Malware Analysis of WanaCry Ransomware

Abdurrahman Akkas, Christos Nestoras Chachamis, Livio Fetahu
{akkas, nhahamis, lfetahu}@mit.edu

May 17, 2017

1 Abstract

In this paper, we present some general observations on ransomwares. Furthermore we focus on the most recent and publicly covered development in the field, the ransomware WanaCry. We show how we analyzed the samples and what conclusions can be drawn from our results. Based on this information, we provide a number of preventive measures that users can take in order to protect their data and have their computers ransomware-free.

2 Introduction

Downloading suspicious or unknown files from the web may turn out to be detrimental for the user and can result in compromising their machine and the data stored in it. One particular type of malware is ransomware. This malware targets the user's files, encrypts them using state-of-the-art public key cryptography, and demands that a ransom is paid (usually in bitcoins and within a limited amount of time) in order for the secret key to be released to the user. Just by looking at the case of CryptoLocker, it is evident that ransomwares constitute a multimillion-dollar industry led by the cyber criminals [1].

The types of attack can vary from encrypting only specific types of files, as in BitCrypt, to encrypting the entire hard drive, as in Petya ransomware. Petya, for instance, inserts itself in the master boot record of the user's machine and doesn't even allow the original OS to execute as normal [2]. After the infection is done and files are encrypted, a message window shows up which usually states the encryption scheme used (e.g. RSA-2048) and states the amount of ransom demanded to decrypt files. The adversary requests the payment in bitcoins and offers communication through the Tor browser that enables anonymous communication in order to hide the traces that can lead to finding out his identity.

The most recent ransomware, WanaCry, is said to have attacked more than 200,000 international targets in 100 countries [3]. As part of our project, we found and analyzed samples of this specific malware. Our approach was to test the virus on a virtual machine and experiment with it. We found what files are created before our data is actually encrypted, which types of files are affected, what Internet connections the program tries to make, and what system libraries the virus uses. We used tools like OllyDbg in order to check the assembly code, WireShark to check the network traffic, and HxD to check the byte decomposition of the unknown files. Finally, we tried to find possible preventive measures. We investigated ways where we could make our files unaffected from the virus, and we ran the ransomware against antiviruses so that we could check their reactions.

3 Beating Ransomwares

Here we present a few optimistic cases of success against ransomwares. The ransomwares mentioned below have been broken by exploiting mistakes of their developers, or by luck, as in the case of TeslaCrypt.

BitCrypt was a ransomware that unintentionally used RSA-426 encryption. While its developers claimed to have used RSA-1024 encryption to encrypt files, the RSA key appeared to be a 128-digit integer, whose length is 426 bits, contrary to what the developers should have been aiming for (128 bytes). An RSA-426 encryption scheme can be feasibly broken on a typical PC. [4]

Petya, although seemed more malicious than BitCrypt because of encrypting the entire hard drive, exhibited an even more amateurish mistake in the encryption scheme. Even though the developer claims to have used a military grade encryption algorithm, the decryption key is hidden in the binaries. After extracting some specific data from the hard drive, a researcher built a decryption algorithm that looks for a the key to unlock the infected machine. [5]

Even more interesting is the case of TeslaCrypt. TeslaCrypt developers were approached by an ESET researcher via the support chat in the TeslaCrypt payment site, after noticing that the ransomware servers were not being maintained, and they decided to release the master key for decryption to the affected users. [6]

4 Software, Tools & Methods

In this section we present the main software, tools, and methods used in our research.

4.1 Ransomware

The ransomwares we have analyzed include WanaCry, Petya, TeslaCrypt, and Locky. [7]

4.2 Tools

The software we used are the following:

- VMware Fusion: The software used to create virtual machine instances where we run several ransomwares in an isolated environment.
- OllyDbg: The software that is used for debugging executable files. It is used for behavioral analysis of malwares where we run binaries step by step using debugger.
- WireShark: Tool used for tracking the network traffic of the virtual machine.
- HxD: The editor used to view and edit contents of binaries.
- Strings: The tool offered by Windows used to list the printable strings found in given executable files.

4.3 Antivirus Software

The antiviruses we used for testing the ransomware mentioned above are Avira Free Antivirus, Eset NOD32 Free Trial Version, and Kaspersky Internet Security Free Trial.

