

# Preliminary Report of Ransomware Petya Analysis

Song Yang

Xin Yang

Zhuohang Li

April 2018

## 1 Experiment Setup

This section mainly introduces the environment setup for our experiment.

### 1.1 MBR Recovery Methodology

The original Petya targets only at Master Boot Record(MBR) partitioned computers, UEFI + GPT partitioned computers will not be infected by this ransomware. In this part, we will test if it's possible to recover the system and user files by using MBR recovery in that Petya will only encrypt the MBR instead of specific types of files.

Since Petya will do severe damage to the file system table, we will launch this malware on a virtual machine. First, we set our virtual machine in VirtualBox as Microsoft Windows XP 32 bit, with no anti-virus software, partitioned by MBR. We kept the virtual optical disk drive and loaded the .iso file for Windows XP installation and recovery. Then we cut off all other connections to the virtual machine to isolate this environment. We will try to use the MBR recovery tools coming with the official installation disk.

After double-clicking on the Petya executable we downloaded, the virtual machine suddenly shut down as a hardware error, this might be shown as a blue screen error on a normal machine. Then it goes into the boot device selection interface. To avoid further damage, we directly enter the Windows XP installation CD. By far we avoided the damage that Petya would do to our file table, only the MBR section and several following unimportant parts are modified. At the main interface of installation CD, we press R to enter recovery mode and select the operating system that we would like to recover. After a few steps, we arrive at the recovery mode command line interface. Here we use the official command "fixmbr", after the confirmation, there will be a message showing that MBR is overwritten successfully. We then enter "exit" to restart. We found that we can directly enter the Windows XP as usual without seeing the Petya interface, which means we have successfully overwritten the malicious code stored in MBR section. Meanwhile, all files are normal as usual, without being encrypted.

Then we tried to recover MBR after the Petya completely executed. In this condition, we can clear the Petya ransom interface by using "fixmbr" command, but the system still failed to boot in that the Master File Table is destroyed while the fake CHKDSK interface is displaying.[2] So we tried to use third-party tools to enter a temporary PE system booting from a live CD or USB, then use disk tools trying to recover the system. However, if the malicious codes have been executed, it's not likely to recover the whole system even if we rebuild the MBR bootloader, but it's still possible for some third-party disk software to recover some of the files.

In our experience, it's still possible to recover the whole file system without paying the ransom, but it would need professional knowledge about lost data recovery and experiences about recovering the MBR and MFT.

### 1.2 Analysis

When we simply load the malware into IDA Pro, we can see the code starts from .text:00401000, and it uses CreateFile and DeviceIoControl. According to MSDN, CreateFile returns a handle

