Basic NGIPS Operation and Management for Intrusion Analysts

GIAC GCIA Gold Certification

Author: Mike Mahurin, mike.mahurin@aos5.com
Advisor: Adam Kliarsky

Accepted: August 1st 2017

Abstract

Next Generation Intrusion Prevention Systems (NGIPS) are often referred to as the panacea to modern malware, network intrusion, advanced persistent threat, and application control for complex modern applications. Many vendors position these products in a way that minimizes the value of tuning and intrusion analysis to get the optimum security capability of the solution. This paper will provide a guide for how to maximize the capabilities of these technologies by providing a basic framework on how to effectively manage, tune, and augment a NGIPS solution with Open Source tools.

1. Introduction

The complexity of network protocols, evolving complex of attacks, and increasing volume of malicious attacks has created a technology arms race. Next Generation Firewalls (NGFW) and Next Generation Intrusion Prevention (NGIPS) technologies have emerged as a minimum baseline control for the perimeter and internal network (Stuart & Beaver, 2013). These devices are characterized by having some form of Threat Intelligence (TI) for automated URL/DNS/IP blocking of bad reputation sites, Deep Packet Inspection (DPI) with an Intrusion Prevention Engine for application aware analysis, file analysis, and an analysis engine (Stuart & Beaver, 2013).

Modern Intrusion Prevention Systems technologies go beyond simple signature matching and include anomaly detection, statistical analysis, target monitoring, Denial of Service (DoS), and combination of these technologies to be effective (Kumar, Singh, & Jayanthi, 2016). NGIPS & NGFW technologies have reduced the workload of tuning these devices through data analysis, automated tuning, and event correlation which reduces the total cost of ownership of IPS solutions (SANS Institute, 2014). Workload reduction usually comes in the form of an analysis engine that performs event correlation analysis, endpoint finger printing, and network behavior baseline. These in turn produce Indicators of Compromise (IOC) based on multiple observed behaviors that provide a greater level of confidence of what events are significant (Datt, 2016).

These technologies certainly reduce the amount of time an intrusion analyst would spend tuning and responding to event. Advanced analysis and response by a skilled analyst is still needed to detect complex intrusion events (Anwar, et al., 2017). Complex server applications require extremely well written analysis and sanitizing technologies to be effective. Due to this limitation, multiple layers of controls are needed as the likelihood of a single solution providing adequate protection is low (Steven, 2016). This holds true for NGIPS in that these devices are still subject to issues such as false positives, false negatives, IPS evasion techniques, and may require tuning to block new and emerging threats.

A common technique that is seen in lab tests is often used in manufacturer marketing collateral that show their solution is 99% - 100% effective in blocking all threats in lab tests. The labor required to tune that device is often omitted, and the labor for ongoing management to reach high performance levels is not mentioned. Cisco provides a good example of this technique; they indicate their solution was rated nearly 100% effective against all exploits and common evasion techniques in NSS lab testing. References are not made to the effort required to maintain in a production environment (NSS Labs, 2016). Trend TippingPoint uses the same approach in their product as well (NSS Labs, 2016). The general sales and marketing for these products is to portray them as being self-tuning and 99%+ effective in stopping malicious traffic. Fortinet also uses the same approach in presenting their product (NSS Labs, 2015).

2. NGIPS Key Components

A NGIPS follows many of the same rules as a traditional IPS unit. The system must be positioned on the network where it is inline with relevant traffic for IPS mode or has traffic visibility only for Intrusion Detection System (IDS) mode. System availability should be implemented so there are redundant units or some form of bypass mechanism that will allow business traffic to fail open or fail closed based on the information security policy.

Appropriately scaling the solution is a much larger issue with NGIPS then it was with traditional IPS. Additional functionality and higher bandwidths drive higher resource consumption. Some technologies such as SSL decryption can introduce 70% or more system overhead in addition to normal functions. The end result is any NGIPS solution should be sized for the services that will be deployed or that may be deployed during the life of the solution. Failure to properly size the solution will result in performance issues that are difficult to identify and remediate.

Several key systems differentiate the NGIPS from a traditional IPS. These include some form of Threat Intelligence (TI), Application Control with Deep Packet Inspection (DPI), Intrusion Preventions System, File Analysis, and some form of analysis engine. These technologies work together to form multiple layers of specialized inspection and control.

Most solutions have a defined order of operations in which these technologies are applied. Usually, the first item that occurs is verifying an ACL has been implemented that will allow the traffic. This is a low system resource operation and most often is a binary comparison. The next layer is comparison to a vendor TI feed which will block based on IP address, DNS, or URL. This operation is a low system resource operation. Once traffic has passed, initial filtering DPI occurs to identify the application, general behavior, and anomalous behavior of the traffic. Control decisions can be made at this point to restrict applications the system understands. IPS is next in the series to identify malicious behavior based on application behavior and traditional IPS signatures.

The system analysis engine usually develops a baseline of what normal traffic on the network is, what endpoint operating systems are in use, and what application versions are actively being used. This is correlated with vulnerability database to develop Indicators of Compromise (IOC). These are multiple points of information that help identify and prioritize a given attack. This helps the analyst quickly identify which events should get priority attention and reduces the time it would take to manually collect this information.

2.1. Threat Intelligence Based Filtering

Threat Intelligence feeds take the form of a vendor provided blacklist of IP addresses, DNS names, and/or URL addresses that have been identified as malicious. These lists are targeted to be very low false positives and are typically updated every 5 – 30 minutes on the device. Most vendors categorize their block list into categories such as phishing, malware, Domain Generation Algorithm (DGA), Command and Control (C2), TOR Exit Nodes, and other categories.

Some vendors will utilize combinations of manual analysis, big data analysis across customers, neural networks, honeypots, finger printing, and behavior profiling to identify malicious IP addresses and DNS names on the Internet. Most solutions allow third party feeds to be installed to provide supplemental or specialized TI feeds. Geolocation capabilities also exist in most applications that allow IP addresses in specific geographic regions to be blocked.

The function of this layer is providing automated filtering for rapidly changing attacks and threats that can be easily identified by IP or DNS entry. Examples would be C2 traffic, attackers using Fast Flux techniques, common drive by download sites, and other rapidly changing threat locations. To be effective this technology must be regularly updated, automatically deployed, have a very low false positive rate, and have an effective vendor research team.

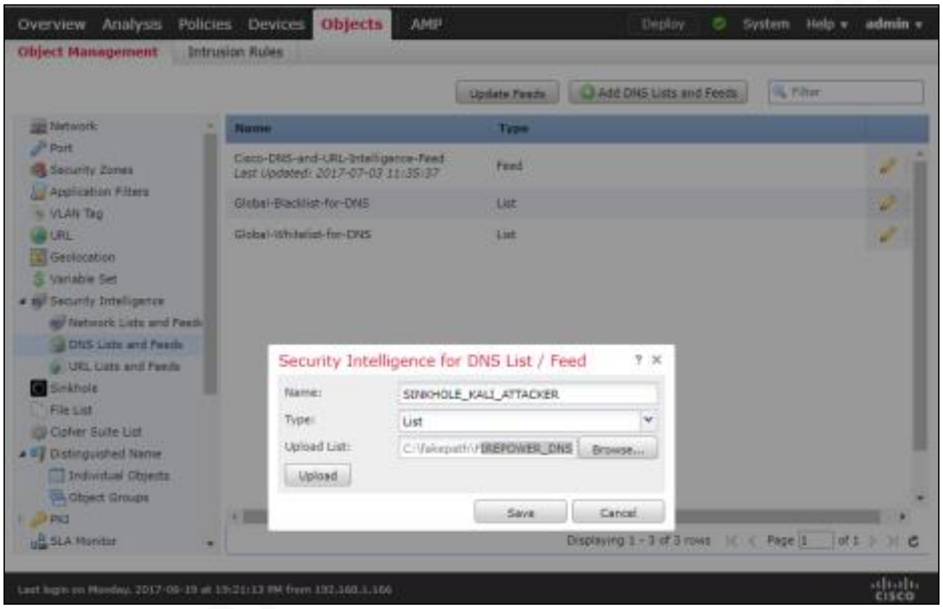
2.1.1. Threat Intelligence Configuration and Tuning

Two types of TI feeds exist on most solutions: one is a vendor or third-party feed and the other is a user defined blacklist and/or whitelist. Configuring the vendor feed usually consists of verifying connectivity to a specific vendor server and configuring a list download frequency. Once connectivity has been established, the system will automatically download the TI database and install it into production. For most technologies, the TI feed blocking will need to be enabled on specific inbound and outbound interface ACL's.

It is key that an intrusion analyst understands the modality of the block (IP vs. DNS resolution) and categories the vendor provides for blocking. In the case of the IP based block, the system will compare the source or destination IP address to the database and will block offending traffic. In the case of a DNS resolution, the system will either give the option to drop the packet, send a "Domain Not Found" DNS response, or reply to the DNS request with the IP of a sinkhole. The impact of these options is key for the analyst to understand.

Dropping an IP request is the simplest form of block on the system. DNS responses will have different impacts depending on the piece of malware. A DNS response of "Domain Not Found" may cause the malware to start checking other DNS names that it has been programmed to respond. In other cases, the malware may shutdown if it cannot communicate back to the DNS host. A DNS sinkhole can be configured to send traffic to an analyst defined IP address. This can be used to make the malware think it has found a live C2 server, provide a way to identify infected internal hosts, or allow for redirection of activity.

