Chapter 41

Cyber Forensics and Incidence Response

Cem Gurkok
Terremark Worldwide, Inc., Miami, FL, United States

1. INTRODUCTION TO CYBER FORENSICS

Cyber forensics and incident response go hand in hand. Cyber forensics reduces the occurrence of security incidents by analyzing the incident to understand, mitigate, and provide feedback to the actors involved. To perform incident response and related activities, organizations should establish an incident plan, a computer security incident response team (CSIRT) or a computer emergency response team (CERT) to execute the plan and associated protocols.

Responding to Incidents

In an organization there is a daily occurrence of events within the IT infrastructure, but not all of these events qualify as incidents. It is important for the incident response team to be able to distinguish the difference between events and incidents. Generally, incidents are events that violate an organization's security policies, end user agreements, or terms of use. SANS (sans.org) defines an incident as an adverse event in an information system or network, or the threat of an occurrence of such an event. Denial-of-service (DoS) attacks, unauthorized probing, unauthorized entry, destruction or theft of data, and changes to firmware or operating systems (OSs) can be considered incidents.

Generally, incident response handling is composed of incident reporting, incident analysis, and incident response. Incident reporting takes place when a report or indications of an event is sent to the incident response team. The team then performs an incident analysis by examining the report, available information, evidence, or artifacts related to the event to qualify the event as an incident, correlate the data, and assess the extent of damage, source, and plan potential solutions. Once the analysis is over, the team responds to mitigate the incident by containing and eradicating the incident. This is followed by the creation of a detailed report about the incident.

Applying Forensic Analysis Skills

Forensic analysis is usually applied to determine who, what, when, where, how, and why an incident took place. The analysis may include investigating crimes and inappropriate behavior, reconstructing computer security incidents, troubleshooting operational problems, supporting due diligence for audit record maintenance, and recovering from accidental system damage. The incident response team should be trained and prepared to be able to collect and analyze the related evidence to answer these questions. Data collection is a very important aspect of incident response since evidence needs to be collected in a forensically sound manner to protect its integrity and confidentiality. The incident responder needs to have the necessary skills and experience to be able to meet the collection requirements.

Forensic analysis is the process where the collected data is reviewed and scrutinized for the lowest level of evidence (deleted data in slack space) it can offer. The analysis may involve extracting email attachments, building timelines based on file times, review of browser history, in-memory artifact review, decryption of encrypted data, and malware reverse engineering. Once the analysis is complete, the incident responder will produce a report describing all the steps taken starting from the initial incident report until the end of the analysis. One of the most important skills a forensic analyst can have is note-taking and logging, which becomes very important during the reporting phase and, if it ever comes to it, in court. These considerations related to forensics should be addressed in organizational policies. The forensic policy should clearly define the responsibilities and roles of the actors involved. The policies should also

Computer and Information Security Handbook. http://dx.doi.org/10.1016/B978-0-12-803843-7.00041-7

address the types of activities that should be undertaking under certain circumstances and the handling of sensitive information.

Distinguishing Between Unpermitted Corporate and Criminal Activity

We previously defined incidents as events that are not permitted by a certain organization's policies. The incident response team should also be aware of several federal laws that can help them to identify criminal activity to ensure that the team does not commit a crime while responding to the incident. Some of these federal laws include:

- The Foreign Intelligence Surveillance Act of 1978
- The Privacy Protection Act of 1980
- The Computer Fraud and Abuse Act of 1984
- The Electronic Communications Privacy Act of 1986
- Health Insurance Portability and Accountability Act of 1996 (HIPAA)
- Identity Theft and Assumption Deterrence Act of 1998
- The USA PATRIOT Act of 2001

When an incident response team comes across incidents relevant to these laws, they should consult with their legal team. They should also contact appropriate law enforcement agencies.

2. HANDLING PRELIMINARY INVESTIGATIONS

An organization should be prepared beforehand to properly respond to incidents and mitigate them in the shortest time possible. An incident response plan should be developed by the organization and tested on a regular basis. The plan should be written in an easily understood and implemented fashion. The incident response team and related staff should also be trained on an ongoing basis to keep them up to date with the incident response plan, latest threats, and defense techniques.

Planning for Incident Response

Organizations should be prepared for incidents by identifying corporate risks, preparing hosts and the network for containment and eradication of threats, establishing policies and procedures that facilitate the accomplishment of incident response goals, and creating an incident response team and an incident response toolkit to be used by the incident response team.

Communicating with Site Personnel

All departments and staff that have a part in an incident response should be aware of the incident response plan and should be regularly trained on its content and implementation. The plan should include the mode of communication with the site personnel. The site personnel should clearly log all activity and communication, including the date and time in a central repository that is backed up regularly. This information should be reviewed by all of the incident response team members to assure all players are on the same page. Continuity and the distribution of information within the team is critical in the swift mitigation of an incident. An incident response team leader should be assigned to an incident and should make sure all team members are well informed and acting in a coordinated fashion.

Knowing Your Organization's Policies

An organization's policies will have an impact on how incidents are handled. These policies are usually very comprehensive and effective computer forensics policies that include considerations, such as contacting law enforcement, performing monitoring, and conducting regular reviews of forensic policies, guidelines, and procedures. Banks, insurance companies, law firms, governments, and health care institutions have such policies. Generally policies should allow the incident response team to monitor systems and networks and perform investigations for reasons described in the policies. Policies may be updated frequently to keep up with the changes to laws and regulations, court rulings, and jurisdictions.

Forensics policies define the roles and responsibilities of the staff involved including users, incident handlers, and IT staff. The policy indicates when to contact internal teams or reach out to external organizations. It should also discuss how to handle issues arising from jurisdictional conflicts. Policies also discuss the valid use of antiforensics tools and techniques (sanitation and privacy versus malicious use, such as hiding evidence). How to maintain the confidentiality of data and the retention time of the data is also governed by organizational policies.

Minimizing Impact on Your Organization

The goals of incident response include minimizing disruption to the computer and network operations, and minimizing the exposure and compromise of sensitive data. To be able to meet these goals, incident response preparedness, planning, and proper execution following related policies is crucial. Incident response teams should minimize the down times of business critical systems once the evidence has been gathered and the systems have been cleared of the effects of the incident. Incident response teams should also identify an organization's risks and work with appropriate teams to continuously test and eliminate any vulnerability. Red team−blue team exercises, where a team plays the role of malicious people and the other team

as incident responders, can provide good training for the staff and expose previously unknown risks and vulnerabilities. To minimize the impact of incidents, organizations should also establish and enforce security policies and procedures, gain management support for security policies and incident response, keep systems updated and patched, train IT staff and end users, implement a strong credential policy, monitor network traffic and system logs, implement and routinely test a backup policy.

Identifying the Incident Life Cycle

SANS (sans.org) defines the phases of the incident life cycle in Fig. 41.1.

Preparation

It's a matter of when, rather than if, an incident will happen. Therefore, it has become a top priority for an organization to be prepared for an incident. To be prepared, an organization must establish security plans and controls, make sure these plans and controls are continuously reviewed and updated to keep up with the evolving threats, and make sure they are enforced in case of an incident. Organizations should be prepared to act swiftly to minimize the impact of any incident to maintain business continuity. Incident response teams should continuously train, test, and update the incident response plan to keep their skills honed.

Detection, Collection, and Analysis

The detection of an incident involves the observance and reporting of security or IT department staff members, customers of irregularities, or suspicious activities. Once an event has been reported and escalated to the incident response team, the event is evaluated to determine if it warrants classification as an incident. If the event has been classified as an incident, the incident response team should move in to perform data collection on the affected systems that will later be used for analysis. During collection, it is important to work in accordance with the organization's policies and procedures and preserve a valid chain of custody. The person involved in collecting the data should make sure that the integrity of the data is maintained on both the original and working copies of the evidence. Once the relevant information has been captured, the incident response team should analyze the data to determine who, what, when, where, how, and why an incident took place.

FIGURE 41.1 Incident response life cycle.

Containment, Eradication, and Recovery

Once the involved systems and offending vectors have been analyzed, the incident response team should move in to contain the problem and eradicate it. It is crucial to contain an incident as fast as possible to minimize its impact on the business. This can be as easy as disconnecting the system from the network or as hard as isolating a whole server farm from the production environment. Containment and eradication should strive to protect service integrity, sensitive data, hardware, and software. The recovery phase depends on the extent of the incidence. For example, an intrusion that was detected while it was affecting a single user is easier to recover from in comparison to an intrusion where the lateral movement of the intruder is extensive. Most of the time, recovery involves backing up the unaffected data to use on the new systems. OSs and applications are usually installed fresh to avoid any type of contamination.

Postincident Activity

The postincident phase involves documenting, reporting, and reviewing the incident. Documentation actually starts as soon as an event has been classified as an incident. The report should include all of the documentation compiled during the incident, the analysis methods and techniques, and all other findings. The person writing the report should keep in mind that the report might someday be used by law enforcement or in court. Finally, the incident response team should go over the report with the IT department and other involved parties to discuss how to improve he infrastructure to prevent similar incidents.

Capturing Volatile Information

Computer systems contain volatile data that is temporarily available either till a process exits or a system is shutdown. Therefore, it is important to capture this data before making any physical or logical changes to the system to avoid tampering with evidence. Many incident responders have destroyed memory-only resident artifacts by shutting down a system in the name of containment.

Volatile data is available as system memory (including slack and free space), network configuration, network connections and sockets, running processes, open files,

login sessions, and OS time. System memory can be captured by using sampling tools (MoonSols Windows Memory Toolkit, GMG Systems' KnTDD) as a file and analyzed with the Volatility Framework to obtain the volatile data previously mentioned. The volatile data can also be captured individually with tools that are specific for each data type. The Microsoft Windows Sysinternals suite provides an extensive set of tools that can capture volatile data, such as login sessions, Registry, process information, service information, shares, and loaded dynamic-link libraries (DLLs).

3. CONTROLLING AN INVESTIGATION

To control an investigation, the incident response team should have a forensics investigation plan, a forensics toolkit, and documented methods to secure the affected environment. An investigator should always keep in mind that the evidence collected, and the analysis performed might be presented in court or used by law enforcement. Related documentation should be detailed and contain dates and times for each activity performed. To avoid challenges to the authenticity of evidence investigators should be able to secure the suspect infrastructure, log all activity, and maintain a chain of custody.

