Introduction to Network Threat Hunting

Thank you to our sponsors!

ACTIVE | COUNTERMEASURES

BLACK HILLS Information Security • 2008 •

WILD WEST HACKIN' FEST

Two options for doing the labs

- Download the class VM
  - Options for VMWare and OVF (VirtualBox)
  - Both are AMD64 (no modern Mac support)
- Build it yourself
  - CentOS, Ubuntu, Rocky supported
  - Docker, so may run on other flavors
- Instructions for each in coming slides

VMs that can be downloaded

VMWare

https://thunt-level1.s3.amazonaws.com/rita5-thunt-vmware.zip

Size: 3.9GB

SHA1: 4BEB757352149236718F16E50D5D461794028AFE

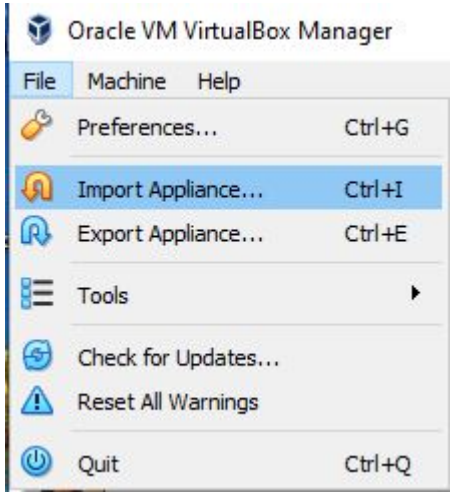
VirtualBox and generic OVF

https://thunt-level1.s3.amazonaws.com/rita5-thunt-ovf.zip

Size 6.8GB

SHA1: E4D4FABB34A0C975E07B46F5A93C43192AC4ED06

Convert OVF to VirtualBox



Then follow prompts to convert OVF to VirtualBox

To set up networking, follow these steps for SSH:

https://www.activecountermeasures.com/port-forwarding-with-virtualbox/

Build the lab system yourself

Spin up a modern Ubuntu, CentOS or Rocky system.
Login with sudo access and run the following commands:

```
wget https://thunt-level1.s3.amazonaws.com/thunt5-labs.tar.gz
tar xvzf thunt5-labs.tar.gz
```

This will create four directories labeled "lab1" through "lab4"
Next, run the following commands:

```
wget https://github.com/activecm/rita/releases/download/v5.0.8/install-rita-zeek-here.sh
chmod +x install-rita-zeek-here.sh
./install-rita-zeek-here.sh
```

Follow the prompts during the install. When prompted for the "BECOME" password, this is your sudo password. When the install is complete, you do not need to run the "zeek start" command.

<shameless_plugs>

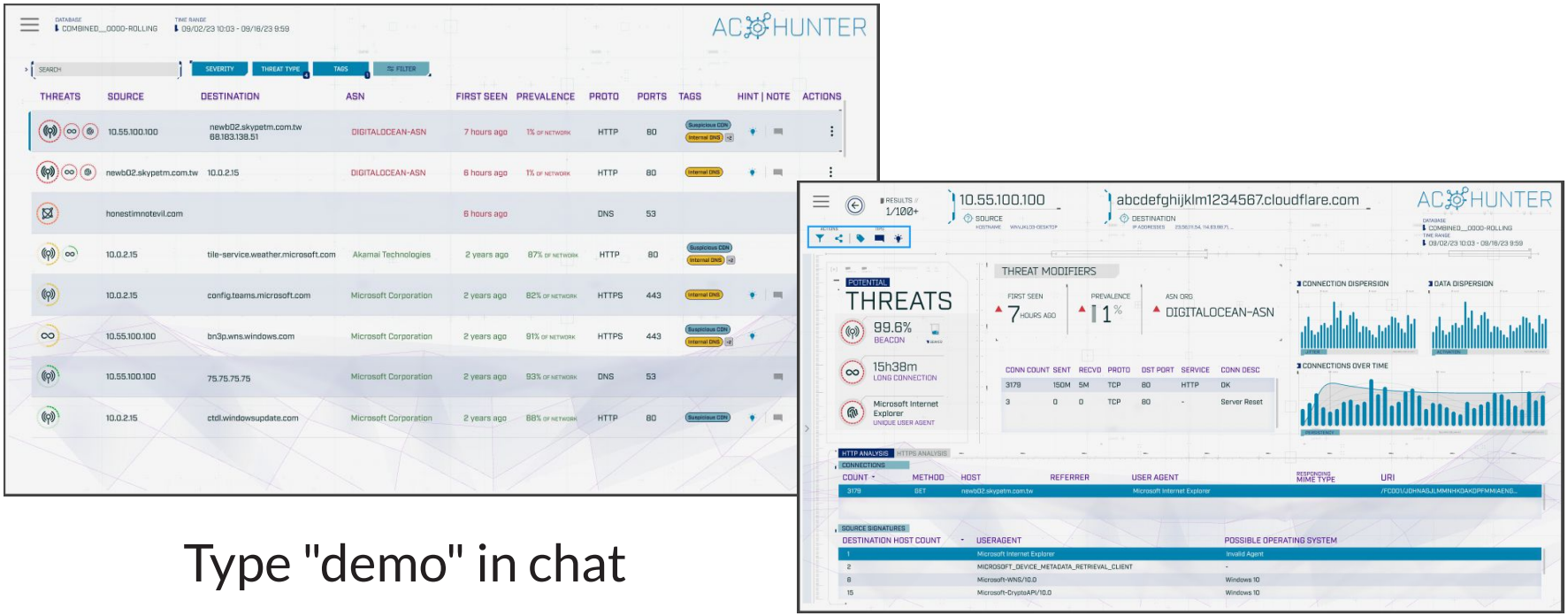
Classes I'm teaching

- Advance Network Threat Hunting
  - WWHF Oct 8th & 9th
  - Virtual tickets still available
- Intro to Docker (new - pay what you can)
- Intro to Packet Decoding (pay what you can)
- Security Compliance & Leadership

https://www.antisyphontraining.com/mission/our-instructors/instructor-profile-chris-brenton/

Want an AC-Hunter demo?

Updated version coming soon!



Type "demo" in chat

</shameless_plugs>

Logistics

- 10 minute break at top of each hour
- 20 minute break at 3 hour point
- Use the Discord channel for discussion
  - #acm-webcast-chat channel
- The team is monitoring for your questions

Help with command line syntax

- We'll be working at the command line
- Some are nested commands

```
<command> | <command> | <command>
```

- I'll explain what's going on
- Try adding one command at a time to observe how it changes the output

https://www.explainshell.com/

Goals for this class

- Define "cyber threat hunting"
- Identify how to perform a threat hunt
- Define and identify connection persistency
- Learn how to investigate endpoints
- Hands on lab time running down real C2 channels used in the wild

What is threat hunting?

- Actively searching your environment for compromised systems
- Triggered by time or process, not by alerts
- Validate the integrity state of every system
  - Not just desktops and servers
  - Not just systems submitting logs to your SIEM
  - Not just the patterns you can hypothesise
- Output is a compromise assessment

But I hunt my SIEM...

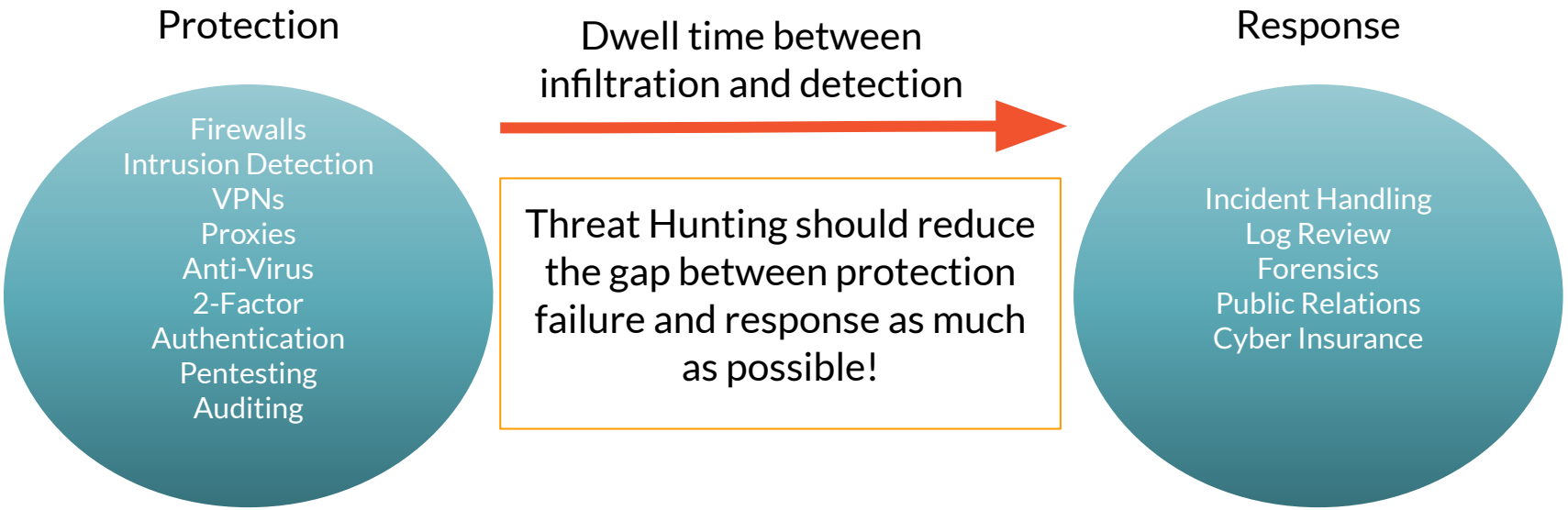
- You don't see everything
- Using data from compromised host
- Do you spend your time "hunting" or tuning the SIEM?
- Most security frameworks require SIEM
- And yet we are seeing no improvement
- Not the silver bullet we thought it was

https://www.activecountermeasures.com/check-the-stats-your-threat-hunting-is-probably-broken/

But AI will fix it, right?

- AI is vaporware and just a marketing term
- What we really have is machine learning
  - Machines do not always "learn" what we want
  - Unexpect bias in the datasets due to lack of real intelligence
  - Neural network AI is extremely challenging to troubleshoot
- Deployment is exceeding our ability to improve
  - Results are not always logical (black and asian nazis)
  - Sometimes they lie a lot (will make up data and news)
  - Write haiku's flaming their owners (can be lead astray)
  - Run over and drag pedestrians (yes this has happened)
  - Run polls to guess the cause of death (no real intelligence means no empathy)
  - Need to remove features to not be racist (this is just sad)
  - Teach kids how to make their disorder worse
  - Diagnose tuberculosis based on age of MRI machine
- If it fails in security, how long before you can tell?

The Purpose of Threat Hunting



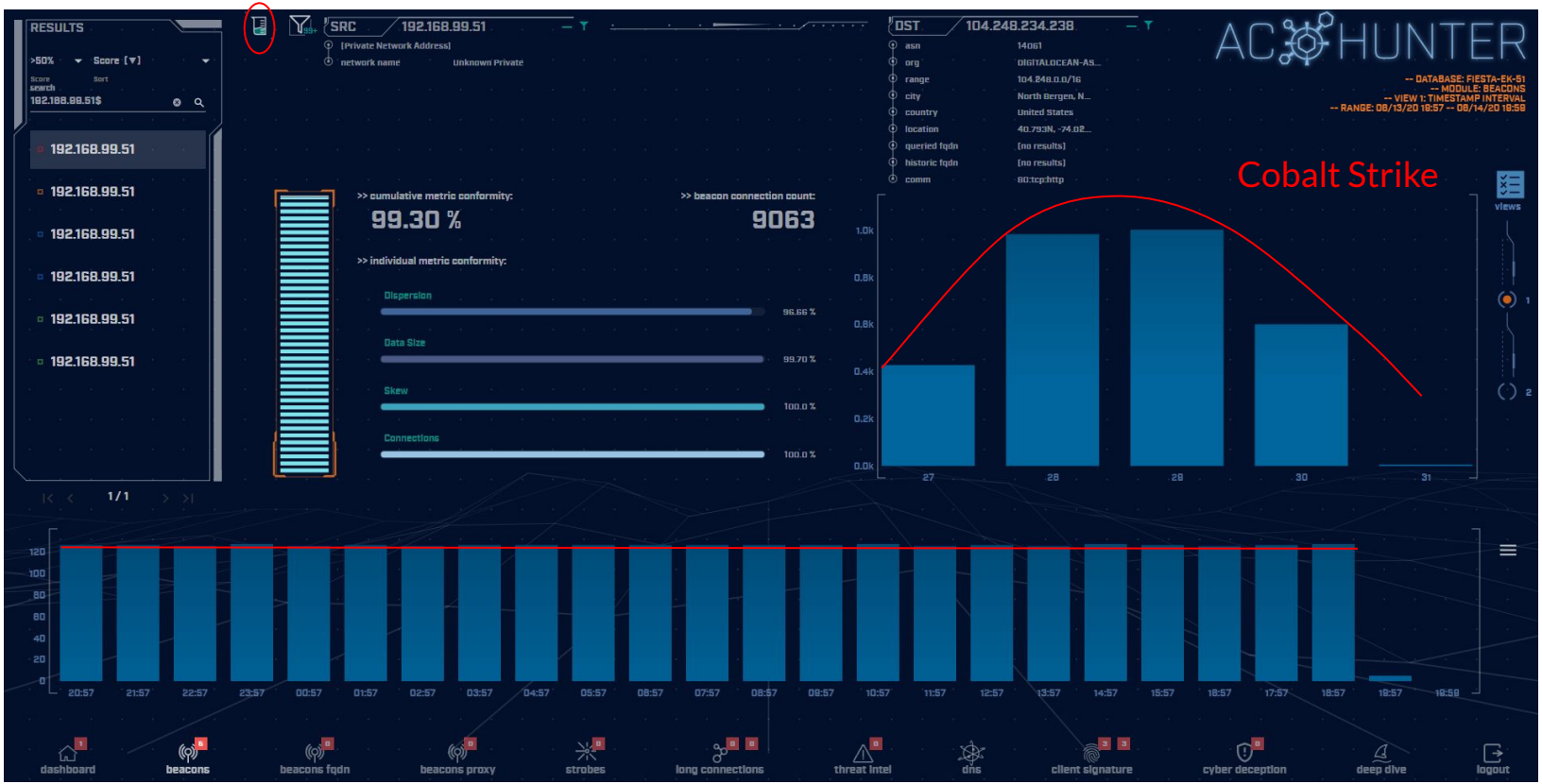
Start with the network

- The network is the great equalizer
  - You see everything, regardless of platform
  - Desktop, servers, IIoT, etc all reviewed the same
- You can hide processes but not packets
- Malware is usually controlled
  - Which makes targeting C2 extremely effective
  - Identify compromise when C2 "calls home"
  - Must be frequent enough to be useful
- Wide view so you can target from there

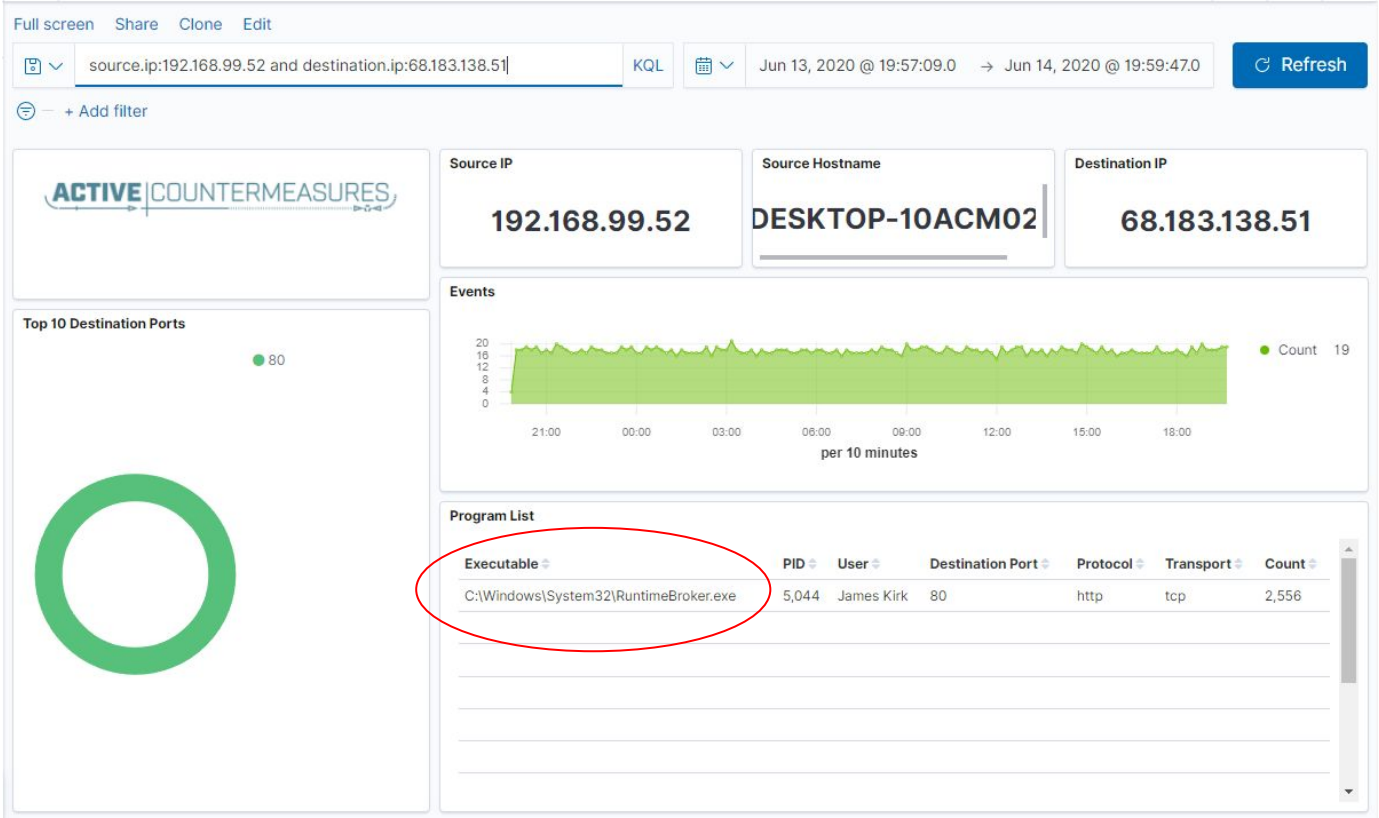
The threat hunting process

- Identify connection persistency
- Business need for connection?
  - Reputation check of external IP
- Abnormal protocol behaviour
- Investigation of internal IP
- Disposition
  - No threat detected = add to safelist
  - Compromised = Trigger incident handling

Start on the network



THEN pivot to the system logs



Don't cross "the passive/active line"

- All threat hunting activity should be undetectable to an adversary
- Passive in nature
  - Review packets
  - Review SIEM logs
- If active techniques are required, we should trigger incident response first
  - Example: Isolating the suspect host
  - Example: Running commands on suspect host

Why have a passive/active line?

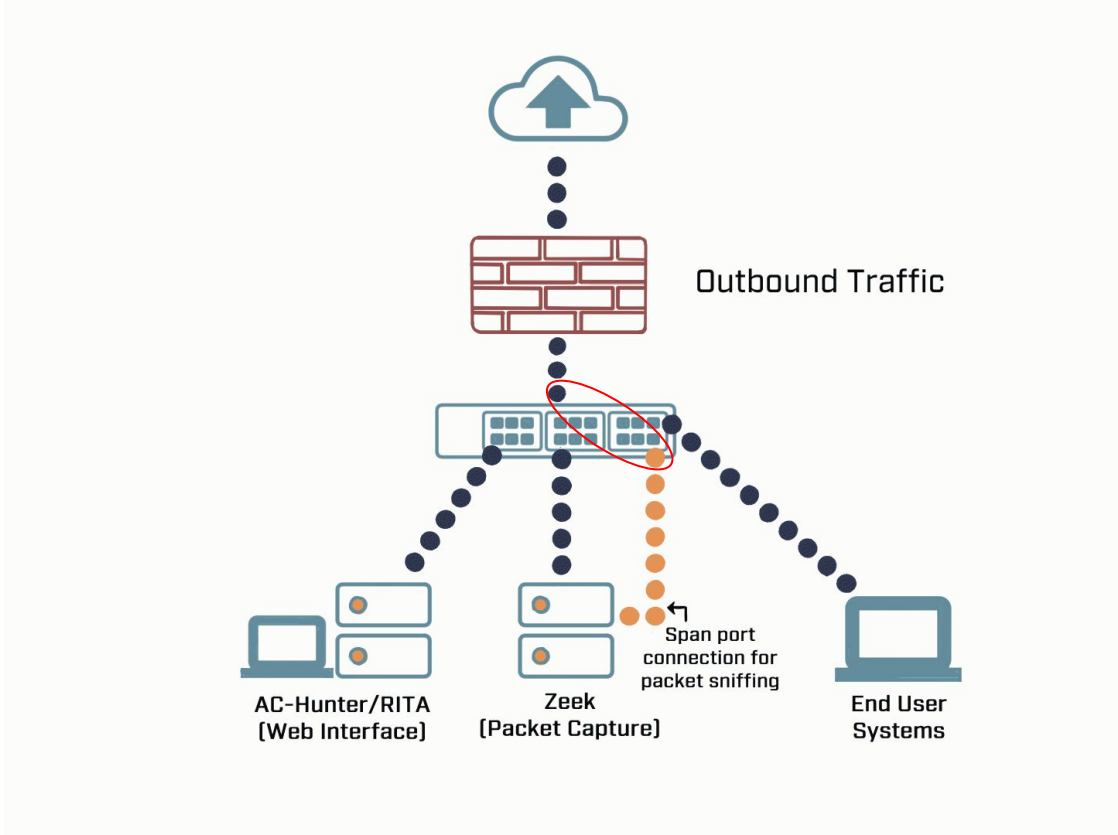
- Run local commands to check system
  - Attacker now knows you are on to them
  - Are you maintaining a proper chain of custody?
  - Are you sure you know what that means?
  - This can impact:
    - Integrity of forensics
    - Law enforcement or legal involvement
- Isolate the system
  - Attacker activates secondary channel
  - Can identify you are now on to them

C2 Detection Techniques

Where to Start

- Monitor traffic to and from the Internet
  - Monitor internal interface of firewall
- Packet captures or Zeek data
- Analyze in large time blocks
  - More data = better fidelity
  - Minimum of 12 hours, 24 is ideal
- Analyze communications in pairs
  - Every outbound session passing the firewall
  - Ignore internal to internal (high false positive)

Typical deployment



Does targeting C2 have blind spots?

