WannaCry Aftershock

On May 12th, 2017, organizations across the world were attacked by a new, fast-spreading piece of malware we now know as WannaCry. It is now considered one of the most widespread, and notoriously destructive malware attacks in history, halted only by a researcher getting a lucky break, registering a domain name embedded in the malware that unexpectedly acted as a kill switch. But even today, more than two years hence, WannaCry continues to affect thousands of computers worldwide.

Peter Mackenzie

Contents

Worm, ransomware...or unwanted vaccine?

Research by Sophos reveals that, despite the availability of security patches and anti-virus protection against WannaCry, more than 12,000 unique variants exist in the wild. These newer variants can spread more effectively, and stay hidden for longer, than the original WannaCry. The most prevalent of these variants are locked in a battle for domination.

In this paper, we will explain why WannaCry's crucial kill switch is preventing a second outbreak, even though the switch is no longer relevant for many victims. We also investigate how the new variants behave almost like a vaccine (though not an entirely benign one), protecting potential victims from the original WannaCry – and raising a red flag for security teams.

We need to stress that "infected with broken WannaCry" is not a reasonable method of protecting your computer, and certainly not any kind of vaccine you want. To get infected in the first place, even with an (essentially) inert variant, means the computer is not patched against the EternalBlue exploit. If you haven't patched against that exploit, then it is highly likely you haven't patched at all in the last two or more years, and this could leave you at risk of a huge number of threats, many far worse than WannaCry.

The basics of WannaCry

To understand the new discoveries, you need to understand how the original WannaCry works.

The now-infamous WannaCry ransomware attack that started on May 12th, 2017, spread to over 200,000 computers across 150 countries. Notable victims included the British National Health Service (NHS).

The WannaCry kill chain can be split into three main components:

The first is the worm element that WannaCry uses to spread to other computers. This behavior extends beyond the local network and also tries to spread the malware to random IP addresses across the internet. It does this by exploiting a vulnerability in the Microsoft SMB protocol, which Windows users typically use to transfer files between machines. This exploit, called EternalBlue, is a piece of stolen property, itself: A group that called itself 'The Shadow Brokers' stole, and then leaked, the exploit, which allegedly originated with the NSA.

The exploit allows WannaCry to copy and execute itself automatically on remote computers. Notably, Microsoft had released a patch for this vulnerability two months prior to the attack, but the victims had not yet updated their computers at the time.

The second component is what has become known as the "kill switch," a line of code which, during the early stages of the attack, checks to see if a specific web domain name is live. If the malware determines that the web domain is live, the attack stops dead.

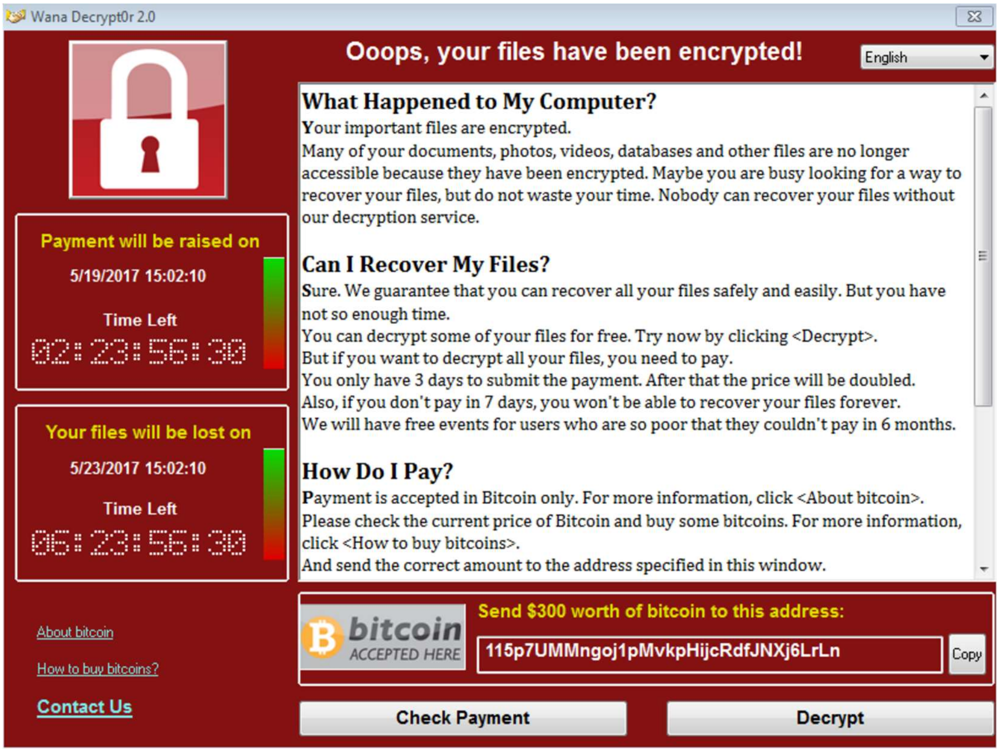
This URL was not registered at the time of the original attack, which meant that WannaCry was able to continue to spread. Two U.K. threat researchers, Marcus Hutchins and Jamie Hankins, registered this domain on May 12, 2017, effectively ending the WannaCry attack.

