HOME NETWORK SECURITY MONITORING

THE CHEAP AND EASY WAY!

WHOAMI

- Aaron J. Scantlin
- GSEC, GCFA
- Security Analyst – ISAM, University of Missouri, Columbia
- Adjunct Instructor – University of Missouri, Columbia
- Vice President – Central Missouri Infragard Chapter
- Member – SANS Advisory Board
- Member – REN-ISAC
- Huuuuuge nerd

WHY MONITOR THE HOME NETWORK?

- Attacks are not restricted to Enterprise networks
- Sanity-checking networked device communications
- Awareness of unauthorized devices on the network
- Provides additional troubleshooting support
- Allows you to create your own events to alert on
- …Why not? ☺

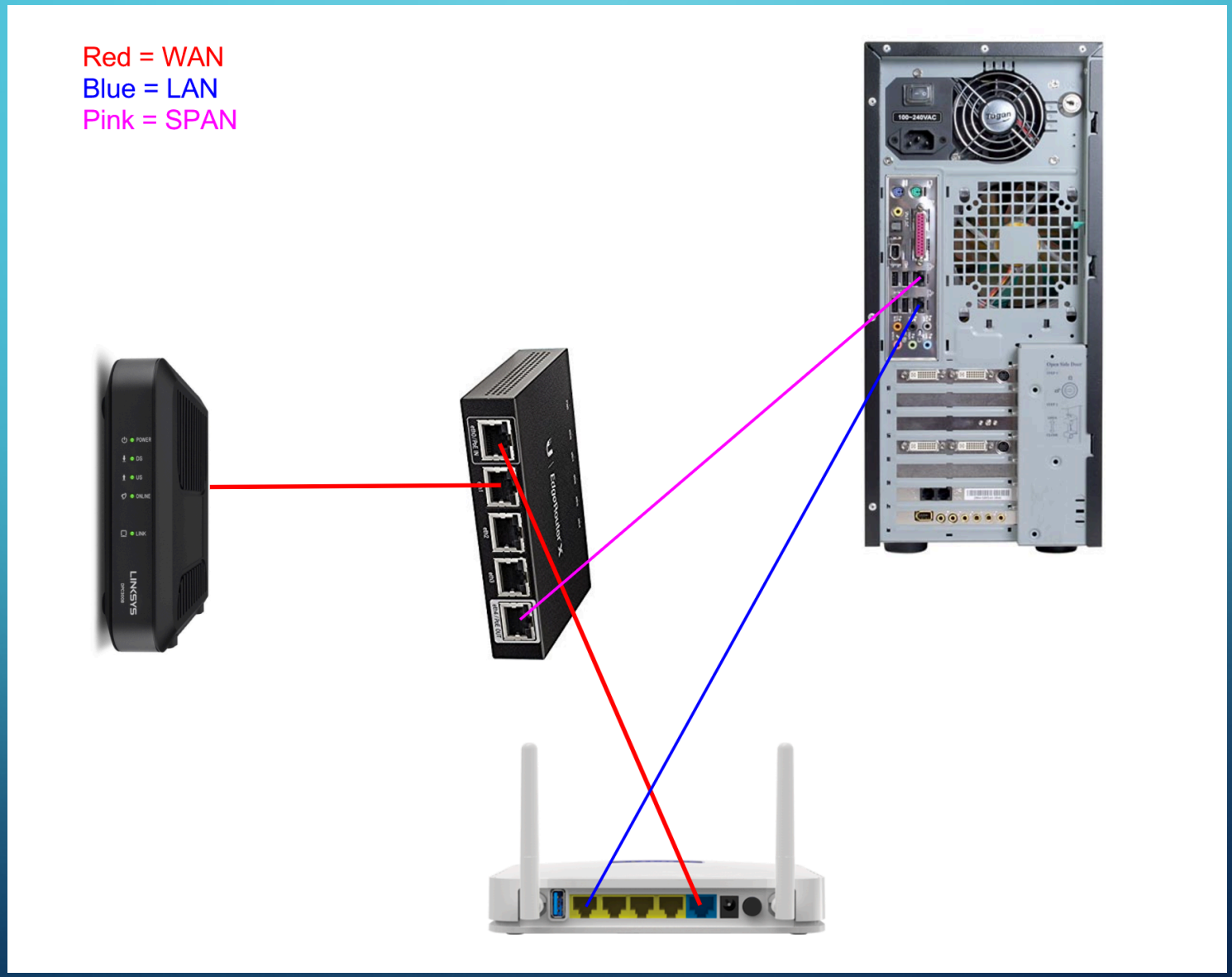
SHOPPING LIST

- Ubiquiti EdgeRouter X - $50
- VirtualBox VM - $0
  - We'll also need the SecurityOnion ISO
  - …also $0 ☺

TOTAL COST TO IMPLEMENT: $50*

*Don't have two NICs? Grab a USB 3.0 Ethernet adapter (total cost rises to ~$65)

EXAMPLE NETWORK CONFIGURATION



DOWNLOADING & INSTALLING VIRTUALBOX

- Visit https://www.virtualbox.org/wiki/downloads
  - Download the appropriate platform package for your OS
  - Grab the Extension Pack as well (the same pack supports all OSes)
- Run the installer and follow the instructions presented on the screen
- Start VirtualBox
- Run the Extension Pack file, select “Install” in VirtualBox, then accept the ToS from Oracle

DEPLOY AND CONFIGURE THE EDGEROUTER X

- Supply power to the ERX and connect your computer to the ERX's eth0 port
  - You'll need to give your computer an IP on the 192.168.1.0/24 subnet; the ERX doesn't have a DHCP server enabled by default
