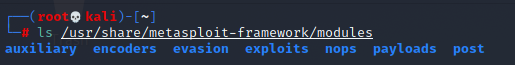
Aryaman Mishra

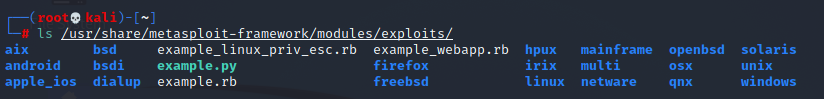
19BCE1027

1. Browse through various folders of Metasploit and explore the folders like payload, exploits and write a paragraph about every folder and one script in every folder

Almost all of your interaction with Metasploit will be through its many modules, which it looks for in two locations. The first is the primary module store under **/usr/share/metasploit-framework/modules/** and the second, which is where you will store custom modules, is under your home directory at **~/.msf4/modules/**.



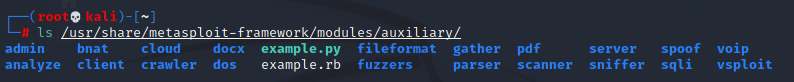
All Metasploit modules are organized into separate directories, according to their purpose. A basic overview of the various types of Metasploit modules is shown below.



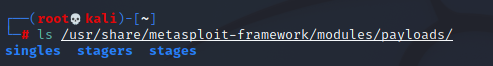
Example.py

Resource scripts provide an easy way for you to automate repetitive tasks in Metasploit. Conceptually, they're just like batch scripts. They contain a set of commands that are automatically and sequentially executed when you load the script in Metasploit. You can create a resource script by chaining together a series of Metasploit console commands and by directly embedding Ruby to do things like call APIs, interact with objects in the database, and iterate actions.

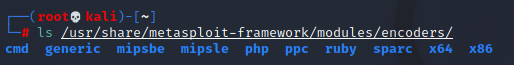
In the Metasploit Framework, exploit modules are defined as modules that use payloads.



Auxiliary modules include port scanners, fuzzers, sniffers, and more.



Payloads consist of code that runs remotely, while encoders ensure that payloads make it to their destination intact. Nops keep the payload sizes consistent across exploit attempts.

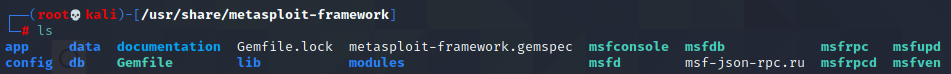


(ii)Run Information Gathering for the protocols like SMTP, secure shell, HTTP. For every protocol minimum of three scanner commands should be run.

Access Framwork folder:



View Contents of Folder:



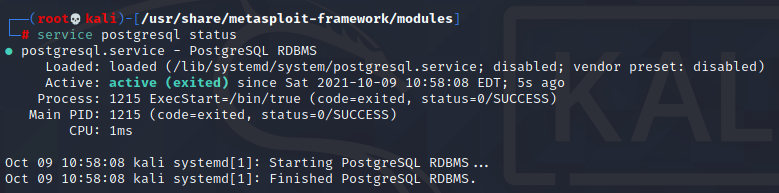
Access Modules folder:



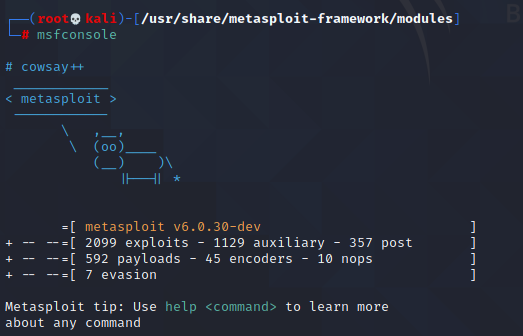
View Contents of Folder:



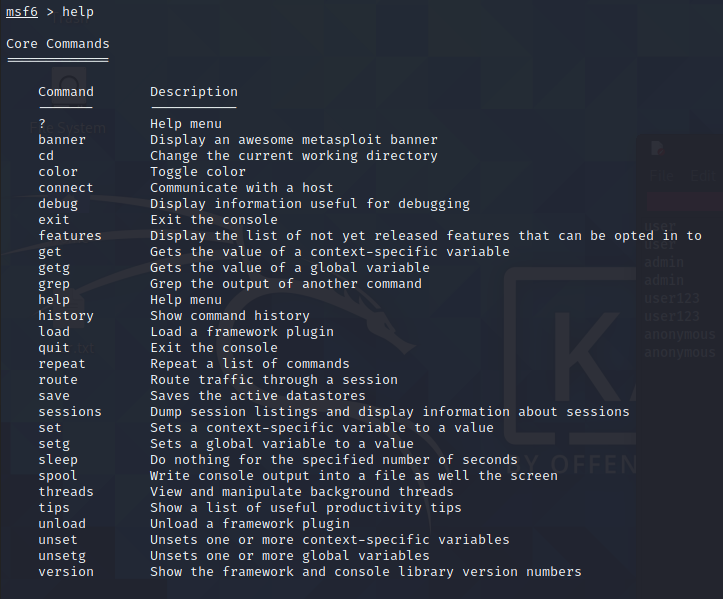
Connect to Database:

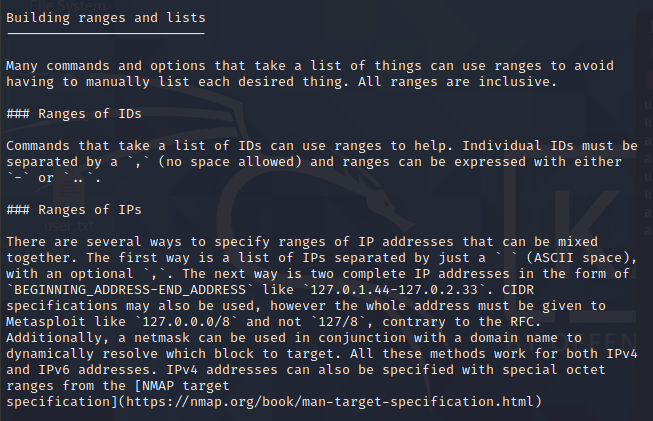
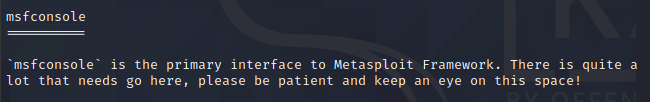
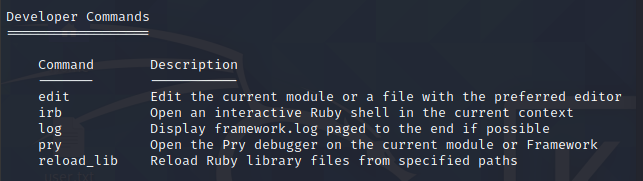
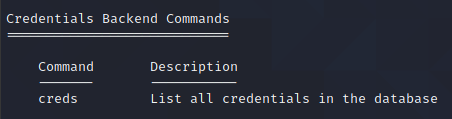
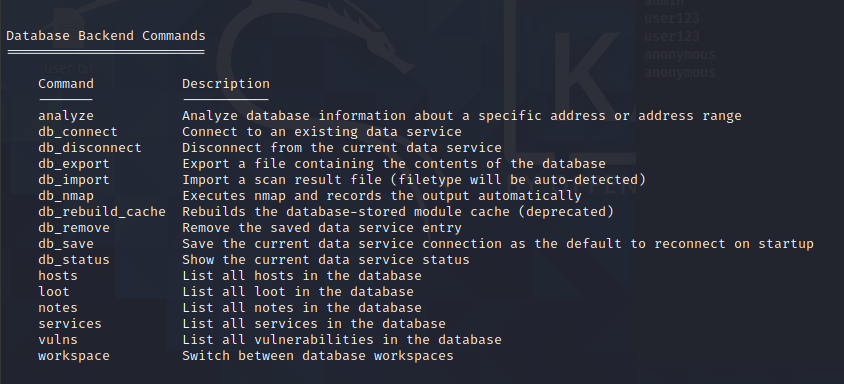
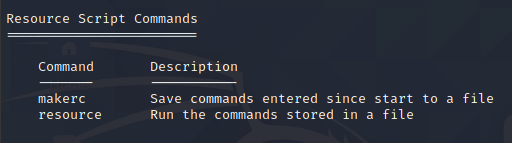
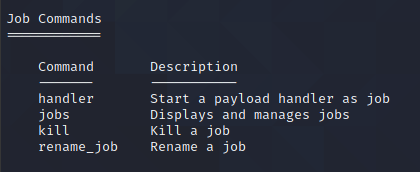
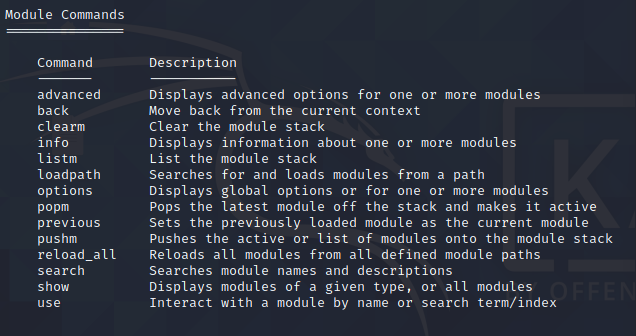
Check database status:

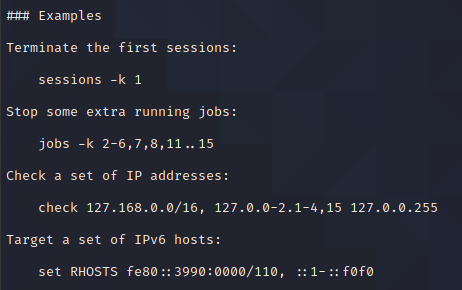
Launch Metasploit:



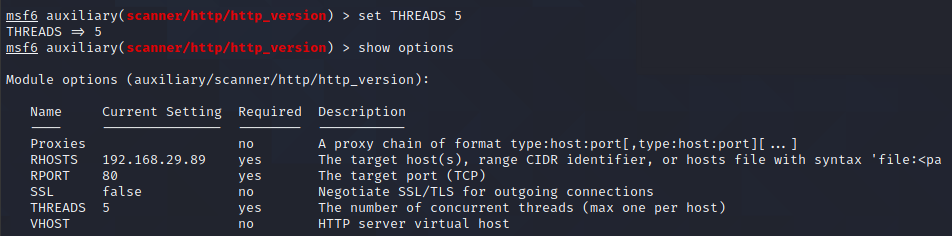
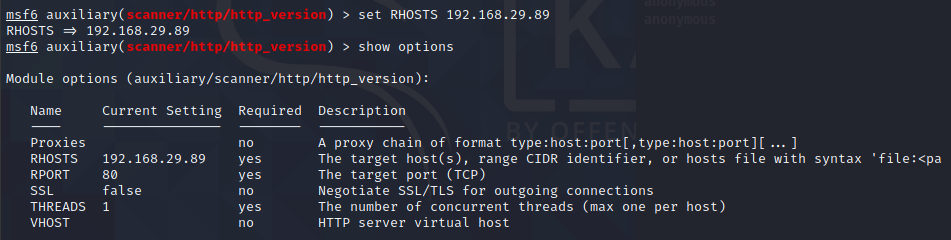
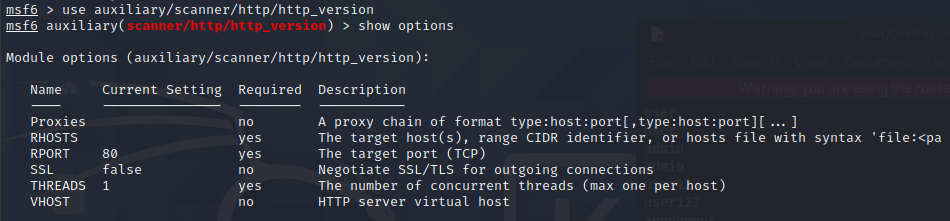
View commands:

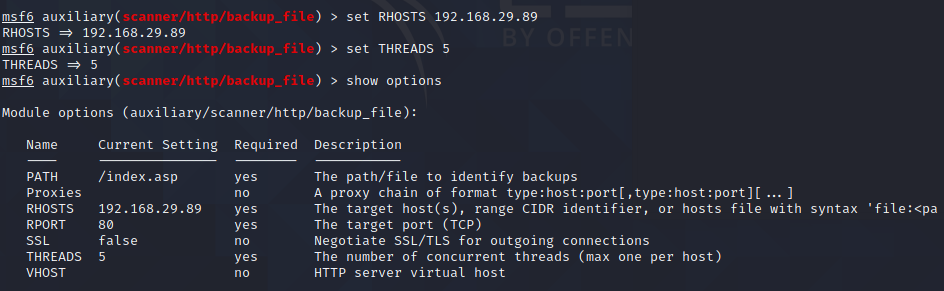
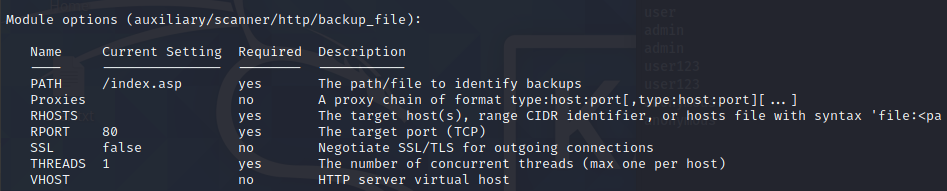
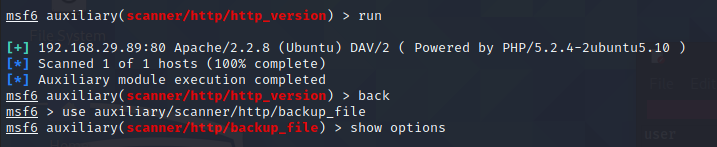
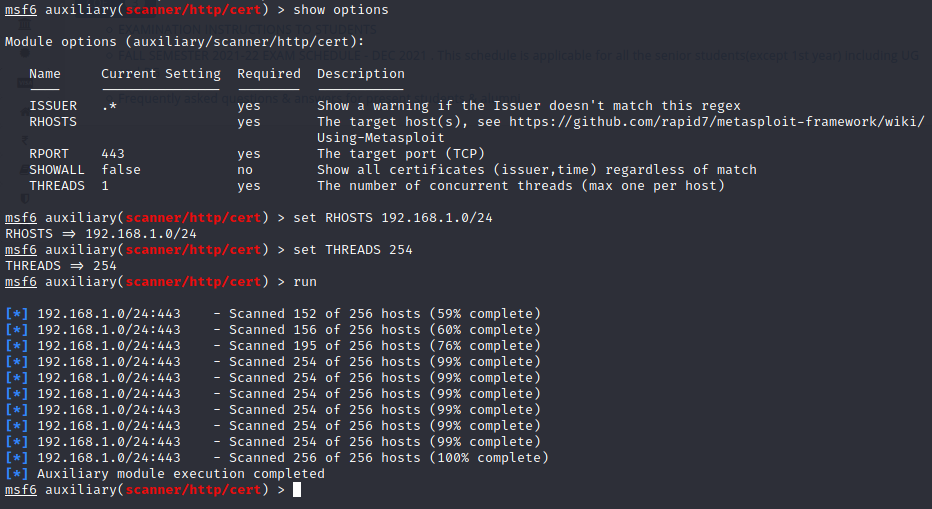


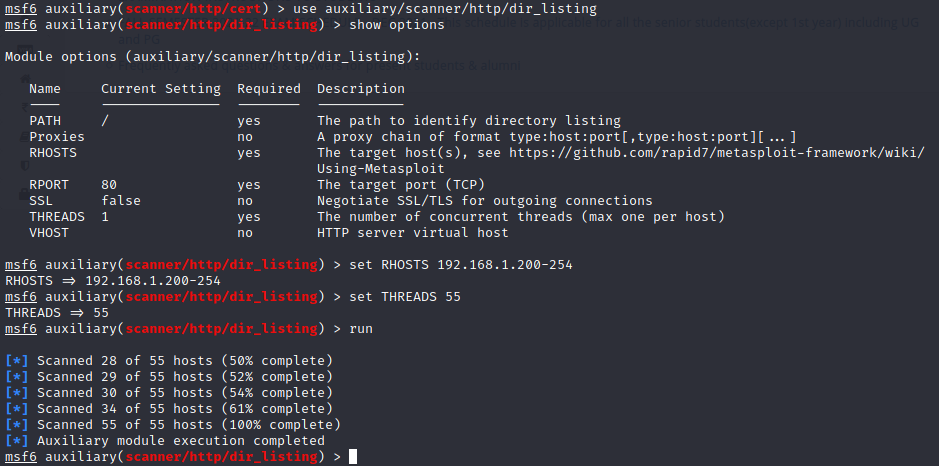
1. 