Collecting Digital Evidence

It is important to an investigator to preserve data related to an incident as soon as possible to avoid the rapid degradation or loss of data in digital environments. Once the affected systems have been determined, volatile data should be captured immediately followed by nonvolatile data, such as system users and groups, configuration files, password files and caches, scheduled jobs, system logs, application logs, command history, recently accessed files, executable files, data files, swap files, dump files, security software logs, hibernation files, temporary files, and complete file listing with times.

Chain of Custody and Process Integrity

The incident response team should be committed to collect and preserve evidence using methods that can support future legal or organizational proceedings. A clearly defined chain of custody is necessary to avoid allegations of tampering evidence. To accomplish this task the team should keep a log of every entity who had physical custody of the evidence, document all of the actions performed on the evidence with the related date and time, make a working copy of the evidence for analysis, verify the integrity of the original and working copy, and store the evidence in secured location when not in use [2]. Also before touching a physical system, the investigator should take a photograph of it. To ensure the integrity of the process a detailed log should be kept of all the collection steps, information about every tool used in the incident response process.

Advantages of Having a Forensics Analysis Team

Forensic analysis is usually thought in the context of crime investigations. Nowadays, due to the increase in computer-related malicious activity and growing digital infrastructure, forensic analysis is involved in incident response, operational troubleshooting, log monitoring, data recovery, data acquisition, audits, and regulatory compliance. Therefore, organizations can no longer rely on law enforcement due to resource and jurisdictional limitations. A violation of organizational policies and procedures might not concern law enforcement leaving the organization to its own devices. It has become evident to organizations that maintaining capabilities to perform forensic analysis has become a business requirement to satisfy organizational and customer needs. While it may make sense for some organizations to maintain an internal team of forensic analysts, some might find it more beneficial to hire outside parties to carry out this function. Organizations should take cost, response time, and data sensitivity into consideration before making this decision [2]. Keeping an internal forensic analysis team might reduce cost depending on the scale of the incident, provide faster response due to familiarity with the infrastructure, and prevent sensitive data from being viewed by third parties.

Legal Aspects of Acquiring Evidence: Securing and Documenting the Scene

Securing the physical scene and documenting it should be one of the first steps an incident responder should take. This involves photographing the system setup, cabling, general area, collecting and documenting all cables and attached devices, write-protecting all media, using antistatic packaging for transportation, maintaining proper temperature for stored devices, avoiding exposure to excessive electromagnetic fields, and logging all access to the area. The incident response team should keep an inventory of evidence-handling supplies (chain of custody forms, notebooks, evidence storage bags, evidence tape), blank media, backup devices, and forensics workstations.

Processing and Logging Evidence

The goal of an investigation is to collect and preserve evidence that can be used for internal proceedings or court of law. Investigators should be able prove that the evidence has not been tampered with. To be able to accomplish this the incident response team members should receive training specifically addressing these issues and should practice these skills on an ongoing basis to stay sharp.

To properly process and log evidence, investigators should keep the evidence within a secured and controlled environment where all access is logged, and should document the collected evidence and its circulation among investigative entities. We cannot stress how important it is to associate each activity with a date and time.

4. CONDUCTING DISC-BASED ANALYSIS

To be able to process evidence in a manner that is admissible in a court of law, a lab and accompanying procedures should be established. This will ensure that the data integrity is not breached and the data remains confidential: in other words, the evidence remains forensically sound.

Forensics Lab Operations

To ensure forensic soundness, an investigator's process needs to be reliable, repeatable, and documented. To have a controlled and secure environment for the investigator to follow these steps, a forensic lab becomes a necessity.

The lab should be established in a physically secure building that is monitored 24/7, should have a dedicated staff, should have regularly upgraded and updated workstations dedicated to forensic analysis with related software installed, and should have a disaster recovery plan in place.

Acquiring a Bit-Stream Image

Acquiring a bit-stream image involves producing a bit-by-bit copy of a hard drive on a separate storage device. By creating an exact copy of a hard drive, an investigator preserves all data on a disc, including currently unused and partially overwritten sectors. The imaging process should not alter the original hard drive to preserve the copy's admissibility as evidence. Selecting a proper imaging tool is crucial to produce a forensically sound copy. The National Institute of Standards and Technology (NIST) lists the requirements for a drive imaging tool as follows [2]:

- The tool shall make a bit-stream duplicate or an image of an original disc or a disc partition on fixed or removable media.
- The tool shall not alter the original disc.
- The tool shall be able to access both integrated development environment (IDE) and small computer standard interface (SCSI) discs.
- The tool shall be able to verify the integrity of a disc image file.
- The tool shall log input/output (I/O) errors.
- The tool's documentation shall be correct.

The imaging of a hard drive can be performed using specialized hardware tools or by using a combination of computers and software.

Specialized Hardware

The Image MASSter Solo series hard drive duplicators generally support serial advanced technology attachment, IDE, Universal Serial Bus (USB), external serial advanced technology attachment, universal serial advanced technology attachment, serial-attached SCSI hard drives and flash memory devices. They can hash the disc images besides providing write-blocking to ensure the integrity of the copies. The imaging process can be either disc-to-disc or disc-to-file.

The Digital Intelligence Forensic Duplicator units have the same properties as the Image MASSter Solo series. But, they provide access to different hard drive formats through their protocol modules.

Software: Linux

The dd or dcfldd has been fully tested and vetted by NIST as a forensic imaging tool. It is a freeware utility for any Linux-based system and can copy every sector of hard drives. The software program dcfldd is dd-based and enhances dd's output by providing status and time-to-completion output as the disc gets imaged and can split the output to smaller chunks. It can also hash the output to ensure data integrity. The following command, as seen in Table 41.1, will read block sizes of 512 bytes, produce 2 GB chunks of a disc device defined as /dev/sdb, and will calculate the MD5 hashes every 2 GB to ensure integrity. The hash values will be written to a file named md5.og. In the event of a read error, dcfldd will write zeroes in the copy.

Windows The AccessData Forensic Toolkit (FTK) Imager tool is a commercial disc-imaging tool distributed by AccessData. FTK supports storage of disc images in EnCase's file format, as well as in bit-by-bit (dd) format.

TABLE 41.1 Creating an Image of a Drive

```
$ dcfldd if=/dev/sdb hash=md5 hashwindow=2G md5log=md5.log  hashconv=after bs=512

conv=noerror,sync split=2G splitformat=aa of=driveimage.dd
```

On the other hand, the Guidance EnCase tool is a commercial disc imaging tool distributed by Guidance Software. Disc images are stored in the proprietary EnCase Evidence File Format, which contains compressed data prefixed with case metadata and contains hashes if the image data.

Enabling a Write Blocker

Write blockers are hardware- or software-based tools that allow the acquisition of hard drive images while preventing any data from being written to the source hard drive, therefore ensuring the integrity of the data involved. Write blockers can do this by only allowing read commands to pass through by blocking write commands or by letting only specific commands through. While copying data with a hardware write blocker, the source and destination drives should both be connected to the write-blocking device and in case of a software blocker, the blocking software should be activated first before copying [1]. After imaging is performed with a write blocker, calculating the hashes of both the source and destination images is essential to ensure data integrity. Some hardware write blockers that are used in the industry is as follows:

- Tableau Forensic Bridges
- WiebeTech WriteBlocker

Establishing a Baseline

It is important to maintain the integrity of the data being analyzed throughout the investigation. When dealing with disc drives, to maintain integrity, calculating the hashes of the analyzed images becomes crucial. Before copying or performing any analysis, the investigator should take a baseline hash of the original drives involved. The hash could be either just MD5 or a combination of MD5, SHA-1, and SHA-512. The baseline hash can be compared with hashes of any copies that are made thereafter for analysis or backup to ensure that the integrity of the evidence is maintained.

Physically Protecting the Media

After making copies of the original evidence hard drives, they should be stored in a physically secure location, such as a safe in a secured storage facility. These drives could be used as evidence in the event of prosecution. The chain of custody should also be maintained by labeling the evidence and keeping logs of date, time, and persons the evidence has come in contact with. During transportation, the hard drives should be placed in antistatic bags and should not be exposed to harsh environmental conditions. If possible, photographs of the evidence should be taken whenever they are processed, starting from the original location to the image acquisition stages.

Disc Structure and Recovery Techniques

Once a forensically sound copy has been made of the evidence, we can proceed to analyzing its contents. There are different kinds of storage media: hard disc drives (HDD), solid state drives (SSD), digital video discs (DVD), compact discs (CD), flash memory, and other kinds. An investigator needs to be mindful about how each media stores data differently. For example, while data in the unused space on an HDD is stored as long as new data is not written, the data in the unused space of an SSD is destroyed within minutes of switching it on. This difference in retaining data makes it difficult to obtain a forensically sound image and recovering data (see checklist: "An Agenda for Action for Data Recovery").

An Agenda for Action for Data Recovery

The cyber forensic specialist should ensure that the following provisional list of actions for data recovery are adhered to (check all tasks completed):

_____**1.** Make sure you are ready and have procedures in place for disasters like floods, tornadoes, earthquakes, and terrorism when they strike.

_____**2.** Make sure you are ready and have a plan in place to take periodic image copies and send them offsite.

_____**3.** Perform change accumulation to reduce the number of logs required as input to the recovery, which saves time at the recovery site. However, performing this step consumes resources at the home site.

_____**4.** Evaluate your environment to decide how to handle the change accumulation question/problem in action/task #3.

_____**5.** Make sure you have procedures in place to implement your plan.

_____**6.** Check your assets to make sure they're ready as part of your plan.

_____**7.** Make sure you build your recovery job control language (JCL) correctly. JCL is tricky, and you need to get it exactly right. Data integrity and your business rely on this task.

_____**8.** Make sure you clean your remote console data sets. It can take hours if done manually, and it's an error-prone process. When your system is down, can you afford to make mistakes with this key resource?

_____**9.** Make sure you test your plan. There's a lot to think about. In the real world, there's much more.

_____**10.** Make sure your plan works before you are required to use it!

_____**11.** Make sure that you have procedures in place to deal with issues of increased availability, shrinking expertise, and growing complexity, failures of many types, and the costs of data management and downtime.

Disc Geometry Components

With regards to HDD geometry: The surface of each HDD platter is arranged in concentric magnetic tracks on each side. To make accessing data more efficient, each track is divided into addressable sectors or blocks as seen in Fig. 41.2. This organization is known as formatting. Sectors typically contain 512 bytes or 2048 bytes of data in addition to the address information. Newer HDDs use 4096 byte sectors. The HDD controller uses the format and address information to locate the specific data processed by the OS.