5 Malware Behavior

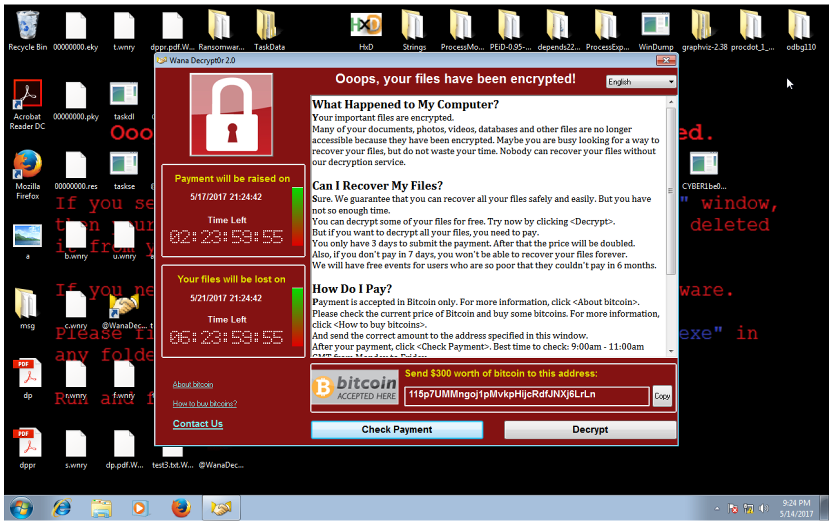
In this section we describe how the malware behaves after the virus has been executed. First a new folder is created, which contains the message of the payment in several different languages. Afterwards, the ransomware extracts six files with extensions *.wnry*, namely the *b.wnry*, *c.wnry*, *r.wnry*, *s.wnry*, *t.wnry* and *u.wnry*. A folder named *TaskData* appears. This folder contains the tor browser and 8 dll files. The next step is the creation of the files *taskdl.exe* and *taskse.exe*. Then, the *@WanaDecryptor@.exe* emerges. In every folder a shortcut of *@WanaDecryptor@.exe* is placed and all the user's files get encrypted. Encrypted files have a *.wnry* extension. The old files are deleted, the desktop background image changes and a ransom note appears. Several temporary files are deleted as well (Figure 1).

The program does not require Internet connection to run. Furthermore, it cannot distinguish between a virtual machine and a computer, a fact that makes the subsequent analysis possible, while it also reveals that the authors took little or no caution for their malware to be analyzed.

6 Malware Analysis

6.1 String Analysis

In this section we present our results from using the Strings tool mentioned above and observing the various strings found in the code of the malware. We were able to identify the specific file extensions that are encrypted by WanaCry.



Ooops, your files have been encrypted!

What Happened to My Computer?
Your important files are encrypted.
Many of your documents, photos, videos, databases and other files are no longer accessible because they have been encrypted. Maybe you are busy looking for a way to recover your files, but do not waste your time. Nobody can recover your files without our decryption service.

Can I Recover My Files?
Sure. We guarantee that you can recover all your files safely and easily. But you have not so enough time.
You can decrypt some of your files for free. Try now by clicking <Decrypt>.
But if you want to decrypt all your files, you need to pay.
You only have 3 days to submit the payment. After that the price will be doubled.
Also, if you don't pay in 7 days, you won't be able to recover your files forever.
We will have free events for users who are so poor that they couldn't pay in 6 months.

How Do I Pay?
Payment is accepted in Bitcoin only. For more information, click <About bitcoin>.
Please check the current price of Bitcoin and buy some bitcoins. For more information, click <How to buy bitcoins>.
And send the correct amount to the address specified in this window.
After your payment, click <Check Payment>. Best time to check: 9:00am - 11:00am