**HTTP:**

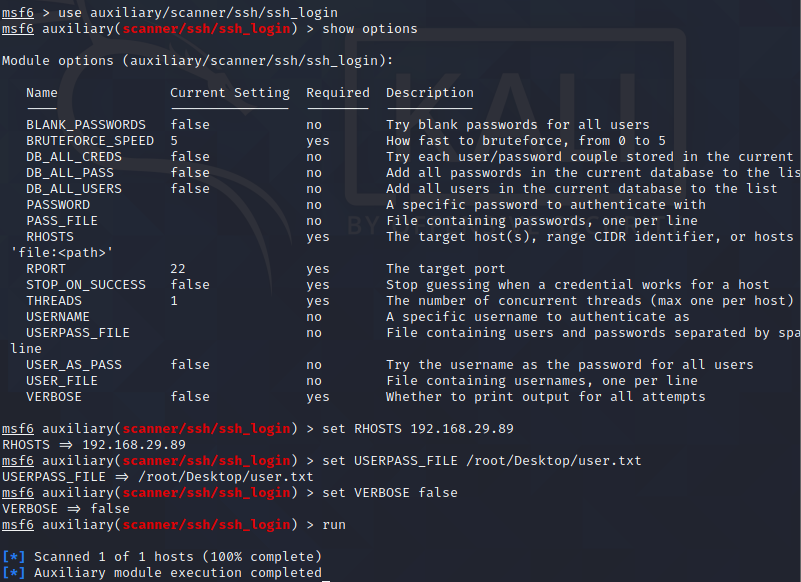
The **http\_version** scanner will scan a range of hosts and determine the web server version that is running on them.

To run the scan, we set the RHOSTS and THREADS values and let it run. 



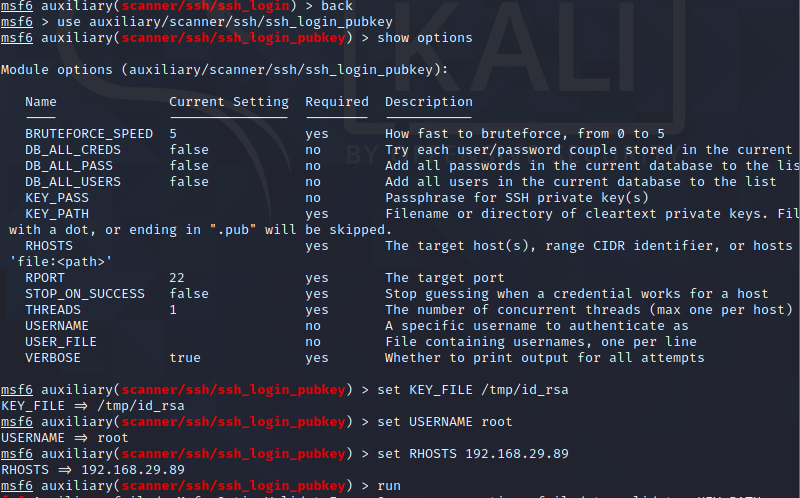
**SECURE SHELL**

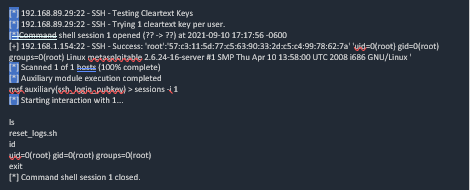
The **ssh\_login** module is quite versatile in that it can not only test a set of credentials across a range of IP addresses, but it can also perform brute force login attempts. We will pass a file to the module containing usernames and passwords separated by a space as shown below. Next, we load up the scanner module in Metasploit and set USERPASS\_FILE to point to our list of credentials to attempt.

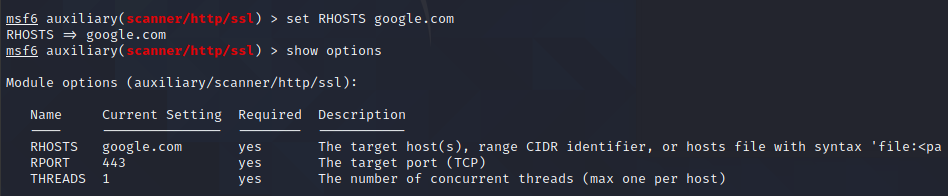
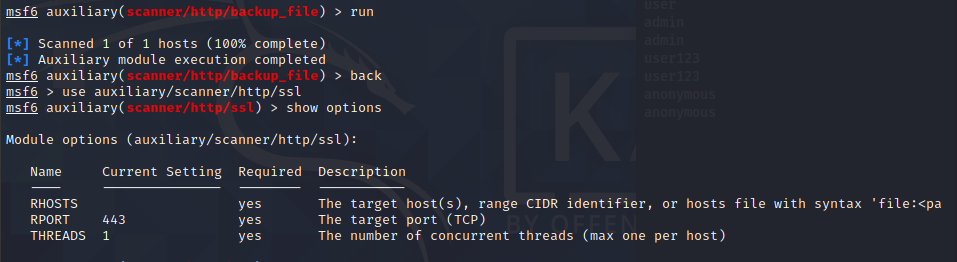


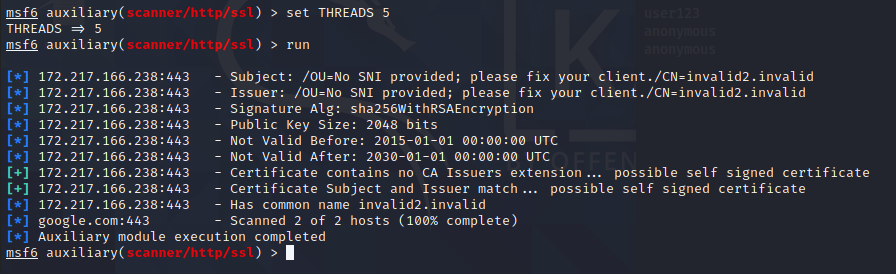
With everything ready to go, we run the module.

Using public key authentication for SSH is highly regarded as being far more secure than using usernames and passwords to authenticate. The caveat to this is that if the private key portion of the key pair is not kept secure, the security of the configuration is thrown right out the window. If, during an engagement, you get access to a private SSH key, you can use the **ssh\_login\_pubkey** module to attempt to login across a range of devices.



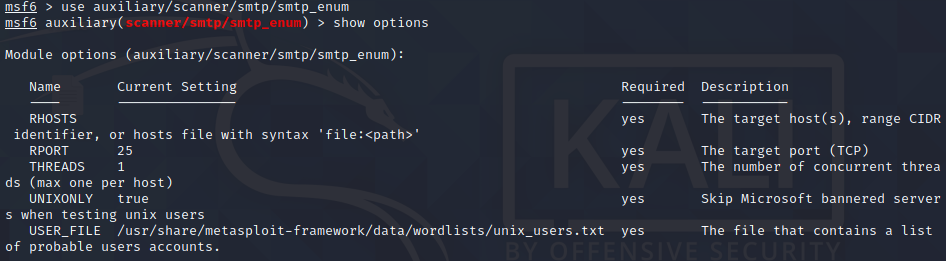


The **ssl** module queries a host or range of hosts and pull the SSL certificate information if present.

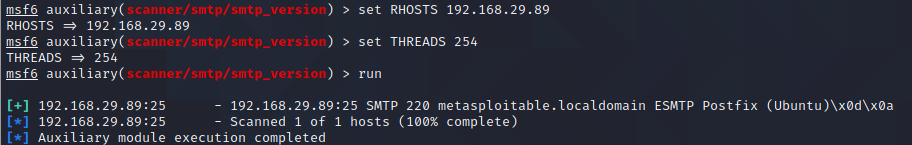
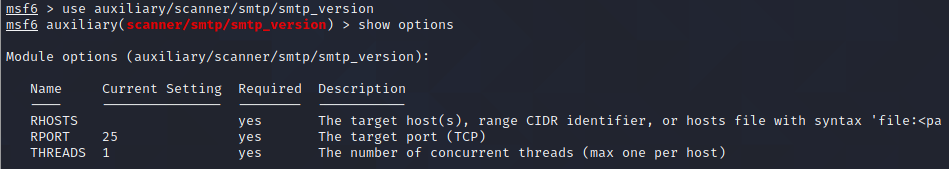
To configure the module, we set our RHOSTS and THREADS values and let it run.

