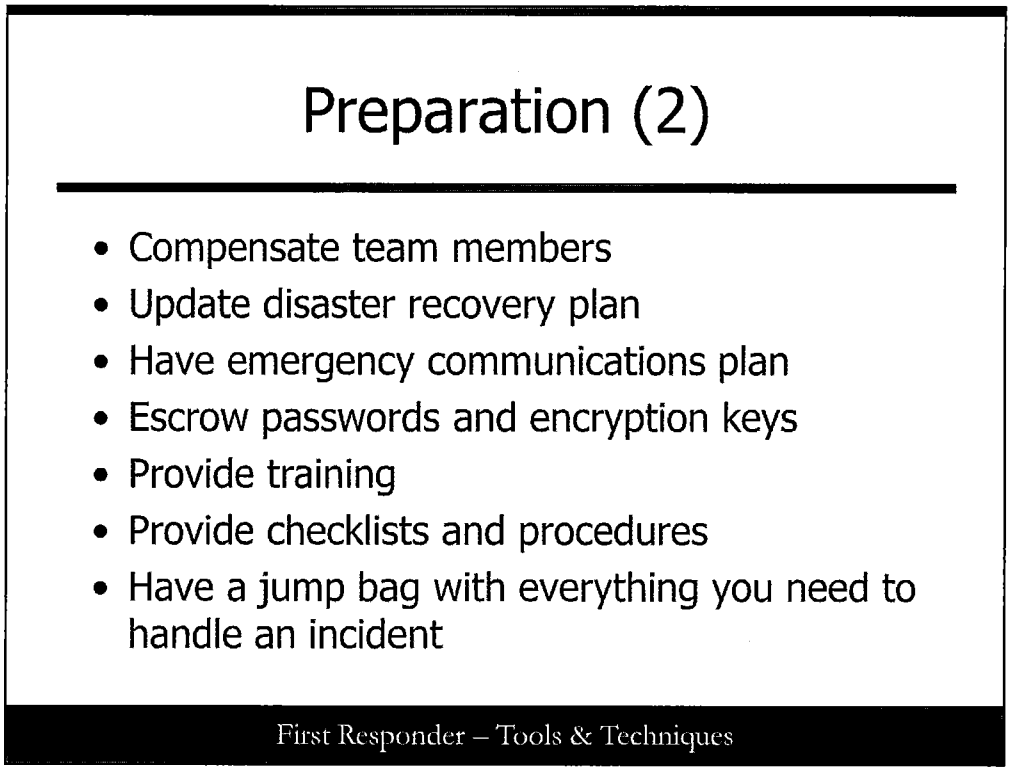
Another important item to consider is whether to contain the incident and move into clean-up phases or to observe the attack in an attempt to gather more evidence. The policy should also contain direction for intra-organization incidents and how the company works with other companies regarding an incident.

Incident handlers can be under extreme pressure. Consider a worm that infects your entire infrastructure, effectively making your network systems unusable. This is one reason incident-handling

teams must never rely of Voice over IP. If you have a VoIP installation, consider the use of cell phones, walkie-talkies, or some other back-up method of communications. Incident handling can become a large-scale effort involving many people on many systems simultaneously. This should be taken into account during the planning phase.

The time to make these types of decisions is before the incident, keeping senior management and legal staff apprised of any changes to policy. Because of the sensitive nature of incident handling, any decisions made could greatly affect your career down the road if you did not get approval or reach consensus with management. The last thing you or your company wants is for senior management to question or doubt the decisions that were made during an incident.

When it comes to selecting members of the team, keep in mind that not everyone makes good incident handlers. There are some smart people in this industry whose personalities do not lend themselves to work under immense pressure and as part of a team. People who like to work solo and need to be the hero usually do not make good team members. Ideally, a person should have a strong technical background, thrive in a team environment, and have the ability to make sound decisions grounded in reality.



Preparation (2)

As the incident response team begins to mature and has responded to several large incidents, it is possible that members of the team will get burned out and leave the team. Although this is certainly understandable, an approach you might want to take is to provide compensation and other rewards for members of the team. This might run counter to your current policies, but keep in mind that incident handlers are often called to perform their duties after normal business hours, weekends, and holidays while under a lot of pressure to get things restored as quickly as possible.

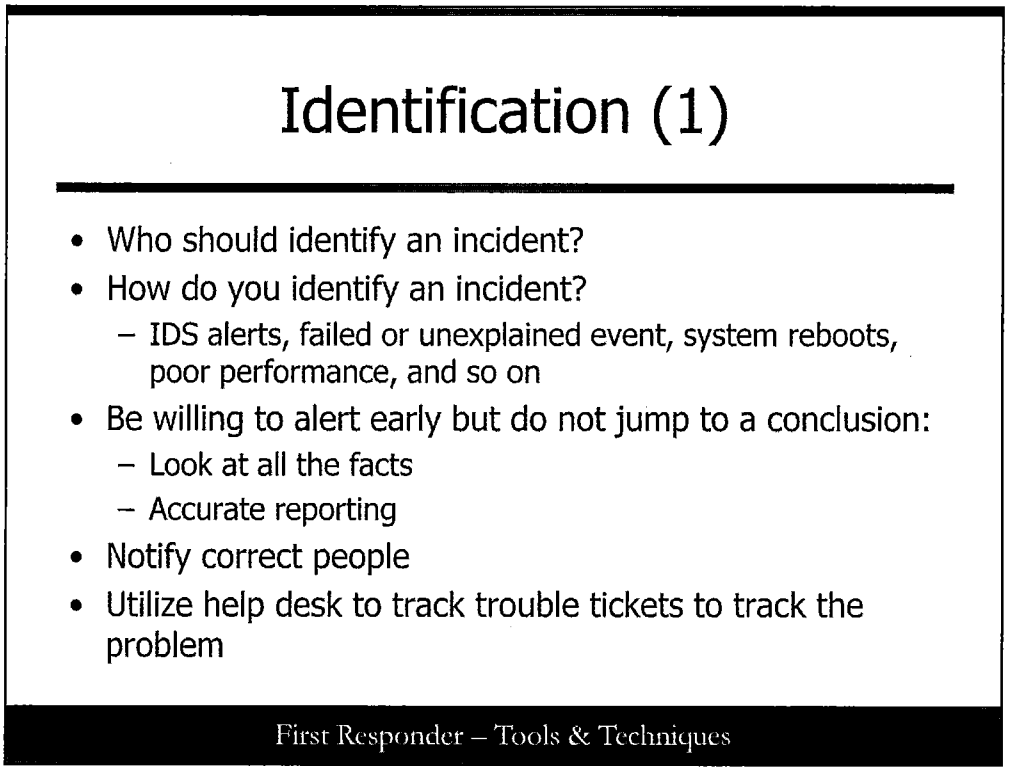
During the preparation phase, an organization should make plans to update its disaster recovery plan to include incident handling. After all, what is a disaster? It is an incident and needs to be handled as such. Although disaster recovery plans are often thought of as a checklist to get a business back up and running as quickly as possible, the skills possessed by the incident-handling team could be put to good use to reach this goal. In addition, the disaster recovery plan and the incident handling procedures guide should contain information for emergency communications.

The issue of making privileged passwords available to others can be a delicate situation. However, in an emergency, a handler might need access to critical systems.

One idea to consider is to incorporate a procedure in which system passwords are kept in sealed envelopes in a locked container or data center until they need to be used. This might seem cumbersome, but it does work and keeps the passwords private until they are needed by the incident-handling team. For this to work, the system administrators must keep the passwords in the sealed envelopes up to date, and the incident handlers must make every effort to tread lightly on the systems, inform the system administrators of any changes made, and above all, never use a privileged password unless they are qualified on that operating system. One thing that will certainly make an incident worse is having someone who has no idea what he is doing issuing commands as administrator or root.

Our computing environments are complex and change over time. Training is critical for each member of the incident-handling team. Memory fades over time, especially if the members are not working on honing their skills on a regular basis. Having a checklist on how to bring a system down safely or on how to restore a system from tape can help in preventing errors and can reduce the stress on the handler. If your team is following a checklist and the resulting operation fails, it might be the fault of using an outdated checklist on a regular basis, so ensure they are updated to your organization's current environment.

Reaction time to an incident is absolutely critical. Every effort should be made to find members of the incident-handling team who can respond on short notice. For example, an incident handler who has a 2-hour commute into work might not be that helpful for a situation that requires immediate attention. One way to mitigate the effects of delayed reaction is using what the military calls a jump bag. This bag should contain in a central location everything needed to respond to an incident. Items such as contact numbers, checklists, telephone, notepad, pencils, and so on, are items that you would want to include. Also, as far fetched as it sounds, spare network cables, a hard drive, a mini-hub, and tools for working on a PC should be considered essential.



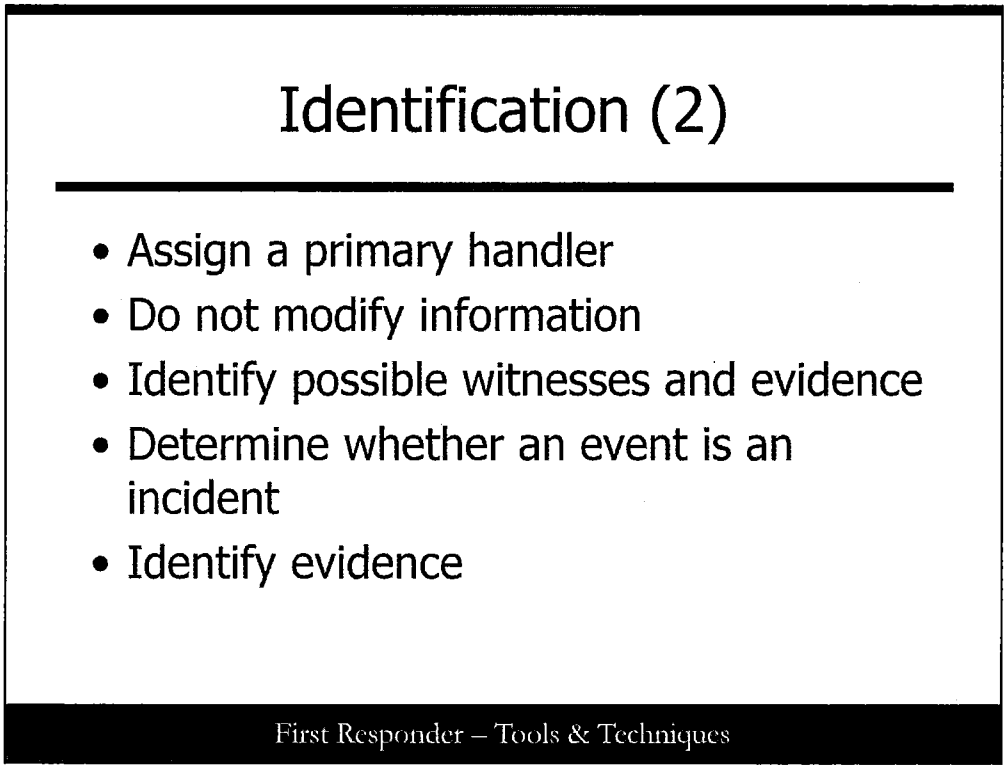
Identification (1)
When it comes to identifying an incident, members of the team should stay with their realm of expertise. You would not want a Windows expert digging around a UNIX system, and vice versa.

Some possible signs of an incident that might warrant further investigation essentially include anything suspicious, such as intrusion detection alerts, unexplained entries in a log file, failed logon events, unexplained events (such as new accounts), system reboots, poor system performance, and so on.

Correctly identifying an incident could be the difference between cleaning up the problem in a few minutes and causing your organization's network to be down for several hours or even days. Obviously, any system outage could potentially cost your company a lot of money, so it is important to identify an incident correctly the first time and respond accordingly. For example, after a fire alarm is pulled and a building evacuated, qualified firefighters respond to the scene and investigate. Only then does the firefighter in charge at the scene authorize re-entry into the building. This should be the paradigm we work under—be willing to alert early, have trained people look at the situation, and stand down quickly at a minimum of expense if nothing is wrong. No matter which course of action you decide to pursue, make certain you have mechanisms in place to correctly identify an incident.

There is nothing wrong with alerting early if you maintain situation-awareness and everyone understands this might not be an actual incident. All attempts should be made to avoid overreacting to the situation and escalating it too fast, only to realize an hour later that you made a mistake. If that happens enough times, you could fall victim to the "boy who cried wolf" syndrome; and then when a real incident occurs, no one will believe you because of the false alarms.

Chances are that your organization has a 24x7 help desk operation that would be ideal for helping out with tracking the incident and maintaining a paper trail. They could also be utilized to facilitate communication and contact other personnel as the situation warrants.

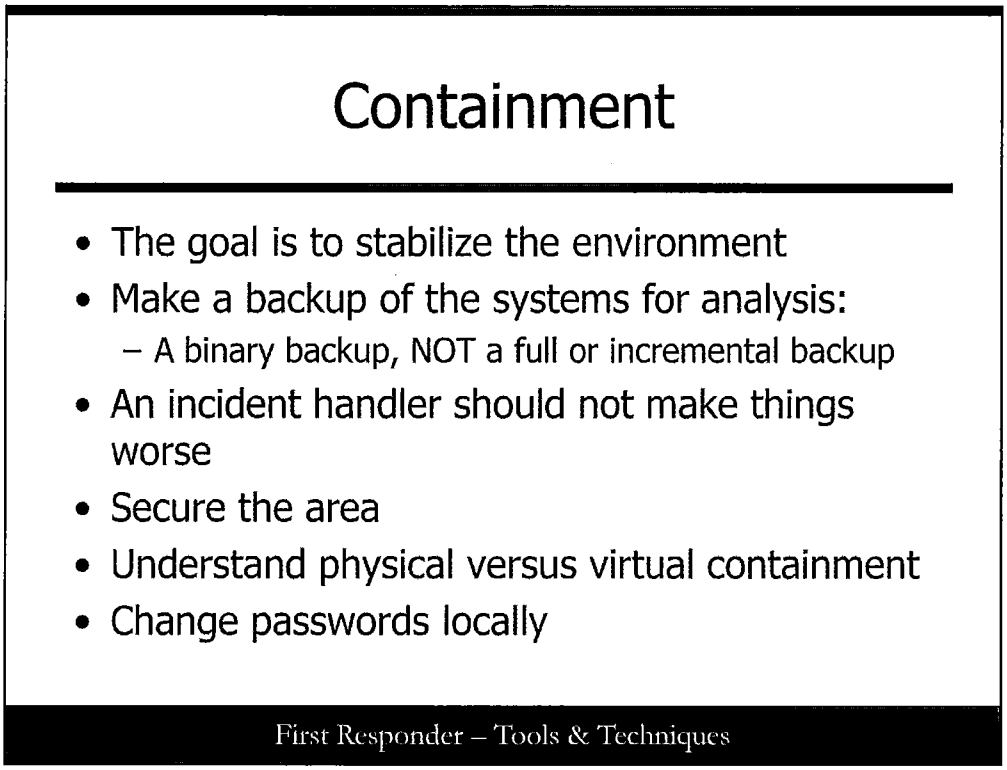


Identification (2)

It is important to keep in mind that a primary handler should be assigned as a team leader to keep the process flowing while also making sure that no steps are overlooked or missed. For smaller incidents, often of the "Would you check this out?" category, there isn't a need to send a core team of incident handlers. It is a recommended practice to have a core team of well-trained handlers and also have incident-handling skills and training as part of the job description for security officers and system administrators. An organization that adopts this approach benefits by having multiple layers of "firefighters."

However, in such a case, it is important to assign tasks in a way that encourages cooperation among the team and allows all members to succeed. When assigning tasks to part-time members of the team, do so in a way that it is clear what is expected of them: the quality of their investigation, their responsibility to preserve and collect evidence, what documentation they should produce, and when it is due. It is also important that they know who they should contact if they feel they need additional guidance or support.

After you determine that the event is actually an incident, the handler might decide to take the steps needed to build a criminal or civil case. In this situation, witnesses should be identified, and a written statement of what they heard or saw should be taken immediately while the information is still fresh in their minds. If a decision is made to involve law enforcement, make sure senior management is notified, unless you have a detailed policy to follow.



Containment

Okay, we have spent countless hours preparing for the eventuality of an incident. We have a good idea of what it takes to identify an incident, but where do we go from there? Identifying an incident solves only part of the problem. We are still left with the task of isolating and eliminating the source of the incident. This section discusses some steps that can be taken to contain an incident and, hopefully, limit its damage to the organization.

The primary responsibility of the incident handler is to make things better while adhering to the basic principles of liability and negligence. Negligence for failure to meet a certain standard of care is generally determined by a court of law. Specifically, negligence is defined as, "the failure to exercise the degree of care expected of a person of ordinary prudence in like circumstances in protecting others from a foreseeable risk of harm in a particular situation." In other words, a handler is responsible for meeting the expectations of the prudent person rule. Typically, a company that acts reasonably or with due care generally will not be found negligent.

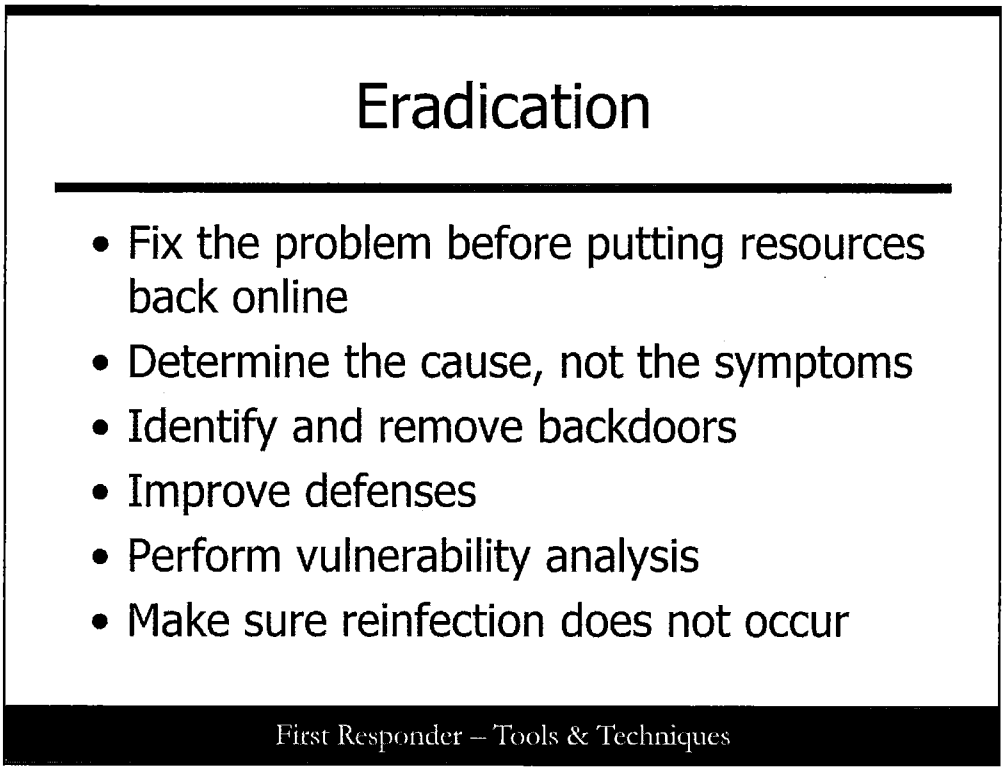
There exists a potential for an incident handler to run into trouble while performing her duties. There is no aspect of incident handling that allows a handler to break the law. For example, if you suspect someone within your organization of downloading child pornography, you can't download these files to your computer to examine them. Also, a handler needs to exercise due care with regards to a person's privacy under the Electronic Communications Privacy Act.

For instance, if you are an Internet service provider, you cannot just release the personal information of a subscriber simply because someone claims she was attacked from the subscriber's IP address.

You should also be aware that corporate officers within your organization might be held liable for your actions if they are considered unlawful.

In containing an incident, you must first secure the area. In doing so, a forensically sound backup should be made of all infected systems. If the original hard drive cannot be kept for evidence, multiple copies of the backups should be made for future analysis, if needed. One copy should be kept for evidence and the other copy used to analyze the incident. At some point in the containment process, a decision needs to be made of whether the systems should be pulled off the network or whether the entire network should be disconnected from the Internet. Also, passwords should be changed as soon as possible to make sure a compromised account couldn't be used for reentry into the system by a remote attacker.

One of the key aspects of the incident-handling process is to be present, with a high level of detail, the different pieces of evidence found, and all the actions performed during the whole process. For this purpose, you should take detailed notes of all the events associated to the incident, from the Identification (step 2) to the Recovery (step 5) phase, preferably using numbered paper notebooks.



Eradication

Before the system goes back online, an incident handler must make sure that she fixes the problem or the vulnerability that the attacker used to compromise the system. At first glance, the tendency might be to wipe out the entire operating system and rebuild it from scratch. Although this is certainly an effective way to remove any malevolent code, the opportunity for re-infection via the same channel still exists. There are a myriad of cases in which systems were taken offline, rebuilt, and put back on the network only to be compromised again within minutes or hours. This is because a root cause analysis wasn't performed to determine why the incident happened in the first place.

It is not enough to simply recover the system and put it back online: The underlying security mechanisms of the affected systems must be altered, fixed, or upgraded to accommodate any new vulnerabilities. If it is a production system, you might hear voices of dissent from the organization about modifying a server running on a production network. This is an important, and to an extent, valid argument, but the counter is that if the system were compromised, then it must contain a vulnerability that might exist on other servers and could be exploited on a continual basis until the problem is fixed. Further, manually cleaning up the damage from an incident does nothing to prevent the problem from occurring again unless the problem is accurately identified and removed, patched, or otherwise mitigated.

Attackers often try to establish additional ways of ensuring remote access to the compromised system, so they have control of it even if the vulnerability exploited originally is fixed. Such backup access methods are known as "backdoors," and are implemented using several methods. Some of the most common ones include a process of listening on a specific port and offering shells access (without requiring authentication), creating a new user account with high privileges, and scheduling jobs that periodically

run programs that open new paths to access the system. As a wide incident handler, you need to not only fix the vulnerability used during the initial system compromise, but also identify and remove every additional backdoor left by the attacker.

After the system is recovered, it is a good idea to run a vulnerability scanner against the affected system to see whether the problem is, indeed, fixed and that no new holes were opened up in the process. A number of commercial products, such as ISS Internet Scanner, work well and produce nice-looking reports, but open source tools such as OpenVas should not be overlooked. If your organization is on a tight budget, and you need tools that perform the task with great efficiency, then you owe it to yourself to explore the open-source options available.

To sum up, your main goal as an incident handler is to make sure that a new compromise using the same, or even a similar, vulnerability does not happen again.

Recovery

The key point to consider in the recovery phase is to ensure you are not restoring vulnerable code that has already proven itself to be exploitable by any number of attack methods. For example, if you restore a system from tape backup, then you could be restoring a previous state that contained the vulnerability exploited by the attacker. Vulnerable code, in this context, refers to operating system software that hasn't been patched to the latest levels, source code, and/or application software used on the affected system. Although there is no easy solution, using a file integrity tool such as Tripwire might help in restoring the system to a known good state. Use Tripwire to take a snapshot of the compromised server, restore from tape backup, and run Tripwire again to compare the results. This method can tell you exactly what files were changed, modified, or deleted during the exploit, and it gives you a better understanding of how the attack occurred and what can be done to prevent it from happening in the future.

The two main options available when restoring a compromised system follow:

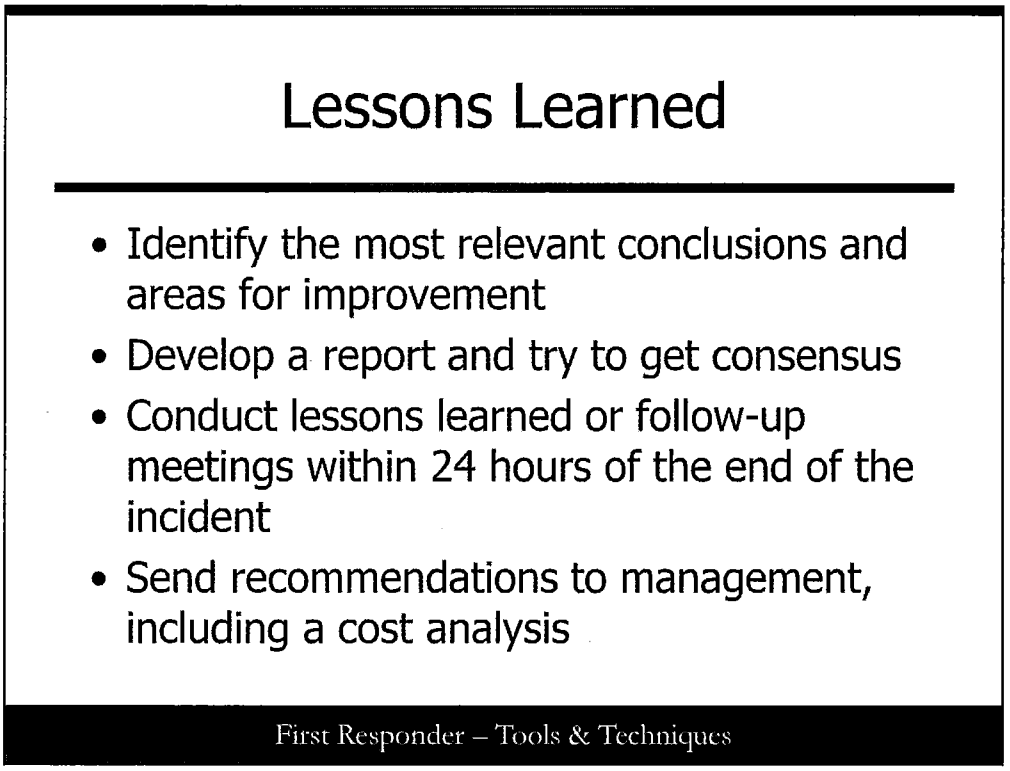
- Installing the operating system (OS) and applications from scratch using the official and original media, adding the latest OS and application software updates (fixing the vulnerability exploited during the incident), and finally restoring the data from a backup.
- Restoring the system from a trusted backup and patching the system, at least fix the vulnerability involved in the incident. The trusted backup already contains the latest system and application data available.

Before the system can be brought back into production, the incident handler needs to validate the system along with the system administrator. Removing the vulnerability could have affected other

functions of the system that are deemed critical by the business. Anything that breaks after the recovery is likely to be blamed on the incident handler, so every effort should be made to ensure the system works as normal before turning it over to the system administrator.

In addition, the decision on when to put the system back into production has to be made by the system owner. The handler can give advice and be as helpful as possible, but, ultimately, the final decision of bringing a system back online rests in the hands of the system owner and/or administrator.

It should go without saying that if the eradication were not complete, or the infection vector were not closed off, there stands a chance of re-infection. Monitor the systems closely for the first few hours of operation to see whether anything crops up that could be attributed to the original incident. Monitoring also helps demonstrate to the organization the importance of an incident-handling team and the dedication of the team members to ensure the problem is taken care of correctly.



Lessons Learned

After the system has been restored and is back in operation, a report outlining the entire process should be drafted by the primary incident handler. It is important to summarize the incident, identifying the most relevant conclusions obtained to aid in avoiding similar incidents in the future. The report should contain areas for improvement, both in the security infrastructure and in the incident-handling process. In addition, the report must point out new security actions or projects identified during the incident and that must be implemented to increase the overall security of the IT environment.

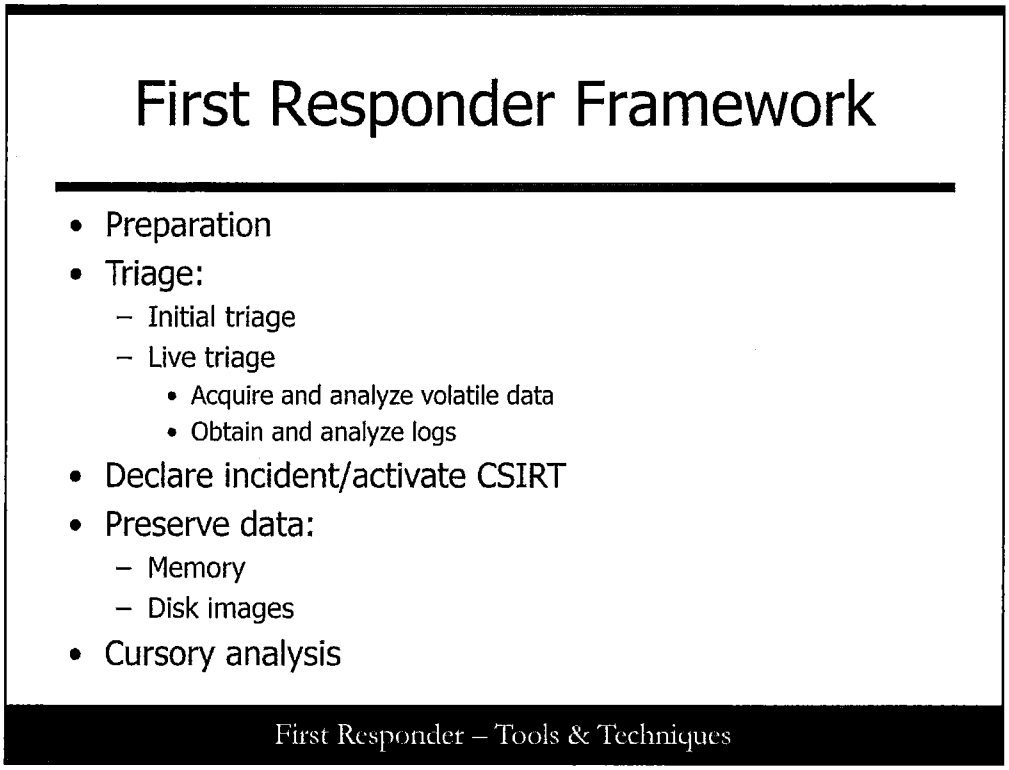
The goal should be to get consensus with everyone involved. After the report has been drafted, all members of the incident-handling team should meet for a "lessons learned" overview. The goal of this meeting is to create a list of items that need to be included in the executive summary of the report. The executive summary should contain a brief synopsis of the entire incident, including the steps taken to recover and recommendations made.

Key Mistakes in Incident Handling

Conducting a follow-up meeting with all involved parties is never a fun task, but it is vital to making sure the organization understands what happened, why it happened, and what steps were taken to make sure it doesn't happen again. During every incident, mistakes occur and there is a tendency to place blame; however, the goal of the follow-up meeting should be to improve the process and learning from the mistakes.

Some key mistakes that are common in many organizations are listed here:

- Failure to report an incident or ask for help
- Incomplete or nonexistent notes
- Mishandling or destroying evidence
- Failure to create working backups
- Failure to contain or eradicate the incident
- Failure to prevent re-infection
- Failure to apply lessons learned

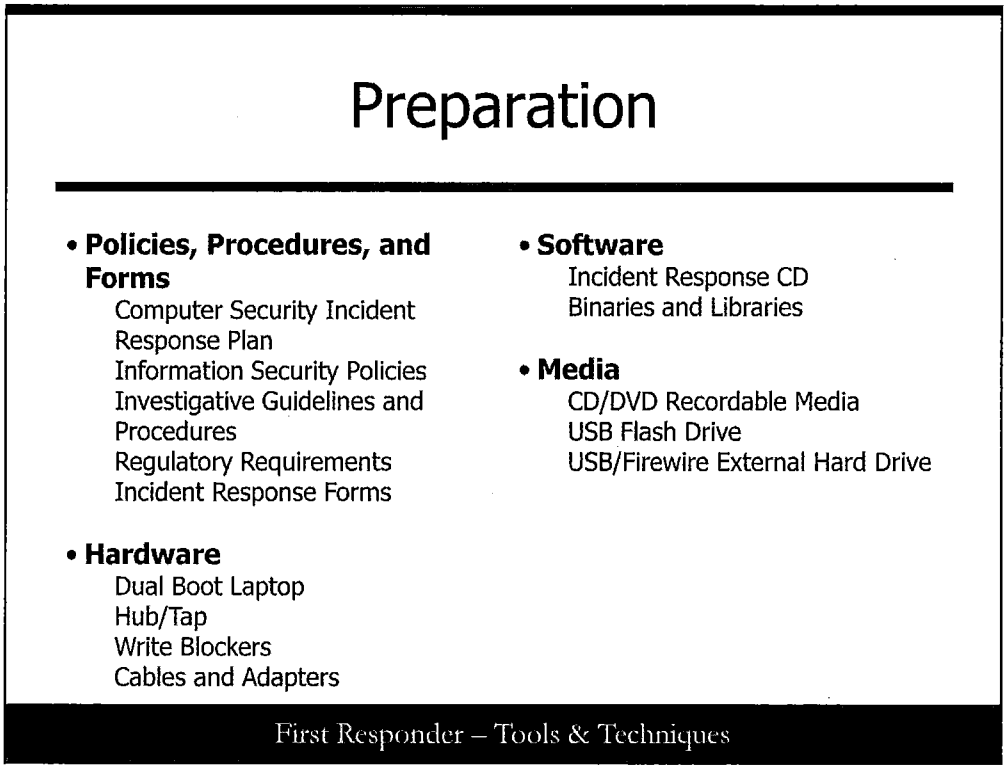


First Responder Framework

As previously mentioned, the focus is on the specific tasks and tools associated with each step in the SANS First Responder Framework. The major components of this framework follow:

- Triage
- Computer Security Incident Response Team (CSIRT) activation
- Acquisition of data
- Cursory analysis

Some components listed require more involvement than others, but each will be discussed in detail in this course.



Preparation

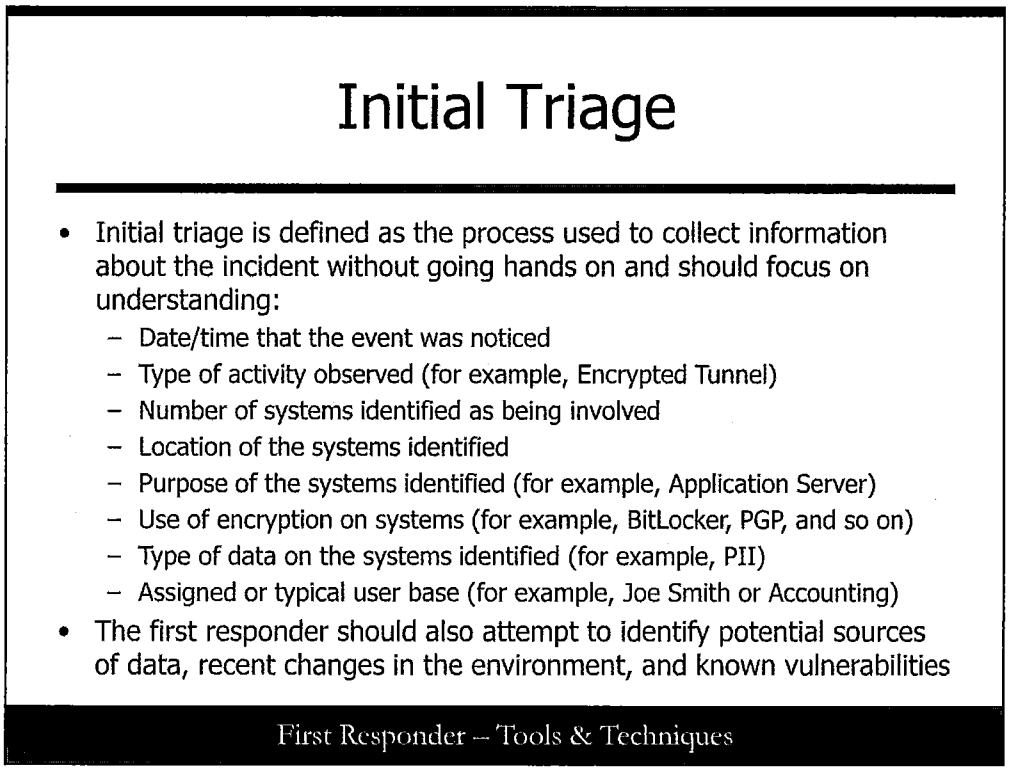
Whenever applicable, the first responder should refer to the Computer Security Incident Response Plan (CSIRP) or the Computer Security Incident Response Team (CSIRT) procedures. Organization-wide information security policies are also important to follow during response, and often overlooked due to the extreme pressure associated with security breaches. Make sure to consider both internal policy and external regulatory requirements. Ultimately, to be prepared, it is ideal if you have knowledge of the security features and auditing capabilities within the environment. In addition, you could proactively recommend that the organization consider incident response concepts such as time synchronization of devices. The first responder plays a critical role in the incident response process on the CSIRT. Primarily, the first responder assesses and secures the scene, collects information, communicates with others on the incident response team, and may take action within their skill base as directed by the CSIRT lead.

You need to have a known, tested, and versatile incident response platform. For example, an Intel-based Apple MacBook Pro is an extremely versatile system that can boot multiple operating systems, including Windows, Linux and OSX. This setup provides the first responder the ability to respond to the most different scenarios. In addition, you want to keep a fast hub or network tap on hand to capture traffic and forensic write-blocking gear for post mortem acquisition.

Throughout this course we utilize the SANS Incident Forensic Toolkit (SIFT) for demonstration purposes. SIFT is an extremely versatile forensic platform; however, we do not utilize all the features on the workstation, and it is not a replacement for a live response CD such as Helix. A live response CD such as Helix provides the ability to boot into a Linux environment, as well as the ability to run tools on a live Windows or Linux host.

To effectively carry out your duties, you might also need to have the requisite equipment and tools in a "jump bag" that is always ready to go. Testing your tools, techniques or methodologies during first response is *not* appropriate. Here are some examples of materials you need to have on hand to deal with an incident:

- **Contacts**: Including at a minimum the telephone number(s) for the primary point(s) of contact
- **Supplies**: Pens, pencils, Sharpie, flashlight, screwdrivers, and such
- **Media**: Blank CD/DVDs, external USB/Firewire HDDs, USB flash drives, and so on
- **Equipment**: Fast Network Hub or Tap, miscellaneous cables and adapters, and IR Laptop(s)
- **Tools**: Course CD or your own CD with known, good response tools that have been tested
- **Forms**: Custody, notes, and acquisition forms
- **References**: SANS, CERT/CC, NIST, DOJ, or your favorite book(s)



When an event is reported or noticed, the first responder will likely be asked to perform triage. Some people think of triage as actually touching the system to confirm if something is wrong but that can have significant adverse effects on the ability to properly analyze the data. In fact, there is a school of thought that would argue that you should not begin collecting data unless you can confirm that there was an incident. If your organization subscribes to this school of thought, it is your duty as a trained first responder to make sure that the decision makers are aware of the implications of conducting live triage too soon. Although this can be debated over and over, it may be a decision that is out of your hands, and depending on the circumstances, it may actually be the correct call from a management perspective. Either way, we must define what we mean by initial triage, and so for purposes of this course, initial triage is defined as the process used to collect information about the incident without going hands-on.

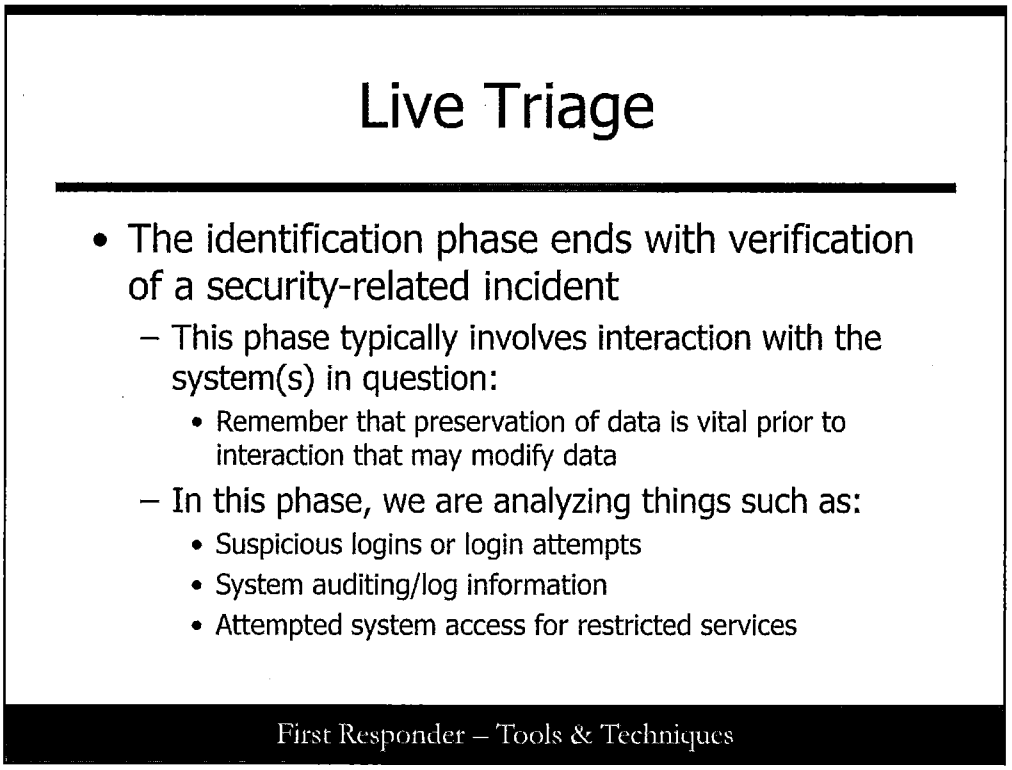
During initial triage, the first responder needs to understand the nature of the reported potential incident to the best of his ability. This includes collecting as much data from the person reporting the event such as:

- Date/time that the event was noticed
- Type of activity observer
- Number of systems identified as being involved
- Location of the systems identified
- Type/purpose of the systems identified
- Type of data on the systems identified
- Assigned or typical user base.

Because a suspected incident can be a legitimate anomalous activity or the result of an improper configuration, it is necessary to confirm that the activity is truly malicious or unwanted. Depending on the nature of the incident and the source of the notification, the first responder should immediately begin to identify potential sources of data that could provide value during the triage process. The answers to the preceding questions are often critical to answer questions from senior executives and legal counsel so that the next decisions are made in an informed manner.

The availability of applicable network device logs and content captures should be determined. Any current host-based or network-based vulnerabilities known to exist within the environment should be discussed as possible leads to investigate. All potential clues identifying the attack vector suspected or applications involved should be listed and reviewed by the response team at this time. Depending on the evidence presented, a determination should be made as to whether the activity resulted from a manual or automated attack.

Although the extent may not be fully known for some time, it is best to get an idea of how many systems may be involved or affected, what the primary role and criticality of each system is, and how these systems are configured. The availability of applicable host-based logs should be determined. It is also necessary to obtain a current understanding of the network architecture including subnets, security and network devices, and external connections. Methods established for remote access such as dial-up or virtual private networks (VPN) should be scrutinized carefully. If recent changes have been made to the architecture or infrastructure, these changes should be noted as they may have created an unexpected or unintentional vector for compromise.



Live Triage

After the initial triage, a decision needs to be made with respect to the declaration of an incident. Often, there is not enough data available to make an extremely informed decision. It many cases, decisions are made based on risk. For example, if the system is known to maintain sensitive, personally identifiable information, that is important to note. Likewise, if the suspected incident is associated with the degraded performance of the primary web application used to take customer orders, then that may influence the decision as well.

Even still, that type of information may not be available or applicable to the incident you are responding to. As a matter of practicality, a first responder may need to perform live triage on a system to determine if an incident needs to be declared. It is at this point that a live response CD such as Helix may be needed.

There are many places that we can find artifacts indicating an incident has occurred or is actually currently taking place. Depending on the nature of the event that is reported, we may need to analyze a live system. Although there is no silver bullet for all security incidents, there are several reactive hands-on steps that should be taken to understand what has happened and limit the overall risk associated with the incident. Because each case involves different components, the decision regarding which steps apply in any given situation will be based on the unique circumstances of the case.

Normally, unless there are unfortunate circumstances, a first responder should begin collecting the most volatile data first and then capture persistent data. Volatile data is data that is no longer available without power to the system. Memory is generally considered the most volatile data that is found on a system.

Depending on the operating system, memory is physically written to a pagefile or a swapfile and may be found on magnetic media if the system were improperly shutdown, but it is not always a safe bet to assume that remnants of memory will be found on a system after the power has been turned off. Because memory can contain extremely valuable evidence, it should be considered unless there is a specific reason or need not to do so.

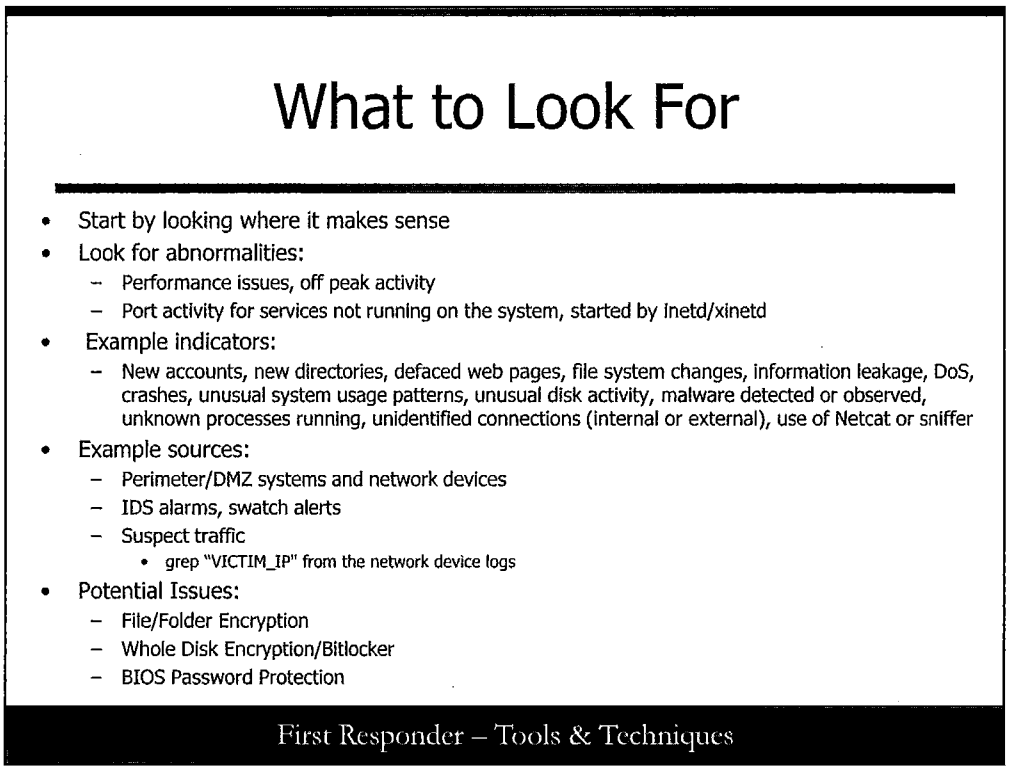
The first responder should also concurrently begin collecting real-time network traffic to monitor ongoing activities and ensuing analysis, as this is also volatile in nature. Capturing live content can provide insight into unusual network activities, large file transfers, and the execution of remote commands. When reviewing content on a system, the responder needs to be sure that explicit authorization exists so as to not violate individual civil liberties and privacy rights.

Systems should be queried to see if unknown user accounts exist. If a root\admin level compromise is suspected to have occurred, the passwords for these accounts may need to be changed (containment). Critical and sensitive files can be examined for modifications, especially if file integrity monitoring is in place. Systems should also be examined for hidden processes and unexplained encrypted files or encrypted traffic. In addition, the most relevant backup tapes should be secured for potential forensic analysis provided the logs are not present. All actions taken and commands entered by first responders should be documented thoroughly to include the system name, time, and exact syntax.

In many cases, the systems involved in an incident are highly critical to the organization's mission and therefore may not be taken offline. When systems may not be taken offline, an additional burden is placed on the responder to properly collect live evidence. Although this is relatively simple for a trained responder to do, it can be complicated and generally increases the amount of time required to collect evidence. In addition, the last thing you want to do as part of your efforts is to bring down a mission critical system, effectively making the situation worse. Therefore, anytime you work on a mission critical system, you should always proceed with extreme caution and notify management of any potential risks or complications associated with the procedures to be performed. For example, if you determine that there is a need to preserve physical memory, you should explain why you would like to perform this action by articulating the benefit of doing so compared to the risk (in this case, the risk that the system will crash).

It is usually at this point in which a decision needs to be made (reference the First Responder Framework) to determine if there is sufficient data to support declaring an incident.

Live triage procedures are covered in the next section of this course.



What to Look For

There are several methods that can be used to help identify whether a system has been compromised. Depending on the method of initial notification, you might already have a good idea of which systems are involved or what is going on. For example, if an automated tool such as swatch is notifying you that a specific system is behaving oddly, or if the event log in Windows sends you an indication of an increase in SMB-related traffic on a system, then you can discern relevant facts from that message. Ultimately, when looking to confirm a suspected incident, you should start in the places that make the most sense. If you suspect it is a network-related issue, you should begin by examining the suspect traffic. If you think it is a host-related issue, or you can identify hosts associated with the event(s), you might want to begin looking at the volatile data on the host. Of course, messages posted to your central log server and IDS/IPS events should always be checked as well.

There are several analysis techniques that you can apply to get a handle on a network security incident. You can look for the top talkers and determine if they are PCs or servers. You can look for unusual port combinations. For instance, one BotNet that the author dealt with used a source port of 153 and communicated to its controller on port 4444. Lastly, using Wireshark (a general purpose protocol analyzer), you can dig into the details of the SMB traffic and see just what type of file/print share connections are being attempted.

When searching for events, consider checking other potential sources of information. By determining what is coming to and going from the system and by checking other systems on the same network segment, you can make some determination of where to start when you get on a suspect system.

Remember that the security of a network and sever environment is built on layers: perimeter protections, network protections, router ACLs, procedures, and system updates. Each layer in an environment can tell you some information that may be useful when you get on the system.

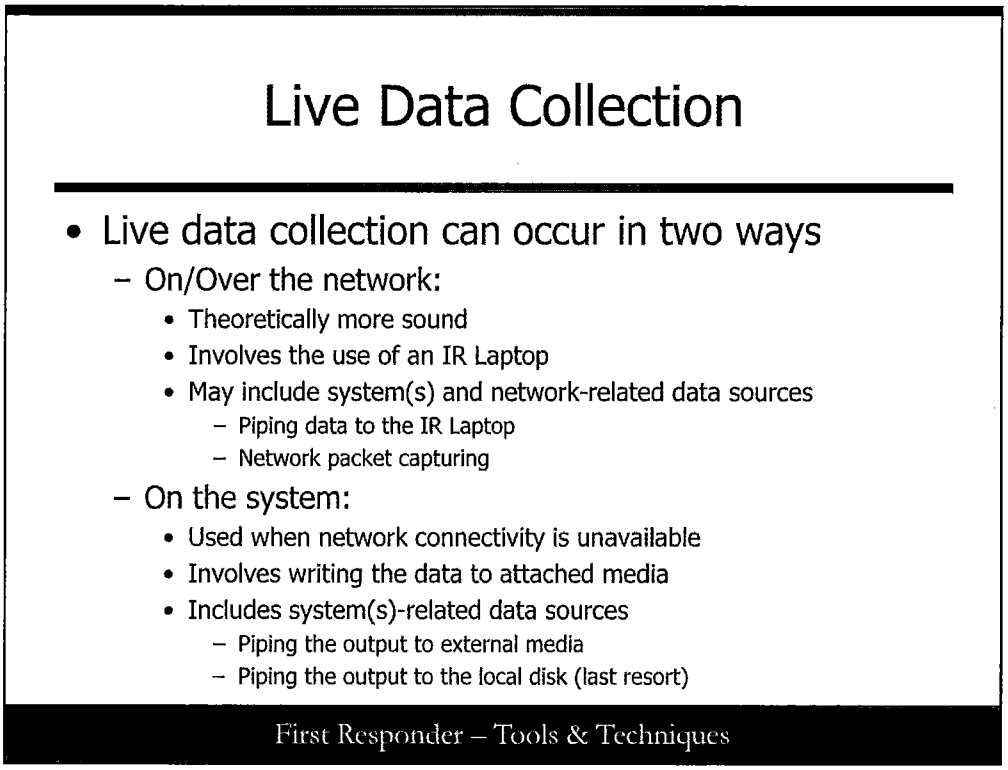
More than likely, the system will running software on well-known and understood ports. There are many clues you can look for on a system to determine if it is used remotely: mismatch ports, unusual or unknown inbound/outbound ports, excessive traffic, or network connection attempts from services that are not supposed to be running. Also, you might find performance issues that might not be easily explained.

If the load on the system is normally approximately 60% CPU utilization and it is now running at 90–100% utilization, this could be an indication of another program that is chewing up spare resources. This could be a real need; additional processing is required. Or it might be a port scanner, IRC bot controller (or Botnet participant) system, FTP "warez" server, or some application a user is running.

In the Windows environment, you can look for many systems attempting to connect to the NetBIOS and/or SMB ports. Also, you can look for the suspect system attempting to make connections to other PCs using these ports. Often, malware or an attacker scans a network; seeing repeated ICMP traffic followed by NetBIOS / SMB traffic may indicate the presence of a worm. In the UNIX/Linux world, most system services can be started from inetd or xinetd. There are also services that are actually started by an init script, such as SSHD and HTTPD (usually Apache).

When some initial data is captured, a number of techniques and tools can be used to glean vital intelligence about the event, such as the simple process of using grep. Obtaining actionable intelligence and data is the focus of this module.

It is important to note that you might encounter a system that uses encryption. Generally, it is a good idea to inquire about the use of this type of technology during the initial triage process. If appropriate, you might obtain the keys or passphrases through inquiry, but this should be done with caution and may need to be in coordination with HR or legal counsel, depending on the circumstances. It may also be a good idea to search the area for external media or paperwork that might contain keys or passphrases in case they are needed later. Also, and more often on laptops, a BIOS password may be set that prevents the HDD from spinning unless connected to the laptop shell. If the system is running, a decision should be made with respect to live imaging; in other words, if there is concern that shutting the system down will make it such that a physical level image is not possible or potentially useless (that is, can't be decrypted), then this information should be factored into the decision to allow the system to run at least until the images have been successfully made.

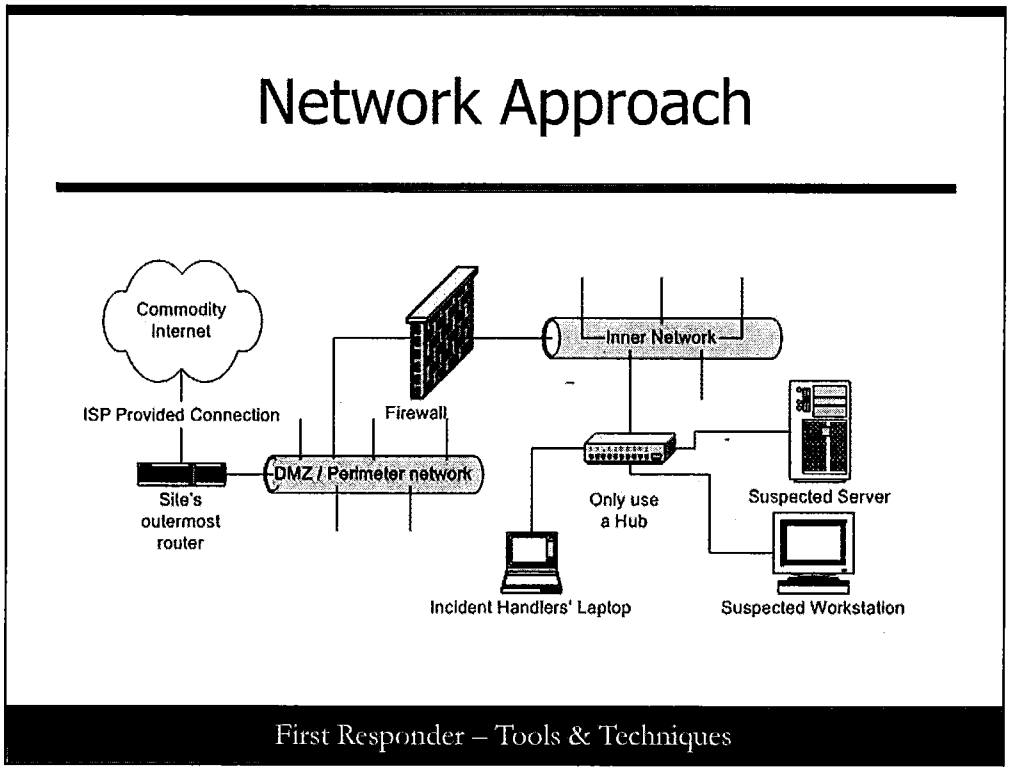


Live Data Collection

During the identification phase, we can, and often should, collect data that is going to and from the suspect system, as well as collecting data on the suspect system. When responding to a live system, we need to interact with the system, and we must do this in the most minimally invasive manner possible. This means that we should attempt to introduce as little as possible on the system.

The best method to approach this is to use your IR Laptop to receive the output from the data collection procedures. By writing the output from the commands to an external HDD attached to the IR Laptop, we are not modifying the system by attaching external media. This approach still requires that the IR CD with our known, good tools and static binaries be placed into the CD drive of the suspect system. Note that in the UNIX/Linux environment, you need to rely on one unknown binary (mount) to mount the IR CD. In addition, using this type of approach, we can potentially capture network traffic.

If the network is not available or you do not have an IR Laptop, you might need to consider utilizing attached storage. In this case, we would first want to proceed with using USB/Firewire external media if possible, but there are still some extremely old systems in production that do not even have USB 1.0. In this type of situation, you might have no choice but to write the output to the local disk. If you make this choice, be prepared to explain your actions.



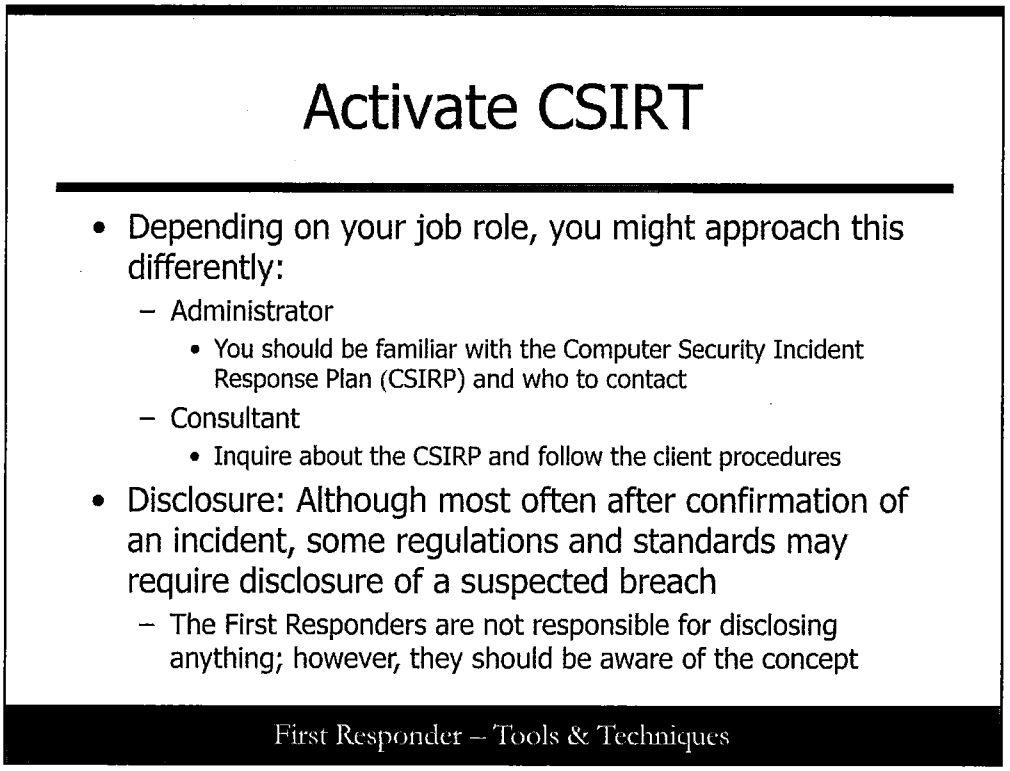
Network Approach

This is an illustration of collecting data live on the network. Here, you can see that there is a hub plugged into the network. Plugged into the hub are both the suspect system and the IR Laptop. By using a hub instead of a switch, the collection machine is electronically "sitting next to" the suspect computer. In real time, you can collect and also see who the victim is communicating with, and what type of data is sent back and forth. After handling many, many incidents on an open network environment, your course authors have the following advice on this subject:

1. At the beginning of the Identification phase, start a full content network trace at the network border (behind the corporate firewall, if possible) with TCPdump, ethereal, or similar tool. Consider using a filter that is scope-limited for the suspect system if you have identified a suspect IP address. Also consider using a network tap if available, as a hub/sniffer combination may not capture everything, depending on the volume of traffic. You may also want to consider introducing a network-based IDS sensor in a strategic location. Note that the organization should have a policy to support a full-content packet dump.
2. Unless you have confirmed there is malicious activity, observe traffic until you have made a determination. This may take some time, but often within 3hours you should have a good idea of whom the suspect is communicating with and what type of data is sent to and from the computer. During this window you can take other supporting actions, such as coordinating with management and a secondary system administrator to get a replacement online.
3. Prepare to capture data close to the system through the use of a span/mirror port on the switch or gather up a network hub and cables. If using a hub, quickly disconnect the system from the network infrastructure and connect it to the hub so that network connections are not lost.

If you are attempting to contain the incident or isolate the suspect system, you might consider connecting the network cable back to a hub or switch that is not connected to any other network or systems to allow the connections and processes running to continue to act as if there were no interruption.

4. When you decide to pull both the network plug and the power plug, you should consider shutting down any process that may corrupt data. For example, if you are running MySQL, use the shutdown script.

