Aryaman Mishra

19BCE1027

i) 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/**.

```
┌──(root💀kali)-[~]
└─# ls /usr/share/metasploit-framework/modules
auxiliary  encoders  evasion  exploits  nops  payloads  post
```

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.

```
┌──(root💀kali)-[~]
└─# ls /usr/share/metasploit-framework/modules/exploits/
aix         bsd     example_linux_priv_esc.rb  example_webapp.rb  hpux   mainframe  openbsd  solaris
android     bsdi    example.py                 firefox            irix   multi      osx      unix
apple_ios   dialup  example.rb                 freebsd            linux  netware    qnx      windows
```

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.

```
┌──(root💀kali)-[~]
└─# ls /usr/share/metasploit-framework/modules/auxiliary/
admin    bnat    cloud    docx  example.py  fileformat  gather  pdf      server   spoof  voip
analyze  client  crawler  dos   example.rb  fuzzers     parser  scanner  sniffer  sqli   vsploit
```

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

```
┌──(root💀kali)-[~]
└─# ls /usr/share/metasploit-framework/modules/payloads/
singles  stagers  stages
```

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.

```
┌──(root💀kali)-[~]
└─# ls /usr/share/metasploit-framework/modules/encoders/
cmd  generic  mipsbe  mipsle  php  ppc  ruby  sparc  x64  x86
```

(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:

```
┌──(root💀kali)-[~]
└─# cd /usr/share/metasploit-framework/
```

View Contents of Folder:

```
┌──(root💀kali)-[/usr/share/metasploit-framework]
└─# ls
app     data  documentation  Gemfile.lock  metasploit-framework.gemspec  msfconsole  msfdb           msfrpc   msfupd
config  db    Gemfile        lib           modules                       msfd        msf-json-rpc.ru  msfrpcd  msfven
```

Access Modules folder:

```
┌──(root💀kali)-[/usr/share/metasploit-framework]
└─# cd modules
```

View Contents of Folder:

```
┌──(root💀kali)-[/usr/share/metasploit-framework/modules]
└─# ls
auxiliary  encoders  evasion  exploits  nops  payloads  post
```

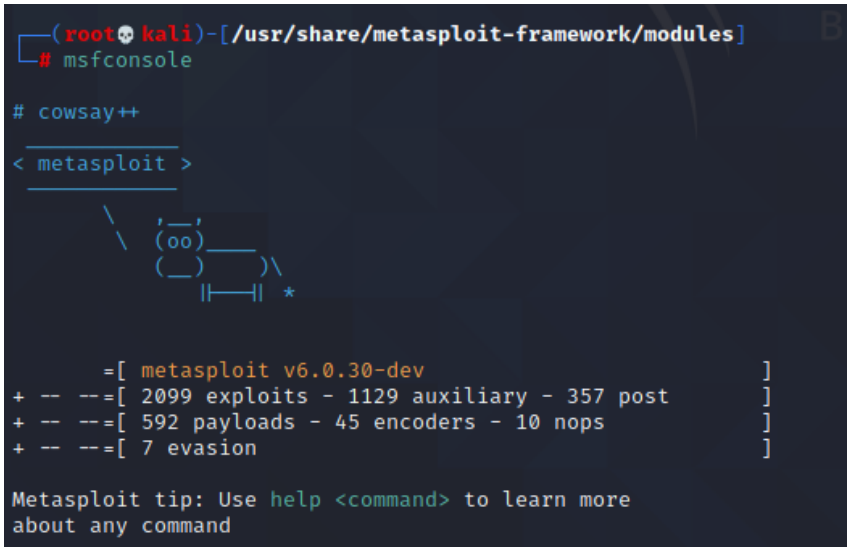
Connect to Database:

```
┌──(root💀kali)-[/usr/share/metasploit-framework/modules]
└─# service postgresql start
```

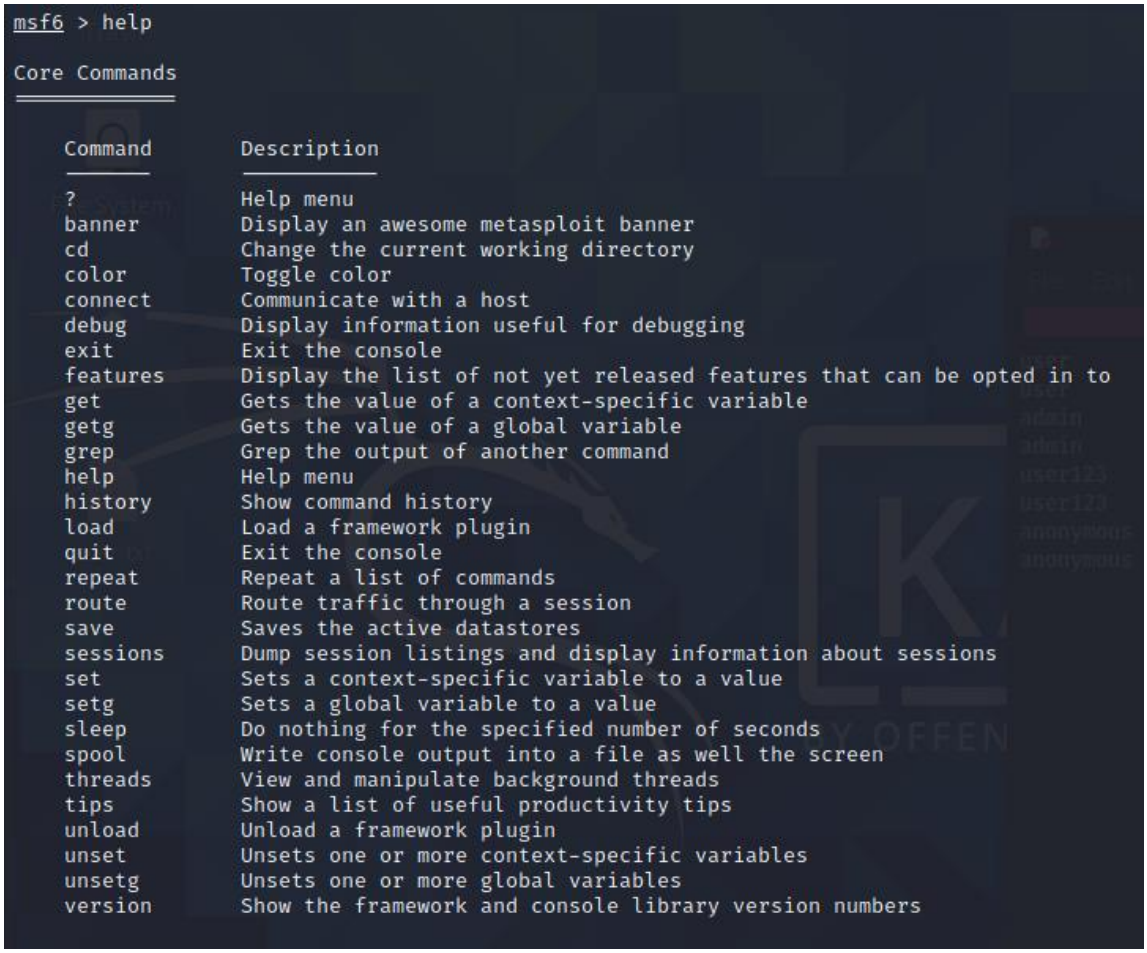
Check database status:

```
┌──(root💀kali)-[/usr/share/metasploit-framework/modules]
└─# service postgresql status
● postgresql.service - PostgreSQL RDBMS
     Loaded: loaded (/lib/systemd/system/postgresql.service; disabled; vendor preset: disabled)
     Active: active (exited) since Sat 2021-10-09 10:58:08 EDT; 5s ago
    Process: 1215 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 1215 (code=exited, status=0/SUCCESS)
        CPU: 1ms

Oct 09 10:58:08 kali systemd[1]: Starting PostgreSQL RDBMS ...
Oct 09 10:58:08 kali systemd[1]: Finished PostgreSQL RDBMS.
```

Launch Metasploit:



View commands:



ii)

```
Module Commands
===============

    Command       Description
    -------       -----------
    advanced      Displays advanced options for one or more modules
    back          Move back from the current context
    clearm        Clear the module stack
    info          Displays information about one or more modules
    listm         List the module stack
    loadpath      Searches for and loads modules from a path
    options       Displays global options or for one or more modules
    popm          Pops the latest module off the stack and makes it active
    previous      Sets the previously loaded module as the current module
    pushm         Pushes the active or list of modules onto the module stack
    reload_all    Reloads all modules from all defined module paths
    search        Searches module names and descriptions
    show          Displays modules of a given type, or all modules
    use           Interact with a module by name or search term/index
```

```
Job Commands
============

    Command       Description
    -------       -----------
    handler       Start a payload handler as job
    jobs          Displays and manages jobs
    kill          Kill a job
    rename_job    Rename a job
```

```
Resource Script Commands
========================

    Command       Description
    -------       -----------
    makerc        Save commands entered since start to a file
    resource      Run the commands stored in a file
```

```
Database Backend Commands
=========================

    Command           Description
    -------           -----------
    analyze           Analyze database information about a specific address or address range
    db_connect        Connect to an existing data service
    db_disconnect     Disconnect from the current data service
    db_export         Export a file containing the contents of the database
    db_import         Import a scan result file (filetype will be auto-detected)
    db_nmap           Executes nmap and records the output automatically
    db_rebuild_cache  Rebuilds the database-stored module cache (deprecated)
    db_remove         Remove the saved data service entry
    db_save           Save the current data service connection as the default to reconnect on startup
    db_status         Show the current data service status
    hosts             List all hosts in the database
    loot              List all loot in the database
    notes             List all notes in the database
    services          List all services in the database
    vulns             List all vulnerabilities in the database
    workspace         Switch between database workspaces
```

Credentials Backend Commands

Command	Description
creds	List all credentials in the database

Developer Commands

Command	Description
edit	Edit the current module or a file with the preferred editor
irb	Open an interactive Ruby shell in the current context
log	Display framework.log paged to the end if possible
pry	Open the Pry debugger on the current module or Framework
reload_lib	Reload Ruby library files from specified paths

msfconsole

`msfconsole` is the primary interface to Metasploit Framework. There is quite a lot that needs go here, please be patient and keep an eye on this space!

Building ranges and lists

Many commands and options that take a list of things can use ranges to avoid having to manually list each desired thing. All ranges are inclusive.

Ranges of IDs

Commands that take a list of IDs can use ranges to help. Individual IDs must be separated by a `,` (no space allowed) and ranges can be expressed with either `-` or `..`.

Ranges of IPs

There are several ways to specify ranges of IP addresses that can be mixed together. The first way is a list of IPs separated by just a ` ` (ASCII space), with an optional `,`. The next way is two complete IP addresses in the form of `BEGINNING_ADDRESS-END_ADDRESS` like `127.0.1.44-127.0.2.33`. CIDR specifications may also be used, however the whole address must be given to Metasploit like `127.0.0.0/8` and not `127/8`, contrary to the RFC. Additionally, a netmask can be used in conjunction with a domain name to dynamically resolve which block to target. All these methods work for both IPv4 and IPv6 addresses. IPv4 addresses can also be specified with special octet ranges from the [NMAP target specification](https://nmap.org/book/man-target-specification.html)

```
### Examples

Terminate the first sessions:

    sessions -k 1

Stop some extra running jobs:

    jobs -k 2-6,7,8,11..15

Check a set of IP addresses:

    check 127.168.0.0/16, 127.0.0-2.1-4,15 127.0.0.255

Target a set of IPv6 hosts:

    set RHOSTS fe80::3990:0000/110, ::1-::f0f0
```

```
Target a block from a resolved domain name:

    set RHOSTS www.example.test/24
```

HTTP:

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

```
msf6 > use auxiliary/scanner/http/http_version
msf6 auxiliary(scanner/http/http_version) > show options

Module options (auxiliary/scanner/http/http_version):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS                    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<pa
   RPORT    80               yes       The target port (TCP)
   SSL      false            no        Negotiate SSL/TLS for outgoing connections
   THREADS  1                yes       The number of concurrent threads (max one per host)
   VHOST                     no        HTTP server virtual host
```

```
msf6 auxiliary(scanner/http/http_version) > set RHOSTS 192.168.29.89
RHOSTS => 192.168.29.89
msf6 auxiliary(scanner/http/http_version) > show options

Module options (auxiliary/scanner/http/http_version):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS   192.168.29.89    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<pa
   RPORT    80               yes       The target port (TCP)
   SSL      false            no        Negotiate SSL/TLS for outgoing connections
   THREADS  1                yes       The number of concurrent threads (max one per host)
   VHOST                     no        HTTP server virtual host
```

```
msf6 auxiliary(scanner/http/http_version) > set THREADS 5
THREADS => 5
msf6 auxiliary(scanner/http/http_version) > show options

Module options (auxiliary/scanner/http/http_version):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS   192.168.29.89    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<pa
   RPORT    80               yes       The target port (TCP)
   SSL      false            no        Negotiate SSL/TLS for outgoing connections
   THREADS  5                yes       The number of concurrent threads (max one per host)
   VHOST                     no        HTTP server virtual host
```

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

```
msf6 auxiliary(scanner/http/http_version) > run

[+] 192.168.29.89:80 Apache/2.2.8 (Ubuntu) DAV/2 ( Powered by PHP/5.2.4-2ubuntu5.10 )
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/http_version) > back
msf6 > use auxiliary/scanner/http/backup_file
msf6 auxiliary(scanner/http/backup_file) > show options
```

```
Module options (auxiliary/scanner/http/backup_file):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   PATH     /index.asp       yes       The path/file to identify backups
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS                    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<pa
   RPORT    80               yes       The target port (TCP)
   SSL      false            no        Negotiate SSL/TLS for outgoing connections
   THREADS  1                yes       The number of concurrent threads (max one per host)
   VHOST                     no        HTTP server virtual host
```

```
msf6 auxiliary(scanner/http/backup_file) > set RHOSTS 192.168.29.89
RHOSTS => 192.168.29.89
msf6 auxiliary(scanner/http/backup_file) > set THREADS 5
THREADS => 5
msf6 auxiliary(scanner/http/backup_file) > show options

Module options (auxiliary/scanner/http/backup_file):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   PATH     /index.asp       yes       The path/file to identify backups
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS   192.168.29.89    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<pa
   RPORT    80               yes       The target port (TCP)
   SSL      false            no        Negotiate SSL/TLS for outgoing connections
   THREADS  5                yes       The number of concurrent threads (max one per host)
   VHOST                     no        HTTP server virtual host
```

```
msf6 auxiliary(scanner/http/cert) > show options

Module options (auxiliary/scanner/http/cert):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   ISSUER   .*               yes       Show a warning if the Issuer doesn't match this regex
   RHOSTS                    yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/
                                       Using-Metasploit
   RPORT    443              yes       The target port (TCP)
   SHOWALL  false            no        Show all certificates (issuer,time) regardless of match
   THREADS  1                yes       The number of concurrent threads (max one per host)

msf6 auxiliary(scanner/http/cert) > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
msf6 auxiliary(scanner/http/cert) > set THREADS 254
THREADS => 254
msf6 auxiliary(scanner/http/cert) > run

[*] 192.168.1.0/24:443    - Scanned 152 of 256 hosts (59% complete)
[*] 192.168.1.0/24:443    - Scanned 156 of 256 hosts (60% complete)
[*] 192.168.1.0/24:443    - Scanned 195 of 256 hosts (76% complete)
[*] 192.168.1.0/24:443    - Scanned 254 of 256 hosts (99% complete)
[*] 192.168.1.0/24:443    - Scanned 254 of 256 hosts (99% complete)
[*] 192.168.1.0/24:443    - Scanned 254 of 256 hosts (99% complete)
[*] 192.168.1.0/24:443    - Scanned 254 of 256 hosts (99% complete)
[*] 192.168.1.0/24:443    - Scanned 254 of 256 hosts (99% complete)
[*] 192.168.1.0/24:443    - Scanned 254 of 256 hosts (99% complete)
[*] 192.168.1.0/24:443    - Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/cert) >
```

```
msf6 auxiliary(scanner/http/cert) > use auxiliary/scanner/http/dir_listing
msf6 auxiliary(scanner/http/dir_listing) > show options

Module options (auxiliary/scanner/http/dir_listing):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   PATH     /                yes       The path to identify directory listing
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS                    yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/
                                       Using-Metasploit
   RPORT    80               yes       The target port (TCP)
   SSL      false            no        Negotiate SSL/TLS for outgoing connections
   THREADS  1                yes       The number of concurrent threads (max one per host)
   VHOST                     no        HTTP server virtual host

msf6 auxiliary(scanner/http/dir_listing) > set RHOSTS 192.168.1.200-254
RHOSTS => 192.168.1.200-254
msf6 auxiliary(scanner/http/dir_listing) > set THREADS 55
THREADS => 55
msf6 auxiliary(scanner/http/dir_listing) > run

[*] Scanned 28 of 55 hosts (50% complete)
[*] Scanned 29 of 55 hosts (52% complete)
[*] Scanned 30 of 55 hosts (54% complete)
[*] Scanned 34 of 55 hosts (61% complete)
[*] Scanned 55 of 55 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/dir_listing) >
```

