CHIRON TRAINING

Arming Small Security Programs

Troopers17

Arming Small Security Programs

OVERVIEW

Network baseline generation with Bropy

Disclaimer

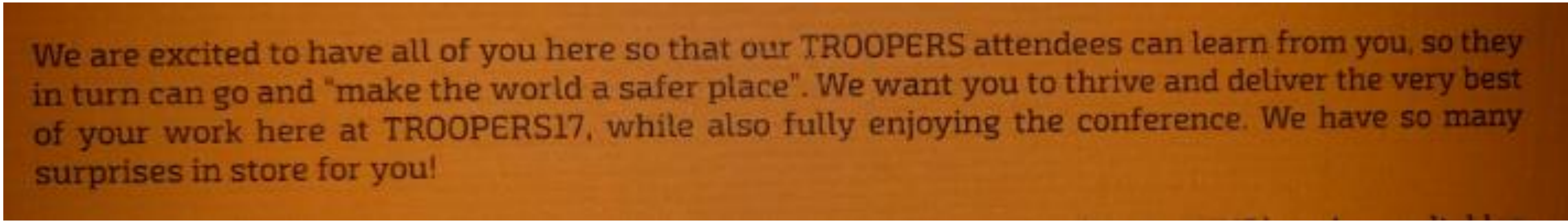
- I “borrowed” my employers slide template
  - – Creating .POT files is hard
- This is NOT my employers material
- TLDR; You can sue me, not my employer

About Me

Matt Domko

- Beard Enthusiast
- Giessen American High School
- Former:
  - Parachutist
  - Enterprise Admin
  - “Cyber Network Defender”
- Instructor at Chiron Technology Services
- Started blogging about Blue Team stuff
  http://goo.gl/uznCag
- Brakesec Slack
  https://brakesec.signup.team
- @hashtagcyber

Why I’m here



We are excited to have all of you here so that our TROOPERS attendees can learn from you, so they in turn can go and "make the world a safer place". We want you to thrive and deliver the very best of your work here at TROOPERS17, while also fully enjoying the conference. We have so many surprises in store for you!

“Make the world a safer place”
{by sharing information}

Get Out!

Leave now if:

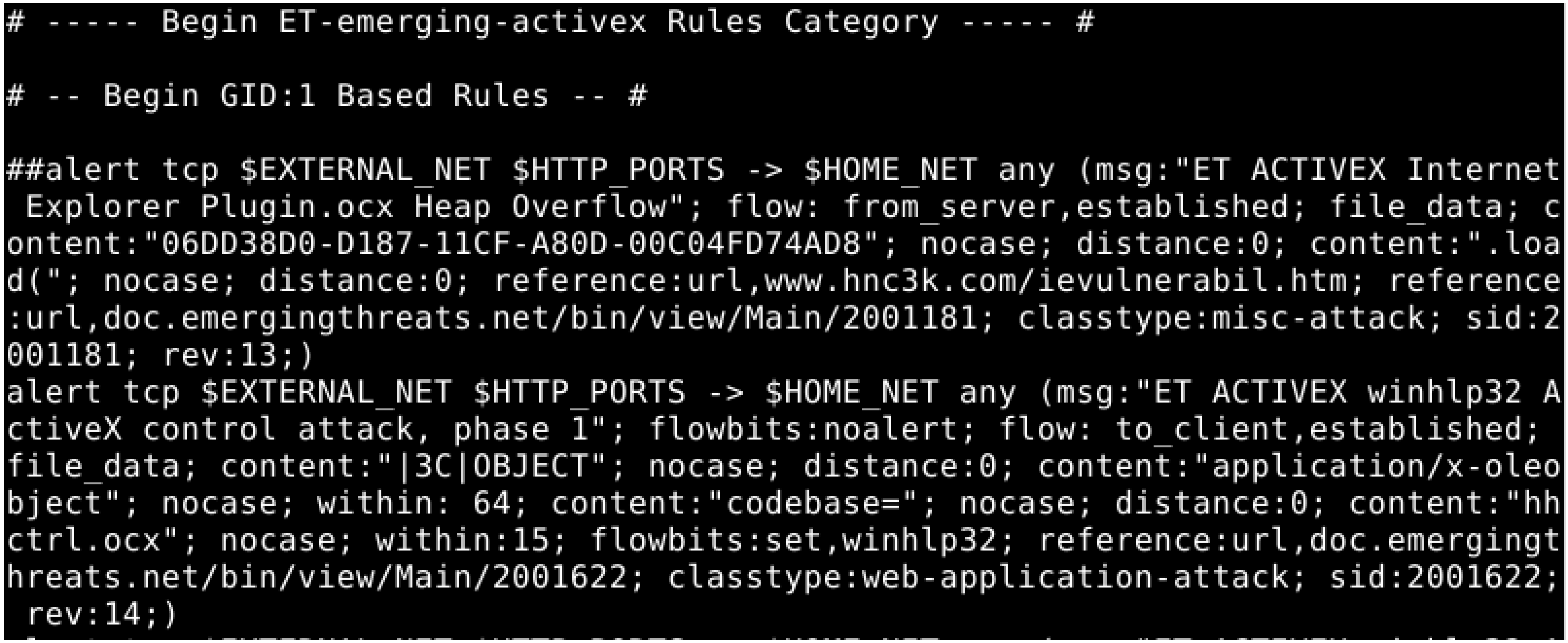
- You were looking for a red team talk
  - Joffrey Czarny is talking about pentesting Citrix in one room
  - Rebecca Shapiro is breaking down bootloaders in the other

Please stay if:

- You want to know every host your critical assets communicate with
- You want a list of every port a server listens on
- You want to do it all in less than 5 mins

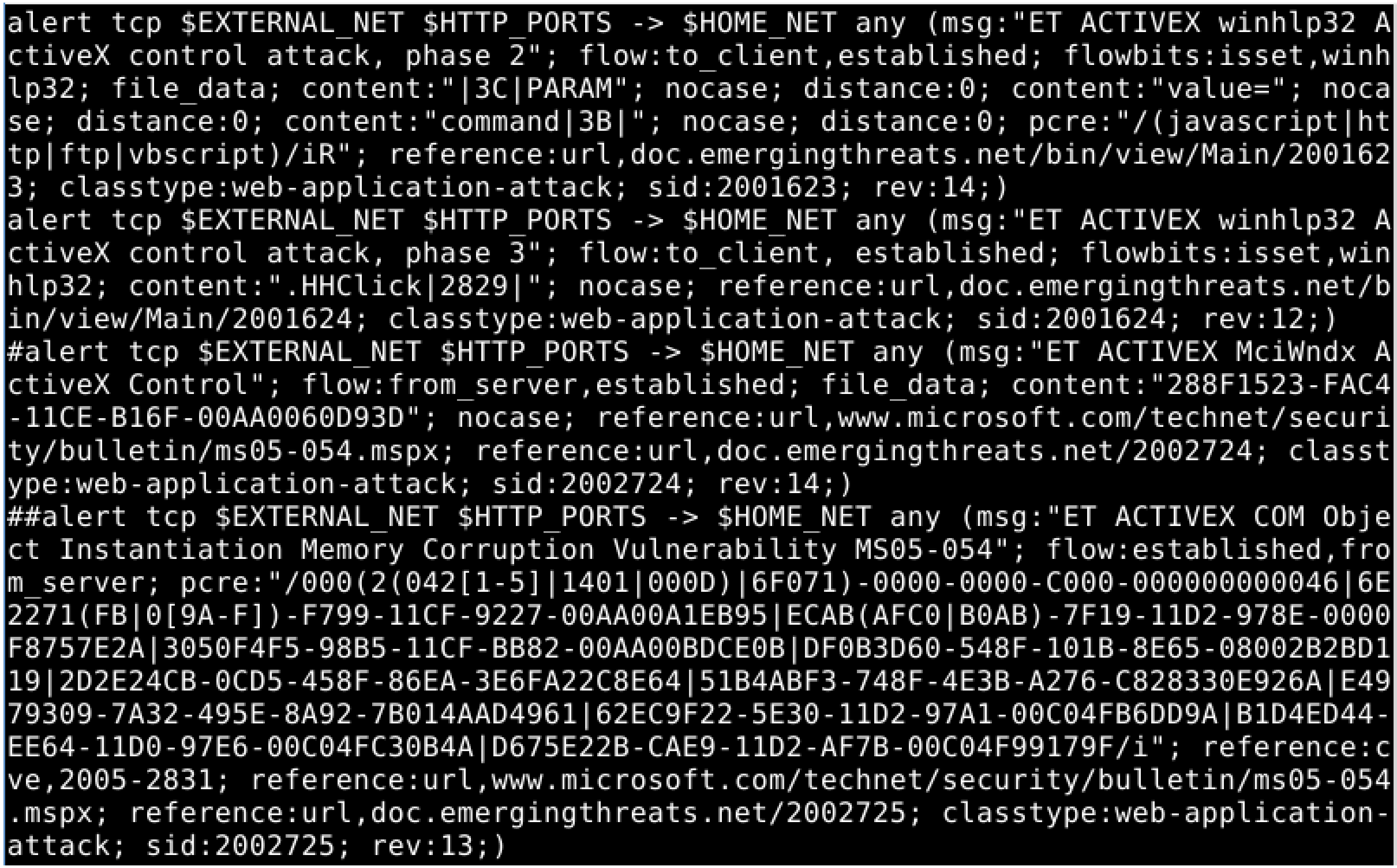
The Problem: Detecting Malicious Network Activity