- Attackers motivated by gain
  - Information
  - Control of resources
- Sometimes "gain" does not require C2
  - Just looking to destroy the target
  - Equivalent to dropping a cyber bomb
  - We are talking nation state at this level
- NotPetya
  - Worm with no C2 designed to seek and destroy
  - These are rare as they frequently go sideways

Start by checking persistency

- Focus on persistent connections
  - Internal system in constantly initiating connections with an outside "system"
  - Long connections
  - Beacons
- Persistent connections should have an identifiable business need
  - Checking the time
  - Checking for patches

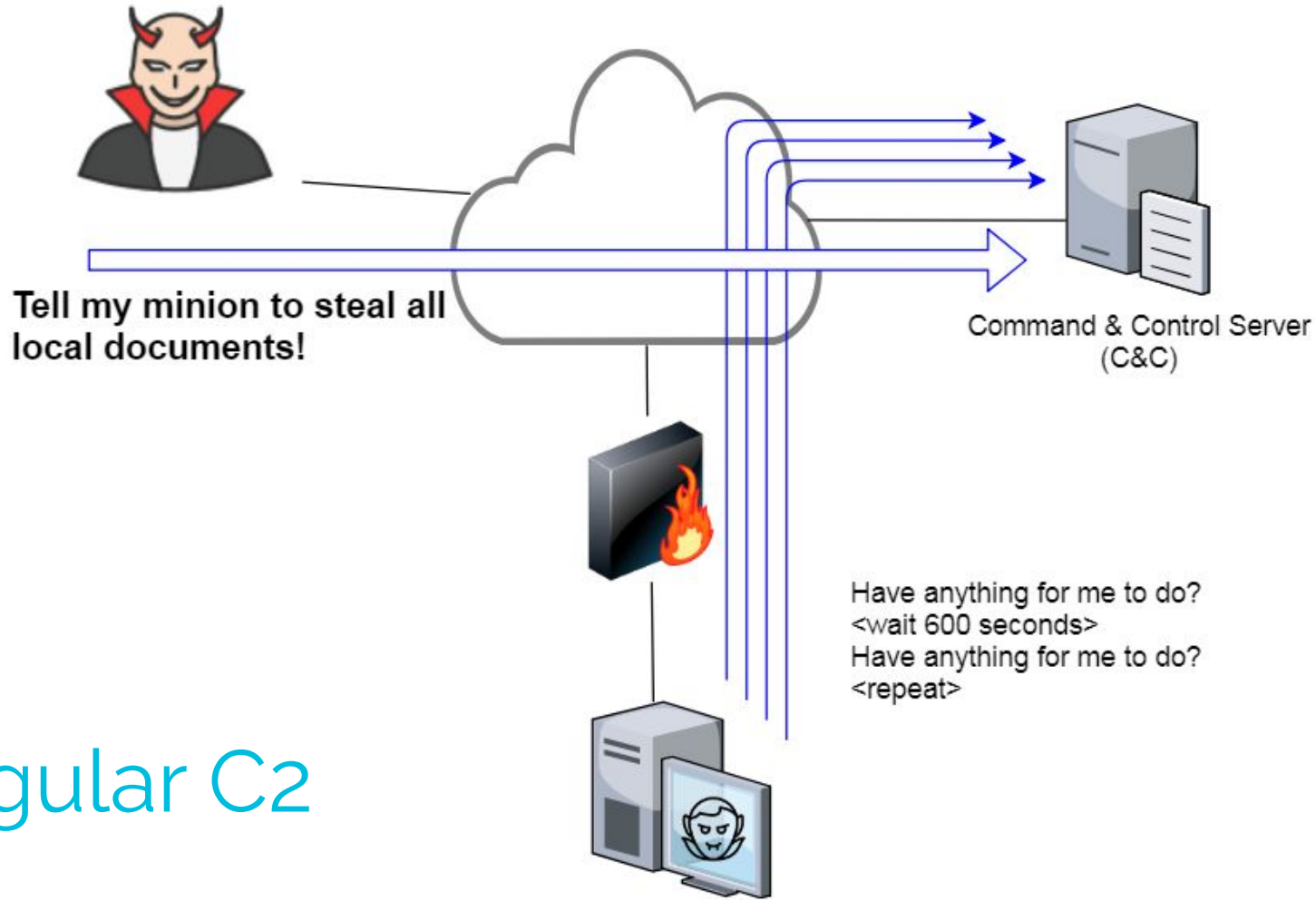
Long connections

- You are looking for:
- Total time for each connection
  - Which ones have gone on the longest?
- Cumulative time for all pair connections
  - Total amount of time the pair has been in contact
- Can be useful to ignore ports or protocols
  - C2 can change channels

Long connection example

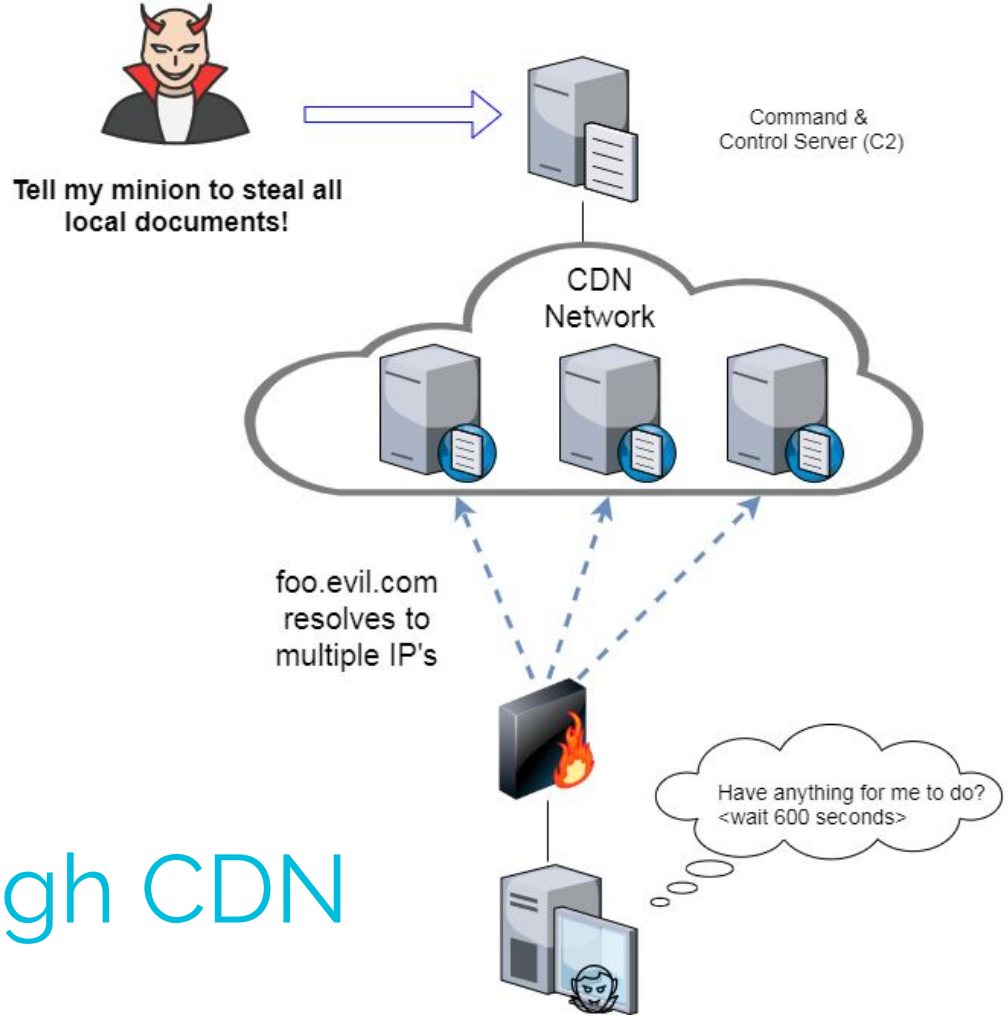
What is a beacon?

- Repetitive connection establishment between two IP addresses
  - Easiest to detect
- Repetitive connection establishment between internal IP and FQDN
  - Target can be spread across multiple IP's
    - Usually a CDN provider
  - Target IPs also destination for legitimate traffic
  - Far more difficult to detect


Tell my minion to steal all local documents!
Command & Control Server (C&C)
Have anything for me to do?
<wait 600 seconds>
Have anything for me to do?
<repeat>

Regular C2

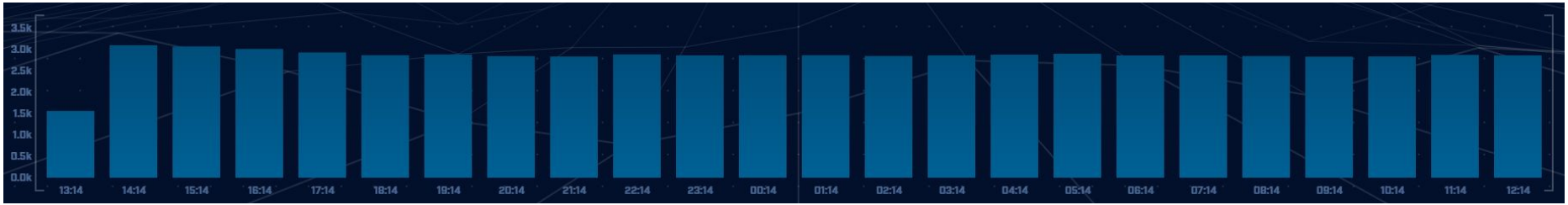
C2 through CDN



Beacon detection based on timing

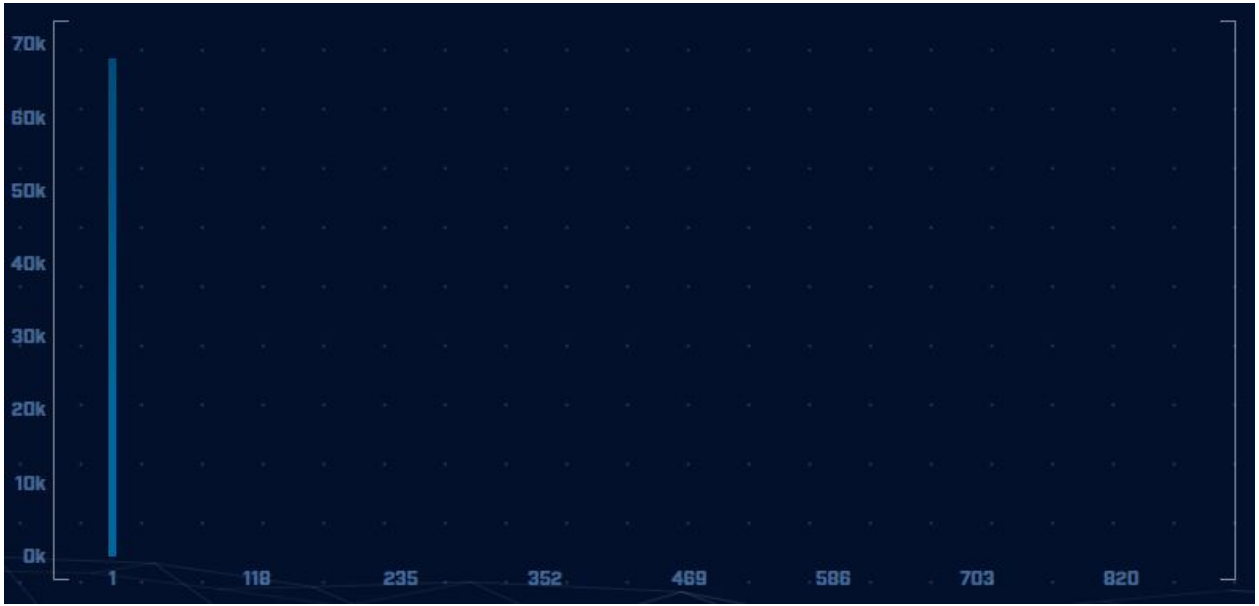
- May follow an exact time interval
  - Technique is less common today
  - Detectable by k-means
  - Potential false positives
- May introduce "jitter"
  - Vary connection sleep delta
  - Avoids k-means detection
  - False positives are extremely rare
- Short enough delta for terminal activities

Connection quantity VS time



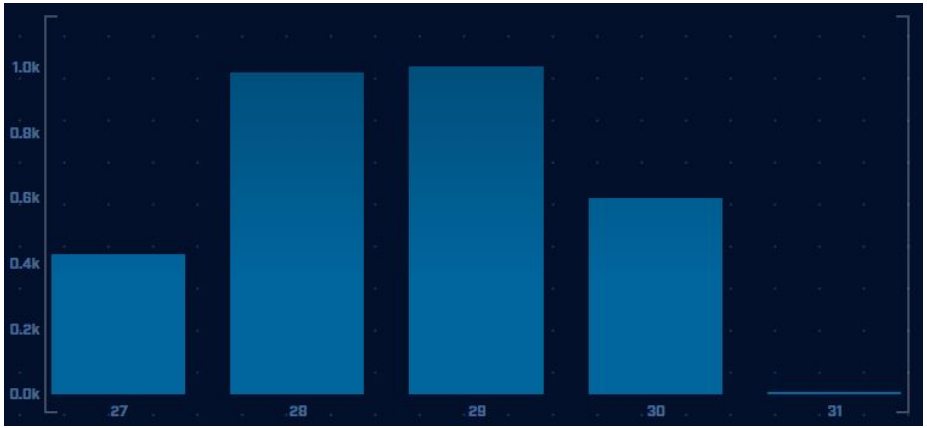
Each bar represents the number of times the source connected to the destination during that one hour time block

Connect time deltas with no jitter



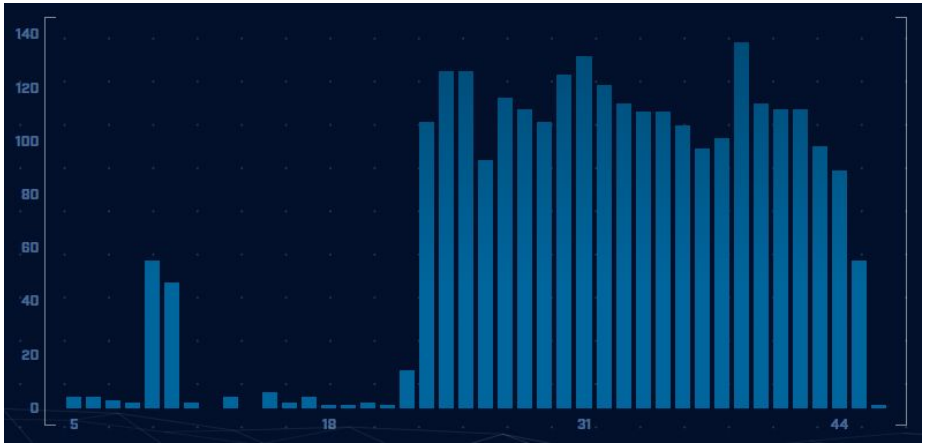
How often a specific time delta was observed

Connection time deltas with jitter



Cobalt Strike will typically produce a bell curve

Pretty well randomized but still a small dwell time "window"



When you don't have a GUI

of connections

```
student@thunt:~/lab3$ beacon-tshark lab3.pcap 192.168.100.136 172.208.51.75
   499 12
   555 13
   556 14
   555 15
   550 16
   555 17
   554 18
   564 19
   551 20
   549 21
   558 22
   557 23
   553 00
   555 01
   556 02
   555 03
   548 04
   548 05
   552 06
   552 07
   557 08
   549 09
   556 10
   554 11
    43 12
```

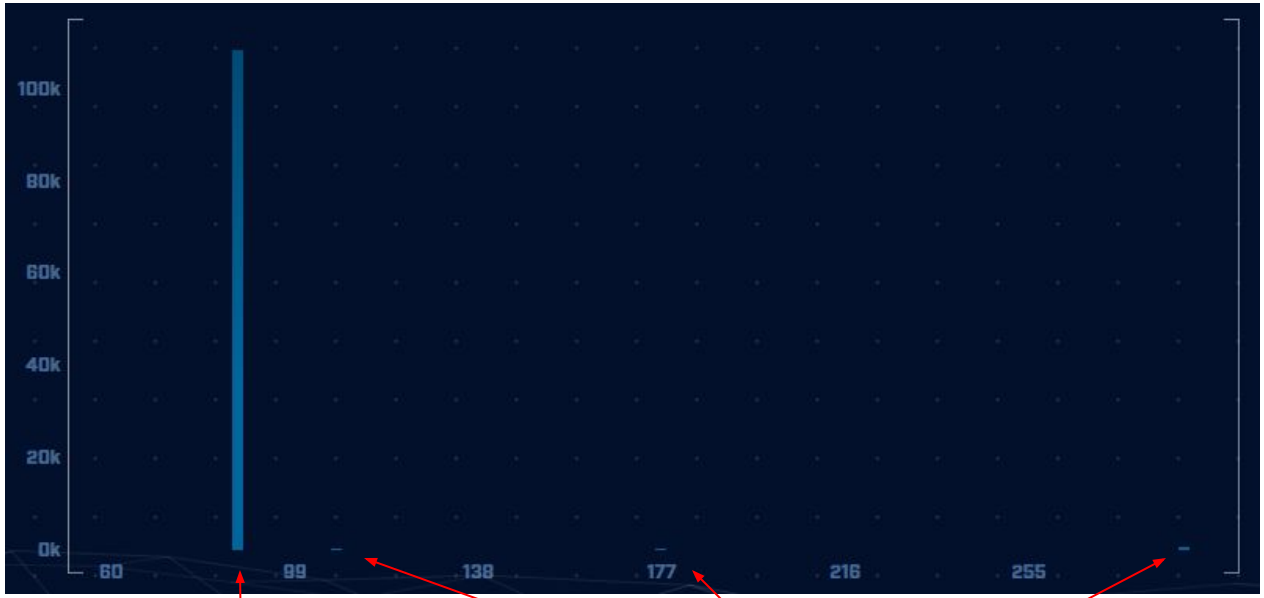
Hour of the day

Runs slower than
equivalent scripts for Zeek

Detection based on session size

- Focuses on detection of the heartbeat
  - Useful for C2 over social media
- Variations from the heartbeat indicate activation of C2 channel
- Session size can help reveal info regarding commands being issued
- Possible to randomly pad but this is extremely rare

Session size analysis



Safelisting

- Not all persistence is "evil"
- Could be part of normal operations
  - Keep computer time in sync
  - Checking for patches
  - Checking on an external service
- When business need can be identified, we should safelist the connection
  - Keep it out of future hunts
  - Don't make safelists any broader than necessary

Identifying business need

- Do you recognize the domain?
  - microsoft.com
  - windows.com
  - ntp.org
- Can you relate the services to a specific department?
- The purchasing group can be helpful
  - Find the company behind the domain
  - Are we purchasing services from them?

Check destination IP address

- Start simple
  - Who manages ASN?
  - Geolocation info?
  - IP delegation
  - PTR records
- Do you recognize the target organization?
  - Business partner or field office
  - Current vendor (active status)
- Other internal IP's connecting?

Some helpful links

https://www.abuseipdb.com/check/<IP Address>

https://otx.alienvault.com/indicator/ip/<IP Address>

https://search.censys.io/hosts/<IP Address>

https://dns.google/query?name=<IP Address>

https://www.google.com/search?q=<IP Address>

https://www.onyphe.io/search/?query=<IP Address>

https://securitytrails.com/list/ip/<IP Address>

https://www.shodan.io/host/<IP Address>

https://www.virustotal.com/gui/ip-address/<IP Address>/relations

C2 Detection Techniques Part 2

What next?

- You've identified connection persistence
- You can't identify a business need
- Next steps
  - Protocol analysis
  - Reputation check of external target
  - Investigate internal IP address

Zeek decodes many apps

- Detect over 55 applications
  - HTTP, DNS, SIP, MYSQL, RDP, NTLM, etc. etc.
- Fairly easy to add new ones
  - Example: HL7 if you are in healthcare
- Checks all analyzers for each port
- Does not assume WKP = application

Zeek example

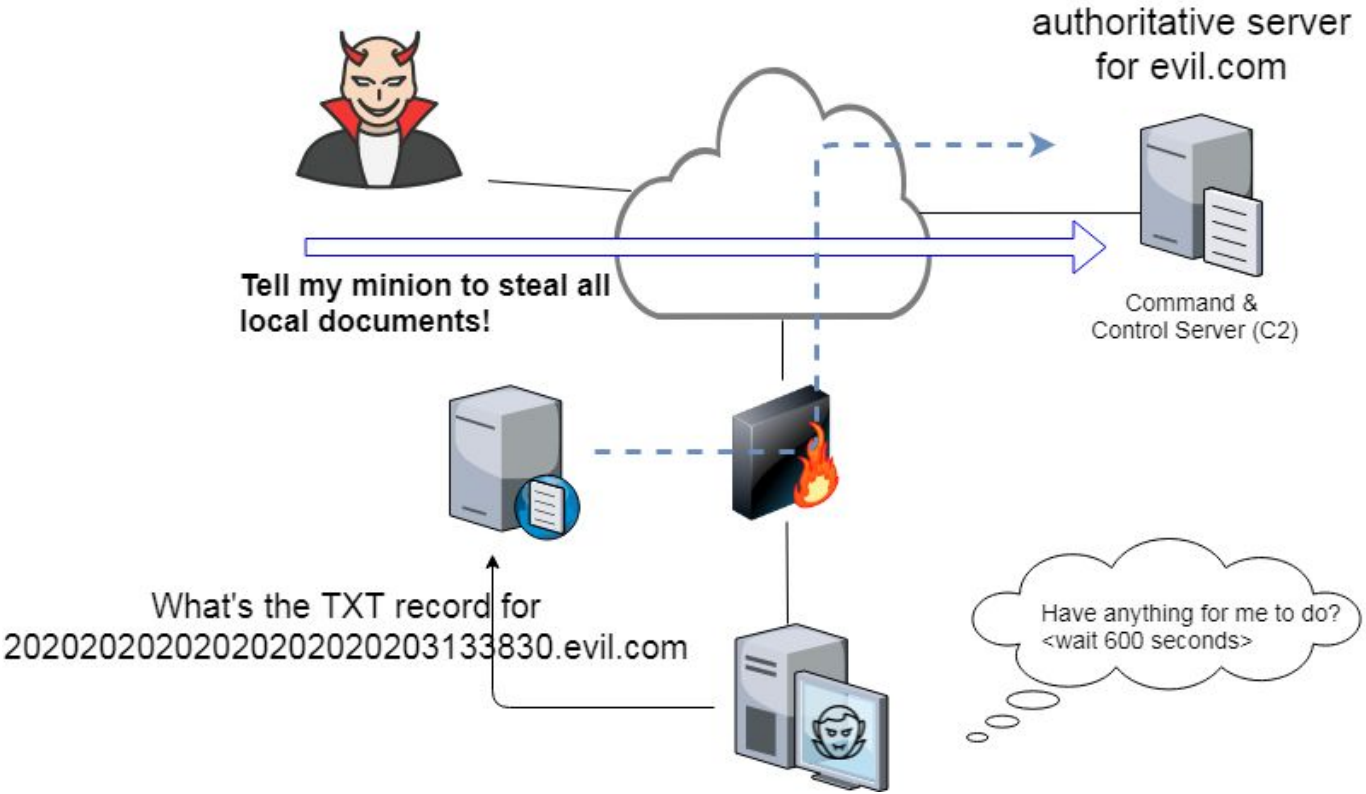
```
thunt@thunt-labs:~/lab1$ cat conn.log | zeek-cut id.orig_h id.resp_h id.resp_p
 proto service orig_ip_bytes resp_ip_bytes | column -t | head
192.168.99.51               104.248.234.238  80    tcp   http  689         403
192.168.99.51               23.223.200.136   80    tcp   -     80          40
192.168.99.51               104.248.234.238  80    tcp   http  729         443
192.168.99.52               224.0.0.251      5353  udp   dns   344         0
fe80::d048:42e0:8448:187c   ff02::fb         5353  udp   dns   424         0
fe80::d048:42e0:8448:187c   ff02::1:3        5355  udp   dns   81          0
192.168.99.52               224.0.0.252      5355  udp   dns   61          0
fe80::d048:42e0:8448:187c   ff02::1:3        5355  udp   dns   81          0
192.168.99.52               224.0.0.252      5355  udp   dns   61          0
192.168.99.51               104.248.234.238  80    tcp   http  689         403
thunt@thunt-labs:~/lab1$
```

AC-Hunter example

Unexpected protocol use

- Look for unknown protocols on standard ports (TCP/80, TCP/443, etc)
- Attackers may bend but not break rules
- This can result in:
  - Full protocol compliance
  - Abnormal behaviour
- Need to understand "normal"
  - For the protocol
  - For your environment

C2 over DNS


authoritative server
for evil.com
Tell my minion to steal all
local documents!
Command &
Control Server (C2)
What's the TXT record for
20202020202020202020203133830.evil.com
Have anything for me to do?
<wait 600 seconds>

Example: Too many FQDNs

- How many FQDNs do domains expose?
  - Most is < 10
  - Recognizable Internet based vendors 200 - 600
    - Microsoft
    - Akamai
    - Google
    - Amazon
- Greater than 1,000 is suspicious
- Could be an indication of C2 traffic

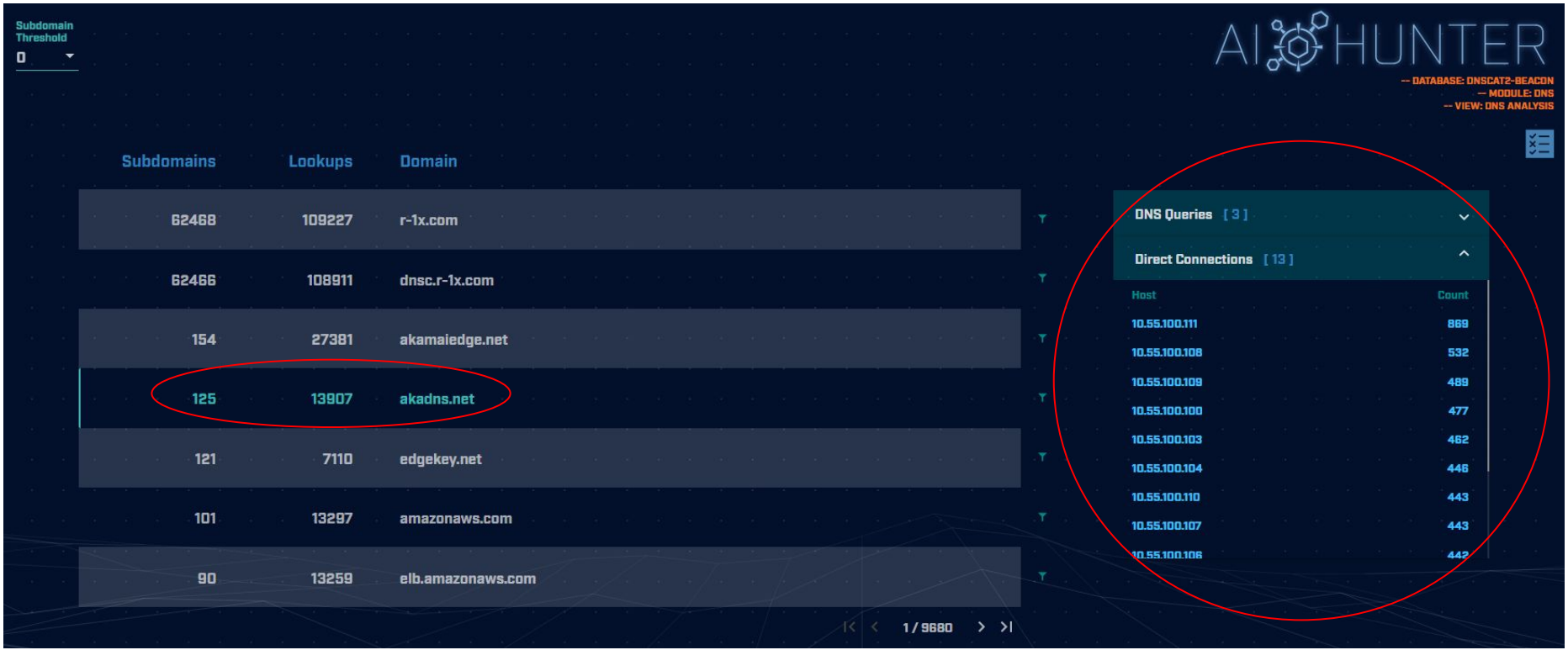
Detecting C2 over DNS

	FQDNs Count	Lookups	Domain
⚠	62468	109227	r-1x.com
⚠	62466	108911	dnsc.r-1x.com
	154	27381	akamaiedge.net
	125	13907	akadns.net

