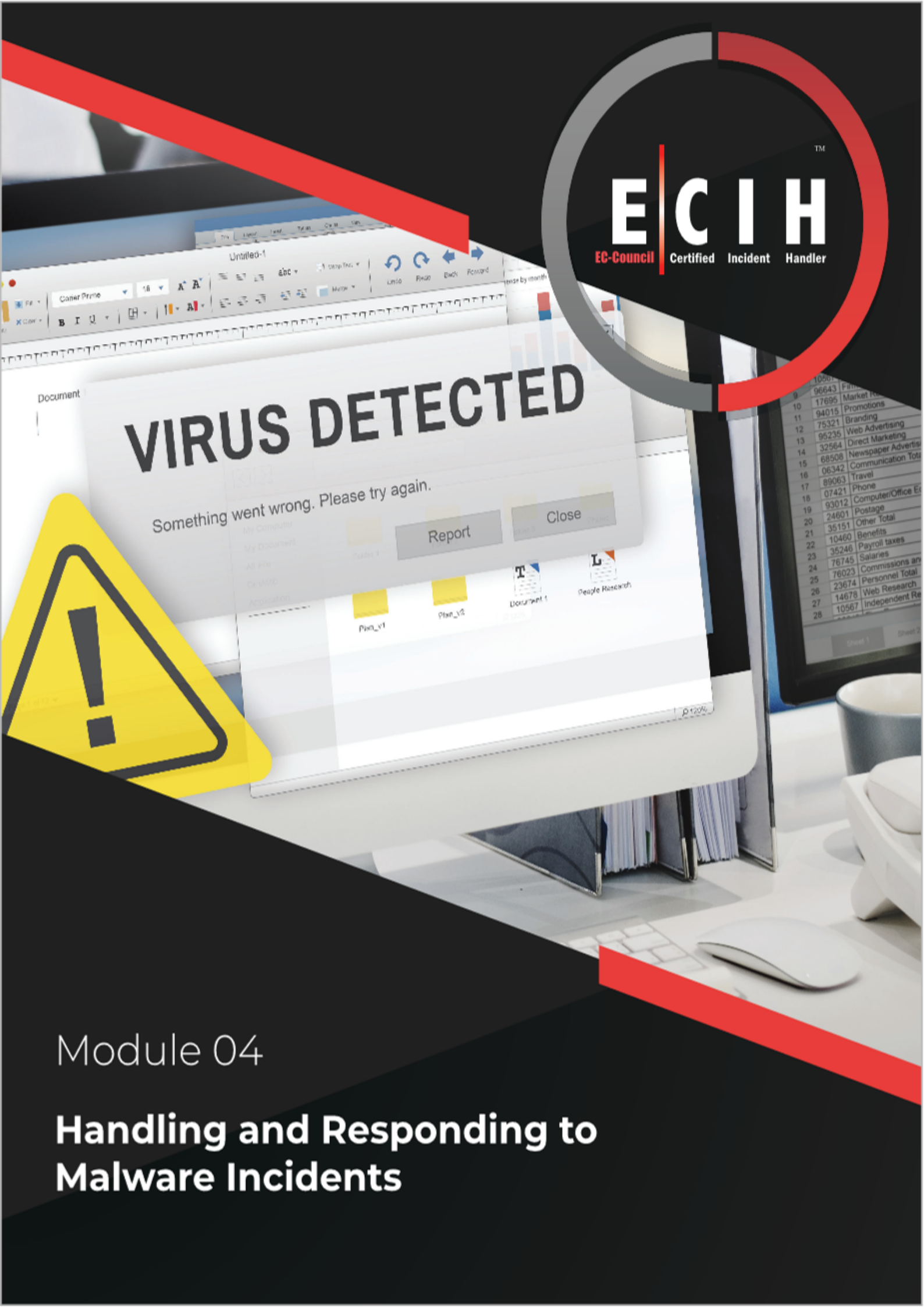
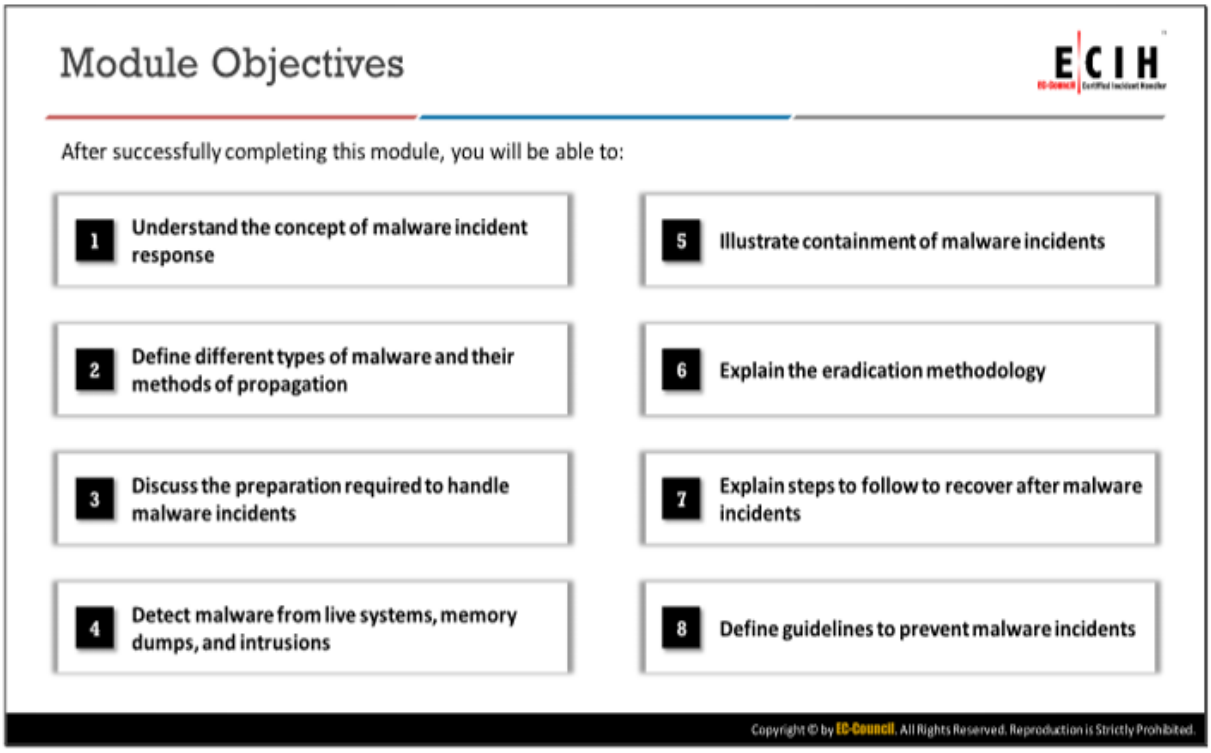


Module 04

Handling and Responding to Malware Incidents

This page is intentionally left blank.

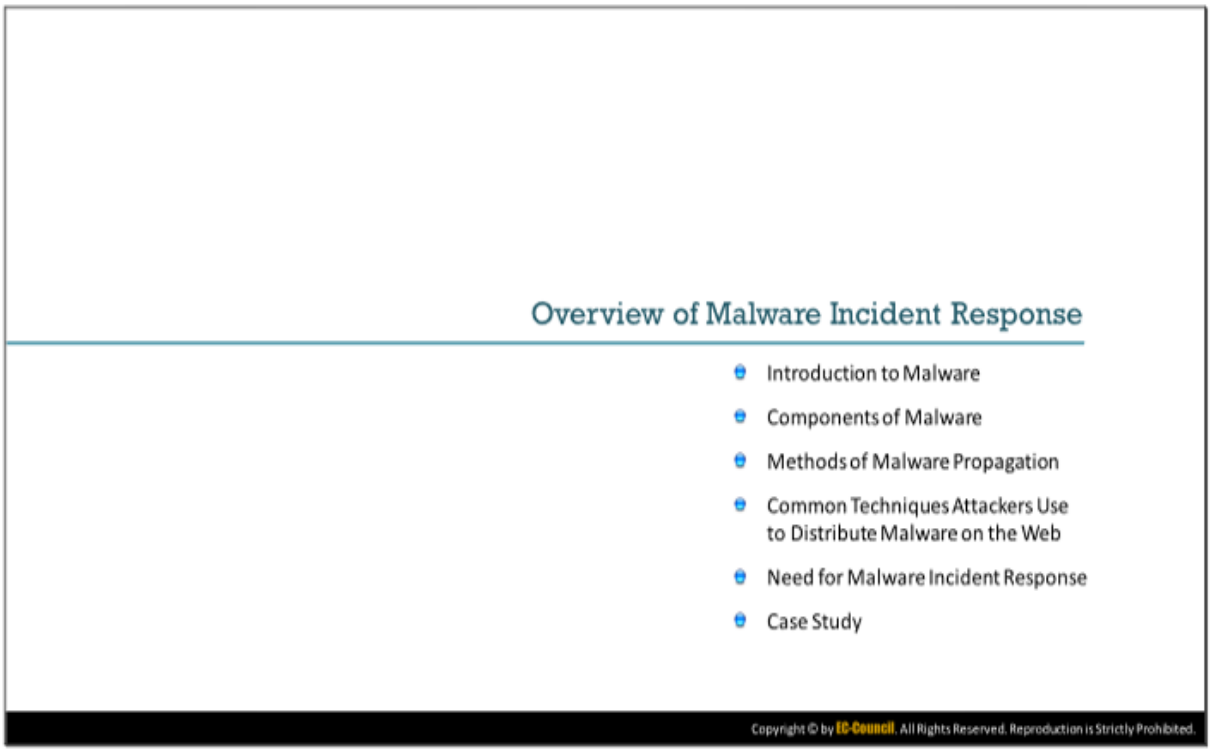


Module Objectives

This module describes different malicious code incidents and media through which they are propagated. It also provides step-by-step descriptions for handling malicious code incidents and tips to prevent them.

At the end of this module, you will be able to:

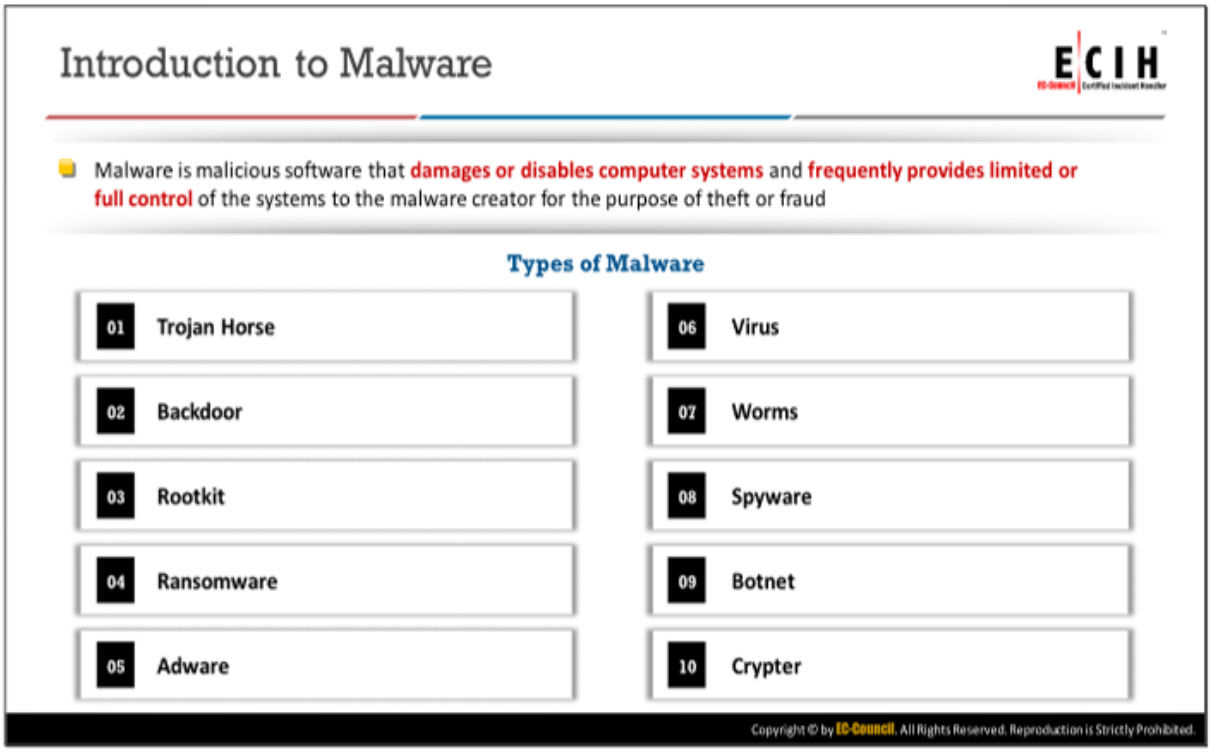
- Understand the concept of malware incident response (IR)
- Define different types of malware and their propagation
- Discuss preparation required to handle malware incidents
- Detect malware from live systems, memory dumps, and intrusions
- Illustrate containment of malware incidents
- Explain eradication methodology
- Explain steps required to recover after malware incidents
- Define guidelines to prevent malware incidents



Overview of Malware Incident Response

Malware attacks are the most common attacks against any enterprise, and as incident handling and response (IH&R) personnel, you must be aware of steps that must be followed to handle malware incidents. We first outline basic concepts of malware to understand importance of malware IR.

This section discusses fundamental malware concepts including types and components of malware, methods of malware propagation, and common techniques attackers use to distribute malware on the web. This section also discusses need for malware IR and presents a case study.



Introduction to Malware

Malware is malicious software that damages or disables computer systems and gives limited or full control of them to its developer for theft or fraud. Malware includes viruses, worms, Trojans, rootkits, backdoors, botnets, ransomware, spyware, adware, scareware, crapware, roughware, crypters, keyloggers, etc., which may delete files, slow down computers, steal personal information, send spam, and commit fraud. Malware can perform various malicious activities ranging from simple email advertising to complex identity and password theft. Malware programmers develop and use it to:

- Attack browsers and track websites visited
- Slow system performance
- Cause hardware failure, rendering computers inoperable
- Steal personal information, including contacts
- Erase valuable information, resulting in substantial data losses
- Attack additional computer systems directly from a compromised system
- Spam inboxes with advertising emails

Types of Malware

The following are types of malware that attackers have used:

- **Trojan Horse**

  A Trojan is a program in which malicious or harmful code is contained inside apparently harmless programs or data such that it can gain control of and damage a system, such as by ruining the file allocation table on your hard disk. Attackers use Trojan horses to trick

the victim into performing predefined actions. Trojans are activated upon users' certain predefined actions like installing malicious software unintentionally and clicking on malicious links, and upon activation, Trojans can grant attackers unrestricted access to all data stored on compromised information systems, thereby causing potentially immense damage.

For example, users could download a file that appears to be a movie. However, when executed, the file unleashes a dangerous program that erases the hard drive or sends credit card numbers and passwords to the attacker. A Trojan is wrapped within or attached to a legitimate program, meaning that the program may have functionality not apparent to the user. In addition, attackers use victims as unwitting intermediaries to attack others.

- **Backdoor**

  A backdoor is a program that can bypass standard system authentication or conventional system mechanisms like intrusion detection systems (IDSs) and firewalls without being detected. In such breaches, hackers leverage backdoor programs to access the victim's computer or network. The difference between this and other types of malware is that backdoors are installed without the user's knowledge, thereby enabling attacks to perform any activity on the infected computer, which can include transferring, modifying, and corrupting files; installing malicious software; rebooting the machine; etc. without user detection. Backdoors are used by the attacker to gain uninterrupted access to the target machine. Most backdoors are used for targeted attacks. Backdoor Trojans are often used to group victim computers into a botnet or zombie network that can be used to perform criminal activities.

- **Rootkit**

  Rootkits are software programs aimed to gain access to a computer without detection. These are malware that help attackers gain unauthorized access to a remote system and perform malicious activities. The goal of the rootkit is to gain root privileges to a system. By logging in as the root user of a system, an attacker can perform any task such as installing software or deleting files. Rootkits work by exploiting vulnerabilities in the operating system and applications and build backdoor logins in the operating system by which the attacker can evade the standard login process.

  Once the user enables root access, a rootkit may attempt to hide traces of unauthorized access by modifying drivers or kernel modules and discarding active processes. Rootkits replace certain operating system calls and utilities with the rootkits' own modified versions, which in turn undermine security of the target system by executing malicious functions. A typical rootkit comprises backdoor programs, distributed denial of service (DDoS) programs, packet sniffers, log-wiping utilities, internet relay chat (IRC) bots, and others.

- **Ransomware**

  Ransomware is a type of malware that restricts access to the computer system it infects, or to critical files and documents stored on the infected system and thereafter demands

an online ransom payment to the malware developer(s) to remove user restrictions. Ransomware might encrypt files stored on the system's hard disk or merely lock the system and display messages meant to trick the user into paying.

Usually, ransomware spreads as a Trojan, entering a system through email attachments, hacked websites, infected programs, app downloads from untrusted sites, vulnerabilities in network services, etc. After ransomware execution, the ransomware payload runs and encrypts the victim's data (files and documents), which can be decrypted only by the malware author. In some cases, user interaction is restricted using a simple payload.

Some common ransomwares include Locky, WannaCry, Petya–NotPetya, GoldenEye, Chimera, Hidden Tear & EDA2, Fantom, Mischa, Shark, and HolyCrypt.

- **Adware**

  Adware refers to software used to display online advertisements in the user interface or on a screen and generate revenue. Attackers use this adware property to display malicious advertisements redirecting users to malicious websites that collect user data without their consent or automatically download other malware.

- **Virus**

  Viruses are the scourge of modern computing and have potential to wreak havoc on both business and personal computers. A computer virus is a self-replicating program that reproduces its code by attaching copies of itself to other executable codes and operates without the user's knowledge or desire. Like a biological virus, a computer virus is contagious and can contaminate other files; however, viruses can infect outside machines only with assistance of computer users. Virus lifetime depends on a virus's ability to reproduce. Therefore, attackers design every virus code such that the virus replicates itself *n* times.

  Some viruses affect computers as soon as their code is executed; others lie dormant until a predetermined logical circumstance is met. Viruses infect a variety of files such as overlay files (.OVL) and executable files (.EXE, .SYS, .COM, or .BAT). Viruses are transmitted through file downloads, infected disk/flash drives, and email attachments.

- **Worms**

  Computer worms are standalone malicious programs that replicate, execute, and spread across network connections independently, without human intervention. Intruders design most worms to replicate and spread across a network, thus consuming available computing resources and in turn causing network servers, web servers, and individual computer systems to become overloaded and stop responding. However, some worms also carry a payload to damage the host system.

  Worms are a subtype of viruses. Although a worm does not require a host machine on which to replicate, the worm's host machine sometimes infects other machines. Initially, Black Hat® professionals treated worms as a mainframe problem. However, following introduction of the internet, attackers mainly concentrated on and targeted Windows

operating systems (OSs) by sharing the same worms in emails and IRCs and through other network functions.

Attackers use worm payloads to install backdoors on infected computers, thereby turning infected computers into zombies and creating botnets that attackers can use to initiate cyberattacks. Some of the latest computer worms include:

- KjW0rm
- SONAR.ProcHijack!g15
- W32.Emotet.B

- **Spyware**

  Spyware is stealthy computer monitoring software that allows attackers to secretly record all user activities on a target computer. It automatically delivers logs to the remote attacker through the internet (by email; file transfer protocol (FTP); command and control through encrypted traffic, hypertext transfer protocol (HTTP), domain name system (DNS) protocol, etc.). Delivery logs include information about all areas of the system such as emails sent, websites visited, all keystrokes (including logins/passwords of Gmail, Facebook, Twitter, LinkedIn, etc.), file operations, and online chat conversations. Spyware also takes screenshots at set intervals, just like a surveillance camera aimed at the computer monitor.

  Spyware is similar to a Trojan horse, which is usually bundled as a hidden component of freeware or software downloaded from the internet. It hides its process, files, and other objects to avoid detection and removal. It allows an attacker to gather information about a victim or organization such as email addresses, user logins, passwords, credit card numbers, and banking credentials.

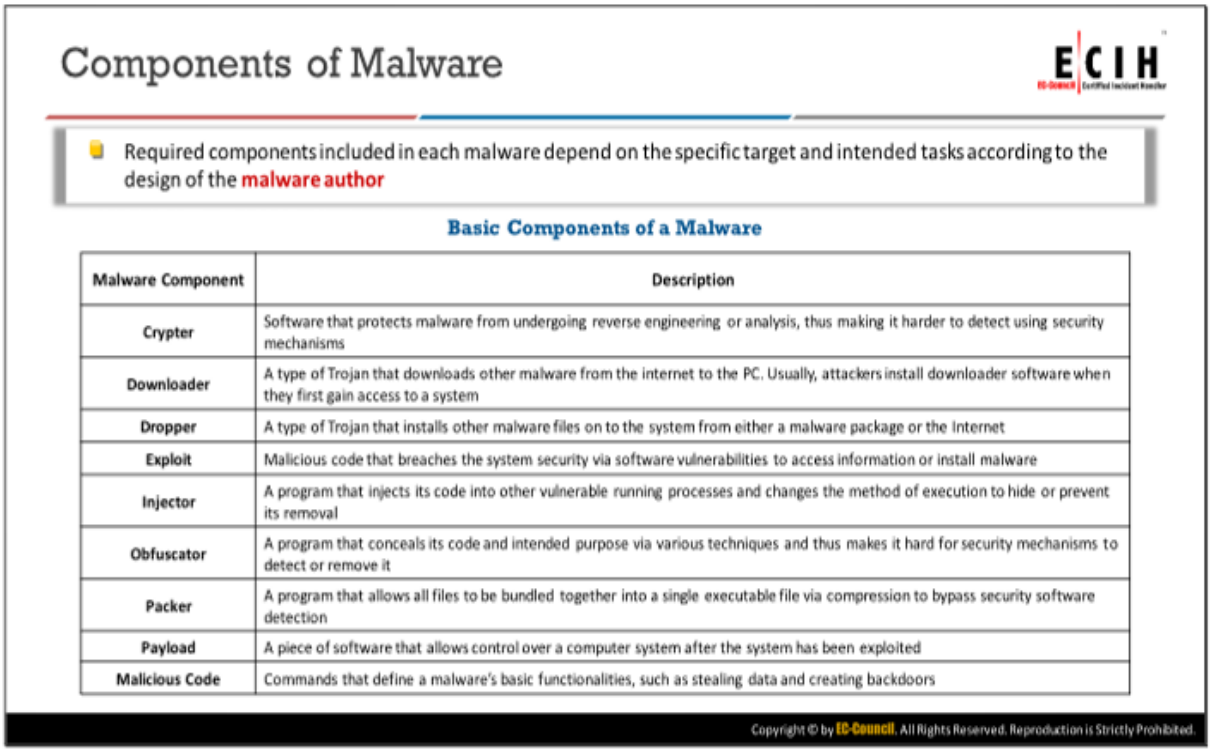
- **Botnet**

  A botnet is a huge network of compromised systems used by attackers to perform denial-of-service (DoS) attacks. Bots are software applications that run automated tasks over the internet. There are different types of bots, including internet, IRC, and chatter bots. Attackers use bots for benign data collection or data mining, such as "web spidering," and to coordinate DoS attacks. The main purpose of a bot is to collect data. Bots in a botnet perform tasks such as uploading viruses, sending emails with botnets attached to them, stealing data, etc.

  Botnets are agents that an intruder can send to a server system to perform some illegal activity. They are hidden programs that allow identification of system vulnerabilities. Attackers can use botnets to perform tedious tasks involved in probing a system for known vulnerabilities.

- **Crypters**

  Crypters are software that encrypts the original binary code of an .exe file. Attackers use crypters to hide viruses, spyware, keyloggers, and remote access Trojans (RATs) among others, thereby rendering malware undetectable by antiviruses.



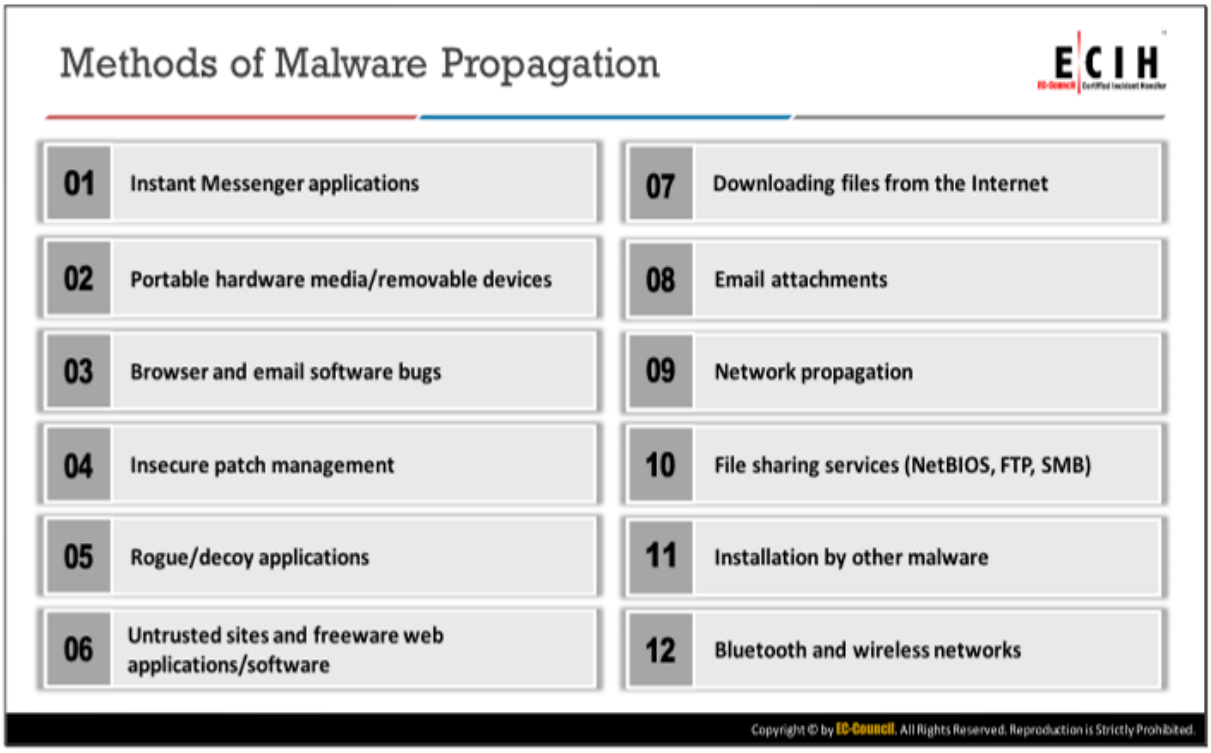
Components of Malware

Malware authors and attackers develop malware using components to help them achieve their goals. They can use malware to steal information, delete data, change system settings, provide access, or merely multiply and occupy system space. Malware is capable of propagating and functioning secretly.

Some essential components of most malware programs include:

- **Crypter**: Refers to a software program that can conceal malware. Attackers use this software to elude antivirus detection. It protects malware from being reverse engineered or analyzed and is thus difficult to detect by the security mechanism.
- **Downloader**: Refers to a Trojan that downloads other malware (or) malicious code and files from the internet onto a personal computer (PC) or device. Usually, attackers install a downloader when they first gain access to a system.
- **Dropper**: Attackers must install the malware program or code on the system to make it run, and this program can do the installation task covertly. The dropper can contain unidentifiable malware code undetected by antivirus scanners and is capable of downloading additional files needed to execute malware on a target system.
- **Exploit**: Part of the malware that contains code or a sequence of commands that can take advantage of a bug or vulnerability in a digital system or device. It is the code that attackers use to breach system security through software vulnerabilities, to spy information, or to install malware. Exploits are categorized according to the type of vulnerabilities they abuse; for example, local and remote exploits.

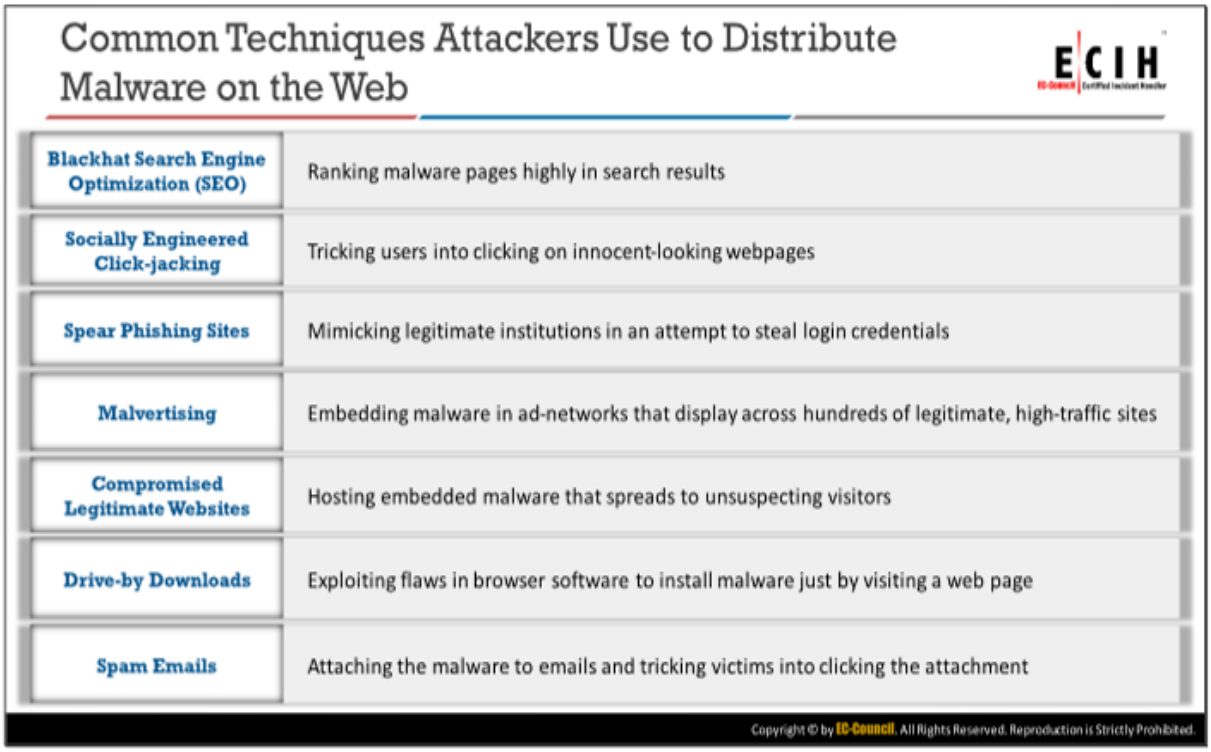
- **Injector**: This program injects exploits or other malicious code available in malware into other vulnerable running processes and changes process execution to hide or prevent removal.
- **Obfuscator**: A program to conceal malware malicious code by various techniques, thus making it difficult for security mechanisms to detect or remove it.
- **Packer**: This software compresses the malware file to convert malware code and data into an unreadable format. Packers use compression techniques to pack malware.
- **Payload**: The part of the malware that performs the desired activity when activated. The payload may be used to delete files or other information, modify files, affect system performance, open ports, change settings, etc. as part of compromising system security.
- **Malicious Code**: This is a piece of code that defines basic malware functionality and comprises commands that result in security breaches. It can many take forms including:
  - Java applets
  - ActiveX controls
  - Browser plug-ins
  - Pushed content



Methods of Malware Propagation

Incident handlers and responders must understand different methods attackers use to spread malware from one system to another across an organization. The most common methods attackers use to infect a system with malware include:

- Instant messenger applications
- Portable hardware media/removable devices
- Browser and email software bugs
- Insecure patch management
- Rogue/decoy applications
- Untrusted sites and freeware web applications/software
- Downloading internet-based files
- Email attachments
- Network propagation
- File sharing services [network basic input/output system (NetBIOS); file transfer protocol (FTP); server message block (SMB)]
- Installation by other malwares
- Bluetooth and wireless networks
- Infected executables, dynamic link library (DLL) files, macros, JavaScripts, and Documents



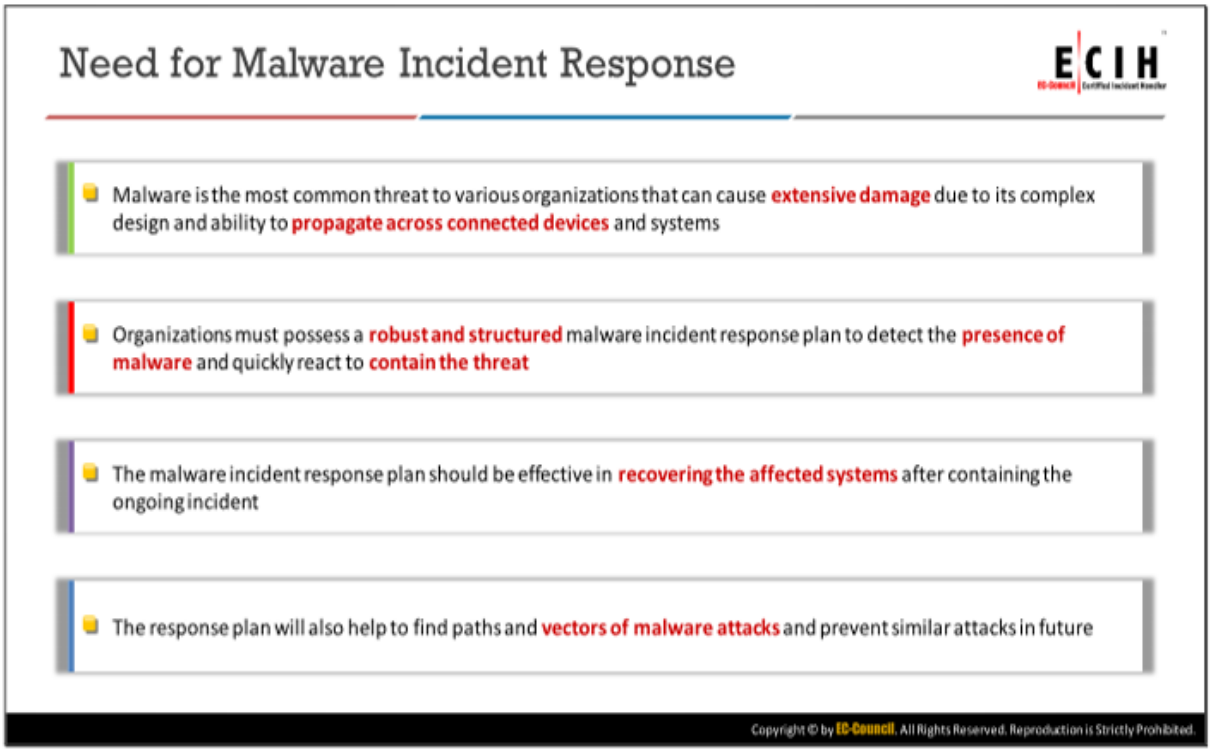
Common Techniques Attackers Use to Distribute Malware on the Web

Source: *Security Threat Report* (*https://www.sophos.com*)

Following are some standard techniques used to distribute malware on the web:

- **Black Hat® Search Engine Optimization (SEO)**: Black Hat® SEO (also referred to as unethical SEO) uses aggressive SEO tactics such as keyword stuffing, doorway pages, page swapping, and adding unrelated keywords to obtain higher search engine rankings for malware pages.
- **Socially Engineered Click-jacking**: Attackers inject malware into legitimate-looking websites to trick users into clicking links to malware. When clicked, malware embedded in the link executes without user knowledge or consent.
- **Spear Phishing Sites**: This technique is used to mimic legitimate institutions such as banks in an attempt to steal passwords, credit card and bank account data, and other sensitive information.
- **Malvertising**: This involves embedding malware-laden advertisements in legitimate online advertising channels to spread malware onto systems of unsuspecting users.
- **Compromised Legitimate Websites**: Often, attackers use compromised websites to infect systems with malware. When an unsuspecting user visits the compromised website, malware is unknowingly installed on the user's system and thereafter executes malicious activities.
- **Drive-by Downloads**: This refers to unintentionally downloading software from the internet. Here, an attacker exploits flaws in browser software to install malware merely by visiting a website.

