Defeating Ransomware Thru Vulnerability Exploitation

By Malvuln - John Page (aka hyp3rlinx)
www.malvuln.com, malvuln13@gmail.com, apparitionsec@gmail.com

Intro:

The malvuln project started Jan 1, 2021 and currently catalogs approximately 666 vulnerable pieces of malware to date. Witnessing the endless ransomware attacks in the news and as I had no ransom trophy kills, I figured I would take a crack at it. At first, ransomware seemed like a big challenge. I am not aware of any sample that listens on any ports that can be abused and local elevation of privilege techniques mean nothing when things get encrypted immediately.

But Wait.. In steps DLL hijacking.

In May 2022, I publicly disclosed a novel strategy that successfully defeated ransomware. Using a well known attacker technique (DLL hijack) to terminate malware pre-encryption. The first malware to be successfully exploited was from the group Lockbit MVID-2022-0572. Followed by Conti, REvil, BlackBasta and CryptoLocker proving many are vulnerable.

DLL hijacking is a method of injecting malicious code into an application by exploiting the way some Microsoft Windows applications search and load Dynamic Link Libraries (DLLs). Only Microsoft operating systems are susceptible to DLL hijacking.

This coding flaw with DLL searching has plagued many pieces of legitimate software for years. If a program is run and side loads an arbitrary DLL it will execute that code in the parent process. During my initial research I found this flaw to be a common mistake made by ransomware authors. Thereafter, I wrote the first ever exploit DLL as a proof-of-concept to successfully mitigate a ransomware attack in the C programming language.

Implementing A Kill Switch:

Not all methodologies are the same, but implementing a kill switch is possible for all pieces of ransomware that suffer from this issue. The code for implementing a kill switch primarily uses Win32 API calls GetCurrentDirectory, OpenProcess and TerminateProcess. Time was spent analyzing and running dozens of ransomware in a virtual machine using the sysinternals "Process Monitor" utility and monitoring for the "NAME NOT FOUND" result. This is an indicator identifying the DLL file being sought by the ransomware.

For example, "Conti Ransom" wants to load "netapi32.dll" when it spawns. Therefore, we can craft a trojan DLL file to call GetCurrentDirectory to return the current directory. Next, we compare the return value from

GetCurrentDirectory with the hardcoded string "C:\Windows\System32" using the standard "strcmp" C language string library function.

If strcmp function returns a non zero value, we know the malware is looking in its own directory and not the legitimate "System32" directory which is normally where "netapi32.dll" lives. Based on that condition, we make the decision to call the WIN32 API OpenProcess() function to get a handle to our own process ID (PID) and terminate. This effectively terminates Ransomware pre-encryption before damage can be done.

In the case of "BlackBasta.Ransom", where it looks for "wow64log.dll" in the "C:\Windows\System32" directory, we compile and copy "wow64log.dll" there and simply call exit() as a System32 check was not required. Moreover, for the wow64log.dll DLL there was a need to export the "WINBASEAPI LONG WINAPI InterlockedExchange" function required by the DLL.

BlackBasta was an unusual case and only ransomware to be exploited where the System32 directory was used for exploit DLL placement. However, copying a DLL there is undesirable as it can cause other programs to fail from starting. Therefore, more research is required for those scenarios.

Code Snippet:

```
BOOL APIENTRY DllMain(HINSTANCE hInst, DWORD reason, LPVOID reserved){
 switch (reason) {
 case DLL_PROCESS_ATTACH:
  TCHAR buf[MAX_PATH];
  if(GetCurrentDirectory(MAX_PATH, buf))
    if(strcmp("C:\\Windows\\System32", buf) != 0){
          HANDLE handle = OpenProcess(PROCESS_TERMINATE, FALSE, getpid());
   if (NULL != handle) {
     TerminateProcess(handle, 0);
     CloseHandle(handle);
```

Exploits To Defend The Network!

Trojan DLLs written to intercept ransomware can be placed in directories that attackers are known to abuse. E.g. C:\Users\Public, C:\Windows\Temp , C:\PerfLogs etc. The existence of these files can mitigate many of the current ransomware strains circulating. DLLs used to mitigate attack can be set as hidden system files using Windows CL attrib +s +h command. In domain environments this can be deployed via GPO. It is possible malware authors may start correcting these coding mistakes, but many historic strains can be stopped.

Related Work:

In July of 2023 I released RansomLord v1, it is a proof-of-concept tool that automates the creation of PE files, used to compromise ransomware pre-encryption and written in the C programming language. RansomLord enables an end user to generate the correct DLL required to defeat a particular Ransomware. DLLs are either x32 or x64 bit and currently total 12. These DLL files can currently defeat 33 different threat groups. Clop, Play, Lockbit, Darkside, Royal, BlackCat (Alphv), Yanluowang, DarkSide, Nokoyawa etc...

There is also a detection option that sets up a custom Windows Event source in the Windows registry for IOC alerts. Events are written to 'Windows Logs\Application' as 'RansomLord' with event ID 1 (Process creation). Malware name and process path are also included in the Event log.

https://github.com/malvuln/RansomLord (v1)
SHA256: b0dfa2377d7100949de276660118bbf21fa4e56a4a196db15f5fb344a5da33ee

Currently, I am working on updates to RansomLord and will release v2, which will bring the total number of ransomware families defeated to 43.

In Summary:

Endpoint security defenses focus on hash signatures, behavioral analysis, indicators of compromise (IOCs), and complicated hooking. However, this simple method intercepts ransomware and acts as a built-in kill switch terminating it pre-encryption. The exploit DLLs just live on the disk waiting. All basic tests were conducted successfully in a virtual machine environment.

References:

https://web.archive.org/web/20220705173141/https://malvuln.com/advisory/38745539b71cf201bb502437f891d799_B.txt

https://web.archive.org/web/20230406020613/https://www.securityweek.com/vulnerabilities-allow-hijacking-most-ransomware-prevent-file-encryption/

https://web.archive.org/web/20230527085450/https://www.bleepingcomputer.com/news/security/conti-revil-lockbit-ransomware-bugs-exploited-to-block-encryption/

Malvuln copyright © Dec, 2023