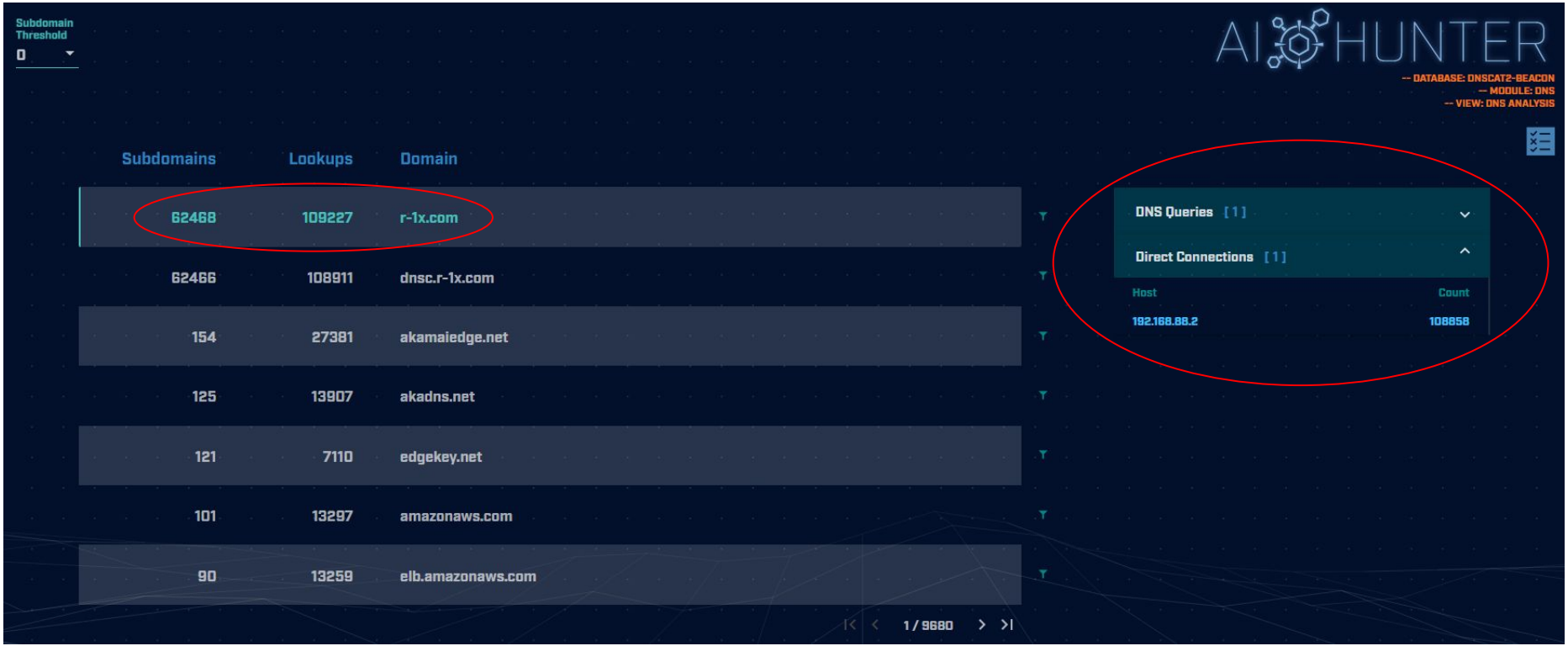
Bonus checks on DNS

- Check domains with a lot of FQDNs
- Get a list of the IPs returned
  - Need DNS answers, not just queries
- Compare against traffic patterns
  - Are internal hosts visiting this domain?
  - Is it just your name servers?
- Unique trait of C2 over DNS
  - Lots or FQDN queries
  - But no one ever connects to these systems

Normal DNS query patten



Things that make you go "hummm"



Look for odd HTTP user agents

```
ritabeakerlab@ritabeakerlab:~/lab1$ cat http.log | zeek-cut id.orig_h id.resp_h user_agent
 | grep 10.0.2.15 | sort | uniq | cut -f 3 | sort | uniq -c | sort -rn
     15 Microsoft-CryptoAPI/10.0
     12 Microsoft-WNS/10.0
      1 Mozilla/5.0 (Windows; U; MSIE 7.0; Windows NT 5.2) Java/1.5.0_08
ritabeakerlab@ritabeakerlab:~/lab1$
```

10.0.2.15 identifies itself as:

Windows 10 when speaking to 27 different IP's on the Internet
Windows XP when speaking to one specific IP on the Internet

Unique SSL Client Hello: Zeek + JA3

SSL/TLS Hash	Seen	Requests	Sources
5e573c9c9f8ba720ef9b18e9fce2e2f7	1	clientservices.googleapis.com	10.55.182.100
bc6c386f480ee97b9d9e52d472b772d8	2	clients4.google.com, 556-emw-319.mktoresp.com	10.55.182.100
f3405aa9ca597089a55cf8c62754de84	2	builds.cdn.getgo.com	10.55.182.100
28a2c9bd18a11de089ef85a160da29e4	2	mediaredirect.microsoft.com	10.55.100.105, 10.55.182.100
08bf94d7f3200a537b5e3b76b06e02a2	4	files01.netgate.com	192.168.88.2

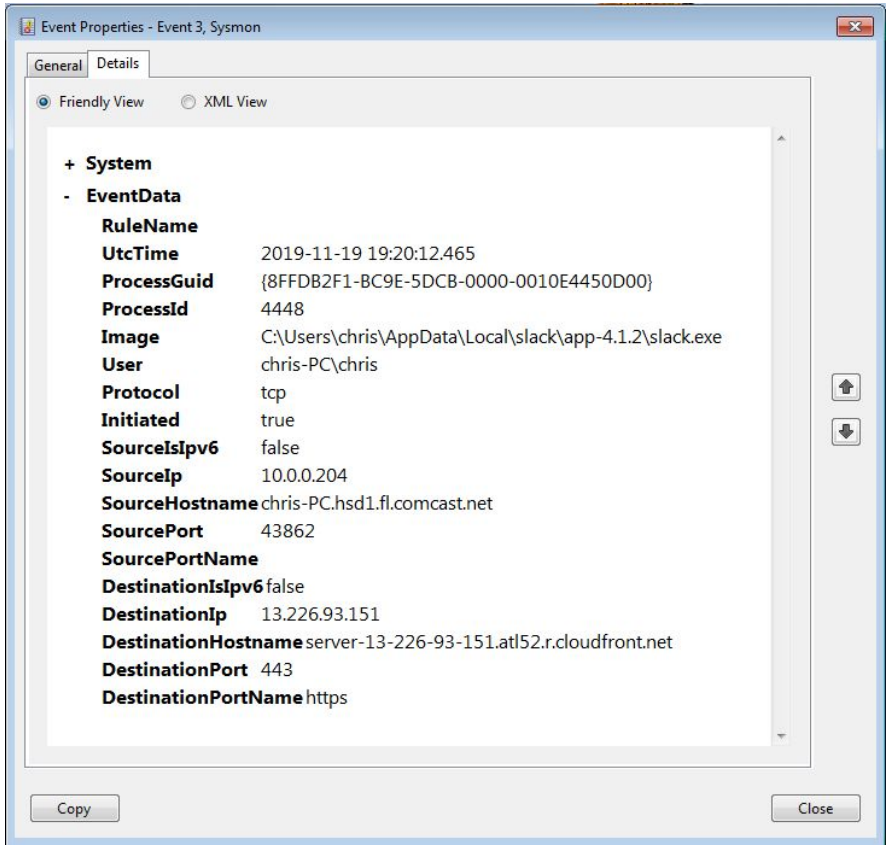
Internal system

- Info available varies greatly between orgs
- Inventory management systems
- Security tools like Carbon Black
- OS projects like BeaKer
- Internal security scans
- DHCP logs
- Login events
- Passive fingerprinting

Leverage internal host logging

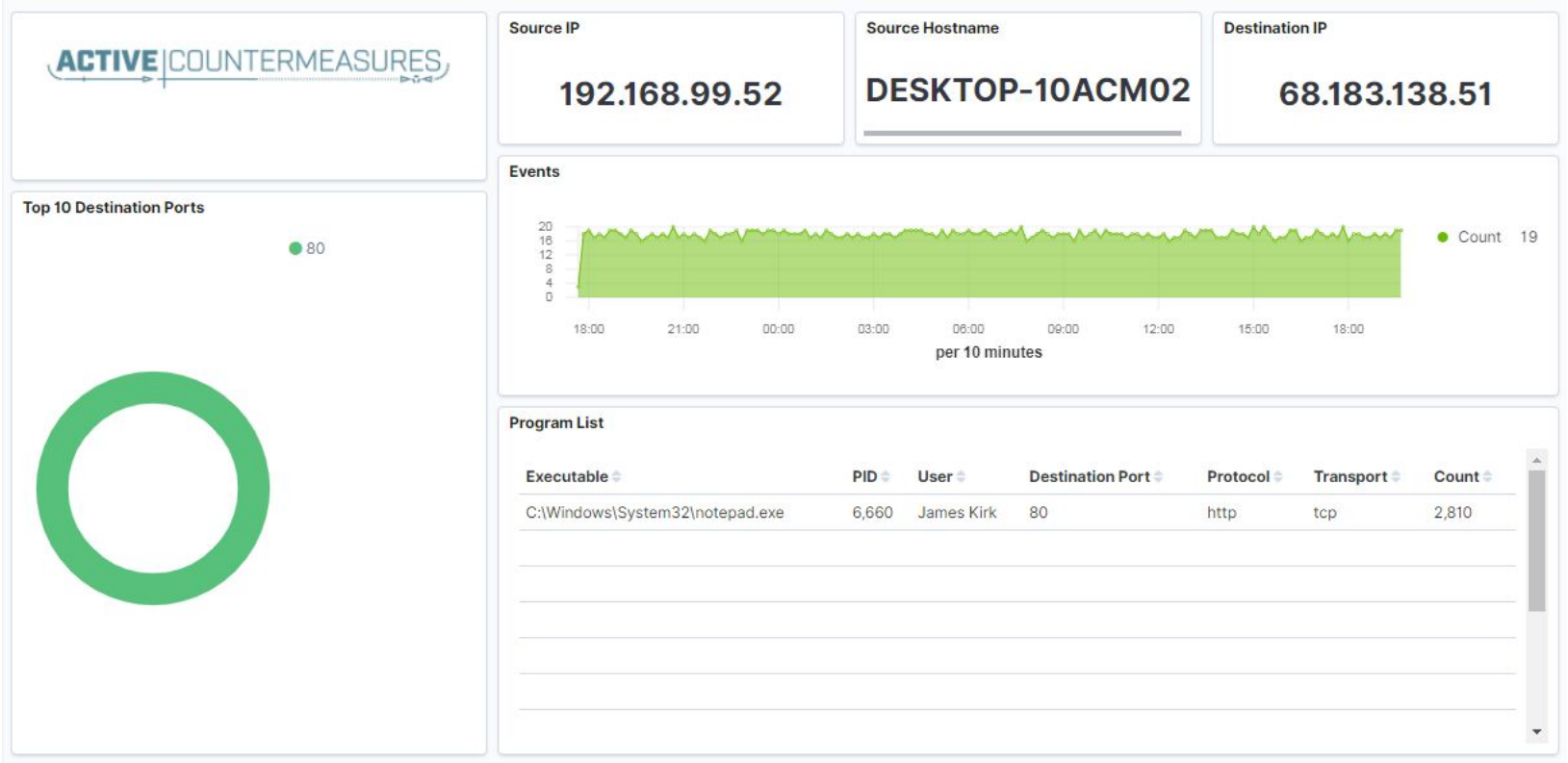
- Network shows suspicious traffic patterns
- Use this data to pivot to host logs
- Filter your logs based on:
  - Suspect internal host
  - Timeframe being analyzed
- Anything stand out as unique or odd?

Sysmon Event ID Type 3's



Map outbound connections to the applications that created them.

Sysmon Type 3 + BeaKer



But I have no system logs!

- Good time to start collecting them
- Full packet captures from system
- Apply additional network tools to collect more data
- Just remember, no detectable actions until we trigger incident response mode!
  - Don't cross the active/passive line

What next?

- Disposition session
  - "I think it's safe" = add to safelist
  - "I think we've detected a compromise" = Incident response mode
- Remember to leave no footprints
  - All actions undetectable to potential adversaries
  - Passive activities only
- Incident response may include active tasks

Network Threat Hunting Tools

tshark

- What's it good for?
  - Extracting interesting fields from packet captures
  - Multiple passes to focus on different attributes
  - Combine with text manipulation tools
  - Can be automated
- When to use it
  - Both major and minor attributes
- Where to get it

https://www.wireshark.org/

Tshark example - DNS queries

```
$ tshark -r thunt-lab.pcapng -T fields -e dns.qry.name
udp.port==53 | head -10

6dde0175375169c68f.dnsc.r-1x.com
6dde0175375169c68f.dnsc.r-1x.com
0b320175375169c68f.dnsc.r-1x.com
0b320175375169c68f.dnsc.r-1x.com
344b0175375169c68f.dnsc.r-1x.com
344b0175375169c68f.dnsc.r-1x.com
0f370175375169c68f.dnsc.r-1x.com
0f370175375169c68f.dnsc.r-1x.com
251e0175375169c68f.dnsc.r-1x.com
251e0175375169c68f.dnsc.r-1x.com
```

Tshark example - user agents

```
$ tshark -r sample.pcap -T fields -e http.user_agent tcp.
dstport==80 | sort | uniq -c | sort -n | head -10
      2 Microsoft Office/16.0
      2 Valve/Steam HTTP Client 1.0 (client;windows;10;1551832902)
      3 Valve/Steam HTTP Client 1.0
     11 Microsoft BITS/7.5
     11 Windows-Update-Agent
     12 Microsoft-CryptoAPI/6.1
    104 PCU
```

Finding display filters

tshark -G | grep '\shttp\.' | less -S -x30

```
F                               Notification                    http.notification               FT_BOO
F                               Response                        http.response                   FT_BOO
F                               Request                         http.request                    FT_BOO
F                               Response number                 http.response_number            FT_UIN
F                               Request number                  http.request_number             FT_UIN
F                               Credentials                     http.authbasic                  FT_STR
F                               Citrix AG Auth                  http.authcitrix                 FT_BOO
F                               Citrix AG Username              http.authcitrix.user            FT_STR
F                               Citrix AG Domain                http.authcitrix.domain          FT_STR
F                               Citrix AG Password              http.authcitrix.password        FT_STR
F                               Citrix AG Session ID            http.authcitrix.session         FT_STR
F                               Response line                   http.response.line              FT_STR
F                               Request line                    http.request.line               FT_STR
:_
```

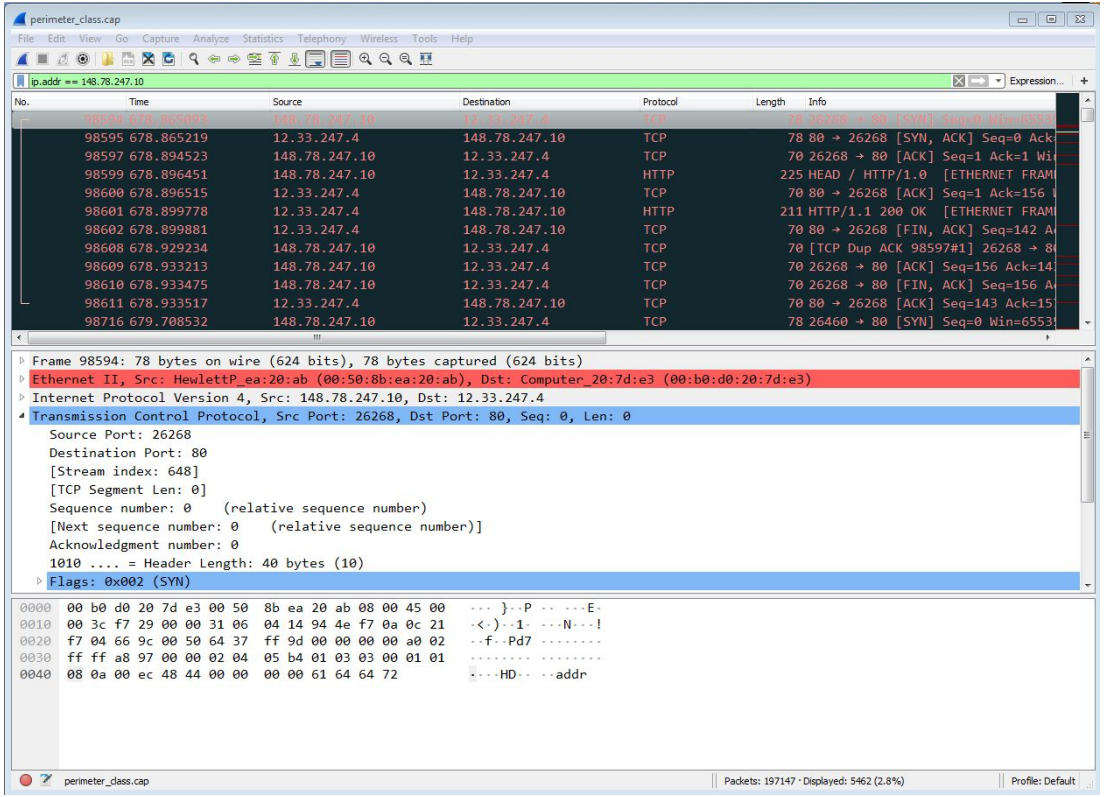
There are just under 185K different display filters!

Wireshark

- What's it good for?
  - Packet analysis with guardrails
  - Stream level summaries
- When to use it
  - As part of a manual analysis
  - When steps cannot be automated
- Where to get it

https://www.wireshark.org/

Useful when I have a target



Zeek

- Network recorder
- What's it good for?
  - Near real time analysis (1+ hour latency)
  - More storage friendly than pcaps
- When to use it
  - When you need to scale
  - When you know what attributes to review
- Docker version included with RITA install

Zeek example - cert check

```
$ cat ssl* | zeek-cut id.orig_h id.resp_h id.resp_p
validation_status | grep 'self signed' | sort | uniq
122.228.10.51   192.168.88.2    9943    self signed certificate in
certificate chain
24.111.1.134    192.168.88.2    9943    self signed certificate in
certificate chain
71.6.167.142    192.168.88.2    9943    self signed certificate in
certificate chain
```

-d for human readable times

- Zeek-cut prints epoch time by default
- "-d" converts to human readable

```
cbrenton@cbrenton-beacon-src-test:~/foo$ cat conn.01\:00\:00-02\
:00\:00.log | zeek-cut ts id.orig_h | head -8
1645578000.318671       167.172.154.151
1645578000.318784       167.172.154.151
1645578000.318841       167.172.154.151
1645578000.334906       167.172.154.151
1645578000.334948       167.172.154.151
1645578000.334977       167.172.154.151
1645578001.228742       167.172.154.151
1645578001.360749       167.172.154.151
cbrenton@cbrenton-beacon-src-test:~/foo$ cat conn.01\:00\:00-02\
:00\:00.log | zeek-cut -d ts id.orig_h | head -8
2022-02-23T01:00:00+0000        167.172.154.151
2022-02-23T01:00:00+0000        167.172.154.151
2022-02-23T01:00:00+0000        167.172.154.151
2022-02-23T01:00:00+0000        167.172.154.151
2022-02-23T01:00:00+0000        167.172.154.151
2022-02-23T01:00:00+0000        167.172.154.151
2022-02-23T01:00:01+0000        167.172.154.151
2022-02-23T01:00:01+0000        167.172.154.151
cbrenton@cbrenton-beacon-src-test:~/foo$
```

zcutter.py

- zeek-cut limited to CSV format
- What if you use JSON?
- zcutter.py to the rescue!
- Like zeek-cut, but supports CSV & JSON
- Will processed compressed files

https://raw.githubusercontent.com/activecm/zcutter/main/zcutter.py

Internal info collection

- Internal IP can be ambiguous
- Generating better intel
  - Host logging
  - Passer - General info collected from the wire
  - Smudge - Passive fingerprinting
  - Internal zone transfers
  - EDR like Carbon Black
  - ADR like wazah
  - Forensics tools like Velociraptor

Datamash

- What's it good for?
  - Similar to the R-base tools, but more extensive
  - Performing simple calculation on data
- When to use it
  - Performing calculations on multiple lines
  - Statistical analysis
- Where to get it

https://www.gnu.org/software/datamash/
sudo apt install datamash

Datamash

- Used for processing raw data at the command line
- Great for sifting through tabulated data
  - Like Zeek logs
- Can perform statistical analysis
  - Min, max, mean, etc.
  - Can add together values

Datamash example

```
cbrenton@cbrenton-lab-testing:~/lab3$ cat conn.log | zeek-cut
id.orig_h id.resp_h duration | sort -k3 -rn | head -5
192.168.1.105   143.166.11.10   328.754946
192.168.1.104   63.245.221.11   41.884228
192.168.1.104   63.245.221.11   31.428539
192.168.1.105   143.166.11.10   27.606923
192.168.1.102   192.168.1.1     4.190865

cbrenton@cbrenton-lab-testing:~/lab3$ cat conn.log | zeek-cut
id.orig_h id.resp_h duration | grep -v -e '^$' | grep -v '-' | sort |
datamash -g 1,2 sum 3| sort -k3 -rn | head -5
192.168.1.105   143.166.11.10   356.361869
192.168.1.104   63.245.221.11   73.312767
192.168.1.102   192.168.1.1     5.464553
192.168.1.103   192.168.1.1     4.956918
192.168.1.105   192.168.1.1     1.99374
```

Duplicate IPs

Beacon/Threat Simulator

- Permits you to test your C2 detection setup
- Target any TCP or UDP port
- Can jitter timing
- Can jitter payload size
- Not designed to exfiltrate data!

```
beacon-simulator.sh <target IP> 80 300 10 tcp 5000
```

Connect to TCP/80 on target IP every 300 seconds, +/-10 seconds, vary payload between 0-5,000 bytes

https://github.com/activecm/threat-tools

What if I need specific app data?

```
#beacon-test (included with class files)
while :
do
        curl -A 'Atari 2600 Frogger Browser' $1 >/dev/null 2>&1
        sleep $(shuf -i200-350 -n1)
done
```

Then run this command with screen:

```
screen -S c2 -d -m ~/bin/beacon-test <Target IP or FQDN>
```

Packet crafting tools like hping3 let you define payload

Create your own scripts!

```
cbrenton@cb-lab:~/lab1$ cat /bin/fq
echo 'DNS info'
cat dns.* | zeek-cut answers query | sort | uniq | grep -Fw $1
echo 'HTTP info'
cat http.* | zeek-cut id.resp_h host user_agent | sort | uniq | grep -Fw $1
echo 'TLS info'
cat ssl.* | zeek-cut id.resp_h server_name validation_status | sort | uniq | grep -Fw $1
cbrenton@cb-lab:~/lab1$ fq 69.172.216.56
DNS info
anycast.fw.adsafeprotected.com,69.172.216.56    fw.adsafeprotected.com
HTTP info
TLS info
69.172.216.56   fw.adsafeprotected.com  ok
cbrenton@cb-lab:~/lab1$
```

Example script you can create to make life easier
"fq" check dns.log, http.log and ssl.log **in the local directory**
Returns info on specified IP address of FQDN
Use "zcat" if logs are in compressed format

Another script example

```
student@thunt:~/bin$ cat beacon-conn
cat conn.* | zcutter -d ts id.orig_h id.resp_h | grep $1 | grep $2 | sed 's/T/:/g' | cut -d ':' -f 2 | uniq -c | tr -s " " | awk '{ print $2 " " $1}'
student@thunt:~/bin$ _
```

```
student@thunt:~/lab1$ beacon-conn 10.0.2.15           68.183.138.51
19 28
20 119
21 44
20 1
21 76
22 119
23 120
00 119
01 120
02 119
03 120
04 119
05 120
06 119
07 120
08 119
09 120
10 119
11 120
12 119
13 120
14 119
15 120
16 119
17 120
18 119
19 92
student@thunt:~/lab1$ _
```

```
student@thunt:~/bin$ ll
total 100
drwxrwxr-x  2 student student  4096 Aug 27 17:20 ./
drwxr-x--- 12 student student  4096 Aug 30 16:21 ../
-rwxr-xr-x  1 student student   150 Aug 27 14:52 beacon-conn*
-rwxr-xr-x  1 student student   145 Aug 27 14:52 beacon-http*
-rwxr-xr-x  1 student student   151 Aug 27 14:52 beacon-ssl*
-rwxrwxr-x  1 student student   120 Aug 27 14:52 beacon-test*
-rwxr-xr-x  1 student student   715 Aug 27 14:52 beacon-tshark*
-rwxr-xr-x  1 student student   264 Aug 27 14:52 fq*
-rwxr-xr-x  1 student student 69281 Aug 27 14:52 zcutter*
student@thunt:~/bin$ _
```

C2 Labs & Walkthroughs

Walkthrough versus labs

- Walkthrough
  - I perform the steps, you follow along
  - Let's you see exactly what I'm doing and mimic
  - Usually the first experience with a tool or process
- Labs
  - I give you a problem to solve, you run with it
  - Next slide is "Hints" if you need help
  - Answer slides are after that
  - Reverse engineer if you are stuck

Working with RITA version 5

- We have completely changed the tool
- New backend, frontend and middleware
- Still working through some minor bugs
- Will call these out as we go through
- These are being addressed
- Should be fixed in the next release
- RITA will tell you when new versions drop

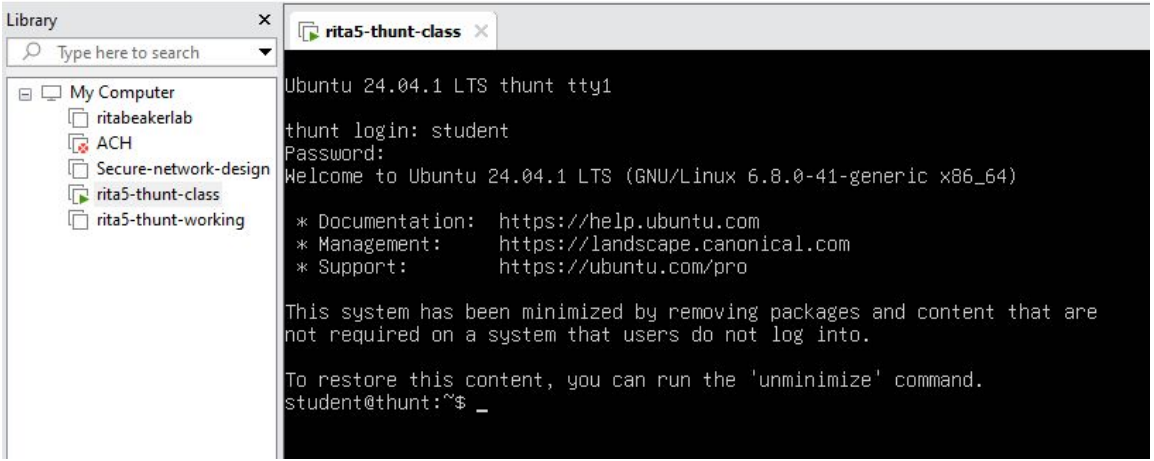
Walkthrough - SSH to VM

- Let's start by connecting to the VM
- I will be using SSH
- This will permit copy/paste of info
  - Like long strings of commands
  - Simplifies doing the labs
- Use the SSH tool of your choice
  - I'll be using SSH from Windows command line
  - I'll also be using SmarTTY