- Open a web browser and go to https://192.168.1.1
  - Accept the cert; you can add it to your trust store or supply your own later
- Login with the default credentials
  - Username: ubnt
  - Password: ubnt

DEPLOY AND CONFIGURE THE EDGEROUTER X

- Download the firmware update by visiting Ubiquity's website
  - Direct link: https://dl.ubnt.com/firmwares/edgemax/v1.9.7/ER-e50.v1.9.7+hotfix.3.5013617.tar
- Upload and install the new firmware image
  - System tab (bottom) > Upgrade System Image > Upload a file
- Reboot and login again

DEPLOY AND CONFIGURE THE EDGEROUTER X

- First we need to run the setup wizard
  - You should be taken there automatically, but if not, Wizards tab > Basic Setup
  - In the “One LAN” section, uncheck “Only use one lan”
    - This gives us a LAN on eth1 with a DHCP server, a separate LAN on eth2, eth3 and eth4 with its own DHCP server, default “Deny” firewall rules, and a WAN link on eth0
  - In the “User setup” section, choose “Create new admin user” and provide a new username and password
- Click “Apply”, then “Apply Changes”, then “Reboot Now”
- While the ERX is rebooting, connect your modem and wireless router to the ERX via eth0 and eth1 respectively; connect the NIC you’ll use for sniffing to the ERX via eth4

DEPLOY AND CONFIGURE THE EDGEROUTER X

- Need to mirror the traffic from eth0 and eth1 to eth4
  - Config tree tab > Interfaces > eth0
    - In the ”mirror” field, type *eth4*
  - Config tree tab > Interfaces > eth1
    - In the “mirror” field, type *eth4*

