Case Name: Communication Artifacts

Couse Name: IST402

Instructor: Robert Price

Date: 04/17/2025

Examiner Name: Jaspreet Singh

Table of Contents

List of Illustrative Materials

Helix

Wireshark

Internet Relay Chat (IRC) Communications

EXECUTIVE SUMMARY

Background

Throughout this lab the examiner is focusing on the forensic analysis of user communication artifacts, with an emphasis on Internet Relay Chat (IRC) and email. These two forms of digital communication could contain useful evidence or data in an investigation whether it is criminal or civil. Through this lab the examiner will analyze the local email data, review the network-based email traffic, and examine the IRC communications.

To begin, the examiner accesses the Microsoft Outlook data files (PST) and used Helix to recover passwords and view the email messages the users have sent and drafted. Then the examiner will capture and analyze network traffic with Wireshark and NetworkMiner while focusing on unencrypted POP3 and SMTP protocols to help get information from the email messages and retrieve user credentials. Finally, the examiner will simulate IRC communication using mIRC and capture the traffic with tcpdump, then the examiner will use Wireshark to examine the message streams.

The examiner will work with both host-based artifacts and network captures to help develop a deeper understanding of how communication data can be identified, extracted, and examined during a forensic investigation.

Evidence

Description	File Name/Path	Source	Date/Time Collected	Examiner
Evidence	Flag.pcap – pop – flag3	Windows Server	04/18/2025 at 9:35 PM	Jaspreet Singh

COLLECTION AND ANALYSIS

Collection

The examiner will begin by accessing the virtualized Windows 7 environment and then launching the Microsoft Outlook application to locate the user's communication artifacts that are stored in the Personal Storage Table (PST) files. Since the data on Microsoft Outlook was password protected, the examiner used the forensic tool called Helix to help extract the password. This would then allow the examiner access to the user's email content, such as messages stored in both the sent and draft folders. These emails would then be examined by the examiner and documented as part of communication analysis.

Next, the examiner will proceed to examine these email messages transmitted over the network by using Wireshark. Which is a network protocol analyzer, the examiner will open a pre-captured .pcap file and apply filters to isolate POP3 and SMTP traffic. By following the TCP streams the examiner will be able to collect unencrypted email contents and login credentials in plaintext, making this critical in an investigation.

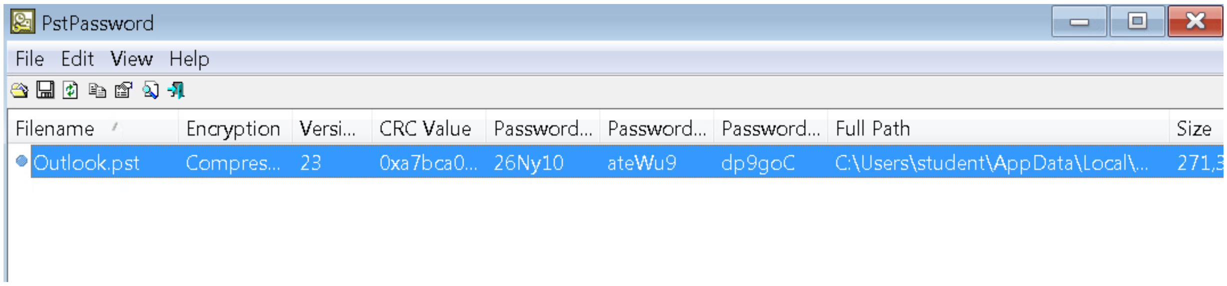
Finally, the examiner will begin the analysis of IRC communications. Which is a Lunix-based sniffer VM that is used to initiate tcpdumps. After establishing a connection and exchanging messages with an IRC channel the traffic was saved as a capture file. Then the examiner will open this saved file in Wireshark on the sniffer VM, and then the examiner will collect the IRC message content and data for analysis.

Overall, the examiner will capture artifacts pertaining to Outlook email messages, network traffic containing POP3 and SMTP communications, and IRC logs.

