Scanning, Encryption and SQL Injection
Optional Lab

This activity will give you an opportunity to explore a number of methods:

- Packet capturing
- Network scanning
- SQL Injection

This activity:

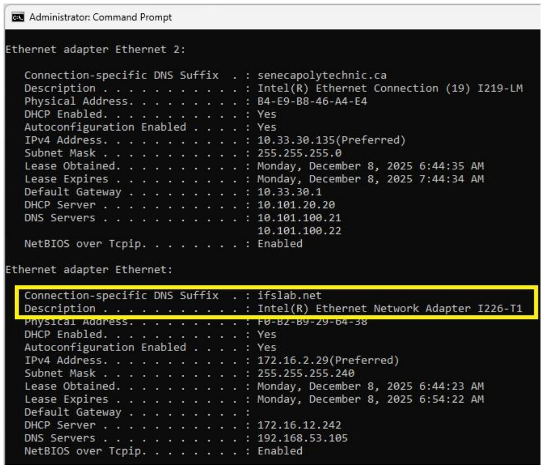
- Should be done on the Lab Workstation.
- Requires access to the Security Lab network: 172.16.11.0/26
- Best done using the Windows VM available on “\\mydrive\courses\SPR100\Win10”

Setup

Your Lab Workstation is already connected to the Security Lab network. Below are the steps to set up (a) a Security Network connection between VMware and, (b) the Lab Workstation, and VMware and your VM.

Steps:

Step 1: *On the Lab Workstation (host) do* ”ipconfig /all” to identify the network card that has the Security Lab network. It will likely start with “Intel (R)..” (see Figure 1)



```
Ethernet adapter Ethernet 2:

   Connection-specific DNS Suffix  . : senecapolytechnic.ca
   Description . . . . . . . . . . . : Intel(R) Ethernet Connection (19) I219-LM
   Physical Address. . . . . . . . . : B4-E9-B8-46-A4-E4
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv4 Address. . . . . . . . . . . : 10.33.30.135(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Monday, December 8, 2025 6:44:35 AM
   Lease Expires . . . . . . . . . . : Monday, December 8, 2025 7:44:34 AM
   Default Gateway . . . . . . . . . : 10.33.30.1
   DHCP Server . . . . . . . . . . . : 10.101.20.20
   DNS Servers . . . . . . . . . . . : 10.101.100.21
                                       10.101.100.22
   NetBIOS over Tcpip. . . . . . . . : Enabled

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : ifslab.net
   Description . . . . . . . . . . . : Intel(R) Ethernet Network Adapter I226-T1
   Physical Address. . . . . . . . . : F0-B2-B9-29-64-38
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv4 Address. . . . . . . . . . . : 172.16.2.29(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.240
   Lease Obtained. . . . . . . . . . : Monday, December 8, 2025 6:44:23 AM
   Lease Expires . . . . . . . . . . : Monday, December 8, 2025 6:54:22 AM
   Default Gateway . . . . . . . . . :
   DHCP Server . . . . . . . . . . . : 172.16.12.242
   DNS Servers . . . . . . . . . . . : 192.168.53.105
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

Figure 1: ”ipconfig /all” shows the interfaces. Identify the interface card for “ifslab.net”

Step 2: *On VMware Workstation Pro*, start the *Virtual Network Editor* (In the Edit menu – see Figure 2), select a current network, select *Bridged*, then select the network card you identified in Step 1. (see Figure 3)

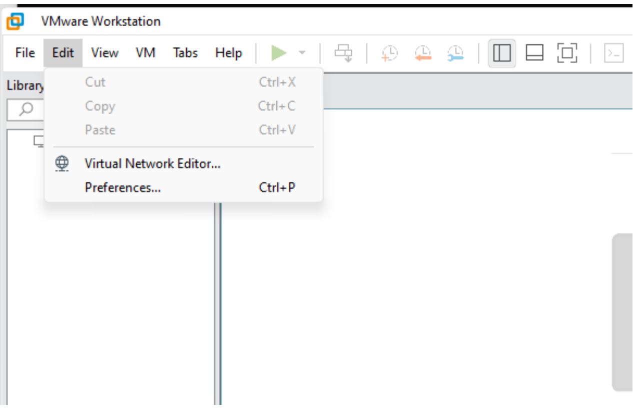
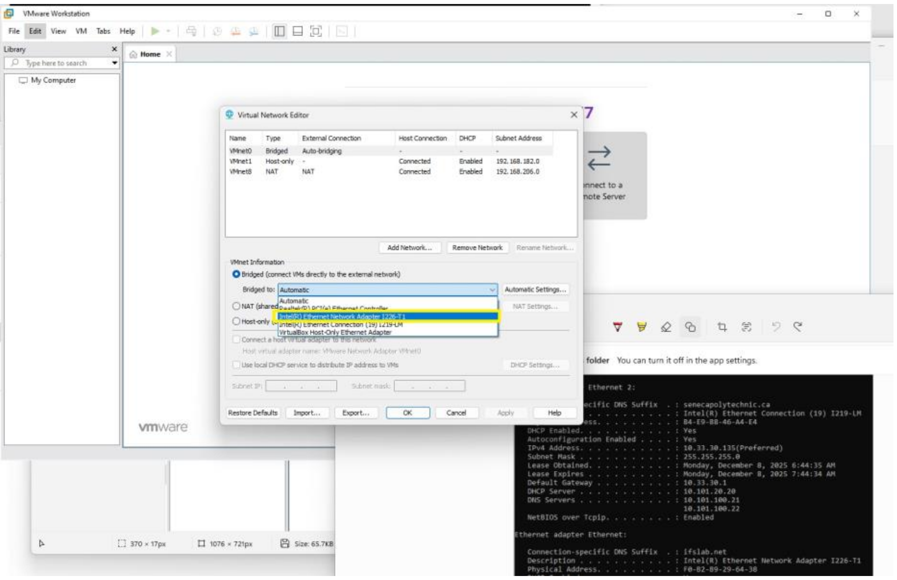


Figure 2: Virtual Network Editor in Edit Menu



Name	Type	External Connection	Host Connection	DHCP	Subnet Address
VMnet0	Bridged	Auto-bridging	-	-	-
VMnet1	Host-only	-	Connected	Enabled	192.168.182.0
VMnet8	NAT	NAT	Connected	Enabled	192.168.206.0

VMnet Information

● Bridged (connect VMs directly to the external network)

Bridged to: Automatic

- Automatic
- Realtek(R) PCI(e) Ethernet Controller
- Intel(R) Ethernet Network Adapter I226-T1
- Intel(R) Ethernet Connection (19) I219-LM
- VirtualBox Host-Only Ethernet Adapter

○ NAT (shared

○ Host-only

□ Connect a host virtual adapter to this network

Host virtual adapter name: VMware Network Adapter VMnet0

□ Use local DHCP service to distribute IP address to VMs

Subnet IP: . . .   Subnet mask: . . .

folder You can turn it off in the app settings.

```
 Ethernet 2:

ecific DNS Suffix  . : senecapolytechnic.ca
 . . . . . . . . . . : Intel(R) Ethernet Connection (19) I219-LM
ess. . . . . . . . . : B4-E9-88-46-A4-E4
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv4 Address. . . . . . . . . . . : 10.33.30.135(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Monday, December 8, 2025 6:44:35 AM
   Lease Expires . . . . . . . . . . : Monday, December 8, 2025 7:44:34 AM
   Default Gateway . . . . . . . . . : 10.33.30.1
   DHCP Server . . . . . . . . . . . : 10.101.20.20
   DNS Servers . . . . . . . . . . . : 10.101.100.21
                                       10.101.100.22
   NetBIOS over Tcpip. . . . . . . . : Enabled