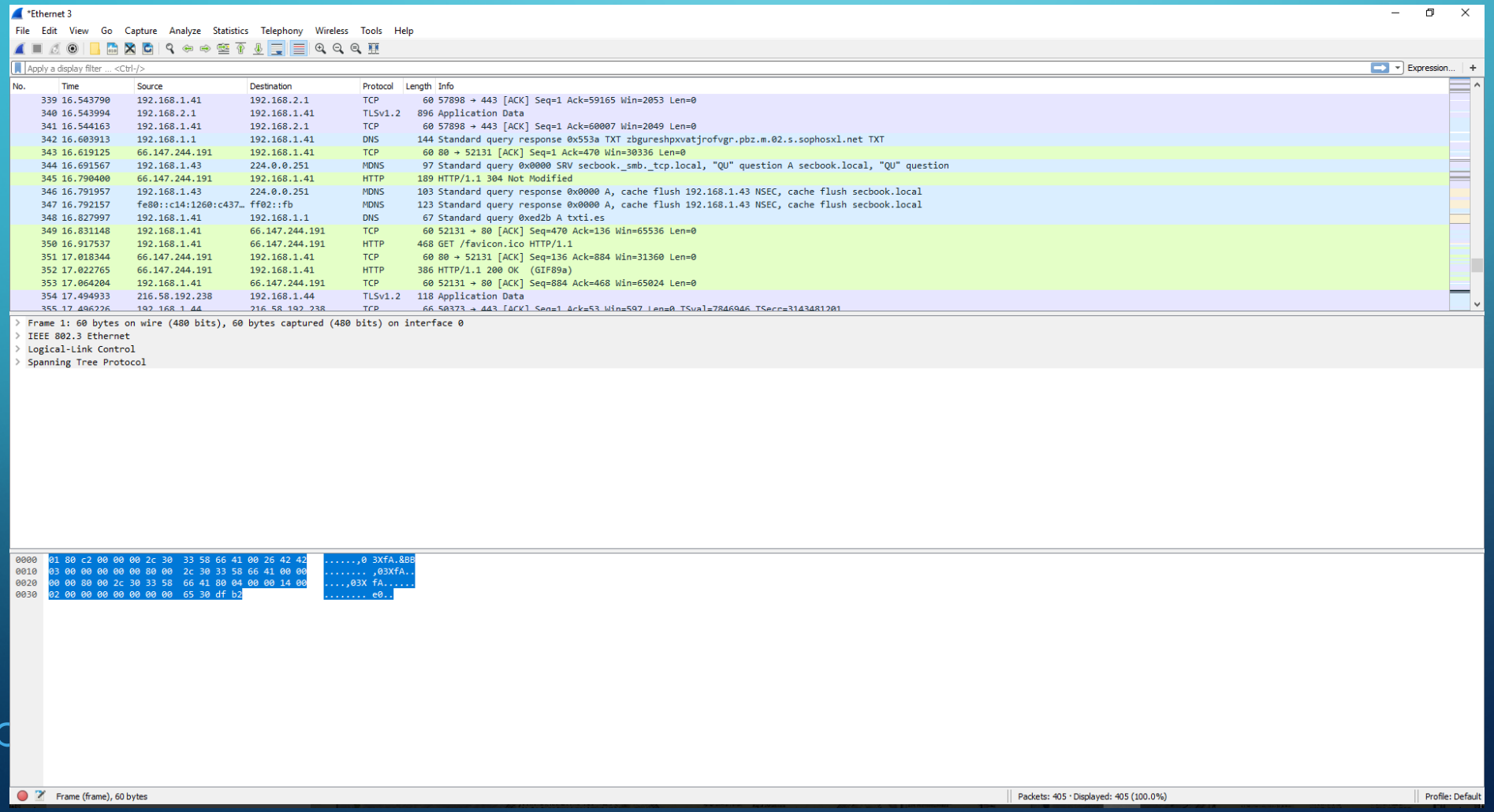
RECONFIGURE YOUR WIRELESS ROUTER

We need to configure your existing wireless router to act as an AP, not a router.

How to do this will vary by manufacturer, but for Netgear products…

- Login to the router web GUI
  - Advanced tab > Advanced setup > Wireless AP
  - Check the “Enable AP Mode” box and click “Apply”
  - Keep the “Get IP settings dynamically” option selection and confirm your choice

VERIFY EVERYTHING WORKS BY RUNNING A PACKET CAPTURE ON THE SPAN INTERFACE



WE ARE NOW MIRRORING ALL NETWORK TRAFFIC!

NOT ONLY THAT, WE'RE DOING IT WITH PER-HOST LEVEL GRANULARITY.