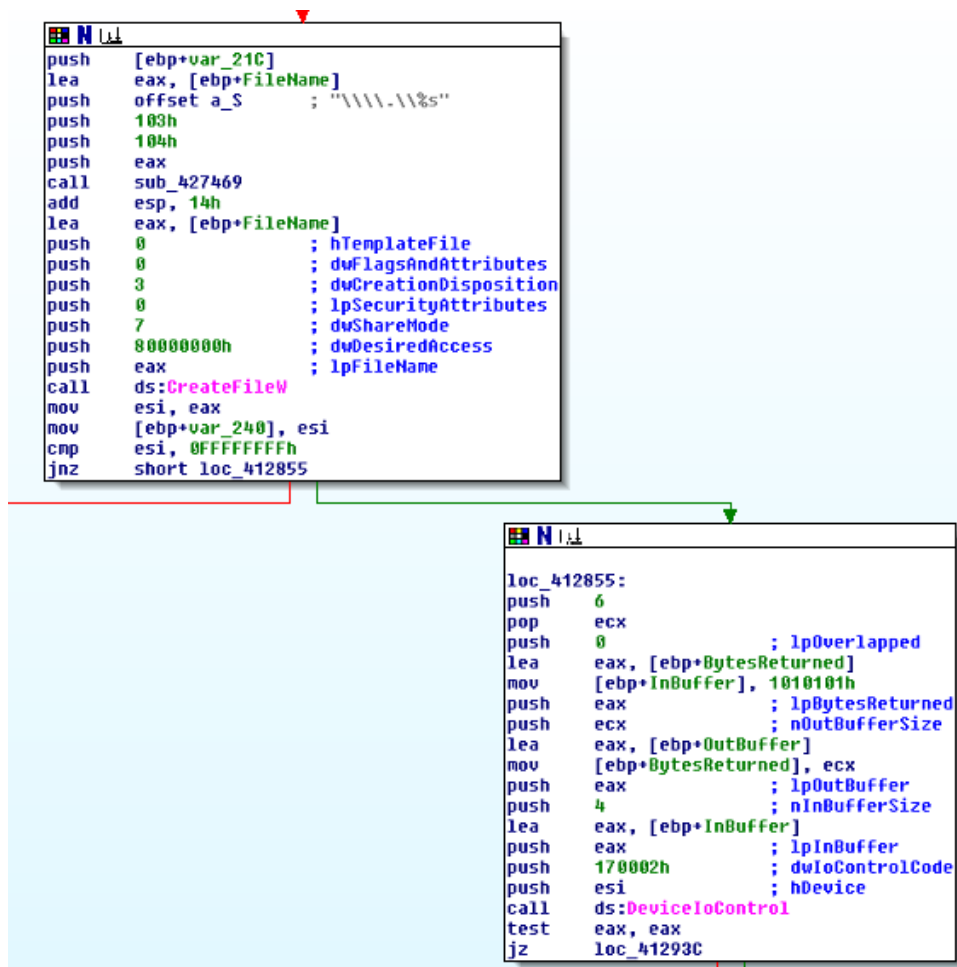


Figure 1: calling I/O function

that can be used to access the file or device for various types of I/O depending on the file or device and the flags and attributes specified, and DeviceIoControl sends a control code directly to a specified device driver, causing the corresponding device to perform the corresponding operation. [1] Therefore we can infer that the program is trying to scan the disks. No further information we can get. The program is actually packed/encrypted with FUD/crypter, which is a pretty strong packer. The malware also disguises itself as a Win32 program, but in fact, it extracts all malicious codes to memory[3].

The malware also contains a lot of misleading strings to prevent people from figuring its actual purpose. Since it's hard to get true malware program and assembly languages, we tried to start up a local debugger, but unfortunately, it's hard to locate actual breakpoint that can stop the programming from restarting the computer. We failed to get the dump file of memory, cause it seems that after programs loaded to memory, it runs at once like another thread and out of control from IDA pro. So we guess that the function that causes system reboot is exactly loaded into memory.

### 1.3 Internet Behavior Detection

In order to find out indicators of Petya's Internet behaviors, we use VirtualBox to build an internal LAN environment without access to the Internet as shown in Figure 2. We set up three virtual machines, VM 1 (IP:192.168.0.10), VM 2 (IP:192.168.0.12) and VM 3 (IP: 192.168.0.14), and get them connected to each other. VM 2 is set up as the host. VM 3 is for detecting how

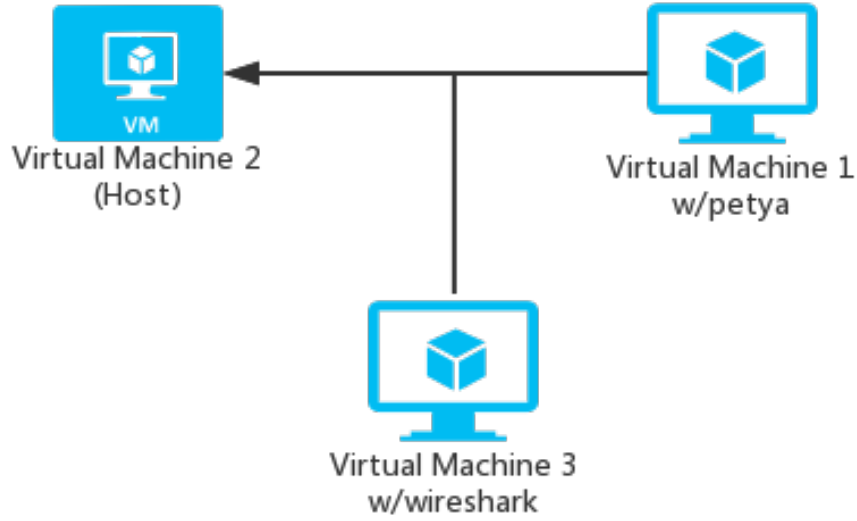


Figure 2: Network Setup

the TCP streams go from the infected computer to the host. So it has to be fully protected. We are using Windows 7 (64 bit) as the testing OS.

## 2 Results

This section would talk about the results and conclusions that we got from our experiment.

