Reflective Loading Runs Netwalker Fileless Ransomware

blog.trendmicro.com/trendlabs-security-intelligence/netwalker-fileless-ransomware-injected-via-reflective-loading/

May 18, 2020

Threat actors are continuously creating more sophisticated ways for malware to evade defenses. We have observed Netwalker ransomware attacks that involve malware that is not compiled, but written in PowerShell and executed directly in memory and without storing the actual ransomware binary into the disk. This makes this ransomware variant a fileless threat, enabling it to maintain persistence and evade detection by abusing tools that are already in the system to initiate attacks.

This type of threat leverages a technique called reflective dynamic-link library (DLL) injection, also referred to as reflective DLL loading. The technique allows the injection of a DLL from memory rather than from disk. This technique is stealthier than regular DLL injection because aside from not needing the actual DLL file on disk, it also does not need any windows loader for it to be injected. This eliminates the need for registering the DLL as a loaded module of a process, and allowing evasion from DLL load monitoring tools. Recently, we have witnessed threat actors using this technique to deploy ColdLock ransomware. Now, we have seen the same attack using a filelessly executed Netwalker ransomware. The payload begins with a PowerShell script detected as Ransom.PS1.NETWALKER.B.

Analysis of the PowerShell Script

Figure 1. Overview of the PowerShell script's behavior

The script hides under multiple layers of encryption, obfuscation, and encoding techniques. For this sample, we were able to reveal three layers of code. The top-most layer executes a base64-encoded command.

Figure 2. Code snippet of top-most layer of code (base64 encoded command)

Decoding this will expose the next layer of code, which is hexadecimal-encoded and XOR-encrypted.

Figure 3. Second layer of code (hexadecimal-encoded and XOR-encrypted)

Decoding and decrypting will then reveal the main script, which is still quite obfuscated, making the content more difficult for analysts to decipher.

Figure 4. Code snippet of the obfuscated main script

The file reflectively injects a ransomware DLL into the memory of the legitimate running process explorer.exe. The ransomware is embedded in the script in hex format.

Figure 5. Ransomware binaries embedded in the script in hex format

Taking the binaries out of the script and decoding them will result in two DLLs; one is an x86 version (for 32 bit OS) of the ransomware, while the other is the x64 version (for 64 bit OS).

It uses the following part of the script to determine the environment it is running on so that it can set the DLL version to use:

Figure 6. Script that determines what environment the ransomware is running on

To successfully perform reflective injection, it first locates the API addresses of the functions it needs from kernell32.dll:

Figure 7. Ransomware collecting API Addresses from kernell32.dll

Then it uses the following functions to set up accurate memory address calculations:

Figure 8. Functions for setting up memory calculations

Figure 9. Code snippet for computing the needed memory addresses

In this manner, the script itself acts as the DLL's own custom loader. This eliminates the need for a traditional windows loader, which usually makes use of the LoadLibrary function. The script itself can compute and resolve its needed memory address and relocations to load the DLL correctly.

It then specifies the process it will inject into; in this case it searches for the running Windows Explorer process.

Figure 10. Code snippet for searching running Windows Explorer process

Afterwards, it will write and execute the ransomware DLL into the memory space of explorer.exe through the following code:

Figure 11. Code snippet of writing the ransomware DLL code into memory

Finally, it deletes Shadow Volume Copies and prevent the victim from using Shadow Volumes to recover their encrypted files.

Figure 12. Code snippet for deleting Shadow Volume Copies

This sample appears to have been derived from PowerSploit's Invoke-Mimikatz module, an open source program that was originally intended to reflectively load Mimikatz completely in memory for stealthy credential dumping.

Analysis of the Fileless Ransomware

This variant of Netwalker is similar to its predecessors in terms of behavior. It renames encrypted files using 6 random characters as extension:

Figure 13. Encrypted files renamed using 6 random characters as extension

It drops ransom notes at various folders in the system and opens one after it has encrypted the data and documents of the victim. As with usual ransomware, it does this to extort money from the victim in exchange for the decryption of their files.

Figure 14. Netwalker ransom note

It adds the following registry entry. Adding this entry is a known behavior of Netwalker:

> HKEY_CURRENT_USER\SOFTWARE\{8 random characters}

> {8 random characters} = {Hex values}

Figure 15. Sample registry entry added by Netwalker

The ransomware terminates some processes and services, some examples of which are related to backup software and data related applications. It is likely that it does this as an attempt to debilitate any efforts the victim may take in performing backup and recovery operations after the ransomware attack.

Below are some examples of services terminated by the ransomware (for the full list of services, please see this report):

- *backup*
- *sql*
- AcronisAgent
- ARSM
- server Administrator
- ShadowProtectSvc
- wbengine

The ransomware also stops security software-related processes to evade detection and termination of its malicious activities.

Additionally, it also terminates processes relating to user data and documents, as well as software for creating backups. Then it will proceed to encrypt files created through those applications.

Example processes terminated by the ransomware (for the full list of processes, please see this report):

- *sql*
- excel.exe
- ntrtscan.exe
- powerpnt.exe
- wbengine*
- winword.exe
- wrsa.exe

Netwalker mainly targets common user files during its encryption routine, such as Office documents, PDFs, images, videos, audio, and text files, among others. Other than that, it apparently doesn't want to render the system completely useless since it generally avoids encrypting any critical files, executables, Dynamic Link Libraries, registries, or other system-related files.

Conclusions and Recommendations

It appears that attackers are now adding Reflective DLL injection into their ransomware arsenal in an attempt to make their attacks untraceable and more difficult to investigate by security analysts. Ransomware in itself poses a formidable threat for organizations. As a fileless threat, the risk is increased as it can more effectively evade detection and maintain persistence. Blended threats such as this make use of multiple techniques, making it necessary for organizations to use various layers of security technologies to effectively protect their endpoints, such as security solutions that employ behavior monitoring and behavior-based detections.

These types of attacks can affect victims tremendously, and they can be painstakingly difficult to recover from. Employing adequate preventive measures, such as applying best practices, will greatly minimize the risk of infection. Here are some of our recommendations to help avoid ransomware attacks:

- Regularly back up critical data to mitigate the effects of a ransomware attack
- Apply the latest software patches from OS and third-party vendors.
- Exercise good email and website safety practices
- For employees, alert the IT security team of potentially suspicious emails and files.
- Implement application whitelisting on your endpoints to block all unknown and unwanted applications.
- Regularly educate employees on the dangers of social engineering.

Below are some of our recommendations to protect systems from fileless threats:

- Secure PowerShell use by taking advantage of its logging capability to monitor suspicious behavior.
- Use PowerShell commands such as ConstrainedLanguageMode to secure systems from malicious code.
- Configure system components and disable unused and outdated ones to block possible entry points.
- Never download and execute files from unfamiliar sources

We also recommend security solutions that utilize behavior monitoring that can work against these types of threats:

> Trend Micro Apex One™ - Features behavioral analysis that protects against malicious scripts, injection, ransomware, and memory and browser attacks related to fileless threats.

Indicator of Compromise

SHA-256	Trend Micro Pattern Detection
f4656a9af30e98ed2103194f798fa00fd1686618e3e62fba6b15c9959135b7be	Ransom.PS1.NETWALKER.B

Ransomware

Ransomware in itself poses a formidable threat for organizations. As a fileless threat, the risk is increased as it can more effectively evade detection. We discuss how Netwalker ransomware is deployed filelessly through reflective DLL injection.

By: Karen Victor May 18, 2020 Read time: (words)

Content added to Folio