DOWNLOAD AND DEPLOY SECURITYONION

- Visit https://securityonion.net/ and download the latest ISO
- Fire up VirtualBox and click “New”
  - Name the virtual machine, choose “Linux” as the Type, “Ubuntu (64-bit)” for Version, then click Next
  - Set the memory to 4096 MB, then click Next
  - Keep the “Create a virtual hard disk now” option selected, then click Create
  - Keep “VDI” selected as the image type, then click Next
  - Keep “Dynamically allocated” selected, then click Next
  - Give the OS disk as much disk space as possible, choose a location to save it, then click Create

DOWNLOAD AND DEPLOY SECURITYONION

- Highlight the virtual machine you just created and make the following changes
  - Click “Display”, adjust the video memory to 128 MB
  - Click “Network”, select the “Adapter 2” tab
    - Enable it
    - Choose “Bridged Adapter” for the network interface
    - Click “Advanced”, and change the Promiscuous Mode drop-down to “Allow All”
  - Click “Storage”, click the disk with the green plus sign, then select “Add Optical Drive”
    - Select “Choose disk”, locate the SecurityOnion ISO you downloaded, then click “Open”
    - Click “OK”
  - Finally, click “Start” to power on the virtual machine

DOWNLOAD AND DEPLOY SECURITYONION

- Once the OS loads, select your language and click “Continue”
  - Select the “Download updates while installing” option, then click “Continue”
  - Click “Install Now”, then click “Continue”
  - Select your time zone (should be detected automatically) and click “Continue”
  - Select your keyboard layout, move the window to the left a bit click “Continue”
  - Enter the user information, then click “Continue”

SecurityOnion will now install itself as the OS of the VM!

When prompted, click “Restart Now” to complete the installation.

CONFIGURE SECURITYONION

- Login with the user credentials you created during deployment
- Click the icon in the upper left, then click “Terminal”
  - Run the command “sudo soup” (without quotes); this will update the SecurityOnion OS
  - If asked, don’t update 50unattended_upgrades
  - Reboot & login when prompted
- Double-click “Setup” and enter your password
  - Click “Yes, Continue!”
- Click “Yes, configure /etc/network/interfaces!”
  - Select ”eth0” for the management interface and click “OK”
  - Select “DHCP” and click “OK” (you can change this later)

CONFIGURE SECURITYONION

- Click “Yes, configure sniffing interfaces!”
  - Ensure “eth1” is checked and click “OK”
- Click “Yes, make changes!” followed by “Yes, reboot!”
- Login again once the system has rebooted, then double-click “Setup” again
  - Enter your password, then click “Yes, Continue!” when prompted
  - Click “Yes, skip network configuration!”

CONFIGURE SECURITYONION

- Select “Production Mode”, then click “OK”
  - Selet “Standalone”, then click ”OK”
  - Select “Best Practices”, then click “OK”
  - Select a Sguil username, then click “OK”
  - Enter a password for the user, click “OK”, confirm the password, and click “OK” again
  - Ensure “Snort” is selected as your IDS Engine, then click “OK”
  - Ensure “Emerging Threats Open” is selected as your IDS ruleset, then click “OK”
  - Leave the default PF_RING min_num_slots value at 4096, then click “OK”
  - Ensure “eth1” is checked, then click “OK”
  - Leave the default HOME_NET value, then click “OK”
  - Click “Yes, proceed with changes!”

CONFIGURE SECURITYONION

- Finally, we'll poke in hole in the firewall so we don't have to use the VM directly all the time…
  - Open a terminal and run *sudo ufw allow https && sudo ufw reload*
  - This gives us access to the web interface of ELSA, our default SIEM for Bro logs, from anywhere on our LAN; you might want to restrict this further if your home network is big enough





WE ARE NOW **CAPTURING** ALL NETWORK TRAFFIC!

STILL DOING IT WITH PER-HOST LEVEL GRANULARITY B)

R.I.P. YOUR DISK SPACE

DO YOU EVEN SCRIPT, BRO?

- Bro is an *awesome* network monitoring tool
  - Event-driven, connection-oriented framework
  - Correlation is left completely to the user
- SecurityOnion sets Bro up to monitor a lot out of the box:
  - Top connections, top services, DNS info, MIME types, HTTP, Kerberos, MySQL, RADIUS, RDP, SNMP, Snort, and more!
- Sguil is provided to help detect and deal with what Bro sees, but what if we want custom alerts?

