



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Advanced Incident Response, Threat Hunting, and Digital Forensics (Forensics
at <http://www.giac.org/registration/gcfa>

Portable System for Network Forensics Data Collection and Analysis

GIAC (GCFA) Gold Certification

Author: Don Murdoch, GSE – donmrdch@gmail.com

Advisor: Richard Carbone

Accepted: July 11, 2016

Abstract

A portable lab environment for network level analysis is a necessary tool today for the forensic analyst. With today's malicious software and myriad of network aware client-side software, one of the tools that should be in the forensic analysts' toolbox is a portable response system for data collection and analysis. This paper will explain how to build a portable forensic workstation that provides several virtual environments installed together with supplemental hardware, such as multiple NICs and modern managed switch in order to provide a network forensic tool. VM's will include pfSense 2.2 running in transparent firewall mode along with other supporting packages, a network security-monitoring platform. A cookbook approach will be used to explore common use cases for the network and system forensic analyst, such as updating rules, sharing data among multiple environments, extracting data from packet captures, and clearing out all of the tools installed to start an investigation. This paper was written to provide a build outline for using pfSense and Security Onion to achieve these goals.

1. Introduction

There are many tools in the forensic analysts' toolbox that focus on analyzing the individual system itself, such as file system, deleted data, and memory analysis. This part of the analysis process inherently limits attention to the system itself; instead, it is often performed after the fact during an investigation, requiring that the analyst interact with the system to some degree. Network level analysis can be a compensating factor because network level analysis can improve data collection, can minimally interrupt the suspect system, can provide reliable data to catch a user or the system "red handed" and can be a tremendous ally in informing a systems forensics analysis. Further, network analysis provides the ability to capture activity that *may never persist to the disk* – a key capability for any malware analysis team. Network analysis also can lend to protocol analysis and the ability to improve anomaly and IDS signature detection, when the analysis team has visibility to the system and the network for trace events that occur at nearly the same time.

When responding to an incident, one of the challenges is the system can easily "lie" to the analyst as the system may be exhibiting rootkit behavior. The network, on the other hand, cannot lie, because for any network aware software to function, the network itself *must* function.

Enterprise grade network level port mirroring is a powerful capability – but it has its limits. It must be setup and is a scarce resource. For example, CISCO switches only support two mirror or Switched Port ANalyzer (SPAN) ports, and security is fortunate to get one of them as they consume a valuable resource and can be computationally intensive on the switch. While solutions like Gigamon can provide multiple port mirror paths, they are also quite expensive. Lastly, these solutions are typically implemented in the data center or at a perimeter point – not in the field. Therefore, they cannot observe suspect traffic on an edge switch, broadcast traffic that is stopped by a router, or localized malware that manipulates traffic at the MAC layer.

Don Murdoch, GSE – donmrdch@gmail.com

This paper provides a walkthrough for the DFIR community for building a very low cost portable analysis capability that can be used in the field to collect real time data for a suspect system with minimal change to the network environment. The techniques herein can also be easily adapted for a malware oriented lab environment.

2. Requirements Analysis

There are several requirements that a portable analysis and collection system must meet in order to be an effective tool in the analysts' toolbox. Each of these requirements, in some way, provide for integrity of the incident response process. The requirements listed below are based on the author's 12+ years of fieldwork as an incident responder working for EDU's, NGO's, commercial sector and discussion with other IR team members.

Table 1: Requirements for a Portable Incident Response and Network Forensics System

Requirement	Purpose	Technology / Tool
Separate network interfaces	Separate interfaces provide isolation, traffic management, separate processing power, collection specifically for network security monitoring, and isolation environments.	Multiple NIC's using USB or other adapters – Examples: StarTech USB/LAN adapter ; Apple Thunderbolt Adapter
Bridging stateful firewall	Provide strong control, suspect client isolation, all without requiring a change in client IP address or changing the TTL value of traffic. Also provides rich logging of <i>any observed traffic</i> .	pfSense 2.2.4 to 2.2.6 , not 2.3. As of May 2016, ntopng is not compatible with <i>pfSense</i> 2.3.
Analysis system and supporting tools	A network analysis system that can be used to monitor all the network traffic to/from the suspect system and provides NSM functions (e.g., IDS, analysis, record keeping...)	Security Onion provides Snort, BRO IDS, log management, IDS rule updates, and PCAP analysis suite
Securable host system	The host system itself must be securable, such as using whole disk encryption. It should	Mac Book Pro – selected because of high-speed USB,

Don Murdoch, GSE – donmrdch@gmail.com

Requirement	Purpose	Technology / Tool
	also provide a usable platform for the analyst to write notes, make screen captures, interoperate with Windows and Linux systems.	Thunderbolt Ethernet for full Pcap monitoring, high res display, and low susceptibility to malicious software.
Common collection point	For this paper, network forensics depends on repeatable processes and data integrity will be demonstrated	Share points through CIFS
Virtualization	Provide restartable and customizable computing platform	Parallels or VMware Fusion

3. System Overview and Deployment

3.1. Overview

The portable analysis system will meet all the requirements listed above, as illustrated in “Figure 1: Logical Deployment Model” on page 4 and explained more fully below.

Isolate and control access to Production Zone from Suspect Zone with minimum impact: A transparent/bridge firewall provides an analyst with two capabilities. The

configuration of the suspect *does not need to change*, which is easily detectable by an intruder, may interfere with normal IT operations and is disruptive to the user.

Collect Best Evidence

network packet capture data: For this solution, the suspect zone will be

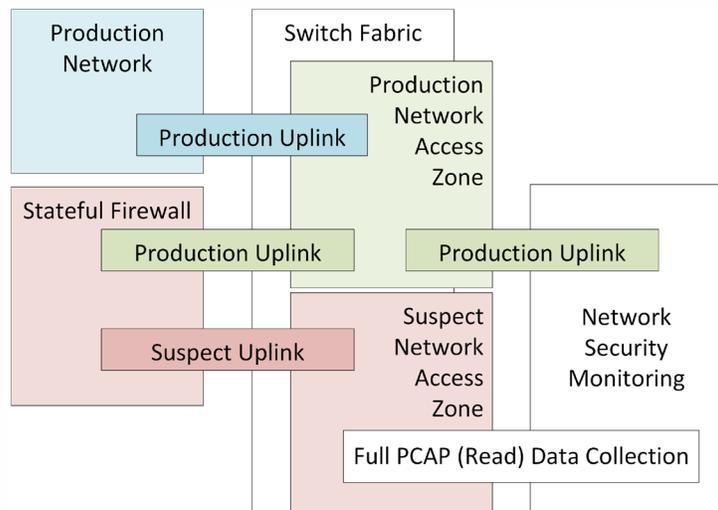


Figure 1: Logical Deployment Model

configured to segregate network traffic from the suspect to the production environment. This mechanism allows the suspect system to be online, the network security-monitoring platform to capture full content data, and for the NSM's analysis capabilities to only focus on communication to and from the suspect. One of the challenges in a corporate or large network setting is reducing the data set for analysis down to the suspect. By inherently limiting that access, the analyst can preserve data solely related to the suspect from the beginning.

Secure data storage, tools preservation, and a common collection point:

By using virtualization techniques and a common shared directory structure, the analyst can also preserve the PCAP data, the environments used to collect that data, and the results of the collection process at a very low price point. Given the very low cost of removable 1 TB hard drives, if there is a need to preserve a complete data set and toolset for a case, all of the collected data, the OS images, and the VM configuration can be stored for years after it is collected.

3.2. OS Image Deployment

3.2.1. MacBook Pro and Other Portable System Options

There are three main platforms available for the solution. Windows, Linux and Mac OS/X. Apple provides a unique capability for a portable solution with its MacBook Pro system. As of April 2016, the MacBook pro has Retina display that is the highest resolution on the market can be configured with PCI/e based SSD drives, has two Thunderbolt 2 interfaces, two USB3 interfaces and an 802.11 a/b/g/n wireless adapter. Having two Thunderbolt interfaces means there are two 10 GB interfaces that can be separately managed for this solution and connected to LAN connections through an inexpensive (\$35 USD) adapter while other necessary ports are unused. There are very few, if any, multi NIC portable computers in the size and weight range of the MacBook Pro.

In contrast to Thunderbolt, the USB 3 specification provide up to 5.0 GB data throughput. While it is true that inexpensive 1 GB USB interfaces are available, overcoming multiple ports requires more hardware, hubs, power adapters, and desk

Don Murdoch, GSE – donmrdch@gmail.com

space. For USB3, StarTech has a dual port NIC adapter described in this solution, costing about \$53 USD. Of course, there certainly are other options on the market today. What is critical to the solution implemented is that the host system not be a bottleneck to LAN and disk throughput. Thus, the MacBook Pro may be one of the best options today for the configuration described.

3.2.2. Transparent Firewall using pfSense

A Transparent/Bridge Firewall¹ is different from a traditional firewall because it is not a routable hop at Layer 3 (network addressing and routing) between networks. A traditional routable firewall functions *as a gateway*, meaning that it needs a routable address on each side, may perform address translation and makes decisions on what traffic can flow. A transparent firewall functions at Layer 2, a “bump in the wire” or a stealth device. It also does not change the Time to Live (TTL) value, it can block traffic and it can cause significant network problems because it passes Layer 2 traffic such as ARP requests and broadcast requests. In this solution, pfSense is configured as a transparent firewall in order to meet the requirements, and preventing the need to change the suspect system’s networking configuration during an incident. If pfSense were configured as a “standard stateful” firewall, then a suspect would need to request a new address via DHCP or need static IP configuration. This process would be disruptive, and *easily detected* by an attacker.

In a lab environment, pfSense can be configured as a traditional stateful firewall. This configuration will allow for a much wider range of security focused pfSense packages installed in the environment as, such as HTTP caching proxy, SSL interception, FTP proxy, and other features.

¹ This definition is adapted from the Cisco ASA product documentation, as the pfSense documentation does not have a formal definition.

3.2.3. pfSense Transparent Firewall Build Notes

An overview of the command and most recommended method for configuring a transparent firewall using pfSense is examined, adapted directly from an article posted to the pfSense forum (dadgbk, 2012)

1. Install pfSense with a WAN and LAN interface, validate that the system is functional, and can perform DNS resolution. Save off this configuration as a baseline.
2. Disable *automatic* NAT (but not the firewall itself). On the Firewall > NAT on the Outbound tab, select “Disable Outbound NAT rule.”
3. VERY IMPORTANT: Go to the 'System -> Advanced -> System Tunables' and set `net.link.bridge.pfil_bridge` from 'default' to '1'. Depending on your specific setup, If you miss this step or do not do this very quickly in the process, and the analysis system is connected to a live network, the system is likely to flood both NIC's with excessive traffic when the bridge is enabled and push CPU > 75% and become unresponsive.
4. Bridge WAN and LAN by going to 'Interfaces → Assign → Bridges’ tab, selecting the WAN and LAN, and then save.
5. Create a third interface, which will be named OPT# (1, 2, 3 ...). Assign the bridge to it by 'Interfaces → Assign → Network Port'.
6. Add an IP address to the bridge interface which is on the production network; this IP is the one you will use to access the firewall long term. In a portable configuration, a local IP will be needed. As an alternative, a fourth logical NIC can be added and connected to the “host only” network for the virtualization platform in use.
7. Add allow all rules to ALL firewall interfaces to avoid being locked out. Ifaces OPT, WAN, and LAN. These will be adjusted later!
8. Set WAN and LAN interface type to 'none', meaning that the IP address is removed from the interface. (Under 'Interfaces' in GUI). If you have any error

or problem and lose access from the Web GUI, an IP address can be assigned back in the text mode interface on the console.

9. Disable pfSense’s DHCP server, which is normally enabled during the text mode initial installation process. There is a checkbox on Services -> DHCP Server to disable this service.
10. Confirm that the firewall is accessible on the IP assigned in step 5.
11. Carefully modify your firewall rules to be more restrictive (i.e., DNS, DHCP, etc.)

3.2.4. pfSense VM Binding Notes

In order to use pfSense with virtualization solution, the logical interfaces (WAN, LAN, and OPT1) need to be connected to *separate* virtual networks, which in turn are bound to specific LAN interfaces. The WAN interface should be bound to one VM

network, which is in turn bound to one LAN adapter. The WAN interface would be accessible on the production network, which requires “allow in” rules in the firewall, would also be used for software updates and would make the

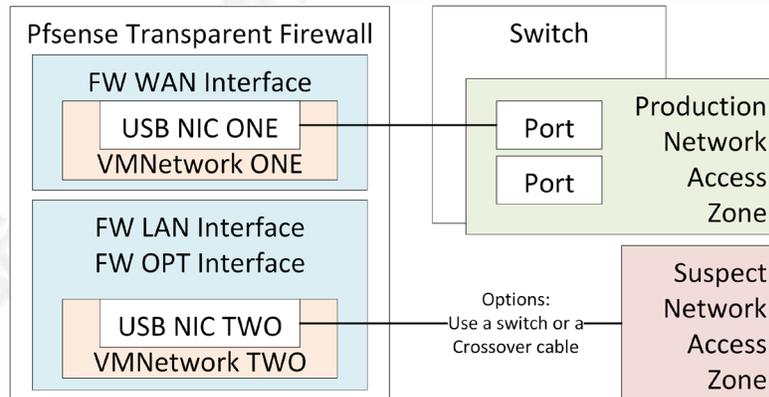


Figure 2: pfSense and VM Network Configuration

management console and *ntop* configuration visible on the “production” network – isolated from the “suspect.” In this solution, it is bound to the first interface on the USB dual NIC. The LAN and OPT interfaces are bound separately to another VM network, which provides complete isolation from production, and prevents any potential issue with bridged traffic. Further, this setup allows the suspect network to be connected to the suspect PC through a switch or a crossover cable.