Sandbox evasion vs. fail-safe hypothesis

Unless the creators of WannaCry explain their motivation, it's impossible to know for certain why they added a kill switch. However, two prominent hypotheses exist: Either the attackers wanted to have a way to stop the attack at their discretion, or, more likely, it was an anti-sandbox evasion technique.

Some sandbox environments fake responses from connections to URLs to make the malware think that it is able to access the internet. As the domain name had not been registered, it meant the attackers knew that if a successful connection was made, it most likely meant they were in a sandbox, so they could end the attack to hide the true nature of the file.

Finally, the third component of WannaCry is, of course, the ransomware payload. In the 2017 outbreak, this payload encrypted thousands of files and left behind a ransom note with instructions to pay $300 in Bitcoin to recover the files.



Wana Decrypt0r 2.0

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.

Payment will be raised on
5/19/2017 15:02:10
Time Left
02:23:56:30

Your files will be lost on
5/23/2017 15:02:10
Time Left
06:23:56:30

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

The WannaCry ransom note popup

When you put those three elements together, you get a more complete picture of how WannaCry works: When a computer infected with WannaCry attempts to spread the malware to a new computer, it first checks if the new computer is already infected with WannaCry. If the computer is clean, WannaCry proceeds to use the EternalBlue exploit to drop a file called **mssecsvc.exe** to the C:\Windows directory of the new computer and then executes it.

WannaCry is not polymorphic. The original file (mssecsvc/r.exe) does not change. It has a unique MD5 Hash of: db349b97c37d22f5ea1d1841e3c89eb4. This file is spread to every newly-infected computer, which then attempts to spread to others.

The first thing that this file does is check for a connection to the kill switch domain (the memorable and intuitive www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com). If the computer cannot connect to this domain, the attack continues, repeating the attempt to spread to other computers, while simultaneously extracting files from a ZIP archive. The contents of this ZIP launch the data-destroying encryption phase of the attack.

Other considerations of the WannaCry killchain

There are a couple of interesting caveats to how WannaCry was portrayed at the time of its initial outbreak. For example, despite significant implications that unpatched Windows XP computers were primarily responsible for WannaCry's rapid spread, more than 97% of WannaCry detections at that time were coming from the newer Windows 7 operating system.

It's also worth noting that, while a computer patched against the EternalBlue exploit is no longer vulnerable to being infected by a remote connection from another WannaCry-infected computer, if that computer was infected *before* it was patched, it will still try to infect other computers; The anti-EternalBlue patch only prevents the vulnerability from being exploited, not from exploiting others.

If everything works as designed and nobody updated WannaCry, the file that started spreading on May 12th, 2017, would be the same as the file seen in the wild today.