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : ifslab.net
   Description . . . . . . . . . . . : Intel(R) Ethernet Network Adapter I226-T1
   Physical Address. . . . . . . . . : F0-82-89-29-64-38
```

Figure 3: Select the interface you identified in Step 1.

Step 3: In your VM's configuration, select the virtual network that you modified in Step 2.

Verification:

1. Start your VM and login.
2. From the *Command Prompt* do "ipconfig /all" and check that you have an IP starting with "172.16.X.X"
3. Check connectivity by doing: 'ping **172.16.11.65'**.

Network Scanning

Single Host

1. Start Zen-map – nmap’s GUI
2. Enter the IP “172.16.11.65” and use the *Ping scan* setting, and look at the results you get back.
3. Now try different scan setting: “Quick” scan and “Intense” setting, in each case reviewing the results you get back and the different types of details they return.

Network Enumeration

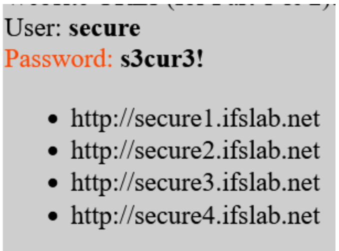
If we are to learn about what is happening on our network, rather than just one computer, we should be familiar with the basic protocols, and service identification.

Using Zen-map, scan the ***Security Lab network: 172.16.11.0/26*** and discover hosts that are listening. This will scan a range of IPs. The number of IPs is determined by the /N. Note, while we give it a range of IPs, this does not necessarily mean there is a host at each IP. In fact, that is part of the purpose of the scan, to determine this.

Cryptography (Web)

Here we are going to do some packet capturing of two types of web traffic, unencrypted and encrypted. For both we will do the following steps:

1. Using a web-browser, going to one of the sites listed in Figure 4.
2. Start *Wireshark* on our VM.
3. Logging into the appropriate account on the website using the credentials given in Figure 4.
4. Logging out.
5. Stopping Wireshark and reviewing the results.



User: **secure**
Password: **s3cur3!**

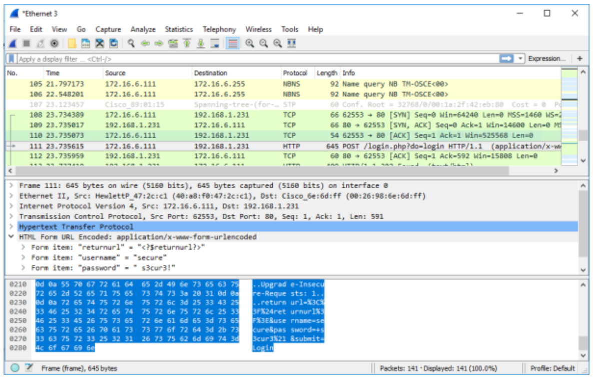
- http://secure1.ifslab.net
- http://secure2.ifslab.net
- http://secure3.ifslab.net
- http://secure4.ifslab.net

Figure 4: Website URLs

Unsecured Web Traffic: HTTP Communication

1. Navigate to website Figure 4.
2. On your Win 10 VM start the application *Wireshark.*
3. Click the **Start** button on *Wireshark* to begin capturing packets.
   - <u>Work as quickly as you can to minimize the number of packets captured</u>.
   - If you see a message saying: **Save capture file before starting a new capture?** Click **Continue Without Saving**.
4. Login to the website (use the ‘login’ link, not the ‘secure login’ link) using the credentials *provided Figure 4*.
5. Click the **Stop** button on *Wireshark*.
6. Analyze the packets and look for the following line in the capture window.
   - Click on the **HTTP** protocol with the **POST** line, as in Figure 5, and the decoded message should appear in the lower window.

- You should see your credentials in plain text.


*Ethernet 3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Expression…