Analysis

Using Helix.exe to reveal Outlook Password

The examiner was able to use the PstPassword tool from Helix.exe to recover the password for the encrypted Outlook.pst file. In the screenshot it shows that the tool revealed multiple possible passwords including "26Ny10" which was later used by the examiner to successfully access the Outlook email allowing the examiner to examine the sent and draft folders.

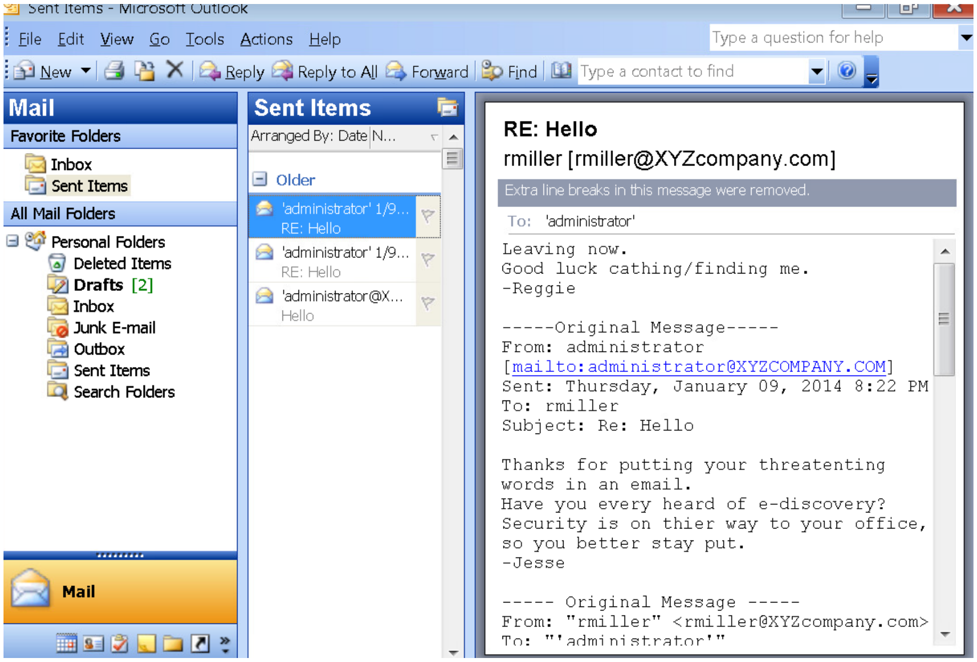


Filename	Encryption	Versi...	CRC Value	Password...	Password...	Password...	Full Path	Size
Outlook.pst	Compres...	23	0xa7bca0...	26Ny10	ateWu9	dp9goC	C:\Users\student\AppData\Local\...	271,3

Figure 1: PstPassord reveal

Reviewing Sent Items

The examiner after being able to access the Outlook application, was able to navigate to the Sent items folder to find three items all in which reveal a message thread that show an intense exchange between rmiller, administrator, and Jesse. With the final message being sent by rmiller stating, "“Leaving now. Good luck catching/finding me. -Reggie,” which suggests the intent to evade detection. This kind of email communication serves as key evidence in an investigation since it shows the individual behavior and intent within a forensic context.



'administrator' 1/9...
RE: Hello

'administrator' 1/9...
RE: Hello

'administrator@X...
Hello

RE: Hello

rmiller [rmiller@XYZcompany.com]

Extra line breaks in this message were removed.

