North Star Software Developers

Company Training Manual

North Star Software Developers

Company Training Manual

Prepared by:
Brett Logan
Strategic Security Consulting Group

Executive Summary and Purpose

Executive Summary

North Star Software Developers operates out of Los Angeles, California, creating widely utilized software solutions. Having many customers, the security of customer data such as credit card numbers is of prime importance. Recently, a network server was compromised, potentially exposing this customer data and harming the reputation of NSSD. This breach revealed a vulnerability in detecting whether the attack vector was internal or external, and showcases the need for updated security policy and enhanced employee training.

Strategic Security Consulting Group has developed this manual to train NSSD IT personnel how to better protect the network to prevent future breaches from being successful. The manual will cover traffic analysis, firewalls, intrusion detection, vulnerability assessment, network assessment, and auditing/log collection.

Using Wireshark, IT personnel at NSSD will have the ability to better understand, and thus defend the network. Network defense is of prime importance in any organization, with the rate of cyber-attacks and data breaches constantly increasing. Some of the methods and strategies included are the ability to scan the network in search of vulnerabilities such as open ports, and also the ability to detect an adversarial network scan, indicating an attacker is probing the network. The combined information in all six of the following sections will create more able IT personnel, thus strengthening the overall security posture at NSSD.

0.1 Mitigation and Incident Response Assessment

A key part of training NSSD employees in the proper use of tools such as Wireshark is the increase in overall security this offers. The ability to monitor and understand the network creates faster detection and results in improved incident response: “there are many cases in which organizations did not know they had been hacked for months, and these situations could easily have been avoided. Proper security-oriented monitoring will alert the relevant people when any threat is encountered, allowing for immediate action to be taken” (Guim 2021). Portions of this manual such as firewall configuration and intrusion detection help mitigate security risks by preventing network breaches and alerting to attempted access and other suspicious activity. In the event of a successful breach, pre-determined incident response methods and strategies can help minimize damages and overall effects. An example of incident response plans from this manual is the collection and analysis of logs as covered in section six, which can reveal details of attack methods and allow personnel to address security vulnerabilities.

0.2 Purpose of This Manual

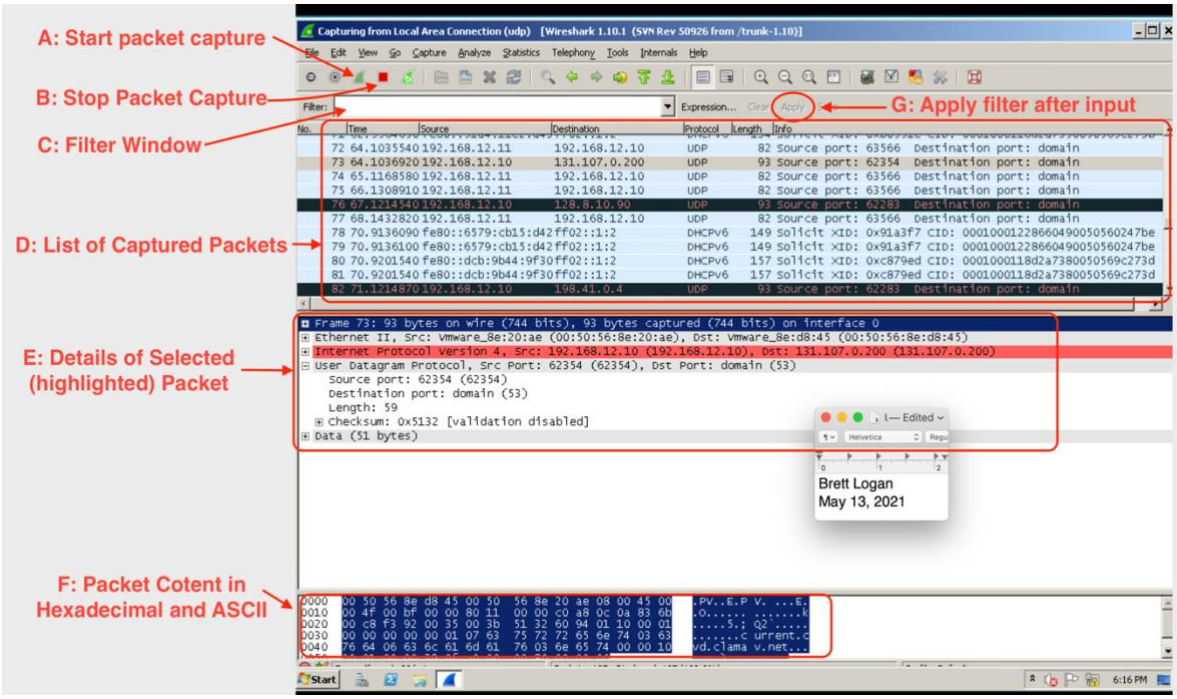
North Star Software Developers (NSSD) operates in a competitive environment where network security is of prime importance. Without proper protocols and precautions, NSSD is more vulnerable to threats such as hacking and data breaches, both of which can affect customers, employees, and stakeholders alike. Successful breaches can potentially cripple operations and open NSSD up to legal battles. Rapidly developing industry regulations make the protection of customer data a legal and ethical responsibility, and maintaining optimal network security policy helps meet these obligations.

To meet NSSD's strategic security goals of protecting the software development process and code, and sensitive client information from internal and external breaches, IT personnel must fully understand how to utilize various network security tools. This manual will explain these tools, and how to test, describe, and recommend them in various situations. By having this information and the ability to apply it appropriately, IT personnel will be able to mitigate the risk of any future breaches, as well as possible effects of a successful breach. The overall value of the policy changes stems from the directed knowledge imparted and how this affects the company. Knowledge is power, and preparedness is key; the purpose of this manual is to impart IT personnel with an understanding of available tools and how to wield them in defense of the NSSD network, resulting in a protected company that can focus on optimal operations and continued profitability.

Section One: Traffic Analysis

1.1 Packet Capturing

For IT personnel to best defend the NSSD network from attack, traffic analysis is conducted to establish baselines and identify anomalies: "traffic analysis is the process by which messages are intercepted and examined for the purpose of performance, security, and general network operation" (Hassan 2017). These messages, or packets, are the flow of information between two hosts on the network, and analyzing them in detail can provide valuable information pertaining to protocol used, IP addresses, MAC addresses, and much more. Doing so increases security by finding malicious or suspicious traffic. NSSD uses Wireshark to capture these packets for analysis, and IT personnel should be familiar with its application and use. Reference the following screenshot to become familiar with the Wireshark interface:



Instructions for capturing packets via Wireshark:

- Press green start button (A) to begin capture
- Press red stop button (B) to stop capture
- All captured packets can be found in the packet list (D)
- Filter for specific protocol by entering specification in filter window (C) and then pressing apply filter button (G)

- Further details of a selected packet header can be found in window (E)

1.2 Filtering through captured packets

When capturing packets via Wireshark, the amount of data collected can be staggering. Fortunately, “wireshark allows the user to quickly filter all that data, so you only see the parts you’re interested in, like a certain IP source or destination. You can even compare values, search for strings, hide unnecessary protocols and so on” (Profitap 2018). This can help IT personnel to identify the source of anomalous traffic and how it is interacting with the NSSD network.

The following commands can be used in the filter window to isolate commonly used protocols while searching for specific packet captures:

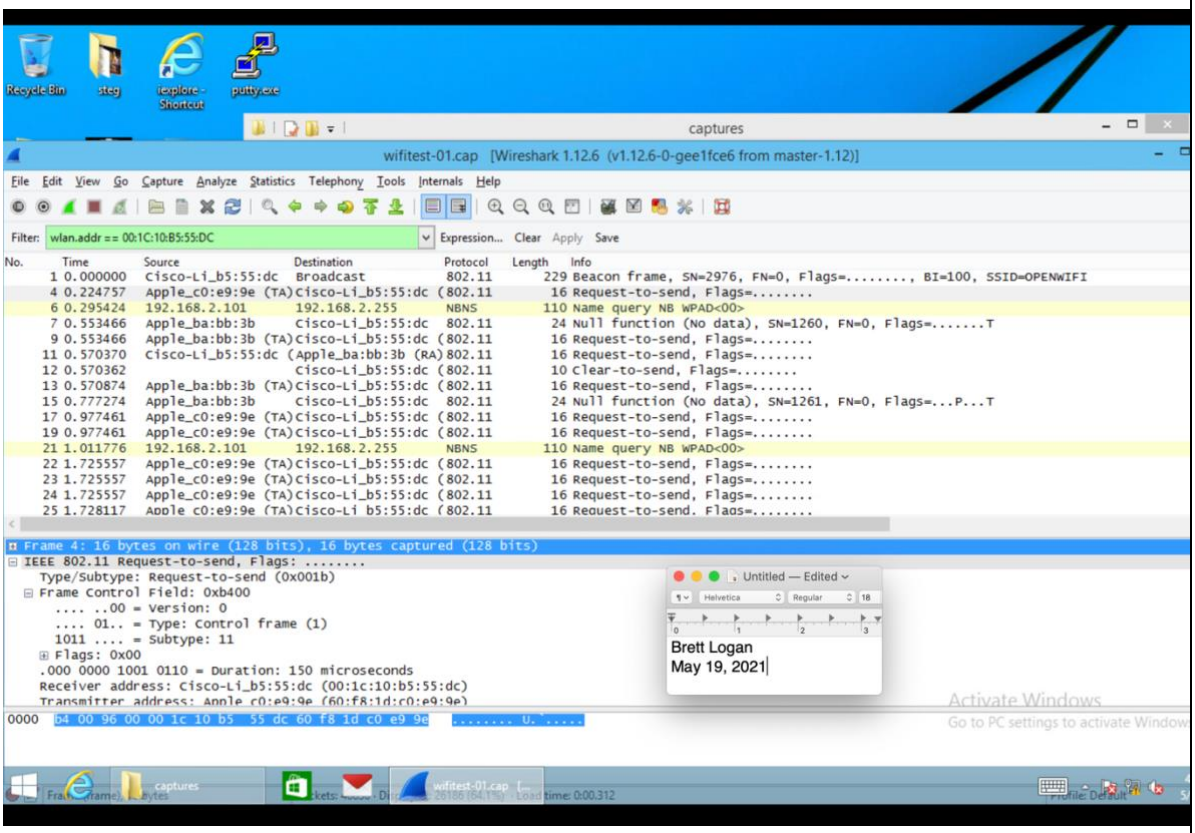
- *ipv6* for ipv6 traffic
- *ip and !ipv6* for ipv4 traffic
- *ip.addr == 224.0.0.0/8* for multicast traffic
- *ip.addr == 172.16.200.255* for broadcast traffic
- *icmp* for icmp traffic
- *arp* for arp trtaffic
- *tcp* for tcp traffic
- *udp* for udp traffic
- *ftp* for ftp traffic
- *smtp* for smtp traffic
- *dns* for dns traffic
- *tcp.port* = = 17 for qotd traffic
- *telnet* for telnet traffic
- *tcp.port* = = *19* for CHARGEN traffic
- *tcp.port* = = *13* for DAYTIME traffic
- *tcp.port* = = *7* for ECHO traffic
- *ssh* for secure shell traffic

- *rdp* for remote desktop traffic
- *smb* for server message block traffic
- *nbns* for NetBIOS Name Service traffic
- *http* for Hypertext Transfer Protocol traffic

Example A:

You are an IT personnel tasked with viewing NSSD's current wireless networks and connected devices in search of anything abnormal. Using Wireshark, you perform a packet capture as instructed above, and then find a router using SSID of OPENWIFI. To accomplish this, use the filter pane and apply the filter: *wlan.addr = = 00:1C:10:B5:55:DC,* resulting in a snapshot of all wireless traffic involving the Sisco Linksys router with the SSID of OPENWIFI.

