Security Operations & Incident Management

Lab: Network Intrusion Analysis

1. Setting up the Environment

As part of this workshop, you will get to analyse malware traffic, as generated by an infected host. You will use the Ubuntu 26 image to see how they detect malware traffic

Ubuntu 26:
username: student
password: student

2. Run snort with the provided pcap file to generate the alerts, using the community rule

Go to the folder "Desktop/data/7-nw-forensics-analysis"

Check if the folder has the following files

```
student@student:~/Desktop/data/7-nw-forensic-analysis$ ls
snort.lua  snort3-community.rules  snort_defaults.lua  wiretap.pcap
```

Create log folder name wiretaplogs

mkdir wiretaplogs

```
student@student:~/Desktop/data/7-nw-forensic-analysis$ mkdir wiretaplogs
```

Run snort
snort -c snort.lua -r wiretap.pcap -i ens33 -A alert_full –lua "alert_full={file=true}" -l wiretaplogs

```
snort -c snort.lua -r wiretap.pcap -i ens33 -A alert_full --lua "alert_full={file=true}" -l wiretaplogs
```

Check if an alert file has been created in the folder wiretaplogs.

```
student@student:~/Desktop/data/7-nw-forensic-analysis$ ls wiretaplogs/
alert_full.txt
```

3. Server attack – Honeynet challenge

Today we will analyse a dataset from a Honeynet challenge, Scan #19.

According to the challenge description:

> "On September 16th a Redhat Linux 6.2 honeypot was compromised. This is the 3rd time this system was compromised by the same intruder. The honeynet is VMware based and uses a modified bash to log to syslog. Syslog is remotely logging to 0.0.0.0 (remote syslog server IP has been replaced). The compromised system has an IP of 192.168.1.102. After successfully breaking

into the box, the attacker ended up using 3 modes of connecting and running commands (some of which is encrypted)."

Dataset Files (both files relate to the same incident)	
Image	**Location**
Ubuntu 26	`Desktop/data/7-nw-forensic-analysis/`**`wiretap.pcap`**

Use the datasets above to analyse them with **Snort** and find out what happened (see section 2). Also use **Wireshark** to further investigate attacking traffic.
To run wireshark:

```
sudo wireshark
```

Tips: After you generate the alerts, you might want to enter VMware into 'Full Screen Mode'.
You might want to consider the following stages of Intrusion Analysis. Use the spaces below to write any commands, notes, BPF filters that you use in this process:

1. If your target is 192.168.1.102, which is the IP Address of the main attacker? Which other IP addresses are involved in the attack? How do you know that?
   Hint: Run Snort in Ubuntu 26 to generate alerts.

2. Which vulnerability did the intruder exploit? Which IDS alerts captured the vulnerability exploitation attacks?

3. At which point do you know that the attack became successful?

Hint: Look for 'Attack Response' alerts

4. Are there any other internal IP addresses involved or affected by this incident? How will you find out?

Hint: Use a BPF filter, or use Wireshark Display Filters

5. What happened after the successful vulnerability exploit?

Hint: Use Wireshark 'Follow TCP Stream' of relevant sessions

6. The intruder downloaded rootkits. What were they called? Use Wireshark's "Edit -> Find Packet" menu option to locate the sessions where rootkits are exchanged.

Hint 1: the rootkits are exchanged over Active FTP, where commands are sent over port 21/tcp ftp-data is exchanged over port 20/tcp

Hint 2: Follow TCP Stream to recover the files, and save then in Raw format.