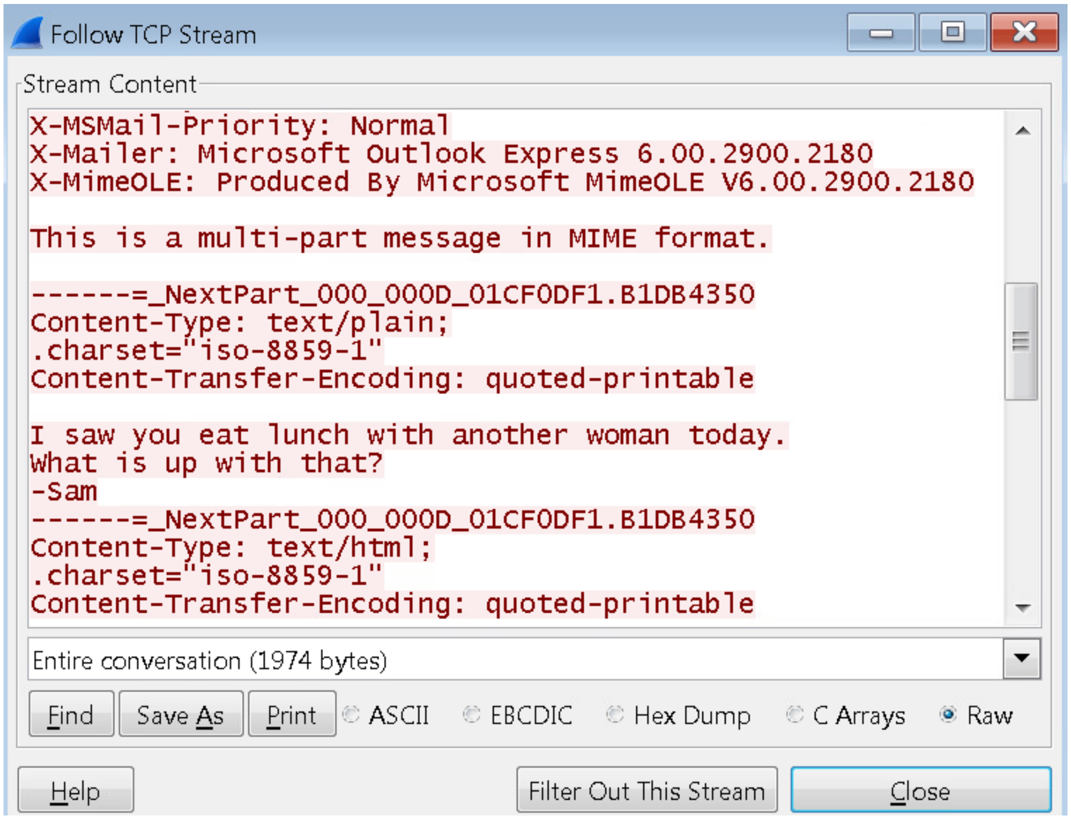
To: 'administrator'

```
Leaving now.
Good luck cathing/finding me.
-Reggie

-----Original Message-----
From: administrator
[mailto:administrator@XYZCOMPANY.COM]
Sent: Thursday, January 09, 2014 8:22 PM
To: rmiller
Subject: Re: Hello

Thanks for putting your threatenting
words in an email.
Have you every heard of e-discovery?
Security is on thier way to your office,
so you better stay put.
-Jesse

----- Original Message -----
From: "rmiller" <rmiller@XYZcompany.com>
To: "'administrator'"
```

Figure 2: Sent emails between Reggie and Administrator

Wireshark Outlook Message

The examiner used Wireshark’s Follow TCP Stream to view an unencrypted email sent via Outlook Express. The message content was fully visible in plain text, showing a personal email from “Sam.” This demonstrates how unsecured SMTP traffic can expose sensitive communications, making it easily accessible through packet analysis.



```
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook Express 6.00.2900.2180
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2900.2180

This is a multi-part message in MIME format.

------=_NextPart_000_000D_01CF0DF1.B1DB4350
Content-Type: text/plain;
.charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

I saw you eat lunch with another woman today.
What is up with that?
-Sam
------=_NextPart_000_000D_01CF0DF1.B1DB4350
Content-Type: text/html;
.charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable
```