3.2.5. Flow Collection and Analysis: Add NTOPNG in pfSense Image

A key component of this solution is a real time and historical visual analysis capability to aid the forensic analyst. This capability is provided by ntop, which provides a rich visualization capability of current flows, recent flows over the past few hours, and can optionally be configured to record and show historical flow data, albeit at a lower performance point. ntopng is an additional package and must be separately installed. Unfortunately, as of July 2016, the pfSense 2.3 distribution does not have an ntop package – thus, pfSense 2.2.6 is recommended.

3.2.6. Network Security Monitor: Security Onion 14.04 NSM Platform

Security Onion is packaged suite of network security monitoring tools that include IDS provided by Snort or Suricata, full Pcap data capture, an interface to Snort through Sguil, the Bro IDS, and log collection for all of its system components through Enterprise Log Search and Archive (ELSA).

3.2.7. Security Onion 14.04 Build Notes

Security Onion is an open source network security monitoring solution built and maintained by Doug Burks. It is distributed as an installable ISO. There are detailed setup instructions on the website. To setup, follow the production deployment instructions, not the evaluation instructions.

Prerequisites:

1. Portable system with multiple NIC's, proven to operate
2. Virtualization software – VMware has worked well for this solution, Parallels proved to be problematic early on. VirtualBox was not attempted.
3. A Snort.ORG (SO) oinkcode, which requires you to sign up for a basic account for SourceFire VRT rules on Snort.Org. SO has four ruleset options which are shown here. Given the purpose of this solution, both the Snort VRT rules and Emerging Threat rules should be added in order to provide maximum coverage. However, the Snort VRT rules are 30 days behind the production SourceFire rules. A subscription costs \$30/year for personal/home use and

\$400/year for business use, as of June 2016.

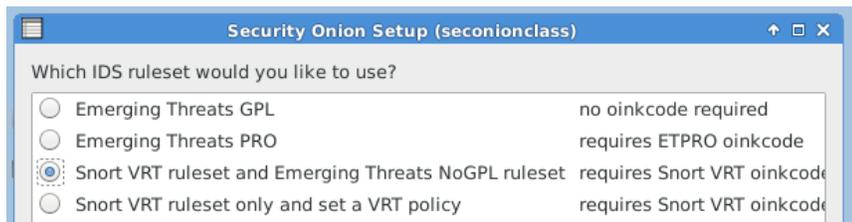


Figure 3: Security Onion Snort Rules

3.2.8. Security Onion GUI Setup Notes

These notes supplement the “Production Deployment” notes provided on the Security Onion website:

1. After the system boots, click on the “setup” icon, and run the setup installer. As the installer runs, these configuration items will be setup.
2. The management interface will be identified – usually eth0. Note which network is bound to the first adapter defined in the virtualization software, as this should be the “management” interface, and accessible on the same production network as the pfSense WAN interface.
3. The network setup will be determined (DHCP or static). Once the system reboots, update the Security Onion package set and install Ubuntu Updates:
 - a. Run `sudo soup`, then run `sudo apt-get update && sudo apt-get dist-upgrade && sudo reboot`. These steps will take several minutes.
 - b. Run the remaining commands as described in the production setup.
4. The monitor interface will usually be eth1. Note that this interface should be bound to the second logical interface defined in the virtualization software, which in turn is bound to the second virtual network, which then allows Security Onion to monitor the connection to the “suspect” system.
5. After the system reboots, click on the “setup” icon, and run the setup installer for the second phase. The first choice is the mode. For this solution, choose “production,” on the next dialog chose “standalone,” and “Best practice” to ensure that all components are installed. The process will take a few minutes.

- a. A Sguil user name and password will be setup – this account is used to authenticate to several of the applications provided with Security Onion.
 - b. Something like “sguiluser” would suffice. For the password, experience advises to avoid special characters for your first deployment. Something like “SguilPass123” is memorable for the setup process.
6. For the IDS engine, choose whichever is more familiar. Snort is the choice for this solution. Three of the four options for rules will require an oinkcode. A single IDS instance and Bro instance will suffice – the solution is designed to monitor one suspect system.
 7. Setup ends with a confirmation of these steps, and then it configures the system.

4. Solution Tool Kits in Practice

4.1. Snort Intrusion Detection System

Snort is a well-known signature based intrusion detection system with several open source inexpensive rulesets to detect common threats.

There are two places where a Snort IDS sensor can be instrumented in this configuration. First, Snort can be installed on the pfSense firewall. Second, snort is installed with Security Onion itself, and can be used to monitor traffic. There is a specific advantage in having multiple snort instances – different default rule sets can be installed, one on pfSense and one on Security Onion. As guidance, the ruleset that is more likely to be configured using manual processes should be enabled on Security Onion.

The pfSense version of Snort is “checkbox oriented,” or fully configured through a rich GUI. It follows some pfSense “naming” conventions and is better integrated with the firewall. For example, snort has several variables which define how it operates that are defined on the “Firewall | Alias” tab, and not edited in the snort configuration files, as

Don Murdoch, GSE – donmrdch@gmail.com

a long time snort user would expect. In contrast, snort on Security Onion is configured by editing text files, updated with Pulled-Pork. Site-specific rules reside in the “local.rules” files, and updating variable definitions in the snort.conf file.

If Security Onion could not retrieve snort rules during the install process, it will set the LOCAL_NIDS_RULE_TUNING option in /etc/nsm/securityonion.conf to ‘yes’. Set that to ‘no’ and run sudo rule-update, which will pull down the most recent rule data sets and apply any new local rules.

4.1.1. Suppressing Snort Rules on pfSense

During an incident, the responder needs to minimize extraneous alerts from IDS systems, with an understanding of their particular environment. An example of a false positive rule is shown below. This rule is triggered by a Roku streaming media server making a DNS query out to Google’s DNS servers, in violation of its local DHCP configuration. Two things are learned from this. 1) A system on a “suspect” network is querying a non-authorized DNS server, 2) In this environment, the device is a preconfigured “appliance,” and can be considered “safe” for this type of alert.

06/09/16 21:45:31	1	UDP	Attempted User Privilege Gain	8.8.4.4  	53	192.168.1.225  	35693	3:19187  	PROTOCOL-DNS TMG Firewall Client long host entry exploit attempt
----------------------	---	-----	--	--	----	--	-------	--	---

Figure 4: Excessively chatty IDS Alert

Therefore, this particular alert – in this specific case – can be suppressed. To suppress the alert, click the gray plus icon left of the red and white X icon under 3:19187. That will add it to the suppression list, which can be edited as needed.

Snort: Suppression List Edit - lansuppress_55ef64343c09b ?

Snort Interfaces Global Settings Updates Alerts Blocked Pass Lists **Suppress** IP Lists SID Mgmt Log Mgmt Sync

Add the name and description of the file.

Name
The list name may only consist of the characters "a-z, A-Z, 0-9 and _". **Note:** No Spaces or dashes.

Description
You may enter a description here for your reference (not parsed).

NOTE: The threshold keyword is deprecated as of version 2.8.5. Use the event_filter keyword instead.

Apply suppression or filters to rules. Valid keywords are 'suppress', 'event_filter' and 'rate_filter'.

Example 1; suppress gen_id 1, sig_id 1852, track by_src, ip 10.1.1.54
Example 2; event_filter gen_id 1, sig_id 1851, type limit, track by_src, count 1, seconds 60
Example 3; rate_filter gen_id 135, sig_id 1, track by_src, count 100, seconds 1, new_action log, timeout 10

```
#PROTOCOL-DNS TMG Firewall Client long host entry exploit attempt
suppress gen_id 3, sig_id 19187
```

Figure 5: Snort Rule Suppression (PfSense)

4.1.2. Snort Supplemental Configuration on pfSense

By default, Snort on pfSense does not have port scanning enabled. Given that it is desirable to detect if a system is being used to scan the network, enable the “Use Portscan Detection to detect various types of port scans and sweeps.” on the “LAN Preprocs” tab. This type of configuration is useful if a suspect system is involved in network scanning.

4.1.3. Snort Supplemental Configuration on Security Onion

Rules and alerts may be overly chatty, or may not be applicable to a site’s environment.

One thing to do early on is to define the HOME and EXTERNAL network variable. To tune the Snort sensor, look at `/etc/nsm/$HOSTNAME-$INTERFACE/snort.conf`. The HOME_NET should be your internal network and EXTERNAL_NET should be “not Home”, or `!$HOME_NET`. There are other variables including DNS_SERVERS that should also be set. Once the IP addresses for the system types are defined run “`sudo rule-update`” to pull in current runes and restart the IDS processes.

As an example, on a busy home network, one particular rule was the most active over a 24-hour period. This rule proved to be triggered by a Roku TV appliance.

QUEUE	SC	DC	ACTIVITY	LAST EVENT	SIGNATURE	ID	PROTO	% TOTAL
15890	3	17		23:59:55	PROTOCOL-DNS TMG Firewall Client long host entry exploit attempt	19187	17	10.280%

Figure 6: Snort Alerts visible in Snort by Security Onion

With three changes to the snort configuration that defines `EXTERNAL_NET`, `HOME_NET` and `DNS_SERVERS`, the alert virtually disappeared. However, there are exceptions. To disable rules by Snort ID and prevent Snort from evaluating them, edit `“/etc/nsm/pulledpork/disablesid.conf”` and add SID’s that should be disabled. The Snort ID for the rule cited above is `“3:19187,”` so that number belongs on a single line in the `disablesid.conf` file.

4.1.4. Manually Updating Rules on Security Onion

Security Onion uses automation to periodically pull in rule updates at 7:01 AM daily. If you want to maintain your own snort rules, edit the `“/etc/nsm/rules/local.rules”` file (an example is covered in “Dark Network Detection” on page 27). To update snort, and retrieve rules, run `“sudo rule-update.”` Assuming that `eth1` is your monitoring interface, review the `“snortu-1.log”` log in the `“/var/log/nsm/SecurityOnion-eth1”` directory for any error conditions. Be aware that cutting and pasting rules from Windows applications, such as Word, into a Linux text file being edited with a SSH application like putty may work visually, but adds Unicode characters which will cause rule compilation or message output errors. An example of this is show in “Figure 24: Snort DarkNet Scan Alerts in ELSA” on page 28.

4.2. Packet Collection using Security Onion

Security Onion uses `netsniff-ng` to write Pcap data, organized by date, to the `“/nsm/sensor_data/seconion-eth1/dailylogs”` folder. This folder and file structure is used by all Security Onion services. Default file rotation is at about 150 MB. The system can be tuned by editing the global BFP file stored here: `/etc/nsm/rules/bpf.conf`. The same syntax used for `tcpdump` is used in BPF tuning, such as `“!(host 1.2.3.4)”` to exclude a specific host for collection. In

Don Murdoch, GSE – donmrdch@gmail.com

order to tune the BPF files, it is best to run tcpdump, ensure that the right BPF filter is collecting the specific data that needs to be filtered out, and *then* edit this file. By default, all components use this file and are symlinked to it.

5. Network Security Monitoring Use Cases

There are two main phases in the six-step incident response process that this solution supports – Identification and Containment. Other phases are supported, of course, but these are the two primary phases.

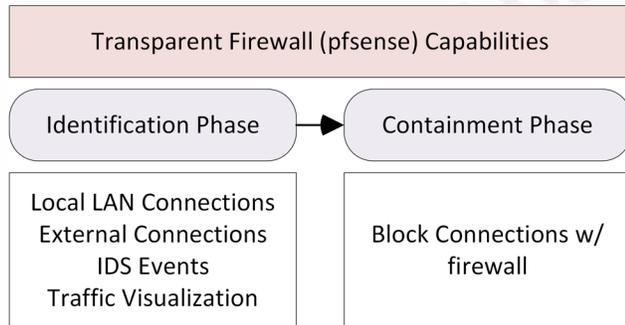


Figure 7: pfSense Incident Response Support

5.1.1. Local and External LAN Connections: pfSense Firewall Logs

From pfSense, firewall log data can be shown by filtering on the source suspect IP (192.168.2.118) and all destinations that are not in the first two octets of this RFC1918 address range.

