Analysis of Cobalt Strike network traffic obfuscation in C2 communication

Vincent van der Eijk & Coen Schuijt

University of Amsterdam

vincent.vandereijk@os3.nl, coen.schuijt@os3.nl

July 3, 2020

Introduction

- Red and Blue Teaming
- RAT $\rightarrow$ Botnet
- Cobalt Strike
- APTs

Figure 1: Cobalt strike logo [https://cobaltstrike.com/]

Research questions

- Main research question

  "How can we distinguish obfuscated Cobalt Strike beacons from genuine traffic based on identifying features?"

- Sub questions
  1. Which features can we extract from network traffic generated by malleable C2 profiles?
  2. Can we detect a Cobalt Strike beacon using a malleable profile with one or more of those features?

State of the art (I/II)

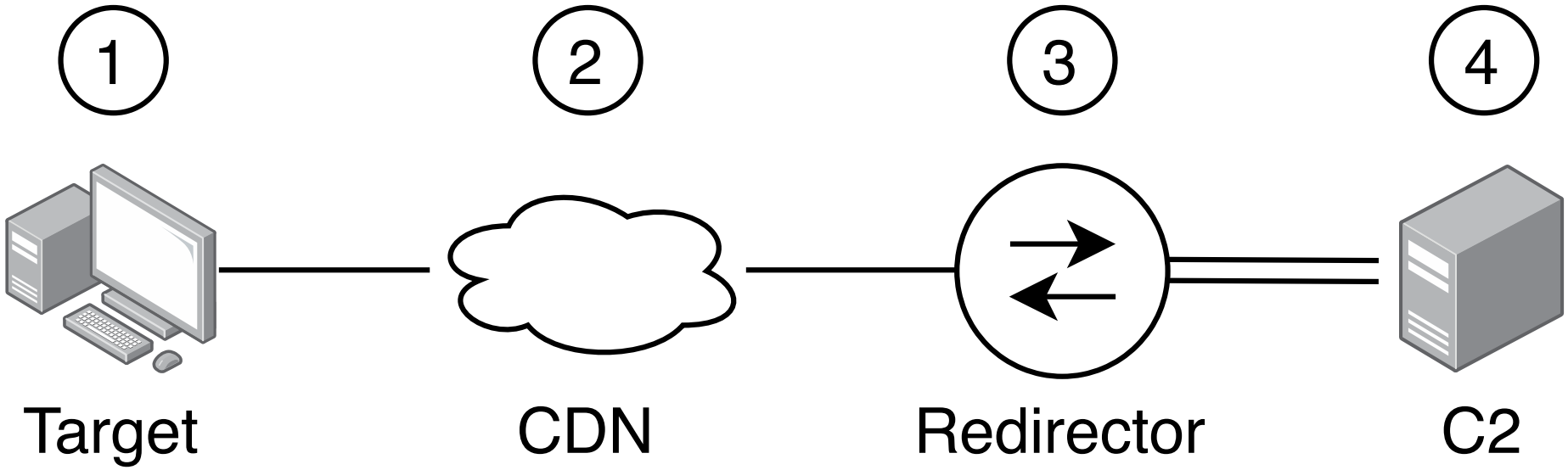

Figure 2: Common C2 network setup

- Beacon
- Domain redirection
- Redirector/proxy
- C2 Server

State of the art (II/II)

- Malleable Profile
  - Defines beaconing behaviour
  - HTTP parameters
  - Encoding
  - Highly customizable

```
1 set sleeptime "5000";
2 set jitter    "0";
3 set useragent "Mozilla/5.0 (Windows NT 6.1; WOW64; Trident
    /7.0; rv:11.0) like Gecko";
4
5 http-get {
6
7     set uri "/s/ref=nb_sb_noss_1/167-3294888-0262949/field-
    keywords=books";
```

Listing 1: Snippet from the amazon.profile

Related work

- Little scientific research on Cobalt Strike
- No research specific to malleable profiles
- Botnet traffic detection researched thoroughly

