

Hunting Naked and Afraid: Shaking the Stick at Cobalt Strike

Michael Haag

Senior Threat Researcher | Splunk

Jose Hernandez

Sr Manager, Threat Researcher | Splunk





Michael Haag

Splunk Threat Research Team

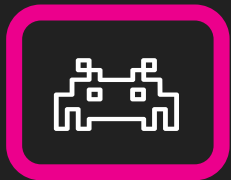


Jose Hernandez

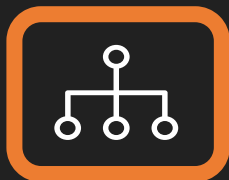
Splunk Threat Research Team

Splunk Threat Research Team (STRT)

**Study
Threats**



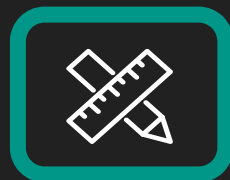
**Create
Datasets**



**Build
Detections**



**Release
Tools**



**Share with
Community**



Agenda

- What is Cobalt Strike
- Why do we hunt it
- Ways to catch a Team Server
- Architecture of Scanning Tool
- Demo
- Detection Development



What is Cobalt Strike

Software for
Adversary Simulations and **Red Team Operations**

[DOWNLOAD](#) [BUY NOW](#)





Adversary Simulations and Red Team Operations are security assessments that replicate the tactics and techniques of an advanced adversary in a network. While penetration tests focus on unpatched vulnerabilities and misconfigurations, these assessments benefit security operations and incident response.

Groups That Use This Software

ID	Name	References
G0079	DarkHydrus	[9][10]
G0073	APT19	[11]
G0037	FIN6	[12]
G0052	CopyKittens	[13]
G0065	Leviathan	[14][15]
G0050	APT32	[16][17][18][19][20][7]
G0096	APT41	[21]
G0016	APT29	[22][23]
G0114	Chimera	[24][25]
G0080	Cobalt Group	[26][27][28][29] [30][31][32][33]
G0102	Wizard Spider	[34][35][36][37][38][39]
G0129	Mustang Panda	[40][41][42][43][44]

Why Hunt for Cobalt Strike

Benefits for a SOC?

- Understanding adversary TTPs
- Keeping pace with tool changes
- Build better analytics

<https://www.darkreading.com/attacks-breaches/d-id>

Cobalt Strike Becomes a Preferred Hacking Tool by ... - Dark ...

May 19, 2021 — Cobalt Strike Becomes a Preferred Hacking Tool by Cybercrime, APT Groups
... Cobalt Strike is one of the top five tools used by attackers.

Cobalt Strike Usage Explodes Among Cybercrooks | Threatpost

Jun 29, 2021 — Cobalt Strike Usage Explodes Among Cybercrooks | Threatpost

<https://www.cybersecuritydive.com/news/cobalt-strike...>

Cobalt Strike rising in prominence among criminal threat actors

Jul 1, 2021 — Malicious actors have increased the use of Cobalt Strike 161% between 2019 and 2020, Proofpoint researchers found. The Cobalt Strike Research

<https://www.proofpoint.com/blog/threat-insight/co...>

Cobalt Strike: Favorite Tool from APT to Crimeware - Proofpoint

Jun 29, 2021 — Cobalt Strike is a legitimate security tool used by penetration testers to emulate threat actor activity in a network. However, it is also ...

Hunting The Cobalt Strike



Scanning the entire Internet

Approaches to Hunting

Pros

Find (realtime) active Cobalt Team servers

Not reliant on services (Shodan/Security Trails/ZoomEye)

Collect WTV you like

Cons

Get in the business of scanning the internet

Finding a “friendly” bulletproof provider

Wiring up zmap/masscan

Slow 🐌

Using 3rd Parties that Scan the Internet

Approaches to Hunting

Pros

Everything from last con slide * -1

Confirmed active data from the last month

Enriched data OOB

Cons

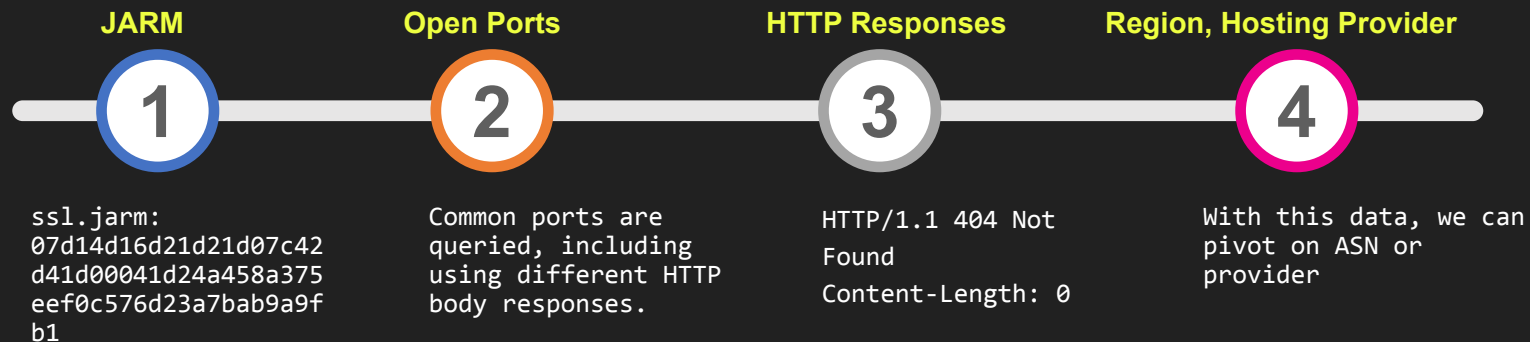
Money 💰💰💰💰

Reliant on “last scanned”