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.

```
msf6 > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) > show options

Module options (auxiliary/scanner/ssh/ssh_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   BLANK_PASSWORDS   false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the current
   DB_ALL_PASS       false            no        Add all passwords in the current database to the lis
   DB_ALL_USERS      false            no        Add all users in the current database to the list
   PASSWORD                           no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   RHOSTS                             yes       The target host(s), range CIDR identifier, or hosts
 'file:<path>'
   RPORT             22               yes       The target port
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a host
   THREADS           1                yes       The number of concurrent threads (max one per host)
   USERNAME                           no        A specific username to authenticate as
   USERPASS_FILE                      no        File containing users and passwords separated by spa
 line
   USER_AS_PASS      false            no        Try the username as the password for all users
   USER_FILE                          no        File containing usernames, one per line
   VERBOSE           false            yes       Whether to print output for all attempts

msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.29.89
RHOSTS => 192.168.29.89
msf6 auxiliary(scanner/ssh/ssh_login) > set USERPASS_FILE /root/Desktop/user.txt
USERPASS_FILE => /root/Desktop/user.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set VERBOSE false
VERBOSE => false
msf6 auxiliary(scanner/ssh/ssh_login) > run

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

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.

```
msf6 auxiliary(scanner/ssh/ssh_login) > back
msf6 > use auxiliary/scanner/ssh/ssh_login_pubkey
msf6 auxiliary(scanner/ssh/ssh_login_pubkey) > show options

Module options (auxiliary/scanner/ssh/ssh_login_pubkey):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the current
   DB_ALL_PASS       false            no        Add all passwords in the current database to the lis
   DB_ALL_USERS      false            no        Add all users in the current database to the list
   KEY_PASS                           no        Passphrase for SSH private key(s)
   KEY_PATH                           yes       Filename or directory of cleartext private keys. Fil
 with a dot, or ending in ".pub" will be skipped.
   RHOSTS                             yes       The target host(s), range CIDR identifier, or hosts
 'file:<path>'
   RPORT             22               yes       The target port
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a host
   THREADS           1                yes       The number of concurrent threads (max one per host)
   USERNAME                           no        A specific username to authenticate as
   USER_FILE                          no        File containing usernames, one per line
   VERBOSE           true             yes       Whether to print output for all attempts

msf6 auxiliary(scanner/ssh/ssh_login_pubkey) > set KEY_FILE /tmp/id_rsa
KEY_FILE ⇒ /tmp/id_rsa
msf6 auxiliary(scanner/ssh/ssh_login_pubkey) > set USERNAME root
USERNAME ⇒ root
msf6 auxiliary(scanner/ssh/ssh_login_pubkey) > set RHOSTS 192.168.29.89
RHOSTS ⇒ 192.168.29.89
msf6 auxiliary(scanner/ssh/ssh_login_pubkey) > run
```

```
[*] 192.168.89.29:22 - SSH - Testing Cleartext Keys
[*] 192.168.89.29:22 - SSH - Trying 1 cleartext key per user.
[*] Command shell session 1 opened (?? -> ??) at 2021-09-10 17:17:56 -0600
[+] 192.168.1.154:22 - SSH - Success: 'root':'57:c3:11:5d:77:c5:63:90:33:2d:c5:c4:99:78:62:7a' 'uid=0(root) gid=0(root)
groups=0(root) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(ssh_login_pubkey) > sessions -i 1
[*] Starting interaction with 1...

ls
reset_logs.sh
id
uid=0(root) gid=0(root) groups=0(root)
exit
[*] Command shell session 1 closed.
```

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

```
msf6 auxiliary(scanner/http/backup_file) > run

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/backup_file) > back
msf6 > use auxiliary/scanner/http/ssl
msf6 auxiliary(scanner/http/ssl) > show options

Module options (auxiliary/scanner/http/ssl):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<pa
   RPORT    443              yes       The target port (TCP)
   THREADS  1                yes       The number of concurrent threads (max one per host)
```

```
msf6 auxiliary(scanner/http/ssl) > set RHOSTS google.com
RHOSTS => google.com
msf6 auxiliary(scanner/http/ssl) > show options

Module options (auxiliary/scanner/http/ssl):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS   google.com       yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<pa
   RPORT    443              yes       The target port (TCP)
   THREADS  1                yes       The number of concurrent threads (max one per host)
```

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

```
msf6 auxiliary(scanner/http/ssl) > set THREADS 5
THREADS => 5
msf6 auxiliary(scanner/http/ssl) > run

[*] 172.217.166.238:443   - Subject: /OU=No SNI provided; please fix your client./CN=invalid2.invalid
[*] 172.217.166.238:443   - Issuer: /OU=No SNI provided; please fix your client./CN=invalid2.invalid
[*] 172.217.166.238:443   - Signature Alg: sha256WithRSAEncryption
[*] 172.217.166.238:443   - Public Key Size: 2048 bits
[*] 172.217.166.238:443   - Not Valid Before: 2015-01-01 00:00:00 UTC
[*] 172.217.166.238:443   - Not Valid After: 2030-01-01 00:00:00 UTC
[+] 172.217.166.238:443   - Certificate contains no CA Issuers extension... possible self signed certificate
[+] 172.217.166.238:443   - Certificate Subject and Issuer match... possible self signed certificate
[*] 172.217.166.238:443   - Has common name invalid2.invalid
[*] google.com:443        - Scanned 2 of 2 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/ssl) >
```

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.

```
msf6 > use auxiliary/scanner/smtp/smtp_enum
msf6 auxiliary(scanner/smtp/smtp_enum) > show options

Module options (auxiliary/scanner/smtp/smtp_enum):

   Name       Current Setting                                                  Required  Description
   ----       ---------------                                                  --------  -----------
   RHOSTS                                                                      yes       The target host(s), range CIDR
 identifier, or hosts file with syntax 'file:<path>'
   RPORT      25                                                               yes       The target port (TCP)
   THREADS    1                                                                yes       The number of concurrent threa
ds (max one per host)
   UNIXONLY   true                                                             yes       Skip Microsoft bannered server
s when testing unix users
   USER_FILE  /usr/share/metasploit-framework/data/wordlists/unix_users.txt    yes       The file that contains a list
of probable users accounts.
```

```
msf6 auxiliary(scanner/smtp/smtp_enum) > set RHOSTS 192.168.29.89
RHOSTS => 192.168.29.89
msf6 auxiliary(scanner/smtp/smtp_enum) > run

[*] 192.168.29.89:25      - 192.168.29.89:25 Banner: 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
```

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.