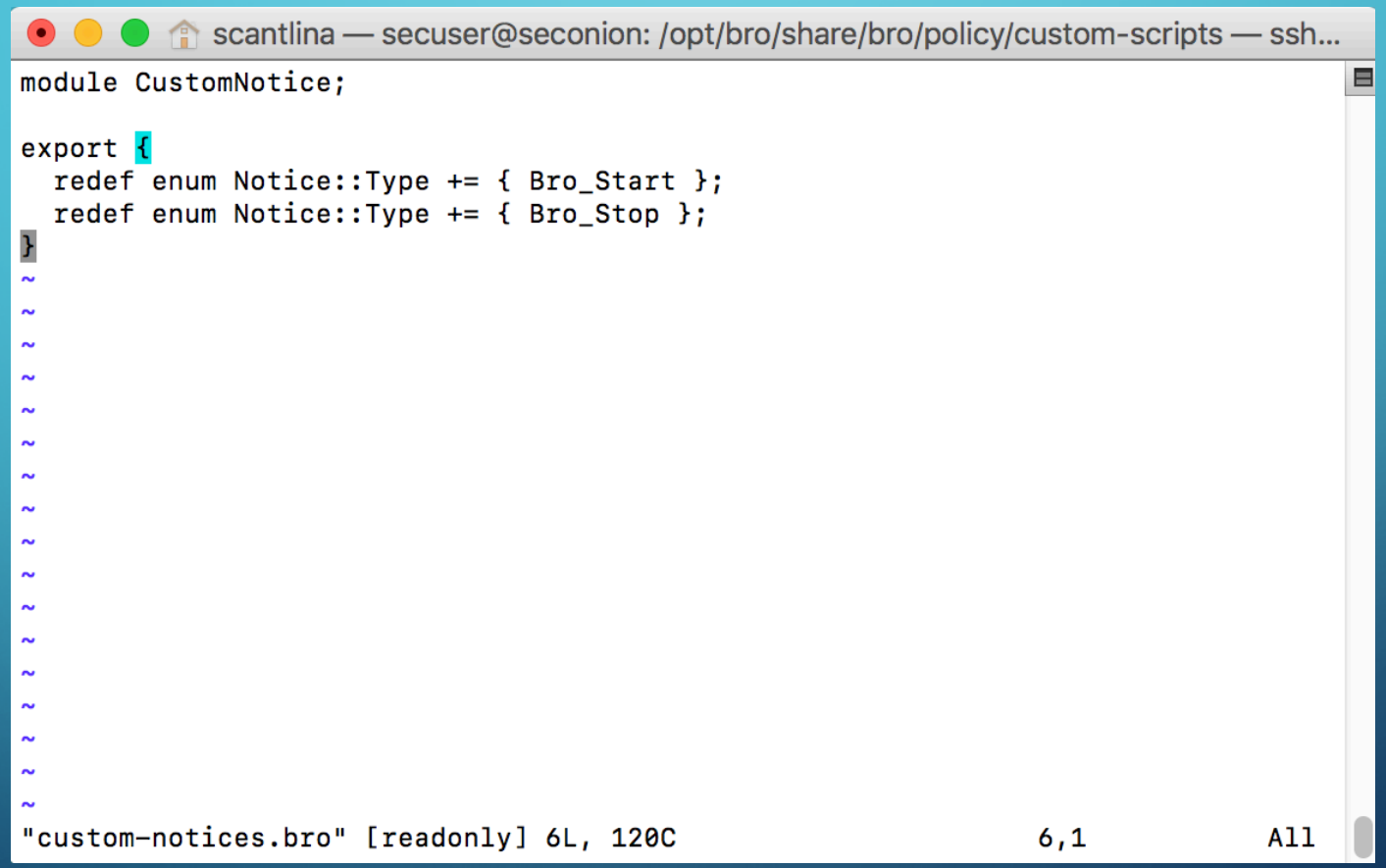
DO YOU EVEN SCRIPT, BRO?

- The answer: Bro scripting
  - Virtually everything Bro does is orchestrated through Bro scripts
  - We will store our custom scripts in a folder at /opt/bro/share/bro/policy/
    - Run the command *sudo mkdir /opt/bro/share/bro/policy/custom-scripts*
- Our collection of scripts will make up a module, which needs
  - main.bro - this is where we implement the logic for our scripts
  - One or more bro scripts - this is where we implement the functionality for our scripts
  - __load__.bro - this is a list of the files that make up your module

DO YOU EVEN SCRIPT, BRO?

