How to Accidentally Stop a WannaCry Global Cyber Attack

Autor:
Data de publicació: 15-05-2017

How to Accidentally Stop a Global Cyber Attacks
May 13, 2017 MalwareTech ms17-010, ransowmare, worm 291

So finally I've found enough time between emails and Skype calls to write up on the crazy events which occurred over Friday, which was supposed to be part of my week off (I made it a total of 4 days without working, so there's that). You've probably read about the WannaCrypt fiasco on several news sites, but I figured I'd tell my story.

I woke up at around 10 AM and checked onto the UK cyber threat sharing platform where i had been following the spread of the Emotet banking malware, something which seemed incredibly significant until today. There were a few of your usual posts about various organisations being hit with ransomware, but nothing significant…yet. I ended up going out to lunch with a friend, meanwhile the WannaCrypt ransomware campaign had entered full swing.

When I returned home at about 2:30, the threat sharing platform was flooded with posts about various NHS systems all across the country being hit, which was what tipped me of to the fact this was something big. Although ransomware on a public sector system isn't even newsworthy, systems being hit simultaneously across the country is (contrary to popular belief, most NHS employees don't open phishing emails which suggested that something to be this widespread it would have to be propagated using another method). I was quickly able to get a sample of the malware with the help of Kafeine, a good friend and fellow researcher. Upon running the sample in my analysis environment I instantly noticed it queried an unregistered domain, which i promptly registered.

Using Cisco Umbrella, we can actually see query volume to the domain prior to my registration of it which shows the campaign started at around 8 AM UTC.

While the domain was propagating, I ran the sample again in my virtual environment to be met with WannaCrypt ransom page; but more interestingly was that after encrypting the fake files I left there as a test, it started connecting out to random IP addresses on port 445 (used by SMB). The mass connection attempts immediately made me think exploit scanner, and the fact it was scanning on the SMB port caused me to look back to the recent ShadowBroker leak of NSA exploits containing….an SMB exploit. Obvious I had no evidence yet that it was definitely scanning SMB hosts or using the leaked NSA exploit, so I tweeted out my finding and went to tend to the now propagated domain.

Now one thing that's important to note is the actual registration of the domain was not on a whim. My job is to look for ways we can track and potentially stop botnets (and other kinds of malware), so I'm always on the lookout to pick up unregistered malware control server (C2) domains. In fact I registered several thousand of such domains in the past year.

Our standard model goes something like this.

Look for unregistered or expired C2 domains belonging to active botnets and point it to our sinkhole (a sinkhole is a server designed to capture malicious traffic and prevent control of infected computers by the criminals who infected them).
Gather data on the geographical distribution and scale of the infections, including IP addresses, which can be used to notify victims that they're infected and assist law enforcement.
Reverse engineer the malware and see if there are any vulnerabilities in the code which would allow us to take-over the malware/botnet and prevent the spread or malicious use, via the domain we registered.

In the case of WannaCrypt, step 1, 2 and 3 were all one and the same, I just didn't know it yet.

A few seconds after the domain had gone live I received a DM from a Talos analyst asking for the sample I had which was scanning SMB host, which i provided. Humorously at this point we had unknowingly killed the malware so there was much confusion as to why he could not run the exact same sample I just ran and get any results at all. As curious as this was, I was pressed for time and wasn't able to investigate, because now the sinkhole servers were coming dangerously close to their maximum load.

I set about making sure our sinkhole server were stable and getting the expected data from the domain we had registered (at this point we still didn't know much about what the domain I registered was for, just that anyone infected with this malware would connect to the domain we now own, allowing us to track the spread of the infection). Sorting out the sinkholes took longer than expected due to a very large botnet we had sinkholed the previous week eating up all the bandwidth, but soon enough I was able to set up a live tracking map and push it out via twitter (you can still see it here).

Around 6:23 PM (BST) I asked an employee to look into the worm code and verify the domain we registered would not change (some malware will periodically change the domain using an algorithm, so we needed to know if there would be new domains so we could register those too), meanwhile I performed some updated to the live map to deal with the rapid influx of new visitors.