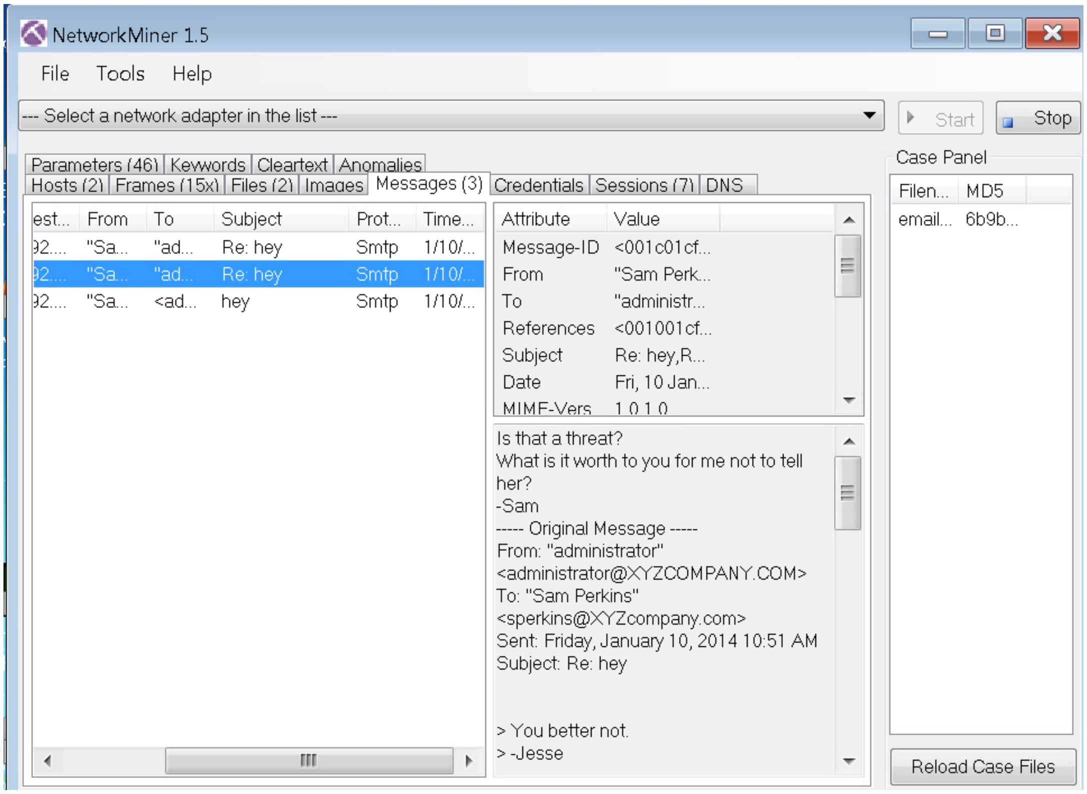
Entire conversation (1974 bytes)

○ ASCII ○ EBCDIC ○ Hex Dump ○ C Arrays ● Raw

Figure 3: Using Wireshark Follow TCP Stream

NetworkMiner Analysis

The examiner used NetworkMiner to analyze the same email capture file. Under the Messages tab, the email content was automatically parsed and displayed. The selected message shows a heated exchange between “Sam Perkins” and “administrator@XYZCOMPANY.COM,” revealing potentially threatening language. This reinforces how NetworkMiner simplifies the extraction of key communication artifacts, making it easier to review and document evidence compared to raw packet analysis.



est...	From	To	Subject	Prot...	Time...
92....	"Sa...	"ad...	Re: hey	Smtp	1/10/...
92....	"Sa...	"ad...	Re: hey	Smtp	1/10/...
92....	"Sa...	<ad...	hey	Smtp	1/10/...

Attribute	Value
Message-ID	<001c01cf...
From	"Sam Perk...
To	"administr...
References	<001001cf...
Subject	Re: hey,R...
Date	Fri, 10 Jan...
MIME-Vers	1.0.1.0

Is that a threat?
What is it worth to you for me not to tell her?
-Sam
----- Original Message -----
From: "administrator" <administrator@XYZCOMPANY.COM>
To: "Sam Perkins" <sperkins@XYZcompany.com>
Sent: Friday, January 10, 2014 10:51 AM
Subject: Re: hey