To configure a sinkhole the first task is to create a DNS List which is a text file that defines the DNS names or namespace you wish to redirect. The '*' can be used as a wildcard or a specific host name can be defined. In this case the DNS name 'kali-lin-attk.attacker.inside' will be defines as a DNS request that will be directed to a DNS sinkhole. Once created the file is loaded into FirePOWER Management Center under Objects>Security Intelligence>DNS Lists and Feeds. A sinkhole must also be defined which will cause the FirePOWER to send a DNS resolution response to the requesting host with the IP of the sinkhole device.



Name	Type
Cisco-DNS-and-URL-Intelligence-Feed Last Updated: 2017-07-03 11:35:37	Feed
Global-Blacklist-for-DNS	List
Global-Whitelist-for-DNS	List

Displaying 1 - 3 of 3 rows | Page 1 of 1

Security Intelligence for DNS List / Feed

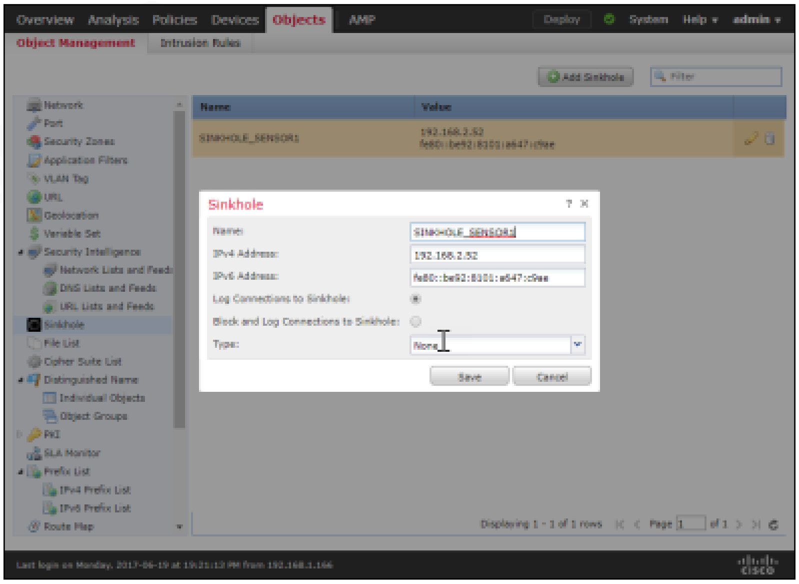
Name: SINKHOLE_KALI_ATTACKER

Type: List

Upload List: C:\fakepath\FIREPOWER_DNS

Last login on Monday, 2017-06-19 at 19:21:13 PM from 192.168.1.166

Figure 1 - DNS Block List



Name	Value
SINKHOLE_SENSOR1	192.168.2.52 fe80::be92:8101:a647:c9ee

Displaying 1 - 1 of 1 rows | Page 1 of 1

Sinkhole

Name: SINKHOLE_SENSOR1

IPv4 Address: 192.168.2.52

IPv6 Address: fe80::be92:8101:a647:c9ee

Log Connections to Sinkhole: ●

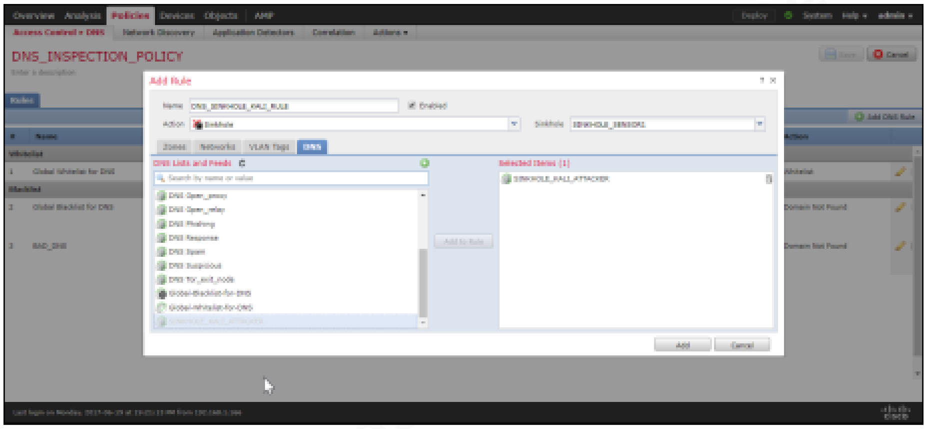
Block and Log Connections to Sinkhole: ○

Type: None

Last login on Monday, 2017-06-19 at 19:21:13 PM from 192.168.1.166

Figure 2- Sinkhole Definition

Once these objects have been defined in the FirePOWER Management Center, a rule is the added to the DNS Policy like a traditional firewall rule. This configuration is located under Policies>DNS under a user defined DNS policy entry. A rule is added to the DNS policy that sets the action of Sinkhole traffic to the previously defined sinkhole sensor and assigns the previously defined DNS List.



DNS_INSPECTION_POLICY

Add Rule

Name: DNS_SINKHOLE_KALI_RULE ☑ Enabled

Action: Sinkhole   Sinkhole: SINKHOLE_SENSOR1

DNS Lists and Feeds

- DNS Open_proxy
- DNS Open_relay
- DNS Phishing
- DNS Response
- DNS Spam
- DNS Suspicious
- DNS Tor_exit_node
- Global-Blacklist-for-DNS
- Global-Whitelist-for-DNS
- SINKHOLE_KALI_ATTACKER

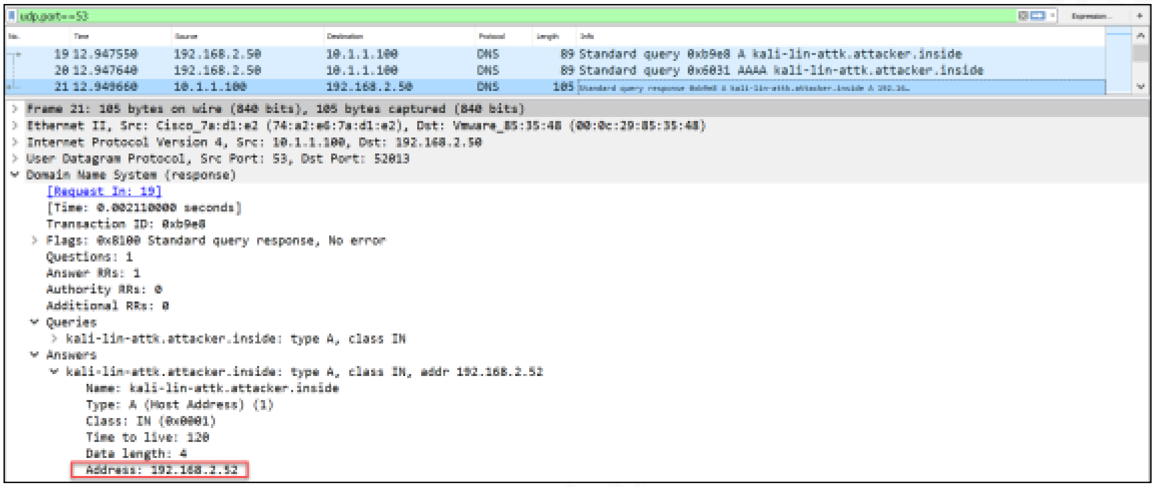
Selected Items (1)

- SINKHOLE_KALI_ATTACKER

#	Name	Action
1	Global Whitelist for DNS	Whitelist
2	Global Blacklist for DNS	Domain Not Found
3	BAD_DNS	Domain Not Found

Figure 3 - DNS Inspection Policy

The FirePOWER sensor will intercept the DNS name lookup request that matches the sinkhole rule. It will then return the IPv4 and IPv6 address of the defined sinkhole to the internal DNS server, which will then respond to the client with the sinkhole IP address. Figure 4 demonstrates the IPv4 DNS sinkhole reply to a client attempting to resolve a DNS name that has been directed to a sinkhole. This technique can be used to facilitate the identification and analysis of malware. It can also be used in the event a kill switch is identified with a specific type of malware to make that host appear reachable.



```
No.  Time        Source          Destination   Protocol  Length  Info
  19 12.947550   192.168.2.50    10.1.1.100    DNS       89      Standard query 0xb9e8 A kali-lin-attk.attacker.inside
  20 12.947640   192.168.2.50    10.1.1.100    DNS       89      Standard query 0x6031 AAAA kali-lin-attk.attacker.inside
  21 12.949660   10.1.1.100      192.168.2.50  DNS       105     Standard query response 0xb9e8 A kali-lin-attk.attacker.inside A 192.16…
> Frame 21: 105 bytes on wire (840 bits), 105 bytes captured (840 bits)
> Ethernet II, Src: Cisco_7a:d1:e2 (74:a2:e6:7a:d1:e2), Dst: Vmware_85:35:48 (00:0c:29:85:35:48)
> Internet Protocol Version 4, Src: 10.1.1.100, Dst: 192.168.2.50
> User Datagram Protocol, Src Port: 53, Dst Port: 52013
v Domain Name System (response)
    [Request In: 19]
    [Time: 0.002110000 seconds]
    Transaction ID: 0xb9e8
  > Flags: 0x8100 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  v Queries
    > kali-lin-attk.attacker.inside: type A, class IN
  v Answers
    v kali-lin-attk.attacker.inside: type A, class IN, addr 192.168.2.52
        Name: kali-lin-attk.attacker.inside
        Type: A (Host Address) (1)
        Class: IN (0x0001)
        Time to live: 120
        Data length: 4
        Address: 192.168.2.52
```

Figure 4 - Sinkhole DNS Response

Manual TI block lists are a very effective tool for an analysis to immediate block malicious activity that is defined by a specific IP address or DNS name. A key use case for the is when a phishing scheme occurs that requires the user to access a specific URL. The analyst can identify the URL and add it to the custom TI blacklist and prevent users from accessing the phishing site. Due to the block being based on location versus a dynamic signature, this is a very fast and effective method to protect against these types of attack.

