Carnegie Mellon University

Information Networking Institute - College of Engineering

14823 - Network Forensics

GROUP 8

Members:

Nandan Desai

Taylor McCampbell

Mohammed Rayyan Shaikh

Irtasam Ali Wains

Table of contents

Introduction **3**

INI Infrastructure and Network Diagram **3**

Attack **4**

Evidence **5**

Snort (Alerts Stored and Searched on Logging Server running ELK Stack) 5

Wireshark (Used to analyze Snort generated PCAPs) 6

Network Miner (Used to analyze Snort generated PCAPs) 6

Sysmon/Winlogbeat (Sent to the logging server by Windows 10 Workstations. Viewed on the logging server running ELK stack) 6

Analysis **7**

Timeline (27th November, 2023) **12**

Conclusion **13**

Appendix A: Attack 15

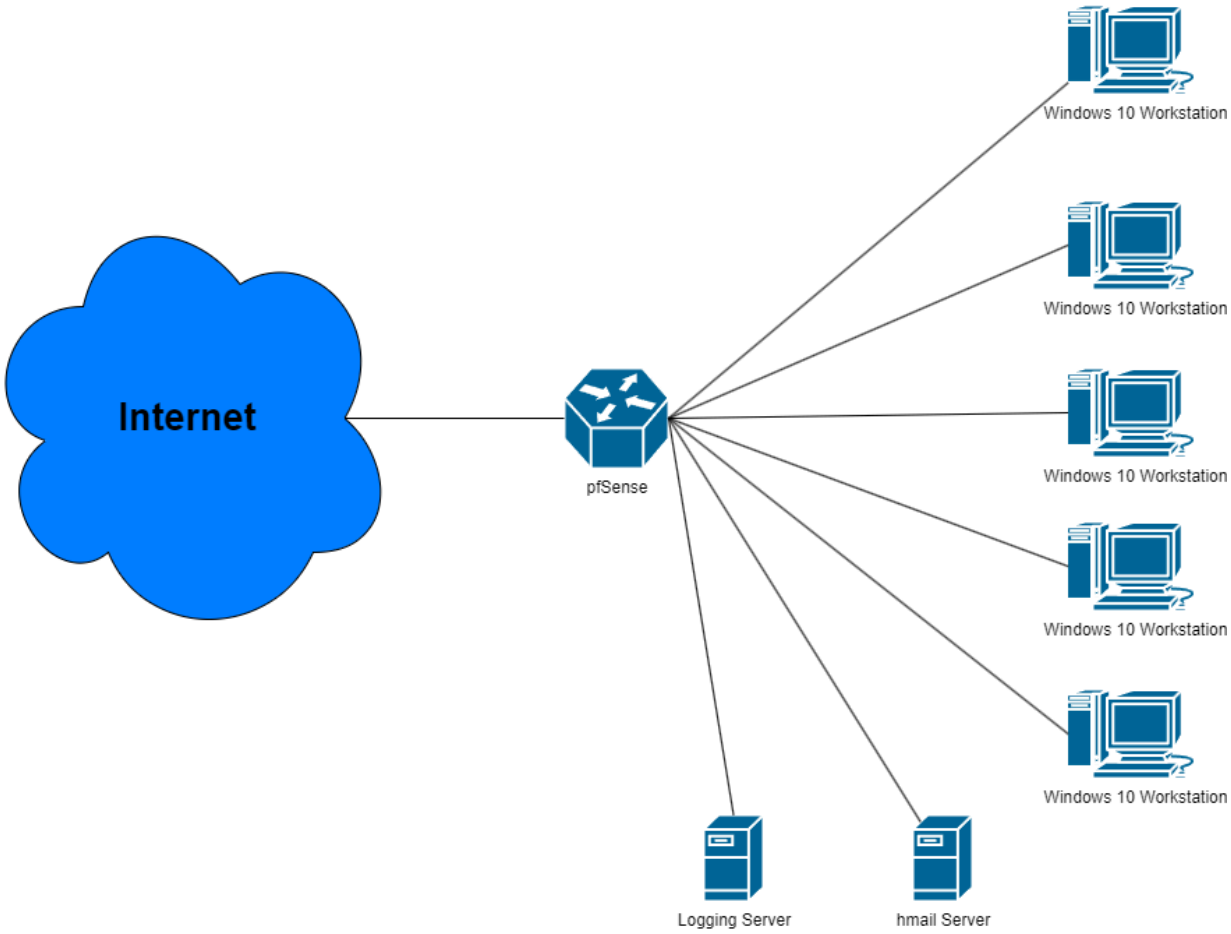
Appendix B: Relevant Evidence 16

Introduction

The Idea Network Institution (INI) is a small business that develops and maintains a web application that allows business leaders to create, share, and discuss ideas. Recently, a disgruntled user of the platform conducted an attack against the company, compromising one of the user workstations. In this project, we will simulate this scenario by configuring the IT infrastructure of the INI, conducting a successful ransomware attack against the infrastructure, and then creating a timeline of events by collecting and analyzing network data generated during the attack.