Caveats to this walkthrough

- I'm working with VMWare
- If you are running VirtualBox
  - Follow port forwarding instructions posted earlier
  - SSH to "student@127.0.0.1:10022"
- If you are running in public cloud
  - Follow vendor instructions to SSH to the system
  - System IP should be listed in their UI
- In both cases, just follow along with the walkthrough so you are familiar with the commands we are using

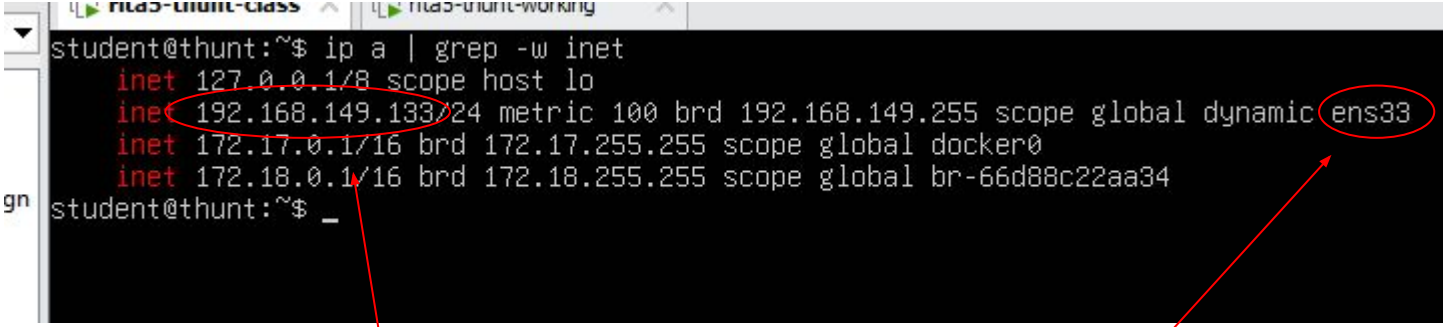
Login to VM



Via VM software console
Login: student Pass: findc2

Find the IP of your VMWare VM

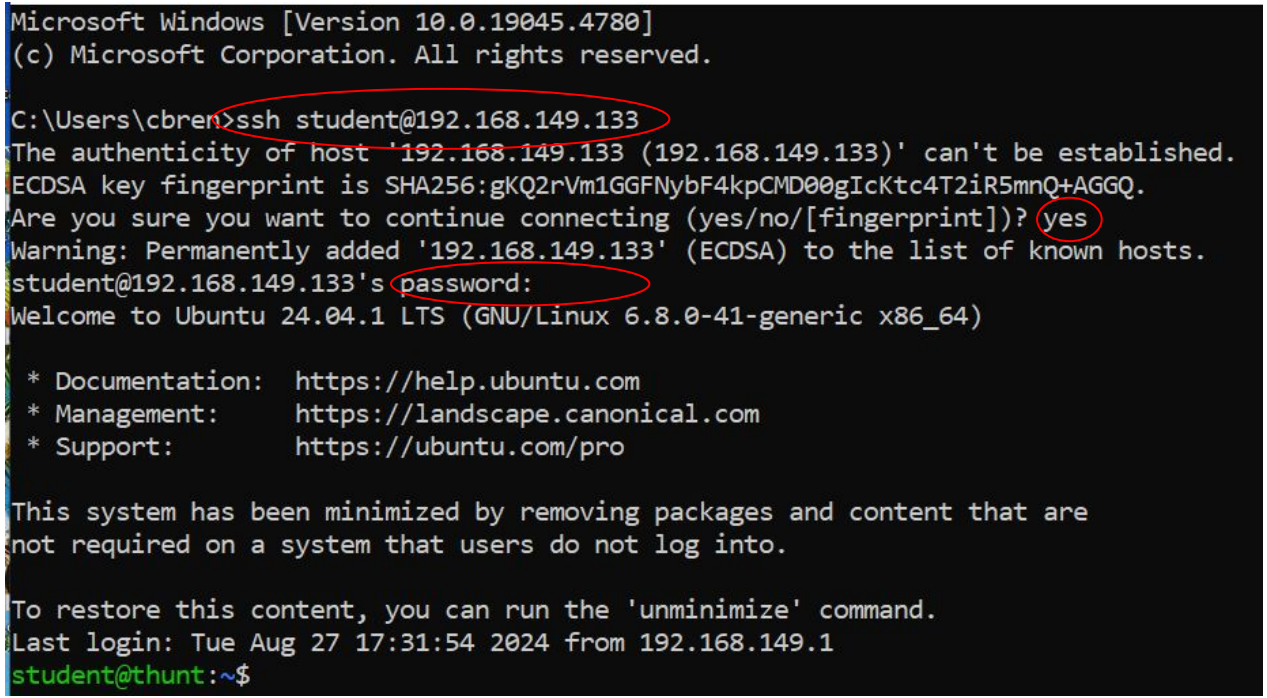
```
ip a | grep -w inet
```



IP address to use
Make a note of it

Network interface

SSH to VM



Password: findc2

Next hands-on walkthrough

- Explore what has been installed
  - Lab files we will be working with
  - Files used by RITA
  - How this "zeek" install is different
- How to process pcaps with Zeek
- How to import Zeek logs into RITA

What RITA installs

```
student@thunt:~$ docker images
REPOSITORY                      TAG       IMAGE ID       CREATED        SIZE
lscr.io/linuxserver/syslog-ng   latest    edeb541b1087   6 days ago     79.4MB
ghcr.io/activecm/rita           v5.0.8    51cbf2933b5b   9 days ago     33.1MB
activecm/zeek                   6.2.1     85d7cfd91bc7   10 days ago    326MB
activecm/zeek                   latest    85d7cfd91bc7   10 days ago    326MB
clickhouse/clickhouse-server    24.1.6    ab7fe0353a83   5 months ago   969MB
student@thunt:~$ docker ps
CONTAINER ID   IMAGE                                  COMMAND            CREATED       STATUS                   PORTS                                NAMES
aff21915525c   lscr.io/linuxserver/syslog-ng:latest   "/init"            45 hours ago  Up 21 hours              6514/tcp, 5514/udp, 6601/tcp         rita-syslog-ng
67a48b8730d8   clickhouse/clickhouse-server:24.1.6    "/entrypoint.sh"   45 hours ago  Up 21 hours (healthy)    8123/tcp, 9000/tcp, 9009/tcp         rita-clickhouse
student@thunt:~$
```

RITA installs:

- A number of Docker containers
- "zeek" script on host which interacts with Zeek Docker container
- "rita" script on host which interacts with remaining containers

Not your usual Zeek executable

```
student@thunt:~$
student@thunt:~$ which zeek
/usr/local/bin/zeek
student@thunt:~$ head -20 /usr/local/bin/zeek
#!/bin/bash
#Sample start/stop script for Zeek running inside docker
#based on service_script_template v0.2
#Many thanks to Logan for his Active-Flow init script, from which some of the following was copied.
#Many thanks to Ethan for his help with the design and implementation, and for the help in troubleshooting readpcap
#V0.5.2

#The --ulimit settings in this file address an issue in an upstream library
#used by zeek where the library allocates two arrays of ints, one entry for
#every possible file descriptor (which is massive in RHEL9 and derivatives
#and allocates 4gb physical, 16gb virtual.  See
# https://github.com/zeek/zeek/issues/2951
#for more details.

#==== USER CUSTOMIZATION ====
#The default Zeek top level directory (/opt/zeek) can be overridden with
#the "zeek_top_dir" environment variable.  Edit /etc/profile.d/zeek and
#add the line (without leading "#"):
#export zeek_top_dir='/my/data/zeek/'
#
student@thunt:~$
```

When you install RITA/Zeek, what you execute is scripts that interact with Docker containers.

Zeek script options

```
student@thunt:~/lab1$ zeek
This script expects a command line option (start, stop, readpcap, restart, status, reload, enable or disable).
In the case of readpcap, please supply the pcap filename as the second command line parameter.
readpcap also accepts an (optional) directory in which to save the logs as the third command line parameter.
Please run again.  Exiting
student@thunt:~/lab1$ zeek status
Zeek docker container status
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
Zeek processes status
Error response from daemon: No such container: zeek
student@thunt:~/lab1$
```

"Zeek" script we created to interact with container.

Zeek script command line switches are not the same as the Zeek binary.

RITA help options

```
student@thunt:~/lab1$ rita -h
[+] Running 3/3
 ✔ Container rita-clickhouse  Healthy
 ✔ Container rita-syslog-ng   Running
 ✔ Container rita-rita-1      Started
[+] Creating 2/0
 ✔ Container rita-clickhouse  Running
 ✔ Container rita-syslog-ng   Running
NAME:
   RITA - Look for evil needles in big haystacks

USAGE:
   rita [-d] command [command options]

VERSION:
   v5.0.8

COMMANDS:
   import    import zeek logs into a target database
   view      view <dataset name>
   delete    delete a dataset
   list      list available datasets
   validate  validate a configuration file
   help, h   Shows a list of commands or help for one command

GLOBAL OPTIONS:
   --debug, -d    Run in debug mode (default: false)
   --help, -h     show help
   --version, -v  print the version
[+] Stopping 1/0
 ✔ Container rita-rita-1  Stopped
student@thunt:~/lab1$
```

RITA's config file - config.hjson

```
student@thunt:~$ head -25 /etc/rita/config.hjson
{
    update_check_enabled: true,
    threat_intel: {
        // Configuration for custom threat intel feeds
        // Allowed format for the contents of both online feeds and custom file feeds is one IP or domain per line
        // Online feeds must be valid URLs
        online_feeds: ["https://feodotracker.abuse.ch/downloads/ipblocklist.txt"],
        // MODIFY THE MOUNT DIRECTORY IN DOCKER COMPOSE, this should rarely need to be changed
        custom_feeds_directory: "/etc/rita/threat_intel_feeds"
    },
    filtering: {
        # These are filters that affect the import of connection logs. They
        # currently do not apply to dns logs.
        # A good reference for networks you may wish to consider is RFC 5735.
        # https://tools.ietf.org/html/rfc5735#section-4

        // internal_subnets identifies the internal network, which will result
        // in any internal to internal and external to external connections being
        // filtered out at import time. Reasonable defaults are provided below,
        // but need to be manually verified before enabling.
        internal_subnets: ["10.0.0.0/8", "172.16.0.0/12", "192.168.0.0/16", "fd00::/8"], # Private-Use Networks  RFC 1918 and ULA prefix

        // always_included_subnets overrides the never_included_* and internal_subnets section,
        // making sure that any connection records containing addresses from these arrays are kept and not filtered
        // Note: the IP address of a proxy must be included here if the proxy is internal
student@thunt:~$
```

Used to tune RITA's detection engine

Contents of home directory

```
student@thunt:~$ student@thunt:~$ ll
total 76
drwxr-x--- 12 student student 4096 Aug 28 17:54 ./
drwxr-xr-x  3 root    root    4096 Aug 27 17:16 ../
-rw-------  1 student student  102 Aug 28 17:54 .Xauthority
drwxrwxr-x  5 student student 4096 Aug 27 17:24 .ansible/
-rw-------  1 student student 3601 Aug 28 17:54 .bash_history
-rw-r--r--  1 student student  220 Mar 31 08:41 .bash_logout
-rw-r--r--  1 student student 3771 Mar 31 08:41 .bashrc
drwx------  2 student student 4096 Aug 27 17:16 .cache/
-rw-------  1 student student   43 Aug 28 17:53 .lesshst
drwxrwxr-x  3 student student 4096 Aug 28 14:40 .local/
-rw-r--r--  1 student student  807 Mar 31 08:41 .profile
drwx------  2 student student 4096 Aug 27 17:16 .ssh/
-rw-r--r--  1 student student    0 Aug 27 17:16 .sudo_as_admin_successful
-rw-rw-r--  1 student student  165 Aug 27 17:23 .wget-hsts
drwxrwxr-x  2 student student 4096 Aug 27 17:20 bin/
drwxrwxr-x  2 student student 4096 Aug 27 17:23 download/
drwxrwxr-x  2 student student 4096 Aug 28 16:27 lab1/
drwxrwxr-x  2 student student 4096 Aug 28 16:20 lab2/
drwxrwxr-x  2 student student 4096 Aug 28 16:20 lab3/
drwxrwxr-x  2 student student 4096 Aug 28 16:21 lab4/
student@thunt:~$
```

Lab directories

What's in the lab1 directory?

```
student@thunt:~$ cd lab1
student@thunt:~/lab1$ ls
lab1.pcap
student@thunt:~/lab1$ capinfos -uae lab1.pcap
File name:           lab1.pcap
Capture duration:    86388.353864 seconds
First packet time:   2020-02-05 19:46:19.233803
Last packet time:    2020-02-06 19:46:07.587667
student@thunt:~/lab1$
```

86,400 seconds is 24 hours

Where do we start?

- Find outbound connection persistency
- See if there is a legitimate business need
- Steps to get there:
  - Process pcap into Zeek logs
  - Import Zeek logs into RITA
  - Review results in RITA

Reading pcaps with Zeek script

```
student@thunt:~$ zeek readpcap ~/lab1/lab1.pcap ~/lab1/
Starting the Zeek docker container
Zeek logs will be saved to /home/student/lab1
student@thunt:~$
```

pcap file to process

Where to store the Zeek logs

What you should get

```
student@thunt:~/lab1$ ll
total 14080
drwxrwxr-x  2 student student     4096 Aug 29 15:39 ./
drwxr-x--- 12 student student     4096 Aug 29 15:43 ../
-rw-r--r--  1 root    root        4423 Aug 29 15:39 capture_loss.log
-rw-r--r--  1 root    root      464294 Aug 29 15:39 conn.log
-rw-r--r--  1 root    root         766 Aug 29 15:39 dhcp.log
-rw-r--r--  1 root    root       88042 Aug 29 15:39 dns.log
-rw-r--r--  1 root    root      469641 Aug 29 15:39 files.log
-rw-r--r--  1 root    root      826789 Aug 29 15:39 http.log
-rw-r--r--  1 root    root         201 Aug 29 15:39 known_hosts.log
-rw-r--r--  1 root    root         265 Aug 29 15:39 known_services.log
-rw-r--r--  1 student student 12333439 Aug 28 14:28 lab1.pcap
-rw-r--r--  1 root    root       35766 Aug 29 15:39 loaded_scripts.log
-rw-r--r--  1 root    root       11814 Aug 29 15:39 notice.log
-rw-r--r--  1 root    root       13398 Aug 29 15:39 ocsp.log
-rw-r--r--  1 root    root         278 Aug 29 15:39 packet_filter.log
-rw-r--r--  1 root    root         379 Aug 29 15:39 reporter.log
-rw-r--r--  1 root    root         651 Aug 29 15:39 software.log
-rw-r--r--  1 root    root       76349 Aug 29 15:39 ssl.log
-rw-r--r--  1 root    root       26564 Aug 29 15:39 stats.log
-rw-r--r--  1 root    root        9969 Aug 29 15:39 x509.log
student@thunt:~/lab1$
```

Import logs into RITA

```
rita import -l ~/lab1/ -d lab1
```

```
student@thunt:~/lab1$ rita import -l ~/lab1/ -d lab1
[+] Running 3/3
 ⠿ Container rita-syslog-ng    Running
 ⠿ Container rita-clickhouse   Healthy
 ⠿ Container rita-rita-1       Started
[+] Creating 2/0
 ⠿ Container rita-syslog-ng    Running
 ⠿ Container rita-clickhouse   Running
```

Where are the Zeek logs?

What to name the database?

```
2024-08-29T15:47:19Z INF Finished Analysis! ⠿  analysis_began=1724946439 analysis_finished=1724946439 elapsed_time=405.424819ms
2024-08-29T15:47:19Z INF Finished Modification! ⠿  elapsed_time=14.733632ms modification_began=1724946439 modification_finished=1724946439
2024-08-29T15:47:19Z INF Finished Importing Hour Chunk day=0 elapsed_time=696.045073ms hour=0
2024-08-29T15:47:19Z INF ⠿ ⠿ Finished Import! ⠿⠿  elapsed_time=1.1s
[+] Stopping 1/0
 ⠿ Container rita-rita-1  Stopped
student@thunt:~/lab1$
```

Success!

```
student@thunt:~/lab1$ rita list
[+] Running 3/3
 ⠿ Container rita-clickhouse  Healthy
 ⠿ Container rita-syslog-ng   Running
 ⠿ Container rita-rita-1      Started
[+] Creating 2/0
 ⠿ Container rita-syslog-ng   Running
 ⠿ Container rita-clickhouse  Running
```

Name	Rolling	Time Range (UTC)
lab1	false	2020-02-05 19:00 - 2020-02-06 19:45

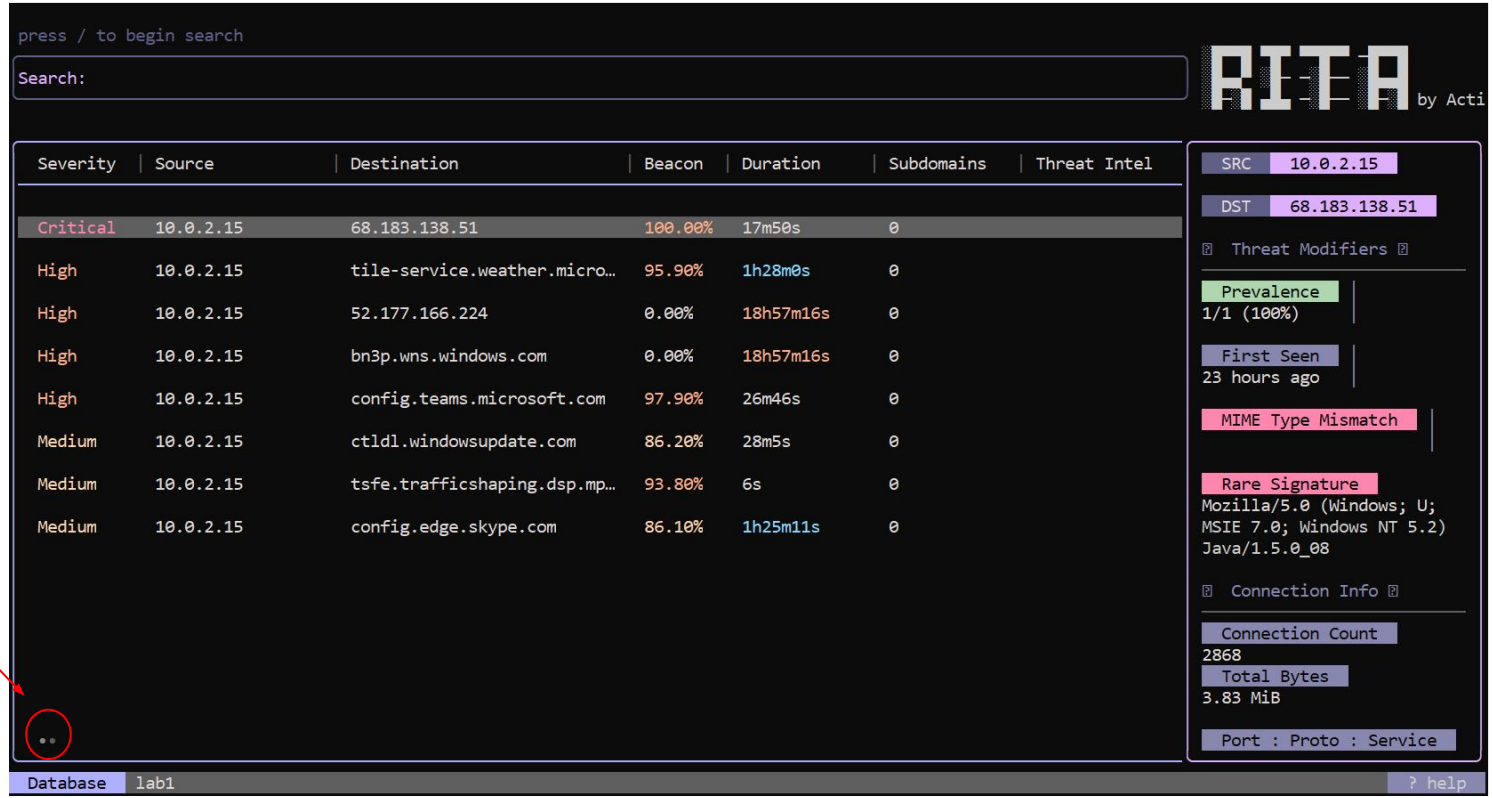
```
[+] Stopping 1/0
 ⠿ Container rita-rita-1  Stopped
student@thunt:~/lab1$
```

Hands-on walkthrough

- First interaction with RITA
- Together we will hunt the first conn pair
- Help you get started using the tool
- Command to get started:

```
rita view lab1
```

First view of RITA



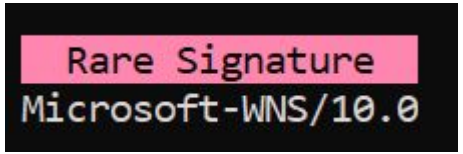
Pages of results

Let's start with the bugs

High	10.0.2.15	52.177.166.224	0.00%	18h57m16s	0
High	10.0.2.15	bn3p.wns.windows.com	0.00%	18h57m16s	0

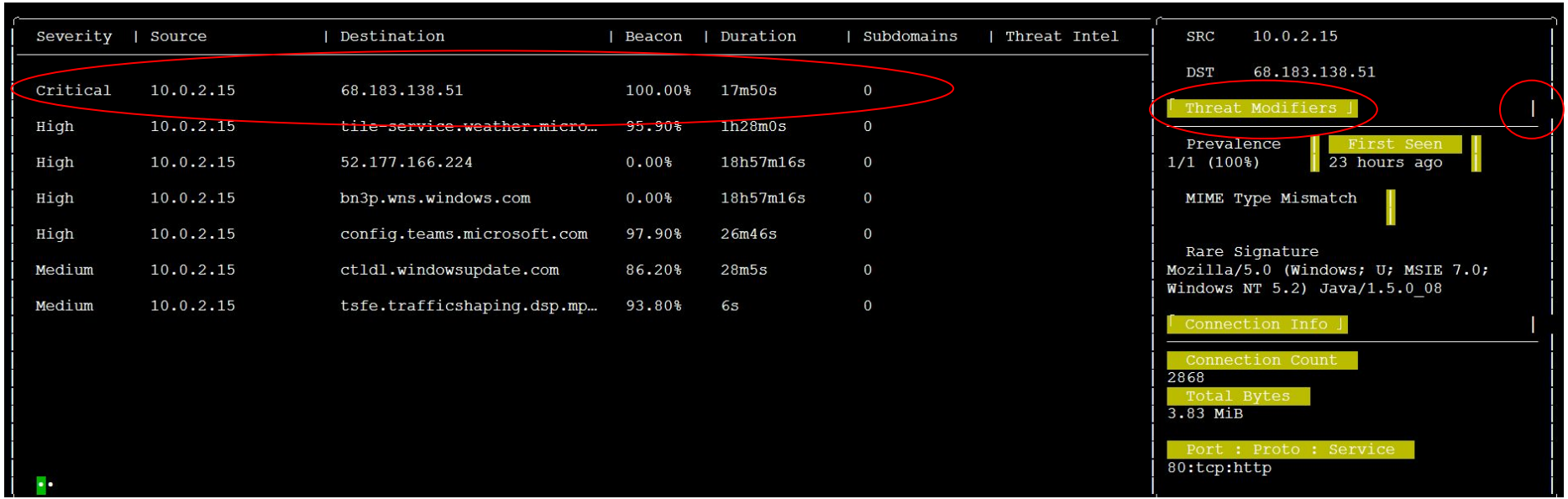
- These are actually the same entry
- One based on IP, the other FQDN
- Exact same duration time is a giveaway
- This is being addressed

Bug #2



- Uniqueness is being calculated based on number of connections to FQDN.
- Based on target IP will be more accurate
- This is being addressed
- For now, manually verify

Hey my screen is messed up!

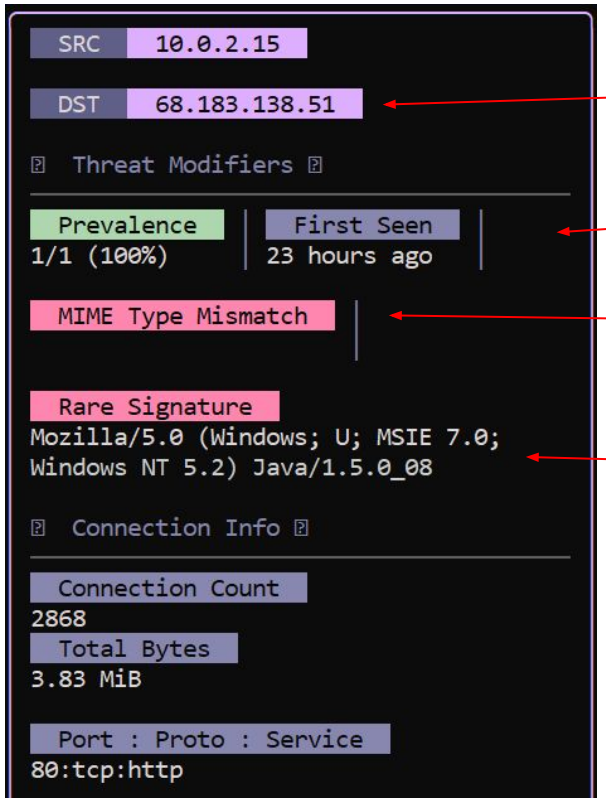


If you get this, hit "q" to quit and run:

```
export TERM=xterm-256color
```

Relaunch RITA. If that does not fix the problem, your terminal app does not support 256 colors (SmarTTY is a known issue).

