8.4.10 Hide Programs Facts

Hackers use a variety of techniques to hide the programs and data they have used.

This lesson covers the following topics:

- Rootkits
- NTFS data streaming
- Steganography

Rootkits

A rootkit is a software program that hackers use to establish root- or admin-level privileges to a system. Rootkits are a set of programs designed to covertly access a system and allow the hacker to control its functions. Using a rootkit, a hacker can hide added applications and processes, obtain sensitive data, and set up the system to act as a server for bot updates.

Rootkits can modify the operating system and the utilities of the target system. Rootkits contain packet sniffers, utilities that remove logs, DDoS programs, IRC bots, and backdoor programs. The following table describes two tools to create rootkits:

Tool	Description
GrayFish	A rootkit tool that runs within the Windows operating system. It contains hidden storage and has invisible command execution. GrayFish isn't flagged in anti-rootkit scans because it sets no hooks on Window kernel functions and doesn't register callback functions.
Sirefef	Sirefef, also known as ZeroAccess, has virus, Trojan horse, and rootkit components. As a rootkit, it is unseen by antivirus and anti-spyware programs. It hides by changing the internal process of the target operating system. Sirefef is difficult to remove and can create problems with Windows Firewall and Defender Service, remote hosts, and browser settings. It creates a folder to store additional malware.

Several methods used to detect and identify rootkits include:

Detection Method	Description
Integrity-based detection	Integrity-based detection works by running a tool to scan a clean system to create a database. The integrity-based detection scans the system and compares the current scan to the clean database. Any dissimilarities between the clean baseline database and the current scan are flagged and a notification is sent.
Signature-based detection	Signature-based detection scans a system's processes and executable files looking for byte sequences of known malicious rootkit programs.
Heuristic or behavior-based detection	Heuristic or behavior-based detection searches for deviations in normal behaviors and patterns of an operating system. One of the patterns it searches for is execution path hooking which allows a function value in an accessible environment to be changed. This is a behavior used by rootkits.
Runtime execution path profiling	Runtime execution path profiling checks for variations in the runtime execution path of all executable files and system processes.
Cross view-based detection	Cross view-based detection uses an algorithm as it goes through the system files, processes, and registry keys to create a baseline that is compared to the data returned by the operating system's APIs.

To prevent rootkits, take the following steps:

- Back up critical data and reinstall the OS and applications.
- Install and routinely update firewalls.
- Patch and regularly update the OS and applications.
- Keep a record of automated installation procedures.
- Harden servers and network stations.
- Train users to confirm that downloads are from a trusted source.
- Check for rootkits through a kernel memory dump analysis.

NTFS Data Streaming

Another way that hackers can hide programs is through NTFS alternate data streams (ADS). When a file is created or copied to NTFS, one data stream stores the attributes, and a second stores the data. NTFS allows each file an unlimited number of data streams with unlimited size. Because they are hidden, a hacker can inject malicious code into these alternate data streams and execute the code without being detected by the user or system administrator.

To get rid of malicious alternate data streams, move suspect files to a partition or device that is formatted using FAT. Since FAT doesn't support alternate data streams, the alternate file streams will be removed when the file is moved. Remember to keep your antivirus software updated. Some tools that detect and remove infected ADS include LADS, Stream Detector, LNS, and Forensic Toolkit.

Steganography

Steganography is the method of embedding data into legitimate files like graphics, banner ads, or plain text messages to hide it and then extracting the data once it reaches its destination. It is very difficult to detect and has become a very popular method for hackers. Steganography can hide identities, communication, code, and content. Hackers can use steganography as an alternative to encryption because data hidden in steganography doesn't have to be encrypted. However, encrypted steganographic information is even more difficult to decipher.

The following table describes several types of steganography.

Steganography Type	Description
Image steganography	The most common form of steganography is hiding information in image files.
Video steganography	Files with extensions can be hidden in video files such as .MPG4, .AVI, and .WMV.
Document or whitespace steganography	The data is hidden in added white spaces and tabs at the end of lines.
Audio steganography	The data is hidden in a digital sound format through least signification bit (LSB) manipulation.
Web steganography	The data is hidden behind a web object when uploaded to the server.
C++ source code steganography	A set of tools is hidden in the C++ code.
Spam/email steganography	Data is embedded in an email.

The following table lists tools used to create steganography.

Tool	Description
StegoStick	A steganography tool that allows a file to be hidden within any image, audio, or video file, even in PDFs and EXE files.
OpenStego	A tool for hiding data in a cover file or watermarked files. It can be used to trace file copying.
OmniHide Pro	OmniHide Pro can hide files in photos, movies, documents, and music. It allows the user to create a password to make the hidden file more secure.
DeepSound	A tool for hiding data in audio files and extracting files from audio tracks. It also has the option to encrypt the files.
Spam Mimic	Spam Mimic encodes data into emails and has the ability to decode the messages.

While it is difficult to detect steganography, there are some actions you can take. The table below identifies where to look for steganography files.

Steganography Type	Description
Text	Check for extra spaces and invisible characters. Look for unusual patterns in spacing, fonts, line heights, and even in the language.
Image	Check for changes in format, size, the color palette, and the last modified timestamp.
Audio	Look for distortions and patterns in frequencies that are above or below the human range of hearing.
Video	Use a combination of the methods used for audio and image files to search for hidden information.

The following tools aid in steganography detection:

Steganography Detection Tool	Description
Discover the Hidden	Scans for known steganography and encryption programs.
StegoHunt	Searches for carrier files through statistical analysis techniques, scans for data hiding tools;,and can crack password-protected data to extract the payload.
Gargoyle	Scans for known steganography files created by tools such as BlindSide, S-tool, and WeavWav.
StegAlyzerSS	Scans media or forensic images for uniquely identifiable byte patterns or known signatures left inside files when a steganography application is used to embed hidden information in them.
Virtual Steganographic Laboratory (VSL)	Uses, tests, and adjusts different steganographic techniques in a simple GUI. VSL is free image steganography and steganalysis software.
Stegdetect	Detects steganographic content in images.