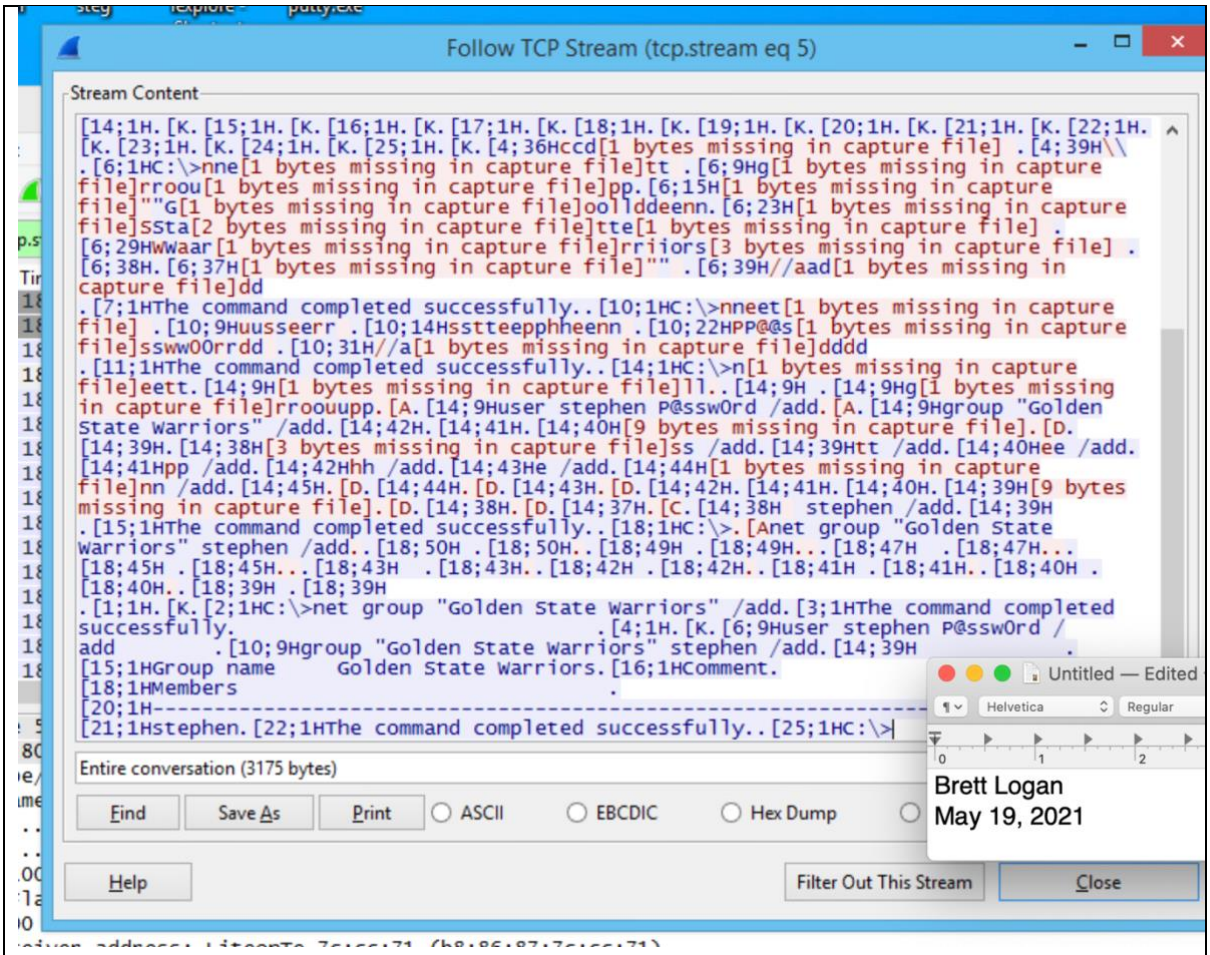
Your output should look something like this:



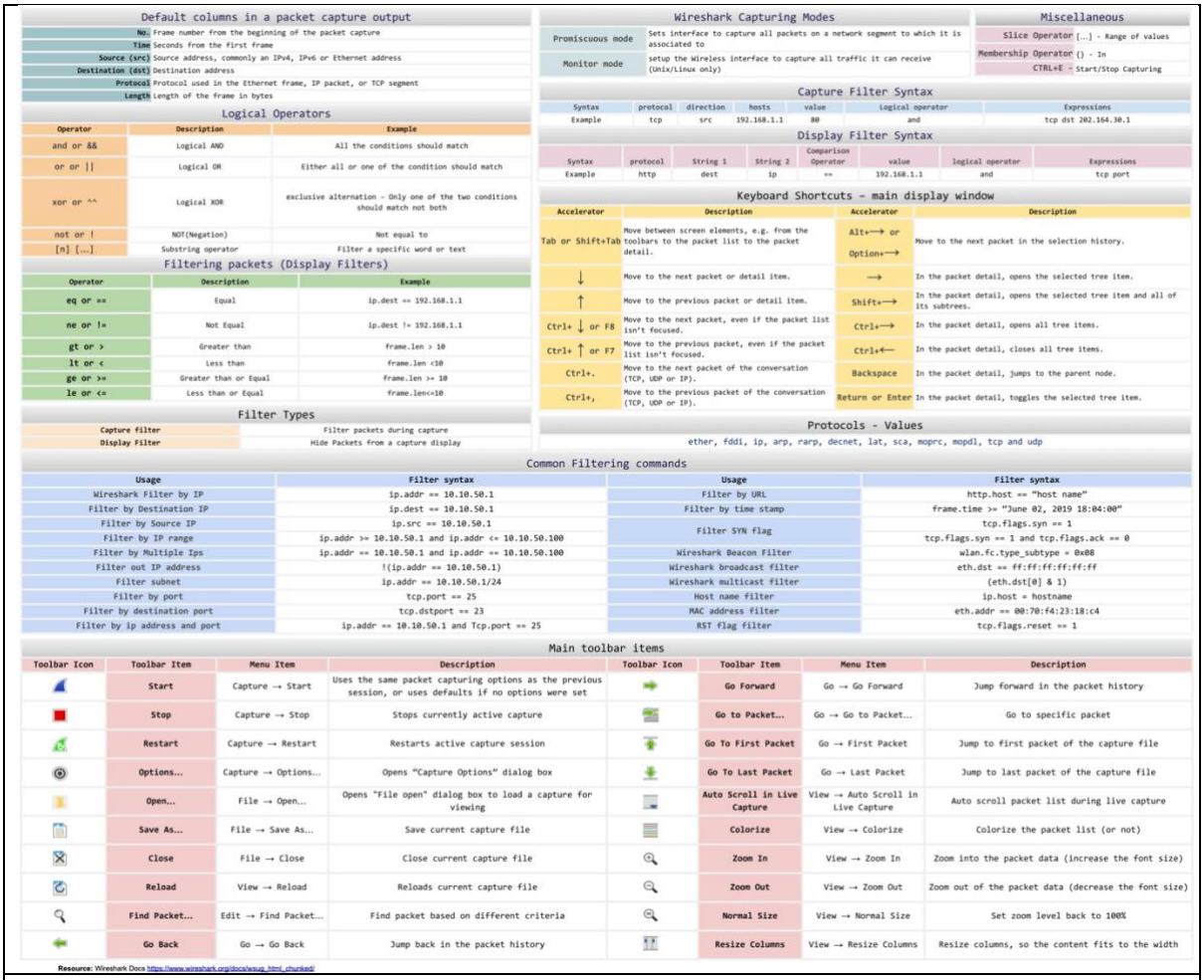
Example B:
While searching through wireshark packets, you need to locate more information by following the TCP stream of a capture. The steps to do so are:

- Perform Packet Capture
- Filter according to specific search purposes
- Highlight packet of interest from the list of captured packets
- *Right-click* on the packet
- Select *follow TCP stream*

This will open a new window with more detailed information, helping to identify things pertinent to security such as the addition of new groups/users. In this screenshot, note the addition of a new group (golden state warriors) and user (Stephen):



The following cheat sheet may also be helpful as you begin using Wireshark:



Default columns in a packet capture output

No.	Frame number from the beginning of the packet capture
Time	Seconds from the first frame
Source (src)	Source address, commonly an IPv4, IPv6 or Ethernet address
Destination (dst)	Destination address
Protocol	Protocol used in the Ethernet frame, IP packet, or TCP segment
Length	Length of the frame in bytes

Logical Operators

Operator	Description	Example
and or &&	Logical AND	All the conditions should match
or or ||	Logical OR	Either all or one of the condition should match
xor or ^^	Logical XOR	exclusive alternation - Only one of the two conditions should match not both
not or !	NOT(Negation)	Not equal to
[n] [...]	Substring operator	Filter a specific word or text

Filtering packets (Display Filters)

Operator	Description	Example
eq or ==	Equal	ip.dest == 192.168.1.1
ne or !=	Not Equal	ip.dest != 192.168.1.1
gt or >	Greater than	frame.len > 10
lt or <	Less than	frame.len <10
ge or >=	Greater than or Equal	frame.len >= 10
le or <=	Less than or Equal	frame.len<=10

Filter Types

Capture Filter	Filter packets during capture
Display Filter	Hide Packets from a capture display

Wireshark Capturing Modes

Promiscuous mode	Sets interface to capture all packets on a network segment to which it is associated to
Monitor mode	setup the Wireless interface to capture all traffic it can receive (Unix/Linux only)

Miscellaneous

Slice Operator	[...] - Range of values
Membership Operator	{} - In
CTRL+E	- Start/Stop Capturing

Capture Filter Syntax

Syntax	protocol	direction	hosts	value	Logical operator	Expressions
Example	tcp	src	192.168.1.1	80	and	tcp dst 202.164.30.1

Display Filter Syntax

Syntax	protocol	String 1	String 2	Comparison Operator	value	logical operator	Expressions
Example	http	dest	ip	==	192.168.1.1	and	tcp port

Keyboard Shortcuts - main display window

Accelerator	Description	Accelerator	Description
Tab or Shift+Tab	Move between screen elements, e.g. from the toolbars to the packet list to the packet detail.	Alt+⟶ or Option+⟶	Move to the next packet in the selection history.
↓	Move to the next packet or detail item.	⟶	In the packet detail, opens the selected tree item.
↑	Move to the previous packet or detail item.	Shift+⟶	In the packet detail, opens the selected tree item and all of its subtrees.
Ctrl+ ↓ or F8	Move to the next packet, even if the packet list isn't focused.	Ctrl+⟶	In the packet detail, opens all tree items.
Ctrl+ ↑ or F7	Move to the previous packet, even if the packet list isn't focused.	Ctrl+⟵	In the packet detail, closes all tree items.
Ctrl+.	Move to the next packet of the conversation (TCP, UDP or IP).	Backspace	In the packet detail, jumps to the parent node.
Ctrl+,	Move to the previous packet of the conversation (TCP, UDP or IP).	Return or Enter	In the packet detail, toggles the selected tree item.