Critical connection pair



HTTP, so destination should be a FQDN, not an IP address

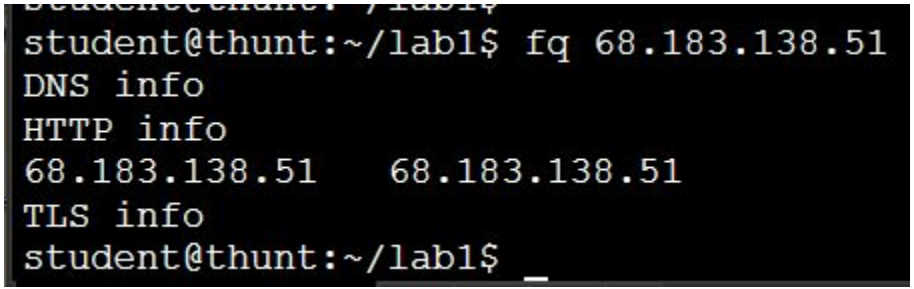
Not really useful for pcaps

File type does not match server MIME

String is unique for this system

Lots of connections but not much data has been moved

Suspicious but not "evil"



We usually connect to Web servers via FQDN. No DNS queries were performed that returned this IP as an answer. So source system did a direct IP address connection without a prior DNS lookup.

Reading the raw Zeek logs

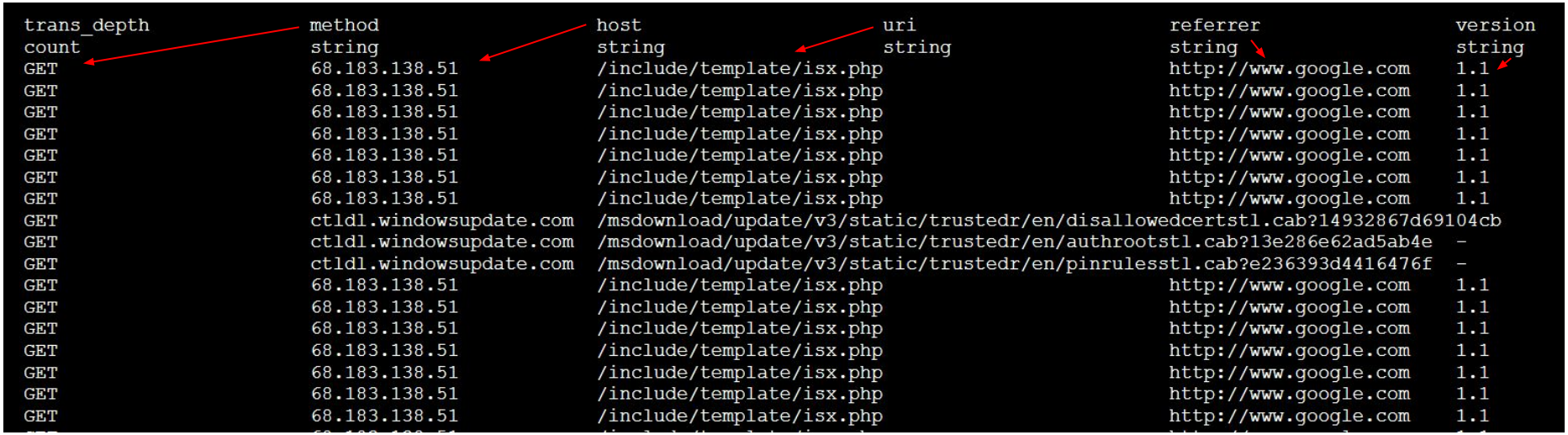
```
less -S -x25 conn.log
```

```
#separator \x09
#set_separator      ,
#empty_field        (empty)
#unset_field        -
#path               conn
#open               2024-08-31-00-13-55
#fields             ts                  uid             id.orig_h       id.orig_p       id.resp_h       id.resp_p
#types              time                string          addr            port            addr            port
1580931979.233803   CzmK432SUc5SmoLpSl  10.0.2.15       49884           68.183.138.51   80              tcp
1580931979.701983   CW1Ug71rqd8G9LN9q8  10.0.2.15       53848           75.75.75.75     53              udp
1580931982.734996   CfIJvW3v2fu3u1RAJh  10.0.2.15       53849           75.75.75.75     53              udp
1580932009.354957   CVl7Dn3NuVhch7xB9   10.0.2.15       49885           68.183.138.51   80              tcp
1580931981.188478   Cdd5sQ1gQPjQCcQnI3  10.0.2.15       138             10.0.2.255      138             udp
1580932039.500298   ChqYq92E1DqOZZkZR8  10.0.2.15       49886           68.183.138.51   80              tcp
1580932053.125526   Cew21014VcSiOkYHX6  10.0.2.15       65426           75.75.75.75     53              udp
1580932069.618701   CQZs6Y1LJU9VmE0eNh  10.0.2.15       49888           68.183.138.51   80              tcp
1580932099.750049   Cu3hwk2L8GAY8OXXWc  10.0.2.15       49889           68.183.138.51   80              tcp
1580932053.142626   CVRjPP1rToZSRhi6ia  10.0.2.15       49887           13.107.3.128    443             tcp
1580932122.685367   CVNCsf28KJ9K4zRynh  10.0.2.15       55180           75.75.75.75     53              udp
1580932129.888218   C0kBIR16ZbikBbouoe  10.0.2.15       49891           68.183.138.51   80              tcp
1580932147.569367   CYyced2cm8gEusIyU2  10.0.2.15       49892           13.107.3.128    443             tcp
1580932160.014154   CUD3Wo4sphYV83fJV9  10.0.2.15       49893           68.183.138.51   80              tcp
1580932182.819756   CZzKz41AI86Zcovew8  10.0.2.15       62299           75.75.75.75     53              udp
```

Search data using "/"

http.log file

```
less -S -x25 http.log
```



trans_depth	method	host	uri	referrer	version
count	string	string	string	string	string
GET	68.183.138.51	/include/template/isx.php		http://www.google.com	1.1
GET	68.183.138.51	/include/template/isx.php		http://www.google.com	1.1
GET	68.183.138.51	/include/template/isx.php		http://www.google.com	1.1
GET	68.183.138.51	/include/template/isx.php		http://www.google.com	1.1
GET	68.183.138.51	/include/template/isx.php		http://www.google.com	1.1
GET	68.183.138.51	/include/template/isx.php		http://www.google.com	1.1
GET	68.183.138.51	/include/template/isx.php		http://www.google.com	1.1
GET	ctldl.windowsupdate.com	/msdownload/update/v3/static/trustedr/en/disallowedcertstl.cab?14932867d69104cb			
GET	ctldl.windowsupdate.com	/msdownload/update/v3/static/trustedr/en/authrootstl.cab?13e286e62ad5ab4e			-
GET	ctldl.windowsupdate.com	/msdownload/update/v3/static/trustedr/en/pinrulesstl.cab?e236393d4416476f			-
GET	68.183.138.51	/include/template/isx.php		http://www.google.com	1.1
GET	68.183.138.51	/include/template/isx.php		http://www.google.com	1.1
GET	68.183.138.51	/include/template/isx.php		http://www.google.com	1.1
GET	68.183.138.51	/include/template/isx.php		http://www.google.com	1.1
GET	68.183.138.51	/include/template/isx.php		http://www.google.com	1.1
GET	68.183.138.51	/include/template/isx.php		http://www.google.com	1.1
GET	68.183.138.51	/include/template/isx.php		http://www.google.com	1.1

Column titles may be offset

MIME type mismatch

```
cat http.log | zcutter id.resp_h uri mime_types | grep 68.183.138.51 | sort | uniq -c
```

```
student@thunt:~/lab1$ cat http.log | zcutter id.resp_h uri resp_mime_types | grep 68.183.138.51 | sort | uniq -c
   2868 68.183.138.51   /include/template/isx.php       text/html
student@thunt:~/lab1$ _
```

MIME type for "isx.php" expected to be "application/x-httpd-php" not "text/html"

List stored at: /etc/rita/http_extensions_list.csv

Also, 2,868 requests for the same PHP file is suspicious

Why zcutter instead of zeek-cut?

- zeek-cut on steroids
- Many functions not supported in zeek–cut
- Supports JSON format, not just CSV
- Can convert between JSON and CSV
- Can process compressed Zeek logs
- Will accept multiple files as input

https://github.com/activecm/zcutter

Rare signature

```
cat http.log | zcutter id.orig_h id.resp_h user_agent | grep 10.0.2.15 | sort | uniq -c
```

```
student@thunt:~/lab1$ cat http.log | zcutter id.orig_h id.resp_h user_agent | grep 10.0.2.15 | sort |uniq -c
      1 10.0.2.15       104.104.10.72   Microsoft-WNS/10.0
      6 10.0.2.15       104.107.60.98   Microsoft-CryptoAPI/10.0
      3 10.0.2.15       104.107.61.83   Microsoft-CryptoAPI/10.0
      3 10.0.2.15       104.112.229.83  Microsoft-CryptoAPI/10.0
      1 10.0.2.15       104.112.229.88  Microsoft-CryptoAPI/10.0
      3 10.0.2.15       104.121.93.214  Microsoft-WNS/10.0
      4 10.0.2.15       104.71.255.238  Microsoft-WNS/10.0
      1 10.0.2.15       104.80.34.253   Microsoft-WNS/10.0
      2 10.0.2.15       104.86.71.221   Microsoft-WNS/10.0
      3 10.0.2.15       13.107.4.50     Microsoft-CryptoAPI/10.0
      6 10.0.2.15       172.232.17.170  Microsoft-CryptoAPI/10.0
      4 10.0.2.15       184.87.56.181   Microsoft-WNS/10.0
      3 10.0.2.15       2.19.89.91      Microsoft-WNS/10.0
     10 10.0.2.15       205.185.216.42  Microsoft-CryptoAPI/10.0
      2 10.0.2.15       23.198.77.93    Microsoft-WNS/10.0
      3 10.0.2.15       23.200.236.232  Microsoft-CryptoAPI/10.0
      3 10.0.2.15       23.37.83.178    Microsoft-WNS/10.0
     20 10.0.2.15       23.67.114.110   Microsoft-WNS/10.0
      2 10.0.2.15       23.74.2.66      Microsoft-CryptoAPI/10.0
      4 10.0.2.15       23.78.105.148   Microsoft-WNS/10.0
   2868 10.0.2.15       68.183.138.51   Mozilla/5.0 (Windows; U; MSIE 7.0; Windows NT 5.2) Java/1.5.0_08
     19 10.0.2.15       72.21.81.240    Microsoft-CryptoAPI/10.0
      3 10.0.2.15       72.246.64.162   Microsoft-CryptoAPI/10.0
      3 10.0.2.15       72.246.64.168   Microsoft-CryptoAPI/10.0
      3 10.0.2.15       8.240.2.254     Microsoft-CryptoAPI/10.0
      3 10.0.2.15       8.240.64.254    Microsoft-CryptoAPI/10.0
      1 10.0.2.15       8.252.166.126   Microsoft-CryptoAPI/10.0
      1 10.0.2.15       95.100.138.18   Microsoft-WNS/10.0
student@thunt:~/lab1$
```

The only time this system uses this user agent string is when it talks to this one target.

Why the high beacon score?

```
Beacon-conn 10.0.2.15 68.183.138.51
```

```
student@thunt:~/lab1$ beacon-conn 10.0.2.15          68.183.138.51
19 28
20 119
21 44
20 1
21 76
22 119
23 120
00 119
01 120
02 119
03 120
04 119
05 120
06 119
07 120
08 119
09 120
10 119
11 120
12 119
13 120
14 119
15 120
16 119
17 120
18 119
19 92
student@thunt:~/lab1$ _
```

Connecting about every 30 seconds in most hours.

ASN associated with target IP

```
whois -h whois.cymru.com " -v 68.183.138.51"
```

```
student@thunt:~/lab1$ whois -h whois.cymru.com " -v 68.183.138.51"
AS      | IP               | BGP Prefix          | CC | Registry | Allocated  | AS Name
14061   | 68.183.138.51    | 68.183.128.0/20     | US | arin     | 2018-09-18 | DIGITALOCEAN-ASN, US
```

Controlled by DigitalOcean
Have they delegated this IP address space?

Who controls 68.183.138.51?

```
dig -x 68.183.138.51 | grep -w arpa
```

```
student@thunt:~/lab1$ dig -x 68.183.138.51 | grep -w arpa
;51.138.183.68.in-addr.arpa.    IN      PTR
138.183.68.in-addr.arpa. 5      IN      SOA     ns1.digitalocean.com. hostmaster.138.183.68.in-addr.arpa. 1725356151 10800 3600 604800 1800
student@thunt:~/lab1$ _
```

No PTR record
Located in a public cloud

Disposition of 68.183.138.51

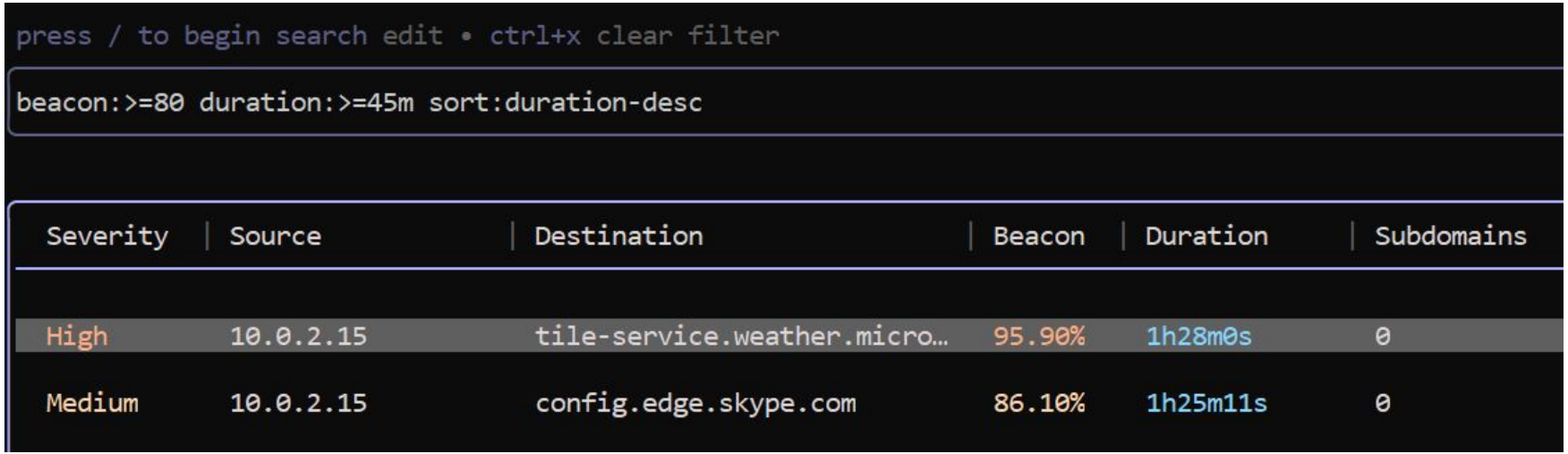
- Absolutely requires deeper investigation
  - Google unique data collected
    - URI
    - user agent string
  - Grab pcaps if we did not already have them
  - What app is creating these connections?
- Would recommend incident response
- URI reveals this is most likely Fiesta C2 delivered by Cloxer.AA

One more hands-on walkthrough

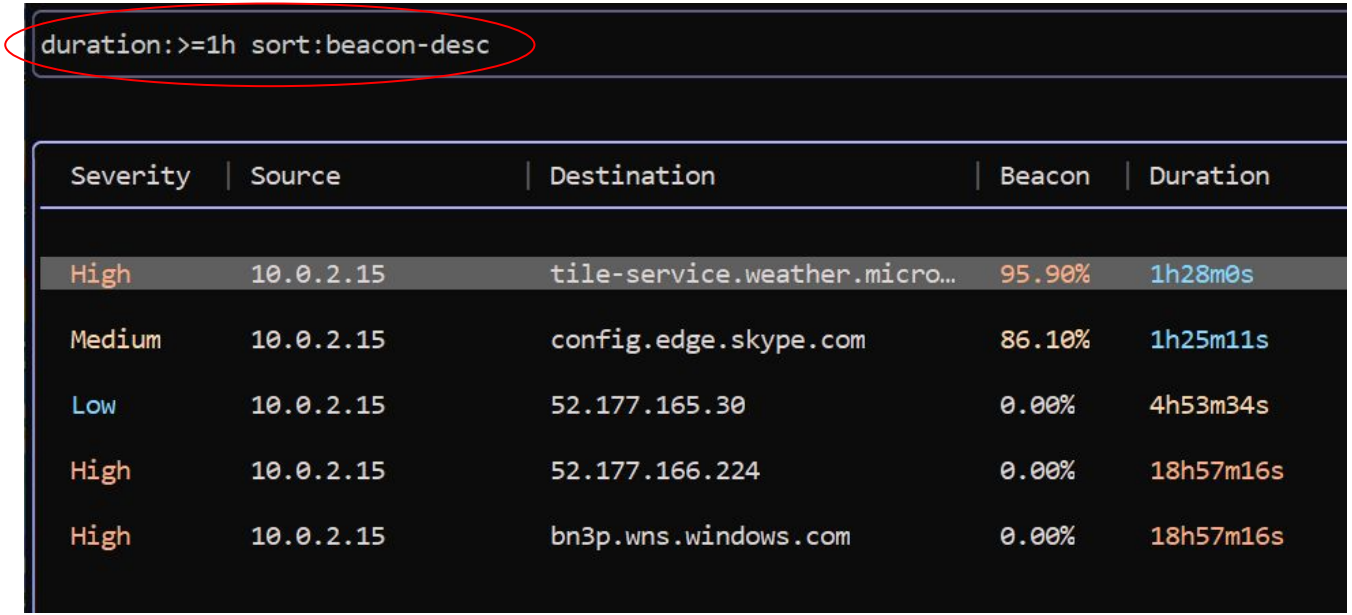
- By default, results sorted by "Severity"
- We can sort data by any column
  - Sort and filters accessed by pressing "/"
  - Sort needs to define ascending or descending
  - Example= sort:duration-desc sort:beacon-asc
- We can also filter by column
  - Will accept "<" and ">"
  - Example: beacon:>=80 duration:>=1h

Filter/sort example

```
beacon:>=80 duration:>=45m sort:duration-desc
```



Another example



Press "?" for help

Press "<ctrl>-x to clear sort/filter settings

Lab time!

- There are three pairs with a severity of "high"
  - Remember one set is a duplicate
- Investigate each of these
- Try to decide if each is:
  - Normal business traffic
  - Possibly a compromised system
- Please use spoilers if posting answers in Discord!
  - Two "|" before and after your text
  - Feel free to test it out now

```
||This lab was easy. The answer is blue.||
```

Hints

- ID if there is a business need for the connection
- Investigate if the endpoint looks legit
- If not, check the protocol for strange behaviour
- Common to have no DNS or app data for long connections that start before the pcap
- "Microsoft-WNS/10.0" is flagged as rare. It actually is not. This is a bug in the code that's being addressed.

Answers

```
student@thunt:~/lab1$ fq tile-service.weather.microsoft.com
DNS info
wildcard.weather.microsoft.com.edgekey.net,e15275.g.akamaiedge.net,104.104.10.72        tile-service.weather.microsoft.com
wildcard.weather.microsoft.com.edgekey.net,e15275.g.akamaiedge.net,104.121.93.214       tile-service.weather.microsoft.com
wildcard.weather.microsoft.com.edgekey.net,e15275.g.akamaiedge.net,104.71.255.238       tile-service.weather.microsoft.com
wildcard.weather.microsoft.com.edgekey.net,e15275.g.akamaiedge.net,104.80.34.253        tile-service.weather.microsoft.com
wildcard.weather.microsoft.com.edgekey.net,e15275.g.akamaiedge.net,104.86.71.221        tile-service.weather.microsoft.com
wildcard.weather.microsoft.com.edgekey.net,e15275.g.akamaiedge.net,184.87.56.181        tile-service.weather.microsoft.com
wildcard.weather.microsoft.com.edgekey.net,e15275.g.akamaiedge.net,2.19.89.91   tile-service.weather.microsoft.com
wildcard.weather.microsoft.com.edgekey.net,e15275.g.akamaiedge.net,23.198.77.93 tile-service.weather.microsoft.com
wildcard.weather.microsoft.com.edgekey.net,e15275.g.akamaiedge.net,23.37.83.178 tile-service.weather.microsoft.com
wildcard.weather.microsoft.com.edgekey.net,e15275.g.akamaiedge.net,23.67.114.110        tile-service.weather.microsoft.com
wildcard.weather.microsoft.com.edgekey.net,e15275.g.akamaiedge.net,23.78.105.148        tile-service.weather.microsoft.com
wildcard.weather.microsoft.com.edgekey.net,e15275.g.akamaiedge.net,95.100.138.18        tile-service.weather.microsoft.com
HTTP info
104.104.10.72   tile-service.weather.microsoft.com
104.121.93.214  tile-service.weather.microsoft.com
104.71.255.238  tile-service.weather.microsoft.com
104.80.34.253   tile-service.weather.microsoft.com
104.86.71.221   tile-service.weather.microsoft.com
184.87.56.181   tile-service.weather.microsoft.com
2.19.89.91      tile-service.weather.microsoft.com
23.198.77.93    tile-service.weather.microsoft.com
23.37.83.178    tile-service.weather.microsoft.com
23.67.114.110   tile-service.weather.microsoft.com
23.78.105.148   tile-service.weather.microsoft.com
95.100.138.18   tile-service.weather.microsoft.com
TLS info
student@thunt:~/lab1$ _
```

tile-service.weather.microsoft.com resolves to multiple Akamai CDNs. Clearly well funded, which seem more legit than evil.

Answers - Dup entry

```
student@thunt:~/lab1$ fq 52.177.166.224
DNS info
bn3p.wns.notify.windows.com.akadns.net,52.177.166.224   bn3p.wns.windows.com
HTTP info
TLS info
52.177.166.224  bn3p.wns.windows.com    ok
student@thunt:~/lab1$
```

- DNS info matches SNI info
- Digital certificate is valid
- Known server used for WNS

Answers - config.teams...

```
student@thunt:~/lab1$ fq config.teams.microsoft.com
DNS info
config.teams.trafficmanager.net,s-0005.s-msedge.net,52.113.194.132      config.teams.microsoft.com
HTTP info
TLS info
52.113.194.132  config.teams.microsoft.com      ok
student@thunt:~/lab1$ _
```

- Similar to last one
- DNS matches SNI, digital cert valid
- Known server used by Teams

Safelisting - Hands-on walkthrough

- When a remote system is safe, you want to safelist the entry
- This will keep it out of future hunts
- Data is still collected, just not scored
  - Can revert later if needed
- Can safelist by IP or FQDN
- Let's create some entries together!

What to safelist

- We had 3 entries with a high score that we deemed safe
- Let's remove them from future hunts
- Entries to safelist:
  - 52.113.194.132
  - 52.177.166.224
  - tile-service.weather.microsoft.com

RITA's config file

- RITA can be tweaked via changes to it's configuration file
  - /etc/rita/config.hjson
- Things you can change
  - Internal network definition
  - Threat intel feeds
  - Score weighting
  - Systems to always include in processing
  - Systems to safelist (exclude)

Config file example

```
student@thunt:~/lab1$ head -20 /etc/rita/config.hjson
{
    update_check_enabled: true,
    threat_intel: {
        // Configuration for custom threat intel feeds
        // Allowed format for the contents of both online feeds and custom file feeds is one IP or domain per line
        // Online feeds must be valid URLs
        online_feeds: ["https://feodotracker.abuse.ch/downloads/ipblocklist.txt"],
        // MODIFY THE MOUNT DIRECTORY IN DOCKER COMPOSE, this should rarely need to be changed
        custom_feeds_directory: "/etc/rita/threat_intel_feeds"
    },
    filtering: {
        # These are filters that affect the import of connection logs. They
        # currently do not apply to dns logs.
        # A good reference for networks you may wish to consider is RFC 5735.
        # https://tools.ietf.org/html/rfc5735#section-4

        // internal_subnets identifies the internal network, which will result
        // in any internal to internal and external to external connections being
        // filtered out at import time. Reasonable defaults are provided below,
        // but need to be manually verified before enabling.
student@thunt:~/lab1$
```

How to safelist

- Need to edit file as root
- Add safelist entries to "never_include"
  - CIDR or FQDN format
  - Double quotes around each entry
- All future hunts will exclude these entries
  - Data still collected
  - Entries will not be scored
  - Remove entries to have them return

What to change

Change:

```
// connections involving ranges entered into never_included_subnets are filtered out at import time
never_included_subnets: [], // array of CIDRs
never_included_domains: [], // array of FQDNs
```

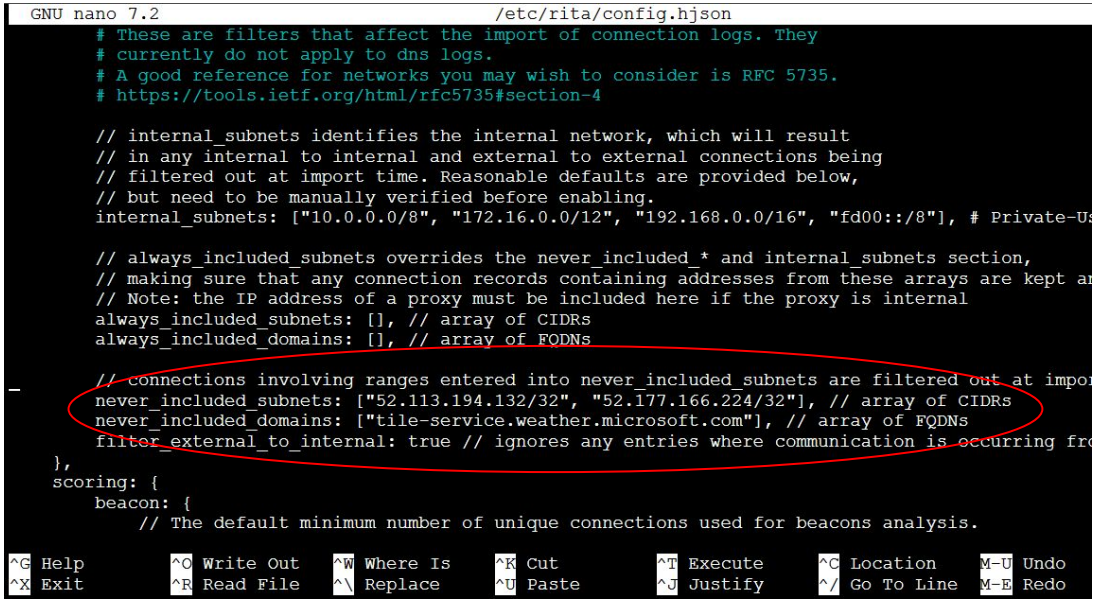
To be:

```
// connections involving ranges entered into never_included_subnets are filtered out at import time
never_included_subnets: ["52.113.194.132/32", "52.177.166.224/32"], // array of CIDRs
never_included_domains: ["tile-service.weather.microsoft.com"], // array of FQDNs
```

Note CIDR format and double quotes around each entry

How to make the change

```
sudo nano /etc/rita/config.hjson
```



CTRL-o to save, CTRL-x to quit

Create a new database

```
student@thunt:~/lab1$ rita import -l ~/lab1/ -d lab1b
[+] Running 3/3
 ✔ Container rita-clickhouse  Healthy
 ✔ Container rita-syslog-ng   Running
 ✔ Container rita-rita-1      Started
[+] Creating 2/0
 ✔ Container rita-syslog-ng   Running
 ✔ Container rita-clickhouse  Running
```

Then run:

```
rita view lab1b
```

High severity entries removed!