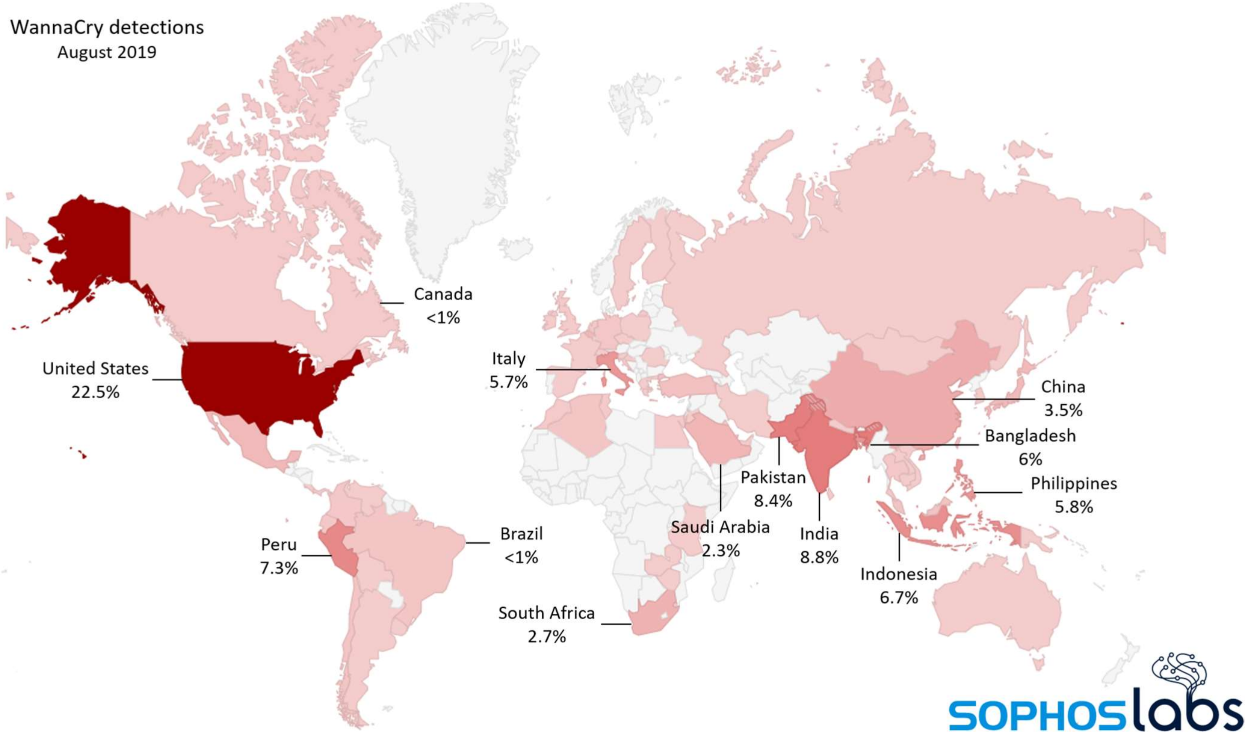
But the reality is very different, and much more intriguing.

New WannaCry findings

Sophos' research is based on a signature named *CXmal/Wanna-A*, the detection name that identifies when a computer, suddenly finds the WannaCry payload (the mssecsvc.exe file) plopped into the **`C:\Windows`** directory. On a Sophos-protected machine, the client application immediately blocks and removes this file.

Using this detection data, we can see how many computers are being attacked, repeatedly, by other computers, as well as the file dropped during the attack. These infected machines could be on the same network as the ones being attacked, or possibly anywhere in the world. All we really know about the infected machines that attempt to spread the infection is that they don't have a working anti-virus product (certainly not ours) on them. Otherwise they would have stopped WannaCry and wouldn't be attempting to infect other machines.

In the three month period between October 1, 2018, and December 31, 2018, Sophos logged 5,140,172 detections (not individual computers) of CXmal-Wanna-A. Nearly two years on from the original attack, as nearly every machine that can install the EternalBlue patch has already done so, why are there still so many detections?



Data from August, 2019 about detected (and blocked) attempts by WannaCry to infect customer machines, broken out by targeted country

Just as a sanity check, since that data was nearly a year old, we reran our queries looking at just one month of attack data, from August, 2019. We discovered that in that month alone, we had recorded more than 4.3 million attacks against customer machines. That seems like a significant increase, but it's worth noting that these numbers can be misleading, because the data is based on customer feedback, and the number of customer reports naturally changes over time as the size of the customer base changes. That can make it seem like the problem is getting worse.

What was important to note is that the proportion of the total number of attacks targeting Sophos customers in specific countries remained consistent in the data from 2018 and 2019, with machines in the US topping the list of countries most subjected to failed attempts at WannaCry infections. We'll keep monitoring these numbers over time to see if the increase in detections continues.

The fact that WannaCry is still going at all raises some very interesting questions, for example:

1. Are all of these machines really still not patched?
2. Why is the kill switch not preventing the infected computers from trying to attack others?
3. Why is no one complaining about files being encrypted?

We knew the answer to question 1 already, as the CXmal/Wanna-A detection is only possible on unpatched machines. To be certain, we did investigate a random selection of computers to manually verify that they had, indeed, not been patched against EternalBlue, or anything else, in the last two years.