Now, with regards to SSD geometry: Compared to HDDs, SSDs store data in 512 kilobyte sectors or blocks, which are in turn divided into 4096 byte long pages. These structures are located in arrays of NAND (Negated AND or NOT AND) transistors.

Inspecting Windows File System Architectures

File systems, including Windows, can be defined in six layers: physical (absolute sectors), data classification (partitions), allocation units (clusters), storage space management (file allocation table [FAT] or master file table [MFT]), information classification (folders), and application level storage (files). Knowing these layers will guide the investigator as to what tool is needed to extract information from the file system. Windows file systems have gone through an evolution starting from FAT and continuing to New Technology File System (NTFS).

File Allocation Table (FAT)

FAT has been available widely on Windows systems starting with the MS-DOS OS. This file system has incarnations, such as FAT12, FAT16, FAT32, and exFAT. The volume is organized into specific sized chunks based on the version numbering of the file system. FAT12 has a cluster size of 512 bytes to 8 kilobytes whereas FAT16 has cluster sizes ranging from 512 bytes to 64 kilobytes. FAT32 file systems can support disc sizes up to two terabytes using cluster sizes ranging from 512 bytes to 32 kilobytes. FAT file systems begin with the boot sector and proceed with FAT areas 1 and 2, the root directory, files, and other directories. FAT provides a table to the OS as to which cluster in the volume is used for a file or folder. In a FAT file system, file deletion is accomplished by overwriting the first character of the object's name with 0xE5 or 0x00 and by setting the table entry of the related clusters to zero. FAT file times are stored by using the local system's time information.

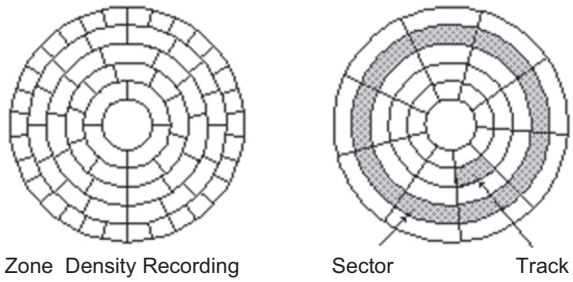

FIGURE 41.2 Physical structure of a drive.

New Technology File System (NTFS)

As the name suggests, NTFS was developed to overcome the limitations inherent in the FAT file system. These limitations were the lack of access control lists (ACLs) on file system objects, journaling, and compression, encryption, named streams, rich metadata, and many other features. The journaling features of NTFS make it capable of recovering itself by automatically restoring the consistency of the file system when an error takes place [2]. It also should be noted that NTFS file times are stored in the Universal Coordinated Time (UTC) compared to FAT where the OS's local time is used. There are mainly two artifacts in NTFS that interests a forensics investigator: MFT and alternate data stream (ADS).

Master File Table (MFT)

MFT or $MFT can be considered one of the most important files in the NTFS file system. It keeps records of all files in a volume, the files' location in the directory, the physical location of the files in on the drive, and file metadata. The metadata includes file and folder create dates, entry modified dates, access dates, last written dates, physical and logical file size, and ACLs of the files. The file and directory metadata is stored as an MFT entry that is 1024 bytes in size. The first 16 entries in the MFT belong to system files, such as the MFT itself. From a forensics investigator perspective, entries are very interesting because when a file is deleted an entry gets marked as unallocated while the file content on the drive remains intact. The file name in the MFT entry can be overwritten due to MFT tree structure reorganization so most of the time file names are not maintained. File data eventually is overwritten as the unallocated drive space gets used.

Alternate Data Streams (ADS)

NTFS supports multiple data streams for files and folders. Files are composed of unnamed streams that contain the actual file data besides additional named streams (main-file.txt:one-stream). All streams within a file share the file's metadata, including file size. Since the file size does not change with the addition of ADSs, it becomes difficult to detect their existence. Open source forensics tools, such as The Sleuth Kit (TSK) can be used to parse MFT entries and reveal the existence of ADSs. Specifically, the TSK command fls can be used to list the files and the associated ADSs.

TABLE 41.2 Listing Files From a Raw Disc Image

```
$ fls -o 63 -rp evidence-diskimage.dd

r/r 11315-158-1:    ads-folder/ads-file.txt

r/r 11315-158-4:    ads-folder/ads-file.txt:suspicious.exe

r/r 11315-158-3:    ads-folder/host.txt:another-stream
```

TABLE 41.3 Extracting a File From a Raw Image With Its Inode

```
$ icat -o 63 evidence-diskimage.dd 11315-158-4 >  suspicious.exe
```

Locating and Restoring Deleted Content

Files can be fully or partially recovered depending on the method of deletion, time elapsed since the deletion, and drive fragmentation. A deleted or unlinked file is one whose MFT entry has been marked as unallocated and is no longer present in the user's view. The file can be recovered based on the metadata still present in the MFT entry given that too much time has not passed since the deletion. TSK can be used to parse the MFT to locate and recover these files. The investigator would need to execute the command fls to get a listing of the deleted file's inode and use that inode to extract the file data with the command icat.

Orphaned files' MFT entries, on the other hand, are no longer fully intact, and therefore the related metadata, such as the file name, might not be available. On the other hand, the file's data may still be recovered using the same method used for deleted files.

Unallocated files' MFT entries have been reused and/or unlinked. In this scenario, the only way to recover the file would be to "carve" the data out of the unallocated drive space. Carving involves using tools (foremost, scalpel) that recognize specific file formats, such as headers and footers to find the beginning and end of the file and extract the data.

Overwritten files' MFT entries and content have been reallocated or reused. Complete recovery would not be possible. Fragments of the file can be recovered by searching through the unallocated spaces on the drive with tools, such as grep.

5. INVESTIGATING INFORMATION-HIDING TECHNIQUES

Hidden data can exist due to regular OS activities or deliberate user activities. This type of data includes ADS, information obscured by malicious software, data encoded in media (steganography), hidden system files, and many others.

Uncovering Hidden Information

Collection of hidden data can be a challenge for an investigator. The investigator needs to be aware of the different data hiding techniques to employ the proper tools.

Scanning and Evaluating Alternate Data Streams

Open source forensics tools, such as TSK can be used to parse MFT entries and reveal the existence of ADSs. Specifically, the TSK command fls can be used to list the files and the associated ADSs as seen in Table 41.2.

In this example, we can see that the file ads-file.txt contains two streams named suspicious.exe and another-stream. The numbers seen in the beginning of each listing is the inode. This value identifies each file and folder in the file system. We should note that 63 bytes were skipped starting from the beginning of the drive since that data belongs to the master boot record (MBR). To extract the file data from the file system, the TSK command icat can be used in combination with the inode values as seen in Table 41.3.

Executing Code From a Stream

Malicious software can attempt to hide its components in ADSs to obscure themselves from investigators. Such components could be executable files. Executable ADSs can be launched with the Windows start command or by other scripting languages, such as VBScript or Perl by referring to

the ADS file directly: start ads-file.jpg:suspicious.exe. Executable hidden in ADSs can be automatically launched on system startup by defining it to do so in the Windows registry key "HKEY_LOCAL_MACHINE\-Software\Microsoft\Windows\CurrentVersion\Run" by creating a string value containing the full path of the ADS file.

Steganography Tools and Concepts

Steganography is the science of hiding secret messages in nonsecret messages or media in a manner that only the person who is aware of the mechanism can successfully find and decode. Messages can be hidden in images, audio files, videos, or other computer files without altering the actual presentation or functionality. While steganography is about hiding the message and its transmission, cryptography only aims to obscure the message content itself through various algorithms. Steganography can be performed by using the least significant bits in image files, placing comments in the source code, altering the file header, spreading data over a sound file's frequency spectrum, or hiding encrypted data in pseudorandom locations in a file. There are several tools that perform steganography:

- S-Tools is a freeware steganography tool that hides files in BMP, GIF, and WAV files. The message can be encrypted with algorithms, such as IDEA, DES, 3DES, and MDC before being hidden in the images.
- Spam Mimic is a freeware steganography tool that embeds messages in spam email content. This tool would be useful when real spam messages are numerous and the fake spam message would not wake any suspicion.
- Snow is a freeware steganography tool that encodes message text by appending white space characters to the end of lines. The tool's name stands for and exploits the steganographic nature of whitespace. It also can employ ICE encryption to hide the content of the message in case of the detection of steganography. While most of the time it's visually undetectable, it can be discovered by a careful investigator or a script looking for this tool's artifacts.
- OutGuess is an open source tool that hides messages in the redundant bits of data sources. OutGuess can use any data format as a medium as long as a handler is provided.

Detecting Steganography

During an incident, an investigator might suspect that steganography has been used by the suspect due to an admission, a discovery of specific tools, or other indicators. Traces of the use of steganography tools can be found in the recently used files (MRU) key, the USERASSIST key, and the MUICache key in the Windows registry; prefetch files, web browser history, and deleted file information in the file system; and in the Windows Search Assistant utility. File artifacts generated by these tools can also be a good indicator of the tools' use.

The presence of files (JPEG, MP3) that present similar properties but different hash values might also generate suspicion. The investigator might be able to discover such pairs of carrier and processed files to apply discovery algorithms to recover the hidden messages. Steganalysis tools can also be used to detect the presence of steganography:

- Stegdetect is an open source steganalysis tool that is capable of detecting steganographic content in images that have been generated by JSteg, JPHide, Invisible Secrets, OutGuess, F5, Camouflage, and appendX.
- StegSpy is a freeware steganalysis tool that can currently identify steganography generated by the Hiderman, JPHideandSeek, Masker, JPegX, and Invisible Secrets tools.
- Stegbreak is used to launch dictionary attacks against JSteg-Shell, JPHide, and OutGuess 0.13b generated messages.

Scavenging Slack Space

File slack or slack space refers to the bytes between the logical end of a file and the end of the cluster the file resides in. Slack space is a source of information leak, which can result in password, email, registry, event log, database entries, and word processing document disclosures. File slack has the potential of containing data from the system memory. This can happen if a file can't fill the last sector in a cluster and the Windows OS uses randomly selected data from the system memory (RAM slack) to fill the gap. RAM slack can contain any information loaded into memory since the system was turned on. The information can include file content resident in memory, usernames, passwords, and cryptographic keys. File slack space can also be used to hide information by malicious users or software, which can get challenging if the investigator is not specifically looking for such behavior. Volume slack is the space that remains on a drive when it's not used by any partition. This space can contain data if it was created as a result of deleting a partition. While the partition metadata no longer exists, its contents still remain on the drive. Partition slack is the area between the ending of a logical partition and the ending of a physical block the partition is located in. It is created when the number of sectors in a partition is not a multiple of the physical block size. Registry slack is formed when a registry key is deleted and the size value is changed to a positive value. Normally, key sizes are negative values when read as signed integer. The registry key data still remains on the drive. Jolanta Thomassen created a Perl script called "regslack" to parse registry hives and extract

TABLE 41.4 Inspecting File Headers

```
$ file hidden.pdf

hidden.pdf: PE32 executable for MS Windows (GUI) Intel 80386 32-bit
```

```
Example of a PDF Header:

0000000: 2550 4446 2d31 2e36 0a25 e4e3 cfd2 0a31  %PDF-1.6.%.....1
```

```
Example of a Windows Executable Header:

0000000: 4d5a 9000 0300 0000 0400 0000 ffff 0000  MZ..............
```

deleted keys by exploiting the negative to positive conversion information.