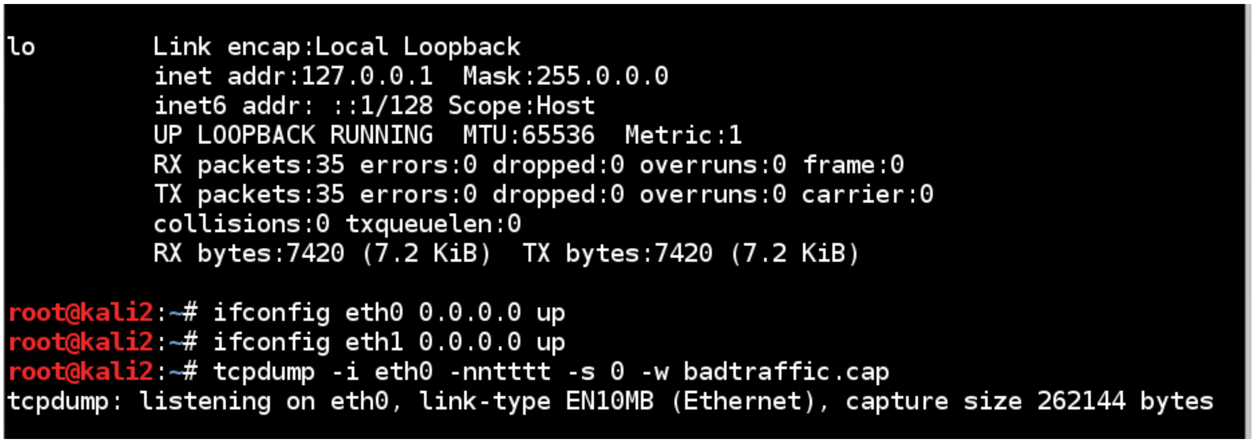
> You better not.
> -Jesse

Filen...	MD5
email...	6b9b...

Figure 4: Using NetworkMiner to analyze messages

Linux sniffer VM

The examiner uses ifconfig command on the Linux sniffer VM to bring up both network interfaces (eth0 and eth1). Then, tcpdump was executed on interface eth0 to capture all network traffic, saving it to a file named badtraffic.cap. This setup ensured that any IRC communication between hosts would be recorded for later forensic analysis. Tcpdump ran in the background, continuously logging packet data as the IRC session was established.

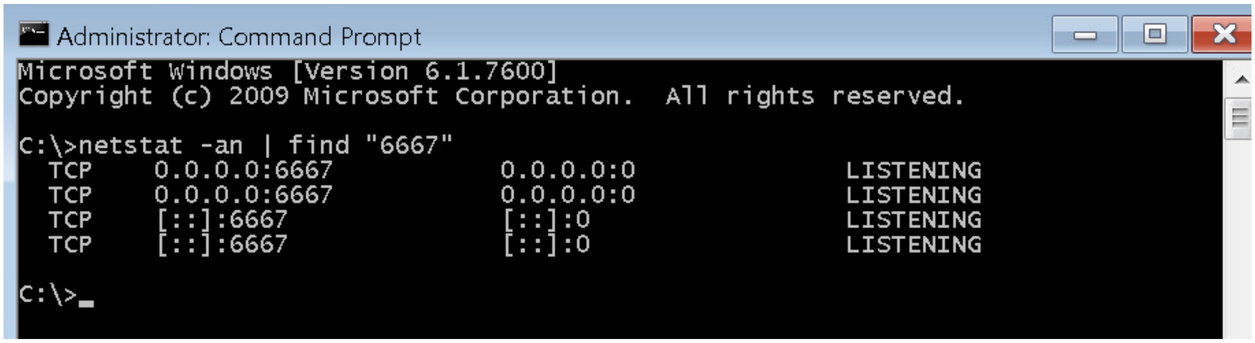


```
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:35 errors:0 dropped:0 overruns:0 frame:0
          TX packets:35 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:7420 (7.2 KiB)  TX bytes:7420 (7.2 KiB)

root@kali2:~# ifconfig eth0 0.0.0.0 up
root@kali2:~# ifconfig eth1 0.0.0.0 up
root@kali2:~# tcpdump -i eth0 -nntttt -s 0 -w badtraffic.cap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
```

Figure 5: Using Linux to start tcpdump

Command Prompt IRC Setup

The examiner ran the command netstat -an | find "6667" on the command prompt within the Windows 7 VM to verify that the IRC server set up is listening on port 667 which it was based on the LISTENING statements in the command prompt. The output confirmed that the system was ready to accept IRC connections, ensuring that any chat traffic during the session would be captured in the tcpdump log from the sniffer machine.