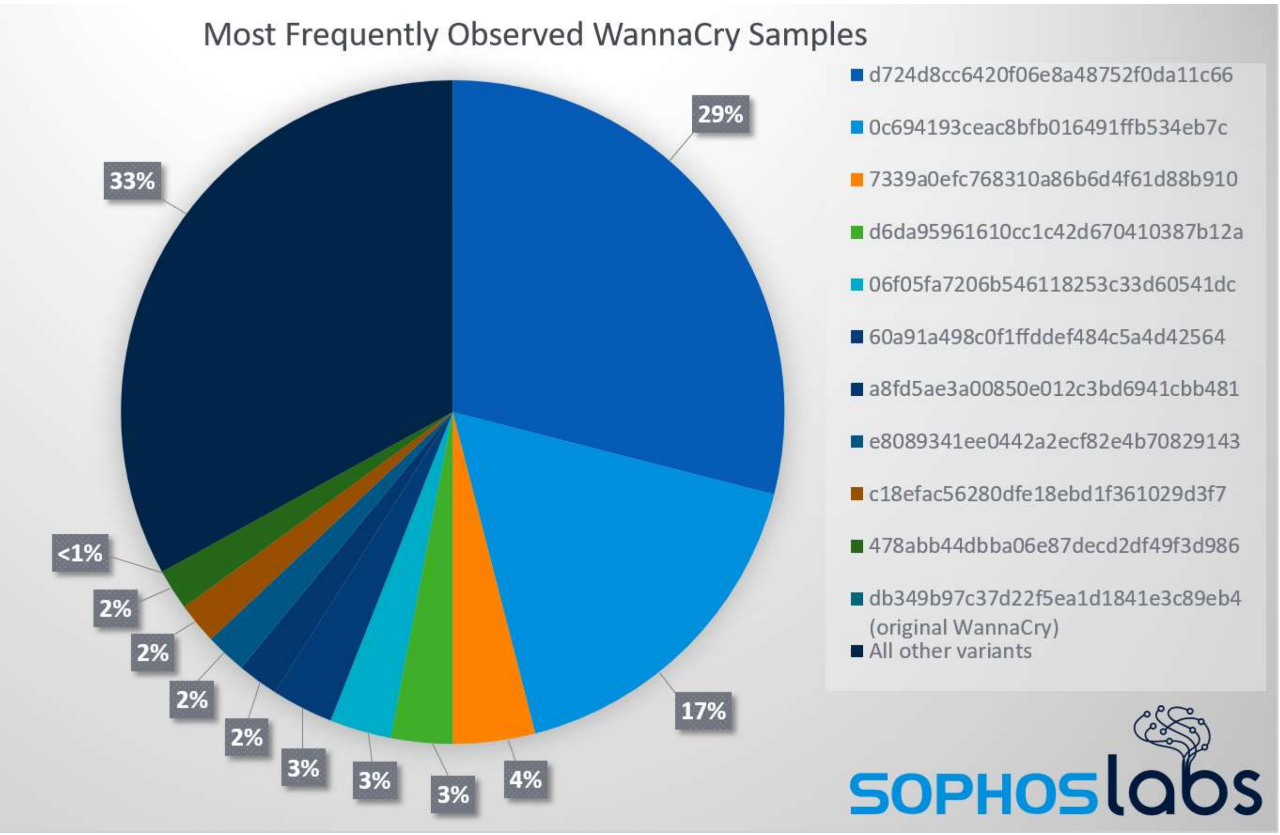
For question 2, we know the computers reporting the detections have internet access, as that is how we get the data. As they are most likely being attacked by infected computers on the same network, it seems plausible that these would also have internet access. So why isn't the kill switch stopping them?

For question 3, we don't know how many infected computers are still out there. We are looking at data from protected computers that are being attacked. Assuming there are many infected computers, we also have to assume that those computers would have been encrypted, by now, and yet nobody is reporting seeing encrypted files or users finding new WannaCry ransom notes.

The WannaCry variants

Analysing the 5.1 million CXmal/Wanna-A detections over the three-month period from October 1 through December 31, 2018, we immediately discovered something unexpected: The malicious file being dropped on these computers was *not* the original WannaCry mssecsvc.exe file (MD5: db349b97c37d22f5ea1d1841e3c89eb4). In fact, among the 5.1 million detections we identified 12,481 unique files.

- The original, true WannaCry file was seen only 40 times, a number so low that it could easily be attributed to testing, rather than a real attack.
- 12,005 of the unique files identified (96.1%) were seen fewer than 100 times each.
- 476 of the unique files (3.8%) accounted for an overwhelming 98.8% of the detections.
- Ten files accounted for 3.4 million (66.7%) of the detections, with the top three accounting for 2.6 million (50.1%). We consider some of these files in greater detail below.



As the data comes from three months at the end of 2018, we also checked the latest data from August, 2019 to see how it compares. In total there were 4.3 million CXmal/Wanna-A detections during August, indicating that, remarkably, the amount of WannaCry detections seems to be rising, and now exceeds long-term malicious worm traffic like that generated by Conficker

WannaCry's infamous kill switch

We analyzed the top 10 most prevalent files and quickly identified that they had all been altered very early in the code. The alterations in all 10 samples bypass the kill switch entirely. This means that these updated WannaCry variants' ability to spread is no longer restrained by the kill switch.