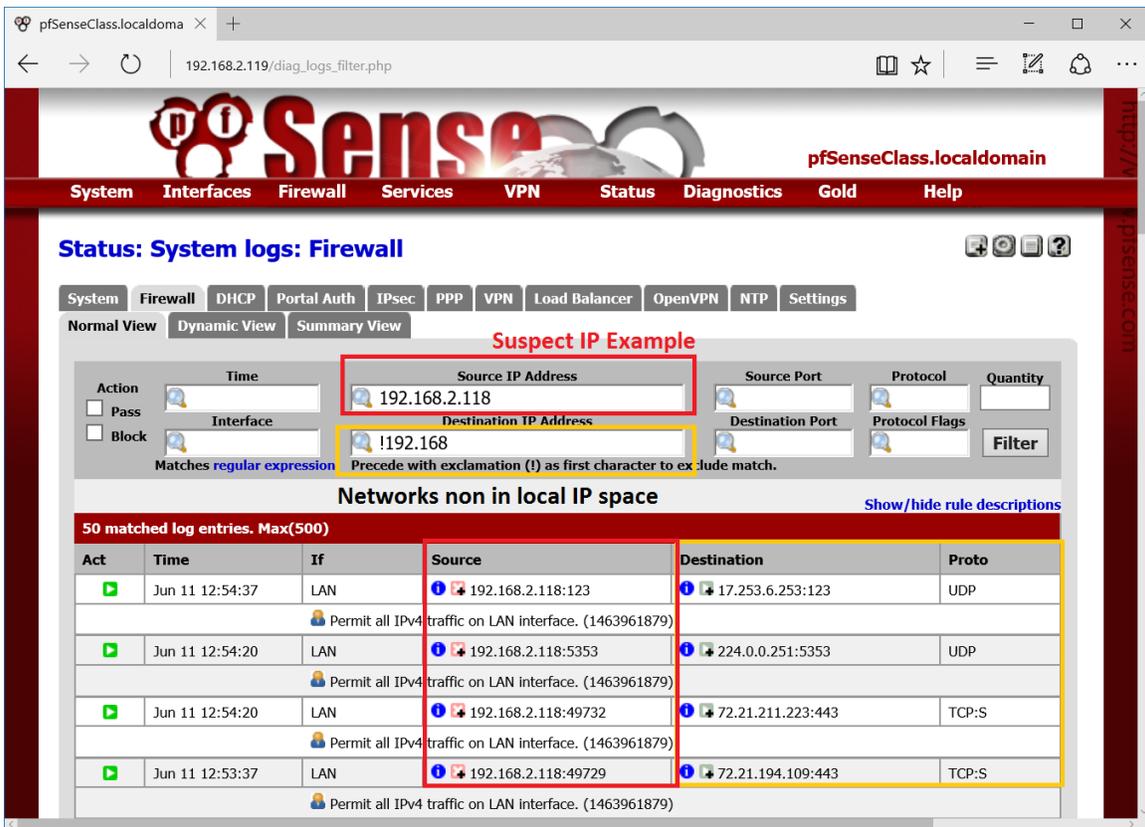


Figure 8: Firewall logs from suspect to external

5.1.2. Local and External LAN Connections: ntop Flow Data

ntop is a network traffic probe for visual traffic analysis, usage, protocol activity, statistics, application flows, and general analysis. For this solution, the ntop package on the pfSense firewall should be configured to monitor the “LAN” interface, perform DNS resolution, and to store historical data. These settings limit analysis to the suspect, provides more usable system names, will also store recent historical data *in memory*, which is stored for the last hour.

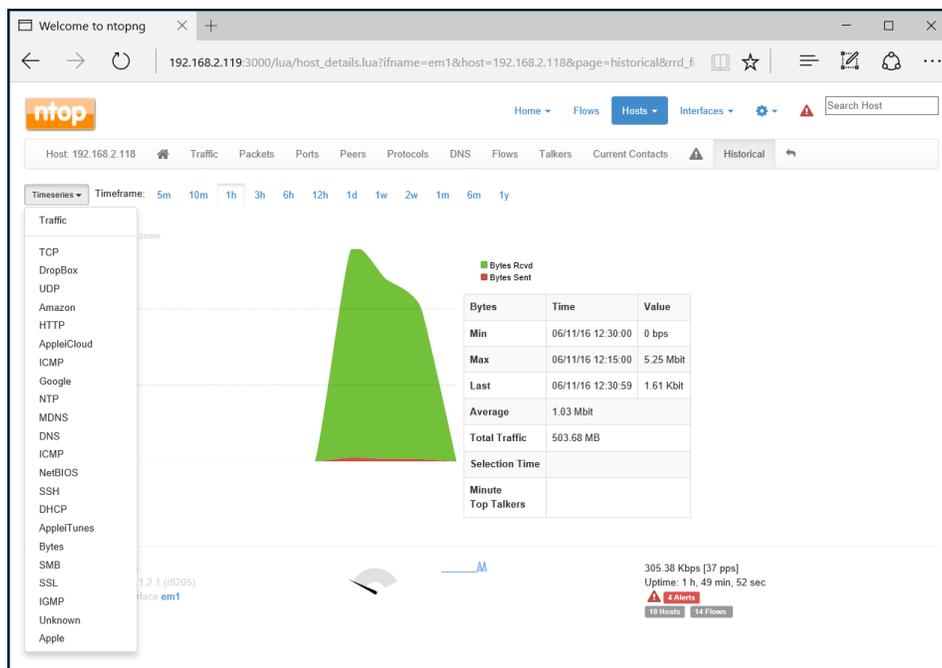


Figure 9: ntop flow history

From the protocol inventory above, the drop box activity may indicate data exfiltration, or other forms of offsite storage. By selecting “DropBox”, changing the time window to a longer time, 3 hours for example, (3h), and then placing the mouse pointer over the highest point, it can be shown that very little data was moved to/from the suspect - 193K with a low rate. *Most likely*, this amount of data exchange indicates a DropBox client was making an initial connection and checking replication status.

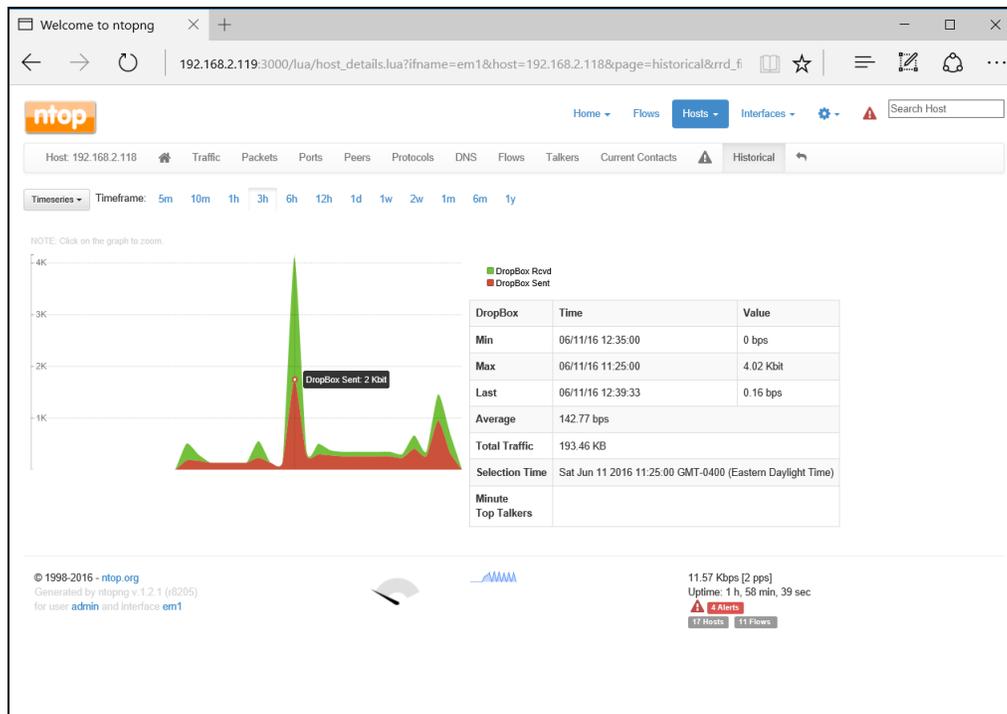


Figure 10: ntop DropBox activity example (pfSense)

ntop Can Fill the Disk! One caution with regard to ntop must be noted. If ntop is configured to record historical data, and a wide amount of scanning or network activity occurs over the observed interface(s), then it is quite possible to fill up the disk. While testing the snort functionality described in “Dark Network Detection” on page 31, this condition manifested itself. pfSense services will fail, the UI becomes nearly unusable, and any change made will likely corrupt the configuration file. In order to remedy this condition, reboot the pfSense VM, and quickly choose ‘single user’ from the pfSense boot menu. Next, remount the file system as writeable with the command “mount -uw /”. Next navigate to “/var/db/ntopng/0” and run “du -sh *”. Most likely, the ‘rrd’ directory will be the culprit in filling up the disk. There are directories by IP address stored here. To clear them out, run a command like this:

```
for outer in `seq 1 255`
do
    for inner in `seq 1 255`
    do
        rm -rf 192.168.$outer.$inner
    done
done
```

Don Murdoch, GSE – donmrch@gmail.com

done

5.1.3. Local and External LAN Connections: IDS Events

The IDS events produced by snort are viewable from with the pfSense UI itself. For this solution, since pfSense should have a local LAN IP address, others aside from the user at the console can view IDS events. In addition, pfSense provides a pushbutton download capability for any IDS detection so an incident responder can preserve supporting data for their case. Being able to download timestamped rules is an important capability to develop incident timeline.

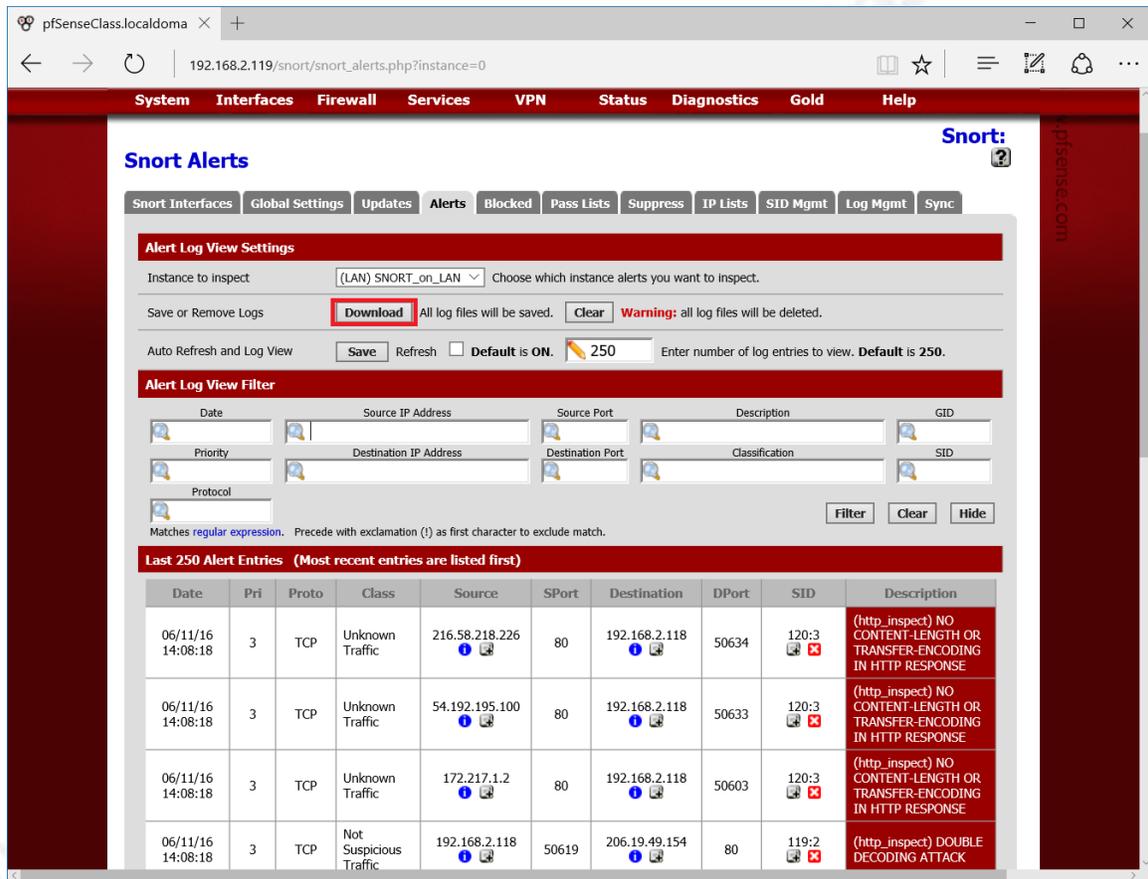


Figure 11: Snort alert example (pfSense)

5.1.4. DNS Requests by Host using Bro and ELSA

To view DNS activity and limit it to a specific host, use a query in ELSA like this: `class=BRO_DNS dstport="53" BRO_DNS.srcip=192.168.1.111`

Don Murdoch, GSE – donmrch@gmail.com

DNS activity may be very revealing, and is useful to document what a system did on the network. From this query, click “Result Options,” and export to Excel. From there, a rich set of timestamped DNS query and response data will be available.