- custom-notices.bro
  - As the name implies, the function of this script will be to facilitate custom notices
  - At a minimum, a module script needs
    - To declare the module name
    - Provide all variables and functions to be accessed by other scripts in the *export* block
  - For now, we'll just create some custom Notice types to raise when Bro starts and stops

DO YOU EVEN SCRIPT, BRO?



DO YOU EVEN SCRIPT, BRO?

- main.bro
  - Event-driven logic component
  - Two events always happen
    - bro_init()
    - bro_done()
  - Our template script will make use of these two events

DO YOU EVEN SCRIPT, BRO?

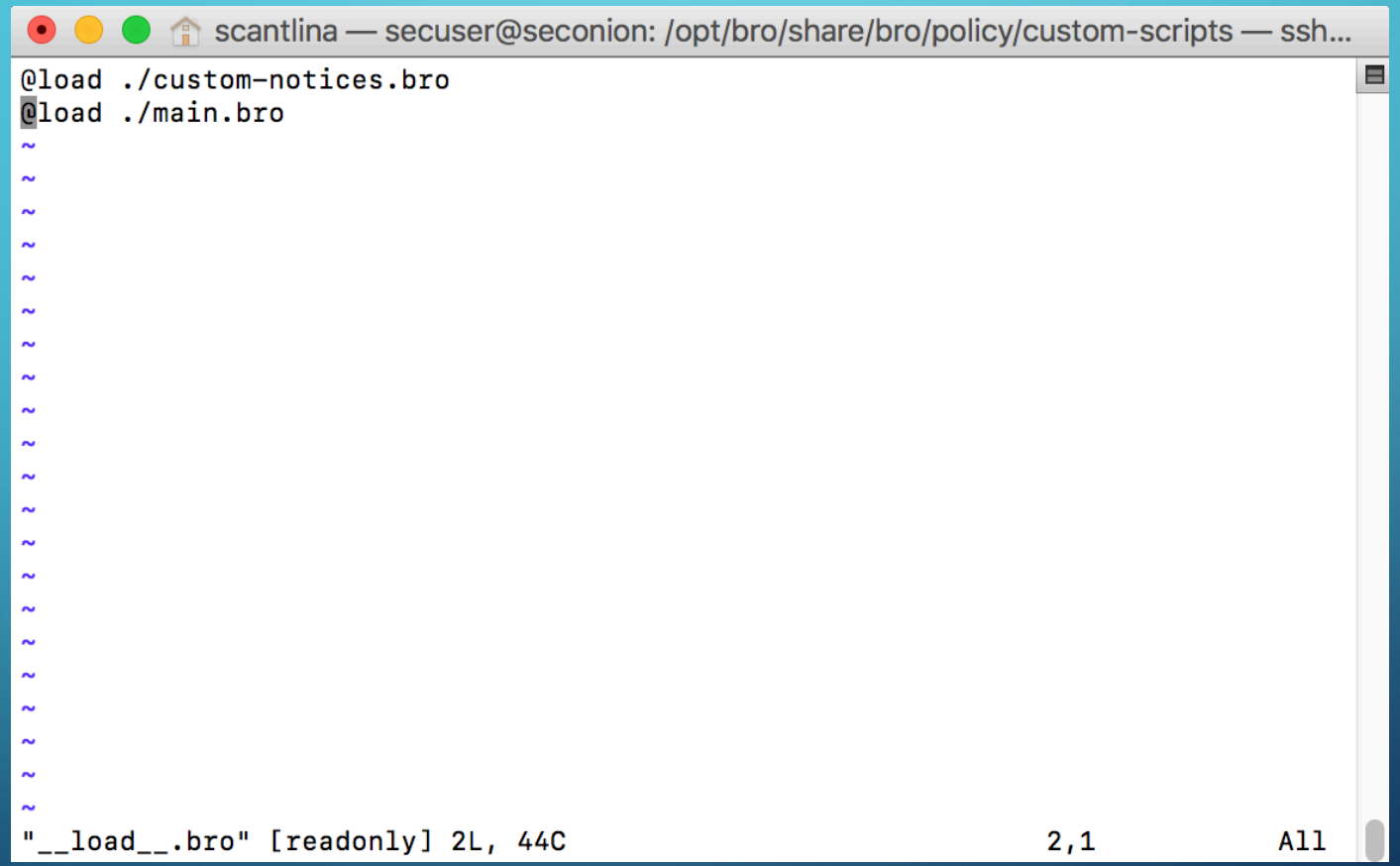
scantlina — secuser@seconion: /opt/bro/share/bro/policy/custom-scripts — ssh...

```
@load custom-scripts/custom-notices

event bro_init() {
  NOTICE([$note=CustomNotice::Bro_Start,
  $msg = "Bro engine started!"]);
}

event bro_done() {
  NOTICE([$note=CustomNotice::Bro_Stop,
  $msg = "Bro engine stopped!"]);
}
~
~
~
~
~
~
~
~
~
~
~
~
"main.bro" [readonly] 11L, 229C                          11,1          All
```

