# 9.1.8 Malware Analysis Facts

One of a penetration tester's roles is to understand malware and how it operates. The penetration tester needs to know how the malware works in order to utilize it in testing and make recommendations for combating threats.

Malware analysis is the process of reverse-engineering a specific piece of malware in order to determine its origin, functionality, and potential impact. There are two methods of malware analysis, static and dynamic.

This lesson covers the following topics:

- Sheep dipping
- Static analysis
- Dynamic analysis

## Sheep Dipping

The process of analyzing emails, suspect files, and systems for malware is known as sheep dipping. The term comes from the process sheep farmers use to dip sheep in chemical solutions to clear them of parasites. A special computer that is used for analysis is called a sheep dip computer. This computer is isolated from all networks, and it has port monitors, file monitors, network monitors, and anti-virus software. This system connects to a network only under extremely strict conditions. Along with the sheep dip computer, an anti-virus sensor system is used. This is a collection of software that detects and analyzes malware.

## Static Analysis

Static analysis is also known as code analysis. This involves going through the actual code of the malware without executing it. This is done using a variety of tools and techniques in order to understand the malware's function and purpose. Because the malware itself is not being run, this method is relatively safe. There are several static analysis techniques:

| Technique | Description |
|---|---|
| File fingerprinting | File fingerprinting is the process of identifying unique malware programs through generating a hash for the program. This hash can be checked throughout the analyzing process to see if it has changed. MD5 or SHA1 are the two most common hash functions used in file fingerprinting. File fingerprinting does not work well with encrypted files, password-secured files, or media files. |
| Scanning | This process involves scanning the malware with a local anti-malware program or using an online scanner. |
| String searching | When the malware's code is not obfuscated, the analyzer can search for strings of plain text in the code. These strings may show the malware's purpose and some of its functions. |
| Identify obfuscation/packing | Hackers use obfuscation techniques and packers to compress and encrypt their malware. Part of the analyzing process is to determine the method that was used. If the method is determined, the analyzer should be able to unpack the code without damaging or changing it, which allows for deeper analysis. |
| Malware disassembly | Disassembling the malware allows the analyzer to learn everything about the program and what it's designed to do. Loading the program into a disassembler or debugging program will generate the raw code, which can be analyzed to determine everything about the malware. |

## Dynamic Analysis

Dynamic analysis is the process of analyzing the malware by running it and observing how it behaves and its effects on the system. This type of analysis can be done only on the sheep dip computer.

The first step in this process is to create a system baseline. System baselining refers to the process of creating a snapshot of the system before the malware is run. This allows the analyzer to determine the changes the malware has on the system.

The process of studying the malware and its effects is known as Host Integrity Monitoring. This involves using the same tools and processes to take a snapshot of the system before and after the malware is executed. Host Integrity Monitoring includes monitoring many components, as explained in the following table.

| Technique | Description |
|---|---|
| Ports | Malware often opens ports on the computers. Using tools such as Netstat will show any open ports the malware is using. |
| Processes | Malware can hide itself by posing as genuine Windows services or processes. Using a tool like Process Monitor can help determine if any processes are actually malware. |

| | |
|---|---|
| Registry | Monitoring the registry for any changes by the malware is important, as malware will often create registry keys. Scanning the registry for suspicious keys can aid in tracking the malware infection. |
| Windows services | Malware can spawn additional Windows services or rename malicious processes to look like a Windows service and evade detection. Windows Service Manager can detect changes in services and can also scan for suspicious Windows services. |
| Startup programs | Malware can set itself to load with Windows in startup programs. Verifying the startup programs can be done manually or with a tool like WinPatrol or Autoruns. |
| Event logs | Event logs should be analyzed to identify malicious or suspicious activities. |
| Installation | When software is installed or uninstalled, traces of the application data can be left on the system. You can install monitor programs such as SysAnalyzer to help track anything being installed or uninstalled. |
| Files and folders | Malware will normally modify a system's files and folders. Use file and folder integrity checkers such as Tripwire or SigVerif, which is the built-in Windows file verifier. |
| Device drivers | Malware can hide itself inside untrusted or invalid device drivers. Verify that device drivers are valid and trusted. |
| Network traffic | Most malware will generate network traffic. Analyzing network traffic with programs like Wireshark will help you see what the malware is doing and track it down. |
| DNS | Some malware is capable of changing a system's DNS information. DNSQuerySniffer, DNSstuff, and similar programs can monitor DNS requests and settings of the system. The analyzer should use these tools to monitor DNS requests and identify whether the malware can change those settings. |
| Application Program Interface (API) calls | APIs are parts of the Windows OS that allow external applications to access OS information such as file systems, threads, and errors. A program like API Monitor can help the analyzer see how the malware is interacting with the operating system. |

TestOut Corporation All rights reserved.