Inspecting Header Signatures and File Mangling

Users or malware with malicious intent can alter or mangle file names or the files themselves to hide files that are used to compromise systems or contain data that has been gathered as a result of their malicious actions. These techniques include but are not limited to renaming files, embedding malicious files in regular files (PDF, Doc, Flash), binding multiple executables in a single executable, and changing file times to avoid event time—based analysis. For example, a malicious Windows executable "bad.exe" can be renamed to "interesting.pdf" and be served by a web page to an unsuspecting user. Depending on the web browser, the user would get prompted with a dialog that asks them if they would like to run the program and most of the time the user will dismiss the dialog by clicking the OK button. To analyze a file disguised in different file extensions, a header based file type checker, such as the Linux "file" command or the tool TrID (also available in Windows) can be used. Table 41.4 provides a sample of the malware Trojan.Spyeye hidden in a file with an Acrobat PDF document extension being detected by the tool file.

Combining Files

Combining files is a very popular method among malware creators. Common file formats, such as Microsoft Office files, Adobe PDF, and Flash files can be used as containers to hide malicious executables. One example is a technique where a Windows executable is embedded in a PDF file as an object stream and marked with a compression filter. The stream is usually obfuscated with XOR. The Metasploit Framework provides several plugins to generate such files for security professionals to conduct social engineering in the form of phishing attacks. The example in Table 41.5 uses Metasploit to generate a PDF file with an embedded executable file.

To discover such embedding, an investigator can use Didier Stevens's tool PDF-parser to view the objects in a PDF file.

Binding Multiple Executable Files

Binding multiple executable files provides the means to pack all dependencies and resource files a program might need while running into a single file. This is advantageous since it permits a malicious user to leave a smaller footprint on a target system and makes it harder for an investigator to locate the malicious file. Certain tools, such as the WinZip Self-Extractor, nBinder, or File Joiner can create one executable file by archiving all related files whose execution will be controlled by a stub executable. When executed, the files will be extracted and the contained program will be launched automatically. Some of these file binders can produce files that can't be detected by some antiviruses and if downloaded and ran by an unsuspecting user, it can result in a system compromise.

File Time Analysis

File time analysis is one of the most used techniques by investigators. File times are used to build a story line that could potentially reveal how and when an event on a system caused a compromise. The file time of a malicious executable could be linked to a user's browser history to find out which sites were visited before the compromise occurred.

The problem with this type of analysis is that sometimes the file times can be tampered with and can't be relied upon as evidence. The tool Timestomp, created by James Foster and Vincent Liu, allows for the deletion or modification of file MACE times (modified, accessed, created, entry modified in MFT times) in the MFT's $STANDARD_INFORMATION attribute. Timestomp can't change the file MACE times in the

TABLE 41.5 Creating a Malicious PDF File With the Metasploit Framework

```
msf > use exploit/windows/fileformat/adobe_pdf_embedded_exe

msf exploit(adobe_pdf_embedded_exe) > set FILENAME evil.pdf

FILENAME => evil.pdf

msf exploit(adobe_pdf_embedded_exe) > set PAYLOAD windows/meterpreter/reverse_tcp

PAYLOAD => windows/meterpreter/reverse_tcp

msf exploit(adobe_pdf_embedded_exe) > set INFILENAME ./base.pdf

INFILENAME => ./base.pdf

[*] Please wait while we load the module tree...

….

INFILENAME => ./base.pdf

payload => windows/meterpreter/bind_tcp

[*] Reading in './base.pdf'...

[*] Parsing './base.pdf'...

[*] Parsing Successful.

[*] Using 'windows/meterpreter/bind_tcp' as payload...

[*] Creating 'evil.pdf' file...

[+] evil.pdf stored at …/local/evil.pdf
```

```
timestomp.exe c:\test.txt -z "Saturday 10/08/2005 2:02:02 PM"
timestomp.exe c:\test.txt -a "Saturday 10/08/2005 2:02:02 PM"
```

Standard Information		File Name Info.	
Creation	10/8/2005 : 14:2:2	Creation	10/15/2008 : 0:37:35
Modifica.	10/8/2005 : 14:2:2	Modifica.	10/15/2008 : 0:37:35
MFT	10/8/2005 : 14:2:2	MFT	10/15/2008 : 0:38:49
Last Acc.	10/8/2005 : 14:2:2	Last Acc.	10/15/2008 : 0:37:35

FIGURE 41.3 Changing timestamps as a result of time stomping.

MFT's $FILE_NAME attribute because this attribute is meant to be modified by Windows system internals only. This time tampering method can be defeated by using Mark McKinnon's MFT Parser tool to view all eight-file times to detect discrepancies as seen in Fig. 41.3.

Executable compile times can also be used as a data point during timeline analysis. The open source Python module pefile can be used to extract this information from the executable header. For example, if malicious software changes MACE times to a future date, but keeps its original

TABLE 41.6 Compilation Time of an Executable Extracted With Pefile

```
Compilation timedatestamp: 2010-03-23 23:42:40

Target machine: 0x14C (Intel 386 or later processors and compatible processors)

Entry point address: 0x000030B1
```

compile time, this can be flagged as suspicious by an investigator as seen in Table 41.6.

6. SCRUTINIZING EMAIL

While many noncommercial users are favoring webmail nowadays, most corporate users are still using local email clients, such as Microsoft Outlook or Mozilla Thunderbird. Therefore, we should still look at extracting and analyzing email content from local email stores. Email message analysis might reveal information about the sender and recipient, such as email addresses, IP addresses, data and time, attachments, and content.

Investigating the Mail Client

An email user will generally utilize a local client to compose and send their message. Depending on the user's configuration, the sent and received messages will exist in the local email database. Deleted e-mails can be also stored locally for some time depending on the user's preferences. Most corporate environments utilize Microsoft Outlook. Outlook will store the mail in a portable storage table (PST) or offline storage table (OST) format. Multiple PST files can exist in various locations on the user's file system and can provide valuable information to an investigator about the user's email-specific activity.

Interpreting Email Headers

Generally speaking, email messages are composed of three sections: header, body, and attachments. The header contains source and destination information (email and IP addresses), date and time, email subject, and the route the email takes during its transmission. Information stored in a header can either be viewed through the email client or through an email forensics tool such as libpff (an open source library to access email databases), FTK, or EnCase.

The "Received" line in Table 41.7 email header shows that the email was sent from IP address 1.1.1.1. An investigator should not rely on this information as concrete evidence because it can be easily changed by a malicious sender (email spoofing). The time information in the header might also be incorrect due to time zones, user system inaccuracies and tampering.

Recovering Deleted E-mails

While most users treat e-mails as transient, the companies they work for have strict data retention policies that can enforce the storage of email, sometimes indefinitely. User emails usually are stored in backup archives or electronic-discovery systems to provide means for analysis in case there is an investigation. The email servers also can keep messages in store although the users remove them from their local systems. Therefore, it has become somewhat difficult for a corporate user to delete an email permanently. Recovery is usually possible from various backup systems. In cases where there is no backup source and users delete an email from their local system, we need to perform several steps on the user's storage drive depending on the level of deletion:

- If the user deletes the message, but does not empty the deleted messages folder, the user can move the messages from the deleted folder to the original folder quite easily.
- If the user deletes the email message and removes it from the deleted messages folder, then the investigator needs to apply disc forensics techniques to recover the email. In case of a Microsoft Outlook PST file, when a message is deleted it is marked as deleted by Outlook and the data remains on the disc unless the location on the drive is overwritten by new data. Commercial tools, such as AccessData's FTK or Guidance's EnCase can be used to recover deleted messages. Another approach would be to use Microsoft's "Scanpst.exe" tool. To apply this technique, the investigator should first backup the PST and then deliberately corrupt the PST file with the command "DEBUG <FILE.pst> -f 107 113 20 -q." If the file is too large and there is insufficient system memory, then the investigator should use a hex editor to make the changes marked in red in the PST file as shown in Fig. 41.4.

TABLE 41.7 An Email Header

```
Return-Path: <example_from@acme.edu>
X-SpamCatcher-Score: 1 [X]
Received: from [1.1.1.1] (HELO acme.edu)
    by fe3.acme.edu (CommuniGate Pro SMTP 4.1.8)
    with ESMTP-TLS id 61258719 for example_to@mail.acme.edu;
Mon, 23 Aug 2004 11:40:10 -0400
Message-ID: <4129F3CA.2020509@acme.edu>
Date: Mon, 23 Aug 2005 11:40:36 -0400
From: Jim Doe <example_from@acme.edu>
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US;
rv:1.0.1) Gecko/20020823 Netscape/7.0
X-Accept-Language: en-us, en
MIME-Version: 1.0
To: John Doe <example_to@mail.acme.edu>
Subject: Sales Development Meeting
Content-Type: text/plain; charset=us-ascii; format=flowed
Content-Transfer-Encoding: 7bit
```

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	21	42	44	4E	4B	E7	0A	20	20	20	20	20	20	20	20	20	!BDNKç.
00000010	20	20	20	20	b0	00	00	00	04	00	00	00	01	00	00	00	▯.....
00000020	B3	93	07	00	00	00	00	00	F8	7E	27	00	D2	D8	00	00	³"....ø~'.ÒØ..
00000030	09	35	01	00	09	2A	01	00	00	36	01	00	28	6E	04	00	.5...*...6..(n..
00000040	84	37	01	00	00	36	01	00	00	36	01	00	E0	DA	00	00	„7...6...6..àÚ..

FIGURE 41.4 Manipulating a PST file for recovery.

Once the PST file has been corrupted, then the "Scanpst.exe" tool should be located on the drive. And it should also be executed to repair the file as seen in Fig. 41.5.