Protocols - Values

ether, fddi, ip, arp, rarp, decnet, lat, sca, moprc, mopdl, tcp and udp

Common Filtering commands

Usage	Filter syntax	Usage	Filter syntax
Wireshark Filter by IP	ip.addr == 10.10.50.1	Filter by URL	http.host == "host name"
Filter by Destination IP	ip.dest == 10.10.50.1	Filter by time stamp	frame.time >= "June 02, 2019 18:04:00"
Filter by Source IP	ip.src == 10.10.50.1	Filter SYN flag	tcp.flags.syn == 1
Filter by IP range	ip.addr >= 10.10.50.1 and ip.addr <= 10.10.50.100		tcp.flags.syn == 1 and tcp.flags.ack == 0
Filter by Multiple Ips	ip.addr == 10.10.50.1 and ip.addr == 10.10.50.100	Wireshark Beacon Filter	wlan.fc.type_subtype = 0x08
Filter out IP address	!(ip.addr == 10.10.50.1)	Wireshark broadcast filter	eth.dst == ff:ff:ff:ff:ff:ff
Filter subnet	ip.addr == 10.10.50.1/24	Wireshark multicast filter	(eth.dst[0] & 1)
Filter by port	tcp.port == 25	Host name filter	ip.host = hostname
Filter by destination port	tcp.dstport == 23	MAC address filter	eth.addr == 00:70:f4:23:18:c4
Filter by ip address and port	ip.addr == 10.10.50.1 and Tcp.port == 25	RST flag filter	tcp.flags.reset == 1

Main toolbar items

Toolbar Icon	Toolbar Item	Menu Item	Description	Toolbar Icon	Toolbar Item	Menu Item	Description
	Start	Capture → Start	Uses the same packet capturing options as the previous session, or uses defaults if no options were set		Go Forward	Go → Go Forward	Jump forward in the packet history
	Stop	Capture → Stop	Stops currently active capture		Go to Packet...	Go → Go to Packet...	Go to specific packet
	Restart	Capture → Restart	Restarts active capture session		Go To First Packet	Go → First Packet	Jump to first packet of the capture file
	Options...	Capture → Options...	Opens "Capture Options" dialog box		Go To Last Packet	Go → Last Packet	Jump to last packet of the capture file
	Open...	File → Open...	Opens "File open" dialog box to load a capture for viewing		Auto Scroll in Live Capture	View → Auto Scroll in Live Capture	Auto scroll packet list during live capture
	Save As...	File → Save As...	Save current capture file		Colorize	View → Colorize	Colorize the packet list (or not)
	Close	File → Close	Close current capture file		Zoom In	View → Zoom In	Zoom into the packet data (increase the font size)
	Reload	View → Reload	Reloads current capture file		Zoom Out	View → Zoom Out	Zoom out of the packet data (decrease the font size)
	Find Packet...	Edit → Find Packet...	Find packet based on different criteria		Normal Size	View → Normal Size	Set zoom level back to 100%
	Go Back	Go → Go Back	Jump back in the packet history		Resize Columns	View → Resize Columns	Resize columns, so the content fits to the width

Resource: Wireshark Docs https://www.wireshark.org/docs/wsug_html_chunked/

(Keary 2020)

1.2 Alert Response Procedures

As part of an increased network security focus, NSSD has engaged in the use of an intrusion detection system (IDS) to monitor the network: “IDS are automated systems that monitor and analyze network traffic and generate "alerts" in response to activity that either match known patterns of malicious activities or is unusual. In some cases, alerts trigger further automated processes such as recording the suspect activity and/or scanning the computer(s) involved for signs of compromise” (Berkeley 2021).

To effectively respond to detected events, IT personnel are to utilize wireshark in investigating the potential threat. The following procedures are to be followed in the event of an IDS alert:

- IDS sends alert that anomalous traffic has been detected
- IT personnel immediately perform packet capture via wireshark
- In accordance with alert type, IT personnel will filter wireshark capture (for example, if IDS detects unusual http traffic, personnel shall filter packet capture by http)
- Once suspicious/unusual activity is located, a report should be generated and submitted to management for immediate review.

Significance of Alert Response

The appropriate response to security alerts could mean the difference between stopping a data breach or malware infestation before it is successful. Network monitoring establishes an activity baseline, and the IDS will help to identify any deviations from this baseline. It is up to IT personnel to then isolate and investigate the security threat. Wireshark is NSSD’s tool of choice for these investigations, and proper utilization will help the department quickly understand what threat is faced and where.

The response to these threats is as important and their identification. Following procedures results in a much quicker event reaction and has the potential to stop or reduce the severity of an attack.

Section Two: Firewalls

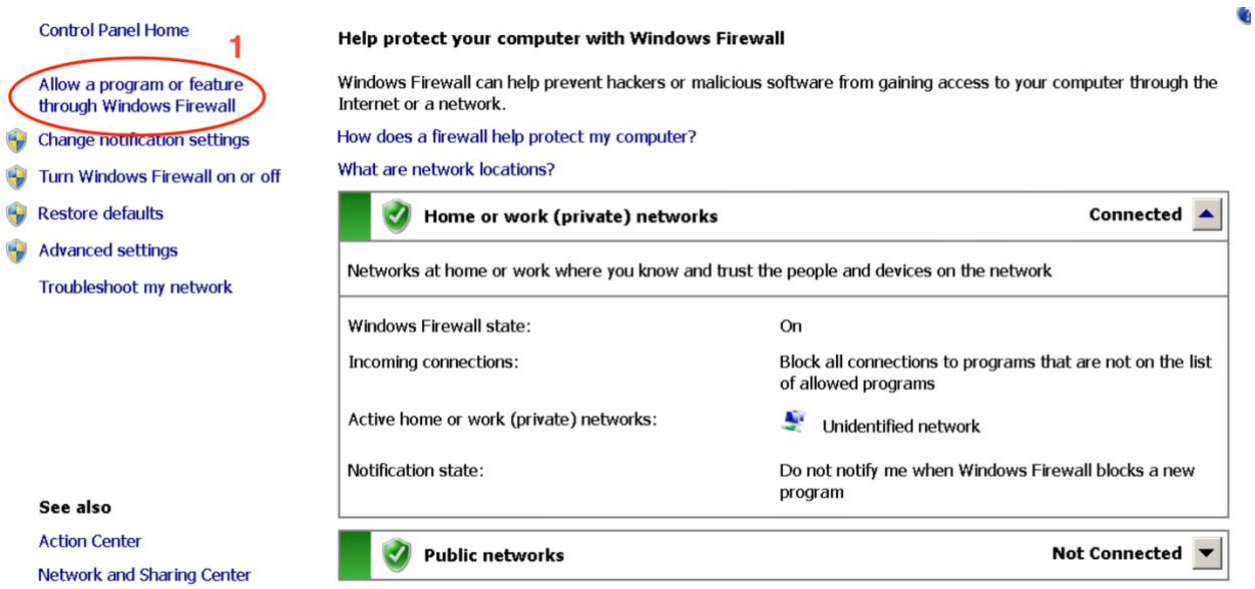
2.1 Firewall Fundamentals

One of the key components of NSSD's network security is the deployment of firewalls. A firewall monitors both incoming and outgoing traffic, and can be configured to block or allow traffic based on predetermined security rules. Another benefit of firewalls is that they can be used to create detailed reports and logs, both of which can help the IT department in troubleshooting issues, as well as maintaining regulatory compliance.

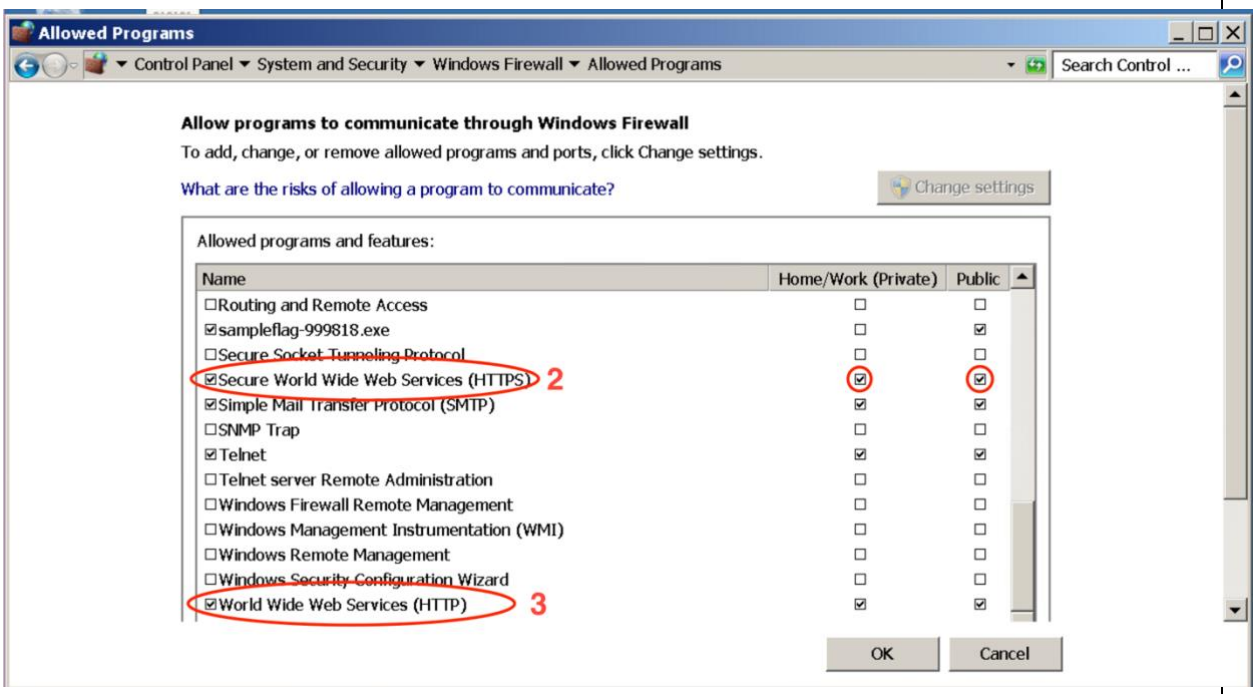
Among the firewall tools that NSSD uses is the Windows Firewall, a host-based firewall that provides protection for computers on our network. This type of firewall is especially helpful in preventing the spread of malicious software via network traffic, such as trojan horse attacks and worms. Due to the widespread use and applicability of Windows Firewall, IT personnel should familiarize themselves with enabling the firewall, rule creation, and how to block/allow/filter traffic.

Example B: Blocking HTTP and Allowing HTTPS traffic using Windows Firewall

First, you will access the Windows Firewall as outlined above in steps 1, 2, 3, and 4 of example A. This time, however, you will click on **Allow a program or feature through windows firewall** (1)



Next, ensure that **secure world wide web services (HTTPS)** (2) is *allowed* by clicking the box next to it and both private/public networks. Ensure the World Wide Web Service (HTTP) (3) is *blocked* by making sure it is *not selected.* Click OK to save these changes and you have successfully blocked HTTP traffic and allowed HTTPS traffic on this computer.