After about 5 minutes the employee came back with the news that the registration of the domain had triggered the ransomware meaning we'd encrypted everyone's files (don't worry, this was later proven to not be the case), but it still caused quite a bit of panic. I contacted Kafeine about this and he linked me to the following freshly posted tweet made by ProofPoint researcher Darien Huss, who stated the opposite (that our registration of the domain had actually stopped the ransomware and prevent the spread).

Having heard to conflicting answers, I anxiously loaded back up my analysis environment and ran the sample….nothing. I then modified my host file so that the domain connection would be unsuccessful and ran it again…..RANSOMWARED.

Now you probably can't picture a grown man jumping around with the excitement of having just been ransomwared, but this was me. The failure of the ransomware to run the first time and then the subsequent success on the second mean that we had in fact prevented the spread of the ransomware and prevented it ransoming any new computer since the registration of the domain (I initially kept quiet about this while i reverse engineered the code myself to triple check this was the case, but by now Darien's tweet had gotten a lot of traction).

So why did our sinkhole cause an international ransomware epidemic to stop?

Talos wrote a great writeup explaining the code side here, which I'll elaborate on using Darien's screenshot.

All this code is doing is attempting to connect to the domain we registered and if the connection is not successful it ransoms the system, if it is successful the malware exits (this was not clear to me at first from the screenshot as I lacked the context of what the parent function may be doing with the results).

The reason which was suggested is that the domain is a "kill switch" in case something goes wrong, but I now believe it to be a badly thought out anti-analysis.

In certain sandbox environments traffic is intercepted by replying to all URL lookups with an IP address belonging to the sandbox rather than the real IP address the URL points to, a side effect of this is if an unregistered domain is queried it will respond as it it were registered (which should never happen).

I believe they were trying to query an intentionally unregistered domain which would appear registered in certain sandbox environments, then once they see the domain responding, they know they're in a sandbox the malware exits to prevent further analysis. This technique isn't unprecedented and is actually used by the Necurs trojan (they will query 5 totally random domains and if they all return the same IP, it will exit); however, because WannaCrypt used a single hardcoded domain, my registartion of it caused all infections globally to believe they were inside a sandbox and exit…thus we initially unintentionally prevented the spread and and further ransoming of computers infected with this malware. Of course now that we are aware of this, we will continue to host the domain to prevent any further infections from this sample.

One thing that is very important to note is our sinkholing only stops this sample and there is nothing stopping them removing the domain check and trying again, so it's incredibly importiant that any unpatched systems are patched as quickly as possible.

As well as the names & companies mentioned in this blog I'd like to give a shout out to:

NCSC UK – Their threat intelligence sharing program provided us with valuable information needed to first identify the malware family behind the attack. They also helped ensure our sinkholes were not mistaken for criminal controlled infrastructure so that we could feed them the information required to notify UK victims.

FBI & ShadowServer – They were a great help in getting non-UK victims notified of the infections in a very short span of time, even if it did mean me staying up all night to link in with them.

2sec4u – For reducing my workload today and providing free panic attacks.

Microsoft – By realeasing an out of bounds patch for unsupported operating systems such as Windows XP and Server 2003, people now are able to patch rather than having to attempt upgrades to newer system in order to be secured against this worm.

If you have anything to patch, patch it. If you need a guide, this one is being reguarly updated: https://www.ncsc.gov.uk/guidance/protecting-your-organisation-ransomware

Now I should probably sleep.

Guidelines for blocking specific firewall ports to prevent SMB traffic from leaving the corporate environment

? Email
? Print

Summary

Malicious users can use the Server Message Block (SMB) protocol for malicious purposes.

Firewall best practices and firewall configurations can enhance network security by helping to prevent potentially malicious traffic from crossing the enterprise perimeter.

Enterprise perimeter firewalls should block unsolicited communication (from the Internet) and outgoing traffic (to the Internet) to the following SMB-associated ports:

137
138
139
445

More Information

These ports can be used to initiate a connection with a potentially malicious Internet-based SMB server. SMB traffic should be restricted to private networks or virtual private networks (VPNs).

