Creating Ransomware Decryptors

Alexander Adamov
CEO/Founder at NioGuard Security Lab,
Senior Teacher at Kharkiv National University of Radio Electronics
email: ada@nioguard.com, oleksandr.adamov@nure.ua

Abstract

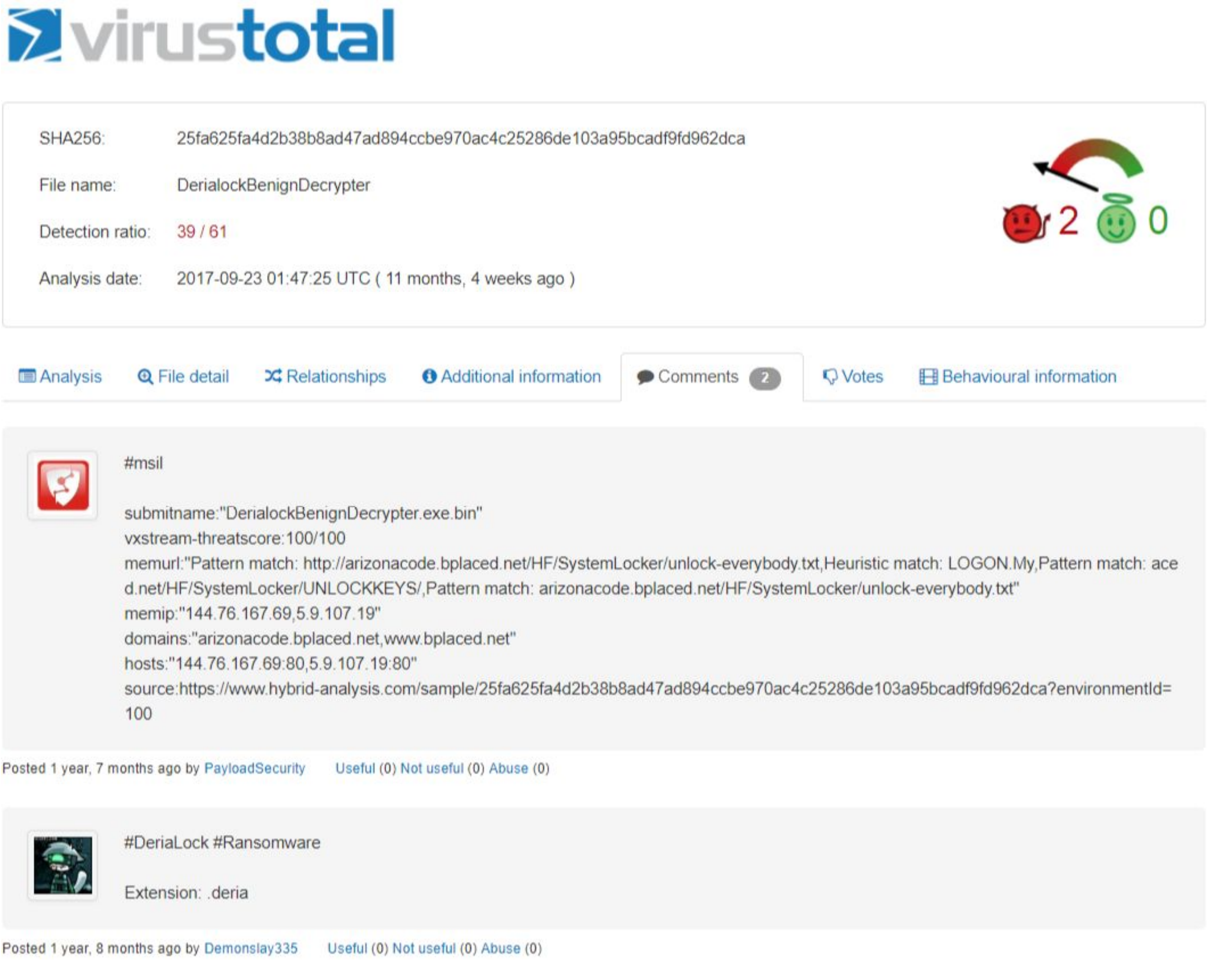
In this talk, I'd like to describe the process of creating ransomware decryptors and what problems exist from both creator's and consumer's points of view. These include but not limited to:

- No detailed documentation explaining how to use a decryption tool and limitations.
- Lack of transparency. It is hard to verify the decryptor's code delivered as a compiled PE file for malicious inclusions, especially in the era of nation-state cyber espionage.
- The new version of a decryptor may stop decrypting files encrypted by old versions of the cryptolocker. And vice versa, the current version of a decryptor when getting outdated cannot decrypt files encrypted by the new version of ransomware.
- Inflexibility. In most cases, you cannot configure decryption tool for your particular environment and easily update the key data.
- It is problematic to extract intermediate data such as a decryption key and initialization vector to continue the decryption process in your own way.
- The decryptor created based on original ransomware code can get detected by antiviruses.