Death of the kill switch?

We wanted to understand if the kill switch bypass was true for all the 12,481 unique files (hashes) seen. Unfortunately, due to the extreme rarity of many of these files we were only able to obtain 2,725 of them. This included the majority of the top 1,000, as well as many that were only seen once. In total, the files we were able to obtain accounted for 97.8% of the detections.

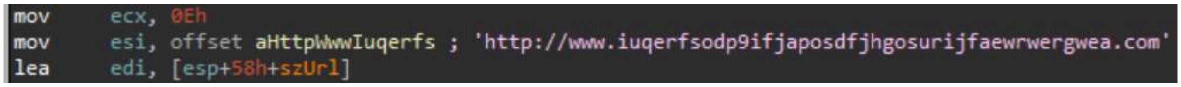
Some form of killswitch bypass was present in all 2,725 files we analyzed.

It's important to note that the changes appear to have been made via the use of a hex editor, rather than through recompilation of the original source code. This suggests that these changes were not made by the original creators.

Taking a deeper look at the changes, we can see four different methods that have been used:

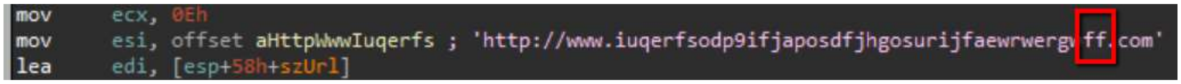
1. Simply removing or changing the kill switch URL. Roughly half of the samples did this, with most just removing the URL completely. The next most common approach was to change the last two letters from “ea” to “ff”.

Original:



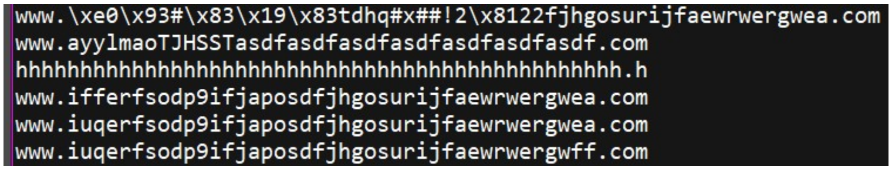
```
mov     ecx, 0Eh
mov     esi, offset aHttpWwwIuqerfs ; 'http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com'
lea     edi, [esp+58h+szUrl]
```

Updated:



```
mov     ecx, 0Eh
mov     esi, offset aHttpWwwIuqerfs ; 'http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwff.com'
lea     edi, [esp+58h+szUrl]
```

We spotted several variations of killswitch domains in these samples.

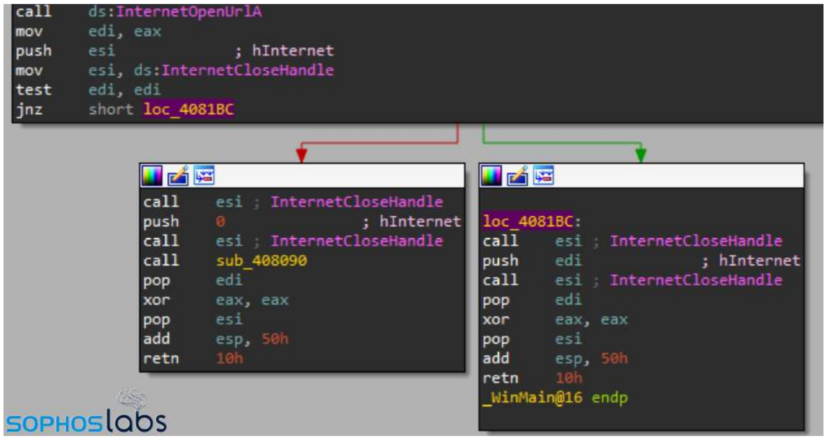


```
www.\xe0\x93#\x83\x19\x83tdhq#x##!2\x8122fjhgosurijfaewrwergwea.com
www.ayylmaoTJHSSTasdfasdfasdfasdfasdfasdfasdf.com
hhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh.h
www.ifferfsodp9ifjaposdfjhgosurijfaewrwergwea.com
www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwff.com
```

Modifying the killswitch domain resulted in these variants of WannaCry being unable to connect. Instead of exiting, they would simply move to the next command, which was to launch the attack.

2. The second method was to change the code to basically instruct the malware: regardless of the result of the kill switch test, move to the next command (execute the attack).