7. VALIDATING EMAIL HEADER INFORMATION

Email header information can be tampered with by users that wish to not disclose their source information or by malicious users that would like to fake the origin of the message to avoid detection and being blocked by spam filters. Email header information can be altered by spoofing by using an anonymizer (removes identifying information), and using a mail relay server.

Detecting Spoofed Email

A spoofed email message is a message that appears to be from an entity other than the actual sender entity. This can be accomplished by altering the sender's name, email address, email client type, and/or the source IP address in the email header. Spoofing can be detected by looking at the "Received" and "Message-ID" lines of the header. The "Received" field will have each email server hop the message that was taken before it had been received by the email client. An investigator can use the email server IP addresses in the header to get their host names from their Domain Name System (DNS) records and verify them by comparing to the actual outgoing and incoming email

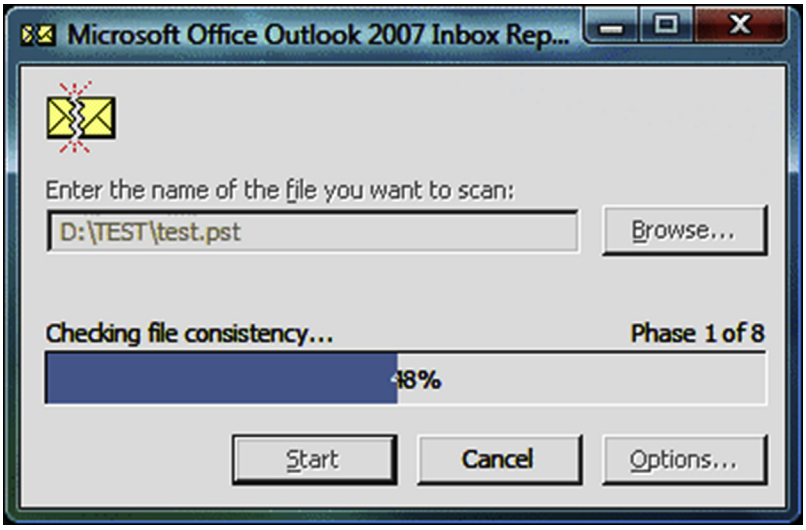

FIGURE 41.5 Use of the Scanpst.exe tool.

servers' information. The "Message-ID" field uniquely identifies a message and is used to prevent multiple deliveries. The domain information in the "Message-ID" field should match the domain information of the sender's email address. If this is not the case, the email is most probably spoofed. An investigator should also look out for different "From" and "Reply-To" email addresses, and unusual email clients displayed in the "X-Mailer" field.

Verifying Email Routing

Email routing can be verified by tracing the hops an email message has taken. This can be accomplished by verifying the "Received" field information through DNS records and if possible obtaining email transaction logs from the email servers involved. The "Message-ID" information can be searched for in the logs to make sure that the message has actually traveled the route declared in the "Received" field.

8. TRACING INTERNET ACCESS

Knowing the path a perpetrator has taken becomes very valuable when an investigator is building a case to present in court. It adds credibility to the claim and solidifies the storyline by connecting the events. For example, knowing the path an attacker has taken to steal a company's source code can reveal the extent of the compromise (loss of domain credentials, customer information leakage, and intellectual property loss), show intent, and prevent the same attack from happening. Tracing Internet access can also be valuable in the case of employees viewing content not compliant with work place rules.

Inspecting Browser Cache and History Files

An investigator can use various data points to trace a perpetrator's activity by analyzing the browser cache and web history files in the gathered evidence. Every action of a user on the Internet can generate artifacts. The browser cache contains files that are saved locally as a result of a user's web browsing activity. The history files contain a list of visited URLs, web searches, cookies, and bookmarked websites. These files can be located in different folders depending on the OS, OS version, and browser type.

Exploring Temporary Internet Files

A browser cache stores multimedia content (images, videos), and web pages (HTML, JavaScript, CSS) to increase the load speed of a page when viewed the next time. For the Internet Explorer web browser on Windows XP and 2003, the cache files can be located in the folder "Documents and Settings\%username%\Local Settings\Temporary Internet Files," in Windows Vista/7/2008 they are located in the folder "Users\%username%\AppData\Local\Microsoft\Windows\Temporary Internet Files":

- On Windows XP/2003, Mozilla Firefox stores the cached files in the folder "C:\Documents and Settings\%username\Local Settings\ Application Data\Mozilla\Firefox\Profiles," and for Windows Vista/7/2008 in "C:\Users\%username%\AppData\Roaming\Mozilla\Firefox\ Profiles."
- On Windows XP/2003 Google Chrome web browser stores the cached files in the folder "C:\Documents and Settings\%username\Application Data\ Google\Chrome\Default\Cache," and for Windows Vista/7/

2008 in "C:\Users\%username%\AppData\Local\Google\Chrome\ Default\Cache."

- The MAC times of these cached files can be used during a timeline analysis to find when certain artifacts, such as malware, get dropped by malicious or compromised websites. Malicious executable PDF files or Java files can be located in the cache unless the cache is cleared by the user or malware.

Visited URLs, Search Queries, Recently Opened Files

The Internet Explorer web browser stores the visited URL, search query, and opened file information in the file "index.dat" accompanied by last modified and last accessed, and expiration times. This file on Windows XP/2003 systems can be located in the folder "Documents and Settings\%username%\Local Settings\Temporary Internet Files\Content.IE5," and in Windows Vista/7/2008 systems it is located in the folder "Users\%username%\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5." The "index.dat" file contains a LEAK record, which is a record that remains when it's marked as deleted, but can't be deleted due to a related temporary internet file (TIF) still being used.

Mozilla Firefox stores the URL, search, and open files-related history in a SQLite 3 database file Places.sqlite. These files on Windows XP/2003 systems can be located in the folder "C:\Documents and Settings\%username\Local Settings\ Application Data\Mozilla\Firefox\Profiles," and in Windows Vista/7/2008 systems it is located in the folder "C:\Users\%username%\AppData\Roaming\Mozilla\Firefox\Profiles."

Google Chrome also stores its user activity data in SQLite 3 database files. These files on Windows XP/2003 systems can be located in the folder "C:\Documents and Settings\%username\Application Data\ Google\Chrome\-default," and in Windows Vista/7/2008 systems it is located in the folder "C:\Users\%username%\AppData\Local\Google\Chrome\ default."

All three browsers' history files, cookies, and cache files can be parsed and interpreted by log2timeline, a tool created by Kristinn Gudjonsson. Log2timeline is capable of parsing multiple data sources and producing a timeline with, including but not limited to, file MAC times, registry write times, and Windows event logs. The tool can be pointed to a raw drive image (dd image) and as a result it can produce a "super" timeline in CSV format for all pursuable time-based data sources as seen Fig. 41.6 and Table 41.8.

Researching Cookie Storage

Internet Explorer cookies can be found in the folder "Documents and Settings\%username%\ Cookies" in

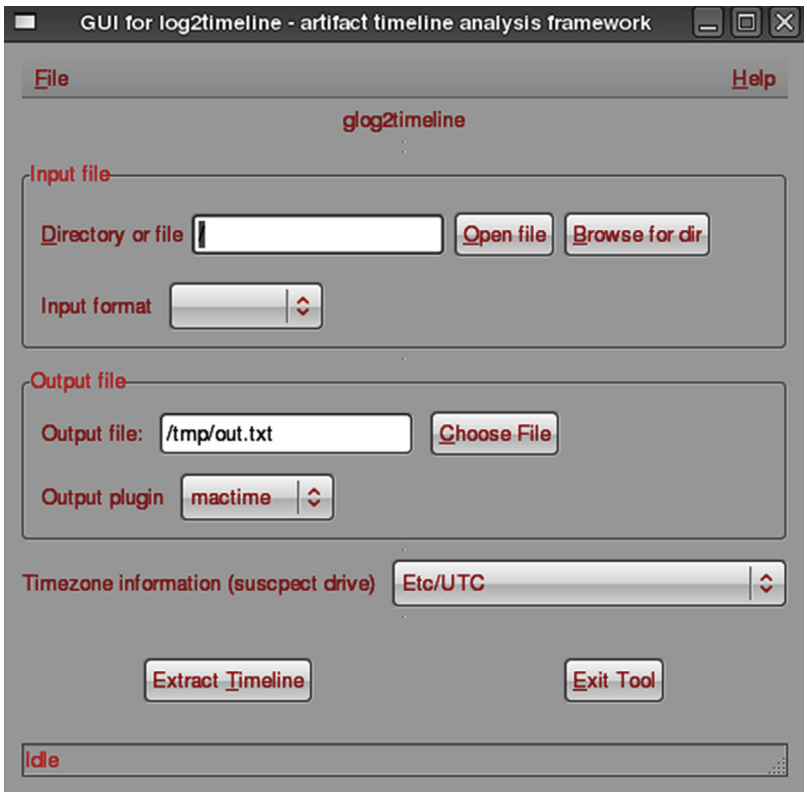

FIGURE 41.6 Graphical user interface (GUI) for log2timeline.

TABLE 41.8 Using Log2timeline From the Command Line

```
log2timeline -p -r -f win7 -z EST5EDT /storage/disk-image001.dd -w supertimeline001.csv
```

Windows XP/2003 systems and in Windows Vista/7/2008 systems in "Users\%username%\AppData\Roaming\Microsoft\Windows\Cookies." The cookies are stored in plain text format.

Mozilla Firefox stores its cookies in a SQLite 3 database Cookies.sqlite located in the folder "C:\Documents and Settings\%username\Local Settings\ Application Data\Mozilla\Firefox\Profiles," and in Windows Vista/7/2008 systems, it is located in the folder "C:\Users\%username%\AppData\Roaming\Mozilla\Firefox\Profiles."

Google Chrome stores its cookies in a SQLite 3 database file "C:\Documents and Settings\%username\Application Data\ Google\Chrome\default\Cookies" in Windows XP/2003 systems, and in the file "C:\Users\%username %\AppData\Local\Google\Chrome\ default\Cookies" in Windows Vista/7/2008 systems. The cookies files of all three browser types can be parsed and viewed by the tool log2timeline as previously mentioned.

Reconstructing Cleared Browser History

It is possible to come across cleared browser histories during an investigation. The user could have deliberately deleted the files to hide their web browsing activity or a malware could have removed its traces to avoid detection and analysis. Nevertheless, an investigator will look into various locations on the suspect system to locate the deleted browser history files. The possible locations are unallocated clusters; cluster slack, page files, system files, hibernation files, and system restore points. Using AccessData's FTK Imager on the suspect drive or drive image, an investigator could promptly locate the orphaned files and see if the browser files are present there. The next step would be to use the FTK Imager to look at the unallocated spaces, which should end up being a time-consuming analysis as seen in Fig. 41.7. If the drive has not been used too much, an investigator has a high chance of locating the files in the unallocated space.