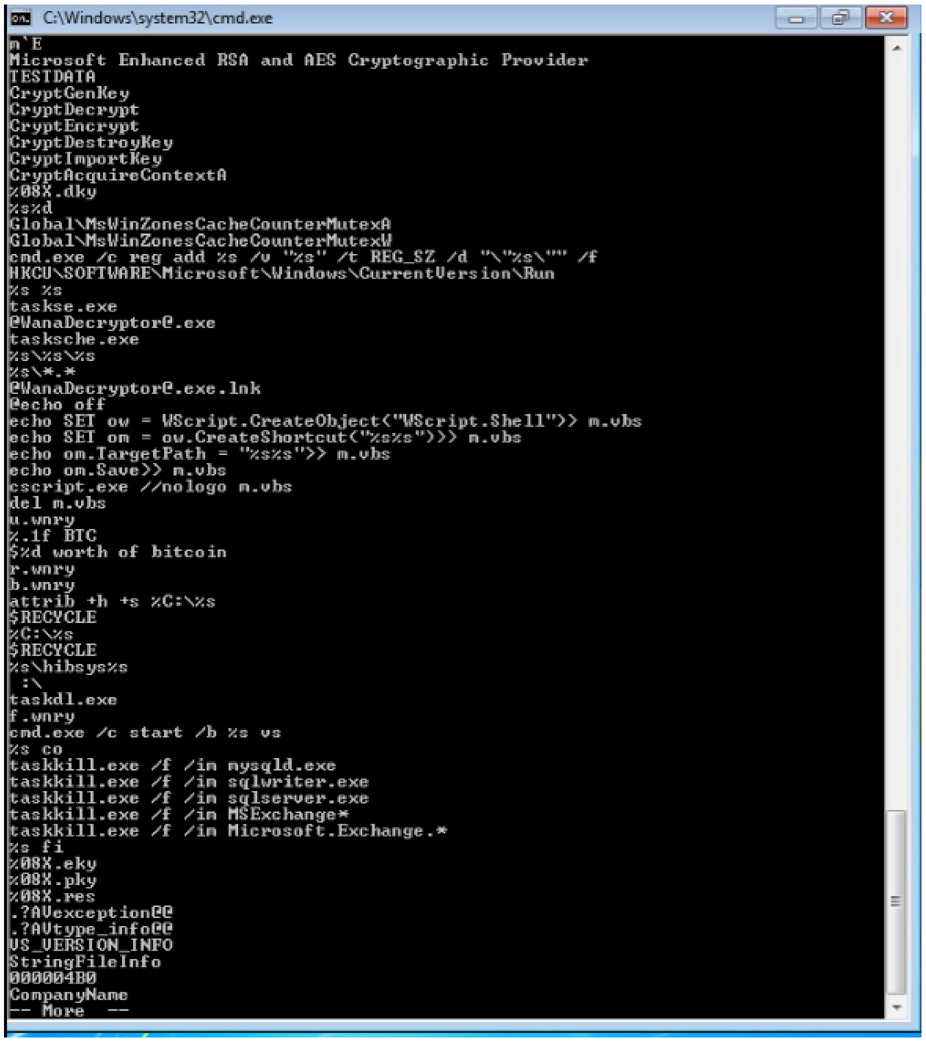
Send $300 worth of bitcoin to this address:
115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn

Figure 1: Computer right after infection

We also found a list of System calls that are used by the program. Some particular ones that caught our attention were the reference in *microsoft enhanced rsa and aes cryptographic provider* and the subsequent calls for key generation, encrypt and decrypt (Figure 2). This was a strong indication that the program used the system machinery to encrypt and decrypt our data. Indeed, the *advapi32.dll* mentioned in the code is the cryptographic package used from Windows. Unfortunately we were not able to confirm by removing or changing the *advapi32.dll*, as this dll file is used by a number of Windows programs and cannot be modified. One other finding which was interesting was that the code includes seven folders in which the files are said to not be encrypted (Table 3). This was indeed true, since we moved test files in the locations specified and none of them ended up being encrypted. A possible explanation is that the virus does not want to mess up with the system files, as it heavily depends on the system to run. In addition, we were able to identify some lines of batch code (Figure 2) which are responsible for creating a shortcut of the ransom note file, *@WanaDecryptor@.exe*, in every folder. Finally, there are five lines of code that terminate SQL database and Microsoft Exchange programs (Figure 2).