- **Spam Emails**: An attacker attaches a malicious file to an email and sends the email to multiple target addresses. The attacker tricks the victim into clicking the attachment. When the attachment is clicked, the malware executes, and the machine is compromised. This technique is the most common method attackers currently use to spread malware. Apart from email attachments, an attacker may also embed malware in an email body.



Need for Malware Incident Response

Malware is the most common threat in the current scenario and is responsible for extensive damage to various organizations owing to its complex design and ability to propagate across connected devices and systems. Therefore, organizations must possess a robust, structured malware IR plan to detect malware presence and quickly react to contain the threat. Malware IR will effectively recover affected systems after containing the ongoing incident. The response plan will also help identify paths and vectors of malware attacks and prevent future similar attacks.

Case Study

Challenge

Maria White, the managing director of an organization, found her system inaccessible and displaying the following image. On understanding this to be some type of a malware attack, she **contacted her incident response team** to investigate the issue. When the incident responders arrived, they found that over 30 other systems in the organization were impacted by a similar ransomware attack.

Process

The responders immediately **separated the impacted systems** from the functioning network and informed the Microsoft organization about the issue. They found that this issue had impacted systems on large scale as a result of using older versions of the Windows operating systems, which contained a vulnerability.

Because the systems were inaccessible, the responders **extracted the hard drives** of some systems to investigate the issue. They extracted the data and transferred it to a sandbox environment for analysis.

Case Study (Cont'd)

Solution

The responders immediately **patched the working system with updates** from Microsoft and initiated data analysis. During analysis, they found that the malware had encrypted all the files on the system. They tried performing static analysis of the files and found that the malware attempted to connect to an unregistered domain and showed signs of failure to connect. The team **used different malware analysis techniques** such as string searching, searching for portable executables (PEs), and file dependency identification, but all attempts were unsuccessful.

They found that the malware was using the domain request as a decryption key, and that any reply from the domain would release the systems. The response team then **used network simulation services such as iNetSim** to simulate a reply as from the domain requested by the malware. After the reply was received, the ransomware unlocked the system. The responders immediately used this technique on all the affected systems and patched them with an update from the manufacturer.

The responders also suggested that a **scheduled automatic update policy** must be implemented to prevent the exploitation of existing system vulnerabilities.

Case Study

The following case study shows the importance and need of malware IR to effectively handle malware security incidents:

- **Challenge**

  Maria White, organizational managing director, found her system inaccessible and displaying the following image. Understanding that it was some type of malware attack,

she contacted the IR team to investigate the issue. When the incident responders arrived, they found that over 30 other systems in the organization were impacted with a similar ransomware attack.

- **Process**

  The responders immediately separated impacted systems from the functioning network and informed the Microsoft organization about the issue. They found that the issue had impacted systems on a large scale and was a result of using older, vulnerable versions of Windows OSs.

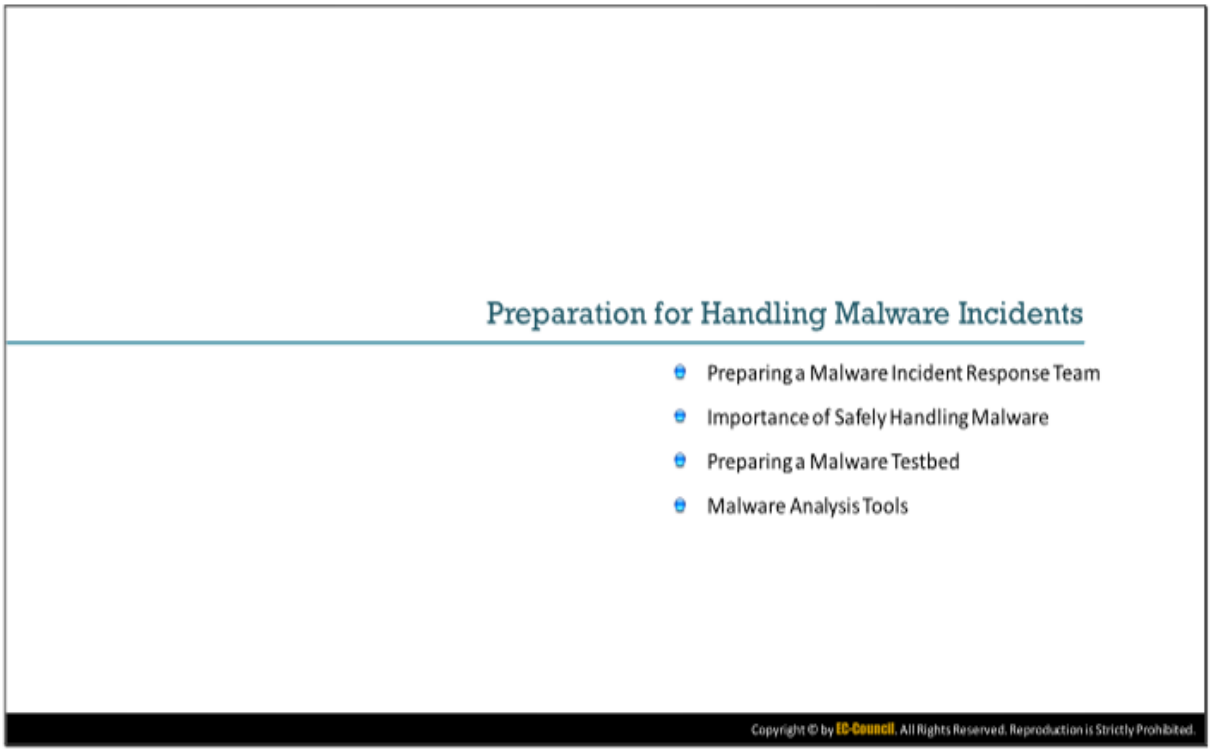
  As the systems were inaccessible, the responders extracted the hard drive memory of some systems. The responders extracted data and transferred it to the sandbox environment to start analysis.

- **Solution**

  The responders immediately patched the working system with Microsoft updates and started analyzing the data. During analysis, they found that the malware had encrypted all the files on the system. They tried to statically analyze the files and found that the malware was trying to connect to an unregistered domain and showed signs of connection failure. The team used different malware analysis techniques such as searching strings, searching for portable executable (PE) files, and identifying file dependencies, but all went in vain.

  They found that the malware was using the domain request as the decryption key and that any reply from the domain could free the systems. The response team then used network simulation services such as the internet services simulation (iNetSim) suite to simulate the reply as if it was from the domain requested by the malware. When applying the same, the ransomware unlocked the system. The responders immediately used this technique on all the systems and patched them with an update from the manufacturer.

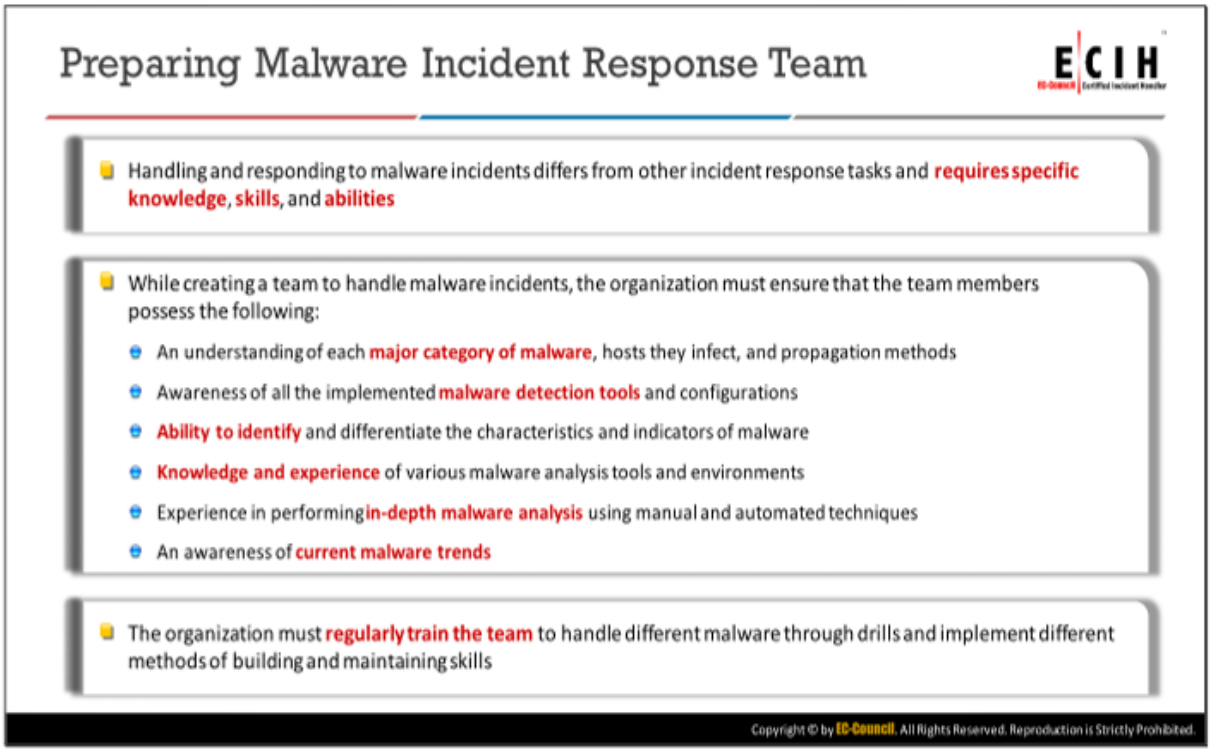
  The responders also suggested that the company must have a scheduled automatic update policy to prevent exploitation of existing system vulnerabilities.



Preparation for Handling Malware Incidents

Any organization can become prey to malware attacks. Handling malware is necessary to contain an ongoing incident. If handled carelessly, malware can severely damage the organization. Therefore, the IH&R team should be well prepared and equipped to handle malware-related security incidents.

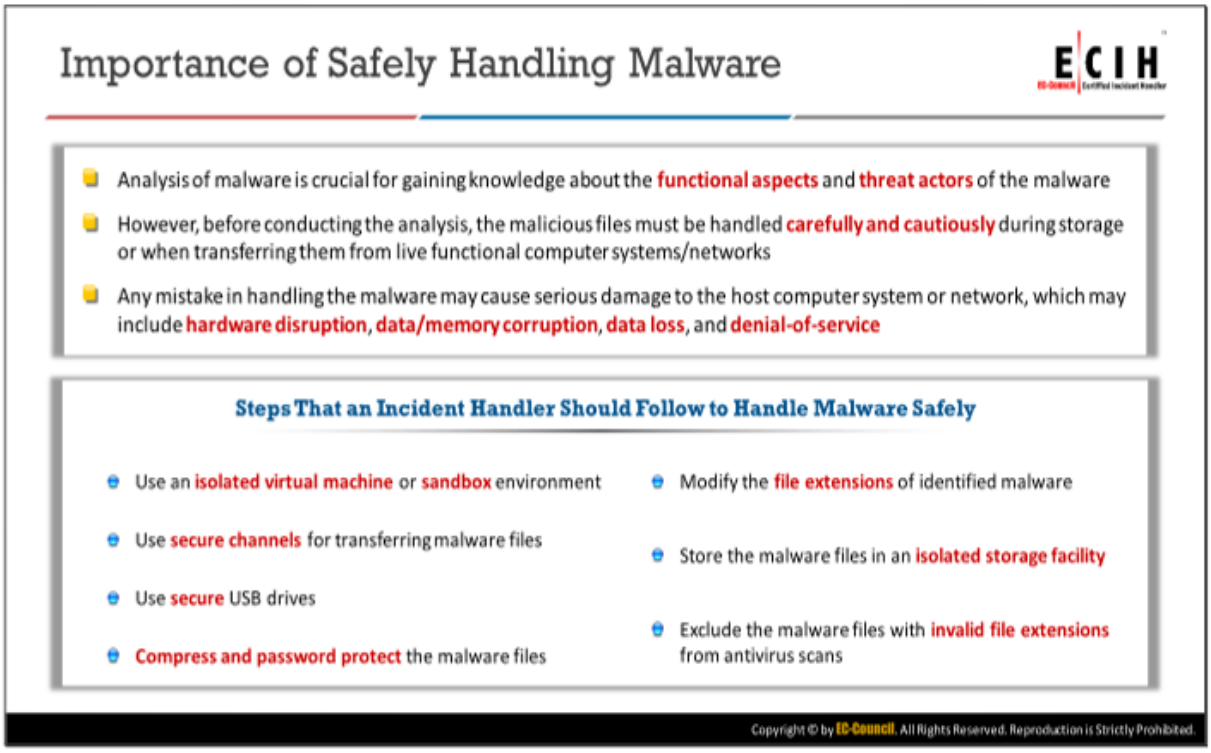
This section discusses the importance of safely handling malware, preparing malware IR teams, preparing a malware testbed, malware analysis, and handling tools.



Preparing Malware Incident Response Team

Handling and responding to malware incidents is different from other IR tasks and requires specific knowledge, skills, and abilities. Therefore, organizations should ensure that IR teams have skills and knowledge required to manage malware incidents. Each team member should have thorough knowledge of each major malware category, propagation method, and infected resources and know how to extract them from compromised devices. The team must include staff having experience in handling incidents, investigations, gathering evidence, and analyzing malware-related incidents. The team also should include vulnerability managers, information security managers, and penetration testers.

Organizations must equip teams with knowledge of organizational malware defense mechanisms, tools, and policies. Teams must understand use of different malware detection tools, techniques, and configurations required to extract malware from different systems and must be able to identify and differentiate characteristics and indicators of malware and have knowledge and experience of various malware analysis tools and environments used to analyze malware securely. They must have experience in deeply analyzing malware using manual and automated techniques. Team members must be aware of current malware trends and methods of detection.

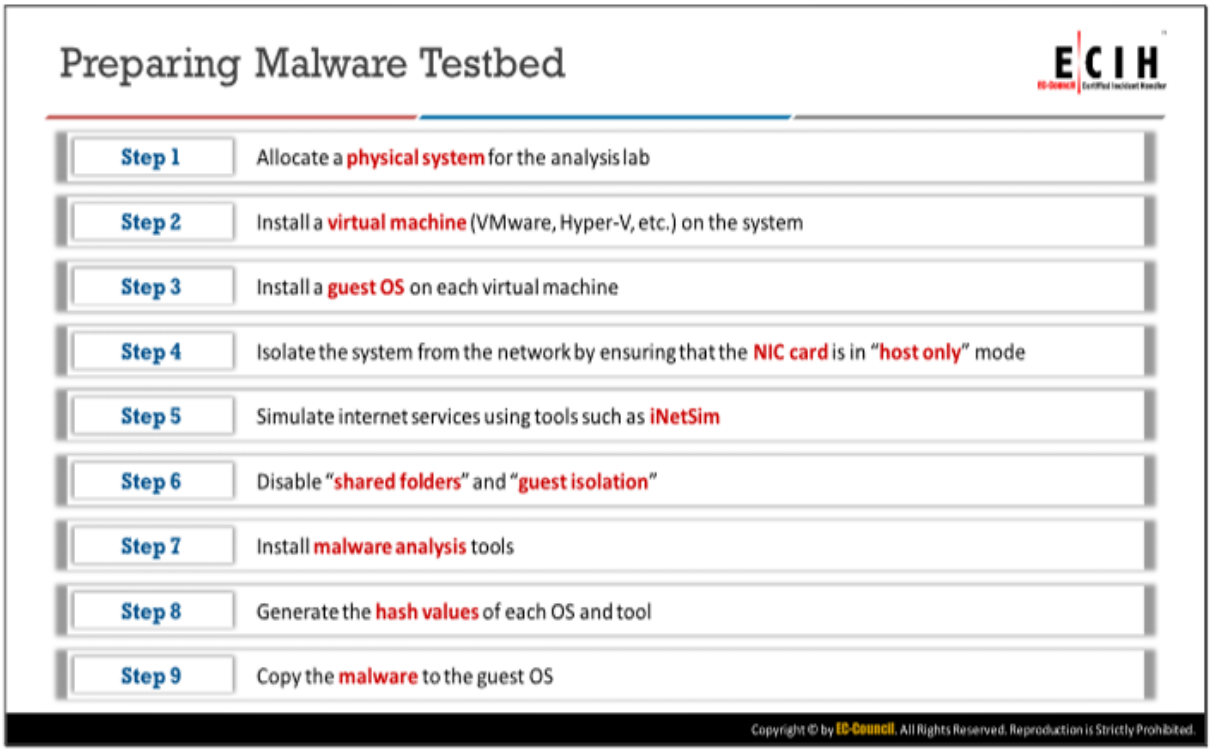


Importance of Safely Handling Malware

Malware analysis is crucial for gaining knowledge about malware functional aspects and threat actors. This knowledge enables the incident responder to develop defense plans against such malicious programs and prevent such future incidents from happening. However, before malware analysis, such malicious files must be handled carefully and cautiously while storing or transferring them from live functional computer systems/networks. Any mistake in handling malware may seriously damage the host computer system or network, which may include hardware disruption, data/memory corruption, data loss, and DoS. Therefore, as IH&R team personnel, you must have prior knowledge and awareness of safely handling malware and its importance.

Following are some steps that an incident handler should follow to handle malware safely:

- Always use a virtual machine or sandbox environment for handling malware.
- Ensure that the virtual machine or sandbox environment is isolated from functional network systems.
- Use secure channels dedicated for transferring malware files.
- Use secure universal serial bus (USB) drives dedicated for transferring malware.
- Keep malware files zipped and password protected to avoid accidental execution.
- Modify identified malware file extensions or add an invalid file extension to malware files to ensure no application is associated with it.
- Store malware files in an isolated storage facility.
- Exclude malware files with invalid file extensions and the directory where malware files are stored from antivirus scans.



Preparing Malware Testbed

The malware analysis lab or testbed must include sandbox environments that can help in developing an isolated network to analyze malware. Ensure that the sandboxes have enough processing power and media to install and run different virtual machines for testing malware. The lab should have tools required for simulating network and databases appropriately. The sandbox must not connect with the functional network as it can disrupt business during malware analysis.

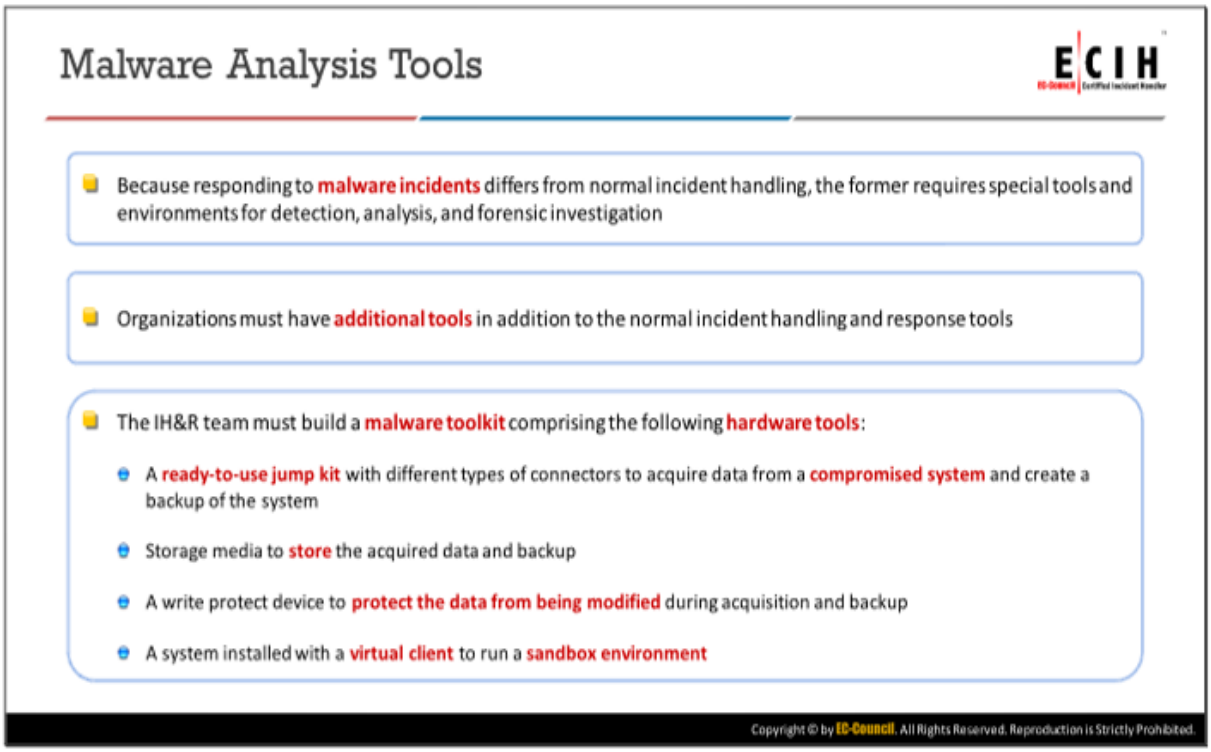
Requirements for building a testbed:

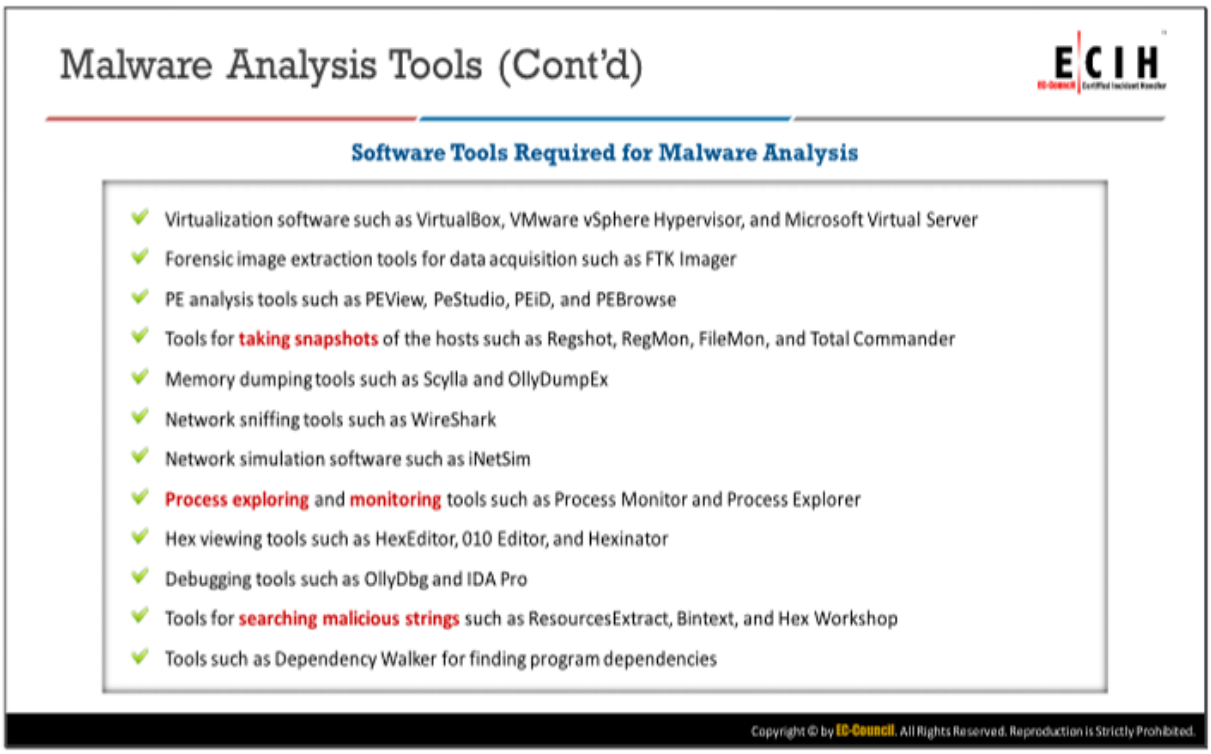
- An isolated test network to host your test bed and isolated network services such as DNS
- Victim machines installed with a variety of OSs and configuration states (nonpatched, patched, etc.)
- Virtualization snapshots and reimaging tools to wipe and rebuild the victim's machine quickly
- Some tools are required for testing. Following are the more important ones:
  - **Imaging tool**: To obtain a clean image for forensics and prosecution
  - **File/data analysis**: To statically analyze potential malware files
  - **Registry/configuration tools**: To help identify the last saved settings because malware infects the Windows registry and other configuration variables.
  - **Sandbox**: To manually perform dynamic analysis

- **Log analyzers**: To extract log files because devices under attack record malware activities and generate log files.
- **Network capture**: To understand how malware leverages the network

Steps required to prepare the testbed:

- **Step 1**: Allocate a physical system for the analysis lab
- **Step 2**: Install a virtual machine (VMware, Hyper-V, etc.) on the system
- **Step 3**: Install guest OSs on the virtual machine(s)
- **Step 4**: Isolate the system from the network by ensuring that the network interface controller (NIC) card is in "host only" mode.
- **Step 5**: Simulate internet services using tools such as iNetSim
- **Step 6**: Disable "shared folders" and "guest isolation"
- **Step 7**: Install malware analysis tools
- **Step 8**: Generate the hash value of each OS and tool
- **Step 9**: Copy malware over to the guest OS





Malware Analysis Tools

Malware IH&R is different and requires special tools and environments for detection, analysis, and forensic investigation because malware can impact other systems in the organization if not handled carefully. Therefore, organizations must provide malware IR teams with additional tools and malware toolkits.

Hardware Tools

The malware toolkit must include hardware tools required to carefully extract, save, and transport malware. It should have tools to encrypt or lock malware to stop it from spreading to other systems.

The IH&R team must build a malware toolkit having the following hardware tools:

- A ready-to-use jump kit with different types of connectors to acquire and back up data from the compromised system
- Storage media to store the acquired and backed up data
- A write-protect device to prevent data modification during acquisition and back up
- A system installed with a virtual client to run the sandbox

Software Tools

The malware toolkit must include a laptop equipped with software tools, devices to store data backups, hardware required to connect to compromised devices, and basic networking equipment and cables. Incident responders should always use removable devices such as digital versatile discs (DVDs) and USBs to collect and transfer suspect malware files from compromised systems to analysis ones.