2.1.2. Threat Intelligence Monitoring

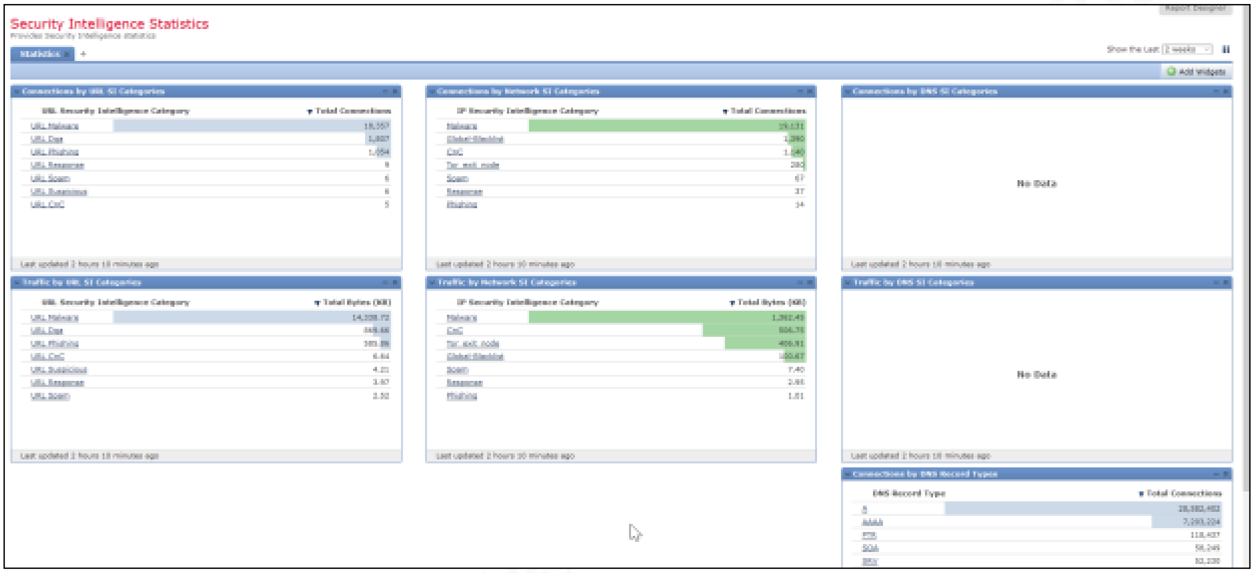
Due to the automated nature of this technology, it is often ignored from a monitoring perspective. Ignoring this system or taking it for granted is a key mistake for the analyst. Several components of this system must be monitored to make sure it remains functional and can provide significant intelligence into what is happening in the environment.

Making sure that the TI feed is being updated on a regular basis is a key responsibility of the analyst. Due to the nature of threats changing on an hour to hour or minute to minute basis, having a real-time update of malicious sites is critical. An alert should be put into place that notifies the system administrator when the system fails to update its TI database. This will allow the analyst to quickly resolve the issue that is preventing the update and restore effective communication. This should be a daily activity of the analyst responsible for this type of system.

Due to this subsystem being highly dependent on a low false positive rate, there may be instances where the TI feeds block a site or DNS name that may be legitimate. In this situation, the system level whitelist is provided to allow exceptions to be implemented. The analyst will need to perform due diligence to verify that the entry is indeed a false positive. This can be accomplished through traditional investigation and analysis, but a whitelisted item should be very rare with this type of technology.

Deploying a NGIPS solution with a very strong vendor TI capability is a key value proposition for the organization using this technology. It provides the ability to mitigate. One of the key questions when scoping this type of solution should be an evaluation of the competence and capability of the vendor’s research organizations.

Changes in the quantity and type of TI blocks can be an indicator of compromised hosts or the organization has been subject to a significant malware campaign. Determining which hosts are effected can be a very time-consuming process using the native FirePOWER tools due to the lag between page refreshes on the GUI console. To provide rapid analysis a combination of open source tools can be used to rapidly identify effected hosts. The figure below shows a significant increase in malware that is being blocked by the Security Intelligence (SI) function of the NGIPS device.



Security Intelligence Statistics
Provides Security Intelligence statistics

Connections by URL SI Categories

URL Security Intelligence Category	Total Connections
URL Malware	18,557
URL Dga	1,807
URL Phishing	1,054
URL Response	9
URL Spam	6
URL Suspicious	6
URL CnC	5

Last updated 2 hours 18 minutes ago

Connections by Network SI Categories

IP Security Intelligence Category	Total Connections
Malware	19,131
Global-Blacklist	1,390
CnC	1,540
Tor_exit_node	280
Spam	67
Response	37
Phishing	14

Last updated 2 hours 10 minutes ago

Connections by DNS SI Categories

No Data

Last updated 2 hours 18 minutes ago

Traffic by URL SI Categories

URL Security Intelligence Category	Total Bytes (KB)
URL Malware	14,208.72
URL Dga	449.66
URL Phishing	185.86
URL CnC	6.64
URL Suspicious	4.21
URL Response	3.87
URL Spam	2.32

Last updated 2 hours 18 minutes ago

Traffic by Network SI Categories

IP Security Intelligence Category	Total Bytes (KB)
Malware	1,362.45
CnC	556.75
Tor_exit_node	408.91
Global-Blacklist	100.67
Spam	7.40
Response	2.66
Phishing	1.81

Last updated 2 hours 10 minutes ago

Traffic by DNS SI Categories

No Data

Last updated 2 hours 18 minutes ago

Connections by DNS Record Types

DNS Record Type	Total Connections
A	28,183,492
AAAA	7,293,224
PTR	118,437
SOA	58,249
SRV	52,230

Figure 5 - Security Intelligence Blocking

Upon identification of this trend the next step would be to go to the Analysis>Security Intelligence screen and use the Report Designer tool to export the Security Intelligence logs into a comma delimited format. FirePOWER will not perform packet captures on SI blocks, only IPS alerts will perform a full packet capture. Simple BASH commands can be used to quickly identify communication pairs and provide a list of hosts that may be compromised which are attempting to communicate with a C2 server.

The figure below provides a good example of how a set of BASH commands can quickly provide meaningful information from the Security Intelligence CSV file very quickly. The “head -n 1 <filename>” command provides a very quick definition of the fields that are present in the log file. Once the field definitions for the SI CSV file has been defined, it is very simple to use a combination of the head, cut, sort, uniq, grep, and wc commands to develop meaningful information from the CSV file. Each iteration of these commands will take a matter of seconds, where attempting these through the normal management console will take minutes to complete.

In general, the cut command can be used to extract fields of interest in the report and provide a means of narrowing down the scope of the search. Using the sort and uniq commands allows for organizing the devices that are communicating and removing duplicate communication requests. Grep is an incredibly powerful tool that can use regular expressions to help refine the networks that are being scrutinized. In the case below the grep command with the regular expression “^10\.” is used to indicate that any line the starts with “10.” is selected. Assuming the internal network is prefixed by “10.” this will show us any internal hosts that are attempting to communicate with external hosts. Finally, the “wc -l” command will tell us a count of how many devices were attempting to communicate with the external host. A good knowledge of BASH commands will allow the analyst to quickly identify and act on compromised network devices.

```
user@host:/tmp$ head -n 1 SI_LOG.csv

First Packet,Last Packet,Action,Reason,Initiator IP,Initiator Country,Responder
IP,Responder Country,Security Intelligence Category,Ingress Security Zone,Egress
Security Zone,Source Port / ICMP Type,Destination Port / ICMP Code,Application
Protocol,Client,Web Application,URL,URL Category,URL
Reputation,Device,Security Context

user@host:/tmp$ head -n-2 SI_LOG.csv | cut -f 5 -d “,” | sort -n | uniq | grep -e “^10\.”
| wc -l

639

user@host:/tmp$
```

Figure 6 - BASH Reporting Commands

A good use case is to identify hosts that could have been compromised by malware and are attempting to communicate to C2 servers. While custom reports or the GUI interface can be used to identify hosts that generated a Malware SI block event, using BASH and/or a shell script is much faster. The example below shows the command set to extract the initiator IP address and reason for the SI block. The data is then sorted and duplicate items have been removed. A grep command is then used to extract lines that begin with internal 10.0.0.0/8 range. Finally, grep is used to only return items that were classified as malware. This information can then be sent to incident handling or desktop support teams to remediate the devices.

```
user@host:/tmp$ head -n-2 SI_LOG.csv | cut -f 5,9 -d “,” | sort -n | uniq | grep -e
“^10\.” | grep Malware
```

Figure 7 - Extract Malware Infected Host List

2.2. Application Control with Deep Packet Inspection

Application Control with Deep Packet Inspection (DPI) is defined as the system’s ability to inspect traffic’s layer 3 – 7 components. Review of these components effectively allow the NGIPS to make decisions based on the application that is used. Traditional firewalls were limited to making decisions based on the layer 3 and layer 4 components of the traffic. This capability is one of the key functionality differences that separates traditional IPS and NGIPS functionality.

Identifying what applications are being used and how they are being used is a key capability that the NGIPS delivers. In traditional IPS visibility was restricted to source and destination port pairs. Today, NGIPS can provide information based on the behavior of a given application over a given port. This allows the user deploying the solution to develop a more refined definition of what applications and use cases are allowed over their network. This allows the organization to better define how their network resources are being used and how they can implement controls that control resource utilization.