```
No.   Time       Source          Destination          Protocol Length Info
  105 21.797173  172.16.6.111    172.16.6.255         NBNS     92 Name query NB TM-OSCE<00>
  106 22.548201  172.16.6.111    172.16.6.255         NBNS     92 Name query NB TM-OSCE<00>
  107 23.123457  Cisco_89:01:15  Spanning-tree-(for-… STP      60 Conf. Root = 32768/0/00:1a:2f:42:eb:80  Cost = 0  Po
  108 23.734389  172.16.6.111    192.168.1.231        TCP      66 62553 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=2
  109 23.735017  192.168.1.231   172.16.6.111         TCP      66 80 → 62553 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MS
  110 23.735073  172.16.6.111    192.168.1.231        TCP      54 62553 → 80 [ACK] Seq=1 Ack=1 Win=525568 Len=0
  111 23.735615  172.16.6.111    192.168.1.231        HTTP    645 POST /login.php?do=login HTTP/1.1  (application/x-ww
  112 23.735959  192.168.1.231   172.16.6.111         TCP      60 80 → 62553 [ACK] Seq=1 Ack=592 Win=15808 Len=0
```

```
> Frame 111: 645 bytes on wire (5160 bits), 645 bytes captured (5160 bits) on interface 0
> Ethernet II, Src: HewlettP_47:2c:c1 (40:a8:f0:47:2c:c1), Dst: Cisco_6e:6d:ff (00:26:98:6e:6d:ff)
> Internet Protocol Version 4, Src: 172.16.6.111, Dst: 192.168.1.231
> Transmission Control Protocol, Src Port: 62553, Dst Port: 80, Seq: 1, Ack: 1, Len: 591
> Hypertext Transfer Protocol
v HTML Form URL Encoded: application/x-www-form-urlencoded
  > Form item: "returnurl" = "<?$returnurl?>"
  > Form item: "username" = "secure"
  > Form item: "password" = " s3cur3!"
```

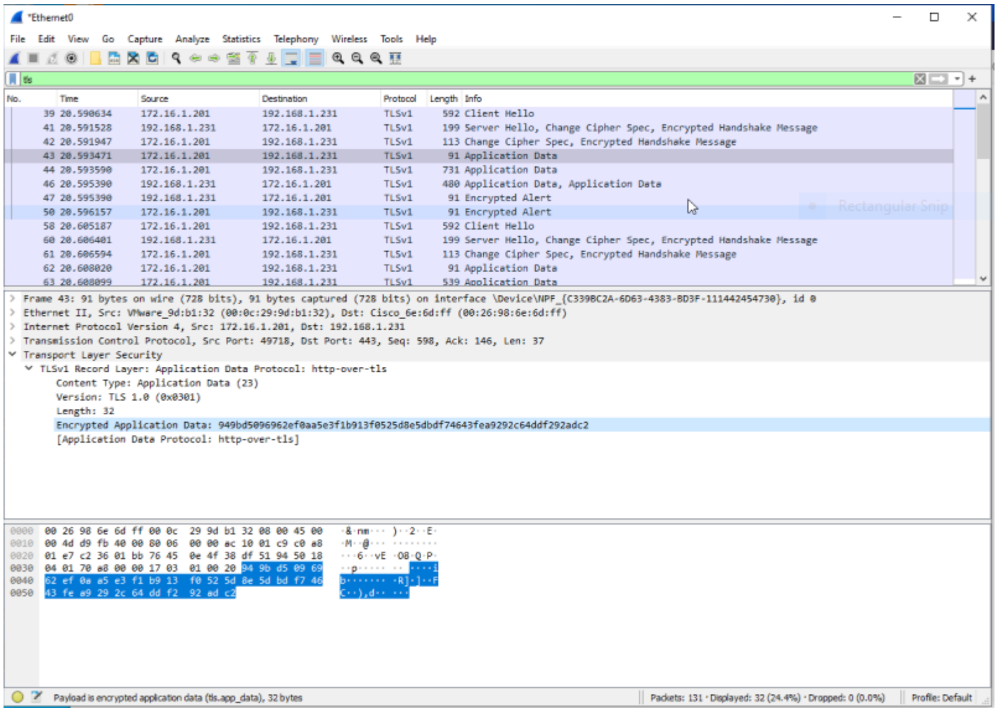
```
0210  0d 0a 55 70 67 72 61 64  65 2d 49 6e 73 65 63 75   ..Upgrad e-Insecu
0220  72 65 2d 52 65 71 75 65  73 74 73 3a 20 31 0d 0a   re-Reque sts: 1..
0230  0d 0a 72 65 74 75 72 6e  75 72 6c 3d 25 33 43 25   ..return url=%3C%
0240  33 46 25 32 34 72 65 74  75 72 6e 75 72 6c 25 33   3F%24ret urnurl%3
0250  46 25 33 45 26 75 73 65  72 6e 61 6d 65 3d 73 65   F%3E&use rname=se
0260  63 75 72 65 26 70 61 73  73 77 6f 72 64 3d 2b 73   cure&pas sword=+s
0270  33 63 75 72 33 25 32 31  26 73 75 62 6d 69 74 3d   3cur3%21 &submit=
0280  4c 6f 67 69 6e                                     Login
```

Frame (frame), 645 bytes | Packets: 141 · Displayed: 141 (100.0%) | Profile: Default

Figure 5: Captured HTTP session

Encrypted Web Traffic: TLS/SSL Communication

1. Open *Wireshark* and begin a packet capture session (you did this above).
2. Login to the website using the secure login link using the credentials *Figure 4*.
3. Stop the capture and examine the captured packets.
    - Look at the packets that appear below "**Client Hello**" and "**Server Hello**".
    - Find a packet labeled "**TLSx.x Application Data**", like packet 138 in Figure 6 below, and click on it in the top pane to select it. Details about the packet will appear in the middle pane.
    - Click the **>** sign to expand **Transport Security Layer**.
    - Expand the layer inside (labeled "**TLSvx.x Record Layer**" so that *the* ***Encrypted Application Data*** *is visible*, as shown at the bottom of the image below on this page. Your user name and password are concealed in that encrypted data.
    - Even though the packet sniffer can see the data go by, it cannot be read. This is how TLS/SSL protects you--all web logins should use TLS/SSL.


*Ethernet0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tls
Rectangular Snip