- Malicious network activity CAN be identified using signatures…



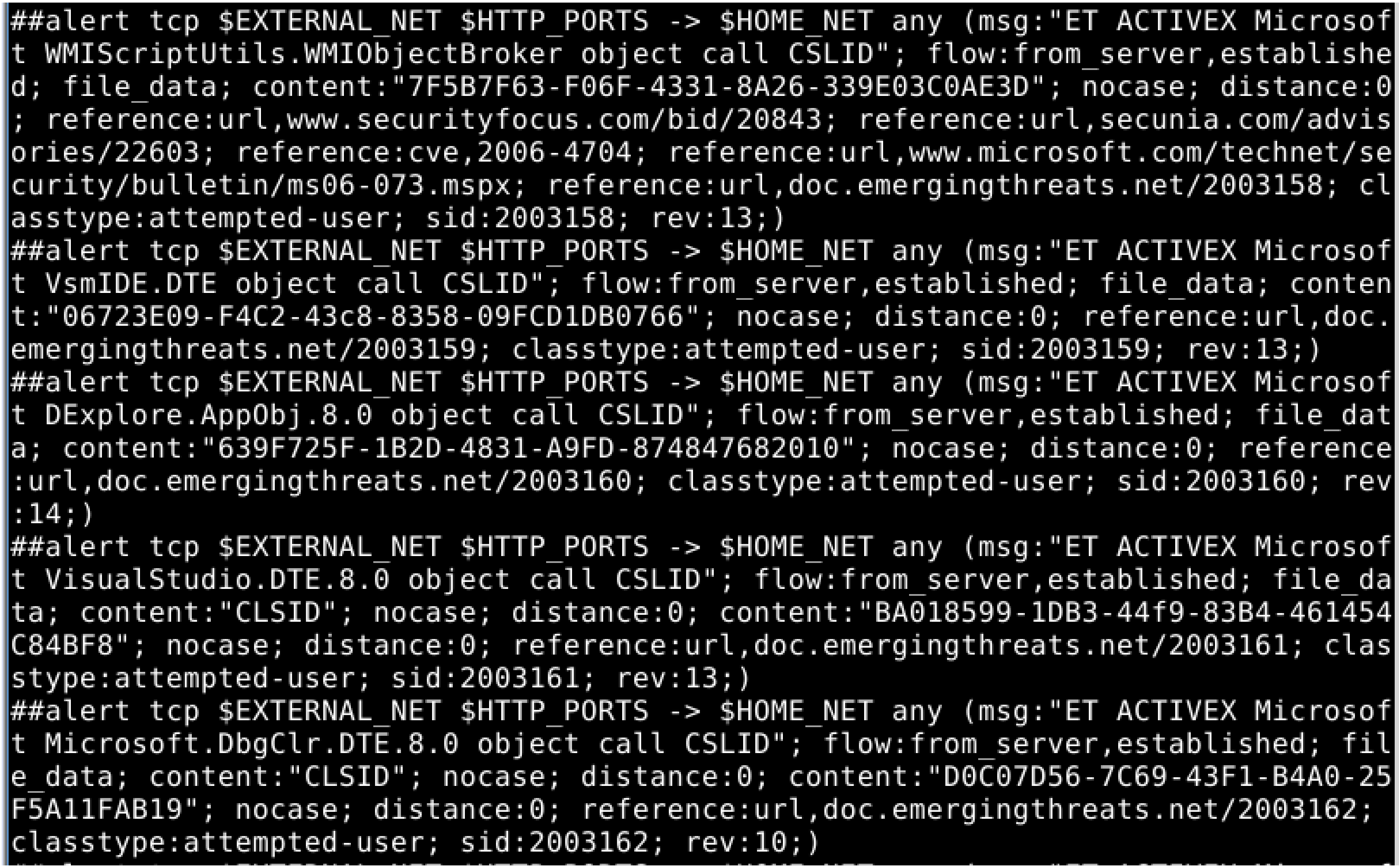
```
# ----- Begin ET-emerging-activex Rules Category ----- #

# -- Begin GID:1 Based Rules -- #

##alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Internet
 Explorer Plugin.ocx Heap Overflow"; flow: from_server,established; file_data; c
ontent:"06DD38D0-D187-11CF-A80D-00C04FD74AD8"; nocase; distance:0; content:".loa
d("; nocase; distance:0; reference:url,www.hnc3k.com/ievulnerabil.htm; reference
:url,doc.emergingthreats.net/bin/view/Main/2001181; classtype:misc-attack; sid:2
001181; rev:13;)
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX winhlp32 A
ctiveX control attack, phase 1"; flowbits:noalert; flow: to_client,established;
file_data; content:"|3C|OBJECT"; nocase; distance:0; content:"application/x-oleo
bject"; nocase; within: 64; content:"codebase="; nocase; distance:0; content:"hh
ctrl.ocx"; nocase; within:15; flowbits:set,winhlp32; reference:url,doc.emergingt
hreats.net/bin/view/Main/2001622; classtype:web-application-attack; sid:2001622;
 rev:14;)
```

The Problem: Detecting Malicious Network Activity



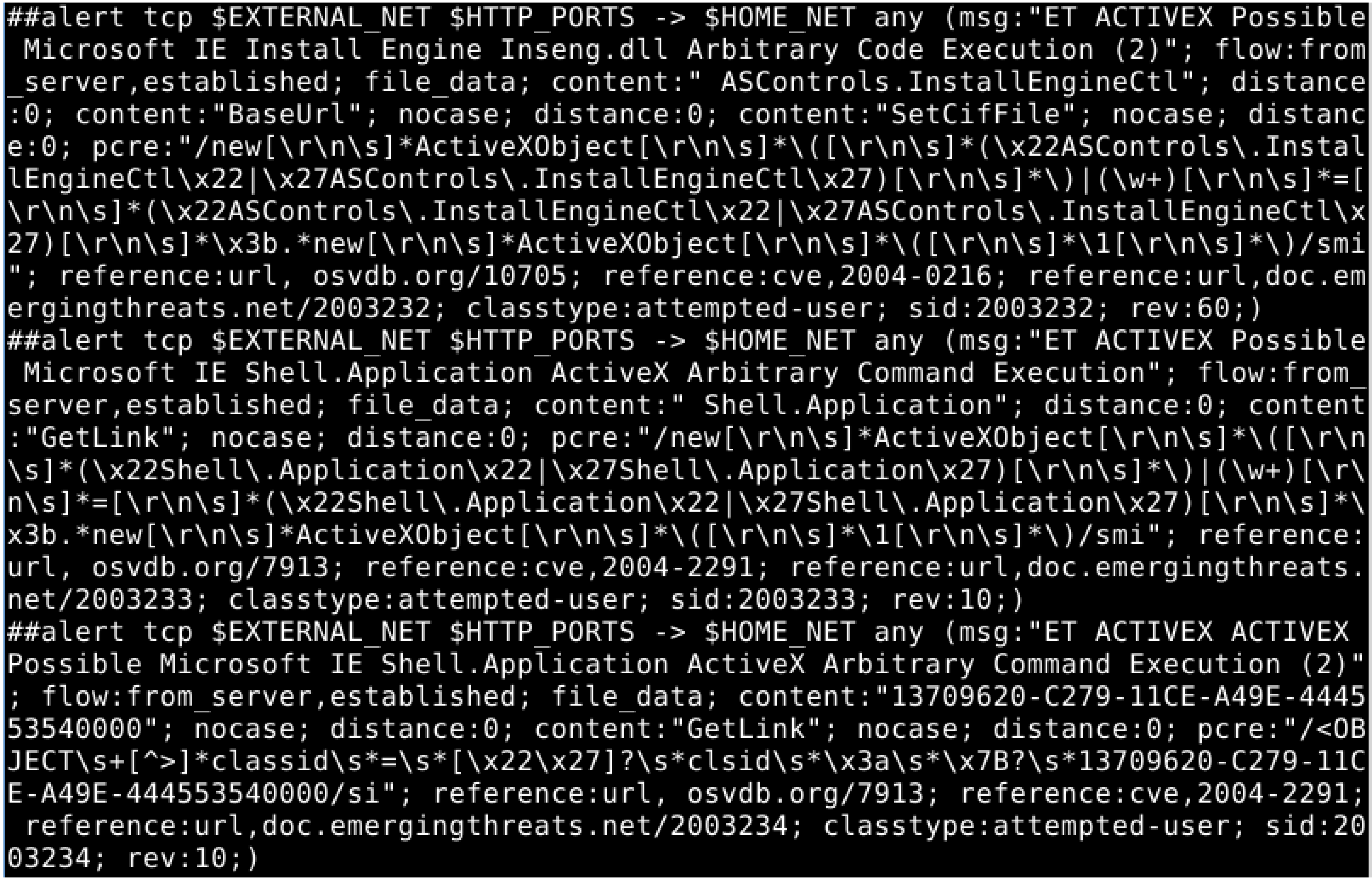
```
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX winhlp32 A
ctiveX control attack, phase 2"; flow:to_client,established; flowbits:isset,winh
lp32; file_data; content:"|3C|PARAM"; nocase; distance:0; content:"value="; noca
se; distance:0; content:"command|3B|"; nocase; distance:0; pcre:"/(javascript|ht
tp|ftp|vbscript)/iR"; reference:url,doc.emergingthreats.net/bin/view/Main/200162
3; classtype:web-application-attack; sid:2001623; rev:14;)
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX winhlp32 A
ctiveX control attack, phase 3"; flow:to_client, established; flowbits:isset,win
hlp32; content:".HHClick|2829|"; nocase; reference:url,doc.emergingthreats.net/b
in/view/Main/2001624; classtype:web-application-attack; sid:2001624; rev:12;)
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX MciWndx A
ctiveX Control"; flow:from_server,established; file_data; content:"288F1523-FAC4
-11CE-B16F-00AA0060D93D"; nocase; reference:url,www.microsoft.com/technet/securi
ty/bulletin/ms05-054.mspx; reference:url,doc.emergingthreats.net/2002724; classt
ype:web-application-attack; sid:2002724; rev:14;)
##alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX COM Obje
ct Instantiation Memory Corruption Vulnerability MS05-054"; flow:established,fro
m_server; pcre:"/000(2(042[1-5]|1401|000D)|6F071)-0000-0000-C000-000000000046|6E
2271(FB|0[9A-F])-F799-11CF-9227-00AA00A1EB95|ECAB(AFC0|B0AB)-7F19-11D2-978E-0000
F8757E2A|3050F4F5-98B5-11CF-BB82-00AA00BDCE0B|DF0B3D60-548F-101B-8E65-08002B2BD1
19|2D2E24CB-0CD5-458F-86EA-3E6FA22C8E64|51B4ABF3-748F-4E3B-A276-C828330E926A|E49
79309-7A32-495E-8A92-7B014AAD4961|62EC9F22-5E30-11D2-97A1-00C04FB6DD9A|B1D4ED44-
EE64-11D0-97E6-00C04FC30B4A|D675E22B-CAE9-11D2-AF7B-00C04F99179F/i"; reference:c
ve,2005-2831; reference:url,www.microsoft.com/technet/security/bulletin/ms05-054
.mspx; reference:url,doc.emergingthreats.net/2002725; classtype:web-application-
attack; sid:2002725; rev:13;)
```