Based on the experience of creating ransomware decryptors in our laboratory, I'm going to highlight the challenges we faced on the way and propose the solutions.

Problems with Current Anti-Ransomware Tools

Typically, security vendors and individual malware researchers deliver these free tools as compiled executables. Sometimes, by merely patching an original ransomware code that causes confusion of being detected at an endpoint with an antivirus as an original malware and showing messages with ransom requests. For example, DeriaLock decryptor [1] is the patched DeriaLocker code (not available for download at the moment of writing this abstract), which is still detected by 39 of 61 antiviruses as the DeriaLock ransomware (Figure 1).



SHA256: 25fa625fa4d2b38b8ad47ad894ccbe970ac4c25286de103a95bcadf9fd962dca

File name: DerialockBenignDecrypter

Detection ratio: 39 / 61

Analysis date: 2017-09-23 01:47:25 UTC (11 months, 4 weeks ago)

#msil

submitname:"DerialockBenignDecrypter.exe.bin"
vxstream-threatscore:100/100
memurl:"Pattern match: http://arizonacode.bplaced.net/HF/SystemLocker/unlock-everybody.txt,Heuristic match: LOGON.My,Pattern match: aced.net/HF/SystemLocker/UNLOCKKEYS/,Pattern match: arizonacode.bplaced.net/HF/SystemLocker/unlock-everybody.txt"
memip:"144.76.167.69,5.9.107.19"
domains:"arizonacode.bplaced.net,www.bplaced.net"
hosts:"144.76.167.69:80,5.9.107.19:80"
source:https://www.hybrid-analysis.com/sample/25fa625fa4d2b38b8ad47ad894ccbe970ac4c25286de103a95bcadf9fd962dca?environmentId=100

Posted 1 year, 7 months ago by PayloadSecurity Useful (0) Not useful (0) Abuse (0)