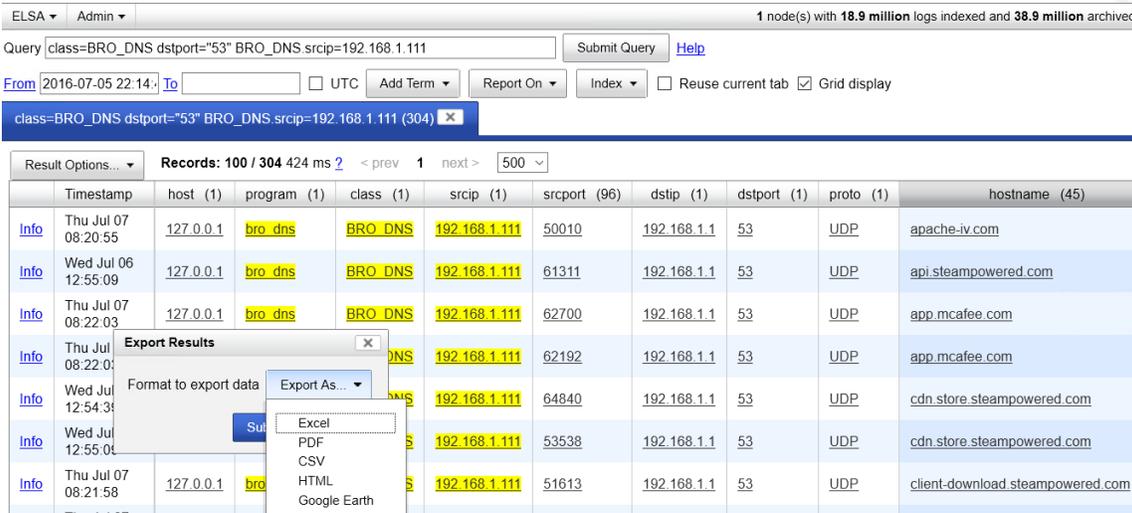


Figure 12: Bro DNS Query and response decodes for a Specific Source IP

Other DNS queries:

To see the hosts that answered DNS queries: `class=BRO_DNS dstport="53" groupby:hostname`

To see the answers to DNS queries: `class=BRO_DNS dstport="53" groupby:answer`

5.2. IP Communications – General Patterns

There are some general use cases for IP communications. Below are some more examples that are designed to help an analyst. The queries are adapted from one of Doug Burk’s presentations on Security Onion.

Query: `Class=BRO_CONN icmp or tcp or udp groupby:srcip`

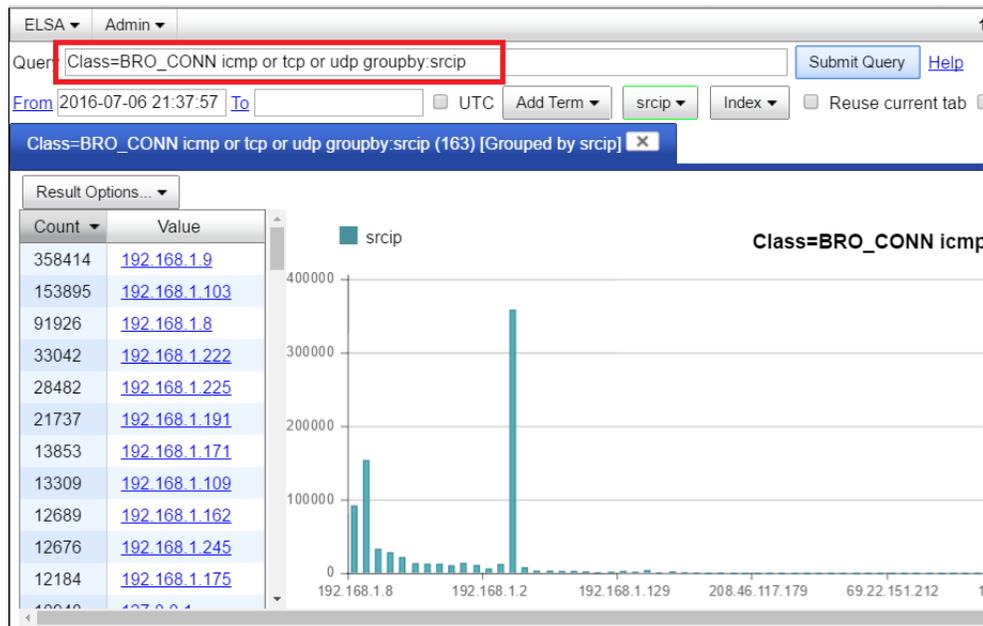


Figure 13: Summary Connection Data from Bro IDS

The next step is to refine the search, or search for a specific address:

Query: `Class=BRO_CONN icmp or tcp or udp srcip=192.168.1.103 groupby:dstip`

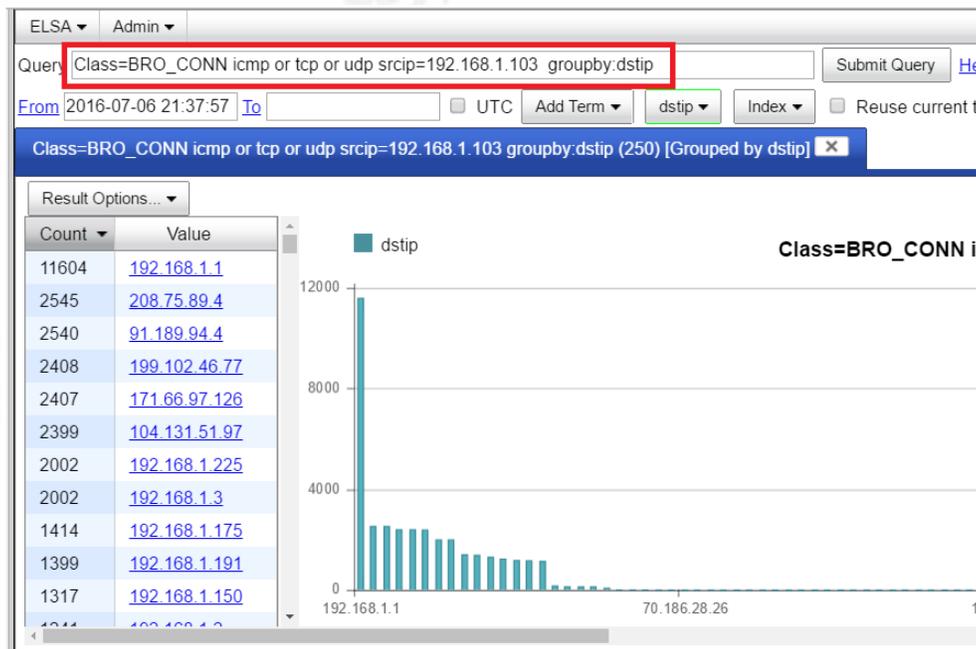


Figure 14: Bro Connection Data for a specific source, group by Destination IP

The next step would be to determine which sensor component has provided data about a specific IP address:

Don Murdoch, GSE – donmrch@gmail.com

Query: groupby:program srcip=192.168.1.103

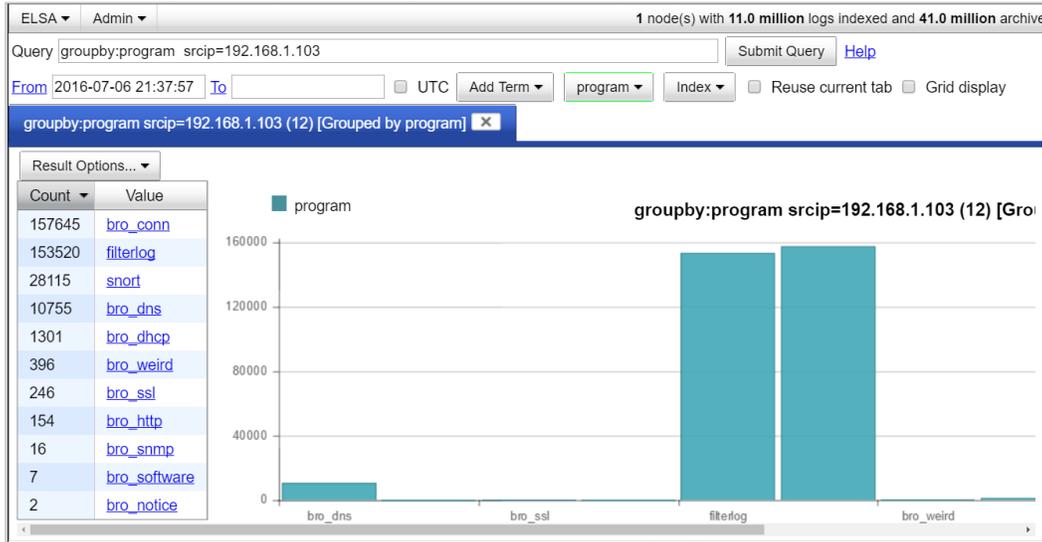


Figure 15: Security Onion Sensor Component Data Summary

A query can be formed for each of the program types. One interesting query is software detection (shown below). For each to the other program types, a similar query can be created to review what that sensor component identified.

Timestamp	Fields
Thu Jul 07 21:46:22	1467942380.908611 192.168.1.103 -HTTP::BROWSER Chrome 50 0 2661 102 -Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/50.0.2661.102 Chrome/50.0.2661.102 Safari/537.36 host=127.0.0.1 program=bro_software class=BR0_SOFTWARE srcip=192.168.1.103 srcport=0 version_major=50 version_minor=0 version_minor2=2661 version_minor3=102 software_type=HTTP_BROWSER name=Chrome version_addl= version=Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/50.0.2661.102 Chrome/50.0.2661.102 Safari/537.36
Fri Jul 08 17:21:33	1468012891.623610 192.168.1.103 22 SSH::SERVER OpenSSH 6 6 1 -jp1 OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.7 host=127.0.0.1 program=bro_software class=BR0_SOFTWARE srcip=192.168.1.103 srcport=22 version_major=6 version_minor=6 version_minor2=1 version_minor3=0 software_type=SSH_SERVER name=OpenSSH version_addl=p1 version=OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.7
Fri Jul 08 18:48:10	1468018089.691037 192.168.1.103 -HTTP::BROWSER Debian APT-HTTP 1 3 -1.0.1ubuntu2 Debian APT-HTTP/1.3 (1.0.1ubuntu2) host=127.0.0.1 program=bro_software class=BR0_SOFTWARE srcip=192.168.1.103 srcport=0 version_major=1 version_minor=3 version_minor2=0 version_minor3=0 software_type=HTTP_BROWSER name=Debian APT-HTTP version_addl=1.0.1ubuntu2 version=Debian APT-HTTP/1.3 (1.0.1ubuntu2)
Fri Jul 08 19:47:53	1468021672.162475 192.168.1.103 22 SSH::SERVER OpenSSH 6 6 1 -jp1 OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.7 host=127.0.0.1 program=bro_software class=BR0_SOFTWARE srcip=192.168.1.103 srcport=22 version_major=6 version_minor=6 version_minor2=1 version_minor3=0 software_type=SSH_SERVER name=OpenSSH version_addl=p1 version=OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.7
Fri Jul 08 20:02:30	1468022548.884223 192.168.1.103 -HTTP::BROWSER Debian APT-HTTP 1 3 -1.0.1ubuntu2 Debian APT-HTTP/1.3 (1.0.1ubuntu2) host=127.0.0.1 program=bro_software class=BR0_SOFTWARE srcip=192.168.1.103 srcport=0 version_major=1 version_minor=3 version_minor2=0 version_minor3=0 software_type=HTTP_BROWSER name=Debian APT-HTTP version_addl=1.0.1ubuntu2 version=Debian APT-HTTP/1.3 (1.0.1ubuntu2)
Fri Jul 08 20:03:25	1468022604.412508 192.168.1.103 -HTTP::BROWSER Python-urllib 3 4 - - Python-urllib 3.4 host=127.0.0.1 program=bro_software class=BR0_SOFTWARE srcip=192.168.1.103 srcport=0 version_major=3 version_minor=4 version_minor2=0 version_minor3=0 software_type=HTTP_BROWSER name=Python-urllib version_addl= version=Python-urllib 3.4
Fri Jul 08 20:09:45	1468022984.575006 192.168.1.103 -HTTP::BROWSER Chrome 51 0 2704 79 -Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/51.0.2704.79 Chrome/51.0.2704.79 Safari/537.36 host=127.0.0.1 program=bro_software class=BR0_SOFTWARE srcip=192.168.1.103 srcport=0 version_major=51 version_minor=0 version_minor2=2704 version_minor3=79 software_type=HTTP_BROWSER name=Chrome version_addl= version=Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/51.0.2704.79 Chrome/51.0.2704.79 Safari/537.36

Figure 16: Bro IDS Software Detection

5.3. Consolidated Log View: ELSA

Each of the analysis sources such as Snort and Bro log data into ELSA. The first thing that an analyst should do is to query for the suspect IP by entering “srcip=192.168.2.118” in the query dialog and then submit.