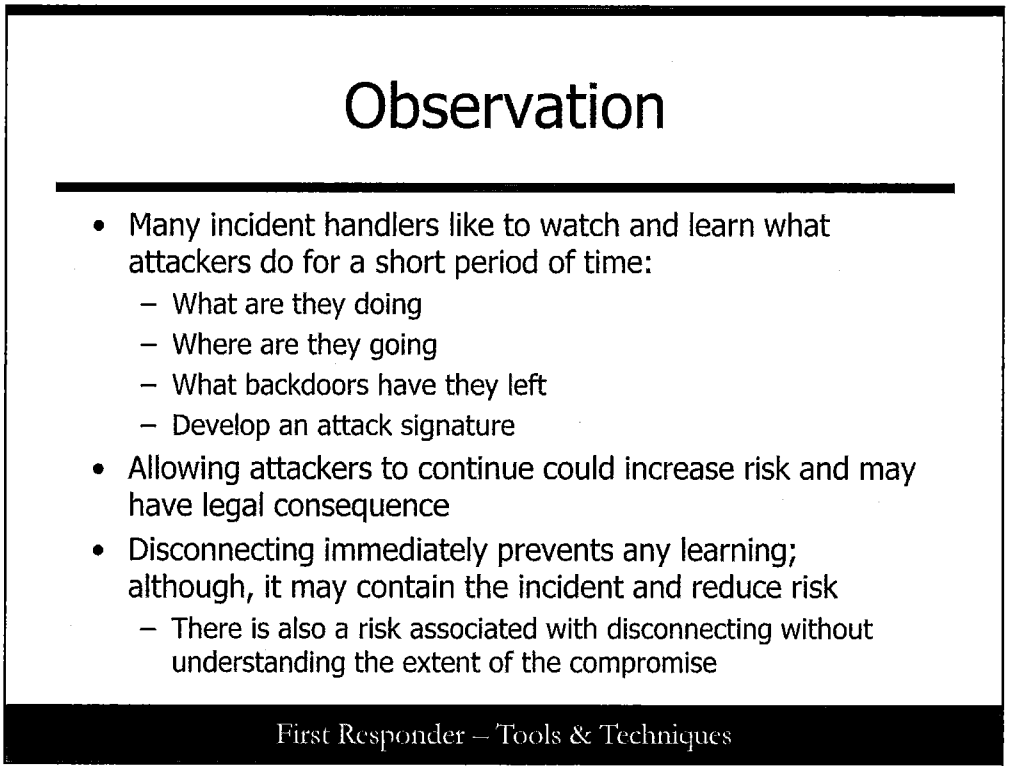


Activate CSIRT

Based on the initial triage, a decision should be made to determine if the event appears to be an incident and/or if it carries significant risk. For example, there is a big difference between finding a virus on a system and finding an encrypted tunnel that nobody can explain the function or purpose. This is the reason for gathering as much data as possible during the initial triage. If it appears that there is significant risk to the organization (that is, financial loss, data loss, and so on), then the First Responder should recommend the activation of the Computer Security Incident Response Team (CSIRT). The purpose of calling out the CSIRT is to ensure that information is shared properly and that all the required decision makers are involved in the next steps.

This is not a course in incident response planning, but as a First Responder, you are a key member of the CSIRT; therefore, you should be aware of the CSIRP documentation and know how to reach the necessary individuals. If you are a first responder for your own organization, you should be familiar with the CSIRP and know who to contact and when. If you are a consultant, you should certainly ask to see the CSIRP and follow any processes and procedures as applicable.

If you become aware of anything that you think might require attention from a disclosure perspective, you should mention this concern to the appropriate parties within the CSIRT.



Observation

During the Identification phase you will be faced with the *tactical* decision to either a) pull the plug on the suspect system or b) watch the suspect system and hopefully learn something about the attack and compromise so that you can better respond. There are real benefits and potential consequences to both decisions.

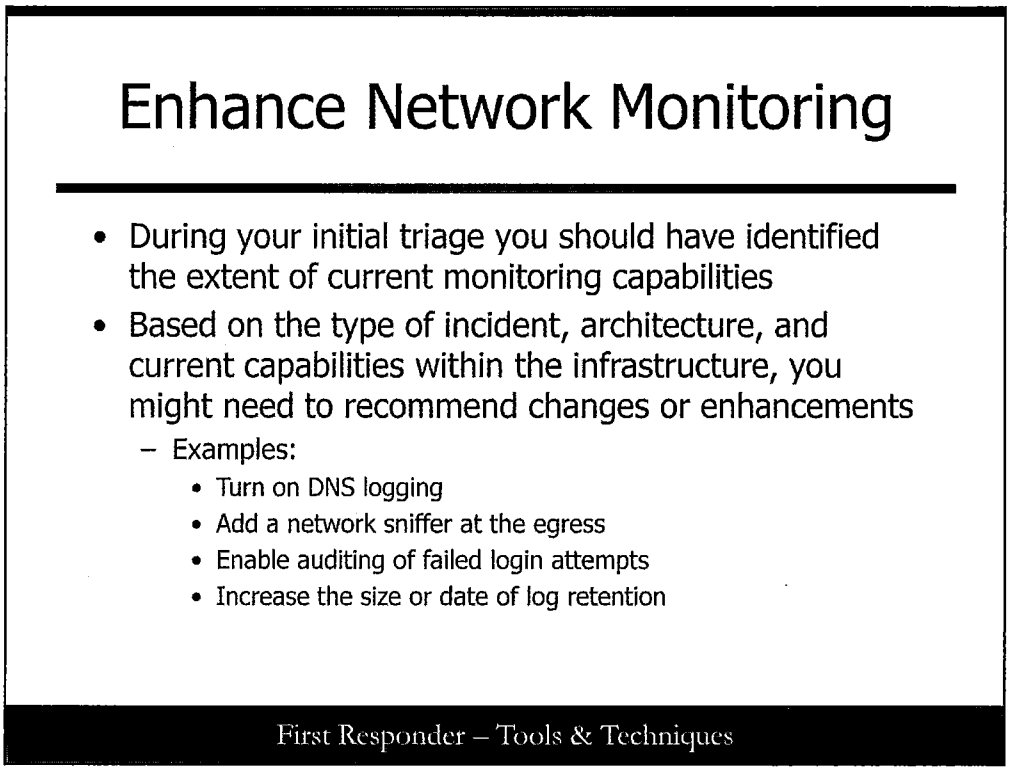
If the choice is to "pull the plug," you minimize your immediate risk and show due diligence by stopping the known actions and ability of an attacker upon discovery, which is a strong and defendable response to the attack; however, there is also a significant risk that you will never know the true extent of the compromise. In addition, if you pull the network plug, you run the risk of automated code potentially damaging the system. If you pull the power plug, the system may end up in a corrupt state and recovery of *critical data* may be difficult.

If the choice is to "watch and learn," you have the opportunity to glean valuable information about how the system is used, where the attacker is coming from, and more important, what ports and possible applications are used on the system; however, you also run the risk of being seen or perceived as allowing the attacker to continue to have access to the system, depending on how long you monitor.

The good news is that this is not your decision. This decision should involve the members of the CSRIT, and this should include a representative from the legal team. The primary decision is a risk-based decision and is centered around leaving the system online and connected to the network for additional observations. At this point, depending on the preceding direction, it may be that only passive monitoring will take place for a period of time. If that is the case, acquisition of RAM may be put on hold even though the system is still live.

In most cases, the decision will likely be to go ahead and acquire RAM and even a live disk image while continuing to monitor network traffic. Although a decision could be made to pull the system off the network immediately due to the nature of the data that is involved, this does not mean that continued or enhanced monitoring is not a good idea.

If the decision is made to disconnect from the network, additional considerations should be made with respect to leaving the system running. If it is determined that it is best to power off the system, consideration should be given to the pros and cons of graceful shutdown versus hard power off. Depending on the operating system and the nature of the processes that might be running (if known), there could be destruction of valuable corporate data and/or evidence. If the system is powered off, a forensic image should be acquired as soon as possible. Then, it can be determined if the system can be powered back on for interactive observation and acquisition of memory. This is often done to see if there is malware on the system that might spawn a process upon startup.



Enhance Network Monitoring

Hopefully, there will be sufficient monitoring in place prior to an incident, but it is possible that no monitoring has been turned on. Then again, it is possible that monitoring has been improperly configured or inadequately tuned. Finally, it is possible that significant efforts to monitor have been put in place but that the incident warrants additional coverage. Whatever the case may be, it is at this stage that the first responder may need to recommend enhancements or fine-tuning to the current monitoring capabilities. Specific examples include increasing log size, adding fields to be captured, or tactically deploying a sniffer.

Keep in mind that this is a parallel action that is taken with the acquisition of memory, volatile data, and file systems. The next steps for this action set include:

- Log collection
- Log examination
- Log correlation
- Reporting

Note: This topic is directly related to Step One (Preparation) of the Six-Step Methodology, so refer back to the previous section as needed.

Note: Hopefully the organization is prepared. Although not necessarily a first responder responsibility, having logs properly configured ahead of time can help prepare for incident handling,

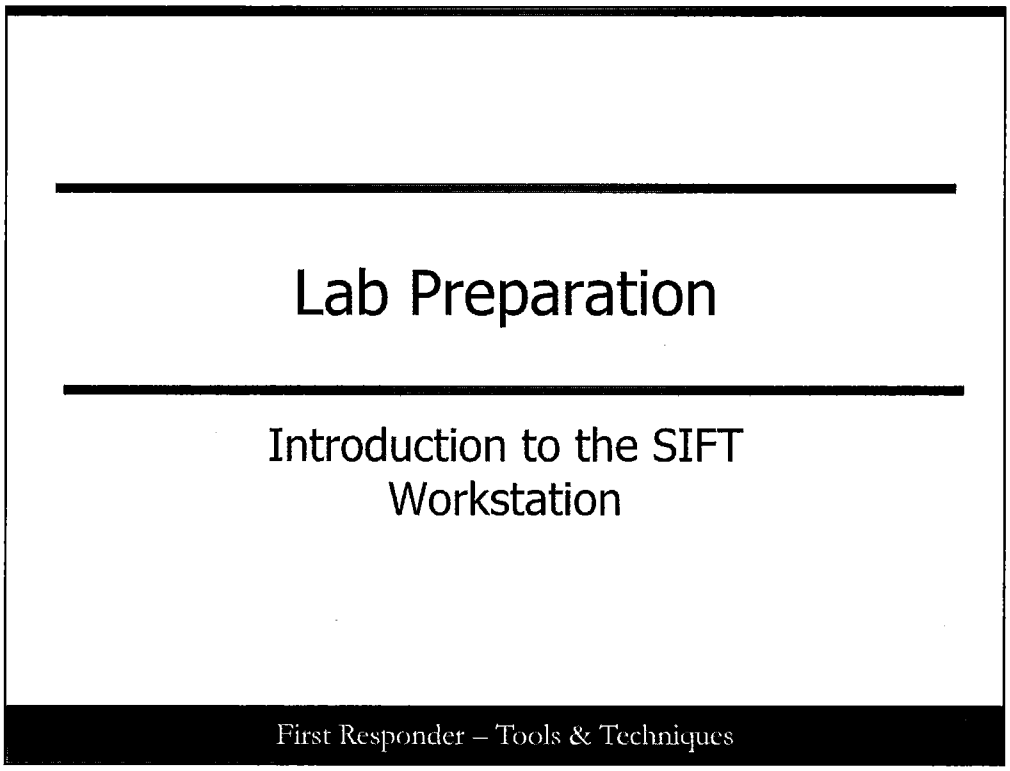
and it can help to form a baseline for the system. Analyzing previous log data can help you determine a pattern for the types of applications used on the server, as well as the types of tasks users routinely perform.

Also, if you expect to see data logged and don't, then you know you need to fix the system. In addition, if logging is not currently enabled or is improperly configured, it can be of value to consider enabling or tuning logging in certain circumstances, with approval from management/counsel.

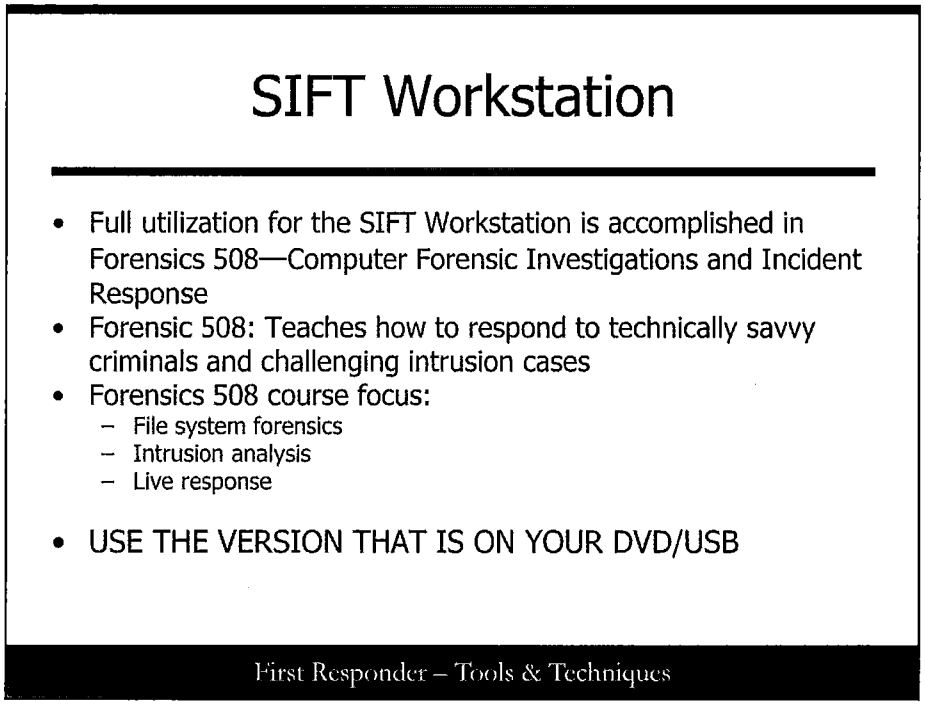
The main set of logs in the Windows environment is the event logs. It should be noted that logs can be written to a default location or can be configured to log to an alternative location, and that different services can log to different directories. We explore event logs later in this course. The main set of logs in a NIX environment are syslogs and are typically stored in /var/log/messages. We explore syslogs later in this course as well.

Also, to know what's different on a system that you might encounter, it is best to have some sort of baseline so that you can compare the current state of the system to the baseline. Perhaps the best baseline is the most-recent known, good full system backup. Another excellent baseline is the original system installation disk or build image. Finally, if baseline auditing tools are used, the output from these tools can be used for comparison. If you are not the system administrator, consider asking the administrator for basic information about the configuration of the system. Practically, this means understanding the operating system, patch level, what applications and services are installed, possible network access points through application ports, and the importance of the system in the enterprise. These days, most organizations have an IDS or IPS, either commercial or open source such as Snort, and there are several viable file integrity tools, such as Tripwire, that may also be an excellent source of valuable information that can assist you in the identification phase.

Often an attacker will attempt to cover her tracks by removing log lines from log data, deleting logs altogether, or disable logging. By having log data sent to an external source such as a centralized log server, log data can be protected. Even if the attacker removes data from the logs at the time the system was compromised, log data is safely stored on the external/centralized log server. This method allows you to view how the attacker compromised the system, while not being on the system. Also, when differences are detected, you have solid clues about skill level and attacker motivation.

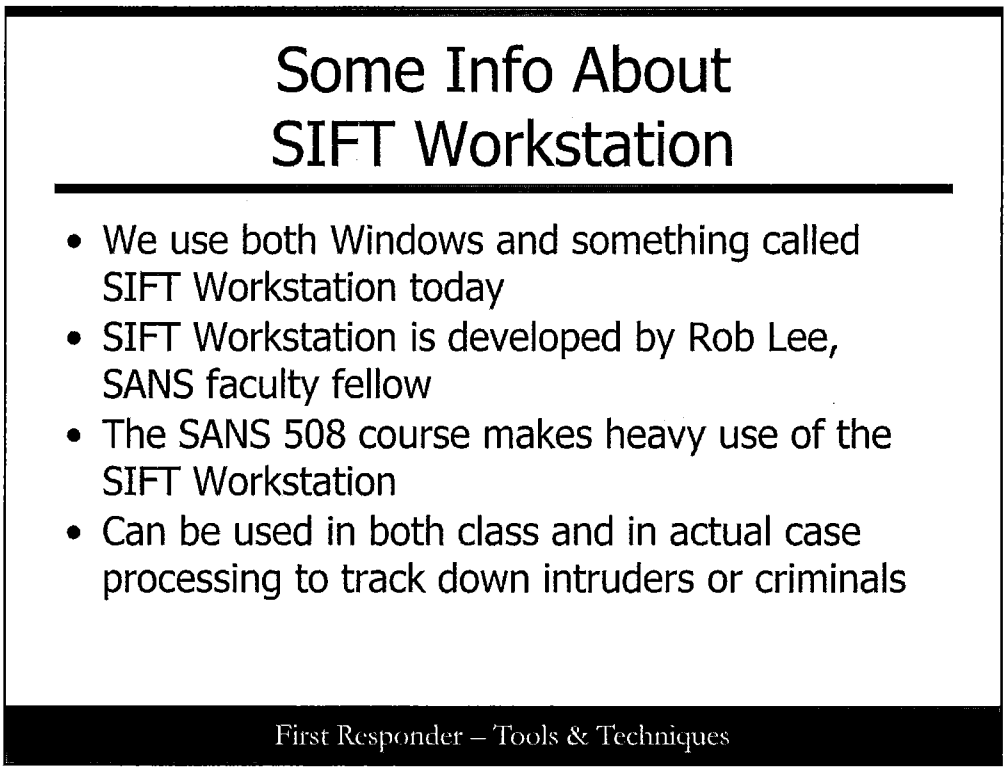


This page intentionally left blank.



SIFT Workstation

In the labs, we utilize the SIFT Workstation to introduce you to the SIFT workstation and to teach hands-on response techniques. The SIFT Workstation was created and is maintained by Rob Lee twitter.com/@robtlee. Full instruction on using the SIFT Workstation is accomplished in Forensics 508—Computer Forensic Investigations and Incident Response. Forensics 508 teaches how to respond to technically savvy criminals and challenging intrusion cases. The course focuses on File System Forensics, Intrusion Analysis, and Live Response. You can stay up to date with the latest SIFT workstation at http://digital-forensics.sans.org/community/downloads.



Some Info About SIFT Workstation

In the labs today we utilize both the SIFT Workstation VMware Virtual Machine environment as well as a Windows XP/Vista/7 instance running on a PC or VMware Virtual Machine that students may have brought along for the course. (Having access to both a SIFT Workstation and a MS Windows environment will go far in understanding the objectives for each lab and further the learning experience for the student.)

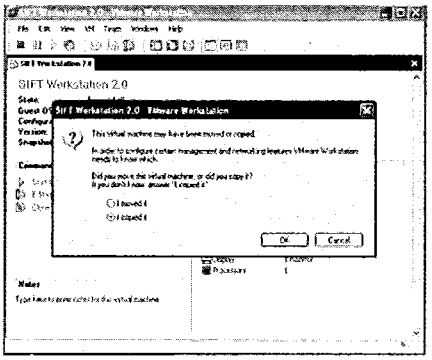
First off, credit for the development of the SIFT Workstation goes to Rob Lee (twitter/@roblee), a fellow faculty of SANS who teaches SANS Forensic 508—Computer Forensic Investigations and Incident Response. The 508 course instructs students of the tactics used by those adversaries on the Internet and high-tech criminals, provides techniques used to match them, and demonstrates the tools used to reveal the facts of the intrusions. The 508 course brings to light the tactics and techniques used to perform File System Forensics, Live Incident Response, and analysis into intrusions to computer systems. Lately, the tactics have been very useful for addressing and combating Advanced Persistent Threat (APT) cases where the intrusions into enterprise networks require an equally high level of detection and analysis capability.

The SIFT Workstation is a key part of effective detection of intrusions. Another is familiarity with Windows operating systems and where Windows keeps its relevant artifacts which are useful for forensic analysis.

Starting SIFT

Next, we spend a few slides preparing the SIFT Workstation so that your environment can help you begin work. If you are already logged in and have your display and settings to where you like them, you can skip ahead to Lab 1.

Moved or Copied?



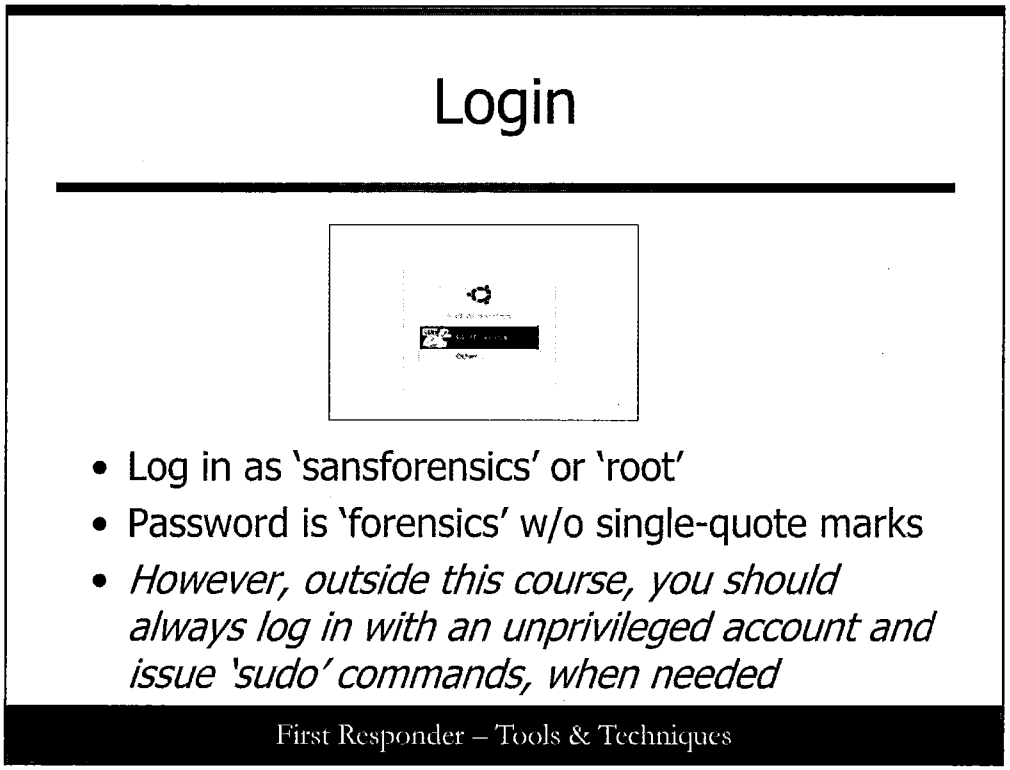
Upon starting the workstation you can expect the "Move or Copied" question—Select "I Copied It"

First Responder – Tools & Techniques

Moved or Copied?

Starting with setting up SIFT and readying it for work, apply it to your VMware (Player, Workstation, or whichever you brought and are permitted to use). Upon starting the workstation you can expect that it will ask you questions.

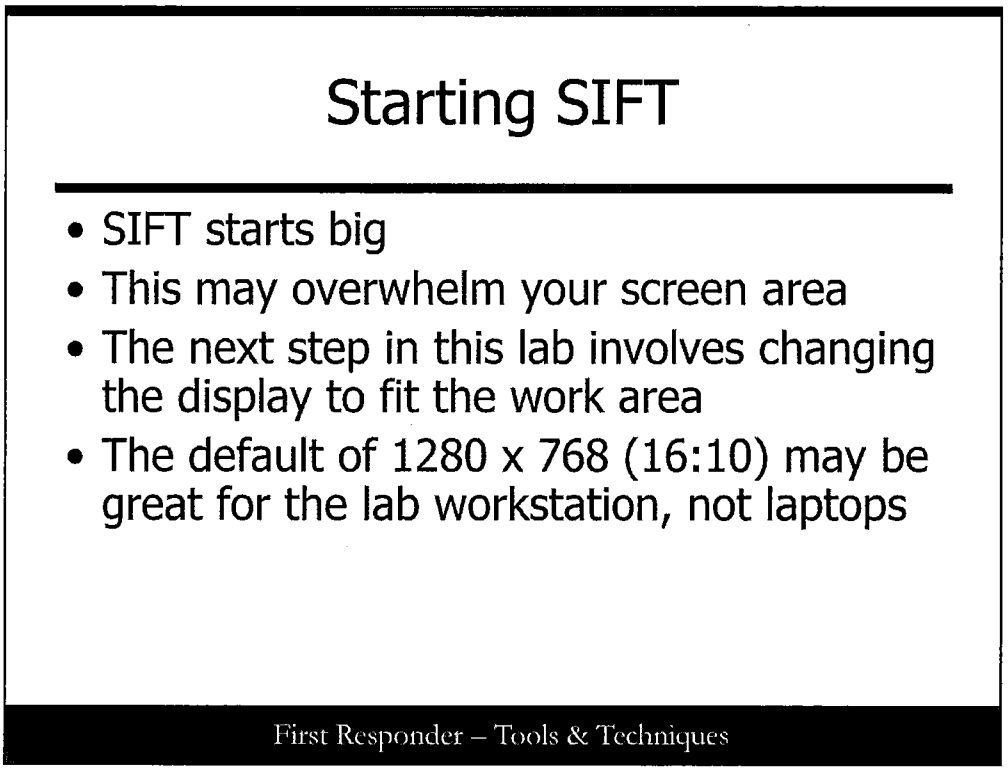
One of those may be if you either moved or copied the machine and its files. The recommended option is to say that "I copied it."



Login

Log in either as 'sansforensics' or 'root.' The password to both accounts is the same for the purposes of this course. Password is 'forensics' (without the single-quote marks.)

Note: Outside of this course, you should always log in with an unprivileged account and issue sudo commands, when needed. This keeps you from making a mistake that could risk critical data that might be irreplaceable.

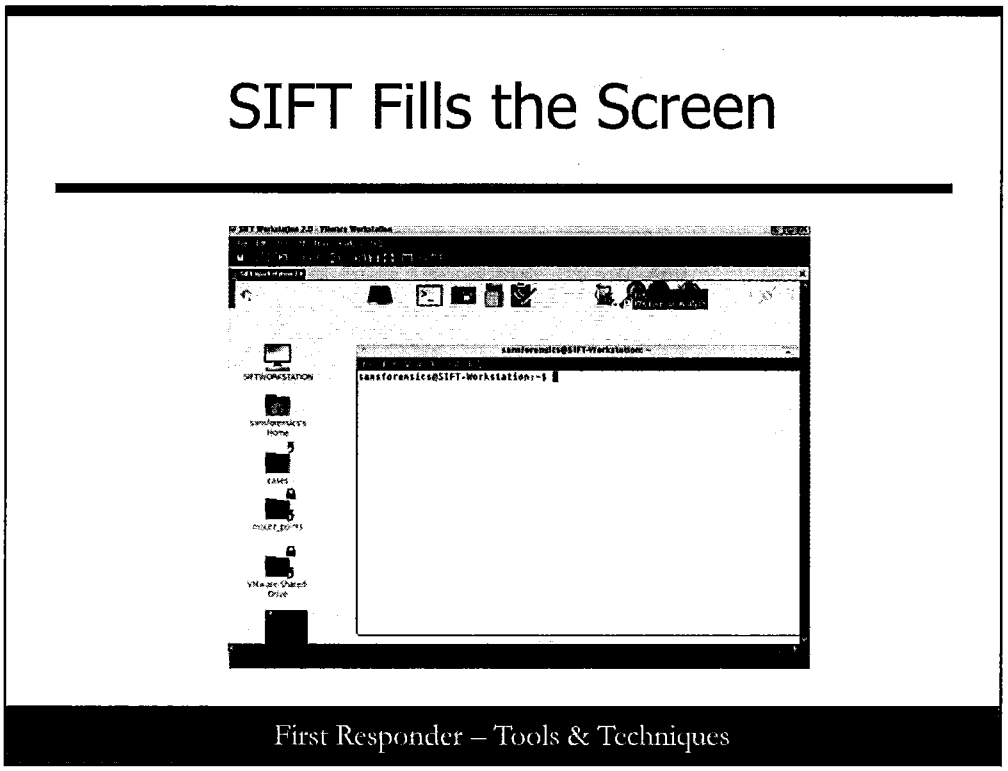


Starting SIFT

SIFT logs you in and you will notice that it made your display area large. This is the default setting fresh from the .ZIP file, and you have to make some changes so that you are not overwhelmed.

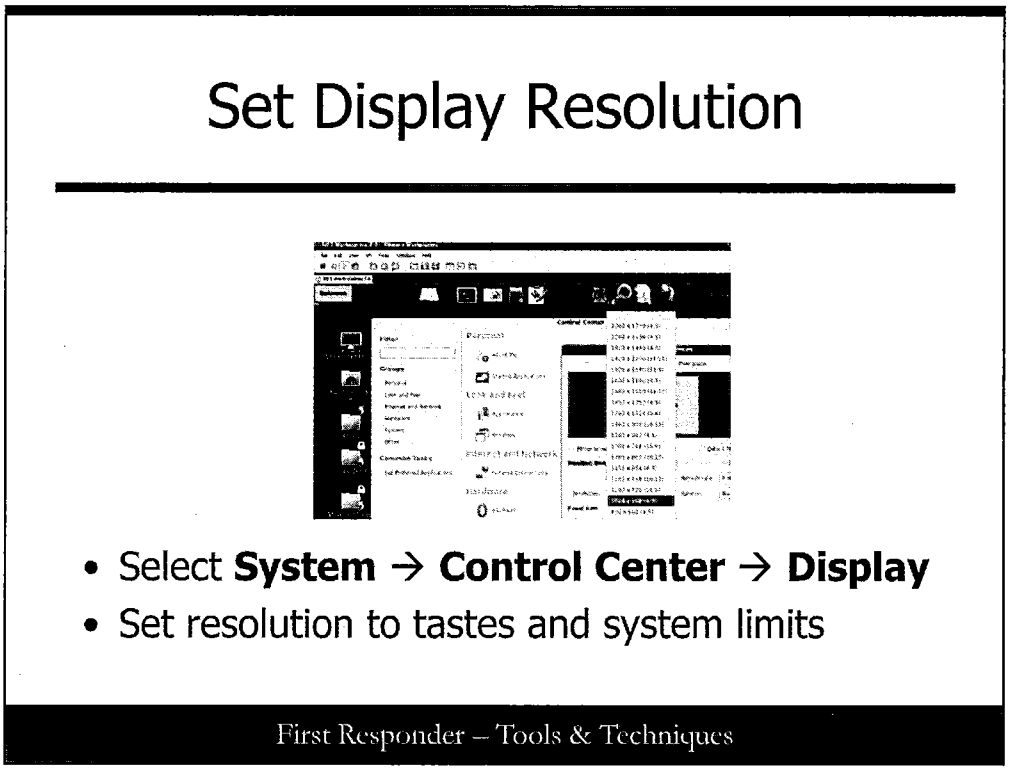
The next step in this lab involves changing the display to fit the work area.

Note: This means you might have to rescale SIFT to fit your screen display limits.



SIFT Fills the Screen

This section intentionally left blank.

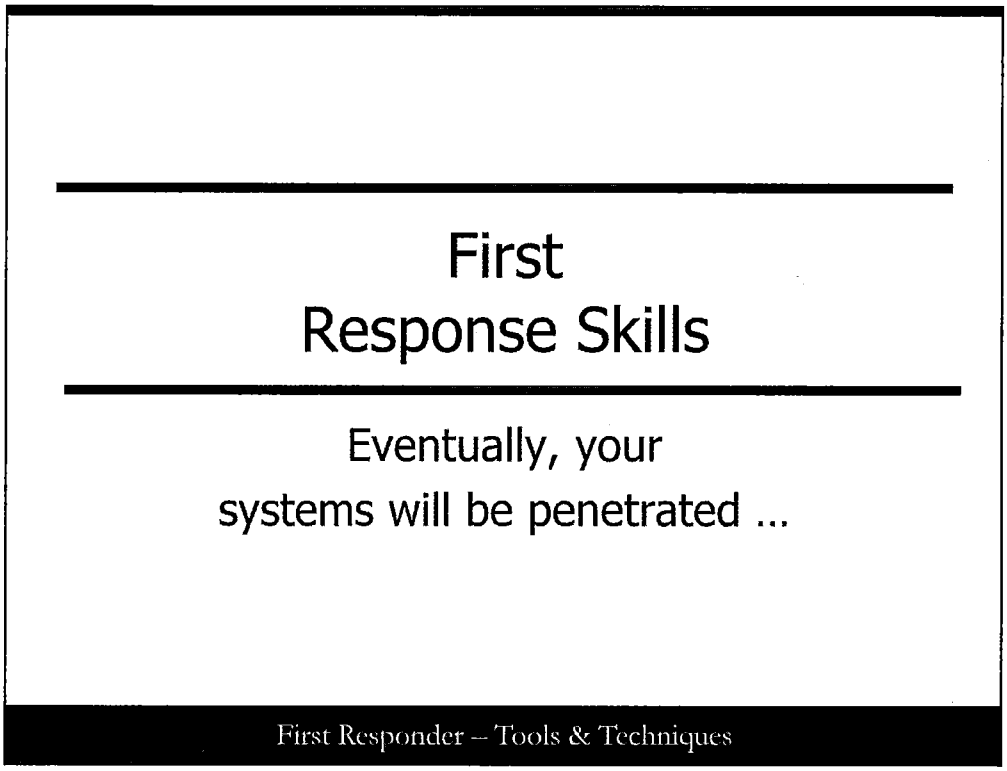


Set Display Resolution

Let's quickly change that resolution. Note in the presentation slide that the System menu option near the top of the illustration has a bright-red box around it to guide you to the next steps.

Select System → Control Center → Display.

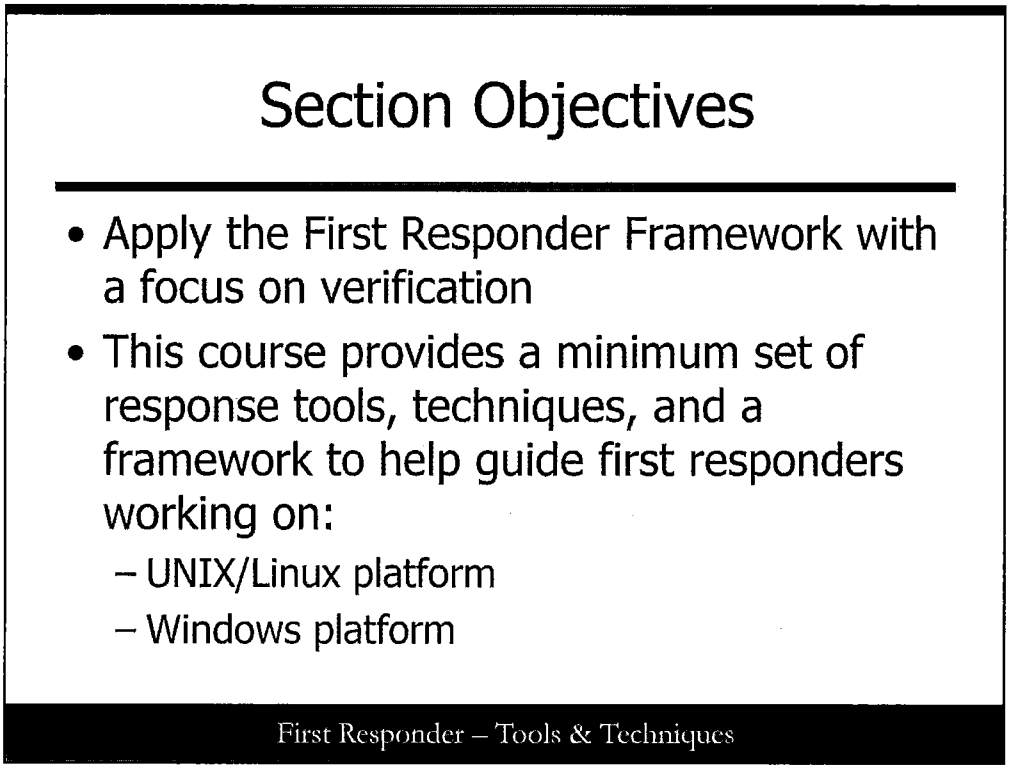
You will find the screen resolution is set to 1280 x 768 (16:10), which might be great for the lab workstation but maybe not laptops. Now you can adjust the screen resolution to something that fits your screen display limits.



SANS Security 501: First Responder

This is module two of the SANS First Responder course. This course is written for anyone who may be interested in this topic but is primarily geared toward security personnel who are members of a Computer Security Incident Response Team (CSIRT), as well as consultants, system administrators, and law enforcement personnel. This is a continuation of the introductory course in incident handling and the basics of digital forensics that is designed to help participants function as first responders—people who are the first on the scene of a computer security-related incident.

In this section of the course, we work through the incident handling process and focus attention on the tools that can be used. This course covers both UNIX/Linux and Windows by developing and applying practical skills on these platforms. This course has lab exercises based on first responder activities and shows several techniques for you to get a handle on what's "on the box."



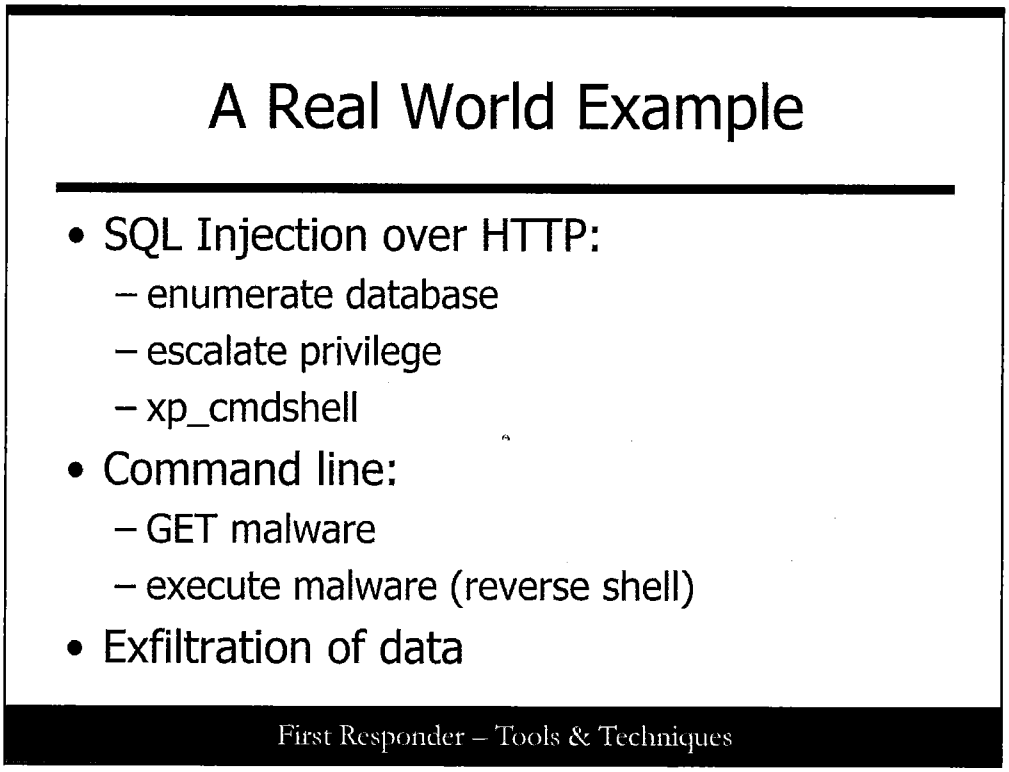
Session Objectives

The majority of the course content is designed to provide the essential skills and techniques you need as a first responder. These tools and techniques help you to recognize actual attacks by a variety of intrusions from automated code to highly skilled attackers. Throughout this section of the course, different aspects of the Six Steps will be applied to both the UNIX/Linux and Windows platforms.

Several tools are discussed, and the lab exercises take you through several of them. The individualized techniques shown can be refined into scripted incident response and tailored to the specific environment. During the course, we demonstrate how to perform data collection and basic analysis.

As previously mentioned, the majority of first responder activities are within the identification phase. As such, this course provides a framework to guide first responders regardless of the platform involved. This framework is consistent with the © SANS Six Step Incident Handling Methodology for effective incident response.

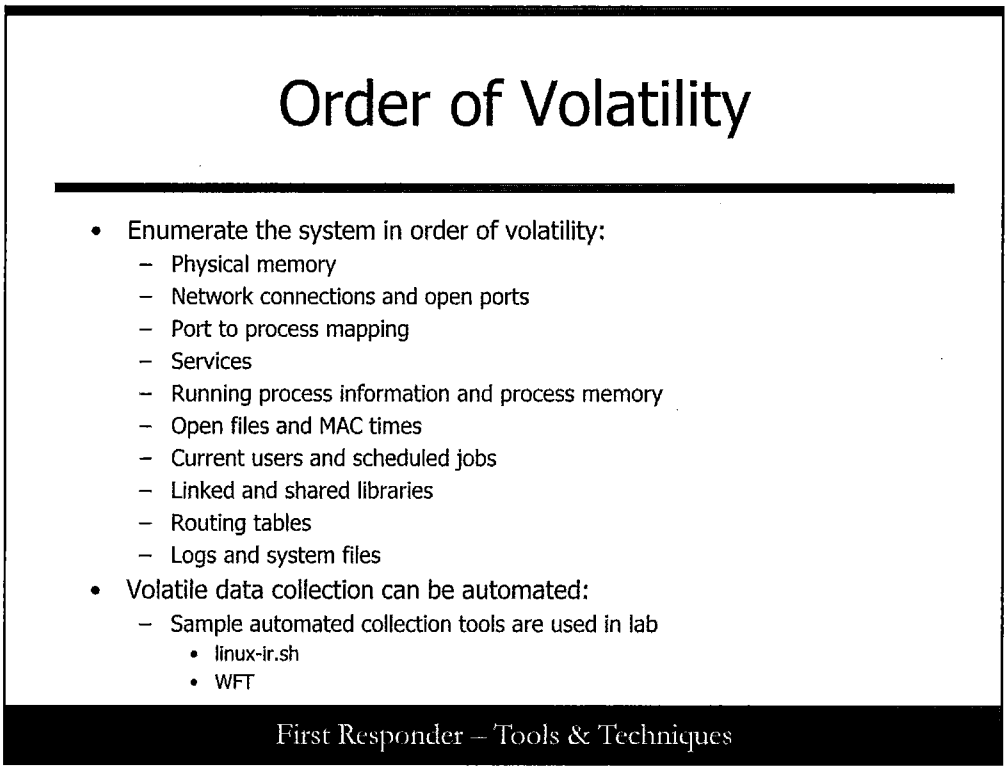
This course provides some practical guidance and case studies on how an incident response team and first responders can best work together to get a handle on these types of events. As the course progresses, there are some team response points that will come up. Also, you should make notes on how you would respond in your own environment. Each site and organization is different, and there is no "one size fits all" in this business.



A Real World Example

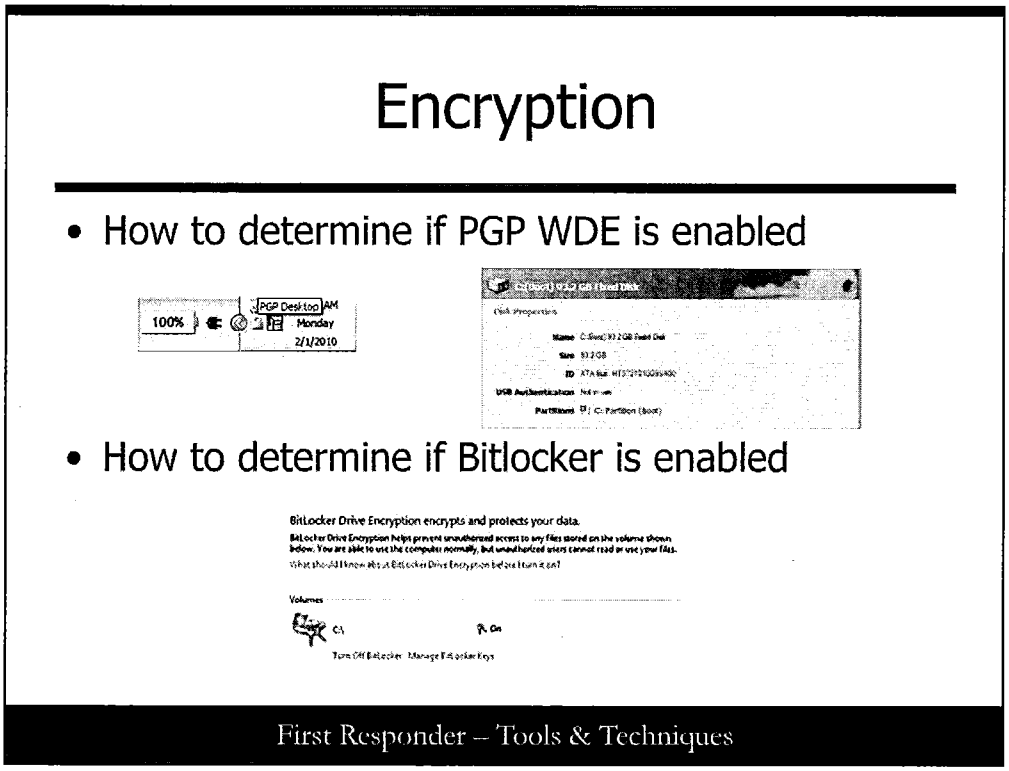
Depending on the type of attack, an attacker typically profiles an organization to determine if it has certain services running or attempts to identify other vulnerabilities. If an organization uses a SQL server backend database (usually on the internal network) tied to a web frontend such as IIS (usually in the DMZ), then direct external access to the SQL server is typically restricted by the firewall and potentially other controls. The problem is that for the web application to function using dynamic content, it must access the SQL server. If command-line access or administrative privilege is obtained, the attacker can use HTTP or FTP to upload malware from the server (typically from a compromised system housing the malware) to the SQL server that is inside of the protected internal network. After the attacker can execute this malware, he essentially owns the network depending on what the malware can do. Usually, this is a combination of backdoors, droppers, redirectors, keyloggers, Trojans, downloaders, installers, password crackers, encryption software, and other bad stuff.

In this case example, our attacker identified .asp pages on a website that were known to accept input in form fields and pass variables to a SQL server. The attacker created an FTP script and injected the script onto the system using SQL echo commands. Then using the extended stored procedure xp_cmdshell, the attacker passed an argument to a command shell that launched the FTP script. Because FTP was allowed out, the attacker could download malware onto the system. Then using xp_cmdshell, the attacker launched the malware. Furthermore, the attacker enumerated the internal network (using netview and add user accounts to the administrators group using netuser) from the compromised SQL server and spread the malware in an attempt to collect sensitive data and transfer from the various locations within the LAN to a different system that the attacker had also compromised. From there they simply access the compromised system (often through many other compromised systems) to obtain the sensitive data.



Order of Volatility

As discussed earlier, during incident response you should make an effort to attempt to collect data in the Order of Volatility. Registers and the CPU are a bit difficult, so we usually begin with memory collection. After that, network state, process state, the file system, and other media are collected. At this point in the incident handling process, you are about to interact with the system. Be sure to take copious notes of actions that you take and include some date/timestamps along the way. Also remember that data should be collected in a forensically sound manner, and remember to check, confirm, and record the location (building, room, office, and jack), condition, and other identifying information about the system (name, MAC address, and IP address).



Encryption

As previously mentioned, if a system uses encryption, we may need to modify our approach from post mortem imaging to live. You may notice a PGP Desktop icon (lock symbol) in the system tray. If you see this, it could mean that the system is using Whole Disk Encryption (WDE) and you can verify this by checking within that application. If you encounter a newer Microsoft OS, you can check to see whether BitLocker runs in the Control Panel or by issuing the following command from an administrator command prompt:

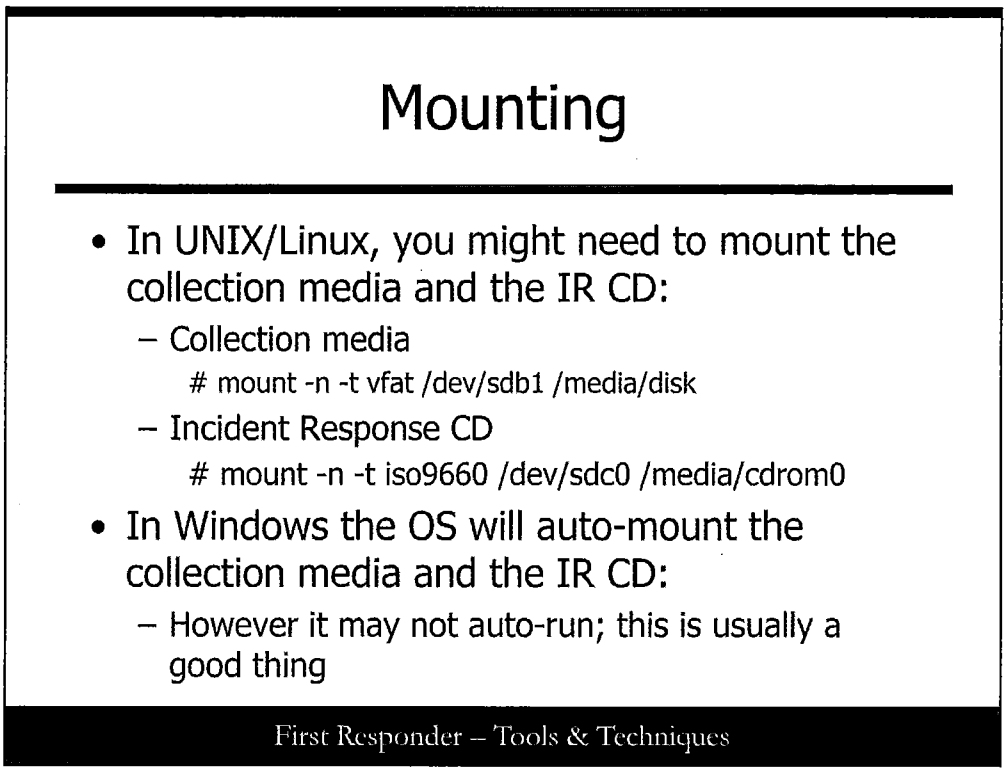
cscript manage-bde.wsf –status

You can also disable BitLocker with the following command:

cscript manage-bde.wsf -protectors -disable c:

Finally, with the following command, you can display the Recovery Key and Numerical Password in case they are later needed:

cscript manage-bde.wsf -protectors -get c:



Mounting

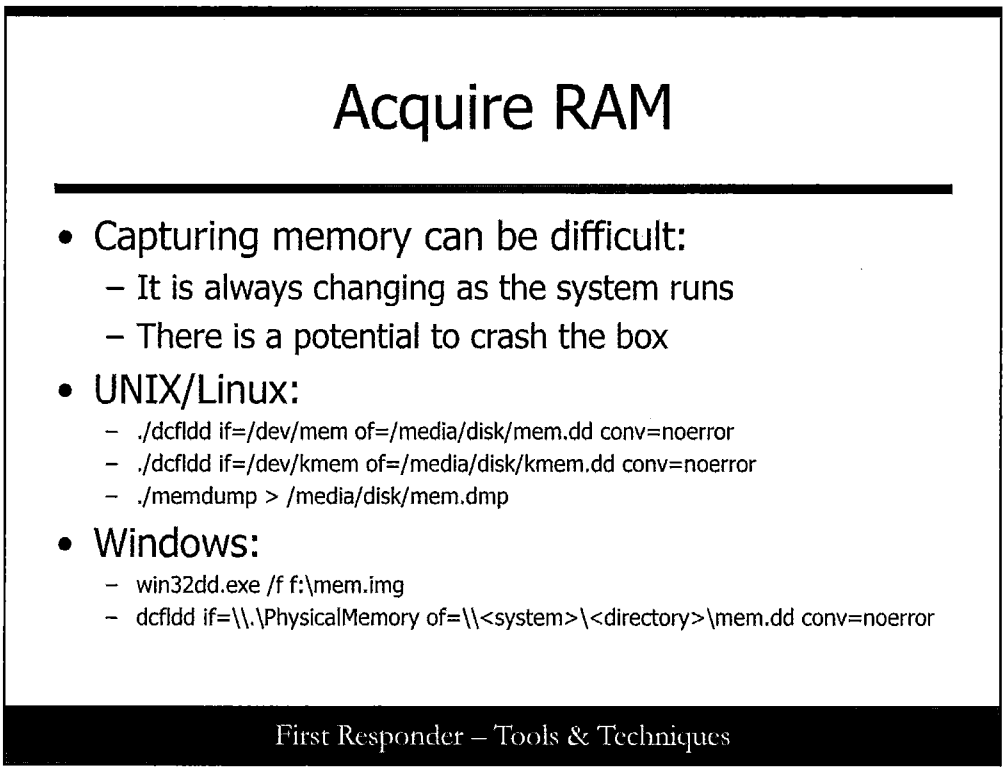
This slide applies to the UNIX/Linux environment. On some UNIX/Linux systems, the CD and USB devices auto-mount; however, if the system does not auto-mount, you need to mount the collection media and IR CD. Because the CD-R is read-only, this is an ideal choice for the response CD so that we know the tools will not be modified. If you have a CD-RW or DVD, you can always mount the device as read-only.

For the collection media, you need to mount the device as read-write. You should have brought with you some form of media to capture data while investigating the suspect system. Depending on the system and hardware attached, you might not have an option to use a USB drive; therefore, you might have to resort to using a floppy disk. It's not guaranteed that support for USB will be available and writing CDs under UNIX/Linux is typically not intuitive. In addition, it may write to the /tmp file system (or directory), which would change the system state.

You may decide early on to pull the plug; you might decide to collect quite a bit of data, or you might decide that the event was nothing at all. In any case, it's critical that you set up the environment to preserve the chain of commands that are executed on the system so that it can be shown what minimal changes were made to the system under investigation.

Use the script command supplied with most distributions of UNIX/Linux. This command records all activity with the terminal it was run from until the application is closed (exit). All data typed after the script command is run will be recorded into typescript by default or a file of your choosing depending on how it is run. All data displayed to the terminal after the command is run is also recorded.

Note: The command on the slide shows the -n option. This option prevents writing data to the file system table (usually /etc/mtab). By using this option, you can possibly prevent an attacker who's on the system from knowing you have mounted additional media.



Acquire RAM

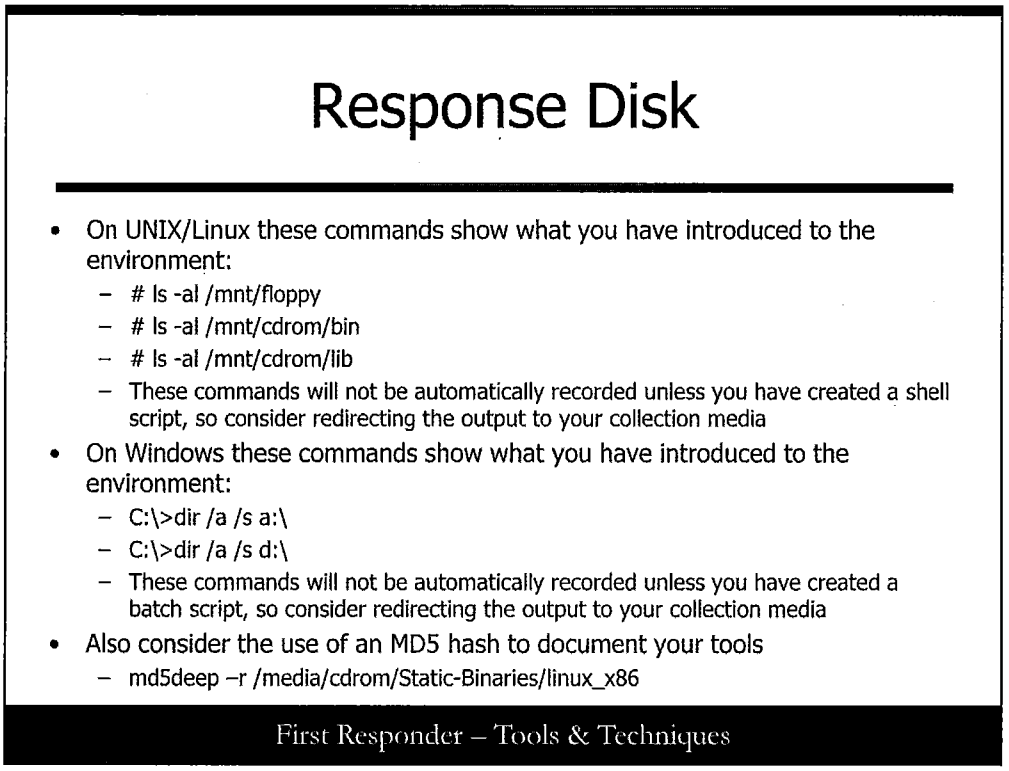
Regardless of the previous decisions made, at some point the first technical action that the first responder will likely perform will be Memory Acquisition. Capturing memory can be quite difficult and risky; any attempt to capture memory must be undertaken with extreme caution, as it can potentially crash the system. Acquisition of memory is extremely important particularly as new methods for analysis evolve. For example, just a few years ago, the method used to examine memory was to search for strings in memory and examine the strings. Today, there are tools such as Volatility and Memoryze that provide the responder with things like running processes and open files. This is critical as it can be the most accurate depiction of what was truly running on a system because advances in rootkit technology makes it more difficult to accurately collect volatile data. This does not mean that collecting volatile data using specific tools is useless; in fact, there may be a need to use these tools if memory cannot be acquired.

dcfldd or dc3dd

These tools are recommended for acquiring memory from a NIX based system. To take a snapshot of the memory, the command dcfldd can be used. This command does a byte-by-byte read and write of binary data. The command line options if= stands for in file and of= stands for out file. It should be noted that the file produced could be equal to the amount of memory on the system. If this system is a server with 2GB of memory, you should expect a 2GB output file. Using dcfldd, you can capture the physical memory on both Windows and Linux systems as well as kernel memory (/dev/kmem) on a Linux system. Also, you might want to use these techniques to copy the swap off of the system.

Win32dd

Although dcfldd can be used on a Windows system (provided you have the Windows binary), win32dd is highly recommended because it has the capability to acquire memory from even the latest Windows operating systems.



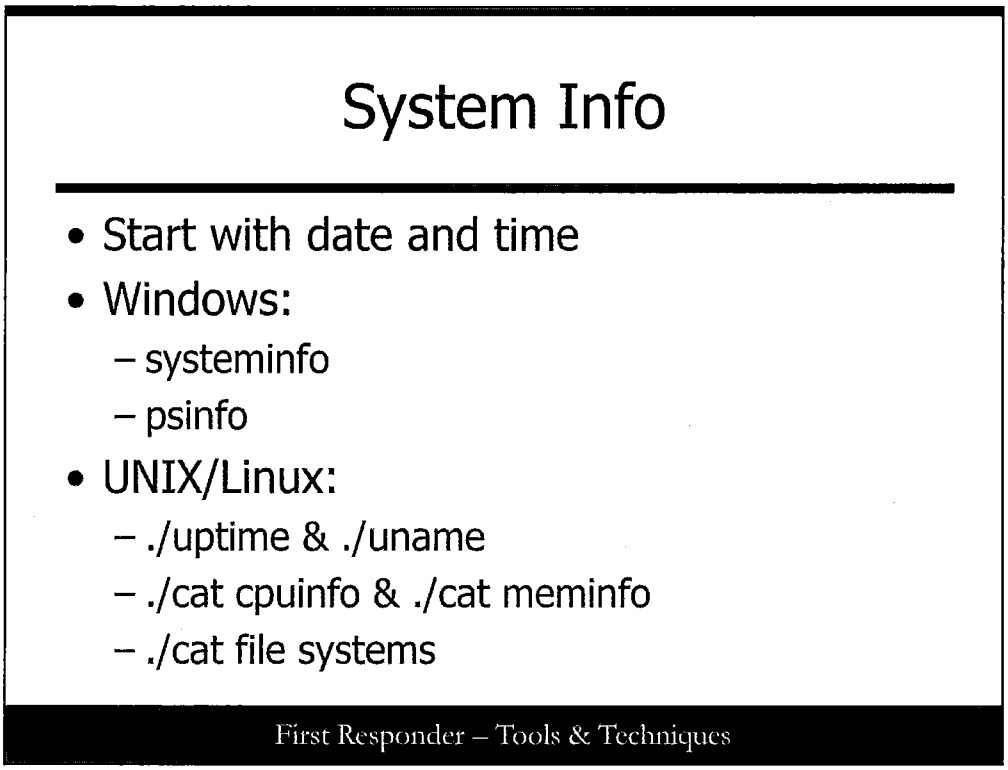
Response Disk

During the incident handling process, it's wise to establish the contents of the collection media. In Linux, begin by running the ls command and show what was on the collection media; it should be blank, except for your script files. Next, show the contents the bin and lib directories. Likewise, in Windows you can always list the contents of media with the dir command. One difference here is that you might want to pipe the output to your collection media as shown here:

```
C:\>dir /s /a dhsra d:\ > a:\IRCD.txt
```

This creates a text file on the collection media (in this case, you would probably use a USB flash drive) named IRCD.txt that should contain a recursive directory listing of all files on the disk. For ultimate documentation, you can utilize an MD5.

Using these techniques, if you are asked to account for the work you are doing, it may be helpful to show that you used a clean set of binaries in your investigation from the response media.

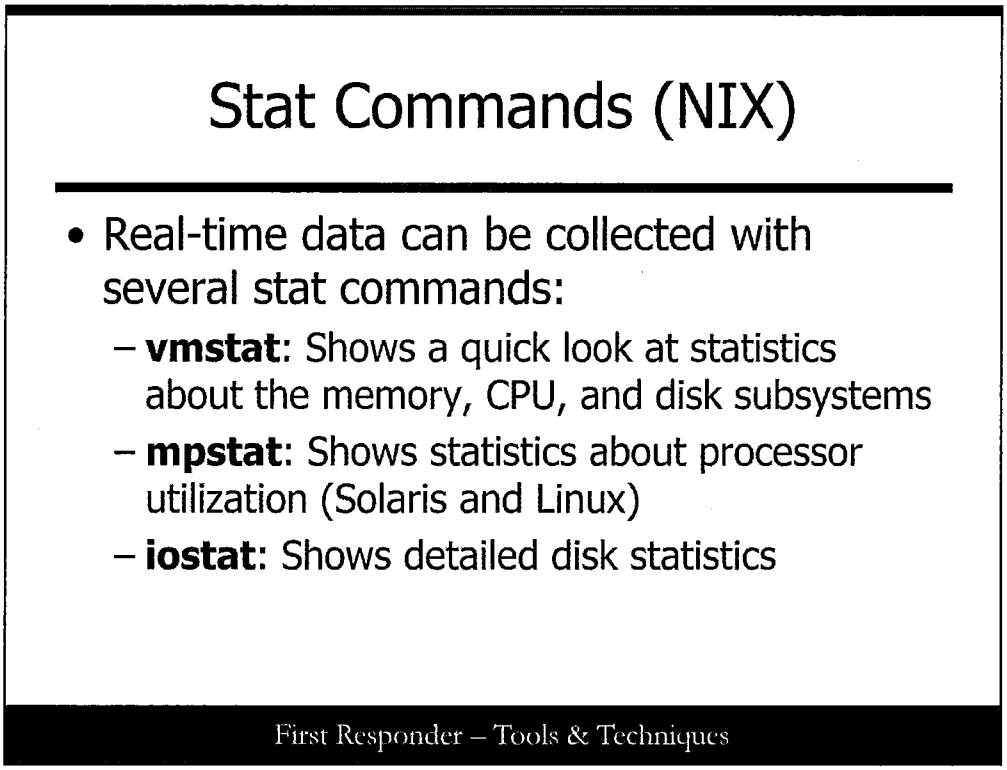


System Info

Generally, it is best to start by documenting the system date and time. In Windows this is done by using the date and time commands. In UNIX/Linux it is done by using the date command. This enables you to compare the current actual date and time with the system date and time.