```
msf6 > use auxiliary/scanner/smtp/smtp_version
msf6 auxiliary(scanner/smtp/smtp_version) > show options

Module options (auxiliary/scanner/smtp/smtp_version):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<pa
   RPORT    25               yes       The target port (TCP)
   THREADS  1                yes       The number of concurrent threads (max one per host)
```

```
msf6 auxiliary(scanner/smtp/smtp_version) > set RHOSTS 192.168.29.89
RHOSTS ⇒ 192.168.29.89
msf6 auxiliary(scanner/smtp/smtp_version) > set THREADS 254
THREADS ⇒ 254
msf6 auxiliary(scanner/smtp/smtp_version) > run

[+] 192.168.29.89:25      - 192.168.29.89:25 SMTP 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)\x0d\x0a
[*] 192.168.29.89:25      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

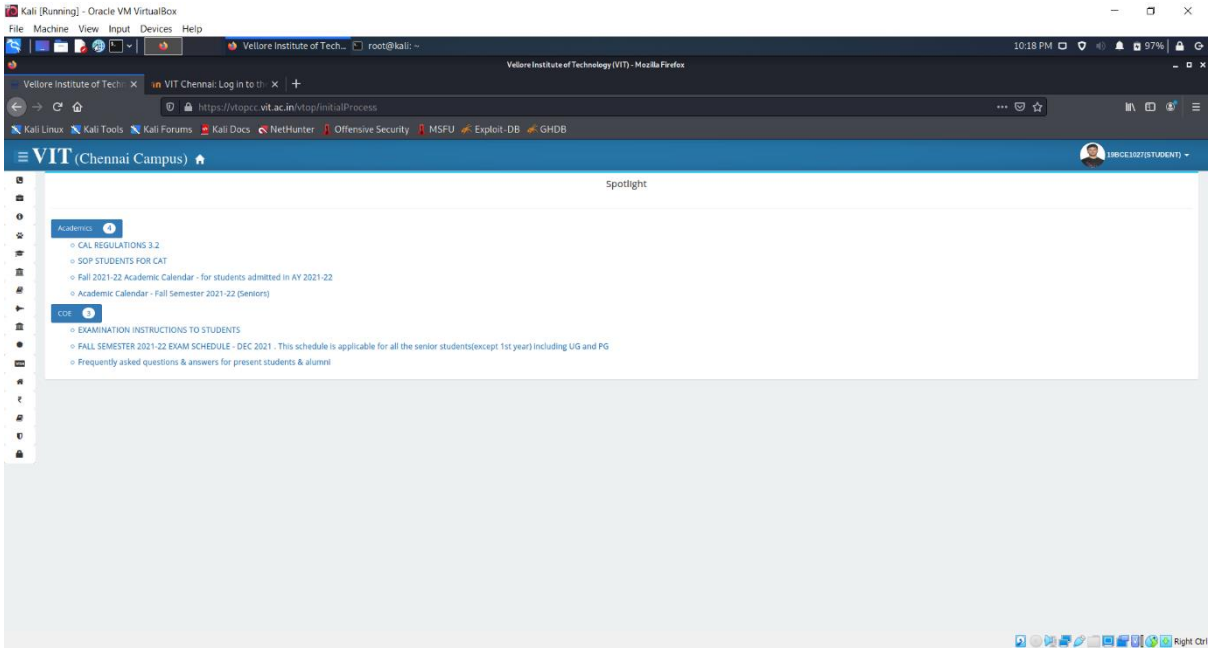
```
msf6 > use auxiliary/scanner/smtp/smtp_enum
msf6 auxiliary(scanner/smtp/smtp_enum) > show options

Module options (auxiliary/scanner/smtp/smtp_enum):

   Name       Current Setting                                                  Required  Description
   ----       ---------------                                                  --------  -----------
   RHOSTS                                                                      yes       The target host(s), range CIDR
 identifier, or hosts file with syntax 'file:<path>'
   RPORT      25                                                               yes       The target port (TCP)
   THREADS    1                                                                yes       The number of concurrent threa
ds (max one per host)
   UNIXONLY   true                                                             yes       Skip Microsoft bannered server
s when testing unix users
   USER_FILE  /usr/share/metasploit-framework/data/wordlists/unix_users.txt    yes       The file that contains a list
of probable users accounts.
```

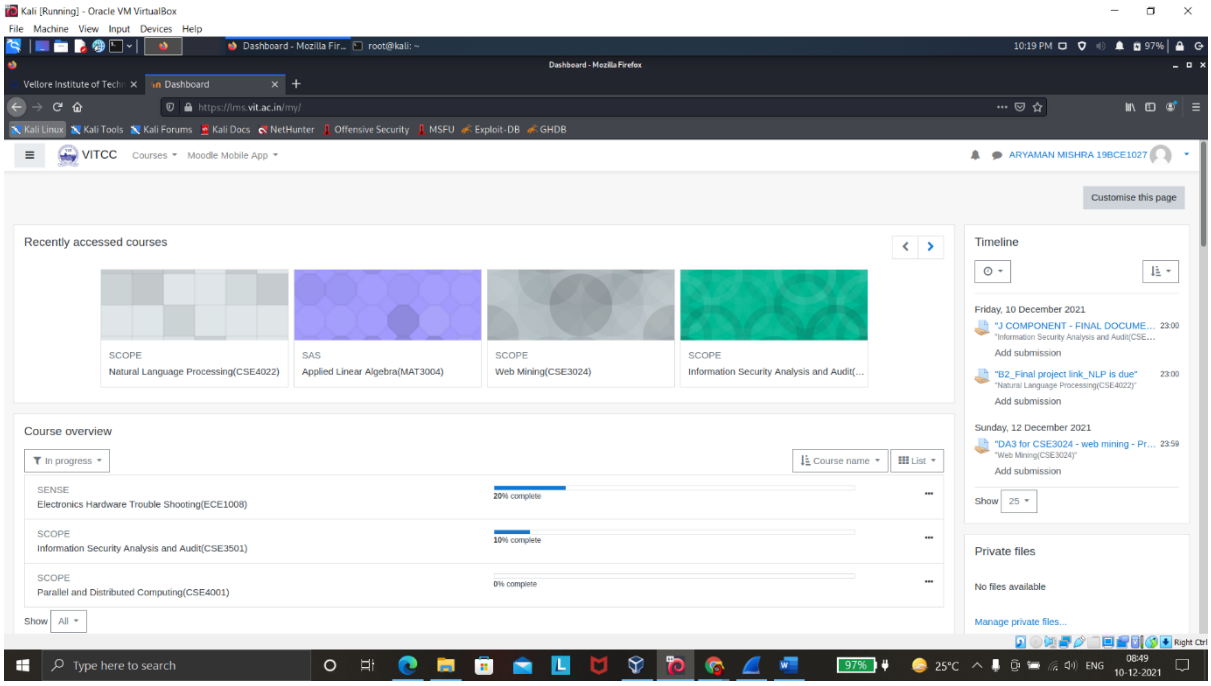
```
msf6 auxiliary(scanner/smtp/smtp_enum) > set RHOSTS 192.168.29.89
RHOSTS ⇒ 192.168.29.89
msf6 auxiliary(scanner/smtp/smtp_enum) > run

[*] 192.168.29.89:25      - 192.168.29.89:25 Banner: 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
```