Initial implementation of this technology often provides a redefinition of how the organization understands that its network is being used. Traditional IPS has let most organizations to view network utilization within the lens of established TCP/IP protocol use based on port number. DPI technology has changed this by providing a more accurate user based behavior foundation of network activity.

2.2.1. Application Control Configuration

Understanding what applications are being used on the network is the first step in controlling how resources are being used. In many cases one aspect of a given application may be beneficial to an organization; other uses of that application may be harmful. NGIPS provides the capability to control the way users use a given application. Having the ability to control these applications provides the organization the capability to make the most effective utilization of its information technology resources.

Applications are generally detected using Deep Packet Inspection (DPI) technology that looks at layers 2 – 7 of packet data to identify the application being used. Typically, the DPI does a signature based inspection of a given packet to review the behavior of the underlying application and comparing it to a signature database. Depending on the technology vendor, this system may or may not include the application preprocessor capability that is usually associated with the IPS engine. This provides the network administrator the capability to establish rules based on the application of subcomponent of an application rather than just the source and destination port. It also reduces the chance an attacker can evade detection by using a non-standard port.

Defining what applications are allowed between network segments can provide an additional layer of protection on the network. Known malicious or business irrelevant applications can effectively be blocked with blocking entire ports of traffic. For example, YouTube could be allowed on a network and Facebook blocked, even though both traversed over port TCP/80. From a basic security standpoint, applications such as peer-to-peer networking, malicious software, bandwidth intensive non-business applications, and other obvious malicious traffic should be controlled.

Other traffic that may be restricted is anonymizers, proxies, and VPN clients depending on the customer's information security policy. These technologies are often used to bypass traditional web content and IPS filtering by end users. Due to these technologies being easier to use and available on a wide range of platforms, they are actively used by many users. In the sample network, implementing blocking of these categories resulted in a 20% decrease in network utilization. These technologies also provide a mechanism for malicious insiders to establish covert channels and cover their tracks.

2.2.2. Application Control Monitoring

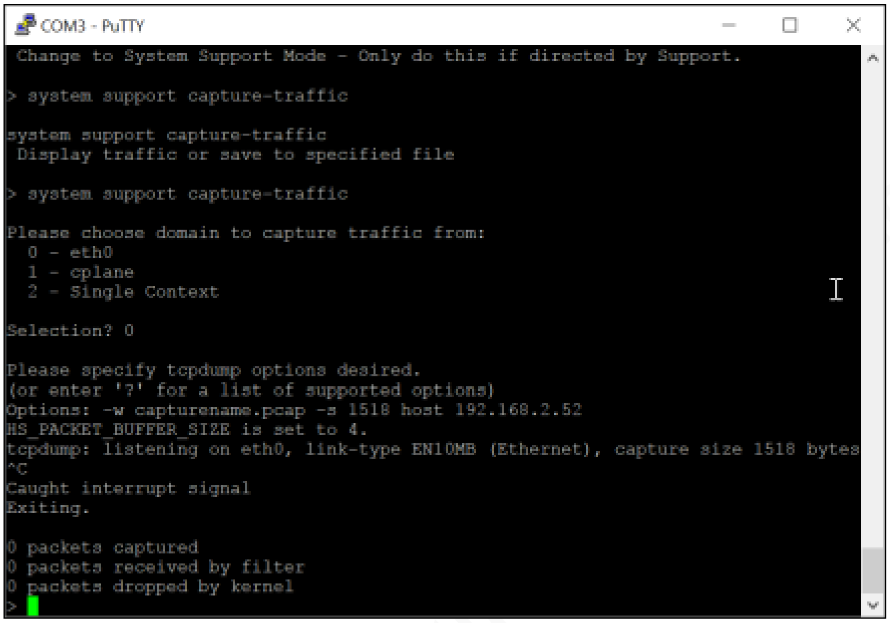
Monitoring application control and application flow through the NGIPS is critical in maximizing the effectiveness of the solution. Due to the application control system using a signature based engine, the analyst must actively monitor that these signatures are being updated on a regular basis. Failure to do this could create a situation where a new application appears or an old application changes which could affect the system's ability to recognize the application.

Several key aspects of traffic flow through the firewall need to be reviewed regularly to effectively take use of the technology. These include macro level application blocking to reduce workload on the IPS subsystems, anomalous traffic monitoring that is not triggering alerts, and unexpected changes in traffic or connection volume.

Basic application rules provide a bulk mechanism like the TI feeds to block traffic before it enters the more resource intensive IPS subsystem. Preventing unwanted applications at this level can reduce the amount of resources and tuning time that is required to manage IPS. This can also help supplement reducing the volume of traffic being directed to secondary inspection technologies. Regular review of the top applications can quickly alert an analyst to abnormal application behavior on the network. If an unusual application appears in the application traffic flow monitoring that violates policy, the analyst can quickly block the applications. Using this to develop an egress application filtering policy can be a low effort layer of defense.

Anomalous traffic that does not trigger an IPS alert or a TI alert can often be identified through monitoring the application control reports. A key priority should be made to investigate applications that are communicating on uncommon ports and are not matching an application signature. A good example of this is on the example network an excessive amount of UDP/31005 traffic was found communicating with an Internet host every 60 seconds. This traffic was not marked as malicious by the TI or IPS systems, but was anomalous network behavior.

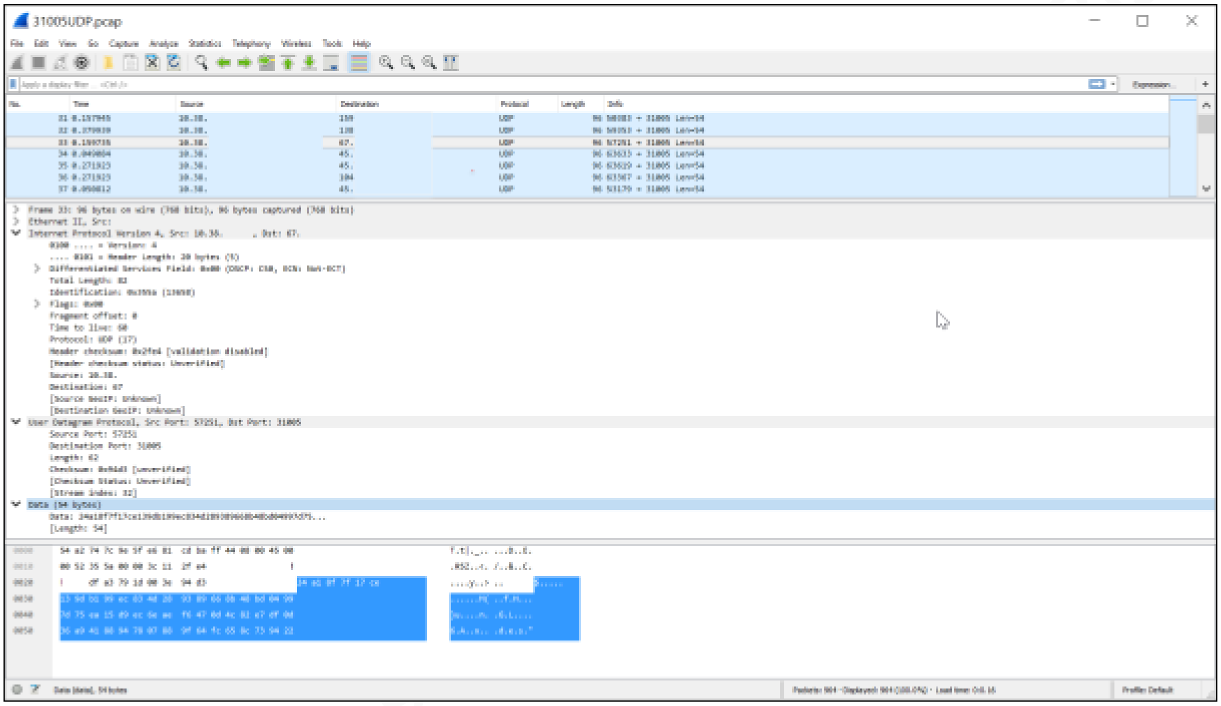
The first step in performing an analysis is to capture a sample of the traffic from the IPS sensor for analysis. FirePOWER allows for tcpdump to be used on the device to capture specific traffic flows and scp to allow for the file to be copied to an analysis system for analysis. Well defined BSD filters should be used to minimize the amount of traffic captured to prevent performance impacts on the sensor.



```
 Change to System Support Mode - Only do this if directed by Support.

> system support capture-traffic

system support capture-traffic
 Display traffic or save to specified file

> system support capture-traffic

Please choose domain to capture traffic from:
  0 - eth0
  1 - cplane
  2 - Single Context

Selection? 0

Please specify tcpdump options desired.
(or enter '?' for a list of supported options)
Options: -w capturename.pcap -s 1518 host 192.168.2.52
HS_PACKET_BUFFER_SIZE is set to 4.
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 1518 bytes
^C
Caught interrupt signal
Exiting.

0 packets captured
0 packets received by filter
0 packets dropped by kernel
> 
```

Figure 8 - FirePOWER Traffic Capture

The figure below shows a simple Wireshark view of the captured data. Since this data is from a live system the layer 2/3 packet information has been obfuscated. The source address for the series of data is the same indicating a single host is generating this traffic. Traffic is being sent to a wide variety of destination IP addresses that are hosted by various cloud hosting providers. Review of the packets indicate a very standard packet configuration of using a packet length of 96 bytes using standard UDP packet construction. The IP & UDP packets are 20 bytes and 22 bytes with a data payload of 54 bytes. Review of the payload across packets shows a changing data value between each packet. Due to the fixed length of the data it appears to be some form of hashed value, but is not being identified by common hash identifiers.