6.2 Traffic Analysis

In this section we present our results from using the WireShark tool and observing the messages and packages sent from and to the malware. One of the earliest actions of the program is to try to send and receive data requested from some IPs over the Internet. Checking those IPs (Table 2) revealed that they were actually Tor exits. The program did communicate with the above IPs, and in fact we were able to find the websites (Table 1) that it tries to connect to. We



```
m`E
Microsoft Enhanced RSA and AES Cryptographic Provider
TESTDATA
CryptGenKey
CryptDecrypt
CryptEncrypt
CryptDestroyKey
CryptImportKey
CryptAcquireContextA
%08X.dky
%s%d
Global\MsWinZonesCacheCounterMutexA
Global\MsWinZonesCacheCounterMutexW
cmd.exe /c reg add %s /v "%s" /t REG_SZ /d "\"%s\"" /f
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
%s %s
taskse.exe
@WanaDecryptor@.exe
tasksche.exe
%s\%s\%s
%s\*.*
@WanaDecryptor@.exe.lnk
@echo off
echo SET ow = WScript.CreateObject("WScript.Shell")> m.vbs
echo SET om = ow.CreateShortcut("%s%s")>> m.vbs
echo om.TargetPath = "%s%s">> m.vbs
echo om.Save>> m.vbs
cscript.exe //nologo m.vbs
del m.vbs
u.wnry
%.1f BTC
$%d worth of bitcoin
r.wnry
b.wnry
attrib +h +s %C:\%s
$RECYCLE
%C:\%s
$RECYCLE
%s\hibsys%s
 :\
taskdl.exe
f.wnry
cmd.exe /c start /b %s vs
%s co
taskkill.exe /f /im mysqld.exe
taskkill.exe /f /im sqlwriter.exe
taskkill.exe /f /im sqlserver.exe
taskkill.exe /f /im MSExchange*
taskkill.exe /f /im Microsoft.Exchange.*
%s fi
%08X.eky
%08X.pky
%08X.res
.?AVexception@@
.?AVtype_info@@
VS_VERSION_INFO
StringFileInfo
000004B0
CompanyName
-- More  --
```

Figure 2: System and Encrypting calls found in the code.

attempted to visit these using our browser but our efforts were unsuccessful. We also couldn't find any DNS records for these websites. However, the malware manages to connect to them, as we've found that there's a large TCP traffic after the initial connection. Moreover, the websites changed every time we ran the virus, indicating that their selection is random, either completely or from a given list. Next, we created a second virtual machine, we created a network between the two virtual machines and ran the virus in one of them. Strangely, we didn't observe the behavior described by the media, namely that the second virtual machine gets infected [8]. One possible explanation for this might be that the connection between the virtual machines has been secure (possibly an update?) or there is a special way to connect virtual machines on the same computer.

www.fjwxgmfsr53oycasfri.com
www.hqksyiqeq.com
www.pdofbgc3ue26xmusjiux.net

Table 1: Random-looking domain names WanaCry tries to connect.

6.3 Assembly Analysis

In this section we present our results from using the OllyDbg tool mentioned above and observing the assembly code produced from the exe file of the WanaCry ransomware. We used breakpoints and step by step execution in some parts, so that we figure out how the first files are created and how user's files are encrypted. We were able to find the exact spots in which data from the exe file are copied into the new files, and also we were able to identify the beginnings of the threads that encrypt the data.

6.4 HxD analysis

In this section we present our results from using the HxD tool. One way to using this tool is for checking the byte representation of several files. Checking file *b.wnry* revealed that it is actually a bitmap, the image that appears as background before ransom note appears. File *c.wnry* turned out to be a list of websites, empty pages with only one character inside (Table 2). File *r.wnry* was the "readme" file. File *s.wnry* was the zip file that contained the tor browser and the dll files. *u.wnry* was the executable file, @WanaDecoder@, which presents the ransom note, while *t.wnry* was an encrypted file. Since *t.wnry* was encrypted, we assume that it contains keys used for the encryption of user's data. We also used HxD tool in order to replace bytes of the code. When we replaced the bytes of *advipi32.dll* in the code with other bytes, the program could not run since the file was missing. Also, when we changed CryptoEncrypt or CryptoDecrypt the program would create the first files and then crash, without an encryption. A probable explanation for file *t.wnry* being created when CryptoEncrypt is missing is that the file is already encrypted. A probable explanation for the crash when decryption is missing is that file *t.wnry* needs to be decrypted in order for our data to be encrypted.

