Hunting beacons

Bartosz Jerzman
agenda

Part I: HTTP beacon detection
Part II: HTTPS beacon detection
Part III: Let’s hunt them early – C2 scanning
whoami

- Sysadmin and network defender for the Polish Navy
- Incident responder
- Pentester
- Cyber threat intelligence analyst & adversary hunter
- @secman_pl
PART I
Beaconing over HTTP
What is beaoning?

- Malware does **not keep long connection** to C2
- Malware connects to C2 **periodically**
- Beaconing can occur regularly at **constant intervals**
- Or it can occur at **pseudorandom** moments of time
Time for x33fcon 2019 most popular meme

ONE DOES NOT SIMPLY DETECT BEACONS

WITH SIGNATURE MATCHING
Signature matching for beaconing?

Cobalt Strike beacon traffic simulating Slack communication
Would your SOC escalate on this?
Would your SOC escalate on this?

IDS detected that HTTP response body **is not gzipped** as it has been declared in the response headers.
Set of hypothesis:

#1: analysis of intervals of connections
#2: same URI for different Host names
#3: same or none Referrer to many URIs
#4: different URIs but length is constant
Dataset:

- Data from Cyber Defence Exercise: „Locked Shields”
- PCAP -> processed by BRO-IDS/ZEEK -> http.log
- Example of data from http.log
- Alternative data sources: flows, webproxy logs

<table>
<thead>
<tr>
<th>srcIP</th>
<th>srcPort</th>
<th>dstIP</th>
<th>dstPort</th>
<th>method</th>
<th>host</th>
<th>uri</th>
<th>user_agent</th>
<th>Req_body_length</th>
<th>Resp_body_length</th>
<th>cookie</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.18.7.3</td>
<td>50474</td>
<td>39.88.160[.]18</td>
<td>80</td>
<td>POST</td>
<td>test.com</td>
<td>/test.php</td>
<td>Mozilla/5.0 (Windows NT 6.1; WOW64)</td>
<td>0</td>
<td>303</td>
<td>Trackr=eDMzZmNvbg==</td>
</tr>
</tbody>
</table>
Hypothesis #1: analysis of connections intervals

**Assumption:** Connection intervals from malware to C2 server are distributed around some average value.

**WHY?**

Beaconing malware often has configuration options for setting:
- **sleep** time
- **jitter** (variations from central value)

```bash
#default Beacon sleep duration and jitter
set sleeptime "60000"
set jitter  "20"
```
Hypothesis #1: analysis of connections intervals

STATISTICS CAN HELP FIND BEACONS?

LET'S GO BACK TO SCHOOL
Hypothesis #1: analysis of connections intervals

https://www.investopedia.com
**Hypothesis #1:** analysis of connections intervals

Beacon A: Cobalt Strike payload with configuration { **60 s sleep, 20% jitter** }

Beacon B: Cobalt Strike payload with **manual sleep** commands from operator

<table>
<thead>
<tr>
<th>Beacon</th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>#6</th>
<th>AVG</th>
<th>STDDEV</th>
<th>Variation Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>48s</td>
<td>51s</td>
<td>62s</td>
<td>69s</td>
<td>55s</td>
<td>60s</td>
<td>57.5s</td>
<td>+/- 7.75 s</td>
<td>13.4 %</td>
</tr>
<tr>
<td>B</td>
<td>1s</td>
<td>2s</td>
<td>100s</td>
<td>14s</td>
<td>70s</td>
<td>27s</td>
<td>35.7s</td>
<td>+/- 40.5 s</td>
<td>113.5 %</td>
</tr>
</tbody>
</table>
Hypothesis #1: analysis of connections intervals

![Hypothesis Diagram](image)

\[
\text{Var. Coeff.} = \frac{\text{STDDEV}}{\text{AVG}} \times 100\%
\]