In a Windows environment, you can use the systeminfo tool on a local system to obtain information such as registered owner, original installation date, system uptime, BIOS version, system directory, login server, hotfixes, and number of network cards installed. In addition, you can use the psinfo tool (PsTools) on a local or remote Windows host to display information such as the type of installation, kernel build, registered organization and owner, number of processors and their type, amount of physical memory, the install date of the system, and if it is a trial version, the expiration date.

In a UNIX/Linux environment, you can use the cat command combined with various other commands such as uptime, version, uname, cpuinfo, file systems, and meminfo to give you a nice display of useful information.



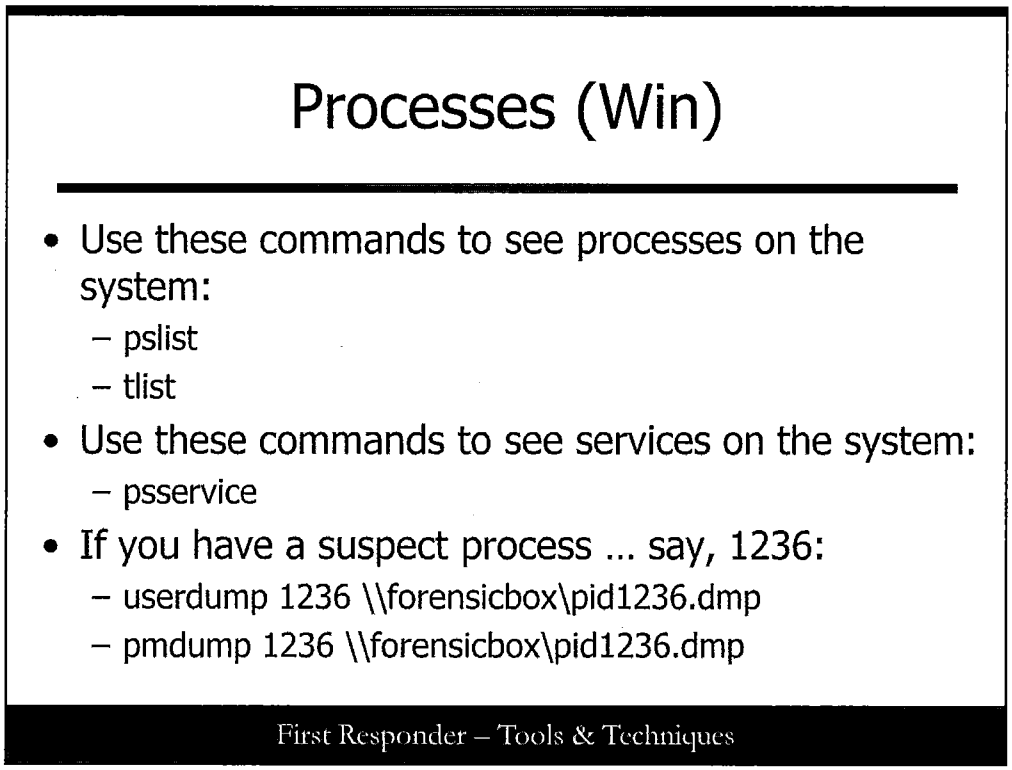
Stat Commands

There are stat commands that show a variety of status information about the system. Some may or may not be available on your particular system, and you might not find these commands to be useful in every incident nor will they be available on every version of Linux. The following are some sample commands:

- **vmstat**: Shows a quick look at statistics about the memory, CPU, and disk subsystems
- **mpstat**: Shows statistics about processor utilization (Solaris and Linux)
- **iostat**: More detailed disk statistics

View the status of the VirtualMemory at a 1-second interval with vmstat:

```
#vmstat 1
procs -----------memory---------- ---swap-- -----io---- --system-- ----cpu----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in    cs us sy id wa
0           0           0          23912       34428       1078720     0
            0          41          125         1193        401         8
            2          87          2
0           0           0          23936       34428 1078720           0
            0           0           0          1069        89          1  1
            98          0
0           0           0          23936       34428 1078720           0
            0           0           0          1091        140         1  0
            99          0
0           0           0          23936       34428 1078720           0
            0           0           0          1217        362         1  1
            98          0
```



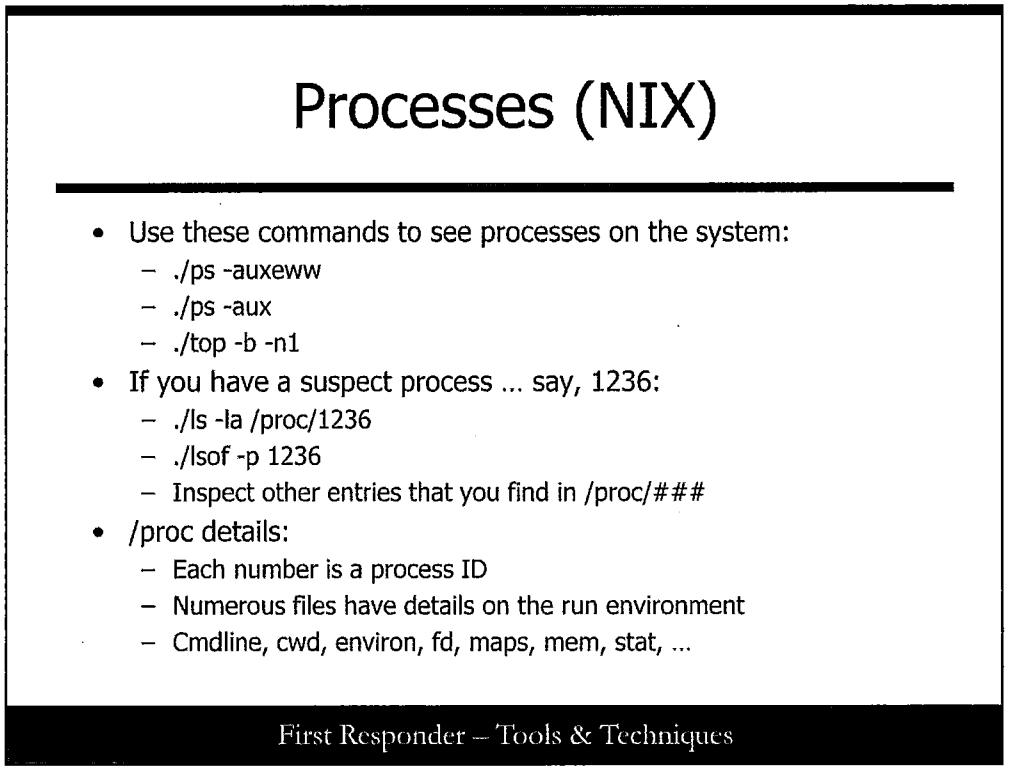
Processes (Win)

Processes are the heart of the system. When you press Ctrl-Alt-Del you can choose the Task Manager view and see processes running on the system. One great clue to malicious activity is which process is using the most of the CPU. Another great clue is to look for processes that "look" like real processes but don't have the same memory footprint.

Each process has a name, which is most often the name of the corresponding EXE that started it. The process ID is the number one way to correlate process information across tools; each process has a unique number on the system. Processes also have a Priority, which determines how often the system allows them to run: The higher the priority, the more runtime a process gets on the system per second. Processes can run or create threads; the more threads there are for a process, the more system resources it consumes. Processes consume memory, and it's important to be on the lookout for process names that don't match normal memory amounts (as mentioned).

There are three times for a process. First, there is the elapsed time: how long the process has been running since the system started. Then there are the Kernel and User times. User time is how much time the process has used in user mode. Kernel time is how much time the process has used in a higher privileged mode.

If you want to learn more about these times and characteristics of a normal system, you should investigate the Performance Monitor and various resource kit tools that are designed to help you profile a system. There is also a great deal of information at msdn.microsoft.com.



Processes (NIX)

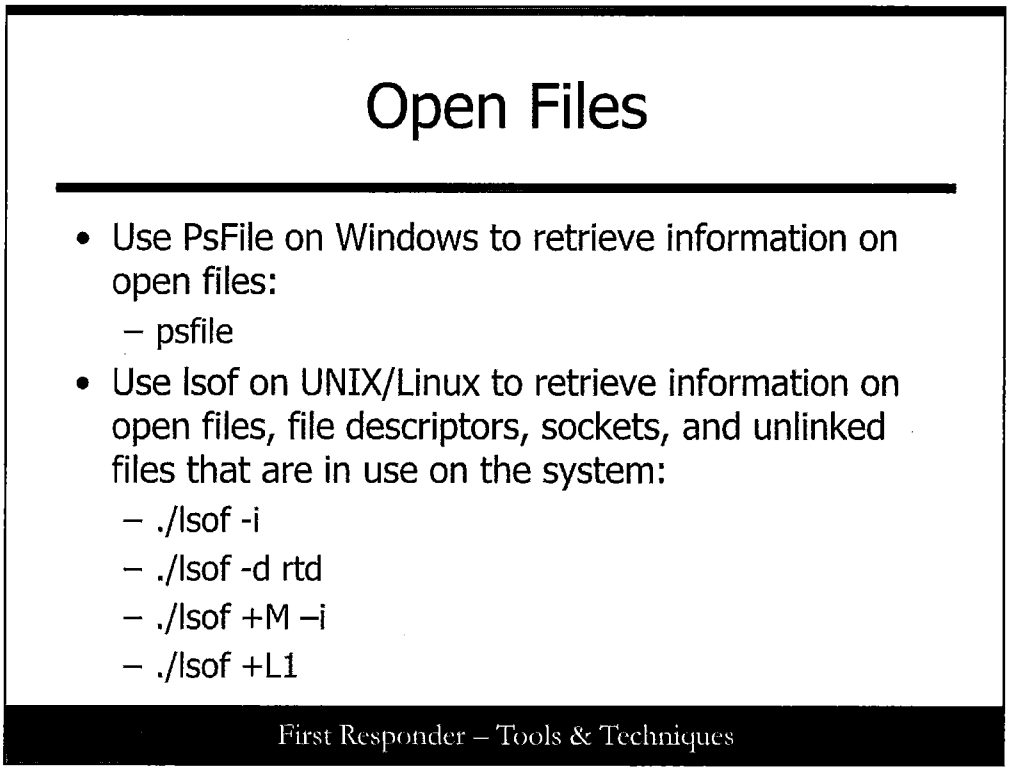
The ps -auxeww command shows processes that are listed on terminals (a), which utilizes user-oriented output (u); processes that are not associated with a terminal (x); and all processes (e); and it uses the wide (ww) format. The top command is normally an interactive command, which refreshes the display frequently and highlights higher CPU utilization processes. Top also shows per-processor utilization information. Here, the -b and the -n1 options tell top to display data in batch mode one time. You could run this with something such as -n30 and get a little more than a minute's worth of output, which would require that you read through a much longer output file.

Frequently during incident response, you identify specific processes that are suspect. To investigate a given process, the /proc file system (a map of processes in memory mapped as a file system) should be checked. Note that the more navigation and the more commands you run on the system, the more likely you are to tip off an attacker, and the more changes you might make on the system. There are two broad choices that can be made at this point: Preserve a copy of /proc for offline analysis, or use lsof to analyze the processes in turn.

The /proc file system is not a real file system; it's a map of process information that is accessible as if it were on the disk drive. This directory contains all information on processes, including the information provided by ps and much more. The /proc directory is broken up into a web of subdirectories and files that change as the state of the system changes. It is a good idea to archive this directory structure when investigating a system. Normal UNIX/Linux commands (ls, cd, pwd) can be used to navigate this file system. Numerous files have special meaning. For instance, /proc/devices contain the list of device drivers configured into the currently running kernel.

You can use /proc to get detailed information about a process. For instance, if you have a process that looks suspect, say 1236, you can look at the raw /proc information by looking at the files contained in the directory /proc/1236. /proc stores process information for each Process ID (PID) in a subdirectory named as the Process ID.

Another useful command to help collect data from processes that are running is lsof. This command lists the processes that are running and the file descriptors they have open. The lsof command is a tremendous ally in system management and also in incident response. This command doesn't come with many versions of UNIX and Linux, and it should be compiled for the specific platforms that are supported or likely to be encountered at a given site. The lsof command stands for "list open files." There are numerous options to lsof. We provide one example here that covers how to obtain details from a specific process, but our next slide covers some additional examples with lsof to obtain details about open files.



Open Files

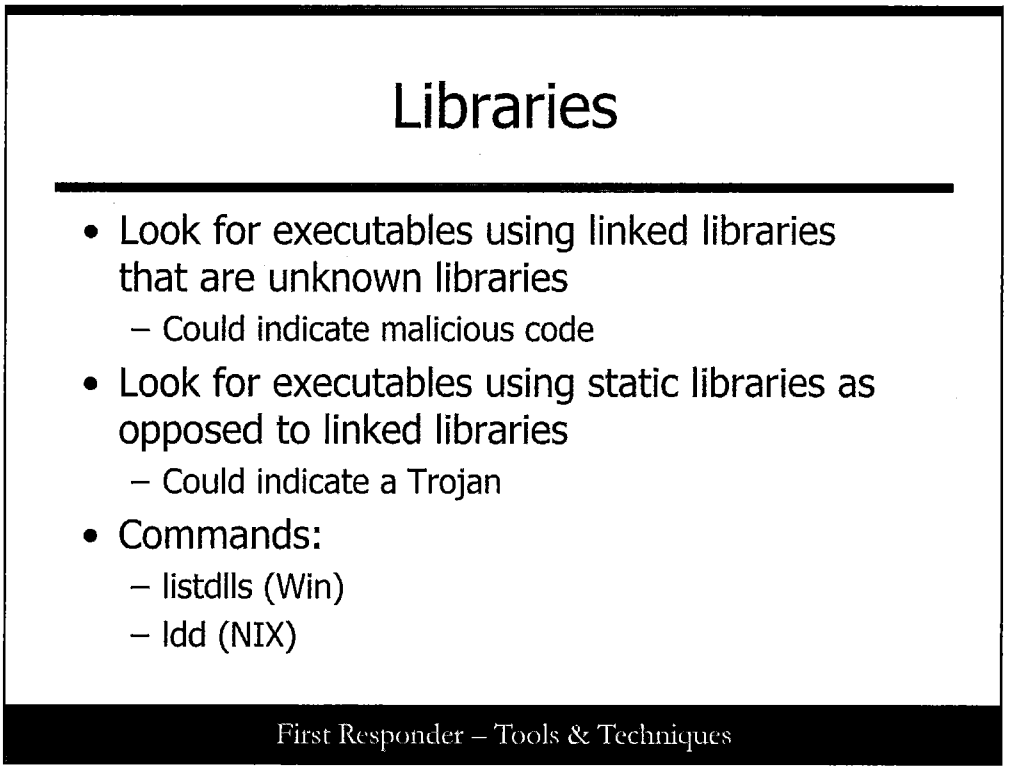
On Windows, running psfile shows you if any files are open remotely. If the situation warrants, there are specific ls and lsof commands that can be run in UNIX/Linux to get details on processes. For example:

ls -la /proc/1236	Listing of process ID 1236 contents
lsof -p 1236	Shows listing of files for this process ID 1236

There are a variety of ways to preserve the contents of /proc. One example, using the tar command, is shown here.

cd /proc; tar -zcvf /media/disk/2005.03.09-proc.tar.gz

Several types of files in the individual /proc/<pid> directories are useful, such as the cmdline file, which has the full command line used to run the process, or the subdirectory fd, which contains the file descriptor information.



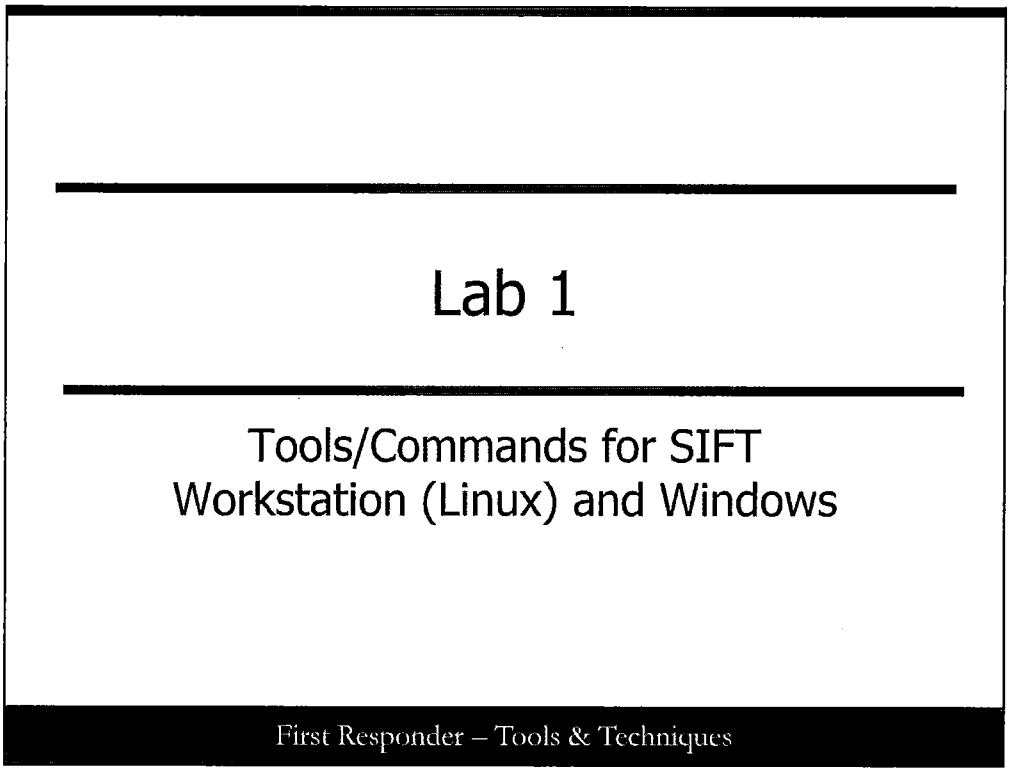
Libraries

Unknown or unusual executables should typically be examined during incident response. Because executables use libraries, the libraries need to be checked as well. Dynamic or Linked libraries are system libraries that are used by multiple executables. When an executable is calling an unknown or unusual library, it could mean that the library is an untrusted library containing malicious code. On the contrary, executables are sometimes statically compiled, so the libraries are packaged with the executable. If you have an executable with a known name running static libraries, this could indicate an attempt to pass the file off as a known good file, whereas the libraries are actually malicious.

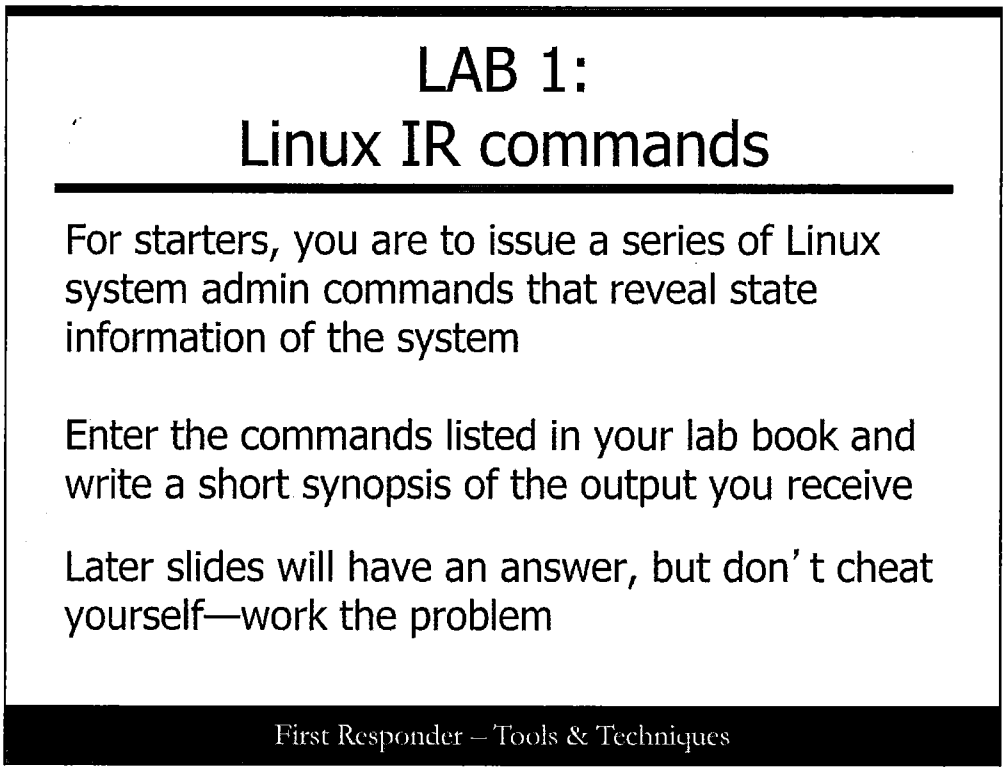
Failure to understand what libraries are called and how they might result in incorrectly classifying an event as not being an incident.

In Windows, listdlls shows all the Dynamically Linked Libraries on the system. If you know the process ID, you can use the –r option to show the DLLs for that specific process.

In Linux, ldd shows Dynamically Linked Libraries; however, you must point this command to a specific file.

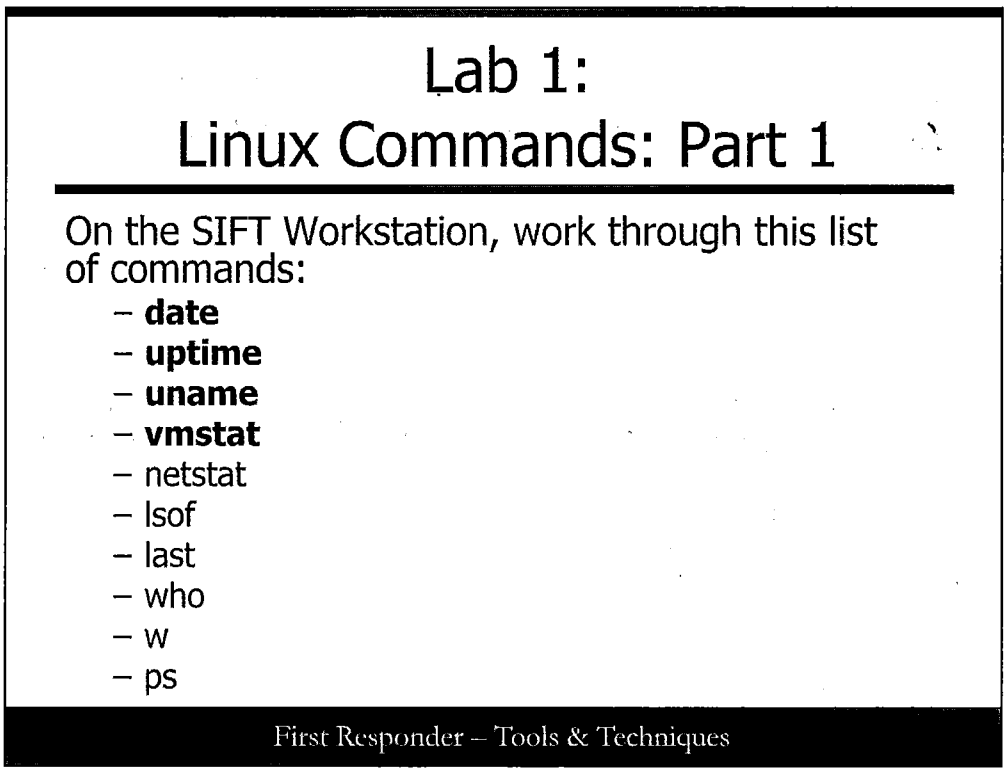


This page intentionally left blank.



LAB 1: Linux IR commands

For starters, you are to issue a series of Linux system admin commands that reveal state information of the system. Enter in the commands listed in your lab book and write a short synopsis of the output you receive. Later slides will have an answer, but don't cheat yourself; work the problem. This is for you to get maximum benefit from the exercise.



Lab 1: Linux Commands: Part 1

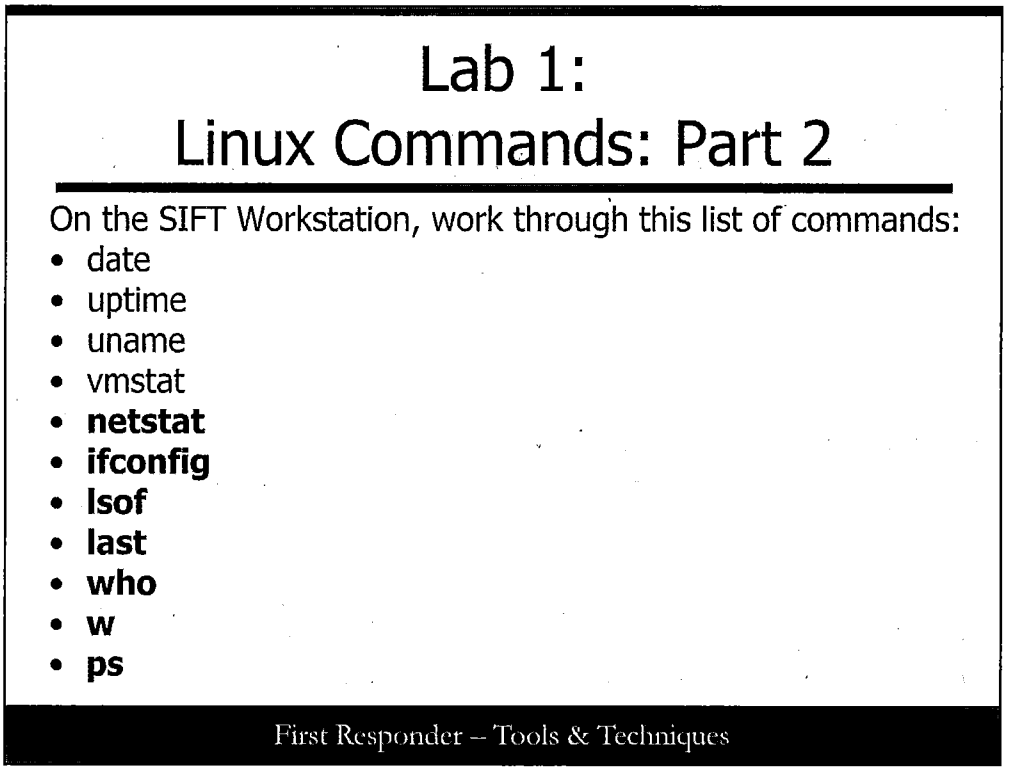
Issue the command env into a terminal window. (If a terminal is not already displayed in the main window, you can open up another instance by clicking on Applications → Accessories → Terminal). Record a brief description of the output in the lines provided here.

Issue the command date into a terminal window. Record the output in the lines provided here.

Issue the command uptime into a terminal window. Record the output in the lines provided here.

Issue the command uname –a into a terminal window. Record a brief description of the output in the lines provided here.

Issue the command vmstat into a terminal window. Record a brief description of the output in the lines provided here.



Lab 1: Linux Commands: Part 2

Issue the command netstat –an4. Record a brief description of the output in the lines provided here.

Issue the command netstat –an6. Record a brief description of the output in the lines provided here.

To the best of your ability and without looking ahead, explain why the slight difference in the two commands in the space provided here.

Issue the command lsof –ni4 into a terminal window. (You have to su to root to issue this command.) Record the overall results, which you see in the output, and how it is either similar or contrasts with the results of the netstat command.

Issue the command who into a terminal window. Record a brief remark of what you see in the output into the lines provided here.

Issue the command w into a terminal window. Record your observations into the lines provided here.

Issue the command ps aux into a terminal window. Record a brief summary of observations into the lines provided here.

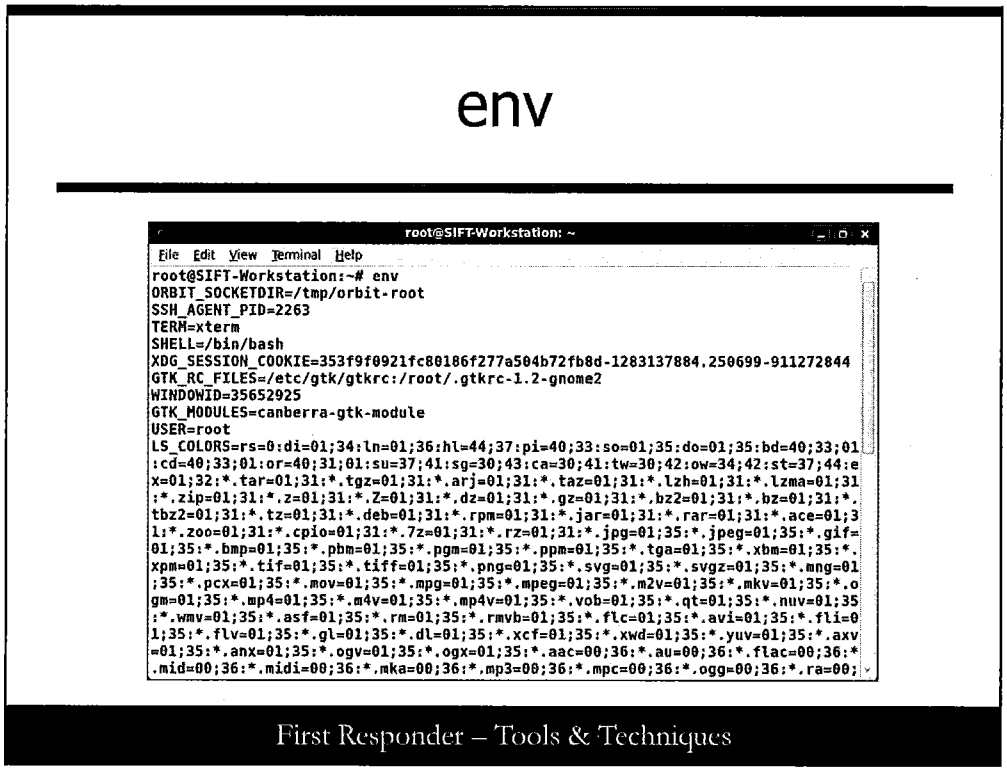
Now change directories in your command prompt by invoking the following commands. (Be sure to do this as the root).

cd /var/log

last –adi –f ./wtmp

last –adi –f ./btmp

Describe what you think you got out of that into the space here.

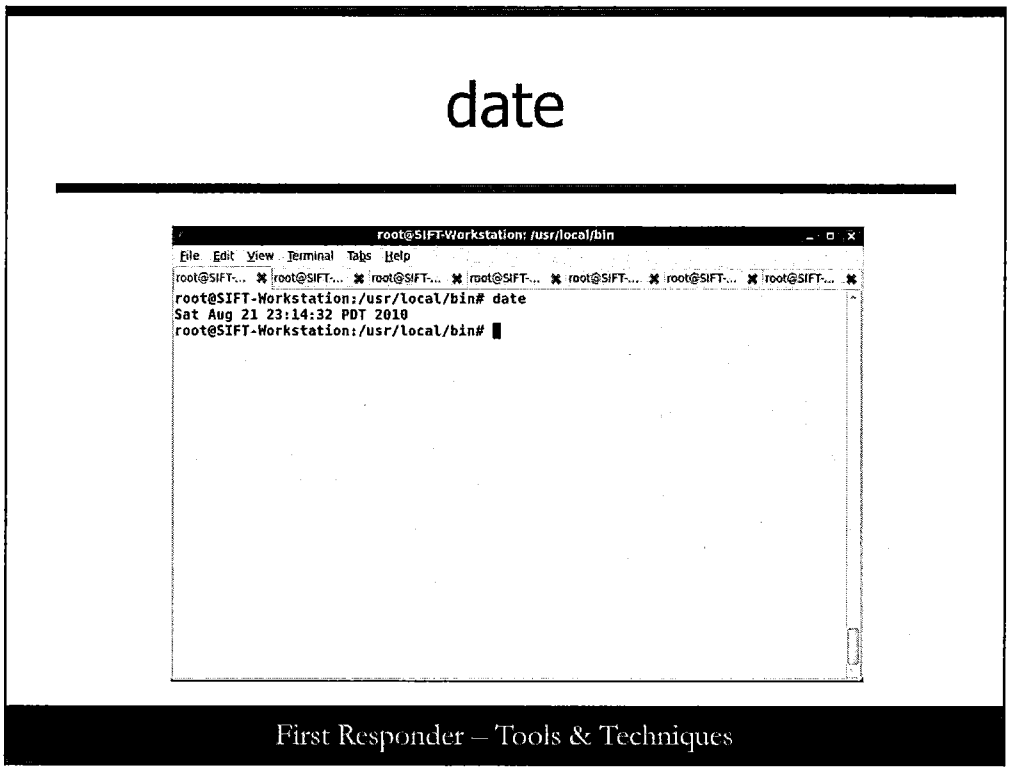


env

The command *env* gets you the currently set variables, which greatly help define the environment in which you work. This is a key collection point.

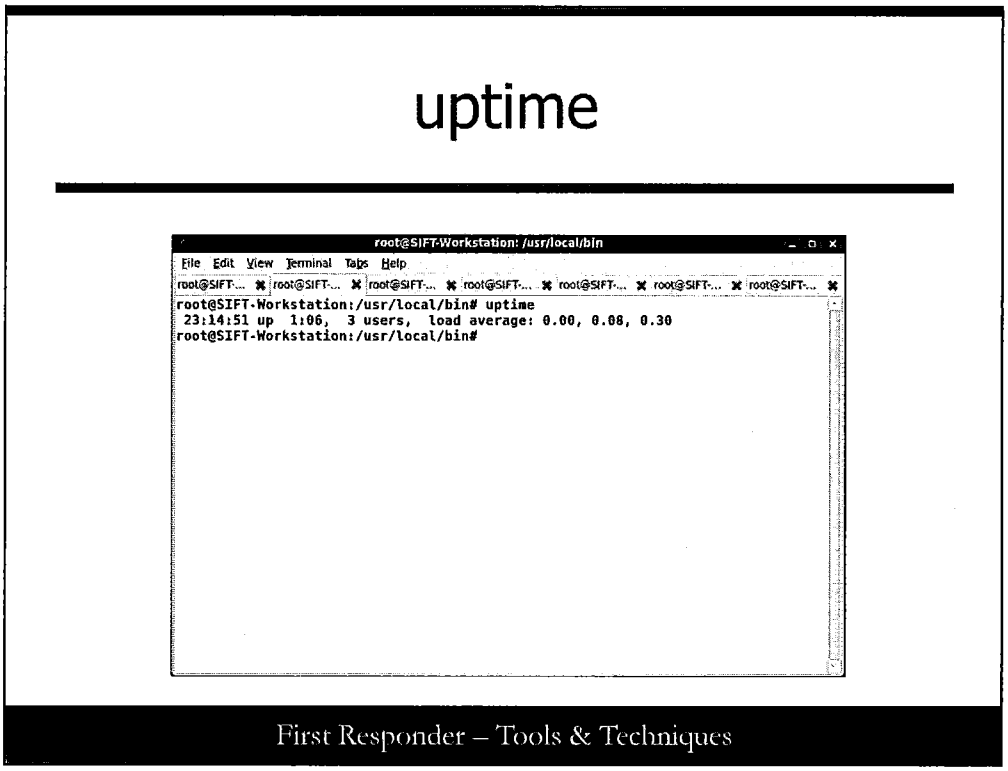
You should get something similar to the output, which follows in the screen shot.

Note: We scrolled up in the terminal window to capture the command issued at the top of the report. So scroll up and compare, but you might experience slight differences depending on what you set your environment to.



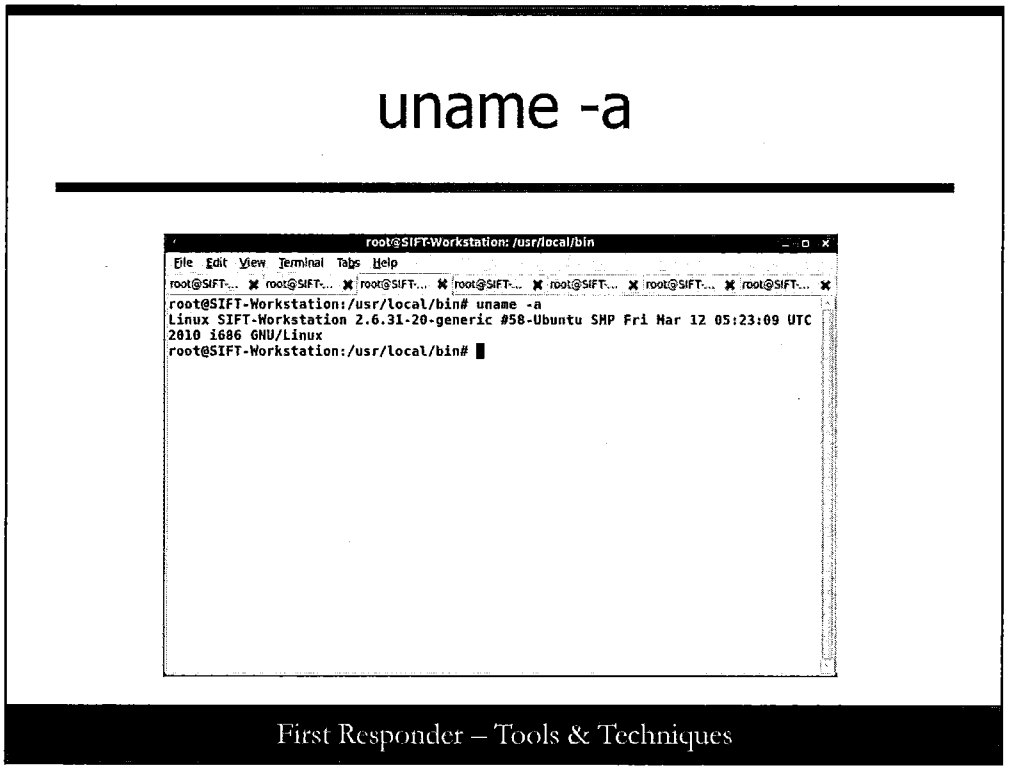
date

The *date* command is self-evident of what it is supposed to do (and it has nothing to do with candy or flowers, movies, or dinner—sorry). You should expect a similar output as shown in the following graphic.



uptime

The *uptime* command is basic because it tells you how long the system has been up and the load conditions (for example, Tasks) it has been burdened with. Notice you also can observe the number of users who are logged on.



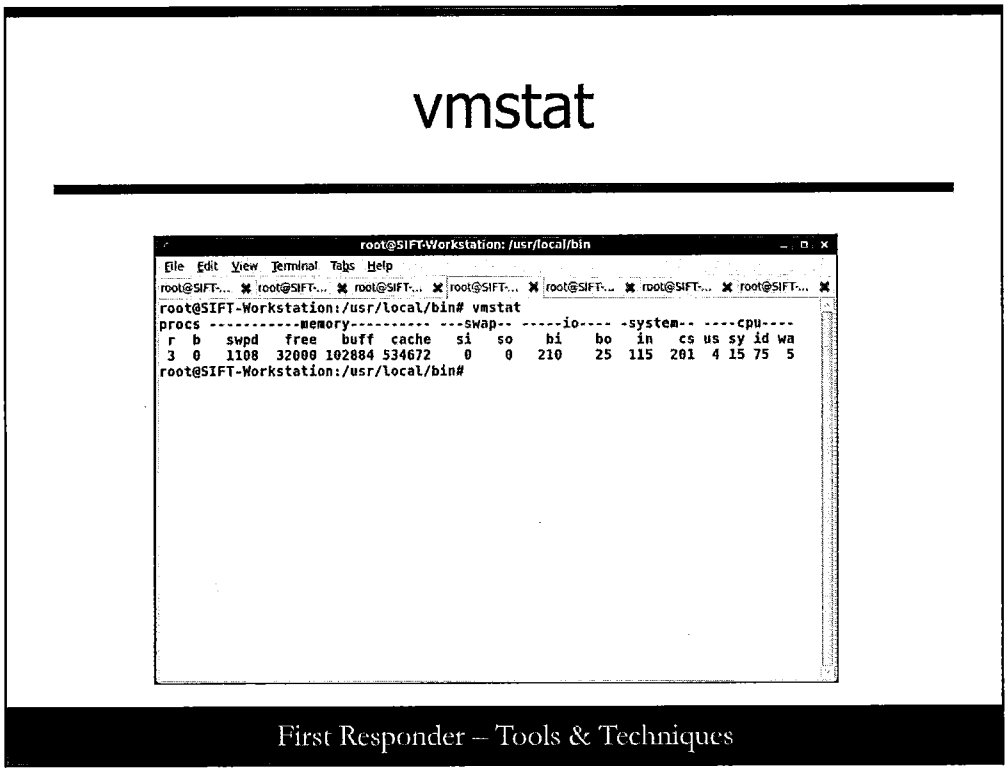
uname –a

Some great information comes out of this command because we issued this command with a "-a" on the tail of the command to modify its behavior. What we did as a result was tell the kernel to inform us *everything* it knew about this system. What follows are the statistics, in order, (and their associated individual switches).

Kernel name (-s)
Node name [what the machine is referred to or hostname] (-n)
Kernel-release (-r)
Kernel-version (-v)
Machine name (-m)
Processor type (-p)
Hardware platform (-i)
Operating system (-o)

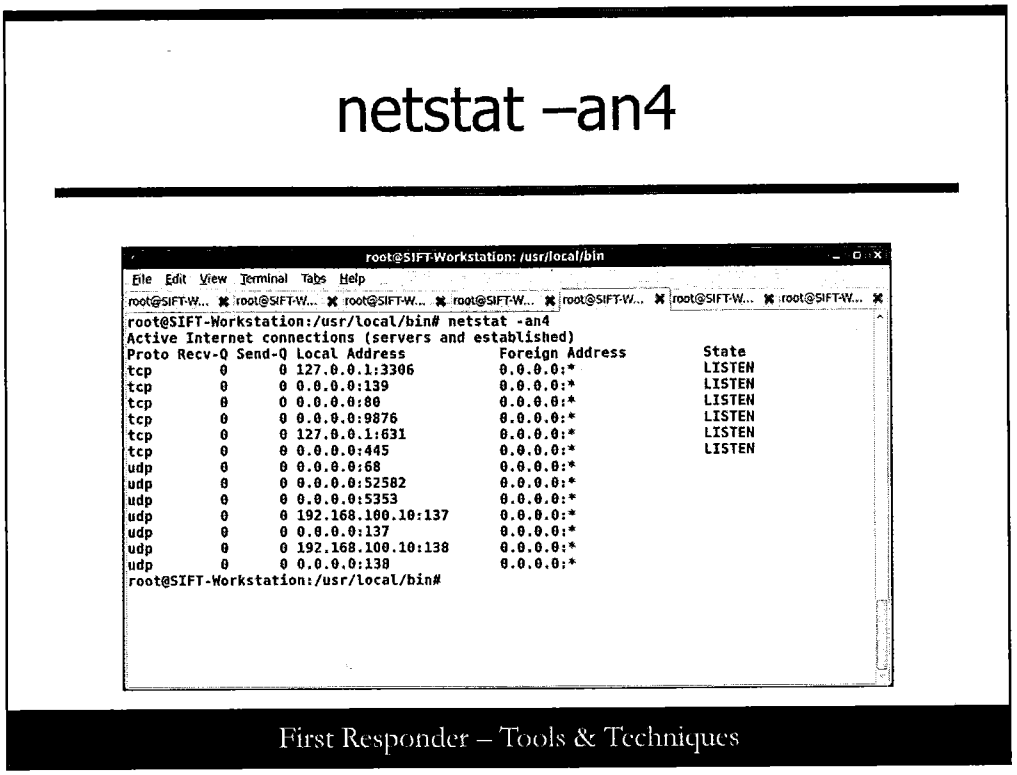
You can see in the following illustration we get a lot. We turned up the kernel name of *Linux*, the node name of *SIFT-Workstation*, the kernel release of *2.6.31-20*, and the kernel version (which includes build date info) of *#58-Ubuntu SMP Fri Mar 12 05:23:09 UTC 2010*. In addition the command reported the machine name of *i686* and the operating system of *GNU/Linux*.

Notice how some of that does not match up in order or some things are left out of the report? That can be explained. If a statistical area could not be collected or was "unknown," it was omitted. This is a big deal for you because you might go blissfully unaware about what a particular aspect of the machine is if you rely only on command output. (Gathering these details by physical inspection, interviews and questions, or build lists helps take care of this problem—hint, hint.)



Vmstat

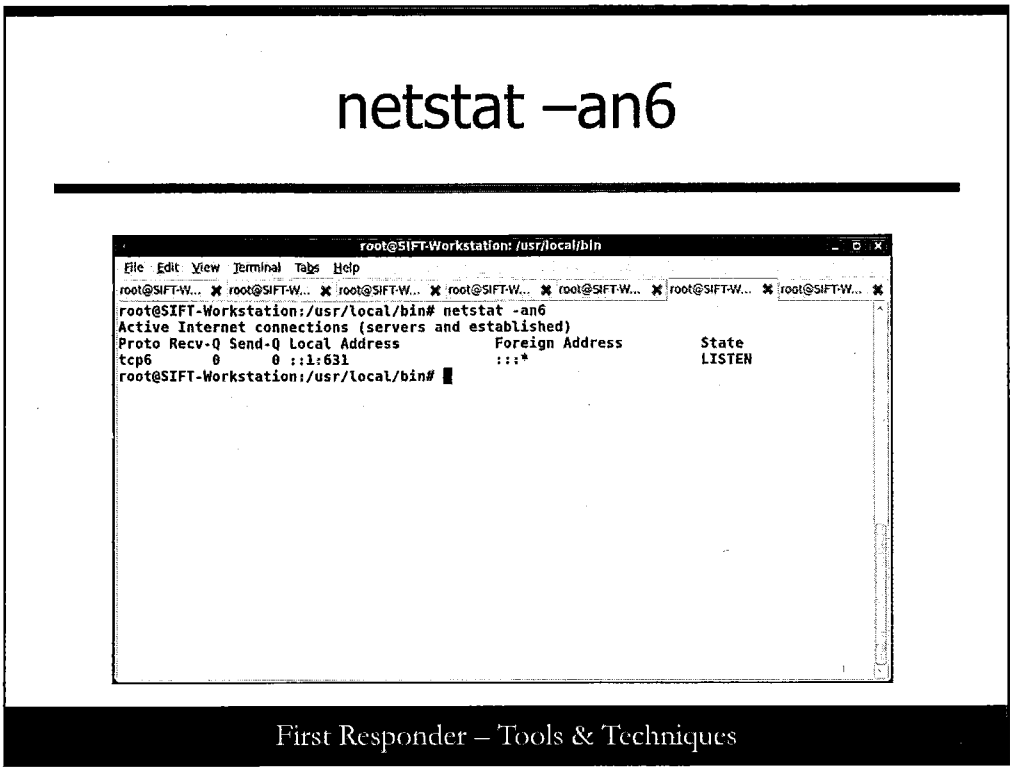
With *vmstat*, you get to see a report of the processes, memory, paging, disk, and even cpu statistics. These results are the averages since the last reboot.



Netstat -an4

The *netstat* series of commands provides a lot of information. It's a key tool most forensic practitioners and system administrators are familiar with. This time we broke its output into two categories: IP version 4 and IP version 6. This distinction becomes more and more important as IPv6 is adopted, so you should getting used to it now.

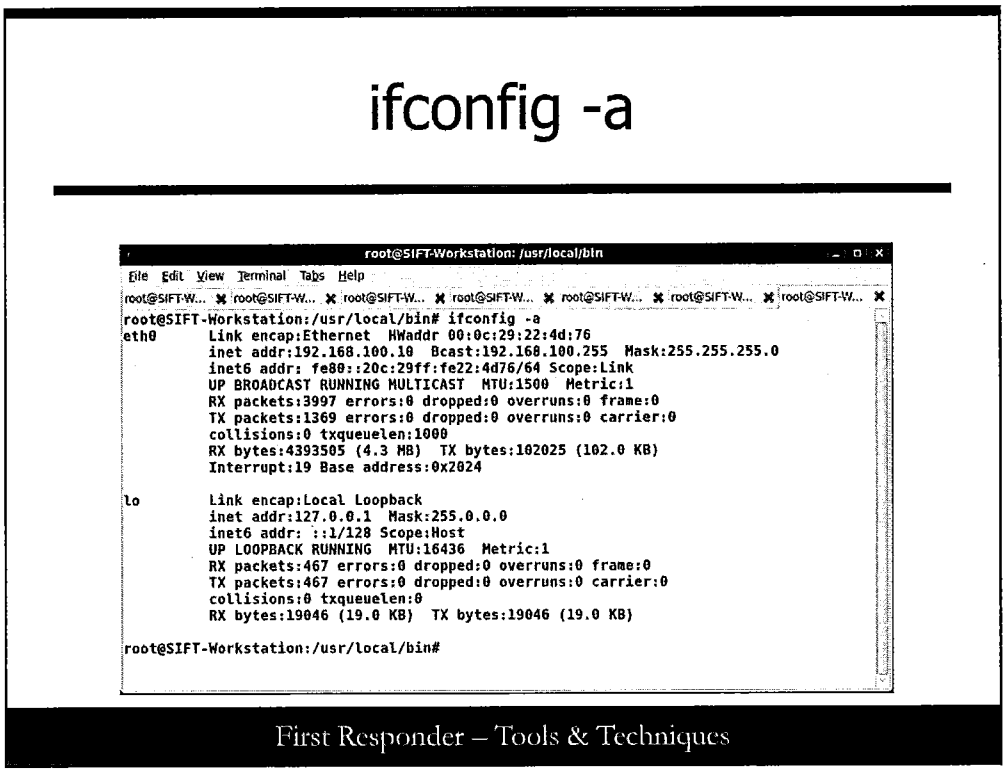
First, consider the output relevant to IPv4. Here when we issued the *netstat –an4* command, we got a listing. The *–a* was to include all connections, not just established connections, but also closed, terminating, and even listening connections. Notice how you see a listening state of the TCP but nothing for the UDP connections? This is explained because UDP is a stateless, best-effort-but-nothing-guaranteed network protocol (up on the Transport layer of the OSI Reference model) and it will not report the state it is in.



Netstat -an6

Then next we had you invoke *netstat –an6,* and you should have gotten a different result: A shorter list because it would be typical that only a short list of services would be configured on your SIFT machine to be communicating or listening on IPv6. Guess what? In this case, it is the Common UNIX Printing System (CUPS) daemon that is running.

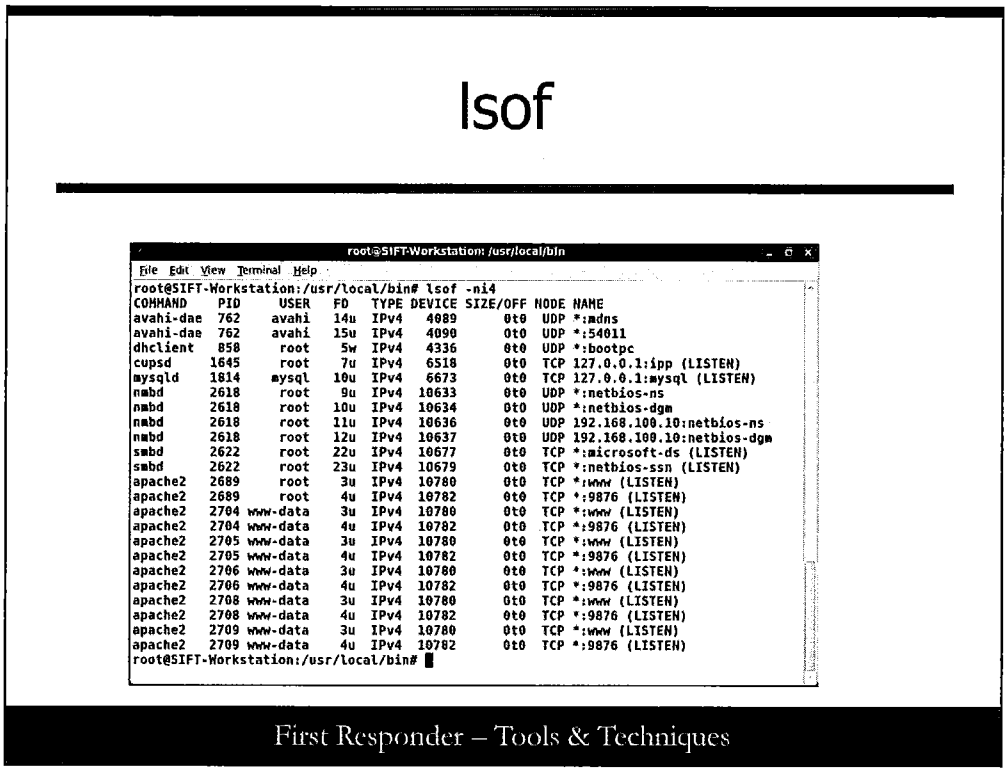
(Perhaps from a hardening aspect your system administrators out there would want to blow this service away or at least contain this to running only in IPv4-land if that is where you have your information security team's IDS sensors listening to ...) You might be surprised by how much is going on with fir communications via IPv6 in your networks that you could be unaware of.



ifconfig –a

Here is another one of the sysadmin- indispensable commands that reveals a ton of information to an incident examiner. But notice we invoked this command with a *–a* ? That's because we want *ifconfig* to show us all adapters, not only the ones that are alive and kicking.

Sometimes, adapters are in an off-state or are misconfigured. You want to see all your adapters, so if a NIC has been turned off, you might still capture that mode (such as promiscuous, for instance) but you have to tell ifconfig to look at every adapter addressable by the kernel.



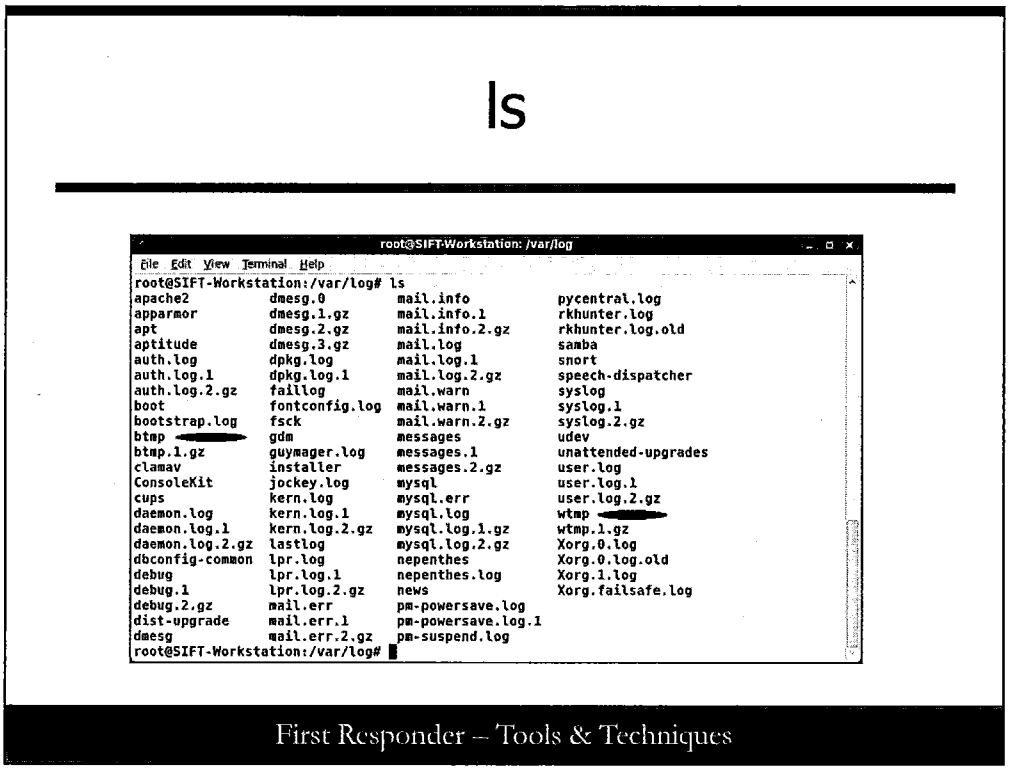
lsof

It's not just for open files. It can be used to determine the state of network connections and the programs, processes, or files involved in those connections.

Notice how many multiple instances of the Apache web server are running and the different source ports the daemon is running on! (At a glance of the upcoming screenshot, we see what appears to be six instances each of port 80 [www] and 9876. Much of this information can be confirmed by looking through the apache.conf file- hint, hint).

We purposely issued this command in this way to show that the *–i* switch is for Internet/network related files and the *–n* keeps it from resolving the IP addresses into canonical names or FQDNs, which slows down the performance of lsof when you need it to be quick. By this time you should have a clear understanding that the *-4* was included to contain the results to those of IPv4.

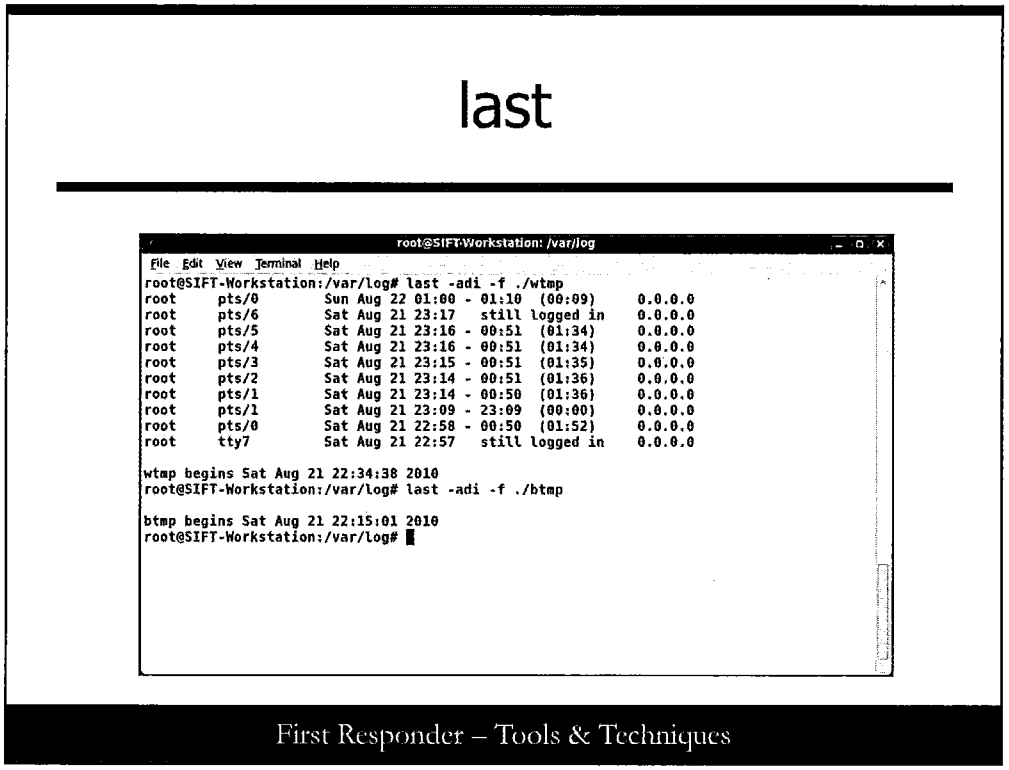
If you would like to, try issuing the command on your own, but this time have it report the status for IPv6. You should expect you can issue a *lsof –ni6* and get the expected results in a form consistent with IPv6 address format.



ls

Ahh … the wtmp and btmp files. Here by doing a listing of the directory contents of /var/log, we attempt to show you these files are currently in a sea of other log files. So we marked them with little pink-ovals to draw your attention to them.

The btmp and wtmp files are binary, nontext files, so viewing them in text-file reader would give you only garble. This is where the last command comes in because it is especially tuned to viewing the data structures of these files and displaying the resulting logon (or attempted logon in the case of btmp) activity.



last

Here is the output for our two login-log files. Notice that at the tail of the *last* commands we specified a *–f* switch and designated either the wtmp or the btmp file. We did this to force it to go to those specific files, one at a time. We also included the *–adi* switches to the *last* command to force it to tell us the following:

Display hostname (-a)

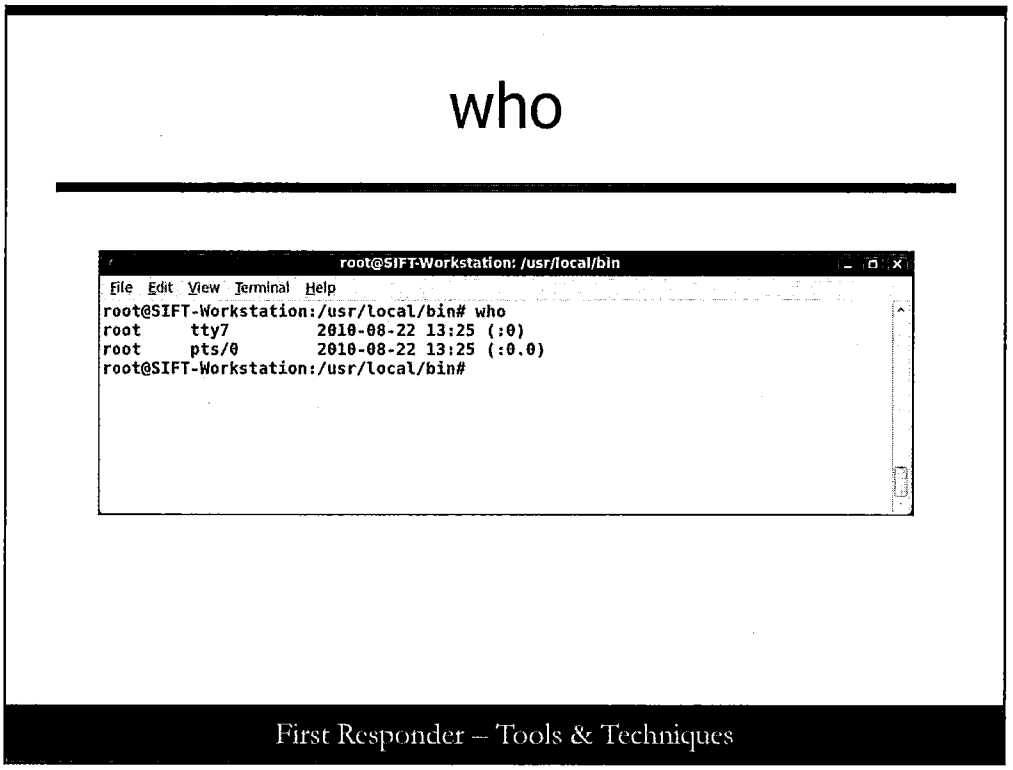
Display the non-local system (such as a remote system) (-d)

Relate this info into IP addresses in a dotted-quad numerical form (-i)

Note: Trying to resolve the addresses into FQDN takes a long time to perform, especially on a busy server. If speed is of the essence and you are trying to minimize the impact onto system during an incident, you can be in keeping with spirit of that intent by ***not resolving IP addresses*** into conical names.