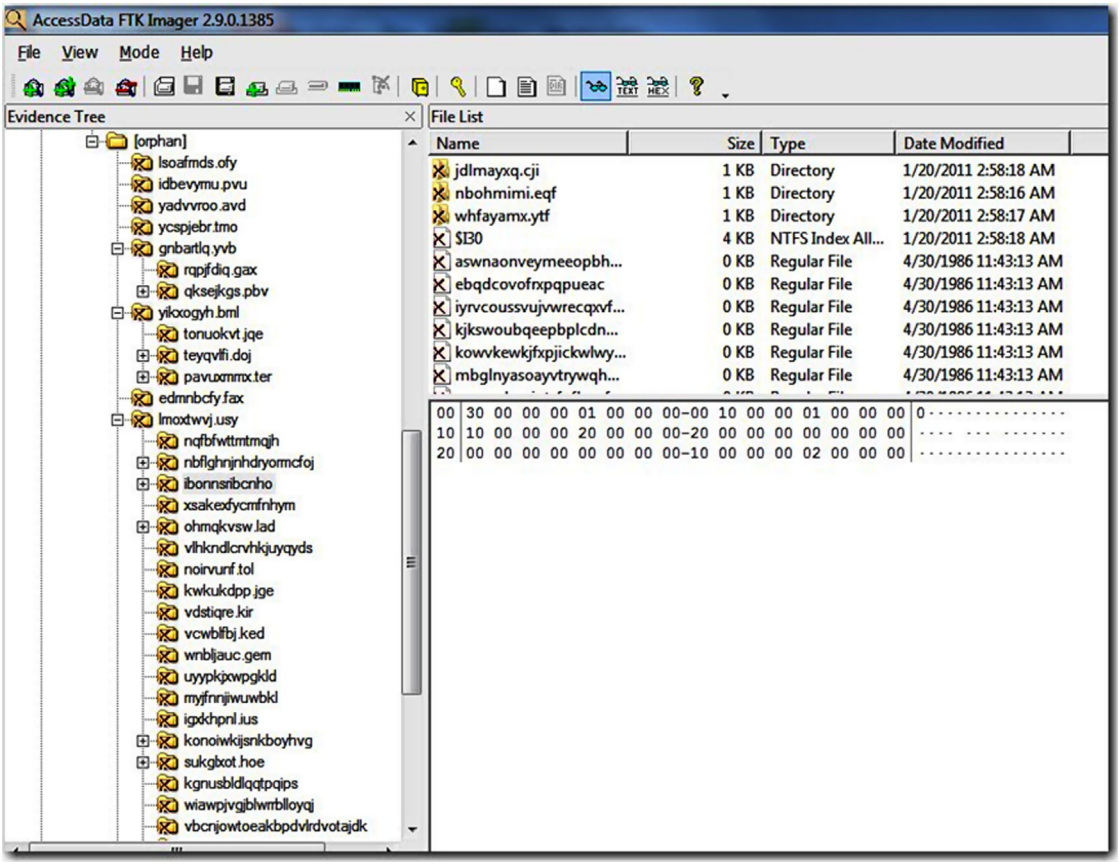

FIGURE 41.7 Use of AccessData FTK Imager.

Auditing Internet Surfing

Knowing what employees are browsing on the web while they are at work has become necessary to prevent the employees from visiting sites that host malicious content (sites with exploits and malware), content that is not compliant with work place rules, and content that is illegal. Employees can use the web to upload confidential corporate information, which can cause serious problems for the employer.

Tracking User Activity

User activity can be tracked by using tools that monitor network activity, DNS requests, local user system activity, and proxy logs. Network activity, on the other hand, can be monitored by looking at netflows. A netflow is a network protocol developed by Cisco Systems for monitoring IP traffic. It captures source and destination IP addresses, IP protocol, source and destination ports, and IP type of service.

Local user system activity can be monitored by installing specific agents on the users' systems that can report their activity back to a centralized server. SpectorSoft offers a product called Spector 360 that can be installed on user system and a central server. The agents on the user systems can track user browser activity by hooking into system application program interfaces (APIs) and enforce rules set by the employer.

DNS requests can be monitored at the corporate DNS server level or by looking at network traffic. When a user requests a web page with its domain name, the name gets translated to an IP address via DNS. User activity can be tracked by monitoring for domains that are not approved by the employer or domains hosting illegal content in the DNS server's logs.

Most corporate environments utilize a proxy server to funnel web traffic through. A proxy server acts as the middle man for requests from users seeking resources from external servers. This position of the proxy server permits tracking user browsing activity and can be used to filter or block certain behavior. Content protected by Secure Socket Layer (SSL) protocol can also be tracked by proxies, by setting the proxy up as an intercept proxy. Squid is one of the most popular open source proxies available and can carry out the necessary functions to track and block user activity.

Uncovering Unauthorized Usage

Unauthorized web usage can take multiple forms, such as downloading or viewing noncompliant or illegal content, uploading confidential information, launching attacks on other systems, and more. Once the unauthorized usage has been detected by the previously mentioned means, an investigator can focus on the user's system to corroborate the unauthorized activity. This can be done by analyzing browser history files and related file system activities. Building a "super" timeline with the tool log2timeline can become very useful to find the created cache and cookie files and the browser history entries around the same time the unauthorized activity was detected.

9. SEARCHING MEMORY IN REAL TIME

Analyzing memory in real time can provide very crucial information about activities of malware or a hacker that would be otherwise unavailable if only looking at a system's drives. This information can be network connections and sockets, system configuration settings, collected private information (user names, passwords, credit card numbers), memory-only resident executables, and much more. Real-time analysis involves analyzing volatile content and therefore requires swift action by the investigator. The investigator has to quickly act to capture an image of the memory using tools, such as MoonSols Windows Memory Toolkit, GMG Systems' KnTDD, or F-response. F-response is different from the other memory imaging tools since it provides real-time access to the target systems memory. Real-time access can reduce the time to analyze by permitting the investigator to analyze the memory right away without waiting for the whole memory to be downloaded into a file. You can read more about F-response at www.f-reponse.com.

Comparing the Architecture of Processes

Generally speaking, Windows architecture uses two access modes, which are user and kernel modes. The user mode includes application processes, such as programs and protected subsystems. The protected subsystems are named so because each of these is a separate process with its own protected virtual address space in memory. The kernel mode is a privileged mode of functioning in which the application has direct access to the virtual memory. This includes the address spaces of all user mode processes and applications and the associated hardware. The kernel mode is also called as the protected mode, or Ring 0:

- Windows processes are generally composed of an executable program, consisting of initial code and data, a private virtual address space, system resources that are accessible to all threads in the process, a unique identifier, called a process ID (PID), at least one thread of execution, and a security context (an access token).
- A Windows thread is what Windows uses for execution within a process. Without threads, the program used by the process cannot run. Threads consist of contents of the registers representing the state of the processor, two stacks (one for the thread for executing

kernel-mode instructions, and one for user mode), private storage area used by the subsystems, run-time libraries and DLLs, and a unique identifier named a thread ID.

- DLLs are set of callable subroutines linked together as a binary file that can be dynamically loaded by applications that use the subroutines. Windows user-mode entities utilize DLLs extensively. Using DLLs are advantageous for an application since they can share DLLs. Windows ensures that there is only one copy of a DLL in memory. Each DLL has its own import address table (IAT) in its compiled form.

Identifying User and Kernel Memory

Windows refers to Intel's linear memory address space as a virtual address space (VAS) since Windows uses the disc space structure to manage physical memory. In other words, 2 GB of VAS is not a one-to-one match to physical memory. Thirty-two-bit Windows divides VAS into user space (linear addresses 0x00000000 – 0x7FFFFFFF, 2 GB) and kernel space (linear addresses 0x80000000 – 0xFFFFFFFF, 2 GB) where user space gets the lower end of the address range and kernel space gets the upper end of the address space. To get an idea of how user space is arranged, we can use the !peb command in the Windows debugger. A list of loaded kernel modules can be obtained by running the command 'lm n' in the Windows debugger.

Inspecting Threads

While it is assumed that user and kernel code are restricted to their own address spaces, a thread can jump from user space to kernel space by the instruction SYSENTER and jump back with the instruction SYSEXIT. Malicious threads can also exist within valid kernel or other user processes. Such hidden or orphan kernel threads can be detected using the Volatility Framework with the plugin threads, which are shown in Table 41.9.

Discovering Rogue Dynamic-Link Libraries (DLLs) and Drivers

DLLs can be used for malicious purposes by injecting them through AppInit_DLLs registry value, SetWindowsHookEx() API call, and using remote threads via the CreateRemoteThread() Windows API call. Injected DLLs can be detected using the Volatility Framework's apihooks plugin. The plugin provides detailed information regarding the DLLs loaded, such as IAT, process, hooked module, hooked function, from–to instructions, and hooking module as seen in Table 41.10.

The Volatility Framework plugin malfind can find hidden or injected DLLs in user memory based on Virtual Address Descriptor (VAD) tags and page. The use of the malfind plugin to discover injected code is shown in Table 41.11.

The plugin dlllist in the Volatility Framework can also be used to list all DLLs for a given process in memory and find DLLs injected with the CreateRemoteThread and LoadLibrary technique. This technique does not hide the DLL and therefore will not be detected by the plugin malfind as seen in Table 41.12.

Employing Advanced Process Analysis Methods

Processes can be analyzed using tools, such as the Windows Management Instrumentation (WMI), and walking dependency trees.

Evaluating Processes with Windows Management Instrumentation (WMI)

WMI is a set of extensions to the Windows Driver Model that provides an OS interface where components can provide information and notifications. The WMI classes Win32_Process can help collect useful information about processes. The Windows command wmic extends WMI for operation from several command-line interfaces and through batch scripts without having to rely on any other programming language. The command wmic uses class aliases to query related information. It can be executed remotely as well as locally by specifying target node or host name and credentials. Various commands that can be used to extract various process-related information through wmic are shown in Table 41.13.

WMI output can be used to get a clean baseline of a system to periodically run comparisons. The comparisons can show any new process that has appeared on the system, and can help to update the baseline if the new process is a known one.