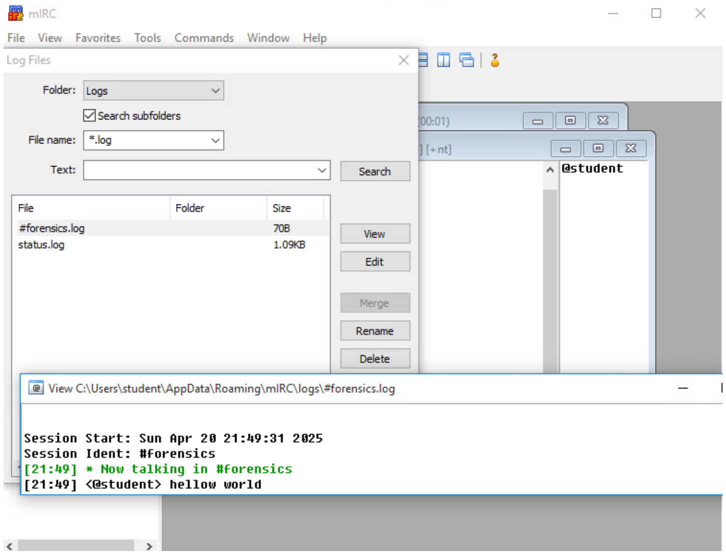


```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\>netstat -an | find "6667"
  TCP    0.0.0.0:6667           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:6667           0.0.0.0:0              LISTENING
  TCP    [::]:6667              [::]:0                 LISTENING
  TCP    [::]:6667              [::]:0                 LISTENING

C:\>_
```

Figure 6: Checking to see if IRC is listening on port 6667

mIRC Client

On the Windows 10 machine, the examiner used the mIRC client to join the #forensics IRC channel. The user student sent a test message: *"hellow world"*. The message was automatically logged and saved to a file named #forensics.log, located in the default mIRC logs directory. This log provides a local record of the IRC session and complements the packet capture collected by the sniffer for cross-verification during forensic analysis.



Folder: Logs

☑ Search subfolders

File name: *.log

Text:

File	Folder	Size
#forensics.log		70B
status.log		1.09KB

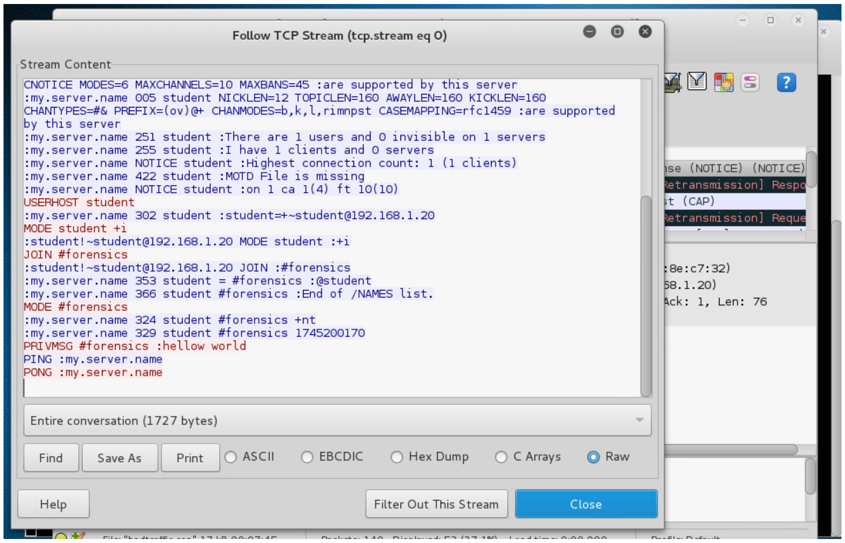
View C:\Users\student\AppData\Roaming\mIRC\logs\#forensics.log

```
Session Start: Sun Apr 20 21:49:31 2025
Session Ident: #forensics
[21:49] * Now talking in #forensics
[21:49] <@student> hellow world
```