Software tools required for malware detection and analysis include:

- Virtualization software such as VirtualBox, VMware vSphere Hypervisor, and Microsoft Virtual Server
- Forensic image extraction tools such as Forensic Toolkit® (FTK®) Imager for data acquisition
- PE analysis tools such as PEView, PeStudio, PEiD, and PEBrowse
- Tools for taking snapshots of hosts such as Regshot, RegMon, FileMon, and Total Commander
- Memory dumping tools such as Scylla and OllyDumpEx
- Network sniffing tools such as WireShark
- Network simulation software such as iNetSim
- Process exploring and monitoring tools such as Process Monitor and Process Explorer
- Hex viewing tools such as HexEditor, 010 Editor, and Hexinator
- Debugging tools such as OllyDbg and IDA Pro
- Tools for searching malicious strings including ResourcesExtract, Bintext, and Hex Workshop
- Tools such as Dependency Walker for finding program dependencies

Other Tools for Supporting Malware Analysis

Following are some supporting tools required to perform malware analysis:

Virtual-Machine Tools

- Hyper-V (*https://docs.microsoft.com*)
- Parallels Desktop 14 (*https://www.parallels.com*)
- Boot Camp (*https://www.apple.com*)

Screen-Capture and Recording Tools

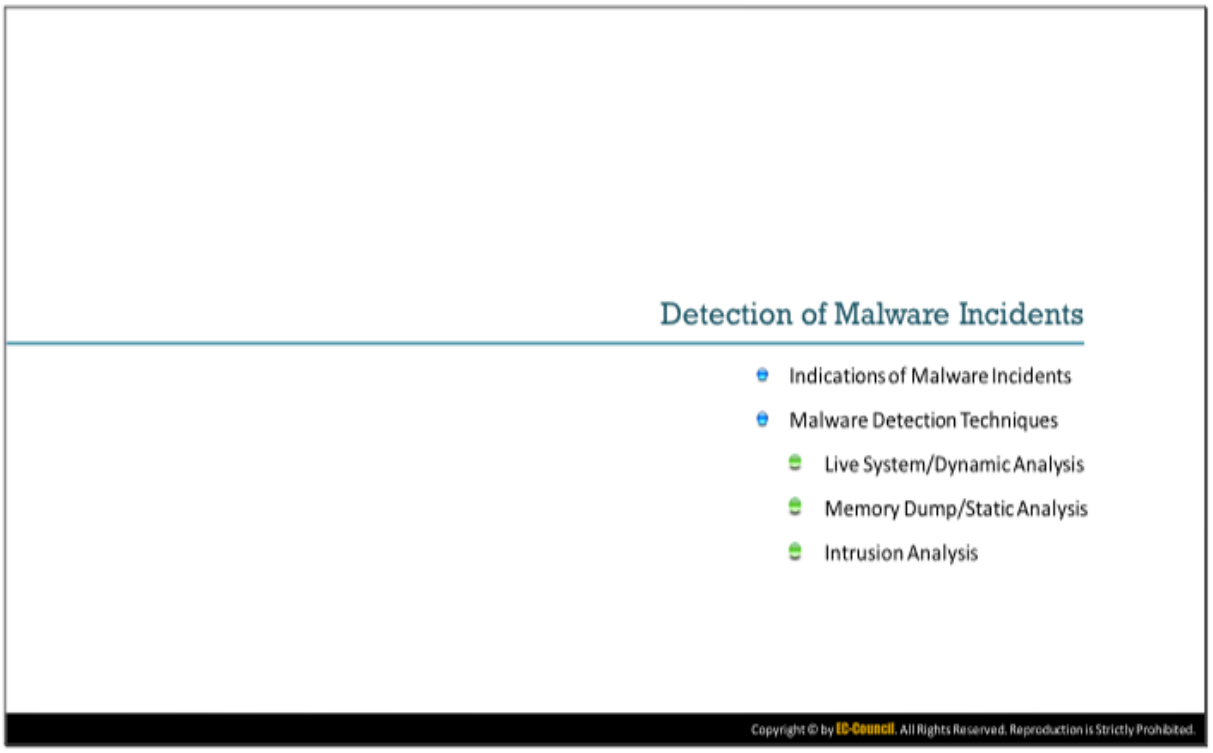
- SnagIt (*https://www.techsmith.com*)
- Jing (*https://www.techsmith.com*)
- Camtasia (*https://www.techsmith.com*)
- Ezvid (*https://www.ezvid.com*)

Network and Internet Simulation Tools

- ns-3 (*https://www.nsnam.org*)
- Riverbed Modeler (*https://www.riverbed.com*)
- QualNet® (*https://web.scalable-networks.com*)

OS Backup and Imaging Tools

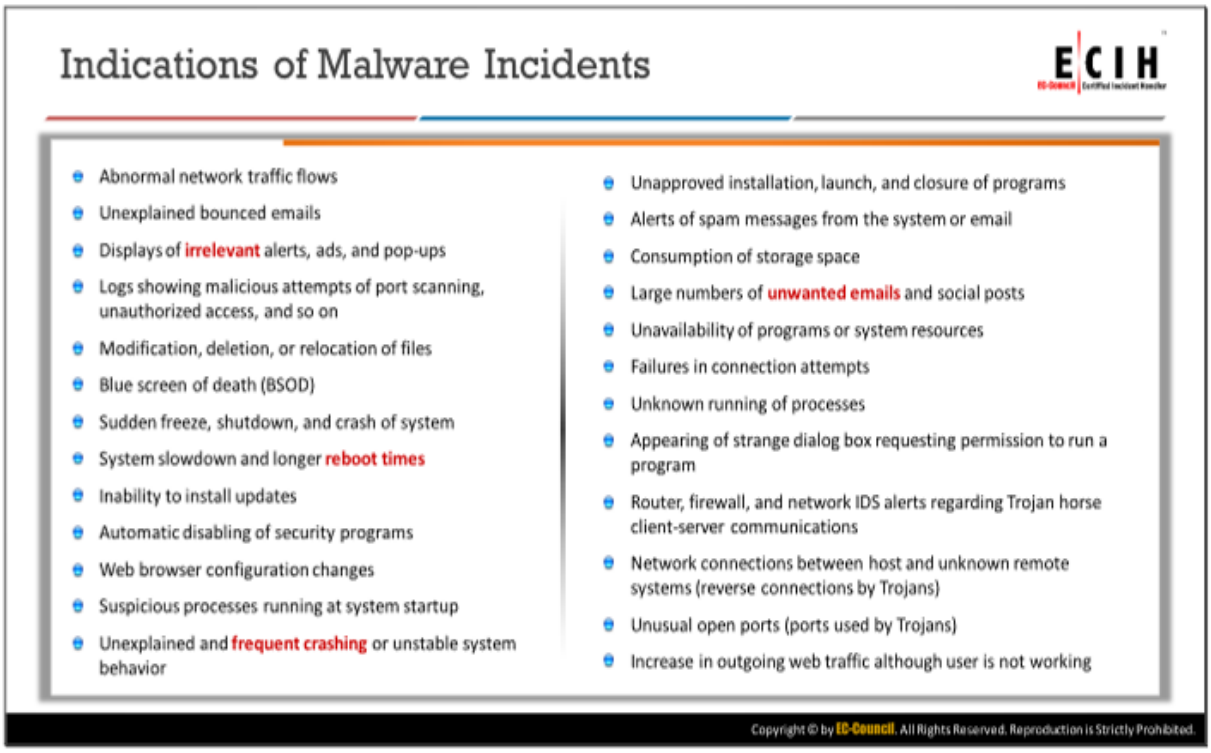
- Genie Backup Manager Pro (*https://www.genie9.com*)
- Macrium Reflect Server (*https://www.macrium.com*)
- R-Drive Image (*https://www.drive-image.com*)
- O&O DiskImage 10 (*https://www.oo-software.com*)



Detection of Malware Incidents

Malware is a program designed to perform malicious acts (The term itself is a contraction of "malicious software."). Malwares such as viruses, Trojans, worms, spywares, and rootkits allow an attacker to breach security defenses and subsequently launch attacks on target systems. Thus, detecting existing infections or malwares quickly is necessary to handle the situation and avoid data corruption.

This section discusses indications of malware security incidents and then discusses in detail various malware detection techniques including live-system/dynamic, memory-dump/static, and intrusion analyses.



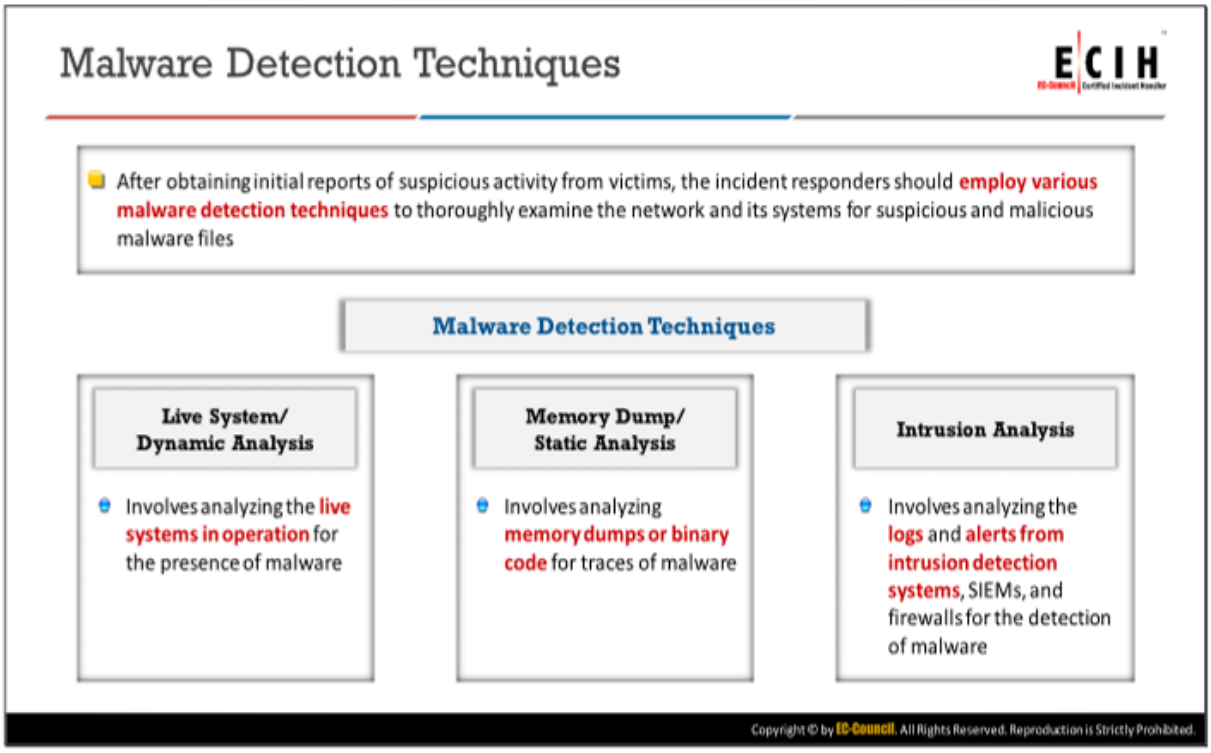
Indications of Malware Incidents

Malwares attack rapidly as they can spread throughout the organization, so incident responders should detect them quickly. Early detection can help minimize the number of infected systems, which in turn reduces the amount of recovery effort required. To detect a malware incident, users, tech support, administrators, and incident responders should be able to identify the following indicators of malware attacks:

- Abnormal network traffic flows
- Unexplained bounced emails
- Displays of irrelevant alerts, ads, and pop-ups
- Logs showing malicious attempts of port scanning, unauthorized access, etc.
- Modification, deletion, or relocation of files
- Blue screen of death (BSOD)
- Sudden freeze, shutdown, and crash of systems
- System slowdown and longer reboot times
- Inability to install updates
- Automatically disabled security programs
- Changes in web browser configurations
- Suspicious processes running at system startup
- Unexplained and frequent crashing or system instability

- Unapproved program installation, launch, and closure
- Alerts of spam messages from the system or email
- Consumption of storage space
- Numerous unwanted emails and social posts
- Unavailable programs or system resources
- Failure in connection attempts
- Unknown running processes
- Strange dialog boxes appearing and requesting permission to run any program
- Router, firewall, and network IDS alerts regarding Trojan horse/client-server communications
- Host and unknown remote systems network connections (i.e., reverse connections by Trojans)
- Unusual open ports (used by Trojans)
- Increased outgoing web traffic even though users are not working
- Unknown registry entries
- Unusual off-hour usages
- Creation of unknown user (guest or administrator) accounts
- Unknown file share access
- Unknown use of protocols
- Unexpected service disruptions
- Presence of unauthorized sessions
- Detection of multiple failed logins
- Detection of any rough hardware
- Presence of unknown malicious software
- The digital versatile disc–read-only memory (DVD–ROM) drawer opens and closes automatically
- The computer screen blinks, flips upside down, or is inverted so that everything is displayed backward
- The default background or wallpaper settings change automatically. This can be performed using pictures either on the user's computer or in the attacker's program.
- Printers automatically start printing documents
- Web pages suddenly open without user input
- OS color settings change automatically

- Screensavers convert to a personal scrolling message
- Sound volume suddenly fluctuates all the way up or down
- Antivirus programs are automatically disabled, and data are corrupted, altered, or deleted from the system
- Computer date and time change
- The mouse cursor moves by itself
- The mouse right-click takes the function of the left-click and vice versa
- The mouse pointer arrow disappears completely
- The mouse pointer and automatic clicks on icons are uncontrollable
- The Windows Start button disappears
- Popups showing bizarre messages suddenly appear
- Clipboard images and text appear to be manipulated
- The keyboard and mouse freeze
- Contacts receive emails that a user did not send from the user's email address
- Strange warnings or question boxes appear. Often these are personal messages directed to the user, asking questions that require the victim to answer by clicking a Yes, No, or OK button
- The system unusually shuts down and reboots
- The taskbar disappears automatically
- The Task Manager is disabled. The attacker, or Trojan, may disable the Task Manager function so that the victim cannot view the task list or be able to end the task on a given program or process.
- Processes take more resources and time, resulting in reduced performance
- The computer beeps with no display
- Drive labels change and the OS cannot be loaded
- Constant antivirus alerts
- Computer freezes frequently or encounters errors such as BSOD
- Files and folders are missing
- Suspicious hard drive activity
- Browser windows "freeze"
- Lack of storage space
- Unwanted advertisements and popup windows



Malware Detection Techniques

After obtaining initial reports of suspicious activity from victims, incident responders should employ various malware detection techniques to thoroughly examine the network and its systems for suspicious and malicious malware files. They must also check whether malware has propagated to other connected devices by finding the shared content and searching for similar suspicious activities. After identifying the compromised system, responders must determine malware type and functionality, impacted system areas, and malware signature for containment and eradication. Malware programs exhibit specific properties that can help responders identify or distinguish them from usual software programs. As a malware incident responder, you must use various techniques and tools to identify malware. Fast malware identification or detection can help responders contain and eradicate malware quickly. Following are malware detection techniques that an incident responder can perform to detect potentially malicious malware:

- **Live-System/Dynamic Analysis**

  It involves analyzing operational live systems for malware.

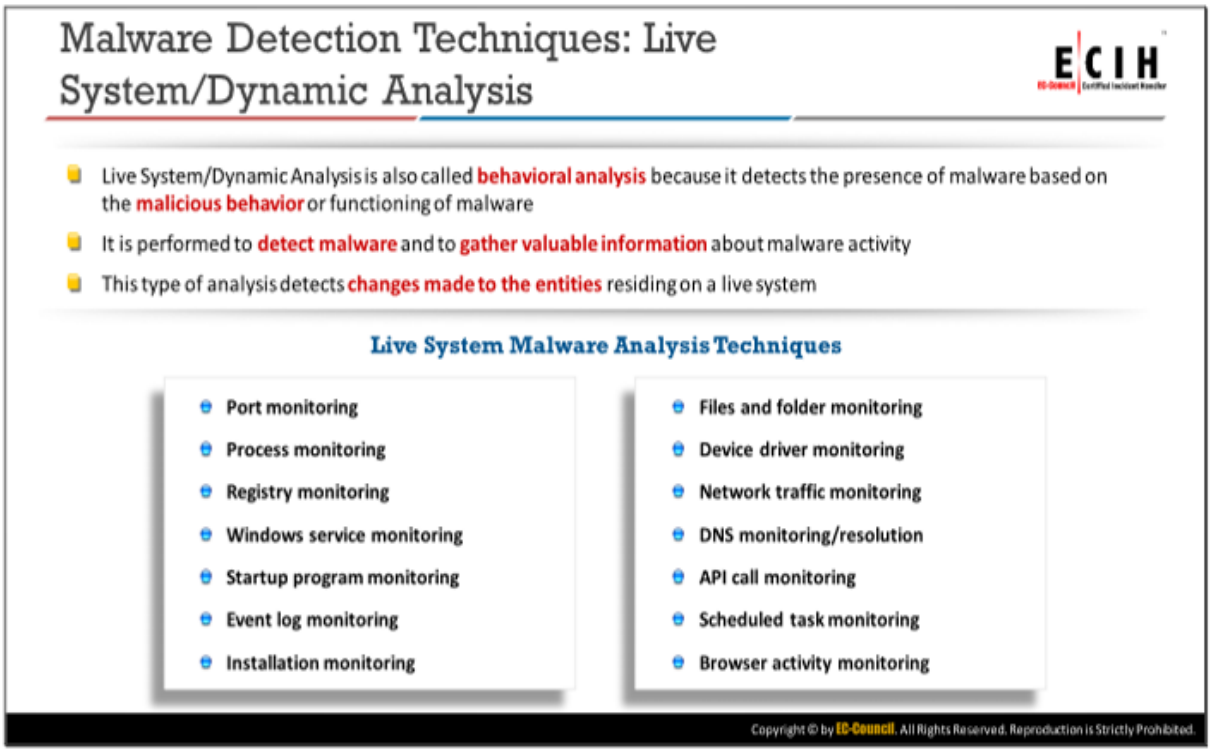
- **Memory-Dump/Static Analysis**

  It involves analyzing memory dumps or binary codes for traces of malware.

- **Intrusion Analysis**

  It involves analyzing logs and alerts of IDSs, security information and event managers (SIEMs), and firewalls to detect malware.

Although these malware detection techniques are intended to understand how malware works, they differ in tools used and analysis time and required skills. It is recommended that all analyses be performed to detect and better understand malware functionality.

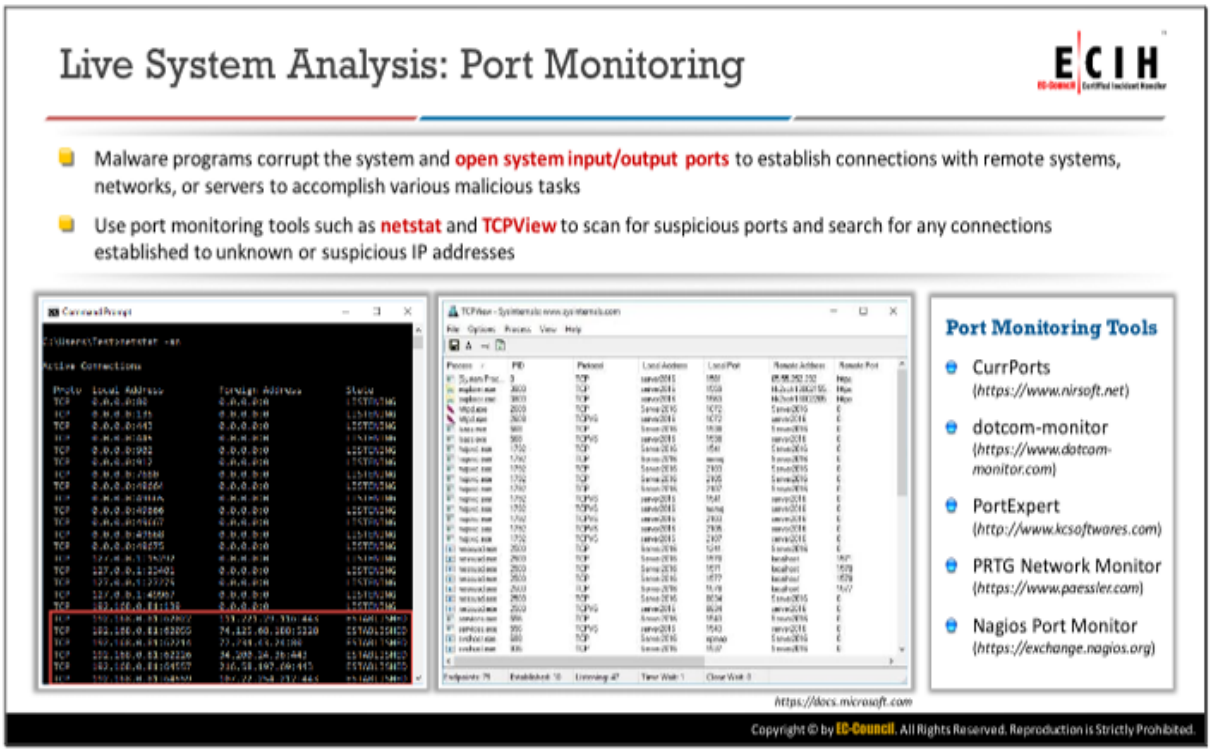


Malware Detection Techniques: Live-System/Dynamic Analysis

Live-system or dynamic analysis is also called "behavioral analysis" because it detects malware based on its malicious behavior or functioning. This type of analysis detects changes made to live-system entities. It is performed to detect malware and gather valuable information about malware activity including files and folders created, ports and uniform resource locators (URLs) accessed, called functions and libraries, applications and tools accessed, information transferred, settings modified, processes and services started, etc. Live-system analysis mainly involves monitoring ports, processes, and registries for any abnormal or malicious activities.

If a user has reported suspicious activity, incident responders must perform the following live-system malware analysis techniques on the compromised system to find traces of malware:

- Port monitoring
- Process monitoring
- Registry monitoring
- Windows service monitoring
- Startup program monitoring
- Event log monitoring
- Installation monitoring
- File and folder monitoring
- Device driver monitoring
- Network traffic monitoring
- DNS monitoring/resolution
- Application programming interface (API) call monitoring
- Scheduled-task monitoring
- Browser activity monitoring



Live-System Analysis: Port Monitoring

Malware programs corrupt the system and open system input/output ports to establish connections with remote systems, networks, or servers and accomplish various malicious tasks. These open ports can also form backdoors for other types of harmful malware and programs. Open ports act as communication channels for malware. They open unused ports on the victim's machine to connect back to the malware handlers. Scanning for suspicious ports will help identify such malware.

As an incident handler, you can also determine whether malware is trying to access a particular port during live-system analysis by installing port monitoring tools such as TCPView and Windows command-line utility tools such as network statistics (netstat). The port monitoring tools will offer details such as protocol, local address, remote address, and connection state. Additional features may include process name and ID and remote connection protocol.

- **Netstat**

  It displays active transmission control protocol (TCP) connections, ports through which the computer is listening, Ethernet statistics, the internet protocol (IP) routing table, IPv4 statistics [for IP, internet control message protocol (ICMP), TCP, and user datagram protocol (UDP)], and IPv6 statistics (for IPv6, ICMPv6, TCP over IPv6, and UDP over IPv6 protocols). Used without parameters, netstat displays active TCP connections.

  Syntax

  ```
  netstat [-a] [-e] [-n] [-o] [-p Protocol] [-r] [-s] [Interval]
  ```

Parameters

- **-a:** Displays all active TCP connections and TCP and UDP ports through which the computer is listening.
- **-e:** Displays Ethernet statistics such as number of bytes and packets sent and received. This parameter can be combined with -s.
- **-n:** Displays active TCP connections; however, addresses and port numbers are expressed numerically, and no attempt is made to determine names.
- **-o:** Displays active TCP connections and includes the process ID (PID) for each connection. You can find the application based on the PID on the Processes tab in Windows Task Manager. This parameter can be combined with -a, -n, and -p.
- **-p Protocol:** Shows protocol connections specified by Protocol. In this case, Protocol can be TCP, UDP, TCPv6, or UDPv6. If this parameter is used with -s to display statistics by protocol, Protocol can be TCP, UDP, ICMP, IP, TCPv6, UDPv6, ICMPv6, or IPv6.
- **-s:** Displays statistics by protocol. By default, statistics are shown in the figure for TCP, UDP, ICMP, and IP. If the IPv6 protocol for Windows XP is installed, statistics are shown for TCP over IPv6, UDP over IPv6, ICMPv6, and IPv6 protocols. Parameter -p can be used to specify a set of protocols.
- **-r:** Displays contents of the IP routing table, which is equivalent to the route print command.

In image shown on the above slide, the command **netstat -an** displays all the active TCP connections and the TCP and UDP ports through which the computer is listening and corresponding addresses and port numbers.

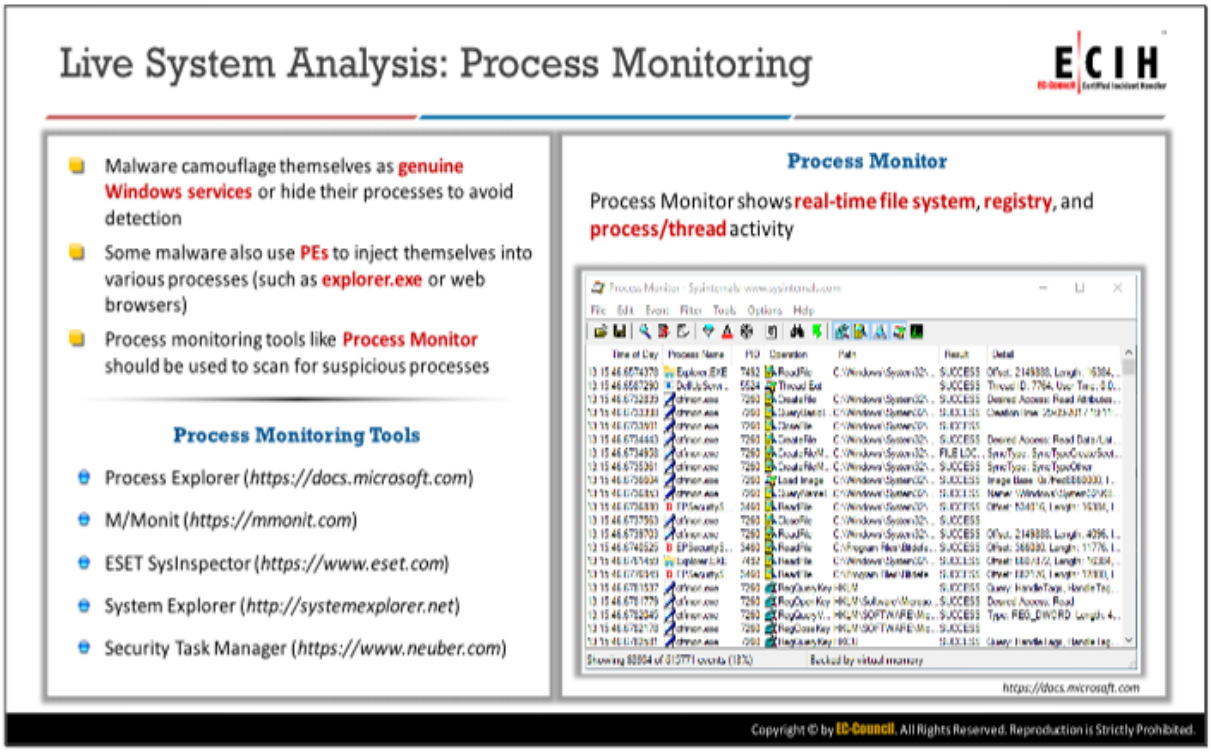
- **TCPView**

  Source: *https://docs.microsoft.com*

  TCPView is a Windows program that shows detailed listings of all TCP and UDP endpoints on the system, including local and remote addresses and TCP connection states. It provides a subset of the netstat program that ships with Windows. The TCPView download includes TCPVCon, a command-line version with the same functionality. When TCPView runs, it enumerates all active TCP and UDP endpoints, resolving all IP addresses to their domain-name versions.

Some additional port monitoring tools include:

- CurrPorts (*https://www.nirsoft.net*)
- dotcom-monitor (*https://www.dotcom-monitor.com*)
- PortExpert (*http://www.kcsoftwares.com*)
- PRTG Network Monitor (*https://www.paessler.com*)
- Nagios Port Monitor (*https://exchange.nagios.org*)



Live-System Analysis: Process Monitoring

Malwares enter a system through pictures, music files, videos, etc. downloaded from the internet, camouflage themselves as genuine Windows services, and hide their processes to avoid detection. Some malwares use PEs to inject themselves into various processes (such as explorer.exe or web browsers). Although malicious processes are visible, they look like legitimate processes and help bypass desktop firewalls. Attackers use specific rootkit methods to hide malware in the system so that antivirus software cannot commonly detect it.

Process monitoring will help understand processes that malware initiates and takes over after execution. Incident handlers should also observe child processes, associated handles, loaded libraries, and functions to define the entire nature of a file or program, gather information about processes running before malware execution, and compare them to processes running after execution. This method will reduce the amount of time required to analyze processes, injected codes, and modified functions and help to easily identify all the processes that malware starts and common techniques for malware process injections. Use process-monitoring tools such as Process Monitor to detect suspicious running processes, malicious parent/child processes, malicious DLLs, and sockets.

- **Process Monitor**

  Source: *https://docs.microsoft.com*

  Process Monitor is a Windows monitoring tool that shows real-time file system, registry, and process/thread activities. It combines features of legacy Sysinternals utilities (Filemon and Regmon) and adds an extensive list of enhancements including rich, nondestructive filtering; comprehensive event properties such session IDs and user names; reliable process information; full thread stacks with integrated symbol support

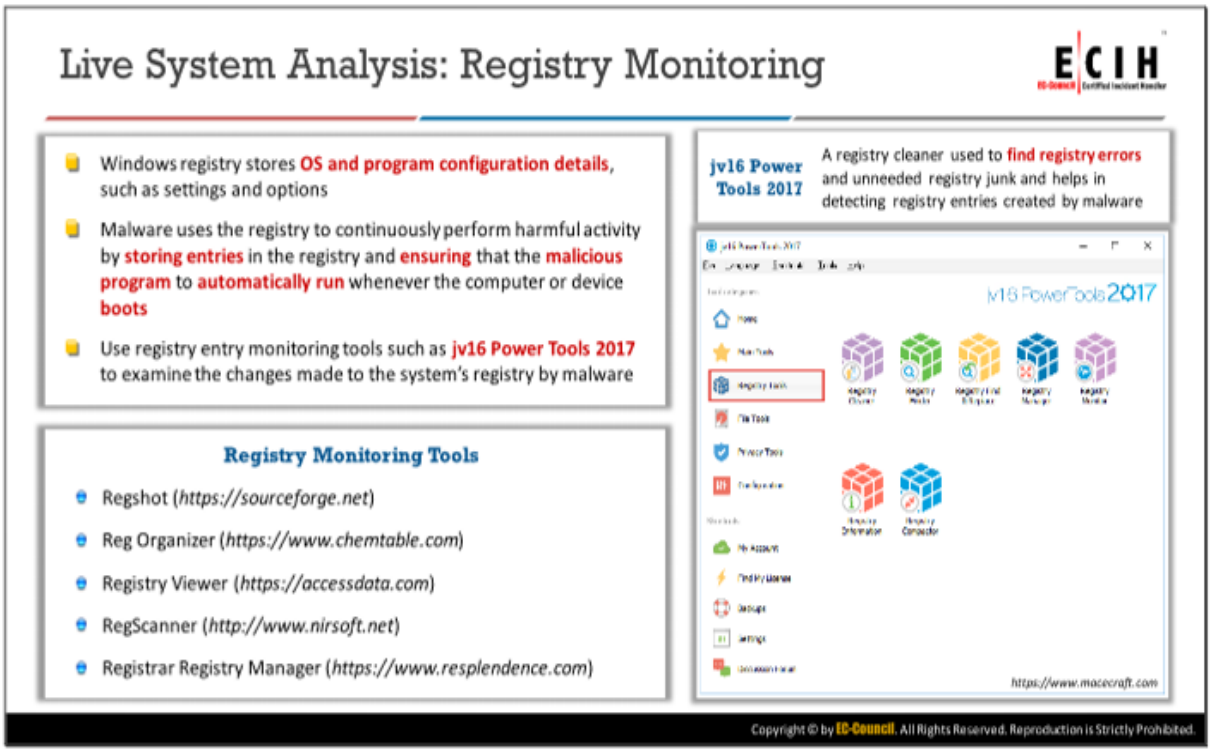
for each operation; simultaneous logging to a file, etc. Unique features of Process Monitor make it a core utility in system troubleshooting and malware hunting.

Features

- More data captured for operation input and output parameters
- Nondestructive filters allow you to set filters without losing data
- Capture of thread stacks for each operation enables operation cause to be identified in many cases
- Reliable capture of process details including image path, command line, user, and session ID
- Configurable and movable columns for any event property
- Filters can be set for any data field including fields not configured as columns
- Advanced logging architecture scales to tens of millions of captured events and gigabytes of log data
- Process-tree tool shows relationships of all processes referenced in a trace
- Native log format preserves all data for loading in a different Process Monitor instance

Some additional process monitoring tools include:

- Process Explorer (*https://docs.microsoft.com*)
- M/Monit (*https://mmonit.com*)
- ESET SysInspector (*https://www.eset.com*)
- System Explorer (*http://systemexplorer.net*)
- Security Task Manager (*https://www.neuber.com*)
- HiJackThis (*https://sourceforge.net*)
- Yet Another (remote) Process Monitor (*http://yaprocmon.sourceforge.net*)
- Process Network Monitor (*https://securityxploded.com*)
- OpManager (*https://www.manageengine.com*)



Live-System Analysis: Registry Monitoring

Windows registry stores OS and program configuration details such as settings and options. If malware is a program, the registry stores its functionality. Malware uses the registry to continuously perform harmful activities by storing entries into the registry and ensuring that the malicious program runs whenever the computer or device automatically boots.

When an attacker installs a type of malware on the victim's machine, the malware generates a registry entry. Consequently, various changes will be noticed, such as the system slows, various advertisements keep popping up, etc.

Windows automatically executes instructions in the following registry sections:

- ***Run***
- ***RunServices***
- ***RunOnce***
- ***RunServicesOnce***
- ***HKEY_CLASSES_ROOT\exefile\shell\open\command "%1" %****

Malware inserts instructions in these registry sections to perform malicious activities. You should have a fair knowledge of the Windows registry and its contents and inner workings to analyze for malware presence. Scanning for suspicious registries will help detect malware. Use registry monitoring tools like jv16 Power Tools 2017 and RegScanner to scan registry values for any suspicious entries that may indicate malware infection.