<table>
<thead>
<tr>
<th>Beacon</th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>#6</th>
<th>AVG</th>
<th>STDDEV</th>
<th>Variation Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>48s</td>
<td>51s</td>
<td>62s</td>
<td>69s</td>
<td>55s</td>
<td>60s</td>
<td>57.5s</td>
<td>+/- 7.75 s</td>
<td>13.4 %</td>
</tr>
<tr>
<td>B</td>
<td>1s</td>
<td>2s</td>
<td>100s</td>
<td>14s</td>
<td>70s</td>
<td>27s</td>
<td>35.7s</td>
<td>+/- 40.5 s</td>
<td>113.5 %</td>
</tr>
</tbody>
</table>

Beacon A: Cobalt Strike payload with configuration {60 s sleep, 20% jitter}

Beacon B: Cobalt Strike payload with manual sleep commands from operator
Hypothesis #1: analysis of connections intervals

Query inspired by: [https://www.splunk.com/blog/2018/03/20/hunting-your-dns-dragons.html](https://www.splunk.com/blog/2018/03/20/hunting-your-dns-dragons.html)
Hypothesis #1: analysis of connections intervals

Aggregate connections
By srcIP,dstIP,User-Agent

Query inspired by: https://www.splunk.com/blog/2018/03/20/hunting-your-dns-dragons.html
Hypothesis #1: analysis of connections intervals

Variation Coeff < 100 %
At least 10 connections
AvgBeaconTime > 1s

Query inspired by: https://www.splunk.com/blog/2018/03/20/hunting-your-dns-dragons.html
Hypothesis #1: analysis of connections intervals

C2 server 78.187.72[.]190
AvgBeaconTime 7s
StdDev +/- 3
= very interactive session
Hypothesis #1: analysis of connections intervals

C2 server 222.186.31[.]162
BeaconTime: 28 min +/- 7 min
Longterm operation for maintaining access
Hypothesis #2: same URI for different Host names

Hypothesis is based on the assumption that:
Adversary is using backdoor that has several C2 backup domains included in the configuration.

Hypothesis #2: same URI for different Host names

<table>
<thead>
<tr>
<th>host</th>
<th>hcount</th>
<th>uri</th>
</tr>
</thead>
<tbody>
<tr>
<td>honeybeer.ex</td>
<td>5</td>
<td>/tr_.gif?mark=_uzeeaEEe7rYQQ_nQvXEheHR3Y1QQbo06oh3fmBxU1_ay21ONM3wAELRvjsY7uqj4ar7TSjsNssPScQrRCsEY30WyRfD17je1N77HnrnyoH2pWFIigTeEvhQQus4</td>
</tr>
<tr>
<td>ls17themoview.ex</td>
<td>5</td>
<td>/tr_.gif?mark=_uzeeh9NB6EPsaPbu8oqlLlb5CqxSjgDeOsyUldxBK7AyCf1tNEhtAypTL0zkTLmNY9HGwS6AxtYGsqs6s71g9KbzWxKtqkH</td>
</tr>
</tbody>
</table>
Hypothesis #2: same URI for different Host names

Datasource is HTTP log from Zeek (request and response data)
Hypothesis #2: same URI for different Host names

Several false positive URIs are excluded
Hypothesis #2: same URI for different Host names

Logic: How many different hosts were requested with same URI?
Hypothesis #2: same URI for different Host names

Detection threshold: 3 different hosts
Hypothesis #2: same URI for different Host names

5 unique C2 domains discovered for 2 similar yet different URI requests

<table>
<thead>
<tr>
<th>URI</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>honeybeer.ex</td>
<td>5</td>
</tr>
<tr>
<td>ls17themovie.ex</td>
<td></td>
</tr>
<tr>
<td>scripts.node.ex</td>
<td></td>
</tr>
<tr>
<td>spend.touristhaus.ex</td>
<td></td>
</tr>
<tr>
<td>theforum.ex</td>
<td></td>
</tr>
</tbody>
</table>
### Hipothesis #3: Same or none Referrer to many URIs

