BEYOND IDS: PRACTICAL NETWORK HUNTING

BSIDES NYC 2016

JOSH LIBURDI

QUICK INTRODUCTION

Currently: Senior Consultant at CrowdStrike

Previously: Large-scale threat detection at Fortune 25

Focus on threat detection, incident response, network forensics

Twitter: @jshlbrd

AGENDA

Hunting overview

Network hunting tools

Hunting techniques & examples

Jackie @find_evil · 19m

Already told you once... #infosec #monitoring #visibility #dfir #threat #hunting

6 9

A FEW WORDS ON HUNTING

What is it?
- Manual / active threat detection
- Driven by people, not computers
- Based on hypotheses of attacker activity

A FEW WORDS ON HUNTING

What is it?
- Manual / active threat detection
- Driven by people, not computers
- Based on hypotheses of attacker activity

Why should I do it?
- Increases likelihood of identifying previously unknown threats
- Provides coverage for attacker tactics, techniques, and procedures (TTPs)

A FEW WORDS ON HUNTING++

What do I need to do it?
- Data! Highly organized data!
- Time
- Buy-in

A FEW WORDS ON HUNTING++

What do I need to do it?
- Data! Highly organized data!
- Time
- Buy-in

When have I succeeded? (Pick one!)
- You've learned something new about your network
- You've come up with a new way to detect attackers in your network
- You've found an attacker in your network

A FEW WORDS ON HUNTING++

What do I do when I'm done?
- Document what worked, what didn't work
- Automate, automate, automate!

A FEW WORDS ON HUNTING++

What do I do when I'm done?
- Document what worked, what didn't work
- Automate, automate, automate!

How do I know if I'm ready?
- detect-respond.blogspot.com/2015/10/a-simple-hunting-maturity-model.html

ADDITIONAL HUNTING RESOURCES

Not widely discussed publicly

David Bianco
- @davidjbianco
- detect-respond.blogspot.com

Scott J Roberts
- @sroberts
- sroberts.github.io

NETWORK HUNTING TOOLS

Bro (thanks ICSI!)

Laika BOSS (thanks Lockheed Martin!)

Moloch (thanks AOL!)

NETWORK HUNTING TOOLS++

What do these tools have in common?

NETWORK HUNTING TOOLS++

What do these tools have in common?

They all produce network metadata!

NETWORK HUNTING TOOLS++

Bro
- Flow data
- Application layer protocol data

Laika BOSS
- File data

Moloch
- Flow data
- Application layer protocol data
- Full packet capture data *

BRO

```
#fields ts      uid     id.orig_h       id.orig_p       id.resp_h       id.resp_p       trans_de
pth     method  host    uri     referrer        user_agent      request_body_len        response
_body_len       status_code     status_msg      info_code       info_msg        filename
tags    username        password        proxied orig_fuids      orig_mime_types resp_fuids
resp_mime_types
#types  time    string  addr    port    addr    port    count   string  string  string  string
string  count   count   count   string  count   string  string  set[enum]       string  string
set[string]     vector[string]  vector[string]  vector[string]  vector[string]
1084443428.222534       ClqHbmkWuersgyFU4       145.254.160.237 3372    65.208.228.223  80
1       GET     www.ethereal.com        /download.html  http://www.ethereal.com/development.html
        Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.6) Gecko/20040113  0       18070
200     OK      -       -       -       (empty) -       -       -       -       -       FwgLBs1a
UNKB1xBIlc      application/xml
```

```
"META_EXIFTOOL": {
  "FlashPix:LastModifiedBy": "Microsoft Office",
  "FlashPix:[illegible]": "",
  "[illegible]Version": 10.07,
  "FlashPix:ModifyDate": "2015:12:22 09:14:00",
  "FlashPix:LinksUpToDate": 0,
  "FlashPix:Software": "Microsoft Office Word",
  "File:FileInodeChangeDate": "2015:12:29 09:41:56-08:00",
  "FlashPix:CodePage": 1251,
  "FlashPix:Company": "Microsoft Corporation",
  "FlashPix:Author": "Microsoft Office",
  "FlashPix:HyperlinksChanged": 0,
  "FlashPix:Characters": 0,
  "FlashPix:Security": 0,
  "FlashPix:TotalEditTime": 0,
  "File:FileModifyDate": "2015:12:29 09:41:56-08:00",
  "FlashPix:Words": 0,
  "FlashPix:TitleOfParts": "",
  "FlashPix:CompObjUserTypeLen": 31,
  "File:FilePermissions": 600,
  "FlashPix:Subject": "",
  "FlashPix:CompObjUserType": "???????? Microsoft Office Word",
  "SourceFile": "/dev/shm/laika_tmp/tmpNogtaW",
  "FlashPix:ScaleCrop": 0,
  "File:MIMEType": "application/msword",
  "FlashPix:Comments": "",
  "File:FileAccessDate": "2015:12:29 09:41:56-08:00",
  "File:FileSize": 93696,
  "FlashPix:Template": "Normal.dot",
  "FlashPix:AppVersion": 11.9999,
  "FlashPix:Paragraphs": 1,
  "FlashPix:Lines": 1,
```