Inactive Team Servers

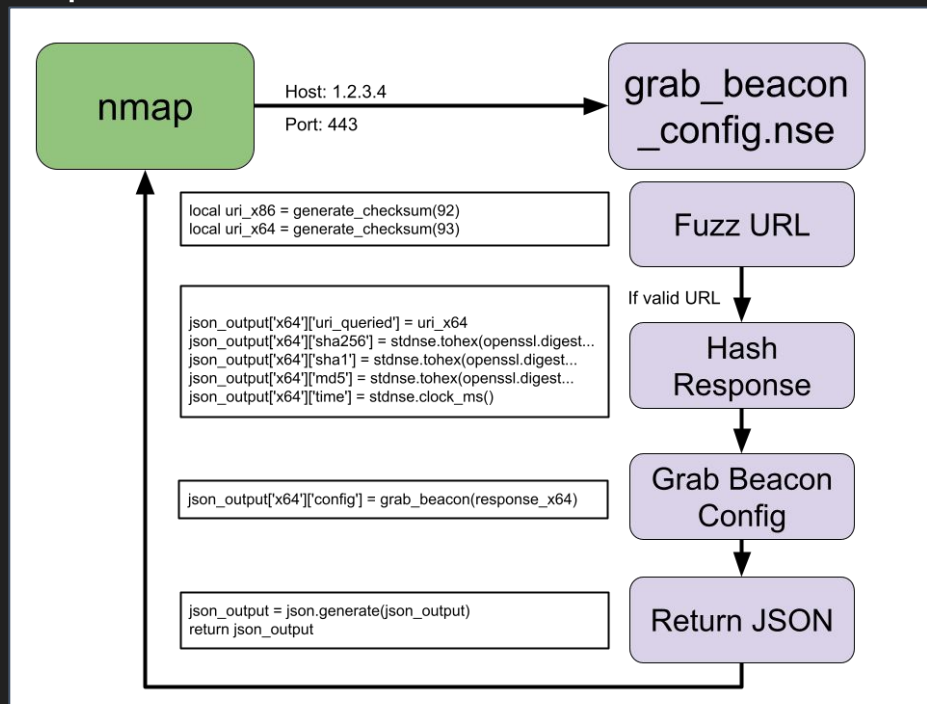
Inaccurate results at time

Hunting Methods



How do you coerce a Team Server

How the NSE script works



Hunting Methods - Output

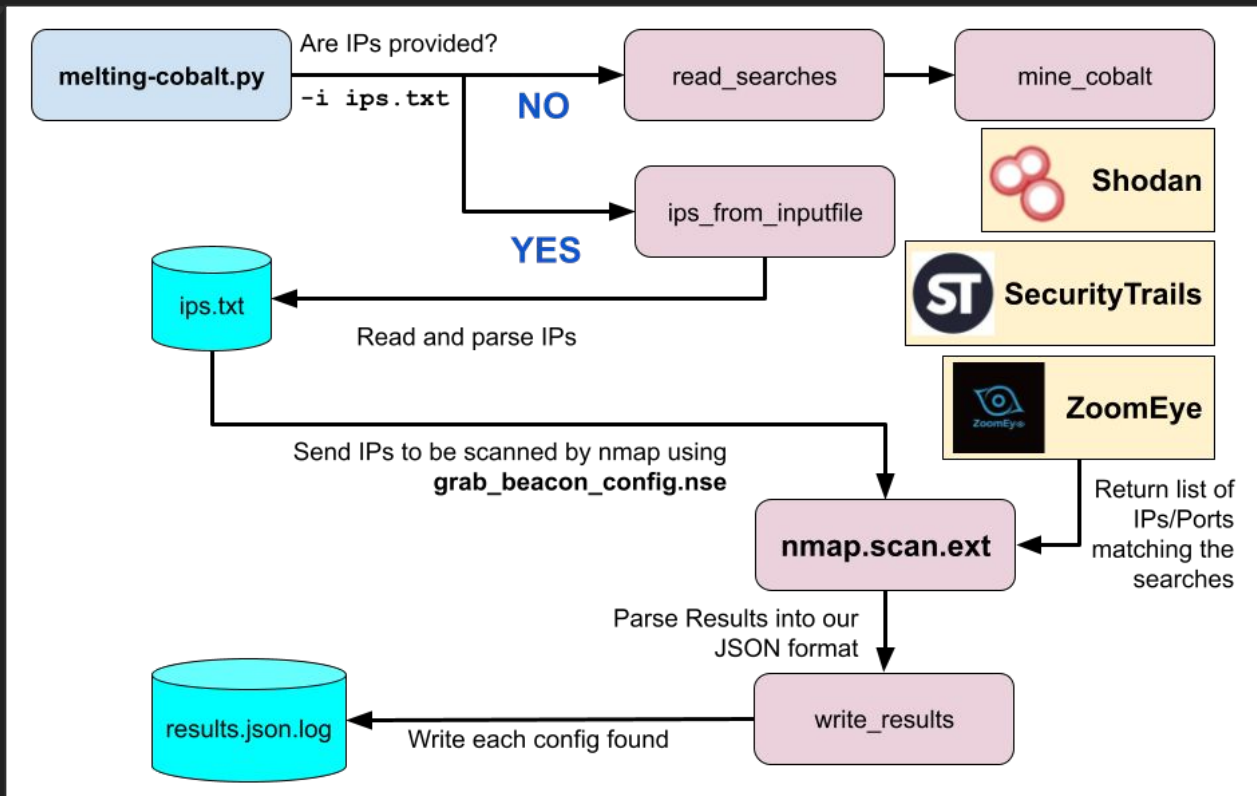
This is what you get

Most settings of Cobalt Strike may be configured with Malleable Profiles

```
dns_idle: 0.0.0.0
dns_sleep: 0
hostnames: null
ip: 45.32.47.23
max_dns: 255
nmap_cmd: /usr/bin/nmap -p 443 --script /home/ubuntu/cobalt-pickaxe/grab_beacon_config.nse -vv -d -n -T5 -oX - 45.32.47.23
port: 443
protocol: tcp
service: https
timestamp: 1627801222.744879
user_agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; WOW64; Trident/6.0; ASU2JS)
watermark: 0
x64_config_beacon_type: 8 (HTTPS)
x64_config_c2_server: [ [-]
  45.32.47.23/cm
]
x64_config_http_method_path_2: /submit.php
x64_config_jitter: 0
x64_config_method_1: GET
x64_config_method_2: POST
x64_config_polling: 60000
x64_config_port: 443
x64_config_spawn_to_x64: %windir%\sysnative\rundll32.exe
x64_config_spawn_to_x86: %windir%\syswow64\rundll32.exe
x64_md5: 9e15f00a05860a9262c7a08592f0faf5
x64_sha1: 94b09f122248f7eaf6386de8542dcc251a0e1a5f
x64_sha256: 15015e7a4d8a7d847dfe64403f08176c17ef380dd773c77c62a40072e898681a
x64_uri_queried: /8pkJ
```

Splunk's Melting-Cobalt Architecture

Melting C



Demo



Detection Development 🛡️

Building Detections

Simulate



Simulate Cobalt Strike
activity
Capture Profiles

Review



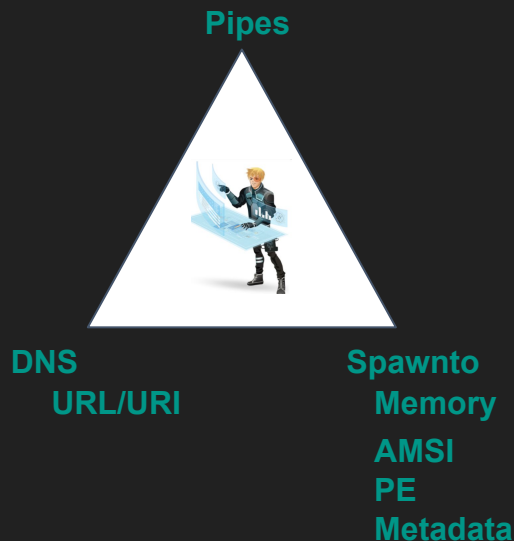
Review data for patterns
or behaviors

Generate



Write detections

Defenders need to watch every angle ▲



- Understand Malleable Profiles
 - <https://www.cobaltstrike.com/help-malleable-c2/>
 - <https://github.com/threatexpress>
- What products in your stack provide visibility?
 - Endpoint, Network
- Familiarize yourself with the data
 - https://github.com/splunk/attack_data/blob/master/datasets/attack_techniques/T1055/cobalt_strike/cobalt_strike.yml
 - https://github.com/splunk/attack_data/blob/master/datasets/attack_techniques/T1572/cobalt_strike/cobalt_strike.yml

Malleable Profiles

- <https://github.com/MHaggis/notes>
<https://github.com/threatexpress/malleable-c2>

```
310 post-ex {
311     # Optionally specify non-existent filepath to force manual s
312     set spawn_to_x86 "%windir%\syswow64\dlhost.exe";
313     # Hardcode paths like C:\Windows\System32\dlhost.exe to