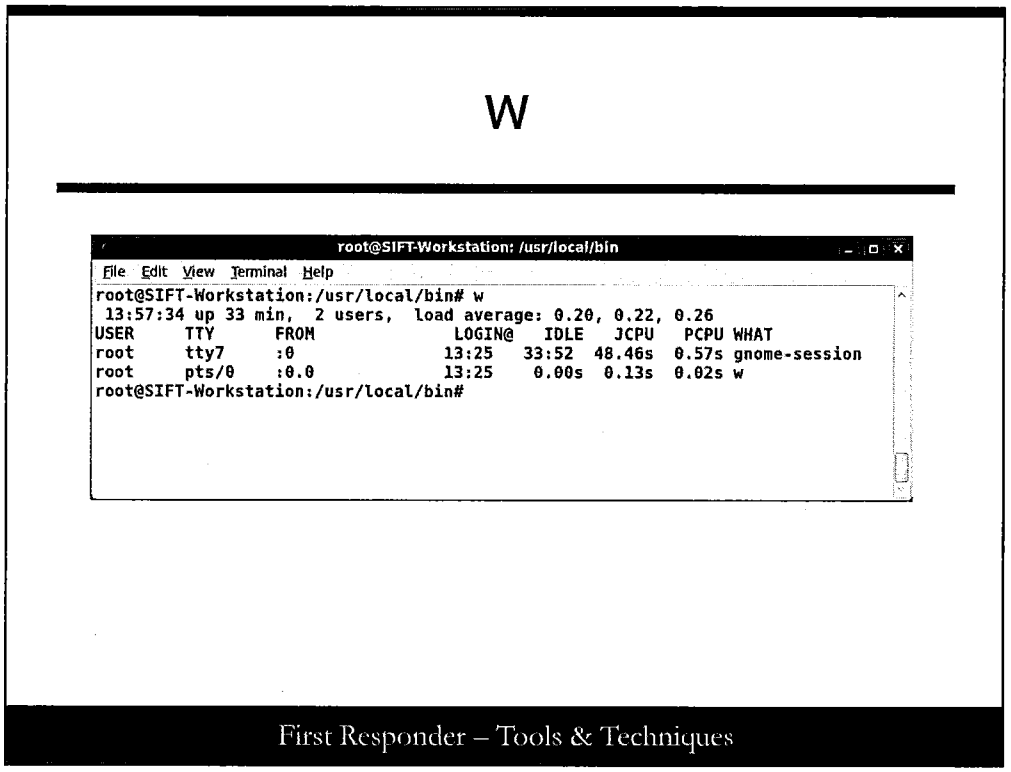
You should have a slightly lower list of root superuser logins (as far as the wtmp log is concerned) because you might have not been logging in to many terminals to create the screen shots of all these commands! But what should not differ (too much, anyway) is the number of bad logins, listed in the btmp log shown in the illustration.

This emphasizes the point that everything on the system is recorded, even opening up a terminal in an additional tab or window. It is a matter of finding these facts and correctly interpreting the meaning of them.



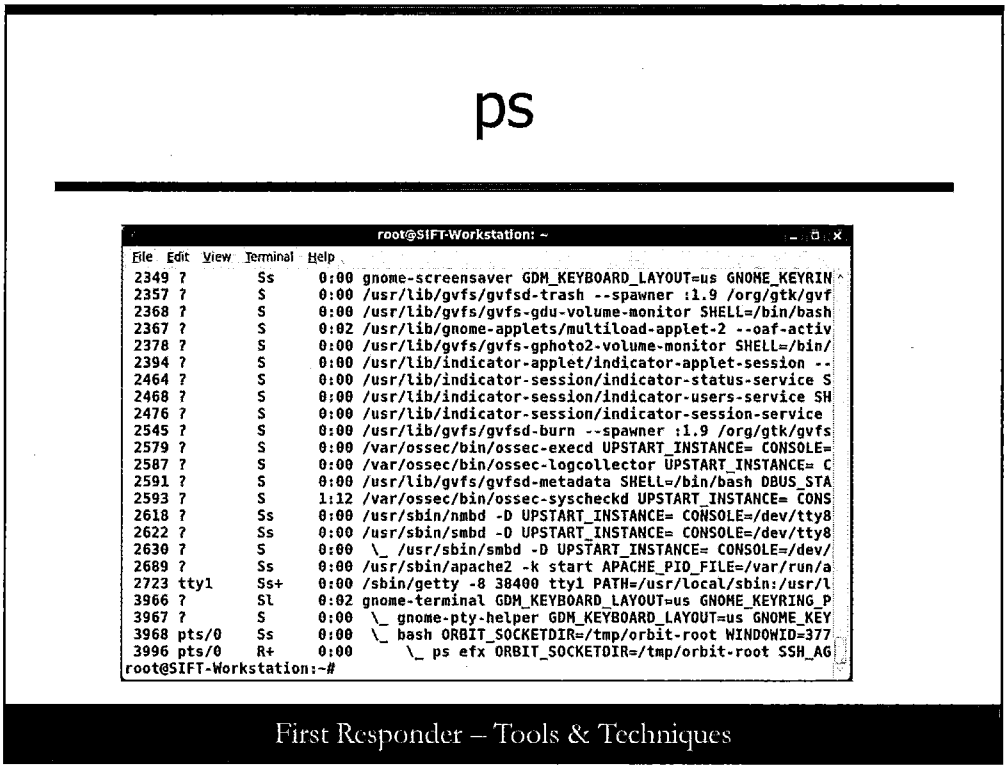
who

Here the *who* command returns the current status of the system, and when no options (in the form of switches) are passed to the command, it shows the names currently logged into the system.



w

W is used to print a summary of the current system usage/load by users. We get to see that the root user is responsible for the gnome-session on one terminal (tty7) and also is currently running *w* on another (in pts/0).

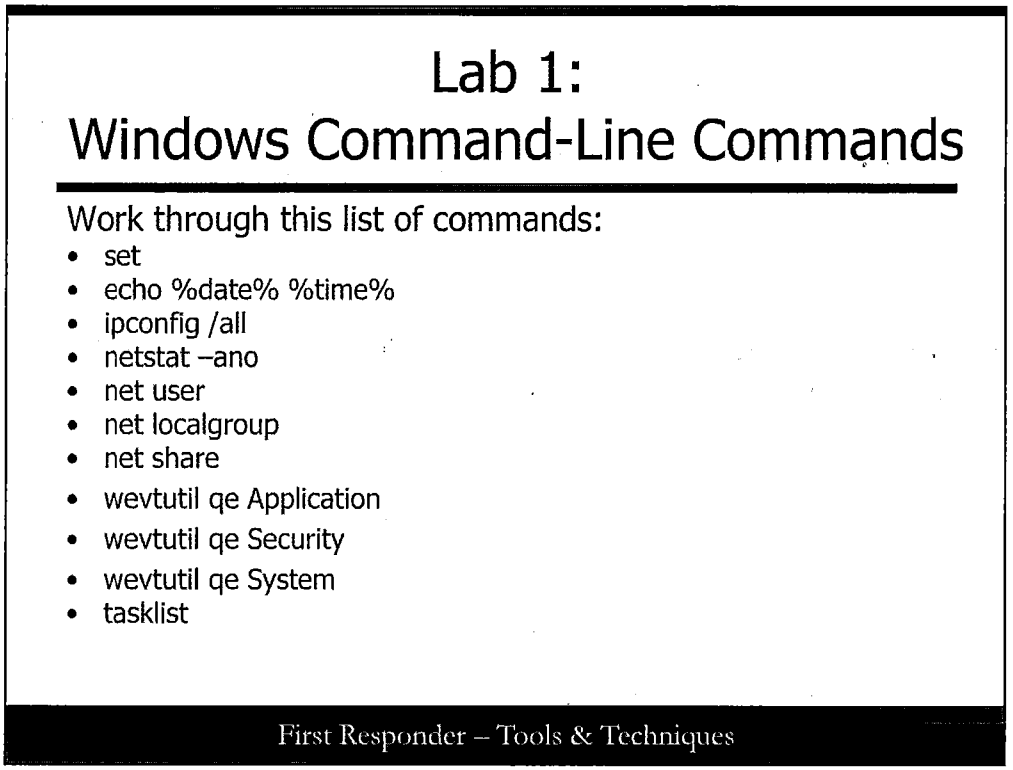


ps

The command *ps -aux* reports a ton of information and ps is considered one of the mainstays to a system administrator's tools in a Linux environment. The *–a* switch reports processes for all users; the *–u* informs the command to report the owning login name to the processes; and *-x* includes processes without an associated terminal. That is because *ps* is one of the few commands that uses these switches (as options) and processes them without the need of a—to give the parser a heads up that a switch follows.

In this command, the results were greater than one single terminal window would allow, so we captured the tail end of the results. But you should always consider the use of paging (with either *more*, *less*, or even *pg*) or use redirection (such as > or even better, >>) to a filename for a closer look and thorough inspection of process activity.

This concludes the section for Linux basic commands. The Windows command line results follow in the next section.



Lab 1: Windows Command-line Commands

Now you need to begin using a VMware virtual machine with an instance of Windows or a Windows 7 computer that you have access to during this course or your studies. Enter in the following commands and record your observations. While you do this, consider how these commands compare or are similar to those in a Linux environment. The explanations come up later in this lab, so do the best you can to get the most out of the exercise.

Issue the command *set* into a command prompt window. Record your observations here.

Issue the command *echo %date% %time%* into a command prompt window. Record the output here. Consider what the significance of the % signs mean.

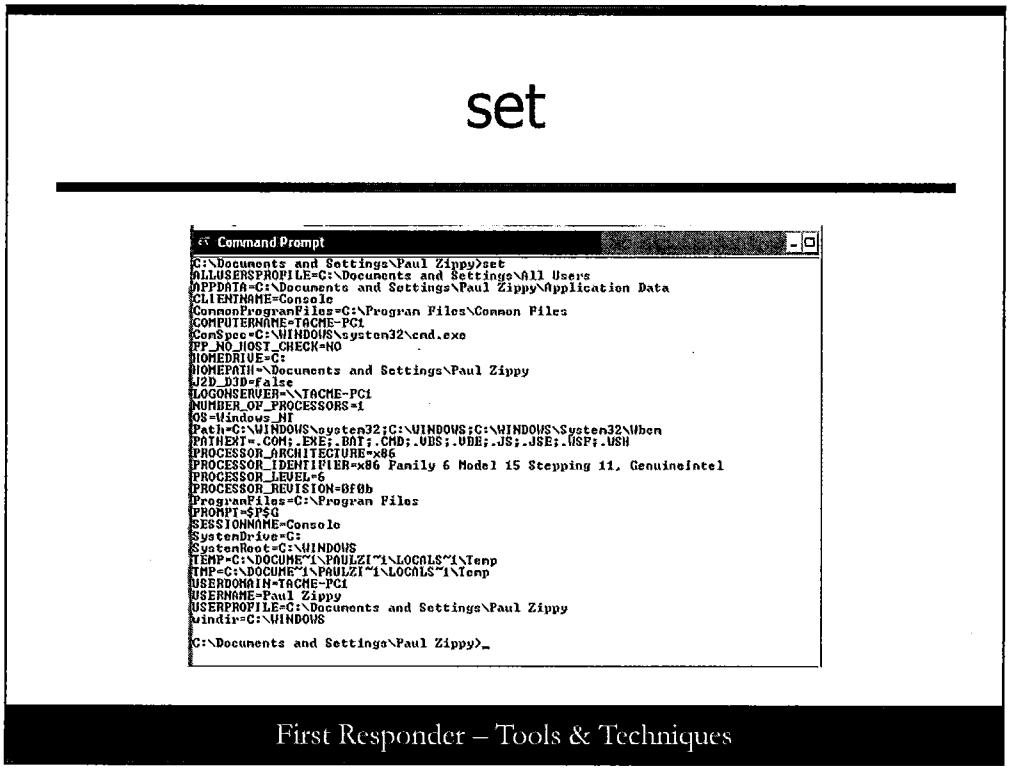
Issue the command *ipconfig /all* into a command prompt window. Record a brief summary of the information that is returned to you as a result.

Issue the command *netstat –ano* into a command prompt window. Write down a brief summary of what is returned to you as a result here.

Issue the following commands into a command prompt window.
net user
net localgroup
Record a brief summary of what is returned to you as a result here.

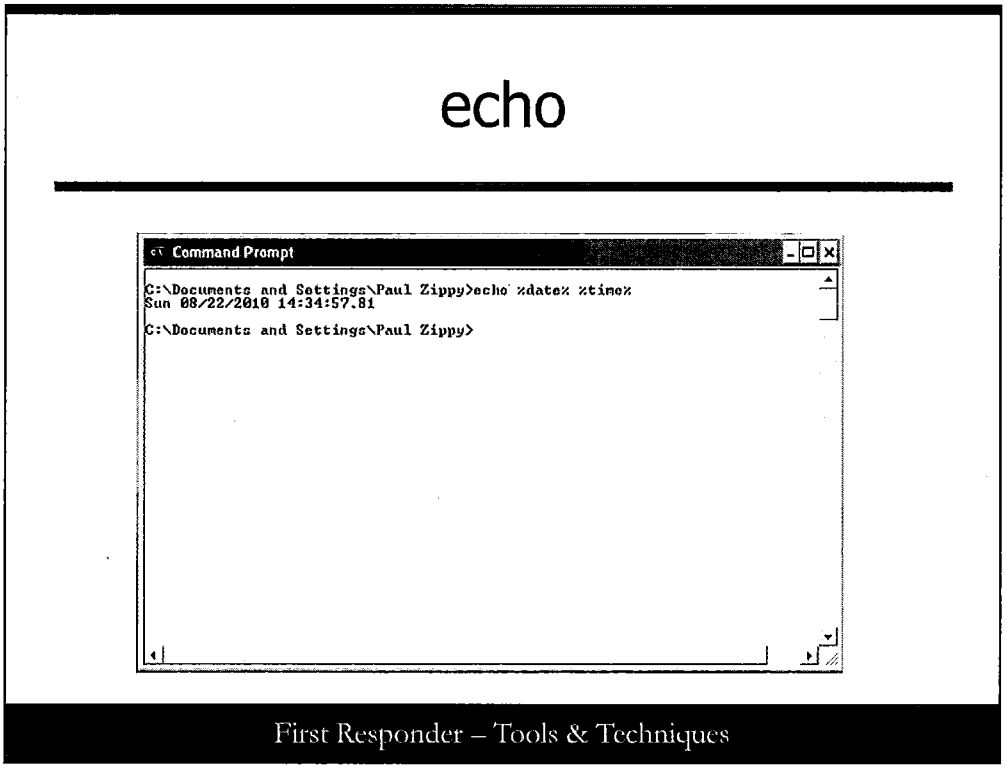
Issue the command *net share* into a command prompt window. Record the observations here.

Issue the command *tasklist* into a command prompt window. Record a brief summary of what is returned to you as a result here.



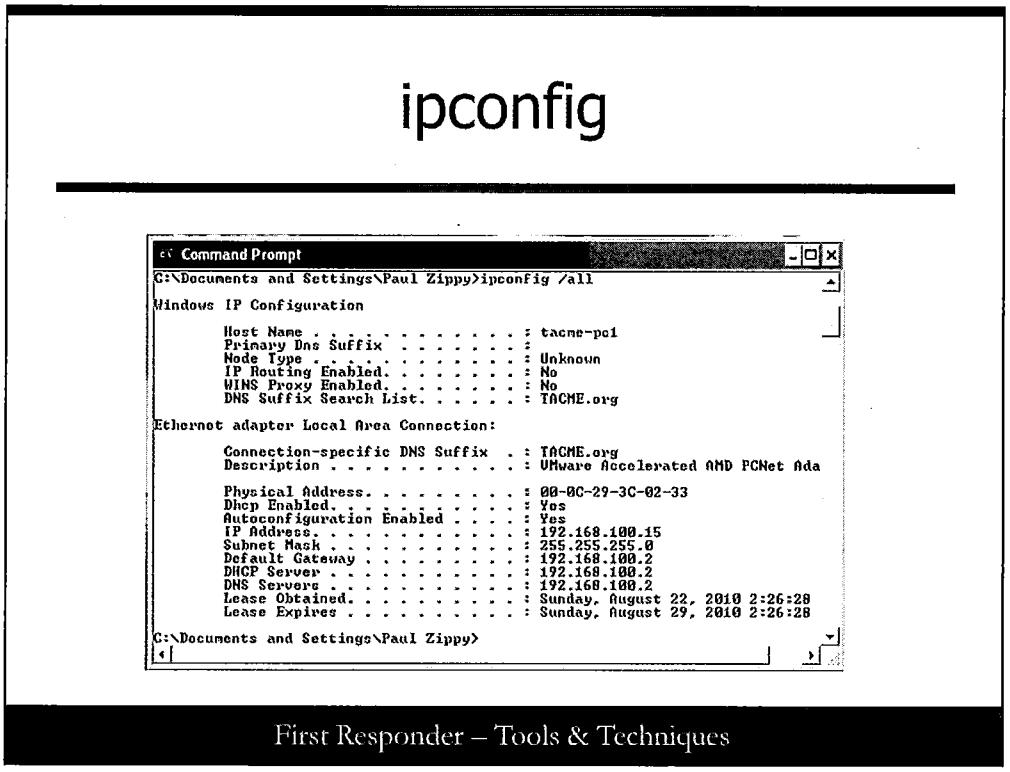
set

The *set* command is great for displaying the currently established variables that the Windows operating system refers to. Many of them are important and are helpful to an incident response analyst. Things such as the command execution path, the system drive, and the currently logged in user are all helpful to review.



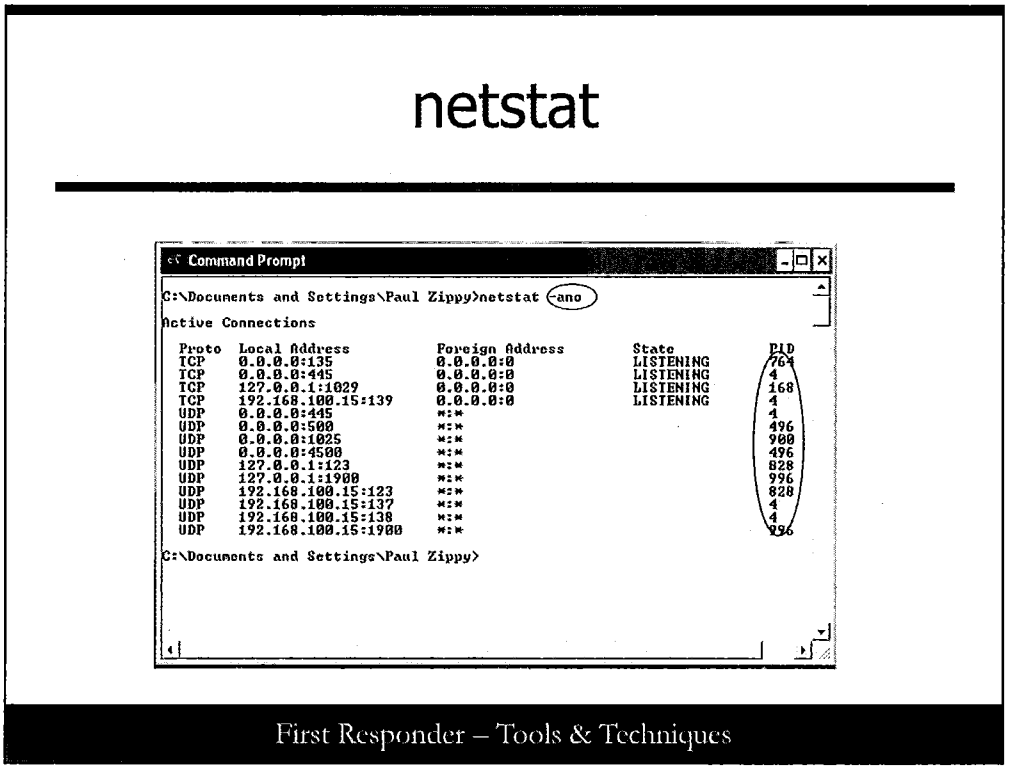
echo

This next command was more helpful and illustrative to issue on one single command line rather than two. By issuing *echo %date% %time%* we hoped you would key in that you can issue multiple variables (which are denoted by being surrounded by the %%) and get a display of the resulting information.



ipconfig

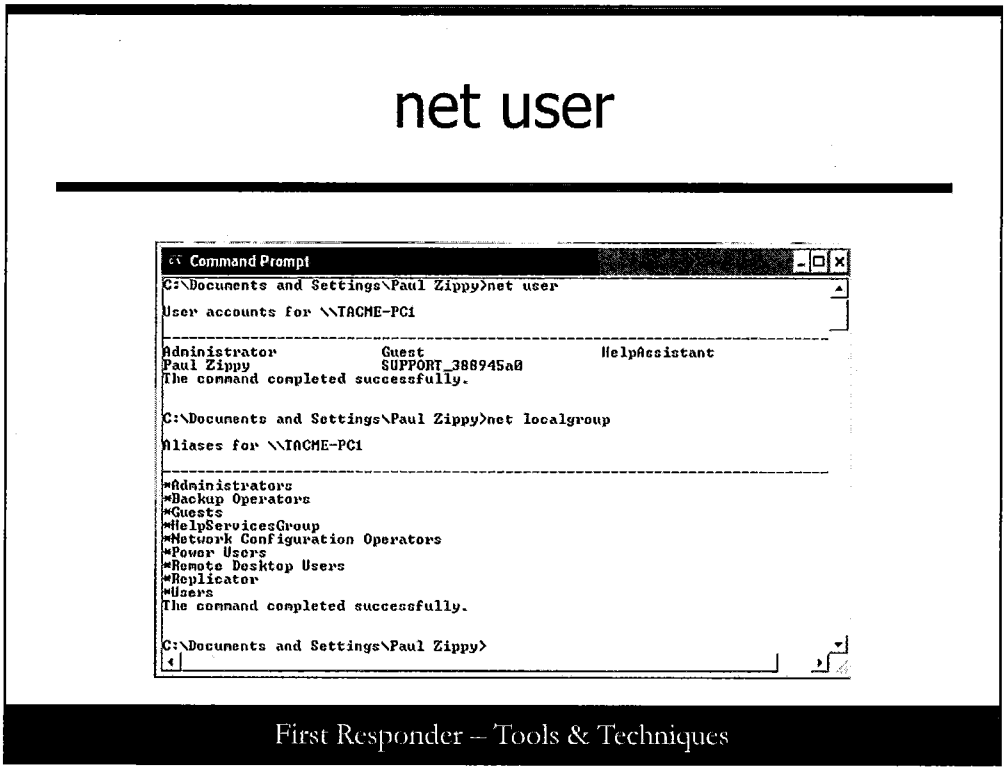
Most any person that had to work with an MS Windows operating system has had to issue an *ipconfig* command sometime in their experience. This time we issued *ipconfig /all* and we get the following display, which is also helpful. The */all* modifies the command to display all adapters that are loaded and do a deeper dive of information reported back, such as the DHCP and DNS servers, the lease time, as well as the hardware address of the network card and the PC hostname.



netstat

Like *ipconfig*, *netstat* is another mainstay command tool for the Windows environment used by technicians and analysts. In this case, we use the *netstat –ano*. The goal was to see all connections *(-a)*, display them in numerical form without performing an FQDN name resolution (*-n*), and for the command to provide the process id (*-o*) with the output.

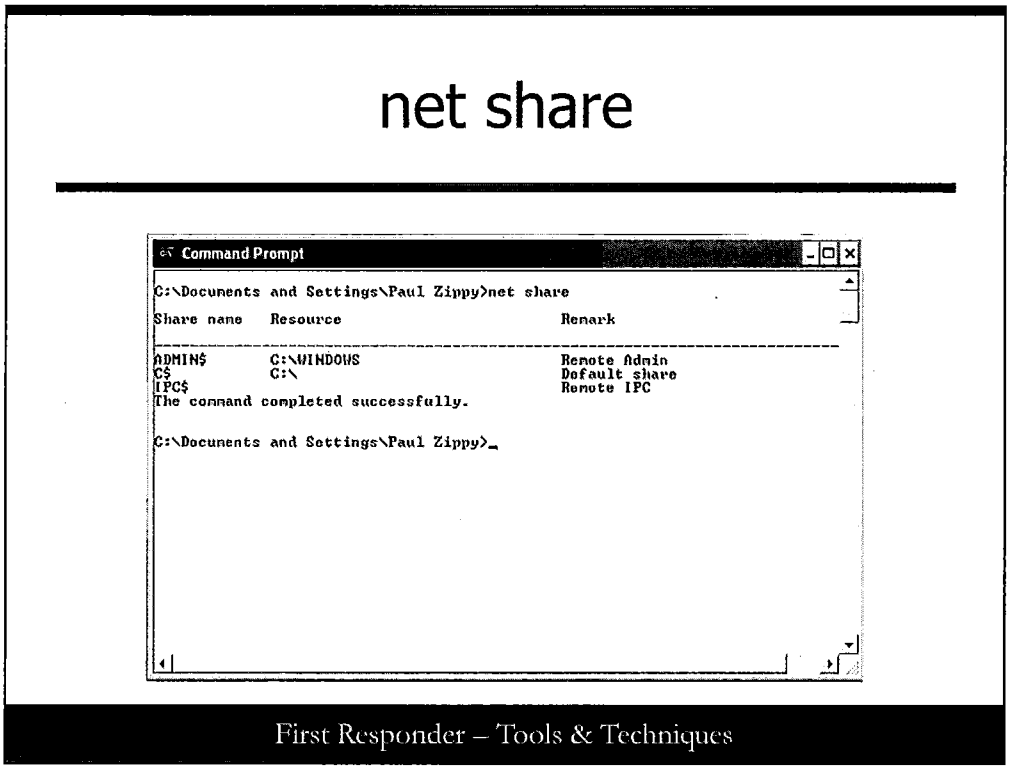
Note we are highlighting the process id (PID) in blue to call your attention to this category of information. Later in this exercise, you see this information again.



net user

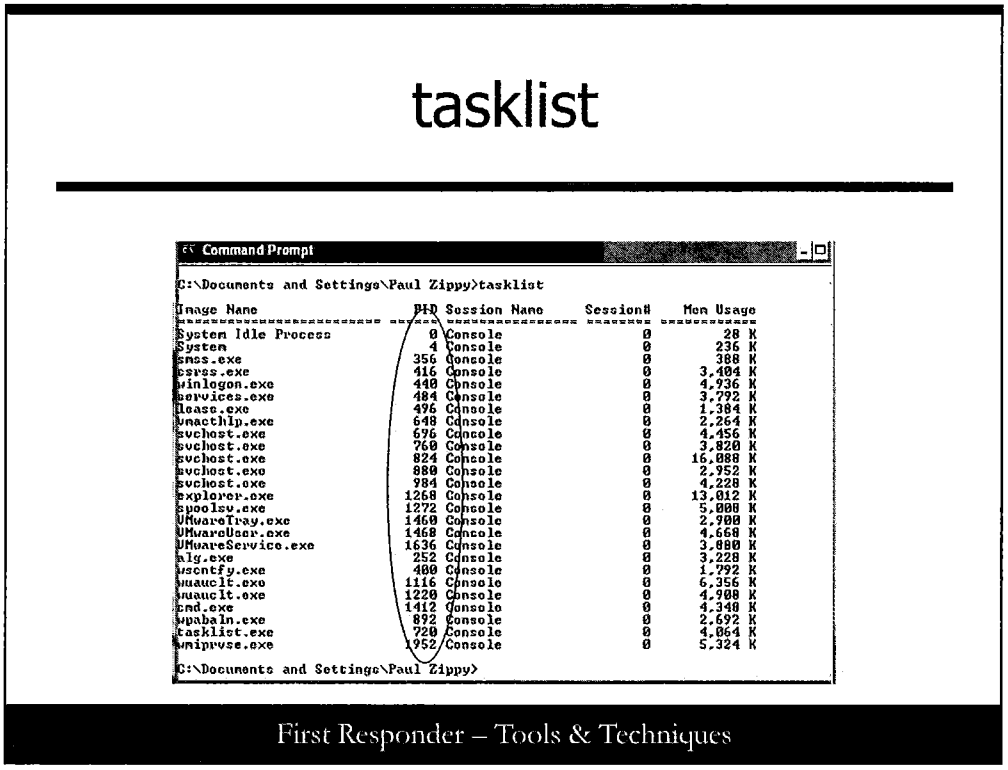
The next commands make use of the net suite of commands.

What you should expect when you issue a *net user* command into a command prompt is a listing of the local user accounts on the computer. To conserve space, we also issued the *net localgroup* command in the same window to show that there were local groups established on this live PC. You should do the same and have a similar experience.



net share

We've reached a point that we can show you the results of the *net share* command. If you have any default or designated share drives that you've established, they should show up when you issue the command as depicted.



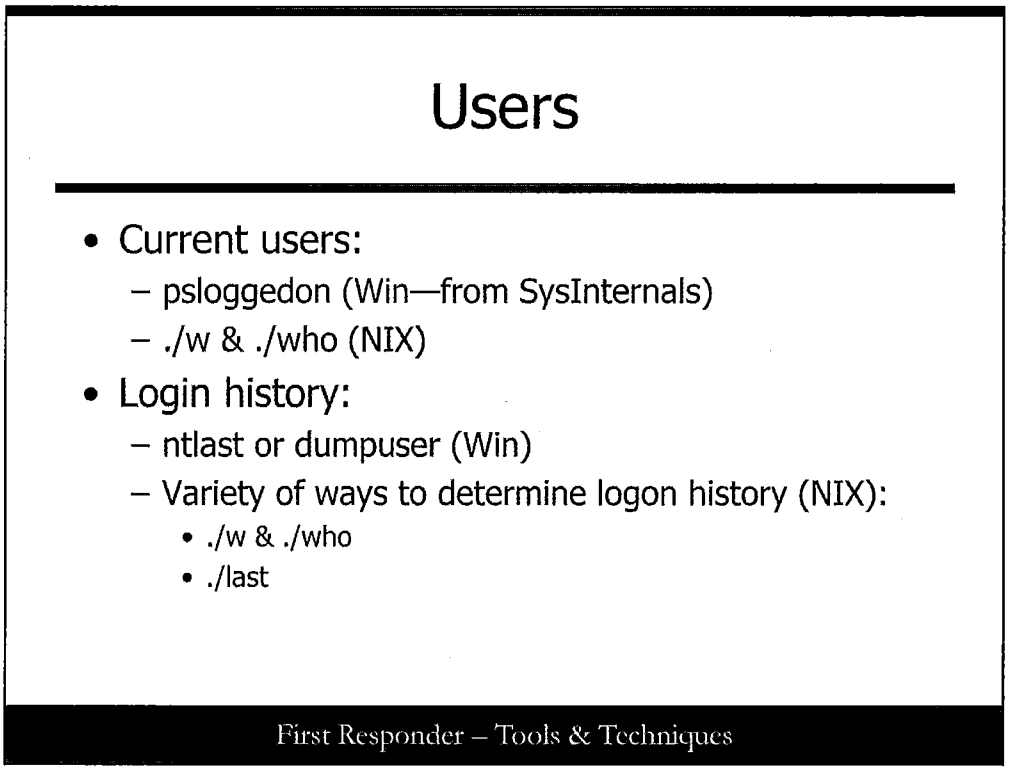
tasklist

We end this section with a usage of the *tasklist* command. This little wonder comes with a high degree of power, and it helps if you read up on it in ways that you can invoke this on a local system as well as remote systems that you have access to and permission to check/maintain.

We wanted you to see the output of *tasklist* without any switches and see that there was a correlation to earlier in this exercise. Here we have highlighted the PID column and ask that you go back to the bit on *netstat* to make a correlation.

For instance, if you go back you can find that PID 4 relates to TCP activity to port 445, which is in a listening state and awaiting a connection.

This concludes the section for Windows command-line commands.



Users

During assessment, it is essential to understand what accounts exist on the system, which accounts are legitimate, which accounts are currently being used, and which accounts have recently been used. For example, if an attacker created a new account, the creation time of the account should fit within your suspected timeframe if already known, or it should give you an idea of the suspected timeframe. If there is a baseline, it may be simple to determine new or unusual accounts. In addition, if you work with IT staff members that have been approved to work on the incident, you might consider asking them to tell you which accounts are legitimate or known.

Often, the attacker uses an existing privileged account or another existing account to escalate privileges. In this type of scenario, you need to understand more about the attack pattern and related information to rule out legitimate use. Specifically, you want to know if the user logging on is inside the network or remote, and if the system used for authentication is one that is typically used by that user account. In other words, if you see a legitimate account being used by an IP address in another country and there should be no foreign users, then this is an indication of something that might be suspect.

In Windows, if a user is remotely accessing the system, there will also be a corresponding NetBIOS session. Using psloggedon in Windows, you can see who is logged in and how (local or remote). Using ./w or ./who in Linux you can see who is logged in and how (local or remote). In addition, there are tools and commands that can be used to see logon history. In Windows, ntlast (www.foundstone.com) can be used to display logon history; however, the output may be dependent on audit settings. If you know the specific user accounts, you can use dumpuser to display statistics, including the last logon.

Following are various Linux commands that can be used:

The w command shows who is currently logged in and what tty or network connection they are using.

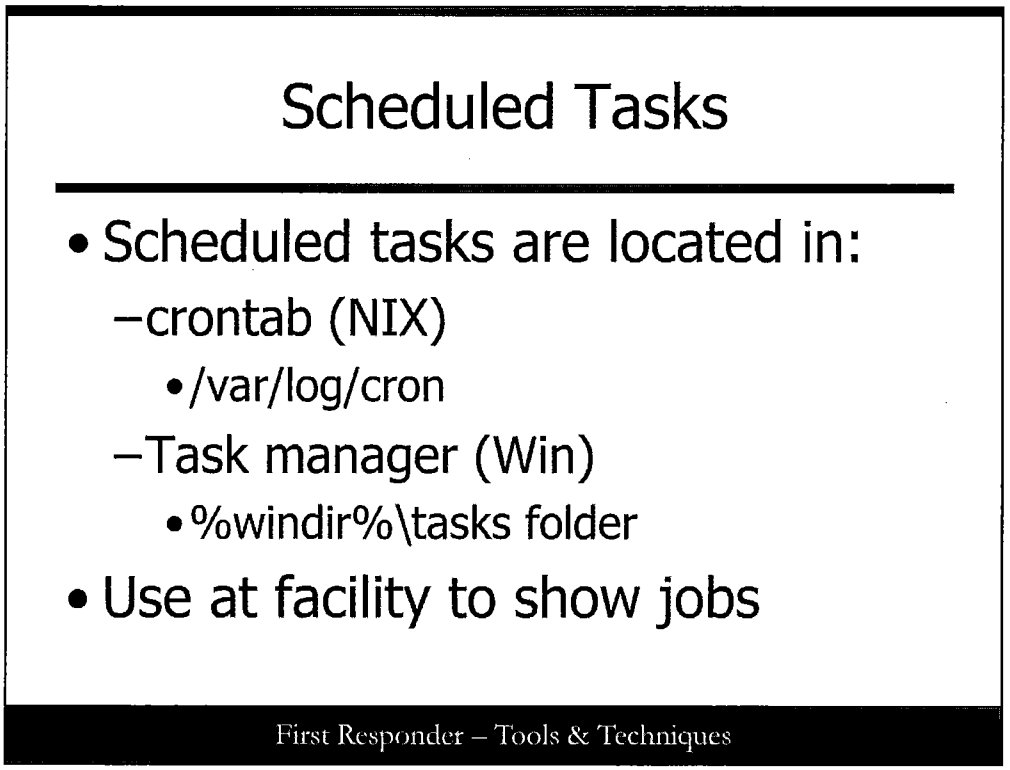
The who -r command can be used to see where someone is connecting from (by IP).

The last command shows who logged in last and their IP address and hostname. This command depends on the /var/log/wtmp file to exist and be readable. Note that hostname resolution will most likely trigger a DNS lookup. Note that the last command (reference the slide entitled "last") with the -aix options shows the hostname, IP address, shutdown, and run level changes. You should pay close attention to those last two items: There should be only shutdown and run level changes on a production system that can actually be accounted for. Note that frequent shutdowns over night may indicate that a user was attempting to ensure that processes started correctly on boot and were testing that facility.

The lastb command shows all failed login attempts and depends on there being a /var/log/btmp file on the system.

The who -Hl command shows who is actually logged in on the system at this time, along with the headings and their idle time. Of particular note would be users whose source is an IP address not on our network. On Debian, replace this command with who -u.

The finger -ls command shows details on when users last logged in, and it includes output from their .project and .plan files.



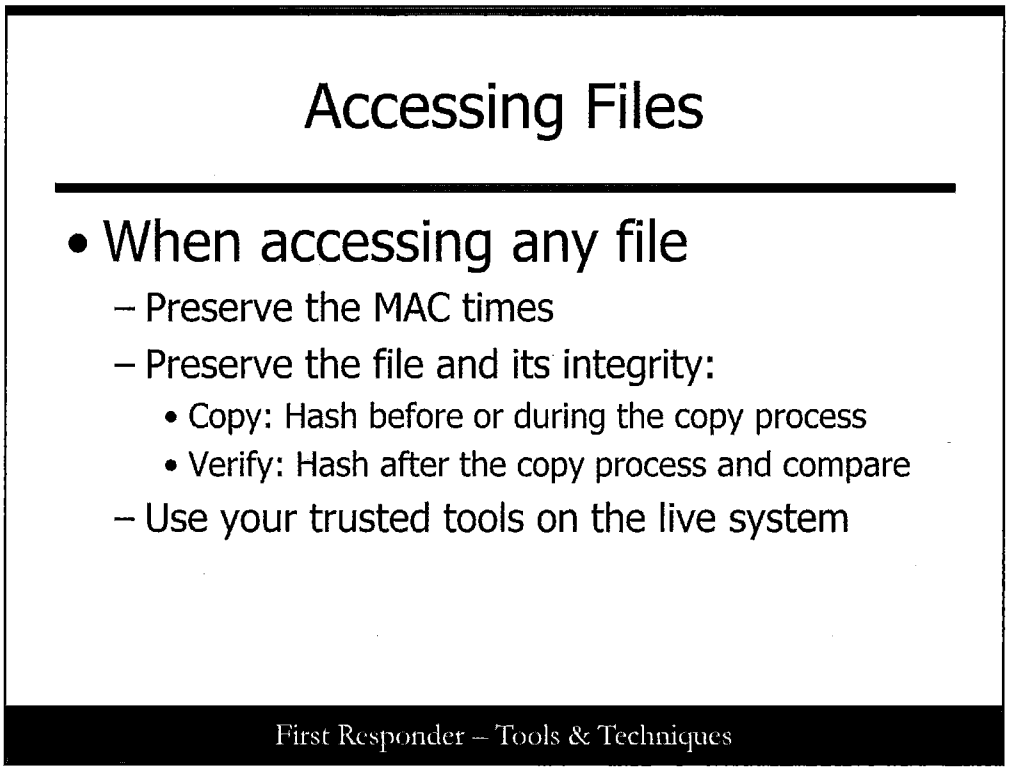
Scheduled Tasks
It is possible that an attacker can configure or add a job to the scheduler service for a variety of tasks: starting a backdoor listener, sending some private data, sending e-mail, or making a reverse connection to a web server to execute a command on the local system. Normally, the scheduler service is used by system administrators to run various scripts and tasks on the system.

There are some other facilities that can be checked during an incident; for instance, root's crontab, and whether users can have cron jobs.

The files in /etc/cron.d control which users can use both the cron and the facilities. The root user should be in /etc/cron.d/cron.allow, it should be root owned, and its permissions should be 600. The /etc/cron.d/at.allow file controls who can use the "at" facility, and it should be set up the same. There should be a /etc/cron.d/cron.deny file, and many sites recommend that all the users except root should be listed in this file. It should be root-owned with 600 permissions. The same holds true for the /etc/cron.d/at.deny file.

Windows has a built-in facility to run scheduled jobs, accessed from the Tasks folder and with the AT command. If necessary, you can also check the tasks folder on the system as part of incident response. Note that it is possible for a job file to be hidden (normally not visible in Explorer) so you should navigate to the tasks folder by using the command prompt. When there, run the attrib -r -s -h *.job command. This command unhides any job files in the directory, as it removes the read-only attribute from a file in this directory and then checks to see what's there.

Note: By using the attrib command, you are changing the MAC time information!

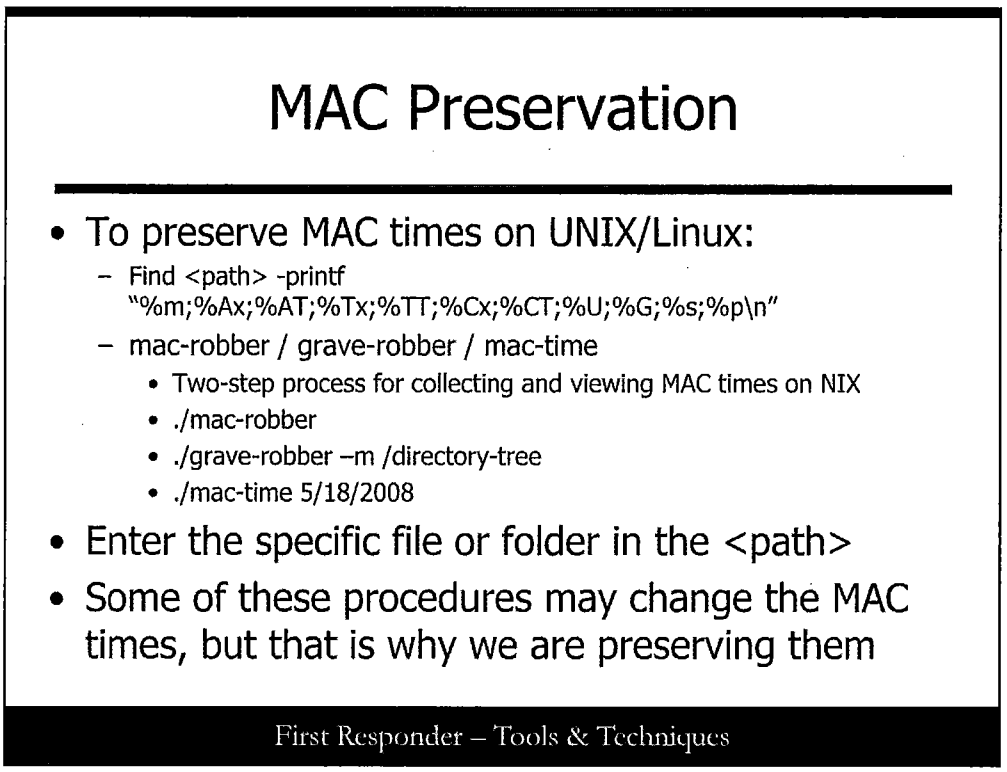


Accessing Files

Accessing files causes changes to the metadata associated with the file that tells you when the file was last accessed. In addition, other actions have an impact on MAC (Modified, Accessed, Created) times.

For a great reference on this, see Symantec's website (http://www.symantec.com/connect/articles/incident-response-tools-unix-part-two-file-system-tools), and the following tables:

- How Common Commands Change MACtimes for a Directory (foo)
- How Common Commands Change MACtimes for a File (f1)



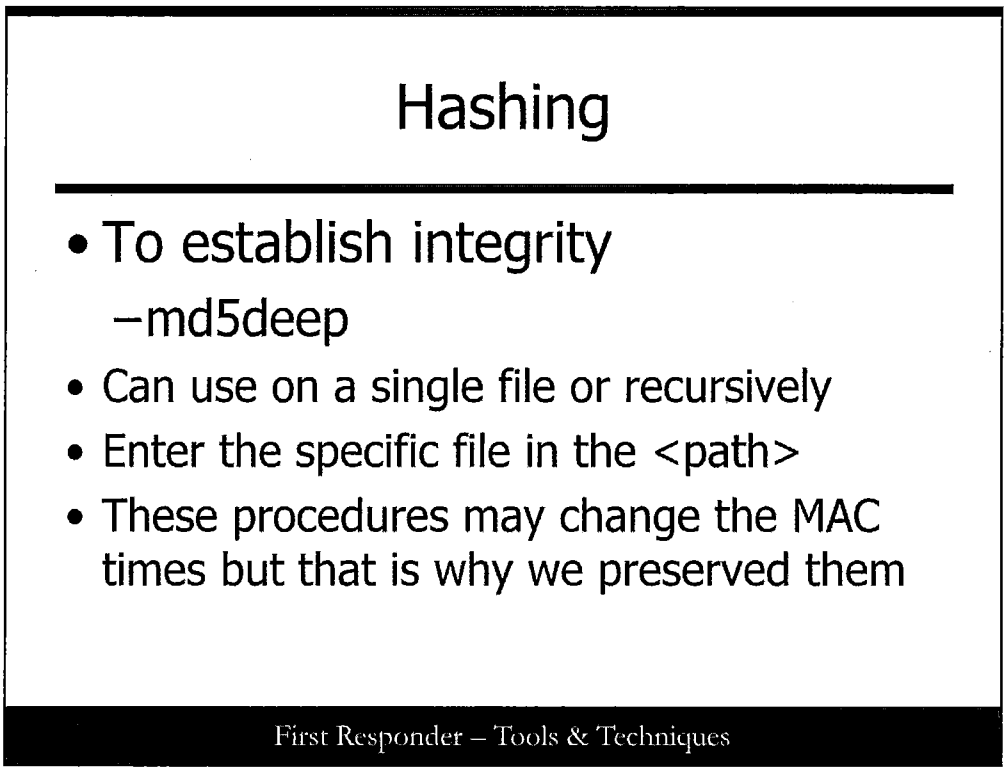
MAC Preservation

At this point in the investigation, the incident handling team needs to make a firm decision: Take the system down and image the disk, or collect MAC times for offline analysis. When volatile data has been collected, we want to have access to the system logs and system files. If the system can't be taken offline, or if we have not yet confirmed an incident, we may want to access these logs and files while the system is running so that we are not wasting time collecting a forensic image.

Accessing files on the system inherently modify the MAC times in a live state. (Unless the file system is mounted read-only, however, this is typically not the case for production file systems.) This included simply querying the file with the find command.

The first method (on this slide) is to use a two-step process to gather MAC times on the system. The first command is mac-robber, which is a digital forensic tool. This tool collects the MAC times from a mounted file system. The output is then used by mactime, which makes a timeline of file activity.

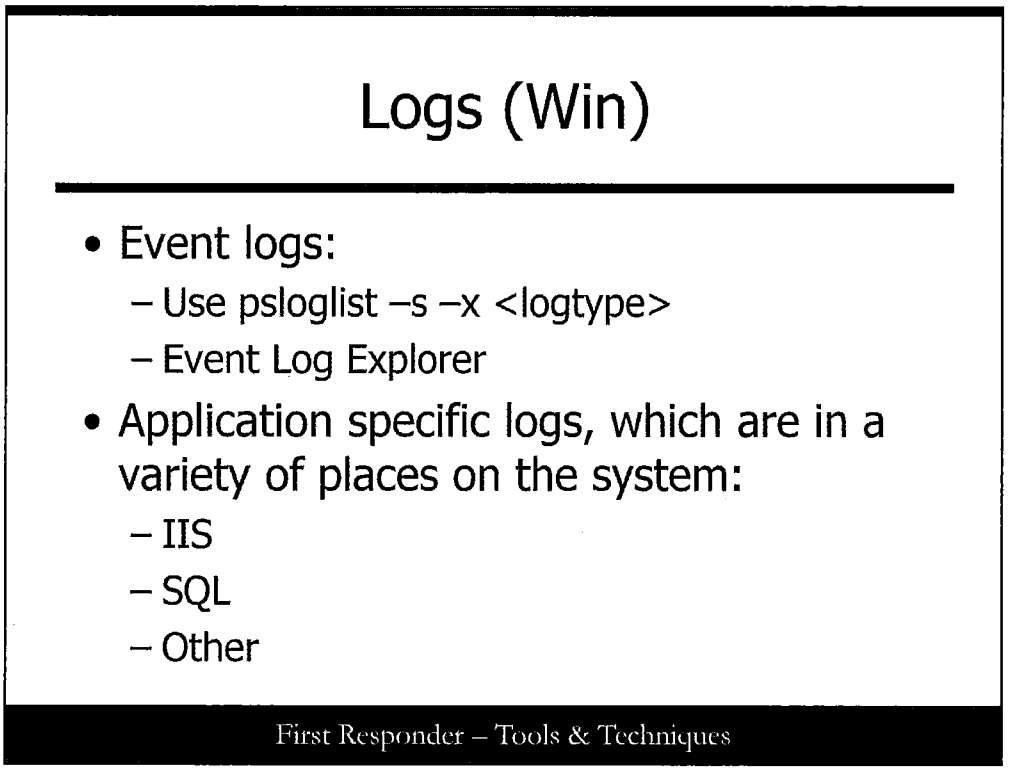
To use mac-robber, the file system must be mounted and should be mounted read-only. mac-robber can modify access times. It does not collect data on deleted files; this is a more advanced forensic skill. The mactime tool can take input from a variety of more advanced forensic tools. For the purposes of this course, it can also take data from the file system produced by mac-robber and then construct a MAC timeline for analysis.



Hashing

Hashing files is an excellent method of establishing integrity. If you use the MD5 algorithm to hash a file, make a duplicate of that file and subsequently hash it; the value should be the same barring any unforeseen issues. The point here is that this is a scientific way to reasonably prove that the files are the same after the copy process.

```
md5deep -r /root
md5deep -r /bin
```

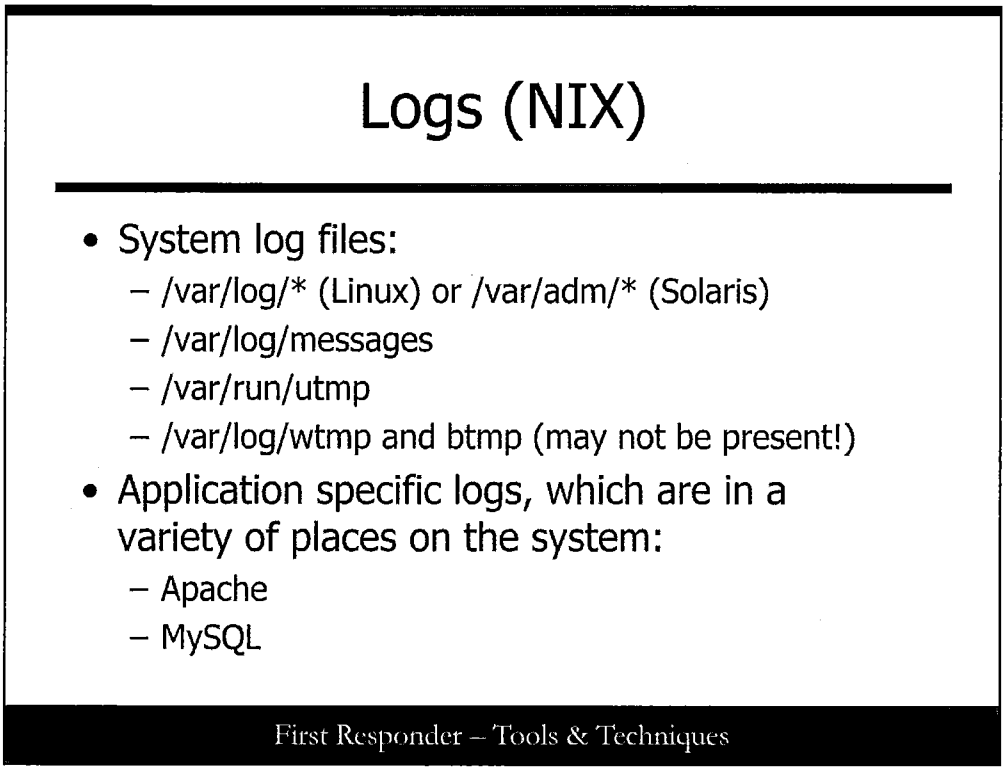


Logs (Win)

Event logs can also be copied and analyzed on another system, to a certain extent. Many of the event log's entries rely on the Registry, so certain details may be missing if you use this method; however, it can still provide you with some significant information.

Prior to Windows Vista (XP, 2003, 2000) these logs were stored on the .evt format and consisted of at least three event logs (Application, System, and Security). Servers also may have additional logs that are relative to their function in the domain; for instance, if the server were used for DNS, it might have a log called DNS Service.evt. Windows Vista, Windows 2008, Windows 7, Windows 8, and newer version use the newer .evtx format, which is stored in XML and uses channels within two groupings: Windows Logs and Application & Services Logs. The Application, System, and Security channels (similar to the .evt format) are stored in the first group (Windows Logs) along with the Setup and ForwardedEvents channels.

As previously mentioned, application logs can be extremely significant. If the compromise involves a database, you might want to look in the IIS logs for SQL commands such as UPDATE, DELETE, INSERT, SELECT, JOIN, and so on because these can be indicators of SQL injection. You may also want to look at application logs for all access during a specific time period or any access from a specific IP address.

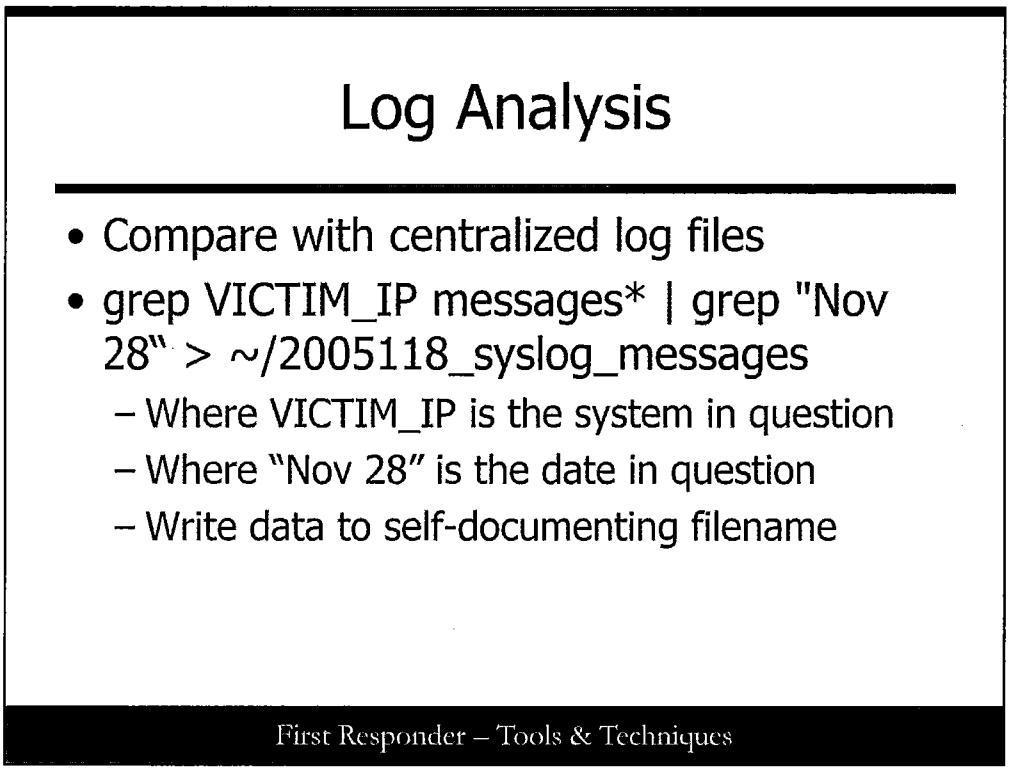


Logs (NIX)

The central syslog system often contains useful information pertaining to the incident you are investigating. On a typical Linux system, the majority of log messages are stored in /var/log/messages. On Solaris, the default log location is /var/adm. This should be the first place you look. If you have trouble finding the data you are looking for, check the /etc/syslog.conf file.

You can collect data for further analysis or preservation from the syslogs by grep'ing out the IP address and dates in question. Depending on the configuration of the syslog daemon, the compromised system's IP may be represented as an FQDN. So if the initial grep turns up empty, try the domain name of the system in place of the IP address.

It may also be possible to grep for the attacker's IP address in your central syslog system's logs. This is one of the advantages of having all systems syslog to a central server; it provides a one-stop shop for checking on issues like this. You may determine other systems that were targeted and possibly compromised. Do not write data to the local system unless there is absolutely no other option.



Log Analysis

Several log files on the system should be captured during incident response. They should be copied to removable media or copied off the system with netcat. Various files have security related data and during an incident should be preserved. As before, the decision must be made to either copy the data off system for real (or near real) time analysis or the incident handling team may choose to capture just volatile data and leave files on the disk, as they are planning on making a forensic backup of the disk for later analysis. This decision is based largely on the criticality of the asset. Does the investigation need to continue because the system must stay up?

The /var/run/utmp file contains information about who is currently logged on to the system, and its contents is normally shown with the w command.

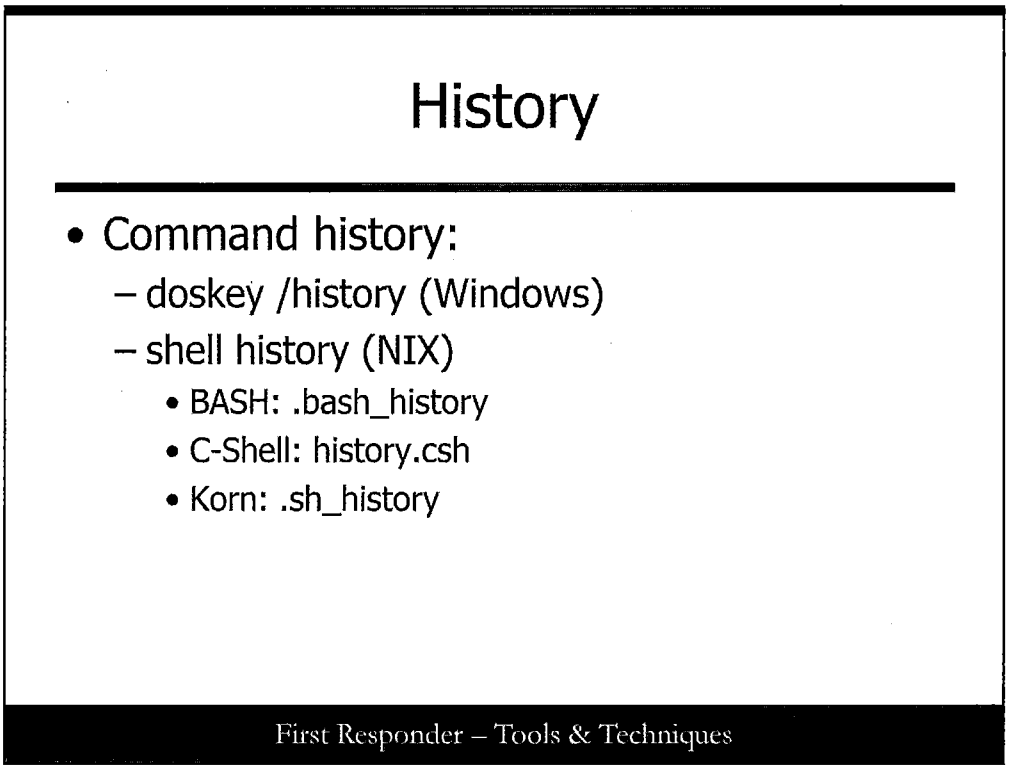
There are two login history files of interest. /var/log/wtmp contains successful login history because this file was created. (Make sure it's on your system.) /var/log/btmp contains the failed login history—bad login attempts, and on many systems it does not exist by default. You need to create it with the touch command, as root. Remember that on Linux (not Solaris) these files are read with the last and lastb commands, respectively.

System log files vary in number and log location. Some system administrators like to change the default location. Also, Sun Solaris systems use /var/adm, whereas Linux systems usually use /var/log. Occasionally, you will find system administrators who manage different operating system environments make their Linux systems match their Solaris systems. Note, also, that the /etc/syslog.conf defines the actual log directory location; so as part of incident response, this file should be checked (and possibly preserved) as well.

Applications can often log to a location outside of the default log directory, or they just might log to the standard log directory.

For example xferlog is the log that is recorded if the compromised system has a functioning FTP server. This log file should contain log records for all the FTP transfers.

Syslog can be configured to log to separate files based on facility and priority. The standard facility supported by most syslog implementations are user, kern, mail, daemon, auth, lpr, news, uucp, cron, local0-7, and mark. Some newer syslog implementations also support security and authpriv. For each syslog facility there are priorities, which represent the importance of the message being logged. The standard priorities are emerg, alert, crit, err, warning, notice, info, debug, and none. Commonly, applications that use syslog for logging enable the administrator to configure the facility and priority that application should use to log to syslog.



History

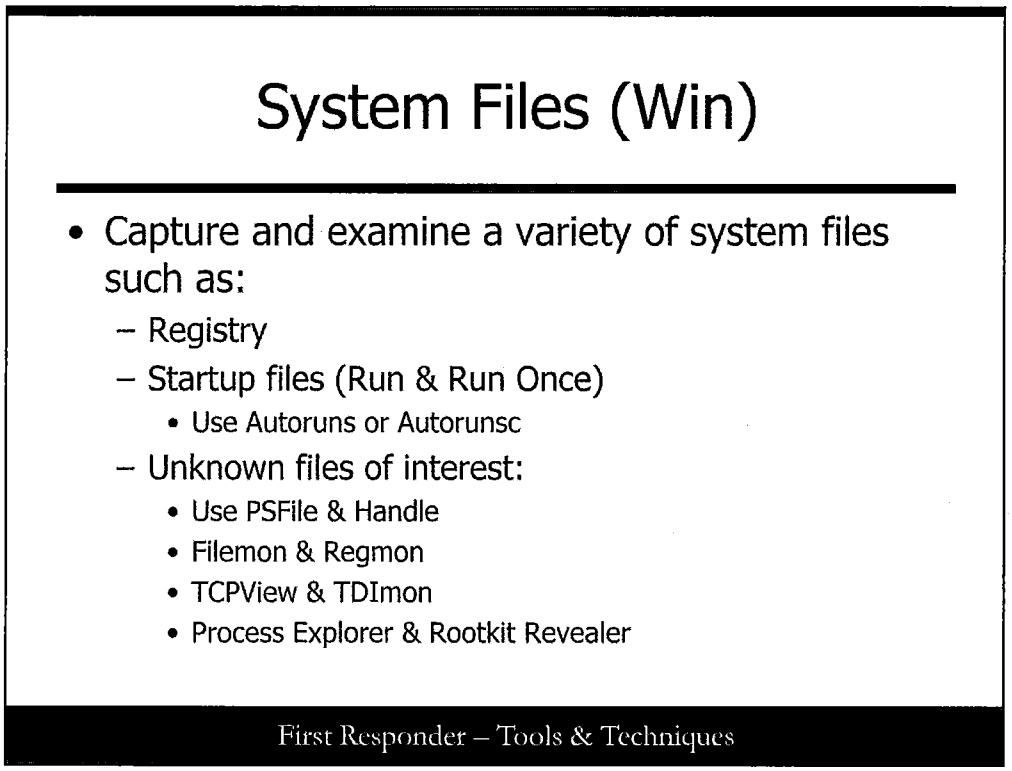
When used with the /history option, the doskey command shows all the commands typed in from the command line. Although many tools are launched from GUI (especially in a Windows environment), many are not. Having the command history when used with other volatile data that was gathered can help provide a more holistic understanding of what may have taken place on the target.