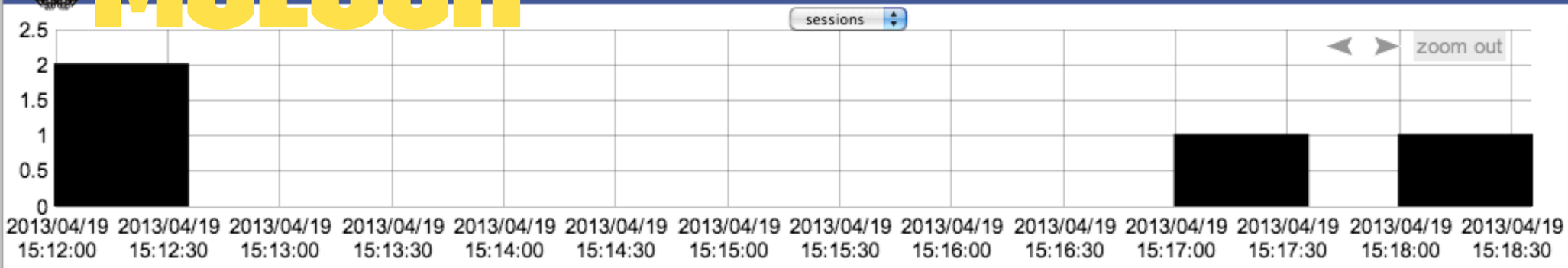
Sessions | SPI View | Connections | Files | Users | Stats | Change Password

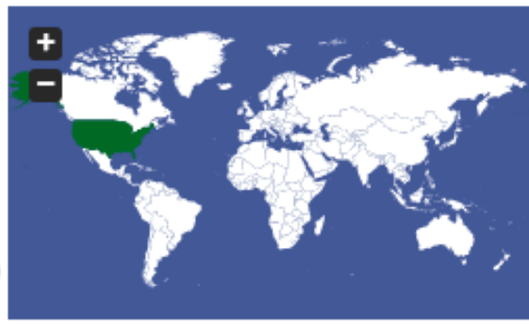
moloch v0.8.0

MOLOCH

ip[illegible]6.60 [illegible] == github.com

Search | Export





100 | Showing 1 to 4 of 4 entries (filtered from 1,887,556,382 total entries) | First | Previous | 1 | Next | Last

	Start	Stop	Src IP	Src Port	Dst IP	Dst Port	Packets	Bytes	Node	Info
udp	2013/04/19 15:12:29	2013/04/19 15:12:29	10.66.66.60	52136	8.8.8.8 USA	53	2	528 / 544	[redacted]	
tcp	2013/04/19 15:12:29	2013/04/19 15:12:29	10.66.66.60	58093	204.232.175.90 USA	80	11	1,001 / 3,896	[redacted]	//github.com/aol/moloch