Walking Dependency Trees

Viewing the dependencies of a process can provide valuable information about the functionality a process contains. A process's dependencies may be composed of various Windows modules, such as executables, DLLs, object linking and embedding control extension (OCX) files; SYS files (mostly real-mode device drivers). Walking a dependency tree means to explore a process's dependencies in a hierarchal view, such as a tree. The free tool Dependency Walker provides an interface that presents such a view, which is shown in Fig. 41.8.

It lists all of the functions that are exported by a given Windows module, and the functions that are actually being called by other modules. Another view displays the minimum set of required files, along with detailed information

TABLE 41.9 Use of the Threads Plugin

```
$ python vol.py threads -f /memory_samples/tigger.vmem -F OrphanThread
Volatile Systems Volatility Framework 2.2_alpha
[x86] Gathering all referenced SSDTs from KTHREADs...
Finding appropriate address space for tables...
------
ETHREAD: 0xff1f92b0 Pid: 4 Tid: 1648
Tags: OrphanThread,SystemThread
Created: 2010-08-15 19:26:13
Exited: 1970-01-01 00:00:00
Owning Process: System
Attached Process: System
State: Waiting:DelayExecution
BasePriority: 0x8
Priority: 0x8
TEB: 0x00000000
StartAddress: 0xf2edd150 UNKNOWN
ServiceTable: 0x80552180
  [0] 0x80501030
  [1] 0x00000000
  [2] 0x00000000
  [3] 0x00000000
Win32Thread: 0x00000000
CrossThreadFlags: PS_CROSS_THREAD_FLAGS_SYSTEM
0xf2edd150 803d782aeff200   CMP BYTE [0xf2ef2a78], 0x0
0xf2edd157 7437             JZ 0xf2edd190
0xf2edd159 56               PUSH ESI
0xf2edd15a bef0d0edf2       MOV ESI, 0xf2edd0f0
0xf2edd15f ff35702aeff2     PUSH DWORD [0xf2ef2a70]
0xf2edd165 ff               DB 0xff
0xf2edd166 15               DB 0x15
0xf2edd167 0c               DB 0xc
…
```

TABLE 41.10 Use of the Apihooks Plugin to Detect Hooking

```
$ python vol.py -f coreflood.vmem -p 2015 apihooks

Volatile Systems Volatility Framework 2.1_alpha

**************************************************************************

Hook mode: Usermode

Hook type: Import Address Table (IAT)

Process: 2015 (IEXPLORE.EXE)

Victim module: iexplore.exe (0x400000 - 0x419000)

Function: kernel32.dll!GetProcAddress at 0x7ff82360

Hook address: 0x7ff82360

Hooking module: <unknown>


Disassembly(0):

0x7ff82360 e8fbf5ffff       CALL 0x7ff81960

0x7ff82365 84c0             TEST AL, AL

0x7ff82367 740b             JZ 0x7ff82374

0x7ff82369 8b150054fa7f     MOV EDX, [0x7ffa5400]

0x7ff8236f 8b4250           MOV EAX, [EDX+0x50]

0x7ff82372 ffe0             JMP EAX

0x7ff82374 8b4c2408         MOV ECX, [ESP+0x8]

...
```

about each file including a full path to the file, base address, version numbers, machine type, and debug information. Dependency Walker can be used in conjunction with the tool Process Explorer (a free Microsoft tool) to detect malicious DLLs. This can be achieved by comparing the DLL list in Process Explorer to the imports displayed by the Dependency Walker.

Auditing Processes and Services

Auditing changes in process and service properties as well as their counts on a system can provide valuable information to an investigator about potentially malicious activity. Rootkits, viruses, Trojans, and other malicious software can be detected by auditing process and service creation or deletion across a period of time. This technique is frequently used in malware behavioral analysis in sandboxes. We can audit Windows processes and services by using tools that utilize system APIs or system memory for live analysis. To view information through the system APIs we can use the tool Process Hacker. The Process Hacker provides a live view of processes and services that are currently being executed or present on the system. It provides an interface to view and search process information in detail, such as process privileges, related users and groups, DLLs loaded, handles opened, and thread information, which is shown in Fig. 41.9. Service information can also be viewed in the services tab. Process and service information can be saved periodically and compared across time to detect any suspicious variations.

System memory can also be analyzed with the Volatility Framework for audit purposes. Periodic memory samples can be obtained using tools, such as F-response, MoonSols Windows Memory Toolkit, or GMG Systems' KnTDD. The

TABLE 41.11 Use of the Malfind Plugin to Discover Injected Code

```
$ python vol.py -f zeus.vmem malfind -p 1645
Volatile Systems Volatility Framework 2.1_alpha


Process: explorer.exe Pid: 1645 Address: 0x1600000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 1, MemCommit: 1, PrivateMemory: 1, Protection: 6


0x01600000  b8 35 00 00 00 e9 cd d7 30 7b b8 91 00 00 00 e9   .5......0{......
0x01600010  4f df 30 7b 8b ff 55 8b ec e9 ef 17 c1 75 8b ff   O.0{..U......u..
0x01600020  55 8b ec e9 95 76 bc 75 8b ff 55 8b ec e9 be 53   U....v.u..U....S
0x01600030  bd 75 8b ff 55 8b ec e9 d6 18 c1 75 8b ff 55 8b   .u..U......u..U.


0x1600000 b835000000       MOV EAX, 0x35
0x1600005 e9cdd7307b       JMP 0x7c90d7d7
0x160000a b891000000       MOV EAX, 0x91
0x160000f e94fdf307b       JMP 0x7c90df63
0x1600014 8bff             MOV EDI, EDI
0x1600016 55               PUSH EBP
```

Volatility Framework plugin pslist can be used to audit processes while the plugin svcscan can be used to audit services.

Investigating the Process Table

The process table (PT) is a data structure kept by the OS to help context switching, scheduling, and other activities. Each entry in the PT, called process context blocks (PCB), contains information about a process, such as process name and state, priority, and PID. The exact content of a context block depends on the OS. For example, if the OS supports paging, then the context block contains a reference to the page table. While to a user a process is identified by the PID, in the OS the process is represented by entries in the PT. The process control block is a large data structure that contains information about a specific process. In Linux this data structure is called task_struct whereas in Windows it is called an EPROCESS structure. Each EPROCESS structure contains a LIST_ENTRY structure called ActiveProcessLinks, which contains a link to the previous (Blink) EPROCESS structure and the next (Flink) EPROCESS structure.

A listing of processes represented in the PT can be obtained by using the plugin pslist in the Volatility Framework. This plugin generates its output by walking the doubly linked list. Certain malware or malicious users can hide processes by unlinking them from this linked list by performing direct kernel object manipulation (DKOM). To detect this kind of behavior, the Volatility Framework plugin psscan can be used since it relies on scanning the memory to detect pools similar to that of an EPROCESS structure instead of walking the linked list.

Discovering Evidence in the Registry

We have already covered one method to inject code into a process. Another method is to add a value in the AppInit_DLLs registry key (HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Windows\AppInit_DLLs) that makes a new process to load a DLL of malicious origin (table XXX).

AppInit_DLLs is a mechanism that allows an arbitrary list of DLLs to be loaded into each user mode process on

TABLE 41.12 Use of the Dlllist Plugin to Detect DLL Injection

```
$ python vol.py dlllist -f sample.vmem -p 468
Volatile Systems Volatility Framework 2.2_alpha
************************************************************************
wuauclt.exe pid:    468
Command line : "C:\WINDOWS\system32\wuauclt.exe"
Service Pack 2

Base             Size Path
---------- ---------- ----
0x00400000    0x1e000 C:\WINDOWS\system32\wuauclt.exe
0x7c900000    0xb0000 C:\WINDOWS\system32\ntdll.dll
0x7c800000    0xf4000 C:\WINDOWS\system32\kernel32.dll
0x77c10000    0x58000 C:\WINDOWS\system32\msvcrt.dll
0x76b20000    0x11000 C:\WINDOWS\system32\ATL.DLL
0x77d40000    0x90000 C:\WINDOWS\system32\USER32.dll
0x77f10000    0x46000 C:\WINDOWS\system32\GDI32.dll
0x77dd0000    0x9b000 C:\WINDOWS\system32\ADVAPI32.dll
0x77e70000    0x91000 C:\WINDOWS\system32\RPCRT4.dll
…
```

TABLE 41.13 List of Windows Management Instrumentations (WMI) Commands to Get Process Information

`wmic /node: /user: process where get ExecutablePath,parentprocessid`	Find the path to a specific running executable and its parent process (for all, leave off 'where name=')
`wmic /node: /user: process where get name,processid,commandline,creationdate`	Find command-line invocation of a specific executable as well as the creation time for the process (for all, leave off 'where name=').
`wmic startup list full`	Find all files loaded at start-up and the registry keys associated with autostart.
`wmic process list brief | find "cmd.exe"`	Search for a specific process name, such as cmd.exe

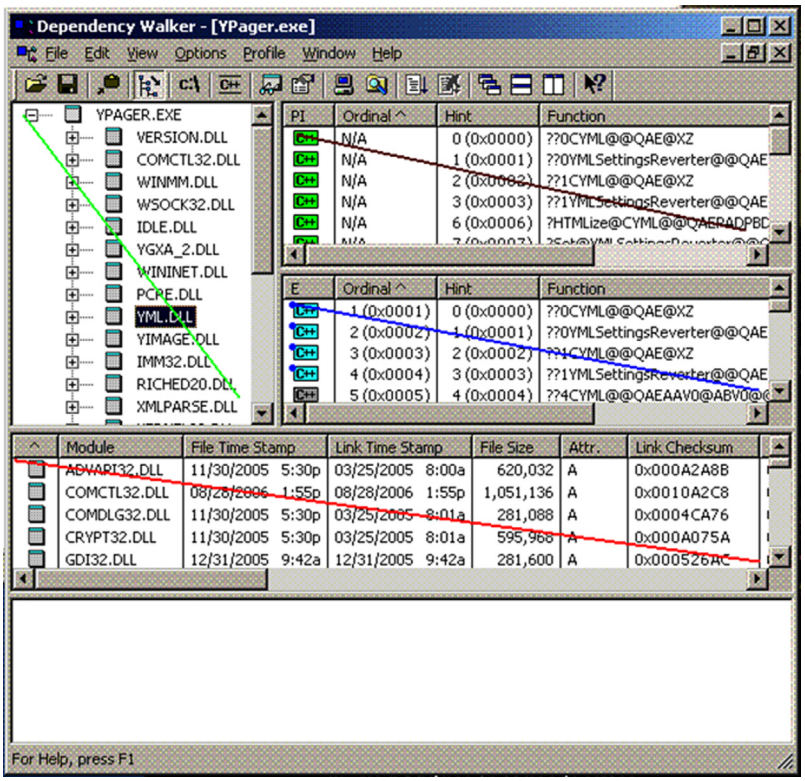

FIGURE 41.8 Using the Dependency Walker.

the system. Although a code-signing requirement was added in Windows 7 and Windows Server 2008 R2, it is still being utilized by various malicious software to hide and persist on a system.