314     set spawn_to_x64 "%windir%\sysnative\dlhost.exe"
315     # change the permissions and content of our post-ex DLLs
316     set obfuscate "true";
317     # pass key function pointers from Beacon to its child jobs
318     set smartinject "true";
319     # disable AMSI in powerpick, execute-assembly, and psinject
320     set amsi_disable "true";
321     # Modify our post-ex pipe names
322     set pipe_name "Winsock2\CatalogChangeListener-###-0.";
323     set keylogger "GetAsyncKeyState";
324     #set threadhint "module!function+0x##"
325 }
```

```
190 dns-beacon {
191     # Options moved into "dns-beacon" group in version 4.3
192     set dns_idle "74.125.196.113"; #google.com (change this to match your campaign)
193     set dns_max_txt "252";
194     set dns_sleep "0"; # Force a sleep prior to each individual DNS request. (in milliseconds)
195     set dns_ttl "5";
196     set maxdns "255";
197     set dns_stager_prepend ".resources.123456.";
198     set dns_stager_subhost ".feeds.123456.";
199
200     # DNS subhosts override options, added in version 4.3
201     set beacon "a.bc.";
202     set get_A "b.1a.";
203     set get_AAAA "c.4a.";
204     set get_TXT "d.tx.";
205     set put_metadata "e.md.";
206     set put_output "f.po.";
207     set ns_response "zero";
```

Default Cobalt Strike Settings

Network

```
dns_idle 0.0.0.0
dns_max_txt 252
dns_sleep 0
dns_stager_subhost
.stage.123456.
dns_ttl 1
maxdns 255
get_A cdn.
get_AAAA www6.
get_TXT api.
put_metadata www.
put_output post.
ns_response drop
```

Pipes

```
msagent_##
status_##
postex_ssh_####
MSSE-###-server
status_##
postex_##
```

SpawnTo

```
rundll32.exe
```

Building Detections

Take what we see used in the wild... and find all the profiles on GitHub

Most common SpawnTo

```
"%windir%\explorer.exe"  
"%windir%\sysnative\<mfmp>.exe"  
"%windir%\sysnative\dlh.exe"  
"%windir%\sysnative\explorer.exe"  
"%windir%\sysnative\gprex.exe"  
"%windir%\sysnative\gpupdate.exe"  
"%windir%\sysnative\msupdate.exe"  
"%windir%\sysnative\notepad.exe"  
"%windir%\sysnative\reg.exe"  
"%windir%\sysnative\regsvr32.exe"  
"%windir%\sysnative\rundll32.exe"  
"%windir%\sysnative\svchost.exe -k netsvcs"  
"%windir%\sysnative\WerFault.exe"  
"%windir%\sysnative\wscript.exe"  
"%windir%\sysnative\WUAUCLT.exe"  
"%windir%\sysnative\wusa.exe"
```