Actions | Download Pcap | Source Raw | Destination Raw | Permalink

Start: 2013/04/19 15:12:29 Stop: 2013/04/19 15:12:29 Node: [redacted] IP Protocol: tcp

Src IP/Port: 10.66.66.60:58093 Dst IP/Port: 204.232.175.90:80 (USA) [AS27357 Rackspace Hosting]

Tags: [redacted] http:content:text/html, http:method:GET, http:statuscode:301, [redacted] protocol:http, tcp

HTTP

Hosts: github.com

User Agents: Mozilla/5.0 (iPhone; CPU iPhone OS 5_1_1 like Mac OS X) AppleWebKit/534.46 (KHTML, like Gecko) Version/5.1 Mobile/9B206 Safari/7534.48.3

Request Headers: accept, accept-encoding, accept-language, connection, cookie, host, user-agent

Response Headers: connection, content-length, content-type, date, location, server, vary

○natural ◉ascii ○utf8 ○hex ☐Line Numbers | ☐Decode GZip | ☐Show Images & Files | ☐Show Timestamps

Source

```
GET /aol/moloch HTTP/1.1
Host: github.com
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 5_1_1 like Mac OS X)
 AppleWebKit/534.46 (KHTML, like Gecko) Version/5.1 Mobile/9B206 Sa
fari/7534.48.3
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q
=0.8
Accept-Language: en-us
Accept-Encoding: gzip, deflate
```

Destination

Destination Bytes:

"WAIT ... I HAVE TO DEPLOY ALL OF THESE TOOLS TO DO NETWORK HUNTING?"

NO!

Look for analogous metadata in logs you already collect

Bro Conn == Firewall, router, NetFlow
Bro HTTP == Web proxy, IIS
Bro DNS == DNS logs
Bro SSH == sshd

BUT ...

Aforementioned tools are extensible

Bro
- Bro scripting language (network metadata)
- BinPac (protocol parsing)

Laika BOSS
- Python (file parsing, extraction)

Easy to turn ideas into production-ready capabilities

THIS PART IS IMPORTANT!

Metadata needs to be centralized and organized

Centralized
- Make it accessible from one location
- SIEM, Splunk / ELK, file server ... wherever

Organized
- Label related groups (systems, sites)
- Keep track of systems of interest
- Becomes critical as scale increases

HUNTING TECHNIQUES

Stacking
- Simple or complex outlier analysis
- Useful for identifying anomalies

Tracking
- Use inside knowledge to track attackers

Visualizing
- Utilize tools to visualize data
- Identifies links of activity that may not be apparent when performing "line-based analysis"

WHAT SHOULD I HUNT?

ARE YOU READY TO STACK?

Problem

- An unidentified system on the network is beaconing via HTTP to an attacker controlled server

- Anti-virus prevention failed, beaconing did not trigger any IDS signatures, and the attack server has never been seen before

- Can we find this system?

STACKING++

Stacking HTTP metadata may help identify this host and the attack server by looking for anomalous HTTP connections

Useful http.log metadata

- HTTP host header value
- HTTP User-Agent header value
- Lack of specific metadata (e.g., no referrer, no User-Agent)

STACKING++

The scale of the network metadata will affect how effective this is

Let's look at a real-world dataset!

STACKING++

24hr period on one network sensor:
1,255 unique source IP addresses
connected to 4,757 unique HTTP hosts

24hr period on 20 network sensors:
38,796 unique source IP addresses
connected to 54,014 unique HTTP hosts

STACKING++

Effectiveness can be increased with aggressive pre-analysis filtering
- Filter by direction (inbound, outbound, internal)
- Filter out known-good servers and services
- Filter for critical systems

Let's filter the previous dataset and focus on domain controllers connecting outbound via HTTP

STACKING++

24hr period on one network sensor:
2 unique source IP addresses connected to
2 unique HTTP hosts

24 period on 20 network sensors:
20 unique source IP addresses connected
to 8 unique HTTP hosts

Focusing our search increases the chance
of finding something interesting!