31005UDP.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
Profile: Default

```
No.   Time       Source   Destination   Protocol  Length  Info
   31 0.157945   10.38.   159           UDP           96  58383 → 31005 Len=54
   32 0.279939   10.38.   138           UDP           96  59353 → 31005 Len=54
   33 0.199735   10.38.   67.           UDP           96  57251 → 31005 Len=54
   34 0.049004   10.50.   45.           UDP           96  63615 → 31005 Len=54
   35 0.271923   10.50.   45.           UDP           96  63619 → 31005 Len=54
   36 0.271923   10.50.   104           UDP           96  63567 → 31005 Len=54
   37 0.050812   10.50.   45.           UDP           96  53179 → 31005 Len=54

> Frame 33: 96 bytes on wire (768 bits), 96 bytes captured (768 bits)
> Ethernet II, Src:
v Internet Protocol Version 4, Src: 10.38.      , Dst: 67.
      0100 .... = Version: 4
      .... 0101 = Header Length: 20 bytes (5)
    > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      Total Length: 82
      Identification: 0x355a (13658)
    > Flags: 0x00
      Fragment offset: 0
      Time to live: 58
      Protocol: UDP (17)
      Header checksum: 0x2fe4 [validation disabled]
      [Header checksum status: Unverified]
      Source: 10.38.
      Destination: 67
      [Source GeoIP: Unknown]
      [Destination GeoIP: Unknown]
v User Datagram Protocol, Src Port: 57251, Dst Port: 31005
      Source Port: 57251
      Destination Port: 31005
      Length: 62
      Checksum: 0x94d3 [unverified]
      [Checksum Status: Unverified]
      [Stream index: 32]
v Data (54 bytes)
      Data: 34a18f7f17ce139db199ec834d20938966084bbd04987d75...
      [Length: 54]

0000  54 a2 74 7c 9e 5f e6 81  cd ba ff 44 08 00 45 00   T.t|._.. ...D..E.
0010  00 52 35 5a 00 00 3a 11  2f e4                     .R5Z..:. /...
0020        df a3 79 1d 00 3e  94 d3 34 a1 8f 7f 17 ce   ....y..> ..4.....
0030  13 9d b1 99 ec 83 4d 20  93 89 66 08 4b bd 04 98   ......M  ..f.K...
0040  7d 75 ea 15 d9 ec 6e ae  f6 47 0d 4c 82 e7 df 0d   }u....n. .G.L....
0050  36 a9 41 88 94 78 07 88  9f 64 fc 65 8c 73 94 22   6.A..x.. .d.e.s."

Data (data), 54 bytes     Packets: 984 · Displayed: 984 (100.0%) · Load time: 0:0.16     Profile: Default
```

Figure 9 - Packet Capture Data

Indicators from this traffic would suggest that the device may be under control of an outside organization and may have established a covert channel. The next step in the analysis process would be to perform a forensic analysis on the device to determine if there is malware on the device or if it is a false positive. This information could then be used to develop a SNORT signature that looks for a UDP packet with a specific length of 96 bytes, is using a UDP port, and has a data payload of 54 bytes. No on device alerts were triggered on the device and this demonstrates how traditional intrusion analysis techniques must be used in conjunction with automated detection methods.

NGIPS provides traffic volume changes in the form of bits/bytes transferred or by the number of connections an application makes. From an analyst's perspective, the distinction between these two measures provides indicators of different behavior. A large increase in the amount of data transmitted can be an indicator of major data exfiltration, bypass of content filtering rules for data heavy applications (Netflix, Bit Torrent, etc.), and use of VPN clients for content filtering evasion. Connection volume provides much different information if there is an increase in high volume low payload traffic. C2 traffic is a good example of this type of traffic. Many small packets being transmitted may not trigger an anomaly if looking at data throughput volume. By reviewing connection volume, this behavior may be very visible in the alert console. Many vendors do not make a clear distinction about which measure they are using in their dashboard consoles.

2.3. IPS Engine Configuration

The IPS engine is the heart of the NGIPS solution with the primary purpose of identifying and blocking malicious traffic that not been blocked by the ruleset, TI, and application control defensive layers. The previous mechanism provides a very low false positive protection suite that reduces the analyst workload and reduces the amount of resource intensive inspection that must be performed by the IPS engine. Traffic that reaches the IPS for inspection will evaluated based on the application pre-processors and signatures that have been defined for the engine. This system allows for analysis of traffic that has a higher false positive rate, but goes beyond basic traffic characters like IP address, DNS name, application type, protocol, port, and IP address information. This allows for greater scalability on the platform and provides a major contribution to the effectiveness of the solution.

Most NGIPS will use a model of packet capture, traffic normalization, application pre-processing, and then the normalized traffic will flow through an inspection engine for analysis. The inspection engine will then compare the traffic to a defined signature set based on the traffic type to analyze for the desired behavior. For purposes of this section, the Cisco FirePOWER solution will be the reference architecture. Due to Cisco FirePOWER being based on SNORT technologies, the examples may be more widely recognized and it increases the interoperability with Open Source technologies. Some of the basic SNORT configuration steps have been automated in this product such as configuring network variables, default application ports, and other basic SNORT configuration tasks. One of the key elements of the solution is to enable network administrators that are not familiar with SNORT to configure a functional IPS deployment.

2.3.1. Packet Capture

Strategic placement of the NGIPS sensor requires a comprehensive security architecture that balances the network architecture, security policy, and budgetary constraints to identify the location and configuration of IPS sensors. Two primary NGIPS configurations exist which are a layer 2 transparent mode or a layer 3 routed mode. Layer 2 deployments are typically implement when the NGIPS needs to be implemented without changing the routing topography of the network. Traffic is placed in line with normal traffic and does not manipulate layer 3 information for data transport. Device failover is usually handled with Network Interface Cards (NIC) that will default to a failed open state where the physical signal is regenerated when a higher-level system failure occurs.

When configured in layer 3 routed mode, network traffic can be directed through the IPS via network routing protocols and in most cases, the device can participate in the network routing protocols. This mode is advantageous when an organization wants to have the ability to direct traffic to specific sensors, have a routing based device failover plan, or want to deploy firewall like capabilities. A common use case for this configuration is to redirect HTTPS traffic to specialized devices for SSL decryption, inspection, and re-encryption activities. In many cases the cost of using content filtering technology for SSL inspection is cheaper then standalone decryption units or adding additional system resources for NGIPS decryption.

Supplemental packet capturing technology may be required to augment analysis when investigating a complex incident. In many cases the NGIPS analysis capabilities are powerful, but they lack the ability to apply customer scripting and can be extremely slow to analyze, even small data sets. An example of this can be seen in the Cisco FirePOWER Management Center where analysis on small data looking sets can take 30 seconds to 2 minutes to refresh on high performance hardware. Performing the same analysis using a raw packet capture would take less than 5 seconds. For most highly targeted analysis operations, an external tool with a raw packet capture is needed to do a rapid detail analysis. Tcpdump on the sensor can be used with BSD filters to capture traffic in a PCAP format directly from most sensors. This can help in collection reoccurring anomalous traffic for analysis on demand. Using an external continuous packet capture solution can also be used to augment the NGIPS.

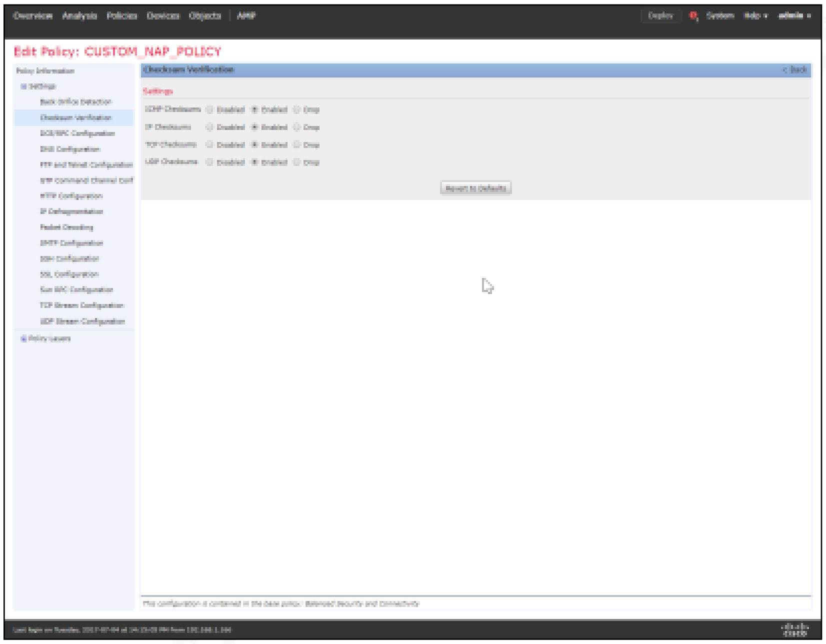
2.3.2. Traffic Normalization

Traffic normalization preprocessors collect and assemble traffic prior to analysis by the IPS engine for a given application. The purpose of these preprocessors is to define the basic rules that a given network application will follow and how that application will be presented to the IPS subsystem. Each application preprocessor has a set of variables that can be defined such as port numbers, application commands, timeouts, encoding types, and other detailed components. These variables help define the basic expected application behavior and to prevent malicious actions like IPS evasion using manipulated protocols.