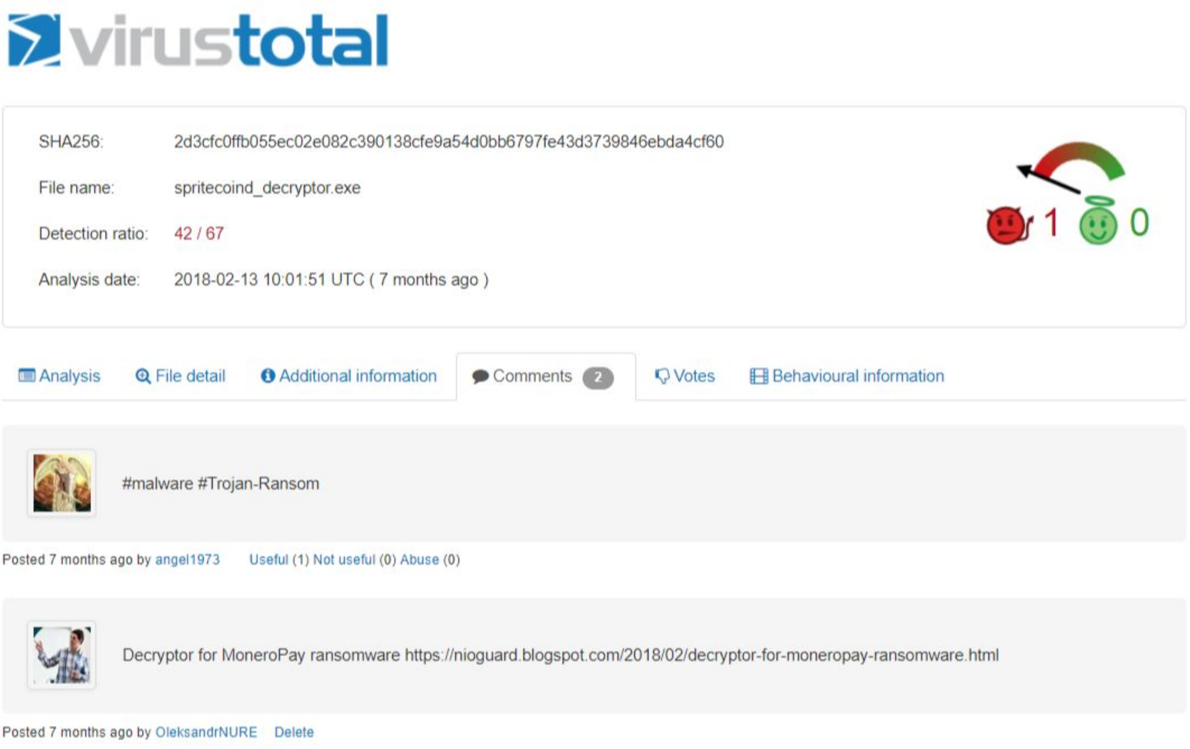
#DeriaLock #Ransomware

Extension: .deria

Posted 1 year, 8 months ago by Demonslay335 Useful (0) Not useful (0) Abuse (0)

Figure 1. The analysis of the DeriaLock ransomware decryptor on VirusTotal [2]

Another example, the MoneroPay decryptor [3] we created by simply modifying several bytes in the original ransomware code is also detected by antiviruses as ransomware (Figure 2). This will be shown during the talk.



SHA256: 2d3cfc0ffb055ec02e082c390138cfe9a54d0bb6797fe43d3739846ebda4cf60

File name: spritecoind_decryptor.exe

Detection ratio: 42 / 67

Analysis date: 2018-02-13 10:01:51 UTC (7 months ago)

#malware #Trojan-Ransom

Posted 7 months ago by angel1973 Useful (1) Not useful (0) Abuse (0)

Decryptor for MoneroPay ransomware https://nioguard.blogspot.com/2018/02/decryptor-for-moneropay-ransomware.html

Posted 7 months ago by OleksandrNURE Delete

Figure 2. The analysis of the MoneroPay ransomware decryptor on VirusTotal [4]

Solutions

To overcome the problem of the patched ransomware being detected by antiviruses, the security vendors create ransomware decryption tool sets, typically delivered as a free standalone application signed with the vendors digital certificate.

Still, the “black box” approach has a drawback - decryption tools are not open source and may contain a backdoor that will allow a vendor to collect data related to an incident including confidential information and deliver them to a malware lab. Moreover, running a third-party decryptor as an executable is usually prohibited by security policies.

To solve the weakness of the black box approach, the decryption tools can be delivered as an open source code. For example, we created an alternative decryptor for the DeriaLock ransomware with the help of Python language [5]. In such case the code of the decryptor can be easily reviewed for any malicious inclusions. Moreover, a decryptor written in Python does not require to be compiled into executable and can be run as is. Still, such a possibility exists.

Another challenge is to keep decryptors up to date. When a new version of ransomware comes out, the appropriate decryptor should be updated as well. In most cases, the key data which is typically stored in the configuration section. In such case, to update a decryptor, it is needed to add a new symmetric key, for example. This can be easily done by a victim if the decryption tool is available in open source and the new key is published by a security vendors.

Running an outdated decryptor may result in corrupting the encrypted files using inappropriate symmetric key during the decryption attempt. To avoid data corruption, ransomware check the encrypted file extension, which is specific for every version, and verify the file’s checksum in the encryption metadata stored in the beginning or end of the file. So do decryptors.

The solution can be also seen in collaborative efforts of security vendors and individual researchers that will help both to increase the variety and improve the quality of the ransomware decryptors. For example, the NoMoreRansom project [6] was launched two years ago to join the forces of Europol EC3, local police, and antivirus vendors in the battle against ransomware. As a result, the web portal gathered almost all available decryption tools under its umbrella. However, not all of them of the same quality and may cause one's frustration when using.

References

[1] http://blog.checkpoint.com/wp-content/uploads/2016/12/Derialock-Decryptor.zip
[2] https://virustotal.com/en/file/25fa625fa4d2b38b8ad47ad894ccbe970ac4c25286de103a95bcadf9fd962dca/analysis/
[3] https://github.com/AlexanderAda/Ransomware-Decryptors/blob/master/MoneroPay/
[4] https://www.virustotal.com/en/file/2d3cfc0ffb055ec02e082c390138cfe9a54d0bb6797fe43d3739846ebda4cf60/analysis/
[5] https://github.com/AlexanderAda/Ransomware-Decryptors/blob/master/DeriaLock/DeriaDecryptor.py
[6] https://www.nomoreransom.org/en/index.html