STACKING++

Many fields can be stacked, but I like ...

dns.log
- query

rdp.log
- cookie
- keyboard_layout

ssl.log
- server_name

LET'S TALK ABOUT TRACKING

Tracking attackers is a more effective (and more difficult) approach

You should consider at least one of two things

- What data the attacker might be after
- How the attacker might achieve their goals

* http://sroberts.github.io/2015/04/14/ir-is-dead-long-live-ir/

TRACKING++

How do we track attackers trying to achieve their goals?

Primarily focused on hunting artifacts left by their tools and tactics, techniques, and procedures (TTPs)

Utilizing threat intelligence and incident notes can increase effectiveness
- Note: threat intelligence, not one-off indicators!

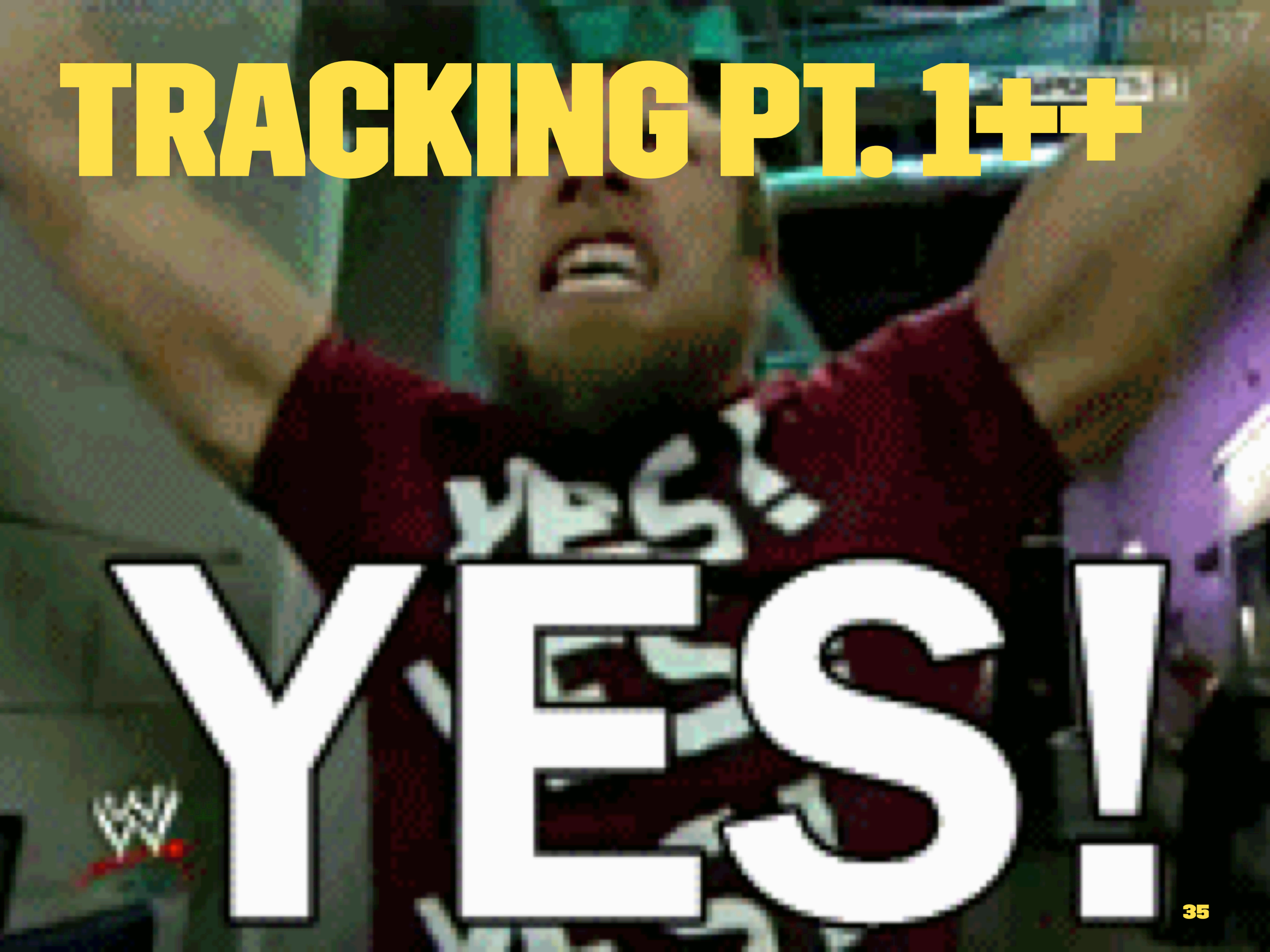
TRACKING BAD GUYS, PT. 1

Problem

- Smart attackers try to protect their infrastructure

- They mask their origins by utilizing VPN services and VPS providers

- Can we track attackers by watching for these services and providers?


TRACKING PT. 1++
YES!
YES!

TRACKING PT. 1++

Not trivial, but possible

Requires knowledge of attacker using service / provider

TRACKED_PROVIDERS.BRO

Available at https://github.com/CrowdStrike/cs-bro

- Accepts lists of VPN / VPS IP addresses and subnets via file input
- If service or provider is seen on network, then tracked_providers.log is written

Note: choosing which VPN / VPS to track is up to you!