<https://github.com/MHaggis/notes>

Pipes

```
rpc_##  
mojo.5688.8052.35780273329370473##  
halfduplex_##  
win_svc  
scerpc##  
scerpc_##  
scerpc  
spoolss_##  
rpc_##  
mypipe-h##  
mojo.5688.8052.35780273329370473##  
demoagent_22  
windows.update.manager###  
f53f##
```

Cobalt-Strike Servers Found

23,848

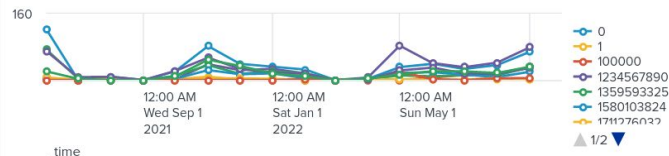
Unique CS Watermarks

288

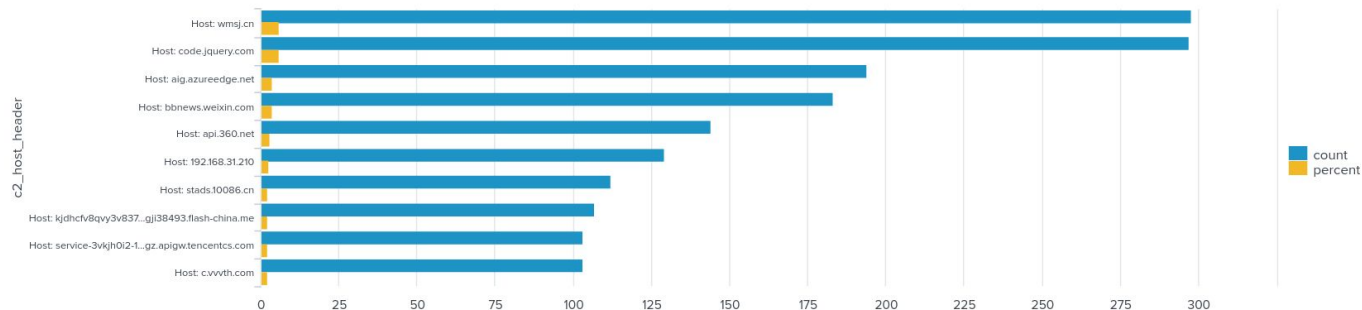
C2 Servers Found

253

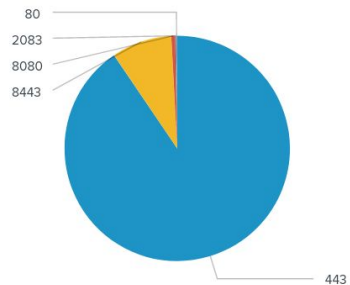
Watermarks



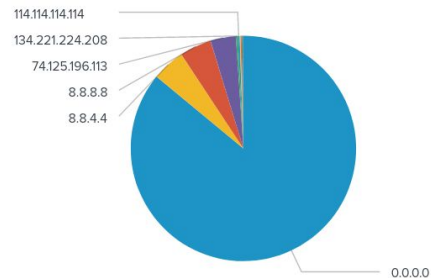
Top C2 Host Headers



Top Listening Ports



DNS Idle



SpawnTo

#	x64_config_spawn_to_x64	percent
1	%windir%\sysnative\rundll32.exe	85.165049
2	%windir%\sysnative\dlhhost.exe	3.106796
3	%windir%\sysnative\gpupdate.exe	1.786408
4	%windir%\sysnative\svchost.exe	1.048544
5	%windir%\sysnative\lmstsc.exe	1.048544
6	%windir%\sysnative\eventvwr.exe	0.893204
7	%windir%\sysnative\svchost.exe -k netsvcs	0.660194
8	%windir%\system32\rundll32.exe	0.504854
9	%windir%\sysnative\wusa.exe	0.504854
10	%windir%\sysnative\runonce.exe	0.504854

Rare SpawnTo

#	x64_config_spawn_to_x64	percent
1	%windir%\Microsoft.NET\Framework64\v4.0.30319\AppLaunch.exe	0.038835
2	%windir%\splwow64.exe	0.038835
3	%windir%\sysnative\adobe64.exe	0.038835
4	%windir%\sysnative\calc.exe	0.038835
5	%windir%\sysnative\cmstp.exe	0.038835
6	%windir%\sysnative\conhost.exe	0.038835
7	%windir%\sysnative\dlhhost.exe -o enable	0.038835
8	%windir%\sysnative\esentutl.exe	0.038835
9	%windir%\sysnative\expand.exe	0.038835
10	%windir%\sysnative\exthost.exe	0.038835

SpawnTo an Analytic

```
"%windir%\sysnative\rundll32.exe"
```

Alternative: <https://github.com/threatexpress/malleable-c2/blob/master/jquery-c2.4.6.profile>

Spawn Bacon.exe

Dump hashes

jump psexec64 <host2> http

shell whoami

parent_process_name ↕	process_name ↕	values(CommandLine) ^
explorer.exe	bacon.exe	"C:\Users\Administrator\Desktop\bacon.exe"
3c0545f.exe	rundll32.exe	C:\Windows\System32\rundll32.exe
svchost.exe	rundll32.exe	C:\Windows\System32\rundll32.exe shell32.dll,S
rundll32.exe	cmd.exe	C:\Windows\system32\cmd.exe /C whoami
bacon.exe	rundll32.exe	C:\Windows\system32\rundll32.exe