The doskey history is ONLY good for the current window. After the command window has been closed, the history goes away with it. The doskey command is, therefore, useful only if you find a suspect system with a command window open.

UNIX-based systems log the command shell history in the users .<shell>_history. The most popular shells store their history files in the following locations with the /home/<user> directory:

BASH: .bash_history
C-Shell: history.csh
Korn: .sh_history
POSIX: .sh_history
Z-Shell: .history

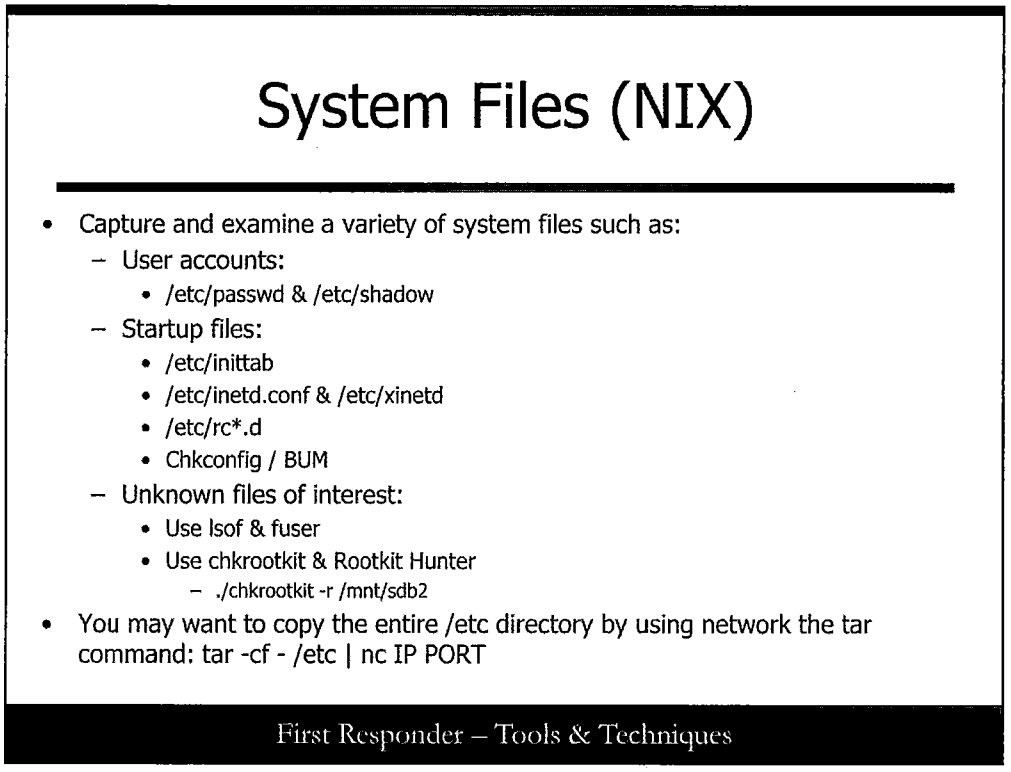
Understand that UNIX is a highly configurable operating system. The history files can literally be placed anywhere. If you cannot find them in the default locations, try using the find command, *find / -name .history.*



System Files (Win)

Autoruns is a utility that enables you to view the Registry keys and full path to programs that are executed at startup in the order in which they start. In addition, you can configure autorunsc to show other locations such as the Windows Explorer shell extensions, toolbars, browser help objects, winlogon notifications, auto-start services, and many other. Note that the Hide Signed Microsoft Entries option enables you to focus on third-party applications that have been added to the system.

Psfile is a utility that enables you to view if any files have been opened remotely on the system.



System Files (NIX)

When examining files and directories, there is at least one method an incident handler should be familiar with: using ls and echo * because these two commands show files but use a different method. Ideally, the list of files output should be the same. If not, there is something nefarious going on.

The /etc directory contains most of the configuration files for software on the system and the startup/shutdown directories - /etc/rc*.d. Based on how the incident is progressing, you might need to either collect important system files off the system or preserve the entire directory on the incident response system. In either case, several files should be checked.

/etc/passwd and /etc/shadow: These two files are the account database on a standard UNIX/Linux system. They should be checked for consistency. For instance, is the only account with UID 0 the actual root account? Do the system accounts (usually accounts with UIDs below 100 or 500, depending on architecture) have "disabled" shells and null or invalid passwords? Are there accounts on the system that cannot be accounted for? Are there accounts with unusual home directories or misspellings in the username field? These are all possible clues to accounts that may have been created on the system by an attacker under the premise that they will return and want to log in to the system.

Two files that should be checked are /etc/inittab and /etc.inetd.conf. The inittab file determines which services are automatically started/spawned at given run levels and what the default run level is. This file actually calls the start script for each of the run levels, starts TTYs and X11, and controls how the system responds to CTRL-ALT-DEL key sequences.

The /etc/inetd.conf file determines which services respond to incoming connection attempts. On most modern UNIX/Linux systems, this file is not used nearly as much as it was years ago. Most systems run a few specific processes, and the majority of the entries in inetd.conf should be commented out. For instance, by default SSH and the Apache web server do not run from inetd.conf.

The various /etc/rc*.d directories start and stop services. Of particular note here would be any file that recently changed. An example command to tar up the etc directory is

cd /etc; tar -zcvf /media/disk/2005.03.09-etc.tar.gz

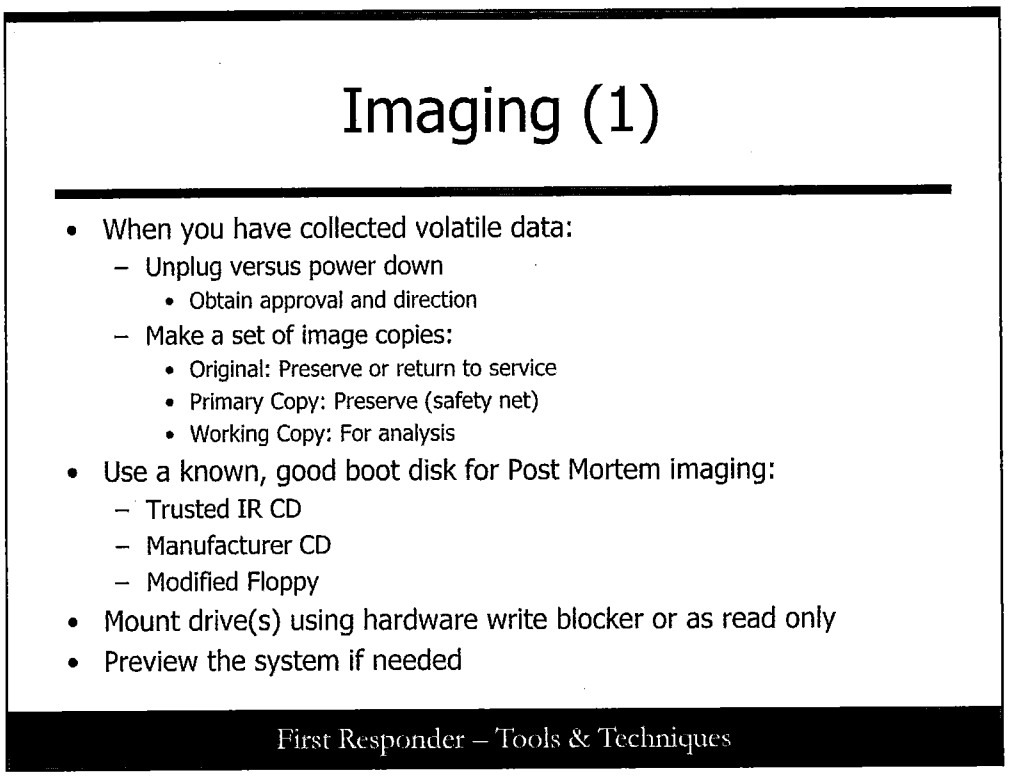
One command that should not be forgotten is the file command. This program uses the "magic number" database to determine what a file actually is. The magic number database is/usr/share/file/magic. Various files have different hex values in the first few characters of the file. You may not want to use the file command from the system; but if you capture data off of the system, using the file command on a response machine can tell you (hopefully) what the file is used for.

The readelf tool reads the Executable Linking and Format (ELF) headers of a binary file.

The strings command can be used to search through a file for ASCII printable character strings. Note that by default strings looks for four (4) character strings, and often this option produces more data than its worth. When using strings, you should send the output to an intermediate file and then use grep to search the output for keywords of interest.

The UNIX/Linux startup environment begins with the bootloader, which basically loads the kernel, which in turns starts the init process (usually in /sbin). First, init processes the /etc/inittab file, which lists the default init level and configures ttys on the system. Based on the init level, the system cycles through the /etc/rc*.d files and processes them in order. Considering this, any of the files in /etc/rc*.d may have been changed.

A variety of checks can be made to see if these startup scripts are valid or altered. First, time and date changes of the files at or near the time of the incident are a good clue, and second, S* and K* scripts that don't match installed services (or services that should be on the system). For instance, if you found S77ircd in /etc/rc3.d, this *may* indicate that someone has installed an IRC daemon on the system.



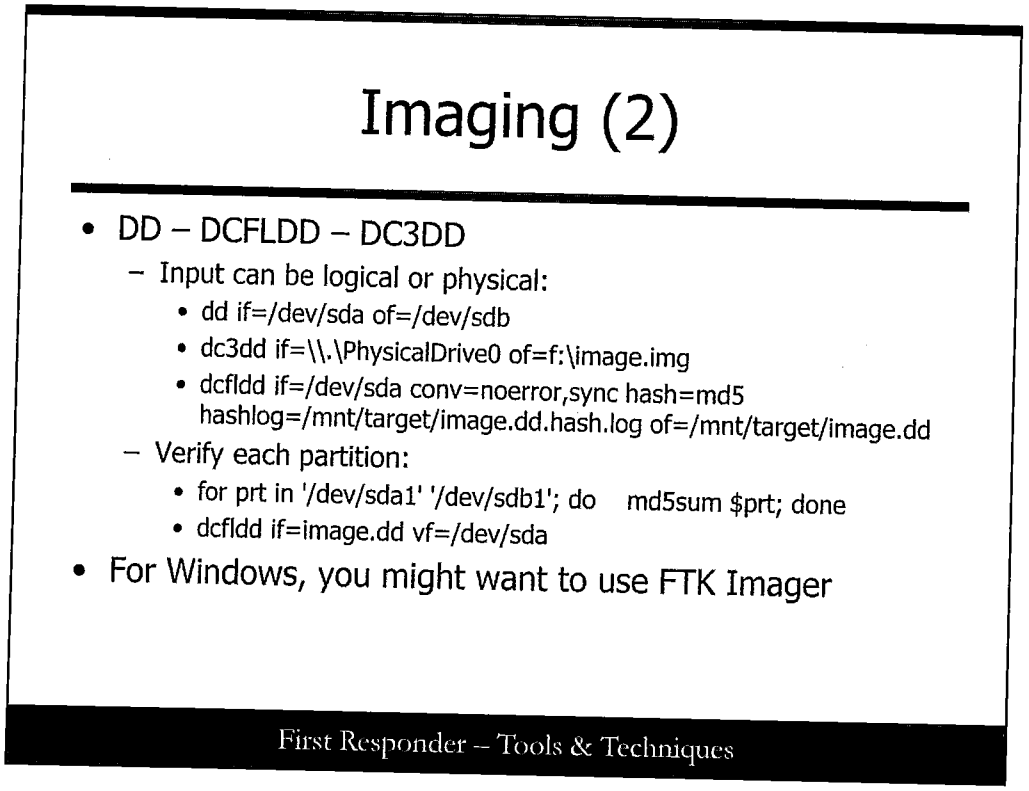
Imaging (1)
Often the incident handling team decides that the system must be disconnected from the network. You want to regain control of the machine, and often the best way to do that is to disconnect it from the network. Make sure that you have collected any network connection data before the system is disconnected.

There is some controversy about the order and type of commands that should be run on a system before it is disconnected and/or powered down. Some feel that the system should be unplugged and then the disks imaged. Some, including your course author, believe that mission-critical applications should be shut down and then the system can be unplugged. Remember, that it's more likely that in cases involving UNIX/Linux systems, a server process is running and many people depend on it, so the validity of that data should be ensured to protect the business.

You should consider going to single-user mode as a potential option (usually done with the init command), and then use some of the steps previously discussed after multi-user services are down. This action may minimize the risk of losing critical data on the system, but it may also alert automated software on the system.

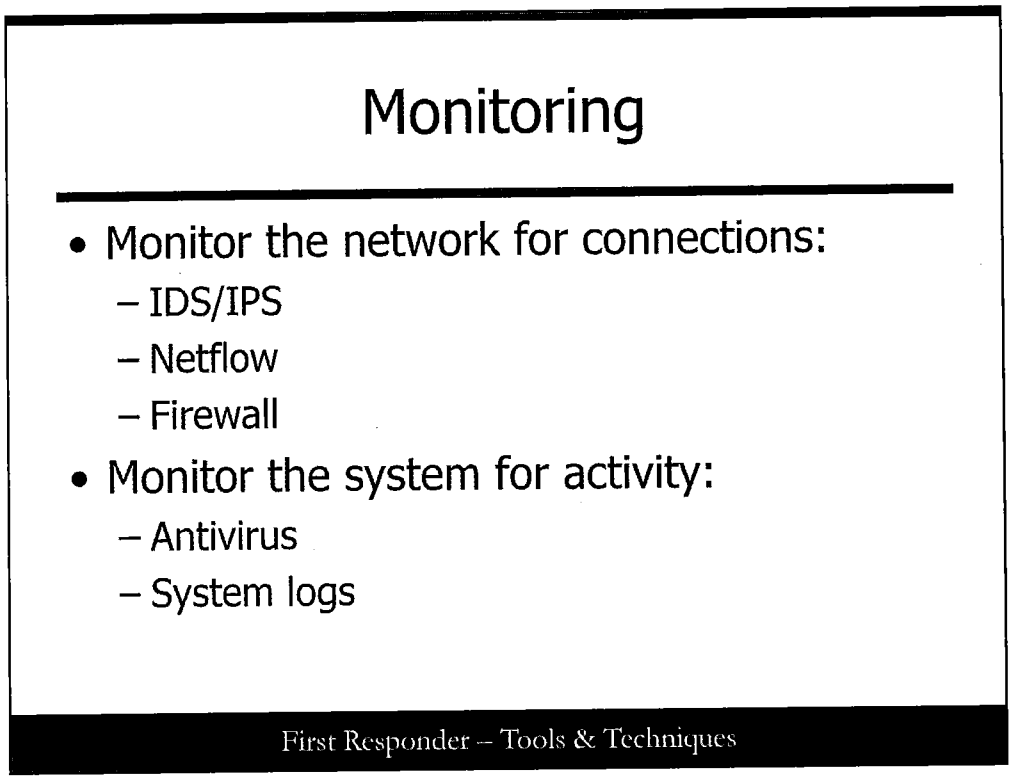
Current best practice states that during an incident, the entire disk should be imaged. When a decision is made to disconnect a system, the incident handler needs to plan for and make several backups of the system's disks. Under the best circumstances, four copies are ideal:

- **Original**: Preserve (as best evidence) or return to service
- **One**: Preserve (would become best evidence if the original is returned to service)
- **Two**: Working copy



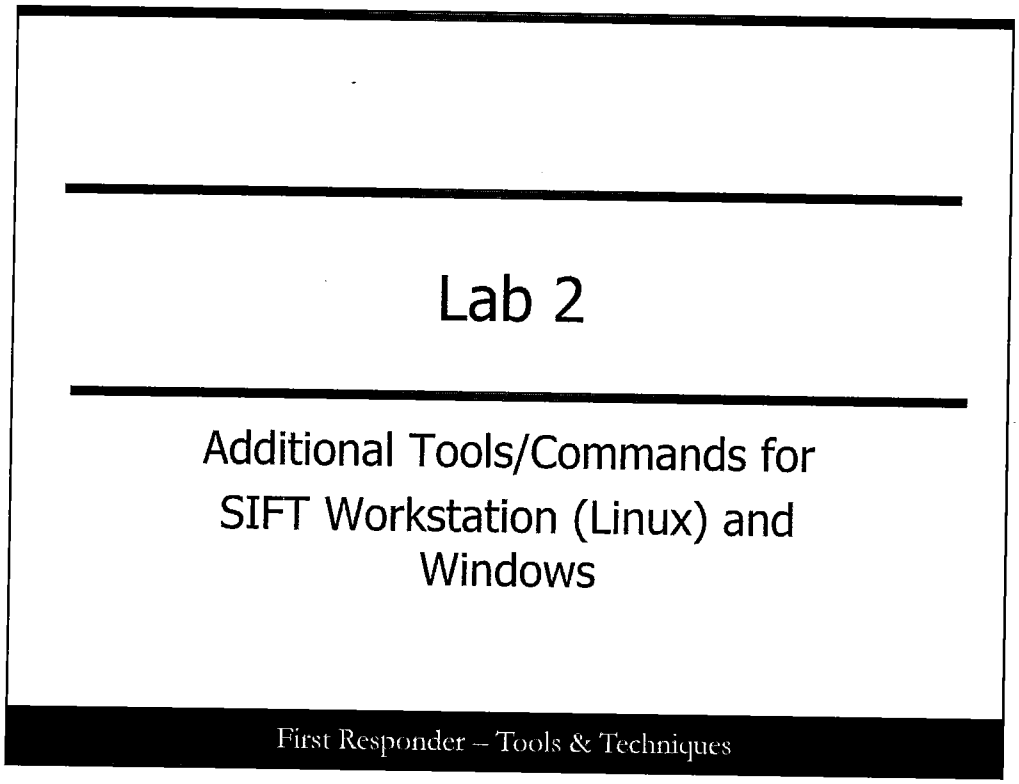
Imaging (2)

The most versatile imaging tool is the dd family of tools because they can essentially be used on any system, and the output format can be read by the most forensic tools. Input for an image can be logical or physical. In addition, imaging tools can be used for logical files. dcfldd and dc3dd have additional functionality such as hashing on-the-fly, which can be of significant value. It is important to verify your images to know if the process were successful.

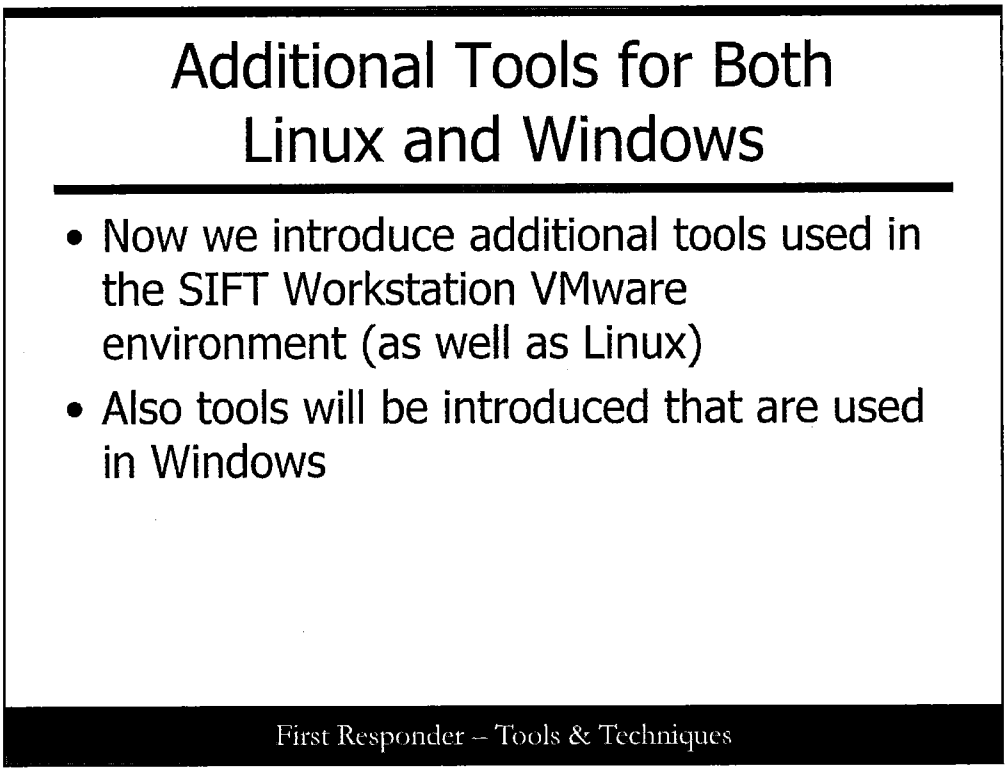


Monitoring

Monitoring is an extremely important part of containment. Using monitoring techniques, you can hopefully tell if the incident has been contained. If you have properly contained the incident, you should not see additional malicious activity; however, keep in mind that a lack of malicious activity does not absolutely mean that the attacker is no longer in the network…the attacker may be sleeping.

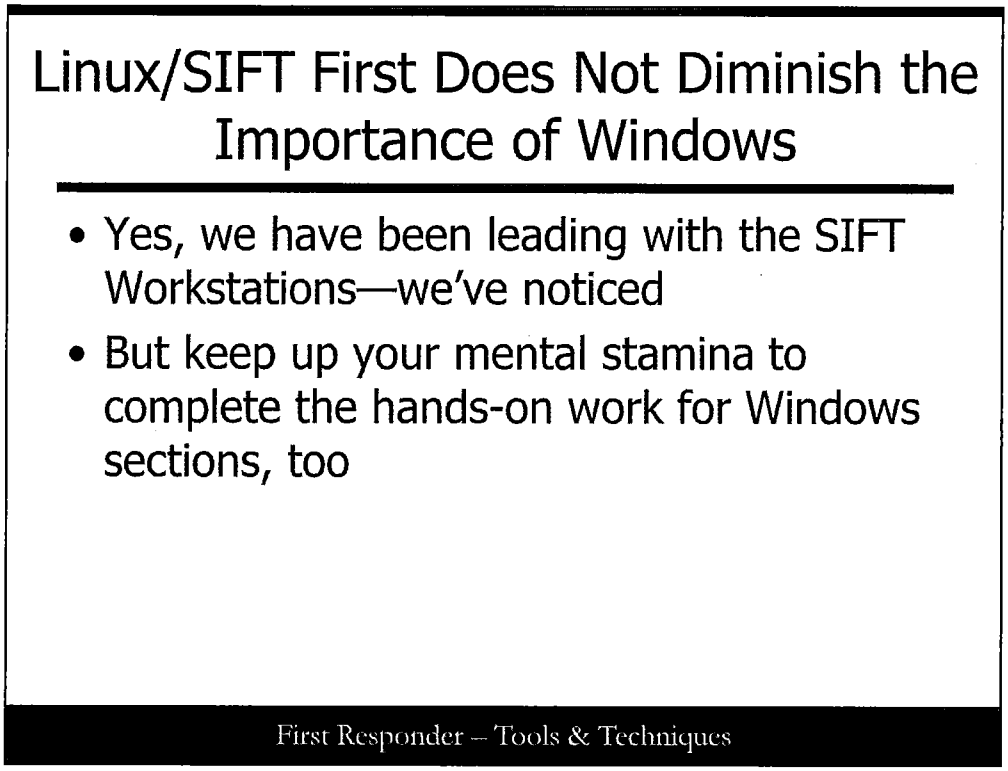


This page intentionally left blank.



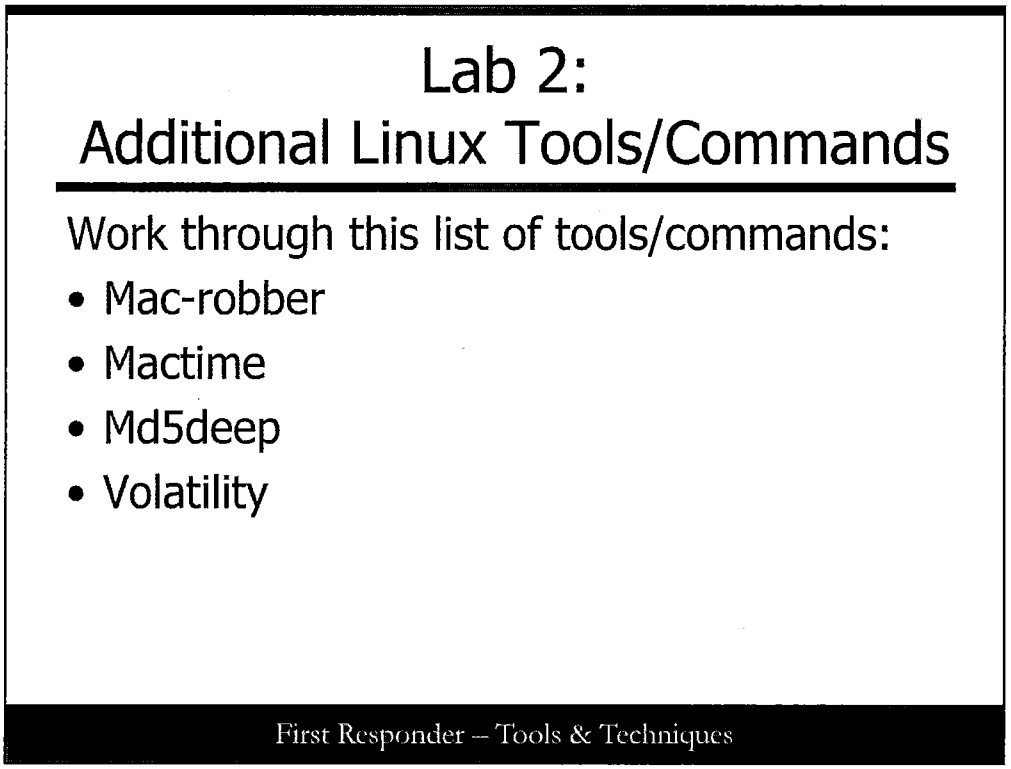
Additional Tools for Both Linux and Windows

In this lab, we perform hands-on demonstrations of additional tools for both the Linux platform under the SIFT Workstation environment, as well as that of the Windows systems, so you can benefit from that perspective.



Linux/SIFT First Does Not Diminish the Importance of Windows

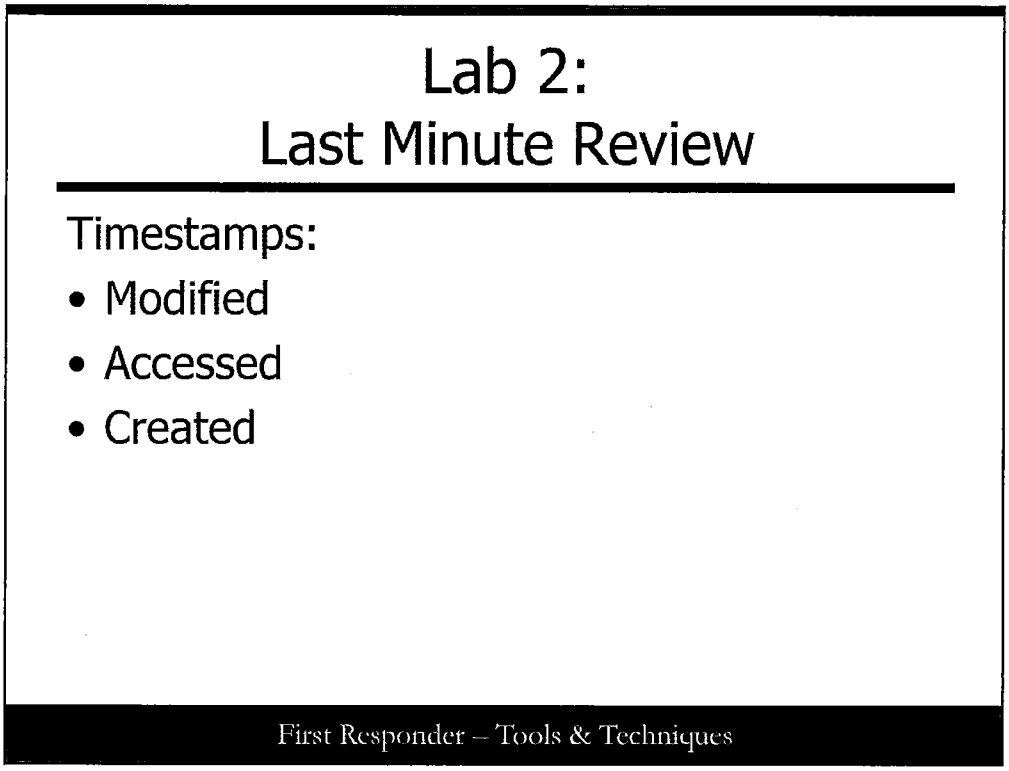
Just because we lead with the SIFT Workstation does not mean that we want you to run out of the room when you are done with that portion of the exercises. The MS Windows portions of the hands-on are important for you to go through. So ensure you can keep you mental stamina up and work the Windows sections of the labs as well so that you put the knowledge to use while you are here or when you resume studying after the course. All this work will come to good in the end.



Lab 2: Additional Linux Tools/Commands

Here is what you will be working on pertaining to Linux using the SIFT Workstation VMware environment. We want you to work through this list of tools/commands:

- Mac-robber
- Mactime
- Md5deep
- Volatility



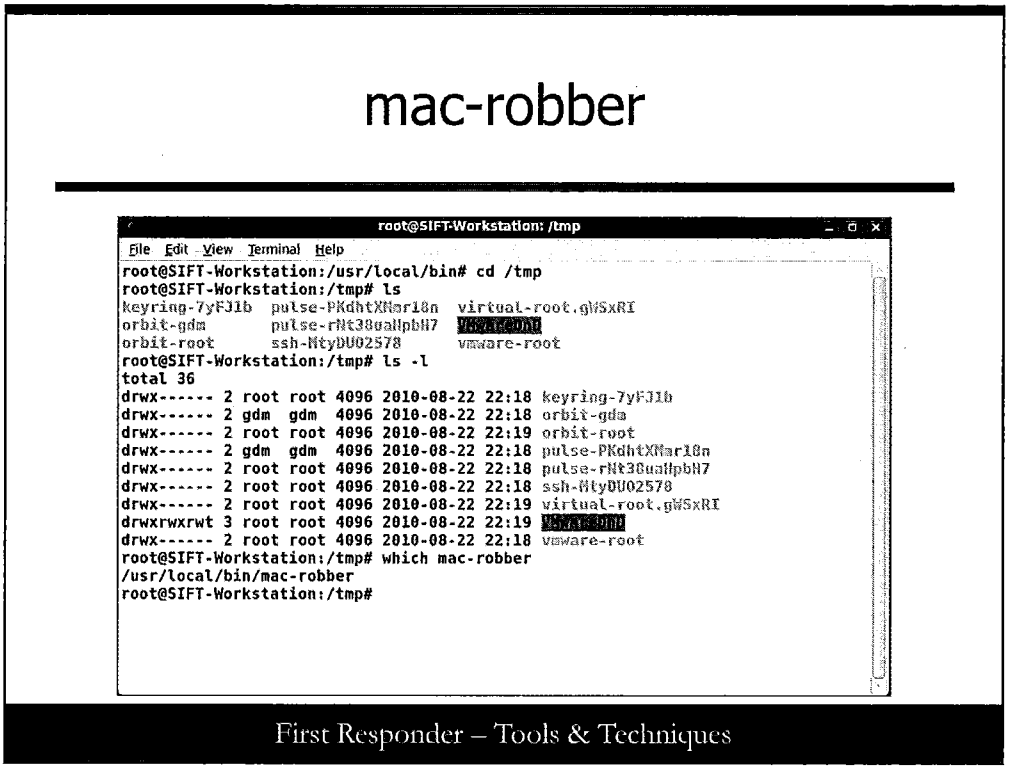
Lab 2: Last Minute Review
Let's dive into the work. Log in to your SIFT station using the root account.
Before you begin your work, start by writing out what you recall or may know of as it relates to file attribute timestamps. In the lines provided here, list out what you know of what times are associated with files and directories.

You should remember that most of the times, files maintain a last Modified, last Accessed, and Created timestamps. Folders have nearly the same but also maintain an Entry Modified attribute to signify when folders get updated with their contents.

But remember, these attributes are largely like boot prints in the mud; you will see only that last one present. So it's a big deal to ensure that the fewest amount and degree of changes are made to a hard drive or storage media as possible. Every effort to minimize the forensic impact and maintain data preservation should be made.

We show you how those timestamps are used in our demonstration on mac-robber and log2timeline.

Issue the following to view the contents of the tmp folder, view the long format of the /tmp folder, and locate the tool mac-robber:
cd /tmp
ls
ls -l



mac-robber

Issue the following to view the contents of the tmp folder, view the long format of the /tmp folder, and locate the tool mac-robber:

cd /tmp

ls

ls –l

which mac-robber

Now that we've verified that mac-robber is part of the SIFT Workstation, let's put it to use by invoking it for the /tmp directory, but we will redirect the output to our home directory.

Note: If you missed the instructions at the beginning of this lab and you did not log in as root, you might end up a bit confused when you go looking for your output file.

$ su - (password is forensics)

cd /root

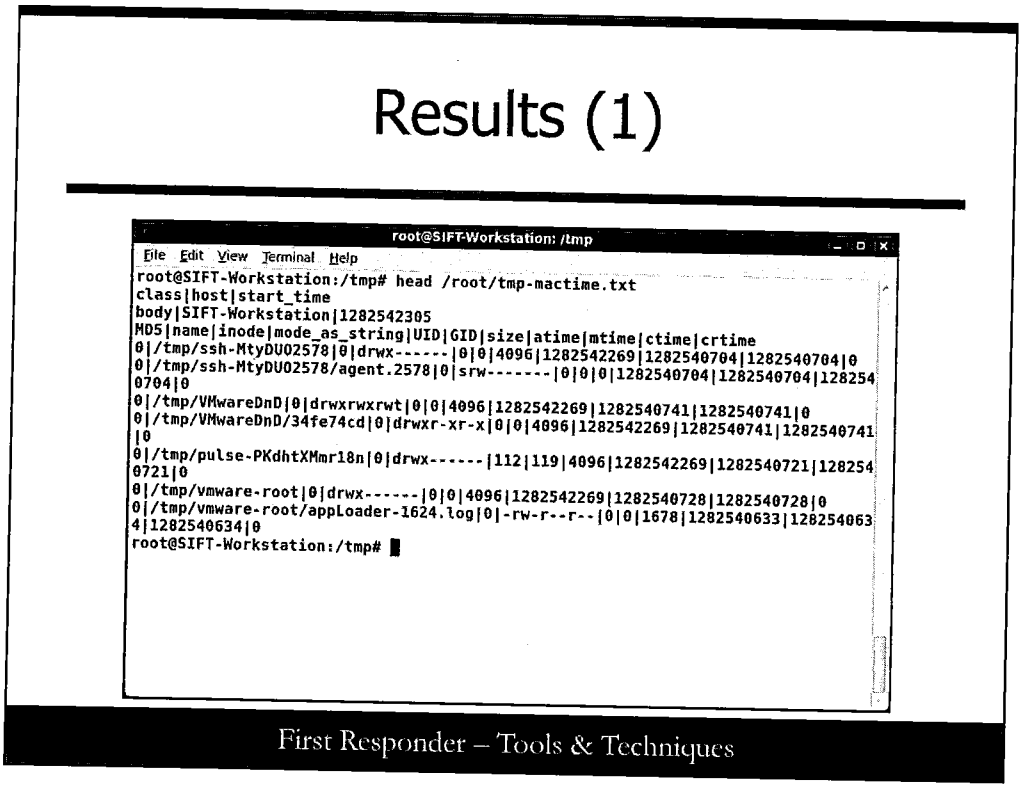
mac-robber /tmp >> /root/tmp-mactime.txt

head /root/tmp-mactime.txt

ls

mactime -b /root/tmp-mactime.txt -z PST8PDT >> /root/tmp-timeline.txt

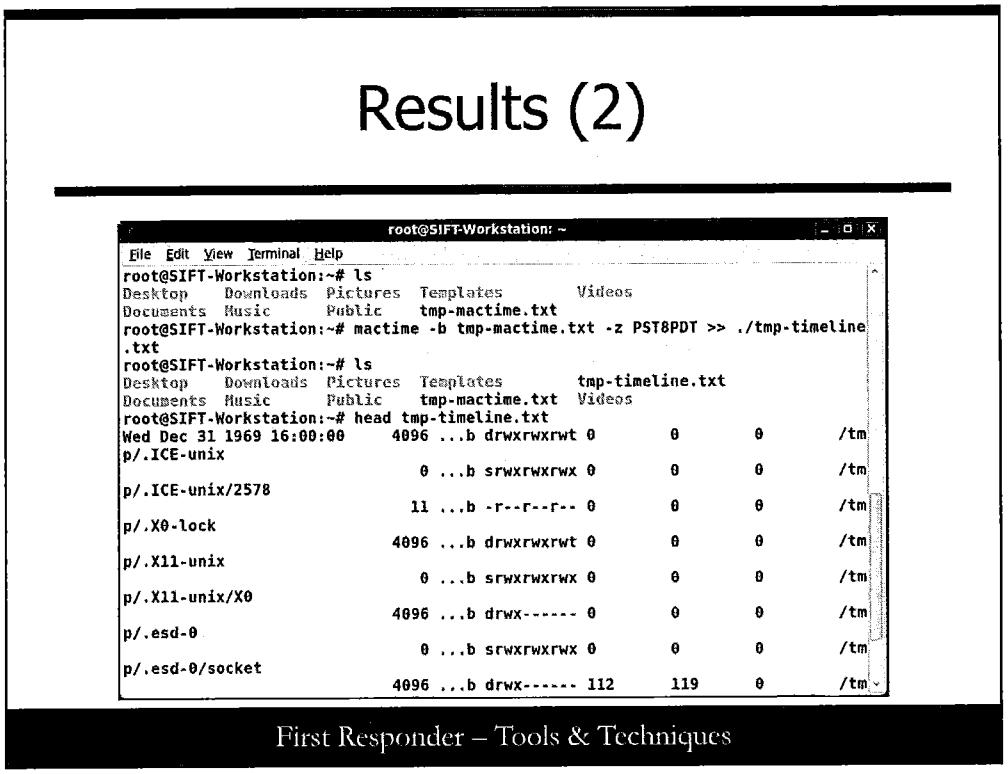
head /root/tmp-timeline.txt



Results (1)

When you look this over, you find that the times don't actually look like a time you are usually familiar with. You see a long number. This is because these timestamps are kept in UNIX Epoch Time form, which is the number of seconds since January 1, 1970, UTC. This method of timekeeping is precise and in our field works well for our purposes.

What we just did was take a folder and its contents and had those written into a format that is parsable. The next step is to sort those events according to the timestamps of Modified, Accessed, and Created.



Results (2)

Enter the tool that gets the file's birthdate info, *mactime*.

Mactime takes files that can be parsed and have date entries in the right format and converts them, which are in a way usable for analysts to review and sort entries.

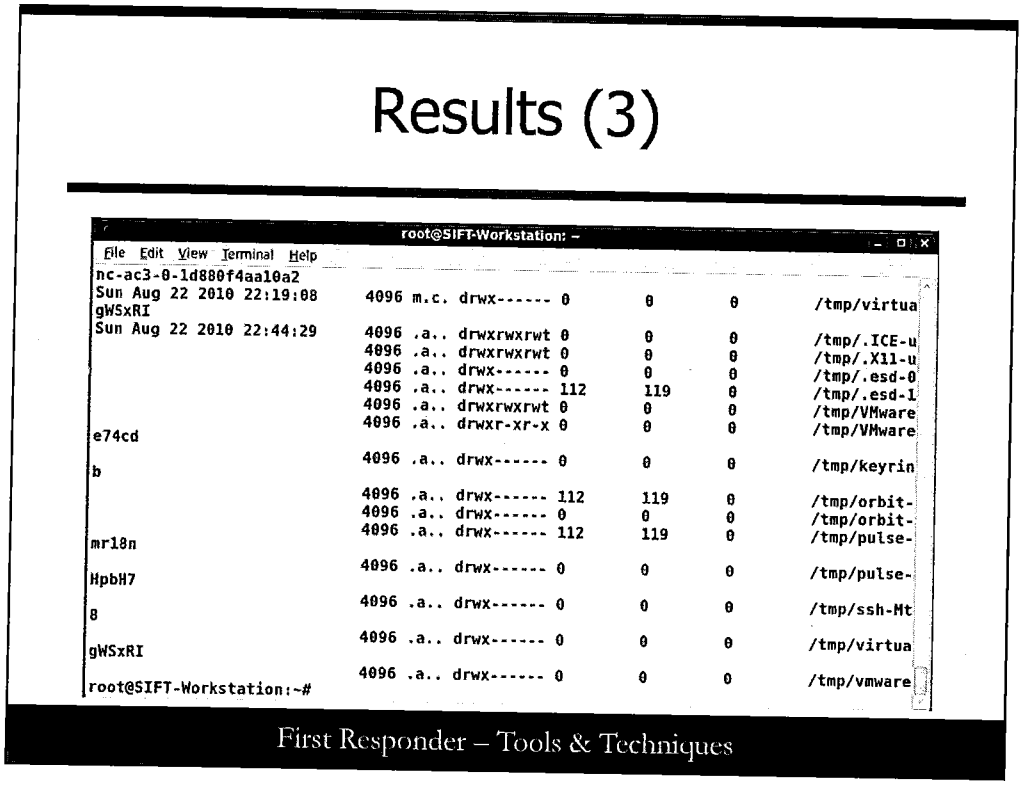
Assuming you are still in your /root/ directory (if not cd to the root directory), issue the following commands, and assuming you want everything in the context of the US West Coast time zone, issue these commands:

ls

mactime –b tmp-mactime.txt –z PST8PDT >> ./tmp-timeline.txt

Then we get the first few lines of the results displayed to the terminal window.

head tmp-timeline.txt



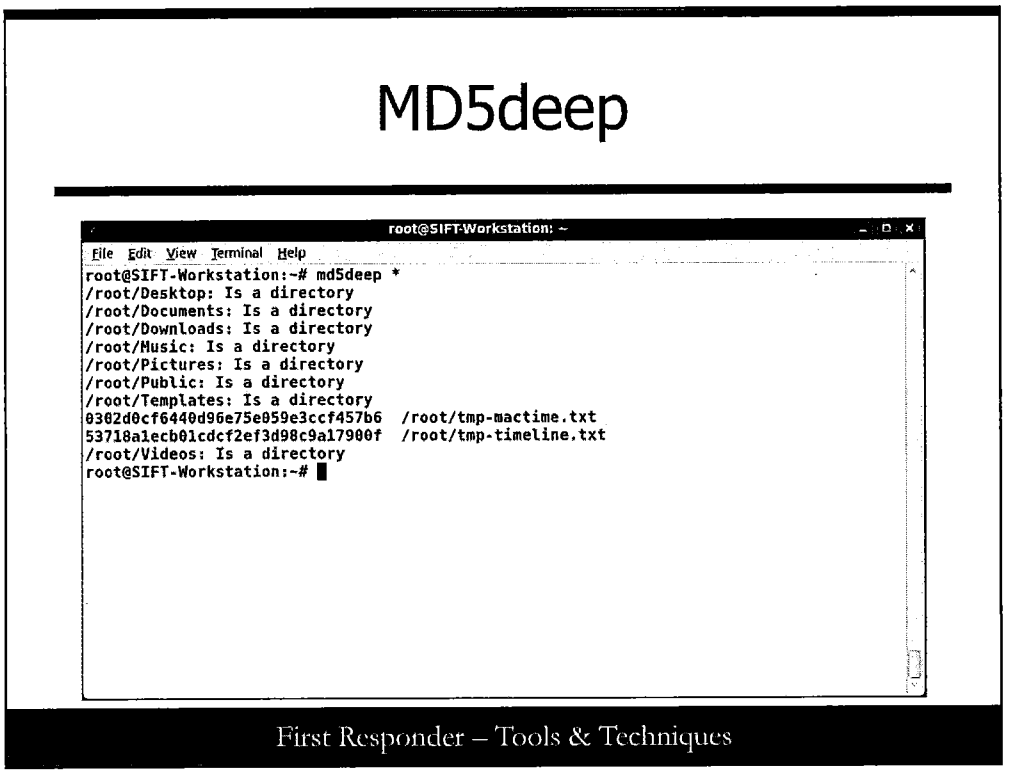
Results (3)

Wow … first events in 1969 … at 4PM (when you do the conversion to a 24-hour clock). But the math checks out and it makes sense. Let's work through how this is explained from what we have so far.

We instructed *mactime* to process the file tmp-mactime.txt by issuing the command : *mactime –b tmp-mactime.txt*. But we needed everything put in terms of the America\Los Angeles (or PST8PDT) time zone, so we include a *–z PST8PDT* in our command.

Everything gets redirected to an output file (tmp-timeline.txt) and we use the *head* command to view the first 20 lines of it.

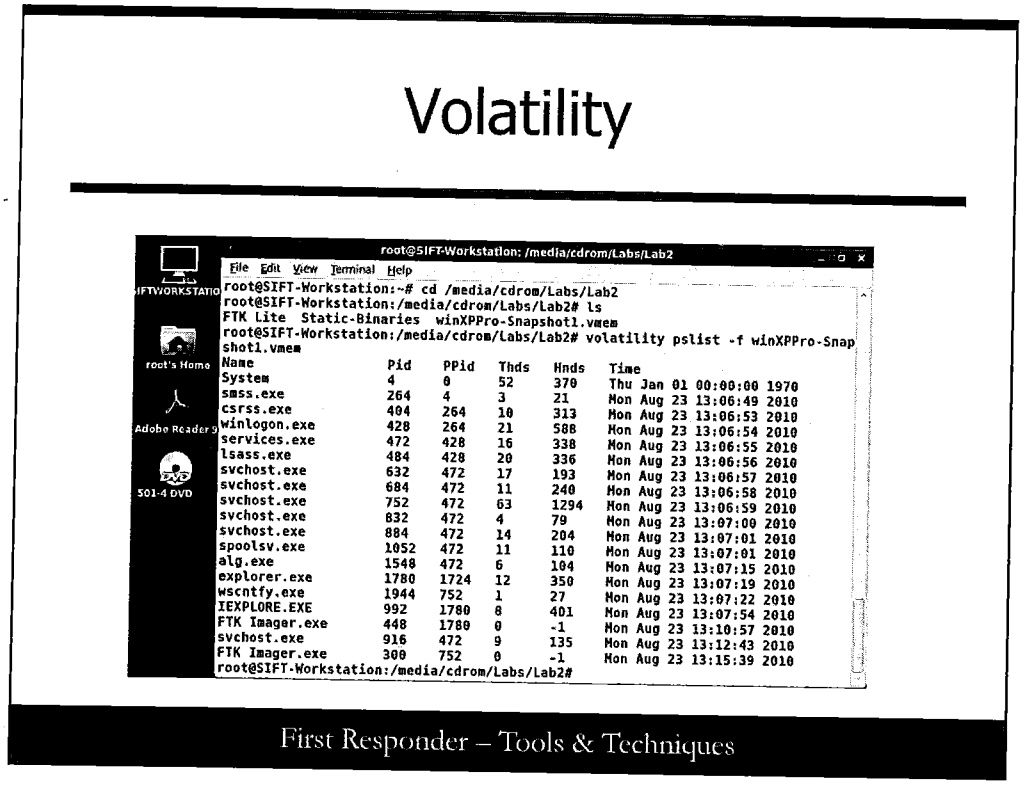
Take the time now to view the file in its entirety. You will find that the last entries in the file have recent timestamps listed at the end of the file as the following illustration depicts. Your experience should be similar, but don't expect the timestamps to be exactly the same as in the picture.



MD5deep

Here, you run the md5deep and you should see similar output. (Note that your fixed-length hashes should appear to be different. The reason is that you got different timestamps when you run these commands in this lab, so you should have different MD5 hashed output.)

md5deep /root/*



Volatility
The last bit in this section is a chance to explore the Volatility memory analysis tool suite.

Volatility is one of the open-source projects that can do memory forensics on memory images captured during live response. Memory is fantastic to grab and a valuable forensic resource because it's about the only thing that shows you the state of the system's mind at the time of a forensic response. Few things get in as close in showing an analyst what was going on at a single moment in time than the system's memory.

It's now time to insert your SANS 501 Course DVD into the CD/DVD-ROM drive. You should expect the Places icon near the top of the screen to enable you to see the DVD disk mounted. (If not there, you should see the DVD mounted on the SIFT Workstation desktop workspace as shown in the screen shot.) We use a memory image in the Labs\Lab2 folder and have Volatility examine it for interesting items.

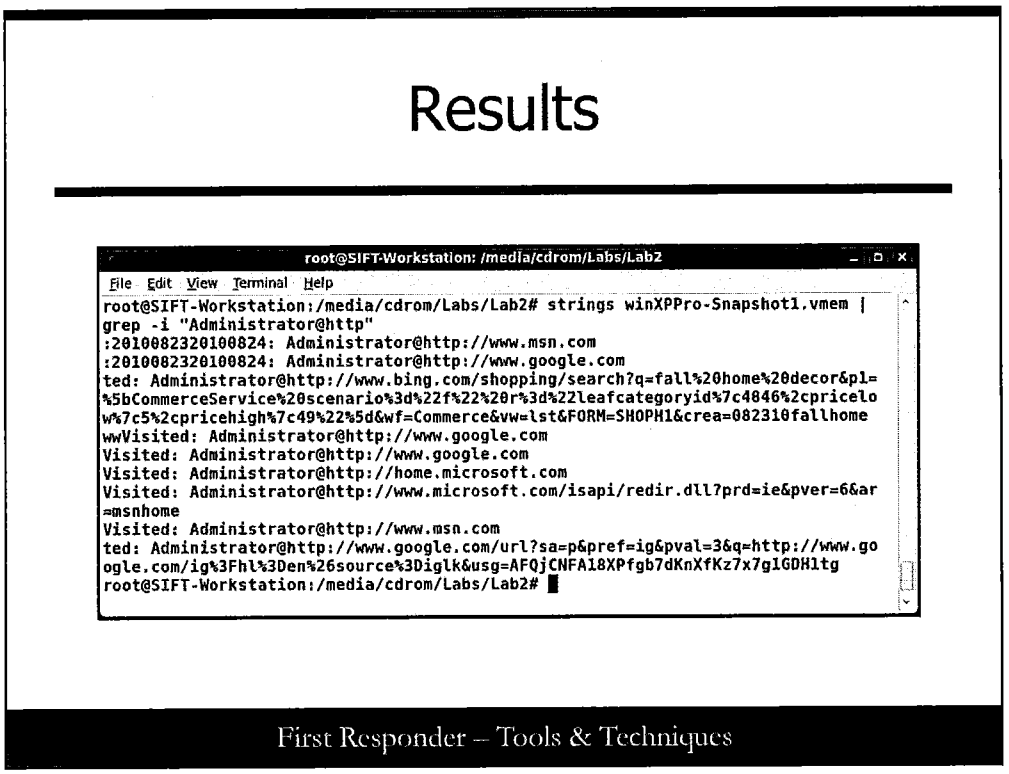
cd /media/cdrom/Labs/Lab2
ls
volatility pslist –f winXPPro-Snapshot1.vmem

As a result, you get a listing of the processes running on the system the moment that this particular virtual machine (VMware) memory snapshot was taken. (Notice that it has a designation of .vmem.)

Some things to notice are that IEXPLORER.EXE running as PID 992 and FTK Imager.exe running as processes 448 and 300. So, could that mean there might be some browsing history of interest on this particular system? Maybe. You just have to go and gather it to find out.

Note: Depending on the version of SIFT, you should use vol.py instead of Volatility:

vol.py pslist -f winXPPro-Snapshot1.vmem



Results

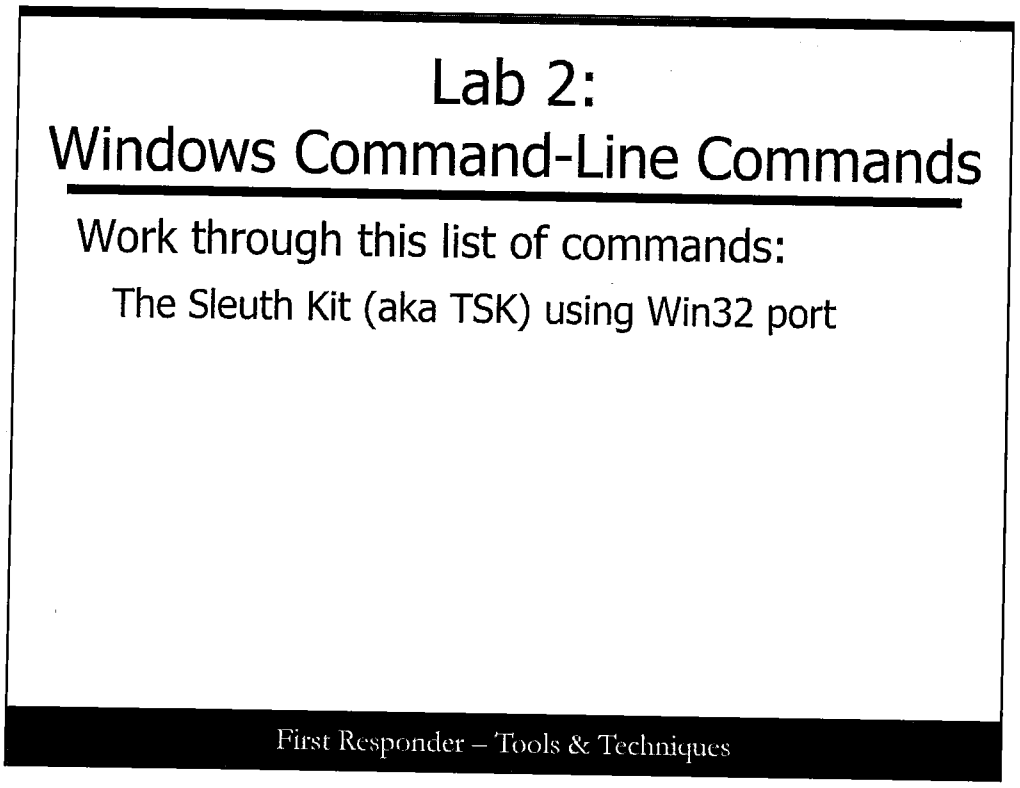
However, don't forget about simple Linux tools such as strings and *grep*. Try this on for size:

strings winXPPro-Snapshot1.vmem | grep –i "Administrator@http"

What we did was direct strings to look at all ASCII printable strings in the memory dump and then piped the output to the grep tool. Grep then used a case-insensitive search (-i) to filter the output to anything that had Administrator@http in the line of strings. Certainly we can see that the web browser was used at some time. But we don't see when one link was visited before another. We need another artifact to track down that sort of history of activity.

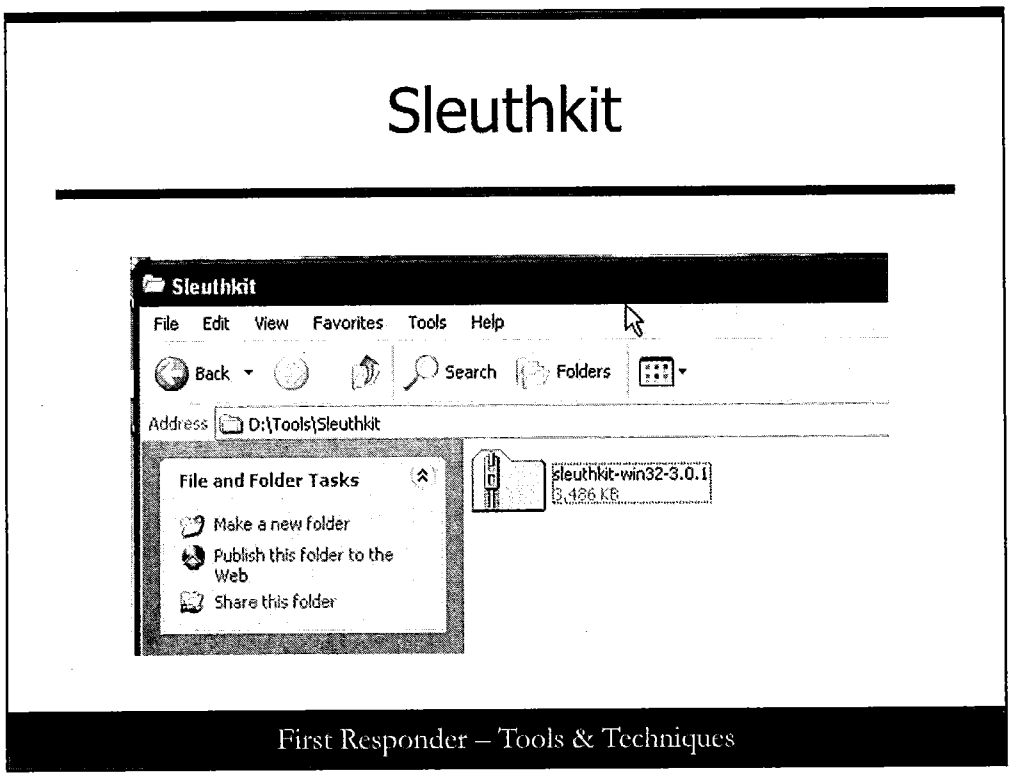
How do we get memory? The answer to that will be to switch from the SIFT workstation environment and switch over to an available Windows workstation. Keep in mind we work on covering Sleuthkit (Win32 port) topics in this part of the lab.

In the upcoming section, we start by demonstrating how to capture the memory contents from within a Windows environment and save them to the hard disk drive or some other location of your choosing. You can then later copy them to a form of removable media or use a file transfer method (such as SCP or netcat) to move the dumpfile from the Windows system and onto an analysis platform.



Lab 2: Windows Command-line Commands

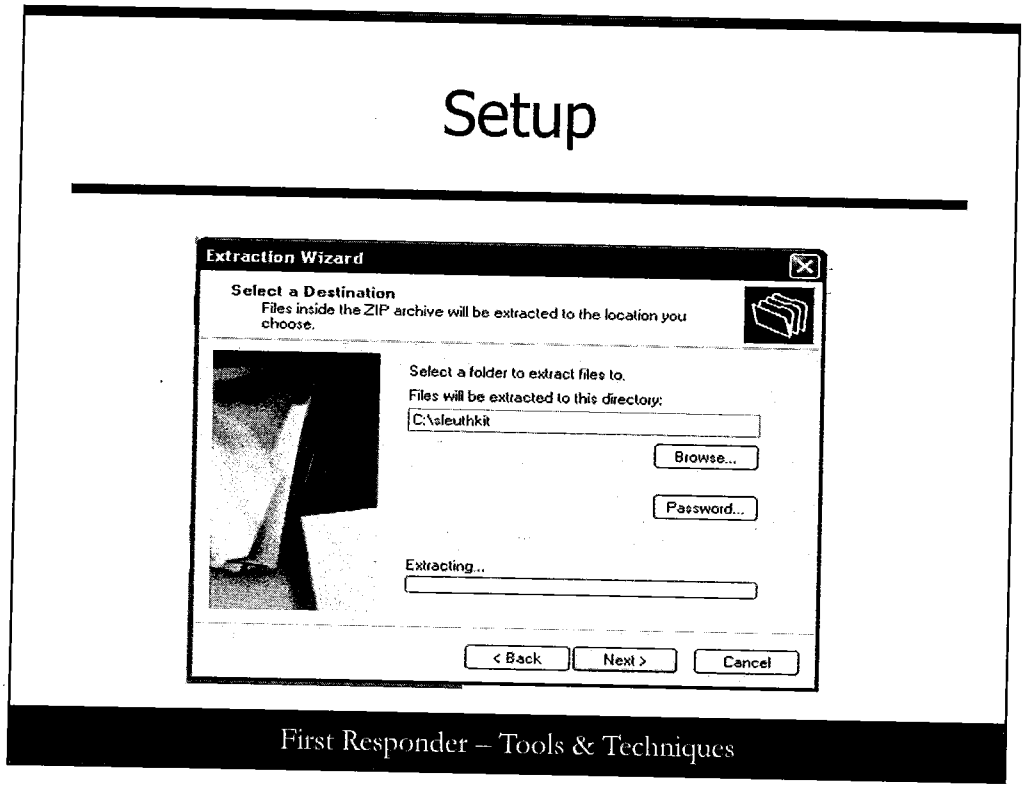
When it is time for you to work on the Windows tools, here is what you need to work through: Sleuthkit (Win32 port), which is also-known-as The Sleuth Kit (TSK).



Sleuthkit

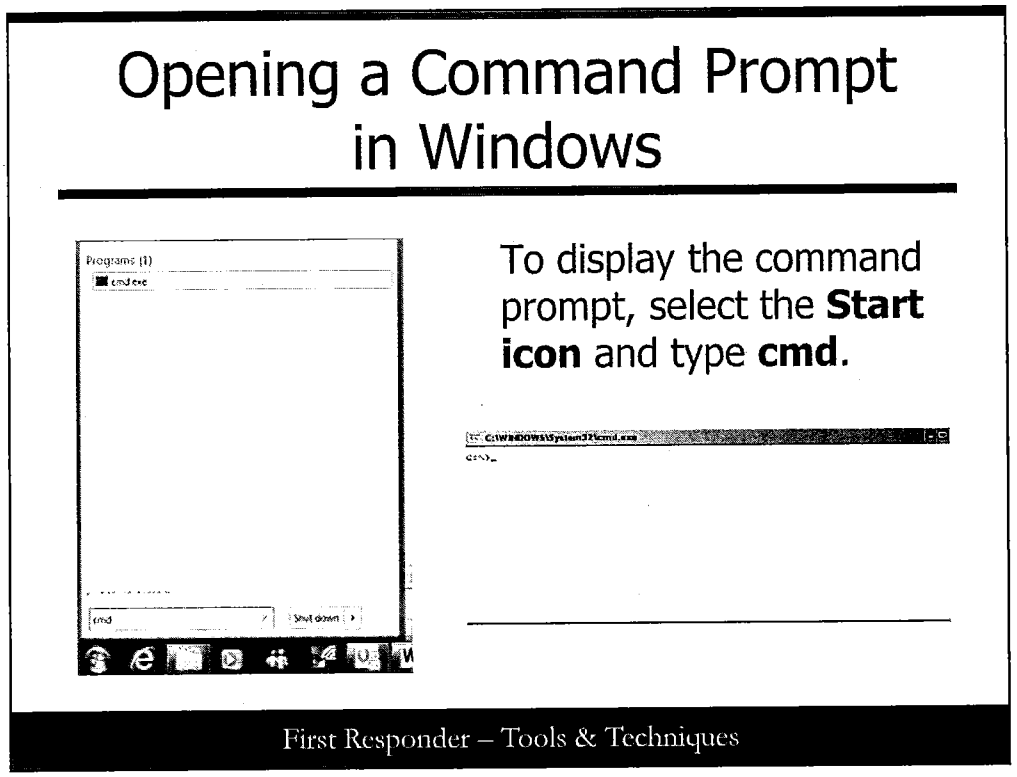
Next, unpack the Sleuthkit suite of tools and perform a basic test. Take note that we are using version 3.0.1 of the Windows-port of Sleuthkit.