- **jv16 Power Tools 2017**

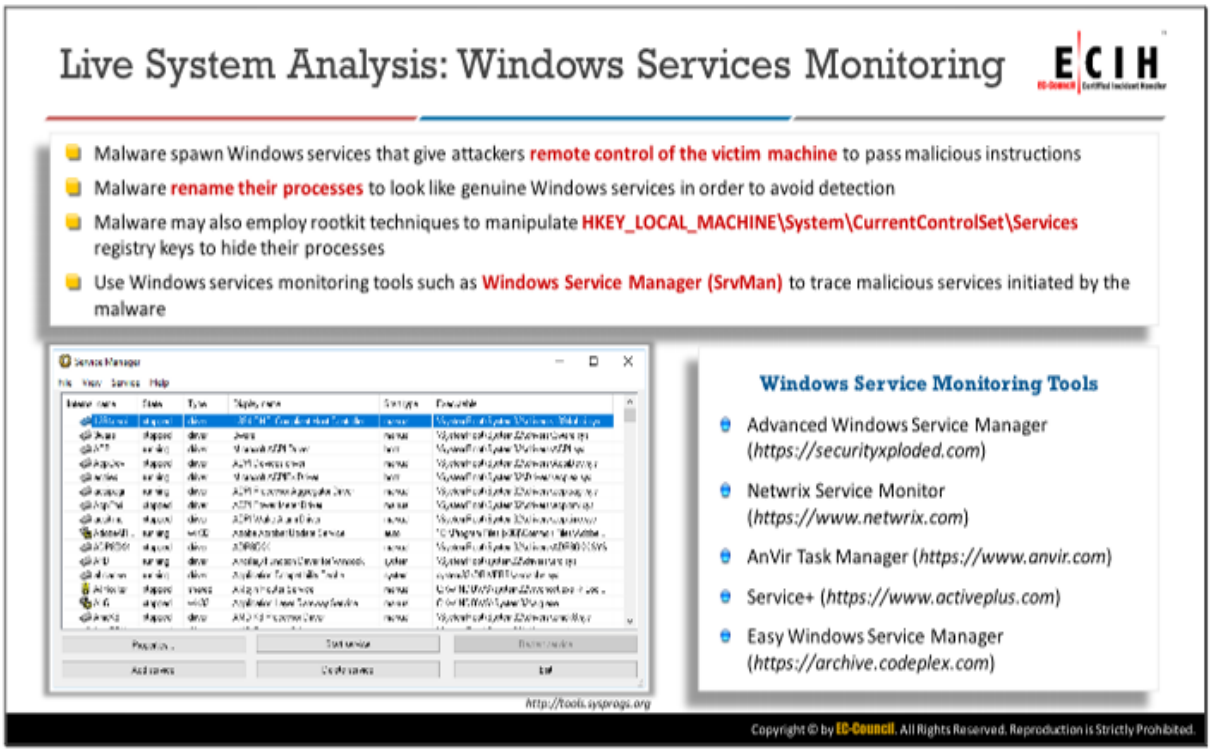
  Source: *https://www.macecraft.com*

  Jv16 Power Tools is PC system utility software that works by cleaning out unneeded files and data, cleaning the Windows registry, automatically fixing system errors, and optimizing the system. It enables the registry and monitor to be scanned.

  It helps to detect registry entries created by malware. The "clean and speedup my computer" feature of Registry Cleaner in jv16 Power Tools 2017 fixes registry and system errors and cleans registry leftovers and unneeded files such as old log and temporary files.

Some additional registry monitoring tools include:

- Regshot *(https://sourceforge.net)*
- Reg Organizer *(https://www.chemtable.com)*
- Registry Viewer *(https://accessdata.com)*
- RegScanner (*http://www.nirsoft.net*)
- Registrar Registry Manager *(https://www.resplendence.com)*
- Active Registry Monitor *(https://www.devicelock.com)*
- MJ Registry Watcher (*https://www.jacobsm.com*)
- Buster Sandbox Analyzer *(https://bsa.isoftware.nl)*



Live-System Analysis: Windows Services Monitoring

Attackers design malware and other malicious code to install and run as a service on a computer device. As most services run in the background to support processes and applications, malicious services are invisible even when they are performing harmful activities on the system and can thus function even without intervention or input. Malware spawns Windows services that enable attackers to control the victim's machine and pass malicious instructions remotely. Malwares rename their processes to look like a genuine Windows service in order to avoid detection. Malware may also employ rootkit techniques to manipulate the following registry keys to hide their processes and services.

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services

These malicious services run as SYSTEM or other privileged accounts, which provide more access than user accounts, making them more dangerous than common malware and executable code. Attackers also try to conceal their actions by naming malicious services similar to names of genuine Windows services to avoid detection.

You can trace malicious services initiated by the suspect file during dynamic analysis using Windows service monitoring tools such as Windows Service Manager (SrvMan), which can detect changes in services and also scan for suspicious Windows services.

- **Windows Service Manager (SrvMan)**

  Source: *http://tools.sysprogs.org*

  SrvMan has both graphical user interface (GUI) and command-line modes. It can also be used to run arbitrary Win32 applications as services (When such service is stopped, the main application window automatically closes.). You can use the SrvMan command line interface to perform the following tasks:

- **Create services**

  ```
  srvman.exe add <file.exe/file.sys> [service name] [display name] [/type:<service type>] [/start:<start mode>] [/interactive:no] [/overwrite:yes]
  ```

- **Delete services**

  ```
  srvman.exe delete <service name>
  ```

- **Start/stop/restart services**

  ```
  srvman.exe start <service name> [/nowait] [/delay:<delay in msec>]
  ```

  ```
  srvman.exe stop <service name> [/nowait] [/delay:<delay in msec>]
  ```

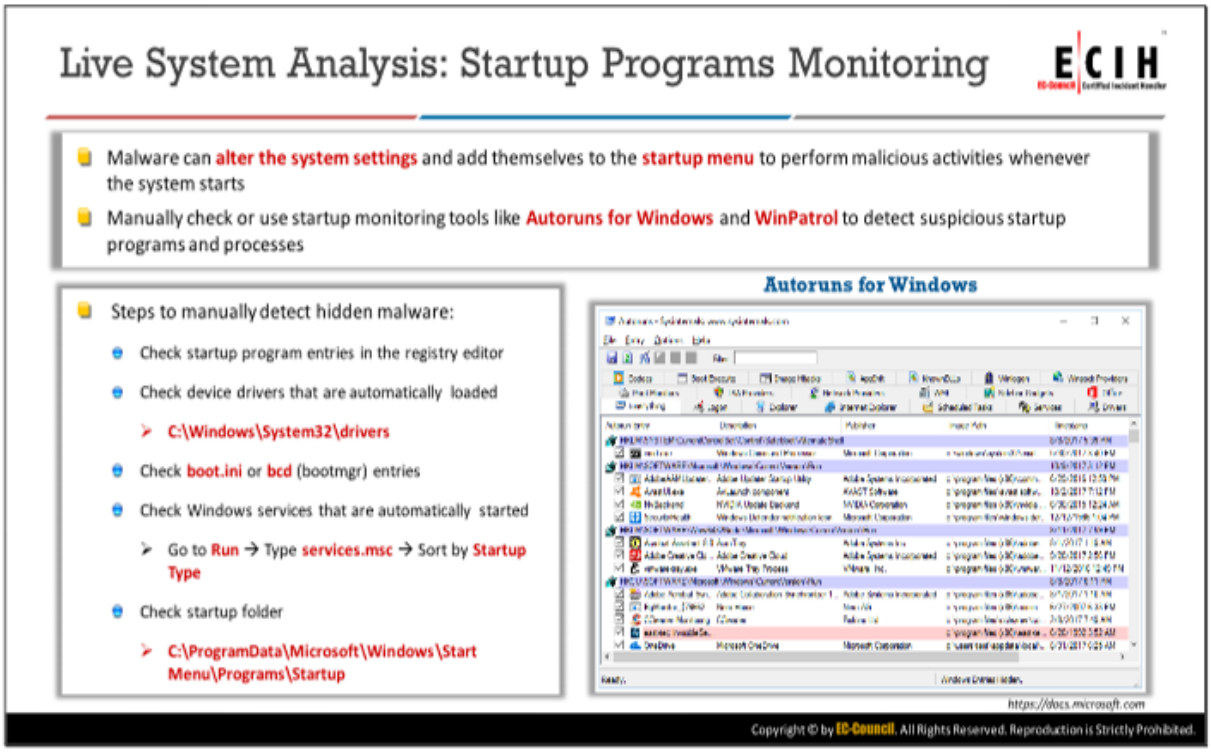
  ```
  srvman.exe restart <service name> [/delay:<delay in msec>]
  ```

- **Install and start a legacy driver with a single call**

  ```
  srvman.exe run <driver.sys> [service name] [/copy:yes] [/overwrite:no] [/stopafter:<msec>]
  ```

Some additional Windows service monitoring tools include:

- Advanced Windows Service Manager (*https://securityxploded.com*)
- Netwrix Service Monitor (*https://www.netwrix.com*)
- AnVir Task Manager (*https://www.anvir.com*)
- Service+ (*https://www.activeplus.com*)
- Easy Windows Service Manager (*https://archive.codeplex.com*)
- Nagios XI (*https://www.nagios.com*)
- Windows Service Monitor (*https://www.manageengine.com*)
- PC Services Optimizer (*https://www.smartpcutilities.com*)
- SMART Utility (*https://www.volitans-software.com*)



Live-System Analysis: Startup Programs Monitoring

Malwares can alter system settings and add themselves to the startup menu to perform malicious activities whenever the system starts. Therefore, scanning for suspicious startup programs manually or using startup program monitoring tools like Autoruns for Windows is essential for detecting malware.

Steps to manually detect hidden malware:

- **Step 1: Check startup program entries in the registry**

  Startup items such as programs, shortcuts, folders, and drivers are set to run automatically at startup when users login to a Windows OS (e.g., Windows 10). Startup items can be added by either installed programs or drivers or manually by the user. Programs that run on Windows 10 startup can be located in registry entries such as Windows Explorer and internet explorer (IE) startup settings.

  - **Windows Startup Settings**

    `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run`

    `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run`

    `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce`

    `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce`

  - **Explorer Startup Settings**

    `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders, Common Startup`

    `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders, Common Startup`

    `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders, Startup`

`HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders, Startup`

- **IE Startup Settings**

  `HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\URLSearchHooks`

  `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\Toolbar`

  `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\Extensions`

  `HKEY_CURRENT_USER\SOFTWARE\Microsoft\Internet Explorer\MenuExt`

- **Step 2: Check automatically loaded device drivers**

  Navigate to `C:\Windows\System32\drivers` to check device drivers.

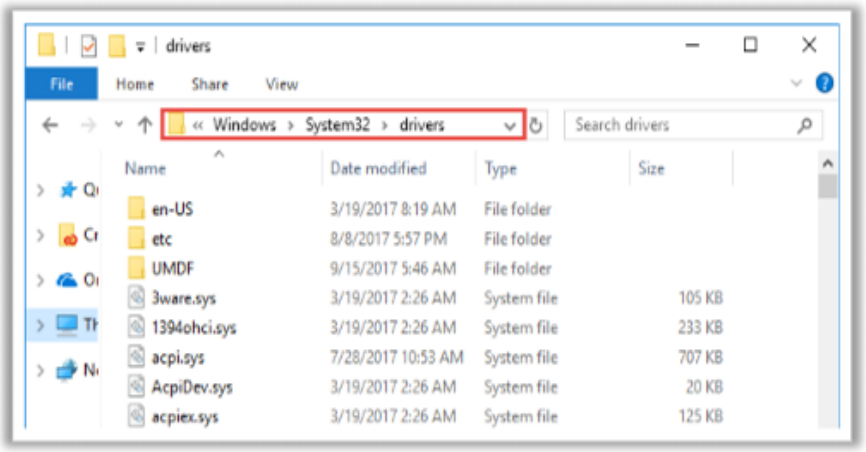


Figure 4.1: Screenshot displaying driver folder.

- **Step 3: Check boot.ini or bcd (bootmgr) entries**

  Check **boot.ini** or **bcd** (bootmgr) entries using the command prompt. Open the **command prompt with administrator privileges**, type the "**bcdedit**" command, and press **Enter** to view all the boot manager entries.

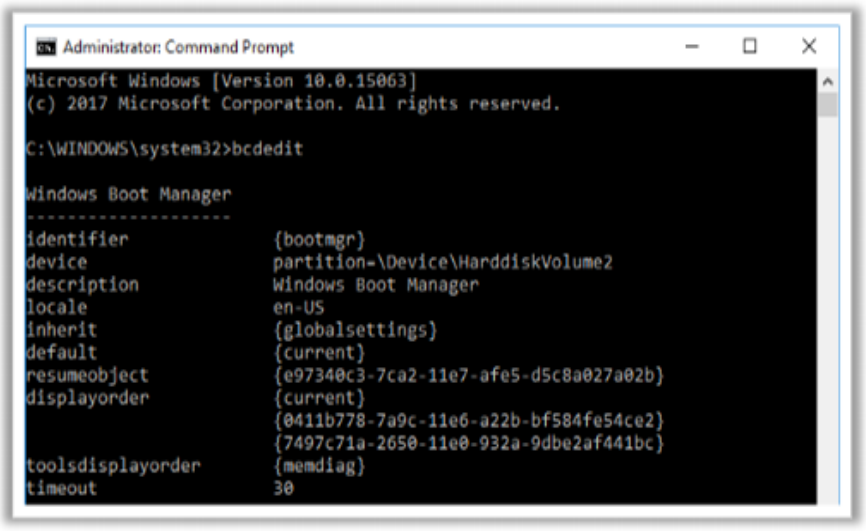


Figure 4.2: Screenshot displaying boot info.

- **Step 4: Check that Windows services automatically started**

  Go to **Run** → Type **services.msc** and press Enter. Sort services by **Startup Type** to check the Windows service list and view services that start automatically when the system boots.

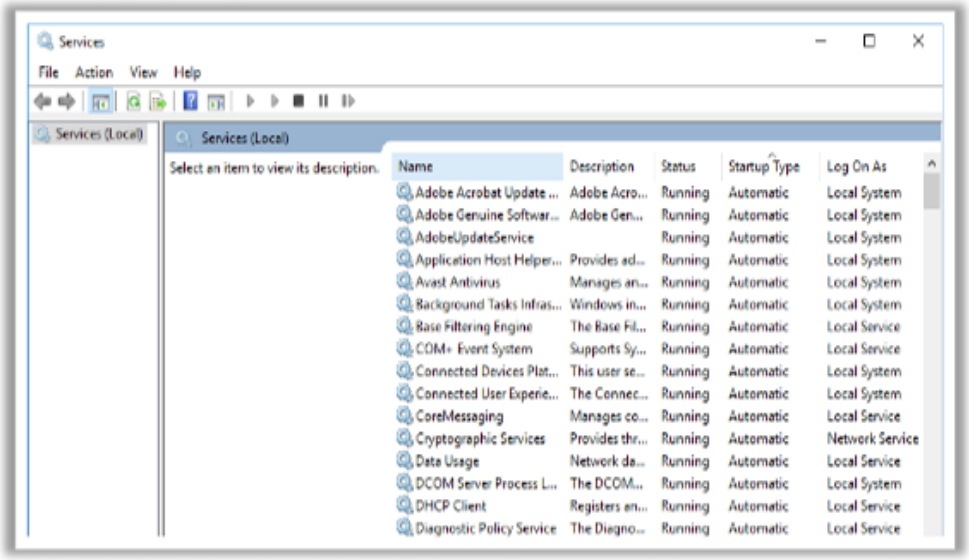


Figure 4.3: Screenshot displaying services.

- **Step 5: Check Startup folders**

  Startup folders store applications or shortcuts for applications that autostart when the system boots. To check **Startup** applications, search the following locations on Windows 10:

  - `C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup`
  - `C:\Users\(User-Name)\AppData\Roaming\Microsoft\Windows\StartMenu\Programs\Startup`

  Another method of accessing startup folders is:

  1. Press the "**Windows" and "R"** buttons simultaneously to open the **Run** box.
  2. Type "**shell: startup"** in the box and click the **OK** button to navigate to the startup folder.

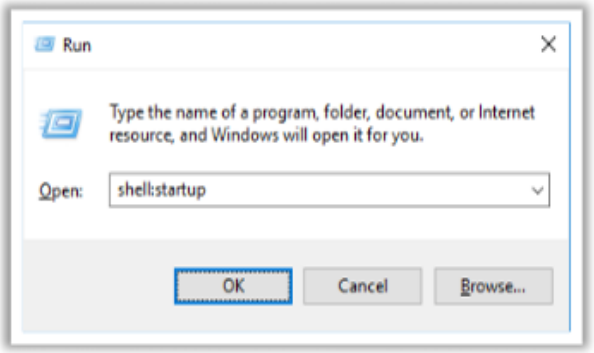


Figure 4.4: Screenshot showing shell: startup command in Run box.

Startup Program Monitoring Tool: Autoruns for Windows

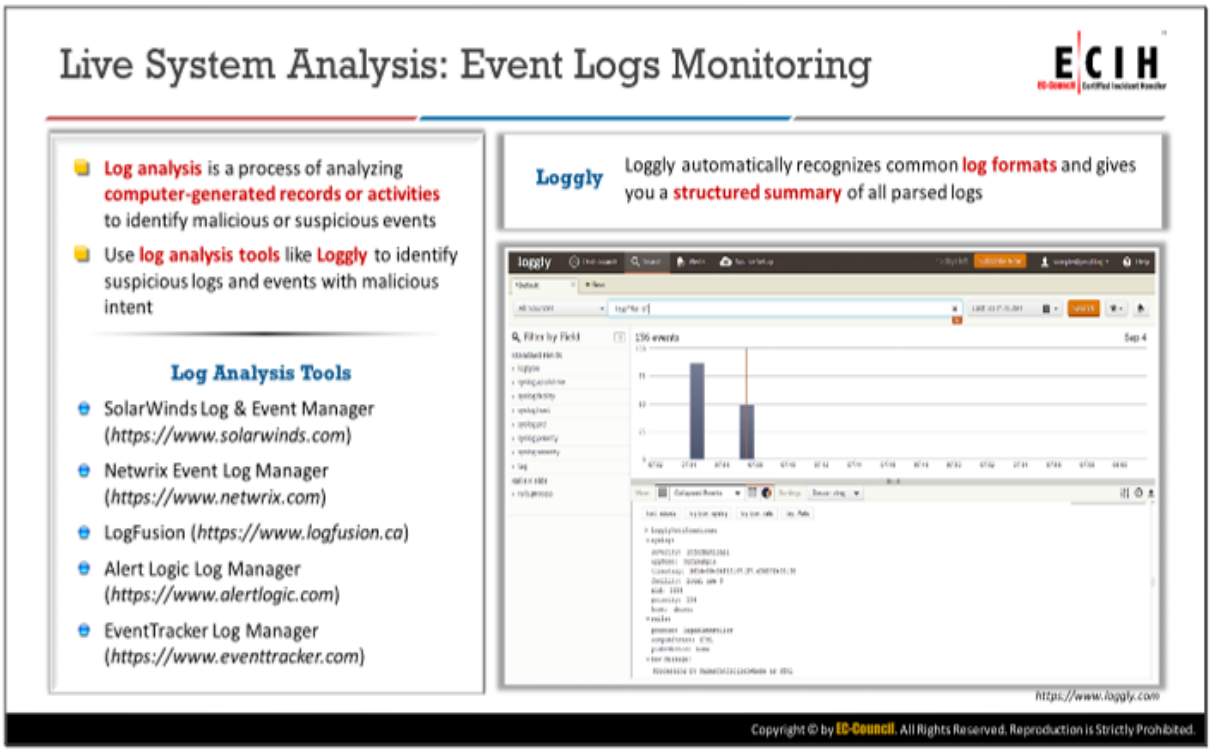
Source: *https://docs.microsoft.com*

This utility can autostart the location of any startup monitor, display what programs are configured to run during system boot or login, and show entries in the order that Windows processes them. As soon as this program includes Run, RunOnce, and other Registry keys in the startup folder, users can configure Autoruns to show other locations including Explorer shell extensions, toolbars, browser helper objects, Winlogon notifications, and autostart services.

Autoruns' "hide signed Microsoft entries" option helps the user zoom in on third-party autostarting images that add to the users' system, and it has support for examining autostarting images configured for other accounts configured on a system.

Some additional startup program monitoring tools include:

- WinPatrol (*https://www.winpatrol.com*)
- Autorun Organizer (*https://www.chemtable.com*)
- Quick Startup (*https://www.glarysoft.com*)
- StartEd Pro (*https://www.outertech.com*)
- Chameleon Startup Manager (*http://www.chameleon-managers.com*)
- BootRacer (*http://www.greatis.com*)
- WinTools.net: Startup Manager (*http://www.wintools.net*)
- EF StartUp Manager (*http://www.efsoftware.com*)
- PC Startup Master (*https://www.smartpcutilities.com*)
- CCleaner (*https://www.piriform.com*)
- Startup Delayer (*https://www.r2.com.au*)



Live-System Analysis: Event Logs Monitoring

Log analysis is a process that provides details of an activity or an event that can extract possible attacks in the form of Trojans or worms on the system. It serves as a primary source of information and helps identify security gaps. This process helps detect zero-day backdoor Trojans or any possible attacks (failed authentication/login attempts) when logs are analyzed for different components. Logs are monitored for components that perform security operations, such as firewall systems, intrusion-detection systems/intrusion-prevention systems (IDSs/IPSs), web servers, and authentication servers. Logs also contain file types, ports, timestamps, and registry entries. In Windows, system, application, and security logs can be analyzed in Event Viewer under "Windows Logs."

Logs are located by the following paths:

- **System logs**

  Start → Windows Administrative Tools → Event Viewer → Windows Logs

- **System Security logs**

  Start → Windows Administrative Tools → Event Viewer → Windows Logs → Security

- **Applications and Services Logs**

  Start → Windows Administrative Tools → Event Viewer → Applications and Services Logs

Command History Monitoring

Certain malwares can use the command prompt to escalate privileges, access restricted locations, find connected systems, and perform other malicious activities in the system.

Command history monitoring will help incident responders determine execution of unauthorized commands. While open, the command prompt will retain command history and will automatically delete it when closed or when the user deletes it manually.

Responders must retrieve command histories/logs from open command prompts using the **doskey/history** command-line tool. This command gives the complete history of commands typed into open command prompts.

Log Analysis Tools

- **Loggly**

  Source: *https://www.loggly.com*

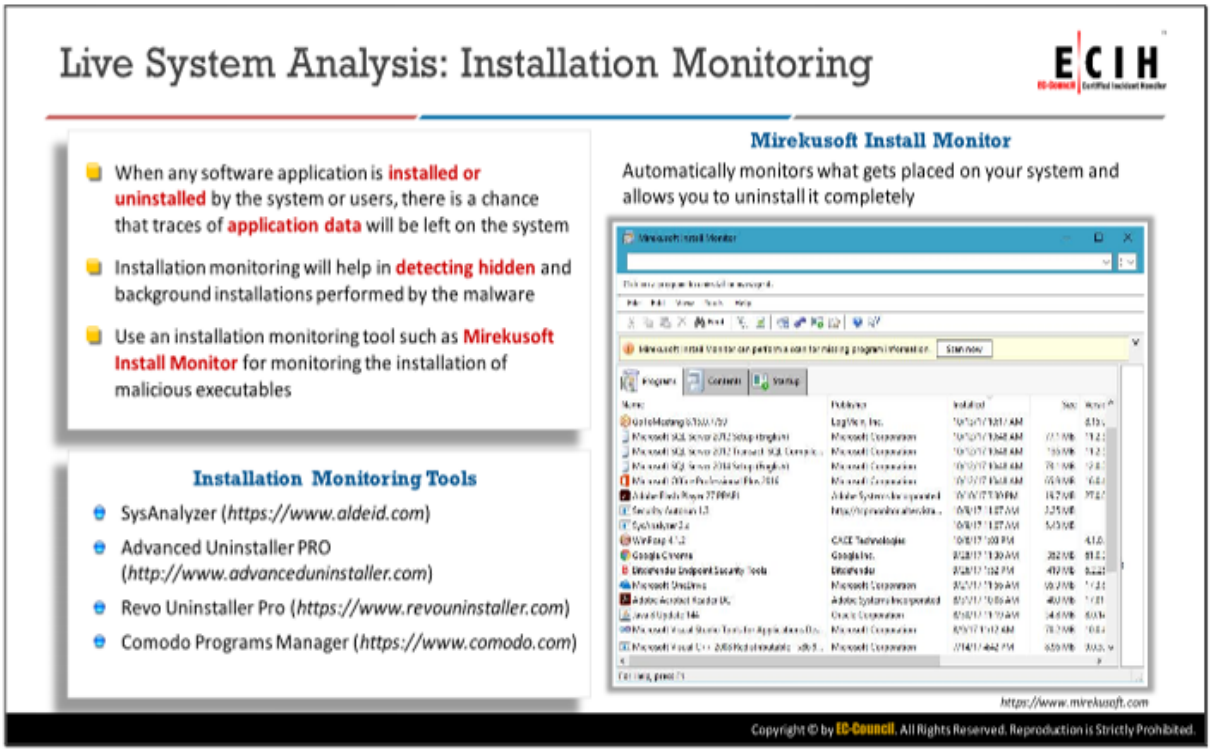
  Loggly automatically recognizes common log formats and gives a structured summary of all parsed logs. It provides real-time monitoring of logs, system behaviors, and unusual activities. It brings logs from the depths of an organization's infrastructure to track activity and analyze trends. It shows how components interact and identifies correlations. Logs can be captured in real-time either on syslog or HTTP.

  Features

  - Tracks service-level agreement (SLA) compliance and identifies anomalies and suspicious events
  - Secures log-data transmission
  - Generates a real-time bird's-eye view of logs
  - Monitors proactively

Some additional log monitoring/analysis tools include:

- SolarWinds Log & Event Manager (*https://www.solarwinds.com*)
- Netwrix Event Log Manager (*https://www.netwrix.com*)
- LogFusion (*https://www.logfusion.ca*)
- Alert Logic Log Manager (*https://www.alertlogic.com*)
- EventTracker Log Manager (*https://www.eventtracker.com*)
- Process Lasso Pro (*https://bitsum.com*)
- Splunk (*https://www.splunk.com*)



Live-System Analysis: Installation Monitoring

When the system or a user installs or uninstalls any software application, there is a chance that it will leave traces of application data on the system. To find such traces, you should know the folders modified or created during installation and files and folders not modified by uninstallation. Installation monitoring will help detect malware hidden and background installations. Tools like Mirekusoft Install Monitor and SysAnalyzer can be used to monitor installation of malicious executables.

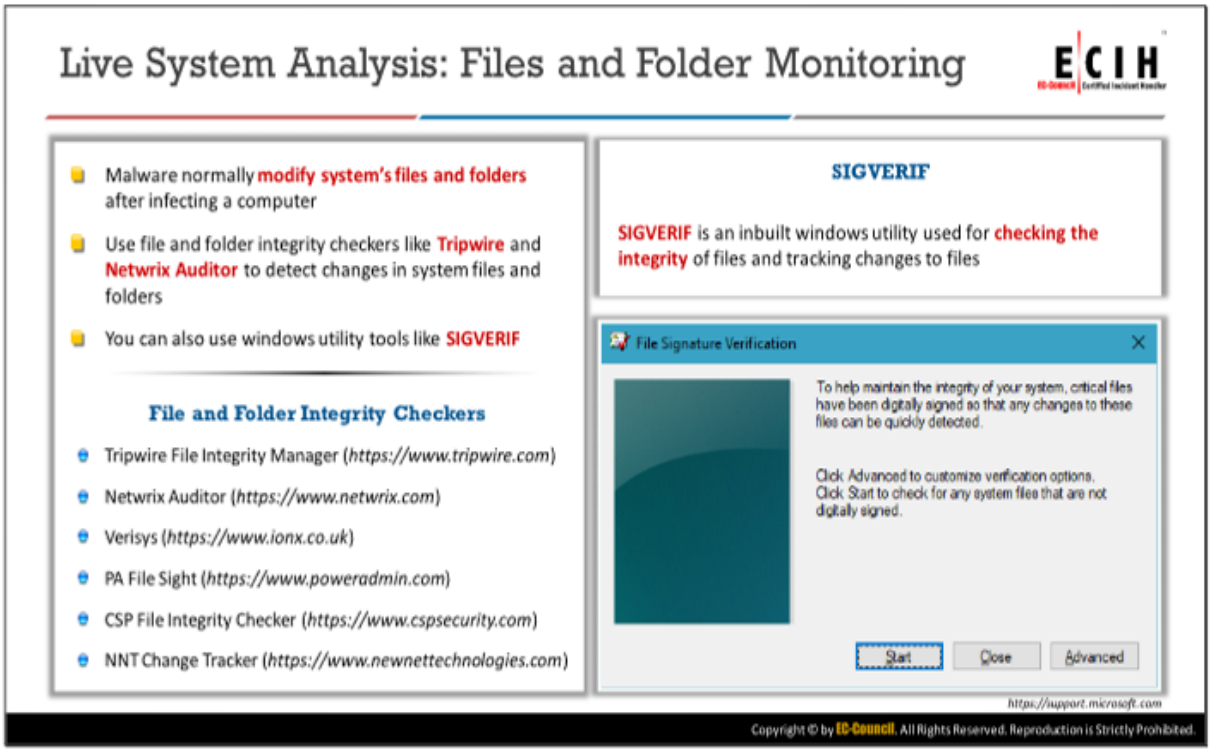
- **Mirekusoft Install Monitor**

  Source: *https://www.mirekusoft.com*

  Mirekusoft Install Monitor automatically monitors what programs are installed on your system and enables them to be completely uninstalled. Install Monitor works by monitoring what resources, such as files and registry entries, are created when a program is installed. It provides detailed information about installed software. You can determine how much disk space, central processing unit (CPU) capacity, and memory your programs are using. It also provides information about how often you use different programs. The program tree is a useful tool that can show you which programs were installed together.

Some additional installation monitoring tools include:

- SysAnalyzer (*https://www.aldeid.com*)
- Advanced Uninstaller PRO (*https://www.advanceduninstaller.com*)
- Revo Uninstaller Pro (*https://www.revouninstaller.com*)
- Comodo Programs Manager (*https://www.comodo.com*)



Live-System Analysis: Files and Folder Monitoring

Malware can modify system files and folders to save information on them. You should be able to find files and folders that malware creates and analyze them to collect any relevant stored information. These files and folders may also contain hidden program code or malicious strings that malware would execute at required time intervals. Incident responders must also check for any files opened by malware or an attacker from a remote location using the command "**openfiles**." It displays a list of currently open files.

Incident responders should also check clipboard contents, prefetch files, etc. to detect malware. Windows creates prefetch files when an application is initially run. This file helps find related files, DLLs, processes, services, and locations of running programs. Responders should analyze prefetch files to obtain information such as whether the attacker ran any programs such as CCleaner to erase activity history. Prefetch files also provide responders with timelines as to when the malware ran its malicious programs. Use tools such as WinPrefetchView to view prefetch files.

Therefore, as an incident responder, you must scan for suspicious files and folders using tools such as SIGVERIF, FCIV, Fastsum, WinMD5, and Tripwire to detect any installed Trojans and system file modifications.

- **SIGVERIF**

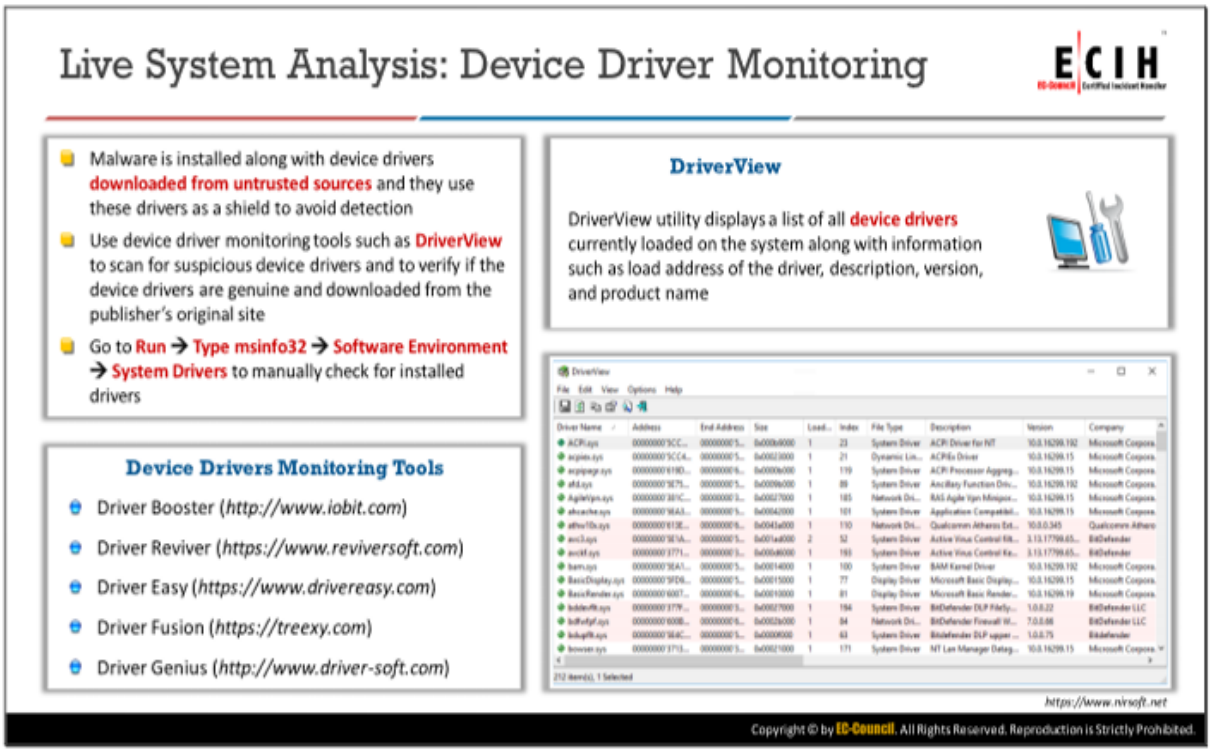
  Source: *https://support.microsoft.com*

  SIGVERIF is a Windows tool, built into Windows 10/8/7, that searches for unsigned system drivers. When an unsigned driver is found, it can be moved to a new folder, the system can be restarted, and the program and its functionality can be tested for errors. Following are steps required for identifying unsigned drivers using SIGVERIF:

- Click **Start → Run**, type **SIGVERIF** and then click **OK**.
- Click the **Advanced** button. Click **Look for other files that are not digitally signed**.
- Navigate to the **Windows\System32\drivers** folder and then click **OK**.
- After Sigverif has finished running its check, it displays a list of all the unsigned drivers installed on the computer. The list of all the signed and unsigned drivers found by Sigverif is in the Sigverif.txt file in the %Windir% folder, typically in the Windows folder.