The Problem: Detecting Malicious Network Activity



```
##alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Microsof
t WMIScriptUtils.WMIObjectBroker object call CSLID"; flow:from_server,establishe
d; file_data; content:"7F5B7F63-F06F-4331-8A26-339E03C0AE3D"; nocase; distance:0
; reference:url,www.securityfocus.com/bid/20843; reference:url,secunia.com/advis
ories/22603; reference:cve,2006-4704; reference:url,www.microsoft.com/technet/se
curity/bulletin/ms06-073.mspx; reference:url,doc.emergingthreats.net/2003158; cl
asstype:attempted-user; sid:2003158; rev:13;)
##alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Microsof
t VsmIDE.DTE object call CSLID"; flow:from_server,established; file_data; conten
t:"06723E09-F4C2-43c8-8358-09FCD1DB0766"; nocase; distance:0; reference:url,doc.
emergingthreats.net/2003159; classtype:attempted-user; sid:2003159; rev:13;)
##alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Microsof
t DExplore.AppObj.8.0 object call CSLID"; flow:from_server,established; file_dat
a; content:"639F725F-1B2D-4831-A9FD-874847682010"; nocase; distance:0; reference
:url,doc.emergingthreats.net/2003160; classtype:attempted-user; sid:2003160; rev
:14;)
##alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Microsof
t VisualStudio.DTE.8.0 object call CSLID"; flow:from_server,established; file_da
ta; content:"CLSID"; nocase; distance:0; content:"BA018599-1DB3-44f9-83B4-461454
C84BF8"; nocase; distance:0; reference:url,doc.emergingthreats.net/2003161; clas
stype:attempted-user; sid:2003161; rev:13;)
##alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Microsof
t Microsoft.DbgClr.DTE.8.0 object call CSLID"; flow:from_server,established; fil
e_data; content:"CLSID"; nocase; distance:0; content:"D0C07D56-7C69-43F1-B4A0-25
F5A11FAB19"; nocase; distance:0; reference:url,doc.emergingthreats.net/2003162;
classtype:attempted-user; sid:2003162; rev:10;)
```

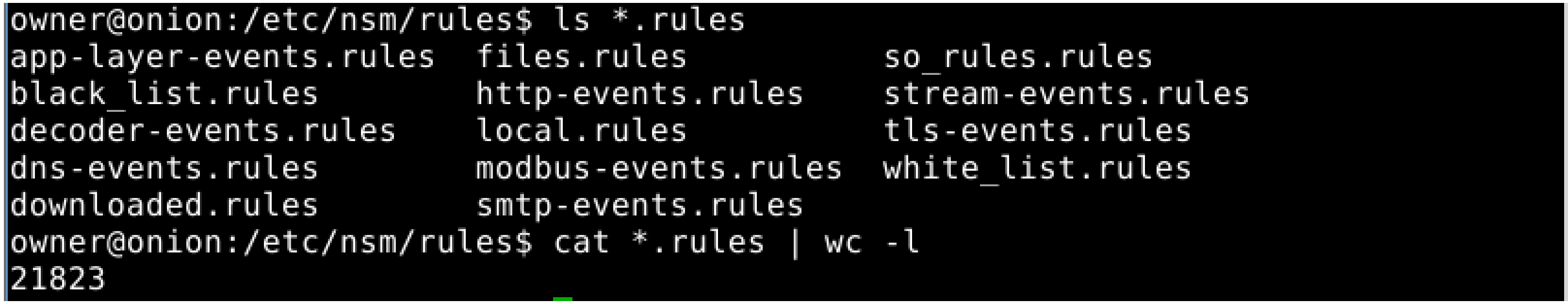
The Problem: Detecting Malicious Network Activity



```
##alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Possible
 Microsoft IE Install Engine Inseng.dll Arbitrary Code Execution (2)"; flow:from
_server,established; file_data; content:" ASControls.InstallEngineCtl"; distance
:0; content:"BaseUrl"; nocase; distance:0; content:"SetCifFile"; nocase; distanc
e:0; pcre:"/new[\r\n\s]*ActiveXObject[\r\n\s]*\([\r\n\s]*(\x22ASControls\.Instal
lEngineCtl\x22|\x27ASControls\.InstallEngineCtl\x27)[\r\n\s]*\)|(\w+)[\r\n\s]*=[
\r\n\s]*(\x22ASControls\.InstallEngineCtl\x22|\x27ASControls\.InstallEngineCtl\x
27)[\r\n\s]*\x3b.*new[\r\n\s]*ActiveXObject[\r\n\s]*\([\r\n\s]*\1[\r\n\s]*\)/smi
"; reference:url, osvdb.org/10705; reference:cve,2004-0216; reference:url,doc.em
ergingthreats.net/2003232; classtype:attempted-user; sid:2003232; rev:60;)
##alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Possible
 Microsoft IE Shell.Application ActiveX Arbitrary Command Execution"; flow:from_
server,established; file_data; content:" Shell.Application"; distance:0; content
:"GetLink"; nocase; distance:0; pcre:"/new[\r\n\s]*ActiveXObject[\r\n\s]*\([\r\n
\s]*(\x22Shell\.Application\x22|\x27Shell\.Application\x27)[\r\n\s]*\)|(\w+)[\r\
n\s]*=[\r\n\s]*(\x22Shell\.Application\x22|\x27Shell\.Application\x27)[\r\n\s]*\
x3b.*new[\r\n\s]*ActiveXObject[\r\n\s]*\([\r\n\s]*\1[\r\n\s]*\)/smi"; reference:
url, osvdb.org/7913; reference:cve,2004-2291; reference:url,doc.emergingthreats.
net/2003233; classtype:attempted-user; sid:2003233; rev:10;)
##alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX ACTIVEX
Possible Microsoft IE Shell.Application ActiveX Arbitrary Command Execution (2)"
; flow:from_server,established; file_data; content:"13709620-C279-11CE-A49E-4445
53540000"; nocase; distance:0; content:"GetLink"; nocase; distance:0; pcre:"/<OB
JECT\s+[^>]*classid\s*=\s*[\x22\x27]?\s*clsid\s*\x3a\s*\x7B?\s*13709620-C279-11C
E-A49E-444553540000/si"; reference:url, osvdb.org/7913; reference:cve,2004-2291;
 reference:url,doc.emergingthreats.net/2003234; classtype:attempted-user; sid:20
03234; rev:10;)
```

The Problem: Detecting Malicious Network Activity

- And more signatures….