SpawnTo an Analytic

The obvious

- Cmd.exe /c spawn from non-standard process (baseline)
- Randomly generated process name, service
 - Service start (7045), admin\$, named pipe (MSSE-6944-server)
- Common processes that spawn rundll32
- Netconn to TeamServer
- Filemod
- registry

parent_process_name ↕	process_name ↕	values(CommandLine) ^
explorer.exe	bacon.exe	"C:\Users\Administrator\Desktop\bacon.exe"
3c0545f.exe	rundll32.exe	C:\Windows\System32\rundll32.exe
svchost.exe	rundll32.exe	C:\Windows\System32\rundll32.exe shell32.dll,S
rundll32.exe	cmd.exe	C:\Windows\system32\cmd.exe /C whoami
bacon.exe	rundll32.exe	C:\Windows\system32\rundll32.exe

SpawnTo an Analytic

The obvious

- Cmd.exe /c spawn from non-standard process (baseline)

index=win EventCode=7045 3c0545f.exe | [table](#) ImagePath ServiceName ServiceType

✓ 1 event (9/22/22 12:00:00.000 PM to 9/23/22 12:35:20.000 PM) No Event Sampling ▾

Job ▾

Events (1) Patterns Statistics (1) Visualization

20 Per Page ▾ [Format](#) Preview ▾

ImagePath ⇅	ServiceName ⇅	ServiceType ⇅
\\WIN-DC-MHAAG-AT\ADMIN\$\3c0545f.exe	3c0545f	user mode service

'sysmon' EventCode IN (12,13,14) 3c0545f.exe | [stats](#) values(registry_hive) by registry_key_name registry_value_name Details

✓ 1 event (9/22/22 12:00:00.000 PM to 9/23/22 12:33:21.000 PM) No Event Sampling ▾

Job ▾ || ▢ → 📄 ⬇

Events (1) Patterns Statistics (1) Visualization

20 Per Page ▾ [Format](#) Preview ▾

registry_key_name ⇅	registry_value_name ⇅	Details ⇅	values(registry_hive) ⇅
HKLM\System\CurrentControlSet\Services\3c0545f	ImagePath	\\WIN-DC-MHAAG-AT\ADMIN\$\3c0545f.exe	HKEY_LOCAL_MACHINE\System

SpawnTo an Analytic

- When this sacrificial process is spawned, there is typically never a command-line
- Default, netconn will egress from rundll32

parent_process_name ↕	process_name ↕	values(CommandLine) ^
explorer.exe	bacon.exe	"C:\Users\Administrator\Desktop\bacon.exe"
3c0545f.exe	rundll32.exe	C:\Windows\System32\rundll32.exe
svchost.exe	rundll32.exe	C:\Windows\System32\rundll32.exe shell32.dll,S
rundll32.exe	cmd.exe	C:\Windows\system32\cmd.exe /C whoami
bacon.exe	rundll32.exe	C:\Windows\system32\rundll32.exe

SpawnTo an Analytic

<https://research.splunk.com/endpoint/e451bd16-e4c5-4109-8eb1-c4c6ecf048b4/>

```
| tstats 'security_content_summariesonly' count FROM datamodel=Endpoint.Processes where 'process_rundll32' by _time span=1h Processes.process_id Processes.parent_process_name Processes.process_name Processes.dest Processes.process_path Processes.process  
| 'drop_dm_object_name(Processes)'  
| 'security_content_ctime(firstTime)'  
| 'security_content_ctime(lastTime)'  
| regex process="(?!)(rundll32\\.exe\\.){0,4}$"
```

✓ 156 events (9/23/22 11:17:00.000 AM to 9/23/22 12:17:16.000 PM) No Event Sampling ▼

Job ▼ || ■ ↻ ⌵

Events (156) Patterns **Statistics (8)** Visualization

20 Per Page ▼ / Format Preview ▼

_time ↕	process_id ↕ /	parent_process_name ^ /	process_name ↕ /	dest ↕	process_path ↕	process ↕
2022-09-23 11:00	0x16c8	3c0545f.exe	rundll32.exe	win-dc-mhaag-attack-range-622.attackrange.local	C:\Windows\System32\rundll32.exe	C:\Windows\System32\rundll32.exe
2022-09-23 11:00	5832	3c0545f.exe	rundll32.exe	win-dc-mhaag-attack-range-622.attackrange.local	C:\Windows\System32\rundll32.exe	C:\Windows\System32\rundll32.exe
2022-09-23 11:00	0x1a44	bacon.exe	rundll32.exe	win-dc-mhaag-attack-range-622.attackrange.local	C:\Windows\System32\rundll32.exe	C:\Windows\system32\rundll32.exe

SpawnTo an Analytic

<https://research.splunk.com/endpoint/35307032-a12d-11eb-835f-acde48001122/>

```
| tstats 'security_content_summariesonly' count FROM datamodel=Endpoint.Processes where 'process_rundll32' by _time span=1h Processes.process_id Processes.process_name Processes.dest Processes.process_path Processes.process Processes
    .parent_process_name Processes.original_file_name
| 'drop_dm_object_name(Processes)'
| 'security_content_ctime(firstTime)'
| 'security_content_ctime(lastTime)'
| regex process="(?!)(rundll32\\.exe\\.){0,4}$"
| join process_id [
| tstats 'security_content_summariesonly' count FROM datamodel=Network_Traffic.All_Traffic where All_Traffic.dest_port != 0 by All_Traffic.process_id All_Traffic.dest All_Traffic.dest_port
| 'drop_dm_object_name(All_Traffic)'
| rename dest as C2 ]
| table _time dest parent_process_name process_name process_path process_id dest_port C2
```