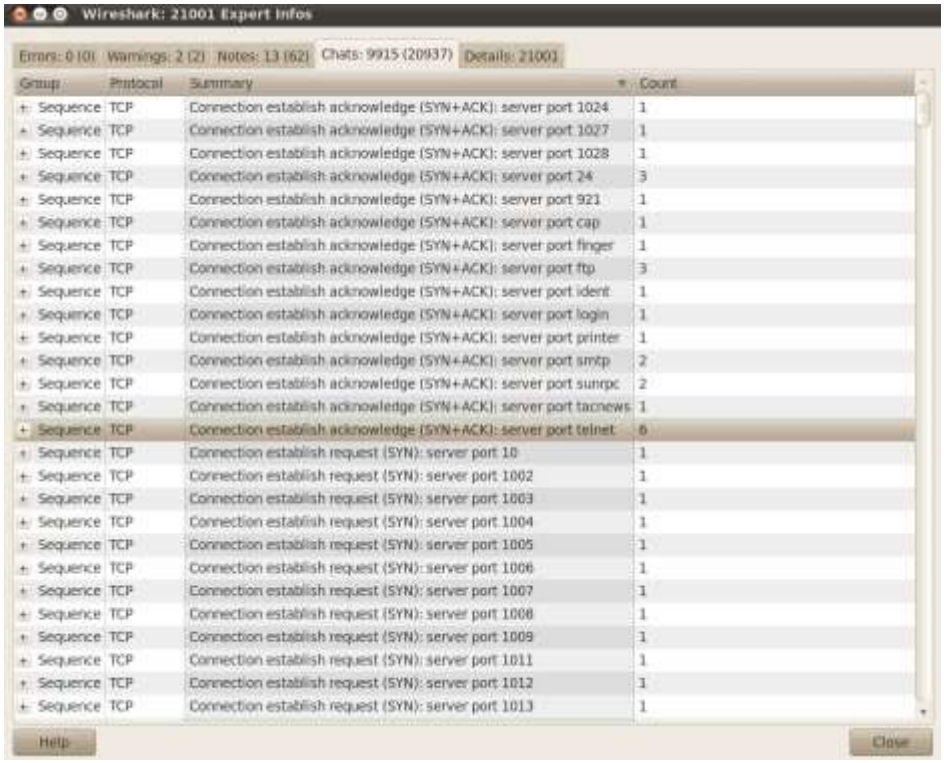
7. BONUS Question: The attacker also establishes a ssh session. Which IP address, server port number and which SSH client does the attacker use?
   Hint 1: Use 'Endpoints' or 'Conversations' from Wireshark to find out.
   Hint 2: Identify port numbers that are "close" to SSH server port 22/tcp

4. Cleaning up the image

Ubuntu 26: Please make sure that you delete all alerts you created, to avoid interfering with other students' work.

5. Hints on Wireshark

Use Expert Info Composite (Analyze -Expert Info Composite) to find some of the successful connections (SYN+ACK) that occurred in this packet stream. When you have completed the traffic reconstruction, select Chats and then reorder thc traffic by clicking on Summary to view all the traffic that shows a successful connection (SYN-ACK). You should verify the validity of these possible connections with a Wireshark filter that will check for SYN-ACK responses and verify that list against the Wireshark list. The list should show 26 SYN-ACK.



Additional Notes

- To recover a rootkit from the traffic logs one must first analyze where the rootkit was transferred to the system. Notice that for an FTP transfer to occur it requires two ports, 21 and 20. TCP Port 21 is the FTP control port. This is where most of the commands are sent from one system to another. The actual file is sent across TCP Port 20. In order to recover a rootkit from the capture, use Wireshark to find where an ftp command that asks for a rootkit is sent. Specifically, you would find the `RETR ftp` command followed by the rootkit name.

- Immediately following the RETR rootkit_name, you will see a new TCP session starting with the three-way hand-shake (SYN, SYN-ACK, ACK). This would be over port 20. In your Wireshark log, it will be listed as protocol FTP-DATA. If you "Follow TCP Stream" on one of these packets you will now see the file in your "Follow TCP Stream" output. Your job is not yet over. You need to switch the display from "ASCII" to "RAW" output. Currently, in ASCII output it will send the stream through a strings filter. You want to display all bytes.

- After switching from ASCII to RAW, you can now select "Save As" and save the dump to disk. The resultant file will be your file. The reason this is successful is due to the reason that a separate FTP-DATA session is set up for every single file being transferred from one system to another. This makes it easy to recover files off of FTP via Wireshark.

FTP Protocol – File Carving

The FTP protocol is somewhat complex and so this section guides you through the carving of the files downloaded via FTP. The first thing to explain is how FTP sends its new port to download a file. When the client request a file to download, the server will send the information to the client Port (hl,h2,h3,b4,p1,p2) -where the server sends the server address and the new port. For example, the server sends Port (192,168,1,102,4,2) indicating that server IP 192.168.1.102 will be using port TCP port 1026 ((4 * 256) +2)) after which the transfer will start.

In order to carve the file out of the data stream, you will need to follow the new port setup by the server. Select the correct port (usually it is after the "Request: RETR filename" information and then Follow TCP Stream to view the Raw file. Make sure you select the correct conversation and save the file using Save As to retrieve. Unlike HTTP carving, there is usually no residual data at the beginning or end of the file. Again, if needed, you may have to use a Hex editor to remove the unwanted data. Additional information on FTP codes and their meaning is provided at the following link:

http://kb.globalscape.com/KnowledgebaseArticle10142.aspx