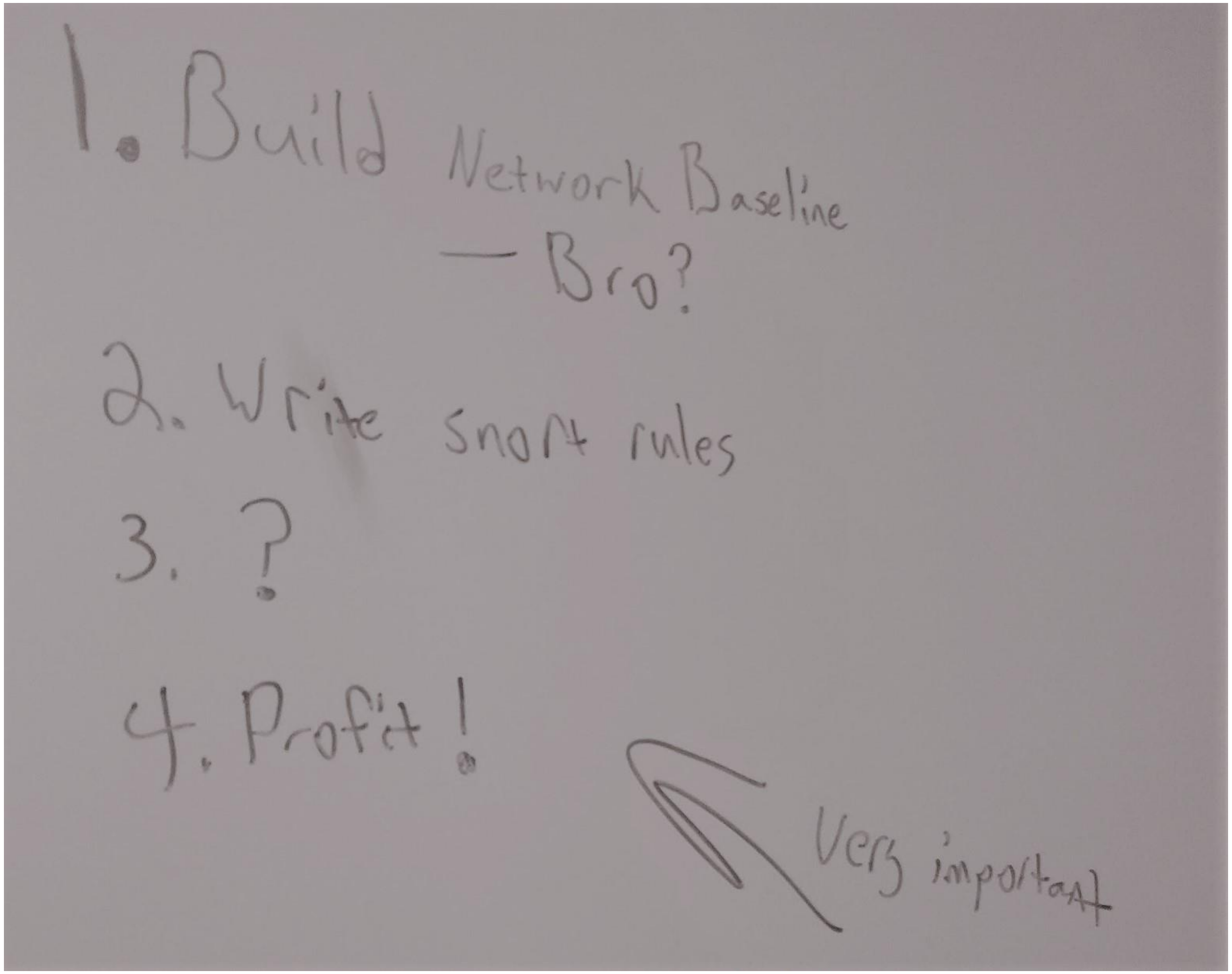


```
owner@onion:/etc/nsm/rules$ ls *.rules
app-layer-events.rules  files.rules          so_rules.rules
black_list.rules        http-events.rules    stream-events.rules
decoder-events.rules    local.rules          tls-events.rules
dns-events.rules        modbus-events.rules  white_list.rules
downloaded.rules        smtp-events.rules
owner@onion:/etc/nsm/rules$ cat *.rules | wc -l
21823
```

- Fingerprinting EVERY attack is impossible

- Signature based detection is USELESS if a signature does not exist for the attack being performed

The Initial Idea



1. Build Network Baseline
   — Bro?
2. Write snort rules
3. ?
4. Profit!

Very important

The Initial Idea

Step 1: Build a network baseline.

- Bro?
- Netflow?

Step 2: Write SNORT rules.

- I need alerts for non-standard traffic

Step 3: ?

- Something … Something … Something…

Step 4: Profit!

- Or at least spend less time worrying

A Similar Problem: Malicious Binaries

- Administrators face a similar problem with detecting malicious binaries.

- Antivirus products initially only used file signatures to identify malware:
  - Evil Hashes
  - Ego Strings
  - Reused code blocks

- This eventually failed, as attackers could easily modify malware to avoid signatures faster than they are generated
  - MSFVenom, Veil-Evasion, Hyperion

A Similar Problem: Malicious Binaries

- Heuristic detection helped, but does it catch everything?
  - – No. (Malware still exists/functions today)

- What else can we do?
  - – Enter Application Whitelisting

A Similar Problem: Malicious Binaries

- Application Whitelisting provides ability to:
  - Log execution of all files except explicitly authorized (whitelist):
    - File Hashes (tedious)
    - File Names (poor protection)
    - Signed Code (Awesome)
    - Source Directory (simple)
  - Prevent execution of files that are not in the whitelist.
  - Prevent execution of explicitly defined files (Blacklisting)

Simple Application Whitelisting Implementation

1. Start with an empty whitelist
2. Apply a policy to log everything not in whitelist
3. Use logs to generate a whitelist
4. Modify policy to block everything not in whitelist
5. Review new logs
   - Investigate blocked files
   - Update whitelist as needed

Malicious Network Activity : Anomaly Detection

- The same concept can be applied to network activity:
  - Start with an empty whitelist
  - Apply a policy to log all traffic not in the whitelist
  - Use logs to update the whitelist
  - Review new logs
    - Investigate new ports/hosts
    - Update whitelist as needed

But Matt, How Do I <do thing>?

- Get data for my whitelist?
  - Bro.
- Create a policy to log traffic?
  - Bro scripts
- Create logs from new traffic?
  - Bro scripts
- Review new logs?
  - ELSA

- Last question… Can you tell me more about Bro?

Disclaimer

- Robin Summer gave a MUCH BETTER presentation on Bro at TROOPERS14

https://www.youtube.com/watch?v=BBl0yaUdq4c

Gathering Data with Bro

- Bro in 30 Seconds
  - Much more than an IDS
  - Logs multiple layers of traffic
    - “Packet String”
    - Similar to NETFLOW
  - Plugins/Scripts
    - Interpret Data
    - Take action
  - Logs are small
    - Allows for longer retention than PCAP
  - Open Source, Built-in to Security Onion

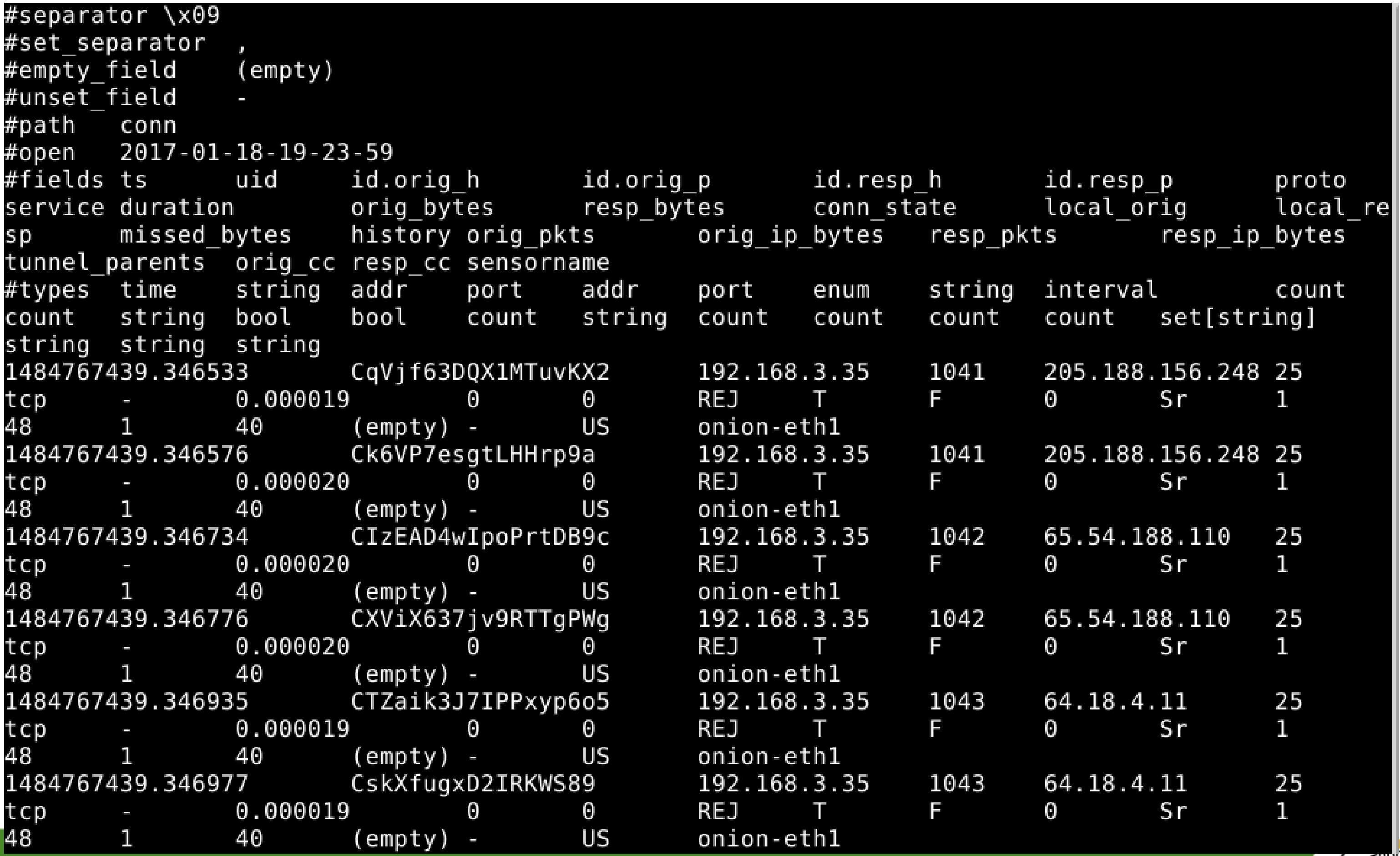
Gathering Data with Bro

- Bro Data Formatting
  - Tab Separated table
  - Headers at top
  - Common Fields:

Timestamp	(ts)
Connection ID	(uid)
IP Source	(id.orig_h)
Source Port	(id.orig_p)
IP Destination	(id.resp_h)
Dest Port	(id.resp_p)