Navigate to the CD/DVD-ROM drive again, and this time go to the Tools\Sleuthkit folder. Unzip the contents of this folder, but this time, put them in the root of our main hard disk drive (for example, *C:\sleuthkit\sleuthkit-win32-3.0.1*). Begin the extraction.



Setup

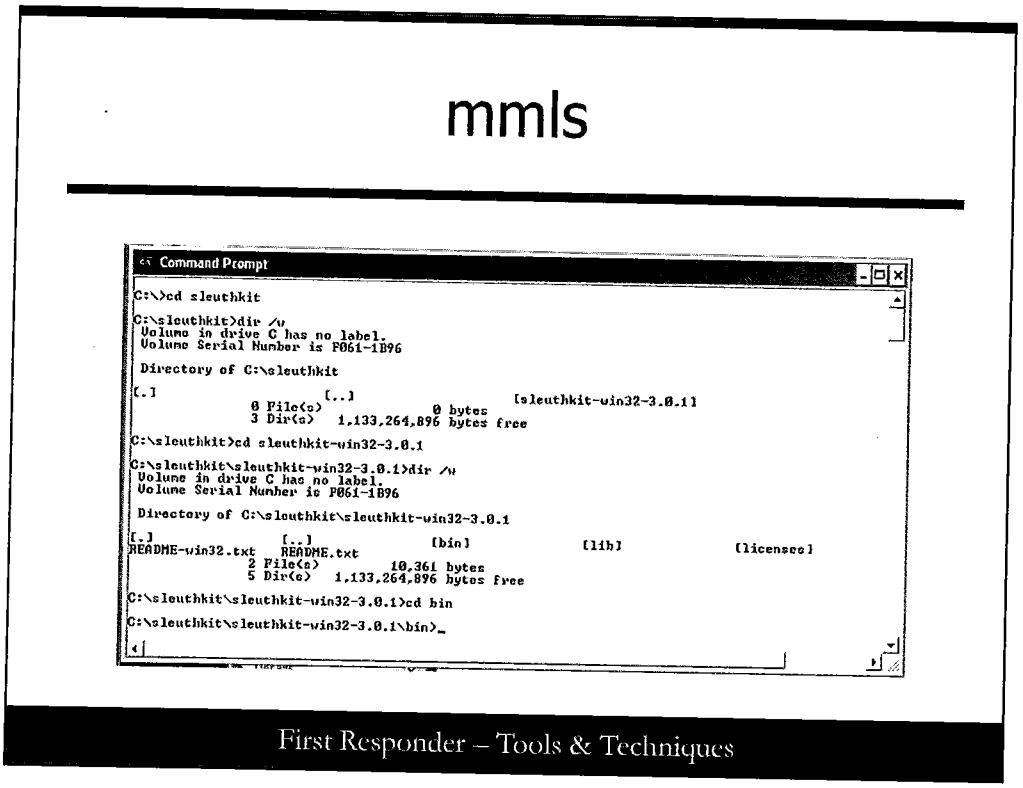
When you click the next button at the bottom of the window, the task starts and puts the contents of the archive into our designated folder.



Opening a Command Prompt in Windows

When it's done, open up a command prompt in Windows. To display the command prompt, select the **Start icon** and type **cmd**.

We spend the rest of the time of this exercise in there.

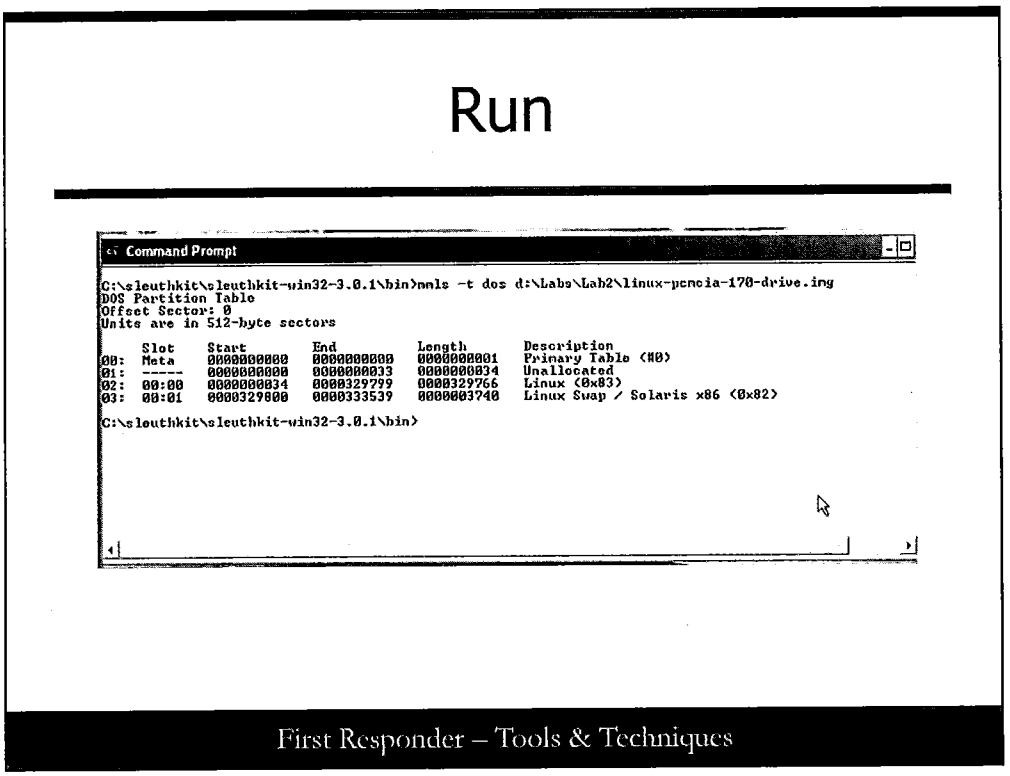


mmls

When you open up the command prompt and navigate to the folder you extracted the files to, you find some subfolders. Change directories into the *C:\Sleuthkit\sleuthkit-win32-3.0.1\bin folder* to continue.

The next thing is to examine a Linux Hard disk image. Yes, we meant what we said when we said Linux. A bit-stream image was taken of a system having a variant of a Knoppix install. The drive was a PCMCIA 170 Mb drive (small by today's standards) and such a small size just allows us to bring more content like this for you to hone your abilities.

The image is located in the Lab folder of your course DVD. Refer to it in the command line as *<DRIVELETTER>:\Labs\Lab2\linux-pcmcia-170-drive.img* where <DRIVELETTER> represents the drive letter for your CD/DVD-ROM drive.



Run

mmls

The next step in the plan is to get a partition listing of the image and print it out on the screen. We use the mmls tool to accomplish this. Enter in the following commands:

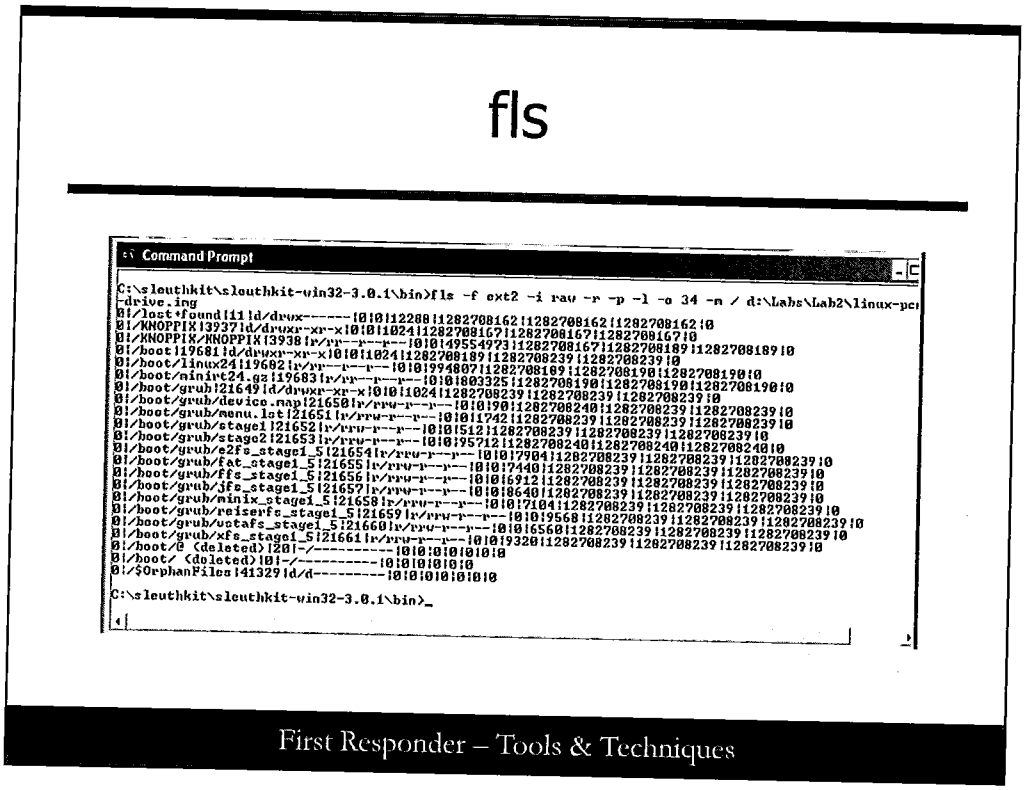
mmls.exe –t dos <DRIVELETTER>:\Labs\Lab2\linux-pcmcia-170-drive.img

(Note: Ensure you substituted in your CD/DVD-ROM drive's drive letter for <DRIVELETTER> or the previous command will not work.)

We specified the –t switch to tell this tool that the partition was a dos partition. Mmls supports a great number of partition types, so you have to give it some input on how it's about to process.

Now you should notice two important details about the output from mmls as it pertains to the Linux drive image. The first discovery is there are two, quite useable and interesting partitions on this image that was recovered from the actual drive. Those are the actual data partition (Linux) and the other is a virtual memory partition, (which Linux calls swap because pages of memory are swapped in and out as needed, so it's called Linux Swap). This means you may harvest both persistent file artifacts from the first partition and maybe also grab contents from the swap partition as memory!

The second detail to notice is the start of the partitions and their lengths are listed. Yes, it may mean some math calculations, but it's helpful for either you or another analyst to be fully aware of how the drive is laid out in respect to data volumes.



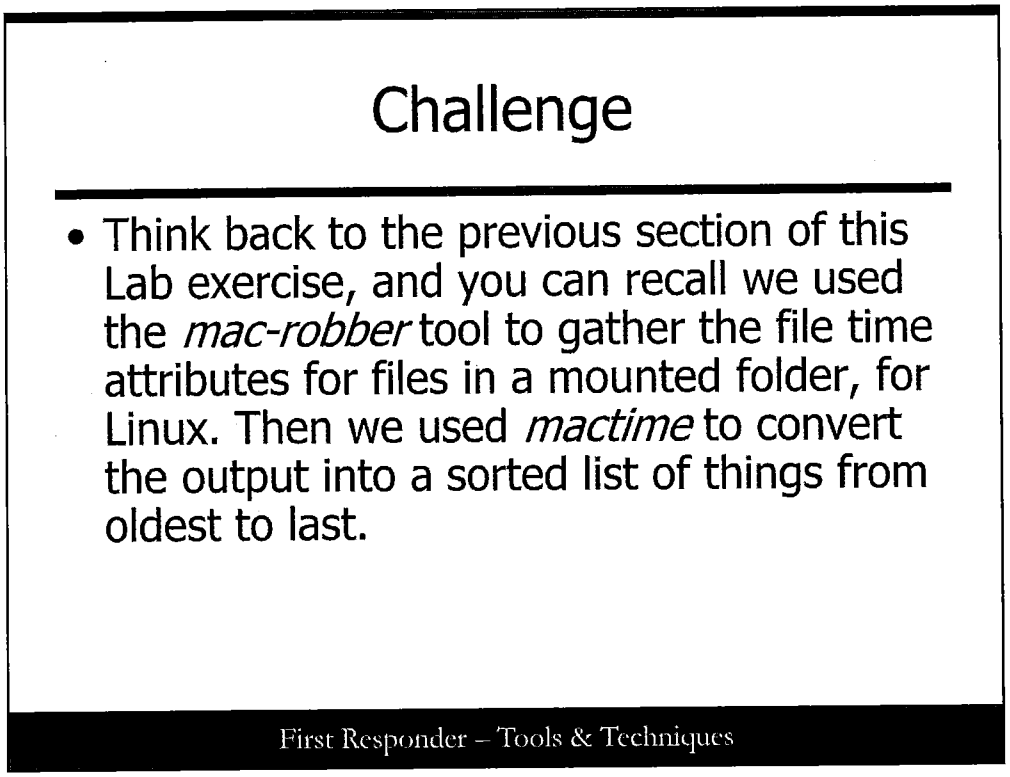
fls

Next we have our last little challenge to perform. We use the *fls* tool to get the MAC-times for all the files and folders in the images and assemble a list of these in what is referred to as mactime-compatible format. This data is critical in many ways for performing timeline of analysis.

A word of warning: There will be a lot of switches, so we explain their use in this course book. But don't limit your understanding of the switches to just us; get out the Sleuthkit site and read a bit of the inner workings and features. You may wind up surprised. Here is the command we want you to enter into your command prompt window:

fls -f ext2 -i raw -r -p -l -o 34 -m / d:\Labs\Lab2\linux-pcmcia-170-drive.img

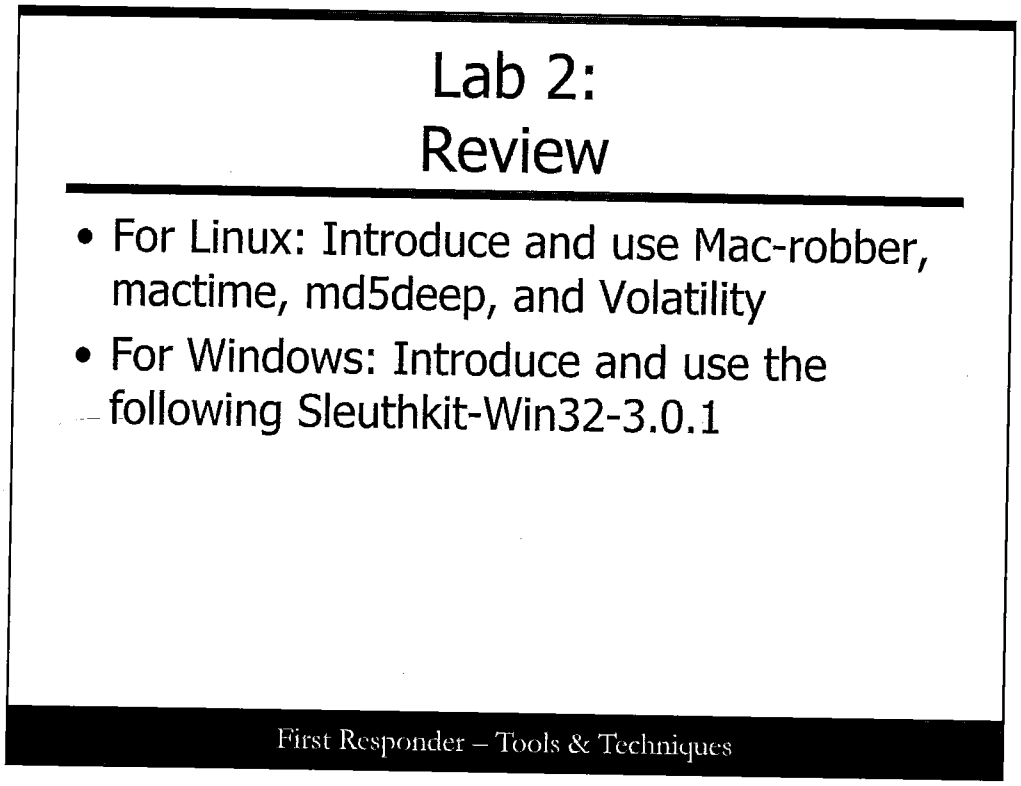
What we created here was a listing to standard output (STNDOUT) of the Modified, Accessed, and Created timestamps of the file system. We've told the fls command to expect a Linux ext2 file system and that the image was going to be raw, thus the –f ext2 and –i raw switches were used, respectively. In addition we told fls to recurse down subfolders/directories while getting the full path of each entry it found; hence the use of the –r and –p switches, in that order. Also we got the listing in the mactime format and designated the mount-point of the file system with a / by using –m / to accomplish this setting. Lastly, we have to tell the fls tool that the file system starts on sector offset # 34, as per the partition list data that came directly from our running of the mmls tool a while ago. When we designated the filename and pressed the Enter key on the keyboard, fls delivered a listing to the screen.



Challenge
Your challenge now is to write out in the upcoming blank lines what you think the process would have to be to take this data we've acquired on this Windows system and make it available to the Linux tools we discussed for a detailed analysis. Just imagine that you as an incident responder on the scene have collected this information using your acquisition tool set built on Windows and you've taken this information back to the lab. The analyst there has access to a SIFT Workstation but could use your coaching and technical expertise to get this data into a timeline for review.

There are multiple right answers that accomplish this. Compare notes with other attendees nearby to refine your technique or possibly learn innovative ways to accomplish this objective.
This completes the lab work for this section.

Note: After this course and in your copious free time, we encourage you to look at timeline and time-sorting tools that are available. One in particular that stands out is called Log2Timeline and is included on your SIFT Workstation image. It can process a large number of input and output formats making it versatile. (Imagine sorting Apache web logs saved in CEF-format with file artifacts in the form we just worked on to find when a user web-request 200 code corresponded with the access to a file! Do you think that would be a big help to you? Possibly so.)

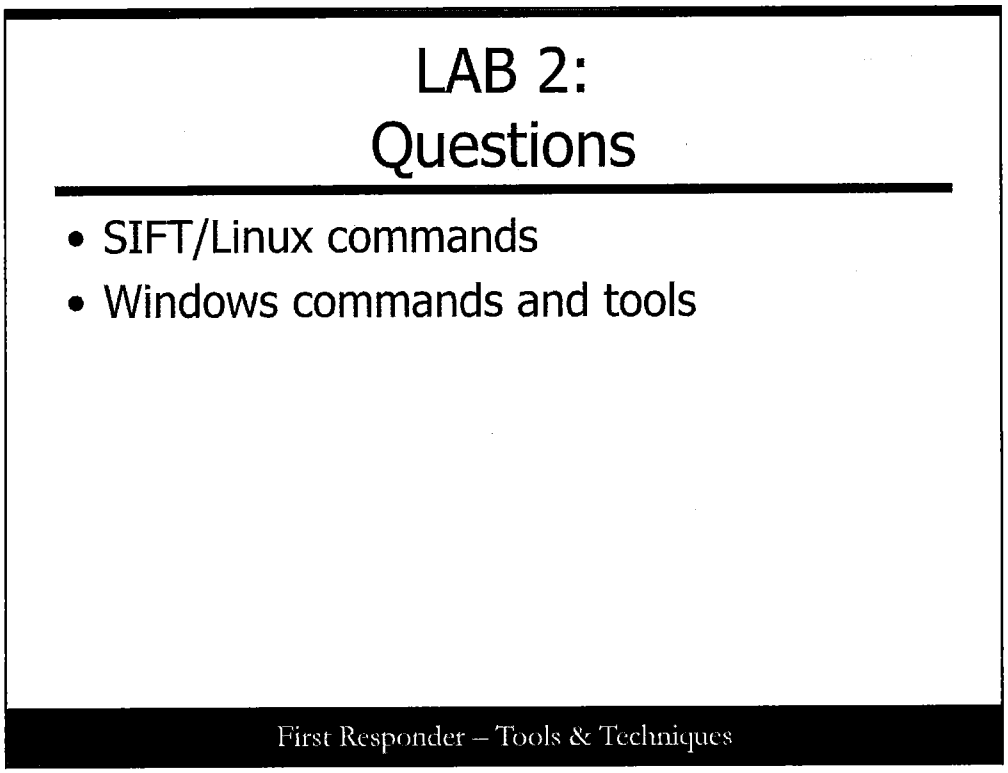


Lab 2: Review

As a review we want to get you introduced and put fingers-on-keyboards to actually use several sets of tools related to the incident response work and computer forensic study. When it comes to the Linux side of the house, using the SIFT Workstation you should have been able to work on getting timestamp artifacts preserved with mac-robber.

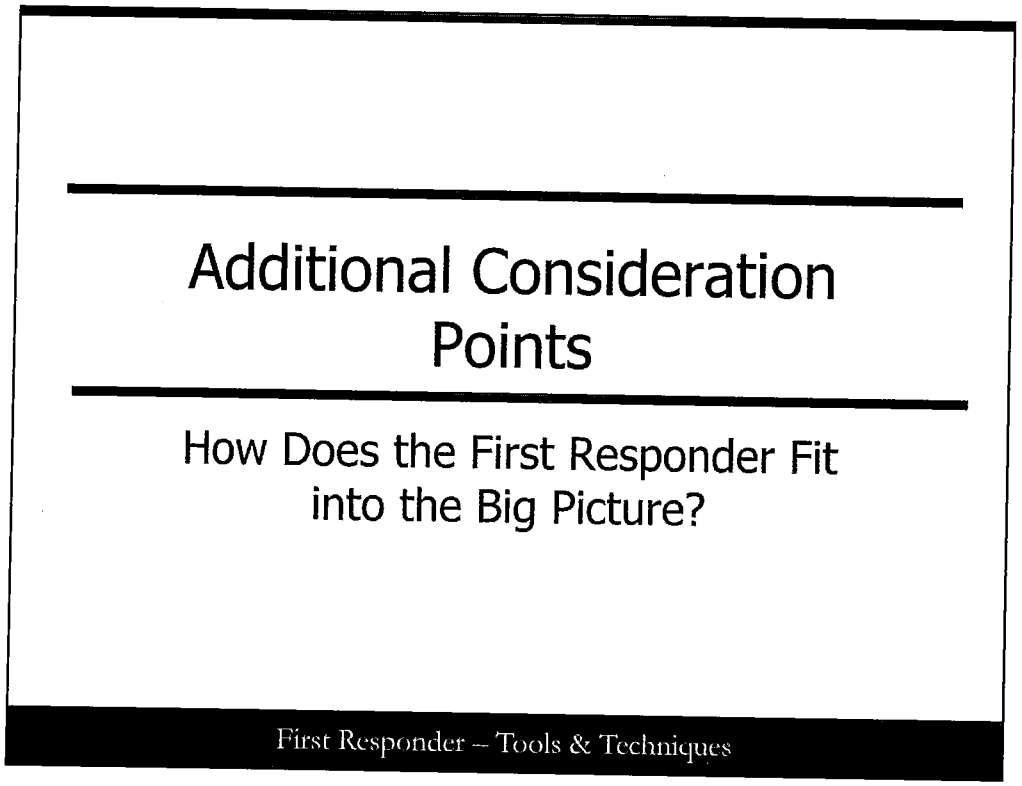
Display timelines with mactime, compute fixed-length hash values using the MD5 algorithm in md5deep, and inspect the running processes in a Windows memory snapshot using the Volatility suite of tools.

Carrying on with that we shifted over to the Windows platform side and got you acquainted with select tools from Brian Carrier's Sleuthkit suite of tools.



LAB 2: Questions

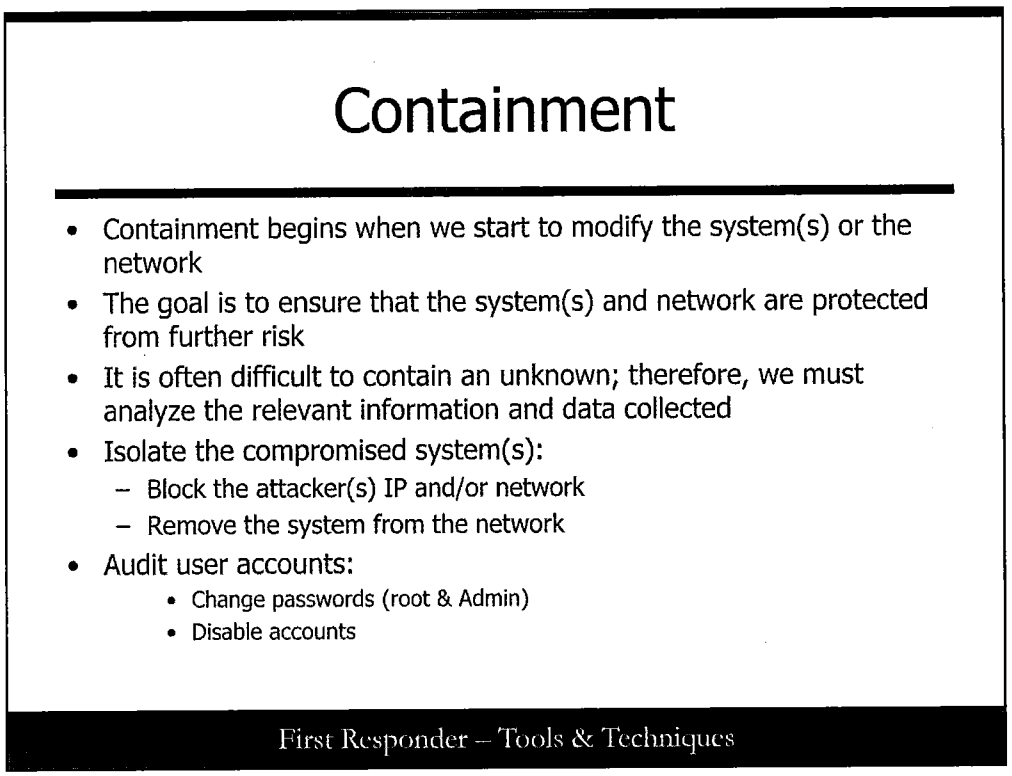
Now if there are any questions, we can open up a brief opportunity to ask questions pertaining to the SIFT Workstation and the tools we used in this block of exercises and move, later, into a few questions, if any, for the Windows tools we used.



Additional Consideration Points

Outcome Statement

The student will understand additional considerations with respect to the © SANS Six Step Incident Handling Methodology to help the student understand where they fit in to the bigger picture and why certain actions are recommended in the First Responder Methodology.



Containment

In the Containment phase we begin to make some changes to the network and the system in response to the issues identified in the previous phase. These changes are designed to protect the environment and the user, but we haven't gone far enough in getting rid of the source. Normally, you would mobilize or send a pair of incident handlers to the scene (if possible). In the case of a first responder, you may be on your own for a while, or you may be working with dedicated system administrators or information security professionals over the phone, or getting ready for their arrival. In either case, because the incident has already occurred, you want to work through diagnosis (and treatment) in a calm, disciplined manner and take notes along the way.

One of the business risk issues is removing application functionality from the user. First responders and incident handlers need to understand what a given computer system is used for, and through that understanding a decision can be made regarding taking the computer offline for a period of time. Also, it is important to understand that management may decide to go straight into containment without preserving host data or capturing any network data, depending on the criticality of the system, even if there is a crime involved. It is the job of the first responder to explain the implications of such actions (that is, not preserving data) but not to argue with management or counsel.

Actions taken should be justifiable and reasonable depending on the nature of the incident. Some examples include

Removing the system from the network or suspending power to the system

Segmenting or segregating the system from other critical assets to minimize risk while continuing to observe the event.

Changing passwords, DNS records, or firewall Access Control Lists (ACL) to block the attacker.

Adding to the current level of monitoring including adding an IDS/IPS device or conducting full packet capturing.

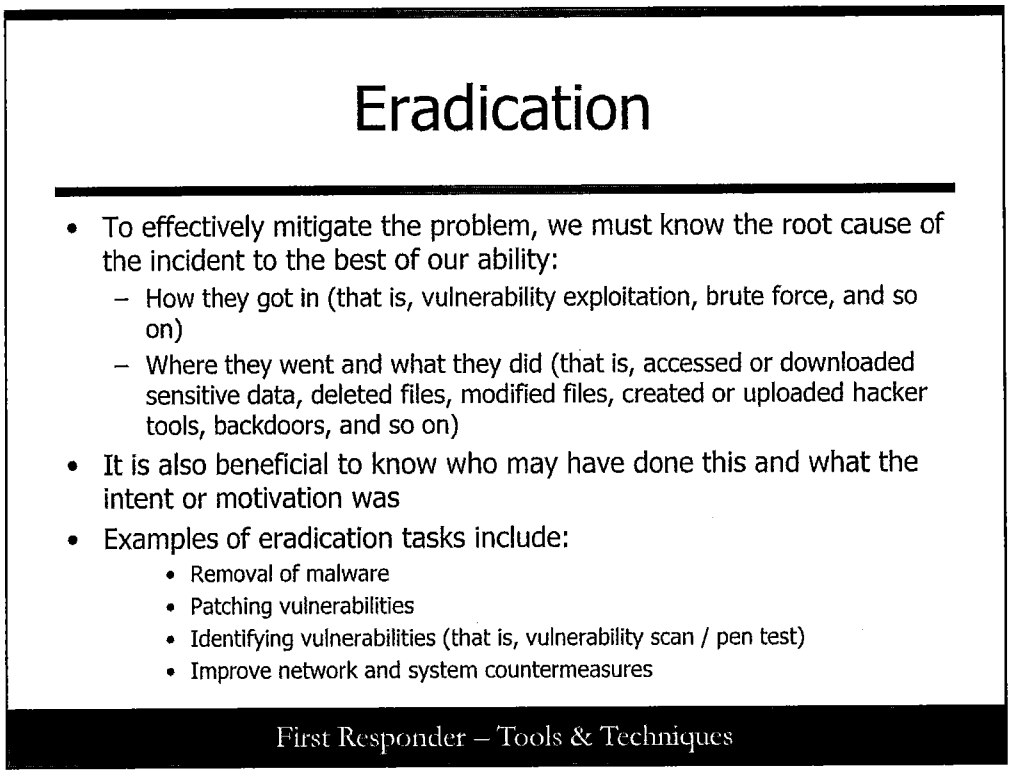
Adjusting or turning up the current level of auditing and logging capabilities.

Collecting a sample of malicious code for analysis through reverse engineering or submission to an Antivirus tool or vendor.

During containment, we must consider risk when making decisions on how to proceed with incident response. When systems on or adjacent to a network involved in collecting, processing, storing, or transmitting sensitive information are suspected to have been compromised, the risks associated with the intrusion are higher and are directly proportional to the classification level. Furthermore, if operations have been affected, the desire to restore normal operations as soon as possible can have a significant impact on the ability to collect relevant evidence for investigation. Therefore, it is important to consider that continuity of operations may heavily depend on your ability to analyze an incident and not necessarily just the ability to restore from backup tapes. If the attack or automated malicious code is not properly analyzed, the event could easily reoccur or may actually remain dormant within the environment on systems that were not identified as being involved or affected.

There are general steps that can be performed when containing an incident. First, blocking the attacking network range or individual IP address at the site's perimeter is a rudimentary technique that is often circumvented due to the nature of dynamic DNS; however, it is still something to consider. Second, is to remove the system from the network, but the first responder should be aware that certain malware might self-destruct or begin to delete files if it is network-aware, so it is important to utilize any intelligence that may be available prior to taking any action. Third, if there is any reason to suspect that accounts have compromised passwords, then a decision needs to be made with respect to changing the passwords. Finally, you should consider coordinating with the upstream ISP, CERT organizations, and Managed Security Service Providers (if applicable) so that the IP addresses involved can be checked against other incident databases.

Keep in mind that there is generally additional containment steps that need to be determined based on the circumstances of the incident.



Eradication

To effectively eradicate, we must know the root cause and the extent of the compromise. This can be determined only through proper analysis and hopefully we know this by the time we reach this step. Eradication primarily refers to the ability to ensure there is no residual risk left after the compromise.

Eradication is typically not a first responder role; however, there are circumstances in which a first responder may be asked to participate in these activities. In the Eradication phase we want to actually remove the affecting malware and malicious tools or otherwise correct the weaknesses identified.

Often during eradication, a system or certain files will be restored from a backup tape. It is imperative that the files being restored are known to be good. Also, keep in mind that simply restoring known good files does not mean that there are no unknown or malicious files on the system. When conducting eradication procedures, it may be necessary to completely rebuild the system. For example, if you suspect that there is a root kit on the system, you might as well rebuild because the time to remediate and the possible risk of not being 100% sure that the system doesn't have some remnants may be the better alternative. In addition, it is highly likely that there are other systems that have some trust with the suspect/compromised system; these relationships should be reviewed.

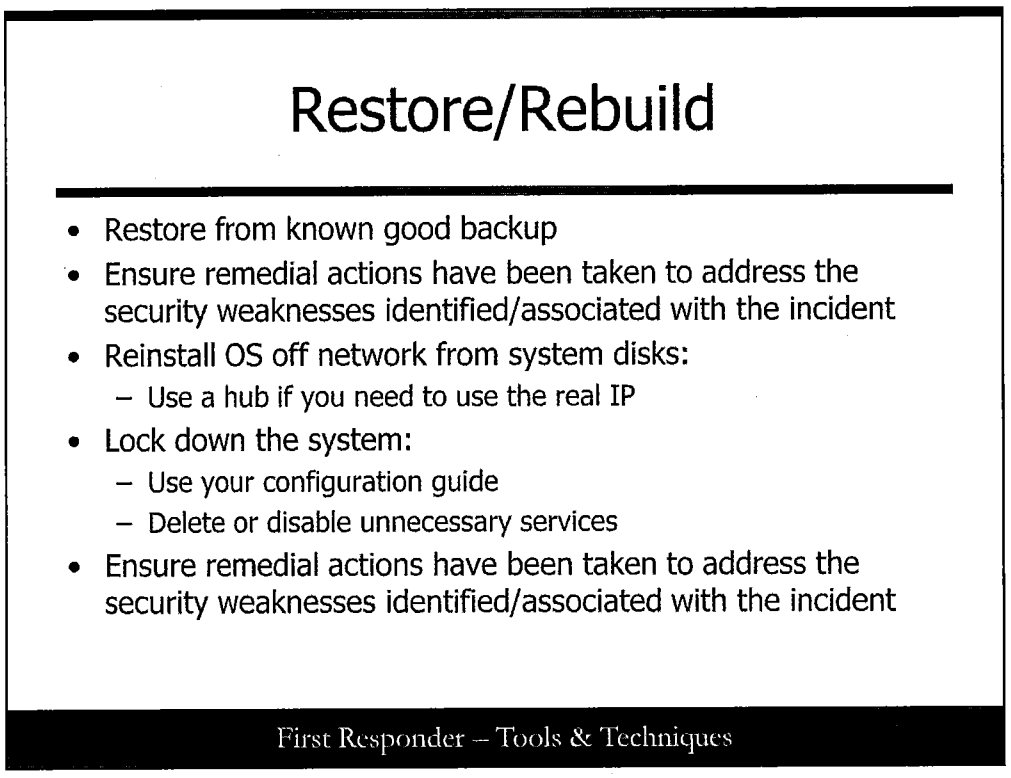
The latest trends indicate that cybercriminals are motivated primarily by financial/economic gain. Although this makes quite a bit of sense, it is not the only motivation for conducting illegal activities online. Depending on the type of incident and the attack vector used (Worm, Botnet, Backdoor, DDOS, or Active Attacker), an attorney may show knowledge and intent.

These are two critical elements of fraud that should be considered when gathering evidence during an investigation. Although gathering all potential evidence may increase the time and effort to complete the investigation, if the goal is to criminally prosecute the perpetrator, it is necessary to look at all evidence, both incriminating and exculpatory. Often an investigation uncovers the identity of an attacker or the responsible parties. If the investigation is done properly and the identity of the perpetrator is known, an organization may collect damages from the perpetrator or the organization that the perpetrator works for. If an organization conducts an investigation with this goal in mind, it will typically increase the amount of time and effort required to get the job done.

Either way, to effectively complete this step, we should have reasonable confidence that the system(s) and network are free from any malicious code or vulnerabilities associated with the incident because part of eradication is to ensure that the vulnerability exploited is no longer an issue. Some questions that you might ask yourself are:

- What was the root cause?
- How did the attacker or malware get in?
- What is the underlying weakness that needs to be fixed?
- What is the nature and extent of the compromise?
- Was this malware or an active attacker?
- Are you sure you know about all unauthorized activity?
- Can you confidently remove or repair the damage?
- Can you back up and/or restore critical data that is needed for operations?
- How far back can or should you restore data?

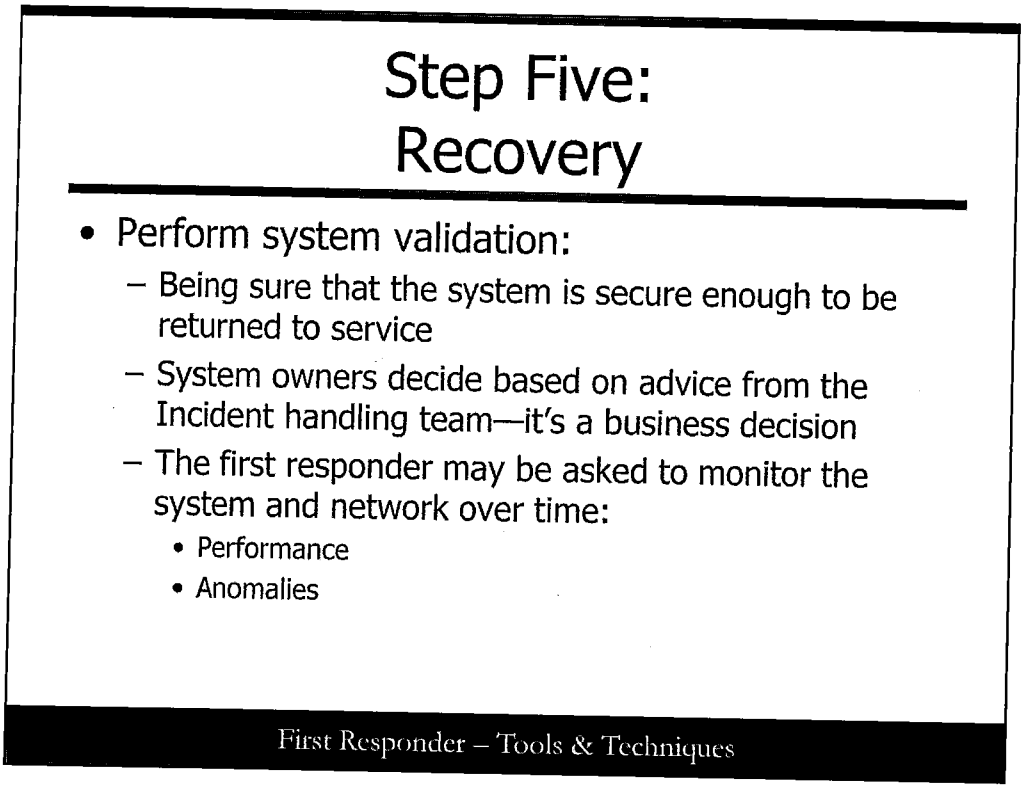
These questions are all important components of the ever-challenging "rebuild or repair" decision. On the rebuild side, there will be a higher degree of trust that the system has integrity, at the cost of the labor to rebuild, and the opportunity costs lost in spending the time on a rebuild. On the repair side there will be a lower degree of trust, but the system can continue operating with minimal down time.



Restore/Rebuild

Restoration is obviously a great way to eradicate malware and hacker artifacts from your environment, provided you are certain that the backup to be used is known to be good. It is often easier to restore than to rebuild, but the choice should be made based on what makes sense. For example, sometimes a combination of rebuilding the OS and restoring the applications and data is the best choice.

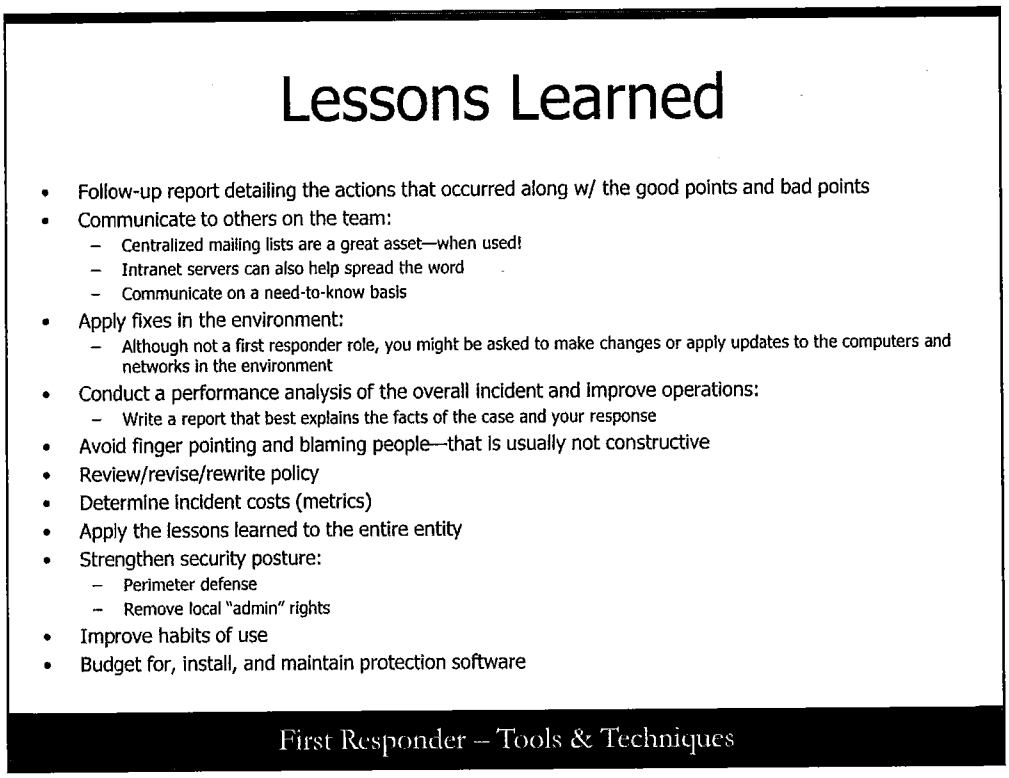
The slide presents many of the common steps involved in rebuilding a system. There are numerous step-by-step guides on the Web and a variety of other resources available at the SANS and GIAC websites.



Recovery

The principle task in the recovery phase is putting a system back into service with a high-level of confidence. That level is likely to vary by individual or company. What's important is that you are comfortable with the decision and that the system provides the business or academic function that it was serving.

Part of recovering from an incident is gaining confidence through continual monitoring over some period of time. Monitoring during recovery is an absolute must and continued monitoring can help ensure that the incident has been contained and that the recovery efforts have been successful (that is, validation).



Lessons Learned

One important concept that has been stressed so far is applying the lessons learned to the rest of the environment. The first responder and the incident handling team need to meet a few days after the incident to assess performance and improve as an incident response team. Here, it is important not to blame people, rather identify the issues and be sure that you have identified the right issue. If there is any blame to be levied, leave it to management to do because first responders should not be involved in HR-type issues.

Also, remember that first responders should want to become stronger and more skilled. It is usually worth taking the time to write down the details of the event and circulate that to others in the organization that may have a need to know because this usually serves to strengthen the skill base of the organization. The attackers are getting smarter, so we need to as well, but keep in mind that the knowledge you have pertaining to the incident is extremely sensitive and that any leak of a known or suspected compromise might mark the end of your time as a first responder.

A post mortem analysis is a sufficiently detailed report that documents the *facts* of the case. Using the report, the organization can conduct a "lessons learned" meeting and determine how the site can better respond in the future and better strengthen system security.

During the lessons learned process, it's wise to avoid finger pointing and blaming people; leave this task to managers and supervisors who are tasked with job performance. Many system administrators are not always aware of everything on a system. Also, with the hectic pace that many organizations set for their staff with the pressure to get systems up, it's easy to miss an important step along the way. Lastly, many organizations do not apply patches and updates immediately, as they have, on occasion, proven to make the system unstable in some unforeseen way even though they deal with security issues.

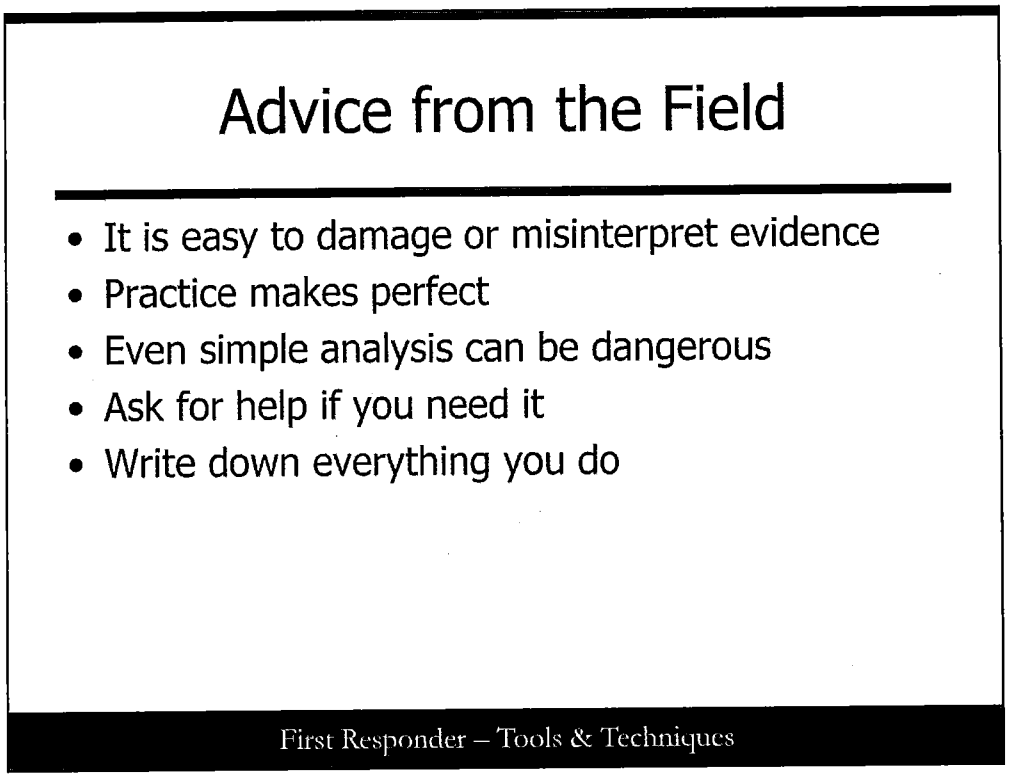
One site in the author's experience had a policy that it did not install updates to open source software for 30 days upon release. In this case, a Trojan was placed in the installation script of Sendmail version 8. The Trojan installer was discovered on the 28th day, and the site was saved installing this malicious code.

Policy may need to be refined based on the incident. One of the critical mistakes made when revising policies after an incident is writing a reactionary policy.

One of the benefits that can come from an incident is greater awareness of the cost of an incident. Many factors are involved in determining costs. Start with down time, lost time from the incident, project delays, lost revenue, and potential impact on company image.

Many of these points have been discussed here in some way. The primary task is to let others know and to assess your overall security response, hopefully improving the state of practice at your site:

- **Check other computers**: Nearby systems may have the same malware or the same security weakness that allowed them to be misused. This might prompt some revision of site policy.
- **Strengthen perimeter defense**: If you don't need the ports open that were used, then close them!
- **Remove local root and sudo rights**: This often gets people into trouble more than it helps them. Also, you should review the sudo configuration file and make sure that it reports when users attempt to use commands they are not authorized to use.

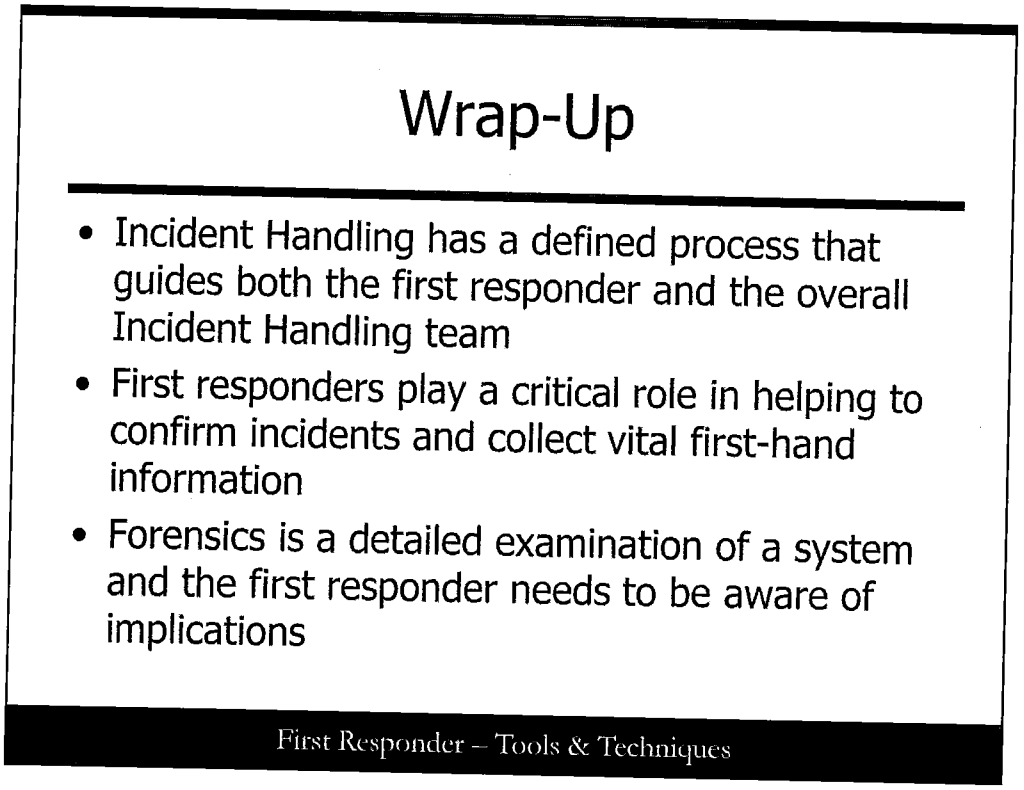


Advice from the Field

When working on a case, it is easy to overwrite evidence of the activity that has occurred. A variety of tools can damage the MAC times in the file system. Another challenge is destroying process and network information for a logged in user, when switching context between the user on the system when you arrive and a supervisory account that has more rights on the system.

Based on the complexity of a system, it is even easier to misinterpret data as you gather data. Rather than relying on a checklist of commands, it is much better if you can collect data in an automated fashion with a repeatable process. However, keep in mind that you must be confident that the tools used are validated so that you know what to expect in the results and so that you do not cause damage to the evidence or the operational environment. Practicing with the tools and testing them prior to using them in incident response is imperative.

You should always ask for help if you are unsure of the situation. Others may know what should or shouldn't be on a system, and it is often best to examine a system with a counterpart. The biggest mistake made by people working in the field is not writing down the steps that they perform, the information they collect, and most important when a given piece of information is collected.

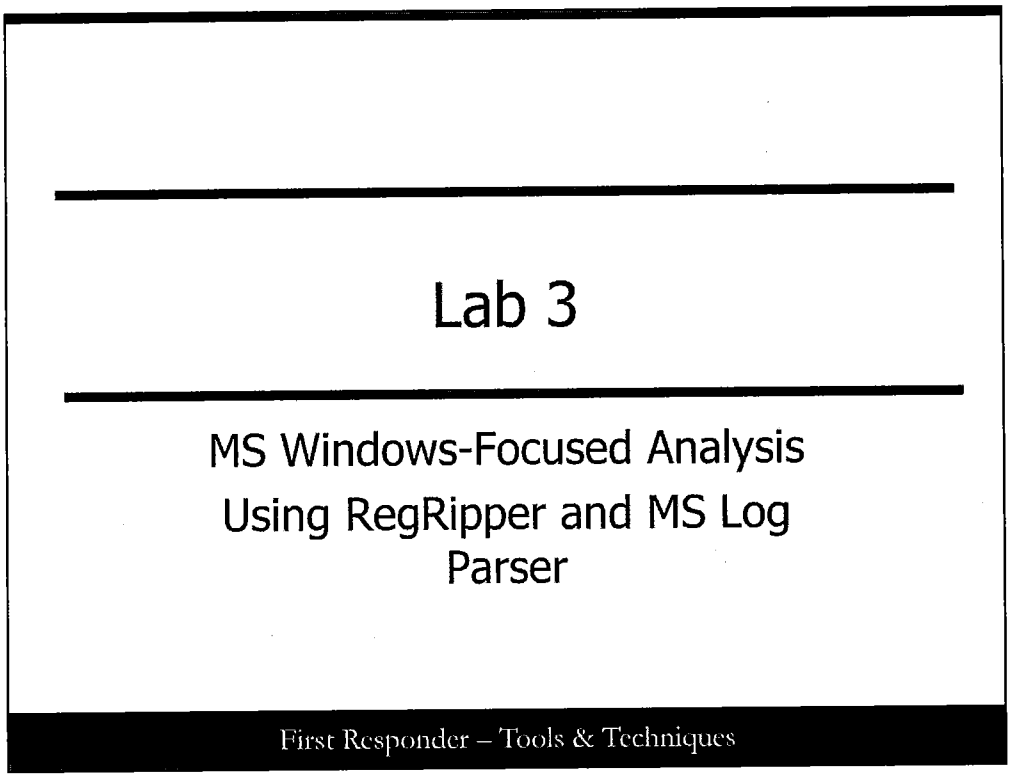


Wrap-Up

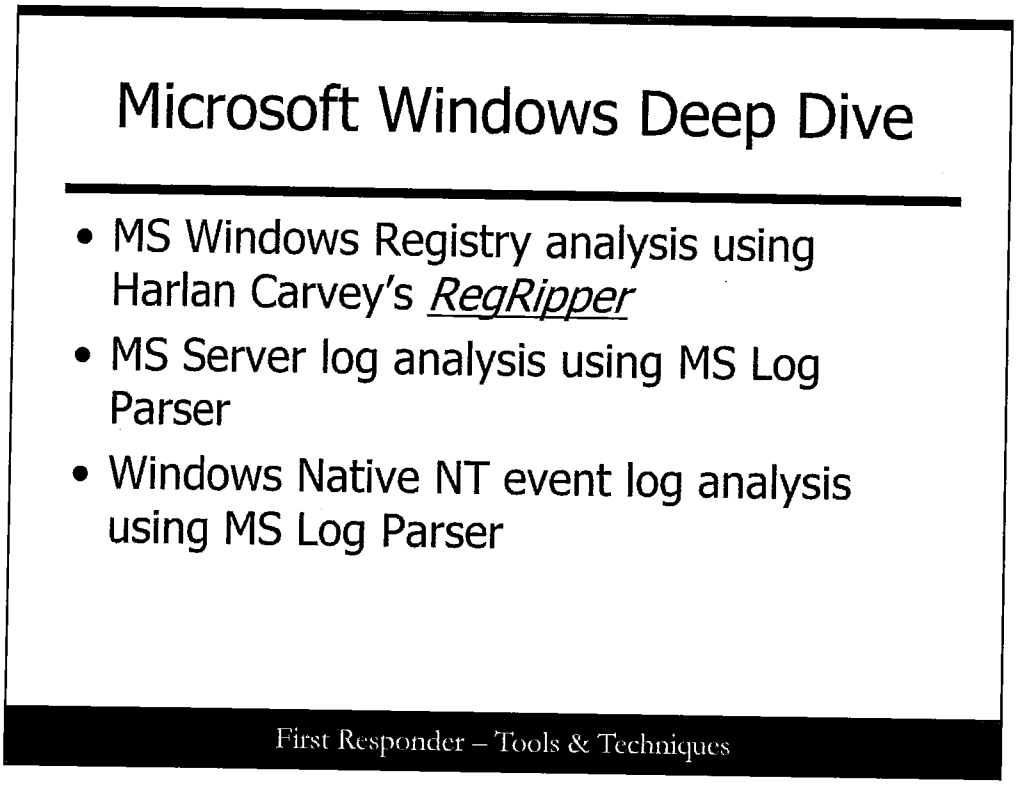
Now that we have been through the theory behind incident response and digital forensics, let' s wrap up with the following key points:

- When you follow the Six Step process to handle an incident, you have both a structure and guidance on what comes next.
- Following a process helps everyone on the team know what they are doing.
- The Six Step process is repeatable. Distinct phases describe and define the actions of the first responder.
- Collecting an initial set of information is a vital skill; by collecting good data early, we can hopefully shorten the overall process.

Forensics is much more than we have talked about here, particularly in the reporting phase. First responders need to be aware of the digital crime scene and need to document and minimize their impact on the system to protect the scene.



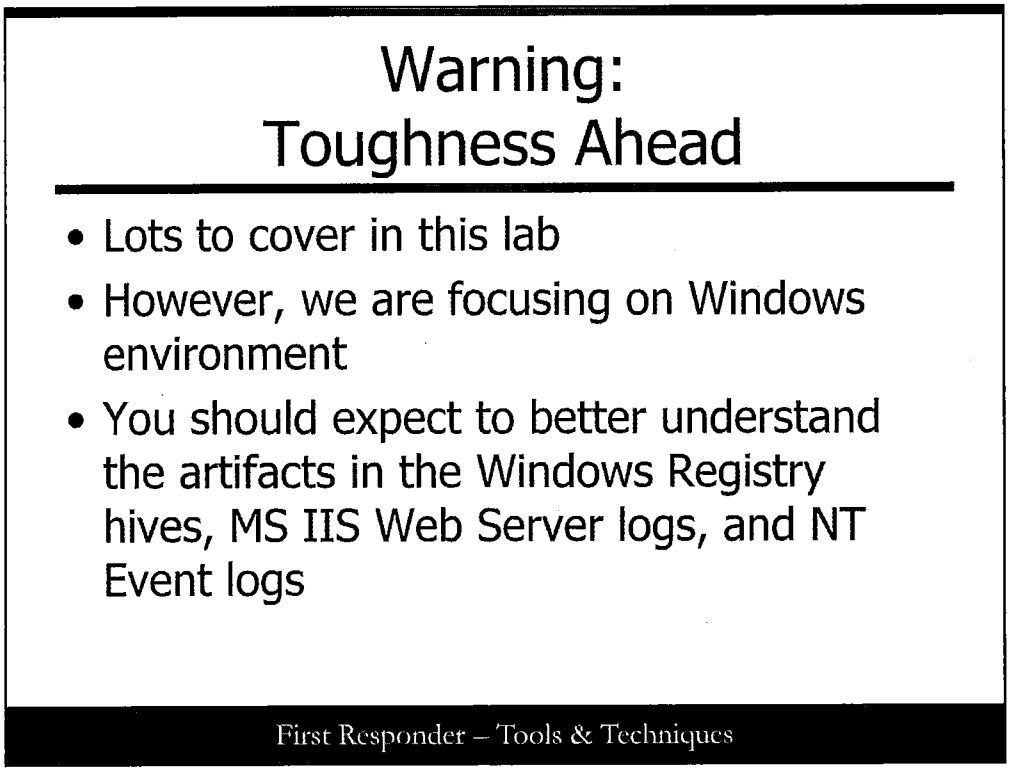
This page intentionally left blank.



Microsoft Windows Deep Dive

In this lab, we perform hands-on demonstrations of the MS Windows Registry using a particular analysis tool called RegRipper. RegRipper is a creation from the mind of Harlan Carvey (who also is an active participant in SANS conferences and events) and is nothing short of spectacular.

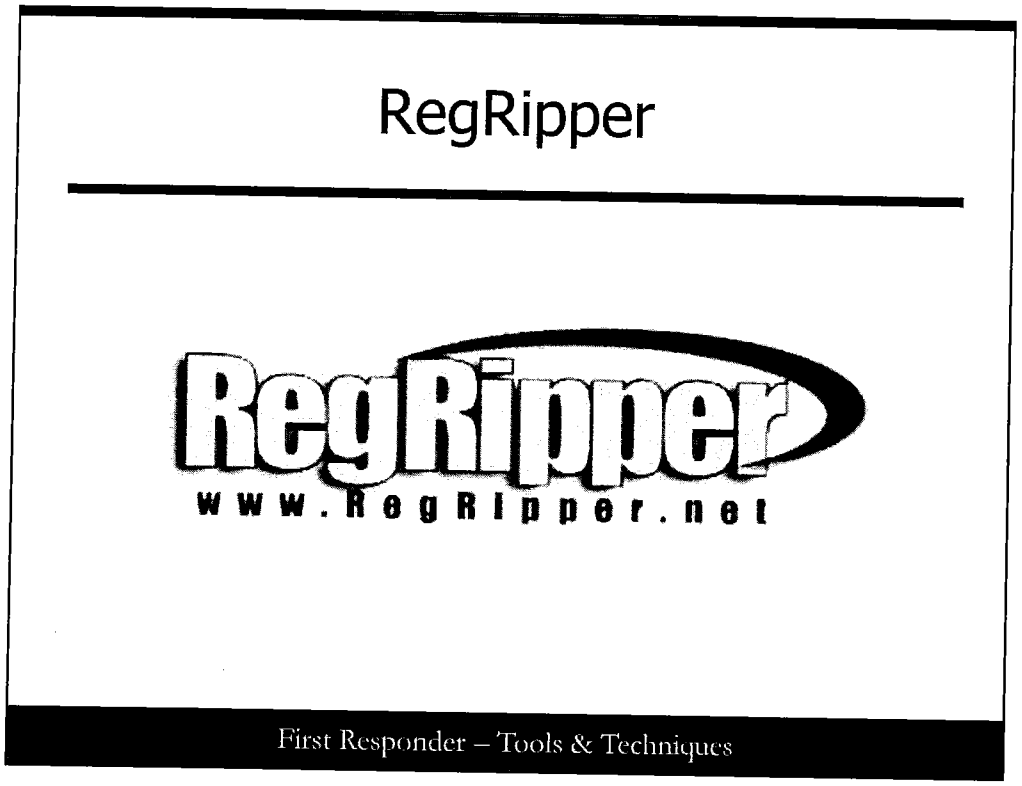
In addition, we look at a Microsoft tool called Log Parser and use it to examine and search through web server and Native NT event logs for clues of unusual activity.