Severity	Source	Destination	Beacon	Duration	Subdomains	Threat Intel
Critical	10.0.2.15	68.183.138.51	100.00%	17m50s	0	
Medium	10.0.2.15	ctldl.windowsupdate.com	86.20%	28m5s	0	
Medium	10.0.2.15	config.edge.skype.com	86.10%	1h25m11s	0	
Medium	10.0.2.15	tsfe.trafficshaping.dsp.mp…	93.80%	6s	0	
Medium	10.0.2.15	23.67.114.110	90.10%	40m48s	0	
Low	10.0.2.15	75.75.75.75	80.80%	7s	0	
Low	10.0.2.15	52.177.165.30	0.00%	4h53m34s	0	

Safelisting in remaining labs

- Safelist entries in the rest of the labs is optional
- How comfortable are you editing Linux text files?
- Can be a little time consuming
- We are on limited time until the end of class
- I'll leave it to your discretion
- Just don't fall behind :-)

Next lab!

- Move to the "lab2" directory
- Run the pcap through Zeek
- Import the Zeek logs into RITA
- Hunt all items with critical or high severity
- Initial commands to run:

```
cd ~/lab2
zeek readpcap ~/lab2/lab2.pcap ~/lab2/
rita import -l ~/lab2/ -d lab2
rita view lab2
```

Lab2 analysis

- There are three connection pairs with a severity of high
- Run each of these down to see if any are potentially malicious
- Sometimes it's easier to start with pairs that may have a legit business need
- Leave the hard ones for last

Hints

- Large number of "Subdomains" data may be an indicator of C2 over DNS
- Could be some interesting info in Zeek's dns.log file
- Search the file for indicated domain
- Think about what is "normal" and pay attention to data that odd or different

Answers - NTP

```
student@thunt:~/lab2$ grep 91.189.89.198 ntp.log | head -5
1623214411.238512       C7aiW8EsYW74vI073        10.20.57.3      43210   91.189.89.198   123     4       3       0       1.000000        1.000000
0.000000        0.000000        \x00\x00\x00\x00        0.000000        0.000000        0.000000        1623214411.055518       0
1623214411.340292       C7aiW8EsYW74vI073        10.20.57.3      43210   91.189.89.198   123     4       4       2       8.000000        0.000000
0.001022        0.022934        17.253.34.123   1623213911.268710       1623214411.055518       1623214411.271790       1623214411.271817       0
1623216459.489070       COOavZ5OyginH5myi        10.20.57.3      47640   91.189.89.198   123     4       3       0       1.000000        1.000000
0.000000        0.000000        \x00\x00\x00\x00        0.000000        0.000000        0.000000        1623216459.113850       0
1623216459.589011       COOavZ5OyginH5myi        10.20.57.3      47640   91.189.89.198   123     4       4       2       8.000000        0.000000
0.001038        0.021729        17.253.34.123   1623216055.417658       1623216459.113850       1623216459.531006       1623216459.531093       0
1623218507.738546       CCTfk31sByM7g0ucu5       10.20.57.3      58182   91.189.89.198   123     4       3       0       1.000000        1.000000
0.000000        0.000000        \x00\x00\x00\x00        0.000000        0.000000        0.000000        1623218507.171939       0
student@thunt:~/lab2$ dig -x 91.189.89.198 | grep arpa
;198.89.189.91.in-addr.arpa.    IN      PTR
89.189.91.in-addr.arpa. 5       IN      SOA     ns1.canonical.com. hostmaster.canonical.com. 2018042656 10800 3600 604800 3600
student@thunt:~/lab2$ _
```

First entry looks like legit NTP traffic
NTP servers are typically accessed via IP address
May want to create a safelist entry for this

Answers - connectivity-check

```
student@thunt:~/lab2$ fq connectivity-check.ubuntu.com
DNS info
-       connectivity-check.ubuntu.com
-       connectivity-check.ubuntu.com.rhodes.edu
34.122.121.32,35.224.170.84,35.232.111.17       connectivity-check.ubuntu.com
34.122.121.32,35.232.111.17,35.224.170.84       connectivity-check.ubuntu.com
35.224.170.84,34.122.121.32,35.232.111.17       connectivity-check.ubuntu.com
35.224.170.84,35.232.111.17,34.122.121.32       connectivity-check.ubuntu.com
35.232.111.17,34.122.121.32,35.224.170.84       connectivity-check.ubuntu.com
35.232.111.17,35.224.170.84,34.122.121.32       connectivity-check.ubuntu.com
HTTP info
34.122.121.32   connectivity-check.ubuntu.com
35.224.170.84   connectivity-check.ubuntu.com
35.232.111.17   connectivity-check.ubuntu.com
TLS info
student@thunt:~/lab2$ grep connectivity-check.ubuntu.com http.log | head -1
1623213040.091771       CAWZrD4Mrzi19Eq8dd      10.20.57.3      59104   35.224.170.84   80      1       GET     connectivity-check.ubuntu.com   /
-       1.1     -       -       0       0       204     No Content      -       -       (empty) -       -       -       -       -       -       -
-       -
student@thunt:~/lab2$
```

Third one is Ubuntu calling home
System implies a benign check
It's actually Canonical tracking installs

Well this is odd…

Severity	Source	Destination	Beacon	Duration	Subdomains
High	10.20.57.3	91.189.89.198	100.00%	1m40s	0
High		cisco-update.com	0.00%	0s	165378

Second entry has a benign domain name
Subdomains listed at 165,378

Does it make sense we would look up this many resource records in 24 hours?

Answers - cisco-update

```
student@thunt:~/lab2$ fq cisco-update.com | head
DNS info
-       0200018ea0233fb9712756a8d59fcf4bdf.cisco-update.com
-       03cc018ea0fa373dfd19faf39296b8e1c3.cisco-update.com
-       056d018ea0b065d773991ea4a1dc0fed4b.cisco-update.com
-       0641018ea09ce100cb2e044a4e8287101b.cisco-update.com
-       0777016cb1bf981e6f0be31ea77085a7f0.cisco-update.com
-       0bbd018ea069a555d6143e181b90bee29e.cisco-update.com
-       0c89016cb1e0e6d92b359aa9b813ed9391.cisco-update.com
-       0cc5018ea00a334dd82c824ceefc9a97b8.cisco-update.com
-       0cfc018ea0904971664e0ed873df6058e0.cisco-update.com
```

Do these look like names humans would use?
Values are 0-9 and a-f. This is Hex!!!
Could be obfuscated data

Can we read the Hex?

- Maybe, need to convert Hex to ASCII
  - There may be other layers of encoding
- Many tools available
  - xxd with "-r" switch
  - CyberChef - Awesome online conversion tool
- This gets a bit beyond an intro class
- I cover these techniques in the advanced threat hunting class

Lab3

- Move to the "lab3" directory
- Run the pcap through Zeek
- Import the Zeek logs into RITA
- Hunt all items with critical or high severity
- Initial commands to run:

```
cd ~/lab3
zeek readpcap ~/lab3/lab3.pcap ~/lab3/
rita import -l ~/lab3/ -d lab3
rita view lab3
```

Lab3 analysis

- First page is all high severity items
  - Unless you previously set extra safelists
- Hunt all 8 items to see if any are of potential concern
- If you can't decide in 3-4 minutes, come back to that connection pair last

Hints

- Same process as before, start with the easy ones and work into more challenging
- Long connections at around 24 hours
  - Not uncommon to have no DNS or header info
  - That data collected at connection start
  - So connection started before pcap was captured
  - In this case, this info being missing not unusual
  - Will not be a problem with live captures

Answers - 162.159.200.1

```
student@thunt:~/lab3$ grep 162.159.200.1 conn.log | head -1
1714219615.329541       CuHlwX3t7No9o4O4G7      192.168.100.139 39260   162.159.200.1   123     udp     ntp     0.004297        48      48      SF
T       F       0       Dd      1       76      1       76      -
student@thunt:~/lab3$ grep 162.159.200.1 ntp.log | head -1
1714219615.329541       CuHlwX3t7No9o4O4G7      192.168.100.139 39260   162.159.200.1   123     4       3       0       1.000000        1.000000
0.000000        0.000000        \x00\x00\x00\x00        0.000000        0.000000        0.000000        1714219615.076723       0
student@thunt:~/lab3$ dig -x 162.159.200.1 | grep arpa
;1.200.159.162.in-addr.arpa.    IN      PTR
1.200.159.162.in-addr.arpa. 5   IN      PTR     time.cloudflare.com.
student@thunt:~/lab3$
```

First entry looks like normal NTP traffic
Normally we would safelist this entry

Answers - 52.226.139.0/24

- Long conn from 4 internal to 2 external
- Conn time close to 24 hours
- Conns started before pcap
- DNS & header info collected at conn start
- No suspicious this is missing
- Not a problem with live captures
  - Only pcaps due to limited time

Windows calling home?

```
student@thunt:~/lab3$ dig -x 52.226.139.185 | grep arpa
;185.139.226.52.in-addr.arpa.   IN      PTR
139.226.52.in-addr.arpa. 5      IN      SOA     ns1-201.azure-dns.com. msnhst.microsoft.com. 1 900 300 604800 60
student@thunt:~/lab3$ dig -x 52.226.139.180 | grep arpa
;180.139.226.52.in-addr.arpa.   IN      PTR
139.226.52.in-addr.arpa. 5      IN      SOA     ns1-201.azure-dns.com. msnhst.microsoft.com. 1 900 300 604800 60
student@thunt:~/lab3$ whois -h whois.cymru.com " -v 52.226.139.180"
AS      | IP               | BGP Prefix          | CC | Registry | Allocated  | AS Name
8075    | 52.226.139.180   | 52.224.0.0/11       | US | arin     | 2015-11-24 | MICROSOFT-CORP-MSN-AS-BLOCK, US
student@thunt:~/lab3$ _
```

Two targets on MS network but no PTR
Cannot blindly trust 8075 anymore!

Can we confirm source is Windows?

```
student@thunt:~/lab3$ cat dns.log | zcutter id.orig_h query | grep 192.168.100.151 | sort | uniq -c | sort -rn | head
    385 192.168.100.151 desktop-bkrdsbg.local
    384 192.168.100.151 desktop-bkrdsbg
    310 192.168.100.151 wpad.localdomain
    127 192.168.100.151 array614.prod.do.dsp.mp.microsoft.com
     72 192.168.100.151 edge.microsoft.com
     69 192.168.100.151 settings-win.data.microsoft.com
     55 192.168.100.151 v10.events.data.microsoft.com
     46 192.168.100.151 array610.prod.do.dsp.mp.microsoft.com
     32 192.168.100.151 ctldl.windowsupdate.com
     32 192.168.100.151 array611.prod.do.dsp.mp.microsoft.com
student@thunt:~/lab3$ _
```

Lots of Windows related queries
Source is most likely Windows
These 4 entries are normal Windows behaviour

Answers - ctldl.windowsupdate.com

```
student@thunt:~/lab3$ fq ctldl.windowsupdate.com | head
DNS info
wu-bg-shim.trafficmanager.net,bg.microsoft.map.fastly.net,199.232.210.172,199.232.214.172       ctldl.windowsupdate.com
wu-bg-shim.trafficmanager.net,download.windowsupdate.com.edgesuite.net,a767.dspw65.akamai.net,184.150.154.120,184.150.154.26,184.150.154.72,184.150
.154.121,184.150.154.25,184.150.154.99  ctldl.windowsupdate.com
wu-bg-shim.trafficmanager.net,download.windowsupdate.com.edgesuite.net,a767.dspw65.akamai.net,184.150.154.123,184.150.154.104,184.150.154.25    ctl
dl.windowsupdate.com
wu-bg-shim.trafficmanager.net,download.windowsupdate.com.edgesuite.net,a767.dspw65.akamai.net,184.150.154.123,184.150.154.67,184.150.154.72,184.150
.154.82,184.150.154.120,184.150.154.99,184.150.154.26,184.150.154.18,184.150.154.74     ctldl.windowsupdate.com
wu-bg-shim.trafficmanager.net,download.windowsupdate.com.edgesuite.net,a767.dspw65.akamai.net,184.150.154.18,184.150.154.17,184.150.154.123,184.150
.154.25,184.150.154.19,184.150.154.121,184.150.154.120,184.150.154.72,184.150.154.26    ctldl.windowsupdate.com
wu-bg-shim.trafficmanager.net,download.windowsupdate.com.edgesuite.net,a767.dspw65.akamai.net,184.150.154.25    ctldl.windowsupdate.com
wu-bg-shim.trafficmanager.net,download.windowsupdate.com.edgesuite.net,a767.dspw65.akamai.net,184.150.154.25,184.150.154.104,184.150.154.17,184.150
.154.19,184.150.154.82,184.150.154.26   ctldl.windowsupdate.com
wu-bg-shim.trafficmanager.net,download.windowsupdate.com.edgesuite.net,a767.dspw65.akamai.net,184.150.154.25,184.150.154.72     ctldl.windowsupdate
.com
wu-bg-shim.trafficmanager.net,download.windowsupdate.com.edgesuite.net,a767.dspw65.akamai.net,184.150.154.25,184.150.154.81     ctldl.windowsupdate
.com
student@thunt:~/lab3$ _
```

Large number of CDNs means this will most likely be legit trafficmanager.net also associated with Microsoft

Answers - Wait? No? What? Wait…

```
student@thunt:~/lab3$ cat http.log | zcutter host uri | grep ctldl.windowsupdate.com | sort | cut -d '?' -f 1 | uniq -c | head
     86 ctldl.windowsupdate.com /msdownload/update/v3/static/trustedr/en/authrootstl.cab
     87 ctldl.windowsupdate.com /msdownload/update/v3/static/trustedr/en/disallowedcertstl.cab
      8 ctldl.windowsupdate.com /msdownload/update/v3/static/trustedr/en/pinrulesstl.cab
student@thunt:~/lab3$
```

Known Windows behaviour with this FQDN
Checking for Digital Cert updates over plaintext
So trusting TLS relies on trusting plaintext
Because connection hijacking is just theoretical

A quick decode

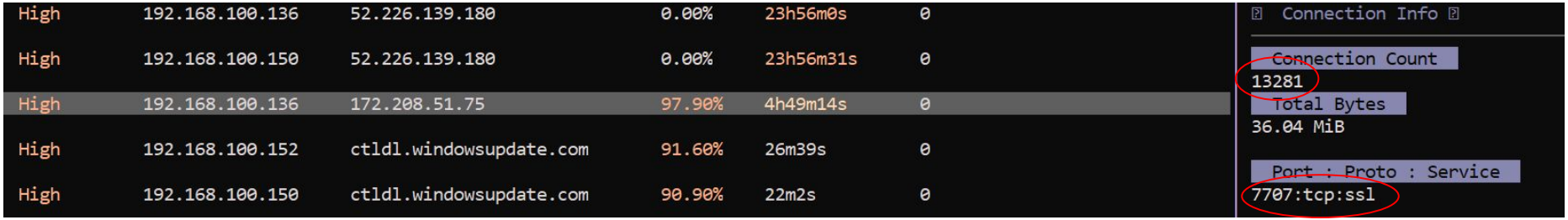
```
14:34:55.412389 IP 192.168.100.136.51708 > 184.150.154.80.80: Flags [P.], seq 287:569, ack 267, win 63974, length 282: HTTP: GET /msdownload/update
/v3/static/trustedr/en/authrootstl.cab?8b439549254404b9 HTTP/1.1
        0x0000:  4500 0142 8bfb 4000 8006 f5a2 c0a8 6488  E..B..@.......d.
        0x0010:  b896 9a50 c9fc 0050 7ce5 ed63 5fc4 3074  ...P...P|..c_.0t
        0x0020:  5018 f9e6 55bc 0000 4745 5420 2f6d 7364  P...U...GET./msd
        0x0030:  6f77 6e6c 6f61 642f 7570 6461 7465 2f76  ownload/update/v
        0x0040:  332f 7374 6174 6963 2f74 7275 7374 6564  3/static/trusted
        0x0050:  722f 656e 2f61 7574 6872 6f6f 7473 746c  r/en/authrootstl
        0x0060:  2e63 6162 3f38 6234 3339 3534 3932 3534  .cab?8b439549254
        0x0070:  3430 3462 3920 4854 5450 2f31 2e31 0d0a  404b9.HTTP/1.1..
        0x0080:  436f 6e6e 6563 7469 6f6e 3a20 4b65 6570  Connection:.Keep
        0x0090:  2d41 6c69 7665 0d0a 4163 6365 7074 3a20  -Alive..Accept:.
        0x00a0:  2a2f 2a0d 0a49 662d 4d6f 6469 6669 6564  */*..If-Modified
        0x00b0:  2d53 696e 6365 3a20 5475 652c 2032 3620  -Since:.Tue,.26.
        0x00c0:  4d61 7220 3230 3234 2031 373a 3339 3a31  Mar.2024.17:39:1
        0x00d0:  3420 474d 540d 0a49 662d 4e6f 6e65 2d4d  4.GMT..If-None-M
        0x00e0:  6174 6368 3a20 2262 3336 3835 3338 3561  atch:."b3685385a
        0x00f0:  3437 6664 6131 3a30 220d 0a55 7365 722d  47fda1:0"..User-
        0x0100:  4167 656e 743a 204d 6963 726f 736f 6674  Agent:.Microsoft
        0x0110:  2d43 7279 7074 6f41 5049 2f31 302e 300d  -CryptoAPI/10.0.
        0x0120:  0a48 6f73 743a 2063 746c 646c 2e77 696e  .Host:.ctldl.win
        0x0130:  646f 7773 7570 6461 7465 2e63 6f6d 0d0a  dowsupdate.com..
        0x0140:  0d0a                                     ..
```

A quick decode

Should ctldl. Be safelisted?

- Expected Windows behaviour
- Not evil, just vulnerable
- Don't block without a plan "B"
  - Really hard to implement
- Kind of stuck with what you've got

Answers - 172.208.51.75



High	192.168.100.136	52.226.139.180	0.00%	23h56m0s	0
High	192.168.100.150	52.226.139.180	0.00%	23h56m31s	0
High	192.168.100.136	172.208.51.75	97.90%	4h49m14s	0
High	192.168.100.152	ctldl.windowsupdate.com	91.60%	26m39s	0
High	192.168.100.150	ctldl.windowsupdate.com	90.90%	22m2s	0

Lots of connections to an odd port

Beacon behaviour

```
student@thunt:~/lab3$ beacon-conn 192.168.100.136 172.208.51.75
12 499
13 555
14 556
15 555
16 550
17 555
18 554
19 564
20 551
21 549
22 558
23 557
00 553
01 555
02 556
03 555
04 548
05 548
06 552
07 552
08 557
09 549
10 556
11 554
12 43
student@thunt:~/lab3$ _
```

Absolutely a beacon
Note small amount of jitter

172.208.51.75 (cont)

```
student@thunt:~/lab3$ fq 172.208.51.75
DNS info
HTTP info
TLS info
172.208.51.75    -         -
student@thunt:~/lab3$ _
```

No DNS queries are suspicious
Note lack of SNI info
Could be TLS 1.3 with SNI obfuscated

TLS info

Obfuscating SNI via TLS 1.3

```
student@thunt:~/lab3$ grep 172.208.51.75 ssl.log | head -5
1714219550.620875       CKaCdw18JdrGBmBLk2      192.168.100.136 50165   172.208.51.75   7707    TLSv13  TLS_AES_128_GCM_SHA256  x25519  -       F
-       -       T       CsiI    -       -       -       -       19e29534fd49dd27d09234e639c4057e        f4febc55ea12b31ae17cfb7e614afda8
1714219551.710555       Cessnz2kUoyOztGnC4      192.168.100.136 50166   172.208.51.75   7707    TLSv13  TLS_AES_128_GCM_SHA256  x25519  -       F
-       -       T       CsiI    -       -       -       -       19e29534fd49dd27d09234e639c4057e        f4febc55ea12b31ae17cfb7e614afda8
1714219559.592323       ChxVGE4YCSJIAFbNe9      192.168.100.136 50167   172.208.51.75   7707    TLSv13  TLS_AES_128_GCM_SHA256  x25519  -       F
-       -       T       CsiI    -       -       -       -       19e29534fd49dd27d09234e639c4057e        f4febc55ea12b31ae17cfb7e614afda8
1714219565.935692       CsjPcAq8aty75jcU1       192.168.100.136 50168   172.208.51.75   7707    TLSv13  TLS_AES_128_GCM_SHA256  x25519  -       F
-       -       T       CsiI    -       -       -       -       19e29534fd49dd27d09234e639c4057e        f4febc55ea12b31ae17cfb7e614afda8
1714219573.004372       Cpq2lB2JQJOXaB8Jkb      192.168.100.136 50169   172.208.51.75   7707    TLSv13  TLS_AES_128_GCM_SHA256  x25519  -       F
-       -       T       CsiI    -       -       -       -       19e29534fd49dd27d09234e639c4057e        f4febc55ea12b31ae17cfb7e614afda8
student@thunt:~/lab3$
```

JA3 hash of client hello is about all we have to go on

Checking TCP port 7707

Open Ports on the ScienceLogic Data Collector Appliance

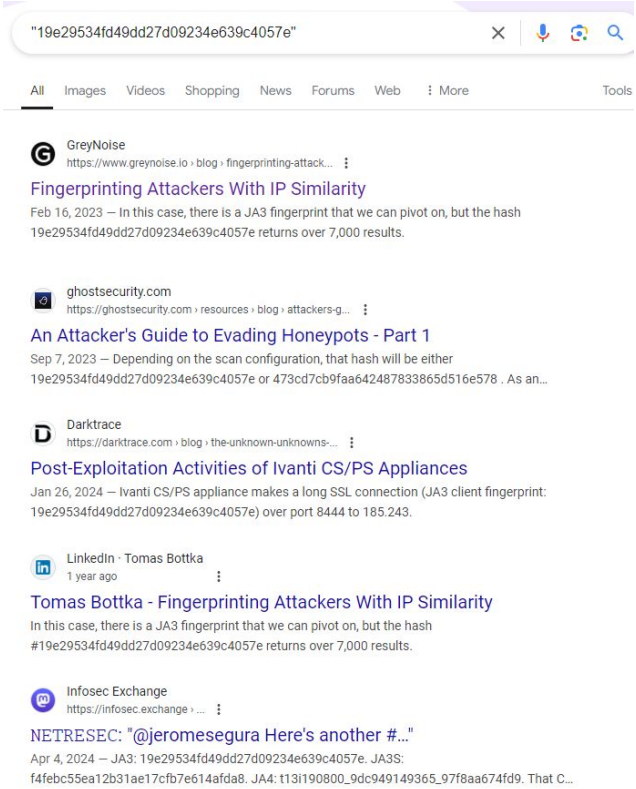
Name	Description	Protocol	Port
Data Pull	Requests from Database Servers to retrieve collected data. In a Phone Home configuration, this port is accessed via an SSH tunnel created by the Data Collector.	TCP	7707
SSH	Optional. For ssh sessions from user workstation.	TCP	22
Web Configurator	Configuration Utility from browser session on user workstation. **NOTE:** For Military Unique Deployment (MUD) configurations, this utility and port are disabled by default. They can be enabled for initial configuration, but must be disabled again after the configuration process is complete.	TCP	7700
SNMP	Optional. SNMP information about the Data Collector can be collected by SL1.	UDP	161
SNMP Traps	Optional. Can receive SNMP traps from managed devices.	UDP	162
Syslog messages	Optional. Can receive syslog messages from managed devices.	UDP	514
HTTPS Secure Interface	Optional. Data from the ScienceLogic Agent running on a monitored device.	TCP	443

Open Ports on the ScienceLogic Message Collector Appliance

Name	Description	Protocol	Port
Data Pull	Requests from Database Servers to retrieve collected data. In a Phone Home configuration, this port is accessed via an SSH tunnel created by the Message Collector.	TCP	7707
SSH	Optional. For ssh sessions from user workstation.	TCP	22
Web Configurator	Configuration Utility from browser session on user workstation. **NOTE:** For Military Unique Deployment (MUD) configurations, this utility and port are disabled by default. They can be enabled for initial configuration, but must be disabled again after the configuration process is complete.	TCP	7700

This does not seem likely

Not getting the warm fuzzies



JA3 hash associated with Sliver C2

Any solid conclusions?

