Reflective Code Injection Attacks - An Overview for Incident Responders

Taz Wake | Halkyn Consulting Ltd | Originally published 2023-11-16

Introduction to Reflective Code Injection

Reflective code injection is a sophisticated technique used by attackers to execute arbitrary code within the memory space of a process without requiring external modules or files. This type of attack is particularly stealthy as it leaves minimal traces on the file system, making detection challenging.

Reflective Code Injection in Windows

Overview: In Windows, reflective code injection involves injecting code into a legitimate process's memory space. This is often achieved through APIs like VirtualAllocEx and WriteProcessMemory, followed by execution control transfer using mechanisms such as CreateRemoteThread.

Execution Flow: The injected code, typically a DLL, is loaded directly from memory, bypassing the standard Windows loader. This is achieved by manually performing tasks usually handled by the loader, such as handling relocations, resolving imports, and executing TLS callbacks.

Stealth and Evasion: This technique is favoured for evading detection tools that monitor file-based module loading activities, as the injected code does not appear in the standard list of loaded modules of the process.

Reflective Code Injection in Linux

Overview: In Linux, reflective code injection can be executed using techniques like process memory manipulation via ptrace or manipulating memory directly through /proc/[pid]/mem.

Execution Flow: The injected code often involves shellcode or ELF-formatted payloads. Unlike Windows, Linux does not have a unified loader mechanism, thus the injected code must be more self-contained, handling its own dependencies.

Comparison with Windows: Unlike Windows, Linux's diverse range of process manipulation capabilities offers a broader attack surface for reflective code injection, but the lack of a standardized injection mechanism makes the attacks more complex and varied.

Detection Techniques:

Memory Analysis

Memory Analysis in Windows: In Windows, tools like Process Hacker or the Windows Debugger (WinDbg) can be used to inspect the Virtual Address Descriptor (VAD) tree of a process. Unusual or unlinked memory regions, especially those with execute permissions, may indicate injected code. You can also use the incredibly effective MemProcFS. This is probably one of the most exciting developments in memory analysis in the last 5 years.

Memory Analysis in Linux: In Linux, the /proc filesystem is a key resource. By examining /proc/[pid]/maps and /proc/[pid]/mem, one can identify anomalies in memory mappings and contents.

Detecting Reflective Code Injection

Anomalies in Memory Regions: Unusual memory regions, such as those with RWX (read, write, execute) permissions, should be treated with suspicion. In both Windows and Linux, these regions could indicate the presence of injected code.

Discrepancies in Process Behavior: Monitoring for discrepancies in process behaviour, such as a text editor executing network operations, can be a sign of code injection.

Use of Uncommon System Calls: In Linux, an increase in the use of system calls related to memory manipulation (like mmap, mprotect, or ptrace) can be a red flag. Similarly, in Windows, frequent calls to memory manipulation APIs outside of normal operation context may indicate injection.

Incident Response and Threat Hunting

Hunting for Reflective Code Injection: Incident responders and threat hunters should monitor for signs of anomalous memory and process behaviour. Tools like Volatility for memory forensics, or YARA rules to identify known patterns of injection, can be effective. In Linux, analyzing the /proc filesystem for each process can reveal inconsistencies indicative of injection.

Summary

Reflective code injection remains a potent threat in both Windows and Linux environments. Understanding the mechanisms of these attacks and employing thorough memory analysis techniques are crucial for effective detection and response. As attackers evolve their methods, continuous education and adaptation of detection strategies remain essential for cybersecurity professionals.

About Halkyn Consulting

Halkyn Consulting Ltd is a UK-based security and risk management consultancy providing 24/7 incident response, digital forensics, threat hunting, virtual security leadership (vCISO) and security compliance support to organisations worldwide. We deliver clear, practical, vendor-independent advice based on real-world threats — no hype, no upsell.

More free security resources are available at www.halkynconsulting.co.uk/security-resources.

Web: www.halkynconsulting.co.uk | **Email:** info@halkynconsulting.co.uk | **Tel:** +44 1244 940858