Warning: Toughness Ahead

The objectives for this lab are tough, but this time we focus on the Microsoft platform to allow you the best opportunity to focus on the goal. At the end, you examine the following types of files harvested from various systems: Windows Registry hives, Microsoft Internet Information Server (IIS) Web Server log files, and Native NT event logs, which are used in Windows operating systems.

You will be introduced to tools that make analysis and initial assessment a bit easier regardless if unusual activity occurred.

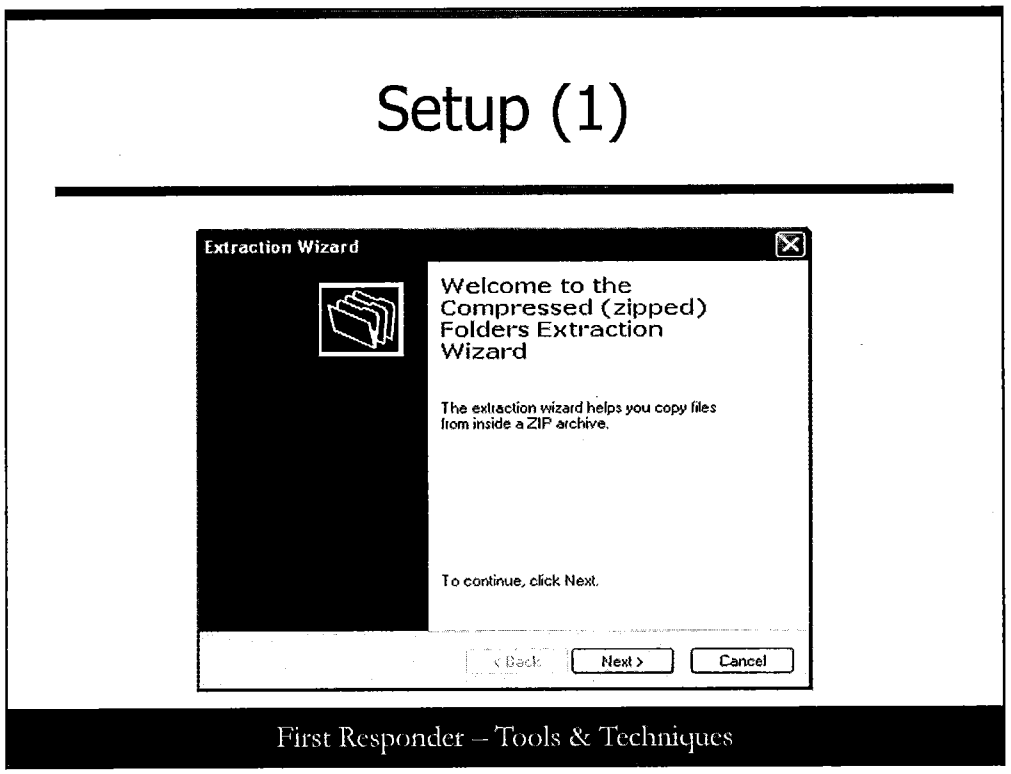


RegRipper

RegRipper enables you as an analyst to determine what state the system was in when it was shut down and to discover a great number of clues of what changes were made to the system, and in many, many ways, what actions were performed by a user. Find more out about RegRipper by going to the website at http://regripper.net/

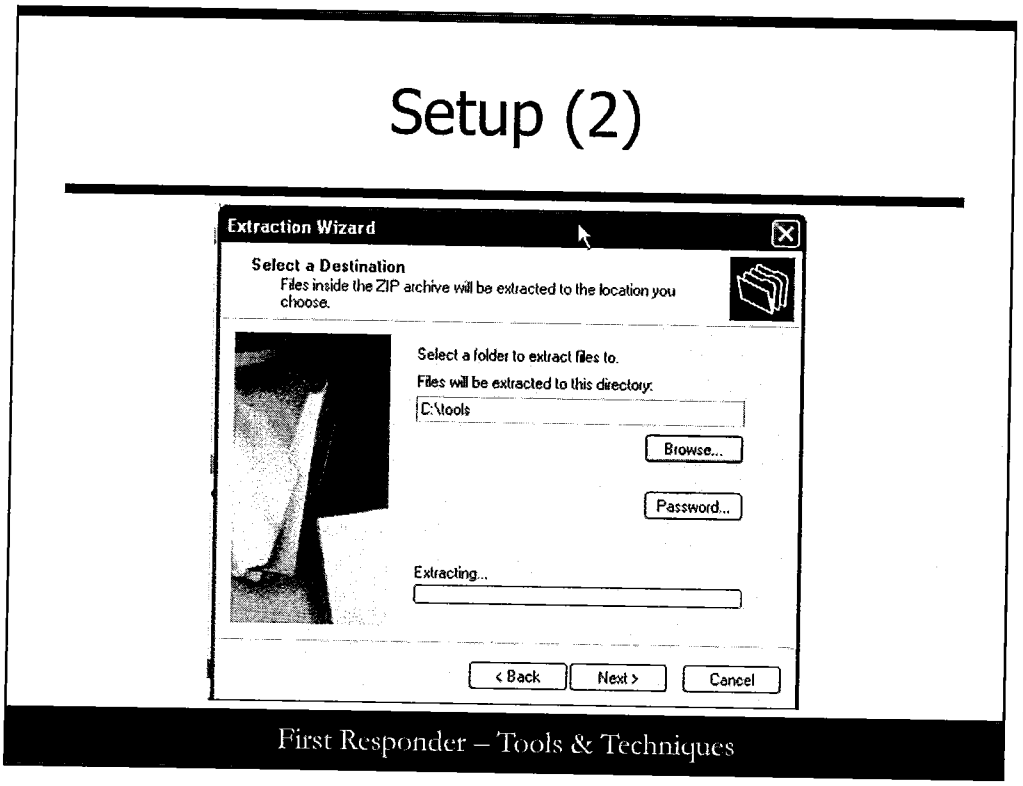
Make a directory called *\tools* off of the root of your C: hard drive and we can proceed: Issue the following commands to perform this action:

C:\> mkdir c:\tools
C:\> mkdir c:\Lab3



Setup (1)

Switching to your CD/DVD-ROM Drive, you then use the Windows Compressed Folders Extraction Wizard to extract RegRipper. By going to the CD/DVD-ROM drive and into the *Tools\RegRipper* folder, you need to right-click the *rr_20080909.zip* file to get the Extract All option. Follow this option to open the archive.



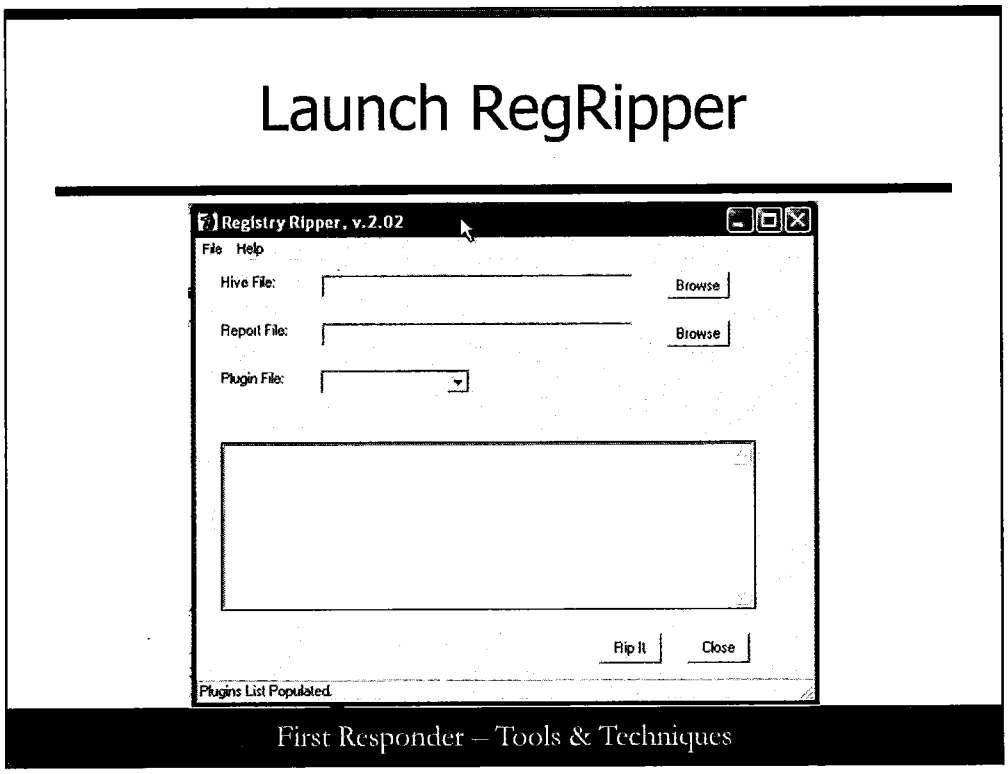
Setup (2)

Your destination folder will be the *C:\tools* folder you established earlier.

You are not obligated to use the selection of Show Extracted Files, but you have the option to.

Now open a command prompt window. From here you need to create a report folder on the root of your C: hard drive called \Lab3\ and change directories to the C:\Tools folder you put RegRipper into:

cd C:\tools

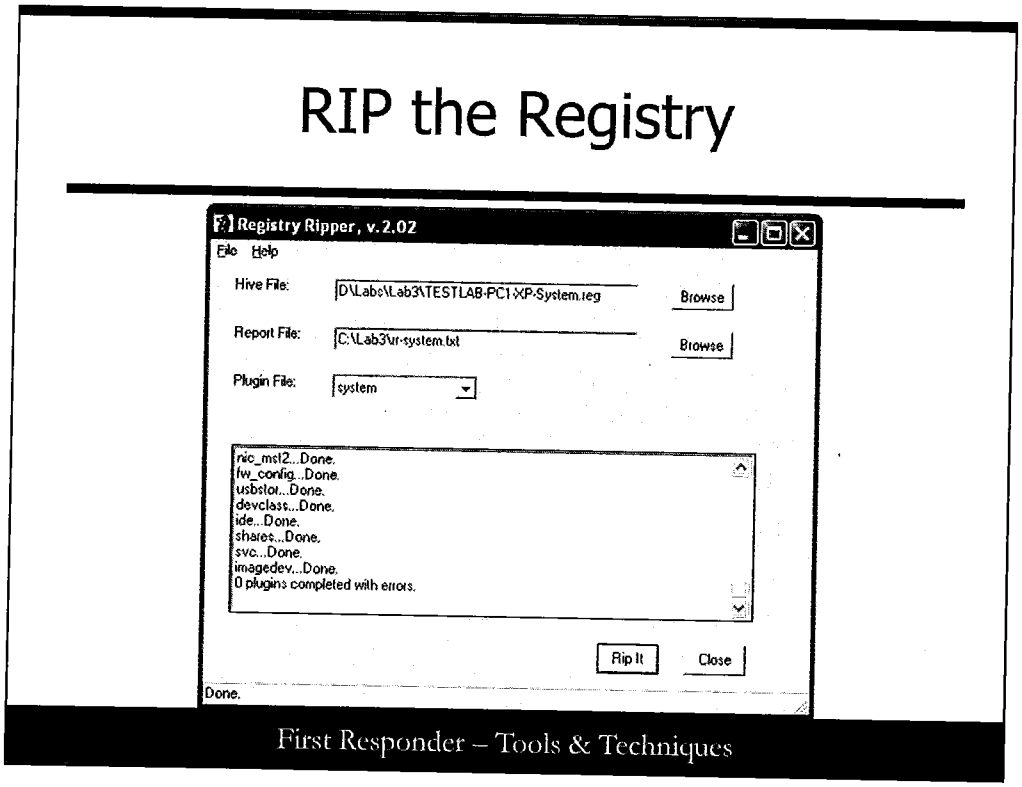


Launch RegRipper

Next, Use the Browse button, which is in line with the Hive File space to get the System Registry hive file located in the D:\Labs\Lab3\Registry folder. It's called TESTLAB-PC1-XP-system.reg.

Next, you need to designate a report filename and a destination folder to put it into. Click the Browse button in line with the Report File space. To remain consistent throughout this, let's prepend our report files with an rr- in front of them so that they are easily spotted later. Let's browse to the C:\Lab3\ folder and use rr-system.txt as our filename.

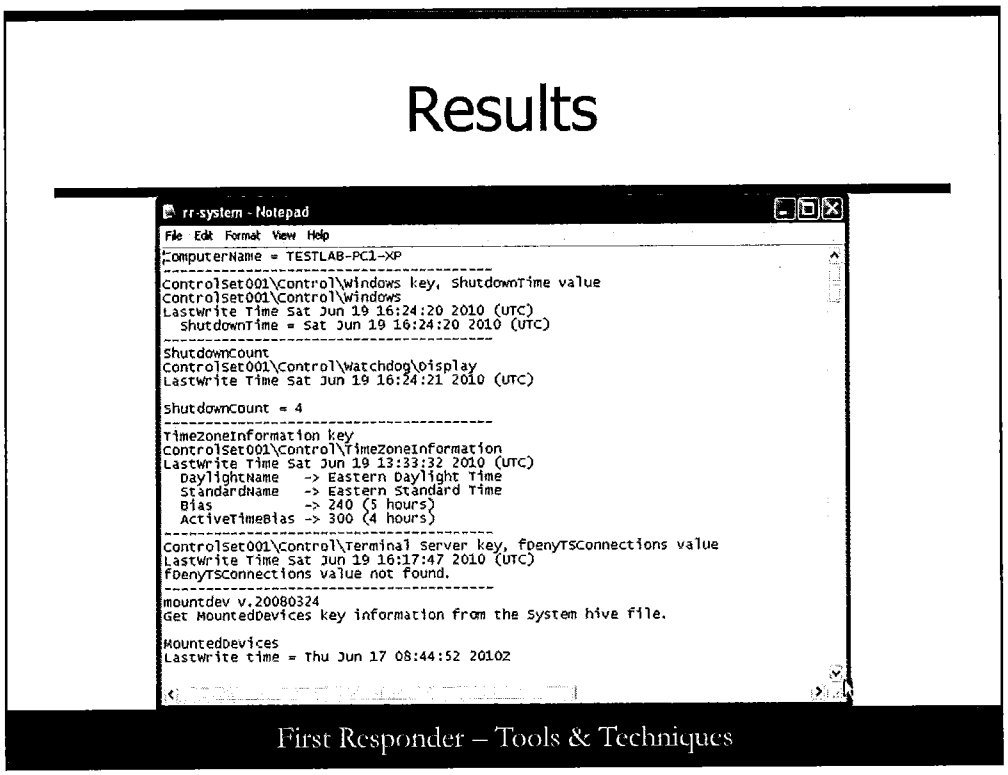
Note: RegRipper keeps a log (designated as a .LOG) file based on the root filename you established and records the success/failure of any applicable plug-ins into it. It also saves the results of the tests in a text (as a .TXT) file also based on the root filename you provided.



Rip the Registry

Next, Select system from the Plugin File selection option. Finish off by clicking Rip It on the lower-right side of the window.

The screen shows the results of what RegRippper looks like when we click Rip It.

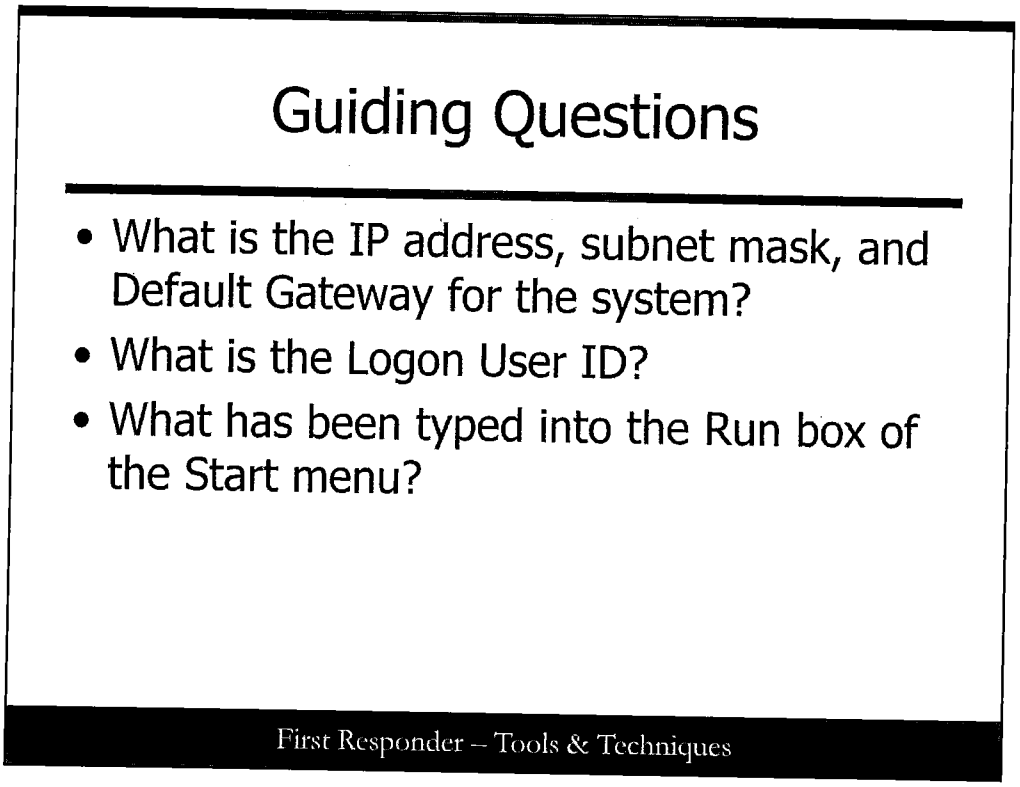


Results

Now use Notepad (or whatever text editor you like to use) to open up the rr-system.txt file in the *C:\Lab3* folder and look over what was picked up. RegRipper throws a battery of tests at numerous sections of the Registry hive it is looking at. Here are a couple areas that are useful to find:

- ShutdownTime
- TimeZoneInformation

This is by no means the only extent of what you can do with this tool. In the interests of limited time and priorities of work, we'll just keep the listing short and encourage you to explore what you find on your own in any spare time you have in this course, and outside of it. It's important that you do so to expand your knowledge on the artifacts that can be found and to fortify your base of knowledge in the subject matter.



Guiding Questions

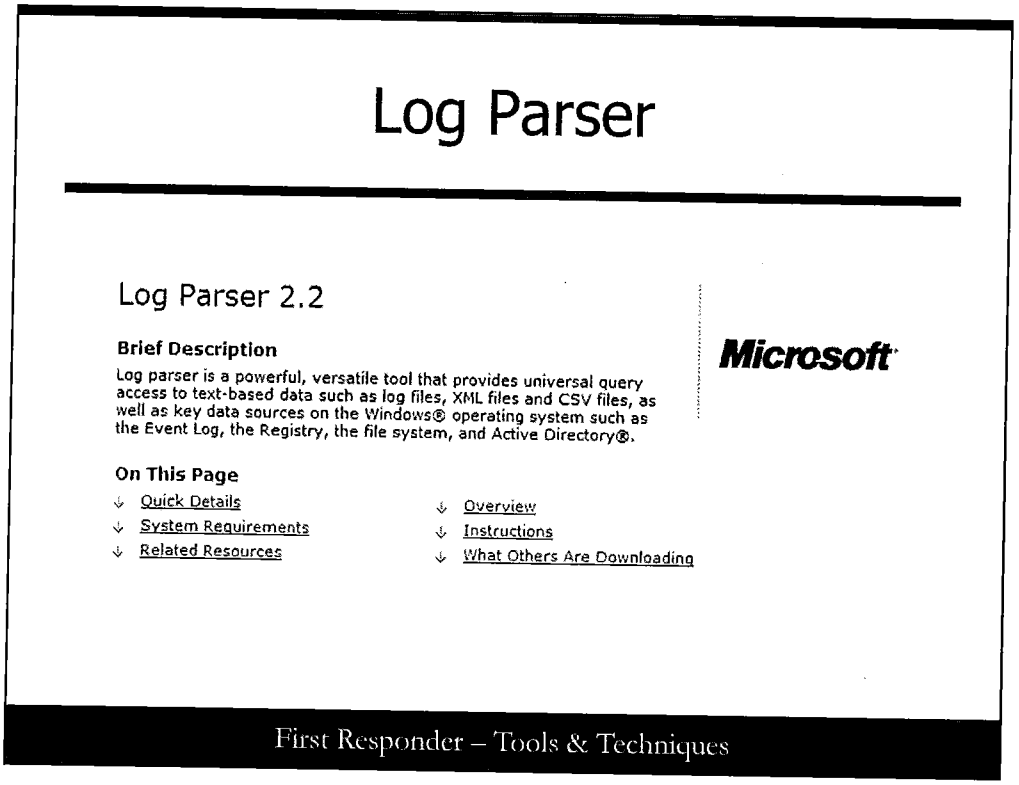
Review the file and specifically the section titled "Network key." Within it, you should find an entry for the Local Area Connection. Record the IP address, subnet mask, and DefaultGateway settings in the lines provided here.

Inside the *D:\Labs\Lab3\Registry* folder, you can find a Software, SAM, and even the NTUSER.DAT profile for a user, all which were harvested from a subject system. Repeat the previous steps short of reviewing them in a text editor until you have gone through all of them. Remember to set the corresponding plugin file selection to that of the Registry hive you are examining. Then, proceed onto the next question.

Examine the output report for the NTUSER and record the Logon User Name setting on the lines provided here:

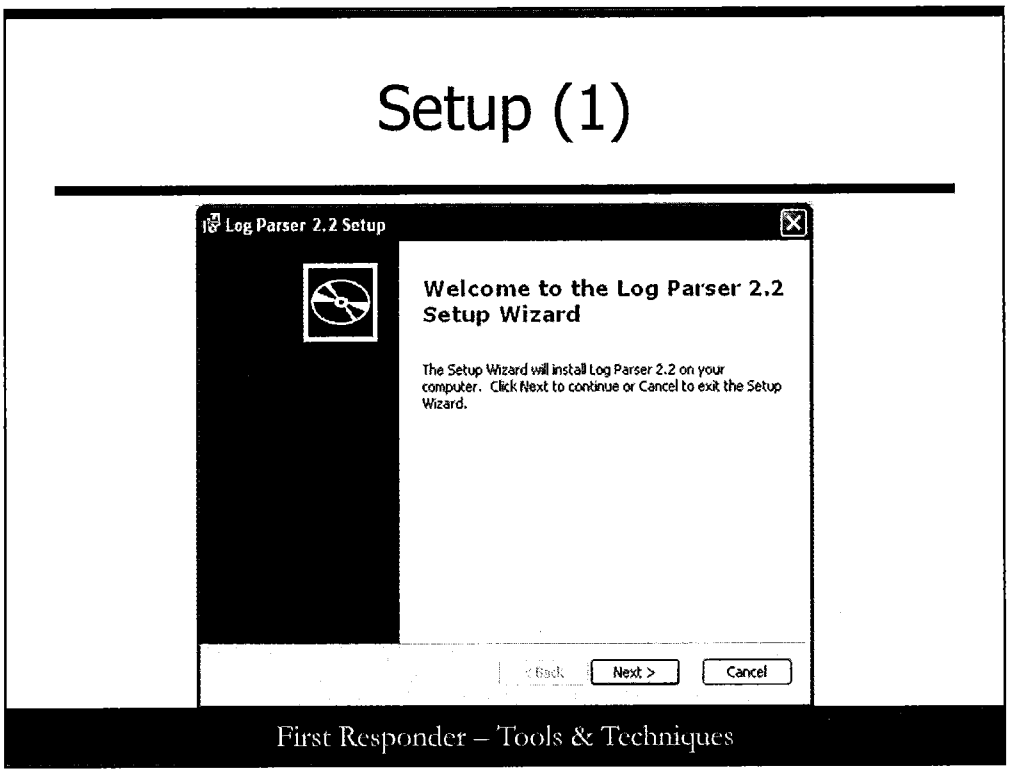
Examine the output report for the NTUSER and record what the user may have typed into the Run box of the start menu by examining the RunMRU (MostRecentlyUsed) settings on the lines provided here:

When you have completed the steps, exit out of RegRipper and close any open command prompt windows that you have open.



Log Parser

The next step in our block of labs is to use Microsoft's Log Parser tool for viewing logs. You will find many times that you have to power log review a number of log formats, and Log Parser goes a log way in helping you achieve that objective. In fact, Log Parser cannot only look at ASCII-text logs and their cousins such as XML and CSV/TSV, but also logs that are based in the MS-world such as the NT Event logs and those that are part of Active Directory®. Even better, it can assess Registry files or parse through a file system. If that is not enough, consider that Log Parser can take the input from a text format and even produce reports or output in HTML, XML, or even dump the data into a database using SQL!



Setup (1)

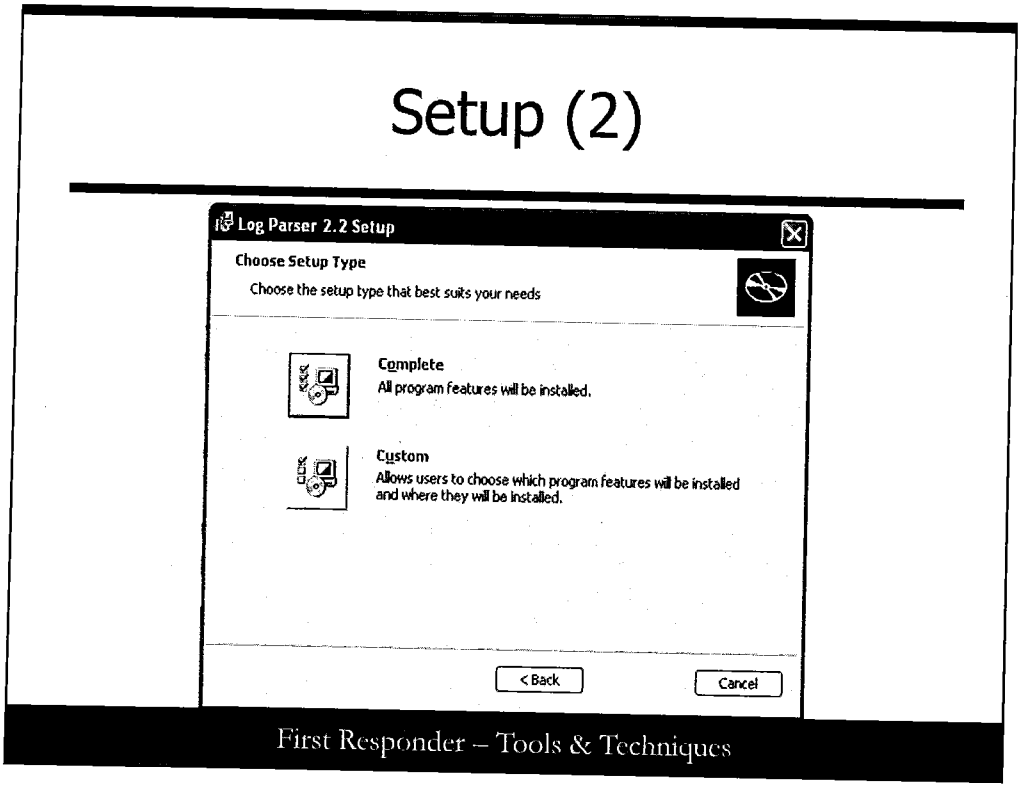
Get Log Parser 2.2 from the Tools folder of your course DVD.

The Setup-MSI file for Log Parser is located in the \Tools\ folder of your course DVD. Refer to it in the command line as <DRIVELETTER>:\Tools\LogParser where <DRIVELETTER> represents the drive letter for your CD/DVD-ROM drive.

Alternatively, you can download it from the Internet, provided you have a working network connection:

http://www.microsoft.com/downloads/details.aspx?FamilyID=890cd06b-abf8-4c25-91b2-f8d975cf8c07&displaylang=en

Double-click the Log Parser Windows Installer Package (.MSI file) to begin setup. Click the Next button.



Setup (2)

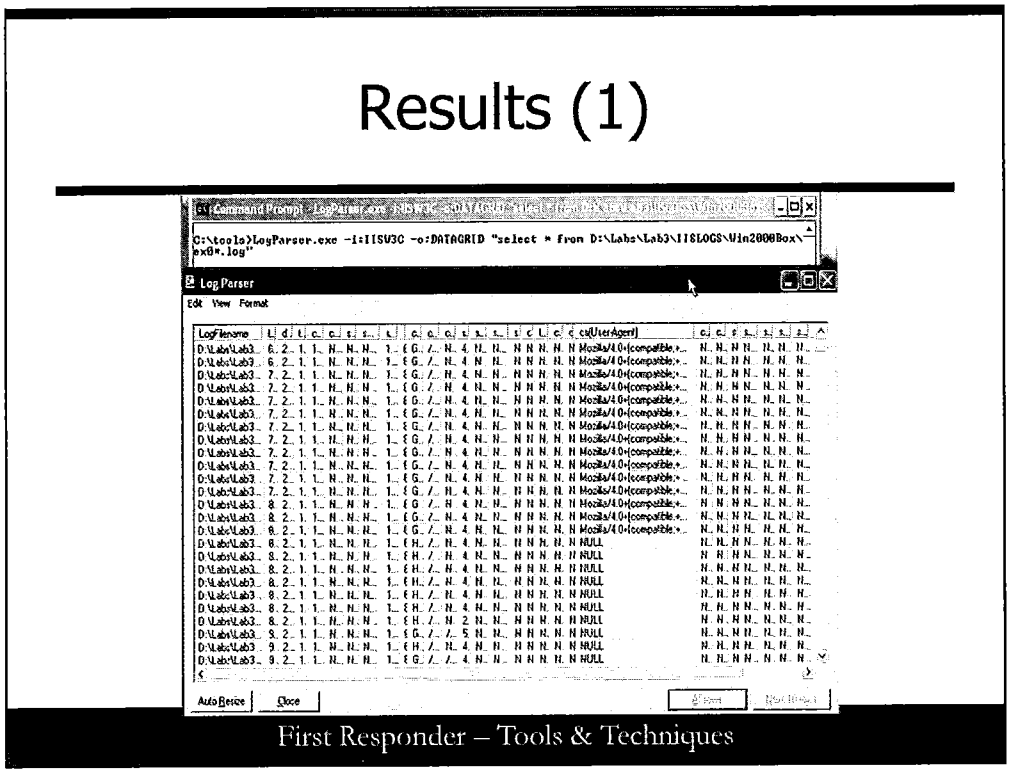
Read through the End User Agreement information and click I accept the terms in the license Agreement button and click Next. In the following window, you get to designate the Setup Type. Select Complete setup.

Then the next screen just gives you a moment to look over what will happen when the install occurs. Click Install to commit to the installation and subsequently the Finish button when it is completed.

Normally Log Parser puts itself into a folder, C:\Program Files\Log Parser; but it does not include this in your command execution path, for your command prompt, that we will invoke Log Parser with. So, we are going to navigate to C:\Program Files\Log Parser, copy ALL THE CONTENTS, and paste them into the \tools\ folder we've already created.

```
cd "c:\program files (x86)\Log Parser 2.2"
xcopy /E * c:\tools
cd c:\tools
```

That way we can have access to Log Parser from the command line simply by typing in C:\tools\LogParser.exe or change to that directory with a CD command and run everything from there. Go ahead and do that now to simplify the next steps.



Results (1)

Next up is Windows Internet Information Server and the analysis of those logs that record when web browsers and clients connect to them. Here we run the Log Parser against all the web server logs within the <DRIVELETTER>:\Labs\Lab3\IISLOGS\Windows 2000Box\ folder. (Substitute in your actual drive letter from now on, for each of the commands that are listed will assume D is the drive that the course DVD is in for the sake of brevity.)

*LogParser.exe –i:IISW3C –o:DATAGRID "select * from D:\Labs\Lab3\IISLOGS\Win2000box\ex0*.log"*

What happens if you properly type in the line is the Log Parser invokes a GUI window called the DATAGRID and shows you your results. It's a quick-and-easy method to look over your results of your search. From now on, we refer to the search as a query because we are actually invoking the mini-SQL engine built into Log Parser to process our query and display the results that match the criteria.

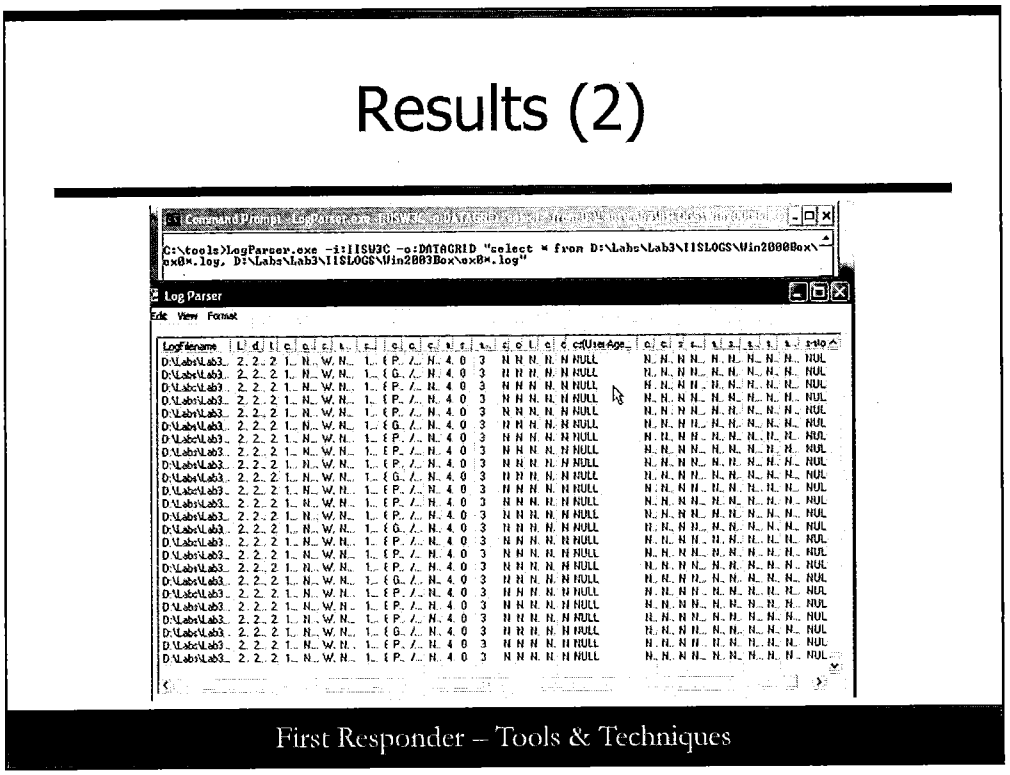
You should expect an experience that looks similar to the one depicted in the following screen shot. In addition, you will find that you can expand the column headings to reveal more information within.

Clicking the mouse cursor on the All Rows Button in the lower-right corner displays all the entries and continues to do so until it has reached the end. (Sometimes this can be slow.)

Clicking the Next 10 Rows button gets you exactly that, 10 rows of results.

Let's quickly review. In our command line, we used Log Parser and told it that we were going to process IIS web server logs, (And we used a regular expression * to refer to many of them in a folder where the * allowed us to match a filename criteria.) We told Log Parser to expect the input type to be IISW3C format, which matches the extended format now used in IIS logs. In addition, we told it that the output will be on a pop-up GUI window. (You can change that by examining the other types of –o: designations that are permitted—Bon Chance.) Lastly but most important, we were not restrictive in the number of events and rows we wanted to look at when we used a select * from <SOMETHING> directive. This is big because it may allow us to see all the events within the log files and then begin to understand and form new criteria to hone a later query. (For instance, maybe we want to select only a date, time, or source IP address of some activity, known in IISW3C format as the c-ip entry in the logs.)

When you finish viewing the results, click the Close button on the lower-left side of the window.



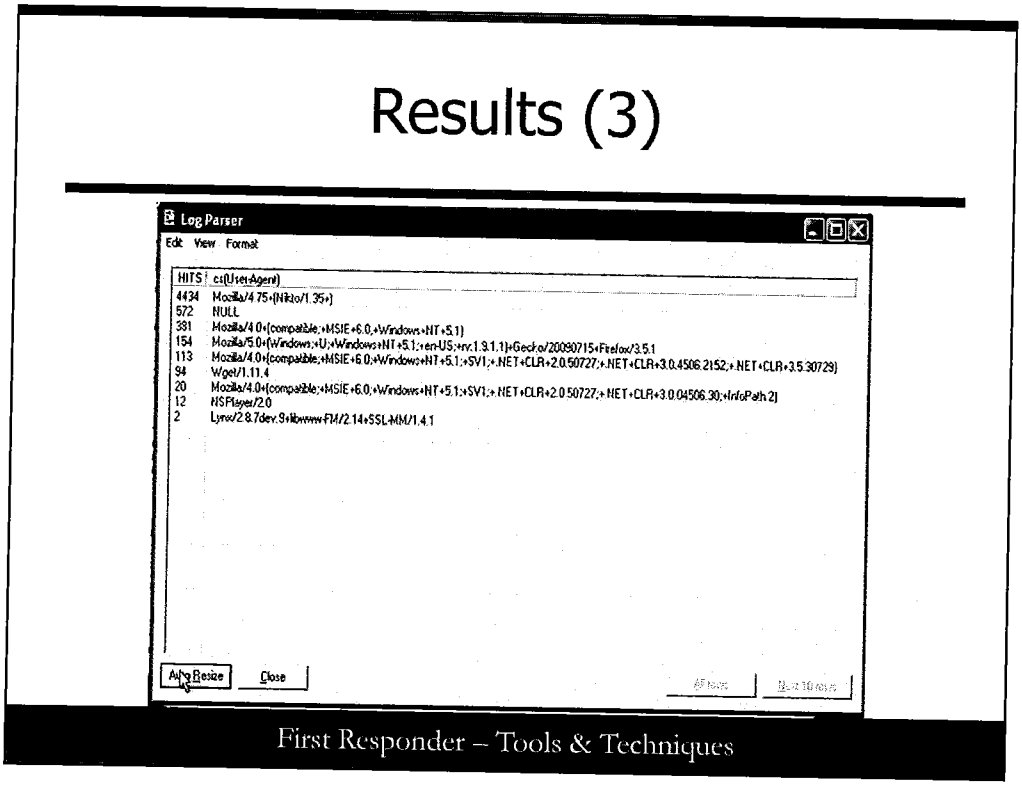
Results (2)

Next, we include multiple directories for Log Parser to search through. There is a simple reason for this. Sometimes there are many, many logs in many different folders that you need to search through and a simple find-in-files method may not do the job. Here is where a properly composed select query in LogParser can help you do this in a powerful manner. Run the following command, paying special attention to the comma (,) between the *D:|Labs\Lab3\IISLOGS\Win2000box\ex0*.log and the D:|Labs\Lab3\IISLOGS\Win2003box\ex0*.log* terms.

*LogParser.exe –i:IISW3C –o:DATAGRID "select * from D:|Labs\Lab3\IISLOGS\Win2000box\ex0*.log, D:|Labs\Lab3\IISLOGS\Win2003box\ex0*.log"*

You should be getting a sense of power log searching at this point because you can select as many folders or even individual files that you choose, and Log Parser dutifully marches through them to execute its query.

Now we are at the point in the exercise where we are going to be selective in what we ask for in our query. Suppose we want to see what number of hits were recorded in the web server logs from different types of web browsers. Sometimes managers and administrators want to know what their clientele use to navigate to their websites, so they can accommodate them with an experience that the browsers can handle. This information is recorded in IIS logs as cs(User-Agent). Let's look for instances of the different web browsers.



Results (3)

Run this query:

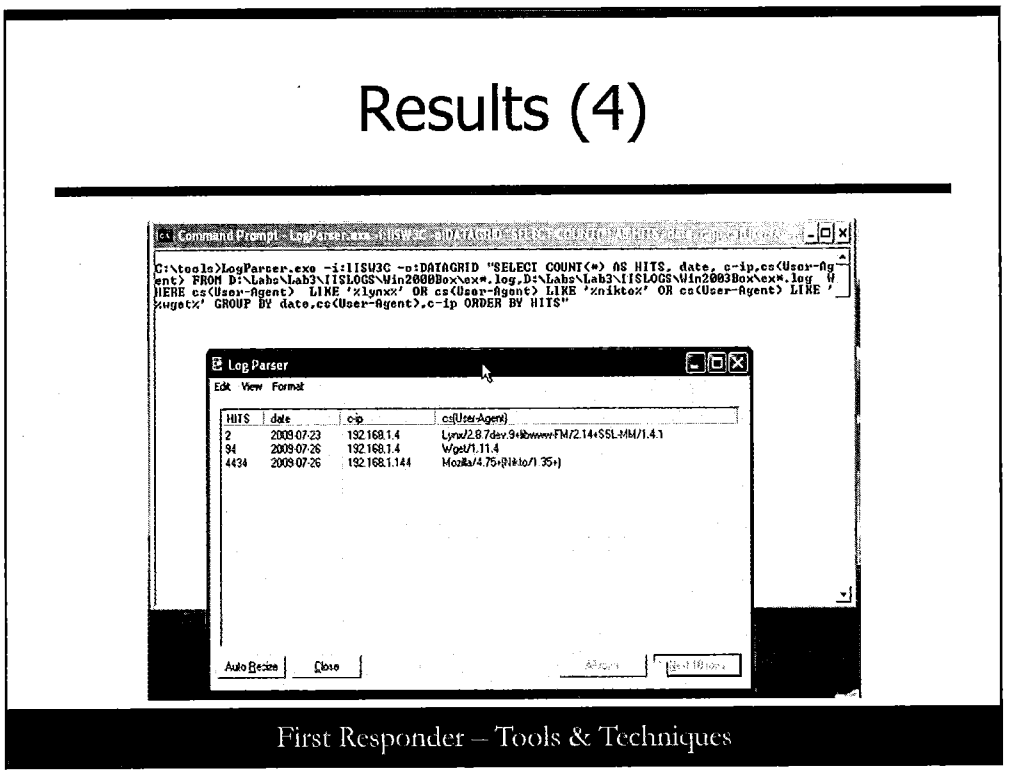
LogParser.exe –i:IISW3C –o:DATAGRID "select count(*) AS HITS, cs(User-Agent) from D:|Labs\Lab3\IISLOGS\Win2000box\ex.log, D:|Labs\Lab3\IISLOGS\Win2003box\ex*.log Group by cs(User-Agent) Order by HITS DESC"*

As a result, you get a concise list of the number of hits to the web server with respect to the type of browser that initiated the connection to the web server. Something will stand out very much now!

You should be asking what is going on with browsers that refer to themselves with strange names such as wget, Mozilla/4.75(Nikto/1.35+), NSPlayer, and Lynx.

These are unusual. How about finding which IP addresses and the dates some of these came from and when they occurred? The specific column terms will be date, c-ip, and cs(User-Agent). We also introduce the concept of using Like to establish a similarity in the SQL statement so that we don't have to exactly match either the beginning or the end of a keyword when we surround it with "tick-percent-percent-tick" (as in '%SOMETERM%').

Pay attention to the commas, single-quotes, and percent signs. Also, to make it easier in the upcoming task, many of the SQL directives in the command were capitalized to help you spot them. Don't worry about the case inside the "tick-percents" as it is not case-sensitive.



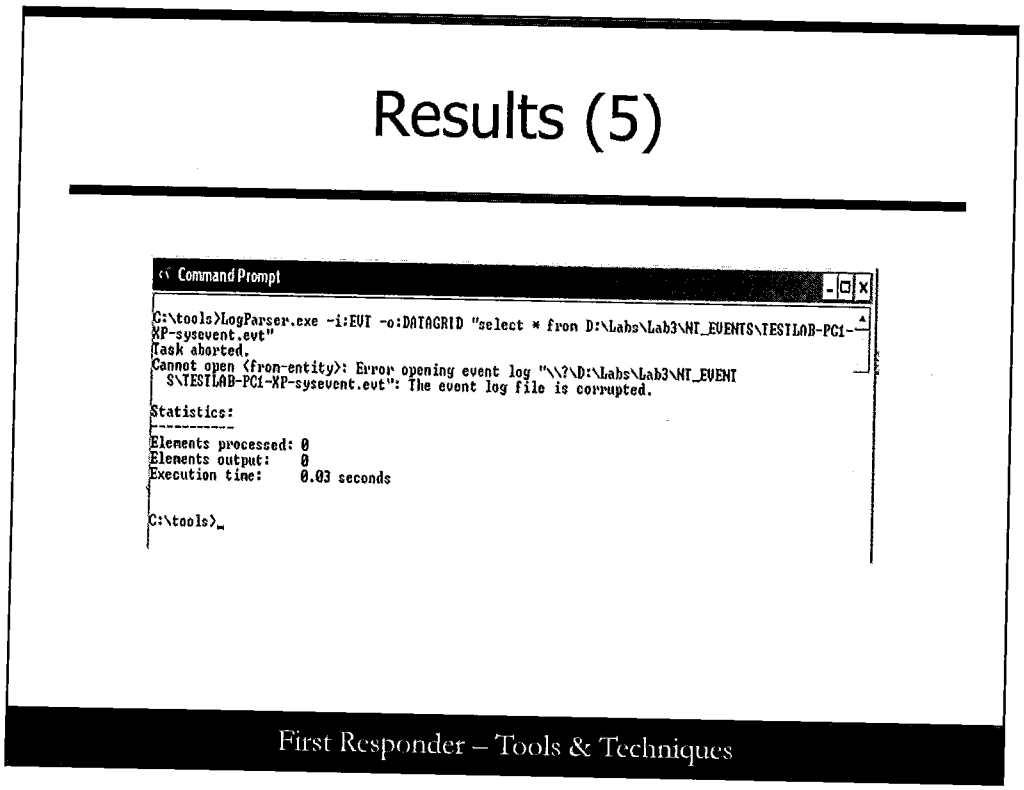
Results (4)

Enter in the following long, long line to get our three unusual browsers to be listed in grouped-by-the-date-of-connection listing and in ascending order according to the number of hits:

LogParser.exe –i:IISW3C –o:DATAGRID "SELECT COUNT (*) AS HITS, date, c-ip, cs(User-Agent) FROM D:\Labs\Lab3\IISLOGS\Win2000box\ex.log, D:\Labs\Lab3\IISLOGS\Win2003box\ex*.log WHERE cs(User-Agent) LIKE '%lynx%' OR cs(User-Agent) LIKE '%nikto%' OR cs(User-Agent) LIKE '%wget%' GROUP BY date, cs(User-Agent),c-ip ORDER BY HITS "*

You should get similar results and see that an IP address stands out.

If we were you, we would be singling out the IP address of 192.168.1.144 as it created a great deal of scanning traffic on July 26, 2009. If you are not already aware of it, Nikto is a popular open-source web server vulnerability scanner. This could be indicative of a reconnaissance and vulnerability assessment. An immediate response should be initiated to determine the cause and intent of this incident if this were an actual finding.



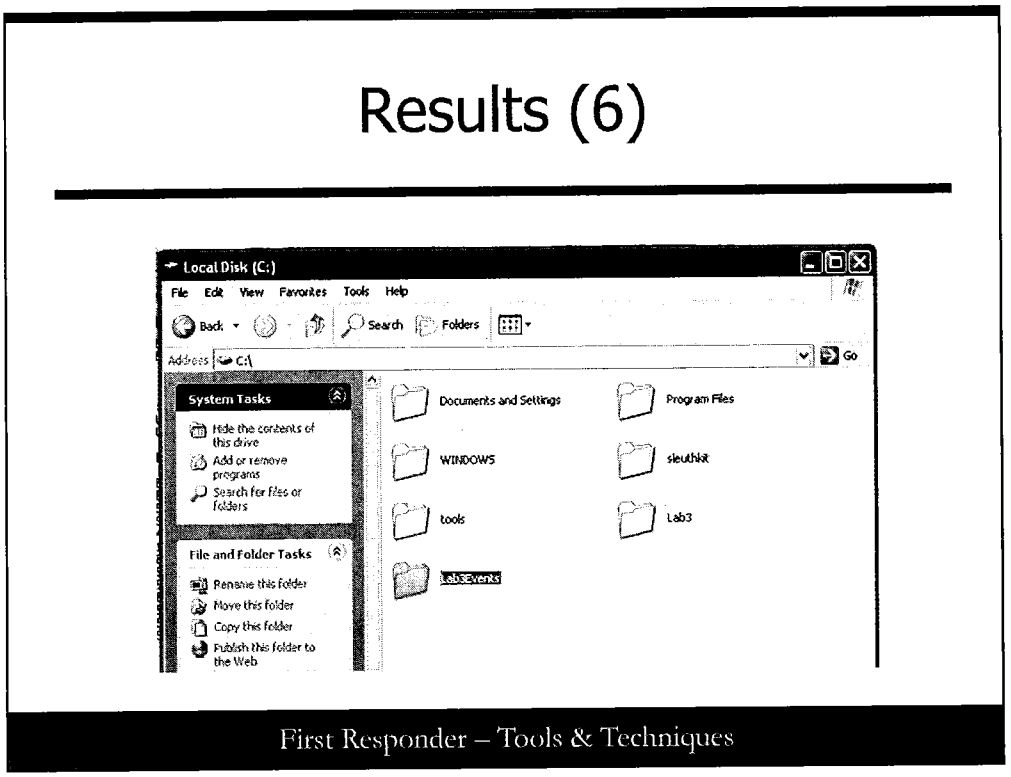
Results (5)

Now we attempt to use Log Parser to review the items that come from a Windows system event log facility. (It's important for success in this part of the lab that you use a Windows system, or VMware Virtual Machine, with Windows loaded on it.)

By changing the –i: designation, we can instruct Log Parser to search through NT-style, Native Event logs (or event logs for short) and review them for patterns or unusual circumstances. This time we use EVT in place of the IISW3C designation for the –i: switch.

*LogParser.exe –i:EVT –o:DATAGRID "select * from D:\Labs\Lab3\NT_EVENTS\TESTLAB-PC1-XP-sysevent.evt"*

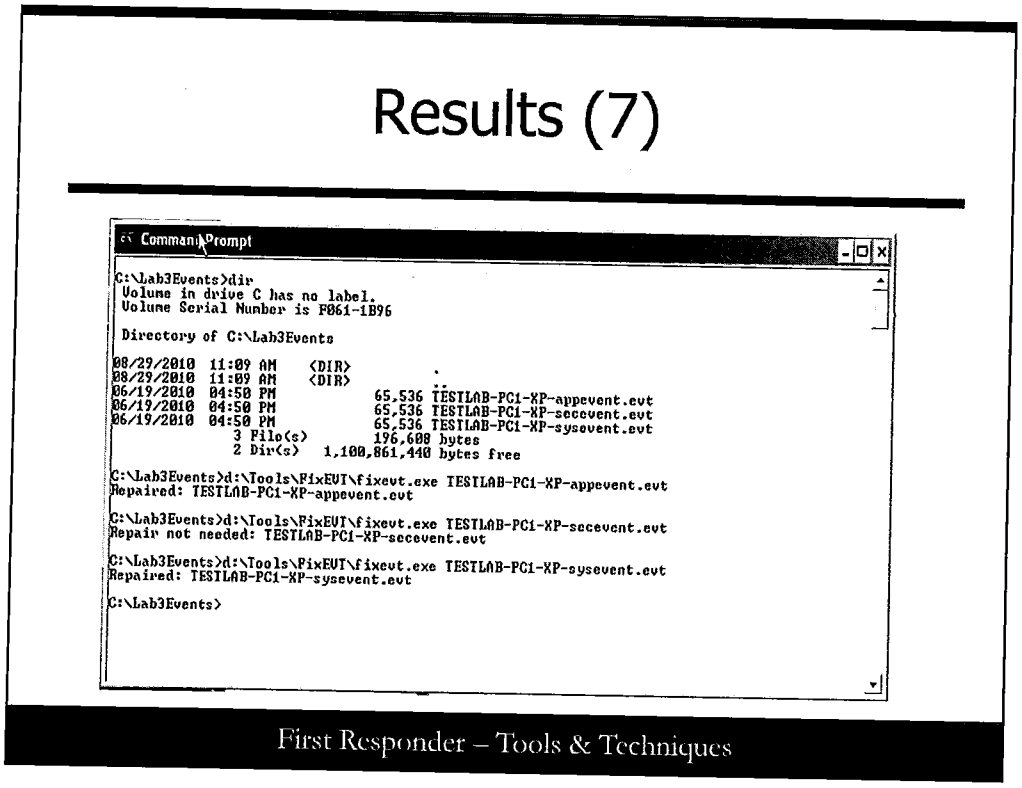
However, you find out it fails! When you run the command, you get a message to the effect that our event file is somehow corrupted for some reason. Compare your results to the screen shot.



Results (6)

This is a common issue with imported event logs. But in our case because we are using a Windows system or VMware Virtual Machine loaded with Windows 7 for our lab, the Event File and the Windows API is not an issue. (Sometimes users get this problem reading logs from one type of Windows system on a vastly different Windows system. For instance Windows XP versus Windows 7; the EventFile structure is completely different and Log Parser is dependent on the host system's Eventlog API to run.) Other instances are like ours here. The Event logs from the subject system and the host analysis system are both of a common operating system, Windows XP. So we can easily fix this.

To fix this, we need to copy our event files over onto our local system. Create a folder on the root folder of your C: hard drive and name it *\Lab3Events* as depicted in the screen shot.

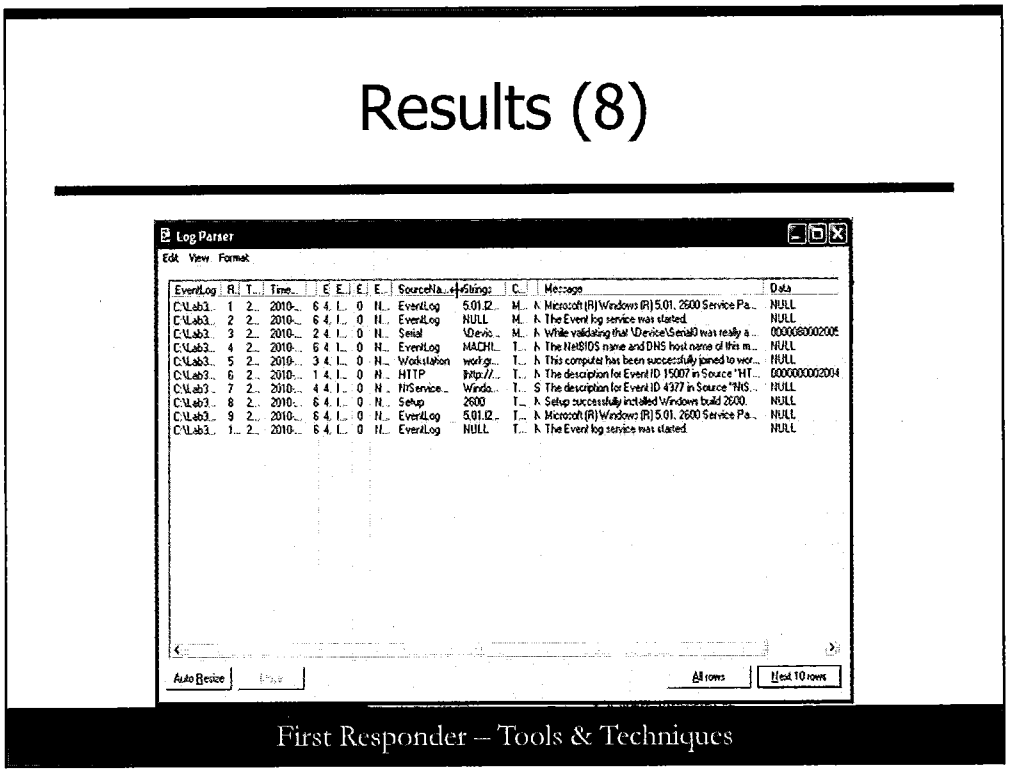


Results (7)

Copy the TESTLAB-PC1-XP event files into it. When they are copied, change to the *C:\Lab3Events* folder you created and then run the *Fixevt.exe* file in the form here in a command prompt window:

D:\Tools\FixEVT\fixevt.exe TESTLAB-PC1-XP-appevent.evt
D:\Tools\FixEVT\fixevt.exe TESTLAB-PC1-XP-secevent.evt
D:\Tools\FixEVT\fixevt.exe TESTLAB-PC1-XP-sysevent.evt

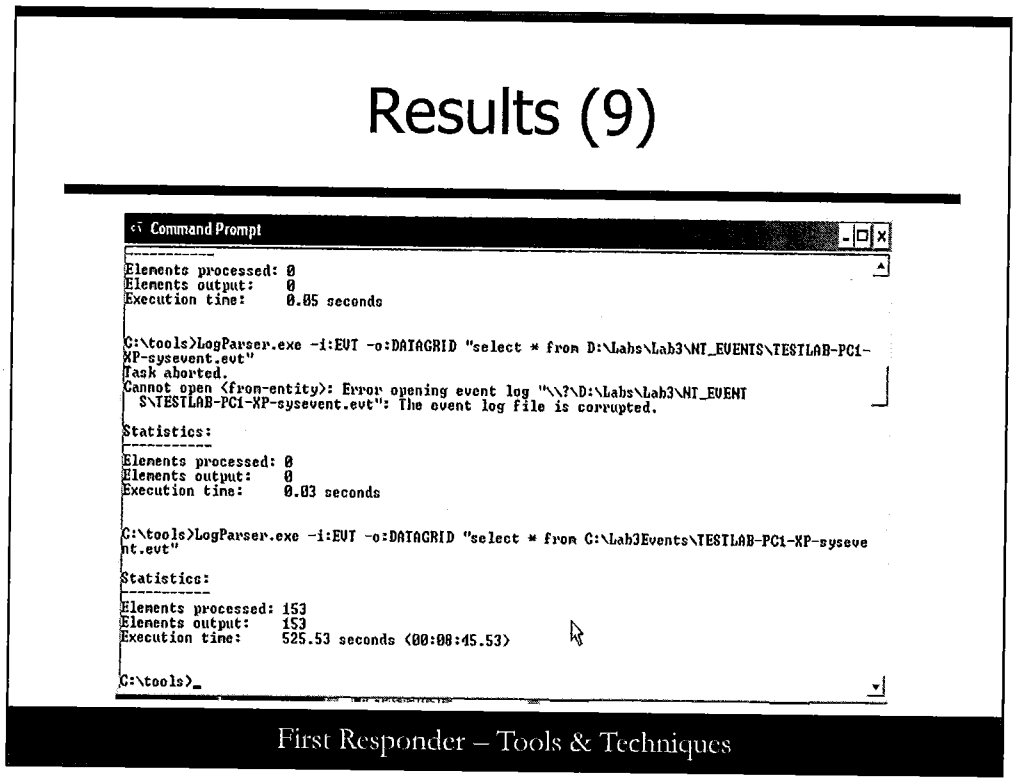
Note: Two things you should notice. One is that if you are in the *C:\Lab3Events* folder you are allowed to refer to the subject Event files without using the full path. The other item to notice is that FixEVT reported one of the three event files that did not need to be repaired. This is expected from time to time.



Results (8)

Now that we have our event files repaired from their corruption,' we can try our LogParser query again. Try it on the System Event log, *TESTLAB-PC1-XP-sysevent.evt.* Compare your results with the screen shot following the instructions.

*LogParser.exe –i:EVT –o:DATAGRID "select * from C:\Lab3Events\TESTLAB-PC1-XP-sysevent.evt"*

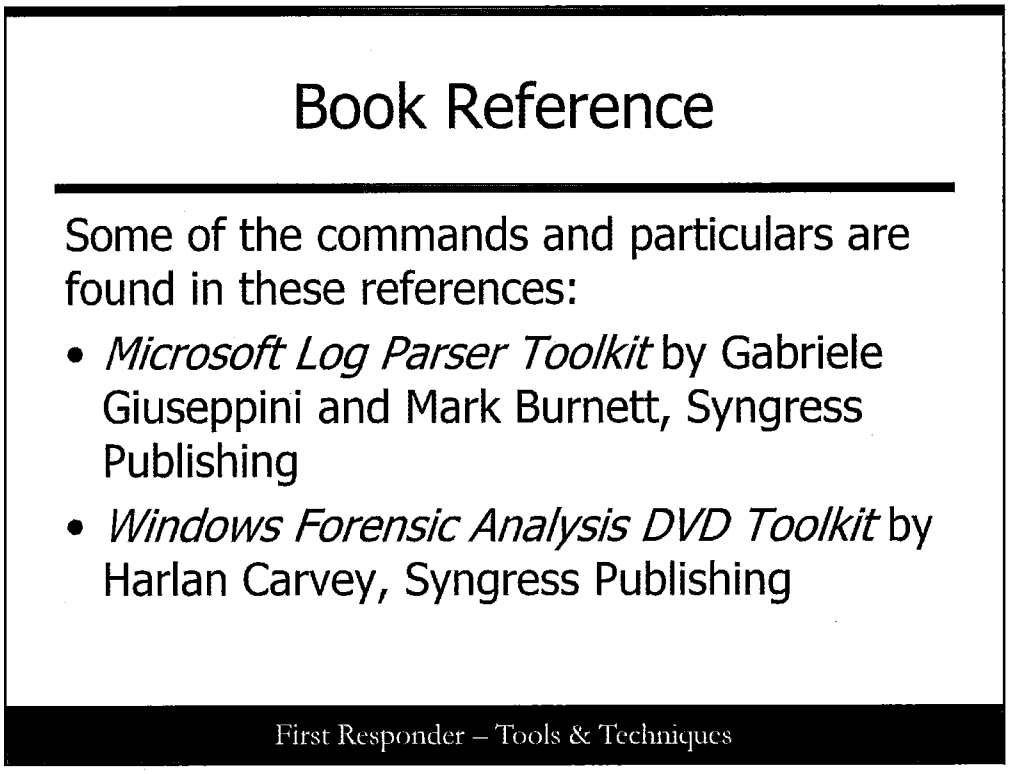


Results (9)

When you are satisfied with the DATAGRID output, you can click the Close button and review the output in the command-line window. This time you should see that all the elements in the *TESTLAB-PC1-XP-sysevent.evt* file were processed. This is good for us.

Sometimes, log file analysis and event log parsing is like a complicated high-math problem, for instance integration in the subject of Calculus. Sometimes it requires a trial-and-error process of trying numerous methods of integration before finding a solution that works. So keep at it in the event your first attempt falls short.

