

Detecting Beaconsing Malware

Malware beaconsing is a technique we frequently encounter in our cybersecurity investigations. It refers to the periodic communication initiated by malware-infected systems with their respective command and control (C2) servers. The beacons, typically small data packets, are sent at regular intervals, much like a lighthouse sends out a regular signal.

In our analysis of beaconsing behavior, we often observe several distinct patterns. The beaconsing intervals can be fixed, jittered (varied slightly from a fixed pattern), or follow a more complex schedule based on the malware's specific objectives. We've encountered malware that uses various protocols for beaconsing, including HTTP/HTTPS, DNS, and even ICMP (ping).

In this section, we will concentrate on detecting the beaconsing behavior associated with a widely recognized Command and Control (C2) framework known as **Cobalt Strike** (in its default configuration).

Let's now navigate to the bottom of this section and click on "Click here to spawn the target system!". Then, access the Splunk interface at [https://\[Target IP\]:8000](https://[Target IP]:8000) and launch the Search & Reporting Splunk application. The vast majority of searches covered from this point up to end of this section can be replicated inside the target, offering a more comprehensive grasp of the topics presented.

Additionally, we can access the spawned target via RDP as outlined below. All files, logs, and PCAP files related to the covered attacks can be found in the `/home/htb-student` and `/home/htb-student/module_files` directories.

```
Detecting Beaconsing Malware

MisaelMacias@htb[/htb]$ xfreerdp /u:htb-student /p:'HTB@cademy_stdnt!' /v:[Target IP] /dynamic-res
```

Related Evidence

- Related Directory: `/home/htb-student/module_files/cobaltstrike_beacon`
- Related Splunk Index: `cobaltstrike_beacon`
- Related Splunk Sourcetype: `bro:http:json`

Detecting Beaconsing Malware With Splunk & Zeek Logs

Now let's explore how we can identify beaconsing, using Splunk and Zeek logs.

```
Detecting Beaconsing Malware

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
```

```
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
```

[Resources](#)[Go to Questions](#)

Table of Contents

Leveraging Windows Event Logs

- [Detecting Common User/Domain Recon](#) ✓
- [Detecting Password Spraying](#) ✓
- [Detecting Responder-like Attacks](#) ✓
- [Detecting Kerberoasting/AS-REProasting](#) ✓
- [Detecting Pass-the-Hash](#) ✓
- [Detecting Pass-the-Ticket](#) ✓
- [Detecting Overpass-the-Hash](#) ✓
- [Detecting Golden Tickets/Silver Tickets](#) ✓
- [Detecting Unconstrained Delegation/Constrained Delegation Attacks](#) ✓
- [Detecting DCSync/DCShadow](#) ✓

Leveraging Splunk's Application Capabilities

- [Creating Custom Splunk Applications](#) ✓

Leveraging Zeek Logs

- [Detecting RDP Brute Force Attacks](#) ✓
- [Detecting Beaconsing Malware](#) ✓
- [Detecting Nmap Port Scanning](#) ✓
- [Detecting Kerberos Brute Force Attacks](#) ✓
- [Detecting Kerberoasting](#) ✓
- [Detecting Golden Tickets](#) ✓
- [Detecting Cobalt Strike's PSEXEC](#) ✓
- [Detecting Zerologon](#) ✓
- [Detecting Exfiltration \(HTTP\)](#) ✓
- [Detecting Exfiltration \(DNS\)](#) ✓
- [Detecting Ransomware](#) ✓

Skills Assessment

index="cobaltstrike_beacon" sourcetype="bro:http:json"

| sort _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

76 events (before 8/30/21 9:00:37000 AM) No Event Sampling

Job

Smart Mode

Events Patterns Statistics (1) Visualization

20 Per Page Format Preview

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

Search Breakdown:

- `index="cobaltstrike_beacon" sourcetype="bro:http:json"`: Selects the data from the `cobaltstrike_beacon` index and filters events of type `bro:http:json`, which represent Zeek HTTP logs.
- `| sort _time`: Sorts the events in ascending order based on their timestamp (`_time`).
- `| streamstats current=f last(_time) as prevtime by src, dest, dest_port`: For each event, calculates the previous event's timestamp (`prevtime`) grouped by source IP (`src`), destination IP (`dest`), and destination port (`dest_port`).
- `| eval timedelta = _time - prevtime`: Computes the time difference (`timedelta`) between the current and previous events' timestamps.
- `| eventstats avg(timedelta) as avg, count as total by src, dest, dest_port`: Calculates the average time difference (`avg`) and the total number of events (`total`) for each combination of `src`, `dest`, and `dest_port`.
- `| eval upper=avg*1.1`: Sets an upper limit for the time difference by adding a 10% margin to the average.
- `| eval lower=avg*0.9`: Sets a lower limit for the time difference by subtracting a 10% margin from the average.
- `| where timedelta > lower AND timedelta < upper`: Filters the events where the time difference falls within the defined upper and lower limits.
- `| stats count, values(avg) as TimeInterval by src, dest, dest_port, total`: Counts the number of events and extracts the average time interval for each combination of `src`, `dest`, `dest_port`, and `total`.
- `| eval prcnt = (count/total)*100`: Calculates the percentage (`prcnt`) of events within the defined time interval limits.
- `| where prcnt > 90 AND total > 10`: Filters the results to only include those where more than 90% of the events fall within the defined time interval limits, and there are more than 10 total events.

VPN Servers

Warning: Each time you "Switch", your connection keys are regenerated and you must re-download your VPN connection file.

All VM instances associated with the old VPN Server will be terminated when switching to a new VPN server.

Existing PwnBox instances will automatically switch to the new VPN server.

US Academy 3

Medium Load

PROTOCOL

☒ UDP 1337 ☐ TCP 443

DOWNLOAD VPN CONNECTION FILE



Connect to Pwnbox

Your own web-based Parrot Linux instance to play our labs.

Pwnbox Location

Skills Assessment

My Workstation

OFFLINE

Start Instance

0 / 1 spawns left

UK

138ms

ⓘ Terminate Pwnbox to switch location

Start Instance

∞ / 1 spawns left

Waiting to start...

☐ Enable step-by-step solutions for all questions ⓘ ✨

Questions

Answer the question(s) below to complete this Section and earn cubes!



Download VPN
Connection File

Target(s): [Click here to spawn the target system!](#)

+ 1 📦 Use the "cobaltstrike_beacon" index and the "bro:httpjson" sourcetype. What is the most straightforward Splunk command to pinpoint beaconing from the 10.0.10.20 source to the 192.168.151.181 destination? Answer format: One word

timechart

Submit

← Previous

Next →

✔ Mark Complete & Next

Powered by



HACKTHEBOX