The query used to analyze the data is:

```
index=* OR index=* sourcetype=zeek_http uri="/" AND uri="/favicon.ico" AND uri="/admin/" |stats values(resp_h) as dest_ip values(uri) as uri values(referrer) as referer by host_dest |eval ucount=mvcount(uri) | eval rcount=mvcount(referrer) | eval dcount=mvcount(dest_ip) | table dest_ip ,dcound,host_dest,referrer,rcount,ucount,uri |where rcount = 1 and ucount > 3 and ucount < 10 and dcount = 1
```

#### Table: Same (or None) Referrer to Many URIs

<table>
<thead>
<tr>
<th>dest_ip</th>
<th>dcount</th>
<th>host_dest</th>
<th>referer</th>
<th>rcount</th>
<th>ucount</th>
<th>uri</th>
</tr>
</thead>
<tbody>
<tr>
<td>39.65.188.147</td>
<td>1</td>
<td>39.65.188.147</td>
<td>-</td>
<td>1</td>
<td>5</td>
<td>/DaaV /EkCI /Hvj4 /W5Sz /hITW</td>
</tr>
<tr>
<td>123.138.215.56</td>
<td>1</td>
<td>apexgames.ex</td>
<td>-</td>
<td>1</td>
<td>4</td>
<td>/blondie.zip /playnow /ucDND12NzY /ucWNTMxMA</td>
</tr>
<tr>
<td>13.107.4.50</td>
<td>1</td>
<td>au.download.windowsupdate.com</td>
<td>-</td>
<td>1</td>
<td>7</td>
<td>/c/msdownload/up /c/msdownload/up /c/msdownload/up</td>
</tr>
</tbody>
</table>
### Hypothesis #3: Same or none Referrer to many URIs

Counting Referrers on single destination

Threshold $>3$ AND $<10$

<table>
<thead>
<tr>
<th>dest_ip</th>
<th>dcount</th>
<th>host_dest</th>
<th>referrer</th>
<th>rcount</th>
<th>ucount</th>
<th>ur</th>
<th>tcount</th>
</tr>
</thead>
<tbody>
<tr>
<td>39.65.188.147</td>
<td>1</td>
<td>39.65.188.147</td>
<td>-</td>
<td>1</td>
<td>5</td>
<td>/DaAV</td>
<td></td>
</tr>
<tr>
<td>39.65.188.147</td>
<td>1</td>
<td>39.65.188.147</td>
<td>-</td>
<td>1</td>
<td>5</td>
<td>/EKCl</td>
<td></td>
</tr>
<tr>
<td>39.65.188.147</td>
<td>1</td>
<td>39.65.188.147</td>
<td>-</td>
<td>1</td>
<td>5</td>
<td>/Hvj4</td>
<td></td>
</tr>
<tr>
<td>39.65.188.147</td>
<td>1</td>
<td>39.65.188.147</td>
<td>-</td>
<td>1</td>
<td>5</td>
<td>/W5Sz</td>
<td></td>
</tr>
<tr>
<td>39.65.188.147</td>
<td>1</td>
<td>39.65.188.147</td>
<td>-</td>
<td>1</td>
<td>5</td>
<td>/hITW</td>
<td></td>
</tr>
<tr>
<td>123.138.215.56</td>
<td>1</td>
<td>apexgames.ex</td>
<td>-</td>
<td>1</td>
<td>5</td>
<td>/blondie.zip</td>
<td></td>
</tr>
<tr>
<td>123.138.215.56</td>
<td>1</td>
<td>apexgames.ex</td>
<td>-</td>
<td>1</td>
<td>5</td>
<td>/playnow</td>
<td></td>
</tr>
<tr>
<td>123.138.215.56</td>
<td>1</td>
<td>apexgames.ex</td>
<td>-</td>
<td>1</td>
<td>5</td>
<td>/ucDND12NzY</td>
<td></td>
</tr>
<tr>
<td>123.138.215.56</td>
<td>1</td>
<td>apexgames.ex</td>
<td>-</td>
<td>1</td>
<td>5</td>
<td>/ucW7MTxMA</td>
<td></td>
</tr>
<tr>
<td>13.107.4.50</td>
<td>1</td>
<td>au.download.windowsupdate.com</td>
<td>-</td>
<td>1</td>
<td>5</td>
<td>/c/msdownload/up</td>
<td></td>
</tr>
<tr>
<td>13.107.4.50</td>
<td>1</td>
<td>au.download.windowsupdate.com</td>
<td>-</td>
<td>1</td>
<td>5</td>
<td>/c/msdownload/up</td>
<td></td>
</tr>
</tbody>
</table>

URIs related to 1st stage malware from C2
### Hypothesis #4: different URIs but length is constant

Another C2 domain discovered with 3 different URIs of same length.