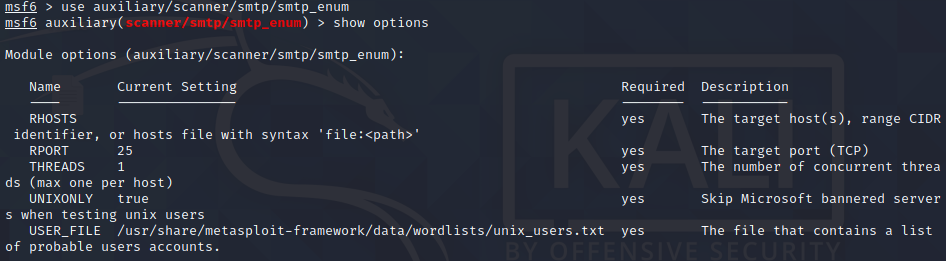
**SMTP:**

The SMTP Enumeration module will connect to a given mail server and use a wordlist to enumerate users that are present on the remote system.



 Since the email username and system username are frequently the same, you can now use any enumerated users for further logon attempts against other network services.

Poorly configured or vulnerable mail servers can often provide an initial foothold into a network but prior to launching an attack, we want to fingerprint the server to make our targeting as precise as possible. The **smtp\_version** module, as its name implies, will scan a range of IP addresses and determine the version of any mail servers it encounters.