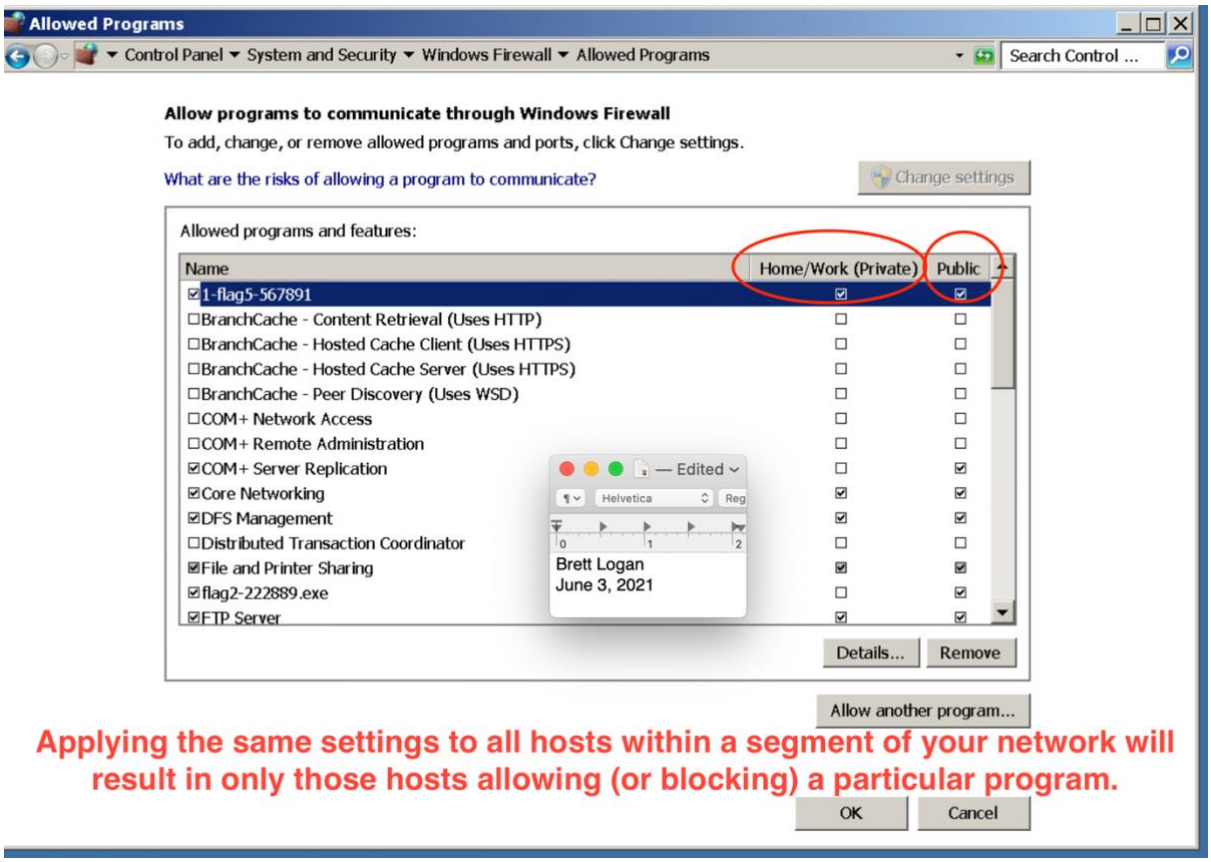


Network Segmentation

Especially with the increase in successful cyber-attacks across a wide swathe of businesses and industries, NSSD seeks to improve our network security through the concept of network segmentation: "network segmentation is the act of dividing a computer network into smaller physical or logical components. Two devices on the same network segment can then talk directly to each other. For communication to happen between segments, the traffic must flow through a router or firewall. This passage allows for traffic to be inspected and security policies to be applied" (NAS 2021). The creation of multiple subnets within a network makes it much more difficult for an attacker to access the network in its entirety, because firewalls between the segments are able to block traffic. As IT personnel, it is essential for you to understand the steps required for segmenting networks, which can be accomplished via Windows Firewall by configuring profiles: Domaine, Private, and Public. Each profile will protect connections according to their network location type, thus grouping different areas of the network into like location types will effectively segment it.

Example C: Windows Firewall Network Locations Public vs. Private

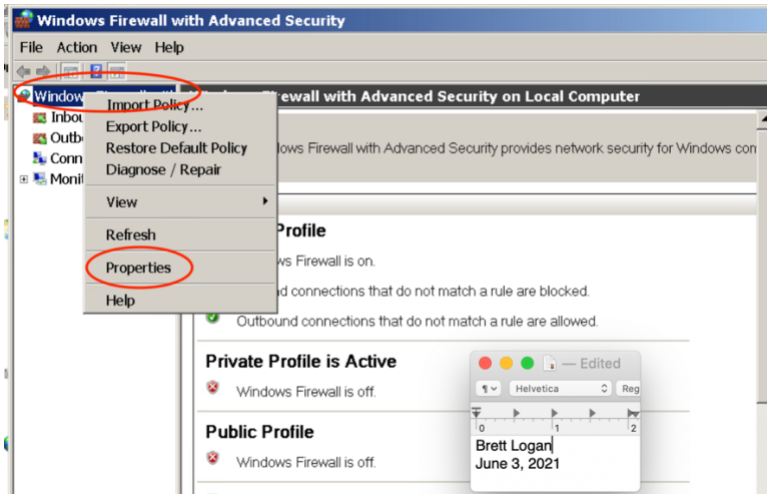
Recall the previous example's allowed programs list. To navigate back, open the firewall and click on **allowed programs and features**. Here you will find the columns are sorted by network location and contain two profiles: public and private. By checking the appropriate boxes, for example, you may allow only HTTP traffic on private profiles and only HTTPS on public profiles.


Allowed Programs
Control Panel ▸ System and Security ▸ Windows Firewall ▸ Allowed Programs
Search Control ...
Allow programs to communicate through Windows Firewall
To add, change, or remove allowed programs and ports, click Change settings.
What are the risks of allowing a program to communicate?
Change settings
Allowed programs and features:
Name
Home/Work (Private)
Public
1-flag5-567891
BranchCache - Content Retrieval (Uses HTTP)
BranchCache - Hosted Cache Client (Uses HTTPS)
BranchCache - Hosted Cache Server (Uses HTTPS)
BranchCache - Peer Discovery (Uses WSD)
COM+ Network Access
COM+ Remote Administration
COM+ Server Replication
Core Networking
DFS Management
Distributed Transaction Coordinator
File and Printer Sharing
flag2-222889.exe
FTP Server
— Edited
Helvetica
Brett Logan
June 3, 2021
Details...
Remove
Allow another program...
Applying the same settings to all hosts within a segment of your network will result in only those hosts allowing (or blocking) a particular program.
OK
Cancel

Guidelines for Implementation of Methods for Detecting Attacks

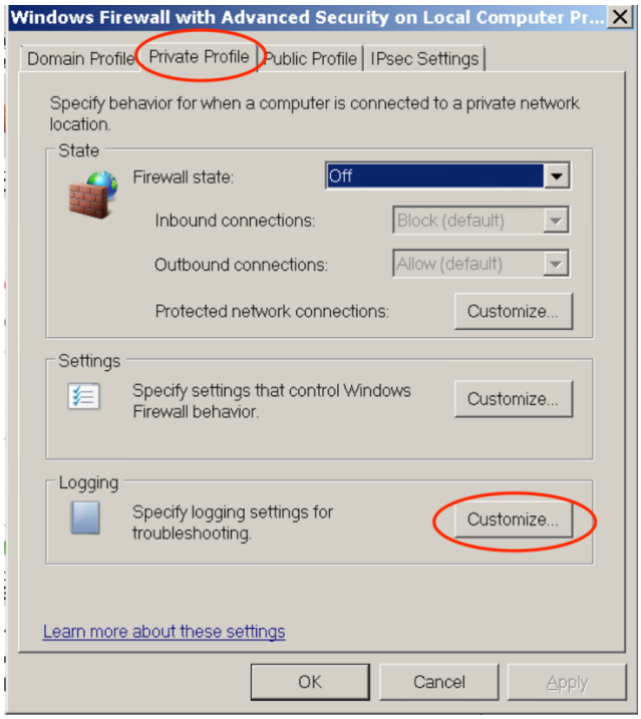
Many attacks can be identified by anomalous traffic on the network. One common tactic that attackers can use against companies is distributed denial of service, or DDoS. This method involves overwhelming a network with large amounts of requests, causing systems and services to shut down. IT personnel are encouraged to identify the potential for this type of occurrence by monitoring unusual firewall activity such as numerous failed access attempts. Firewall logs keep a record of all activity, and IT Personnel are able to set up logging in Windows Firewall by undertaking the following:

-Open Windows Firewall with Advanced Security. Right-Click on Windows Firewall with Advanced Security and go to Properties.



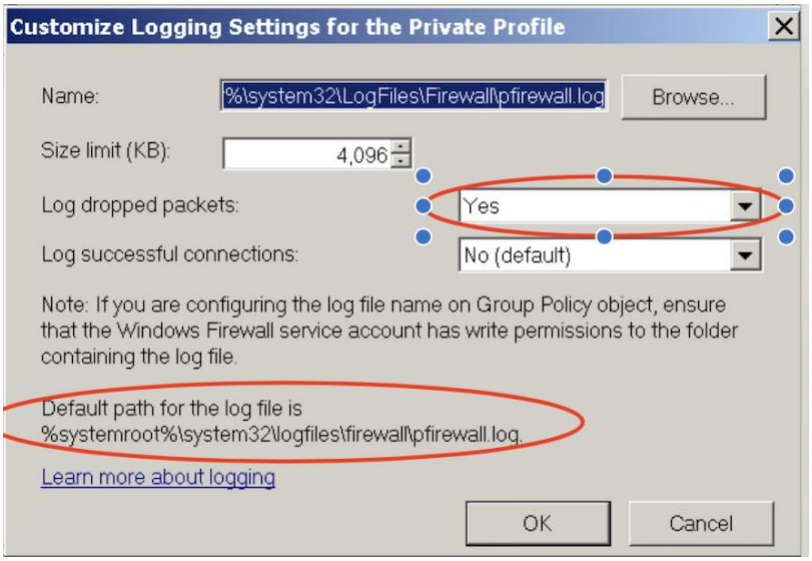
-Recalling the previously mentioned profiles, for this example choose *private*

-Click Private Profile > Logging > Customize



-Under log dropped packets, switch to *YES*

-copy the default path for the log file. (%systemroot%\system32\LogFiles\Firewall\pfirewall.log) and then Press OK



-Open File Explorer and Browse to where the Windows Firewall log is stored, you will see a pfirewall.log Copy this to your desktop to create a shortcut to the logs.

Creating a shortcut such as this makes reviewing and analyzing firewall logs easier, and IT Personnel are encouraged to familiarize yourselves with types of traffic and what typical logs look like. While this is just one method for detecting attacks using firewalls, it can be highly effective.

Section Three: Intrusion Detection and Prevention

3.1 Significance of Intrusion Detection and Prevention Systems (IDPS)

IDPS is a core piece of NSSD's overall cybersecurity posture. Our intrusion detection system allows the ability to monitor and analyze network traffic to uncover potential security incidents, violations, or threats, while the intrusion prevention system acts to stop those detected threats. Our IDPS relies on anomaly-based methods to detect attacks, for example "a profile for a network might show that Web activity comprises an average of 13% of network bandwidth at the Internet border during typical workday hours. The IDPS then uses statistical methods to compare the characteristics of current activity to thresholds related to the profile, such as detecting when Web activity comprises significantly more bandwidth than expected and alerting an administrator of the anomaly" (NIST 2021). This why all IT personnel must understand how to navigate and utilize our chosen intrusion detection system, Snort.

3.2 IDPS Tools and Methodology

A. *Snort:* Snort is a program that allows capture and analysis of traffic, helping to identify anomalies such as unwanted incoming or outgoing traffic. For IT personnel to effectively use snort, you must first use a sniffer to record traffic, and then send the captured traffic to a file to be analyzed.

*note: a sniffer should operate in promiscuous mode to see all network traffic. At NSSD this is accomplished by connecting the sniffer to the Switched Port Analyzer (SPAN) port.

B. *Linux distribution Kali 2:* this is the sniffer used to capture traffic and send it to a file for analysis using snort. The Linux/UNIX utility *tcpdump* is used by IT personnel to capture network traffic.