Suggestion
Blocking these ports at the enterprise edge or perimeter firewall helps protect systems that are behind that firewall from attempts to leverage SMB for malicious purposes. Organizations can allow port 445 access to specific Azure Datacenter IP ranges (see the following reference) to enable hybrid scenarios where on-premises clients (behind an enterprise firewall) use the SMB port to talk to Azure file storage.

Approaches
Perimeter firewalls typically use “Block listing” or “Approved listing” rule methodologies, or both.

Block listing
Allow traffic unless a deny (block listed) rule prevents it.

Example 1
Allow all
Deny 137 name services
Deny 138 datagram services
Deny 139 session service
Deny 445 session service

Approved listing
Deny traffic unless an allow rule allows it.

To help prevent attacks that may use other ports, we recommend that you block all unsolicited communication from the Internet. We suggest a blanket deny, with allow rule exceptions (approved listing).

Note The approved listing method in this section blocks NetBIOS and SMB traffic implicitly by not including an allow rule.

Example 2
Deny all
Allow 53 DNS
Allow 21 FTP
Allow 80 HTTP
Allow 443 HTTPS
Allow 143 IMAP
Allow 123 NTP
Allow 110 POP3
Allow 25 SMTP

The list of allow ports is not exhaustive. Depending on corporate needs, additional firewall entries may be needed.

Impact of workaround
Several Windows services use the affected ports. Blocking connectivity to the ports may prevent various applications or services from functioning. Some of the applications or services that could be affected include the following:

Applications that use SMB (CIFS)
Applications that use mailslots or named pipes (RPC over SMB)
Server (file and print sharing)
Group Policy
Net Logon
Distributed File System (DFS)

Terminal server licensing
Print spooler
Computer browser
Remote procedure call locator
Fax service
Indexing service
Performance logs and alerts
Systems Management Server
License logging service

How to undo the workaround
Unblock the ports at the firewall. For more information about ports, see TCP and UDP port assignments.

References
Azure remote apps https://azure.microsoft.com/en-us/documentation/articles/remoteapp-ports/

Azure datacenter IPs http://go.microsoft.com/fwlink/?LinkId=825637

Microsoft Office https://support.office.com/en-us/article/Office-365-URLs-and-IP-address-ranges-8548a211-3fe7-47cb-abb1-355ea5aa88a2

Properties
Article ID: 3185535 - Last Review: Aug 31, 2016 - Revision: 1

Applies to
Windows 10, Windows 10 Version 1511, Windows 10 Version 1607, Windows Server 2012 R2 Datacenter, Windows Server 2012 R2 Standard, Windows Server 2012 R2 Essentials, Windows Server 2012 R2 Foundation, Windows 8.1 Enterprise, Windows 8.1 Pro, Windows 8.1, Windows RT 8.1, Windows Server 2012 Datacenter, Windows Server 2012 Datacenter, Windows Server 2012 Datacenter, Windows Server 2012 Datacenter, Windows Server 2012 Standard, Windows Server 2012 Standard, Windows Server 2012 Standard, Windows Server 2012 Standard, Windows Server 2012 Essentials, Windows Server 2012 Foundation, Windows Server 2012 Foundation, Windows Server 2012 Foundation, Windows Server 2012 Foundation, Windows Server 2008 R2 Service Pack 1, Windows Server 2008 R2 Datacenter, Windows Server 2008 R2 Enterprise, Windows Server 2008 R2 Standard, Windows Web Server 2008 R2, Windows Server 2008 R2 Foundation, Windows 7 Service Pack 1, Windows 7 Ultimate, Windows 7 Enterprise, Windows 7 Professional, Windows 7 Home Premium, Windows 7 Home Basic, Windows 7 Starter, Windows Server 2008 Service Pack 2, Windows Server 2008 Datacenter, Windows Server 2008 Enterprise, Windows Server 2008 Standard, Windows Web Server 2008, Windows Server 2008 Foundation, Windows Server 2008 for Itanium-Based Systems, Windows Vista Service Pack 2, Windows Vista Ultimate, Windows Vista Enterprise, Windows Vista Business, Windows Vista Home Premium, Windows Vista Home Basic, Windows Vista Starter