Figure 7: Joining the #forensics IRC channel

Wireshark Follow TCP Stream

After capturing the IRC session using tcpdump, the examiner opened the badtraffic.cap file in Wireshark and followed the IRC TCP stream. The stream shows a full transcript of the IRC connection, including server responses, channel join events, and the test message *"hellow world"* sent by the user student. This confirms that the IRC traffic was successfully captured in transit and demonstrates how unencrypted protocols allow full reconstruction of conversations through packet analysis.



```
CNOTICE MODES=6 MAXCHANNELS=10 MAXBANS=45 :are supported by this server
:my.server.name 005 student NICKLEN=12 TOPICLEN=160 AWAYLEN=160 KICKLEN=160
CHANTYPES=#& PREFIX=(ov)@+ CHANMODES=b,k,l,rimnpst CASEMAPPING=rfc1459 :are supported
by this server
:my.server.name 251 student :There are 1 users and 0 invisible on 1 servers
:my.server.name 255 student :I have 1 clients and 0 servers
:my.server.name NOTICE student :Highest connection count: 1 (1 clients)
:my.server.name 422 student :MOTD File is missing
:my.server.name NOTICE student :on 1 ca 1(4) ft 10(10)
USERHOST student
:my.server.name 302 student :student=+~student@192.168.1.20
MODE student +i
:student!~student@192.168.1.20 MODE student :+i
JOIN #forensics
:student!~student@192.168.1.20 JOIN :#forensics
:my.server.name 353 student = #forensics :@student
:my.server.name 366 student #forensics :End of /NAMES list.
MODE #forensics
:my.server.name 324 student #forensics +nt
:my.server.name 329 student #forensics 1745200170
PRIVMSG #forensics :hellow world
PING :my.server.name
PONG :my.server.name
```

Entire conversation (1727 bytes)

Figure 8: Finding hellow world in Wireshark after stopping tcpdump

Conclusion

Overall, throughout this lab, the examiner was able to gain hands-on experience using multiple forensic tools to capture, analyze, and interpret communication artifacts across host and network environments. The examiner was able to identify users through email and IRC traffic, while also emphasizing the importance of monitoring both local and transmitted communication data during an investigation.

Helix

The examiner was able to use the forensic tool Helix to recover the password protecting the Outlook .pst file. This tool allowed the examiner to access local email artifacts, including messages stored within the Sent Items and Drafts folders. Helix's PstPassword utility allowed the examiner to extract multiple potential passwords one of which was "26Ny10" that was able to successfully unlock the email archive. This tool was critical for accessing stored communication data when credentials were not initially known to the examiner.

Wireshark

The examiner was able to use the forensic tool Wireshark to examine POP3 and SMTP traffic from a pre-captured .pcap file called email.pcap. By applying filters and using Follow TCP Stream features, the examiner was able to extract email content and review them in plain text, demonstrating the risks of transmitting data over an unencrypted channel. This process helped highlight the value of packet analysis in identifying sensitive information within network traffic.

Internet Relay Chat (IRC) Communications

Finally, the examiner was able to simulate an Internet Relay Chat (IRC) session throughout this investigation. The examiner captured IRC traffic using tcpdump and analyzed it in Wireshark. The captured stream showed connection details, user commands, and the message *"hellow world"* sent by the user. In addition, local mIRC logs were reviewed to confirm the chat session. This portion of the lab demonstrated how easily IRC traffic, when unencrypted, can be monitored and reconstructed for forensic review.