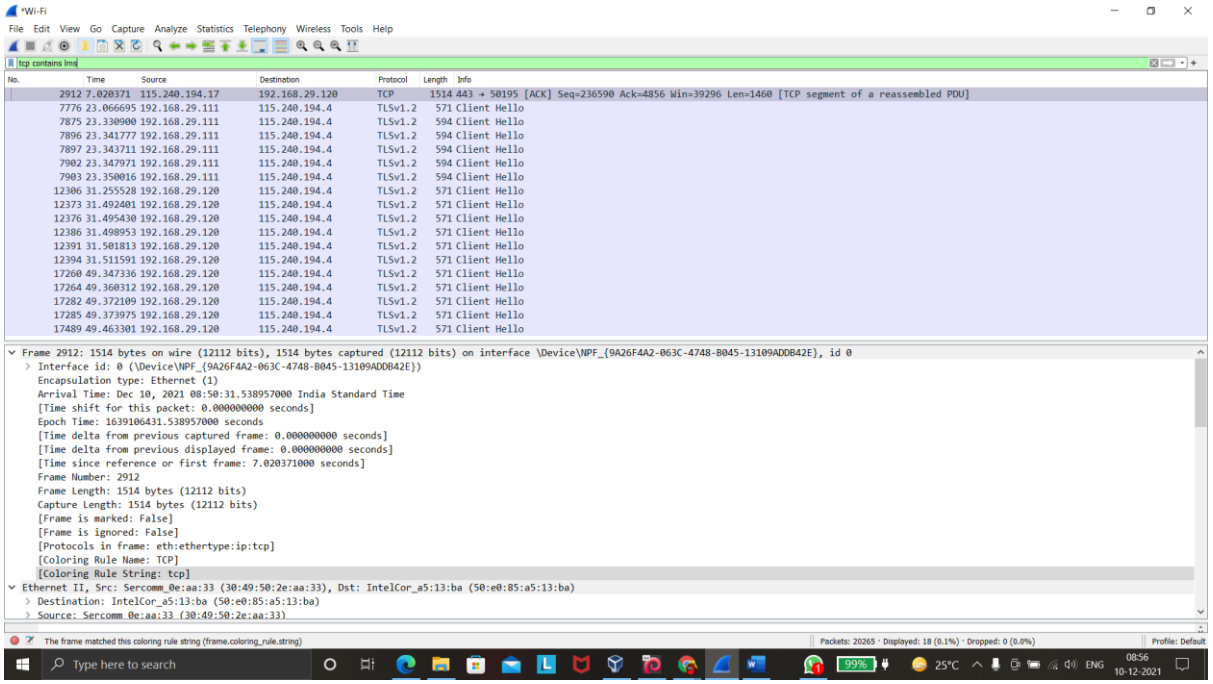
ARYAMAN MISHRA

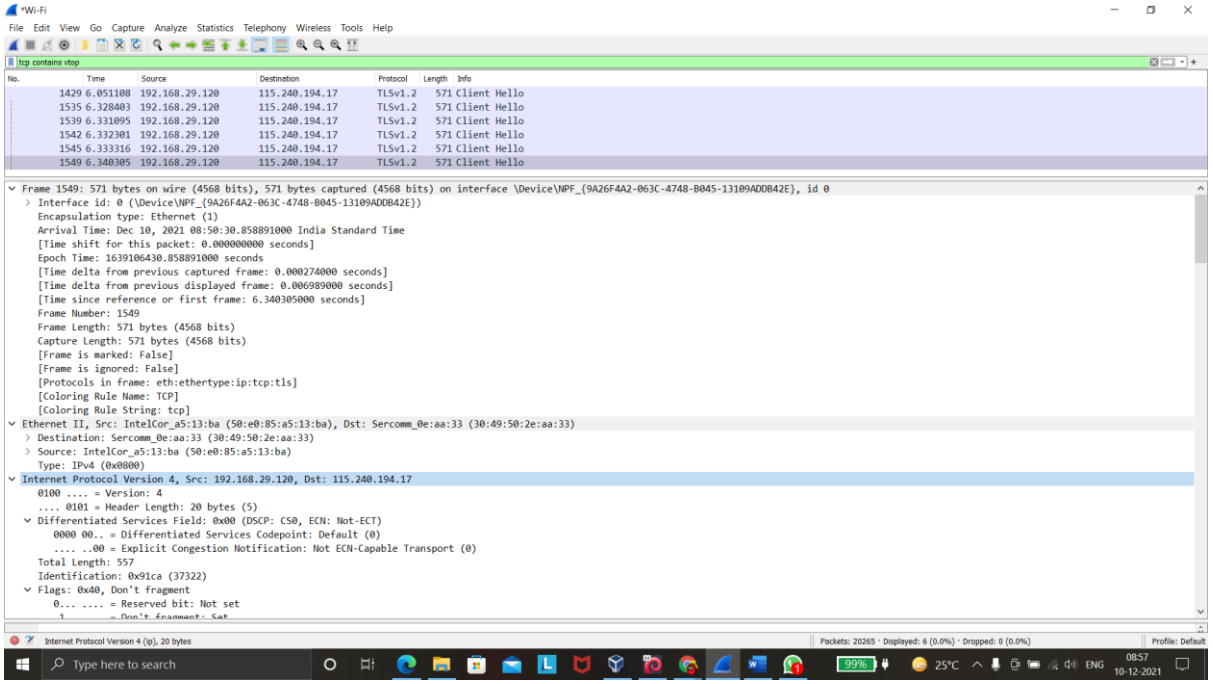
19BCE1027

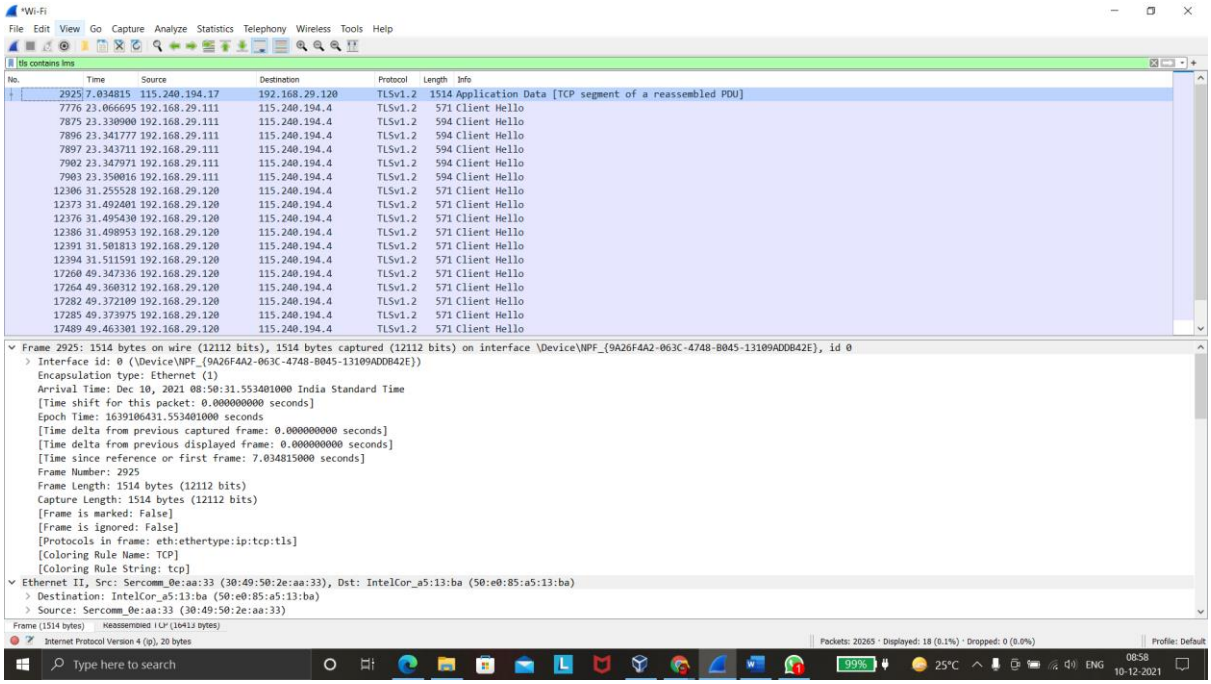
Enter into MOODLE and VTOP applications using the necessary login credentials and identify the user names and passwords of the two applications in the trace files of Wireshark. Display the screenshots that are showing the usernames and passwords in the trace and the respective ASCII codes as well.


Kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Vellore Institute of Technology (VIT) - Mozilla Firefox
https://vtopcc.vit.ac.in/vtop/initialProcess
VIT (Chennai Campus)
Spotlight
Academics
CAL REGULATIONS 3.2
SOP STUDENTS FOR CAT
Fall 2021-22 Academic Calendar - for students admitted in AY 2021-22
Academic Calendar - Fall Semester 2021-22 (Seniors)
COE
EXAMINATION INSTRUCTIONS TO STUDENTS
FALL SEMESTER 2021-22 EXAM SCHEDULE - DEC 2021 . This schedule is applicable for all the senior students(except 1st year) including UG and PG
Frequently asked questions & answers for present students & alumni


Kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Dashboard - Mozilla Firefox
https://lms.vit.ac.in/my/
VITCC
Courses
Moodle Mobile App
ARYAMAN MISHRA 19BCE1027
Customise this page
Recently accessed courses
SCOPE
Natural Language Processing(CSE4022)
SAS
Applied Linear Algebra(MAT3004)
SCOPE
Web Mining(CSE3024)
SCOPE
Information Security Analysis and Audit(...
Timeline
Friday, 10 December 2021
"J COMPONENT - FINAL DOCUME... 23:00
Add submission
"B2_Final project link_NLP is due" 23:00
Add submission
Sunday, 12 December 2021
"DA3 for CSE3024 - web mining - Pr... 23:59
Add submission
Show 25
Course overview
In progress
Course name
List
SENSE
Electronics Hardware Trouble Shooting(ECE1008)
20% complete
SCOPE
Information Security Analysis and Audit(CSE3501)
10% complete
SCOPE
Parallel and Distributed Computing(CSE4001)
0% complete
Show All
Private files
No files available
Manage private files...
Type here to search
25°C
ENG
08:49
10-12-2021


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp contains lms
No. Time Source Destination Protocol Length Info
2912 7.020371 115.240.194.17 192.168.29.120 TCP 1514 443 → 50195 [ACK] Seq=236590 Ack=4856 Win=39296 Len=1460 [TCP segment of a reassembled PDU]
7776 23.066695 192.168.29.111 115.240.194.4 TLSv1.2 571 Client Hello
7875 23.330900 192.168.29.111 115.240.194.4 TLSv1.2 594 Client Hello
7896 23.341777 192.168.29.111 115.240.194.4 TLSv1.2 594 Client Hello
7897 23.343711 192.168.29.111 115.240.194.4 TLSv1.2 594 Client Hello
7902 23.347971 192.168.29.111 115.240.194.4 TLSv1.2 594 Client Hello
7903 23.350016 192.168.29.111 115.240.194.4 TLSv1.2 594 Client Hello
12306 31.255528 192.168.29.120 115.240.194.4 TLSv1.2 571 Client Hello
12373 31.492401 192.168.29.120 115.240.194.4 TLSv1.2 571 Client Hello
12376 31.495430 192.168.29.120 115.240.194.4 TLSv1.2 571 Client Hello
12386 31.498953 192.168.29.120 115.240.194.4 TLSv1.2 571 Client Hello
12391 31.501813 192.168.29.120 115.240.194.4 TLSv1.2 571 Client Hello
12394 31.511591 192.168.29.120 115.240.194.4 TLSv1.2 571 Client Hello
17260 49.347336 192.168.29.120 115.240.194.4 TLSv1.2 571 Client Hello
17264 49.360312 192.168.29.120 115.240.194.4 TLSv1.2 571 Client Hello
17282 49.372109 192.168.29.120 115.240.194.4 TLSv1.2 571 Client Hello
17285 49.373975 192.168.29.120 115.240.194.4 TLSv1.2 571 Client Hello
17489 49.463301 192.168.29.120 115.240.194.4 TLSv1.2 571 Client Hello
Frame 2912: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface \Device\NPF_{9A26F4A2-063C-4748-B045-13109ADDB42E}, id 0
Interface id: 0 (\Device\NPF_{9A26F4A2-063C-4748-B045-13109ADDB42E})
Encapsulation type: Ethernet (1)
Arrival Time: Dec 10, 2021 08:50:31.538957000 India Standard Time
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1639106431.538957000 seconds
[Time delta from previous captured frame: 0.000000000 seconds]
[Time delta from previous displayed frame: 0.000000000 seconds]
[Time since reference or first frame: 7.020371000 seconds]
Frame Number: 2912
Frame Length: 1514 bytes (12112 bits)
Capture Length: 1514 bytes (12112 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: eth:ethertype:ip:tcp]
[Coloring Rule Name: TCP]
[Coloring Rule String: tcp]
Ethernet II, Src: Sercomm_0e:aa:33 (30:49:50:2e:aa:33), Dst: IntelCor_a5:13:ba (50:e0:85:a5:13:ba)
Destination: IntelCor_a5:13:ba (50:e0:85:a5:13:ba)
Source: Sercomm_0e:aa:33 (30:49:50:2e:aa:33)
The frame matched this coloring rule string (frame.coloring_rule.string)
Packets: 20265 · Displayed: 18 (0.1%) · Dropped: 0 (0.0%)
Profile: Default
Type here to search
99%
25°C
ENG
08:56
10-12-2021

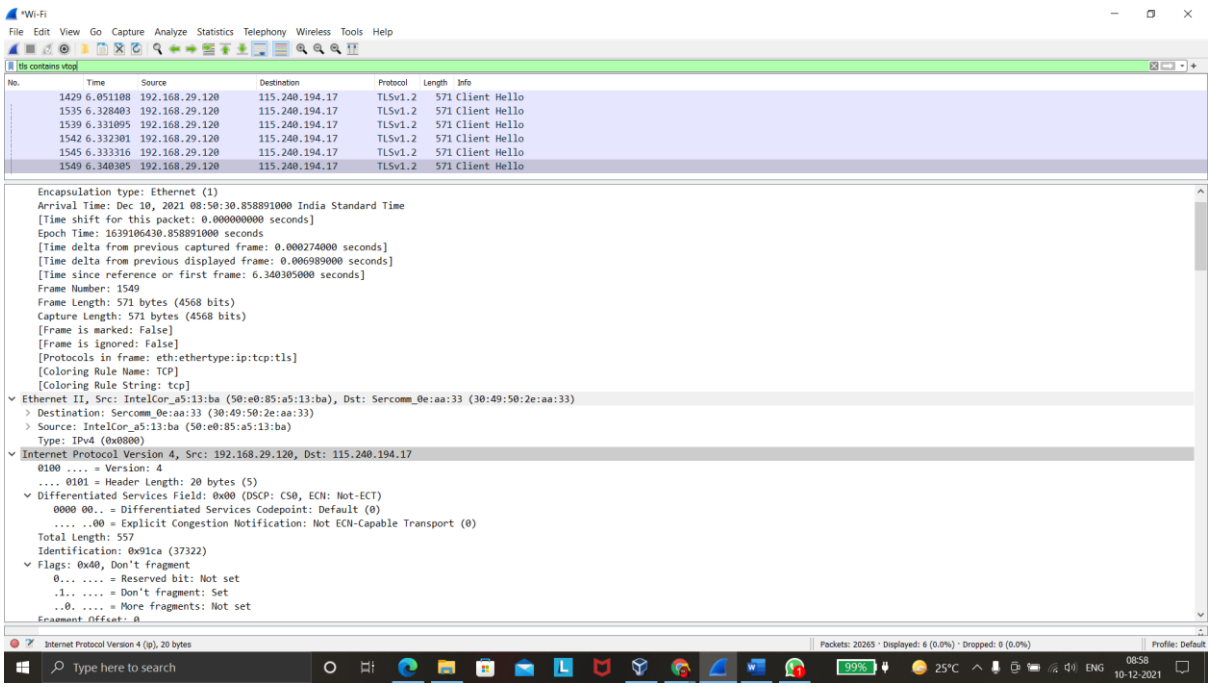

*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp contains vtop
No. Time Source Destination Protocol Length Info
1429 6.051108 192.168.29.120 115.240.194.17 TLSv1.2 571 Client Hello
1535 6.328403 192.168.29.120 115.240.194.17 TLSv1.2 571 Client Hello
1539 6.331095 192.168.29.120 115.240.194.17 TLSv1.2 571 Client Hello
1542 6.332301 192.168.29.120 115.240.194.17 TLSv1.2 571 Client Hello
1545 6.333316 192.168.29.120 115.240.194.17 TLSv1.2 571 Client Hello
1549 6.340305 192.168.29.120 115.240.194.17 TLSv1.2 571 Client Hello
Frame 1549: 571 bytes on wire (4568 bits), 571 bytes captured (4568 bits) on interface \Device\NPF_{9A26F4A2-063C-4748-B045-13109ADDB42E}, id 0
Interface id: 0 (\Device\NPF_{9A26F4A2-063C-4748-B045-13109ADDB42E})
Encapsulation type: Ethernet (1)
Arrival Time: Dec 10, 2021 08:50:30.858891000 India Standard Time
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1639106430.858891000 seconds
[Time delta from previous captured frame: 0.000274000 seconds]
[Time delta from previous displayed frame: 0.006989000 seconds]
[Time since reference or first frame: 6.340305000 seconds]
Frame Number: 1549
Frame Length: 571 bytes (4568 bits)
Capture Length: 571 bytes (4568 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: eth:ethertype:ip:tcp:tls]
[Coloring Rule Name: TCP]
[Coloring Rule String: tcp]
Ethernet II, Src: IntelCor_a5:13:ba (50:e0:85:a5:13:ba), Dst: Sercomm_0e:aa:33 (30:49:50:2e:aa:33)
Destination: Sercomm_0e:aa:33 (30:49:50:2e:aa:33)
Source: IntelCor_a5:13:ba (50:e0:85:a5:13:ba)
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.29.120, Dst: 115.240.194.17
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
0000 00.. = Differentiated Services Codepoint: Default (0)
.... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
Total Length: 557
Identification: 0x91ca (37322)
Flags: 0x40, Don't fragment
0... .... = Reserved bit: Not set
Internet Protocol Version 4 (ip), 20 bytes
Packets: 20265 · Displayed: 6 (0.0%) · Dropped: 0 (0.0%)
Profile: Default
Type here to search
99%
25°C
ENG
08:57
10-12-2021