DO YOU EVEN SCRIPT, BRO?

- __load__.bro
  - Tells the Bro control process what files are needed for this module to operate

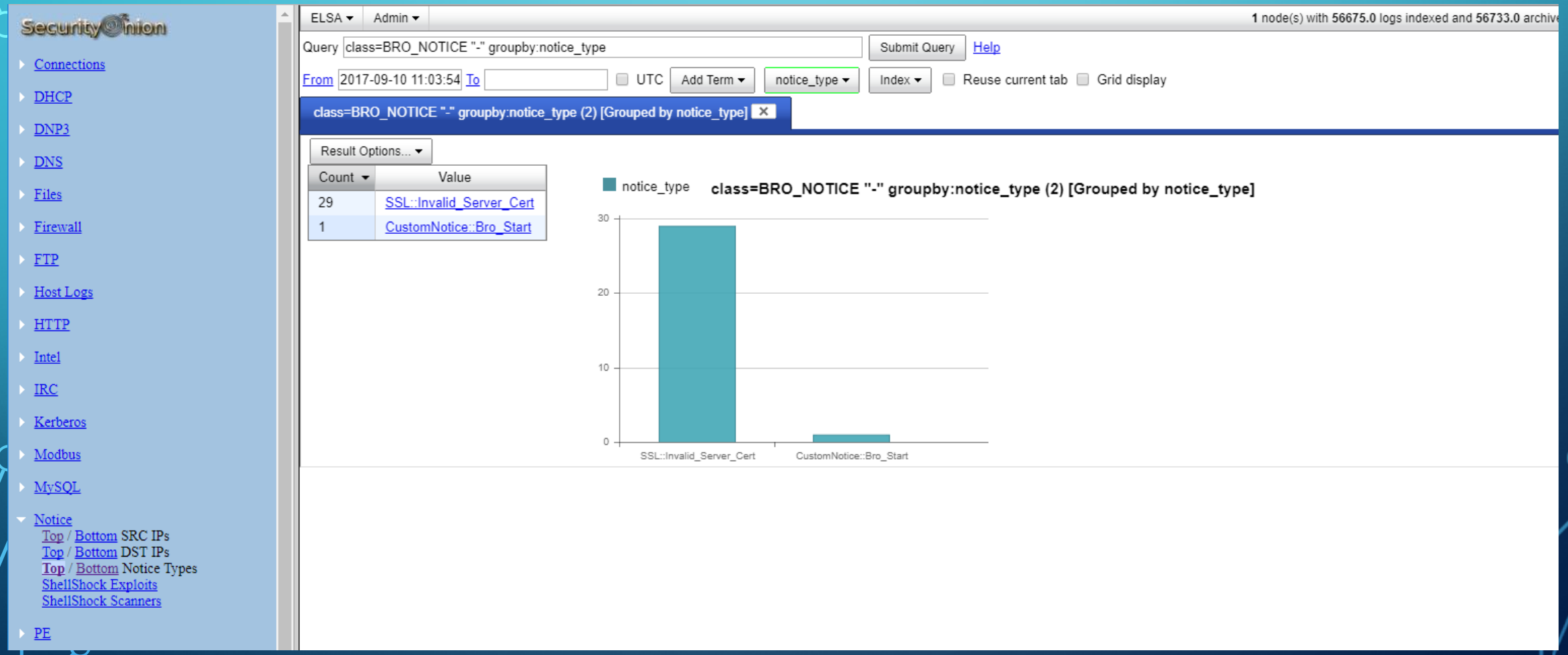
DO YOU EVEN SCRIPT, BRO?