Gathering Data with Bro

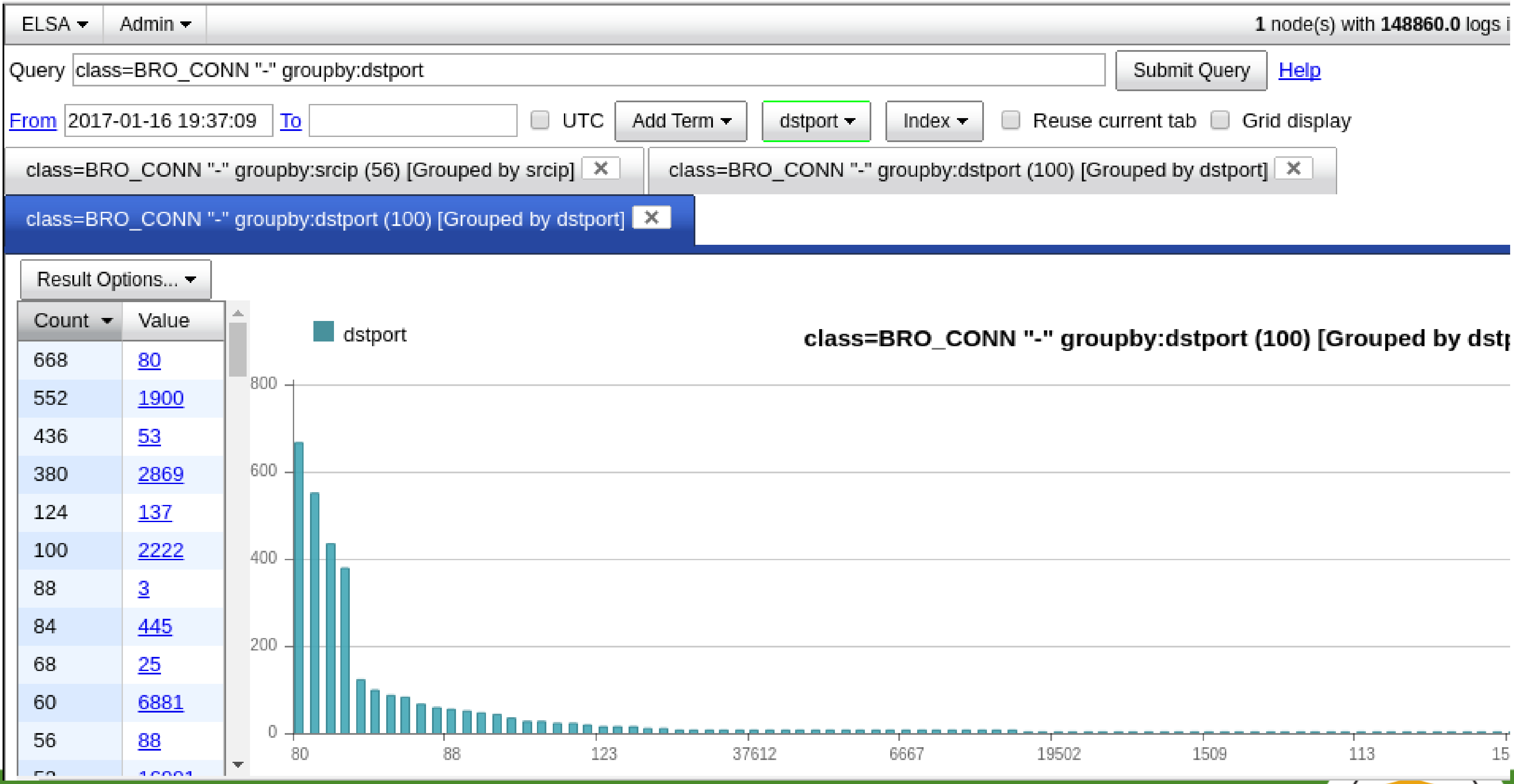
- Logs are simple to parse …. programmatically



```
#separator \x09
#set_separator  ,
#empty_field    (empty)
#unset_field    -
#path   conn
#open   2017-01-18-19-23-59
#fields ts      uid     id.orig_h       id.orig_p       id.resp_h       id.resp_p       proto
service duration        orig_bytes      resp_bytes      conn_state      local_orig      local_re
sp      missed_bytes    history orig_pkts       orig_ip_bytes   resp_pkts       resp_ip_bytes
tunnel_parents  orig_cc resp_cc sensorname
#types  time    string  addr    port    addr    port    enum    string  interval        count
count   string  bool    bool    count   string  count   count   count   count   set[string]
string  string  string
1484767439.346533       CqVjf63DQX1MTuvKX2      192.168.3.35    1041    205.188.156.248 25
tcp     -       0.000019        0       0       REJ     T       F       0       Sr      1
48      1       40      (empty) -       US      onion-eth1
1484767439.346576       Ck6VP7esgtLHHrp9a       192.168.3.35    1041    205.188.156.248 25
tcp     -       0.000020        0       0       REJ     T       F       0       Sr      1
48      1       40      (empty) -       US      onion-eth1
1484767439.346734       CIzEAD4wIpoPrtDB9c      192.168.3.35    1042    65.54.188.110   25
tcp     -       0.000020        0       0       REJ     T       F       0       Sr      1
48      1       40      (empty) -       US      onion-eth1
1484767439.346776       CXViX637jv9RTTgPWg      192.168.3.35    1042    65.54.188.110   25
tcp     -       0.000020        0       0       REJ     T       F       0       Sr      1
48      1       40      (empty) -       US      onion-eth1
1484767439.346935       CTZaik3J7IPPxyp6o5      192.168.3.35    1043    64.18.4.11      25
tcp     -       0.000019        0       0       REJ     T       F       0       Sr      1
48      1       40      (empty) -       US      onion-eth1
1484767439.346977       CskXfugxD2IRKWS89       192.168.3.35    1043    64.18.4.11      25
tcp     -       0.000019        0       0       REJ     T       F       0       Sr      1
48      1       40      (empty) -       US      onion-eth1
```

Gathering Data with Bro

- Humans should use ELSA, Splunk, etc…



ELSA ▾ Admin ▾ 1 node(s) with 148860.0 logs i

Query class=BRO_CONN "-" groupby:dstport Submit Query Help

From 2017-01-16 19:37:09 To ☐ UTC Add Term ▾ dstport ▾ Index ▾ ☐ Reuse current tab ☐ Grid display

class=BRO_CONN "-" groupby:srcip (56) [Grouped by srcip] ✕ class=BRO_CONN "-" groupby:dstport (100) [Grouped by dstport] ✕

class=BRO_CONN "-" groupby:dstport (100) [Grouped by dstport] ✕

Result Options... ▾

Count ▾	Value
668	80
552	1900
436	53
380	2869
124	137
100	2222
88	3
84	445
68	25
60	6881
56	88
52	16001

class=BRO_CONN "-" groupby:dstport (100) [Grouped by dstp

Gathering Data with Bro

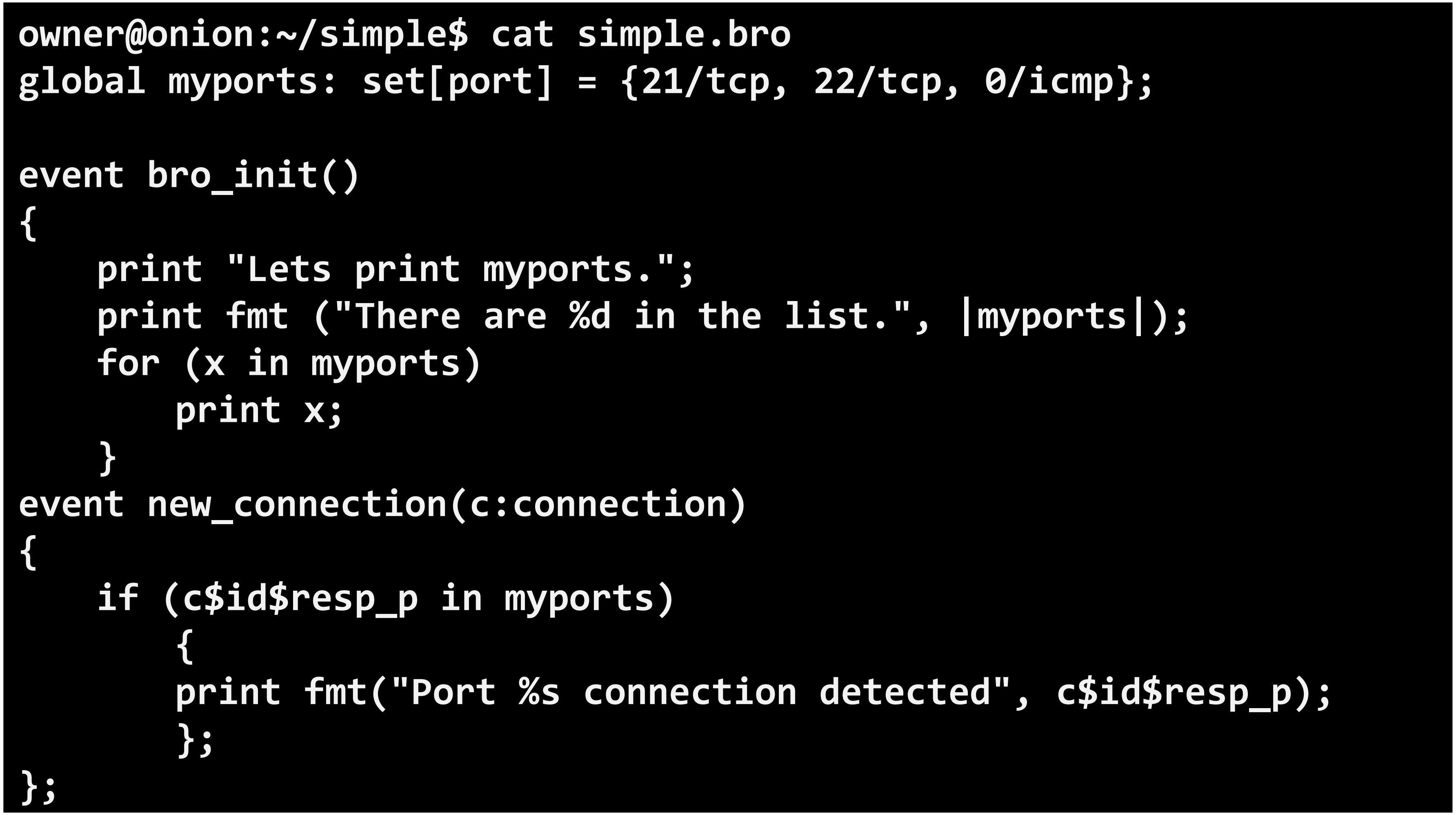
- Key Directories:
  - /nsm/bro/logs/current
    - notice.log
    - conn.log
    - weird.log
  - /opt/bro/share/bro/policy
    - Contains scripts loaded by Bro
  - /opt/bro/share/bro/site/local.bro
    - Add path to custom scripts to this file to load when bro starts

