CRYING IS FUTILE: SandBlast Forensic Analysis of WannaCry

By bferrite

Published: 2017-05-16 · Archived: 2026-04-19 02:05:44 UTC

Using the NSA exploit EternalBlue released by the Shadow Brokers, the WannaCry ransomware developers have added their names to malware lore. Given the number of institutions hit and the amount of media generated, it seemed appropriate to show what the ransomware actually does on a system through our SandBlast Agent Forensics product.

The WannaCry outbreak has been a good test case for the recently launched SandBlast Anti-Ransomware. AR and Forensics work together as part of our SandBlast Agent product. As we had expected, Anti-Ransomware was up to the task and has successfully blocked all WannaCry samples we've thrown at it, without requiring any signatures or updates.

For this report, we disabled Anti-Ransomware, Anti-Malware and Threat Emulation (all 3 catch the attack) so that we could see what the attack does when encrypting a system.

We let SandBlast Agent Forensics perform its automated analysis of the incident. The interactive forensics report is here:

http://freports.us.checkpoint.com/wannacryptor2_1/

We invite you to click and explore the analysis.

Figure 1. Forensics Overview Screen for WannaCrypt. Click to access the online report.

The report generated by forensics will reduce the time taken to determine, analyze and understand the impact of an incident from hours or days to mere minutes. Let us see how by following what the malware is doing.

The first executable in the infection chain is not shown in the report because it checks for a specific URL before continuing the attack (so called kill switch). Malware researchers have discovered most of the URLs in the different samples of the attack and so the ransomware component is not created and executed.

We start the analysis from the actual ransomware executable itself as shown in Figure 2. Having analyzed multiple samples of the ransomware, we noticed that the behavior is fairly consistent.

Figure 2. Forensics Execution Tree starting with wcry.exe. Click to access the online report.

In our report, the attack starts with the launching of the wcry.exe sample. This executable drops a lot of files that are most likely configuration/data files needed to continue execution. We see the dropped files by clicking on the wcry.exe process and then viewing the File Ops Tab. A large number of files with the "wnry" extension are created for example.

Figure 3. attrib.exe and icacls.exe are the first processes launched by wcry.exe

This sample then proceeds to hide all the files in its own folder. This is done through the Windows “Attrib.exe” process as shown in Figure 3. We believe this is done so that the sample does not accidentally encrypt itself, though it could also be a basic technique to hide from investigators.

WCry.exe then executes Windows “icacls.exe” to modify the current folders permissions. We are still investigating as to why this is. This is the first ransomware family we have seen that actually utilizes this Windows process.

Figure 4. Last stage of the encryption process per file includes a rename.

Wcry.exe then begins the encryption process starting with files on the desktop. By following the flow of any one of the encrypted documents, we see that the malware wrote into a newly created file with the extension wncryt (t for temp?) and then after the encryption of the original file was completed it renamed the file to have the extension wncry.

For example:

1. The file 2014-financial-statements-en.pdf was read
2. The file 2014-financial-statements-en.pdf.wncryt was created.
3. The file 2014-financial-statements-en.pdf.wncryt was modified with encrypted content of the original 2014-financial-statements-en.
4. The file 2014-financial-statements-en.pdf.wncryt was renamed to 2014-financial-statements-en.pdf.wncry

It also creates an executable called @ and launches it. This executable creates the Tor Application folder, and installs Tor in it. This can be seen with suspicious event Tor Application Download. @ then launches taskhsvc.exe that is used to begin TOR communication.

Figure 5 cmd.exe execution triggered uac command prompt

After the encryption of files is finished we see a UAC prompt pop up because of a CMD that wishes to elevate privileges. The cmd.exe requires elevated privileges in order to delete shadow copies and modify boot options. If the user clicks OK then Shadow Copy Deletion occurs through both vssadmin.exe and wmic.exe. BCEdit and wbadmin executons are meant to occur based on the cmd.exe arguments (/c vssadmin delete shadows /all /quiet & wmic shadowcopy delete & bcdedit /set {default} bootstatuspolicy ignoreallfailures & bcdedit /set {default} recoveryenabled no & wbadmin delete catalog –quiet). However, neither are executed.

Figure 6. Suspicious Events for the process, Wall Paper Change details are shown

After the encryption the wall paper is also changed as seen in Suspicious events Wall Paper Change. Like Cerber and Locky, the wallpaper is changed to display a ransom message.

Persistence on boot is meant to occur based on the registry run key with the process named: tasksche.exe, but this process was never created by the attack and so nothing happens on reboot of the system. This process apparently should have been created from the downloader that detects if a kill switch is present. However, given that we executed this without executing the downloader it was unable to persist.

Finally the process called @ is also used to display the UI asking for payment.

For more information on Check Point’s Sandblast Agent Forensics please visit:

http://blog.checkpoint.com/tag/sandblast-agent-forensics/