To view available switches for *tcpdump*, you can use the command:

root@kali2:~# tcpdump --help

```
root@kali2:~# tcpdump --help
tcpdump version 4.6.2
libpcap version 1.6.2
OpenSSL 1.0.1k 8 Jan 2015
Usage: tcpdump [-aAbdDefhHIJKlLnNOpqRStuUvxX#] [ -B size ] [ -c count ]
                [ -C file_size ] [ -E algo:secret ] [ -F file ] [ -G seconds ]
                [ -i interface ] [ -j tstamptype ] [ -M secret ] [ --number ]
                [ -Q in|out|inout ]
                [ -r file ] [ -s snaplen ] [ --time-stamp-precision precision ]
                [ -T type ] [ --version ] [ -V file ]
                [ -w file ] [ -W filecount ] [ -y datalinktype ] [ -z command ]
                [ -Z user ] [ expression ]
```

To run tcpdump on the network interface (eth0 for example) is connected to, use the command: **root@kali2:~# tcpdump –i eth0**

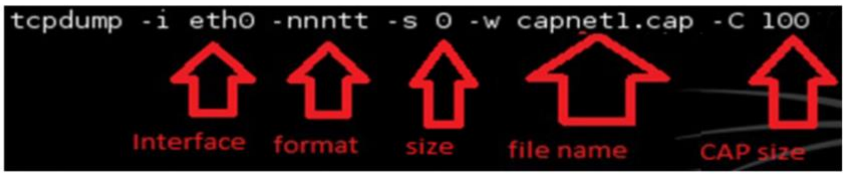
```
root@kali2:~# tcpdump -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
```

To stop capturing traffic via the sniffer, press: ***Ctrl + C***

To capture traffic and send it to a file for analysis, use command:
root@kali2:~# tcpdump –i eth0 -nnntt -s 0 -w capnet1.cap -C 100

And finally, to better understand the above syntax, please refer to the following table:

-i eth0	Use interface zero
-nnntt	Disable DNS resolution, date and time format
-s 0	Disables default packet size of 96 bytes, full packet size
-w	Write to a capture file, instead of displaying to the screen
-C	Split the captures into files of this size



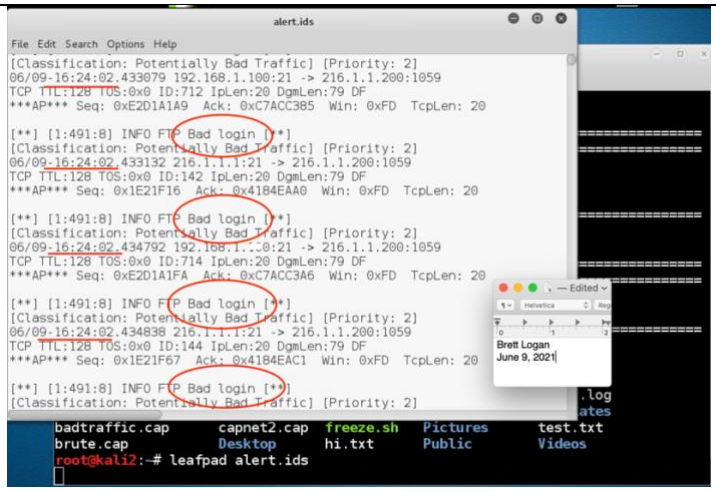
Example: Detect an attack using Snort to analyze network traffic log files.

In this example the NSSD network is under attack by a hacker using brute force tactics in an attempt to gain administrator account access.

Step 1: Use the following command to view a newly created alert.ids file: **root@kali2:~# ls**

Step 2: Use the following command to analyze the file: **root@kali2:~# leafpad alert.ids**

Step 3: Be on the lookout for clusters of alerts grouped close together, an indicator of attack. In this example the numerous repeated bad logins timed closely together are indicative of the brute force attack software trying various passwords in quick succession.



C. Whitelisting and Blacklisting

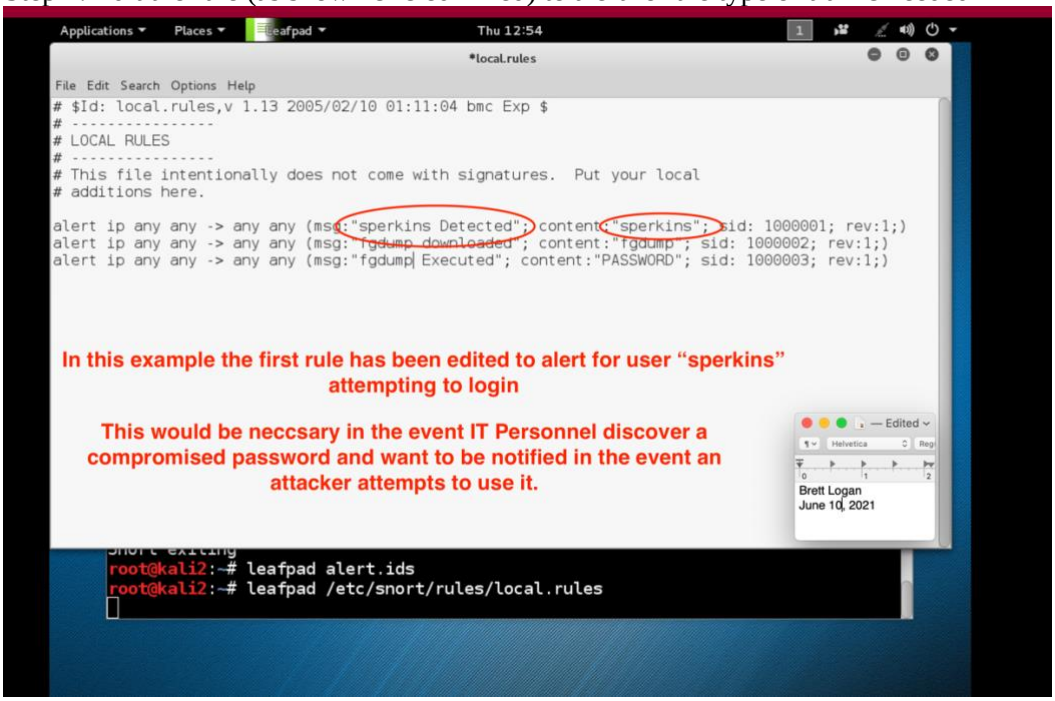
Two methods of managing network access are whitelisting and blacklisting. NSSD's IDPS utilizes these lists in an effort to prevent malicious activity and only allow access for approved users and programs. Whitelists are a list of benign entities that are allowed access, while on the opposite end of the spectrum blacklists are a list of entities that are blocked. These entities might include hosts, applications, usernames, specific ports, and file names/extensions. All NSSD IT personnel should be familiar and up to date with these lists, which can change frequently depending on updated threats and new users.

As you attempt to identify network traffic that may need to be whitelisted or blacklisted, it is helpful to create custom rules to alert for certain occurrences such as file name, account name, or output from an executed file. You can do so by editing the local.rules file in Snort.

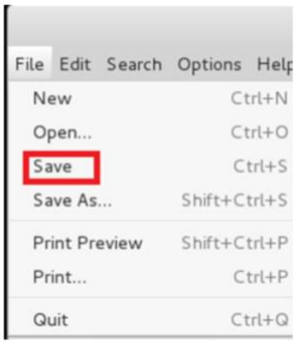
Example: Editing The local.rule File in Snort to Detect Particular Types of Traffic

Step 1: Open the local rules file by entering the following command:
root@kali2:~# leafpad /etc/snort/rules/local.rules

Step 2: Edit the rule (as shown circled in red) to alert for the type of traffic needed



Step 3: Be sure to save the edit by clicking **File→Save**



Congratulations, you have now successfully edited a rule to look for certain types of traffic on the NSSD network. As you can see, the combination of sniffers, log analysis, and custom rule editing enables IT Personnel to exercise greater control and utilization of our intrusion detection and prevention systems, thus keeping our network safe.

Section Four: Vulnerability Assessment

Vulnerability scans are an important part of NSSD's network defense because they provide feedback on the health and security of the network, helping identify abnormal behaviors and vulnerable devices.

Using Wireshark, IT employees may discover malicious activity of an attacker based on the output after certain filters are applied. These outputs are a result of a potential attacker running various scans in an attempt to breach our network, typically by finding open ports. As part of a vulnerability assessment, IT personnel may also use the same nmap scanning methods that a would-be attacker does. This gives critical insight into network vulnerabilities that security teams can then mitigate.

4.1: NMAP Scanning and Detecting

The following chart shows the scan technique, the command used to run the scan, and the Wireshark filter which will reveal resulting traffic to indicate the scan has taken place. The <target> section of the command is replaced with the desired IP address.

Technique Used	Wireshark Filter	Command
TCP SYN	tcp.flags.syn==1 and tcp.flags.ack==0 and tcp.window_size<=1024	nmap -sS <target>
TCP connect ()	tcp.flags.syn==1 and tcp.flags.ack==0 and tcp.window_size>1024	nmap -sT <target>
TCP Null	tcp.flags==0	nmap -sN <target>
TCP FIN	tcp.flags==0x001	nmap -sF <target>
TCP Xmass	tcp.flags.fin==1 && tcp.flags.push==1 && tcp.flags.urg==1	nmap -sX <target>

UDP port	icmp.type==3 and icmp.code==3	nmap -sU <target>

(Infosec 2021)

Summary of scan technique and purpose:

TCP SYN scan: detection of numerous packets in a short time indicates an attacker may be running SYN scans, SYN port sweeps, or SYN floods.

TCP connect () scan: identical to SYN scan, but with a larger window size revealing TCP connection which indicates an attacker may be conducting port scans or sweeps.

TCP Null: An attacker might send packets without flags set in an attempt to discover open ports.

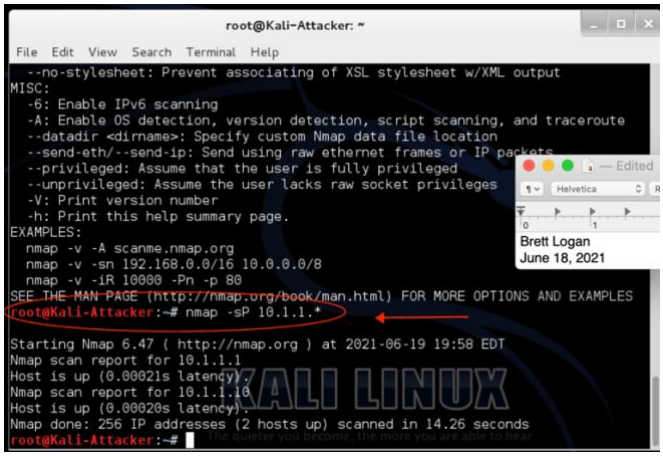
TCP FIN: An attacker might also set only the FIN flag in attempt to bypass firewalls and discover open ports.

TCP Xmass: Another way an attacker may seek to discover open ports is by sending packets with FIN, PUSH, and URG flags set.

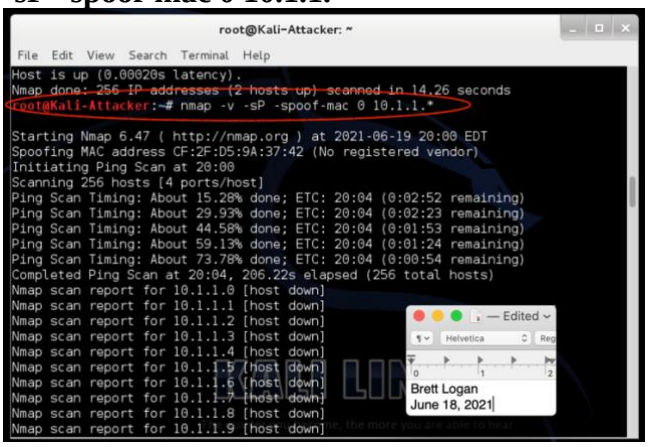
UDP Port scan: A high number of ICMP packets showing port/destination unreachable may indicate the attacker is running UDP port scans.

It is helpful to think like a would-be attacker might, trying various methods to poke and prod the network in search of any openings or weak points. Following are examples using nmap to conduct network vulnerability tests.