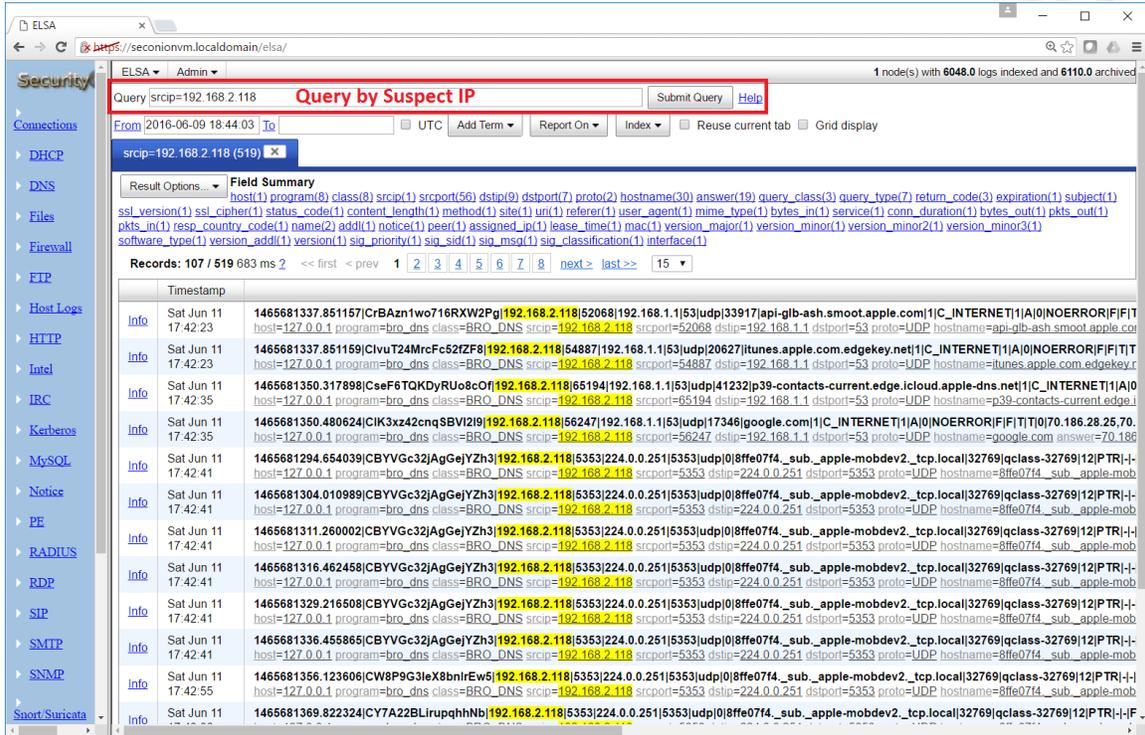


Figure 17: ELSA Consolidated view for suspect IP

Each of the hyperlink items is a group by term, which can then be subsequently exported. For example, to see the result set sorted by the reporting source, click on the “program (8)” link to see which analysis engine provided data into ELSA.

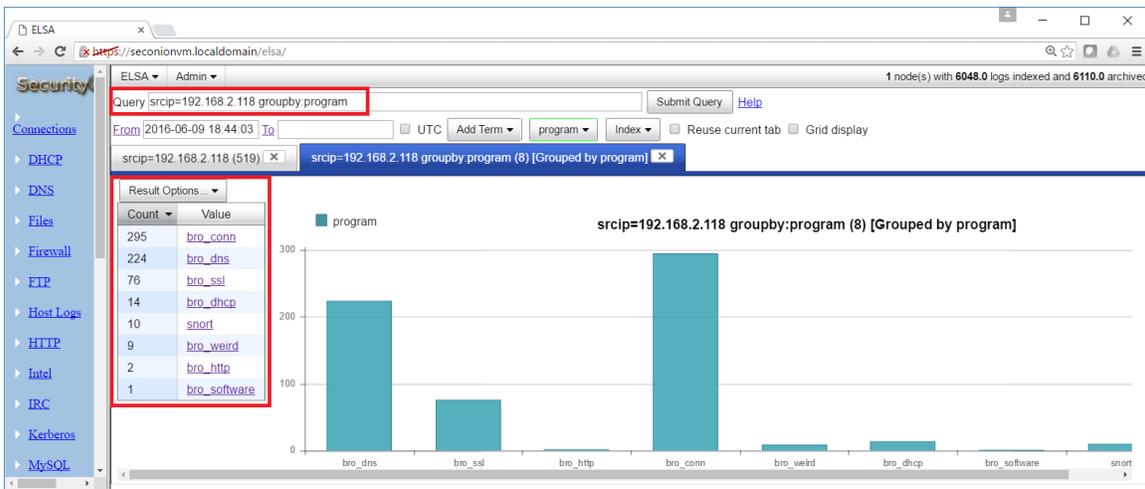


Figure 18: ELSA reporting view

Data can also be exported. From the original query, click on “hostname (30)” to see the DNS name queries that were analyzed by Bro, and reported on during the period.

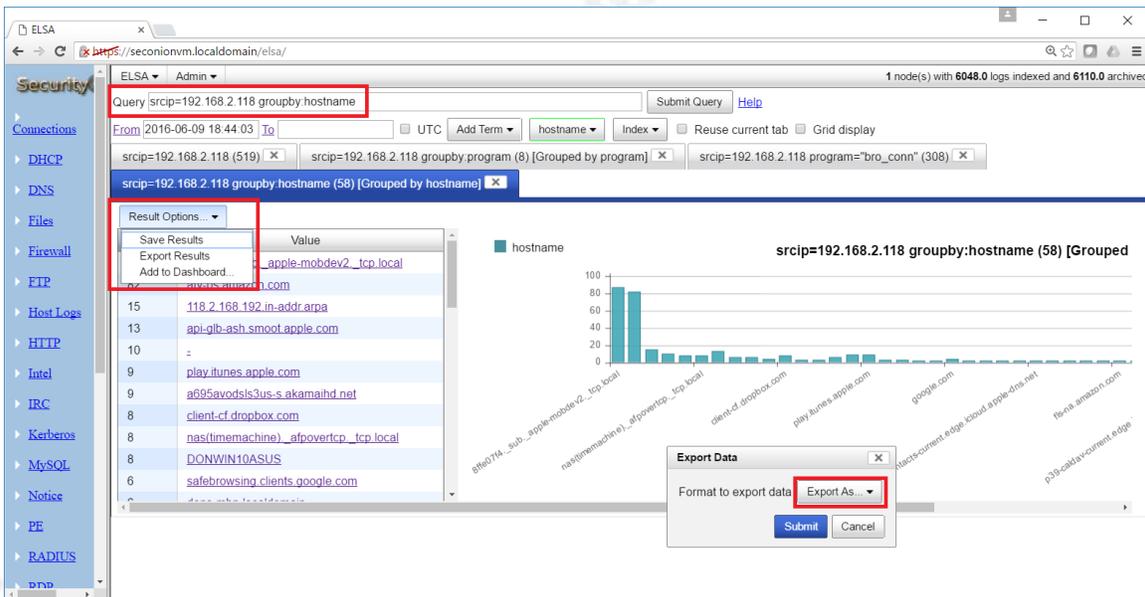


Figure 19: ELSA Data Export

Data can be exported to Excel, CSV, PDF or HTML.

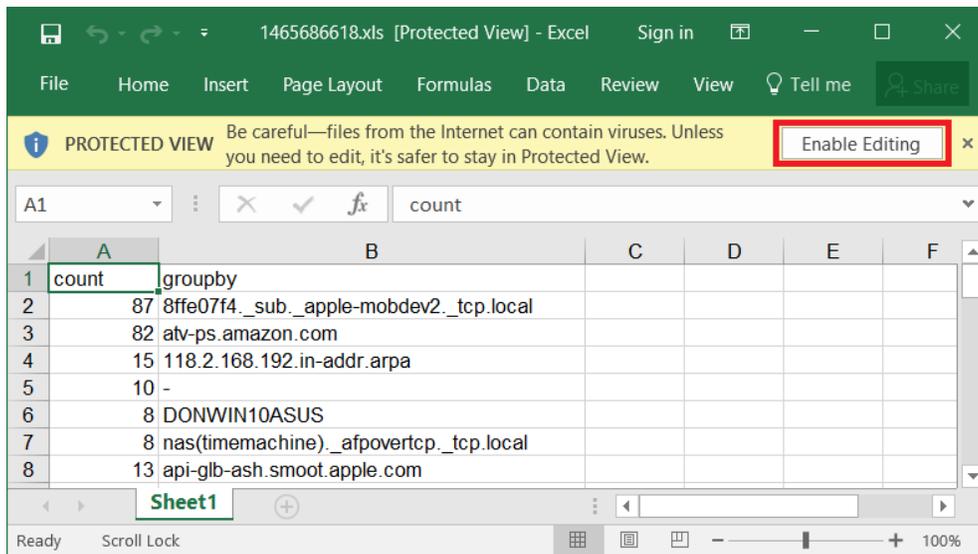


Figure 20: Example export from ELSA to MS Excel

ELSA also stores and can present firewall data. By searching for the TCP protocol and then grouping by the destination IP, it is very easy to show which IP's were most frequently connected to over the period.

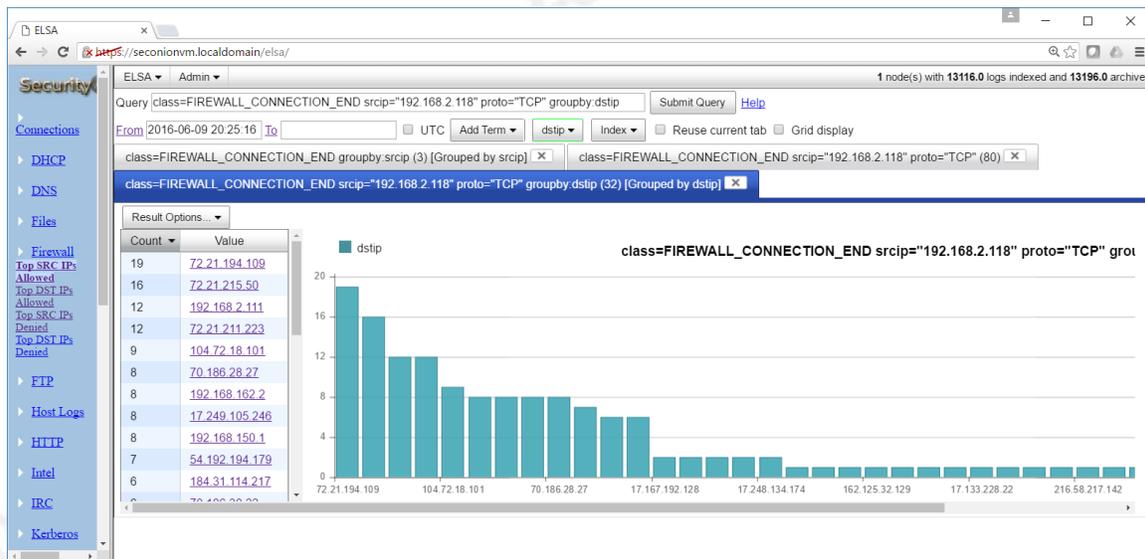


Figure 21: Example firewall data in ELSA

5.4. Pcap Data Extraction Tools and Methods

Security Onion uses `net-sniff-ng` to write full pcap files into a well-organized folder structure. These Pcap files are readable by any libpcap compatible

tool, such as tcpdump. File times are the time that the file was rolled over to the new file – the last “write” time. On the system used for this paper, a sample directory is:

```
/nsm/sensor_data/seconion-eth1/dailylogs/2016-07-09
```

Breaking this down, the system was named “seconion,” the Ethernet monitor interface is eth1, and the date was July 9, 2016. In order to extract a Pcap data set over a one-day period, the set of Pcap files can be processed and then “stitched” together. Below is a sample script that can be used from the ‘daily\DATE’ directory. This script goes through the current Pcap files and pulls out all TCP, UDP, and ICMP traffic to or from host 192.168.1.103 on a file by file basis, and then merges them into one:

```
#!/bin/sh
for file in `ls -a -u snort*`
do
    tcpdump -n -r $file -w /tmp/$file.ex "host 192.168.1.103
and (tcp or udp or icmp)"
    echo "Output file size: "`ls -lah /tmp/$file.ex |cut -f 5
-d' '`
done
mergcap -w /tmp/merged.pcap /tmp/snort*.ex
ls -lah /tmp/merged.pcap
```

One of the more useful things Bro does is to extract EXE files when it can observe them over a clear text network protocol. These are stored in /nsm/bro/extracted. A forensic analyst should always check here for executables. In the example below, the Pcap file snort.log.1465697489 was written a few minutes after the Bro tool extracted two EXE files, which it stores in /nsm/bro/extracted and can be identified as Windows executables using the ‘file’ command. By understanding the file layout and how to read the file’s time, a forensic analyst can quickly narrow focus on Pcap data at a particular event time, and pull out network trace data.

