Chapter 1 Study Guide: Basic Static Techniques

Practical Malware Analysis – Michael Sikorski & Andrew Honig

Overview

Basic static analysis is the process of examining a malware executable without running it. The primary goal is to safely collect information about the file, determine whether it is malicious, and identify indicators of compromise (IOCs). Static analysis is typically the first step in malware triage.

Antivirus Scanning

Antivirus scanning is a useful first step for identifying known malware. AV engines rely on signatures, heuristics, and behavioral patterns. Because malware authors frequently modify their code, AV results should be treated as informational rather than definitive. Using multiple AV engines provides better coverage.

Hashing Malware Samples

Hashing produces a unique fingerprint for a file using algorithms such as MD5 or SHA-1. Hashes allow analysts to label samples, share them with other analysts, and search online databases to determine if malware has been previously identified. Any change to the file results in a different hash.

Strings Analysis

Strings are human-readable sequences embedded within a program. Analyzing strings can reveal URLs, IP addresses, file paths, registry keys, error messages, and Windows API function names. Analysts commonly extract both ASCII and Unicode strings. Not all extracted strings are meaningful, so analysts must filter irrelevant data.

Packed and Obfuscated Malware

Packing and obfuscation are techniques used to hide malware functionality and evade detection. Indicators of packing include few readable strings, abnormal section names, and high entropy. Packed malware significantly limits the effectiveness of basic static analysis and often requires dynamic or advanced analysis.

Portable Executable (PE) File Format

Windows executables use the Portable Executable (PE) format. Examining PE headers and sections provides insight into how malware interacts with the operating system. Analysts review imports, exports, section names, and metadata to infer behavior.

Imported and Exported Functions

Imported functions reveal which Windows APIs the malware relies on. These functions often indicate malware intent, such as networking, persistence, or file manipulation. Exported functions show functionality the malware exposes to other programs.

Resources Section Analysis

The resources section of a PE file may contain icons, configuration data, strings, or embedded payloads. Malware frequently hides secondary components or configuration data in this section.

Static Analysis in Practice

Unpacked malware is generally easier to analyze statically, while packed malware often prevents analysts from gathering meaningful information. Static analysis is most effective when combined with dynamic analysis.

Key Takeaways

- Static analysis is fast and safe but limited
- Antivirus results provide clues, not conclusions
- Hashes uniquely identify malware samples
- Strings frequently reveal intent and infrastructure
- Packed malware requires advanced techniques