Most systems are deployed with a default configuration for the preprocessors from the vendor. These are the typical behaviors that the vendor expects with configurations that have the least negative impact on the system. For most users, these configurations will never be changed. A skilled intrusion analyst can use the features to help detect malicious activity and better tailor the solution for their environment. These configuration options can do things like remove anomalous TCP flags, clear reserved bits, define packet decoding protocols, and in general control how traffic is presented to the IPS engine.

Preprocessor configuration in Cisco FirePOWER is located under the access control menu under the Network Analysis Policy (NAP). Once accessed there are numerous protocol preprocessors that can be customized based on the desired effect to be applied to the traffic before being presented to the IPS engine. Due the complexity involved with the polices, it is highly recommended that changes be made in a lab/test environment using proper packet crafting and packet analysis to verify the rule changes have the desired effect. Many of the configuration options can also be applied in a SNORT environment to develop and test the preprocessor configuration.

A good example of how a preprocessor could be tuned is that checksum verification can be configured for ICMP, IP, TCP, and UDP traffic. This can allow the preprocessor to ignore, drop, or maintain failed checksums on each of those packet types. In the event an attacker is attempting to manipulate checksum values for IPS evasion, the intrusion analyst can change the behavior of the preprocessor to deliver the IPS engine the desired traffic flow for inspection. Another example would be packet decoding which can be configured to detect Teredo packets on non-standard ports. This can be used to detect when IPv6 to IPv4 traversal is being used on odd ports and deliver that traffic to the IPS engine for analysis.



Edit Policy: CUSTOM_NAP_POLICY

Checksum Verification

Settings

ICMP Checksums ○ Disabled ● Enabled ○ Drop
IP Checksums ○ Disabled ● Enabled ○ Drop
TCP Checksums ○ Disabled ● Enabled ○ Drop
UDP Checksums ○ Disabled ● Enabled ○ Drop

This configuration is contained in the base policy: Balanced Security and Connectivity

Last login on Tuesday, 2017-07-04 at 14:15:03 PM from 192.168.1.166

Figure 10 - Network Analysis Policy

When an analyst is tuning these policies, they should be aware that changing these variables could have a significant effect on the NGIPS solution's performance, behavior of the IPS engine, and the expected behavior on traffic flowing through the device. These configuration options have an incredibly powerful impact on the underlying operation of the NGIPS engine and should be fully understood by the analyst before changes are made. They do provide the knowledgeable analyst a very powerful resource to customize the solution to very specific network environments.

2.3.3. Inspection Engine and Signatures

Data that has been normalized is presented to the inspection engine to be analyzed by signatures that have been defined for that traffic type. The signature looks for characteristics like header information, traffic directionality, and payload to decide if malicious or abnormal traffic is present. When a signature is matched, the inspection engine decides to ignore, alert, or block the offending traffic. The Cisco FirePOWER engine follows an inspection and rule format like SNORT and will be a point of reference for this section.

Typically, a base set of signatures that is updated regularly is provided by the product vendor as part of their solution. These signatures typically are broken into categories of signatures based on the behavior they are designed to detect. Each signature has some form of a validity rating that identifies how likely it will generate a false positive result. New signatures are typically released when a new attack or vulnerability emerges and the vendor’s research team develops a high-quality signature. High quality signatures are typically automatically enabled when updated and low-quality signatures require manual activation. Users also can create their own signatures to detect customer traffic.

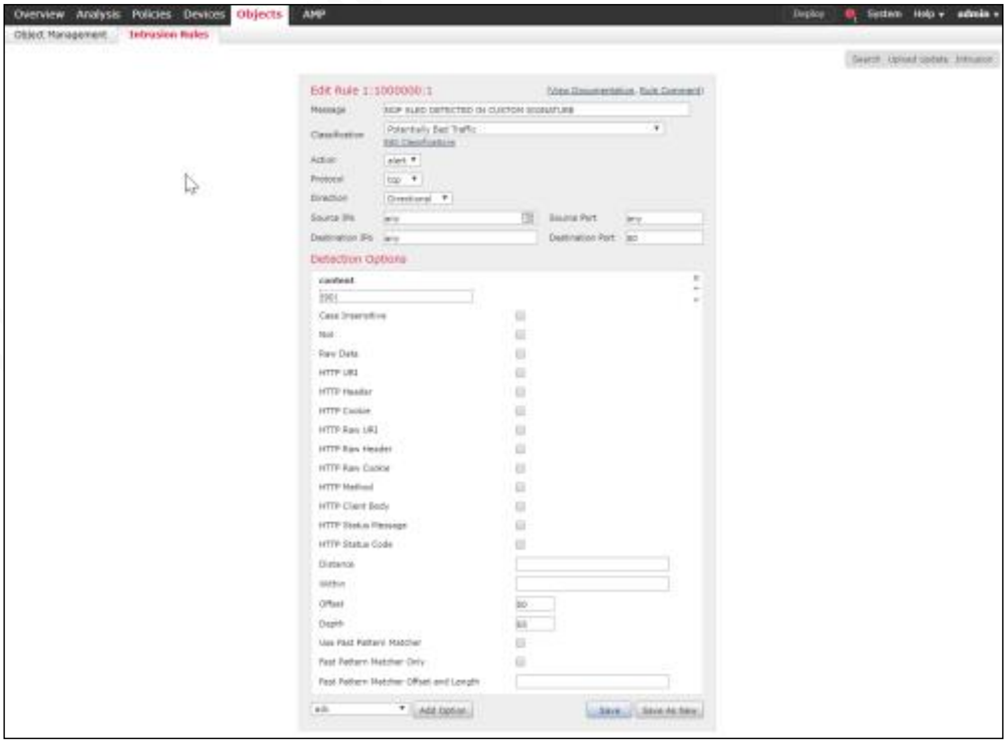
User defined scripts can be created using a GUI interface on the console or a SNORT rule can be imported into the system. These signatures are applied on top of a common rule base so that user defined signatures will override signatures provided by the vendor. Rules can be applied to control how customer rules are applied in complex organizations. It is recommended that user defined rules first be developed in a SNORT lab environment. Rule development should be conducted using SNORT best practices (creating specific rules, using fast match, avoiding complex PCRE sequences, etc.) and fully tested in a simulated environment. The performance impact on the test sensor should also be evaluated. Once the custom rules have been created, it should be imported to the NGIPS and configured in an audit mode. After a trial window, it can then be moved into blocking mode.

Custom signatures are an important feature for any IPS, upon detection of a new attack or identification of a zero day exploit the vendor may not have an adequate signature. In these cases, the intrusion analyst must create a custom detection signature to identify and block the offending traffic. In the example, an exploit may have a NOP sled between bytes 50 and 65 targeting tcp/80. The analyst would begin by developing a basic snort signature that searches between byte 50 – 65 for NOP bytes (|90|). An example of the SNORT rule can be seen below.

```
alert tcp any any -> any 80 (msg:"NOP SLED DETECTED IN CUSTOM
SIGNATURE"; content:"|90|"; offset:50; depth:65;)
```

Figure 11 - SNORT Sample Rule

The rule could then be tested in a lab environment on a FirePOWER test system or the rule could be validated using a simple virtual environment with SNORT and a packet crafting tool. Once the rule has been validated it can be installed in FirePOWER by going to the Objects>Intrusion Rules screen and either use the GUI or import the SNORT rule from a text file. The GUI interface provides a more user-friendly way to create rules, but then requires testing to be performed using the FirePOWER console. This may not be the most effective way to generate a new rule.



Edit Rule 1:1000000:1

Message: NOP SLED DETECTED IN CUSTOM SIGNATURE

Classification: Potentially Bad Traffic

Edit Classifications

Action: alert

Protocol: tcp

Direction: Directional

Source IPs: any

Source Port: any

Destination IPs: any

Destination Port: 80

Detection Options

content: |90|

Case Insensitive □

Not □

Raw Data □

HTTP URI □

HTTP Header □

HTTP Cookie □

HTTP Raw URI □

HTTP Raw Header □

HTTP Raw Cookie □

HTTP Method □

HTTP Client Body □

HTTP Status Message □

HTTP Status Code □

Distance:

Within:

Offset: 50

Depth: 65

Use Fast Pattern Matcher □

Fast Pattern Matcher Only □

Fast Pattern Matcher Offset and Length:

ack

Figure 12 - FirePOWER IPS Rule GUI

Once the rule has been created it must be activated on the relevant intrusion policy. The intrusion policy is located under Policies>Intrusion. Manually created rules are stored in the local rule category. To enable the rule the Rule State must be changed from Disable to Generate Events or Drop and Generate Events based on whether the administrator wants a simple alert or to block traffic. In most cases the rule should be tested in Generate Event state before it is moved to drop traffic. This can help with rule tuning and prevent a denial of service condition.