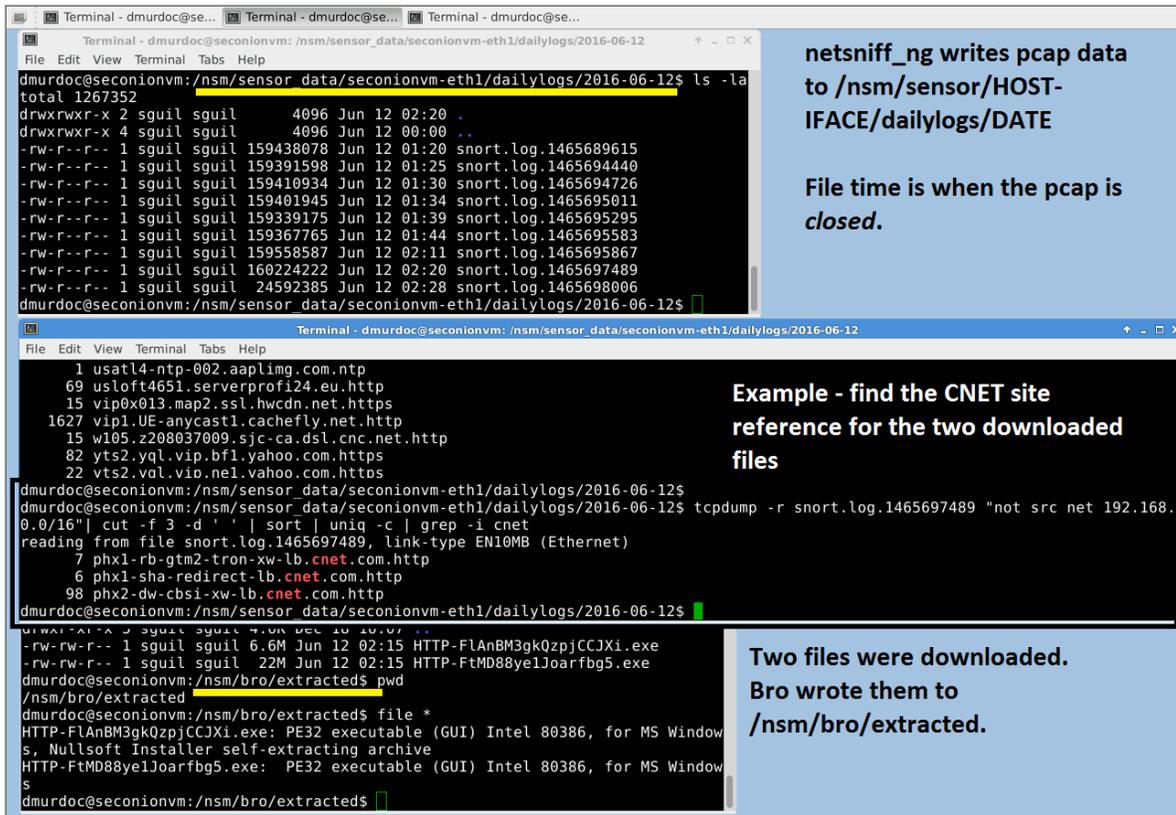


Figure 22: Security Onion PCAP Data and Bro File Extraction

5.5. Dark Network Detection

Most organizations have network segments that are unused. If a suspect system is observed communicating to local dark nets, it is likely up to no good. Detection rules for these networks are not in the downloadable rule sets, because they would be almost useless, quickly disabled, and always have to be specifically configured for a site’s IP network layout. In the case of the test network, a few “dark nets” are defined and a few unused RFC 1918 ranges in order to have something observable:

```
ipvar DARK_NET [192.168.7.0/24, 192.168.8.0/24,
192.168.8.0/24, 10.0.0.0/8, 172.16.0.0/16]
alert udp HOME_NET any -> $DARK_NET any (msg:"UDP to
defined DarkNet"; classtype: attempted-recon; sid:1000020;
rev:1;)
alert tcp HOME_NET any -> $DARK_NET any (msg:"TCP to
defined DarkNet"; classtype: attempted-recon; sid:1000021;
rev:1;)
```

```
alert icmp HOME_NET any -> $DARK_NET any (msg:"ICMP to
defined DarkNet"; classtype: attempted-recon; sid:10000202;
rev:1;)
```

To test rules like these, run a command like “nmap 192.168.0.0/16” and “nmap -sP 192.168.0.0,” wait a few minutes, and then check the Sguil interface. Change the IP range for the actual network in use, of course!

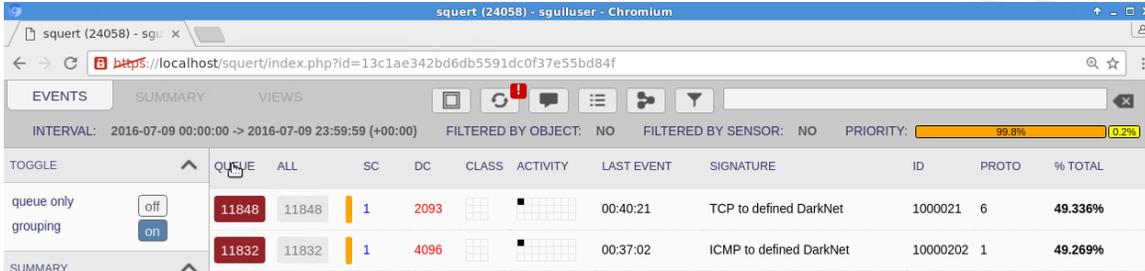


Figure 23: Snort DarkNet Scan Alerts in Sguil

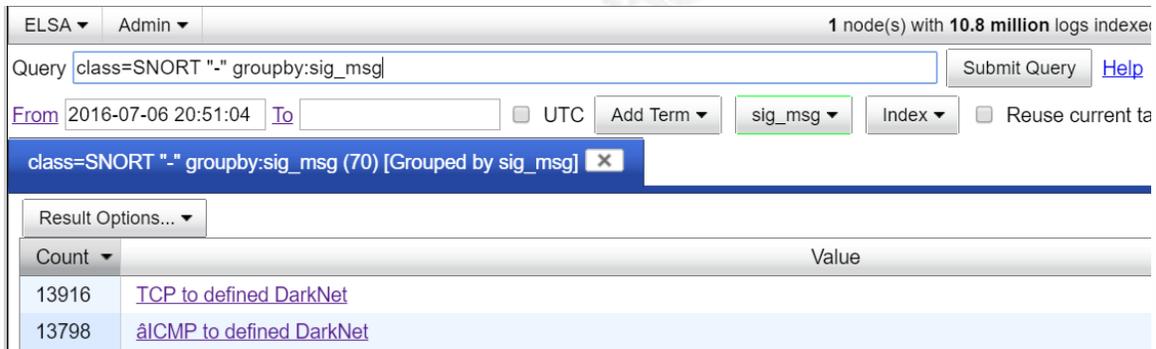


Figure 24: Snort DarkNet Scan Alerts in ELSA

Notice the odd text in the “ICMP to defined DarkNet” message field. Extraneous characters appear when rules are copied from Windows word processing programs and then pasted in a shell window – it is best to type a rule in, or to copy a file over and use “dos2unix” to remove Unicode characters.

5.6. Network Access Control Use Case

The firewall can be used to limit and control access going through it very discretely. For example, HTTP traffic can be stopped while DNS and DHCP traffic are permitted. The rules below, in order, block HTTP, allow SSH, DNS, and HTTP. Because pfSense processes rules in top down order, the first “block and log” HTTP

Don Murdoch, GSE – donmrdch@gmail.com

traffic rule overrides the allow HTTP, and then allow all rules. An analyst can add a block rule to the rule set, then move it to the top of the list, reload the rules, and then stop any offending traffic while permitting other services to operate on the suspect.

ID	Proto	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description
	*	*	*	LAN Address	80 22	*	*		Anti-Lockout Rule
	IPv4 TCP	*	*	*	80 (HTTP)	*	none		Block HTTP on the LAN Interface.
	IPv4 TCP	*	*	*	22 (SSH)	*	none		Allow SSH on the LAN interface.
	IPv4 TCP/UDP	*	*	*	53 (DNS)	*	none		Permit DNS traffic on LAN interface.
	IPv4 TCP	*	*	*	80 (HTTP)	*	none		Permit HTTP on the LAN Interface.
	IPv4 TCP/UDP	*	*	*	67 - 68	*	none		
	IPv4 *	*	*	*	*	*	none		Permit all IPv4 traffic on LAN interface.

Figure 25: pfSense Block and Accept ruleset example

Output:

Act	Time	If	Source	Destination	Proto
✗	May 26 14:46:27	LAN	192.168.1.178:49629	104.72.9.115:80	TCP:S
✓	May 26 14:46:28	LAN	192.168.1.178:55959	192.168.1.1:53	UDP
✗	May 26 14:46:28	LAN	192.168.1.178:49630	131.253.33.50:80	TCP:S
✗	May 26 14:46:31	LAN	192.168.1.178:49630	131.253.33.50:80	TCP:S
✗	May 26 14:46:33	LAN	192.168.1.178:49629	104.72.9.115:80	TCP:S
✓	May 26 14:46:36	LAN	192.168.1.178:62017	192.168.1.1:53	UDP
✗	May 26 14:46:37	LAN	192.168.1.178:49630	131.253.33.50:80	TCP:S
✓	May 26 14:46:46	LAN	192.168.1.178:55152	192.168.1.1:53	UDP
✗	May 26 14:46:46	LAN	192.168.1.178:49631	104.72.9.115:80	TCP:S

Figure 26: Block Rule logging example

On the client system, a DNS lookup returns the IP addresses, but the browser is non-responsive:

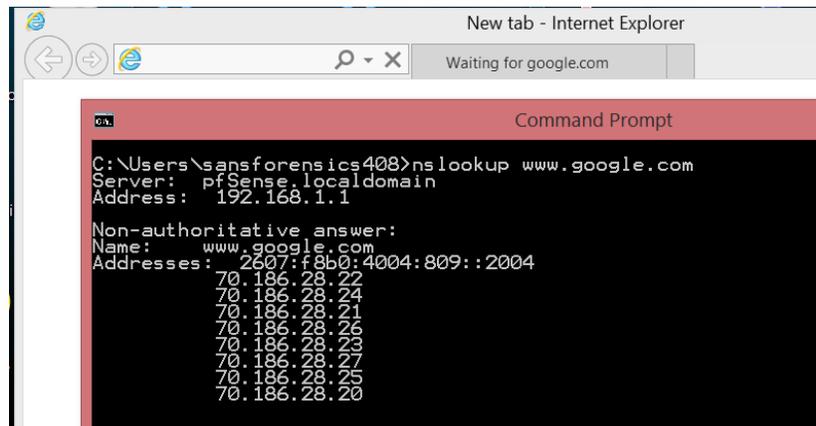


Figure 27: DNS functional, Web Browser is not

Appendix A: Configurations

A.1 Lab Configuration with VMware and Windows 10

There are specific techniques presented can be used to build this solution configuration for a lab environment, as well. Here, a typical PC can become a full-fledged isolation environment with the addition of an inexpensive PCI/E dual port LAN card. The configuration is illustrated in “Figure 28: VMware Workstation, Switch and pfSense Configuration,” and explained below. The “production switch” is the uplink to the production environment. The suspects, or lab systems, are all connected to a separate switch with a mirror port enabled so that any data for the suspects can be presented to the NSM platform for data collection. The suspect side of the firewall connects to the isolation switch, which is also where any suspects are plugged in. The VM network assignments match several screen captures below.

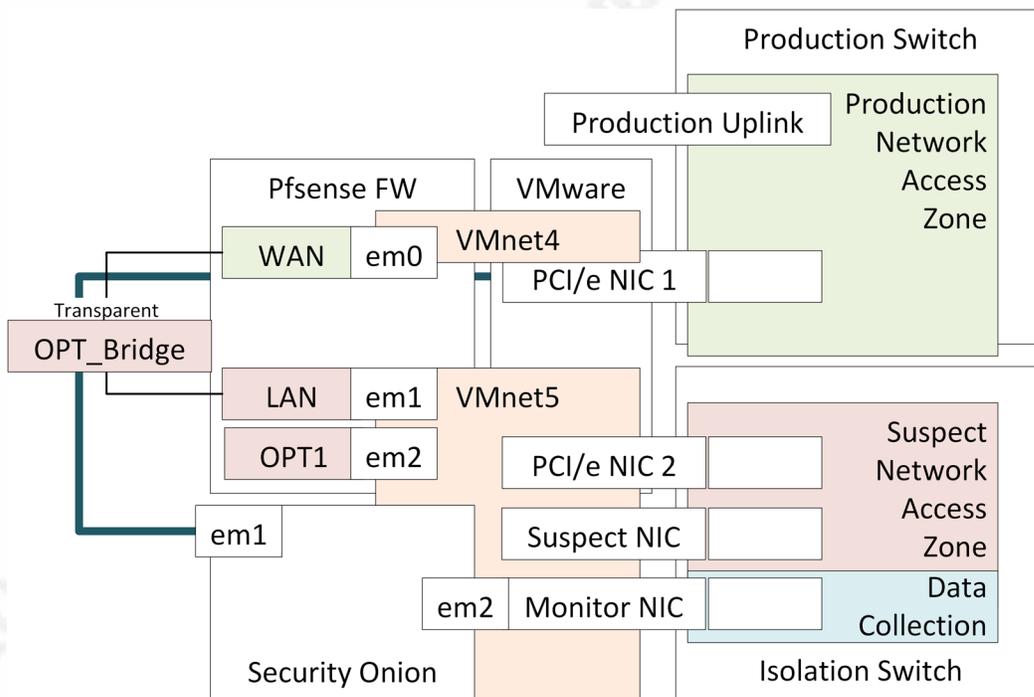


Figure 28: VMware Workstation, Switch and pfSense Configuration

A.2 Mainboard Ethernet and PCI Add on Card Configuration

For the sake of simplicity, keep the mainboard Ethernet connection as the primary LAN connection. Use the LAN add-in card connection for pfSense and Security Onion. It is also useful to rename the ports on the add-in card.