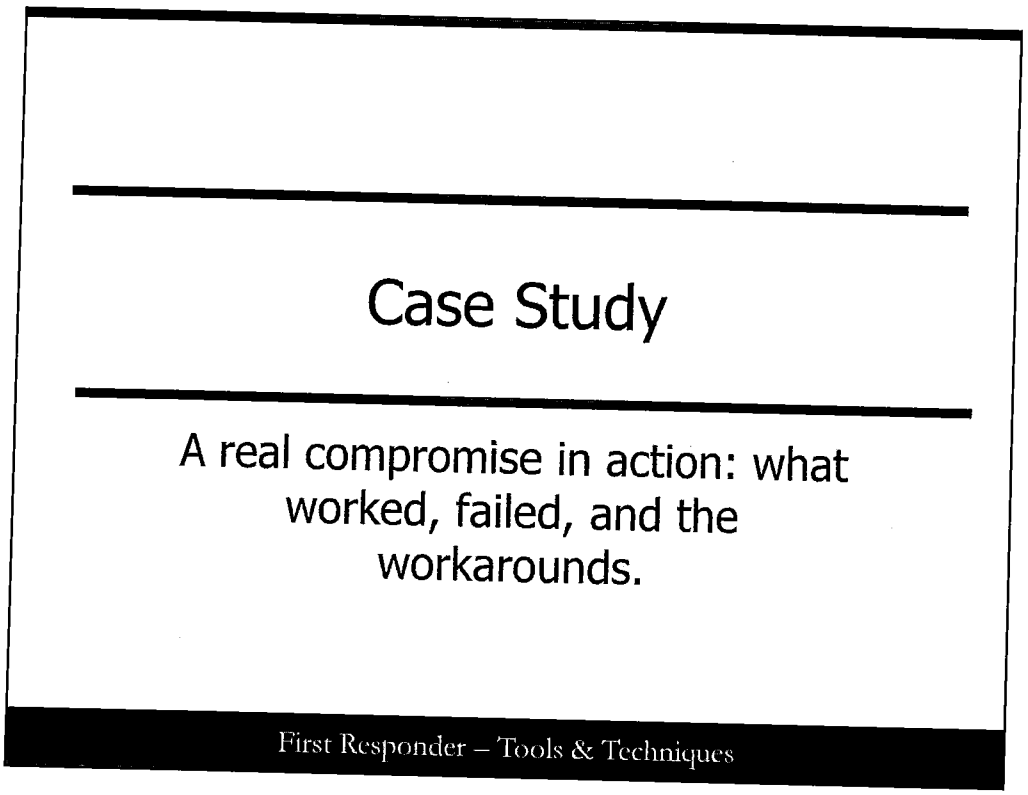
Now you should jot down what you've learned in this lab and work to derive other methods in which you can use what you picked up here in this lab exercise.



Book Reference

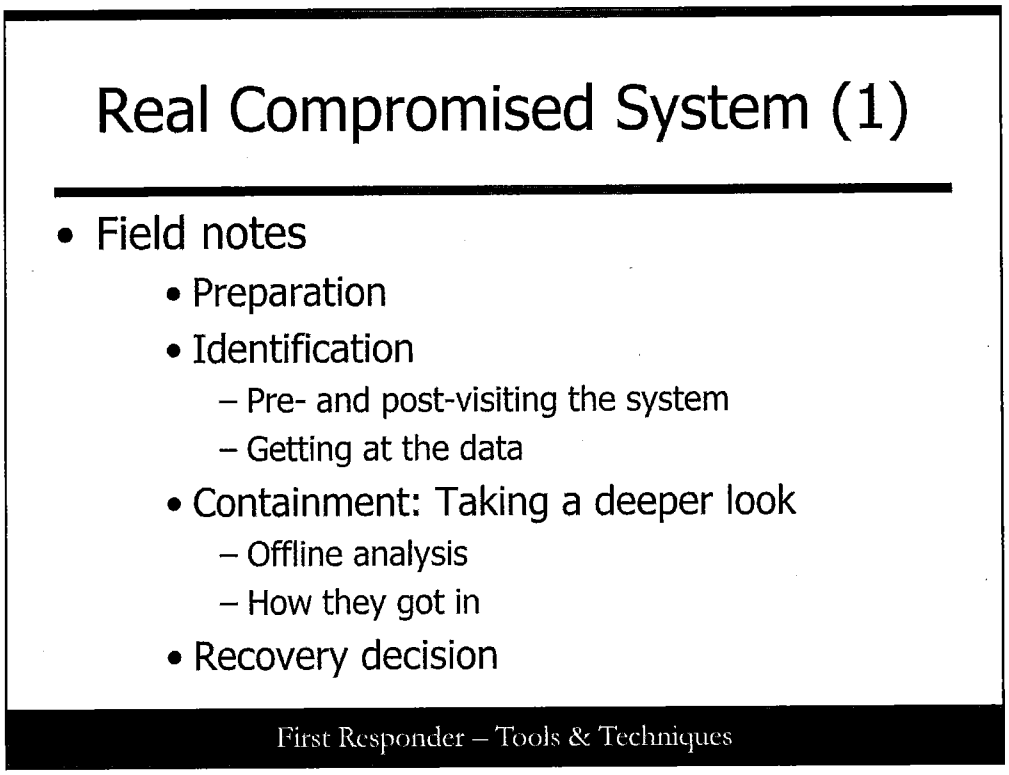
The following references may help you expand what you've learned in this exercise, and we recommend that you go find them and add them to your bookshelf for reference:

- *Microsoft Log Parser Toolkit,* by Gabriele Giuseppini & Mark Burnett, Syngress Publishing, ISBN 978-1-932266-52-8
- *Windows Forensic Analysis DVD Toolkit* by Harlan Carvey, Syngress Publishing, 2nd Edition ISBN 978-1597494229



Case Study

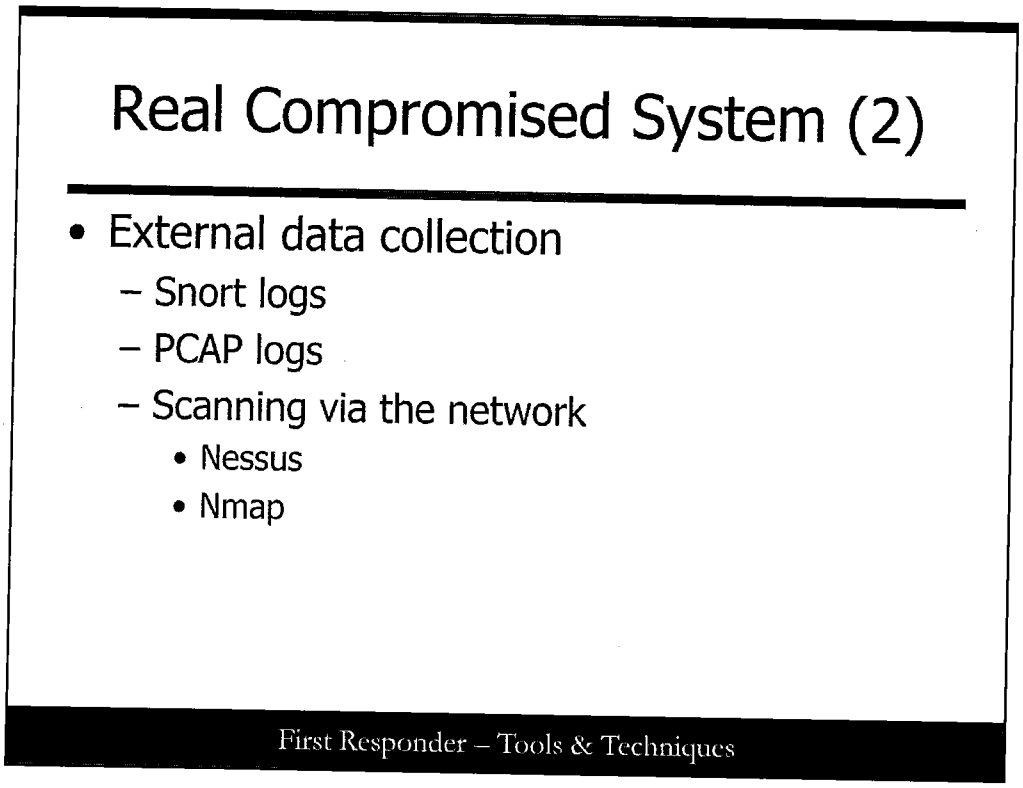
This is an actual case. Many of the tools and techniques work, and many don't. This case provides some workarounds when things don't quite go as planned.



Real Compromised System (1)

This section discusses the steps taken during a real incident to identify and remediate the compromised system. The system was a low-usage project support system. It provided FTP, Web, and e-mail support for a select group of people.

This system was initially identified as being compromised when a user noticed that the main website this system hosted was defaced. It was replaced with a message informing the administrator that their system has a vulnerability and that it needs to be patched. At the bottom of the page was a promotion for warez and hack groups.



Real Compromised System (2)

After being informed that this system was compromised, we started to collect data from external sources. These external sources include snort logs, PCAP logs, and scanners.

The first step was to attempt to capture live data that could be post processed for inbound and outbound hacking or scanning. This data could be useful in determining what type of applications are running that should or should not be running. This task may be performed by a security administrator or network engineer. If this task falls to you, you might use an application such as TCPdump (UNIX/Linux) or windump (Windows). The command we used to collect data follows:

tcpdump -i eth1 -s 2048 -w incident_id.dump "host 10.1.1.10"

The second step was to troll the snort logs for the IDS system looking for alerts that may correspond to a compromise of the system in question. There was nothing of significance shown in the snort logs for the previous day and current day that this system was reported to be compromised.

Our next step was to get a list of ports open on this host. We used Nessus to scan this system for vulnerabilities and open ports. We enabled the Nmap feature of Nessus, so we did not need to run Nmap as a separate process. The Nessus scan pointed out the possible version of software listening on various ports and whether they may be vulnerable. We externally determined this system had ports 80/tcp (apache) and 21/tcp (proftpd) open among others. External scans give you invaluable information about the system before you visit it. With this information we knew that we should expect to see an FTP and web server running on this compromised system.

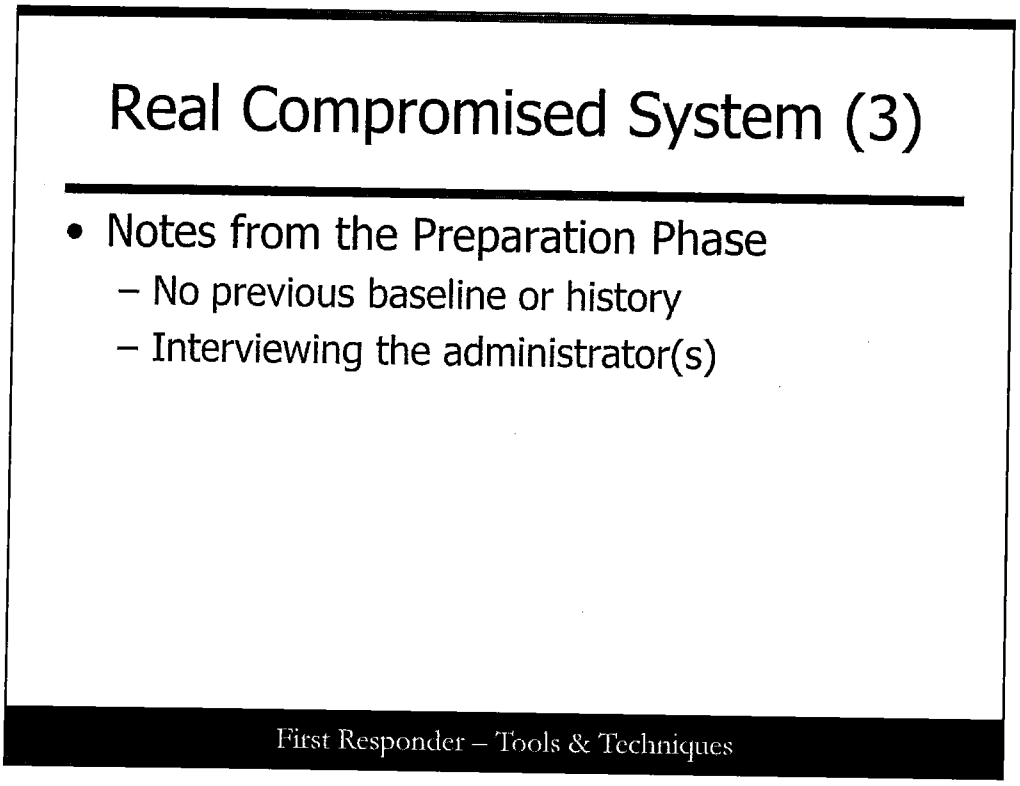
During the interview section, which you learn more about on the next slide, this information is useful in comparing what is running versus what should be running.

If Nessus is unavailable, you can use Nmap to get a fingerprint of the system, which includes type of OS and open ports. The following command can be used to run Nmap on a system:

nmap -O 10.1.1.10

For more information about TCPdump, Windump, Nessus, or Nmap, visit the following links:

TCPdump: http://www.tcpdump.org/
Windump: http://www.winpcap.org/windump/
Nessus: http://www.nessus.org/
Nmap: http://www.insecure.org/nmap/

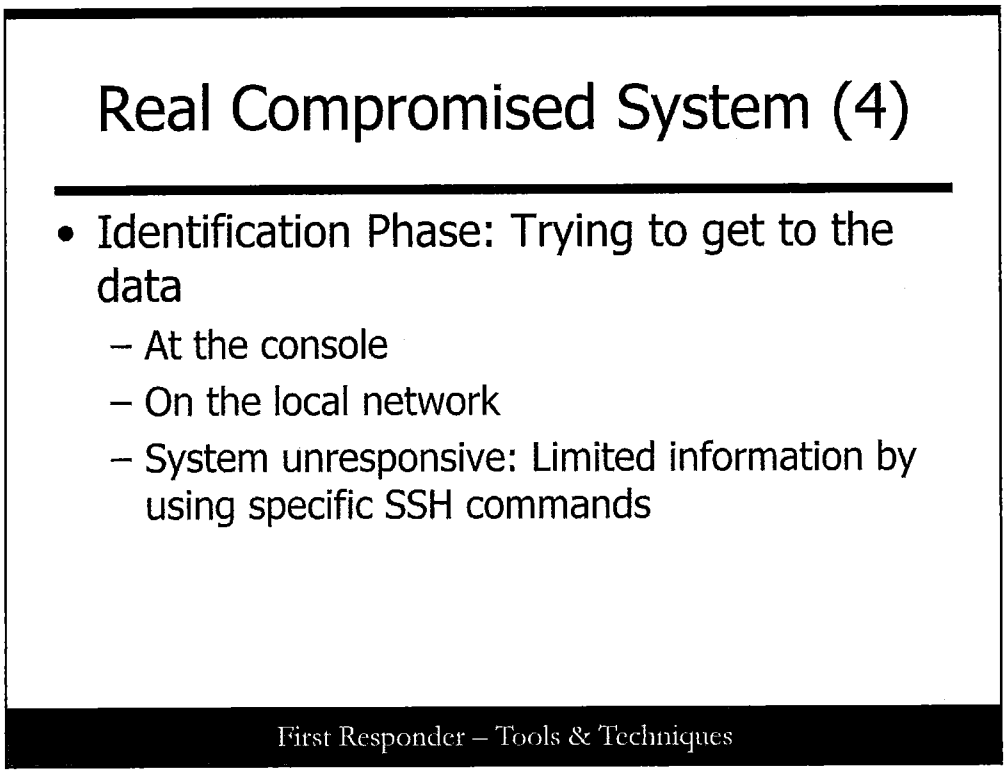


Real Compromised System (3)
This system was privately maintained by another department. When this happens the level of logging history and baseline may not be available. If you know the time frame of the compromise, you might determine when files should be known as "good." In our case we had not previously seen this box nor were we familiar with the practices of its system administrator(s).

The system was unresponsive during console logins, both root and regular users. The system would pause, not completing the login process. To get a better understanding of what applications and services should be running on this, the system administrator(s) were interviewed. The kinds of questions that were asked follow:

- Does this system's syslogs centrally log to an external source?
- When was the last time the system was updated?
- When was the last time the root user account was used?
- When was the last time the system was rebooted or shutdown?
- What applications and services should be running on this system?
- Does the system have backups?
- How many people have user-level access to this system?
- How many people have admin-level access to this system?

Using the answers from the questions asked here, it was possible to learn that this system was used by 10 or less users, had 2 primary admins, was updated/patched less than a month ago, ran a web server for hosting information for the 10 or so user population, allowed FTP access to upload/download files, and was not configured to centrally log syslog logs.



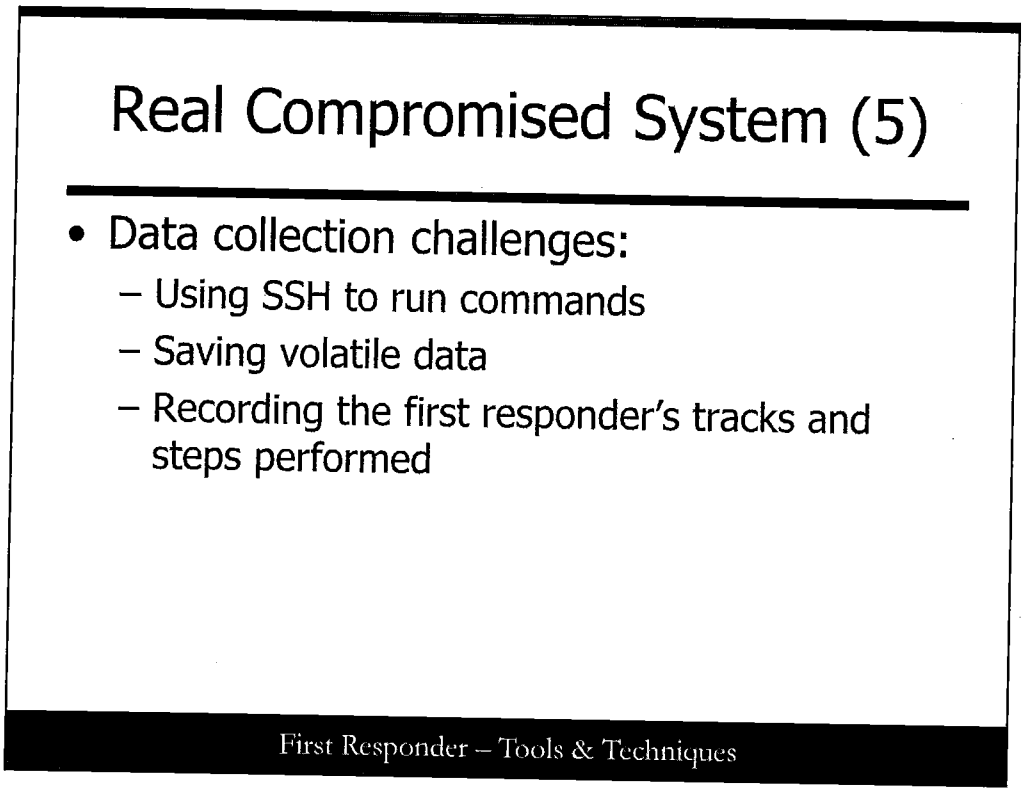
Real Compromised System (4)
We were unable to log into the console of this compromised system. The next step was to use a crossover cable and plug a Linux laptop directly into the server. Normally, you want to get onto the system before bringing down the network link. The reason for this is that you do not want the behavior of the system to change. Some applications, malware, are network link-aware meaning the malware may exit or go dominant if it notices that the network link has dropped. Keeping this in mind the amount of time that the link was down was minimized.

Next the laptop was configured using the IP address of the gateway for the network of the compromised system. This made it possible for the laptop to respond to basic ping or ARPs for the gateway IP address of the compromised system. Running a TCPdump on the laptop also allowed for the collection of all packets directed to the "gateway" by the compromised system.

After the laptop was connected and configured, a connection to the compromised system was opened using SSH. Logging in to the system via SSH resulted in the same behavior found while trying to log in to the console. A final attempt to get into the system was attempted using SSH in command-line fashion. This means that instead of trying to establish an interactive SSH session, one command per SSH run was attempted. This allowed us to get onto the system and record volatile information such as ps output, data stored in memory, the state of proc, data stored in /tmp, and so on. The following syntax was used to run SSH in a command-line fashion:

ssh -l root 'ps -auwwx > /incident/ps-auwwx.txt'

The system prompts you for a password, runs the command without invoking a shell, and then exits.



Real Compromised System (5)

Now that the system was accessible, data needed to be collected and preserved. At this point, collecting the valid data was important. Using the sshd command-line syntax, the /proc and /tmp directories were archived into a tar-ball.

```
ssh -l root 'tar -zcvf /incidents/2005.07.18-proc.tar.gz /proc'
ssh -l root 'tar -zcvf /incidents/2005.07.18-tmp.tar.gz /tmp'
```

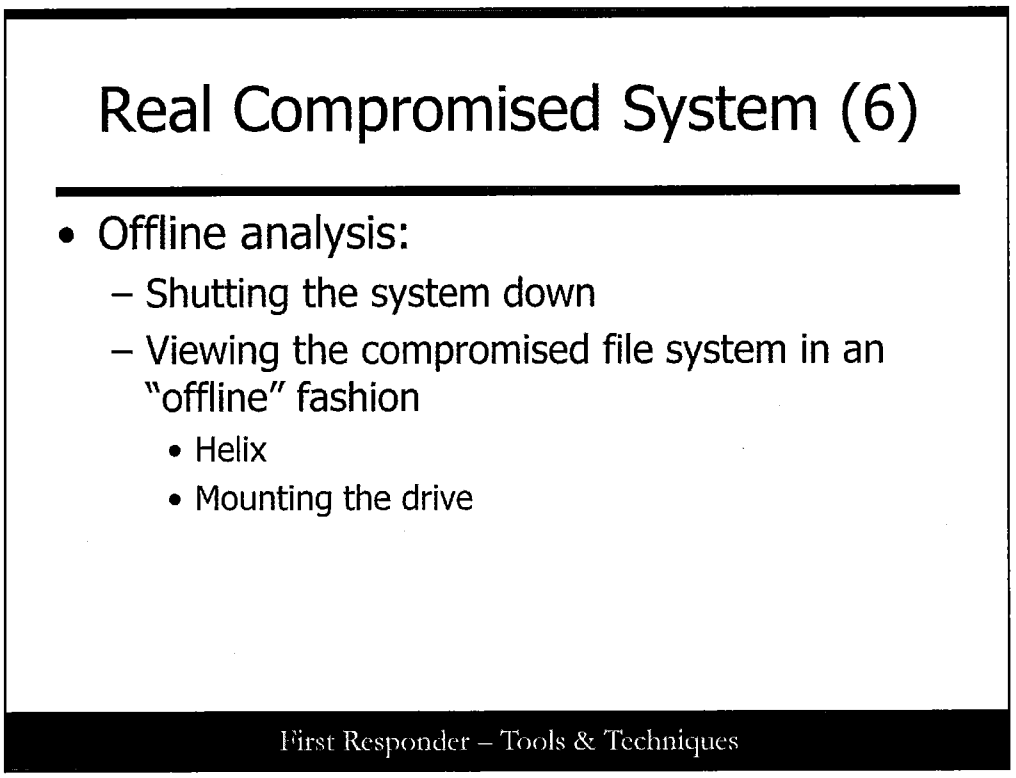
The memory for the system was dumped into a file for later processing as well.

```
ssh -l root 'cat /dev/mem > /incidents/2005.07.18-mem.raw'
```

Other text files were created to record dmesg, system time, ps, and ls. Some of the commands that were run are as follows:

```
ssh -l root 'dmesg > /incidents/2005.07.18-dmesg.txt'
ssh -l root 'date > /incidents/2005.07.18-date.txt'
ssh -l root 'ps -auwwx > /incidents/2005.07.18-ps-auwwx.txt'
ssh -l root 'ls -lactr /etc > /incidents/2005.07.18-ls-lactr-etc.txt'
```

On the Linux laptop before we started to process data, we used the script command. By default this command creates a file called typescript in the current working directory. From the point this command is run until **exit** is typed; it will record every keystroke you type, or every output to the screen that displays. Instead of writing the output of the previous files to a file on the compromised system, changing access information on the file system, you could simply run the command, and script logs the command and the output into a file. This becomes an invaluable tool to use while investigating a compromised system.

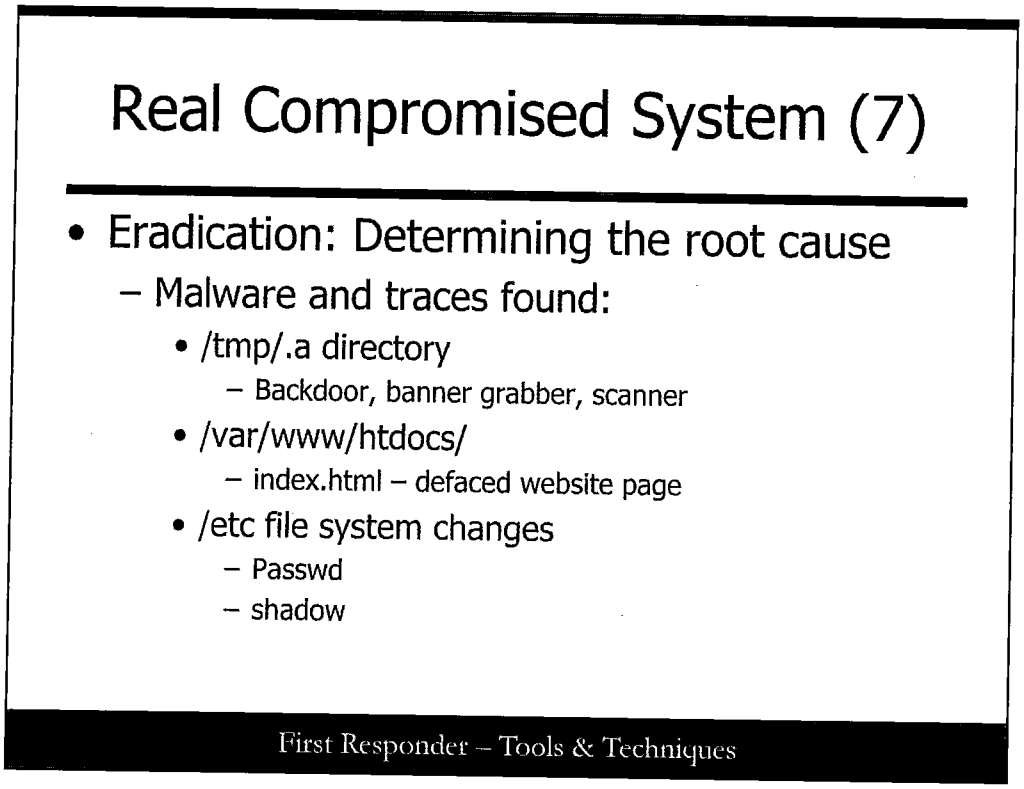


Real Compromised System (6)
All volatile data has been archived or copied to a secure location. The next step was to shut the system down and begin the analysis of the system. There are two methods to shutting down a compromised system. One method is to simply pull the power plug out of the computer, forcing the system to halt then and there. The other method is to shut the system down in a safe fashion ensuring data integrity. If the system has mission critical services running such as a database, it is best to shut those services down prior to pulling the plug. In the case with this compromised system, getting onto the console has been virtually impossible, and therefore the power cable was pulled from the system without any shutdown of services.

The system was then booted using Helix. After the Helix version of Linux was booted, the system disks from the compromised system were mounted read-only. This is an important step, which allows tools that would normally damage file access and modification times to run without the threat of destroying evidence. The command to mount a file system read-only is one of two ways:

```
mount -o ro /device/drive1 /mnt/drive1
mount -r /device/drive1 /mnt/drive1
```

More information about Helix can be found at http://www.e-fense.com/helix/.



Real Compromised System (7)
Findings:

/tmp/.a directory

Local – ssh server backdoor

Grabbb – network app banner grabber

Madscan – scan for DUP broadcast traffic

atac2.tgz

/var/www/htdocs/

index.html – defaced website page

/etc

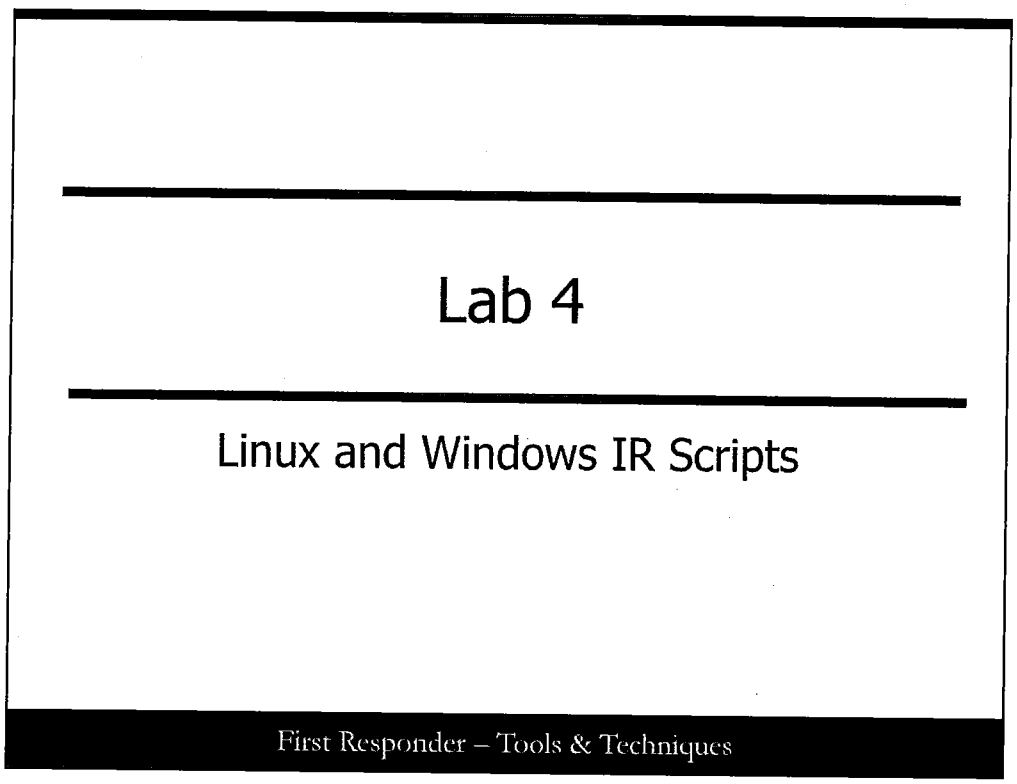
Passwd and shadow

The key indication this system was compromised was the defaced website. The search started in /var/www/htdocs, standard location for an rpm installed version of Apache. Using ls -lactr it was determined that on the morning in question, three files were modified/added to the htdocs directory, index.html, and two image files referenced by the index.html. Also the original index.html had been renamed. There were no further signs of malware or tampering.

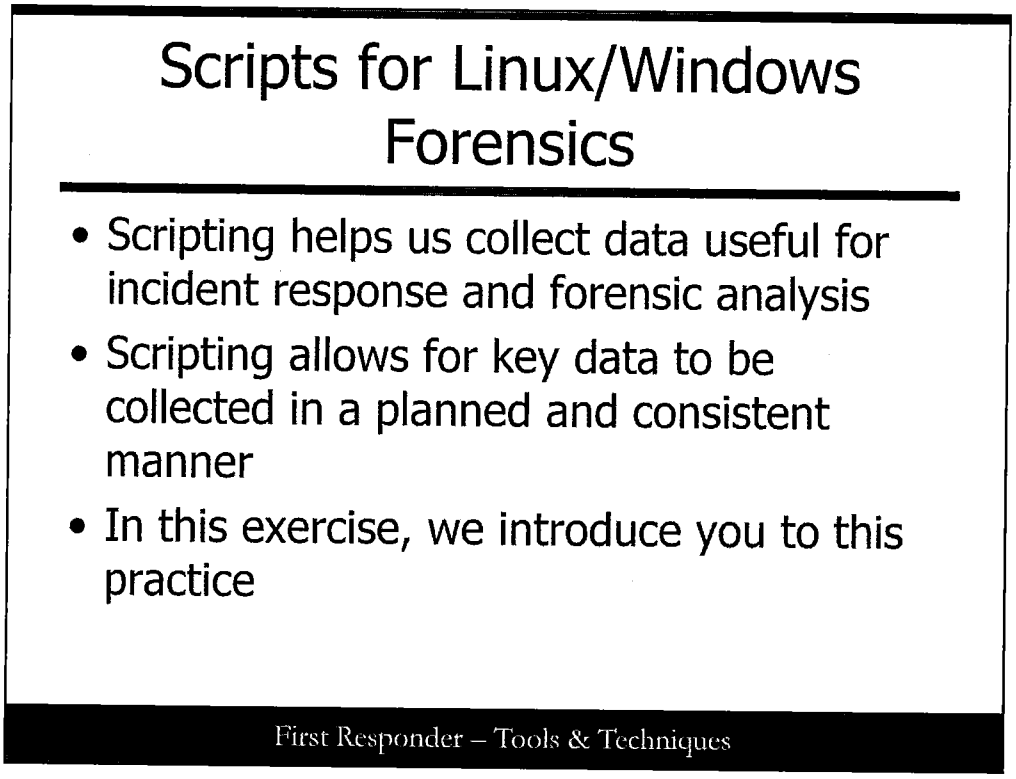
Upon inspecting of /tmp there was found a ".a" directory, which is normally a dead give away that malware was installing into temporary space. In the .a directory there were several files including additional tar-balls. Some of the files found were local, grabbb, madscan, and atac2.tgz. It was determined that local contained an SSH server that upon connection would instantly log you in to the system as root. No references to the exact file could be found on the Internet.

Grabbb was found on the Internet in source format and was determined to be an application network banner collect for SSH, FTP, SMTP, and NNTP. Madscan was also found on the Internet in source format; it uses the ping command to do broadcast pings checking for "DUP" and reporting back the results in a file. Atac2.tgz was not extracted but was found on the system. Upon later examination this tar-ball was found to contain an SSH scanner. There was no direct reference on the Internet that could be found for this file.

One of the most important things that was found was that in /etc, the password and shadow file had been modified on the same day as the html files were installed. A dummy account with user id (uid) 0 was found in the password file. This was most likely installed by the attacker as a means to get back on the system. In some cases the modification date/time on the /etc/shadow and /etc/password files may not be useful. If a user had changed his password after the compromise, then this information may have not been noticed. This gets back to preparation. Knowing the types of accounts that should be on a system can still be used to weird out the accounts added by attackers.

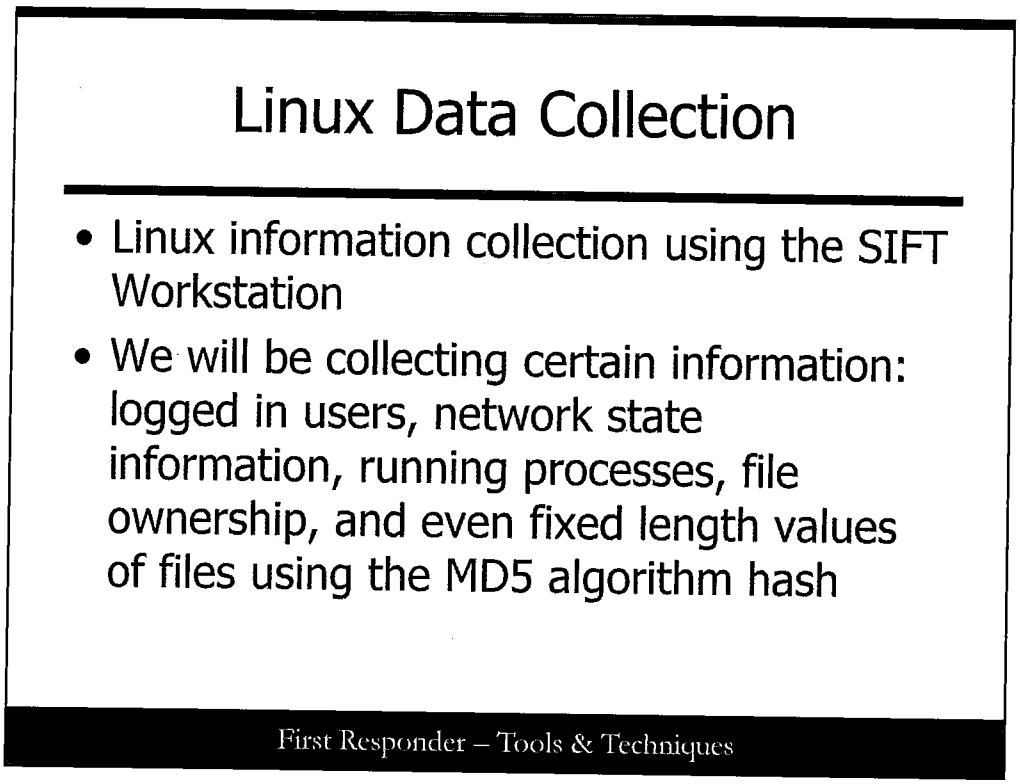


This page intentionally left blank.



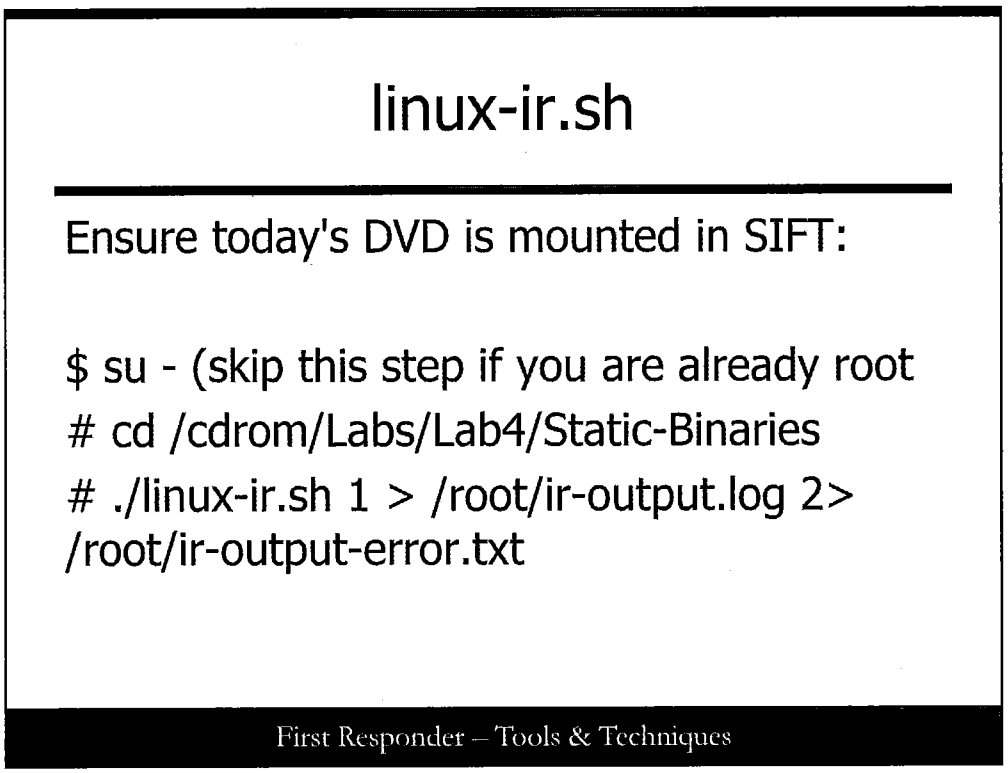
Scripts for Linux / Windows Forensics

In this lab, we perform a great deal of data collection work here but using scripting to collect data useful to an incident response capacity in a consistent fashion. That is, always part of the plan is collecting certain information, whenever the situation allows. The purpose for this next exercise is to introduce you to a scripted data collection, which can be used in Linux environments, as well as those on a MS Windows platform.



Linux Data Collection

Harvesting Linux information from the SIFT Workstation happens first, and we set out collecting certain information with emphasis on the following: logged in users, network state information, running processes, file ownership, and even fixed length values of files using the MD5 algorithm hash.



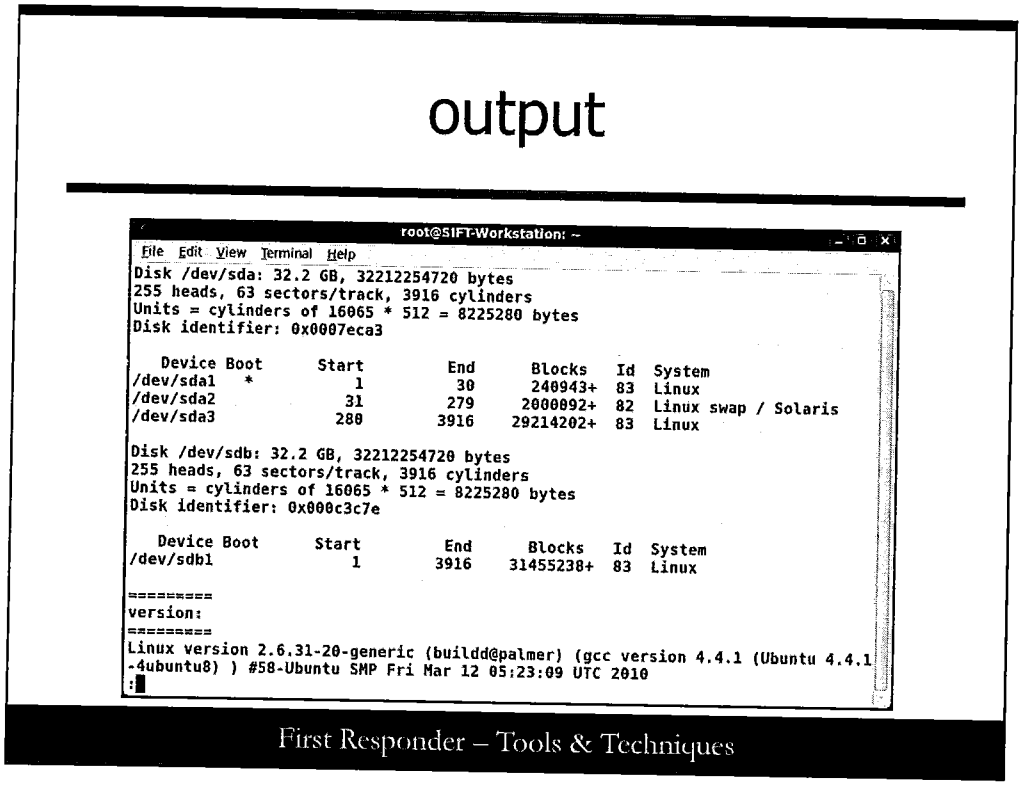
linux-ir.sh
When you run the linux-ir.sh script, you get a great deal of output. If you were prepared you can either redirect it to a file that will store the output, or you can pipe it into netcat to send it over to another system that is waiting to receive the data.

If you navigate to the */Lab/Lab4/Static-Binaries* folder, you can find the linux-ir.sh shell script ready to be used to collect data.

In addition, if you navigate into the */Lab/Lab4/Static-Binaries/linux_x86* folder, you can find a number of programs that were included. Many of these files have a name that looks familiar but possess a t_ in front of them to signify they are trusted and will not inadvertently run an executable on the system, to the extent possible. The goal here is making the lightest touch possible to the system while you are collecting the information.

It's good to know that many of the files are statically compiled, so they do not depend on other library files that normally a host system would be glad to offer up, but also may be suspect if a compromise is afoot.

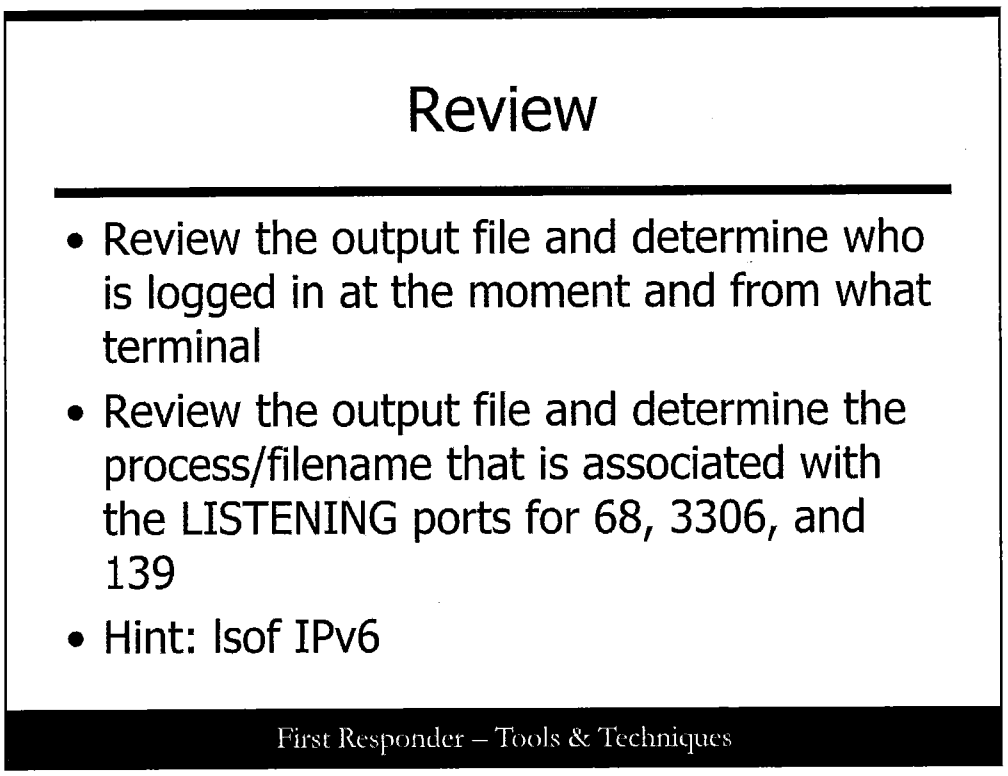
Mount the Course DVD if you have not already done so. Open up a terminal window if you don't have one open already.
cd */Lab/Lab4/Static-Binaries*
Next, run the *linux-ir.sh* command but redirect the output to an output file in our ~/ (home) folder so that you can inspect it at your leisure. We also redirect standard error (denoted by 2>) to provide a means for the errors that might occur to go to a file for inspection later. Doing this on one single command line is allowed and recommended.



output

Typically, the command takes a few minutes to run completely and capture all the information. When it's done it returns you to a prompt. Use either the pagin commands *more*, or *less* to review the file inside your command prompt, or you can use the GUI text file editor, which is part of the SIFT Workstation. (It's found under Applications → Accessories → gedit Text Editor.)

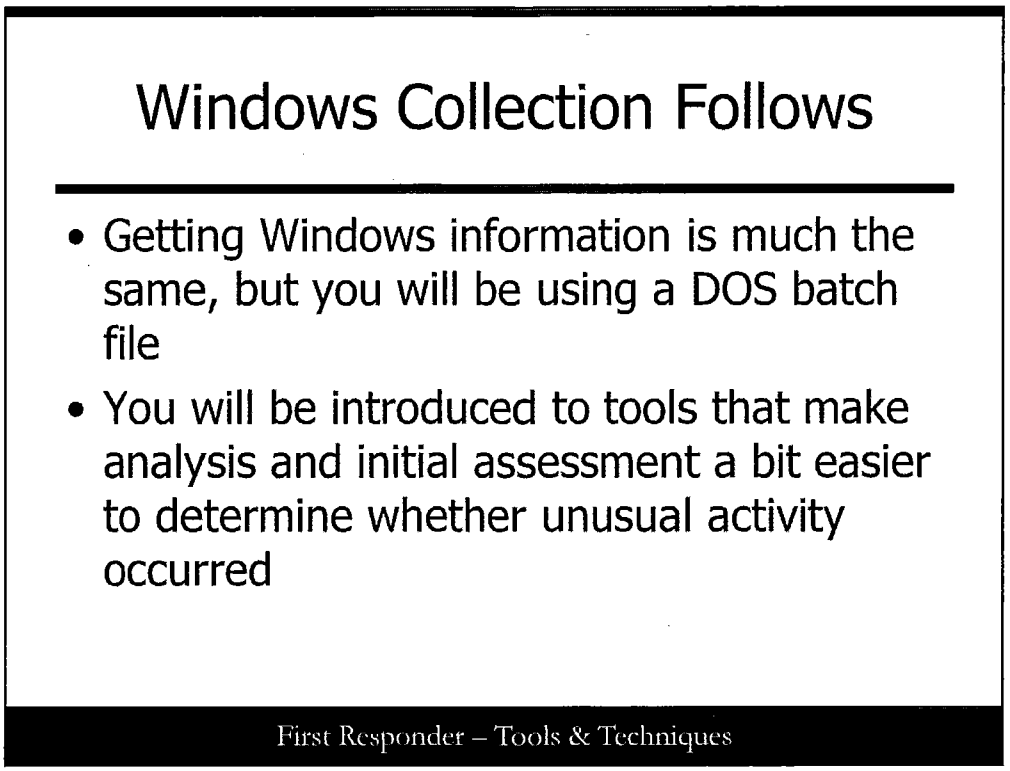
This screen shot is what you can expect to see when you review the file; it should look something quite similar to this file, but keep in mind there are many, many lines stored in the output of the report.



Review the output file and specifically the section titled "Currently logged in users." Within it you should spot who is logged in at the moment and from what terminal. Take the space in the lines below and write down who is logged in and from what terminal ID.

Review the output file and specifically the section titled "netstat output(current connections)." Within it you need to write down the process/filename that is associated with the LISTENING ports for 68, 3306, 139. Hint: lsof IPv6

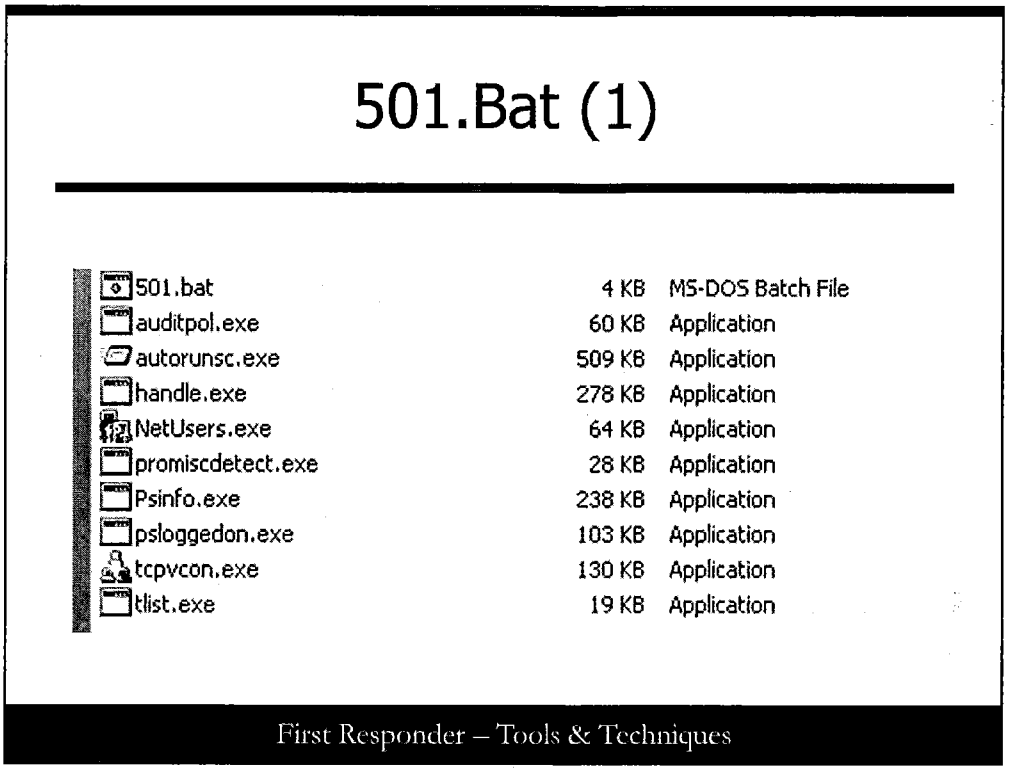
You are all done with the Linux part of this exercise.



Windows Collection Follows

Getting Windows information is much the same but you use a DOS Batch file.

You will be introduced to tools that make analysis and initial assessment whether unusual activity occurred a bit easier to determine.



501.bat (1)

Next, we run a similar script within an MS Windows environment. Use your host Windows workstation (it does not have to be XP) or VMware Virtual Machine with Windows running on it.

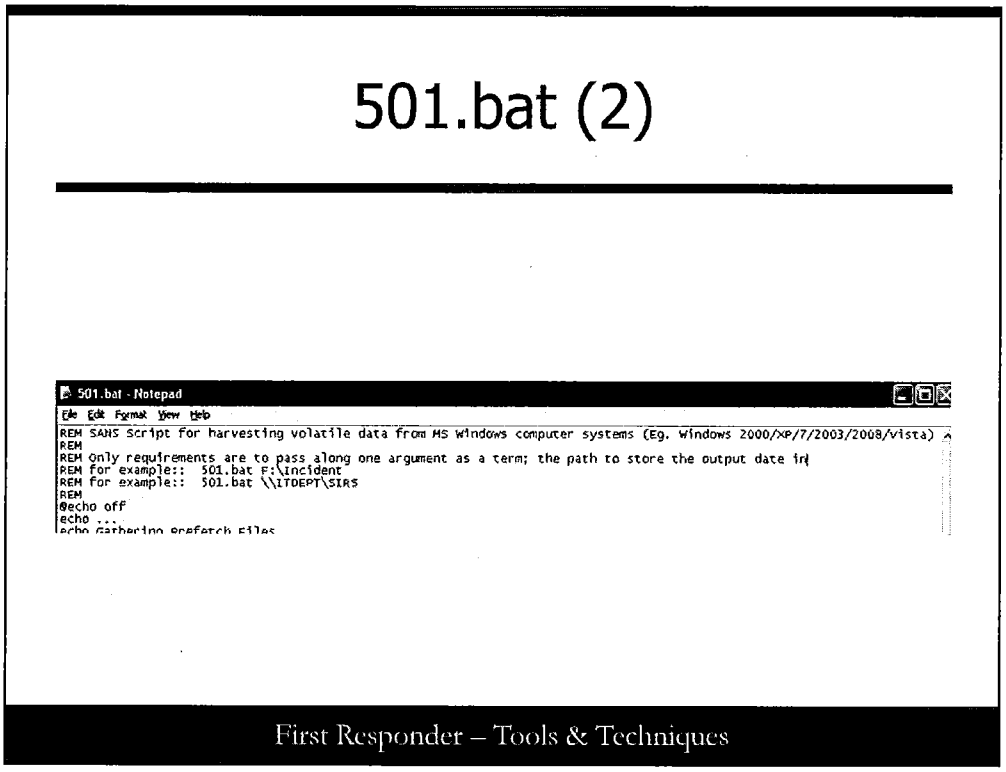
The 501.bat batch file is located in the *D:\Labs\Lab4* folder along with several files that are DOS-based static executable files. To the extent possible, we want you to copy them either to a writable CD/DVD or a USB removable thumb-drive and run them.

Run cmd.exe as Administrator

mkdir c:\Lab4

copy d:\Labs\Lab4* c:\Lab4

cd c:\Lab4



501.bat (2)

When running the 501.bat batch file, we expect you to include a term along with the batch file to designate the output folder that the data will go to. Otherwise, everything will be dumped into the file that is the present working directory, which could be the CD/DVD-ROM drive, which may not allow writing to occur. For instance, if you run it from within your CD/DVD-ROM drive on D:\ and you want to send the output to a directory (call it SIRS; completely fictional and notional) you've created on a network share drive \\ITDEPT, then this is how you would be expected to invoke the command:

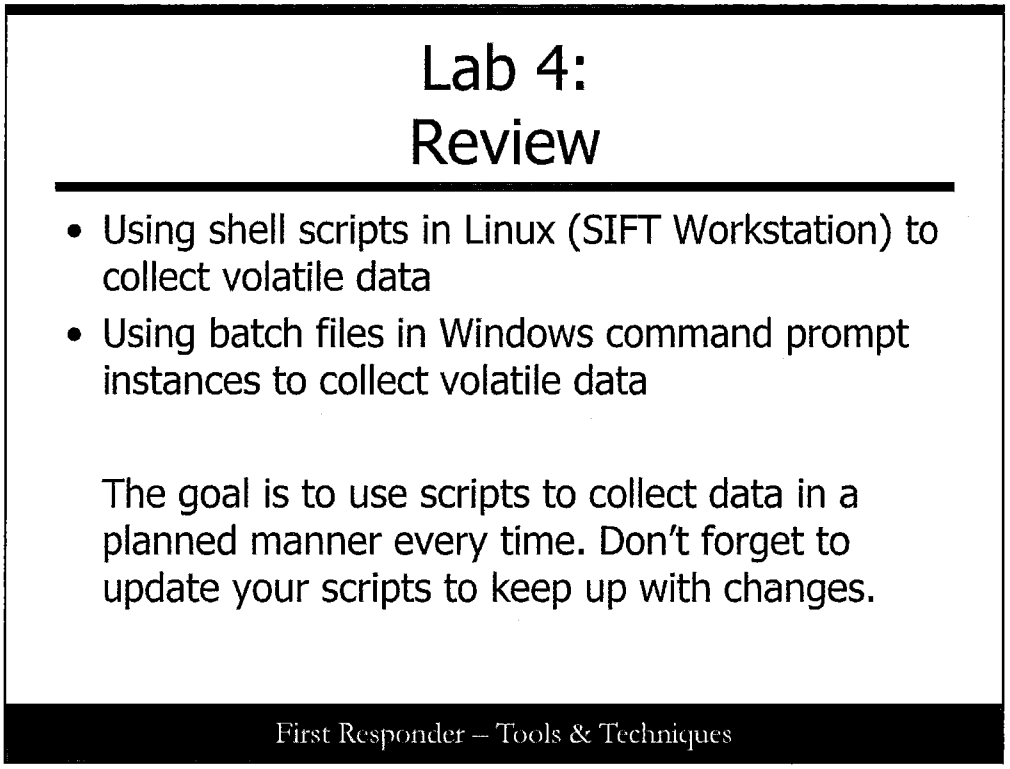
D:\501.bat \\ITDEPT\SIRS

Give it a run. For example, you could type:

501.bat c:\Lab4

type "dir" when complete

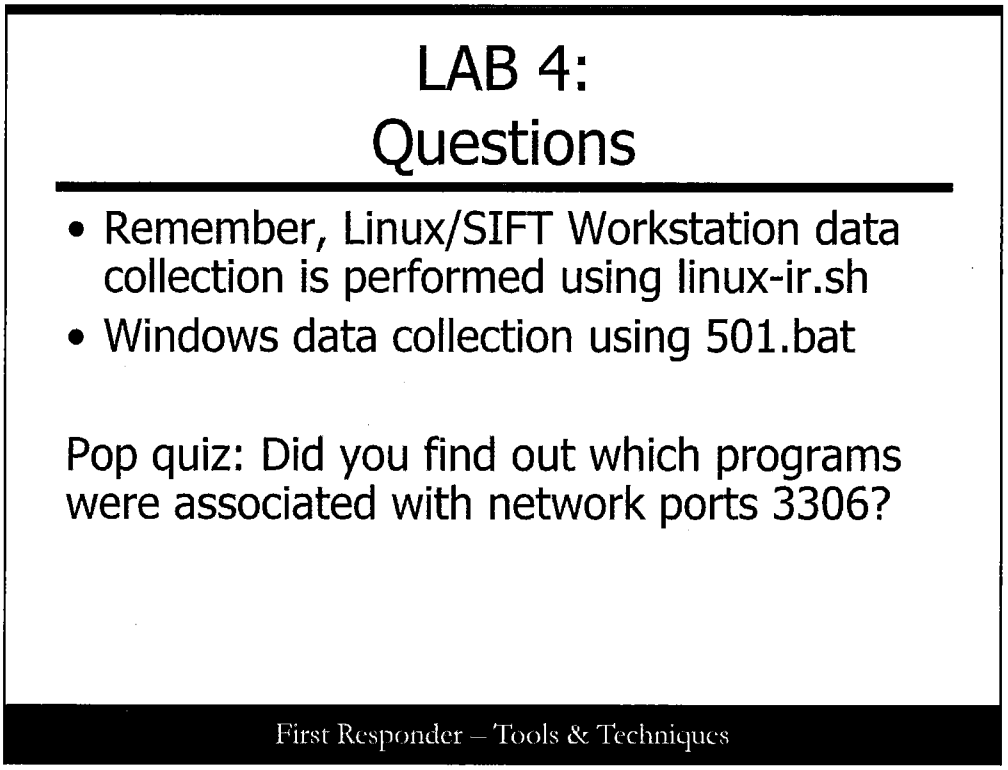
Then inspect the contents of the designated output directory. You should also notice that the collected data is usually stored into .LOG files where the .TXT is reserved for files that end with -err.txt in their names. Most the time you expect the sizes of these particular sets of files to be of a zero length as there may not have been errors encountered, but when errors do occur, they can help you troubleshoot or determine the cause.



Lab 4: Review

Review what we set out to learn from this exercise. We can use shell scripts in a Linux environment (which also includes the SIFT Workstation so that we can maintain our skills) to collect volatile data. In Windows environments, we can use batch-files in a DOS command prompt to collect data.

The goal is to use scripts to collect data in a planned manner every time. Don't forget to update your scripts to keep up with changes. When you learn new or improved methods for collecting information in a forensically sound manner, test them; then once they pass muster, update your scripts and teach the use to others who use the scripts.

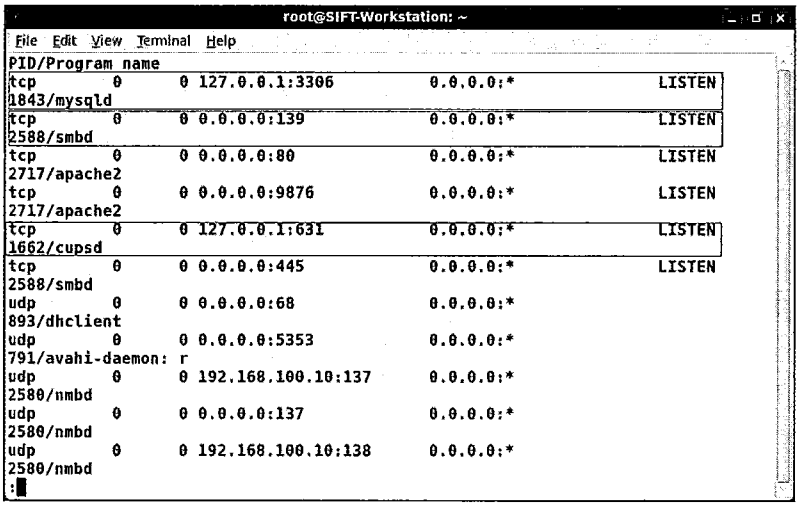


LAB 4: Questions

Review questions about IR Collections Scripts for both Linux and Windows. We remind you that the *linux-ir.sh* script is used to collect data from Linux systems. It operates in a shell environment and expects you to type in commands.

For the Windows users out there, we make use of a DOS batch file known as *501.bat,* which is used to collect data from Windows systems.

Review Answer



First Responder – Tools & Techniques

Review Answer

mysqld running on port 3306

smbd running on 139