The Volatility Framework can also be used to analyze the registry that has been loaded to memory. The plugins that are available from the Volatility Framework for registry analysis are shown in Table 41.14.

A listing of services can be obtained from the registry in memory from the registry key "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services." This key contains the database of services and device drivers that get read into the Windows service control manager's (SCM) internal database. SCM is a remote procedure call (RPC) server that interacts with the Windows service processes.

Other registry keys that might be of interest are the keys located in the registry key "HKEY_LOCAL_MACHINE\SOFTWARE\ Microsoft\ Windows\CurrentVersion." The keys that are part of the autostart registries are "Run," "RunOnce," "RunOnceSetup," "RunOnceEx," "RunServices," and "RunServicesOnce". Malware can set values in these keys to persist across system restarts and get loaded during system start up.

Deploying and Detecting a Rootkit

Rootkits are composed of several tools (scripts, binaries, and configuration files) that permit malicious users to hide their actions on a system so they can control and monitor the system for an indefinite time. Rootkits can be installed either through an exploit payload or installed after system access has been achieved. Rootkits are usually used to provide concealment, command and control (C2), and surveillance. A rootkit most of the time will try to hide system resources, such as processes, Registry information, files, and network ports. API hooking is a popular rootkit technique that intercepts system calls to make the OS report inaccurate results that conceal the presence of the rootkit. To skip all of the system-level subversion, we can look into the memory directly to detect rootkits. The Volatility Framework provides various plugins to detect rootkit concealment techniques as seen in Table 41.15:

10. SUMMARY

In this chapter we have seen the importance of having a well-documented incident response plan and process, and having an incident response team that is experienced in cyber forensics analysis. Besides having these important components, an organization needs to have strong policies and procedures that back them. Incident response is not only about countering the incident, but also about learning from it and improving on the weaknesses exposed. We should always keep in mind that preparedness is paramount since it is a matter of when rather than if an incident will strike.

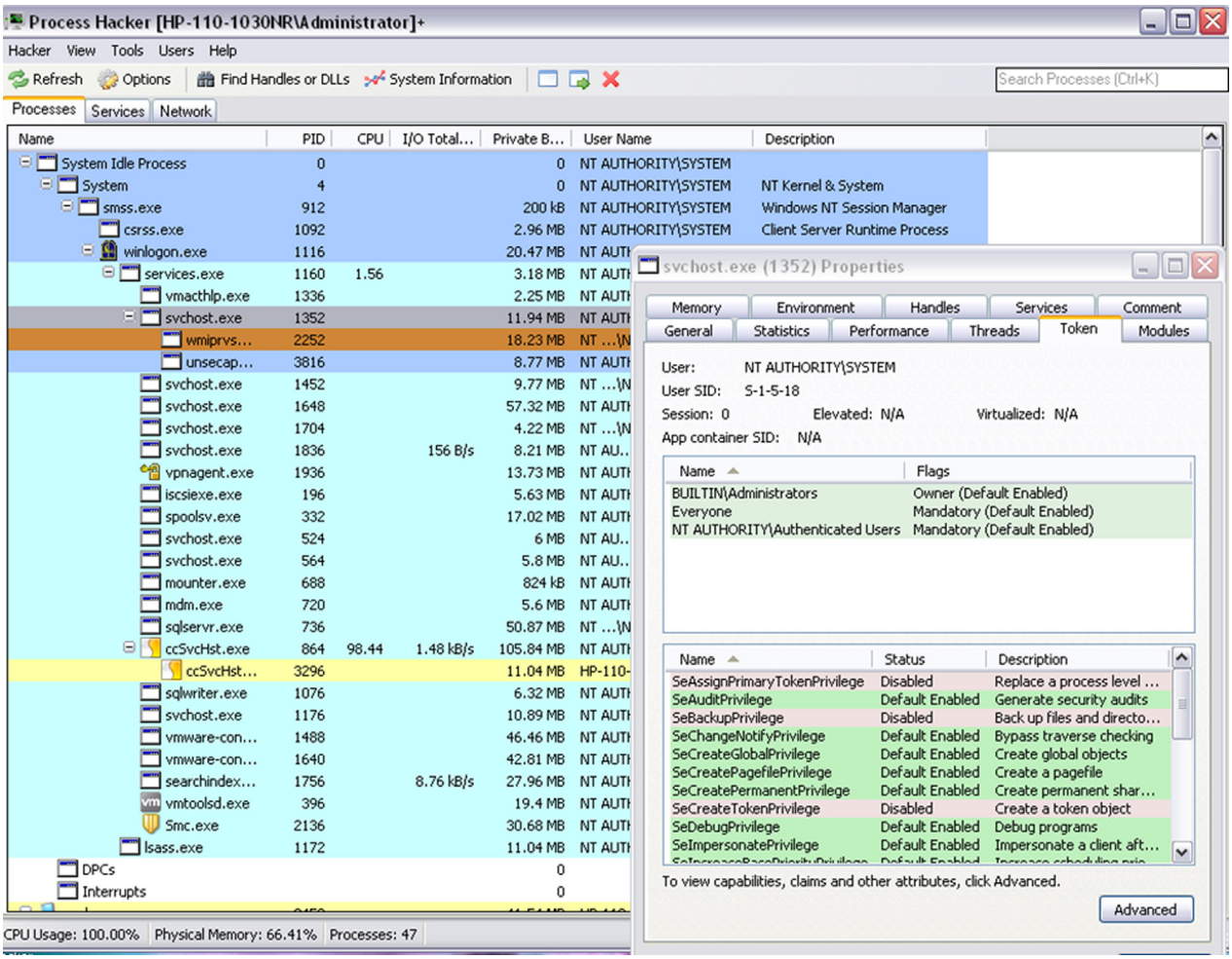

FIGURE 41.9 Using Process Hacker to view process and service information.

TABLE 41.14 List of Volatility Framework Plugins to Extract Registry Information From Memory

hivescan	Finds the physical address of CMHIVE structures, which represent a registry hives in memory.
hivelist	Takes a physical address of one CMHIVE, returns the virtual address of all hives, and their names.
printkey	Takes a virtual address of a hive and a key name (e.g., 'ControlSet001\Control'), and display the key's timestamp, values, and subkeys.
hashdump	Dump the LanMan and NT hashes from the registry.
lsadump	Dump the LSA secrets (decrypted) from the registry.
cachedump	Dump any cached domain password hashes from the registry.

Looking into the near future, the amount of data that needs to be gathered and analyzed is increasing rapidly and as a result we are seeing the emergence of big-data analytics tools that can process disparate data sources to deal with large cases. Tomorrow's incident response teams will need to be skilled in statistical analysis as well as forensics to be able to navigate in this increasingly hostile and expanding cyberspace. As you can see, incident response and cyber forensics needs to be a step ahead of the potential causes of threats, risks, and exploits.

Finally, let's move on to the real interactive part of this Chapter: review questions/exercises, hands-on projects,

TABLE 41.15 List of Volatility Framework Plugins to Detect Rootkit Hooking

Rootkit Technique	The Volatility Framework Plugin
IAT (import address table) hooks	apihooks: detect overwritten IAT entry for a PE file
EAT (export address table) hooks	apihooks
Inline application program interface (API) hooks	apihooks
IDT (interrupt descriptor table) hooks	idt: detects overwritten IDT entries that point to malicious interrupts ot processor exceptions
Driver IRP (I/O request packets) hooks	driverirp: detects overwritten IRP function table entries (modified to monitor buffer data)
SSDT (system service descriptor table) hooks	ssdt: detects hooking of pointers to kernel mode functions in the SSDT that occurs per thread
Hiding with orphan threads in kernel	threads: detects orphan threads that can unlinking or unloading its driver

case projects, and optional team case project. The answers and/or solutions by chapter can be found in Appendix K.

CHAPTER REVIEW QUESTIONS/ EXERCISES

True/False

1. True or False? Cyber forensics and incident response go hand in hand.
2. True or False? In an organization there is a daily occurrence of events within the IT infrastructure, but not all of these events qualify as incidents.
3. True or False? Forensic analysis is not usually applied to determine who, what, when, where, how, and why an incident took place.
4. True or False? When an incident response team comes across incidents relevant to these laws, they should consult with their legal team.
5. True or False? An organization should be prepared beforehand to properly respond to incidents and mitigate them in the longest time possible.

Multiple Choice

1. How do incident response and cyber forensics fit together?
   A. They don't
   B. Incident response helps cyber forensics in analyzing evidence
   C. Cyber forensics provides answers to questions that need to be answered for proper incident response
   D. None of the above
   E. All of the above
2. Which option might be classified as an incident?
   A. Phishing attack
   B. Unauthorized access
   C. Intellectual property theft
   D. Denial of service attack
   E. All of the above
3. Which one should be considered volatile data and captured immediately?
   A. Configuration files
   B. Database files
   C. Documents
   D. System memory
   E. All of the above
4. Which considerations would be involved in monitoring employee email?
   A. Technical factors
   B. Legal factors
   C. Organizational factors
   D. All of the above
   E. None of the above
5. Which tool can be used to extract the MFT of a Windows system from a drive?
   A. The Sleuth Kit
   B. The Volatility Framework
   C. The KntDD
   D. Truecrypt
   E. All of the above

EXERCISE

Problem

How does an organization ship their hard drives?

Hands-On Projects

Project

How does an organization get their data back?

Case Projects

Problem

As an exercise to practice what you have learned in this chapter, you will analyze your own Windows workstation:

1. Create an image of your disc drive and save it to an external drive using AccessData FTK Imager.
2. Create a memory sample from your system using MoonSols Windows Memory Toolkit and save it to an external drive.
3. Create a super timeline of the disc image using log2timeline and list the files created within 24 h.
4. Using the Volatility Framework, get a list of the following objects from the memory sample: processes, DLLs, modules, services, connections, sockets, API hooks.

Optional Team Case Project

Problem

When should an organization consider using a computer forensic examiner?

REFERENCES

[1] Disk Imaging Tool Specification, NIST, 2001.
[2] Guide to Integrating Forensic Techniques into Incident Response, NIST, 2006.