Last 60 minutes

✓ 132 events (9/23/22 11:08:00.000 AM to 9/23/22 12:08:44.000 PM) No Event Sampling ▾ Job ▾ || ▢ ↶ ⚙ ⬇ 🗨 Verbose Mode ▾

Events (132) Patterns Statistics (1) Visualization

20 Per Page ▾ ↗ Format Preview ▾

_time ⚙	dest ⚙	parent_process_name ⚙	process_name ⚙	process_path ⚙	process ⚙	process_id ⚙	dest_port ⚙	C2 ⚙
2022-09-23 11:00:00	win-dc-mhaag-attack-range-622.attackrange.local	3c0545f.exe	rundll32.exe	C:\Windows\System32\rundll132.exe	C:\Windows\System32\rundll132.exe	5832	80	54.188.59.68

Pipes Gone Wrong

Default Pipes

- \postex_*
- \postex_ssh_*
- \status_*
- \msagent_*
- \MSSE-*
- *-server

ITW Pipes

- rpc_##
- mojo.5688.8052.35780273329370473##
- halfduplex_##
- win_svc
- scerpc##
- scerpc_##
- scerpc
- spoolss_##
- rpc_##
- mypipe-h##
- mojo.5688.8052.35780273329370473##
- demoagent_22
- windows.update.manager###
- f53f###

Pipes Gone Wrong

```
`sysmon` EventID=17 OR EventID=18 PipeName IN (\\msagent_*, \\DserNamePipe*, \\srvsvc_*, \\postex_*, \\status_*, \\MSSE-*, \\spoolss_*, \\win_svc*, \\ntsvcs*, \\winsock*, \\UIA_PIPE*)  
| stats count min(_time) as firstTime max(_time) as lastTime by Computer, process_name, process_id process_path, PipeName  
| rename Computer as dest  
| `security_content_ctime(firstTime)`  
| `security_content_ctime(lastTime)`
```

✓ 6 events (9/23/22 11:21:00.000 AM to 9/23/22 12:21:37.000 PM) No Event Sampling ▼

Events (6) Patterns Statistics (4) Visualization

20 Per Page ▼ Format Preview ▼

dest ↕	process_name ↕	process_id ↕	process_path ↕	PipeName ↕	count ↕
win-dc-mhaag-attack-range-622.attackrange.local	3c0545f.exe	5876	\\WIN-DC-MHAAG-AT\\ADMIN\$\\3c0545f.exe	\\MSSE-6944-server	2
win-dc-mhaag-attack-range-622.attackrange.local	bacon.exe	6180	C:\\Users\\Administrator\\Desktop\\bacon.exe	\\MSSE-7439-server	2
win-dc-mhaag-attack-range-622.attackrange.local	bacon.exe	6180	C:\\Users\\Administrator\\Desktop\\bacon.exe	\\postex_cff0	1
win-dc-mhaag-attack-range-622.attackrange.local	rundl132.exe	6724	C:\\Windows\\system32\\rundl132.exe	\\postex_cff0	1

Pipes Gone Wrong

Buyer Beware

These pipes are *noisy*

ITW Pipes

- rpc_##
- mojo.5688.8052.35780273329370473##
- halfduplex_##
- win_svc
- scerpc##
- scerpc_##
- scerpc
- spoolss_##
- rpc_##
- mypipe-h##
- mojo.5688.8052.35780273329370473##
- demoagent_22
- windows.update.manager###
- f53f##

Pipes Gone Wrong

Buyer Beware

These pipes are *noisy*

ITW Pipes
rpc ##

process_name ↕	process_id ↕	process_path ↕	PipeName ↕	count ▼
svchost.exe	1292	C:\Windows\system32\svchost.exe	\scerpc	7
3c0545f.exe	5876	\\WIN-DC-MHAAG-AT\ADMIN\$\3c0545f.exe	\MSSE-6944-server	2
bacon.exe	6180	C:\Users\Administrator\Desktop\bacon.exe	\MSSE-7439-server	2
bacon.exe	6180	C:\Users\Administrator\Desktop\bacon.exe	\postex_cff0	1
explorer.exe	4236	C:\Windows\explorer.exe	\UIA_PIPE_4236_00003759	1
rundll32.exe	6724	C:\Windows\system32\rundll32.exe	\postex_cff0	1

f53f##

Pipes Gone Wrong

Buyer Beware

These pipes are *noisy*

ITW Pipes

- rpc_##

PipeName ↕	count ↕
<Anonymous Pipe>	2634
\wkssvc	429
\srvsvc	426
\lsass	27
\PSHost.133084089897848585.6100.DefaultAppDomain.powershell	1
\PSHost.133084089929706886.3792.DefaultAppDomain.powershell	1
\PSHost.133084089942944358.7148.DefaultAppDomain.powershell	1
\PSHost.133084089953279572.3244.DefaultAppDomain.powershell	1
\PSHost.133084089956903808.7968.DefaultAppDomain.powershell	1
\PSHost.133084089960046524.6812.DefaultAppDomain.powershell	1
\PSHost.133084089963287261.3640.DefaultAppDomain.powershell	1
\PSHost.133084089966360536.5680.DefaultAppDomain.powershell	1
\PSHost.133084089969639865.1892.DefaultAppDomain.powershell	1
\PSHost.133084089976180598.5176.DefaultAppDomain.powershell	1