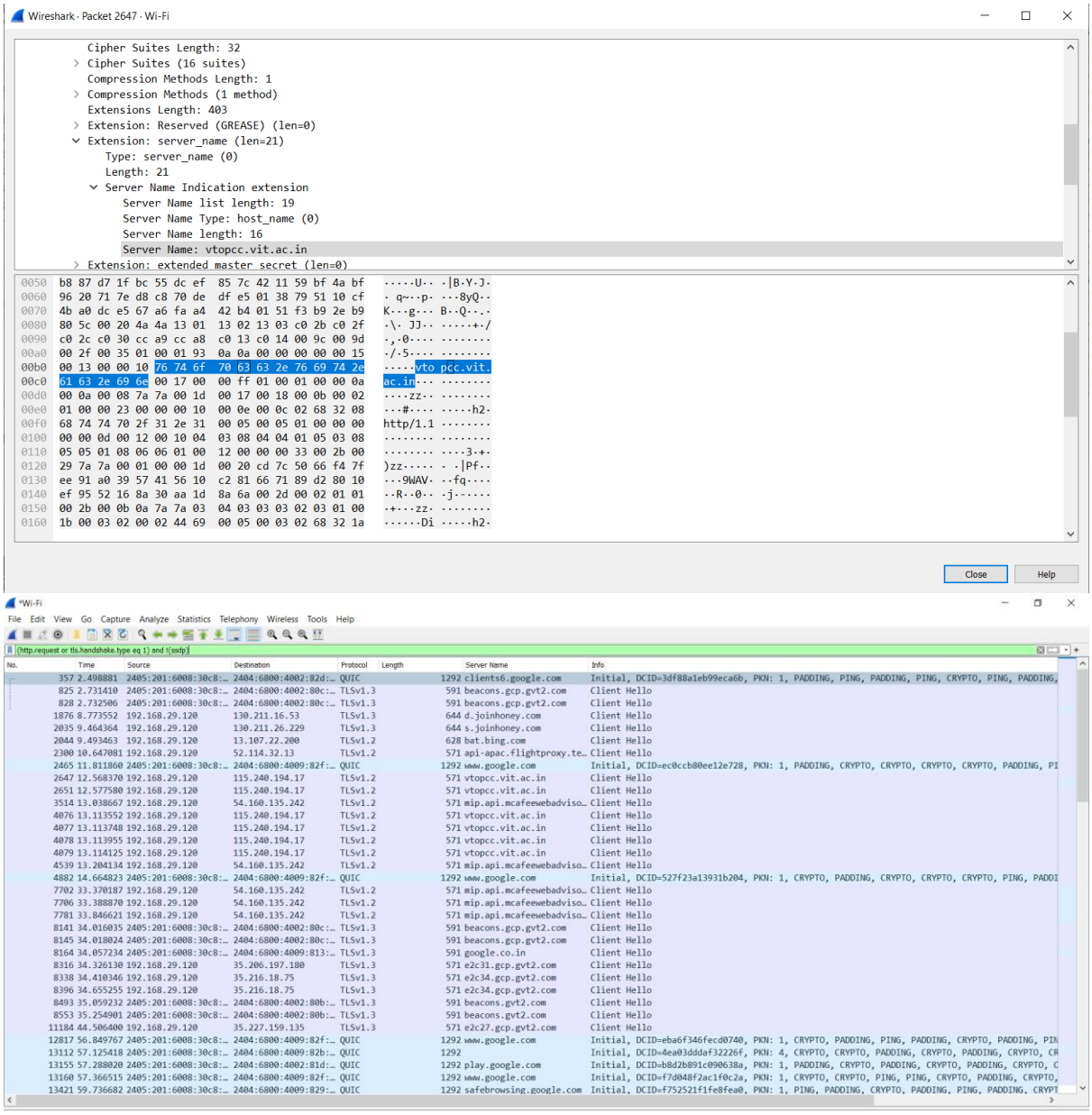
Open ***Wireshark-tutorial-on-decrypting-HTTPS-SSL-TLS-traffic.pcap*** in Wireshark. Use a basic web filter as described in this previous tutorial about Wireshark filters. Our basic filter for Wireshark 3.x is:

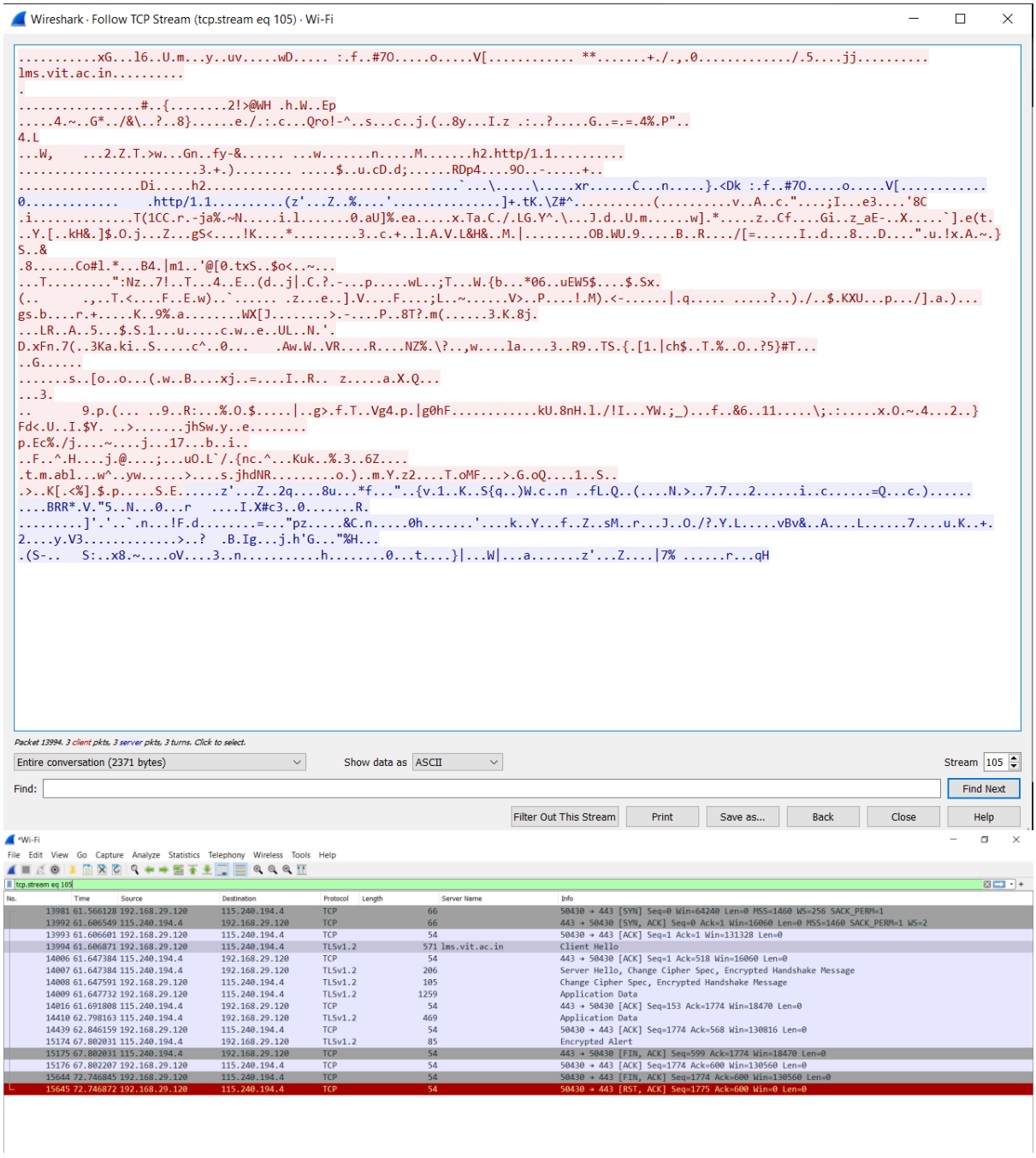
```
(http.request or tls.handshake.type eq 1) and !(ssdp)
```