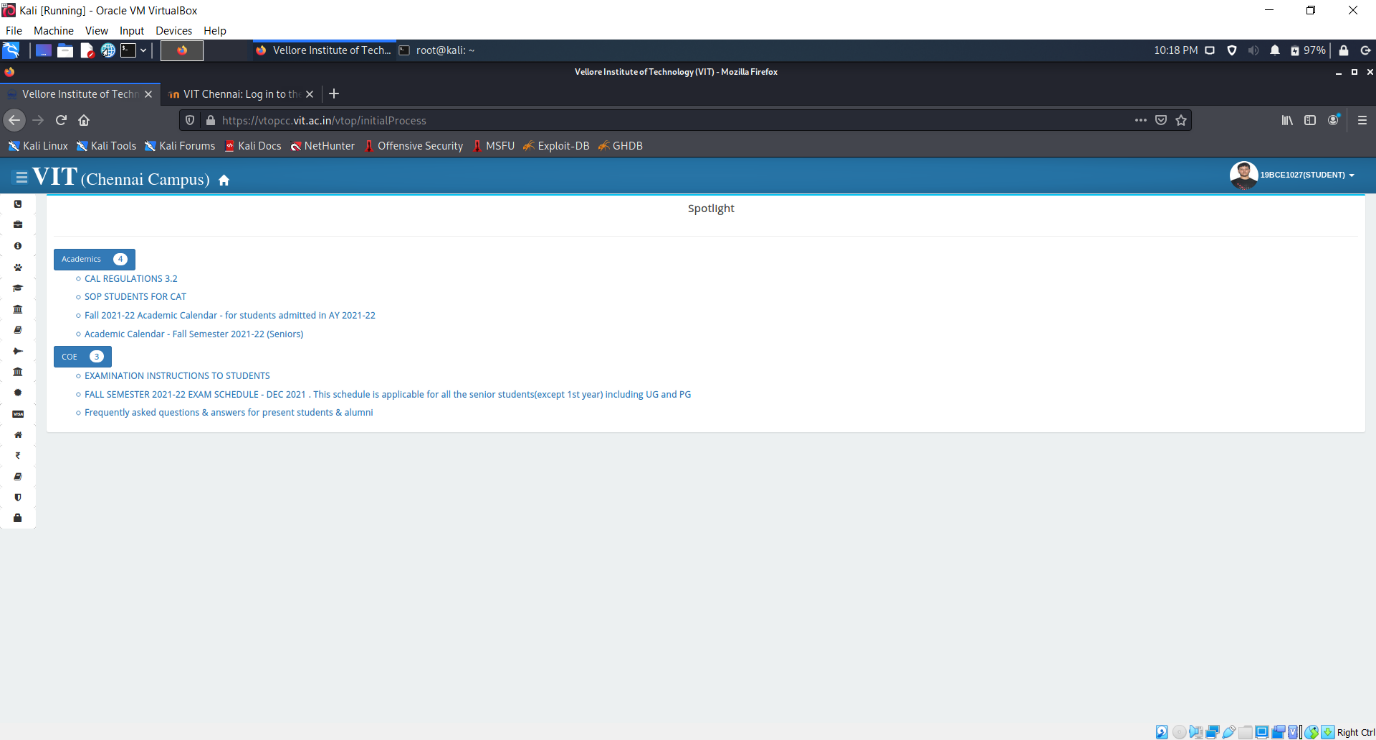


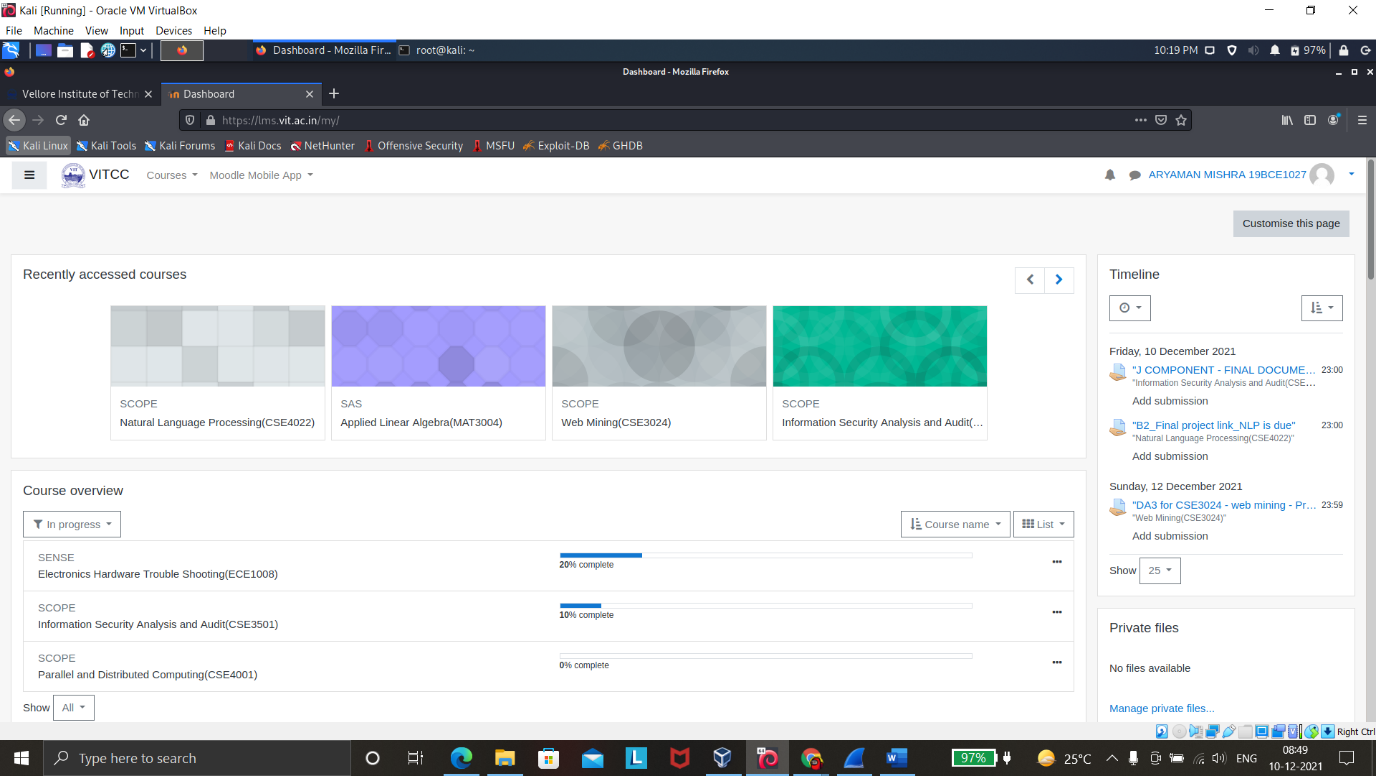


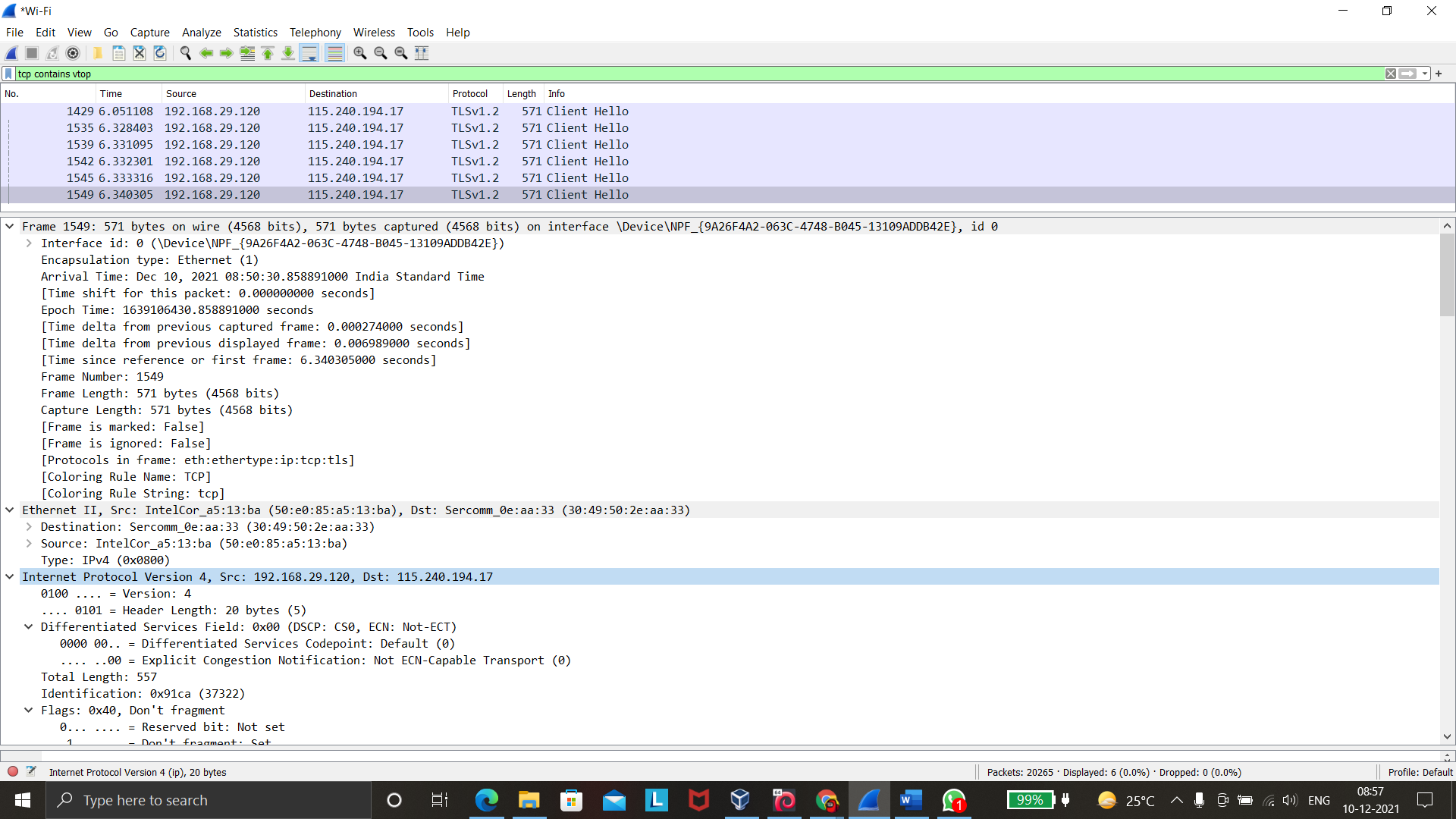
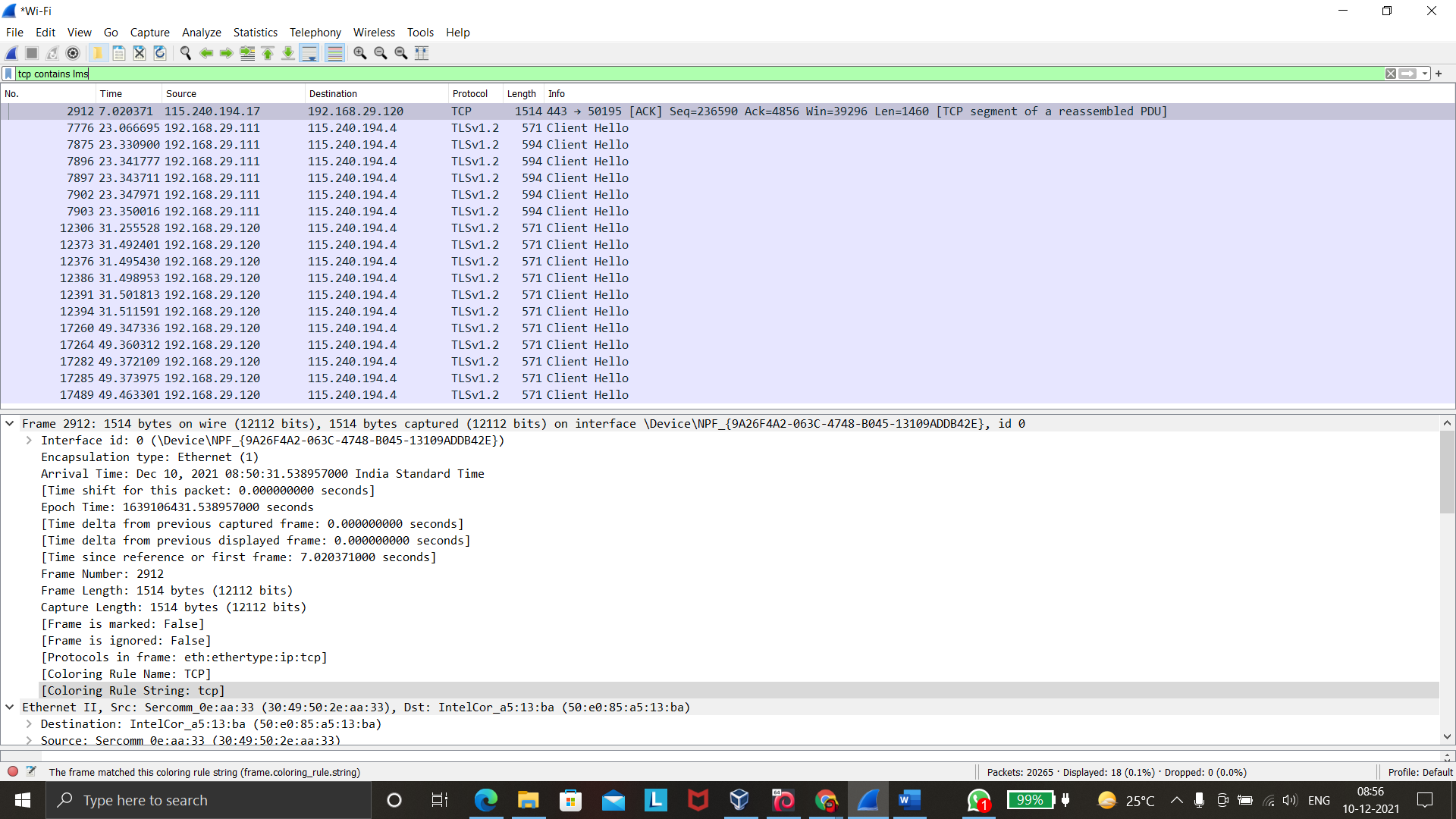
ARYAMAN MISHRA