Sources


L. van Duijn (2014)
Beacon detection in PCAP files


J. Dreijer (2015)
StealthWare - Social Engineering Malware

Methodology

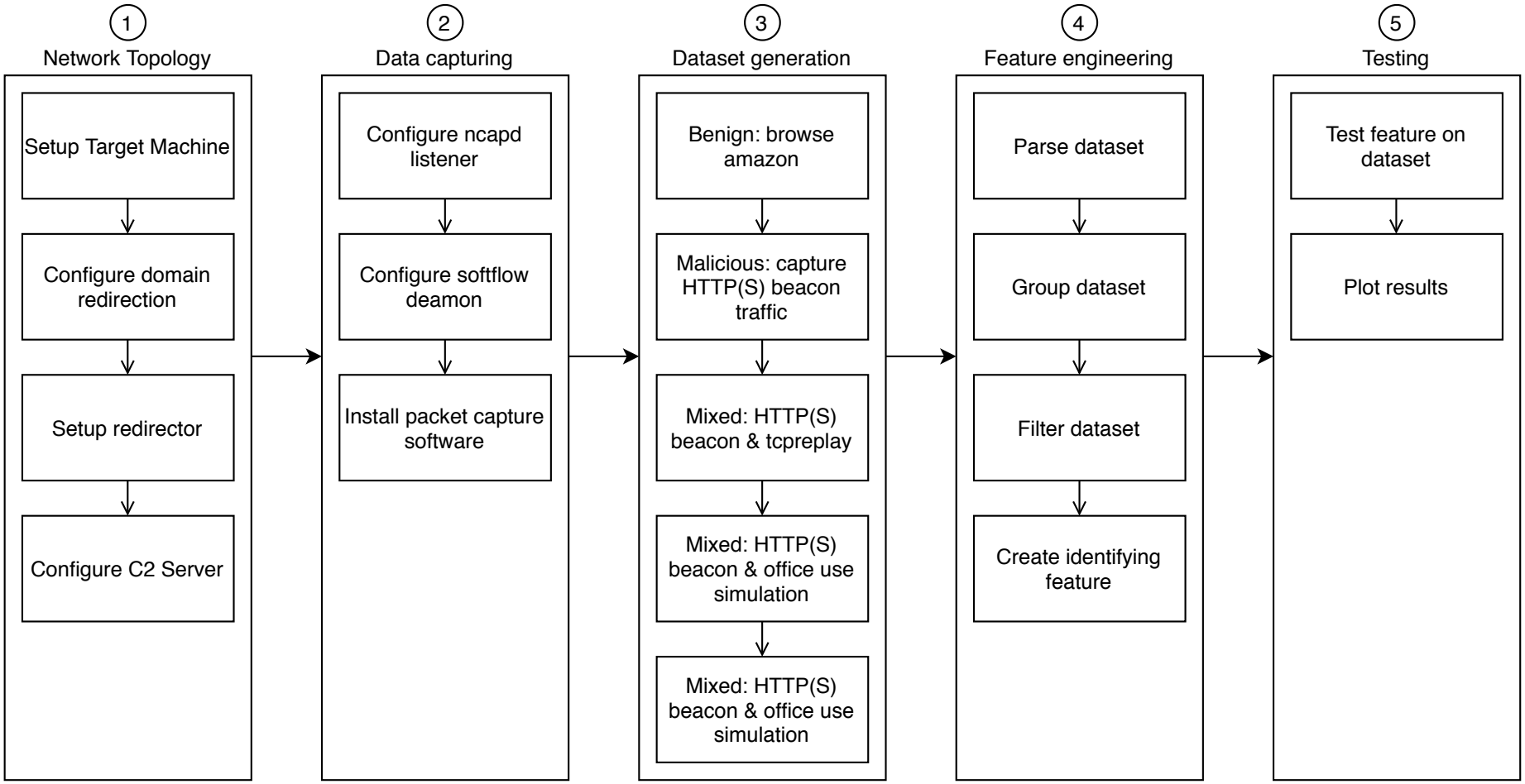

Figure 3: Project approach

Infrastructure setup (I/II)