- We absolutely need host data
- Need to know which app is making these conns
- Sysmon/BeaKer data would be perfect
- If not, time for incident response
  - Don't cross active/passive line
- If we go down the rabbit hole, this is AsyncRAT

https://www.activecountermeasures.com/malware-of-the-day-asyncrat/

Remember "don't trust 8075"?

```
student@thunt:~/lab3$ dig -x 172.208.51.75 | grep arpa
;75.51.208.172.in-addr.arpa.    IN      PTR
51.208.172.in-addr.arpa. 5      IN      SOA     ns1-32.azure-dns.com. azuredns-hostmaster.microsoft.com. 1 3600 300 2419200 300
student@thunt:~/lab3$ whois -h whois.cymru.com " -v 172.208.51.75"
AS      | IP               | BGP Prefix          | CC | Registry | Allocated  | AS Name
8075    | 172.208.51.75    | 172.208.0.0/13      | GB | ripencc  | 2002-02-13 | MICROSOFT-CORP-MSN-AS-BLOCK, US
student@thunt:~/lab3$ _
```

ASN 8075 now overlaps Azure
Anyone who knows what they are doing can spin up instances in 8075!
Pay attention to PTR records

Let's talk about TLS 1.3

- SNI can be encrypted
- This obfuscates it from view
- Client makes an "A" record query for IP address of website
- It then makes a "HTTPS" record query for server's public key
- Shared secret generated to obfuscate SNI

Query examples

"A" query then "HTTPS"

```
student@thunt:~/lab3$ cat dns.log | zcutter query qtype_name answers | head
www.bing.com    A       wwwprod.www-bing-com.akadns.net,www.bing.com.edgekey.net,e86303.dscx.akamaiedge.net,23.53.4.107,23.53.4.16,23.53.4.24,23.53
.4.19,23.53.4.11,23.53.4.18,23.53.4.34,23.53.4.32
www.bing.com    HTTPS   wwwprod.www-bing-com.akadns.net,www.bing.com.edgekey.net,e86303.dscx.akamaiedge.net
www.bing.com    A       wwwprod.www-bing-com.akadns.net,www.bing.com.edgekey.net,e86303.dscx.akamaiedge.net,23.53.4.34,23.53.4.26,23.53.4.107,23.53
.4.18,23.53.4.32,23.53.4.24,23.53.4.33
www.bing.com    HTTPS   wwwprod.www-bing-com.akadns.net,www.bing.com.edgekey.net,e86303.dscx.akamaiedge.net
r.clarity.ms    A       clarity-ingest-eus2-b-sc.eastus2.cloudapp.azure.com,20.119.174.243
r.clarity.ms    HTTPS   clarity-ingest-eus2-b-sc.eastus2.cloudapp.azure.com
edge.microsoft.com      A       edge-microsoft-com.dual-a-0036.a-msedge.net,dual-a-0036.a-msedge.net,204.79.197.239,13.107.21.239
edge.microsoft.com      HTTPS   edge-microsoft-com.dual-a-0036.a-msedge.net
edge.microsoft.com      HTTPS   edge-microsoft-com.dual-a-0036.a-msedge.net
edge.microsoft.com      A       edge-microsoft-com.dual-a-0036.a-msedge.net,dual-a-0036.a-msedge.net,204.79.197.239,13.107.21.239
```

SNI encrypted

Which causes Zeek to report

```
student@thunt:~/lab3$ cat ssl.log | zcutter id.resp_h version server_name | grep TLSv13 | sort | uniq
13.107.246.36   TLSv13  edgestatic.azureedge.net
172.208.51.75   TLSv13  -
23.215.25.190   TLSv13  www.microsoft.com
23.53.4.107     TLSv13  -
23.53.4.11      TLSv13  -
23.53.4.16      TLSv13  -
23.53.4.16      TLSv13  www.bing.com
23.53.4.18      TLSv13  -
23.53.4.25      TLSv13  -
23.53.4.26      TLSv13  -
23.53.4.26      TLSv13  r.bing.com
23.53.4.26      TLSv13  www.bing.com
23.53.4.33      TLSv13  -
23.53.4.34      TLSv13  -
23.53.4.34      TLSv13  www.bing.com
23.53.4.8       TLSv13  -
23.53.4.9       TLSv13  -
52.123.251.167  TLSv13  config.edge.skype.com
52.123.251.180  TLSv13  config.edge.skype.com
52.123.251.184  TLSv13  -
52.123.251.184  TLSv13  config.edge.skype.com
student@thunt:~/lab3$
```

So are we out of luck?

- Not exactly
- Our job is now harder
- But not impossible
- We still have that original "A" query that we can work with

Leveraging DNS

```
student@thunt:~/lab3$ cat ssl.log | zcutter id.resp_h version server_name | grep TLSv13 | grep '-' | sort | uniq
172.208.51.75   TLSv13  -
23.53.4.107     TLSv13  -
23.53.4.11      TLSv13  -
23.53.4.16      TLSv13  -
23.53.4.18      TLSv13  -
23.53.4.25      TLSv13  -
23.53.4.26      TLSv13  -
23.53.4.33      TLSv13  -
23.53.4.34      TLSv13  -
23.53.4.8       TLSv13  -
23.53.4.9       TLSv13  -
52.123.251.184  TLSv13  -
student@thunt:~/lab3$ fq 23.53.4.107 | head -2
DNS info
p-static.bing.trafficmanager.net,r.bing.com.edgekey.net,e86303.dscx.akamaiedge.net,23.53.4.26,23.53.4.9,23.53.4.1
34,23.53.4.32   r.bing.com
student@thunt:~/lab3$ fq 23.53.4.25 | head -2
DNS info
wwwprod.www-bing-com.akadns.net,www.bing.com.edgekey.net,e86303.dscx.akamaiedge.net,23.53.4.10,23.53.4.16,23.53.4
32,23.53.4.107,23.53.4.33       www.bing.com
student@thunt:~/lab3$ fq 52.123.251.184 | head -2
DNS info
config.edge.skype.com.trafficmanager.net,mira.config.skype.com,svc.ha-teams.office.com,svc.ms-acdc-teams.office.c
187,52.123.251.166      config.edge.skype.com
student@thunt:~/lab3$ _
```

This was the C2 server

Wait, so you still see the FQDN???

- Usually, yes
- So it has made life harder for security folks without really improving privacy
  - I'm convinced RFC writers officially hate us
- Caveat is DNS over TLS
  - Combine it with v1.3 and we are totally screwed
- Within corporate, just say no to both

Down the rabbit hole with Lab4

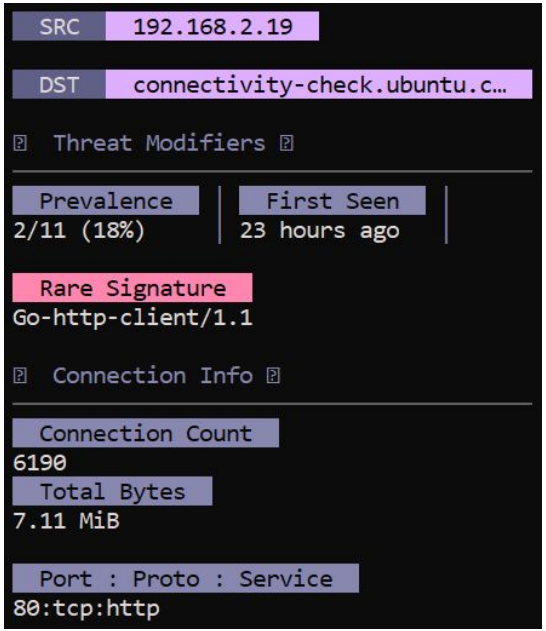
- Move to the "lab4" dir and run data
- During Zeek import, minor soft error
  - "line 30: Failed to open GeoIP…"
  - Volume mapping being addressed
- This soft error is safe to ignore

```
cd ~/lab4
zeek readpcap ~/lab4/lab4.pcap ~/lab4/
rita import -l ~/lab4/ -d lab4
rita view lab4
```

Walkthrough collaboration

- Threat hunting can be messy
- Not always as clean or obvious as the labs
- Let's go through a noisy dataset
- Lots of severity "High" scores
- May or may not contain C2
- Let's go through each line together
- Please share your techniques for running down each suspect connection

First entry



What do you think?
Please share in Discord

First entry - What is it?

- Ubuntu calling home
- Appears to be a mis-configured system
  - Connecting 260 times per hour (1/14 sec)
  - That should be per day (1/300 sec)
- Should we safelist this?
  - Appears benign
  - But then we would not see misconfiguration
  - But is this the best tool to check for that?
- Personally I would safelist this

What Zeek sees

```
student@thunt:~/lab4$ cat http.log | zcutter host uri user_agent status_code | grep connectivity-check.ubuntu.com | sort | uniq -c | sort -rn
   5902 connectivity-check.ubuntu.com   /       Go-http-client/1.1      204
    288 connectivity-check.ubuntu.com   /       -       204
student@thunt:~/lab4$
```

Request for default index.html
Status code 204 means "No Content"
Unsure why sometimes user agent is missing

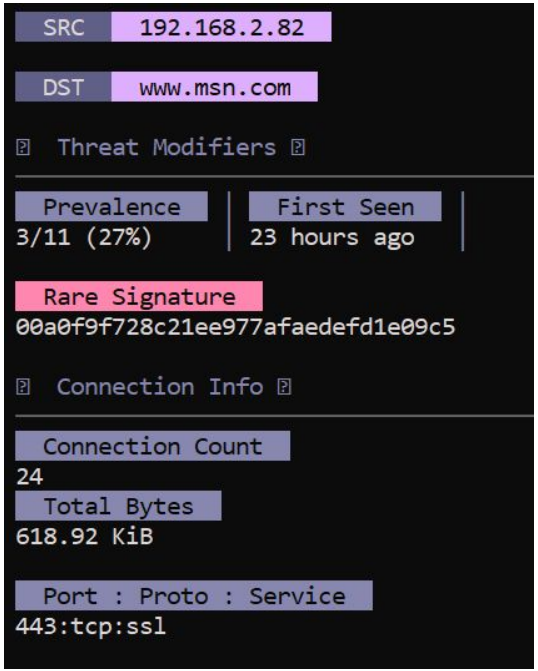
First entry - packet decode

```
14:08:17.236878 IP 192.168.2.19.45565 > 185.125.190.17.80: Flags [P.], seq 1:
], length 129: HTTP: GET / HTTP/1.1
        0x0000:  4500 00b5 f3ce 4000 4006 0c2a c0a8 0213  E.....@.@..*....
        0x0010:  b97d be11 b1fd 0050 1618 9f29 a8f3 dcbd  .}.....P...).....
        0x0020:  8018 01f6 63d5 0000 0101 080a 3f09 7885  ....c.......?.x.
        0x0030:  f1ba f9fc 4745 5420 2f20 4854 5450 2f31  ....GET./.HTTP/1
        0x0040:  2e31 0d0a 486f 7374 3a20 636f 6e6e 6563  .1..Host:.connec
        0x0050:  7469 7669 7479 2d63 6865 636b 2e75 6275  tivity-check.ubu
        0x0060:  6e74 752e 636f 6d0d 0a55 7365 722d 4167  ntu.com..User-Ag
        0x0070:  656e 743a 2047 6f2d 6874 7470 2d63 6c69  ent:.Go-http-cli
        0x0080:  656e 742f 312e 310d 0a41 6363 6570 742d  ent/1.1..Accept-
        0x0090:  456e 636f 6469 6e67 3a20 677a 6970 0d0a  Encoding:.gzip..
        0x00a0:  436f 6e6e 6563 7469 6f6e 3a20 636c 6f73  Connection:.clos
        0x00b0:  650d 0a0d 0a                             e....
14:08:17.314892 IP 185.125.190.17.80 > 192.168.2.19.45565: Flags [P.], seq 1:
85], length 189: HTTP: HTTP/1.1 204 No Content
        0x0000:  4500 00f1 0d7c 4000 3706 fb40 b97d be11  E....|@.7..@.}..
        0x0010:  c0a8 0213 0050 b1fd a8f3 dcbd 1618 9faa  .....P..........
        0x0020:  8018 01fd e260 0000 0101 080a f1ba fa4b  .....`.........K
        0x0030:  3f09 7885 4854 5450 2f31 2e31 2032 3034  ?.x.HTTP/1.1.204
        0x0040:  204e 6f20 436f 6e74 656e 740d 0a73 6572  .No.Content..ser
        0x0050:  7665 723a 206e 6769 6e78 2f31 2e31 342e  ver:.nginx/1.14.
        0x0060:  3020 2855 6275 6e74 7529 0d0a 6461 7465  0.(Ubuntu)..date
        0x0070:  3a20 5468 752c 2032 3320 4d61 7920 3230  :.Thu,.23.May.20
        0x0080:  3234 2031 343a 3038 3a31 3720 474d 540d  24.14:08:17.GMT.
        0x0090:  0a78 2d63 6163 6865 2d73 7461 7475 733a  .x-cache-status:
        0x00a0:  2066 726f 6d20 636f 6e74 656e 742d 6361  .from.content-ca
        0x00b0:  6368 652d 696c 332f 300d 0a78 2d6e 6574  che-il3/0..x-net
        0x00c0:  776f 726b 6d61 6e61 6765 722d 7374 6174  workmanager-stat
        0x00d0:  7573 3a20 6f6e 6c69 6e65 0d0a 636f 6e6e  us:.online..conn
        0x00e0:  6563 7469 6f6e 3a20 636c 6f73 650d 0a0d  ection:.close...
        0x00f0:  0a                                       .
:
```

"No Content" but header includes status info for NetworkManager

Second entry

Critical	192.168.2.82	www.msn.com	97.70%	3m5s	0



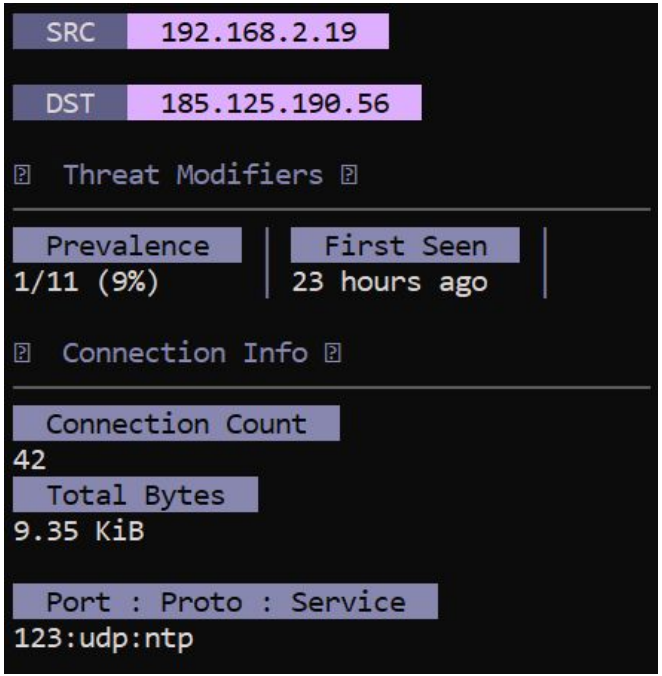
Your thoughts?

Second entry - What is it?

- Windows calling home
- There is an MSN app, but connection frequency is too slow
- Reporting that this is used to deliver ads to Windows
- Should we safelist?
  - Same caveats as Ubuntu checkin
  - I would personally safelist

Third entry

High	192.168.2.19	185.125.190.56	100.00%	3s	0



Your thoughts?

Third entry - What is it?

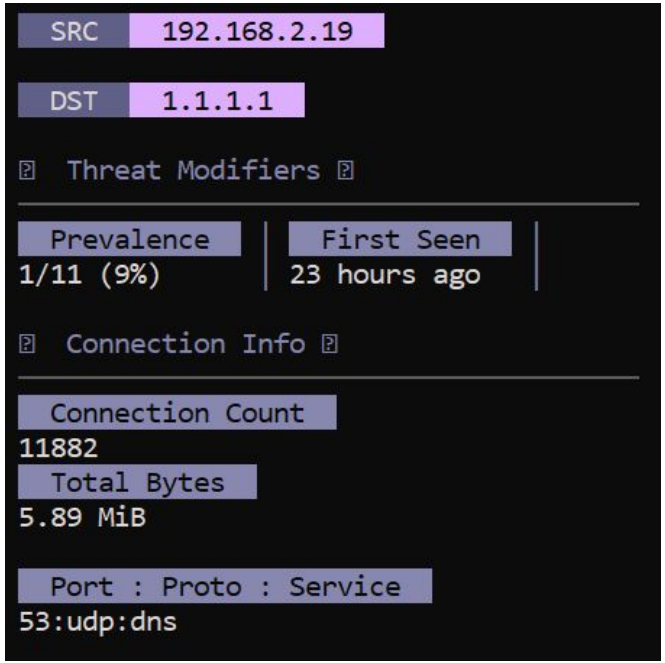
```
student@thunt:~/lab4$ grep 185.125.190.56 ntp.log | head -3
1716474394.945852       CVOmX22THZfWJicavg      192.168.2.19    49911   185.125.190.56  123     4       3       0       1.000000        1.000000
0.000000        0.000000        \x00\x00\x00\x00        0.000000        0.000000        0.000000        1716474394.220218       0
1716474394.945942       CVOmX22THZfWJicavg      192.168.2.19    49911   185.125.190.56  123     4       3       0       1.000000        1.000000
0.000000        0.000000        \x00\x00\x00\x00        0.000000        0.000000        0.000000        1716474394.220218       0
1716474395.027188       CVOmX22THZfWJicavg      192.168.2.19    49911   185.125.190.56  123     4       4       2       1.000000        0.000000
0.001175        0.000153        79.243.60.50    1716474382.028316       1716474394.220218       1716474394.986689       1716474394.986726       0
student@thunt:~/lab4$ dig -x 185.125.190.56 | grep arpa
;56.190.125.185.in-addr.arpa.   IN      PTR
56.190.125.185.in-addr.arpa. 5  IN      PTR     prod-ntp-3.ntp4.ps5.canonical.com.
56.190.125.185.in-addr.arpa. 5  IN      PTR     prod-ntp-3.ntp1.ps5.canonical.com.
student@thunt:~/lab4$ fq 185.125.190.56
DNS info
HTTP info
TLS info
student@thunt:~/lab4$ _
```

Legit NTP

Safelist by IP since FQDN not being used

Fourth entry

High	192.168.2.19	1.1.1.1	100.00%	1m15s	0



Your thoughts?

Fourth entry - What is it?

- Cloudflare public DNS resolver
- Fast, claims higher privacy
- Used by many orgs

```
student@thunt:~/lab4$ dig -x 1.1.1.1 | grep arpa
;1.1.1.1.in-addr.arpa.          IN      PTR
1.1.1.1.in-addr.arpa.   5       IN      PTR     one.one.one.one.
student@thunt:~/lab4$ whois -h whois.cymru.com " -v 1.1.1.1"
AS      | IP               | BGP Prefix          | CC | Registry | Allocated  | AS Name
13335   | 1.1.1.1          | 1.1.1.0/24          | AU | apnic    | 2011-08-11 | CLOUDFLARENET, US
student@thunt:~/lab4$
```

Is it evil?

- Wait… 11,882 connections
- Could this be C2 over DNS???
- Note "Subdomains" did not trigger
  - Looking for excessive number of FQDNs in domain
  - Not detected in this situation
  - Just a busy DNS server
- Should we safelist this?
  - Absolutely not!
  - Will be blind to C2 over DNS
  - Make a note and live with it

Checking for C2 over DNS

Settings in /etc/rita/config.hjson

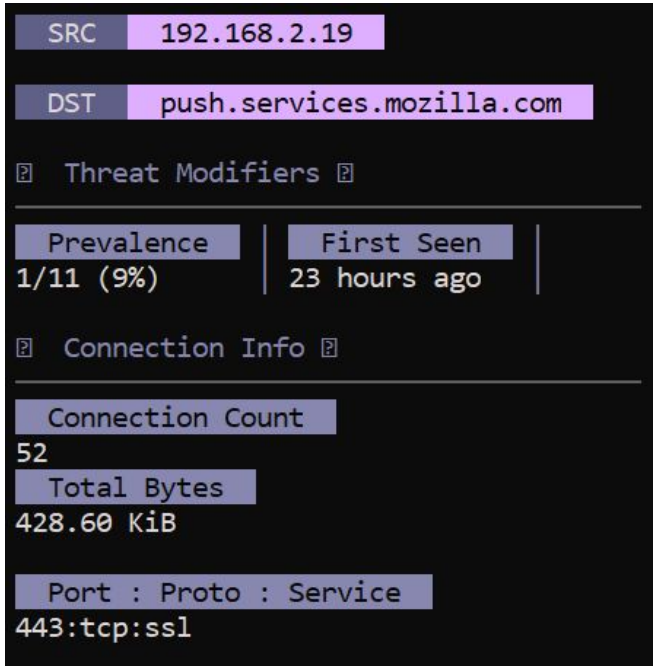
```
        c2_score_thresholds: {
            // number of subdomains
            base: 100,
            low: 500,
            medium: 800,
            high: 1000
        },
```

What was seen in dns.log

```
student@thunt:~/lab4$ cat dns.log | zcutter id.resp_h query | grep '1\.1\.1\.1' | cut -f 2 | sort | cut -d . -f 1 | uniq -c
     44 1
  23764 connectivity-check
student@thunt:~/lab4$ _
```

Fifth entry

High	192.168.2.19	push.services.mozilla.com	90.50%	25h3m15s	0



Your thoughts?

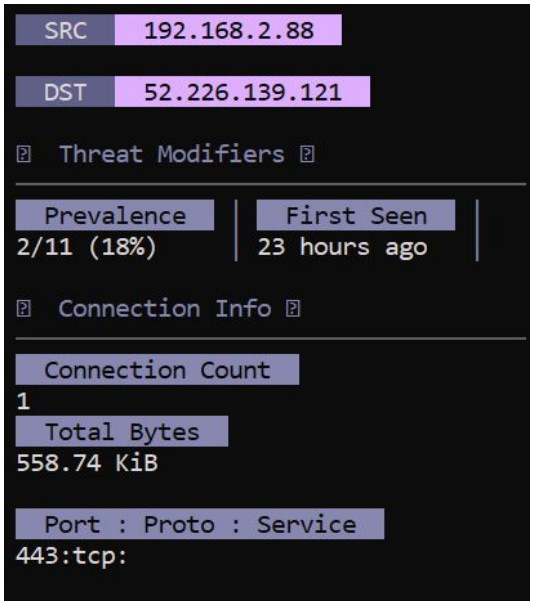
Fifth entry - What is it?

- Firefox service for website notifications
- Let's approved sites send you pop-ups
- Because we all agree that the Internet needs more pop-up notifications ;-)
- This can be safelisted, but may want to disable in the browser

https://support.mozilla.org/en-US/kb/push-notifications-firefox

Sixth & Seventh entry

High	192.168.2.88	52.226.139.121	0.00%	23h59m29s	0
High	192.168.2.87	52.226.139.185	0.00%	23h37m33s	0



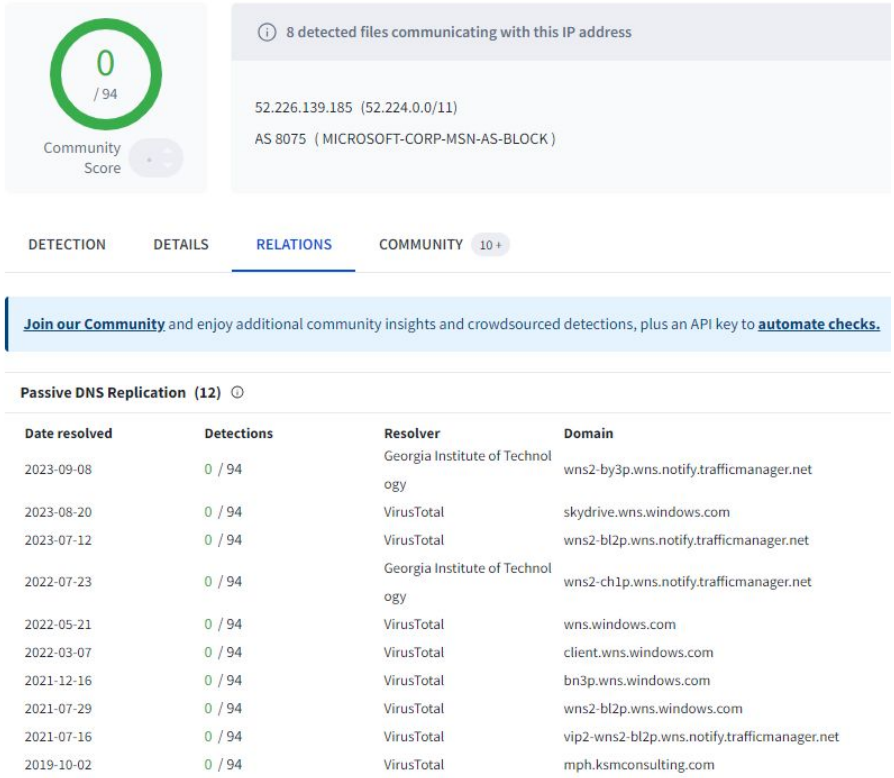
These two are nearly identical.

Thoughts?

Sixth & Seventh - What is it?

- These are a challenge
- No header info to work with
- No DNS info to work with
- whois points at Microsoft but no PTR
- What does VirusTotal think?