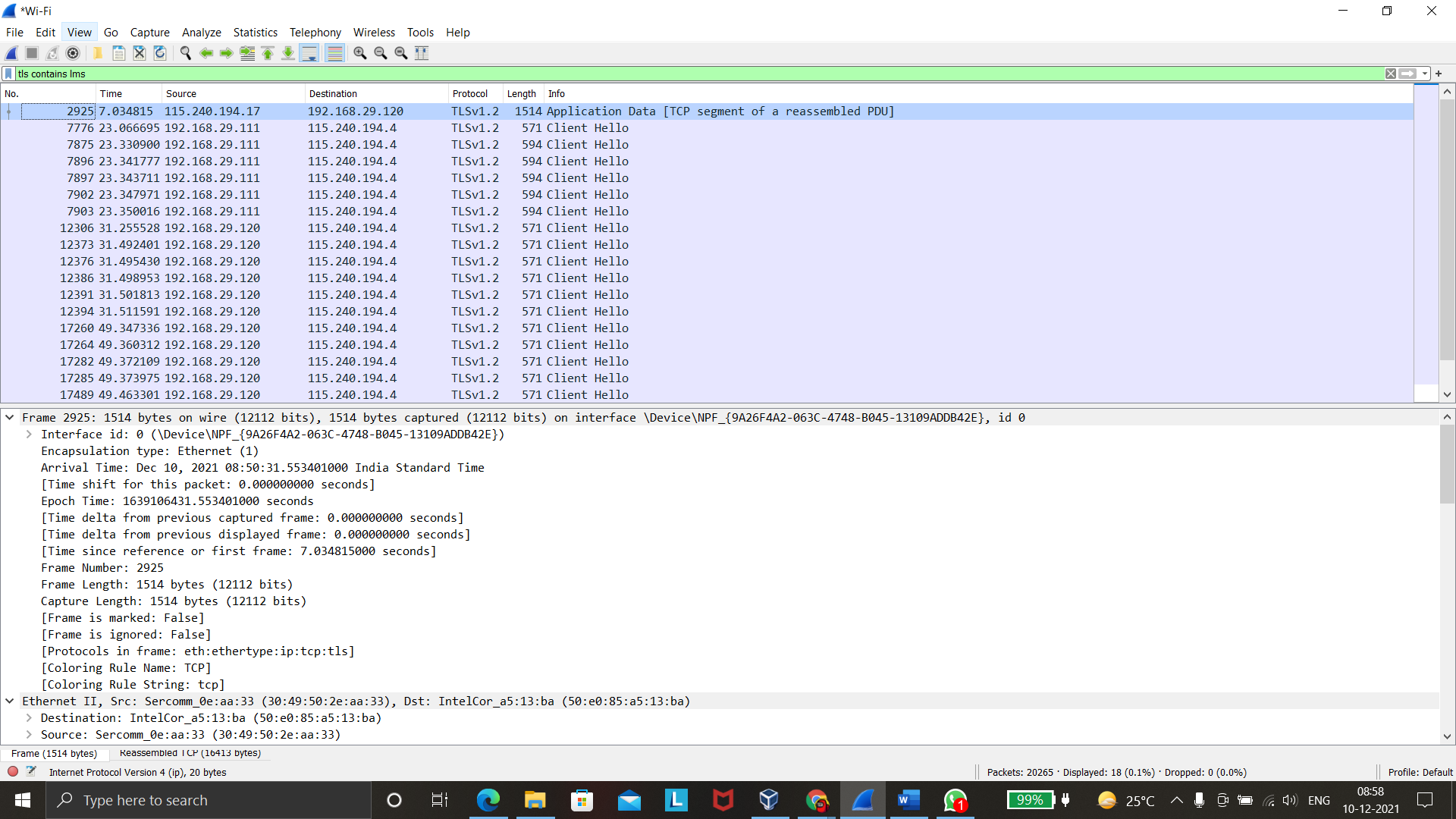
19BCE1027

Enter into MOODLE and VTOP applications using the necessary login credentials and identify the user names and passwords of the two applications in the trace files of Wireshark. Display the screenshots that are showing the usernames and passwords in the trace and the respective ASCII codes as well.