INI Infrastructure and Network Diagram

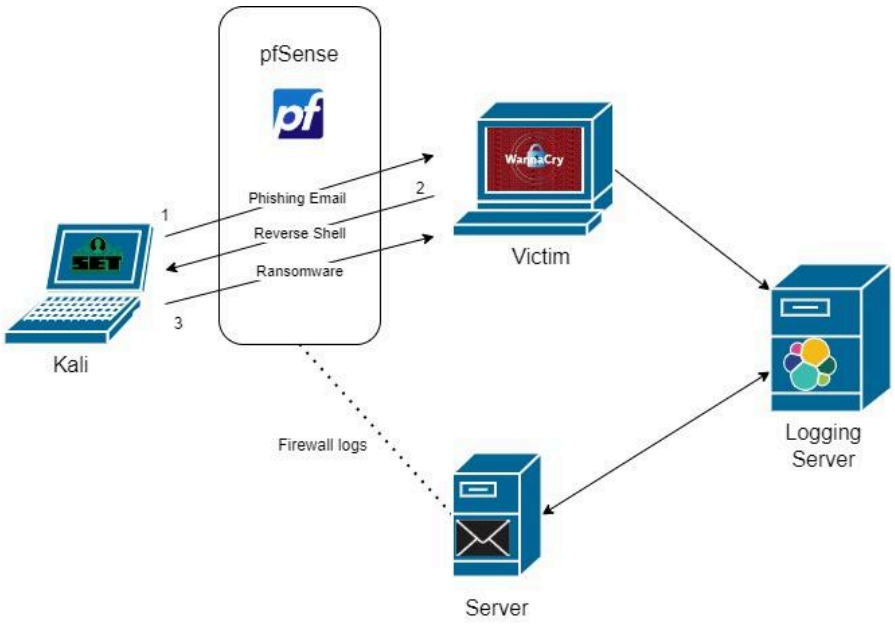
The INI network consists of several Windows 10 Employee Workstations, an Email Server, a Logging Server, and a pfSense deployment. The INI utilizes pfSense as a firewall, default gateway, and DNS server for the internal workstations. Snort is also configured as an Intrusion Detection System through pfSense and forwards alerts to the logging server. Snort also adds packets to a PCAP file stored on the pfSense machine every time it alerts. This PCAP file rolls over once it reaches 128 megabytes and uses the FIFO policy to incorporate the new data. The Email Server is configured using SMTP and runs the open source email server Hmail. The logging server is running the ELK stack and ingests logs through port 5140. The pfSense web interface can be accessed anywhere on the internal subnet by network administrators. Hmail can be configured directly through the server machine. The INI IT department uses wireshark, network miner, and elasticsearch to analyze network traffic and conduct investigations and incident response.



Attack

A classic ransomware scenario was used for this simulation. The social engineering toolkit was used to generate an email from a fake email address that contained a malicious attachment (Figure 2.1). The email was constructed to cause the user to feel a sense of urgency to download and review the attachment stating that there was a business expense report that needed immediate review. The sense of urgency was heightened as the spoofed email address claimed to be the user's boss. The user was fooled into opening the attachment through these tactics. Once the attachment was opened it spawned a meterpreter reverse shell giving the threat actor full control of the victim machine. After control was established, the attacker uploaded the

wannacry.exe ransomware file and ran the executable using the meterpreter shell, rendering the victim machine useless (Figure 2.2). The attacker then tried to move laterally but was unsuccessful.



Evidence

Snort (Alerts Stored and Searched on Logging Server running ELK Stack)

Snort was configured with the Snort Community Ruleset Version 2.9 and forwarded alerts to the logging machine. Snort also created a PCAP and added traffic on alert.

[Fig 2.14]: Snort Alert on IMAP traffic

[Fig 2.15]: Snort Alert of windows executable detected

[Fig 2.16]: Snort Alert of a root shell

Wireshark (Used to analyze Snort generated PCAPs)

[Fig 2.1]: SMTP traffic of Email sent from attacker to Mail Server

[Fig 2.2]: IMAP traffic of Email being forwarded from Mail Server to Victim

[Fig 2.6]: Meterpreter Reverse Shell in action Traffic

Network Miner (Used to analyze Snort generated PCAPs)

[Fig 2.7]: The hosts on the network

[Fig 2.8]: Incoming and outgoing sessions from suspicious machine

[Fig 2.10]: Copy of the malicious attachment

[Fig 2.11]: Email sent from the attacker to victim relayed by mail server

[Fig 2.12]: Protocols involved in the Email Exchange

[Fig 2.13]: Sessions pointing out the reverse tcp shell

Sysmon/Winlogbeat (Sent to the logging server by Windows 10 Workstations. Viewed on the logging server running ELK stack)

[Fig 2.4]: Process creation of report.pdf.exe

[Fig 2.5]: Process creation of wannacry.exe

[Fig 2.17]: wannacry.exe Alert in Detail

[Fig 2.18]: Compromised Machine IP

Hmail Server Logs:

[Fig 2.19]: Shows the email logs for the phishing email

Analysis

The company had three major sources of evidence. First was the alerts from snort, second the alerts from Winlogbeat logs, and lastly the PCAP file generated from the snort alerts. The first two pieces of evidence were viewed using the Elasticsearch-Logstash-Kibana (ELK) Stack running on the logging server. Furthermore, wireshark and Network Miner were used to analyze the PCAP file obtained from the snort alerts.

For the analysis, we started with the Firewall (pfSense) logs to look for any alerts. Since the company had set up Snort with an up-to-date signature database, we were alerted of the ransomware attack [Fig 2.15]. We were then able to piece together the first fragments of our timeline, source IP and the destination IP from this alert. The source IP and port were found to be 10.5.5.100:1111 and the destination IP and port were 192.168.1.105:65500. The alert was about a suspicious Windows executable detected. The time for this alert was 01:00:27 on 27th November, 2023. The IP address 192.168.1.105 was recognized to be associated with a Windows 10 Machine in the company's network. However, the IP address 10.5.5.100 was new and did not belong to any of the machines within the network.

The next step was to look for other snort alerts that could be related to this incident. While digging through the logs, we found alerts for IMAP traffic as seen in Figure 2.14. The IMAP traffic is generated when the mail server forwards the email it got to the recipient. The alert for this traffic was generated at 00:59:35 on 27th November, 2023. The IP addresses involved in this conversation were 192.168.1.105 and 192.168.1.102. The IP 192.168.1.105 was involved in the last alert as well and 192.168.1.102 was found to belong to the Mailing Server. We start to suspect with this alert that a phishing attack has taken place and one of our users was tricked into clicking on a malicious link.

The next step in our investigation was to look at the PCAP file associated with these alerts. The PCAP file was analyzed first using Wireshark and then with network miner as well. While analyzing the PCAP with wireshark, the IMAP filter was used to check what the email was about. As seen in Figure 2.2 the email was intended for aia@netfor.com and the sender was shown to be ini@cmu.edu. Note that the sender ini@cmu.edu is outside of our domain and is not registered with our email server as having ever been received or sent to before this alert. From this, we can note this email as suspicious. Additionally, there was an attachment also seen in Figure 2.2 which was "report.pdf.exe". This attachment can clearly be seen as a misleading report as there is a ".exe" after .pdf which makes it an executable. Following on the lead regarding a phishing email having taken place, the SMTP filter was used in wireshark because SMTP protocol is used for email delivery. Close to the time the victim (192.168.1.105, aia@netfor.com) got the email that there was SMTP traffic coming from 10.5.5.100 to the mailing server (192.168.1.102). On closer inspection, as seen in Figure 2.1, the same sender, receiver, and attachment can be seen. After looking at this piece of information, it can be deduced that 10.5.5.100 is ini@cmu.edu. More details regarding this email were found in the Hmail Server logs which can be seen in Figure 2.19.