```
No.  Time       Source         Destination    Protocol Length Info
  39 20.590634  172.16.1.201   192.168.1.231  TLSv1    592 Client Hello
  41 20.591528  192.168.1.231  172.16.1.201   TLSv1    199 Server Hello, Change Cipher Spec, Encrypted Handshake Message
  42 20.591947  172.16.1.201   192.168.1.231  TLSv1    113 Change Cipher Spec, Encrypted Handshake Message
  43 20.593471  172.16.1.201   192.168.1.231  TLSv1     91 Application Data
  44 20.593590  172.16.1.201   192.168.1.231  TLSv1    731 Application Data
  46 20.595390  192.168.1.231  172.16.1.201   TLSv1    480 Application Data, Application Data
  47 20.595390  192.168.1.231  172.16.1.201   TLSv1     91 Encrypted Alert
  50 20.596157  172.16.1.201   192.168.1.231  TLSv1     91 Encrypted Alert
  58 20.605187  172.16.1.201   192.168.1.231  TLSv1    592 Client Hello
  60 20.606401  192.168.1.231  172.16.1.201   TLSv1    199 Server Hello, Change Cipher Spec, Encrypted Handshake Message
  61 20.606594  172.16.1.201   192.168.1.231  TLSv1    113 Change Cipher Spec, Encrypted Handshake Message
  62 20.608020  172.16.1.201   192.168.1.231  TLSv1     91 Application Data
  63 20.608099  172.16.1.201   192.168.1.231  TLSv1    539 Application Data
```

