CS 460 Project Report

Donald Cha (ddcha3)
Nihal Pathuri (npathur2)
May 11, 2014

Overview

This is a comprehensive document of our development of detecting false positive of a certain malware family. Our code processes sample CryptoLocker automated feeds, analyzes the behavior of the given binary code through information obtained from Sandbox run of the binary, and determines whether the binary is CryptoLocker or not.

CryptoLocker is a ransomware trojan which encrypts a set of files using the RSA public-key encryption scheme. After encrypting the files, the private key to decrypt the files is stored only on the adversary's server. A typical symptom that a victim will know if he or she is infected by CryptoLocker is when the malware displays a message which offers to decrypt the data if a payment is made in bitcoins by a stated deadline. Once a payment is made, the program will legitimately decrypt the files, but if the deadline passes, the program uninstalls itself, leaving the files locked without a realistic way to decrypt them.

There are many different knock off variants of this malware. For example, some variants encrypt files using some symmetric key encryption and prompt the user to make a payment to decrypt with no intention of decrypting. This implicitly shows a problem that any Antivirus programs that tries to detect a valid CryptoLocker binary unfortunately has a high false positive rate. To mitigate this problem, our group has created a program that reads the an XML file which contains information about the binary run in a sandbox, analyzes the executable file's behavior, and determines whether the report from the Antivirus program is a false positive or not.

Development Process

1. Research on CryptoLocker

- We have spent excessive amount of time researching the behavior of CryptoLocker, how the infection process works, and known characteristics that CryptoLocker has.
- We did not look for a specific code signature of CryptoLocker because viruses nowadays hide their signature through encrypting itself or through other stealth techniques.
- We have spent a good amount of time to decipher what each xml log is doing such that we are able to make our code retrieve only the necessary information from the xml log.

2. CryptoLocker Overview

- CryptoLocker will install itself into %AppData% so it can execute without Administrator privileges.

    - On XP, "`C:\Documents and Settings\USER\Local Settings\Application Data\`"
    - On Vista and above, "`C:\Users\USER\AppData\Local\`"
- The binary usually has a random name in one of two formats:
    - `Rlatviomorjzlefba.exe`
    - `{34285B07-372F-121D-311F-030FAAD0CEF3}.exe`
- The binary will add itself to auto run by adding the following registry entries:
    - `KEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run`

      `"CryptoLocker.exe"`
    - `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce`

      `"*CryptoLocker.exe"`
- It will store a list of encrypted files in the registry usually under the following entry:
    - `HKEY_CURRENT_USER\Software\CryptoLocker_0388\Files`

3. Program Overview

- Our program assigns points when the mentioned conditions are true.
- The program obtains a list of suspicious executables of a sandbox run by filtering out the information from internet explorer and other core Windows executables
- We then examine if the executable writes an exe file into %AppData% mentioned above.
- We also check to see if the executable stores a registry entry into "...CurrentVersion\Run", and if the value it stores is an executable in %AppData%
- We then check to see if a PublicKey is stored in the registry by the executable.
- We also see if file paths are stored in the registry (indicated the encrypted files).

- We give points for each behavior, and sum them to a total
- If the total exceeds a predetermined amount, we can definitively say that this is a CryptoLocker binary.
- All false positives have a total under a predetermined amount.
- There is a certain range where in which we can't positively identify whether it is a CryptoLocker or not. We believe the program starts, matches some of the big conditions, but stops executing after detecting that its being run in a sandbox.

4. Tests

- The tests are done for both negative responses and positive responses to make sure our code covers all cases.

In general, our program shows various results from running some test cases. The way we determine if an executable file is CryptoLocker is by a point base system. If an executable file consists of several characteristics that CryptoLocker would have, we will give a high score.

For an executable file that is suspected to be a CryptoLocker, the code gives a high point and shows what files are encrypted by this malware.

For an executable file that is suspected to be “false positive”, you can observe that no file has been encrypted and the score that it received is close to zero.

Our code also considers an unidentifiable case. This is a case when there are not enough characteristics to clearly determine if an executable file is CryptoLocker. This case usually receives an average score such that the code cannot distinguish if an executable file’s behavior is closer to CryptoLocker or some other malware that acts very similarly as CryptoLocker.