154.35.175.225	gx7ekbenv2riucmf.onion
176.10.104.243	57g7spgrzlojinas.onion

Table 2: Tor IPs and onion websites

6.5 Chosen Ciphertext Size Analysis

In this section we present the results of our experiments using several different-sized *txt* files. We observed that two *txt* files of size 1 byte and 15 bytes respectively produced ciphertexts of size 296 bytes, two *txt* files of size 17 bytes and 31 bytes respectively produced ciphertexts of size 312 bytes, and a *txt* file of size 33 bytes produced a ciphertext of size 328 bytes. If we assume that the first encryption is a block encryption using fixed block size and padding, like AES, then we can conclude that this block size is 16 bytes, or 128 bits, and the remaining 280 bytes might be randomly generated AES key.

7 Preventive Measures

7.1 Specific Locations Not Encrypted

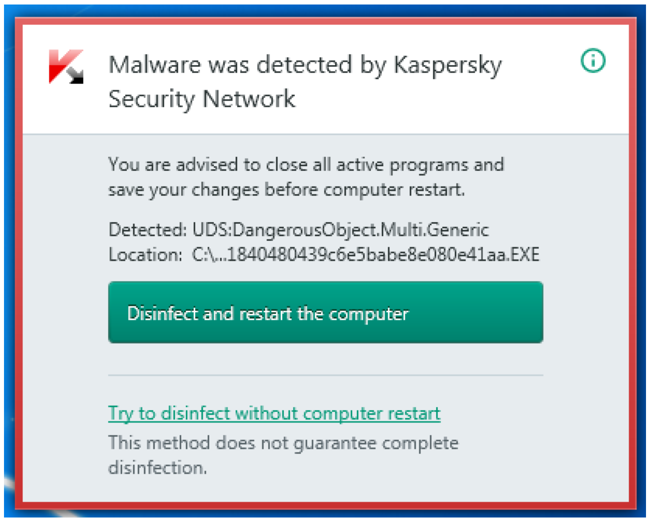
As mentioned in section 6.1 there are seven locations that are not encrypted by the WanaCry ransomware. This happens so that the virus does not accidentally encrypt one file that is critical for its code to run. This might be the case for other ransomware as well. One possible measure then can be to store a copy of the most important files in such special folders.

/Local Settings/Temp	/Program Files (x86)	/WINDOWS
/AppData/Local/Temp	/Program Files	/ProgramData

Table 3: Folders excluded from WanaCry

7.2 Antivirus Software

We installed the antiviruses mentioned in section 4.3 to our virtual machine and tested them against WanaCry. Even though they were free trial versions,


Malware was detected by Kaspersky Security Network

You are advised to close all active programs and save your changes before computer restart.

Detected: UDS:DangerousObject.Multi.Generic
Location: C:\...1840480439c6e5babe8e080e41aa.EXE

Disinfect and restart the computer
Try to disinfect without computer restart

This method does not guarantee complete disinfection.

Figure 3: Kaspersky's detection of WanaCry

all antiviruses successfully detected the malware threat and deleted the virus. The same results were observed when we used the antiviruses against the other ransomware viruses of section 4.1. Moreover, when we first tried to download WanaCry binary through Firefox, we were prompted a security message stating that we were about the download a malicious software. Thus having an up to date antivirus software and browser is an effective way to protect computer from ransomware threats.

7.3 Files With Special Extensions Not Affected

The extensions mentioned in section 6.1 and shown in Table 2 are always encrypted. However, this is not true for file extensions that do not belong in the list. One possible solution is, thus, to change all the extensions of the files to other extensions defined by us and use the normal programs to the new extensions. For instance, consider changing the *.pdf* extension of a file into *.pdf1*. We can still open the program using Adobe Reader, though the ransomware cannot encrypt this unknown extension.