Some additional file integrity checking tools include:

- Tripwire File Integrity Manager (*https://www.tripwire.com*)
- Netwrix Auditor (*https://www.netwrix.com*)
- Verisys (*https://www.ionx.co.uk*)
- PA File Sight (*https://www.poweradmin.com*)
- CSP File Integrity Checker (*https://www.cspsecurity.com*)
- NNT Change Tracker (*https://www.newnettechnologies.com*)
- AFICK (Another File Integrity Checker) (*http://afick.sourceforge.net*)
- Fsum Frontend (*http://fsumfe.sourceforge.net*)
- OSSEC (*https://www.ossec.net*)
- IgorWare Hasher (*https://www.igorware.com*)



Live-System Analysis: Device Driver Monitoring

The system installs malware along with device drivers when a user downloads infected drivers from untrusted sources; malware uses these drivers as a shield to avoid detection. One can scan for suspicious device drivers using tools such as DriverView and Driver Detective to verify if they are genuine and downloaded from the publisher's original site.

The path to Windows System Drivers is:

Run → Type **msinfo32** → **Software Environment** → **System Drivers**

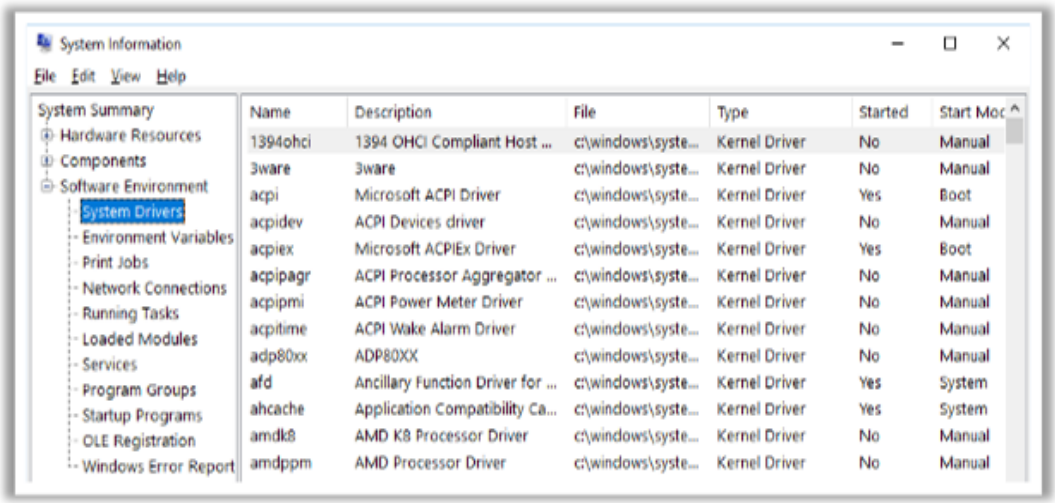


Figure 4.5: Screenshot displaying Windows System Drivers.

- **DriverView**

  Source: *https://www.nirsoft.net*

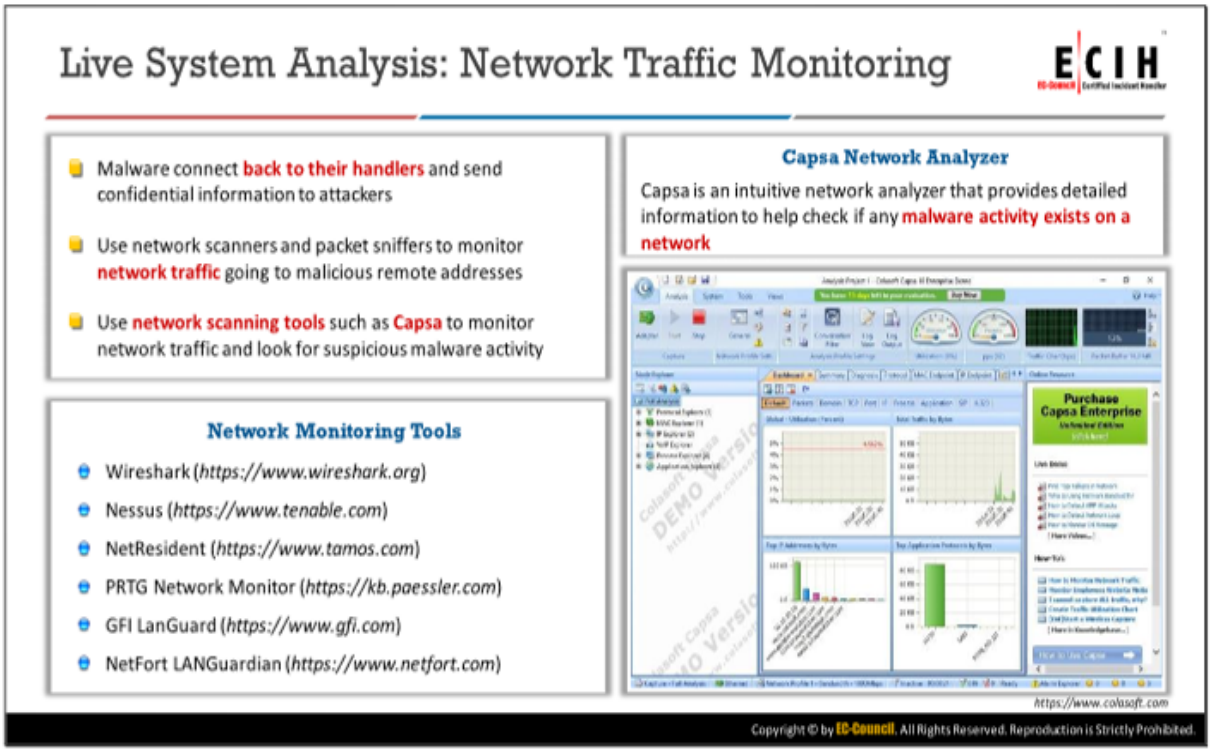
  The DriverView utility displays the list of all device drivers currently loaded on the system. For each driver in the list, additional information is displayed such as load address, description, version, product name, and development company.

  Features

  - Displays the list of all drivers loaded on your system
  - Standalone executable

Some additional device driver monitoring tools include:

- Driver Booster (*https://www.iobit.com*)
- Driver Reviver (*https://www.reviversoft.com*)
- Driver Easy (*https://www.drivereasy.com*)
- Driver Fusion (*https://treexy.com*)
- Driver Genius (*http://www.driver-soft.com*)
- Unknown Device Identifier (*http://www.zhangduo.com*)
- Driver Magician (*http://www.drivermagician.com*)
- DriverHive (*http://www.driverhive.com*)
- InstalledDriversList (*https://www.nirsoft.net*)
- My Drivers (*http://www.zhangduo.com*)
- Driver Agent Plus (*https://scan.driverguide.com*)
- DriverPack (*https://drp.su*)



Live-System Analysis: Network Traffic Monitoring

Network analysis is the process of capturing network traffic and investigating it carefully to determine malware activity. It helps to find types of traffic/network packets or data transmitted across the network.

Malware depends on the network for various activities such as propagation, downloading malicious content, transmitting sensitive files and information, and offering a remote control to attackers. Therefore, you should adopt techniques that can detect malware network send/receive loops, malware artifacts, and usage across networks. Some malware connects back to handlers and sends confidential information to attackers. Network traffic monitoring, statistical analysis, and manual traffic-review all increase probability of incident responders detecting any malware privilege-escalation activities over network computers.

Network monitoring tools such as Capsa Network Analyzer and Wireshark can be used to monitor and capture live network traffic to and from the victim system during execution of the suspect program. This will help understand the malware's network artifacts, signatures, functions, and other elements.

- **Capsa Network Analyzer**

  Source: *https://www.colasoft.com*

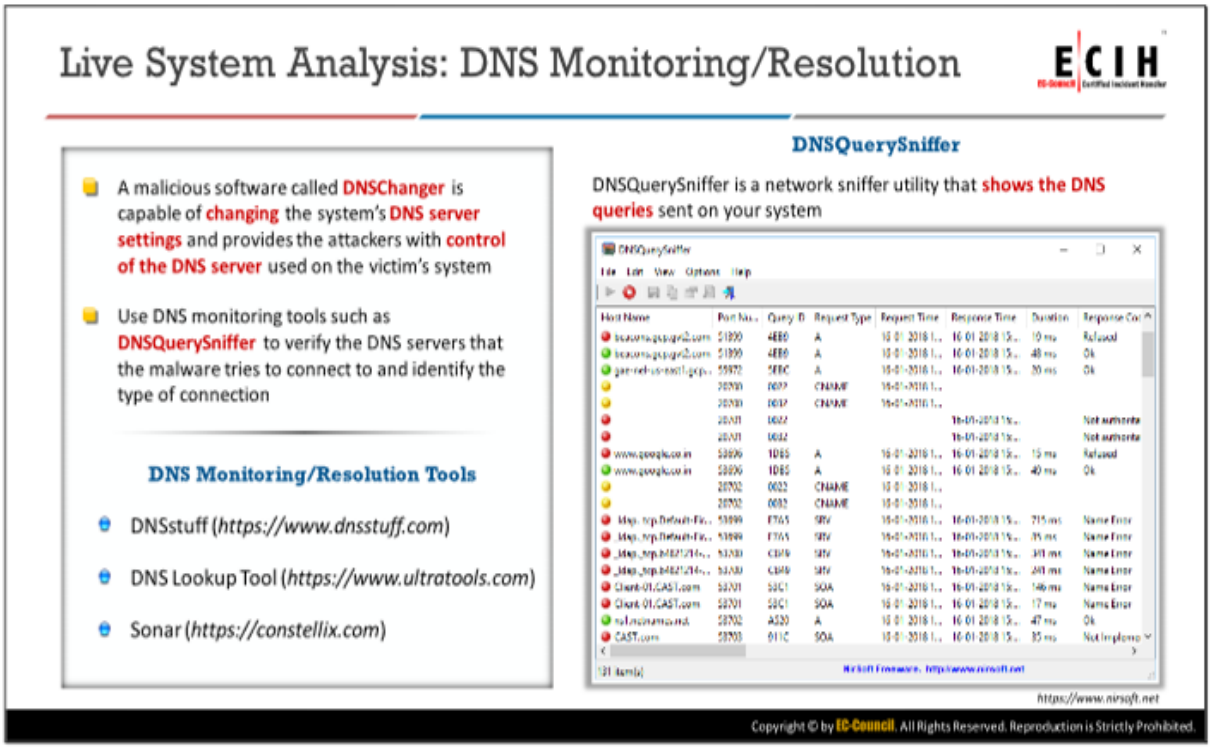
  Capsa is a portable network analyzer application for both local area networks (LANs) and wireless LANs (WLANs), which captures packets in real time, monitors networks 24/7, analyzes advanced protocols, decodes packets in depth, and automatically and expertly diagnoses network problems. Capsa is an intuitive network analyzer, which provides detailed information to help check whether there are any Trojan activities on a network. It helps incident responders pinpoint and resolve application problems.

Features

- Real-time packet capture and ability to save data transmitted over local networks, including wired networks and wireless ones like 802.11a/b/g/n.
- Identifies and analyzes network protocols and network applications according to protocol analysis.
- Identifies "Top Talkers" by monitoring network bandwidth and usage by capturing and summarizing data packets transmitted over networks and decoding packet information.
- Monitors and saves internet email and instant-messaging traffic, thereby helping identify security and confidential-data-handling violations.
- Diagnoses and pinpoints network problems by detecting and locating suspicious hosts.
- Maps traffic, IP address, and media access control (MAC) of each host on the network, allowing identification of each host and through traffic.

Some additional network activity monitoring tools include:

- Wireshark (*https://www.wireshark.org*)
- Nessus (*https://www.tenable.com*)
- NetResident (*https://www.tamos.com*)
- PRTG Network Monitor (*https://kb.paessler.com*)
- GFI LanGuard (*https://www.gfi.com*)
- NetFort LANGuardian (*https://www.netfort.com*)
- CapMon (*https://www.capmon.dk*)
- Nagios XI (*https://www.nagios.com*)
- Total Network Monitor (*https://www.softinventive.com*)



Live-System Analysis: DNS Monitoring/Resolution

Malicious software called DNSChanger can change system DNS server settings and give attackers control of the DNS server on the victim's system, thereby enabling attackers to control internet sites to which the user tries to connect and force the victim to connect to a fraudulent website or interfere with their online web browsing.

Therefore, while performing dynamic analysis, you should identify whether malware can change any DNS server settings. You can use tools such as DNSQuerySniffer and DNSstuff to verify DNS servers to which malware tries to connect and identify connection type.

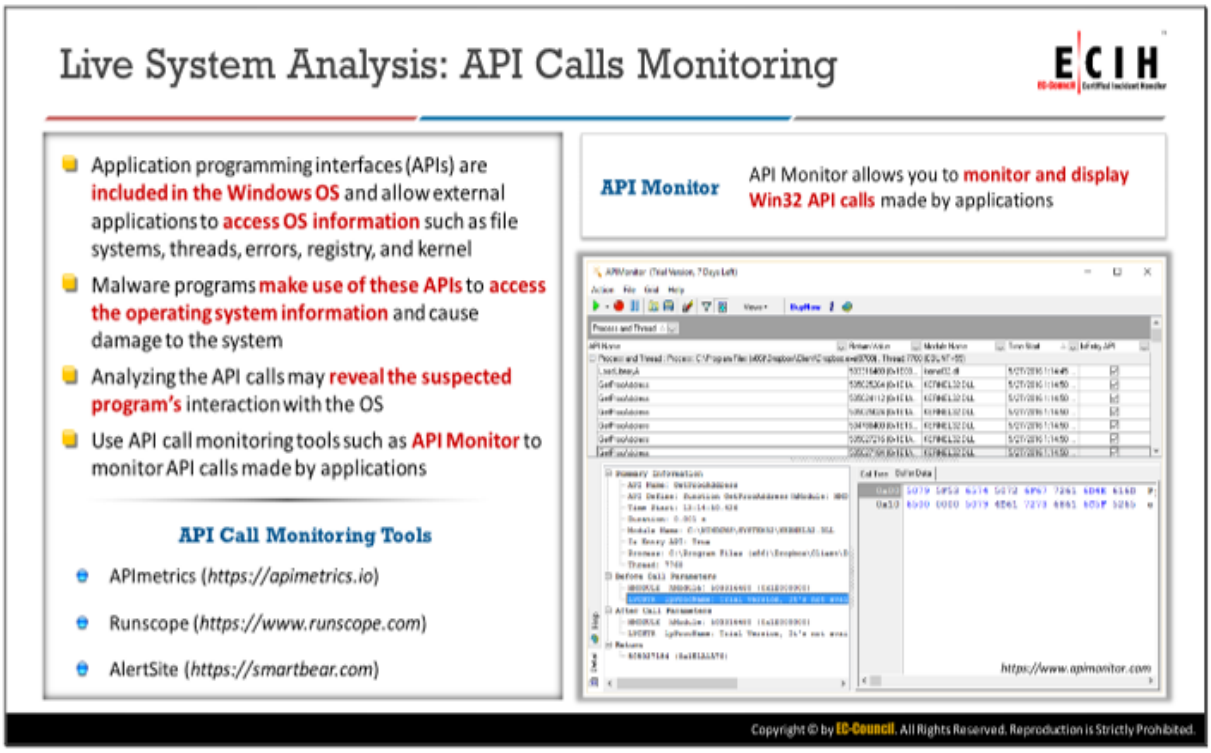
- **DNSQuerySniffer**

  Source: *https://www.nirsoft.net*

  DNSQuerySniffer is a network sniffer utility that shows DNS queries sent on your system. For every DNS query, the following information is displayed: Host Name, Port Number, Query ID, Request Type (A, AAAA, NS, MX, etc.), Request Time, Response Time, Duration, Response Code, Number of records, and content of returned DNS records. You can easily export DNS query information to comma-separated value (CSV)/tab-delimited/extensible markup language (XML)/hypertext markup language (HTML) files or copy DNS queries to the clipboard and then paste them into Excel or other spreadsheet applications.

Some additional DNS monitoring/resolution tools include:

- DNSstuff (*https://www.dnsstuff.com*)
- DNS Lookup Tool (*https://www.ultratools.com*)
- Sonar (*https://constellix.com*)



Live-System Analysis: API Calls Monitoring

APIs are parts of the Windows OS that allow external applications to access OS information such as file systems, threads, errors, registries, kernels, buttons, mouse pointers, network services, webs, and the internet. Malware programs also use APIs to access OS information and damage systems.

You must gather malware-related APIs and analyze them to reveal their OS interactions and system activities. Use API call monitoring tools such as API Monitor to monitor application API calls.

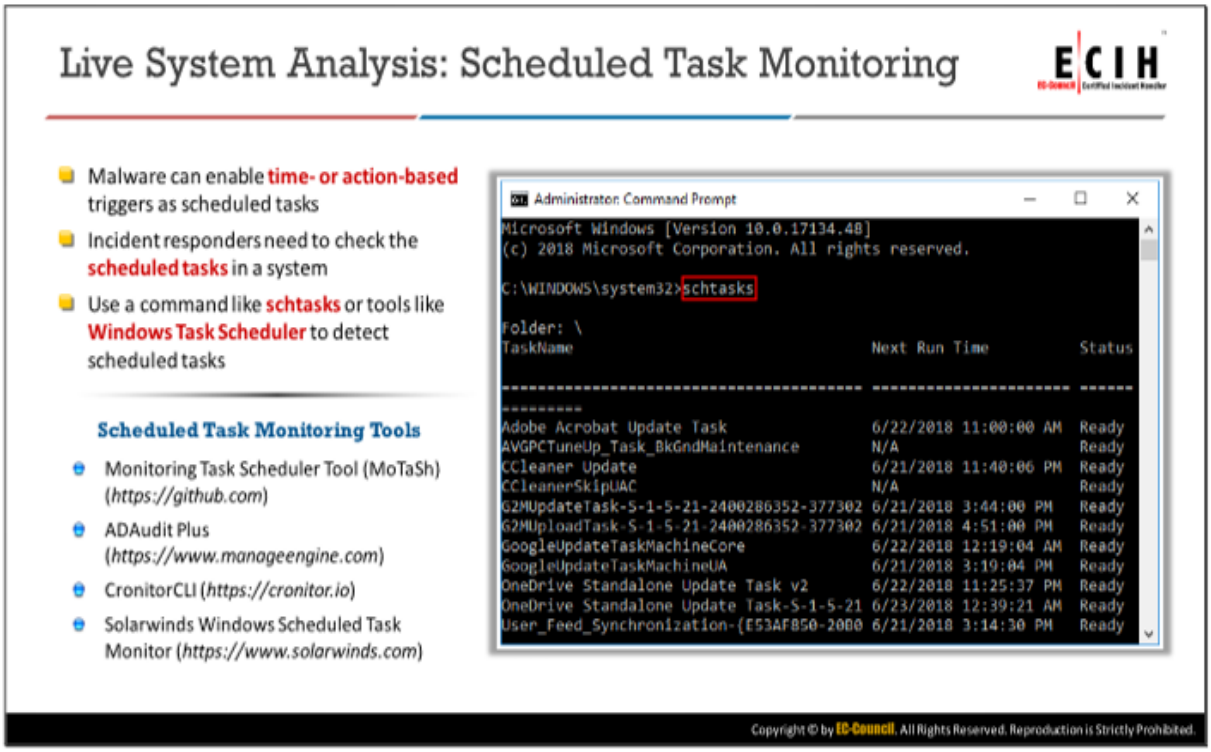
- **API Monitor**

  Source: *https://www.apimonitor.com*

  API Monitor is software that allows you to monitor and display Win32 API calls made by various applications. It can trace any exported APIs and display a wide range of information including function name, call sequence, input and output parameters, function return values, and more. This developer tool is useful for determining how Win32 applications work and learning their tricks.

Some additional API monitoring tools include:

- APImetrics (*https://apimetrics.io*)
- Runscope (*https://www.runscope.com*)
- AlertSite (*https://smartbear.com*)



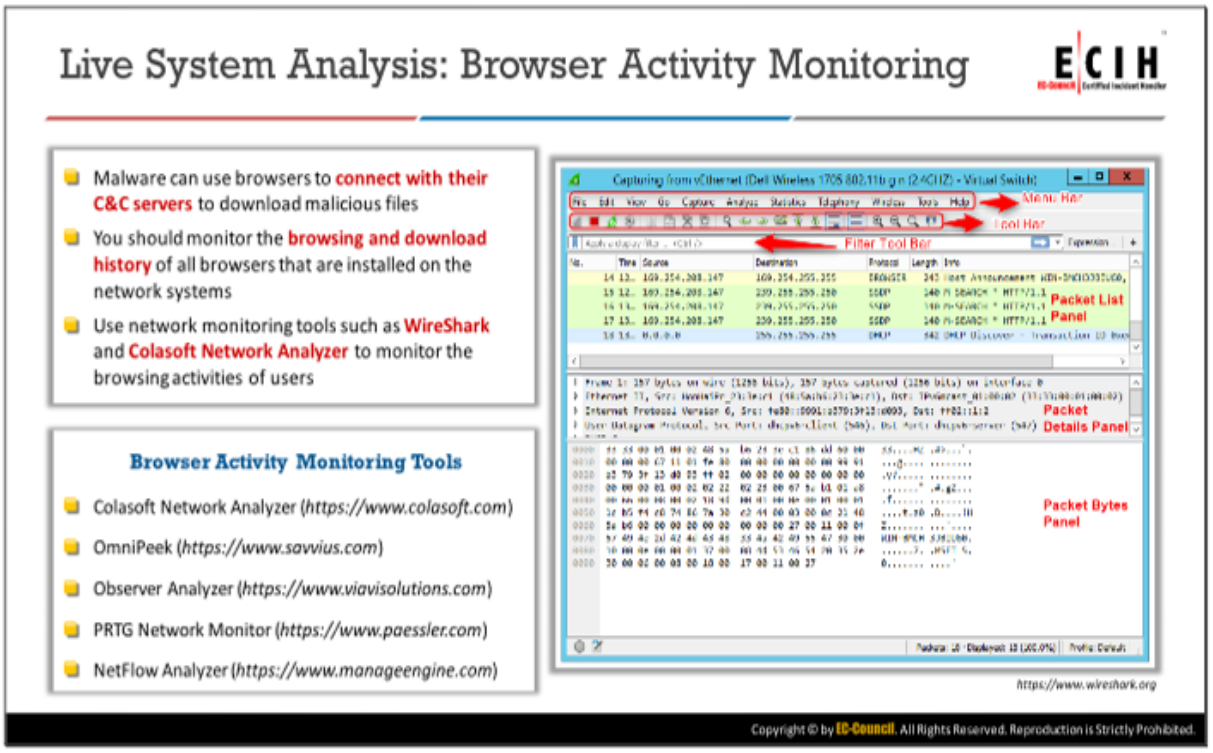
Live-System Analysis: Scheduled Task Monitoring

Attackers design malware to remain inactive and trigger at a specific date or event. Attackers enable malware to schedule tasks using Windows Task Scheduler. Therefore, checking for scheduled tasks will help you find malware such as logic bombs capable of executing at different triggers.

Use command-line arguments such as **schtasks** to display a list of all the system scheduled tasks. Responders can also use the Windows Task Scheduler tool to view scheduled tasks.

Some additional Windows-scheduled-task monitoring tools are as follows:

- Monitoring Task Scheduler Tool (MoTaSh) (*https://github.com*)
- ADAudit Plus (*https://www.manageengine.com*)
- CronitorCLI (*https://cronitor.io*)
- SolarWinds Windows Scheduled Task Monitor (*https://www.solarwinds.com*)



Live-System Analysis: Browser Activity Monitoring

Malware can use browsers to connect with command-and-control (C&C) servers, malicious websites, and other DNS servers to download malicious files. Therefore, incident responders must inspect suspicious browsing activities to identify malware type and system location. You should monitor browsing and download histories of all browsers installed on network systems.

As browsers use 80, 443, or 8080 ports to connect to networks, check for any browsing activities that might have occurred through other ports. Examine web caches, monitor web access at firewalls, and filter web access by URL and malicious strings in web logs. Use network monitoring tools such as WireShark and Colasoft Network Analyzer to monitor user browsing activities.

- **Wireshark**

  Source: *https://www.wireshark.org*

  Wireshark is a widely used network protocol analyzer. It captures and intelligently browses traffic passing through a network. Components and features of Wireshark are as follows:

  Components

  - **Menu Bar**: Hosts features of Wireshark
  - **Tool Bar**: Hosts more frequently used tools and icons
  - **Filter Tool Bar**: Filters traffic according to various options
  - **Packet List Panel**: Displays captured packets
  - **Packet Details Panel**: Displays detailed granular information about captured packets

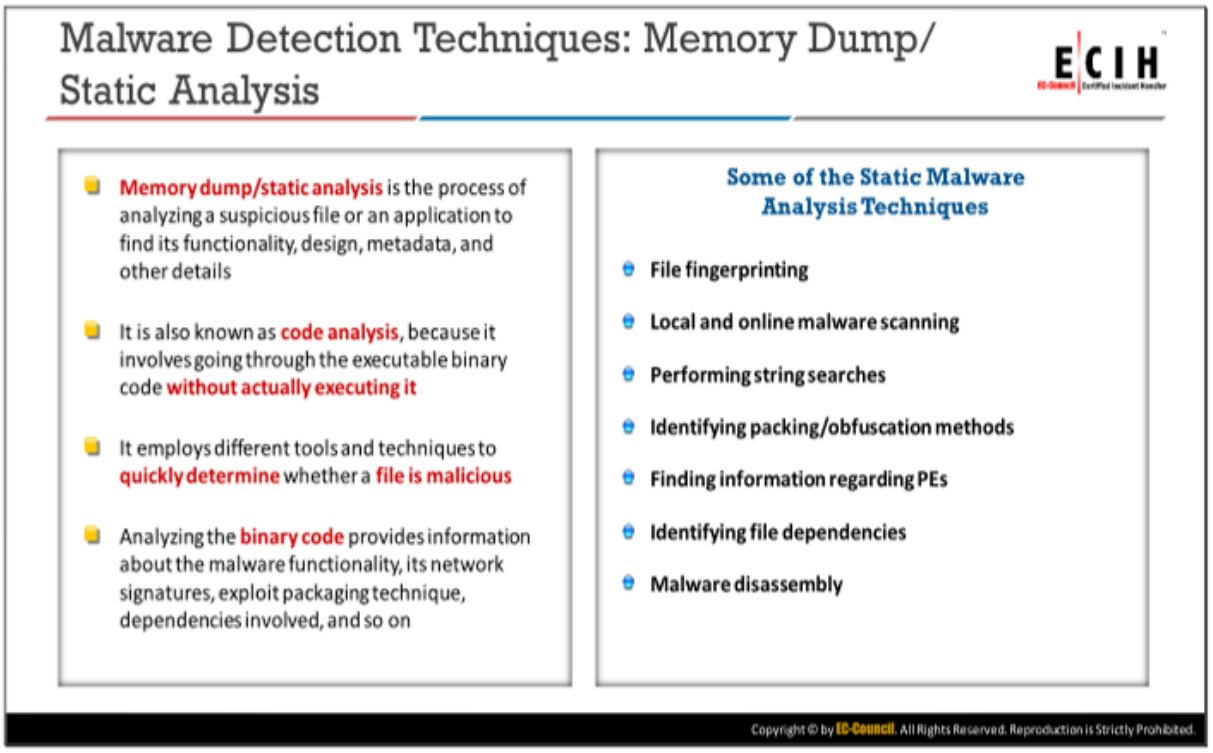
- **Packet Byte Panel**: Displays captured-packet bytes in hex dump format

Features

- Deep inspection of hundreds of protocols
- Live capture and offline analysis
- Standard three-pane packet browser
- Runs on Windows, Linux, OS X, Solaris, Free Berkeley software distribution (FreeBSD), NetBSD, and many other OSs
- Captured network data can be browsed using a GUI or the teletypewriter (TTY)-mode TShark utility

Some additional network traffic monitoring tools include:

- Colasoft Network Analyzer (*https://www.colasoft.com*)
- OmniPeek (*https://www.savvius.com*)
- Observer Analyzer (*https://www.viavisolutions.com*)
- PRTG Network Monitor (*https://www.paessler.com*)
- NetFlow Analyzer (*https://www.manageengine.com*)



Malware Detection Techniques: Memory-Dump/Static Analysis

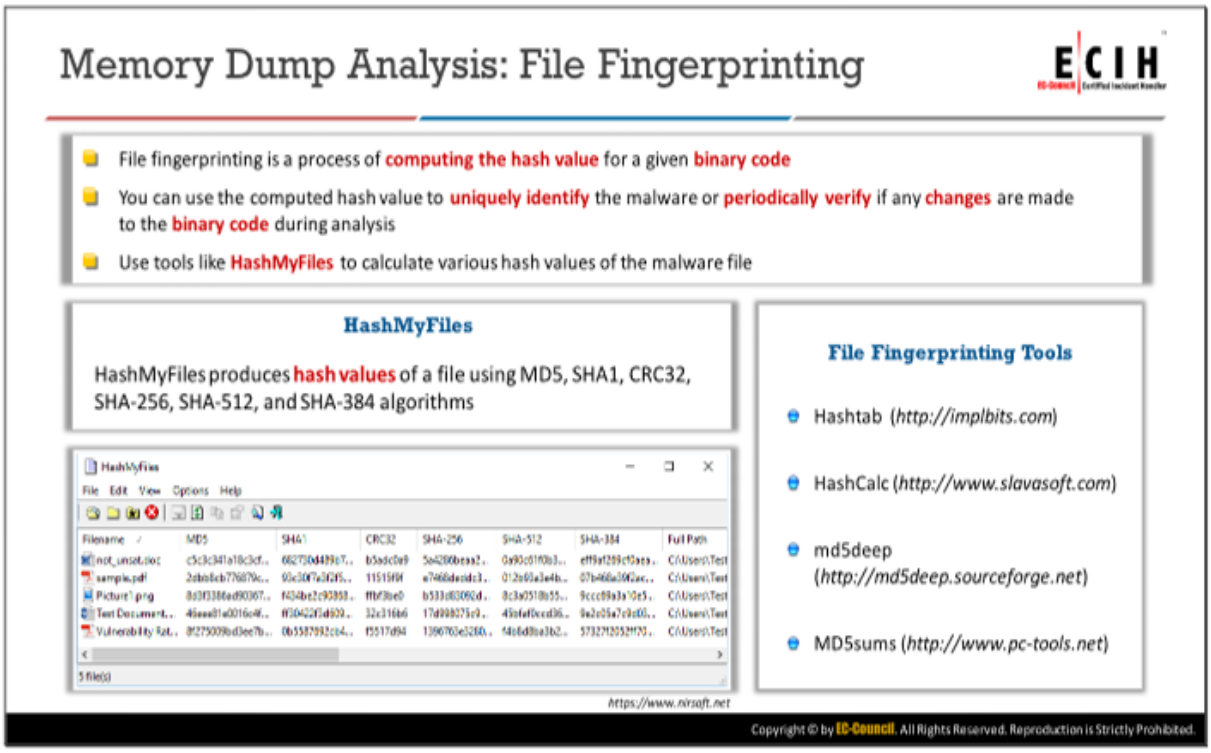
Memory-dump or static analysis is the process of analyzing a suspicious file or application to determine its functionality, code, metadata, and other details. It is also known as code analysis because it involves stepping through executable binary code line by line without actually executing it to better understand the malware and its purpose. In other words, it is the process of investigating an executable file without actually running or installing it. It is safe to statically analyze such suspect files because incident responders do not install or execute them. However, some malware does not require installation to performing malicious activities, so it is better that incident responders statically analyze such files in a controlled environment.