Figure 29: Windows 10 Network Connections renamed to represent configuration details

Only bind the PC’s Windows services and IPv4 / IPv6 services to the primary Ethernet card (Properties tab), and the VMware Bridge Protocol to the add-in card.

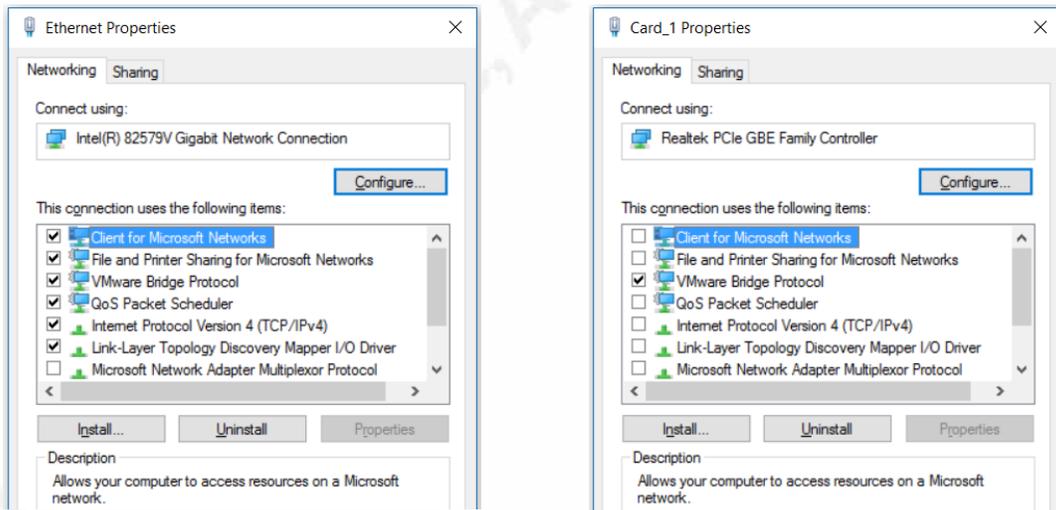


Figure 30: Windows 10 Adapter Properties

A.3 VMware Workstation Pro 12 VMnet Configuration

VMware should be configured to map physical network interfaces to physical cards in order to prevent any possible mismatch. In order to do this, VMNET0 needs to be specifically assigned to the mainboard Ethernet controller, and then other VMnet interfaces can be assigned. The example here is VMnet4 and VMnet5, bound

to ports 1 and 2 on the add-in card. During development of this solution, the network adapter was changed from a USB adapter, to an Intel Proset dual NIC, and then to the RealTek adapter. So long as the VM's themselves used VMnet4 and 5, they operated properly.

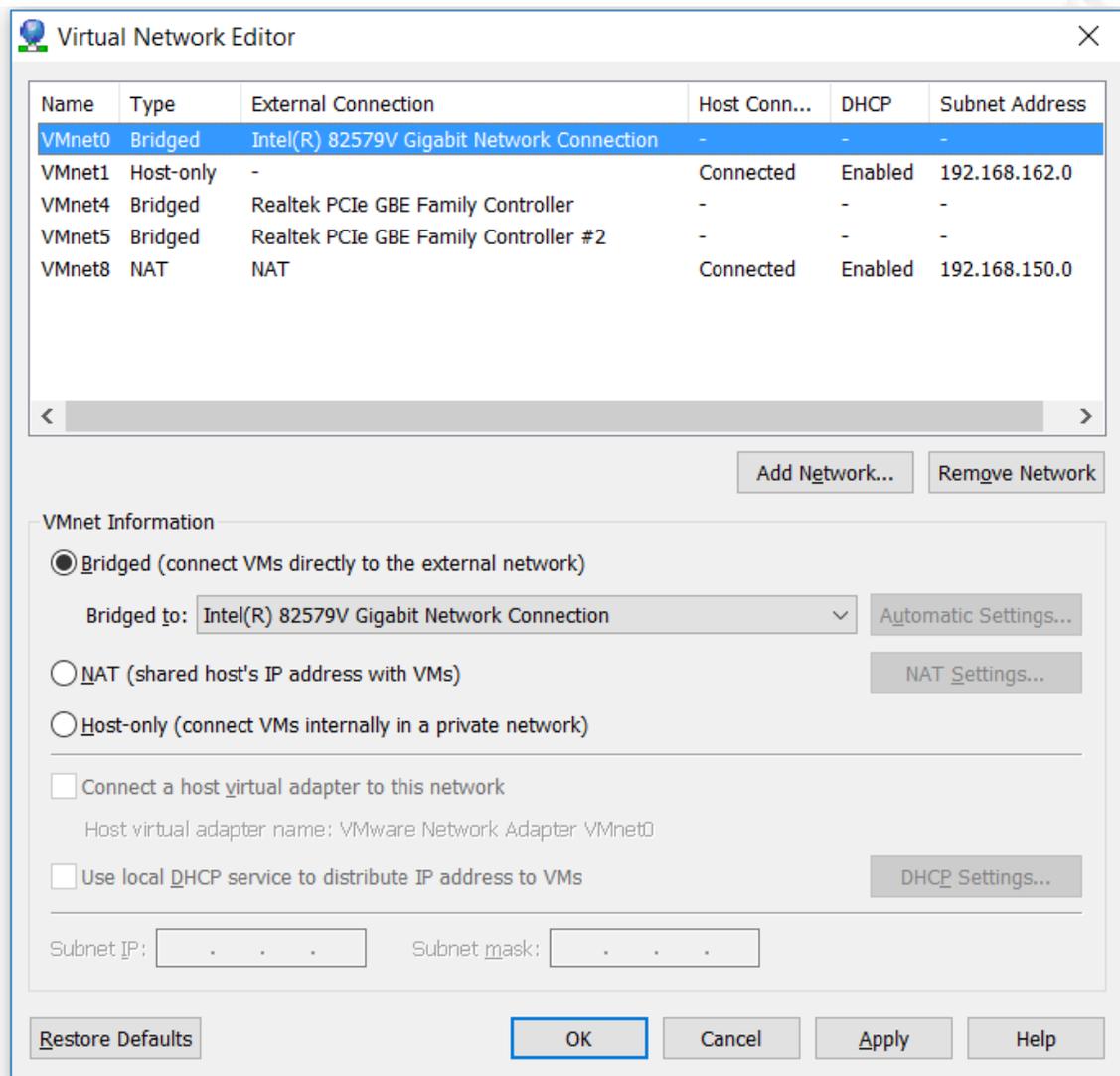


Figure 31: VMware Network Configuration, mapping to physical adapter

Once applied, the IP addresses that VMware assigned to the VMnet4 and VMnet5 adapters are visible. For the purposes of getting pfSense and Security Onion setup and configured, the WAN interface can be assigned an IP address on 192.168.231.0/24 and the LAN interface can be assigned an IP address on 192.168.214.0/24.

Don Murdoch, GSE – donmrdch@gmail.com

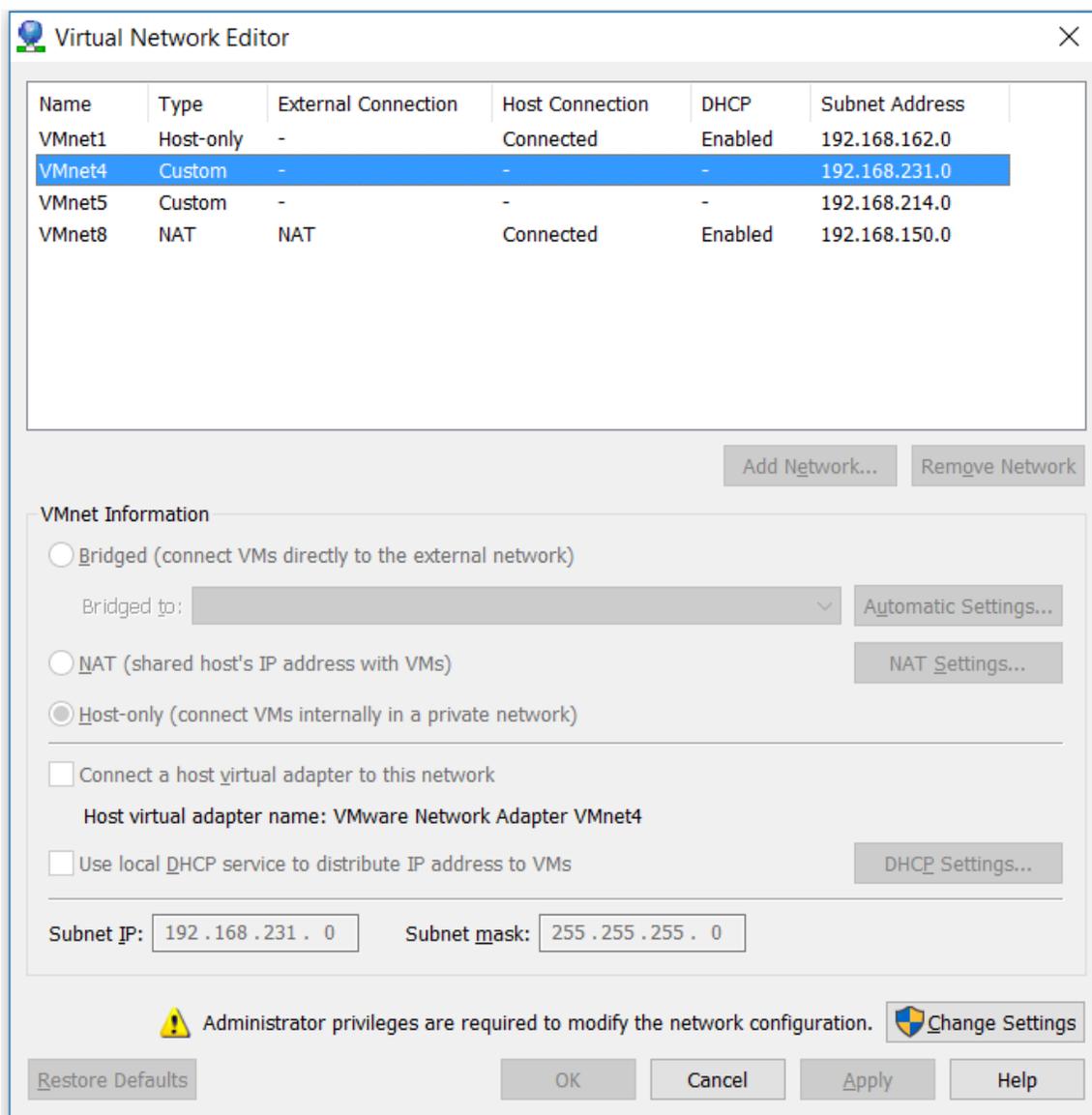


Figure 32: VMware network configuration with IP ranges

A.4 pfSense Configuration with VMWare Pro 12

pfSense should be set to VMnet4 for the first network interface and VMnet5 for the second interface. In this configuration, there is a third interface defined so that pfSense can be configured for another purpose, such as setting the Web interface to a host only network.