7.4 Problem Solved By Updates

It has been revealed that the network propagation of the WanaCry ransomware is down to a Windows Server Message Block (SMB) security flaw [8], called EternalBlue and revealed in March 2017. The following April, Microsoft released an update [9], which did not include the previous versions of Windows. As a result, older versions of Windows were vulnerable to the threat above. WanaCry exploited this vulnerability and managed to infect whole networks [3]. Microsoft claims to have solved the problem in a recent update, for all versions [10]. Whether this is true or not, updates in general solve the latest bugs, flaws and security threats, so it is a good idea that every user have the latest versions of their operating system and programs.

8 Conclusion

In this paper we first described the working mechanisms of ransomwares and then analyzed several different ransomware samples. Our analysis mostly focused on WanaCry, a ransomware that had an enormous impact as we were writing this paper. Our results showed that WanaCry is a malware that is not so hard to analyze and understand. We've managed to run it properly on a virtual machine, unlike some other sophisticated ransomwares that can detect whether they are in a virtual machine and behave differently. We've figured out that it relies on the Windows cryptography libraries to do RSA and AES encryptions. The malware stops SQL database and Microsoft Exchange servers before the encryption, and it uses Tor browser to connect to Command&Control servers as we've found using different analysis tools. We've also analyzed the files that malware extracts before encryption phase and identified their content.

After the analysis, we've presented some novel prevention measures for WanaCrypt. For instance, we found some specific folders in the system are not encrypted and argued that moving critical files to these locations might be a good idea. Furthermore, we've found the set of extensions that WanaCrypt encrypts and showed that files can be protected renaming extensions to some pseudorandom values. Up to date antiviruses are shown to be effective against ransomwares in our analysis and a discussion on Windows security update is also provided.

We believe that WanaCry does not show innovative or genuine features, or particular elements that make it more dangerous than other ransomwares. Its uniqueness comes from the fact that it takes advantage of the EternalBlue exploit of Windows to spread.

9 What's Next

The current paper is not entirely complete, as there are more questions that we wanted to investigate but we found ourselves out of time. We present those questions in this section.

One behavior we tried to replicate was the described ability of WanaCry to infect nearby computers. Since it didn't work for two virtual machines on the same computer, we can investigate what other ways exist to connect them, like using different computers.

Another question is why the program unzips two exe files, instead of having them in its code. We plan to use string and assembly analysis on them as well. One possible explanation is that they can perform actions that are not related to WanaCry, but some other virus instead, so we can change some bytes and check if the program still runs.

Finally, it would be interesting to find the secret key decrypting *t.wnry*. The method we are going to use is finding the thread in which the keys are extracted from *t.wnry* and checking the nearby code step by step.

Bibliography

1. Turkel, D. (2016). Victims paid more than $24 million to ransomware criminals in 2015 — and that's just the beginning. http://www.businessinsider.com/doj-and-dhs-ransomware-attacks-government-2016-4

2. Gallagher, S. (2016). New ransomware installs in boot record, encrypts hard disk. https://arstechnica.com/security/2016/03/new-ransomware-installs-in-boot-record-encrypts-hard-disk/

3. Reuters. (2017). Cyberattack hits 200,000 in at least 150 countries - Europol. http://www.cnbc.com/2017/05/14/cyber-attack-hits-200000-in-at-least-150-countries-europol.html

4. Perigaud, F. (2010). Bitcrypt broken. http://blog.cassidiancybersecurity.com/post/2014/02/Bitcrypt-broken

5. leo-stone. hack-petya mission accomplished!!!. https://github.com/leo-stone/hack-petya

6. Adams, L. (2016). TeslaCrypt shuts down and Releases Master Decryption Key. https://www.bleepingcomputer.com/news/security/teslacrypt-shuts-down-and-releases-master-decryption-key/

7. ytisf. https://github.com/ytisf/theZoo/tree/master/malwares/Binaries

8. Graham, J. (2017). How to Rapidly Identify Assets at Risk to WannaCry Ransomware and ETERNALBLUE Exploit. https://blog.qualys.com/securitylabs/2017/05/12/how-to-rapidly-identify-assets-at-risk-to-wannacry-ransomware-and-eternalblue-exploit

9. MSRC Team. (2017). Protecting customers and evaluating risk. https://blogs.technet.microsoft.com/msrc/2017/04/14/protecting-customers-and-evaluating-risk/

10. Microsoft. (2017). https://technet.microsoft.com/en-us/library/security/ms17-010.aspx