DNS Beaconsing

Option	Default Value	Example	Description
dns_idle	0.0.0.0	1.2.3.4	IP address used to indicate no tasks are available to DNS Beacon; Mask for other DNS C2 values
dns_max_txt	252	199	Maximum length of DNS TXT responses for tasks
dns_sleep	0	1	Force a sleep prior to each individual DNS request. (in milliseconds)
dns_stager_prepend		doc-stg-prepend	Prepend text to payload stage delivered to DNS TXT record stager
dns_stager_subhost	.stage.123456.	doc-stg-sh.	Subdomain used by DNS TXT record stager.
dns_ttl	1	5	TTL for DNS replies
maxdns	255	200	Maximum length of hostname when uploading data over DNS (0-255)
beacon		doc.bc.	DNS subhost prefix used for beaconing requests
get_A	cdn.	doc.1a.	DNS subhost prefix used for A record requests
get_AAAA	www6.	doc.4a.	DNS subhost prefix used for AAAA record requests
get_TXT	api.	doc.tx.	DNS subhost prefix used for TXT record requests
put_metadata	www.	doc.md.	DNS subhost prefix used for metadata requests
put_output	post.	doc.po.	DNS subhost prefix used for output requests
ns_response	drop	zero	How to process NS Record requests. "drop" does not respond to the request (default), "idle" responds with A record for IP address from "dns_idle", "zero" responds with A record for 0.0.0.0

DNS Beaconsing

DNS Beaconsing

There are default patterns:

- Default beacons to 0.0.0.0 until IP
- Queries start with:
 - dns_idle 0.0.0.0
 - get_A cdn.
 - get_AAAA www6.
 - get_TXT api.
 - put_metadata www.
 - put_output post.

TXT records are *noisy*

Latest JQuery Profile

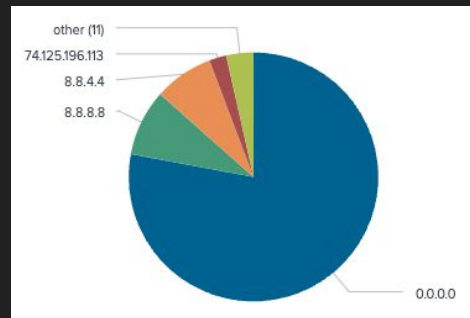
```
set dns_stager_prepend ".resources.123456.";
set dns_stager_subhost ".feeds.123456.";

# DNS subhosts override options, added in version 4.3
set beacon "a.bc.";
set get_A "b.1a.";
set get_AAAA "c.4a.";
set get_TXT "d.tx.";
set put_metadata "e.md.";
set put_output "f.po.";
set ns_response "zero";
```

DNS Beaconsing

DNS Idle

```
.getbobspizza.com", "rrtype": "A", "ttl": 1, "rdata": "103.123.181.80"}], "grouped": {"A": ["103.123.181.80"]}}  
  
ns.getbobspizza.com", "rrtype": "A", "ttl": 1, "rdata": "45.149.86.12"}], "grouped": {"A": ["45.149.86.12"]}}  
  
ns.getbobspizza.com", "rrtype": "A", "ttl": 1, "rdata": "73.115.9.240"}], "grouped": {"A": ["73.115.9.240"]}}  
  
eepdish.dns.getbobspizza.com", "rrtype": "A", "ttl": 1, "rdata": "0.0.0.0"}], "grouped": {"A": ["0.0.0.0"]}}  
  
c6.57cc0fb6.deepdish.dns.getbobspizza.com", "rrtype": "A", "ttl": 1, "rdata": "0.0.0.0"}], "grouped": {"A": ["0.0.0.0"]}}  
  
c6.57cc0fb6.deepdish.dns.getbobspizza.com", "rrtype": "A", "ttl": 1, "rdata": "0.0.0.0"}], "grouped": {"A": ["0.0.0.0"]}}
```



DNS Beaconing

- "api.017784db6.2d7e6eba.c2.dns.getbobspizza.com"
- "cdn.061362a57.57cc0fb6.deepdish.dns.getbobspizza.com"
- "post.2830be959d752afaa9039f60a589cff61fed5c8adff0a68b6.df60370897410454dbb991d3e3f2f1b8741dc56f7bc7d56d.17c2b2cc3.57cc0fb6.resolver.dns.getbobspizza.com"

values(dns.answers[].rrname) ^

```
api.011d032c9.57cc0fb6.resolver.dns.getbobspizza.com
api.012745cc4.57cc0fb6.resolver.dns.getbobspizza.com
api.0135c42c3.57cc0fb6.c2.dns.getbobspizza.com
api.013e25d77.57cc0fb6.c2.dns.getbobspizza.com
api.0142f2722.57cc0fb6.deepdish.dns.getbobspizza.com
api.015ad66e3.57cc0fb6.c2.dns.getbobspizza.com
api.016352b9d.57cc0fb6.c2.dns.getbobspizza.com
```

values(dns.answers[].rrname) ◆

```
cdn.063b6397a.74102efe.deepdish.dns.getbobspizza.com
cdn.1000063b6397a.74102efe.deepdish.dns.getbobspizza.com
cdn.1000163b6397a.74102efe.deepdish.dns.getbobspizza.com
cdn.1000263b6397a.74102efe.deepdish.dns.getbobspizza.com
cdn.1000363b6397a.74102efe.deepdish.dns.getbobspizza.com
cdn.1000463b6397a.74102efe.deepdish.dns.getbobspizza.com
cdn.1000563b6397a.74102efe.deepdish.dns.getbobspizza.com
cdn.100063b6397a.74102efe.deepdish.dns.getbobspizza.com
```

