Netwalker

Fileless Ransomware

June 5th, 2020

ATLAS CYBERSECURITY

www.atlas-cybersecurity.com

The Attack

- Well-known Netwalker ransomware with a twist
  - Utilizing ‘reflective dynamic-link library injection,’ this variant of the ransomware is written in PowerShell and executed directly in memory, without storing the actual ransomware binary into the disk.
  - As a result, this technique is stealthier than regular DLL injection because not only does it not need 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.
- The Netwalker ransomware is your typical ransomware attack, which includes:
  - It encrypts and renames file using six characters as extensions
  - it drops ransom notes at various folders in the system and opens one after it has encrypted the data and documents of its victims
  - it terminates processes and services related to backup software and data related applications in an attempt to maintain persistence and in an effort to prevent the victim from restoring from backups

Analyzing the PowerShell Script

- The script hides under multiple layers of encryption, obfuscation, and encoding techniques; in one example, Trend Micro was able to reveal three layers of code
  - The top-most layer executes a base64-encoded command
    - Decoding this will expose the next layer of code, which is hexadecimal-encoded and XOR-encrypted
      - Decoding and decrypting this will then reveal the main script. The file reflectively injects a ransomware DLL into the memory of the legitimate running process 'explorer.exe.' The ransomware is then embedded in the script in hex format
      - The script contains two DLLs, of which one will be utilized depending on the OS architecture of the victim system
- In order to successfully perform reflective injection, it first locates the API addresses of the functions it needs from kernel32.dll and then uses a series of functions to set up accurate memory address calculations.
- With all the previous actions taken together, this allows the script to act as the DLL's custom loader, thus eliminating the need for a traditional Windows loader
- The script itself can compute and resolve its needed memory address and relations to load the DLL correctly; it then specifies the process it will inject into
  - In a Trend Micro case study, the script injected itself into the Windows Explorer process (explorer.exe)

Prevention and Remediation

- It is becoming increasingly common for threat actors to incorporate Reflective DLL Injection into their attacks, in an attempt to make their attacks untraceable and more difficult to investigate
  - While ransomware itself poses a formidable threat, fileless attacks increase the threat by making the malware more effective at evading detection and maintaining persistence
- Some prevention actions available to users and network administrators to combat the newly emerging fileless threat are:
  - 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
  - And, as always, never download and execute files from unfamiliar sources