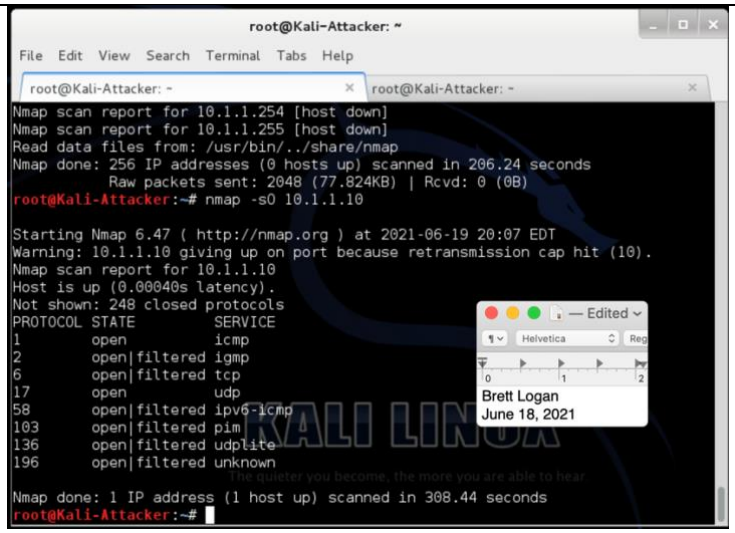
To start, one might seek to identify any live hosts in the network, which can be accomplished via ping scan. For example, to identify all live hosts with a network ID of 10.1.1.* you would use the command: root@Kali-Attacker:~# **nmap –sP 10.1.1.***



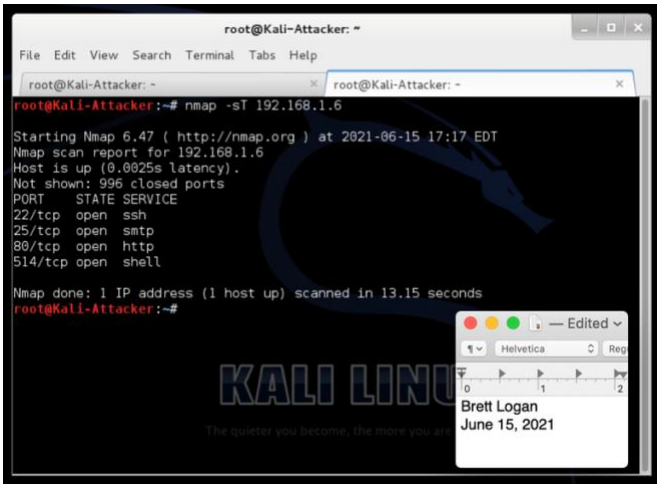
To initiate a ping scan while simultaneously spoofing the source MAC address, use command **nmap –v –sP –spoof-mac 0 10.1.1.***



To scan for active systems on the network, use command **nmap –sO 10.1.1.10**

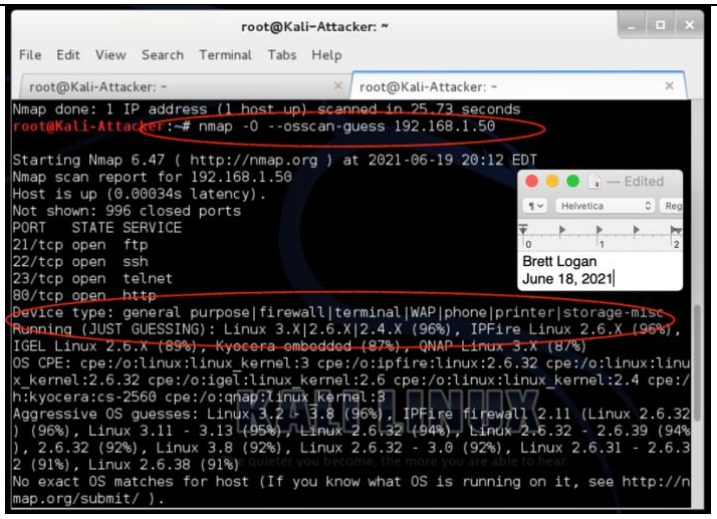


In the following image, the ***nmap -sT 192.168.1.6*** command is used to scan the network for vulnerable systems (note this command is a TCP connect () scan as indicated in the chart above). The output indicates that TCP ports 22, 25, 80, and 514 are all open.



NMAP may even be used to discover what operating system and device is in use by making approximate guesses until the correct one is discovered using command:

nmap –O --osscan-guess 192.168.1.50



These types of scans can be detected in Wireshark based upon certain traffic patterns they create.

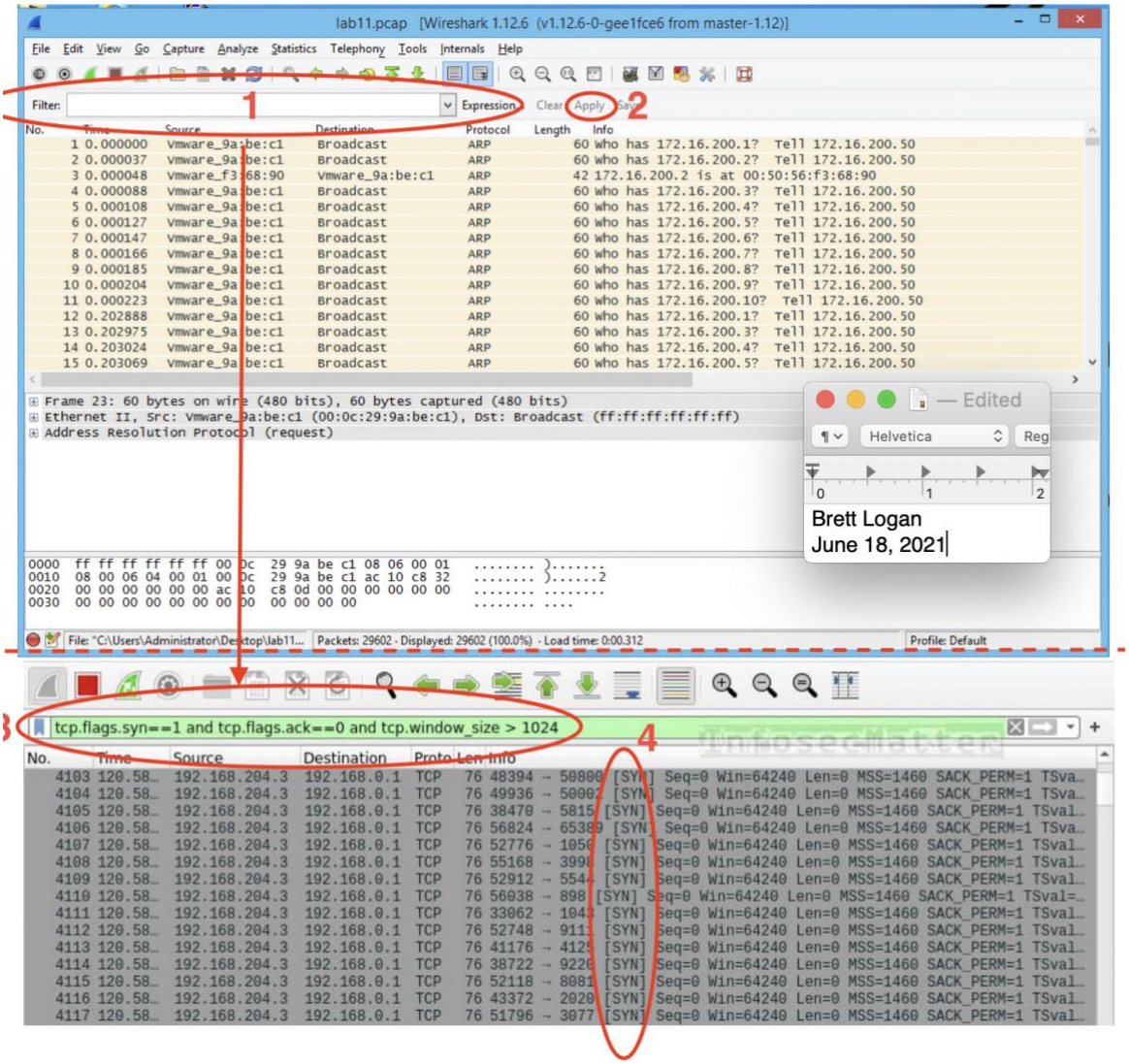
Example A: Detecting SYN scans using Wireshark

Step 1: After opening Wireshark and running capture, enter filter as shown in chart above

Step 2: Click *Apply*

Step 3: Correct filter syntax will appear with a green background. In this instance we are filtering out packets with Syn flag not set, ACK flag not set, and window size <= 1024 bytes

Step 4: The scan shows a high amount of SYN packets, indicating an attacker may be running nmap -sS <target> to run SYN scans in the network.



4.2 Detecting Network Attacks Using Wireshark

Just as various port scans can be detected using Wireshark filters, so too can potential network attacks. IT personnel are instructed to use the following filters in Wireshark to reveal corresponding attacks

Technique	Wireshark Filter	Command/Tool
ARP poisoning	arp.duplicate-address-detected or arp.duplicate-address-frame	arpspoof, ettercap
ICMP flood	icmp and data.len > 48	fping, hping

VLAN hoping	dtp or vlan.too_many_tags	frogger, yersinia
Unexplained packet loss	tcp.analysis.lost_segment or tcp.analysis.retransmission	Not Applicable

(Infosec 2021)

Summary of scan technique and purpose:

ARP Poisoning: Also known as ARP spoofing, this allows an attacker to initiate a man-in-the-middle attack. Using this filter will reveal if a single IP address is being claimed by more than one MAC address, a likely indicator of such attack.

ICMP flood: By filtering for large data size, IT personnel can detect oversized ICMP packets indicative of an attacker attempting a denial-of-service attack.

VLAN hoping: This filter can reveal packets tagged with multiple VLAN tags, indicating an attacker's attempt at bypassing network access controls.

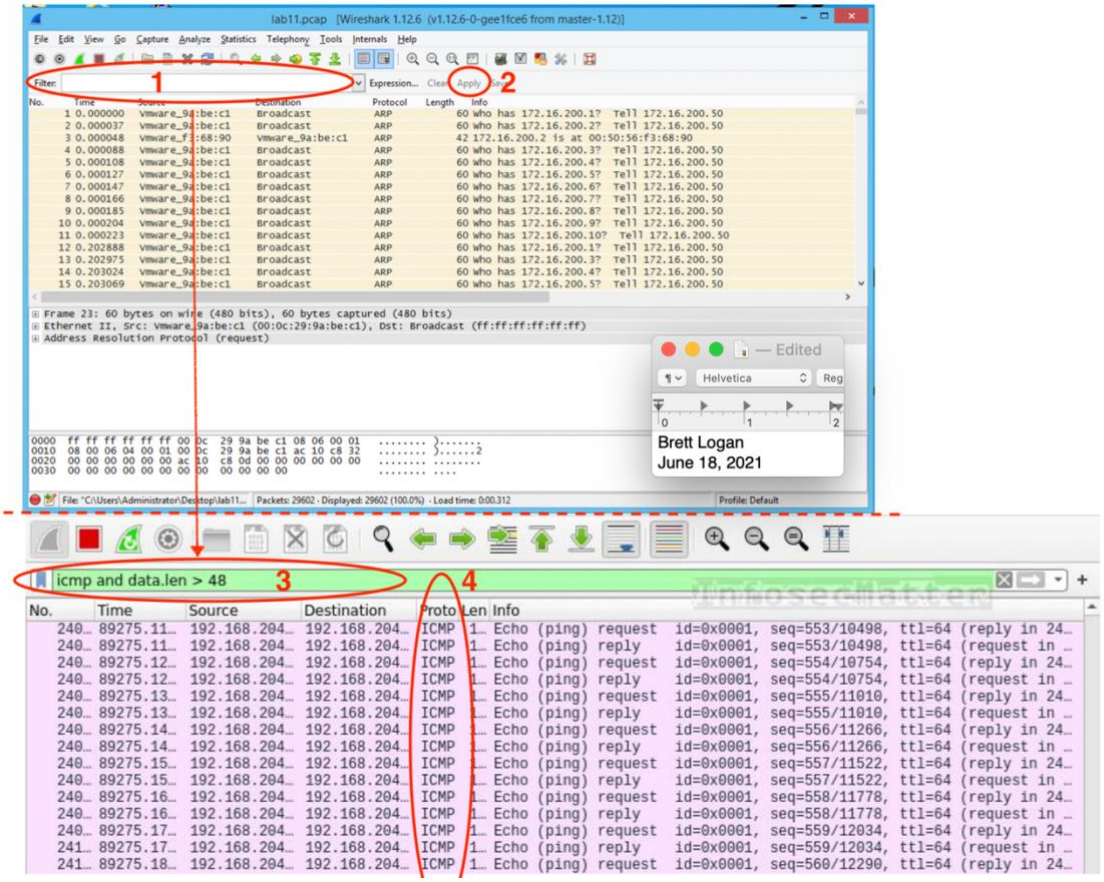
Example B: Detecting Denial-of-Service Attack Using Wireshark

Step 1: After opening Wireshark and running capture, enter filter as shown in chart above

Step 2: Click *Apply*

Step 3: Again, if the syntax is correct, filter background should appear green. With this filter you are filtering for packets with larger than 48 bytes of data.