Original:



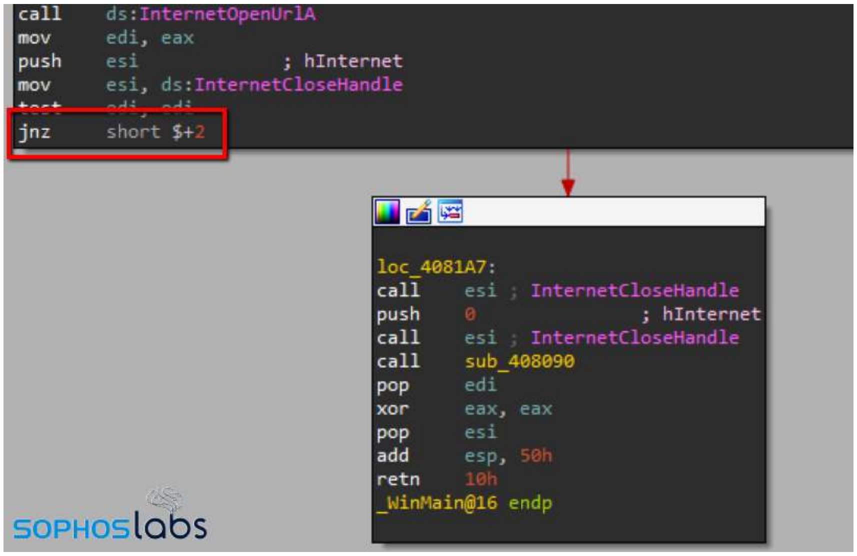
```
call    ds:InternetOpenUrlA
mov     edi, eax
push    esi             ; hInternet
mov     esi, ds:InternetCloseHandle
test    edi, edi
jnz     short loc_4081BC
```

```
call    esi ; InternetCloseHandle
push    0               ; hInternet
call    esi ; InternetCloseHandle
call    sub_408090
pop     edi
xor     eax, eax
pop     esi
add     esp, 50h
retn    10h
```

```
loc_4081BC:
call    esi ; InternetCloseHandle
push    edi             ; hInternet
call    esi ; InternetCloseHandle
pop     edi
xor     eax, eax
pop     esi
add     esp, 50h
retn    10h
_WinMain@16 endp
```

Subroutines that instruct WannaCry either to attack (left) or to quit, depending on the results of the killswitch test

Updated:



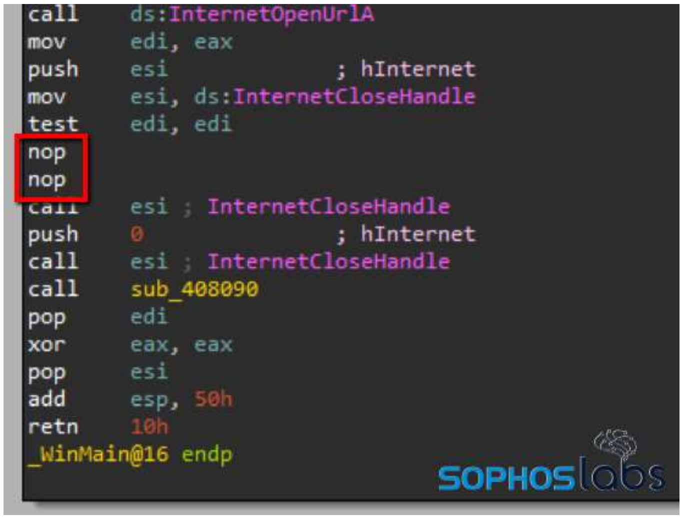
```
call    ds:InternetOpenUrlA
mov     edi, eax
push    esi             ; hInternet
mov     esi, ds:InternetCloseHandle
test    edi, edi
jnz     short $+2
```

```
loc_4081A7:
call    esi ; InternetCloseHandle
push    0               ; hInternet
call    esi ; InternetCloseHandle
call    sub_408090
pop     edi
xor     eax, eax
pop     esi
add     esp, 50h
retn    10h
_WinMain@16 endp
```

Modified WannaCry skipping past the "exit" subroutine

3. The third method was to replace the kill switch with “nop” (No Operation) opcodes, which means: do nothing. These replaced the code that would check the result of the kill switch connection attempt. This means that it doesn’t check the result and just continues with the attack.