DO YOU EVEN SCRIPT, BRO?

- Now we need to make sure Bro will use our custom module
  - Crack open */opt/bro/share/bro/site/local.bro* and add the line *@load custom-scripts* to the end of the file
- Time to reload Bro!
  - Run the command *sudo broctl*
    - From BroControl, run the command *install*
    - Then, run the command *restart*
    - Next, run the command *scripts* and confirm your *custom-scripts* module is in the list
    - Finally, run the command *exit* to return to the shell

DO YOU EVEN SCRIPT, BRO?



DO YOU EVEN SCRIPT, BRO?

Ok, that's cool… but not terribly useful. Bro scripting is too complex to cover in a matter of hours, but one important event for security alerts is *new_connection*

- *event new_connection (c: connection)*
  - Generated for every new connection
  - Event contains a *connection* object
  - Among other things, this object provides the origin IP and destination IP
- Using this connection, we can come up with some more meaningful alerts
  - Compare origin and/or destination with a known malicious IP list, alert on match
  - Examine port and protocol, alert on atypical results
  - Compare HTTP info with a known malicious domain list, alert on match

A FUN, TRIVIAL EXAMPLE…

Bing_User_Detected|Bing user detected at 192.168.1.47! Teach 'em to fish or drown 'em, your choice.|-|-|-|-|-|-|

OTICE srcip=127.0.0.1 srcport=0 dstip=127.0.0.1 dstport=0 mime_type=- desc=- protocol=-
notice_msg=Bing user detected at 192.168.1.47! Teach 'em to fish or drown 'em, your choice. sub_msg=-

A FUN, TRIVIAL EXAMPLE…

scantlina — secuser@seconion: /opt/bro/share/bro/policy/custom-scripts — ssh...

```
@load custom-scripts/custom-notices

event bro_init() {
  NOTICE([$note=CustomNotice::Bro_Start,
  $msg = "Bro engine started!"]);
}

event new_connection(c: connection) {
  if (c$id$resp_h == 13.107.21.200 || c$id$resp_h == 204.79.197.200) {
    NOTICE([$note=CustomNotice::Bing_User_Detected,
    $msg = fmt("Bing user detected at %s! Teach 'em to fish or drown 'em, your c
hoice.", c$id$orig_h)]);
  }
}

event bro_done() {
  NOTICE([$note=CustomNotice::Bro_Stop,
  $msg = "Bro engine stopped!"]);
}
~
~
~
~
"main.bro" 18L, 502C                                          18,1          All
```

WHERE TO GO FROM HERE?

- Start reading up on the Bro documentation
  - This will help you write better Bro scripts
- Check out Sguil
  - Out of the box alerting; might give you some good ideas
- Check out the plethora of Bro script repos on GitHub etc.
  - Don't re-invent the wheel (unless it's in the name of learning!)
- ZodiacFX (https://northboundnetworks.com/products/zodiac-fx)
  - OpenFlow capable switch for ~$85
  - Use software-defined network + Bro's NetControl framework to give Bro some teeth!

LINKZ

https://dl.ubnt.com/guides/edgemax/EdgeRouter_ER-X_QSG.pdf
https://www.netgear.com/support/product/gs105e.aspx
https://www.virtualbox.org/wiki/VirtualBox
https://github.com/Security-Onion-Solutions/security-onion/wiki/Bro
http://try.bro.org/#/?example=hello
https://www.bro.org/sphinx/frameworks/notice.html
https://www.bro.org/sphinx/scripts/base/init-bare.bro.html#type-connection
https://www.bro.org/sphinx/scripting/#the-connection-record-data-type

END

Aaron J. Scantlin
Twitter: @sysaaron
GitHub: https://github.com/coffeebro