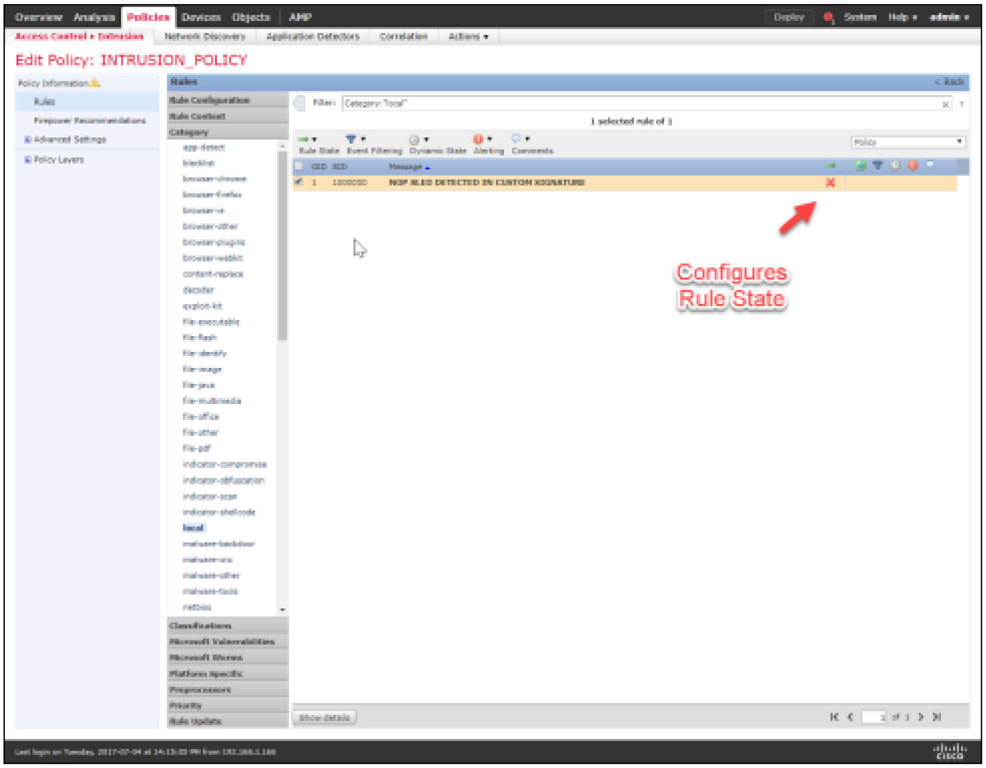

Overview Analysis Policies Devices Objects AMP
Deploy System Help ▾ admin ▾
Access Control ▸ Intrusion Network Discovery Application Detectors Correlation Actions ▾
Edit Policy: INTRUSION_POLICY
Policy Information
Rules
Firepower Recommendations
Advanced Settings
Policy Layers
Rules
< Back
Rule Configuration
Rule Content
Category
Filter: Category:"local"
1 selected rule of 1
Rule State Event Filtering Dynamic State Alerting Comments
Policy
GID SID Message
1 1000000 NOP SLED DETECTED IN CUSTOM SIGNATURE
Configures Rule State
Classifications
Microsoft Vulnerabilities
Microsoft Worms
Platform Specific
Preprocessors
Priority
Rule Update
Show details

Figure 13 - Custom Signature Activation

Network and host profiling is a mechanism that NGIPS uses to help automatically tune the IPS signature set. Using the same technology that is used to develop Indicators of Compromise (IOC), the system will identify which rules are relevant to the current environment and will enable/disable rules as necessary. An example would be if no BSD systems were in the environment, then the system may disable BSD related signatures. This reduces system overhead and lowers the chance for false positives that may be triggered based on an unnecessary signature. Due to changes that occur in the network over time, this function should be run monthly to ensure that a current network profile is used as the basis for the operating ruleset. Failing to do this could result in a false negative situation where an application appears, but rulesets have not been applied.

Using the host profiling method to enable rules is time saving especially for an organization that does not have an intrusion analyst on staff. This provides a baseline of functionality that can provide an organization with IPS capabilities they may have otherwise never been able to have. It is not a replacement for traditional tuning that an intrusion analyst would perform. The baseline rule recommendations will provide a starting point that requires minimal effort. Additional rules should be enabled by review of applications that are on the network, security architecture, and rules that reflex unique business requirements. Unique requirements could be things such as Data Loss Prevention (DLP), SCADA environments, Internet of Things (IoT) systems, and other areas that have specific rule requirements.

2.4. IPS Engine Monitoring

The monitoring process for NGIPS is usually based on some form of basic alert which is like traditional IPS. Most NGIPS solutions go a step further by adding the concept of an Indicator of Compromise (IOC) which is a correlation of multiple factors including end device operating system, application versions in use, multiple behaviors, and other indicators that can help identify if a machine has been compromised. This technology helps reduce the time an analyst spends monitoring the solution. These two components provide the primary data outputs that the intrusion analyst will use daily to manage events that are detected by the IPS subsystem.

Several other systems that must be monitored on a regular basis include the frequency of signature updates, system resource utilization, and ongoing management of custom signatures. Failure to monitor these systems can result in false positive or false negative situations without showing any alert information. Most systems have automated health checks that can be configured to alert when there is a change in these components.

2.4.1. IPS Alerts and Indicators of Compromise

IPS alerts appear when an event triggers a signature in the NGIPS solution. The alert is usually prioritized by the signature that has been defined. Alerts are standalone elements that are not using contextual information to prioritize and gain more information about the alert. Standard alerts have been the traditional output of most IPS technologies since the inception of the technology. Security Information and Event Management (SIEM) systems were used to correlate information from other sources to provide a better reference to things like operating system type, application type, and other factors that validate or invalidate the alert.

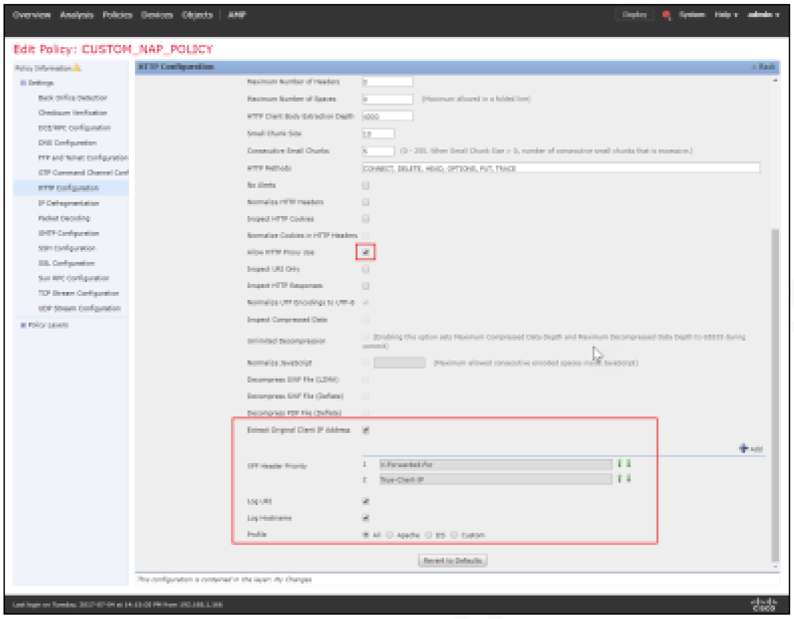
Traditional alerts can be monitored and reviewed typically based on priority. Depending on the volume of alerts and analysts available for the organization, this model allows for the inspection of all alerts that may occur providing greater detail. This is also a good method to identify customer signatures that may be needed for low volume alerts, but does not warrant configuring and an IOC in the environment. Most NGIPS systems will provide the individual alert's IOC rating with the alert to provide easier indexing. Alerts are best described as a view of a network event that triggered the alert that is independent of other variables on the network. This independent view can be helpful especially for malicious traffic that may be designed to evade the IOC classification systems.

IOCs are typically alerts that are generated by NGIPS reviewing the traffic in context of the operating systems, application versions, traffic volumes, device vulnerability status, user identity, and other characteristics. This is used to develop a profile of the hosts that are on the current network that helps put the nature of the host into context when an alert is triggered. Host information can be gathered using a wide range of built in tools, but can also be expanded on using third party tools such as vulnerability scanners, endpoint security applications, and port scanning utilities. The more information that can be gathered about the host, the more accurate the system can evaluate the criticality of an observed security event.

IOC is a function of the system analysis engine that compares the IPS alert with the data that has been gathered by the host and traffic profiles. The goal is to identify if the system has been compromised and to develop a score of the priority of a given incident. Using this method helps reduce the amount of time that is spent by an intrusion analyst gathering this information and manually investigating. It also increases the accuracy of automatically blocking attacks to help reduce the number of false positives that are encountered in the environment. This capability has helped increase the adoption of NGIPS in organizations that would otherwise not have the staffing capability to manage an IPS solution. It has also lead to a greater blurring of the designation of NGIPS and NGFW in the market.

Intermediate systems like proxy servers, transparent firewalls, load balancers, traditional firewalls, and Network Address Translation (NAT) can give the host profile system an inaccurate view of the true nature of the hosts on each end of the network communication. An example of this would be a clustered web content filtering proxy solution using Web Cache Communication Protocol (WCCP) to redirect outbound HTTP/HTTPS traffic for SSL decryption and content inspect. The proxy server will appear as the source or destination for any given HTTP/HTTPS traffic depending on the placement of the NGIPS sensor. The result will be that the host profiling agent on the NGIPS will see all the network host's traffic as if it were originating or destined from the proxy server. The result is the IOCs will appear as if the proxy server if the device is compromised.

An issue like this can be resolved by implementing a technology like X-Forwarded-For (XFF) headers to inject an end station identified in the packet. The NGIPS can then be configured to extract that header information and store it in the host profile to positively identity the host. If a network administrator does not realize this behavior occurs, then a situation can arise where compromised hosts are not identified. Other examples that could occur include large number of clients using different IP addresses in a DHCP pool, personal firewalls, and network firewall configurations which can change a host or application fingerprint. The XFF Header configuration is located under the Network Analysis Policy assigned to the relevant intrusion policy. The figure below highlights this configuration.