Bro Scripts

“The best way to learn to write Bro scripts, is to write Bro scripts”

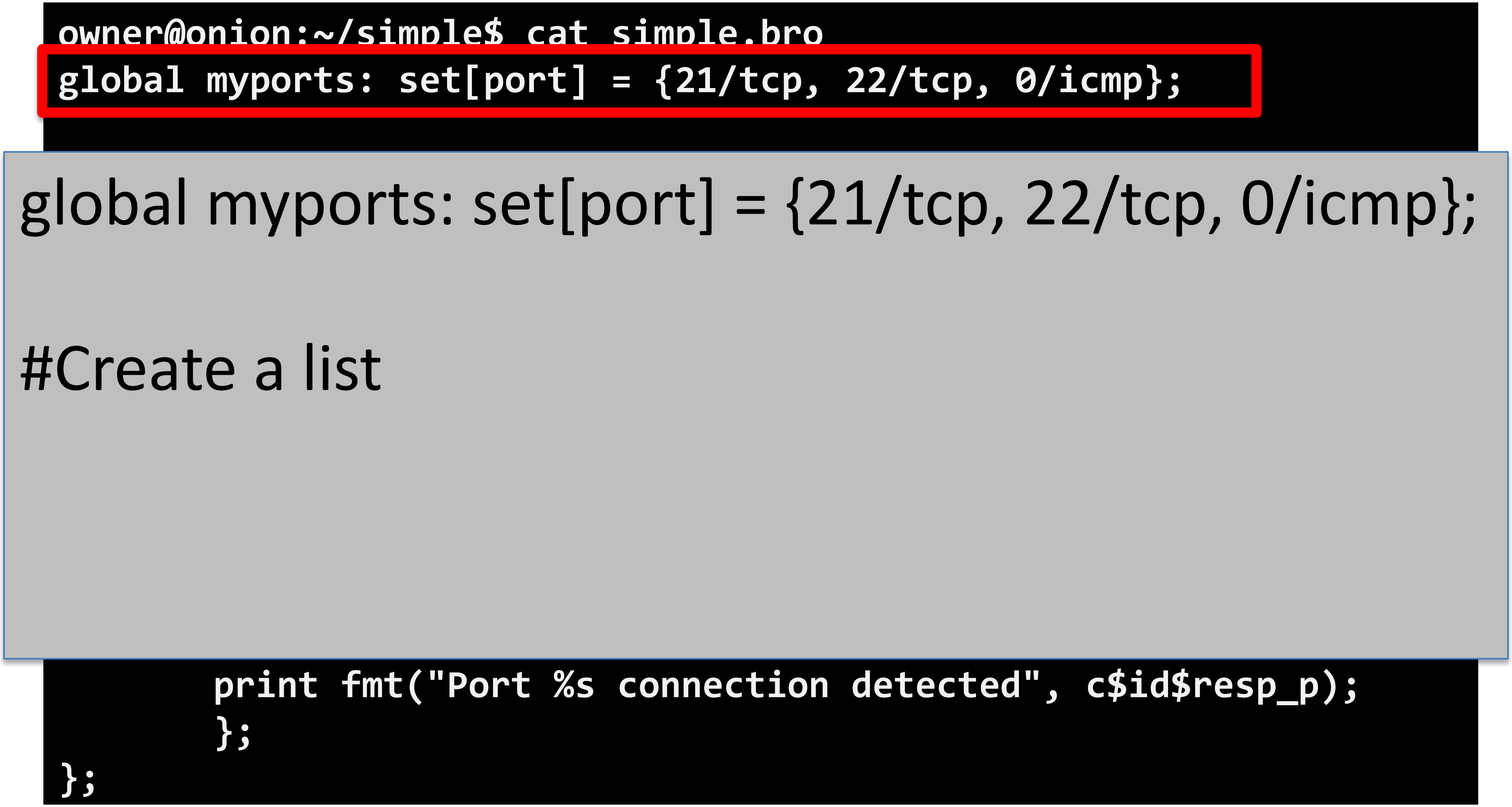
– Seth Hall, SecurityOnion Conference 2015

Bro Scripts



```
owner@onion:~/simple$ cat simple.bro
global myports: set[port] = {21/tcp, 22/tcp, 0/icmp};

event bro_init()
{
    print "Lets print myports.";
    print fmt ("There are %d in the list.", |myports|);
    for (x in myports)
        print x;
    }
event new_connection(c:connection)
{
    if (c$id$resp_p in myports)
        {
        print fmt("Port %s connection detected", c$id$resp_p);
        };
};
```

Bro Scripts



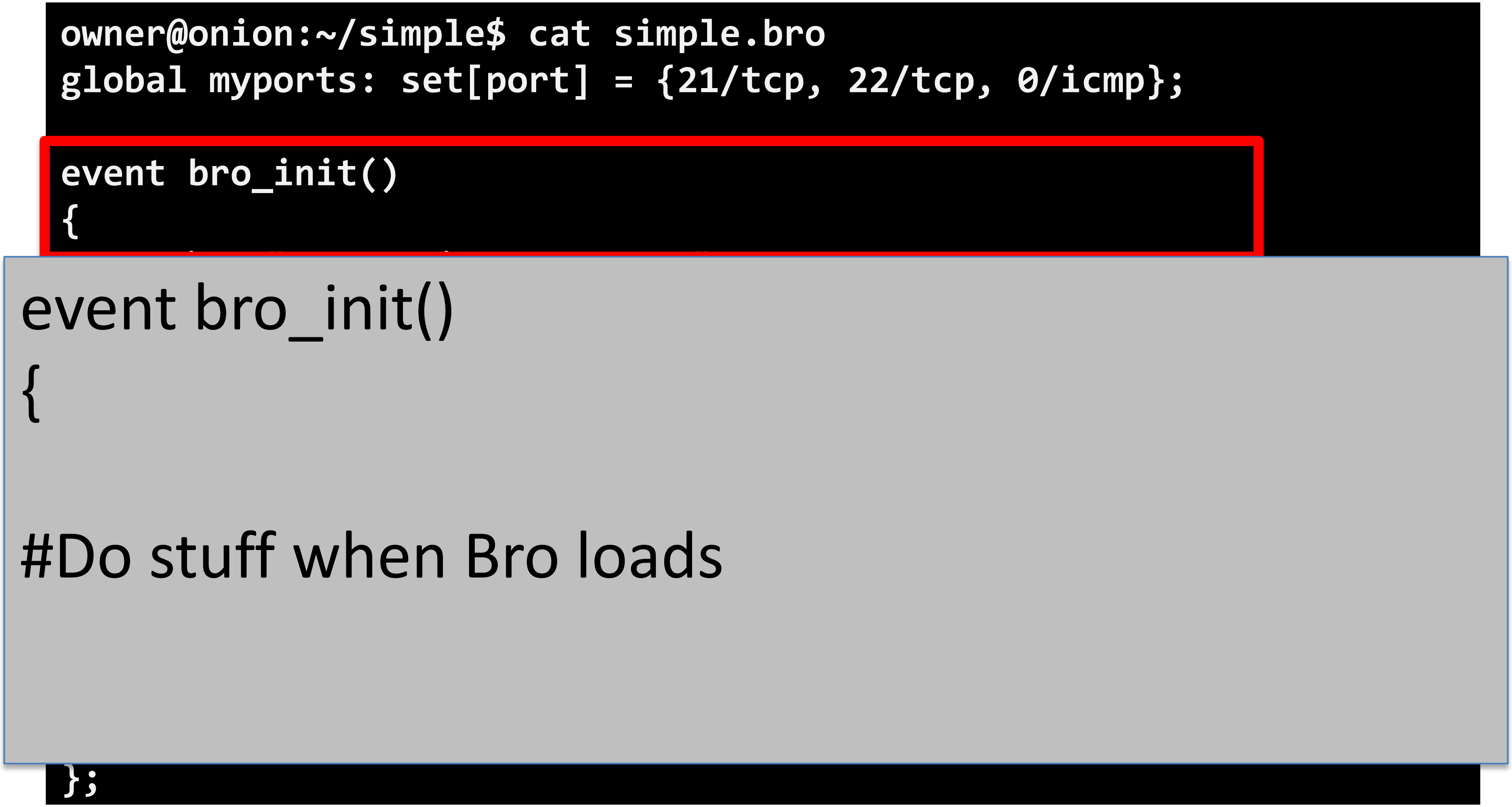
```
owner@onion:~/simple$ cat simple.bro
global myports: set[port] = {21/tcp, 22/tcp, 0/icmp};
```

global myports: set[port] = {21/tcp, 22/tcp, 0/icmp};