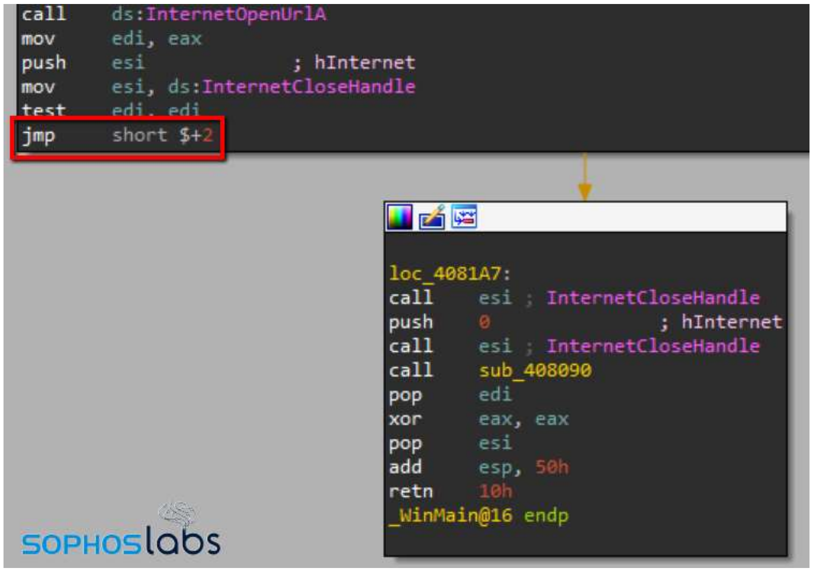
Updated:



```
call    ds:InternetOpenUrlA
mov     edi, eax
push    esi             ; hInternet
mov     esi, ds:InternetCloseHandle
test    edi, edi
nop
nop
call    esi ; InternetCloseHandle
push    0               ; hInternet
call    esi ; InternetCloseHandle
call    sub_408090
pop     edi
xor     eax, eax
pop     esi
add     esp, 50h
retn    10h
_WinMain@16 endp
```

Killswitch substituted for a pair of nop *calls*

4. The final method seen was to use a 2 byte “jmp”, which jumps over the code that checks the result of the kill switch connection, also resulting in it just continuing the attack.



```
call    ds:InternetOpenUrlA
mov     edi, eax
push    esi             ; hInternet
mov     esi, ds:InternetCloseHandle
test    edi, edi
jmp     short $+2
```

```
loc_4081A7:
call    esi ; InternetCloseHandle
push    0               ; hInternet
call    esi ; InternetCloseHandle
call    sub_408090
pop     edi
xor     eax, eax
pop     esi
add     esp, 50h
retn    10h
_WinMain@16 endp
```

Leaping right over the killswitch subroutine with a jmp *instruction*

Kill switch survival

At this point you might be tempted to believe that we had answered our question of “Why isn’t the kill switch stopping the attacks?” After all, this data indicates the kill switch is effectively dead, and looking for connections to the URL (as an indicator of compromise) no longer is relevant. This is not the case. In a recent interview, Jamie Hankins said that, in “June 2019 alone the kill switch prevented about 60 million ransomware detonations,” indicating that there are still (potentially) thousands of computers infected with the original WannaCry, and keeping the kill switch domain online is the only thing stopping a second outbreak.

We can even see that, despite the kill switch, there are still people paying ransom demands to the creators of WannaCry. We know this because WannaCry includes three hardcoded Bitcoin addresses, to which you must send your $300 worth of Bitcoins if you choose to pay the ransom. While Bitcoin accounts are mainly anonymous, all transactions are completely public, so we can see every payment in and out of these accounts:

- `13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94`
- `12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw`
- `115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn`

At the time of writing this, over two years since the original attack, we can still see that the occasional victim is paying the ransom demand.

Important*: If you do happen to find a machine infected with WannaCry,* **do not** *pay the ransom. Doing so will result in you losing your money* and *getting nothing in return. The attackers do not monitor these payments, nor provide a decryption tool. Restoring from backups is your only hope at recovery.*