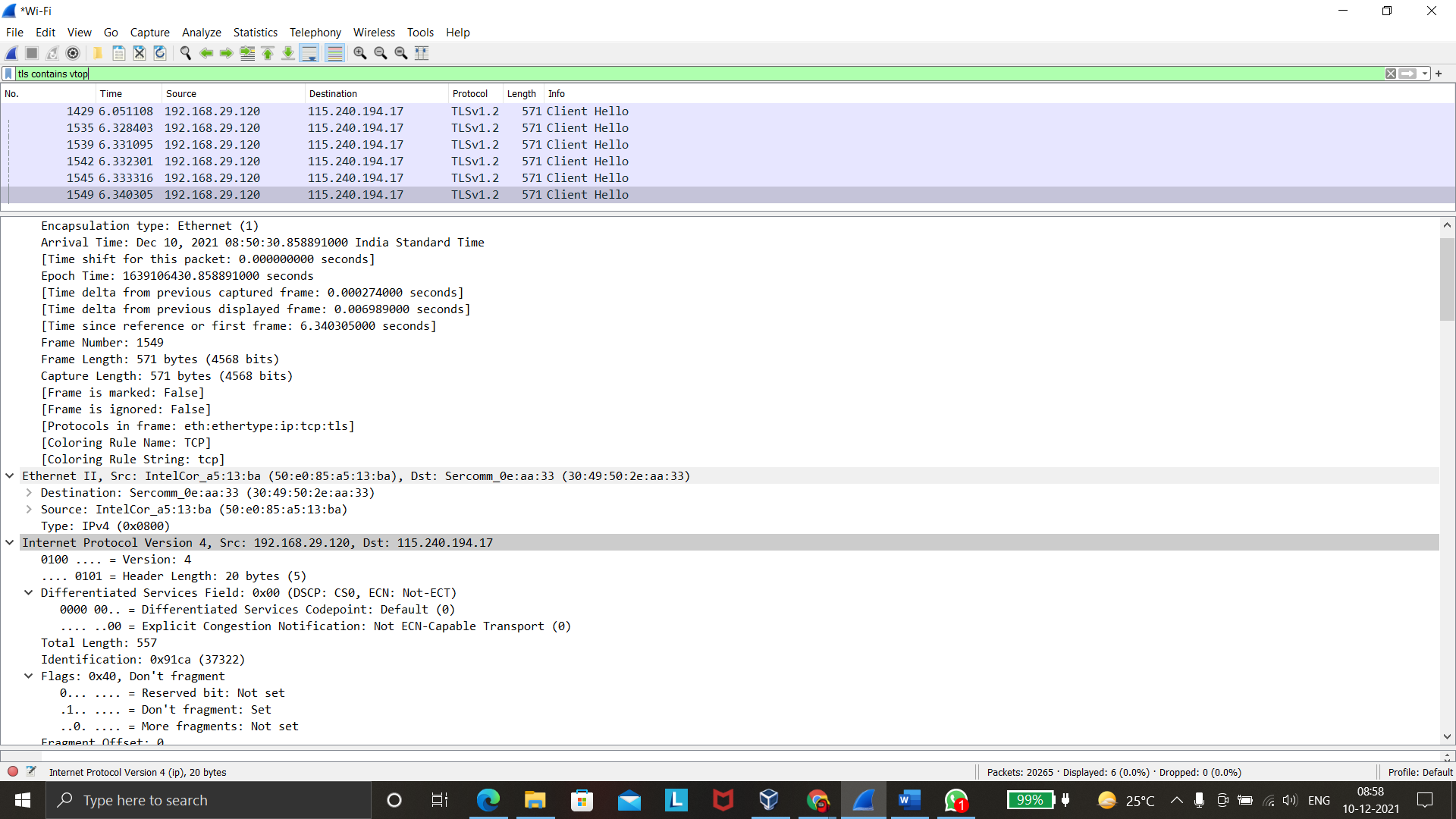


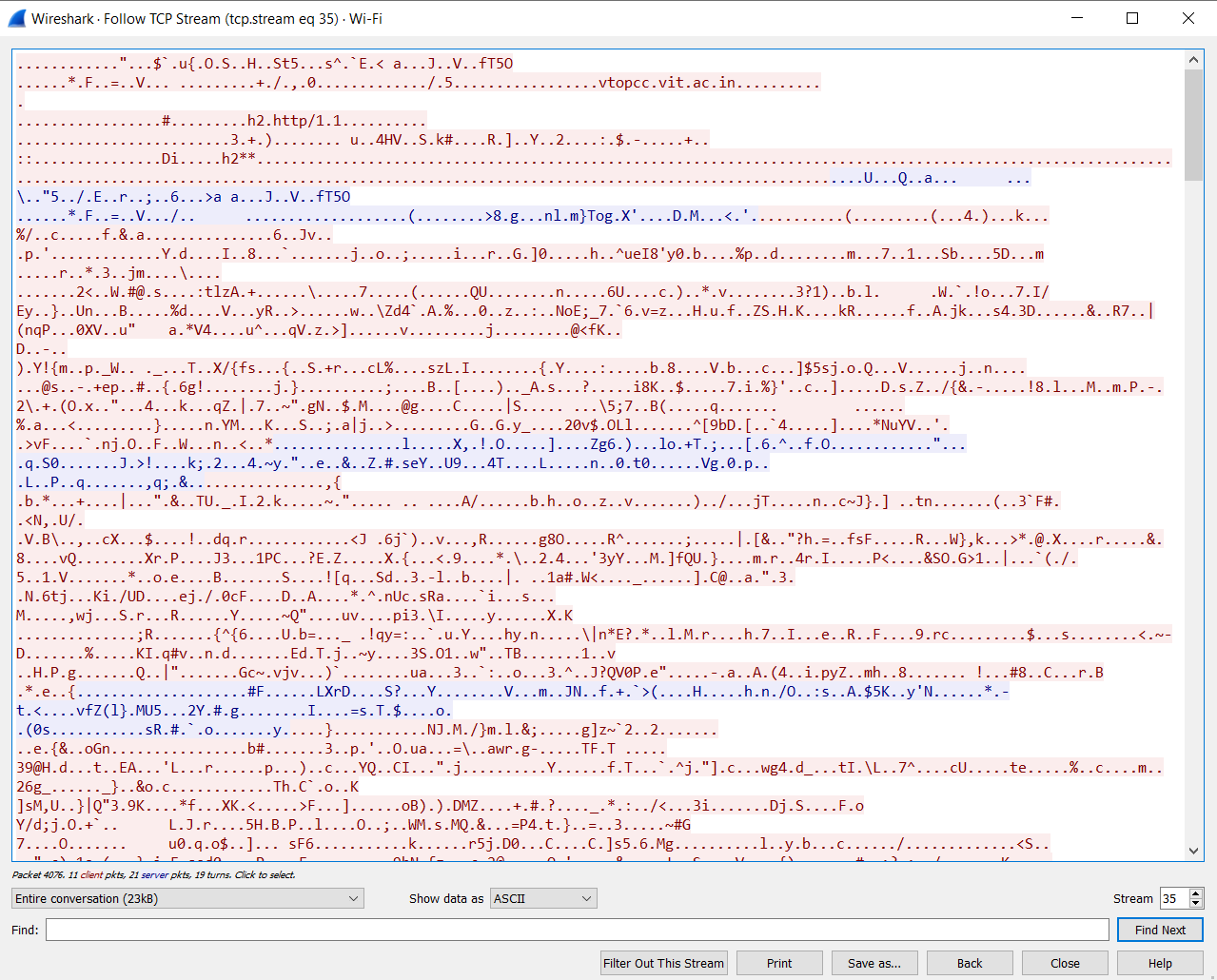
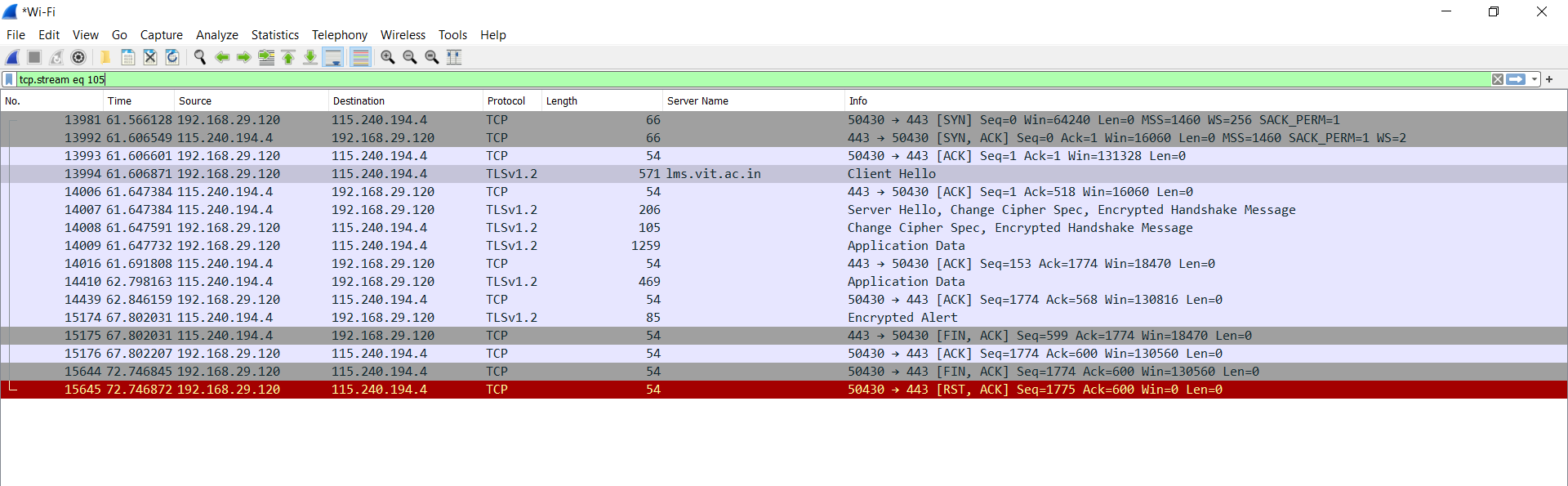
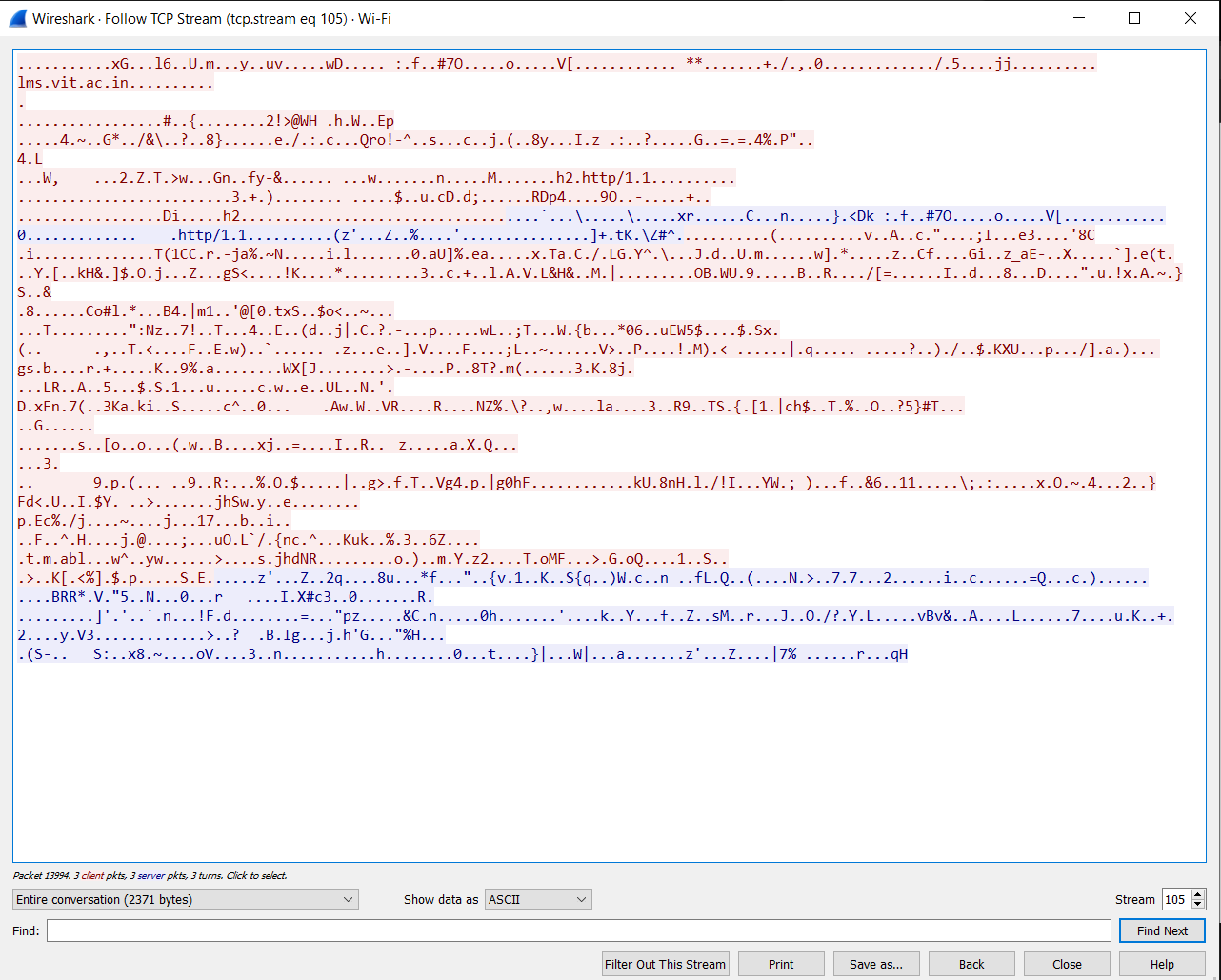
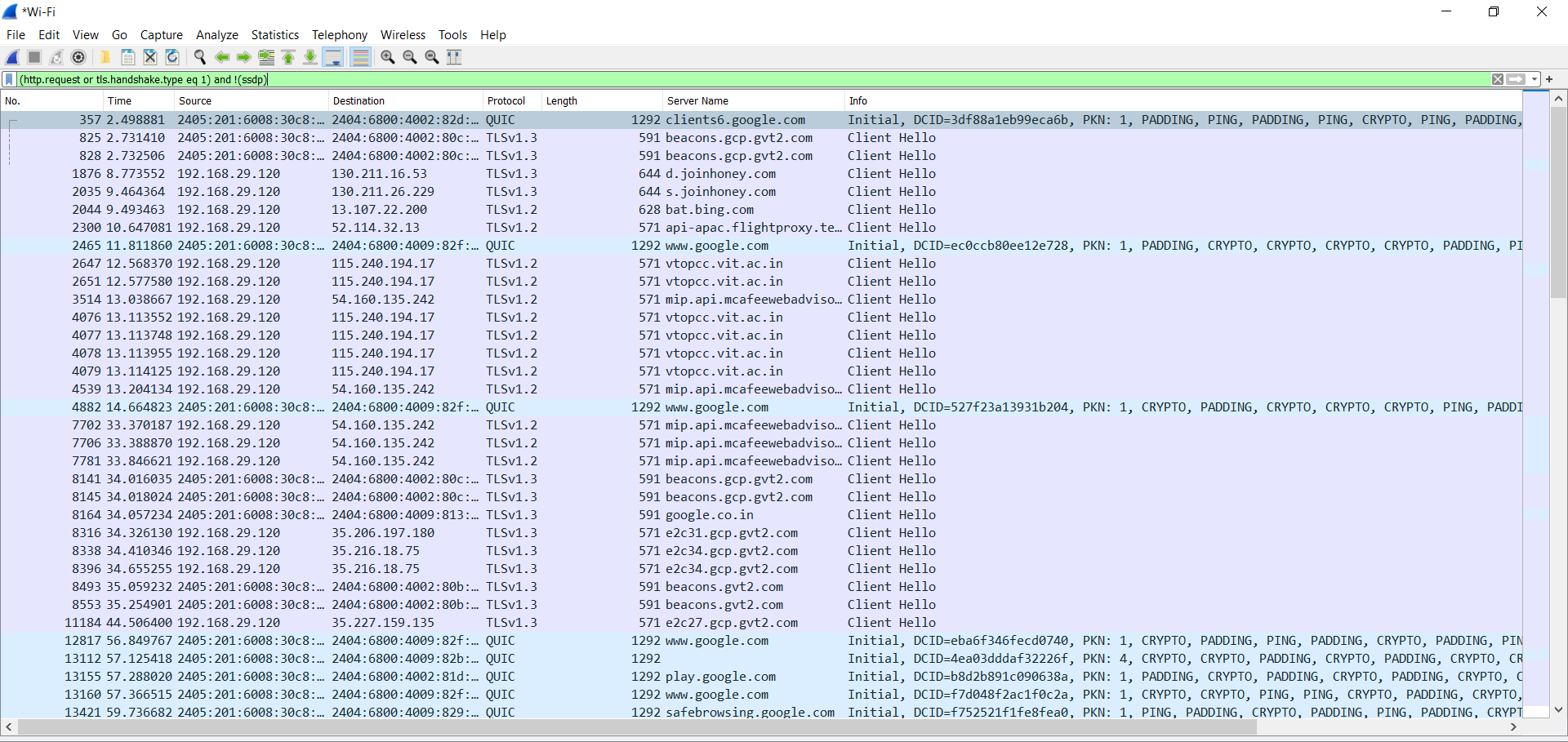
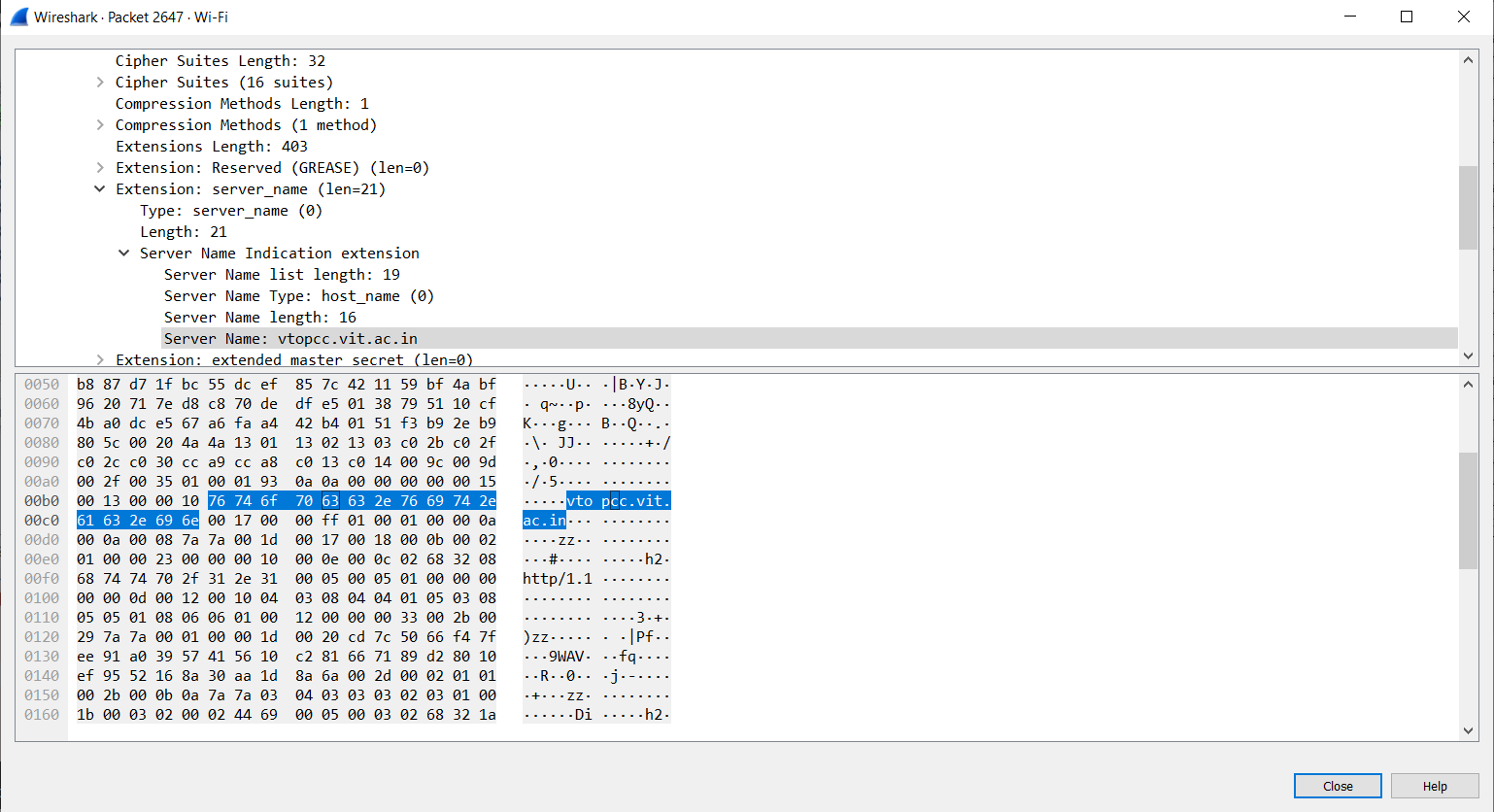


Open **Wireshark-tutorial-on-decrypting-HTTPS-SSL-TLS-traffic.pcap** in Wireshark. Use a basic web filter as described in this previous [tutorial about Wireshark filters](https://unit42.paloaltonetworks.com/using-wireshark-display-filter-expressions/). Our basic filter for Wireshark 3.x is:

(http.request or tls.handshake.type eq 1) and !(ssdp)

Open **Wireshark-tutorial-on-decrypting-HTTPS-SSL-TLS-traffic.pcap** in Wireshark. Then use the menu path **Edit --> Preferences** to bring up the Preferences Menu



Once you have clicked “OK,” when using the basic filter, your Wireshark column display will list the decrypted HTTP requests under each of the HTTPS lines

SHA Values for Passwords are

31cf42b2a7c5c558f44cfc67684cc344c17d4946d3a1e0b2cecb8eb58173cb2f

29!-!.

0d5ad14787be9df27b50dd7b99279b1b64c8bffd752e19a485d189467191bdca

Y|!B