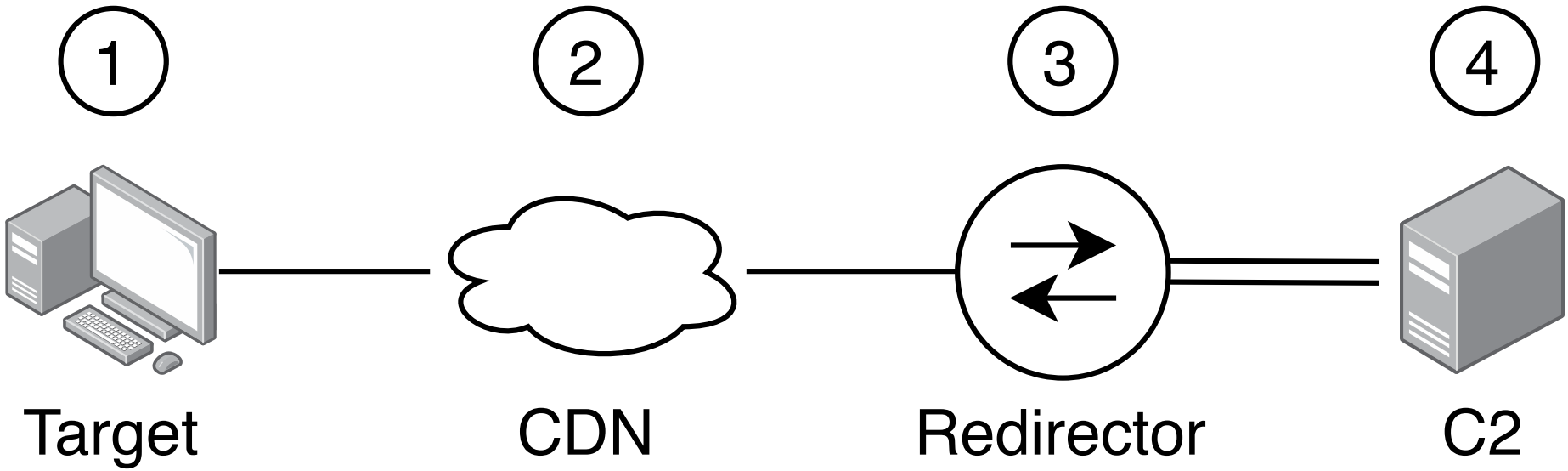

Figure 4: Infrastructure setup

1. Target
   - Windows 10 (1909)
   - NAT interface
2. CDN
   - Amazon CloudFront
   - Domain redirection (Host Header, Redirector IP)

Infrastructure setup (II/II)

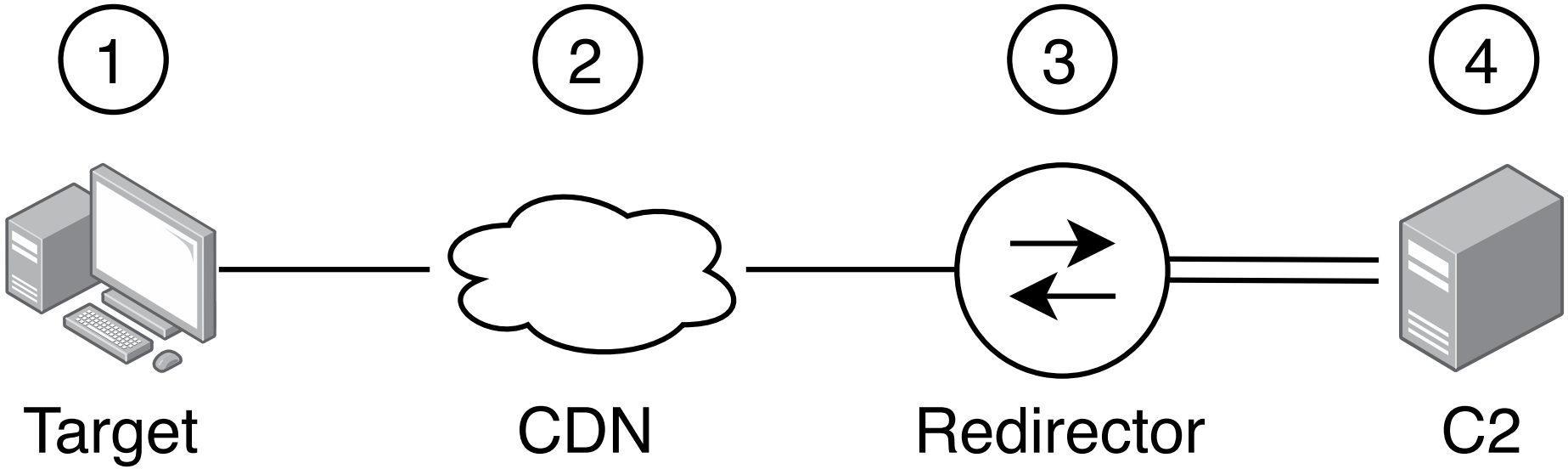

Figure 4: Infrastructure setup

3. Redirector
   - socat proxy
   - 443, 80
4. C2 Server
   - Cobalt Strike 4.0
   - amazon.profile

Data gathering (I/V)

- Benign
  - PCAPS for HTTP
- Malicious
  - NetFlow for HTTPS
- Mixed
  - Active beacon
  - Simulate user
    - browsing
    - updating
    - mailing
    - ...
  - Reproduceable dataset
- External
  - CTU-13 (Botnet-43)[1]
  - 6M flows, university network
  - Stratosphere Research Laboratory (CZ)

[1] https://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-43/

Detection algorithm (I/II)

