Project Proposal

"Evading Signature-Based Antivirus Software Using Custom Reverse Shell Exploit"

Russell Sullivan, Harish Subburaj, Nick Oneal, Johnathan Carrizales, Sebastian Bejarano

Overview:

The paper discusses the "primary" line of defense against malicious software, and key exploitations to such software. Antivirus software is determined to be the primary defense against malicious code, where the software uses signature-based detection algorithms in order to detect malicious code. The authors suggest that being able to get past such signature detection would be key to breaking through the antivirus software, thus leading them to introduce "source code obfuscation of a typical Metasploit reverse shellcode exploitation". In other words, simply taking the source code of a typical reverse shellcode, a type of malware that establishes connection from the victim's machine to the attacker's machine and modifying it in a way that avoids detection by the signature-based engine. Although typical reverse shellcode is extremely easy to detect, the paper proposes a program obfuscation that involved finding a way to achieve the same result as the original program while also changing the programs signature, where the authors solution was to add another stager to the program therefore instead of the shellcode being generated and stored within the program it is generated separately and stored on a remote server. Furthermore the paper goes on to describe implementation details such as system, network, and remote code execution. Finally it was concluded that the implementations of the authors design resulted in reduced detection by 97% in comparison to the typical reverse shell program therefore changing the reverse shells program signature substantially impacts antivirus softwares

ability to detect such malicious exploits where the alarming point is made that there are many other ways to obfuscate malicious code to bypass antivirus.

Intellectual Merit:

Signature-based antivirus uses an algorithm to compare code to a database of known malicious signatures. This method of antivirus is typically very accurate and quick. Its only flaw is new zero-day attacks because of its reliance on pre-existing exploits.

About 95% of data breaches start with a phishing email which an employee of a company opens. Unfortunately, it only takes one click to mistakenly compromise important data. The key takeaway from "Evading Signature-Based Antivirus Software Using Custom Reverse Shell Exploit" is the exposure of an exploit that uses Windows 64-bit operating system to successfully bypass almost all antivirus signature-based detection. This bypass specifically uses Metasploit reverse shellcode exploitation as well as Msfvenom within Metasploit to generate different types of shellcode exploits for multiple operating systems.

Task 2: Proposed Work and Methodology

The specific topic I will investigate is code obfuscation, as described in the paper's method of modifying the reverse shell's source code to evade antivirus detection. Code obfuscation, as outlined in the paper, allows for subtle modifications in the reverse shell's source code that doesn't alter functionality but makes detection much more difficult for signature-based antivirus systems. It also closely ties to our class discussions on cybersecurity vulnerabilities and malware

evasion techniques. The concept of altering the signature of the reverse shell to bypass signature-based detection is relevant to our studies on attack methods and defenses. This technique can be applied to areas such as penetration testing, malware analysis, and improving antivirus detection mechanisms. The broader application of this technique extends to fields like penetration testing and malware analysis, where stealth is critical, and evading detection is a key goal.

Task 3: Demonstration Software

For the final presentation, we will demonstrate the modified reverse shell exploit from the paper, focusing on how code obfuscation (C.O.) reduces detectability by signature-based antivirus systems. Code obfuscation involves altering the shellcode's signature without changing its functionality, making it less recognizable to antivirus software. The key idea is to host the shellcode on a remote server and fetch it during execution, so it never touches the disk, further minimizing detection. By applying these obfuscation techniques, we will compare the detectability of the reverse shell before and after obfuscation, highlighting how the change in signature effectively bypasses antivirus systems. This presentation will emphasize the role of code obfuscation in making malware stealthier.