A special case that we have encountered from our test cases is the following. Although the code analyzes the same executable file, it gives two different results. One of the results displays “False Positive” while the other one shows that it is CryptoLocker. We have noticed that “Metal”, which is a type of virtual Sandbox, is assumed to be easy for a virus to notice that it is running on a Sandbox environment. We assume that CryptoLocker removes itself from the computer as soon as it detects itself being ran on a Sandbox environment. This is why running the same executable file on two different Sandbox environments give different results.

5. Determining Less Helpful Rules

To determine the less helpful rules, we looked at the overall accuracy of AV, Snort, and Yara.

a. AV

To test the accuracy of AV's rules, we have chosen 7 days of AV CryptoLocker run (From May 4, 2014 to May 10, 2014).

As a result, we have found 174 valid xml logs that we were able to analyze and the result is the following.

Not False Positive: 2 executable files were determine to be not false positive. Out of those two files, only one of them was encrypting documents in the Sandbox Environment.

Unidentifiable case: 15 executable files were unidentifiable due to lack of characteristics to be CryptoLocker. Also, none of the documents were encrypted by these executable files.

False Positive: 157 executable files were known of be False Positive by our CryptoLocker analyzer.

As a result, we have determined that AV's CryptoLocker false positive rate is almost 90.23% while AV actually finding CryptoLocker is only 1.15%.

b. Snort

To test the accuracy of Snort's rules, we have chosen 7 days of Snort CryptoLocker run (From May 4, 2014 to May 10, 2014).

As a result, we have found 19 valid xml logs that we were able to analyze and the result is the following.

Not False Positive: 11 executable files were determine to be not false positive. But only four executable files encrypted the documents in the Sandbox environment. We assume the reasoning is because there were no files to encrypt or the malware removed itself before making any action.

Unidentifiable case: 0 executable files were unidentifiable. This shows that Snort rule either gets all of the characteristics or none.

False Positive: 8 executable files were known to be False Positive by our CryptoLocker analyzer.

As a result, we have determined that Snort's CryptoLocker false positive rate is about 42.11% while Snort actually finding CryptoLocker is about 57.89%. This shows that Snort has an average accuracy on determining CryptoLocker.

c. Yara

To test the accuracy of Yara's rules, we have chosen 12 days of Yara CryptoLocker runs (Between Apr 10, 2014 to May 9, 2014).

As a result, we have found 18 valid xml logs that we were able to analyze and the result is the following.

Not False Positive: 0 executable files were determine to be not false positive. This is unfortunate result for Yara. However, we have found that the reasoning for hitting 0 accuracy is because it ran half of executable files on Metal, which is a low-interactive Sandbox environment that recent malwares can acknowledge themselves being ran on Sandbox and take a measure to remove itself from the computer. Also some executables are known to be "CryptoLocker.txt" spammer. This is an executable file that acts very similar as regular CryptoLocker and notifies the users to pay the ransom through spamming text files on most directories, but it does not contain any characteristics that CryptoLocker should have.

Unidentifiable case: 1 executable file was unidentifiable.

False Positive: 17 executable files were known of be False Positive by our CryptoLocker analyzer.

As a result, we have determined that Yara's CryptoLocker false positive rate is about 94.44%

d. Conclusion

From the test runs on each detection rule, we have figured that Yara and AV have very high rate of hitting False positive. We believe that the most accurate rule for now is Snort. It is important to note that we are specifically looking for the actual CryptoLocker and not any of the knock off variants, which still may be malicious. Even though we determine that these files aren't CryptoLocker, they could still be one of the many knock off variants

References

http://www.bleepingcomputer.com/virus-removal/cryptolocker-ransomware-information

http://blog.malwarebytes.org/intelligence/2013/10/cryptolocker-ransomware-what-you-need-to-know/

http://en.wikipedia.org/wiki/CryptoLocker

http://www.reddit.com/r/sysadmin/comments/1mizfx/proper_care_feeding_of_your_cryptolocker/