Edit Policy: CUSTOM_NAP_POLICY

HTTP Configuration

Maximum Number of Headers: 0
Maximum Number of Spaces: 0 (Maximum allowed in a folded line)
HTTP Client Body Extraction Depth: 1000
Small Chunk Size: 10
Consecutive Small Chunks: 5 (0 - 255. When Small Chunk Size > 0, number of consecutive small chunks that is excessive.)
HTTP Methods: CONNECT, DELETE, HEAD, OPTIONS, PUT, TRACE
No Alerts ☐
Normalize HTTP Headers ☐
Inspect HTTP Cookies ☐
Normalize Cookies in HTTP Headers ☐
Allow HTTP Proxy Use ☑
Inspect URI Only ☐
Inspect HTTP Responses ☐
Normalize UTF Encodings to UTF-8 ☑
Inspect Compressed Data ☐
Unlimited Decompression ☐ (Enabling this option sets Maximum Compressed Data Depth and Maximum Decompressed Data Depth to 65535 during commit)
Normalize Javascript ☐ (Maximum allowed consecutive encoded spaces inside Javascript)
Decompress SWF File (LZMA) ☐
Decompress SWF File (Deflate) ☐
Decompress PDF File (Deflate) ☐
Extract Original Client IP Address ☑
XFF Header Priority: 1 X-Forwarded-For; 2 True-Client-IP
Log URI ☑
Log Hostname ☑
Profile: ● All ○ Apache ○ IIS ○ Custom

This configuration is contained in the layer: My Changes

Last login on Tuesday, 2017-07-04 at 14:13:20 PM from 192.168.1.100

Figure 14 - XFF Header Configuration

Intentional IOC evasion is the second issue that could arise if an attacker was able to identify the NGIPS system that is in use. In theory, a host could be compromised to report back information that would convince the IOC engine to view the target as not vulnerable to an attack. An example would be to change the application banners, application behavior, default packet fields (TTL, MSS, MTU, etc.), odd fragmenting, or other factors that the fingerprinting system is based on to make the host appear as not vulnerable to an exploit. This would allow an attacker to reduce the likelihood of detection if the IOC was the only alerts that were being monitored by the NGIPS.

From a workflow perspective, it is recommended for analysts to start with IOC monitoring during their normal analysis activity. The technology helps maximize the positive value the analyst provides the organization by reducing the analysis overhead associated with profiling an intrusion. With that said, the IOC alerts should not be the only alerts that are monitored on the device. Manual analysis of IPS alerts, application activity, TI activity, and the other device is necessary on a routine basis to get the most value out of the solution. Just like signatures, IOCs should not be viewed as 100% accurate 100% of the time.

3. Conclusion

NGIPS can dramatically have a positive impact on virtually any network environment if designed, deployed, and monitored effectively. Even though the vendors of these solutions like to portray these systems as standalone magic bullets for modern security events, they are not. A skilled intrusion analyst can use these systems to increase their effectiveness, provide better security defenses, and leverage multiple layers of security to provide effective defense-in-depth capabilities. These technologies allow the analyst to spend more time on value added investigation activities versus low value data collection actives.

Even though they may appear easier to use, it is now more important than ever for analysts to pursue education on the fundamentals of how these technologies work. This will allow for the fair evaluation of these technologies and the impact of the configuration options to be known. Security professionals can leverage this to help provide more effective security defenses while making better use of security resources. Organizations that do not wish to invest in the skill set to manage these technologies can deploy and forget these technologies. Initially they will see an improvement, but that will degrade with time as the system ages. This fact needs to be clearly presented when a vendor is proposing NGIPS as a self-administering system.

References

Anwar, S., Zain, M. J., Zolkipli, M., Inayat, Z., Kahn, S., Anthony, B., & Chang, V. (2017). From Intrusion Detection to an Intrusion Response System: Fundamentals, Requirements, and Future Directions. *Algorithms*, 39.

Datt, S. (2016). *Learning Network Forensics.* Birmingham: Packet Publishing Ltd.

Kumar, D., Singh, M. K., & Jayanthi, M. (2016). *Network Security Attacks and Countermeasures.* Hershey: IGI Global.

NSS Labs. (2015). *NEXT GENERATION FIREWALL TEST REPORT: Fortinet FortiGate 3200D v5.2.4, build 5069.* Austin: NSS Labs.

NSS Labs. (2016). *Breach Detection Systems Test Report: Cisco Firepower 8120 with NGIPS v6.0 and Advanced Malware Protection v5.3.2016071117 .* Austin: NSS Labs.

NSS Labs. (2016). *NEXT GENERATION INTRUSION PREVENTION SYSTEM (NGIPS) REPORT: Trend Micro TippingPoint 7500NX v3.8.4.4525 – Tuned Policy.* Austin: NSS Labs.

SANS Institute. (2014, February). *Calculating Total Cost of Ownership on Intrusion Prevention Technology.* Retrieved from SANS Instituite InfoSec Reading Room: https://www.sans.org/reading-room/whitepapers/analyst/calculating-total-cost-ownership-intrusion-prevention-technology-34745

Steven, M. B. (2016). *Thinking Secruity: Stopping Next Year's Hackers.* Crawfordsville: Addison-Wesley Professional.

Stuart, D., & Beaver, K. (2013). *Next Generation IPS for Dummies.* Hoboken: John Wiley & Sons, Inc.

Appendix A

The technologies and techniques in this document were applied at a large public-sector education organization in the Midwest. Key characteristics of this organization are that it currently has approximately 70,000 devices on the network, over 100 sites, and utilizes an approximately 5 Gbps Internet link. Due to the nature of the organization, they are constrained on the number of employees and resources that they can spend on security. Complicating this is that their geographic location salary scale and training budgets are not conducive to attracting the talent level that a similarly sized commercial organization would attract. This organization has 2 Full Time Employee (FTE) equivalents in security while other smaller and similar size organizations in the area have 8 – 24 FTEs.

Modern attackers are attracted to this type of organization for financial crimes, crimes against children, and theft of computing resources. To complicate this issue, Internet technologies have become a key element of education, government testing, back office operations, and is considered critical to the organization. Traditionally a simple combination of traditional statefull firewalls, web/email content filtering, and traditional desktop antivirus was used. These solutions were not providing adequate protection against modern attacks as characterized by incidents of malware and intrusion incidents.

These characteristics drove to development of a risk driven security assessment and redesign of the organization’s security architecture. Key design considerations were deploying technology that would target low to mid-range attackers, leverage automation to reduce management labor, and provide a widely supportable platform. Several projects arose from this effort including the deployment of a NGIPS solution with 8 – 16 hours of weekly incident analyst time to support the solution. While one piece of a defense-in-depth strategy, this solution needed to clearly demonstrate its effectiveness in the environment by reducing perimeter threats.

The organization was found to have 1.77 billion requests made to and from the Internet every month. On average 194 Terabytes (TB) were transferred between the Internet and the organization. General monthly averages of 100 TB of HTTP and 19.2 TB of YouTube were consumed. Approximately 9 million files per month are transferred to and from the Internet. Review of this basic usage data exposed that 12 – 20% of the organization's bandwidth was being used by students that were using anonymous proxies, VPN proxies, and content evasion software to bypass content filtering rules. Additionally, roughly 60% of a Guest network was being consumed by Netflix and Hulu activity.

Deployment of application control to restrict anonymous proxy, VPN, and TOR traffic reduced bandwidth consumption by approximately 15%. While it cannot be quantitatively verified, a positive outcome on classroom learning is assumed to have resulted. Implementation of application filtering on the Guest network resulted in a 90% decrease of bandwidth consumption. The combination of these controls took less than 15 minutes of administrative overhead to implement and manage.

Deployment of the Threat Intelligence (TI) based blocks (Cisco terms Security Intelligence) averages 800,000 threats blocked on a monthly basis. The Intrusion Prevention System blocks on average 900 attacks per month that were not blocked by other sub-systems. File analysis filtering analyzes roughly 9 million files per month and blocks approximately 100,000 files that identified as malicious. Quantifying file analysis effectiveness can be difficult due to the TI component proactively blocking hostile sites before a file can be downloaded. Incidents of false positives have been very low with an average of 1 per month. Overall administration of this technology and intrusion analysis averages 12 hours per week to get these levels of results.

The general organization result has been a dramatic reduction in incidents of ransomware, malware, network latency, and circuit upgrades due to growth have been postponed. High value assets have additional security layers, and there have not been instances of users being negatively impacted by the technology. In general, the technology has had a major positive impact on the organization and their security posture. The ability to effectively communicate how connectivity is used and the effectiveness of the security controls has driven administration support.

Even though there are many positive factors of this technology, there were also several deficiencies and areas where other tools were needed to augment intrusion analysis. Due to the large datasets and slow system processing speeds, performing intrusion analysis for the GUI console tends to be extremely slow. Applying filters and isolating specifically targeted traffic tends to take much longer than it should. Vendor recommended database tuning did not resolve the issue and the systems is scaled to a point where limited systems resources were not an issue. The GUI interface tended to be very good at providing detailed dashboards, performance graphs, and allowing for downloading PCAP files from intrusion events.

To compensate for these issues, it is helpful to maintain a packet capture using a supplemental system. Due to the high volume of data transfer in this systems case, doing complete packet captures was not feasible. In the case of recurring events, a packet capture is configured on the NGIPS sensor to capture the traffic flow and then save it as a PCAP file. It can then be moved to an instance of Security Onion for further analysis with tools like bro, Wireshark, and other analysis tools.

Event detection is the final major issue that was identified. The solution does an excellent job at detecting anomalies and malicious traffic that is in their solution database. Items that are not within those confines are easily missed by the solution. This is where the intrusion analyst comes in as a component of the solution. They have the skill set that is necessary to review the various system logs and identify anomalous traffic that is traversing the network. Once the traffic is identified, they will be able to customize one of the system security components to block the offending traffic. The result of the solution is that it reduces the work required to perform adequate intrusion analysis by 60 – 70%; it does not remove the need for an intrusion analyst altogether.