Step 4: The protocol section of Wireshark reveals a large number of ICMP packets, a good indication that an attacker is flooding the network intentionally to perform a denial-of-service attack.



*dotted line to indicate top portion from lab and bottom portion from infosecmatter

As you can see, there are many uses of NMAP in vulnerability assessment, as well as corresponding Wireshark filters to detect such activity if it is malicious rather than being performed for assessment purposes.

Using these tools helps identify and rectify any weaknesses or misconfigurations in the NSSD network.

Section Five: Network Scanning and Assessment

Network scanning is an essential piece of NSSD's overall cybersecurity program. Scans can help IT personnel protect the network from attack by identifying weaknesses or vulnerabilities. Often an attacker will initiate a scan themselves to surveil the network before attack, and the identification of such scans can help analysists understand the method and route of attack. Scan results can offer details such as

-IP Addresses

-Open Ports

-Operating System

-Vulnerabilities

5.1 Network Scanning

Using the filter in Wireshark allows detection of malicious activity based upon certain atypical traffic which reveals that a scan has taken place. The following scan types and their corresponding Wireshark filters are a valuable tool:

Ping Sweep

This type of scan reveals which IP addresses are active on the network. To detect the presence of such scans using Wireshark, the following filters are used:

Detecting an icmp ping sweep

icmp.type==8

icmp.type==0

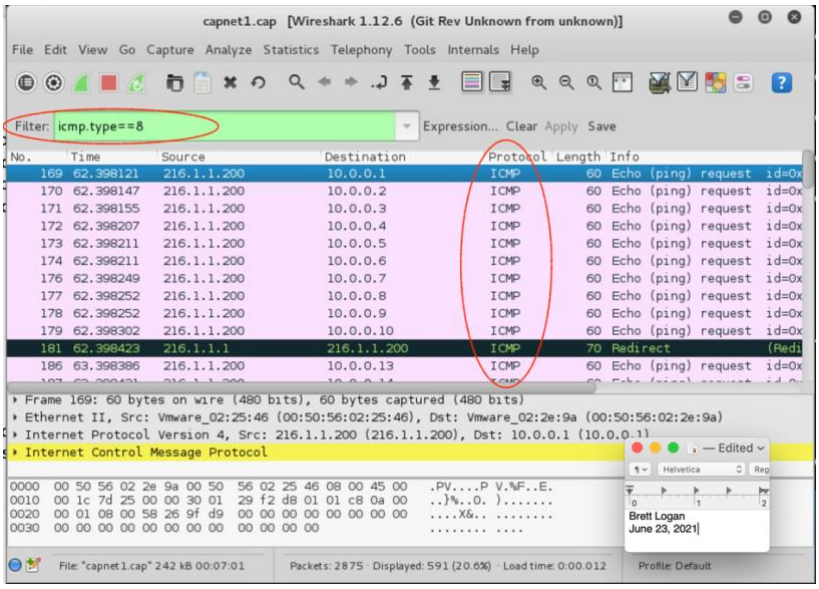
Detecting a TCP ping sweep

tcp.dstport==7

Detecting a UDP ping sweep

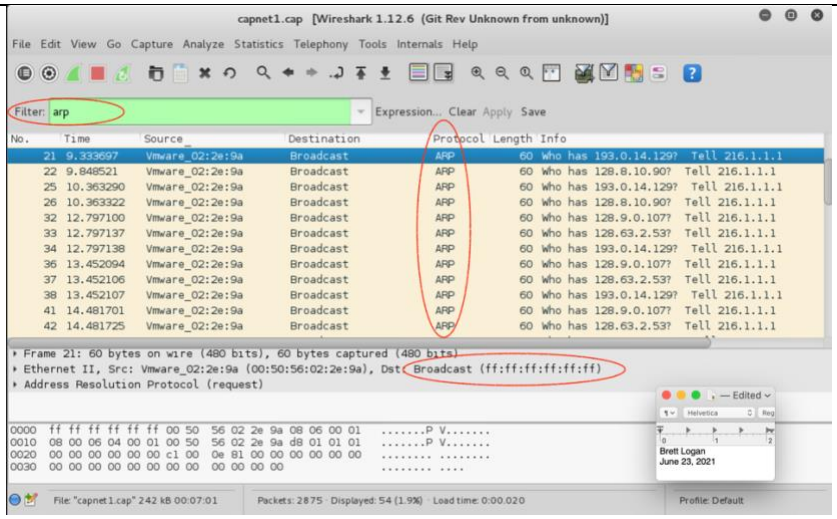
udp.dstport==7

An unexpected increase in packets as revealed by these filters may indicate a ping sweep.



ARP Sweep

When filtering by ARP in Wireshark, personnel can look for an ARP sweep which indicates an attacker has sent an ARP broadcast for every IP in a subnet, hoping for a response which would indicate which IP addresses are active. The signature of this scan type is the use of “ff:ff:ff:ff:ff:ff” for the destination MAC which is used for broadcast.



Stealth Scan/TCP Half Open

Used to detect which ports are open or closed, an attacker can initiate a TCP connection by sending a SYN packet to the targeted port. Open ports will respond with SYN+ACK and closed ports will respond with RST of RST+ACK. When the attacker discovers an open port, they will respond with RST to avoid making an actual TCP connection. In Wireshark, the signs of this type of scanning include numerous TCP sessions with less than 4 packet communications, and/or many RST packets.

TCP Full Connect Scan

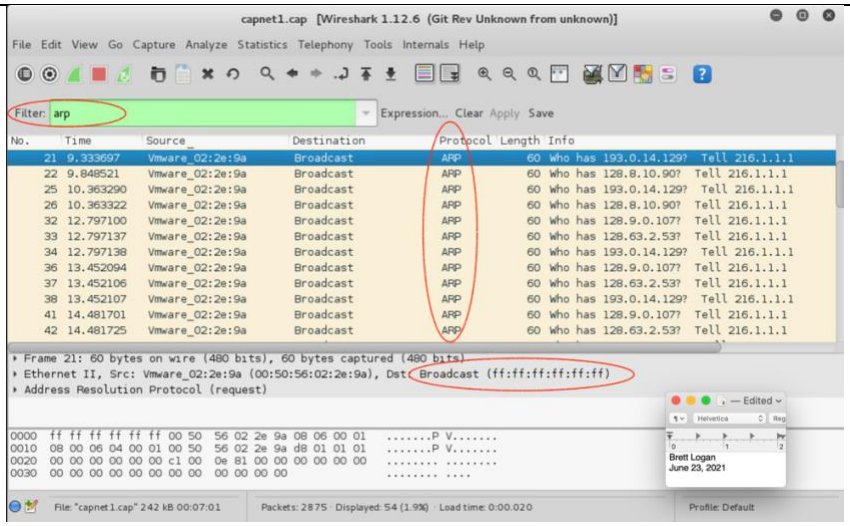
Unlike a stealth scan, this method results in the attacker responding to the TCP+ACK with ACK to establish a TCP connection and then terminate it. The following Wireshark filters can be utilized to find SYN, SYN+ACK,RST, and RST+ACK packets:

tcp.flags==0x002

tcp.flags==0x012

tcp.flags==0x004

tcp.flags==0x014



TCP Null Scan

Here the attacker sends a TCP packet with no flags to look for an RST response from closed ports. To look for this type of scan in Wireshark use filter:

tcp.flags==0x000

UDP Scan

A high number of packets with ICMP type 3 Code 3 can indicate an attacker sending UDP packets to a port to discover whether it is open or closed (this response indicates to the attacker that the port is closed). To search for this type of scanning, use Wireshark filter:

icmp.type==3

icmp.code==3

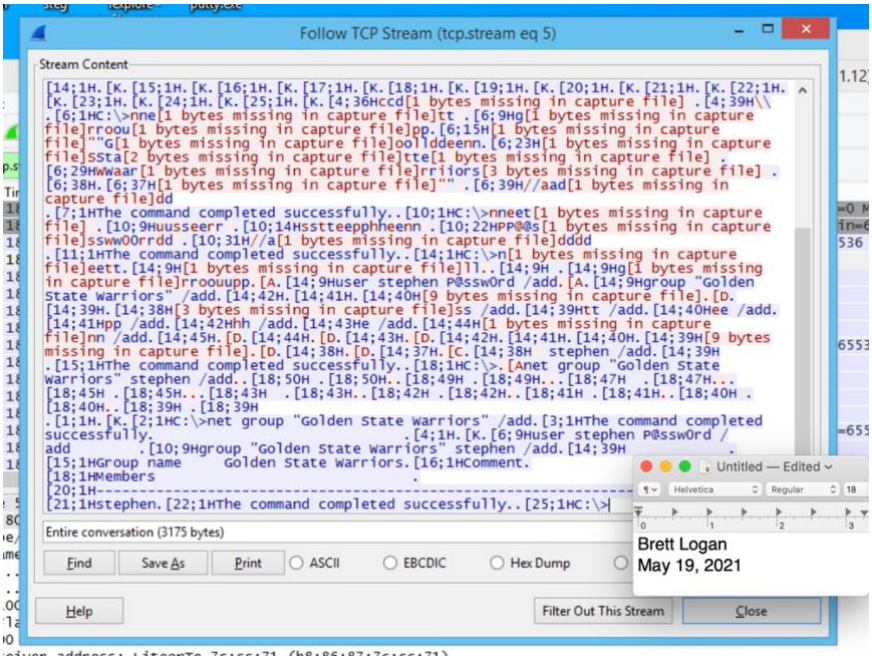
5.2 Network Assessment Using Wireshark

Wireshark may also be used by IT Personnel to detect things such as rogue devices and malicious connections. To search for this type of occurrence, follow these directions:

1: Start Wireshark Capture as directed in Section 1 of this manual

2: Review Captured Packets, paying close attention to *unknown* MAC or IP addresses

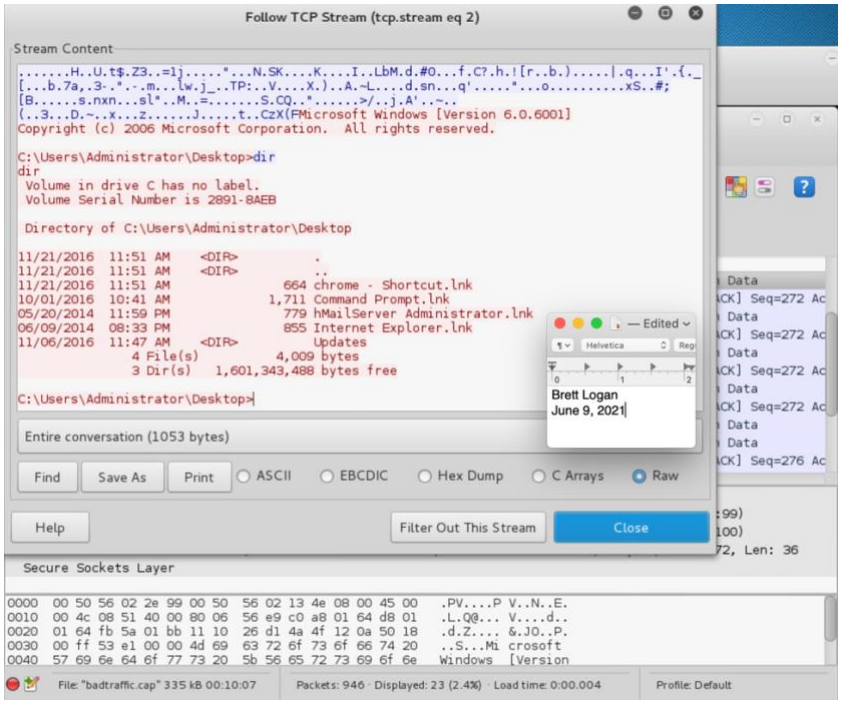
3: Upon discovery of suspicious packet, perform R*ight Click→Follow→TCP Stream*



*Note that blue indicates data from server to client, while red indicates data from client to server.