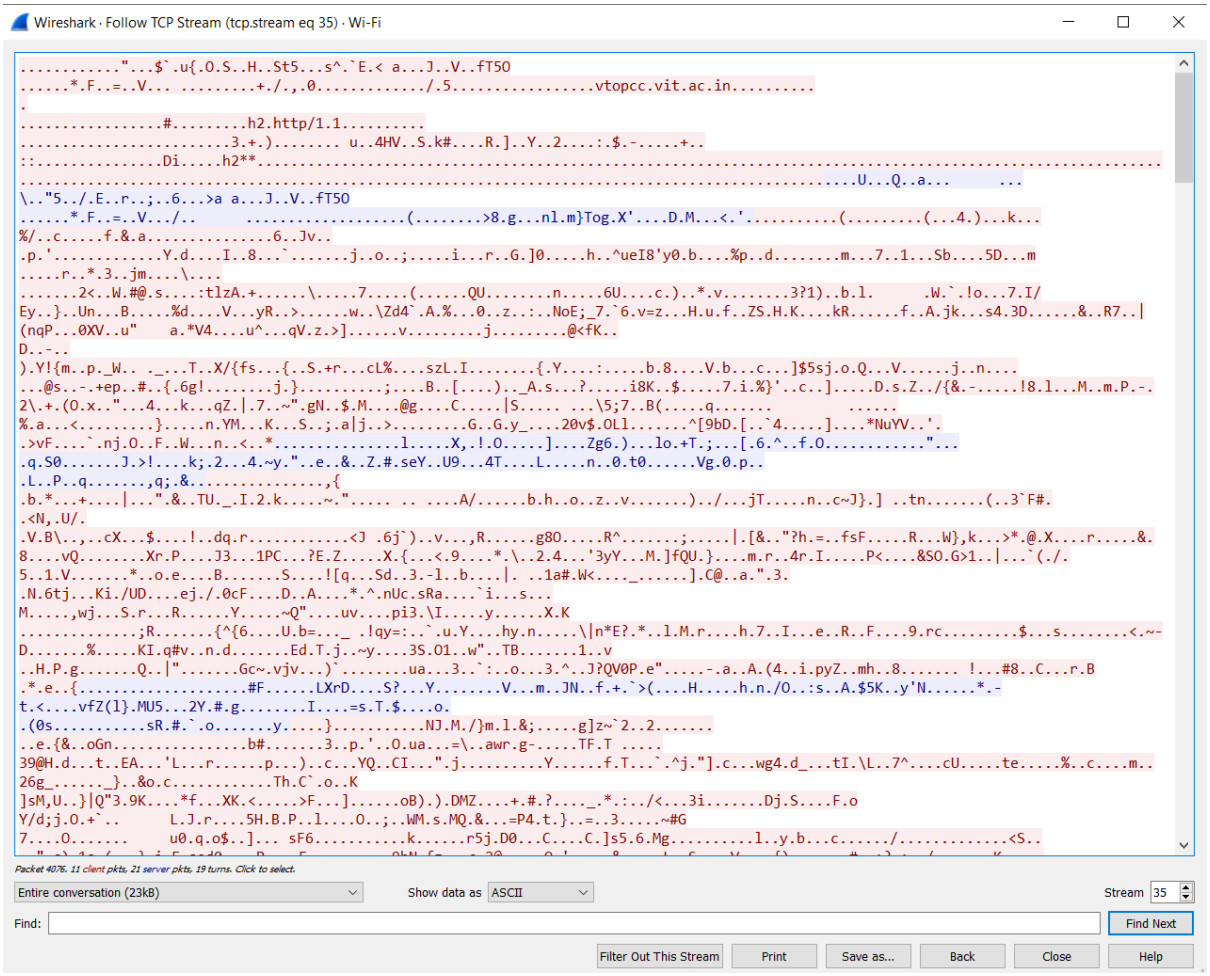
Open ***Wireshark-tutorial-on-decrypting-HTTPS-SSL-TLS-traffic.pcap*** in Wireshark. Then use the menu path ***Edit --> Preferences*** to bring up the Preferences Menu



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




Wireshark · Follow TCP Stream (tcp.stream eq 105) · Wi-Fi
lms.vit.ac.in
Packet 13994. 3 client pkts, 3 server pkts, 3 turns. Click to select.
Entire conversation (2371 bytes)
Show data as ASCII
Stream 105
Find:
Find Next
Filter Out This Stream
Print
Save as...
Back
Close
Help
*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp.stream eq 105
No. Time Source Destination Protocol Length Server Name Info
13981 61.566128 192.168.29.120 115.240.194.4 TCP 66 50430 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
13992 61.606549 115.240.194.4 192.168.29.120 TCP 66 443 → 50430 [SYN, ACK] Seq=0 Ack=1 Win=16060 Len=0 MSS=1460 SACK_PERM=1 WS=2
13993 61.606601 192.168.29.120 115.240.194.4 TCP 54 50430 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0
13994 61.606871 192.168.29.120 115.240.194.4 TLSv1.2 571 lms.vit.ac.in Client Hello
14006 61.647384 115.240.194.4 192.168.29.120 TCP 54 443 → 50430 [ACK] Seq=1 Ack=518 Win=16060 Len=0
14007 61.647384 115.240.194.4 192.168.29.120 TLSv1.2 206 Server Hello, Change Cipher Spec, Encrypted Handshake Message
14008 61.647591 192.168.29.120 115.240.194.4 TLSv1.2 105 Change Cipher Spec, Encrypted Handshake Message
14009 61.647732 192.168.29.120 115.240.194.4 TLSv1.2 1259 Application Data
14016 61.691808 115.240.194.4 192.168.29.120 TCP 54 443 → 50430 [ACK] Seq=153 Ack=1774 Win=18470 Len=0
14410 62.798163 115.240.194.4 192.168.29.120 TLSv1.2 469 Application Data
14439 62.846159 192.168.29.120 115.240.194.4 TCP 54 50430 → 443 [ACK] Seq=1774 Ack=568 Win=130816 Len=0
15174 67.802031 115.240.194.4 192.168.29.120 TLSv1.2 85 Encrypted Alert
15175 67.802031 115.240.194.4 192.168.29.120 TCP 54 443 → 50430 [FIN, ACK] Seq=599 Ack=1774 Win=18470 Len=0
15176 67.802207 192.168.29.120 115.240.194.4 TCP 54 50430 → 443 [ACK] Seq=1774 Ack=600 Win=130560 Len=0
15644 72.746845 192.168.29.120 115.240.194.4 TCP 54 50430 → 443 [FIN, ACK] Seq=1774 Ack=600 Win=130560 Len=0
15645 72.746872 192.168.29.120 115.240.194.4 TCP 54 50430 → 443 [RST, ACK] Seq=1775 Ack=600 Win=0 Len=0



SHA Values for Passwords are

`31cf42b2a7c5c558f44cfc67684cc344c17d4946d3a1e0b2cecb8eb58173cb2f`

`29!-!.`

0d5ad14787be9df27b50dd7b99279b1b64c8bffd752e19a485d189467191bdca

`Y|!B`