```
> Frame 43: 91 bytes on wire (728 bits), 91 bytes captured (728 bits) on interface \Device\NPF_{C339BC2A-6D63-4383-BD3F-111442454730}, id 0
> Ethernet II, Src: VMware_9d:b1:32 (00:0c:29:9d:b1:32), Dst: Cisco_6e:6d:ff (00:26:98:6e:6d:ff)
> Internet Protocol Version 4, Src: 172.16.1.201, Dst: 192.168.1.231
> Transmission Control Protocol, Src Port: 49718, Dst Port: 443, Seq: 598, Ack: 146, Len: 37
v Transport Layer Security
  v TLSv1 Record Layer: Application Data Protocol: http-over-tls
      Content Type: Application Data (23)
      Version: TLS 1.0 (0x0301)
      Length: 32
      Encrypted Application Data: 949bd5096962ef0aa5e3f1b913f0525d8e5dbdf74643fea9292c64ddf292adc2
      [Application Data Protocol: http-over-tls]
```

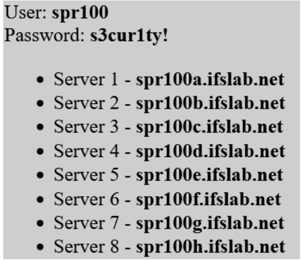
```
0000  00 26 98 6e 6d ff 00 0c  29 9d b1 32 08 00 45 00   ·&·nm··· )··2··E·
0010  00 4d d9 fb 40 00 80 06  00 00 ac 10 01 c9 c0 a8   ·M··@··· ········
0020  01 e7 c2 36 01 bb 76 45  0e 4f 38 df 51 94 50 18   ···6··vE ·O8·Q·P·
0030  04 01 70 a8 00 00 17 03  01 00 20 94 9b d5 09 69   ··p····· ·· ····i
0040  62 ef 0a a5 e3 f1 b9 13  f0 52 5d 8e 5d bd f7 46   b······· ·R]·]··F
0050  43 fe a9 29 2c 64 dd f2  92 ad c2                  C··),d·· ···
```

Payload is encrypted application data (tls.app_data), 32 bytes | Packets: 131 · Displayed: 32 (24.4%) · Dropped: 0 (0.0%) | Profile: Default

Figure 6: Captured TLS session

Cryptography (SSH)

Here we will be doing something similar to the previous exercise, though in this case we will be looking at SSH traffic.

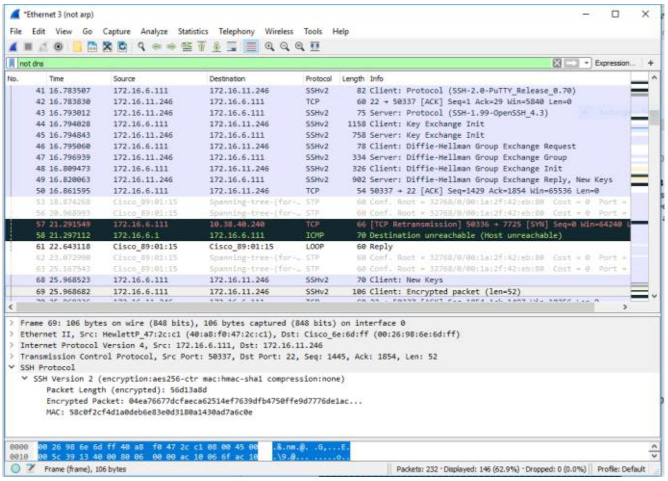


User: **spr100**
Password: **s3cur1ty!**

- Server 1 - **spr100a.ifslab.net**
- Server 2 - **spr100b.ifslab.net**
- Server 3 - **spr100c.ifslab.net**
- Server 4 - **spr100d.ifslab.net**
- Server 5 - **spr100e.ifslab.net**
- Server 6 - **spr100f.ifslab.net**
- Server 7 - **spr100g.ifslab.net**
- Server 8 - **spr100h.ifslab.net**

Figure 7: SSH Servers

1. Begin another screen capture in *Wireshark*.
   - You may want to set the **Filter Options** to '*not arp and not dns' to reduce the amount of data generated.*
2. Click the '**Start'** button in *Wireshark*.
3. Start *PuTTY – this is installed on the Win10 VM*.
4. Enter one of the server names given in *Figure 7*.
5. Click the '**Open'** button on Putty Configuration.
   - You get a Security Alert window. Click' **Yes'** to add the server's public key to the registry.
6. Login using the credentials *provided in class*.
7. At the command prompt type the **ls** command.
8. Logout of the server.
9. Click '**Stop'** capture in *Wireshark* .
10. Scroll to the beginning of the capture window and review the communication between the client and server (see Figure 8).
    - Notice that SSHvx uses the Diffie-Hellman protocol to exchange keys.
    - Notice that a key and handshake are agreed to by the client and server before the exchange of data.
    - Notice that each packet is encrypted within an encrypted tunnel which protects the login information from replay attacks.



Filter: not dns