4: Review this data to determine the specific actions taking place on the network and whether they are atypical or not.

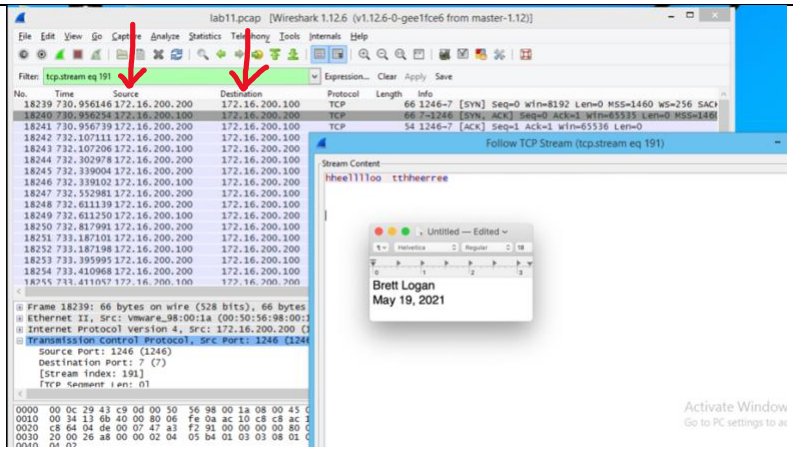
Example: In this image IT Personnel can discover unwanted outgoing traffic, indicating a malicious actor has breached the network.



5.3 Viewing Network Topologies

While Wireshark doesn't specifically offer a topology illustration, it is still possible to determine network topology by monitoring traffic and paying close attention to IP/MAC addresses within the network. Monitoring traffic between these addresses helps to create a picture of network locations and establish norms for network activity.

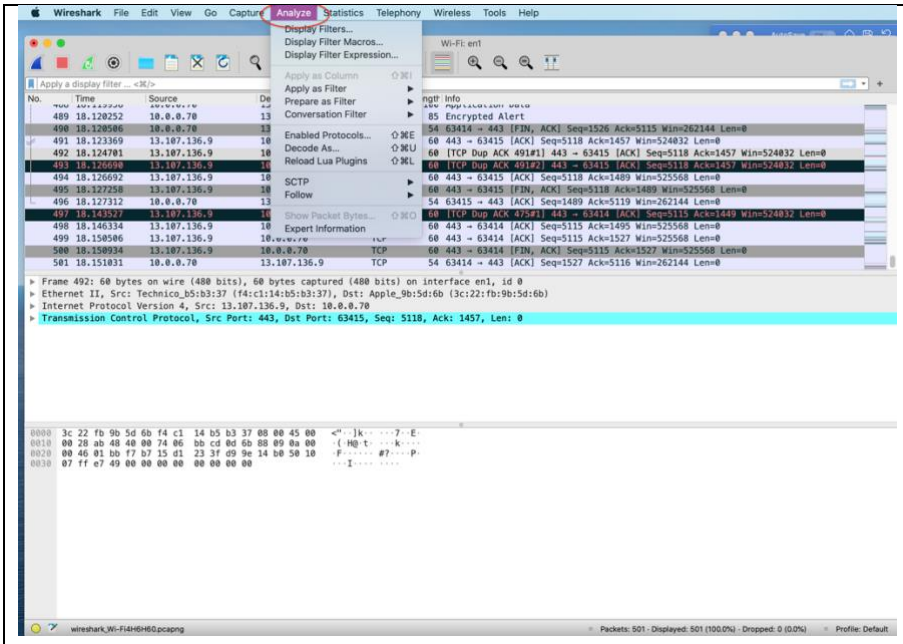
Note the source and destination columns in your Wireshark Capture revealing IP addresses:



Section Six: Auditing and Log Collection

Network auditing is a key responsibility of IT personnel at NSSD. A properly executed audit provides information such as security risks, allowing for changes to the network to keep our data safe. These audits also help map and inventory the network to ensure visibility and proper administration efforts.

When auditing to diagnosing a specific network problem, Wireshark is especially useful due to the specificity of data that it reveals. Among the tools that personnel can use to garner deeper levels of information is the analyze menu:



A breakdown of menu items and their description is as follows:

Menu Item	**Accelerator**	**Description**
Display Filters…		Displays a dialog box that allows you to create and edit display filters. You can name filters, and you can save them for future use.
Display Filter Macros…		Shows a dialog box that allows you to create and edit display filter macros. You can name filter macros, and you can save them for future use.
Apply as Column	**Shift+Ctrl+I**	Adds the selected protocol item in the packet details pane as a column to the packet list.

Apply as Filter		Change the current display filter and apply it immediately. Depending on the chosen menu item, the current display filter string will be replaced or appended to by the selected protocol field in the packet details pane.
Prepare as Filter		Change the current display filter but won't apply it. Depending on the chosen menu item, the current display filter string will be replaced or appended to by the selected protocol field in the packet details pane.
Conversation Filter		Apply a conversation filter for various protocols.
Enabled Protocols…	**Shift**+**Ctrl**+**E**	Enable or disable various protocol dissectors..
Decode As…		Decode certain packets as a particular protocol..
Follow → TCP Stream		Open a window that displays all the TCP segments captured that are on the same TCP connection as a selected packet..
Follow → UDP Stream		Same functionality as "Follow TCP Stream" but for UDP "streams".

Follow → TLS Stream		Same functionality as "Follow TCP Stream" but for TLS or SSL streams. See the wiki page on TLS for instructions on providing TLS keys.
Follow → HTTP Stream		Same functionality as "Follow TCP Stream" but for HTTP streams.
Expert Info		Open a window showing expert information found in the capture. Some protocol dissectors add packet detail items for notable or unusual behavior, such as invalid checksums or retransmissions. Those items are shown here. The amount of information will vary depend on the protocol

(Wireshark 2021)

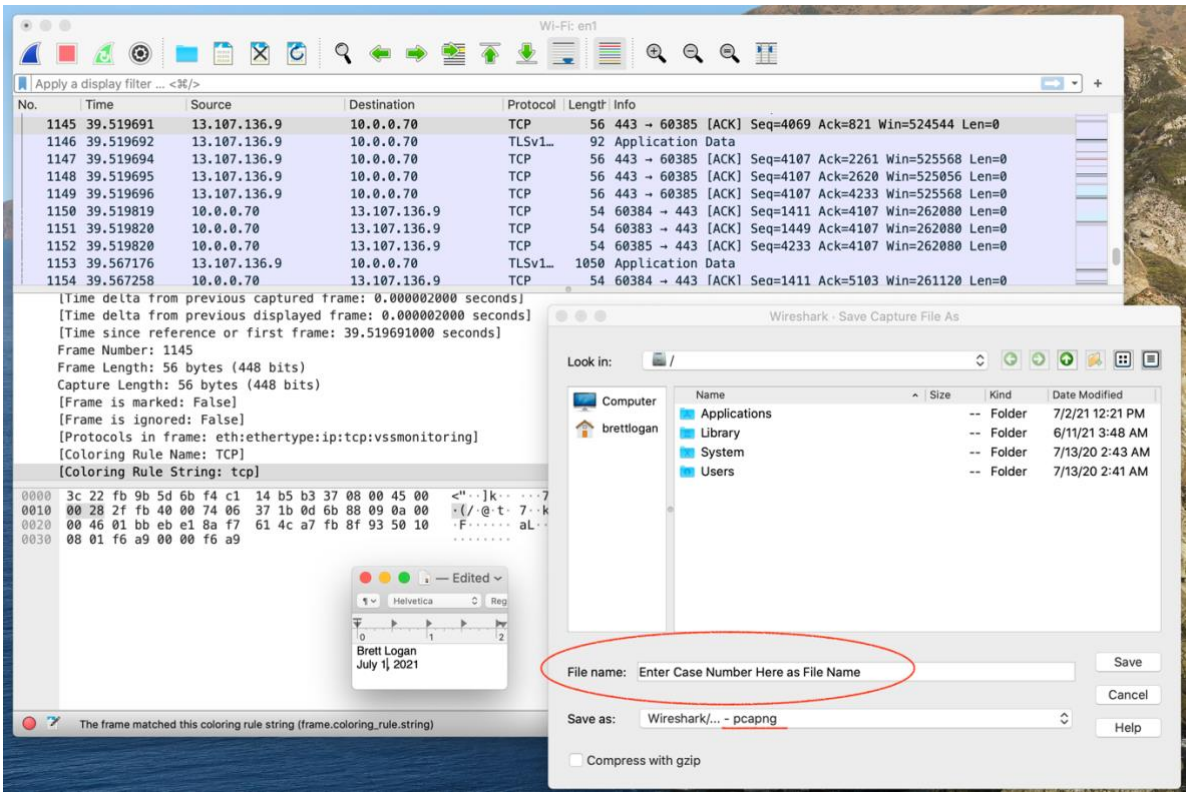
Loopback Traffic

Useful for diagnostics and troubleshooting, loopback traffic is the communications a device has with itself. To view this traffic, be sure to install ncap along with Wireshark. This creates the loopback interface, and on the initial Wireshark screen select and double click the *adapter for loopback traffic capture*. This traffic can be analyzed like other captured traffic using the above menu to gain greater detail.

Creating/Saving Logs

Logs are useful for IT personnel because they allow revisiting of network activity, giving the ability to search, troubleshoot, and compare. Using Wireshark, follow these directions to save captured traffic in files for later use and regulatory compliance:

1. Open Wireshark
2. Run Capture
3. If there is a known issue of concern on the network, reproduce it
4. Stop the capture
5. Click File→Save As and enter file name (such as case number)
6. The log file will be saved in .pcapng format



Troubleshooting Tip:

Creating two Wireshark analysis sets can help IT personnel in diagnosing errors. Running an analysis on a problematic machine and also on a functional machine and then comparing the results can offer valuable insight.

Tools Overview

Wireshark

The focus of this manual, “wireshark is the world’s foremost and widely-used network protocol analyzer. It lets you see what’s happening on your network at a microscopic level and is the de facto (and often de jure) standard across many commercial and non-profit enterprises, government agencies, and educational institutions. Wireshark development thrives thanks to the volunteer contributions of networking experts around the globe and is the continuation of a project started by Gerald Combs in 1998” (Wireshark 2021). IT personnel at NSSD will benefit from familiarization with Wireshark and its many useful features in ensuring better network understanding and security.

Nmap

Short for “network mapper”, this utility for network discovery and security auditing provides useful functions such as determining open ports.

Snort

This is an intrusion detection and prevention system that offers real-time traffic analysis and packet logging.

Kali 2

Used for penetration testing to ensure network defenses are adequate.

Windows Firewall

Built in firewall that filters traffic and can be used to block unwanted connections.

Section Eight: References

Berkeley. 2021. Berkeley Information Security Office. Intrusion Detection Guideline. https://security.berkeley.edu/intrusion-detection-guideline

Guim, Timothy. 2021. PCH Technologies. Why Do I Need Network Security Monitoring For My Network? https://pchtechnologies.com/why-do-i-need-network-security-monitoring-for-my-network/

Hassan, Waqar UI. 2017. Mena Entrepreneur. The Importance of Network Traffic Analysis. https://menaentrepreneur.org/2017/02/importance-network-traffic-analysis/

Infosec Matters Blog. 2021. Detecting Network Attacks with Wireshark. https://lab.infoseclearning.com/lab/intrusion-detection-using-snort

Keary, Tim. 2020. Comparitech. Wireshark Cheat Sheet. Commands, Captures, Filters, and Shortcuts. https://www.comparitech.com/net-admin/wireshark-cheat-sheet/

National Institute of Standards and Technology (NIST). 2021. https://www.nist.gov

Profitap. 2018. Profitap Blog. 14 Powerful Wireshark Filters Our Engineers Use. https://insights.profitap.com/14-powerful-wireshark-filters-to-use

Wireshark. 2021. User Reference, Chapter 3. The Analyze Menu.

https://www.wireshark.org/docs/wsug_html_chunked/ChUseAnalyzeMenuSection.html