1. Read NetFlow data
2. Creating host objects
3. Append flow to host (src IP)

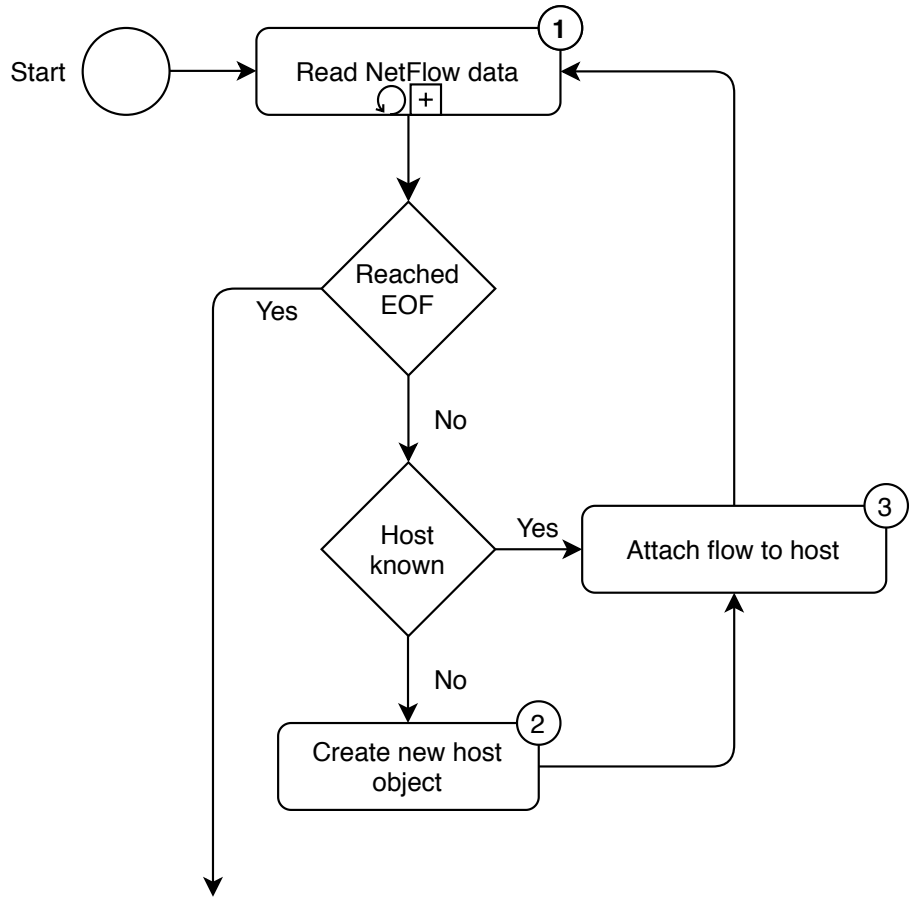

Figure 5: Detection algorithm pt.1

Detection algorithm (II/II)

4. Filter flows
5. Apply feature (Host)
6. Alert

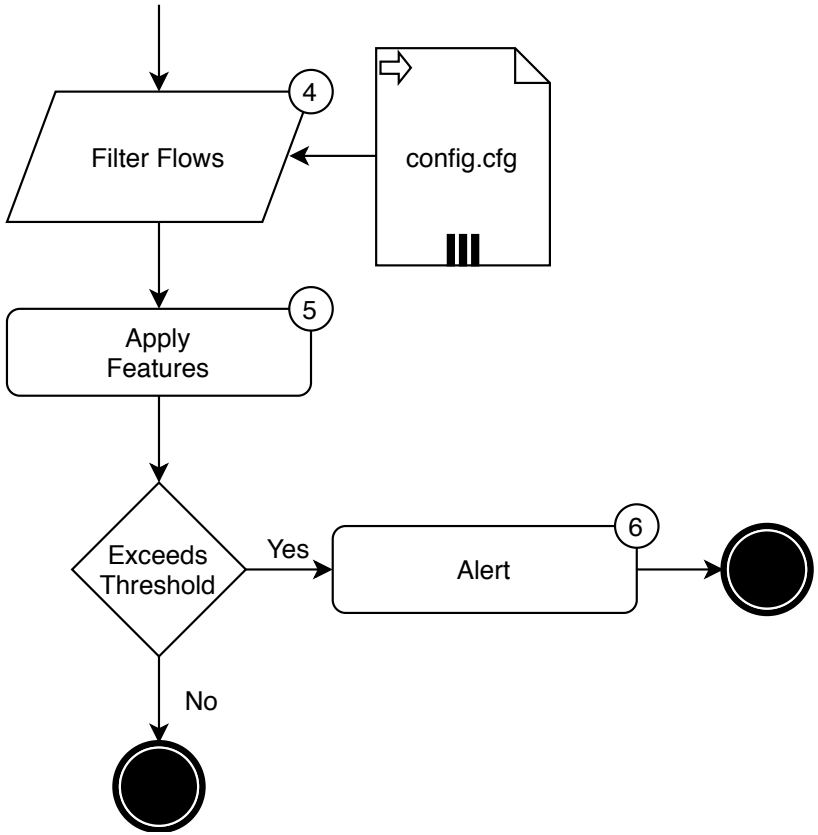

Figure 6: Detection algorithm pt.2

Results

- Amazon.profile traffic analysis (Cobalt Strike)
  - HTTP Beacon
  - Benign Amazon network traffic
  - HTTPS Beacon
- Beacon detection algorithm
- Detection accuracy