TRACKED_PROVIDERS.BRO++

"How does this differ from a traditional IDS IP blacklist?"

"How does this differ from IP addresses I receive in my #threatintel #indicator feed?"

TRACKED_PROVIDERS.BRO++

IP addresses in blacklists and indicator lists are (or were) known-bad

Doesn't focus on one server, treats them all as suspects of interest

Hint hint, you could do this with Python as well

TRACKED_PROVIDERS.BRO++

Usefulness of Bro really shines here

Immediately gain context on what the server is doing w/o need for PCAP

Scanning?
- Correlate with Scan:: alerts

Webshell access?
- Correlate with http.log or ssl.log

Exfiltration?
- Correlate with conn.log

TRACKING BAD GUYS, PT. 2

Lateral movement: methods an attacker performs to move throughout the network to reach their target

Hunting lateral movement is something that only seems achievable via endpoint data

TRACKING BAD GUYS, PT. 2

Lateral movement: methods an attacker performs to move throughout the network to reach their target

Hunting lateral movement is something that only seems achievable via endpoint data

... but is it?

TRACKING PT. 2++

Problem

- Network analysts tend to focus on hunting command and control and exfiltration of data

- Traditionally, network-based threat detection appliances are placed at the borders of a network

- What could we find if we monitored internal traffic between critical business sites and VPN nodes?