#Create a list

```
        print fmt("Port %s connection detected", c$id$resp_p);
        };
};
```

Bro Scripts

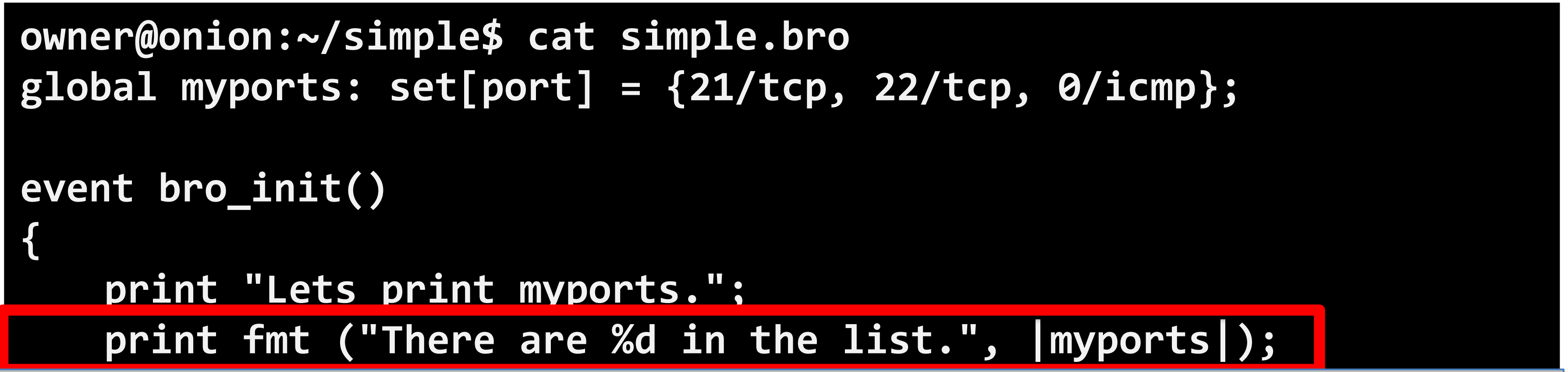


```
owner@onion:~/simple$ cat simple.bro
global myports: set[port] = {21/tcp, 22/tcp, 0/icmp};

event bro_init()
{
```

```
};
```

```
event bro_init()
{

#Do stuff when Bro loads
```

Bro Scripts



```
owner@onion:~/simple$ cat simple.bro
global myports: set[port] = {21/tcp, 22/tcp, 0/icmp};

event bro_init()
{
    print "Lets print myports.";
    print fmt ("There are %d in the list.", |myports|);
```

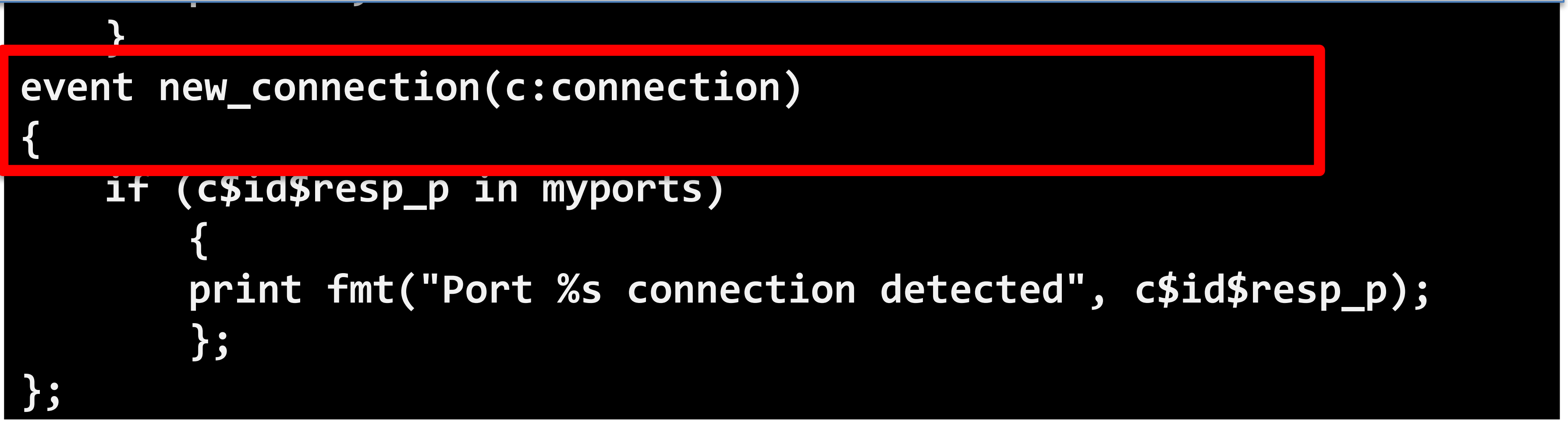
```
print fmt ("There are %d in the list.", |myports|);

    #Format string

    # |var| gets length of list
```

Bro Scripts

event new_connection(c:connection)

{

# Do the thing in curley braces when Bro detects a new connection

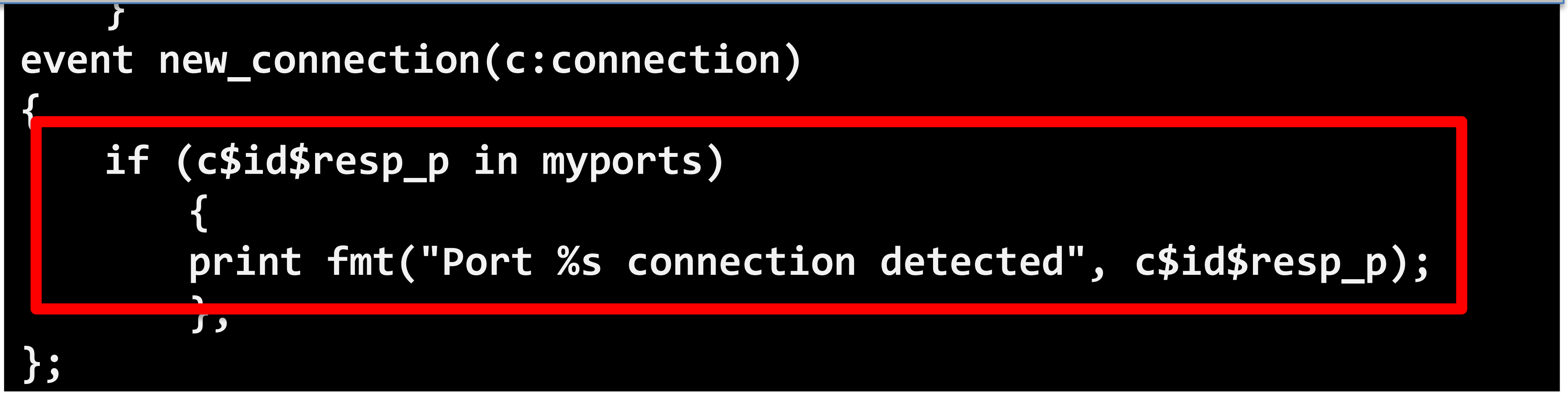


```
    }
event new_connection(c:connection)
{
    if (c$id$resp_p in myports)
        {
        print fmt("Port %s connection detected", c$id$resp_p);
        };
};
```

Bro Scripts

```
if (c$id$resp_p in myports)
    {
```

#If the destination port (c$id$resp_p) is in the list, do the thing in curley brackets



```
    }
event new_connection(c:connection)
{
    if (c$id$resp_p in myports)
        {
        print fmt("Port %s connection detected", c$id$resp_p);
        };
};
```

Something a little more useful…

Baselinereport.bro ::Pseudocode

1. Load table (baseline.data)
2. Check every new connection:

   – Is the destination on the baselined subnet?

     • If so, is it in the baseline?

   – If it's in the baseline, is the source address allowed to use that port?