Analyzing the PCAP further, we employ Network Miner to easily extract artifacts and visualize data from the conversation between the two hosts. From Figure 2.7 we can see that Network Miner confirms the hosts involved and the IP addresses in the conversation. We can see that the suspicious actor is using a Linux based operating system and the other machines on the network, all of which we recognize except for the suspicious actor with the IP 10.5.5.100. Conducting further analysis using Network Miner as seen in Figure 2.8 we can see one outgoing session and 6 incoming sessions on the suspicious machine. The outgoing session is to the IP

192.168.1.102 which we know from previous analysis is the SMTP traffic. This reinforces that the email was initially sent from the suspicious actor to aia@netfor.com. Additionally, there is an incoming session from 192.168.1.105, the compromised machine, opening a connection to the suspicious actor (10.5.5.100) on port 1111. This is the same port and IP address that was identified in the snort alert that read "Windows Executable Detected" [Figure 2.15]. Further, we can label the suspicious actor as a threat actor as we have multiple sources of evidence that confirm 10.5.5.100 actively compromised the INI's internal network and sent a phishing email to aia@netfor.com.

With Network Miner we can also extract a copy of the malicious attachment the threat actor sent with the email [Figure 2.10]. This attachment is called "report.pdf.exe" and on inspection, is an autogenerated exploit that takes advantage of an apache vulnerability. When running this extracted artifact in a virtualized sandbox we can see that it attempts to create a connection to 10.5.5.100. Upon setting up a fake attacker machine in the sandbox that has the IP address and starting a meterpreter session, this exploit gave the victim a meterpreter shell with full control of the initiating machine. We have now proven that the email attachment was a reverse shell exploit that opened a connection to the threat actor and fully compromised the victim Windows 10 workstation.

Moving even further with Network Miner, it displays the subject and body of the phishing email [Figure 2.11]. The subject of the email was "Please review the business expense report" and the body was "please review the report and get back to me ASAP." We can also see that the display name of the user was "Director." This supports our conclusion that this was a phishing attempt targeted at INI employees. Also, in Figure 2.12, we can see that the malicious actor sent the email using the SMTP protocol on port 25 to the mailing server. On top of this, we

can confirm that the mailing server forwarded the email to 192.168.1.105 on port 143 using the IMAP protocol. This is the traffic corresponding to the IMAP alert discussed earlier in the report. This traffic also allows us to confirm the attacker was in control of 192,168.1.105.

Moving forward, the last thing discovered by Network Miner was the encrypted meterpreter traffic. This can be seen in Figure 2.6 and Figure 2.13. Both Network Miner and Wireshark can easily extract this information. Due to the traffic being encrypted, all that we can verify is source, destination, and its presence. By default, meterpreter sessions are encrypted and it is safe to conclude from our previous analysis that this traffic is associated with the meterpreter session.

Our last source of suspicious activity comes from Sysmon/Winlogbeat logs. Sysmon logs revealed a suspicious process creation event corresponding to the execution of the ransomware on the compromised Windows 10 workstation giving the executable name “wannacry.exe” [Figure 2.5]. Detailed information about the process, including the executable path, command line arguments, parent process, time, etc. was captured [Figure 2.17]. This alert was generated at 2023-11-27 06:42:31.174 UTC time. Note that UTC time is five hours ahead of Eastern Time (GMT -5), which is what the rest of the network is synced on. For the purposes of this report, we will translate these timestamps into Eastern Time (GMT -5) from UTC time when referring to Winlogbeat logs. The INI has remedied this problem by syncing the Windows 10 workstation times with the rest of the network. However, for the purposes of this investigation, it is listed in UTC time and then translated to Eastern Time. Winlogbeat logs confirmed the host computer on which this Sysmon log was generated [Figure 2.18]. This can also be seen in [Fig 2.5]. Note the time difference as mentioned above.

From all of this analysis, we can finally build the timeline of events based on correlating the logs for all of our sources of evidence. The string of events starts by the attacker sending the malicious email to the network. This can be seen as the first event at 00:59. Then, the attacker's email is retrieved by the victim machine at 01:00. The user then downloaded and ran the malicious executable at 01:00:12, after which the attacker had taken control of the victim machine. After some meterpreter shell traffic where the attacker uploaded the ransomware (01:00:27), the ransomware wannacry.exe was executed on the victim machine at 01:01:28. This timeline is summarized with supporting evidence for each event listed in the “Source” section. We can see that the logging and alert infrastructure at the INI were able to investigate and accurately tell a story with multiple confirming pieces of evidence. Please see the timeline table on the next page for a more easily visualized version.

Timeline (27th November, 2023)

Event	**Time (Hour:Minutes:Seconds)**	**Source**
Attacker's Malicious Email Reaches Mail Server (SMTP)	00:59	Wireshark [Fig 2.1]
Attacker's Email is Retrieved by Victim Machine (IMAP4)	01:00	Snort Alert [Fig 2.14] + Generated PCAP Analyzed Using Wireshark [Fig 2.2]
User Downloads and Executes Reverse Shell Payload "report.pdf.exe"	01:00:12	Snort Alert [Fig 2.16] + Generated PCAP Analyzed Using Wireshark [Fig 2.2] Winlogbeat logs (Process creation) [Fig 2.4]
Attacker Uploads wannacry.exe to Victim Machine	01:00:27	Snort Alert [Fig 2.15]
Attacker Executes wannacry.exe	01:01:28	Winlogbeat logs (Process creation) [Fig 2.5]

Conclusion

The project involved 3 stages. The first was setting up the network , the second phase was performing the attack and lastly utilizing forensics tools to trace back the attack. After going through all three phases, we have identified areas of improvement:

The logging infrastructure can be improved in several ways.

1. Sysmon logs: The default Sysmon configuration includes the capturing of Process Creation and Process Termination events, among others, but doesn't include Network Connection and File Creation events for each of the processes. Enabling this configuration would help us gather more accurate evidence.
2. Auto-Updating Snort Signatures: In the current firewall configuration, Snort alert signatures are manually added and updated. Enabling auto-update would help us obtain the latest signatures and protect the company against new threats.
3. Enabling 'Security' in ELK configuration: For the scope of this project, we are using the default configuration for the ELK stack. In this default setup, 'Security' features are disabled. This should never be done in a production environment, and it is especially relevant to forensics. Disabling 'Security' would compromise authentication to the Elasticsearch database, potentially jeopardizing the integrity of our logs—something critical from a forensics perspective.
4. Using an NTP server for time synchronization instead of relying on manually configuring system times.