<table>
<thead>
<tr>
<th>orig_h</th>
<th>host_dest</th>
<th>uri</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.18.2.41</td>
<td>39.65.188.147</td>
<td>/DaaV, /Hvj4, /VW5z</td>
</tr>
<tr>
<td>10.18.3.175</td>
<td>fourthgate.ex</td>
<td>/ucDNDI2NzY, /ucWMTg5MzE, /ucWOTAyMzM</td>
</tr>
</tbody>
</table>
Jack Crook (still waiting for you, Jack, at x33fcon) has a great set for hypothesis inspirations:

https://twitter.com/jackcr/status/1029457184164335617
PART II
Beaconing over HTTPS
{ FakeTLS example from LAZARUS APT }
FakeTLS – how does it work?
FakeTLS – how does it work?

The Funny Part of mimicking TLS to popular sites e.g. wetransfer.com
FakeTLS – how does it work?

C2 sends back real (often expired) certificate
FakeTLS – how does it work?

Non-TLS encryption with symmetric, shared RC4 key
FakeTLS – does it beacon?

C2 COMMS (encrypted messages sizes in Bytes)
FakeTLS – does it beacon?

Maximum message size of 808 Bytes
FakeTLS – interesting part shortly after handshake

The beginning of REAL comms has fixed size messages
FakeTLS – is it really hardcoded?

# Message 2 construction in code

```plaintext
push 0x17  # Encrypted Data Header in SSL message
push 1    # TLS 1.0
lea edx, [esp + 0x34]
push 0x18  # 24 bytes - Encrypted Message Length
```
FakeTLS detection using SSL profiling

Analysing the **sizes of first 5 messages** of Encrypted Application Data (after TLS handshake) can help you detect traffic to **unknown C2 infrastructure** that uses FakeTLS.
FakeTLS – what’s wrong with those msg sizes?

In TLS algorithms every message is hashed (e.g. md5) for integrity check:
\[
\text{length(md5(msg))} = 16B
\]
\[
8B < 16B ;)
\]
FakeTLS – where to hunt unknown C2 infrastructure?

Reactive:
- own network traffic detection
- Can your network traffic analyser process TLS data after the handshake?

Proactive:
- pcaps from sandboxes e.g. Hybrid-Analysis
PART III
Let’s hunt them early – C2 scanning
NBA in 1990s – „Offense starts with defense”
Quick intro to wide topic

Groups
Groups that use this software:

APT19
APT29
APT32
Cobalt Group
CopyKittens
DarkHydrus
FIN6
Leviathan

https://attack.mitre.org/
Finding defaults: #1 Cobalt Strike console port

Management console port for Teams Server is by default: 50050/tcp
Finding defaults: #2 Cobalt Strike idle DNS answer

DNS answer for ANY request is: 0.0.0.0
Finding defaults: #3 Cobalt Strike 404 answer

CS (NanoHTTPD) answers with:

HTTP/1.1 404 Not Found
Content-Type: text/plain
Date: Mon, 30 Feb 2019 13:37:00 GMT
Content-Length: 0
Finding defaults: #4 Cobalt Strike „space”

CS responds with additional space after 200 OK
Hunting for NanoHTTPD servers.
Corrected in Cobalt Strike v. 3.13
Conclusion

• Adversary tools and procedures very often have **patterns**
• Threat analyst job is to **uncover** human traces and adversaries weaknesses
• Burn the **defaults**, burn what is **known** (opensource, commercial C2)