Static scrutiny involves analyzing suspicious files without executing malicious code or instructions. The process includes using different tools and techniques to determine malicious parts of programs or files, gathering information about malware functionality, and collecting generated technical pointers or simple signatures. Such pointers include file name, message-digest algorithm 5 (MD5) checksums or hashes, file type, and file size. Static analysis also involves accessing source or binary code to find data structures, function calls, call graphs, etc. representing malicious behavior. Incident responders can use various tools to analyze binary code to understand file architecture and system impact. Compiling system source code into a binary executable will result in data losses, which makes code analysis more difficult. Analyzing binary code provides information about malware functionality, network signatures, exploit packaging, dependencies, etc.

A binary executable file is mostly examined manually without actually executing it, which requires extraction of intriguing data such as data structures, utilized functions, and call graphs from the malicious file, all data that incident responders cannot see after program compilation.

Some static malware analysis techniques include:

- File fingerprinting
- Local and online malware scanning
- Searching strings
- Identifying packing/obfuscation methods
- Finding PE information
- Identifying file dependencies
- Malware disassembly



Memory-Dump Analysis: File Fingerprinting

File fingerprinting is a process of computing hash values for given binary codes to identify and track data across networks. The process includes calculation of cryptographic hashes and other binary-code key encryption material to recognize binary-code functions and compare them to those of other previously encountered binary codes and programs. The process also includes identification of common binary cryptographic algorithms and comparing them with those of suspicious binaries to identify whether the given binary is malware. The process also reveals implementation weaknesses in encryption algorithms. Therefore, computed hashes can be used to uniquely identify malware or periodically verify whether any changes have been made to binary code during analysis.

These fingerprints are used to track and identify similar programs from a database. Fingerprinting does not work for certain records including encrypted or password-secured, picture, audio, and video files, which have different contents than the predefined fingerprint.

MD5 and secure hash algorithm 1 (SHA-1) are the most commonly used hash functions for malware analysis. Various tools such as HashMyFiles can be used to fingerprint suspect files as part of static analysis. HashMyFiles is a GUI-based tool that can calculate various hashes.

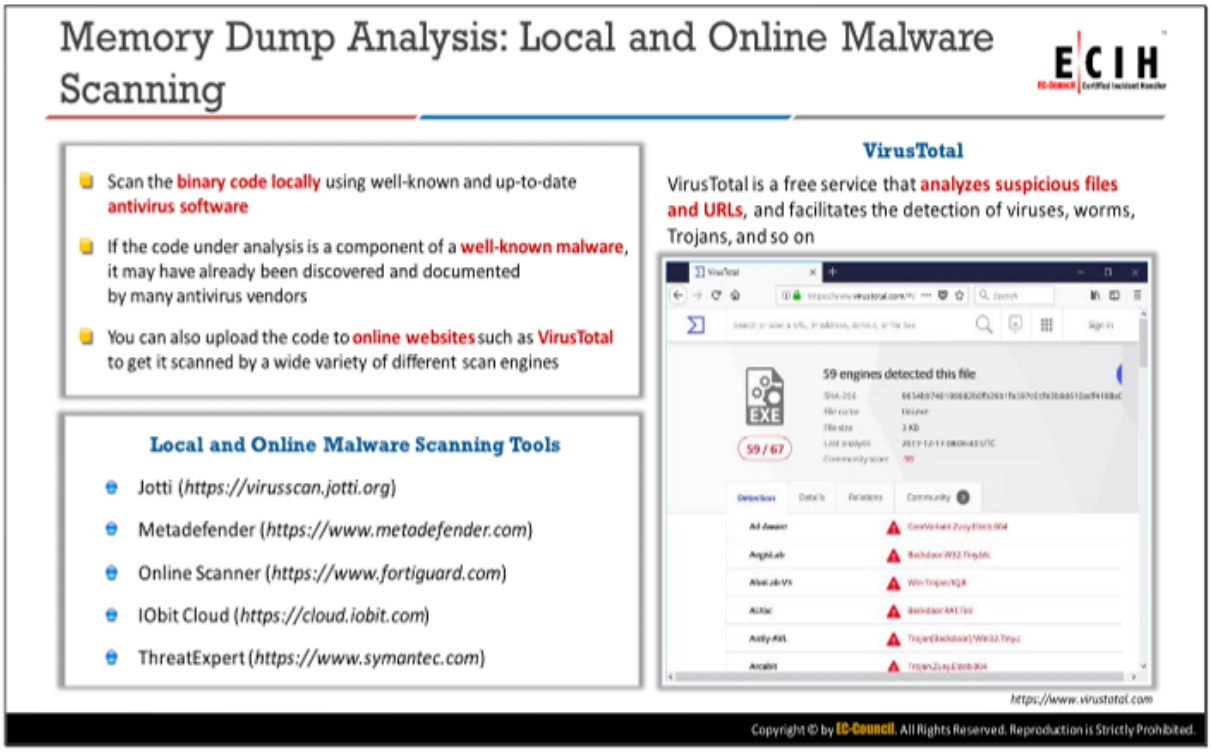
- **HashMyFiles**

  Source: *https://www.nirsoft.net*

  HashMyFiles uses MD5, SHA1, CRC32, SHA-256, SHA-512, and SHA-384 algorithms to produce file hashes. The program also provides file information such as the full path, creation and modification dates, and file size, attributes, version, and extension. This data will help search and compare similar files.

Some additional file fingerprinting tools include:

- Hashtab (*http://implbits.com*)
- HashCalc (*http://www.slavasoft.com*)
- md5deep (*http://md5deep.sourceforge.net*)
- MD5sums (*http://www.pc-tools.net*)
- tools4noobs—Online hash calculator (*https://www.tools4noobs.com*)
- Cryptomathic (*http://extranet.cryptomathic.com*)



Memory-Dump Analysis: Local and Online Malware Scanning

You can scan binary code locally using well-known up-to-date antivirus software. If analyzed code is a component of well-known malware, it already may have been discovered and documented by many antivirus software vendors. You can also upload code to online websites such as VirusTotal to scan it by a wide variety of scan engines.

VirusTotal calculates hashes of suspect files and compares them to known malware hashes stored in online and offline malware databases to recognize malicious codes. This process simplifies further investigation by offering better insight into code, its functionality, and other essential details.

- **VirusTotal**

  Source: *https://www.virustotal.com*

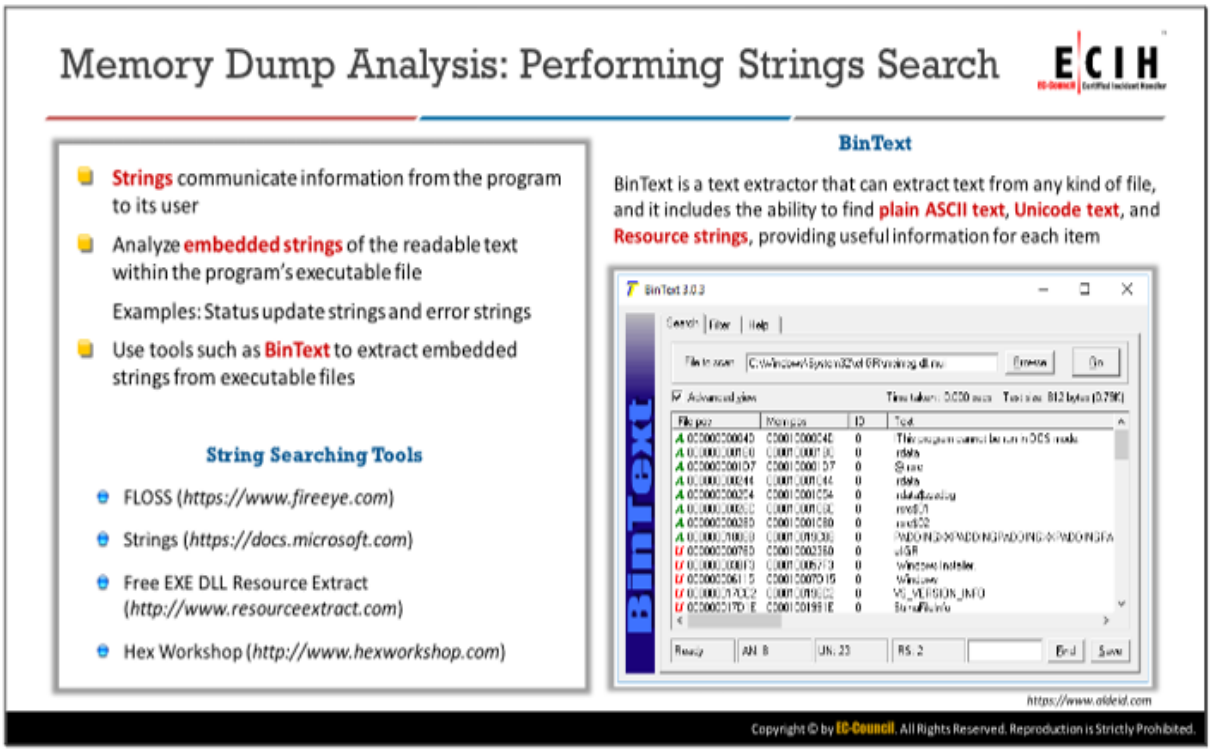
  VirusTotal is a free service that analyzes suspicious files and URLs and facilitates detection of viruses, worms, Trojans, etc. It generates a report providing the total number of engines that marked the file as malicious, the malware name, and additional malware information if available.

  It also offers important online-file-analysis details such as target machine, compilation timestamp, file type, compatible processors, entry point, PE sections, dynamic link libraries (DLLs), PE resources, different hashes, IP addresses accessed or contained in the file, program code, and types of connections established.

Some additional local and online malware scanning tools include:

- Jotti (*https://virusscan.jotti.org*)
- Metadefender (*https://metadefender.opswat.com*)

- Online Scanner (*https://www.fortiguard.com*)
- IObit Cloud (*https://cloud.iobit.com*)
- ThreatExpert (*https://www.symantec.com*)
- Malwr (*https://malwr.com*)
- Valkyrie (*https://valkyrie.comodo.com*)
- Dr.Web® Online Scanners (*https://vms.drweb.com*)
- UploadMalware.com (*http://www.uploadmalware.com*)
- ThreatAnalyzer (*https://www.threattrack.com)*
- Payload Security (*https://www.payload-security.com*)
- Anubis (*https://sourceforge.net*)
- Windows Defender Security Intelligence (WDSI) (*https://www.microsoft.com*)
- Bitdefender® Quickscan (*https://www.bitdefender.com*)



Memory-Dump Analysis: Performing Strings Search

Software programs include some strings that are commands for performing specific functions such as printing output. Strings communicate information from the program to its user. Various strings could represent malicious program intent such as reading internal memory or cookie data embedded in compiled binary code.

Searching through strings can provide information about basic program functionality. During malware analysis, search for malicious strings that could determine harmful actions a program could perform. For instance, if the program accesses a URL, it will have that particular URL string stored in it. You should be attentive while looking for strings and search for embedded and encrypted strings included in suspect files.

Use tools such as BinText to extract embedded strings from executable files. Ensure that the tool can scan and display both American standard code for information interchange (ASCII) and Unicode strings. Some tools can extract all strings and copy them to a text or document file. Use such tools to copy strings to a text file for ease in searching malicious strings.

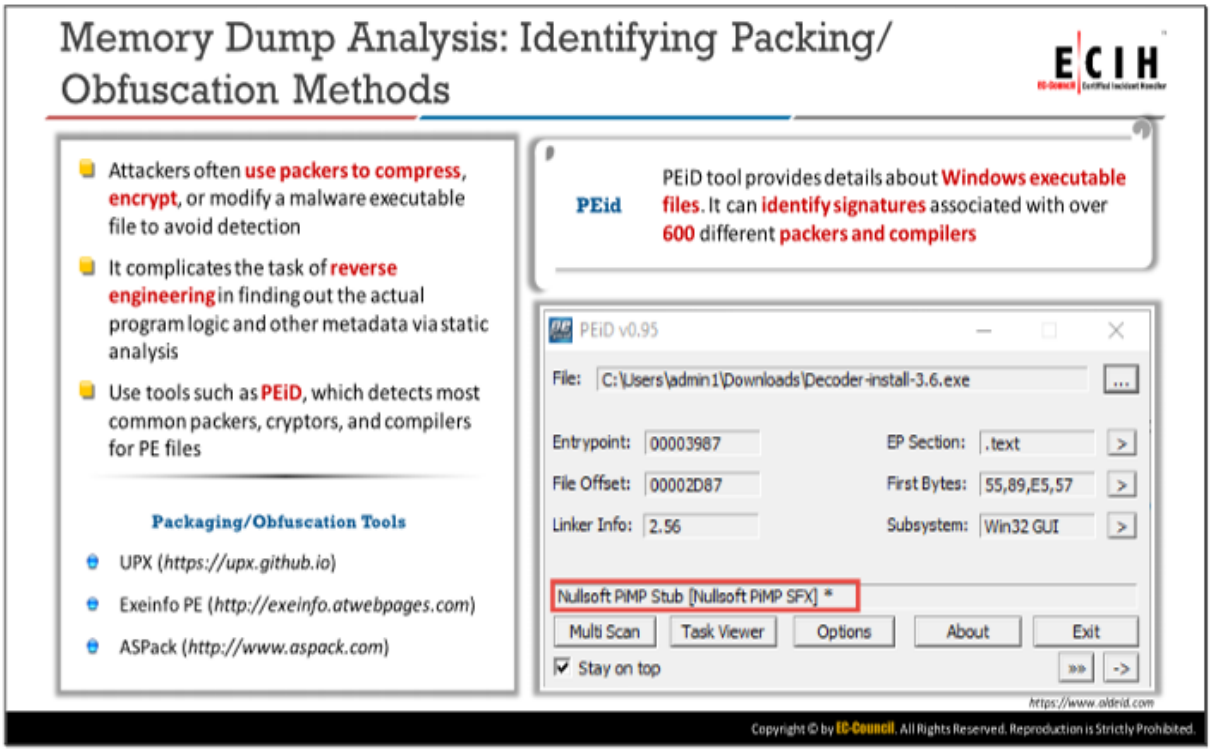
- **BinText**

  Source: *https://www.aldeid.com*

  BinText is a text extractor that can extract text from any kind of file and includes ability to find plain ASCII text, Unicode text, and resource strings, providing useful information for each item.

Some additional string searching tools include:

- FLOSS (*https://www.fireeye.com*)
- Free EXE DLL Resource Extract (*http://www.resourceextract.com*)
- Hex Workshop (*http://www.hexworkshop.com*)
- Strings (*https://docs.microsoft.com*)



Memory-Dump Analysis: Identifying Packing/Obfuscation Methods

Attackers use packing and obfuscation or packers to compress, encrypt, or modify malware executable files to avoid detection. Obfuscation also hides program execution. When the user executes a packed program, a small wrapper program also runs to decompress the packed file and then runs the unpacked file, which complicates reverse engineering by static analysis to determine actual program logic and other metadata.

Therefore, you should try to determine whether files include packed elements and locate tools or methods used to pack it. For this task, use tools such as PEiD, which detects the most commonly used packers, cryptors, and compilers for PE executable files. Finding packers will ease the task of selecting a tool for unpacking or restoring code. You can also attempt to rebuild malicious-code executable content from memory dumps by correlating malware-induced memory artifacts with those of on-disk applications.

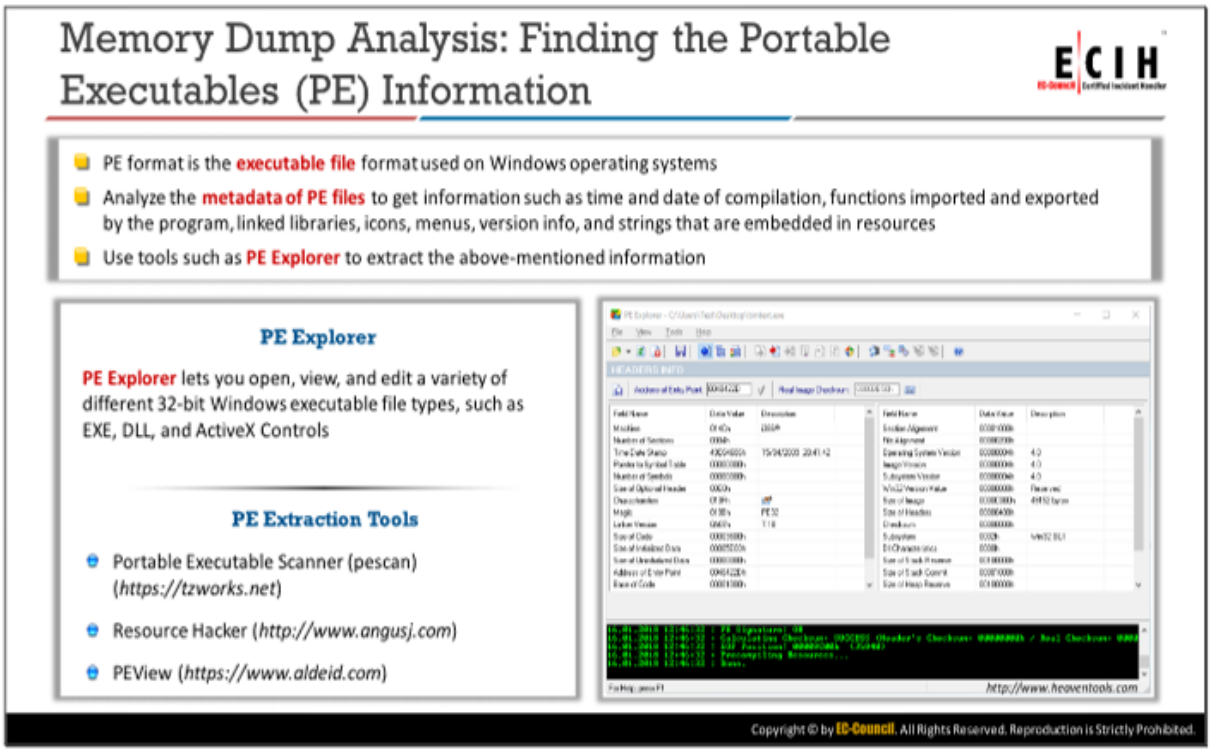
- **PEiD**

  Source: *https://www.aldeid.com*

  PEiD is a free tool that provides details about Windows executable files. It can identify signatures associated with over 600 different packers and compilers. This tool also displays types of program packers as well as entry points, file offsets, EP sections, and packing subsystems.

Some additional packaging/obfuscation tools include:

- UPX (*https://upx.github.io*)
- Exeinfo PE (*http://exeinfo.atwebpages.com*)
- ASPack (*http://www.aspack.com*)



Memory-Dump Analysis: Finding the Portable Executables (PE) Information

PE is the Windows-OS executable-file format, which stores the information a Windows system requires to manage executable codes. PE also stores program metadata, which helps find additional file details. For instance, Windows binary code is in PE format and contains information such as file creation and modification times, import and export functions, compilation time, DLLs, linked files, strings, menus, and symbols. PE format contains headers and sections, which store file metadata and OS code mapping.

File PEs contain the following information:

- **.text:** Contains instructions and program codes that the CPU executes
- **.rdata:** Contains import and export information and other read-only data used by the program
- **.data:** Contains the program's global data, which the system can access from anywhere
- **.rsrc:** Comprises executable-employed resources such as icons, images, menus, and strings offering multilingual support

You can use header information and tools such as PE Explorer to gather/extract additional file or program details/features.

- **PE Explorer**

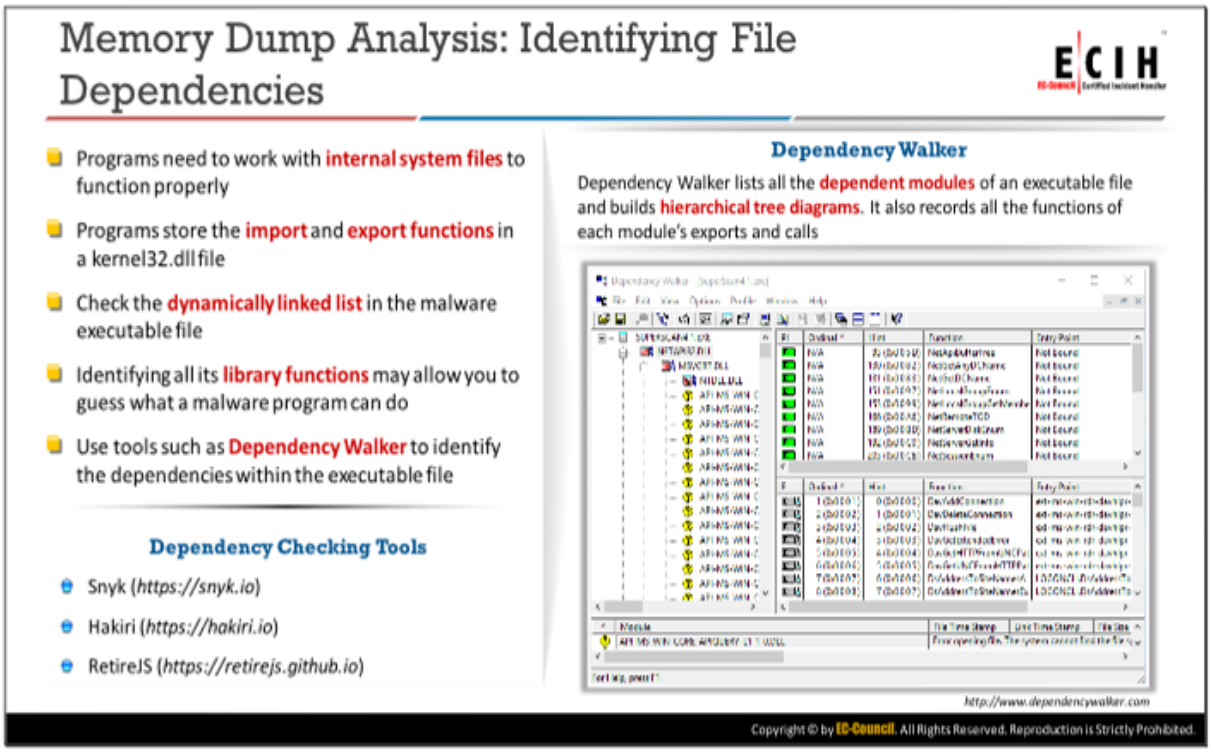
  Source: *http://www.heaventools.com*

  PE Explorer lets you open, view, and edit different 32-bit Windows executable files (also called "PE files") ranging from common types such as EXE, DLL, and ActiveX Controls to less familiar ones such as SCR (screensavers), CPL (control panel applets), SYS,

MSSTYLES, BPL, DPL, and more (including executable files that run on the MS Windows Mobile platform).

Some additional PE extraction tools include:

- Portable Executable Scanner (pescan) (*https://tzworks.net*)
- Resource Hacker (*http://www.angusj.com*)
- PEView (*https://www.aldeid.com*)



Memory-Dump Analysis: Identifying File Dependencies

Software programs depend on various OS built-in libraries to perform specified system actions and must work with internal system files to function correctly. Programs store import and export functions in a kernel32.dll file. File dependencies contain information about internal system files that programs require to install, register, and locate on the machine and function properly thereafter.

You must find libraries and file dependencies, as they contain information about application run-time requirements, and then check whether they can be located and analyzed as they can provide information about malware files. File dependencies include linked libraries, functions, and function calls. Check dynamically linked lists in malware executable files. Determine all the library functions that may allow you to infer what malware programs can do. You should know all the various dlls used to load and run programs.

Some standard dlls include:

dll	Description
Kernel32.dll	Core functionality such as access and manipulation of memory, files, and hardware
Advapi32.dll	Provides access to advanced core Windows components such as Service Manager and Registry
User32.dll	User interface components such as buttons and scrollbars and components for controlling and responding to user actions/inputs
Gdi32.dll	Functions for displaying and manipulating graphics
Ntdll.dll	Windows kernel interface
WSock32.dll and Ws2_32.dll	Networking DLLs that help connect to a network or perform network-related tasks
Wininet.dll	Supports higher-level networking functions

Table 4.1: Standard dlls.

You can use tools such as Dependency Walker to identify dependencies within executable files.

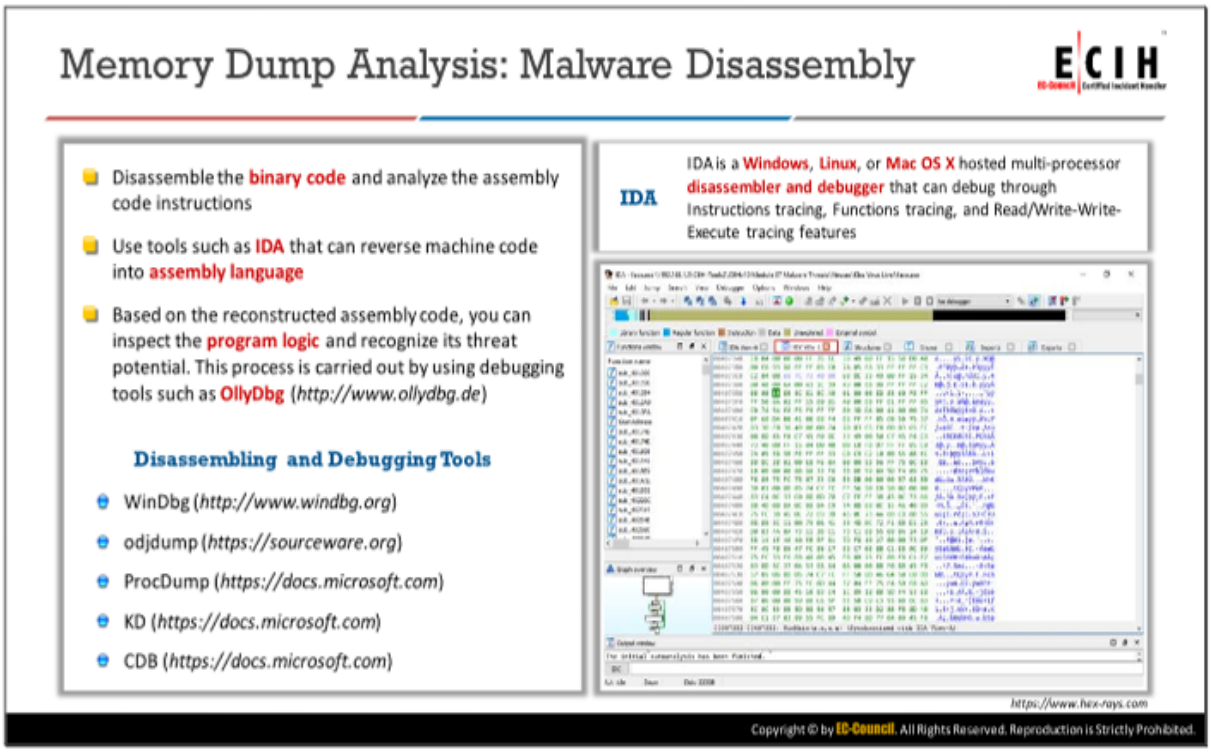
- **Dependency Walker**

  Source: *http://www.dependencywalker.com*

  Dependency Walker lists all the dependent modules of an executable file and builds hierarchical tree diagrams. It also records all the functions that each module exports and calls and detects many common application problems such as missing and invalid modules, import/export mismatches, circular dependency errors, mismatched machine modules, and module initialization failures.

Some additional dependency extraction tools include:

- Snyk (*https://snyk.io*)
- Hakiri (*https://hakiri.io*)
- Retire.js (*https://retirejs.github.io*)



Memory-Dump Analysis: Malware Disassembly

Memory-dump/static analysis also involves dismantling suspicious executables into binary format to study their functionalities and features, which helps identify the language used to program malware, reveal API functions, etc. From reconstructed assembly code, you can inspect program logic and recognize its threat potential.

Sometimes attackers develop malware to connect back to their C&C servers. Therefore, while disassembling suspicious files, incident responders must also look for any malicious function calls in binary-code subroutines. Attackers usually design malware to evade reverse engineering and maintain their confidentiality. Therefore, incident responders also must analyze disassembled code for any antireverse-engineering malware, which can be carried out using debugging tools such as IDA Pro and OllyDbg.

- **IDA Pro**

  Source: *https://www.hex-rays.com*

  IDA Pro is a multiplatform disassembler and debugger that explores binary programs, for which source code is not always available, to develop binary-program execution maps. IDA Pro symbolically represents instructions in assembly language, the same way a processor executes them; thus, it is easy for you to find harmful or malicious processes.

  Features

  - **Disassembler**

    The IDA Pro disassembler explores binary programs, for which source code is not always available, to develop binary-program execution maps.

- **Debugger**

  The IDA Pro debugger is an interactive tool complementing the dissembler to statically analyze binary programs in one step. It bypasses obfuscation, which helps the assembler to process hostile code in depth.