Amazon profile traffic analysis: HTTP Beacon (I/V)

http.request

No.	Time	Source	Destination	Protocol	Length	Info
4	0.000785	172.16.22.129	145.100.104.47	HTTP	549	GET /s/ref=nb_sb_noss_1/167-3294888-0262949/field-keywords=books HTTP/1.1
14	5.026288	172.16.22.129	145.100.104.47	HTTP	549	GET /s/ref=nb_sb_noss_1/167-3294888-0262949/field-keywords=books HTTP/1.1
35	10.059287	172.16.22.129	145.100.104.47	HTTP	549	GET /s/ref=nb_sb_noss_1/167-3294888-0262949/field-keywords=books HTTP/1.1
45	15.071252	172.16.22.129	145.100.104.47	HTTP	549	GET /s/ref=nb_sb_noss_1/167-3294888-0262949/field-keywords=books HTTP/1.1
57	20.082889	172.16.22.129	145.100.104.47	HTTP	549	GET /s/ref=nb_sb_noss_1/167-3294888-0262949/field-keywords=books HTTP/1.1
70	25.108927	172.16.22.129	145.100.104.47	HTTP	549	GET /s/ref=nb_sb_noss_1/167-3294888-0262949/field-keywords=books HTTP/1.1
79	30.135122	172.16.22.129	145.100.104.47	HTTP	549	GET /s/ref=nb_sb_noss_1/167-3294888-0262949/field-keywords=books HTTP/1.1
88	35.145898	172.16.22.129	145.100.104.47	HTTP	549	GET /s/ref=nb_sb_noss_1/167-3294888-0262949/field-keywords=books HTTP/1.1
97	40.157684	172.16.22.129	145.100.104.47	HTTP	549	GET /s/ref=nb_sb_noss_1/167-3294888-0262949/field-keywords=books HTTP/1.1

```
> Frame 4: 549 bytes on wire (4392 bits), 549 bytes captured (4392 bits) on interface \Device\NPF_{FDC4E157-1133-4C81-B1A8-9FCB5346F432}, id 0
> Ethernet II, Src: VMware_ed:c5:d8 (00:0c:29:ed:c5:d8), Dst: VMware_f2:bf:fa (00:50:56:f2:bf:fa)
> Internet Protocol Version 4, Src: 172.16.22.129, Dst: 145.100.104.47
> Transmission Control Protocol, Src Port: 52250, Dst Port: 80, Seq: 1, Ack: 1, Len: 495
v Hypertext Transfer Protocol
  > GET /s/ref=nb_sb_noss_1/167-3294888-0262949/field-keywords=books HTTP/1.1\r\n
    Accept: */*\r\n
    Host: www.amazon.com\r\n
  > [truncated]Cookie: skin=noskin;session-token=R4wPMKTXf4iHjL9QIsbLhnATDuHI53+f7CZB7xmsmgkMccS8riu6vynW+VYMLOILS2qwNB4OshR+98hULuWnfuU/20BGYJ+AOvgaDBI…
    User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko\r\n
    Connection: Keep-Alive\r\n
    Cache-Control: no-cache\r\n
    \r\n
    [Full request URI: http://www.amazon.com/s/ref=nb_sb_noss_1/167-3294888-0262949/field-keywords=books]
    [HTTP request 1/1]
    [Response in frame: 6]
```

Figure 7: Packet capture for HTTP beacon

Amazon profile traffic analysis: Benign (II/V)