```
No.   Time       Source          Destination          Protocol Length Info
   41 16.783507  172.16.6.111    172.16.11.246        SSHv2      82 Client: Protocol (SSH-2.0-PuTTY_Release_0.70)
   42 16.783830  172.16.11.246   172.16.6.111         TCP        60 22 → 50337 [ACK] Seq=1 Ack=29 Win=5840 Len=0
   43 16.793012  172.16.11.246   172.16.6.111         SSHv2      75 Server: Protocol (SSH-1.99-OpenSSH_4.3)
   44 16.794028  172.16.6.111    172.16.11.246        SSHv2    1158 Client: Key Exchange Init
   45 16.794843  172.16.11.246   172.16.6.111         SSHv2     758 Server: Key Exchange Init
   46 16.795060  172.16.6.111    172.16.11.246        SSHv2      78 Client: Diffie-Hellman Group Exchange Request
   47 16.796939  172.16.11.246   172.16.6.111         SSHv2     334 Server: Diffie-Hellman Group Exchange Group
   48 16.809473  172.16.6.111    172.16.11.246        SSHv2     326 Client: Diffie-Hellman Group Exchange Init
   49 16.820063  172.16.11.246   172.16.6.111         SSHv2     902 Server: Diffie-Hellman Group Exchange Reply, New Keys
   50 16.861595  172.16.6.111    172.16.11.246        TCP        54 50337 → 22 [ACK] Seq=1429 Ack=1854 Win=65536 Len=0
   53 18.874268  Cisco_89:01:15  Spanning-tree-(for-… STP        60 Conf. Root = 32768/0/00:1a:2f:42:eb:80  Cost = 0  Port =
   56 20.968983  Cisco_89:01:15  Spanning-tree-(for-… STP        60 Conf. Root = 32768/0/00:1a:2f:42:eb:80  Cost = 0  Port =
   57 21.291549  172.16.6.111    10.38.40.240         TCP        66 [TCP Retransmission] 50336 → 7725 [SYN] Seq=0 Win=64240 L
   58 21.297112  172.16.6.1      172.16.6.111         ICMP       70 Destination unreachable (Host unreachable)
   61 22.643118  Cisco_89:01:15  Cisco_89:01:15       LOOP       60 Reply
   62 23.072990  Cisco_89:01:15  Spanning-tree-(for-… STP        60 Conf. Root = 32768/0/00:1a:2f:42:eb:80  Cost = 0  Port =
   63 25.167543  Cisco_89:01:15  Spanning-tree-(for-… STP        60 Conf. Root = 32768/0/00:1a:2f:42:eb:80  Cost = 0  Port =
   68 25.968523  172.16.6.111    172.16.11.246        SSHv2      70 Client: New Keys
   69 25.968682  172.16.6.111    172.16.11.246        SSHv2     106 Client: Encrypted packet (len=52)
```