Some additional debugging tools include:

- OllyDbg (*http://www.ollydbg.de*)
- WinDbg (*http://www.windbg.org*)
- objdump (*https://sourceware.org*)
- ProcDump (*https://docs.microsoft.com*)
- KD (*https://docs.microsoft.com*)
- CDB (*https://docs.microsoft.com*)
- NTSD (*https://docs.microsoft.com*)

Memory Dump Analysis Using Volatility Framework

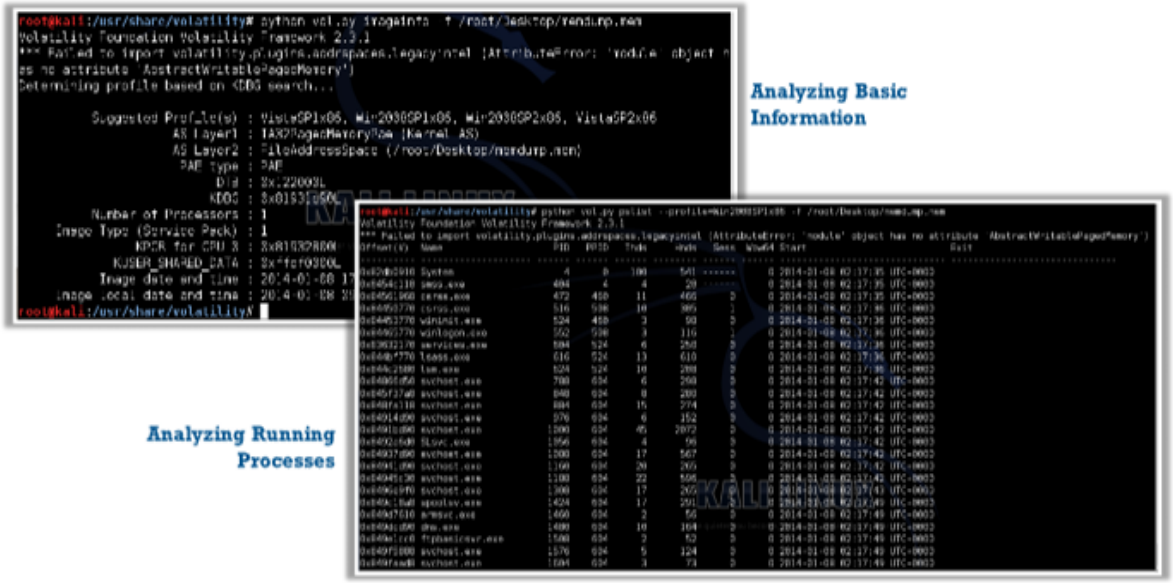
- **Volatility** is a python-based memory analysis tool that is capable of performing **various forensic operations**
- It can be used by the incident handler to analyze digital artifacts from **memory dumps** in order to identify any **anomalies**

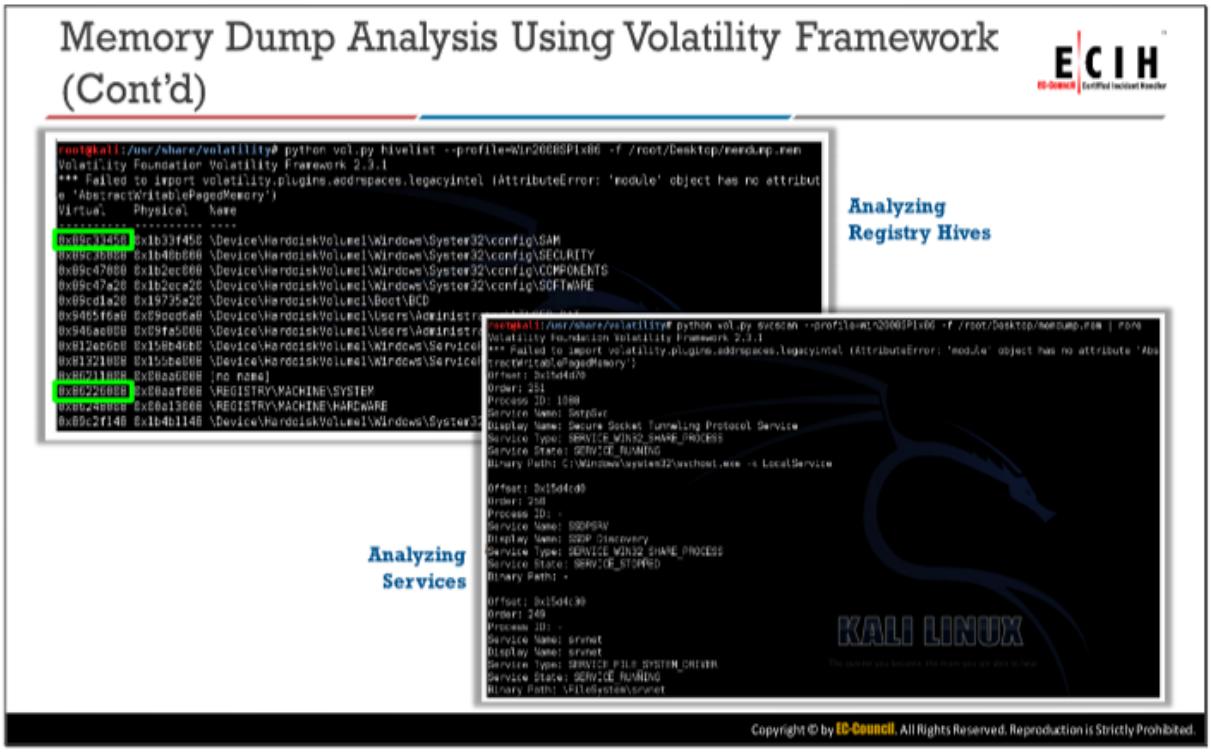
Steps to Analyze Memory Dump of the Compromised System Using Volatility Framework

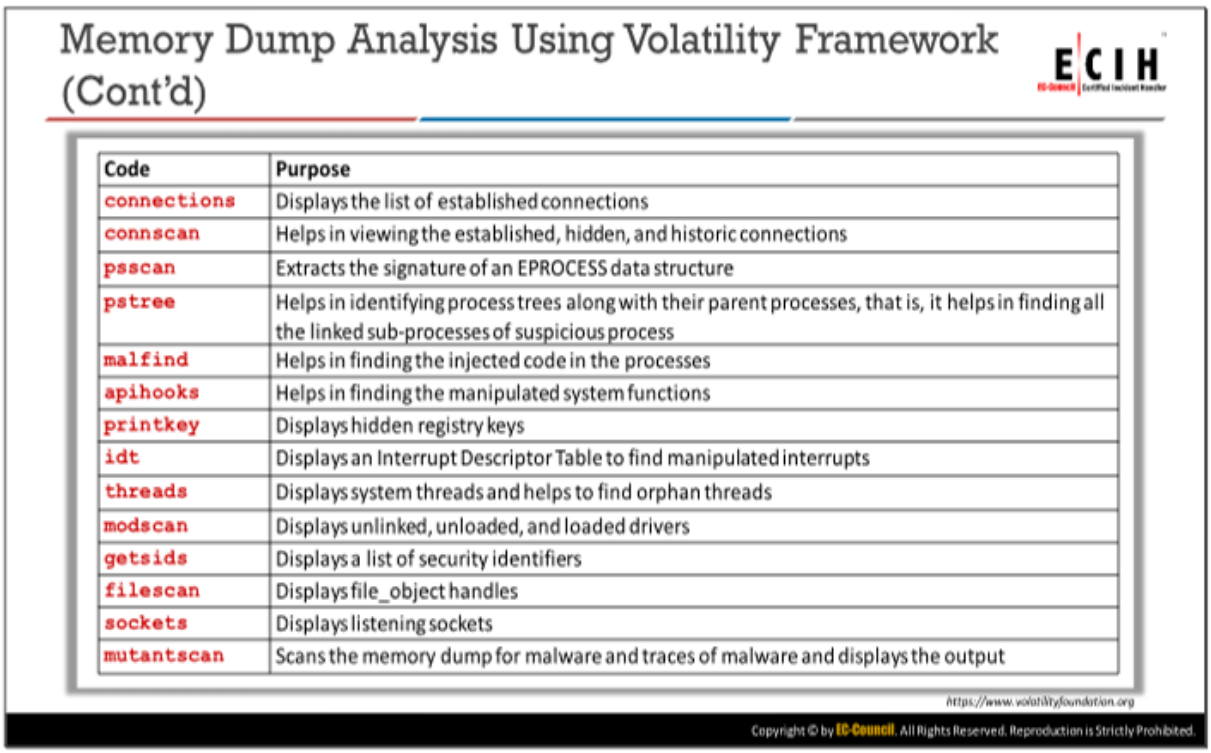
- Create a memory dump of the system and store it as a **.dd image file** or **.mem file** on the analysis system. Here we save the memory file as **memdump.mem**
- Use a **sandbox environment**, preferably a Linux-based virtual machine, to **analyze the memory dump** of a compromised system for detection and analysis of malware
- Install Volatility memory forensics tool on the analysis system by using Linux command `apt-get install volatility` and copy the image file onto it
- To analyze the image using the Volatility forensics tool in the command line interface of the Linux system, the incident responder should navigate to /usr/share/volatility by using the `cd /usr/share/volatility` command and use the following command syntax: `python vol.py [plugin] -f [image] –profile=[profile name]`

https://www.volatilityfoundation.org

Memory Dump Analysis Using Volatility Framework (Cont'd)







Code	Purpose
connections	Displays the list of established connections
connscan	Helps in viewing the established, hidden, and historic connections
psscan	Extracts the signature of an EPROCESS data structure
pstree	Helps in identifying process trees along with their parent processes, that is, it helps in finding all the linked sub-processes of suspicious process
malfind	Helps in finding the injected code in the processes
apihooks	Helps in finding the manipulated system functions
printkey	Displays hidden registry keys
idt	Displays an Interrupt Descriptor Table to find manipulated interrupts
threads	Displays system threads and helps to find orphan threads
modscan	Displays unlinked, unloaded, and loaded drivers
getsids	Displays a list of security identifiers
filescan	Displays file_object handles
sockets	Displays listening sockets
mutantscan	Scans the memory dump for malware and traces of malware and displays the output

Memory-Dump Analysis Using Volatility Framework

Source: *https://www.volatilityfoundation.org*

Memory-dump analysis helps collect various malware artifacts from systems without power supplies, which is crucial because turning on systems may spread malware, change system settings or memory, and disrupt malware-infection evidence. Memory-dump analysis also helps incident responders conduct deeper analyses to assess malware impact, location, and

propagation methods. Incident responders can use common memory analysis frameworks or tools like Volatility to perform memory-dump analysis.

Volatility is a Python-based forensics memory analysis tool that can be used by incident handlers to analyze memory-dump digital artifacts to identify anomalies and detect data erasure applications.

Following are steps required to analyze compromised-system memory dumps using the Volatility framework:

- Create a system memory dump file and store it as a **.dd** image file or **.mem** file on the analysis system. Here, we saved the memory file as **memdump.mem**.
- Use a sandbox environment, preferably a Linux-based virtual machine, to analyze the compromised-system memory dump for malware.
- Install the Volatility memory forensics tool on the analysis system using the Linux command **`apt-get install volatility`** and copy the image file onto the analysis system.

  To analyze the image using the Volatility forensics tool, incident responders should navigate to the Volatility folder on the analysis system using the **`cd /usr/share/volatility`** command at the Linux command-line interface and then use the following syntax:

  `python vol.py [plugin] -f [image] -profile=[profile name]`

  In the following screenshot, the Volatility-tool help option is executed using the **`python vol.py -h`** command.

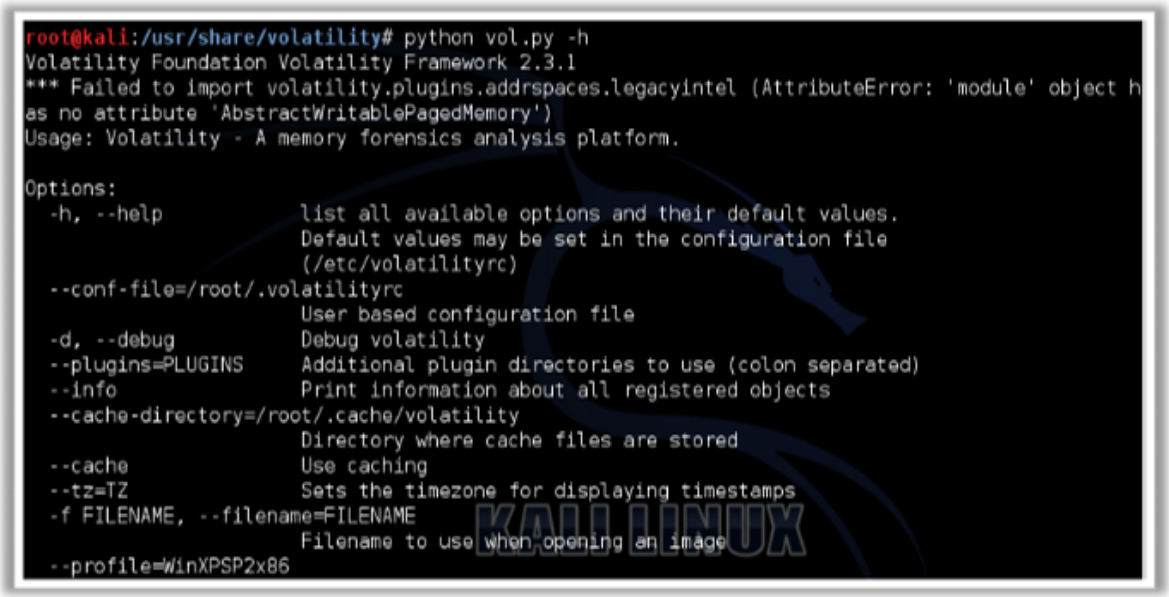


Figure 4.6: Screenshot displaying Volatility-tool help options.

- Use the following command to analyze basic image information like operating system and image date and time in the Volatility tool:

  ```
  python vol.py imageinfo -f /root/Desktop/memdump.mem
  ```

- Use the following command to analyze the image-file running process in the Volatility tool:

  ```
  python vol.py pslist --profile=Win2008SP1x86 -f
  /root/Desktop/memdump.mem
  ```

  The screenshot shown on the above slide, you can observe the following columns:

  - **Offset**: Indicates hexadecimal location of the random-access-memory (RAM) process
  - **Name**: Indicates process name
  - **PID**: Indicates process ID
  - **PPID**: Indicates parent-process ID

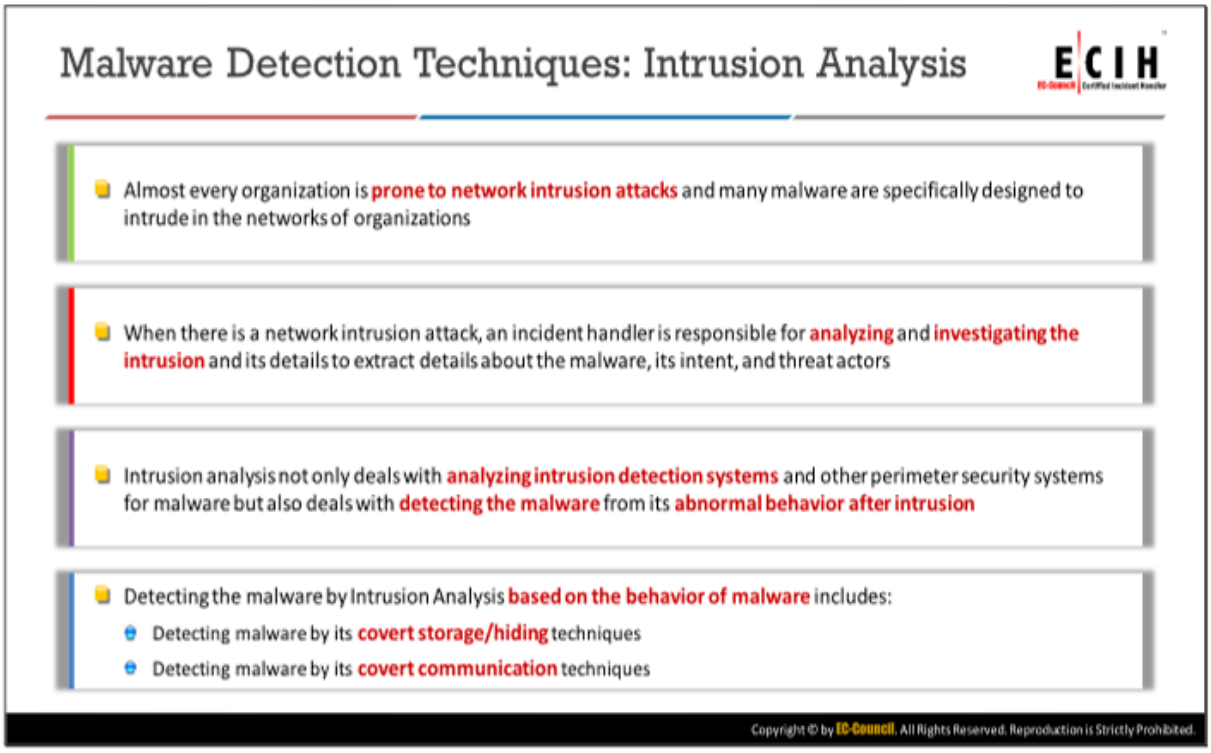
- Use the following command to analyze image-file services in the Volatility tool:

  ```
  python vol.py svcscan --profile=Win2008SP1x86 -f
  /root/Desktop/memdump.mem | more
  ```

- Use the following command to analyze image-file registry hives in the Volatility tool:

  ```
  python vol.py hivelist --profile=Win2008SP1x86 -f
  /root/Desktop/memdump.mem
  ```

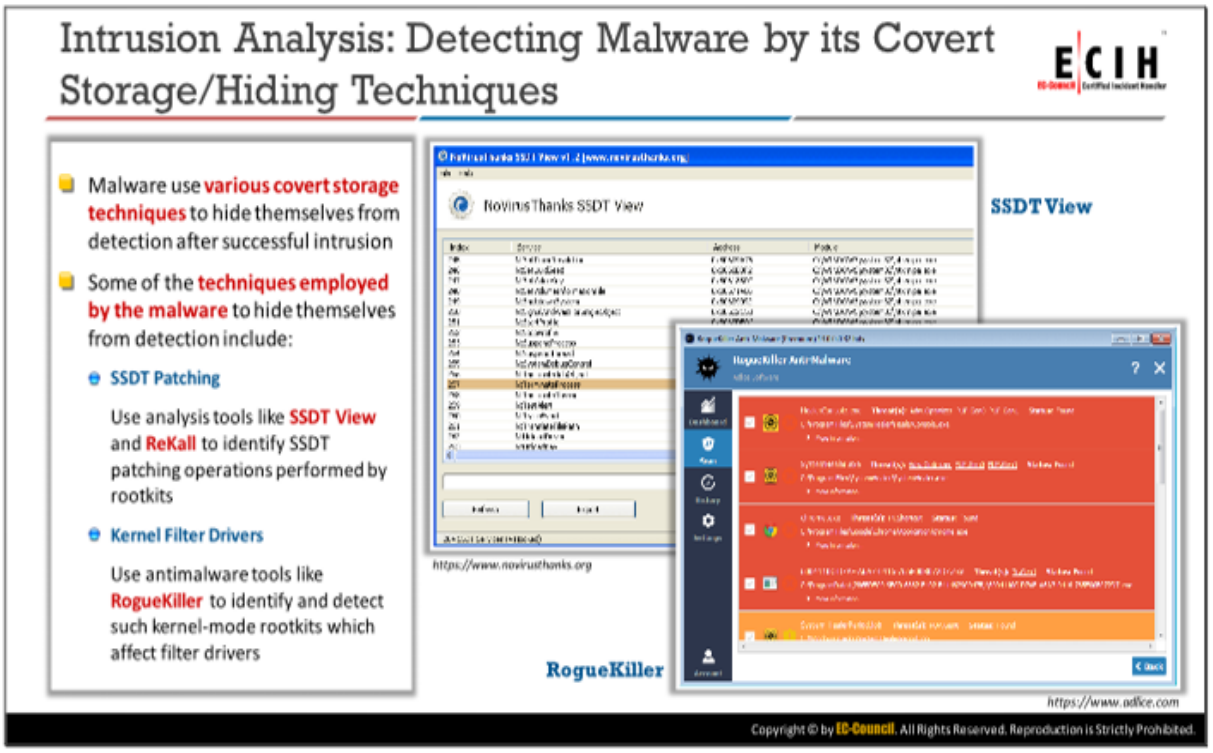
The table shown on the above slide contains other Volatility commands that incident responders can use to obtain required artifacts during forensic investigation of infected machines.



Malware Detection Techniques: Intrusion Analysis

Almost every organization is prone to network intrusion attacks, and many malwares are specifically designed to intrude an organization's networks and spread the infection throughout. When there is a network intrusion attack, incident handlers are responsible for analyzing and investigating the intrusion and its details to determine the extract malware details, intent, and threat actors. Intrusion analysis not only involves analyzing IDSs and other perimeter security systems for malware but also involves detecting malware based on its abnormal behavior after intrusion. Intrusion analysis also assists incident responders in identifying and detecting malware spreading throughout the network and detection of network misuses and anomalies.

Malware-behavior-based intrusion analysis includes detecting malware by its covert storage/hiding and communication techniques.



Intrusion Analysis: Detecting Malware by its Covert Storage/Hiding Techniques

Malware uses various covert storage techniques to hide from detection after successful intrusion. As an incident responder, you must be aware of various obfuscation techniques employed by malwares to hide their detection, and you are also responsible for detecting such covertly hidden malwares.

Following are some techniques employed by malwares to hide from detection:

- **SSDT Patching/Hooking**

  A system service descriptor table (SSDT) is a table in the Windows OS kernel, which stores entry-level addresses and is similar to a kernel-instruction index page. Each SSDT entry indicates a set of functions to be performed based on user requirements. Because every kernel function or operation is instructed by the SSDT, malware such as kernel-mode rootkits alter and modify the kernel data structure and attempt to hook with the SSDT data structure. By this modification, the SSDT alone can enable the rootkit to affect all the user operations. A rootkit attaches itself to an SSDT by overwriting legitimate SSDT instructions, pointing the SSDT toward rootkit functions and performing desired malicious operations.

  You can use analysis tools like SSDT View, ReKall, and SSDTViewer to identify such SSDT patching operations performed by rootkits.

- **SSDT View**

  Source: *https://www.novirusthanks.org*

  SSDT View is a Microsoft Windows OS utility designed to list the most significant SSDT aspects including service indexes, addresses, and names and the module name corresponding to the service address. You can export the report to a file for further analysis. This utility can be useful for listing table hooks despite the x64 OS shipping with kernel patch protection (KPP)/PG-ready kernels, which translates to kernel bug checking if the SSDT is modified. However, this feature can be disabled through some malware-like tricks; hence, the development of SSDT View.

- **Kernel Filter Drivers**

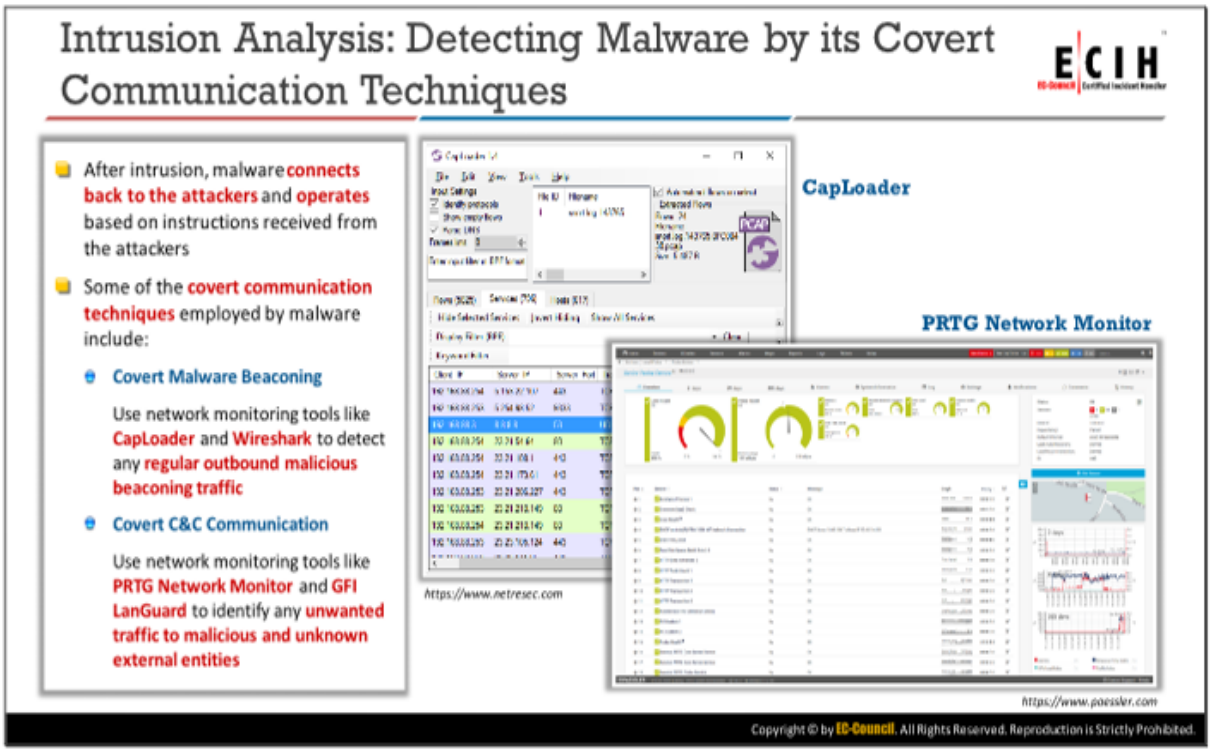
  Most rootkit malwares attempt to hide themselves in kernel filter drivers, which follow a multilayered approach to enable system peripheral devices to communicate with each other. A driver stack consists of a set of drivers that pass information from one driver to another; for example, if you press any key on your keyboard, information is converted multiple times by multiple drivers before the desired content is displayed on the computer screen. Rootkits like TDL4 and keyloggers use such kernel filters to hide themselves from detection.

  As an incident responder, you must remove filter devices from attached pointers and use antimalware tools like RogueKiller to identify and detect such kernel-mode rootkits that affect filter drivers.

  - **RogueKiller**

    Source: *https://www.adlice.com*

    RogueKiller is antimalware that can detect and remove generic malware and advanced threats like rootkits, rogues, and worms. It also detects controversial potentially unwanted programs (PUPs) and possible bad system modifications/corruptions (PUMs).



Intrusion Analysis: Detecting Malware by its Covert Communication Techniques

After intrusion, certain malwares attempt to connect back to attackers and operate on the basis of instructions received from attackers through encrypted communication. Malwares employ various covert communication techniques that enable malwares to stay hidden within host computer resources and communicate with malware developers or C&C servers. To detect such malware covert communication techniques, incident responders must be aware of them.

Following are some malware covert communication techniques:

- **Covert Malware Beaconing**

  Malwares that infiltrate live systems remain unnoticeable and hidden within computer system resources awaiting instructions from attackers. Such malwares generate regular outbound signals called "malware beacons" to external malicious servers to receive instructions and maintain connections. Beaconing is a regular process that can happen at any time depending on the malware.

  As an incident responder, you can use network monitoring tools like CapLoader and Wireshark to detect any regular outbound malicious beaconing traffic.

  - **CapLoader**

    Source: *https://www.netresec.com*

    CapLoader is a Windows tool designed to handle large volumes of captured network traffic. CapLoader indexes PCAP/PCAP next generation (PcapNG) files and visualizes their contents as a list of transmission control protocol (TCP) and user datagram protocol (UDP) flows. Users can select flows of interest and quickly filter out those

packets from loaded PCAP files. This tool can be useful for analyzing network traffic activity and determining whether a single host makes multiple DNS requests to 8.8.8.8. Such flows are merged together as one row in the services tab of the CapLoader interface.

- **Covert C&C Communication**

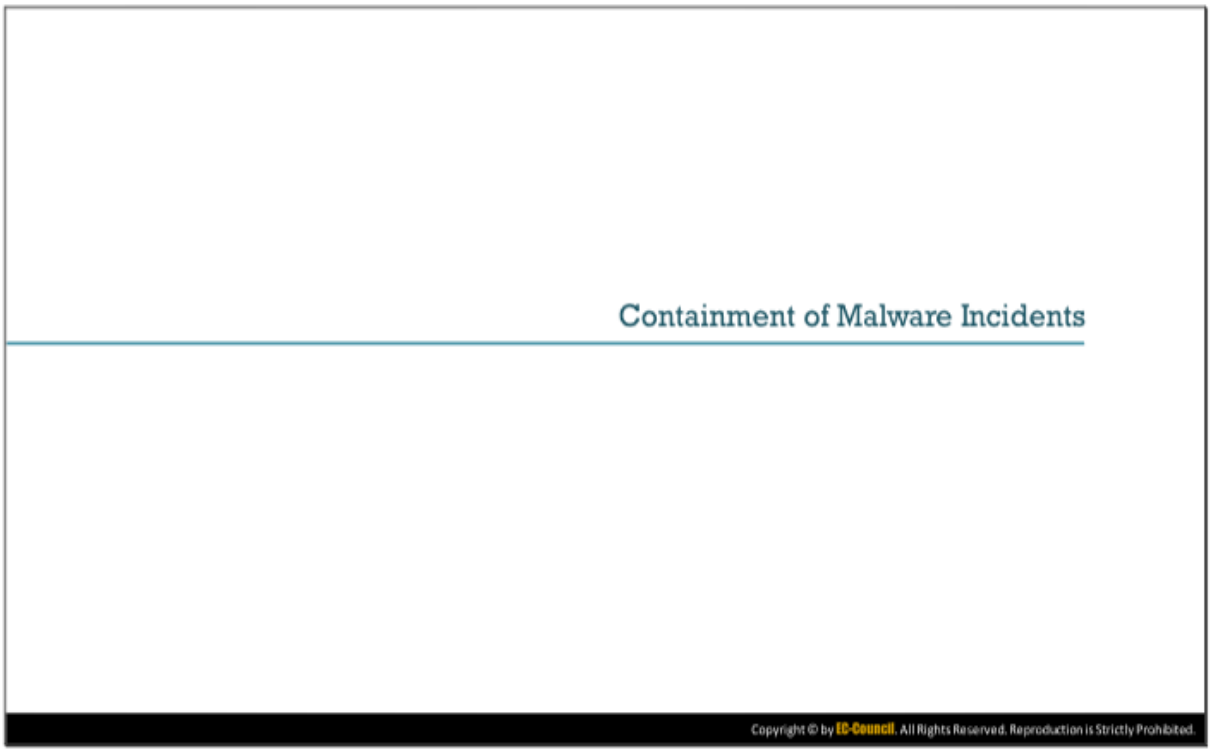
  Malwares often hide in intruded systems and keep open discreet communication channels with C&C servers to await instructions from or provide remote access to attackers. Such channels transmit very low intensity encrypted transmission signals under cover of general outbound traffic that might contain system data or instruction requests from remote C&C servers. C&C communication can occur using social media pages and images, discreet DNS traffic, and anonymous communication servers like Tor.

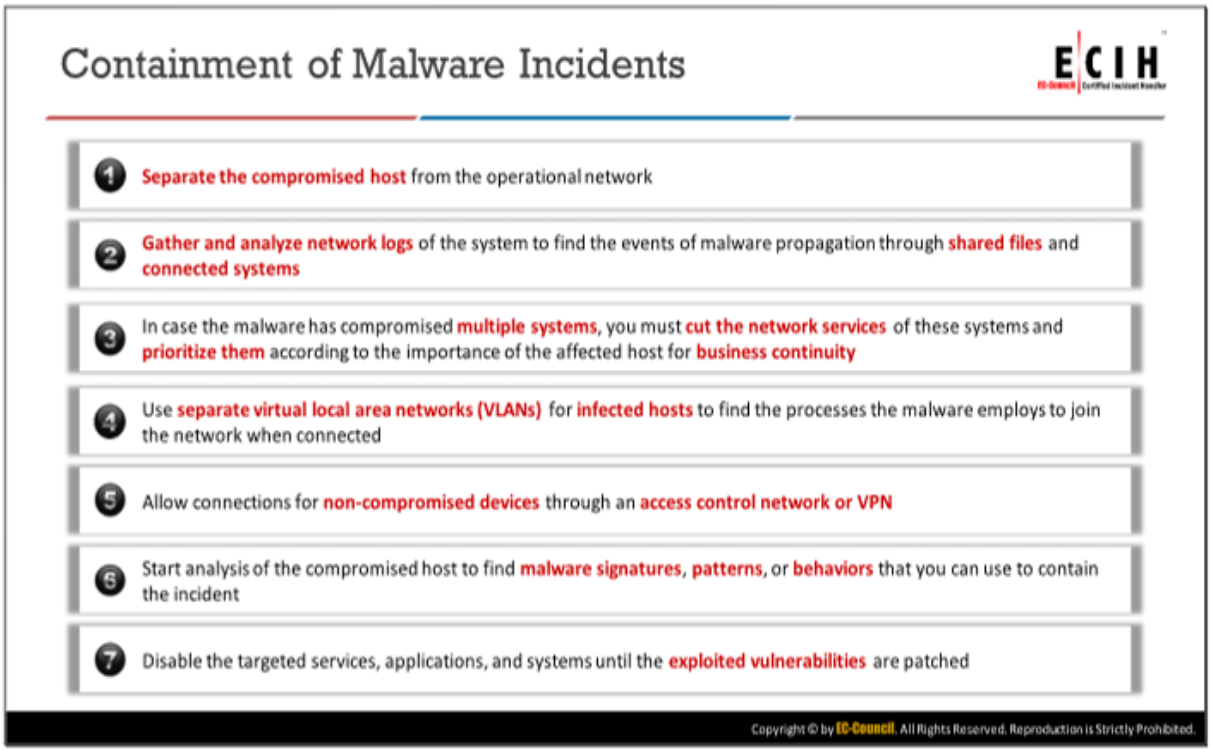
  You can use network monitoring tools like PRTG Network Monitor and GFI LanGuard to identify any unwanted traffic to malicious and unknown external entities.

  - **PRTG Network Monitor**

    Source: *https://www.paessler.com*

    PRTG Network Monitor is a network monitoring tool effectively used to monitor entire network infrastructures. PRTG supports most technologies including simple network management protocol (SNMP) (all versions), flow technologies (i.e., NetFlow, jFlow, sFlow), secure shell (SSH), Windows management instrumentation (WMI), Ping, and structured query language (SQL). Powerful APIs (Python, EXE, DLL, PowerShell, Visual Basic (VB), Batch Scripting, and representational state transfer (REST)) are used to integrate everything else. PRTG Network Monitor is available for every platform.





Containment of Malware Incidents

Once a responder validates a malware incident, the IH&R team should focus on containing it after obtaining approval from all the concerned authorities. The main intentions of malware containment are to prevent further spread throughout the network and minimize impact to the organization. In critical incidents involving widespread malware infection, incident handlers

should adopt advanced strategies. It is therefore important for an organization to outline its containment strategies for various malware incidents in its IR plan.