Attack:

5. More Sophisticated Attack: For this project a classic ransomware situation was demonstrated where one user machine is compromised. In reality, this specific ransomware attack could be easily mitigated by a modern antivirus or firewall

Overall, through moving through the phases of the project, we have learned a considerable amount about the implementation of a robust logging infrastructure. The common tools used, how to configure pfsense, windows, snort, the ELK stack, how to synchronize system times, etc. On top of this, we learned how to successfully use meterpreter and the Social Engineering Toolkit to perform penetration testing. Lastly, we reinforced the skills learned throughout the semester to conduct the analysis on all of the logs our infrastructure collected.

Appendix A: Attack

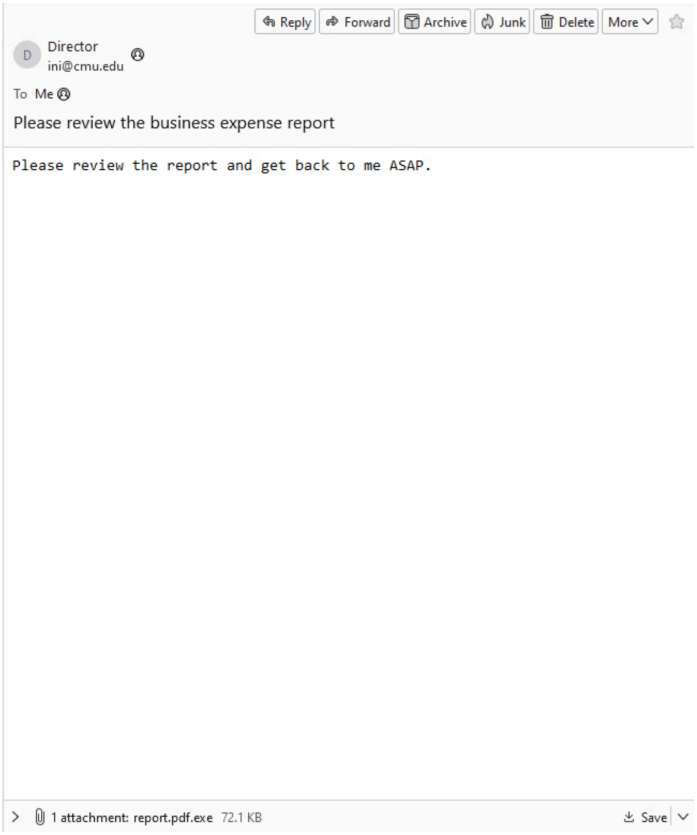


Figure 1.1

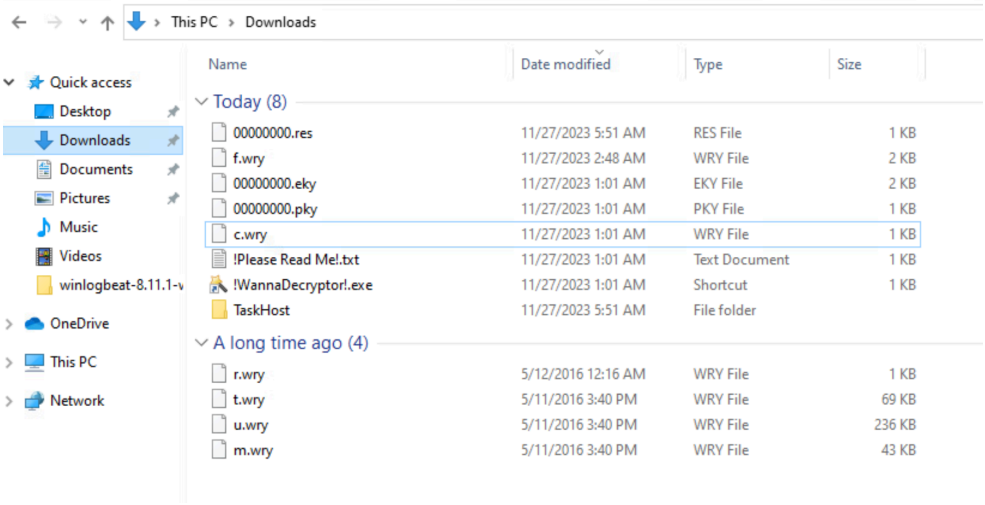


Figure 1.2

Appendix B: Relevant Evidence

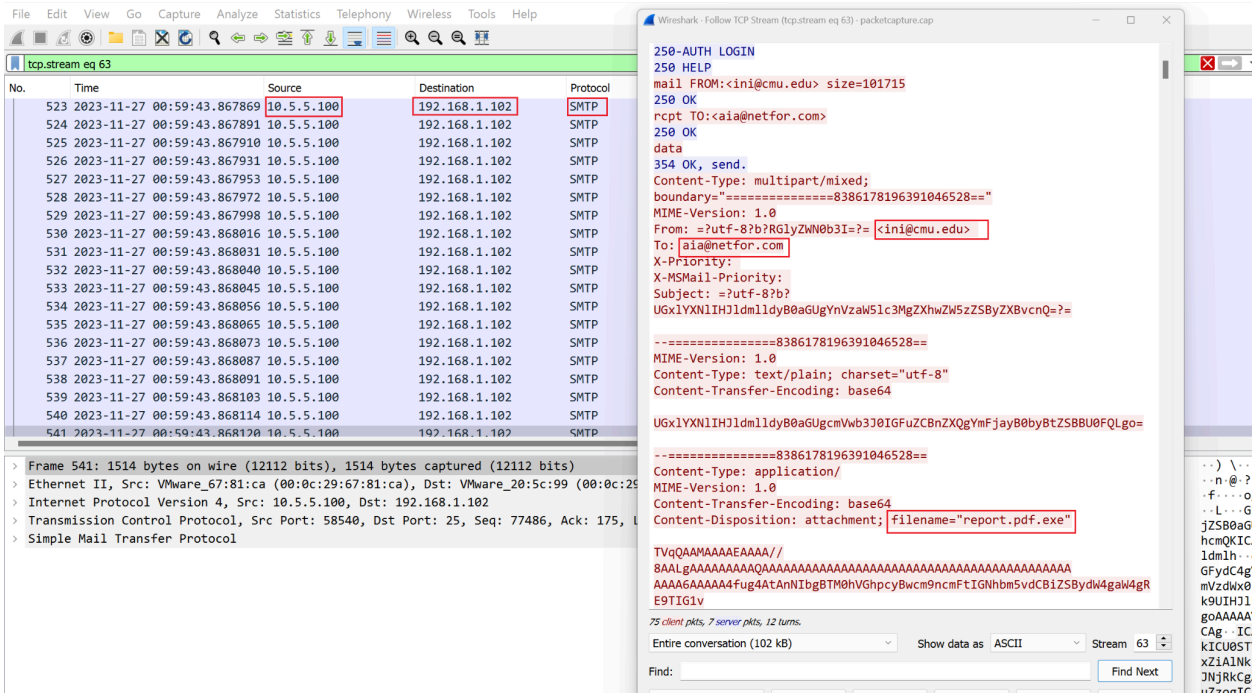


Figure 2.1: Snort PCAP of SMTP Traffic

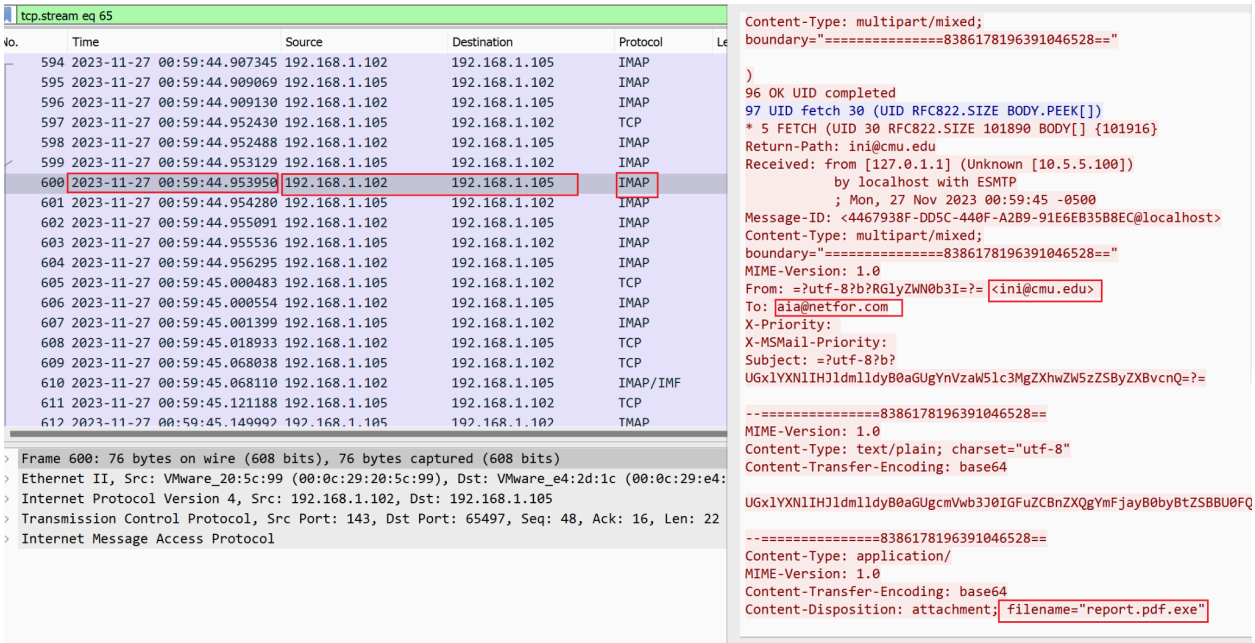


Figure 2.2: Snort PCAP of IMAP4 Traffic

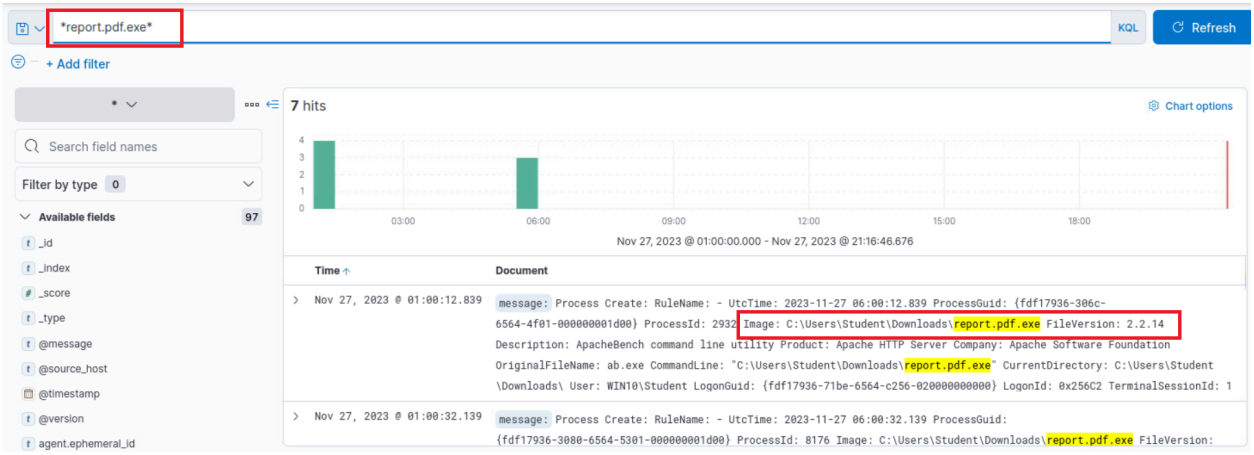