Checking VirusTotal



Date resolved	Detections	Resolver	Domain
2023-09-08	0 / 94	Georgia Institute of Technology	wns2-by3p.wns.notify.trafficmanager.net
2023-08-20	0 / 94	VirusTotal	skydrive.wns.windows.com
2023-07-12	0 / 94	VirusTotal	wns2-bl2p.wns.notify.trafficmanager.net
2022-07-23	0 / 94	Georgia Institute of Technology	wns2-ch1p.wns.notify.trafficmanager.net
2022-05-21	0 / 94	VirusTotal	wns.windows.com
2022-03-07	0 / 94	VirusTotal	client.wns.windows.com
2021-12-16	0 / 94	VirusTotal	bn3p.wns.windows.com
2021-07-29	0 / 94	VirusTotal	wns2-bl2p.wns.windows.com
2021-07-16	0 / 94	VirusTotal	vip2-wns2-bl2p.wns.notify.trafficmanager.net
2019-10-02	0 / 94	VirusTotal	mph.ksmconsulting.com

Note history of Microsoft related domains

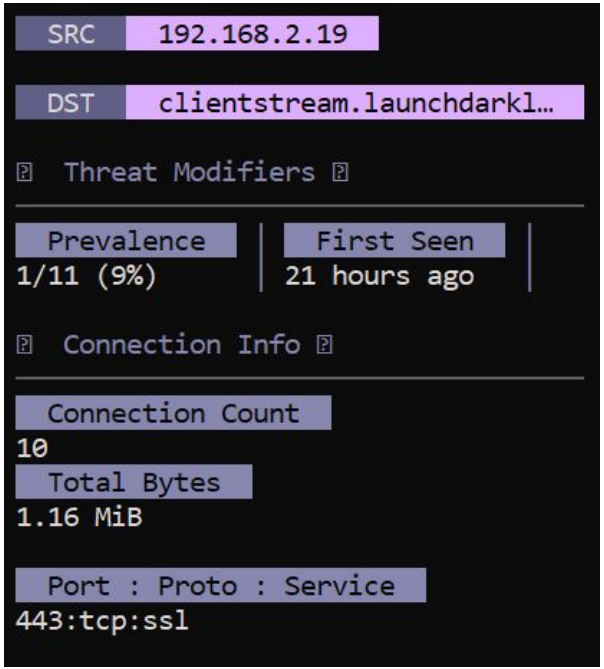
WNS known to make long conns

What to do

- Leaning towards it's OK to safelist
- Would be nice to have more data
- We could reboot the source systems
  - Connection should re-establish
  - Capture original DNS query
  - Capture transport negotiation
  - Would give us definitive info

Eighth entry

High	192.168.2.19	clientstream.launchdarkly.…	0.00%	39h46m27s	0



Your thoughts?

Eighth entry - What is it?

- SaaS service for software development
- Let's you bug and monitor your code
  - Make live changes in production
  - Segregate who sees which features
  - Monitor app usage and collect statistics
- I would really want to know which app is reporting data
- As a security person…I have concerns
  - Would not safelist this (at least for now)

https://launchdarkly.com/how-it-works/

Sanity check time

- We have 5 entries we could safelist
- Let's add them in now

```
connectivity-check.ubuntu.com
www.msn.com
185.125.190.56    (NTP)
push.services.mozilla.com
52.226.139.0/24    (MS WNS)
```

How to make the change

```
sudo nano /etc/rita/config.hjson
```

```
    // connections involving ranges entered into never_included_subnets are filtered out at import time
    never_included_subnets: ["52.113.194.132/32", "52.177.166.224/32", "185.125.190.56/32", "52.226.139.0/24"], // array of CIDRs
    never_included_domains: ["tile-service.weather.microsoft.com", "connectivity-check.ubuntu.com", "www.msn.com", "push.services.mozilla.com"], // arra
    filter_external_to_internal: true // ignores any entries where communication is occurring from an external host to an internal host
},
scoring: {
```

Save changes, then re-import data:

```
sudo rita import -l ~/lab4/ -d lab4b
rita view lab4b
```

New view of our data

Severity	Source	Destination	Beacon	Duration	Subdomains
High	192.168.2.19	1.1.1.1	100.00%	1m15s	0
High	192.168.2.19	34.107.243.93	86.90%	25h8m17s	0
High	192.168.2.19	clientstream.launchdarkly.…	0.00%	39h46m27s	0
High	192.168.2.19	3.33.235.18	0.00%	14h1m59s	0
High	192.168.2.77	64.23.195.234	0.00%	23h59m55s	0
High	192.168.2.77	172.208.51.75	0.00%	48h1m41s	0
High	192.168.2.19	events.launchdarkly.com	99.90%	4h22m58s	0
High	192.168.2.19	76.223.31.44	0.00%	11h51m0s	0

Not safelisted

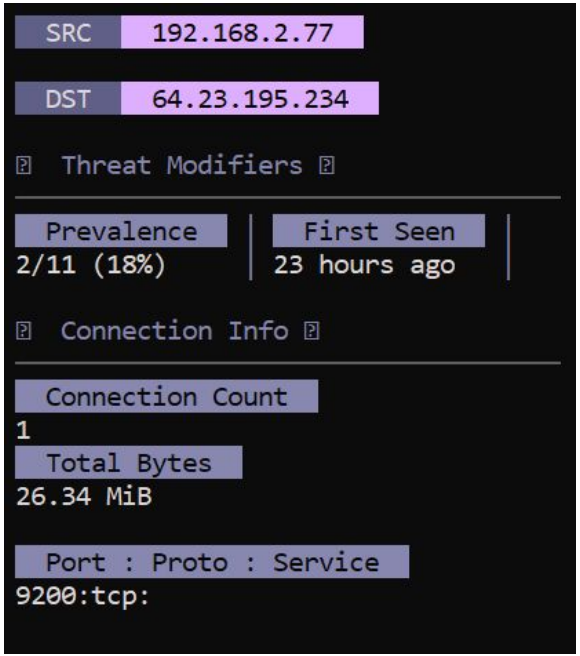
Appears due to long conn bug being addressed

Not safelisted

New entries from here down

Next entry - 64.23.195.234

High	192.168.2.77	64.23.195.234	0.00%	23h59m55s	0



Your thoughts?

64.23.195.234 - What is it?

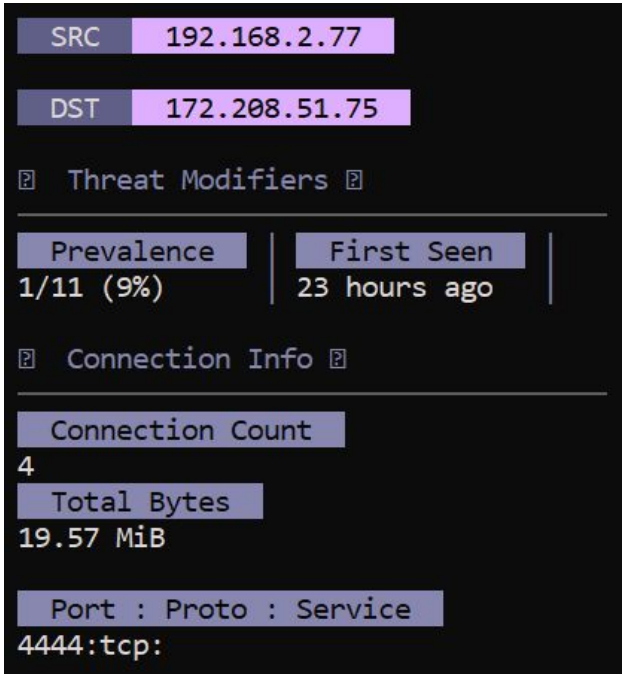
- Long conn - No DNS or app data to use
- Running dig & whois shows DigitalOcean
  - But no useful host info
- VirusTotal info not definitive
- TCP/9200 is Elasticsearch
  - This may help run down why it's in use
- App is usually a browser, so BeaKer type info many not be helpful
- Chat with user or power cycle the source

Wait, so safelist or not?

- Do not yet have a definitive answer on 64.23.195.234
- If it is Elasticsearch, it's probably not evil
- But best to check and be sure
- We will usually not be able to solve everything in a quick easy pass
- Some items will require additional research

Next entry - 172.208.51.75

High	192.168.2.77	172.208.51.75	0.00%	48h1m41s	0



Your thoughts?

172.208.51.75 - What is it?

- Long conn with no DNS or app info
- Connecting to strange port - TCP/4444
  - SOHO router console port
  - Metasploit default listener
  - Various malware
- Similar to the last one, not much to go on within the datastream without seeing initial connection

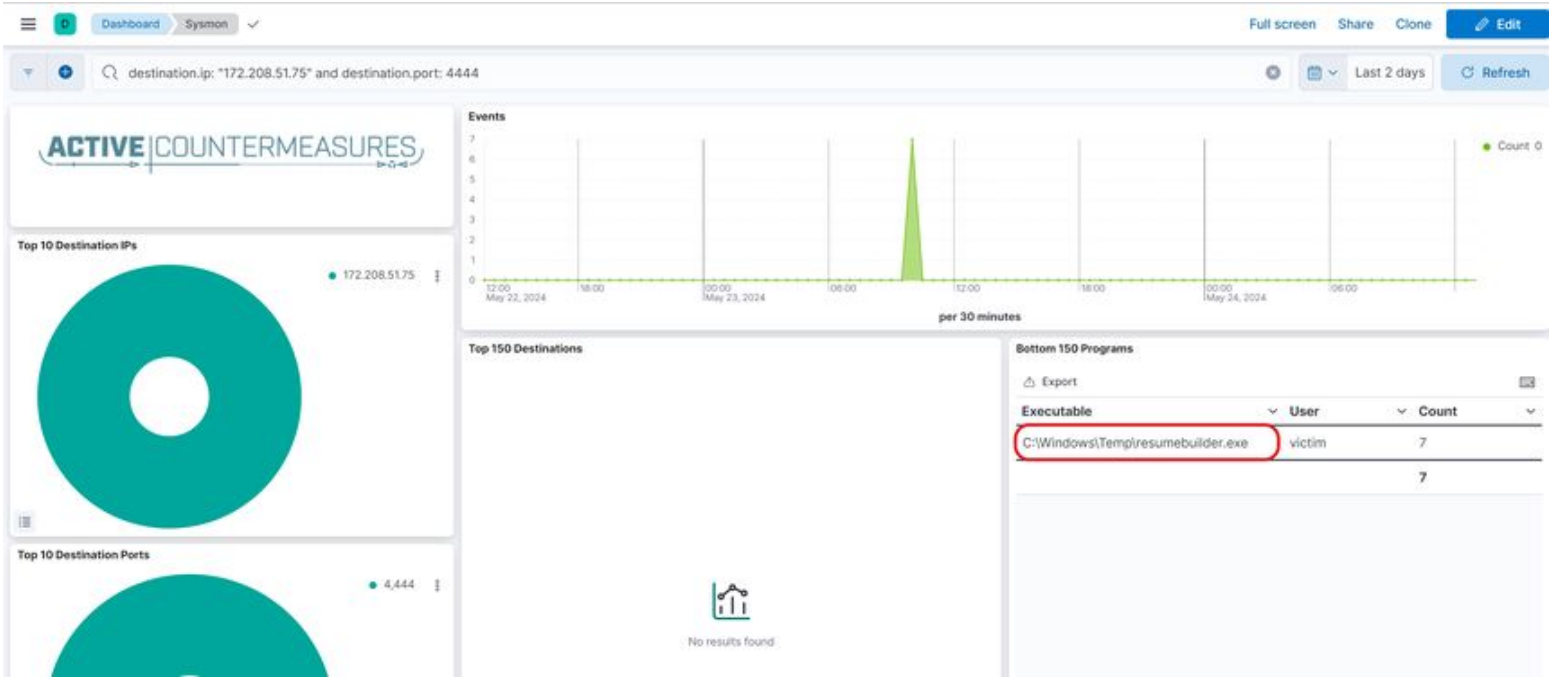
How 48 hours in 24 hour pcap?

```
cat conn.log | zcutter -d ts id.orig_h id.orig_p id.resp_h id.resp_p service duration | grep 172.208.51.75
```

```
student@thunt:~/lab4$ cat conn.log | zcutter -d ts id.orig_h id.orig_p id.resp_h id.resp_p service duration | grep 172.208.51.75
2024-05-23T14:07:10+0000        192.168.2.77    63576   172.208.51.75   4444    -       122.245912
2024-05-23T14:06:41+0000        192.168.2.77    63573   172.208.51.75   4444    -       29.273174
2024-05-23T14:06:42+0000        192.168.2.77    63575   172.208.51.75   4444    -       86375.033262
2024-05-23T14:06:42+0000        192.168.2.77    63574   172.208.51.75   4444    -       86375.274844
student@thunt:~/lab4$
```

Four connections made
Some connections ran concurrently
Sum of durations is just over 48 hours

What if we can pivot to BeaKer?



"resumebuilder.exe" running out of C:\Windows\Temp

Red flags in BeaKer data

- A binary executable named "resumebuilder" doesn't make sense
- Running in the Windows temp directory
  - Not where apps are usually run
  - Leveraged by malware due to loose perms
- Leaning towards thinking its evil
- Binary analysis would be helpful
- BeaKer data can really help to clarify

Final walkthrough - Tuning RITA

- Still have 4 pages of severity "high"
- Can we tune some of these out?
- RITA evaluates 4+ conns as beacons
  - We changed this to 12 for the labs
  - Low conn count a concern in high security envs
  - Are we worried about high level nation state?
  - If not, we could increase this value further

Increase beacon count

```
sudo nano /etc/rita/config.hjson
```

Change:

```
  scoring: {
      beacon: {
          // The default minimum number of unique connections used for beacons analysis.
          // Any two hosts connecting fewer than this number will not be analyzed. You can
          // safely increase this value to improve performance if you are not concerned
          //  about slow beacons.
          unique_connection_threshold: 12, // min number of unique connections to qualify as beacon
```

To this:

```
scoring: {
    beacon: {
        // The default minimum number of unique connections used for beacons analysis.
        // Any two hosts connecting fewer than this number will not be analyzed. You can
        // safely increase this value to improve performance if you are not concerned
        //  about slow beacons.
        unique_connection_threshold: 20, // min number of unique connections to qualify as beacon
```

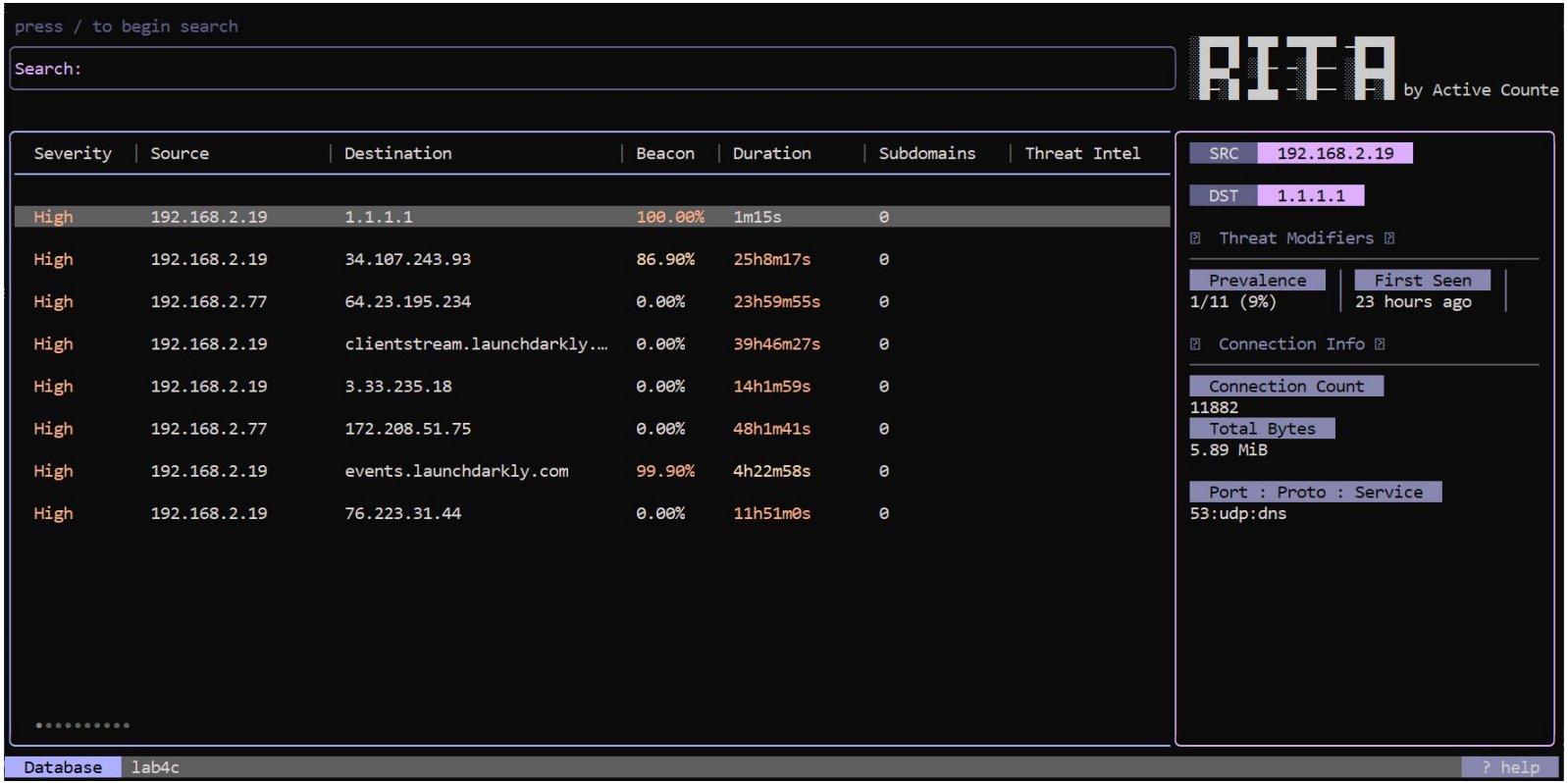
Then save and exit

Recheck the data

```
rita import -l ~/lab4/ -d lab4c

riva view lab4c
```

New results


press / to begin search
Search:
RITA
by Active Counte
Severity | Source | Destination | Beacon | Duration | Subdomains | Threat Intel
High 192.168.2.19 1.1.1.1 100.00% 1m15s 0
High 192.168.2.19 34.107.243.93 86.90% 25h8m17s 0
High 192.168.2.77 64.23.195.234 0.00% 23h59m55s 0
High 192.168.2.19 clientstream.launchdarkly.… 0.00% 39h46m27s 0
High 192.168.2.19 3.33.235.18 0.00% 14h1m59s 0
High 192.168.2.77 172.208.51.75 0.00% 48h1m41s 0
High 192.168.2.19 events.launchdarkly.com 99.90% 4h22m58s 0
High 192.168.2.19 76.223.31.44 0.00% 11h51m0s 0
SRC 192.168.2.19
DST 1.1.1.1
Threat Modifiers
Prevalence
1/11 (9%)
First Seen
23 hours ago
Connection Info
Connection Count
11882
Total Bytes
5.89 MiB
Port : Proto : Service
53:udp:dns
Database lab4c
? help

What did the change do?

- Reduced the number of severity high items
  - 8 removed
  - 3 pages instead of 4
  - Stuff we care about is still there
- Could we improve further?
  - Increasing to 47 removed half remaining entries
  - Change long conn thresholds
    - High to 20 hours
    - Medium to 12
    - Removes another half page of entries

Old school output

```
student@thunt:~/lab4$ rita view --stdout lab4f | head
[+] Running 3/3
 ✔Container rita-syslog-ng   Running0.0s
 ✔Container rita-clickhouse  Healthy0.5s
 ✔Container rita-rita-1      Started0.8s
[+] Creating 2/0
 ✔Container rita-syslog-ng   Running0.0s
 ✔Container rita-clickhouse  Running0.0s
Viewing database: lab4f
Severity,Source IP,Destination IP,FQDN,Beacon Score,Strobe,Total Duration,Long Connection Score,Subdomains,C2 Over DNS Score,Threat Intel,Prevalence,First Se
en,Missing Host Header,Connection Count,Total Bytes,Port:Proto:Service,Modifiers
High,192.168.2.19,1.1.1.1,,1,false,75.863075,0,0,0,false,0.09090909,23 hours ago,false,11882,6173838,"53:udp:dns",""
High,192.168.2.19,34.107.243.93,,0,false,90497.625,0.8,0,0,false,0.09090909,23 hours ago,false,54,440040,"443:tcp:ssl,443:tcp:",""
High,192.168.2.19,::,clientstream.launchdarkly.com,0,false,143187.56,0.8,0,0,false,0.09090909,21 hours ago,false,10,1218730,"443:tcp:ssl",""
High,192.168.2.77,172.208.51.75,,0,false,172901.83,0.8,0,0,false,0.09090909,23 hours ago,false,4,20519582,"4444:tcp:",""
High,192.168.2.77,64.23.195.234,,0,false,86395.15,0.8,0,0,false,0.18181819,23 hours ago,false,1,27615573,"9200:tcp:",""
High,192.168.2.19,::,events.launchdarkly.com,0.999,false,15778.463,0.40957263,0,0,false,0.09090909,23 hours ago,false,97,1731119,"443:tcp:ssl",""
High,192.168.2.19,::,www.expressapisv2.net,0.83,false,51.69783,0,0,0,false,0.18181819,23 hours ago,false,432,10162508,"443:tcp:ssl","rare_signature:871a754af
286dfb70c1b53c6887c62e0"
High,192.168.2.19,3.33.235.18,,0,false,50519.387,0.6508291,0,0,false,0.09090909,11 hours ago,false,3,427089,"443:tcp:ssl",""
write /dev/stdout: broken pipe
[+] Stopping 1/0
 ✔Container rita-rita-1  Stopped0.0s
student@thunt:~/lab4$ _
```

Currently "--stdout" is an undocumented switch.

Closing thoughts

- Remember the process
  - Identify connection persistency
  - Identify business need if present
  - Investigate external IP
  - Investigate internal IP
- Disposition each IP
  - Pretty certain it's still pristine
  - Pretty certain it's compromised
- Don't cross the passive/active line

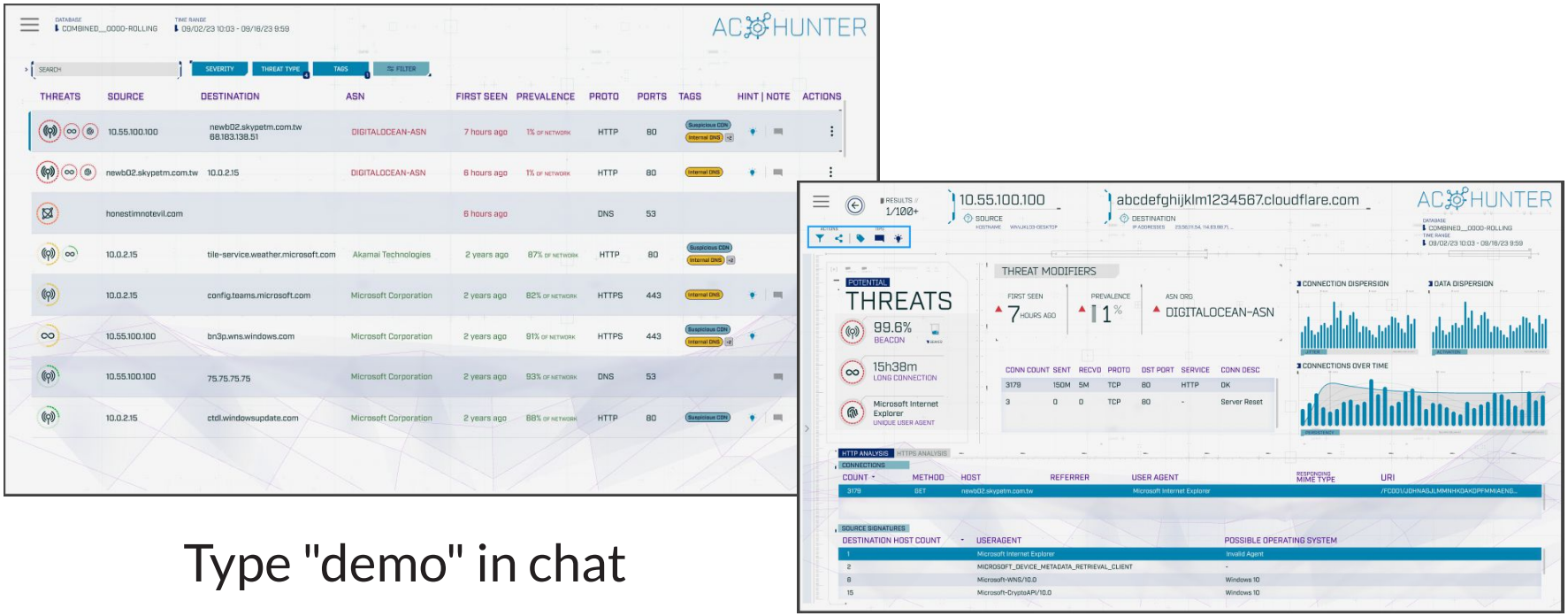
If you want to keep practicing

- Check our malware of the day blog
- Skip to the bottom, download the 24 hour long pcap file
- Import into RITA
- Review the results
- When done, check the blog for answers
  - Did you miss anything?

https://www.activecountermeasures.com/?s=malware+of+the+day

Want an AC-Hunter demo?

Updated version coming soon!



Type "demo" in chat

Classes I'm teaching

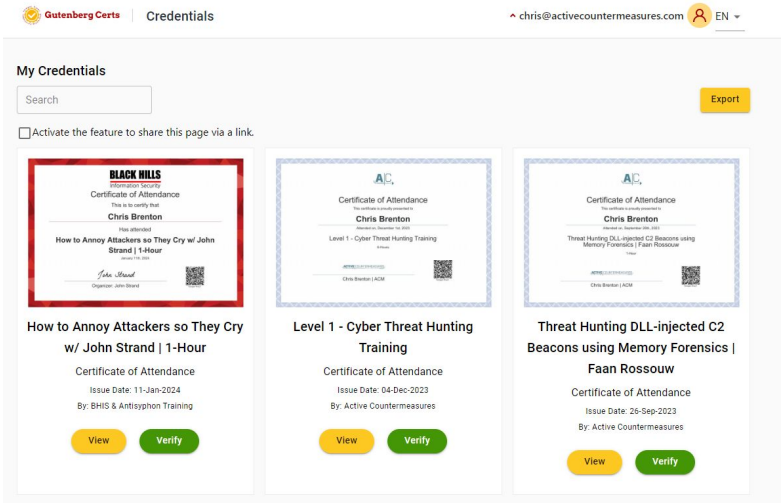
- Advance Network Threat Hunting
  - WWHF Oct 8th & 9th
  - Virtual tickets still available
- Intro to Docker (new - pay what you can)
- Intro to Packet Decoding (pay what you can)
- Security Compliance & Leadership

https://www.antisyphontraining.com/mission/our-instructors/instructor-profile-chris-brenton/

When will I get my cert?

Certs go out within 24 hours.

You can also retrieve your cert from Accredible:
https://v2.accounts.accredible.com/retrieve-credentials



Thank you for attending!

- Thanks for sharing your valuable time with us today
- We hope the class has been helpful
- The team will monitor Discord for any last minute question