### 2.1 Defense Mechanism Based on MBR Recovery

From our experiments, we can see that ransomware like Petya, which targets MBR partitioned machines and only encrypts the MBR section instead of specific files can be easily recovered at the first stage after infected. We can prevent further steps from happening before the malware does harm to other parts by launching MBR rebuild or overwrite. Advanced methodologies are that we can backup the MBR and MFT regularly, and if any modification towards these parts detected, a recovery boot item will be added and users can automatically enter the recovery stage before the malware execute other malicious codes. In this way, we can stop the malware at the first time and help protect users from losing data.

### 2.2 Defense Petya from Internet Attack

This original version of Petya does not have any indicators of being able to have a positive Internet attack. It can only infect the computer which receives the malware file and opens it. Wireshark cannot receive any package sent from the virtual machine infected Petya.

Other than the original version of Petya, we also test the updated version on our test platform. The recent update addict worm features to it. We detected that a copy of Petya is sent after the malware runs before the system shut down. The worm would scan the whole internal network and form a list and finally broadcast its copy which is harmful.

In this case, Petya has to gain some privilege to call system level functions.

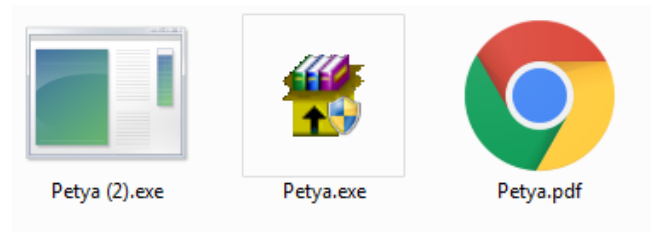


Figure 3: Petya Icons

## 2.3 Other Original Defense Methods

Since Petya is such a low-level malware, once it gets rebooted, there are not many things we can do at the software level. Other than methods talked about below, there are still some ways to protect personal computers not only for Petya but for other similar user-cheating malware.

### 2.3.1 Icon Protection Mechanism

Learning the reason that user was deceived, is that Petya disguise itself as a pdf or zip file which is pretty common and user-friendly in work, but actually it is an executable file. Executable file icon is one of the alarming icons, so the malware developers change them to 'harmless' icons.

As a result, the 'user-believed' icons must be strictly used and should build a strong icon-using rule. The Icon Protector should be a system level detector. Icons including pdf, zip, word, ppt, excel and folder files should be strictly used and cannot be used privately. Microsoft provided software should have a brief indicator showing that it is a safe file. Executable file has to be strictly matched with exe file icons to alert users.

Claiming that, the Protector should not play a role as a anti-virus software, but should detect if the new file saved in the computer matched with icon-extension name rule and provide reasonable warnings.

### 2.3.2 Privilege Protection Mechanism

By just clicking yes to grant all permissions to a software is dangerous. Combining with Android, when installing apk software, the system gets the permission labels and list all applied permissions to the user. It is a very 'Android' setting, but easy to discover potential risks. Permissions are numerous, but we can classify them. By classifying different kinds of users, the system should provide different permissions. Give an example of classification, for preliminary users, system-sensitive permissions are unavailable or hard to apply, then the probability of getting infected for them will be reduced. For master users, they can tell if a pdf file needs system rebooting permission.

Classifying permissions is such a huge imagination, but should be a solution with Significant effect to preventing malware.

## 2.4 User's Perspective

Petya is a targeted malware. The original version spreads only through email. It baits user to click on the executable file, requests for the privilege, and restart the device to get full access. Petya attacks the computer by taking advantage of user's incautious. Since it is specially designed and targeted, we should have a more cautious user behavior.

The advice given by NJCCIC is to apply patches to all out-of-date software and discontinue the use of unsupported/EoL software or hosts, update anti-virus software with the latest definitions and, if possible, set it to automatically update. Because it asks for the privilege, we should run all software as a non-privileged user to diminish the effects of a successful attack.



Permission Table	
 Preliminary User	 Master User
(Partial Privilege) File Read File Write Read System File Using I/O ...	(Full Privilege) File Read File Write Service Create I/O control ...

Figure 4: Assumed Privilege Table

Applying the Principle of Least Privilege to all systems and services is a good way to prevent ourselves from this kind of attack.[2]

## References

- [1] DeviceIoControl function. [https://msdn.microsoft.com/en-us/library/windows/desktop/aa363216\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa363216(v=vs.85).aspx).
- [2] Petya. <https://www.cyber.nj.gov/threat-profiles/ransomware-variants/petya>. [Accessed: 2016-12-7].
- [3] Petya – taking ransomware to the low level. <https://blog.malwarebytes.com/threat-analysis/2016/04/petya-ransomware/>. [Posted: April 1, 2016].