Figure 2.4: WinLogBeat Entry for Reverse Shell Process Creation

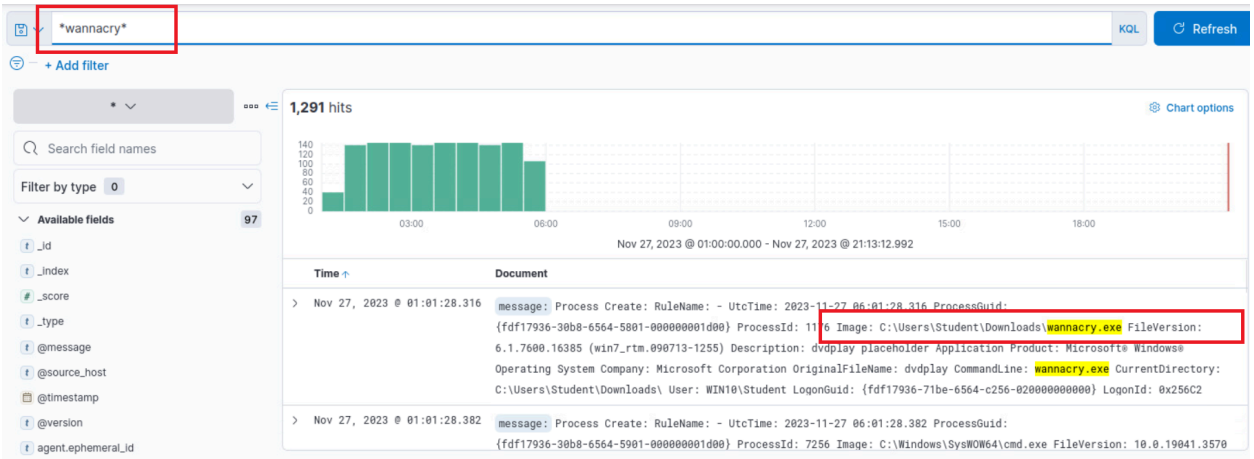


Figure 2.5: WinLogBeat Entry for Ransomware Process Creation

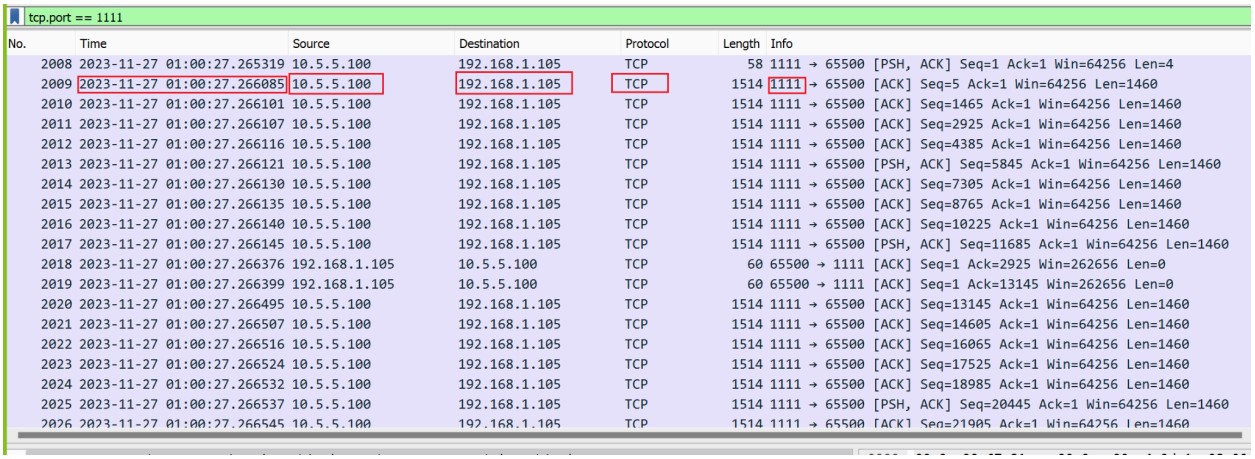


Figure 2.6: PCAP on Meterpreter Traffic

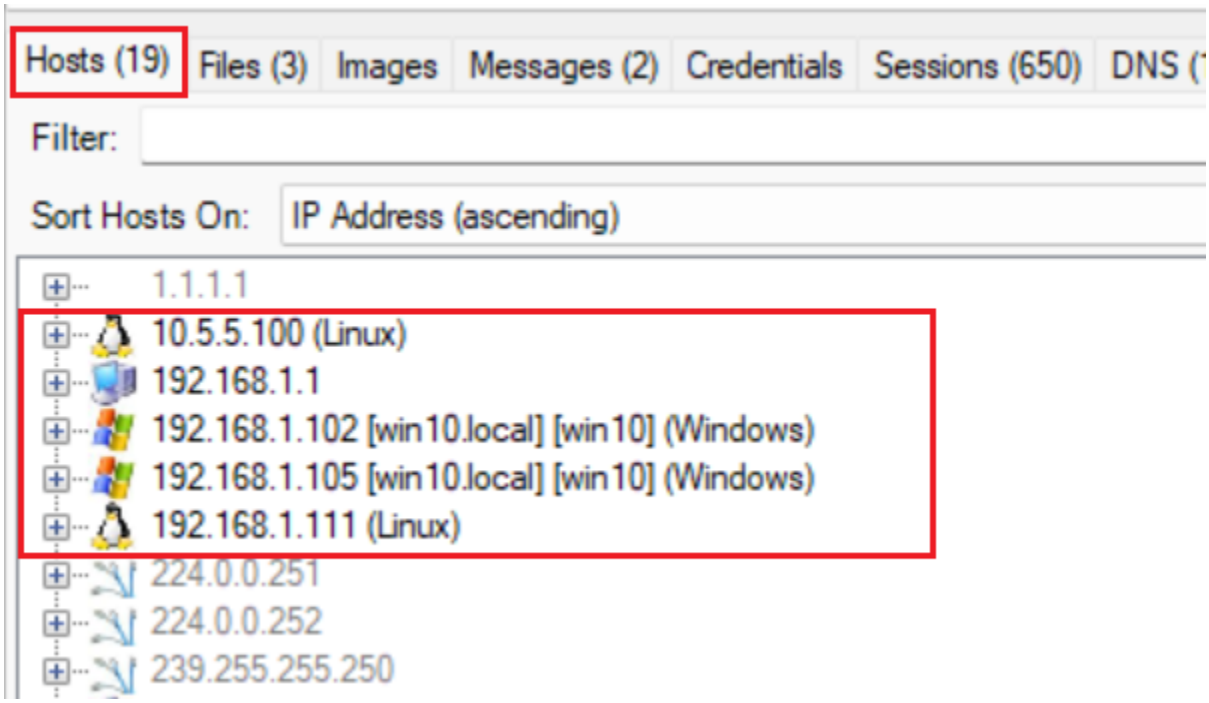


Figure 2.7: Network Miner Active Hosts

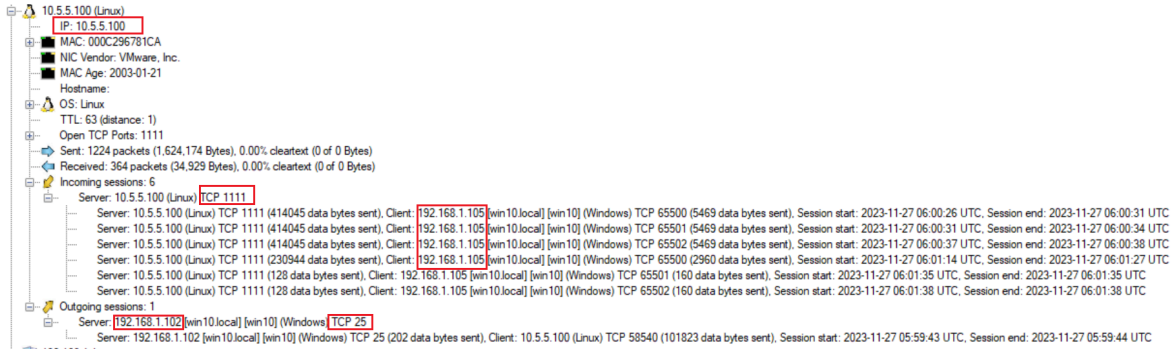


Figure 2.8: Network Miner Attacker IP Address and Port

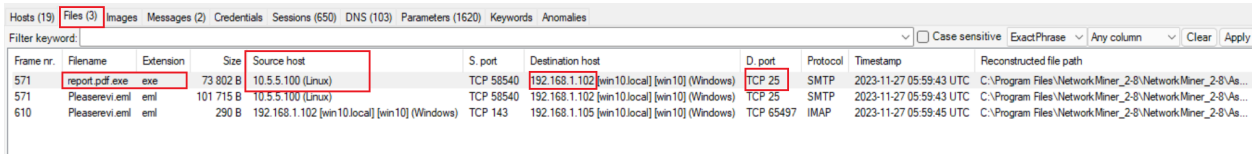


Figure 2.10: Network Miner Malicious Attachment

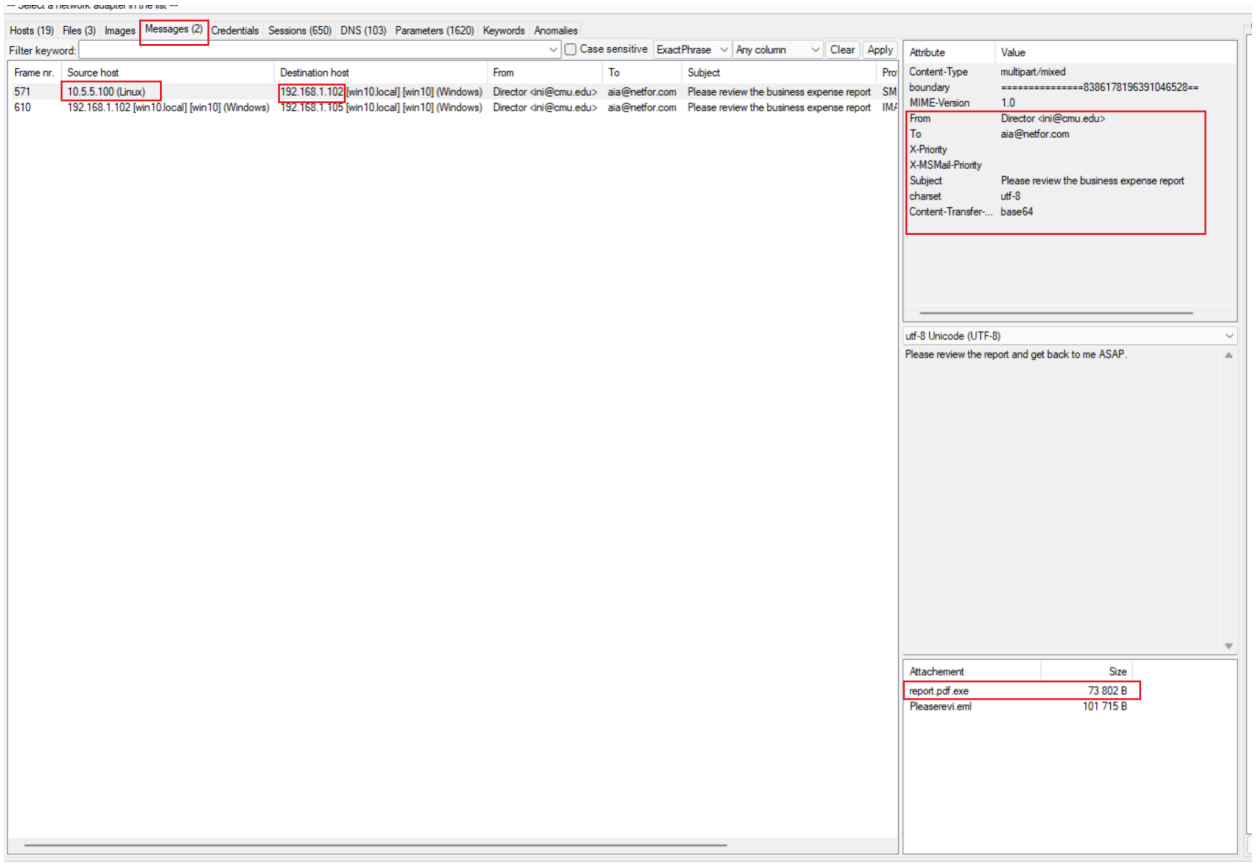


Figure 2.11: Network Miner Email Traffic

Frame nr.	Client host	C. port	Server host	S. port	Protocol (application layer)	Start time
466	10.5.5.100 (Linux)	58540	192.168.1.102 [win10.local] [win10] (Windows)	25	SMTP	2023-11-27 05:59:43 UTC
594	192.168.1.105 [win10.local] [win10] (Windows)	65497	192.168.1.102 [win10.local] [win10] (Windows)	143	IMAP	2023-11-27 05:59:44 UTC

Figure 2.12: Network Miner Email Server Protocols

Hosts (19) Files (3) Images Messages (2) Credentials Sessions (650) DNS (103) Parameters (1620) Keywords Anomalies

Filter keyword:

Frame nr.	Client host	C. port	Server host	S. port	Protocol (application layer)	Start time
5672	192.168.1.111 (Linux)	39814	1.1.1.1	53		2023-11-27 06:02:02 UTC
5670	192.168.1.111 (Linux)	39810	1.1.1.1	53		2023-11-27 06:02:02 UTC
5668	192.168.1.111 (Linux)	39800	1.1.1.1	53		2023-11-27 06:02:02 UTC
5660	192.168.1.111 (Linux)	39790	1.1.1.1	53		2023-11-27 06:02:02 UTC
5658	192.168.1.111 (Linux)	39780	1.1.1.1	53		2023-11-27 06:02:02 UTC
5656	192.168.1.111 (Linux)	39778	1.1.1.1	53		2023-11-27 06:02:02 UTC
5648	192.168.1.111 (Linux)	39774	1.1.1.1	53		2023-11-27 06:02:02 UTC
5646	192.168.1.111 (Linux)	39758	1.1.1.1	53		2023-11-27 06:02:02 UTC
5644	192.168.1.111 (Linux)	39752	1.1.1.1	53		2023-11-27 06:02:02 UTC
5541	192.168.1.105 [win10.local] [win10] (Windows)	65497	192.168.1.102 [win10.local] [win10] (Windows)	143		2023-11-27 06:01:43 UTC
5520	192.168.1.105 [win10.local] [win10] (Windows)	65502	10.5.5.100 (Linux)	1111		2023-11-27 06:01:38 UTC
5506	192.168.1.105 [win10.local] [win10] (Windows)	65501	10.5.5.100 (Linux)	1111		2023-11-27 06:01:35 UTC

Figure 2.13: Network Miner Meterpreter Shell Traffic

Figure 2.14: Snort IMAP4 Alert

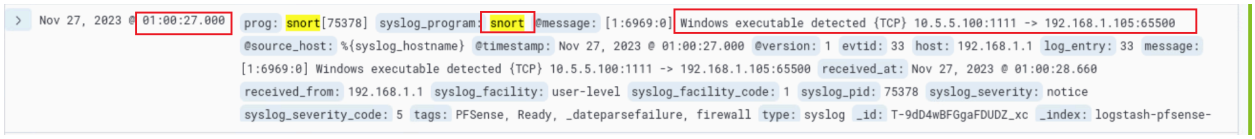


Figure 2.15: Snort Windows Executable Alert

Figure 2.16 Snort Root Shell Alert

```
Unset
Process Create:
UtcTime: 2023-11-27 06:42:31.174
ProcessGuid: {fdf17936-3a57-6564-6202-000000001d00}
ProcessId: 660
Image: C:\Users\Student\Downloads\!WannaDecryptor!.exe
OriginalFileName: SyncHost.EXE
CommandLine: !WannaDecryptor!.exe c
CurrentDirectory: C:\Users\Student\Downloads\
User: WIN10\Student
LogonGuid: {fdf17936-71be-6564-c256-020000000000}
LogonId: 0x256C2
Hashes: SHA256=78E3F87F31688355C0F398317B2D87D803BD87EE3656C5A7C80F0561EC8606DF
ParentProcessGuid: {fdf17936-30b8-6564-5801-000000001d00}
ParentProcessId: 1176
ParentImage: C:\Users\Student\Downloads\wannacry.exe
ParentCommandLine: wannacry.exe
ParentUser: WIN10\Student
```

Figure 2.17 Winlogbeat log showing process related to the deployed ransomware

```
Unset
"winlog.computer_name": [
  "win10"
],
"winlog.event_data.ParentUser.keyword": [
  "WIN10\\Student"
],
"host.ip.keyword": [
  "fe80::106e:890b:d66e:59ce",
  "192.168.1.105"
]
```

Figure 2.18 WinLogBeat IP Address Confirmation

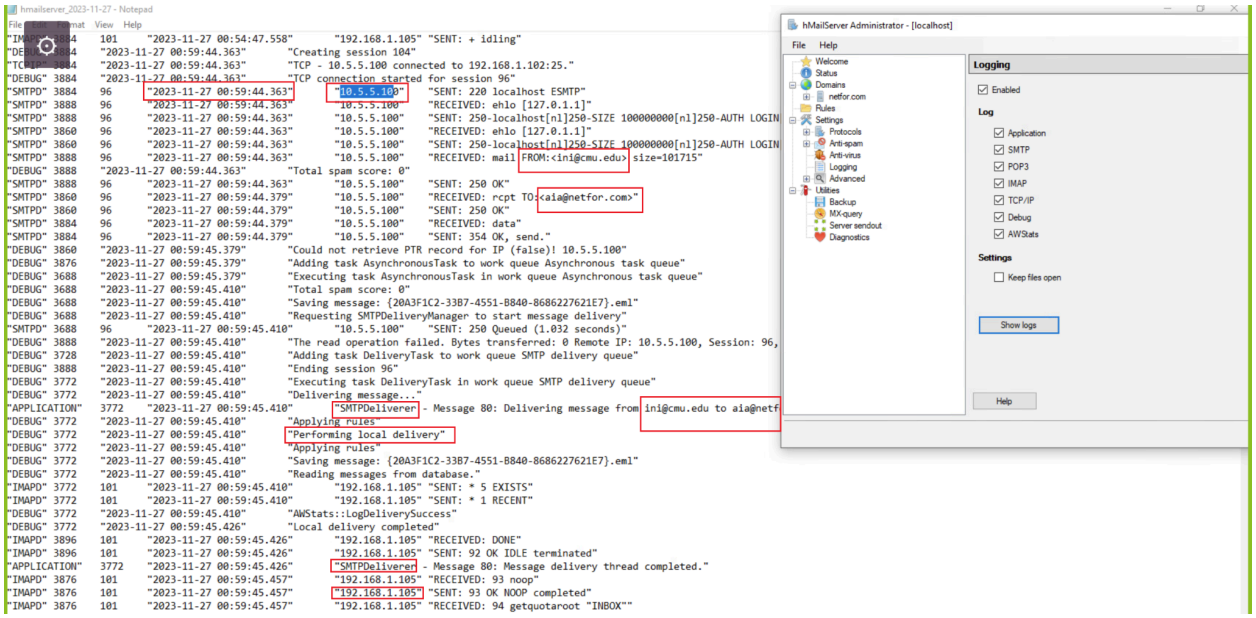


Figure 2.19 Hmail Server Logs