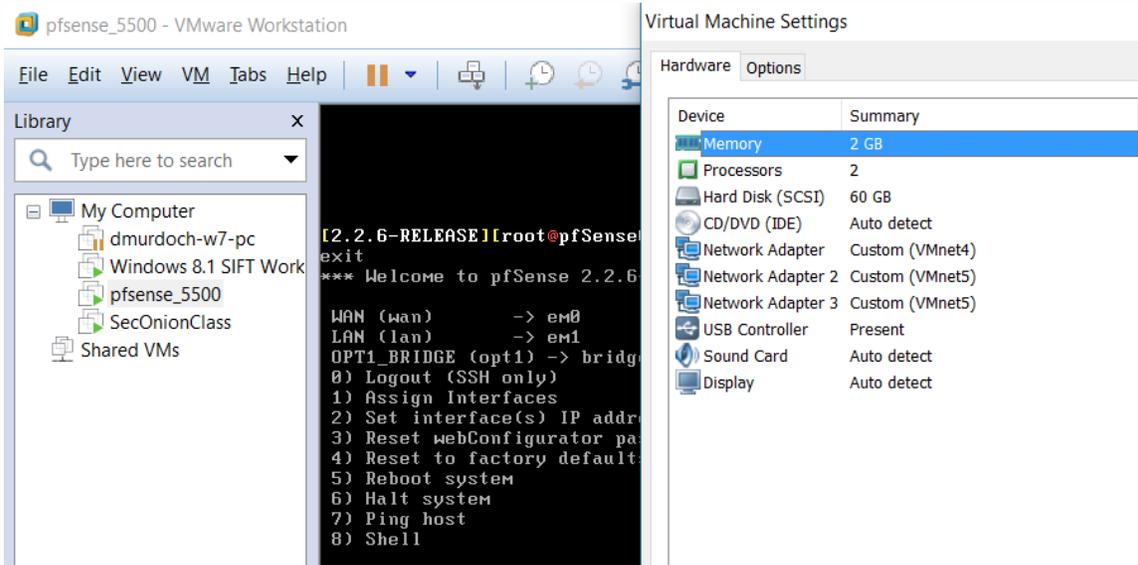


Figure 33: pfSense mapping to virtual adapters

The first and second interfaces match up with the WAN and LAN interfaces, defined within pfSense itself. The bridge is a logical construct.

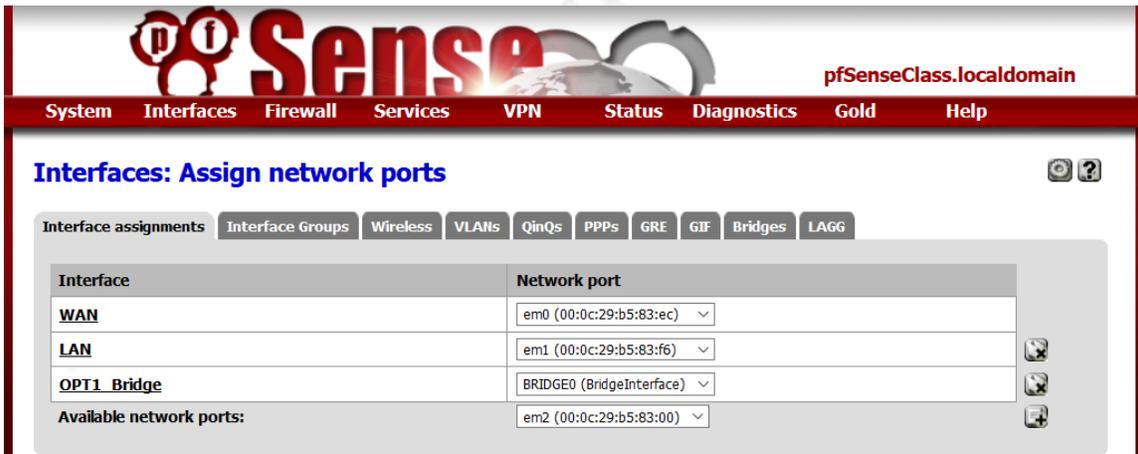


Figure 34: pfSense Interface mapping to LAN adapters

A.5 DLink DGS-1100-08 Configuration Port Mirroring

The “D-Link 8-Port EasySmart Gigabit Ethernet Switch (DGS-1100-08)²” switch is an inexpensive desktop and fanless switch which implements port mirroring capabilities, similar to an enterprise switch, and can be taken in the field

² \$49.66 as of 05/26/16 at Amazon: http://www.amazon.com/D-Link-EasySmart-Gigabit-Ethernet-DGS-1100-08/dp/B008ABLU2I/ref=sr_1_1?s=pc&ie=UTF8&qid=1464479585&sr=8-1&keywords=dgs-1100-8

in a jump kit and deployed to monitor a suspect system. Note that *any change* made to the DLink must be saved using the “Save | Save Configuration” option from the main menu bar in order for the change to persist!

The Port Mirroring feature will copy data from one port to another. For this solution, all of the data from the *suspect* side of the firewall should be copied to the *monitor port* configured as a “receiver,” where Security Onion *monitor interface* (*en1*) will be plugged in. As an example, the suspect system would be connected to port 6, with all “receive” data sent to port 8.

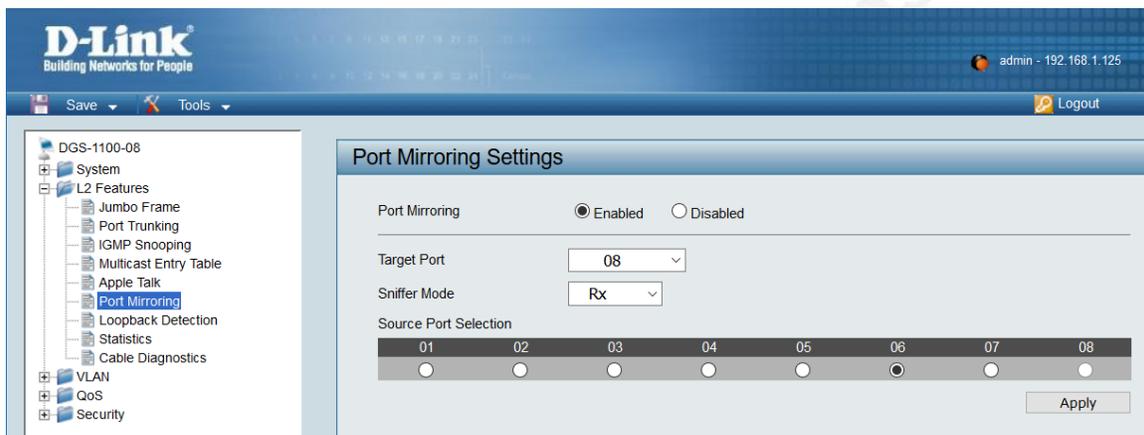


Figure 35: DLink Mirror configuration

Appendix B: Security Onion Architecture Diagram

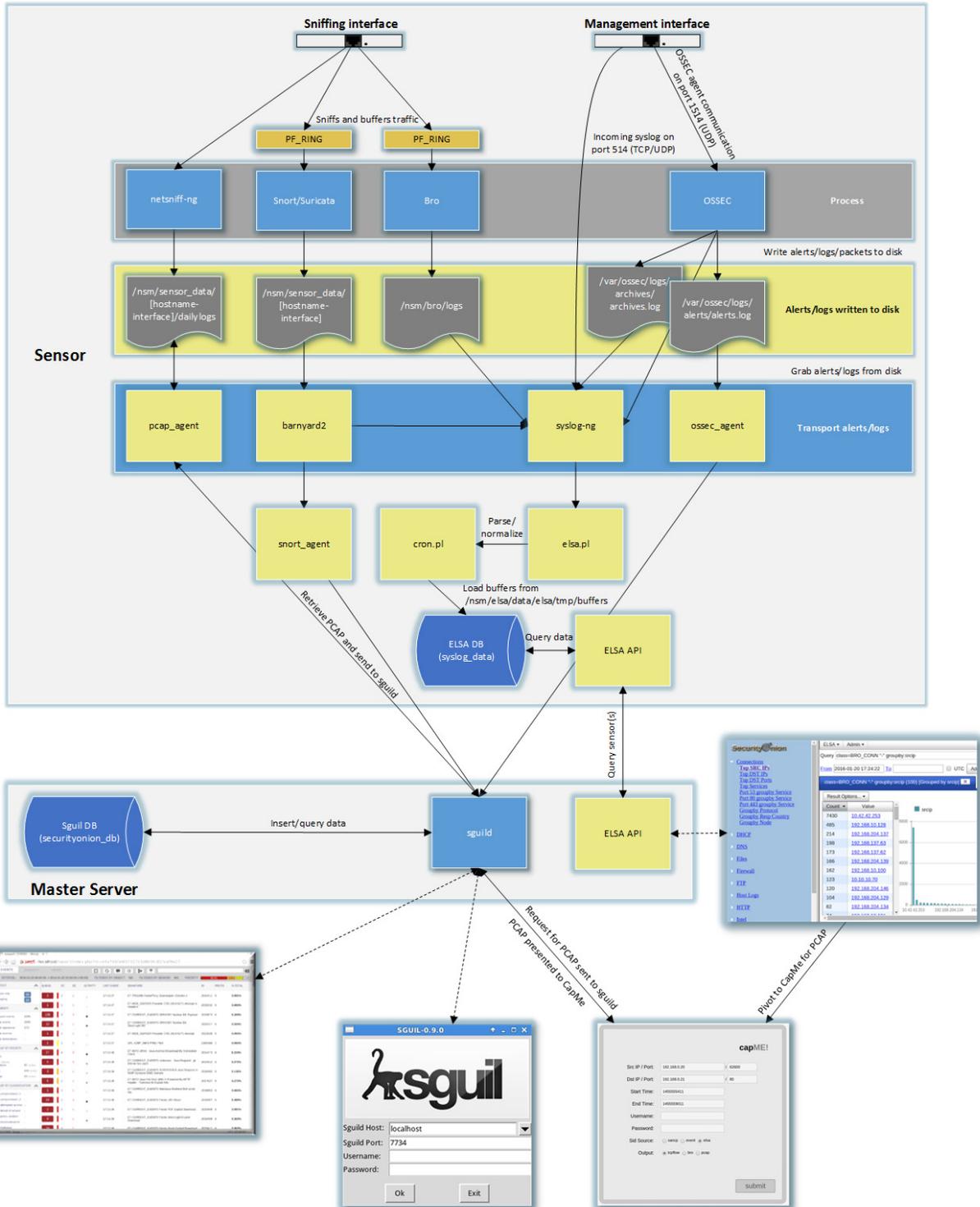


Figure 36: Architecture Diagram for Security Onion (Source: URL: <https://github.com/Security-Onion-Solutions/security-onion/wiki/Architecture>)

Don Murdoch, GSE – donmrch@gmail.com

Appendix C: Key Terms

There are several terms used to describe the components of the system, defined here to avoid confusion:

Forensically Sound Practice: The procedures used to acquire digital (electronic) information that ensures it can “prove the content of writing, recording, or photograph”. *For this solution, this means full content pcap network capture data.*

Network Security Monitoring: The ability to analyze traffic flows, perform full content capture and instrument other network analysis capabilities.

Production Zone: Network segments where business is conducted, and therefore must be protected from disruption.

Solution: The combined technical and procedural components of the system (software, hardware, cabling, practices and disciplines).

Stateful Firewall: A security device that can track, manage and control the operating state of LAN traffic traversing its interfaces.

Suspect Zone: The section of the network where a suspect resides; this is the broadcast and collision domain together for the suspect.

Suspect: A system under evaluation and analysis.

Transparent/Bridge Firewall³: A traditional firewall is a routed hop at Layer 3 (network addressing and routing) between networks and functions as a gateway. A transparent firewall functions at Layer 2, a “bump in the wire”, or a stealth device. It also does not change the Time to Live (TTL) value.

³ This definition is adapted from the Cisco ASA product documentation, as the pfPFSSense documentation does not have a formal definition.

References

Bejtlich, Richard (2013). *The Practice of Network Security Monitoring*, San Francisco: No Starch Press.

Burks, Doug (2016), Security Onion Architecture, Retrieved from <https://github.com/Security-Onion-Solutions/security-onion/wiki/Architecture> (06/17/16)

Burks, Doug (2016). Production Deployment, Security Onion. URL: <https://github.com/Security-Onion-Solutions/security-onion/wiki/ProductionDeployment> (6/15/16)

Burks, Doug, Security Onion Wiki, Security Onion Solutions. URL: <https://github.com/Security-Onion-Solutions/security-onion/wiki/IntroductionToSecurityOnion> (7/1/16)

CISCO (2015), Cisco ASA Series CLI Configuration Guide, 9.0. Retrieved from http://www.cisco.com/c/en/us/td/docs/security/asa/asa90/configuration/guide/asa_90_cli_config/intro_fw.html (05/26/16)

Dadbdk (2012). pfsense 2.0 transparent firewall / firewall bridge. PFSense Forum, URL: <https://forum.pfsense.org/index.php?topic=50711.0> (07/08/16)

Davidoff, Sherri. Ham, Jonathan (2012). Network Forensics: Tracking Hackers through Cyberspace. Upper Saddle Lake: Pearson Education, Inc.

Gray, Dggory, Setting up pfSense as a Stateful Bridging Firewall. URL: <http://users.ox.ac.uk/~clas0415/assets/Setting-up-pfSense-as-a-Stateful-Bridging-Firewall-with-commodity-hardware.pdf> (6/4/16)

Don Murdoch, GSE – donmrdch@gmail.com

Hagan, Ed, & Judish, Nathan. (2009). CCIPS DOCUMENTS AND REPORTS. Retrieved from <https://www.justice.gov/sites/default/files/criminal-ccips/legacy/2015/01/14/ssmanual2009.pdf> (05/26/16)

pfSense (2016). *pfSense Documentation*. Website: https://doc.pfsense.org/index.php/Main_Page (6/1/16)

Sanders, Chris. Smith, Jason (2014). *Applied Network Security Monitoring*, Waltham, Syngress.