The ransomware payload

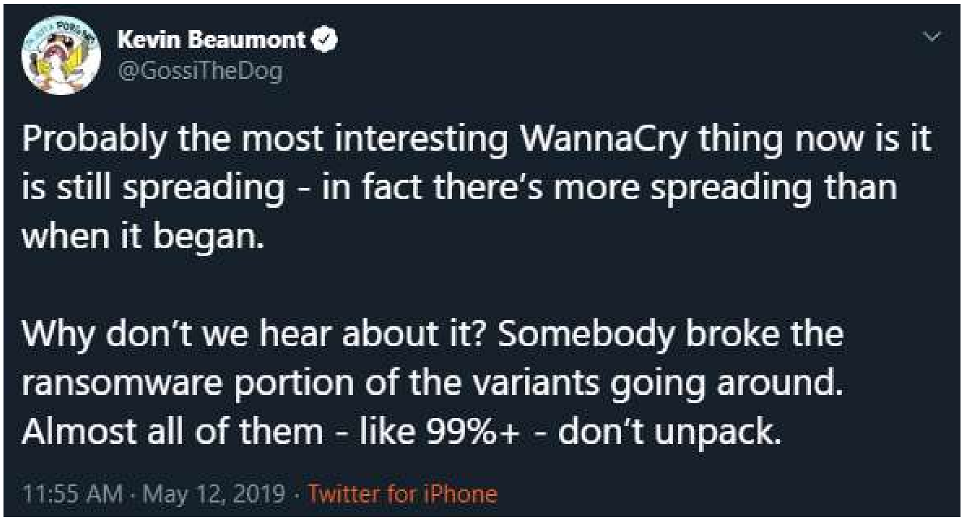
Now onto our third question: why is nobody complaining about WannaCry encrypting their files?

Using the same 2,725 samples, we wanted to understand if they had been altered in any other way. We executed a random selection of samples, including the top ten that were most prevalent on unprotected computers. In each case, no files were encrypted, and no ransom notes were created.

To understand why, you need to look at how WannaCry works. There is one component that spreads the malware to other machines, and then there is a separate component that does the encryption. This second component is contained in a password protected ZIP archive. The contents of the ZIP archive are extracted to the computer and then used to execute the ransomware attack.

In all 2,725 samples, the ZIP archive was corrupt. Errors appeared after only a few files had been extracted from the contents. This was the discovery we were looking for; everything now made sense: the large volume of detections were due to the lack of a kill switch, with nobody complaining about encrypted files because almost every sample seen in the wild had a corrupt archive that doesn’t encrypt anything.

Sophos researchers were not the only ones to spot this. In May, 2019, researcher Kevin Beaumont tweeted the following:



Kevin Beaumont
@GossiTheDog

Probably the most interesting WannaCry thing now is it is still spreading - in fact there's more spreading than when it began.

Why don't we hear about it? Somebody broke the ransomware portion of the variants going around. Almost all of them - like 99%+ - don't unpack.

11:55 AM · May 12, 2019 · Twitter for iPhone

Dawn of the broken payload

We can, however, track this discovery back even further, to just two days after the original attack.

On May 14, 2017, researchers at Kaspersky discovered a variant of WannaCry that had been uploaded to VirusTotal earlier that day. They shared the sample with researcher Matt Suiche, and in a blog post that same day he confirmed that the sample did not have a kill switch, and that the archive was corrupt. It was also noted that while the sample had been uploaded to VirusTotal, it had not been seen in the wild. This sample led to our final discovery.

The MD5 hash of the file uploaded to VirusTotal, which doesn't have a kill switch and doesn't encrypt files, is none other than the exact same file we now see causing the highest number of WannaCry detections:

MD5: d724d8cc6420f06e8a48752f0da11c66

It is number one on the of unique file variants list provided earlier, causing 29% of all WannaCry detections in our data. Even more amazing is that the top three files on our list are all variants of this same file. The other two files contain the same corrupt archive; the only difference is in how the kill switch has been removed.

The hangover

We now know that the WannaCry variants we see spreading in the wild today are an evolved version of the original. Without the kill switch they spread more effectively, and with no encryption they stay hidden on a network, drawing little attention from users or admins (who are, hopefully, busy improving their patch management process).

Infected computers benefit, slightly, from a feature of the malware that seeks to avoid duplication of effort: if a computer targeted for infection by a "potent" version of WannaCry has already been infected with a corrupted version of WannaCry, the dangerous version ignores the infected computer, and moves on to the next victim.

Of course, a computer infected with any type of WannaCry should be a cause for great concern, as an indication of how desperately out of date that computer is.

Recommendations and advice

The most important advice we can share is patch your computers, all of them. Do it now!

You can use the instructions in the following article to check if your computer is patched against EternalBlue: How to Verify if a Machine is Vulnerable to EternalBlue - MS17-010.

If you are a Sophos customer and have seen detections for CXmal/Wanna-A or some of the related detections, such as Troj/Ransom-EMG or HPMal/Wanna-A, please follow the instructions in How to resolve multiple detections for CXmal/Wanna-A, Troj/Ransom-EMG, HPMal/Wanna-A.

SophosLabs has published a list of Indicators of Compromise (IoCs) relating to this research on our Github page, at https://github.com/sophoslabs/IoCs

Acknowledgments

The author wishes to thank SophosLabs researchers Fraser Howard and Anton Kalinin for their assistance with the research and technical guidance for this report.