3. Log any "No's"

Installing Baselinereport.bro

1. git clone https://github.com/hashtagcyber/baseliner.git

2. Edit line 32 of baselinereport.bro, replace with a comma separated list of subnets
3. Copy both files to /opt/bro/share/bro/policy/misc

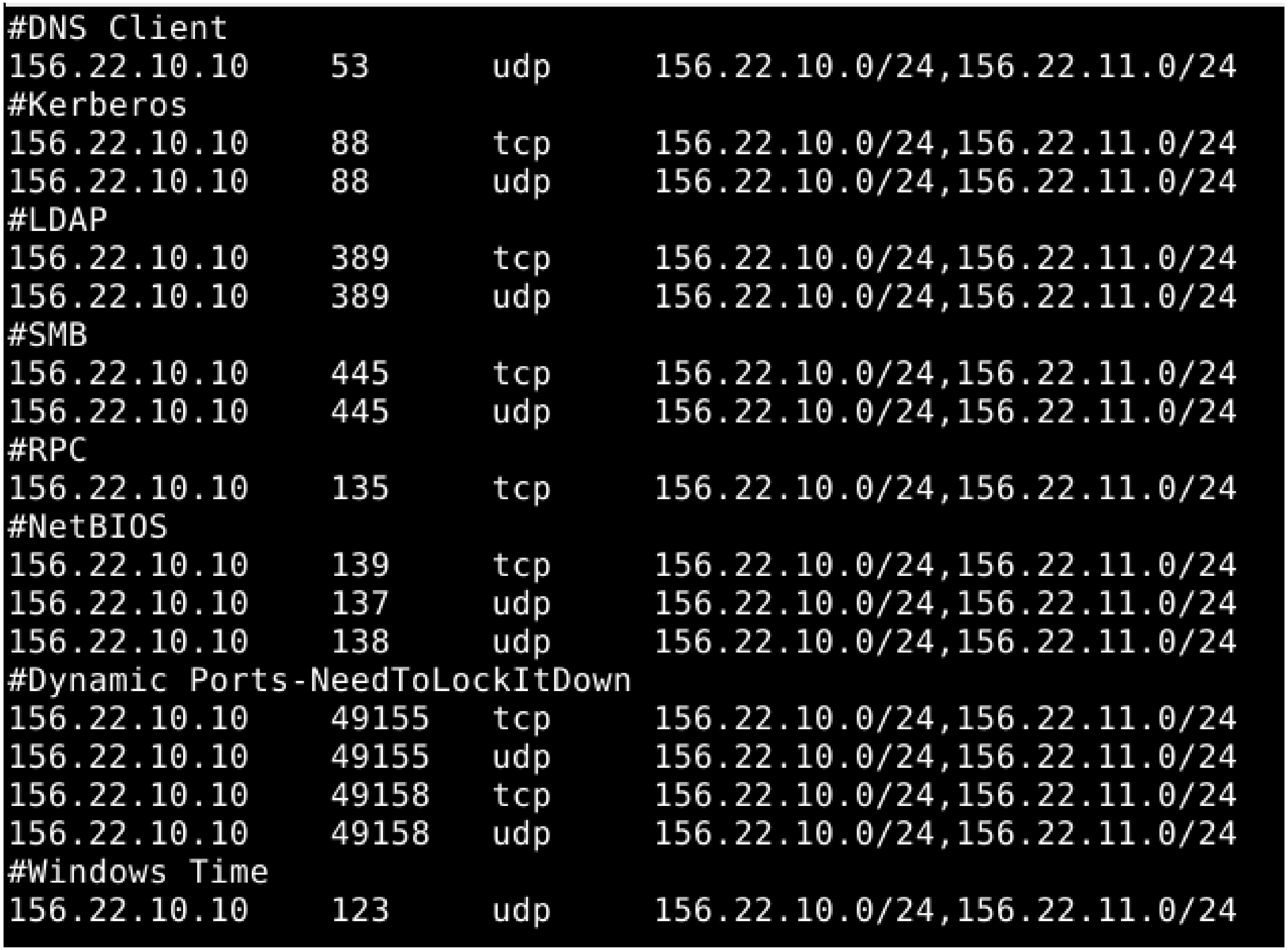
4. Add "@load misc/baselinereport" to /opt/bro/share/bro/site/local.bro

5. Restart Bro

ELSA Demo

- Useful search terms:
  - Show all notice’s generated by baselinereport
    - class=BRO_NOTICE "-" notice_type="TrafficBaselineException“
  - Show all connections to an IP, grouped by destination port
    - BRO_CONN.dstip=156.22.10.10 groupby:dstport
  - Show all connection to an IP/Port pair grouped by source IP
    - *BRO_CONN.dstip=156.22.10.10 BRO_CONN.dstport=445 groupby:srcip*

Updating Baseline w/ ELSa & VI



```
#DNS Client
156.22.10.10    53      udp     156.22.10.0/24,156.22.11.0/24
#Kerberos
156.22.10.10    88      tcp     156.22.10.0/24,156.22.11.0/24
156.22.10.10    88      udp     156.22.10.0/24,156.22.11.0/24
#LDAP
156.22.10.10    389     tcp     156.22.10.0/24,156.22.11.0/24
156.22.10.10    389     udp     156.22.10.0/24,156.22.11.0/24
#SMB
156.22.10.10    445     tcp     156.22.10.0/24,156.22.11.0/24
156.22.10.10    445     udp     156.22.10.0/24,156.22.11.0/24
#RPC
156.22.10.10    135     tcp     156.22.10.0/24,156.22.11.0/24
#NetBIOS
156.22.10.10    139     tcp     156.22.10.0/24,156.22.11.0/24
156.22.10.10    137     udp     156.22.10.0/24,156.22.11.0/24
156.22.10.10    138     udp     156.22.10.0/24,156.22.11.0/24
#Dynamic Ports-NeedToLockItDown
156.22.10.10    49155   tcp     156.22.10.0/24,156.22.11.0/24
156.22.10.10    49155   udp     156.22.10.0/24,156.22.11.0/24
156.22.10.10    49158   tcp     156.22.10.0/24,156.22.11.0/24
156.22.10.10    49158   udp     156.22.10.0/24,156.22.11.0/24
#Windows Time
156.22.10.10    123     udp     156.22.10.0/24,156.22.11.0/24
```

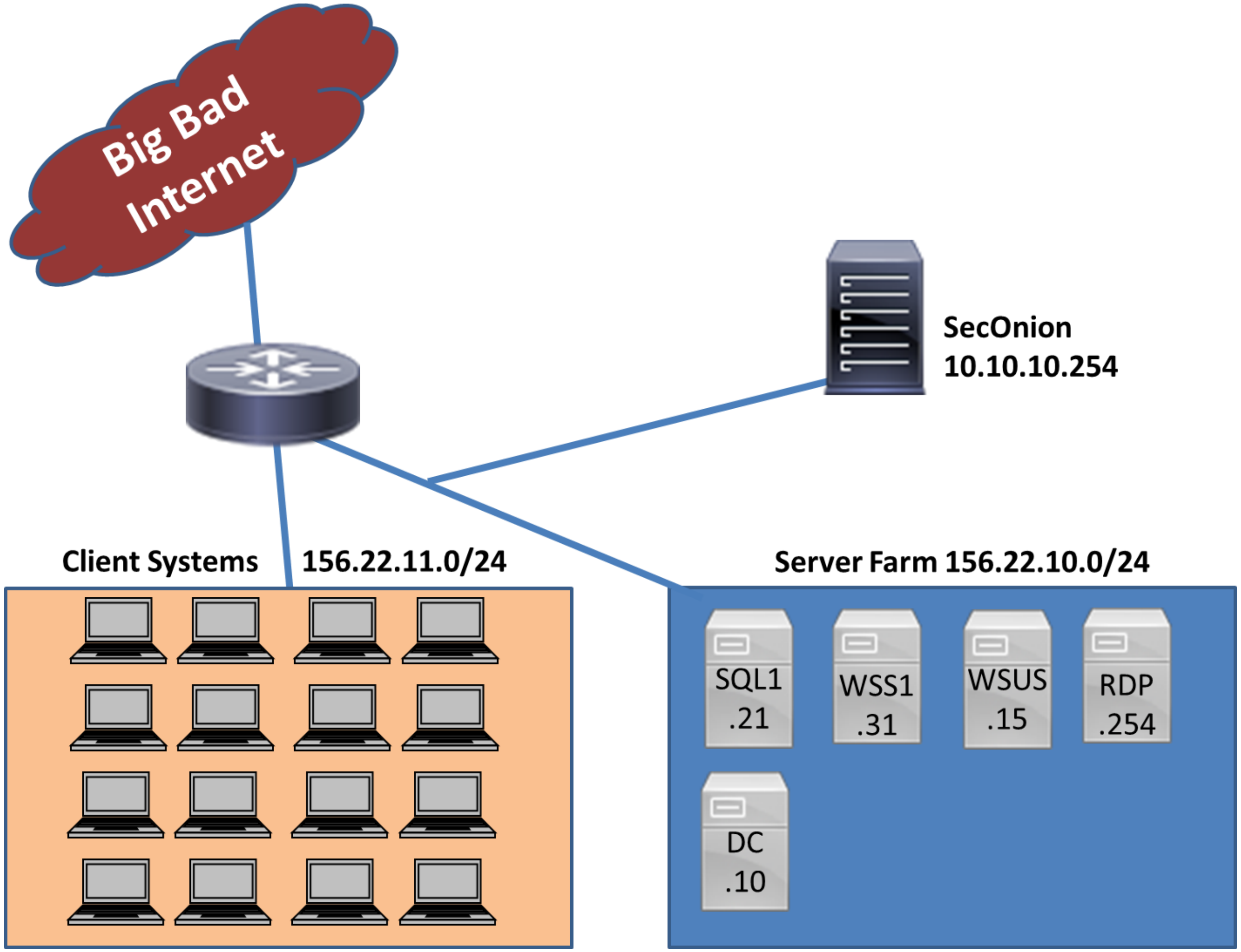
But ….

- That sounds like a lot of work…

Bropy

- Written in Python
- Installs baselinereport.bro script
- Parses notice.log
- Generates network baseline automatically
- Simple Yes/No interface

Scenario Network



Bropy

- Demo Time

Bropy on SecurityOnion

1. git clone https://github.com/hashtagcyber/bropy
2. cd bropy
3. sudo ./bropy.py
   - Select option 3 to install
     - Enter the subnet and CIDR that you would like to monitor
     - Example: 156.22.10.0/24
4. Select Y to restart Bro
5. Wait for logs to be generated….
6. sudo ./bropy.py
   - Select option 1 to “Auto Baseline”
   - Select option 2 for Y/N prompting

Use Case

- Generate a list of every port/protocol critical hosts receive connections on
- Receive alerts when non-standard connections are detected
- Baseline data can be used to generate firewall lists

CHIRON TRAINING

Questions?