No.	Time	Source	Destination	Protocol	Length	Info
19	0.964718	52.7.114.31	10.0.2.15	TCP	60	443 → 49753 [ACK] Seq=1 Ack=6327 Win=65535 Len=0
20	0.964719	52.7.114.31	10.0.2.15	TCP	60	443 → 49753 [ACK] Seq=1 Ack=7625 Win=65535 Len=0
21	0.966209	10.0.2.15	10.0.2.3	DNS	74	Standard query 0x9a91 A www.amazon.com
22	0.970128	10.0.2.15	104.99.233.153	TLSv1.2	194	Application Data
23	0.970372	104.99.233.153	10.0.2.15	TCP	60	443 → 50020 [ACK] Seq=1 Ack=141 Win=65535 Len=0
24	0.991400	10.0.2.15	10.0.2.3	DNS	74	Standard query 0x9a91 A www.amazon.com
25	0.994661	10.0.2.3	10.0.2.15	DNS	169	Standard query response 0x9a91 A www.amazon.com CNAME tp.4
26	0.994663	10.0.2.3	10.0.2.15	DNS	169	Standard query response 0x9a91 A www.amazon.com CNAME tp.4
27	0.995109	10.0.2.15	10.0.2.3	DNS	89	Standard query 0x2871 A d3ag4hukkh62yn.cloudfront.net
28	0.995418	10.0.2.3	10.0.2.15	DNS	105	Standard query response 0x2871 A d3ag4hukkh62yn.cloudfront
29	0.995632	10.0.2.15	10.0.2.3	DNS	89	Standard query 0x37c3 AAAA d3ag4hukkh62yn.cloudfront.net
30	1.011023	10.0.2.3	10.0.2.15	DNS	89	Standard query response 0x37c3 AAAA d3ag4hukkh62yn.cloudfr
31	1.061799	52.7.114.31	10.0.2.15	TLSv1.2	96	Application Data
32	1.064981	52.7.114.31	10.0.2.15	TLSv1.2	248	Application Data
33	1.065019	10.0.2.15	52.7.114.31	TCP	54	49753 → 443 [ACK] Seq=7625 Ack=237 Win=63532 Len=0
34	1.438204	104.99.233.153	10.0.2.15	TLSv1.2	733	Application Data
35	1.438208	104.99.233.153	10.0.2.15	TLSv1.2	1474	Application Data
36	1.438209	104.99.233.153	10.0.2.15	TLSv1.2	1474	Application Data [TCP segment of a reassembled PDU]
37	1.438282	10.0.2.15	104.99.233.153	TCP	54	50020 → 443 [ACK] Seq=141 Ack=3520 Win=64240 Len=0
38	1.438344	104.99.233.153	10.0.2.15	TLSv1.2	1263	Application Data, Application Data
39	1.438911	10.0.2.15	104.99.233.153	TCP	54	50020 → 443 [ACK] Seq=141 Ack=4729 Win=63031 Len=0
40	1.444395	104.99.233.153	10.0.2.15	TLSv1.2	1474	Application Data, Application Data
41	1.444397	104.99.233.153	10.0.2.15	TLSv1.2	1474	Application Data [TCP segment of a reassembled PDU]
42	1.444399	104.99.233.153	10.0.2.15	TLSv1.2	901	Application Data [TCP segment of a reassembled PDU]
43	1.444684	10.0.2.15	104.99.233.153	TCP	54	50020 → 443 [ACK] Seq=141 Ack=8416 Win=64240 Len=0

Figure 8: Packet capture for benign Amazon traffic

Amazon traffic analysis: HTTPS Beacon (III/V)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.22.129	145.100.104.47	TCP	66	52424 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000491	145.100.104.47	172.16.22.129	TCP	60	443 → 52424 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
3	0.000649	172.16.22.129	145.100.104.47	TCP	54	52424 → 443 [ACK] Seq=1 Ack=1 Win=65535 Len=0
4	0.001061	172.16.22.129	145.100.104.47	TLSv1.2	238	Client Hello
5	0.001401	145.100.104.47	172.16.22.129	TCP	60	443 → 52424 [ACK] Seq=1 Ack=185 Win=64240 Len=0
6	0.010356	145.100.104.47	172.16.22.129	TLSv1.2	144	Server Hello
7	0.010448	172.16.22.129	145.100.104.47	TCP	54	52424 → 443 [ACK] Seq=185 Ack=91 Win=65535 Len=0
8	0.010804	145.100.104.47	172.16.22.129	TLSv1.2	60	Change Cipher Spec
9	0.010857	172.16.22.129	145.100.104.47	TCP	54	52424 → 443 [ACK] Seq=185 Ack=97 Win=65535 Len=0
10	0.010934	145.100.104.47	172.16.22.129	TLSv1.2	99	Encrypted Handshake Message
11	0.010958	172.16.22.129	145.100.104.47	TCP	54	52424 → 443 [ACK] Seq=185 Ack=142 Win=65535 Len=0
12	0.011396	172.16.22.129	145.100.104.47	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
13	0.011563	145.100.104.47	172.16.22.129	TCP	60	443 → 52424 [ACK] Seq=142 Ack=236 Win=64240 Len=0
14	0.013011	172.16.22.129	145.100.104.47	TLSv1.2	578	Application Data
15	0.013227	145.100.104.47	172.16.22.129	TCP	60	443 → 52424 [ACK] Seq=142 Ack=760 Win=64240 Len=0
16	0.054741	145.100.104.47	172.16.22.129	TLSv1.2	339	Application Data
17	0.054763	145.100.104.47	172.16.22.129	TLSv1.2	85	Encrypted Alert
18	0.054834	172.16.22.129	145.100.104.47	TCP	54	52424 → 443 [ACK] Seq=760 Ack=459 Win=65535 Len=0
19	0.055028	172.16.22.129	145.100.104.47	TCP	54	52424 → 443 [FIN, ACK] Seq=760 Ack=459 Win=65535 Len=0
20	0.055125	172.16.22.129	145.100.104.47	TCP	54	52424 → 443 [RST, ACK] Seq=761 Ack=459 Win=0 Len=0
21	0.055190	145.100.104.47	172.16.22.129	TCP	60	443 → 52424 [ACK] Seq=459 Ack=761 Win=64239 Len=0
22	0.055206	172.16.22.129	145.100.104.47	TCP	54	52424 → 443 [RST] Seq=761 Win=0 Len=0
23	5.064351	172.16.22.129	145.100.104.47	TCP	66	52425 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
24	5.064843	145.100.104.47	172.16.22.129	TCP	60	443 → 52425 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460