DNS Beaconsing

- ```
- "api.017784db6.2d7e6eba.c2.dns.getbobspizza.com"
- "cdn.061362a57.57cc0fb6.deepdish.dns.getbobspizza.com"
- "post.2830be959d752afaa9039f60a589cff61fed5c8adfff0a68b6.df60370897410454dbb991d3e3f2f1b8741dc56f7bc7d56d.17c2b2cc3.57cc0fb6.resolver.dns.getbobspizza.com"
```

```
values(dns.answers{}.rrname) ^
```

```
values(dns.answers{}.rrname) ⚡
```

post.102df5237.c2d0e7b30.57cc0fb6.deepdish.dns.getbobspizza.com

post.103228c54.426e10788.57cc0fb6.resolver.dns.getbobspizza.com

post.1035497e1.135036d64.57cc0fb6.c2.dns.getbobspizza.com

post.10367b49d.926e10788.57cc0fb6.resolver.dns.getbobspizza.com

post.103e5e49c.c60f747af.57cc0fb6.deepdish.dns.getbobspizza.com

post.10428d906654ada7feb4f80916e451e4c848b7fd28ccba681610bd269.32269d576b.7bd8c5be.resolver.dns.getbobspizza.com

post.104735942.c71ad65c4.57cc0fb6.resolver.dns.getbobspizza.com

Call: 1800030033/4, 77102010, 400941311, 4113, 50000301220, 50111

# DNS Beacons

## TXT Records

| _time ▾                    | values(dns.answers[].rrname) ⚡                          | values(dns.grouped.TXT[]) ⚡                                                                         |
|----------------------------|---------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| 2021-03-18<br>21:10:54.744 | api.12eb58752c7d.74102efe.deepdish.dns.getbobspizza.com | cHqGTM4+0vFMabBRHcxdz7FRAepMcfikVjZre62rDYCYwHd3TYkTJyJRPHoR9wGBY4kt1IgXCEjYjrDNuRs=                |
| 2021-03-18<br>21:10:54.742 | api.12ea58752c7d.74102efe.deepdish.dns.getbobspizza.com | 0yWvWja2gq8XRqyZQCwopnz+kPCxVGh9UGhTvC7IoINT8d03XM5w4okbVSkjWbrzX43K1unCM3RU61dCztyu0B12qe1wm8UnPL  |
| 2021-03-18<br>21:10:54.740 | api.12e958752c7d.74102efe.deepdish.dns.getbobspizza.com | JpHvN8TwOz3kJ8041NZdjeVuS0h1DG122n17RQbLK/+s5I1f7L+m1cqdmBf3MBLB1wgJeVf6JcOHGUa/w6ybXwMjVgxrk0Y/rA  |
| 2021-03-18<br>21:10:54.739 | api.12e858752c7d.74102efe.deepdish.dns.getbobspizza.com | 37NrUiQBG+nGUbS5xwWn2NrR1Lpajk9TC3L0pECmoHoK+wxofSs2FPrhqe9ln3fSrTADG1Djc2TC3bfceZC3nv0jypcfadB9jf  |
| 2021-03-18<br>21:10:54.737 | api.12e758752c7d.74102efe.deepdish.dns.getbobspizza.com | 3gGXRo0Fm8jY1GMqEuF4gjdUGfCoPpJDMLOkGAzR0vJ51TE9FOu7vvAb1Dgk5PiS0sag6qq9LiY41JzKc1vv80RKDiFpm9r63   |
| 2021-03-18<br>21:10:54.735 | api.12e658752c7d.74102efe.deepdish.dns.getbobspizza.com | 2Bv60oLxQ+3s6AnP0dFLcIp6qdsr6fdDLXU3aSxij5AW6gnwRUCvjrp19fSWaGiYRmD8J6G9IXV3EGpc+k1sKvFfMJJ3A+sfMv  |
| 2021-03-18<br>21:10:54.733 | api.12e558752c7d.74102efe.deepdish.dns.getbobspizza.com | /NvLUj8fbAZTWgqC7vEMYMh/ksqS+rzldMnS5LA0QYcgRL8ju8ju01CXgcJKnjygzgTph3s50tLQ//A5hYa71scJSYL85GSdo1e |
| 2021-03-18<br>21:10:54.731 | api.12e458752c7d.74102efe.deepdish.dns.getbobspizza.com | Md2aKoLjrc/heaJ2q6B2VEpGaTWjIuoYGVreJ5E72SXmAH1DKUpR1Jww9iGCDzQeuczWrkDZ1Xo+URqxECPq05thVrb00KtrL+  |
| 2021-03-18<br>21:10:54.730 | api.12e358752c7d.74102efe.deepdish.dns.getbobspizza.com | TvoPHPLU416Kn1KGuCOrzB11Ndr1UNnHWvPFmwBy8AW5p6jCjzy8++Bsguhs6Yn4hY0mHmBCB41qn555YnULQR7dwAHpk9o2fG  |

# DNS Beaconsing

## Tips for Hunting

Find that noisy TXT requestee

- Why? What process? Why is the data <weird>?

A Records

- Hunting the defaults will be easy. Most profiles change it though.

Hunt Queries


- `source=suricata dns.answers{}.rrname IN ("api.*", "cdn.*", "post.*") | stats values(dns.answers{}.rrname) by src_ip`
- `source=suricata dns.answers{}.rrtype=TXT dns.answers{}.rrname IN ("api.*", "cdn.*", "post.*") | stats values(src_ip) count by dns.answers{}.rrtype | where count >= 500`
- `| timechart count by dns.answers{}.rrtype limit=10`




# Key Takeaways

- Splunk Melting-Cobalt can help you stay ahead of the curve
- Build security content around the latest Cobalt Strike tradecraft
- Enrich Threat Intelligence with identified Cobalt Strike Profiles

 Splunk Melting Cobalt: <https://github.com/splunk/melting-cobalt/>

 Splunk Melting Cobalt Dash:  
<https://gist.github.com/d1vious/54048a8c701fa073cebdcd7b894068df>

 Melting Cobalt inputs.conf:  
<https://gist.github.com/d1vious/c84458fa92181f2825b0920f5a8c6566>

Thank you for listening 🙏