TRACKING PT. 2++

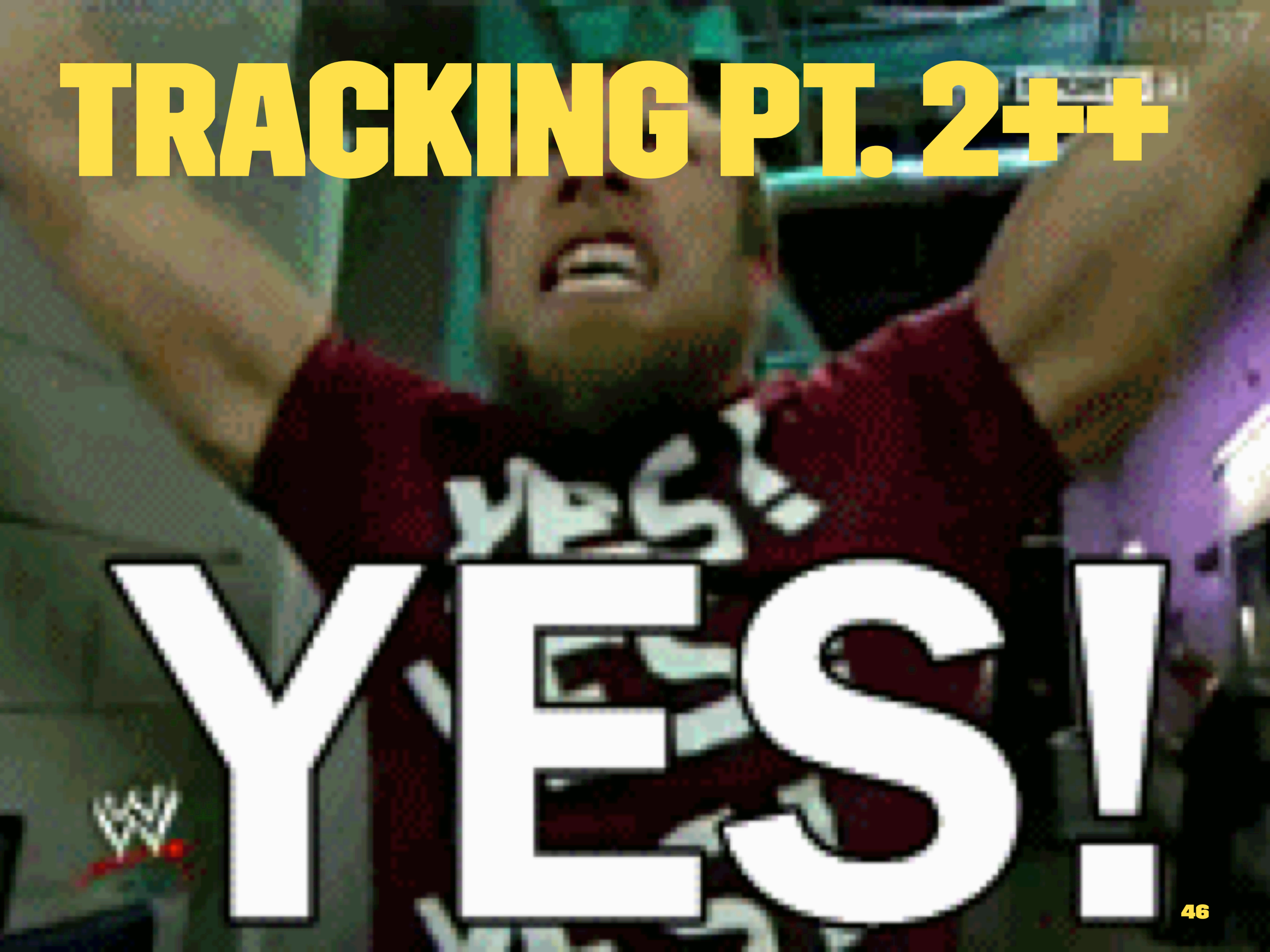
Think about tools and network services that attackers typically use
- Remote desktop protocol (RDP)
- File shares (SMB)
- AT jobs / scheduled tasks (SMB and DCE-RPC)
- Windows Management Instrumentation (DCE-RPC)

TRACKING PT. 2++

Think about tools and network services that attackers typically use
- Remote desktop protocol (RDP)
- File shares (SMB)
- AT jobs / scheduled tasks (SMB and DCE-RPC)
- Windows Management Instrumentation (DCE-RPC)

Can we find these artifacts in network traffic and collect them?

TRACKING PT. 2++



BRO + RDP

RDP protocol analyzer
- Captures metadata from RDP sessions pre-encryption
- Contains enough metadata to successfully hunt suspicious sessions
- Included by default as of Bro 2.4

BRO + SMB

SMB protocol analyzer
- Captures metadata from SMB transactions
- Quickly identify file shares and AT jobs
- Analyzer is not stable in production and current development is frozen

BRO + DCE-RPC

DCE-RPC protocol analyzer
- Captures metadata from DCE-RPC connections
- Includes bind / interface UUID, operation numbers, stub data
- Wide range of possibilities, including identifying scheduled tasks and WMI
- Not ported to Bro 2.x ...

BRO + DCE-RPC

DCE-RPC protocol analyzer
- Captures metadata from DCE-RPC connections
- Includes bind / interface UUID, operation numbers, stub data
- Wide range of possibilities, including identifying scheduled tasks and WMI
- Not ported to Bro 2.x ... just kidding!