Containment strategies may vary with the nature of malware incidents. Following are some steps IH&R personnel should follow to contain malware incidents:

- After confirming malware infection, incident responders must separate the compromised host from the operational network.
- Incident responders must simultaneously gather and analyze network system logs to find malware propagation events through shared files and connected systems.
- When malware has compromised multiple systems, you must cut off compromised-system network services and prioritize compromised systems according to importance of affected hosts to maintain business continuity.
- Use separate virtual local area networks (VLANs) for infected hosts to determine which processes the malware employs to join the network when connected.
- Allow connections through an access control network or virtual private network (VPN) for noncompromised devices.
- Start analyzing the compromised host to find malware signatures, patterns, or behaviors that you can use to contain the incident.
- Disable targeted services, applications, and systems until exploited vulnerabilities are patched.
- Block all unnecessary host and firewall ports.
- Run host-based antivirus, firewall, and IDS software.
- Run registry monitoring tools to find malicious registry entries added by the backdoor, Trojan, or virus.
- Remove or uninstall programs or applications installed by the backdoor, Trojan, or virus.
- Remove malicious registry entries added by the backdoor, Trojan or virus.
- Delete malicious backdoor-, Trojan-, or virus-related files.

Eradication of Malware Incidents

- Eradication of Malware Incidents
- Antivirus Tools

Eradication of Malware Incidents

Content Filtering Tools: Use the **static characters of the malware**, such as strings and loaders, as filters to block the malware from entering systems, servers, emails, and other propagation elements

Network Security Devices: Add the **malware signature** to the **network security devices** such as firewalls and IPS to stop it from breaching the organization perimeter

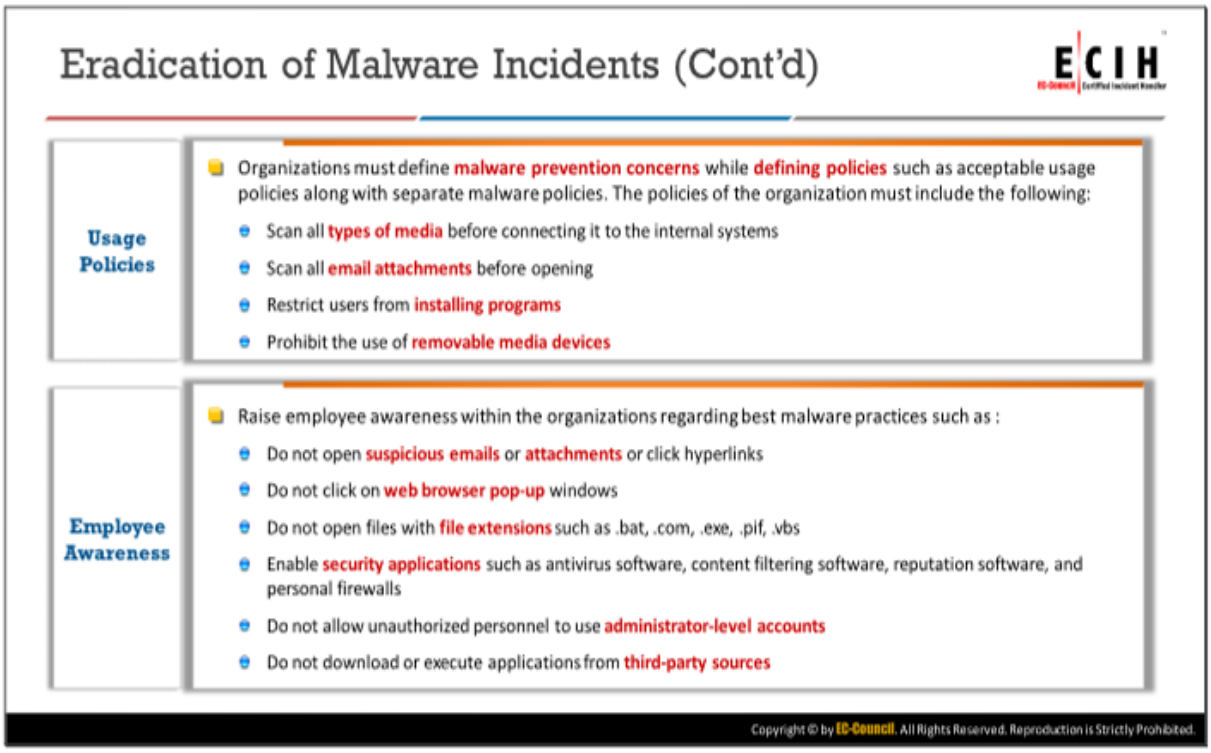
Malware Blacklisting: Block the **harmful URLs**, **IP addresses**, and **email-Ids** that have acted as a source for the spread of malware. **Blacklist** the services, programs, applications, and executables that install malware onto the system

Antivirus Tools: Update the antivirus tools to **detect the newly found malware** using signature, string, or heuristics-based techniques

Updating Malware Databases: Include the **Hashsums of the malware** to the online and offline databases for future reference of the organization as well as for public recognition

Fixing Devices: **Update the browsers**, applications, and operating systems, and **patch the vulnerabilities** that the malware had exploited as entry points

Manual Malware Scanning: Run a **full scan of the compromised system** with an updated antivirus program to remove the malicious codes, binaries, and related registry entries



Eradication of Malware Incidents

Although malware eradication implies completely removing malware from infected hosts, it involves more than this. Apart from just disinfecting infected hosts, it also includes eliminating vulnerabilities that led to infection of the host/network. Eradication should also eliminate further chances of similar infections.

Following are some steps that incident responders should follow to eradicate malware security incidents:

- **Content Filtering Tools**

  An organization must use various tools to mitigate malware threats. Antivirus software, intrusion prevention systems (IPSs), content filtering tools, and firewalls can help the IH&R team blacklist malware.

  Content filtering tools are highly useful when blocking malware showing static characteristics such as strings and loaders. For example, responders can block malicious spam email by appropriately configuring email servers and clients and installing antispam software to filter suspicious emails based on attributes such as content, attachment name and type, and email origin and signature.

- **Network Security Devices**

  Compromised-host network connectivity plays a major role in malware spreading and establishing communication between the attacker's tool and the malware C&C server. Thus, imposing temporary restrictions on network connectivity very effectively curbs malware attempts to compromise other hosts.

Incident responders must disconnect compromised devices by blocking their IP addresses or by physically removing their network cables. This approach includes isolating uncompromised subnets from the main network or eliminating network access to remote VPN users. For example, responders can maintain servers and workstations on separate subnets to ensure minimal functionality disruption.

IPS devices can prevent malware infections based on malware signatures and heuristics. Inline-network-based IPSs can identify and block malwares from infecting organizational hosts. Incident handlers should reconfigure IPS sensors according to severity of malware infections. IPS devices can contain malware spreading from both incoming and outgoing attacks. Furthermore, responders can customize IPS devices according to malware attributes and signatures.

- **Malware Blacklisting**

  Incident responders can also blacklist malwares to block them from executing. This method is applicable even if responders did not receive malware signatures from vendors. To blacklist malware, incident responders can simply enter names of files that should not be executed in OSs, host-based IPS products, and other network security tools. Although antivirus tools can quarantine and contain existing malwares, such tools may not be very effective for quarantining and containing new malwares. Therefore, incident responders should update antivirus software with the latest malware signatures/definitions.

- **Antivirus Tools**

  Responders may eradicate malware by cleaning infected systems/networks, quarantining infected files with antivirus software, using malware removal tools, manually removing malware, implementing vulnerability management technologies, and using network access control software. Organizations may also use automated eradication methods such as remotely triggering regular antivirus scans.

  To remove malware binaries and related registry entries, responders should run full antivirus scans with antivirus software containing updated signatures/definitions. Incident responders may also run online antivirus scans or use best practices suggested by antivirus software vendors.

  Responders should remember that when eradicating sophisticated malwares, rebuilding the system from scratch from trusted sources such as system installation disks or clean system images is always the safest approach.

- **Updating Malware Databases**

  Organizational malware databases should be regularly updated with signatures/definitions of new malware and the same should be reported to antimalware software vendors and antivirus software developers. Save malware signatures/definitions in the form of hashes for future organizational reference and public awareness/protection.

- **Fixing Devices**

  Once IH&R teams have detected network vulnerabilities that malwares have exploited, teams also must address such vulnerabilities on other network systems. Some examples include security misconfigurations, which incident responders can rectify by implementing proper access controls, or vulnerabilities in shared drives, which incident responders can update or stop using to contain attacks.

- **Manual Malware Scanning**

  Run a full scan of the compromised system with an updated antivirus program to remove malicious codes, binaries, related registry entries, scheduled tasks, and other malware-related files and folders. Recheck recovered devices for traces of malware before reintroducing the devices to the functional network.

- **Usage Policies**

  Organizations must clearly define malware prevention concerns while defining policies such as separate acceptable usage and malware policies.

  Organizations must include the following usage policies:

  - Scan all types of media before connecting them to internal systems
  - Scan all email attachments before opening them
  - Restrict users from installing unknown programs
  - Prohibit use of removable media devices

- **Employee Awareness**

  Organizations must teach their employees about malware best practices such as:

  - Do not open suspicious emails or attachments or click hyperlinks
  - Do not click on web browser popup windows
  - Do not open files with file extensions such as .bat, .com, .exe, .pif, or .vbs
  - Enable security applications such as antivirus, content filtering, reputation, and personal firewall software
  - Do not allow unauthorized personnel to use administrator-level accounts
  - Do not download or execute applications from third-party sources

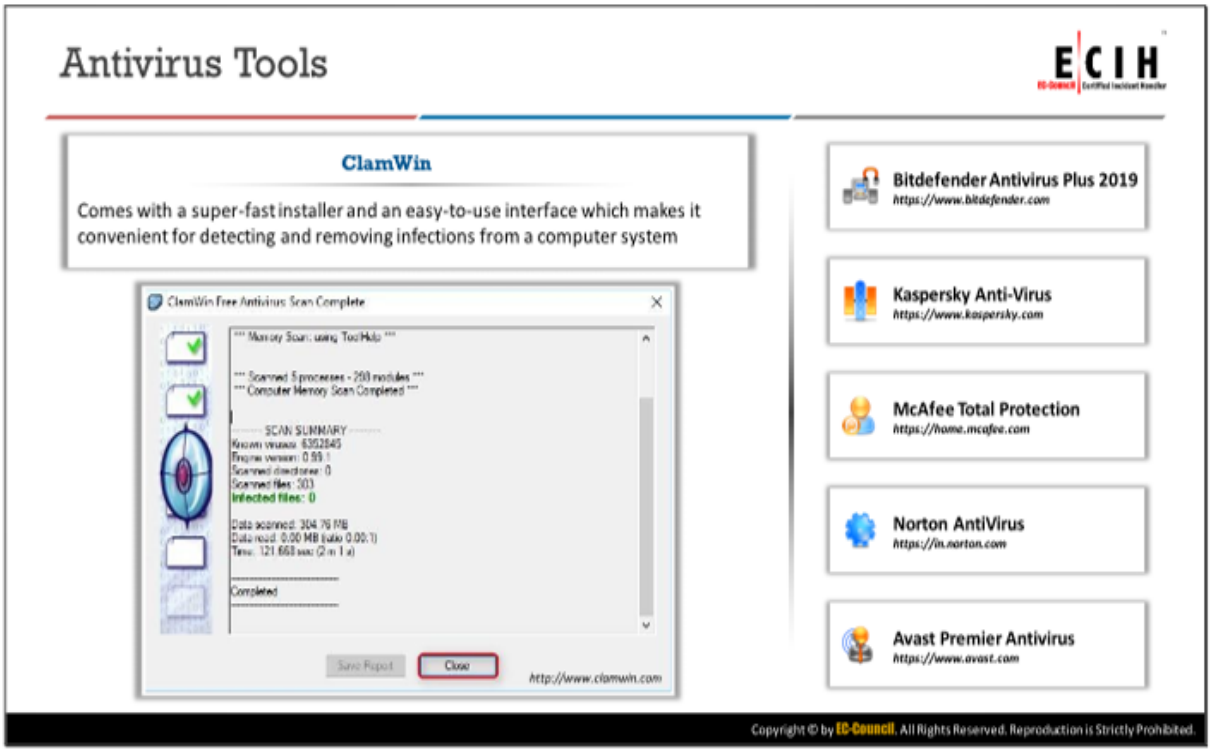
Apart from the aforementioned policies, organizations should implement many other strategies to eradicate malware security incidents as follows:

- Organizations must patch all the discovered vulnerabilities that malwares had exploited and inform antimalware manufacturers or developers
- Incident responders must submit malware information to online databases for further scrutiny and share malware details with their peers to prevent future similar incidents

- Organizations must implement application whitelisting to only allow network access to crucial applications
- Organizations must prohibit users from installing new applications by restricting user-account privileges
- Organizations must block websites containing malicious content and initiating auto-downloads to prevent automatic malware downloads
- Organizations must block all access to system and server basic input/output systems BIOSs by segregating functional systems using virtualization technologies
- Organizations must implement the following browser settings to prevent malware incidents:
  - Block popups
  - Disable auto-downloads
- Organizations must implement the following policies to eradicate Trojan incidents:
  - Avoid opening email attachments received from unknown senders
  - Block all unnecessary host and firewall ports
  - Avoid accepting programs transferred by instant messaging
  - Strengthen weak default configuration settings and disable unused functionalities including protocols and services
  - Monitor internal network traffic for odd ports or encrypted traffic
  - Avoid downloading and executing applications from untrusted sources
  - Regularly install patches and security updates for OSs and applications
  - Scan external USB drives, compact discs (CDs), and DVDs with antivirus software before use
  - Restrict desktop-environment permissions to prevent malicious-application installation
  - Avoid blindly typing commands and implementing prefabricated programs or scripts
  - Manage local workstation file integrity through checksums, auditing, and port scanning
  - Run host-based antivirus, firewall, and IDS software
- Organizations must implement the following policies to eradicate backdoor incidents:
  - Most commercial antivirus products can automatically scan and detect backdoor programs before they can cause damage
  - Educate users not to install applications downloaded from untrusted internet sites and email attachments

- Avoid installing untrusted software and ensure that a firewall protects each device
- Use antivirus tools such as McAfee and Norton to detect and eliminate backdoors
- Track open-source projects, such as code repositories, that enter the enterprise from untrusted external sources
- Inspect network packets using protocol monitoring tools
- If a computer is infected by backdoors, restart the infected computer in safe mode with networking

- Organizations must implement the following policies to eradicate virus and worm incidents:
  - Install antivirus software to detect and remove infections in real time
  - Develop an antivirus policy for safe computing and distribute it to all staff members
  - Pay attention to instructions while downloading any internet-based files or programs
  - Update antivirus software regularly
  - Avoid opening attachments received from unknown senders because viruses can spread by email attachments
  - Because viral infections can corrupt data, ensure you are regularly backing up data
  - Schedule regular scans for all drives after installation of antivirus software
  - Do not accept disks or programs without checking them first using a current version of an antivirus program
  - Ensure that any organizational executable code has been approved
  - Do not boot machines with infected bootable system disks
  - Stay informed about the latest virus threats
  - Check DVDs and CDs for viral infections
  - Ensure popup blockers are turned on and use internet firewalls
  - Run disk clean up and registry scanner once a week
  - Run antispyware or adware removal software once a week
  - Do not open files with more than one file type extension
  - Be cautious with files sent through instant messenger applications



Antivirus Tools

Incident handlers can employ various antivirus tools to eradicate malware from infected systems. Following are some important tools for eradicating malware:

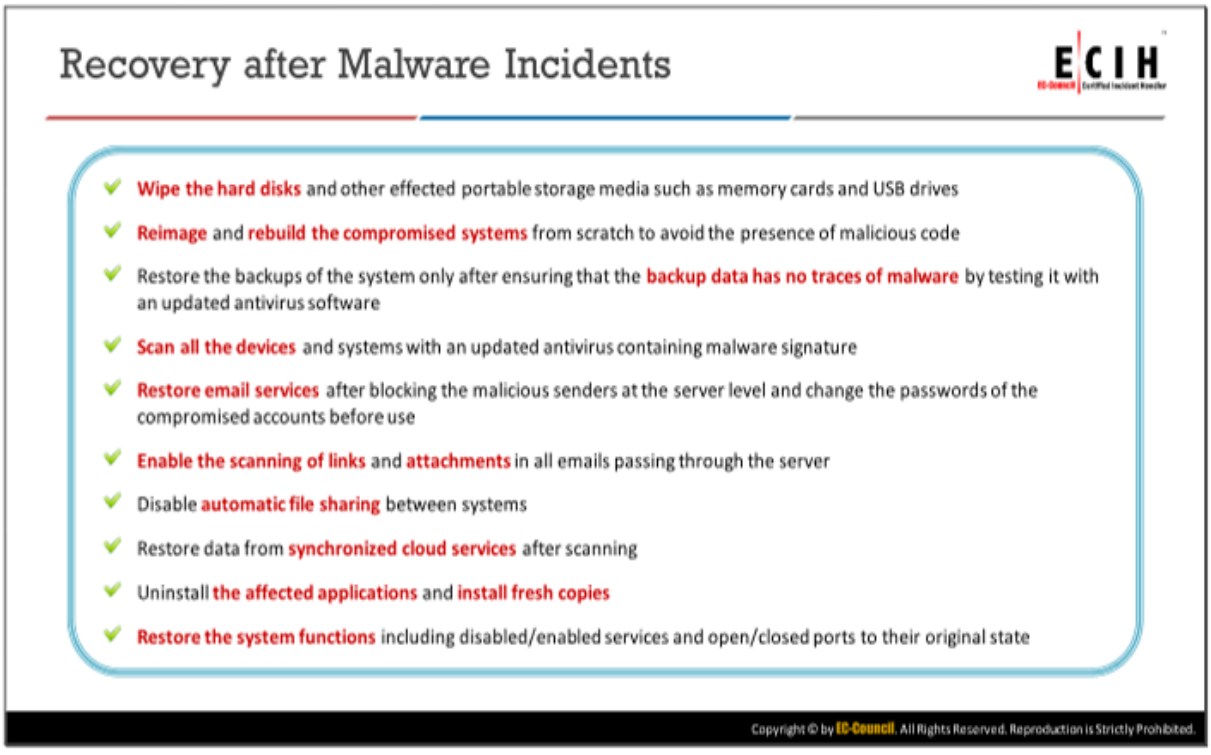
- **ClamWin**

  Source: *http://www.clamwin.com*

  ClamWin is a free, open-source antivirus program for Windows systems. It comes with a super-fast installer and an easy-to-use interface, which makes it convenient for detecting and cleaning infections from computer systems. It provides high detection rates for viruses and spywares and a scanning scheduler.

Following are some additional tools for eradicating malware:

- Bitdefender Antivirus Plus 2019 (*https://www.bitdefender.com*)
- Kaspersky Anti-Virus (*https://www.kaspersky.com*)
- McAfee Total Protection (*https://home.mcafee.com*)
- Norton AntiVirus (*https://in.norton.com*)
- Avast Premier Antivirus (*https://www.avast.com*)
- ESET Smart Security (*https://www.eset.com*)
- AVG Antivirus FREE (*https://www.avg.com*)
- Avira Antivirus Pro (*https://www.avira.com*)



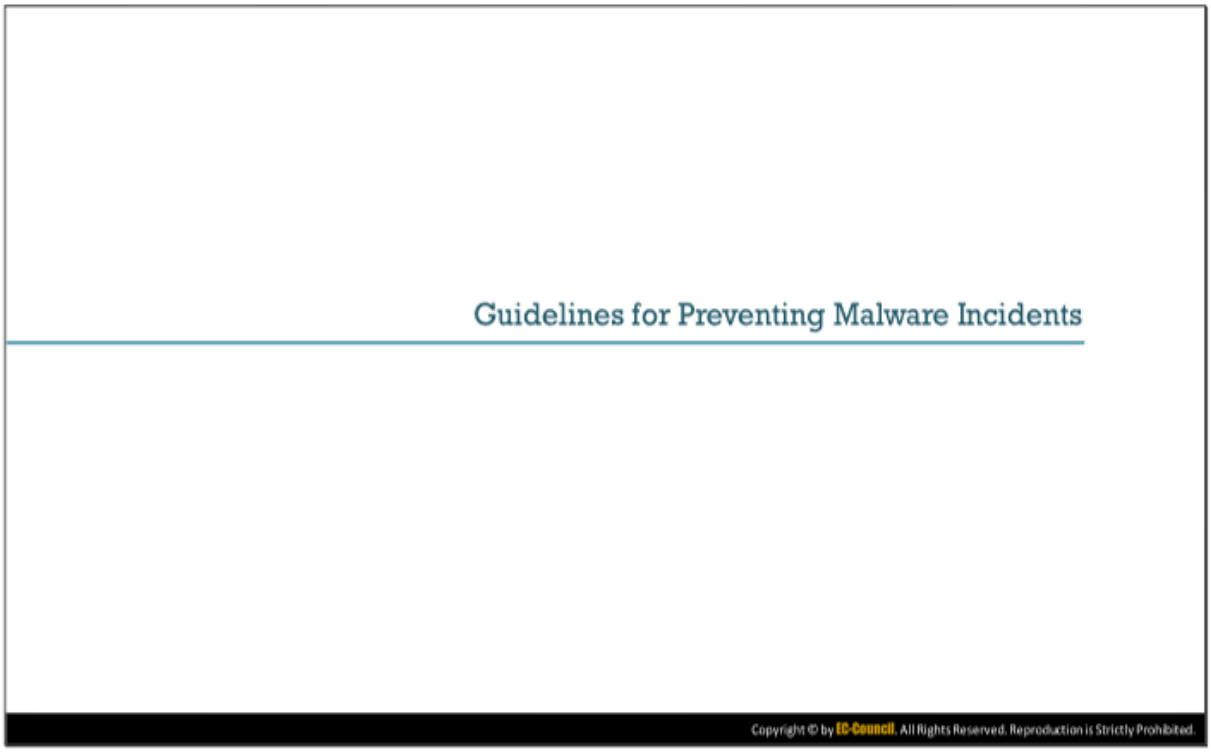
Recovery after Malware Incidents

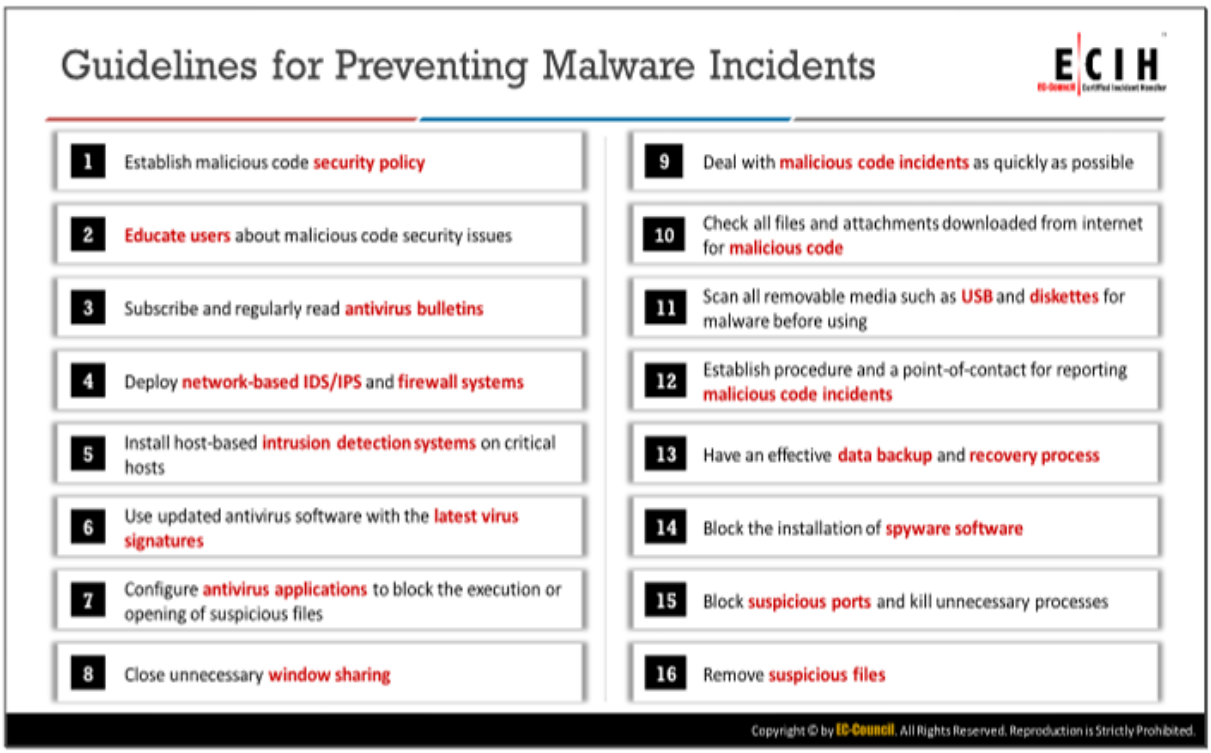
Following are some recovery steps that incident responders must follow after malware security incidents:

- Incident responders may recover infected systems by reimaging and rebuilding them from scratch or by removing temporary containment imposed on them. Delete all the

data from compromised systems and wipe impacted storage media such as hard disks, memory cards, and USBs.

- Incident responders should recover any data lost owing to infection using data recovery tools, trusted clean backup sources, or backup data in cloud synchronization
- Incident responders should scan hosts and file shares with updated antivirus software signatures/definitions, eliminate system/network vulnerabilities, and update routers
- Incident responders should run full scans on system and device backups to ensure that all traces of malware have been removed prior to using backups to restore servers, systems, and databases. Incident responders should use antivirus software containing updated malware signatures/definitions to ensure complete eradication of malware from hosts, networks, and servers.
- Prior to recovering infected services, incident responders should assess preproduction security risks to ensure that no more infections are detected and that the cause of the original infection has been entirely eradicated. It is good practice to inform all concerned parties, such as users, employees, and stakeholders, about recovery of suspended services. Then, the organization should restore recovered functions in stages and by adding better monitoring facilities.
- Restore email services after blocking at server level malicious email senders. All the employees must change their account passwords after attacks to prevent account compromise if malware accessed account security credentials. Enable two-factor authentication for organizational email, login, and user accounts. Enable scanning of links and attachments in all emails passing through the server.
- The organizations should disable automatic file sharing between systems. If files must be shared among connected systems, the organization should enable file sharing with authentication. Update file sharing applications or remove old versions and their associated files and install new versions directly from developer websites. Restore system functions including disabled/enabled services, open/closed ports, etc. to their original states.





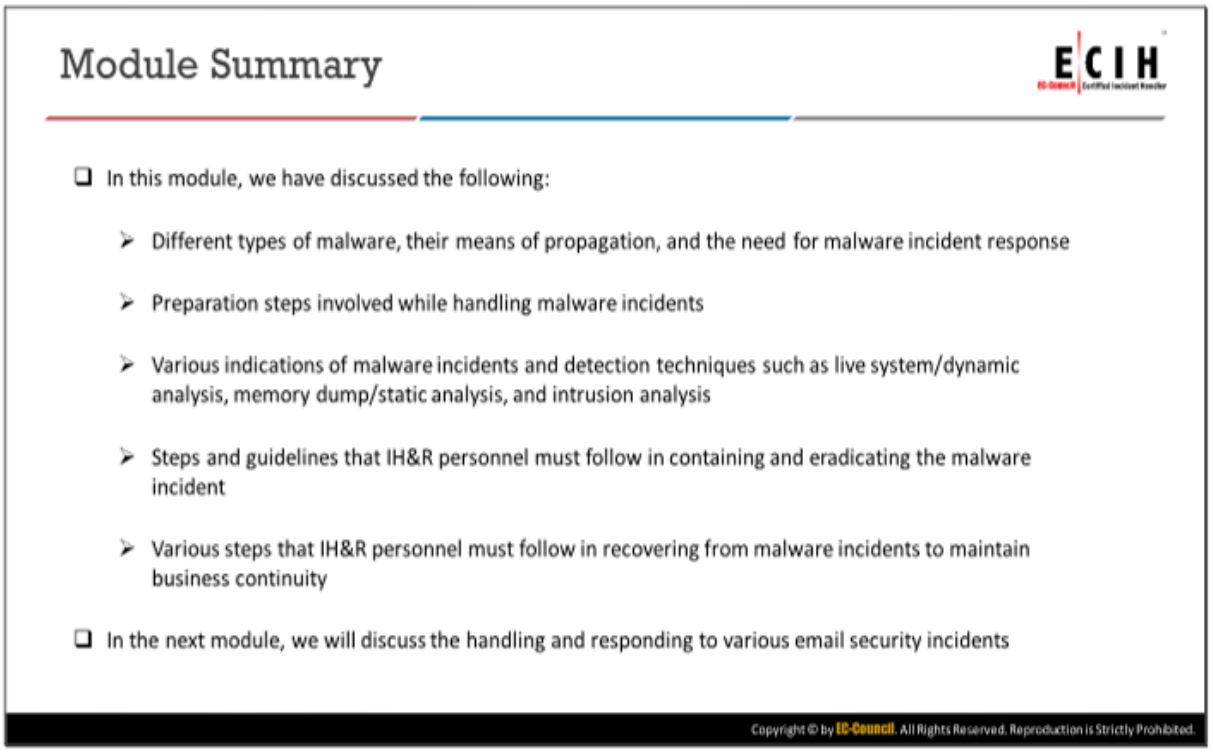
Guidelines for Preventing Malware Incidents

Following are recommended guidelines based on prevention, awareness, and vulnerability and threat mitigation for preventing malware security incidents. The guidelines apply to all users and information technology (IT) staff involved in preventing organizational malware infections.

Strictly following these guidelines is helpful for reducing the number of malware incidents that occur:

- IH&R teams must regularly review and update computer security and malware prevention policies as per reports and outcomes of recent malware incidents.
- IH&R teams must inform and educate users, clients, stakeholders, and employees about recent malware attacks and specify actions that can prevent such future incidents.
- IH&R teams should subscribe to and regularly follow antivirus bulletins to prevent and/or smoothly handle future malware attacks.
- According to outcomes of recent malware incidents, organizations may review procedures and points of contact for reporting future malware incidents.
- IH&R teams should ensure that organizations follow effective data backup and recovery processes.
- Organizations should educate employees about how to safely handle email attachments.
- IH&R teams should deploy network-based IDSs/IPSs and firewall systems to receive timely alerts about any intrusion attempts into organizational networks.
- Organizations should identify critical hosts and install host-based IDSs on them to monitor traffic flow and detect anomalies.
- IH&R teams should ensure that all organizational devices use the latest versions of antivirus software.
- Incident responders may configure malware-signature-based antivirus applications to block opening or execution of suspicious files.
- Organizations should provide strict guidelines to employees and users to check all downloaded internet-based files and attachments for malware.
- Organizations also should restrict use of removable devices on networks/systems unless absolutely necessary. In that case, organizations should ensure that employees use clean storage devices and scan all removable media such as USB drives and diskettes for malware before use on organizational networks/systems.
- Incident responders should check whether all hosts have updated firewalls and antivirus software signatures/definitions that can block spyware installation.
- Incident responders should regularly check and block suspicious ports and kill unnecessary processes to prevent possible malware attacks.
- Incident responders should configure organizational network share settings to prevent unnecessary window shares.
- Organizations should regularly review and update IH&R processes and facilities to respond to malware incidents as quickly as possible with minimal losses.

- Incident responders should regularly check and scan all organizational systems and remove all suspicious files.
- All organizational employees and network users should have email filters that can detect and filter out spam.
- Incident responders should limit use of unnecessary file transfer protocol (FTP) programs and eliminate sources and causes of unwanted network traffic.
- All network users should activate and use web browser security features such as disabling JavaScript, enabling popup blocking, and configuring and customizing security settings as per best practices to avoid malware infections.
- IH&R teams should implement strict security features on all organizational emails and ensure that emails are securely encrypted and transmitted.
- Organizations should use secure email clients including features such as digital signatures, pretty good privacy (PGP) encryption, and automatic attachment scanning.
- IH&R teams should regularly check and patch all organizational systems and applications.
- IH&R teams should regularly check whether all the organizational hosts have proper hardening measures such as patching, configuration management, and security partitioning to limit possible malware attacks.



Module Summary

In this module, we have discussed different types of malware, their means of propagation, and their impacts on networks/systems and organizations. We also have discussed the need for malware IR, preparation for handling malware incidents, and various malware-incident indicators and detection techniques such as live-system/dynamic, memory-dump/static, and intrusion analyses. Furthermore, we discussed steps and guidelines that IH&R personnel must follow to contain, eradicate, and recover from malware incidents while maintaining business continuity.

In the next module, we will discuss in detail handling and responding to various email security incidents.