Figure 9: Packet capture for Amazon HTTPS beacon

Amazon traffic analysis: HTTPS Beacon (IV/V)

Date first seen	Duration	Proto	Src IP Addr:Port		Dst IP Addr:Port	Flags	Tos	Packets	Bytes
2020-07-02 20:39:39.472	0.064	TCP	172.16.22.129:50223	->	145.100.104.47:443	.APRSF	0	13	1297
2020-07-02 20:39:44.552	0.056	TCP	172.16.22.129:50224	->	145.100.104.47:443	.APRSF	0	13	1297
2020-07-02 20:39:49.616	0.059	TCP	172.16.22.129:50225	->	145.100.104.47:443	.APRSF	0	12	1257
2020-07-02 20:39:54.689	0.058	TCP	172.16.22.129:50226	->	145.100.104.47:443	.APRSF	0	12	1257
2020-07-02 20:39:59.754	0.060	TCP	172.16.22.129:50227	->	145.100.104.47:443	.APRSF	0	12	1257
2020-07-02 20:40:04.827	0.059	TCP	172.16.22.129:50228	->	145.100.104.47:443	.APRSF	0	12	1257
2020-07-02 20:40:09.893	0.058	TCP	172.16.22.129:50229	->	145.100.104.47:443	.APRSF	0	12	1257
2020-07-02 20:40:14.972	0.064	TCP	172.16.22.129:50230	->	145.100.104.47:443	.APRSF	0	12	1257
2020-07-02 20:40:20.046	0.060	TCP	172.16.22.129:50231	->	145.100.104.47:443	.APRSF	0	13	1297
2020-07-02 20:40:25.111	0.062	TCP	172.16.22.129:50232	->	145.100.104.47:443	.APRSF	0	13	1297
2020-07-02 20:40:30.183	0.055	TCP	172.16.22.129:50233	->	145.100.104.47:443	.APRSF	0	12	1257
2020-07-02 20:40:35.248	25.360	TCP	172.16.22.129:50234	->	145.100.104.47:443	.APRSF	0	13	1297
2020-07-02 20:40:40.328	0.060	TCP	172.16.22.129:50235	->	145.100.104.47:443	.APRSF	0	13	1297
2020-07-02 20:40:45.402	0.059	TCP	172.16.22.129:50236	->	145.100.104.47:443	.APRSF	0	12	1257
2020-07-02 20:40:50.466	0.060	TCP	172.16.22.129:50237	->	145.100.104.47:443	.APRSF	0	13	1297
2020-07-02 20:40:55.540	13.428	TCP	172.16.22.129:50238	->	145.100.104.47:443	.APRSF	0	13	1297
2020-07-02 20:41:00.608	0.064	TCP	172.16.22.129:50239	->	145.100.104.47:443	.APRSF	0	12	1257
2020-07-02 20:41:05.684	0.059	TCP	172.16.22.129:50240	->	145.100.104.47:443	.APRSF	0	13	1297
2020-07-02 20:41:10.755	0.052	TCP	172.16.22.129:50241	->	145.100.104.47:443	.APRSF	0	11	1217
2020-07-02 20:41:15.819	0.063	TCP	172.16.22.129:50242	->	145.100.104.47:443	.APRSF	0	12	1257
2020-07-02 20:41:20.892	0.059	TCP	172.16.22.129:50243	->	145.100.104.47:443	.APRSF	0	12	1257
2020-07-02 20:41:25.956	0.063	TCP	172.16.22.129:50244	->	145.100.104.47:443	.APRSF	0	12	1257

Summary: total flows: 22, total bytes: 27974, total packets: 272, avg bps: 2100, avg pps: 2, avg bpp: 102
Time window: 2020-07-02 20:39:39 - 2020-07-02 20:41:26
Total flows processed: 47, Blocks skipped: 0, Bytes read: 3888
Sys: 0.002s flows/second: 20991.5 Wall: 0.000s flows/second: 770491.8

Figure 10: NetFlow data for HTTPS beacon

Amazon traffic analysis: Summary (V/V)

- We identified the following features:
  - Periodicity
  - Consistent byte size of flows
  - Short flow duration
  - TCP Flags
  - Lack of DNS requests