DCE-RPC PROTOCOL ANALYZER

Analyzer code ships with each install of Bro 2.x, just not enabled

Requirements to get it working
- DCE-RPC payload signature to enable the analyzer
- dcerpc/main.bro file to handle logging the metadata

DCE-RPC PROTOCOL ANALYZER ++

Logs interface UUID, operation numbers, and length of stub data

Months of testing on production systems

Scheduled tasks and WMI can be found by hunting for interface UUIDs related to those services

DCE-RPC PROTOCOL ANALYZER ++

```
1438553222.724284       C9nGZaHppkW3AM101       172.18.20.76    51833   10.16.40.23     54771   12345678-1234-abcd-ef00
-01234567cffb    netlogon        4       REQUEST 150
1438553222.724805       C9nGZaHppkW3AM101       172.18.20.76    51833   10.16.40.23     54771   12345678-1234-abcd-ef00
-01234567cffb    netlogon        4       RESPONSE        12
1438553222.825566       C9nGZaHppkW3AM101       172.18.20.76    51833   10.16.40.23     54771   12345678-1234-abcd-ef00
-01234567cffb    netlogon        26      REQUEST 220
1438553222.827106       C9nGZaHppkW3AM101       172.18.20.76    51833   10.16.40.23     54771   12345678-1234-abcd-ef00
-01234567cffb    netlogon        26      RESPONSE        20
1438553222.933682       C9nGZaHppkW3AM101       172.18.20.76    51833   10.16.40.23     54771   12345678-1234-abcd-ef00
-01234567cffb    netlogon        21      REQUEST 184
1438553222.934240       C9nGZaHppkW3AM101       172.18.20.76    51833   10.16.40.23     54771   12345678-1234-abcd-ef00
-01234567cffb    netlogon        21      RESPONSE        40
1438553223.001264       C9nGZaHppkW3AM101       172.18.20.76    51833   10.16.40.23     54771   12345678-1234-abcd-ef00
-01234567cffb    netlogon        29      REQUEST 856
1438553223.002876       C9nGZaHppkW3AM101       172.18.20.76    51833   10.16.40.23     54771   12345678-1234-abcd-ef00
-01234567cffb    netlogon        29      RESPONSE        3384
```

DCE-RPC PROTOCOL ANALYZER ++

Available at https://github.com/CrowdStrike/cs-bro

Immediate todos
- Support for object UUIDs to better track connections

Longterm todos
- Connection-based logging
- Intelligent stub data extraction

ONE MORE THING ...

If you haven't looked at PCAP of a WMI connection ...

```
T 192.168.132.142:26387 -> 192.168.142.207:49154 [AP]
  ........ ................T..`..../R."..................f...D.lX..;.q....User............W.Q.L...UserH.......H...s.e.l
  .e.c.t. .N.a.m.e.,.C.S.D.V.e.r.s.i.o.n.,.T.o.t.a.l.V.i.r.t.u.a.l.M.e.m.o.r.y.S.i.z.e. .f.r.o.m. .W.i.n.3.2._.O.p.e.r.
  a.t.i.n.g.S.y.s.t.e.m.........................e......
```

ONE MORE THING ...

If you haven't looked at PCAP of a WMI connection ...

```
T 192.168.132.142:26387 -> 192.168.142.207:49154 [AP]
  ........ ................T..`..../R."..................f...D.lX..;.q....User............W.Q.L...UserH.......H...s.e.l
  .e.c.t. .N.a.m.e.,.C.S.D.V.e.r.s.i.o.n.,.T.o.t.a.l.V.i.r.t.u.a.l.M.e.m.o.r.y.S.i.z.e. .f.r.o.m. .W.i.n.3.2._.O.p.e.r.
  a.t.i.n.g.S.y.s.t.e.m.........................e......
```

WMI analyzer, anyone?

TLDR?

Work with the data you have, consider new tools

Centralize and organize your data

Look for opportunities to meaningfully increase visibility

Focus on post-exploitation attacker activity

View your network like an attacker would - "How would I do X?"