Beacon detection

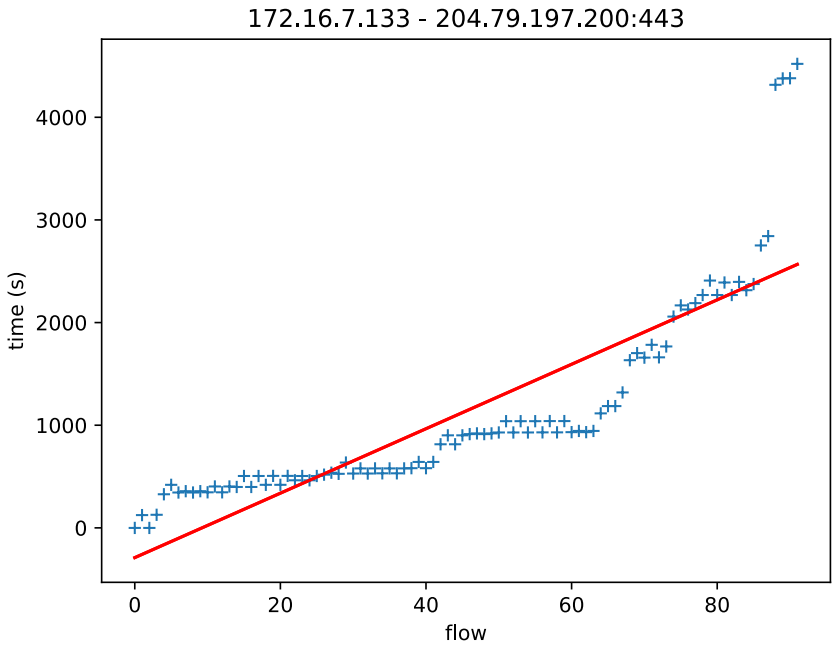

Figure 11: Linear regression for regular HTTPS network traffic shows a weak correlation (r=0.854)

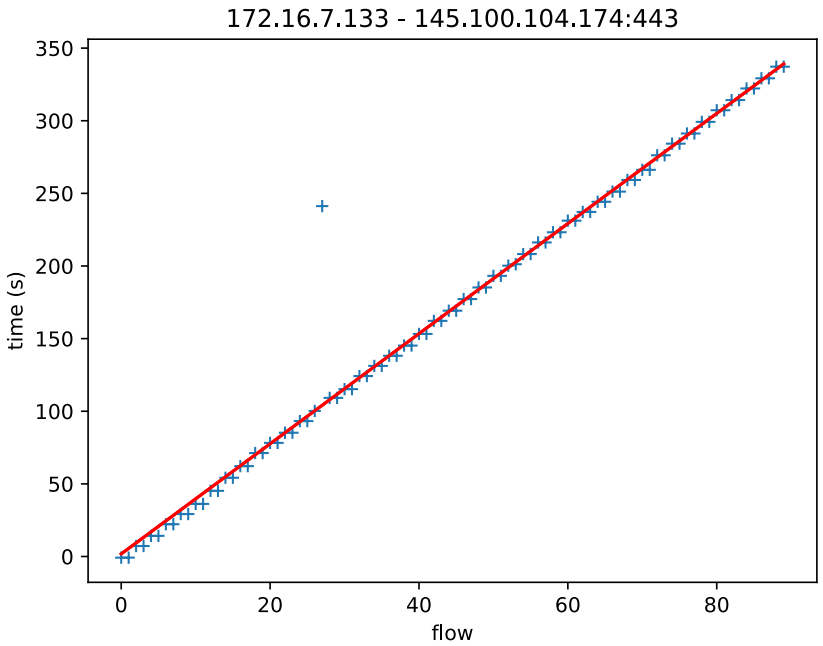

Figure 12: Linear regression for C2 server network traffic shows a high correlation (r=0.999)

Results: Accuracy

Table 1: Overview of NetFlow streams that the detection algorithm was able to classify correctly as either benign (good) or malicious (bad)

		Actual	
		Good	Bad
Predicted	Good	128910	2
	Bad	5	15

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} = \frac{13 + 128267}{13 + 128267 + 5 + 2} = 99,996\%$$

Discussion

- Difficult to obtain a large dataset with benign network traffic
- Only tested on our own malware samples and infrastructure

Conclusions I/II

- *Q1: Which features can we extract from network traffic generated by malleable C2 profiles?*
  - Time interval
  - Byte size of flow
  - Flow duration
  - TCP flags
  - DNS requests
- *Q2: Can we detect a Cobalt Strike beacon using a malleable profile with one or more of those features?*
  - All features except the correlation to DNS requests and the TCP RST flag are useable

Conclusions II/II

- How can we distinguish obfuscated Cobalt Strike beacons from genuine traffic based on identifying features?
  - Filter rules based on identified features
  - Detection algorithm using linear regression

Future Work

- Further research the TCP RST flag behaviour
- Expand the detection algorithm to fingerprint threat actors
- Modify the detection algorithm to support real-time detection

Key findings

- C2 communication of Cobalt Strike shows periodicity
- We are able to detect other profiles than the Amazon profile
- Avoid detection by changing the beaconing interval regularly