```
> Frame 69: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface 0
> Ethernet II, Src: HewlettP_47:2c:c1 (40:a8:f0:47:2c:c1), Dst: Cisco_6e:6d:ff (00:26:98:6e:6d:ff)
> Internet Protocol Version 4, Src: 172.16.6.111, Dst: 172.16.11.246
> Transmission Control Protocol, Src Port: 50337, Dst Port: 22, Seq: 1445, Ack: 1854, Len: 52
v SSH Protocol
  v SSH Version 2 (encryption:aes256-ctr mac:hmac-sha1 compression:none)
      Packet Length (encrypted): 56d13a8d
      Encrypted Packet: 04ea76677dcfaeca62514ef7639dfb4750ffe9d7776de1ac...
      MAC: 58c0f2cf4d1a0deb6e83e0d3180a1430ad7a6c0e
```

```
0000  00 26 98 6e 6d ff 40 a8  f0 47 2c c1 08 00 45 00   .&.nm.@. .G,...E.
0010  00 5c 39 13 40 00 80 06  00 00 ac 10 06 6f ac 10   .\9.@... .....o..
```

```
Frame (frame), 106 bytes    Packets: 232 · Displayed: 146 (62.9%) · Dropped: 0 (0.0%)    Profile: Default
```

Figure 8: Captured SSH Session

SQL Injection

Here you will be looking at SQL injection attacks. OWASP lists Injection attacks in their Top 10 for 2021 (https://owasp.org/Top10/). This includes SQL Injection. This was true for 2010, 2013 and 2021. Here are some references for SQL injection:

- https://en.wikipedia.org/wiki/SQL_injection
- https://www.owasp.org/index.php/SQL_Injection

We will be doing manual SQL injection attacks using the DVWA. DVWA stands for (Damn Vulnerable Web Application):

- It's a PHP/MySQL web application that is damn vulnerable.
- Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a classroom environment.

We will be injecting 'always true' SQL statements into the *User ID* field of the database. Our ultimate goal is to obtain username and raw-MD5 password contents from the users table of DVWA.

Setup

- Using the results of the network scan in the Enumeration section, identify the machine (and its IP) that is running DVWA.
- Once you have identify the machine, connect to the machine using the browser and the machines IP address and enter 'the store'.
- You are now prompted to login. Login with username: admin, password: password and you should see DVWA home page.
- Set *DVWA Security Level (left-hand column)*, by clicking on DVWA security on the left hand menu, selecting 'low' and submitting the selection.
- Go to the SQL Injection option by selecting 'SQL Injection' from the left-hand menu.

1. Basic Injection

Input: in the *User ID* text box input *"1"* and click Submit.

- The webpage/code is **supposed to** print ID, First name, and Surname to the screen.

The PHP *select* statement that we will be exploiting, specifically $id is:

- $getid = "SELECT first_name, last_name FROM users WHERE user_id = '**$id**'";

Always True Scenario

Input: the text *"%' or '0'='0' "* into the *User ID* text box and click Submit – don't enter the double quotes.

In this scenario, we are saying to display all records that are **false** and all records that are **true**.

- %' - Will probably not be equal to anything, and will be false.
- '0'='0' - Is equal to true, because 0 will always equal 0.

Database Statement:

- mysql> SELECT first_name, last_name FROM users WHERE user_id = '**%' or '0'='0**';

2. Display Database Version, Database User and Database Name

Input: the text *“%' or 0=0 union select null, version() #”* into the *User ID* Textbox and click Submit.

Notice in the last displayed line the version number is displayed in the surname. This is the version of the MySQL database used to store the data for the application.

Display Database User and Database Name
You can get the *Database User* by entering a string similar to the above, substituting *user ()* for *version (), and fo*r *Database Name* you’d substitute *database ()*

3. Database Schema and Table Names

To really make progress in hacking the database, what you want to know is the schema, that is, the tables that are in the database.

To display all the tables in the information_schema you need to do the following:
Input: *“%' and 1=0 union select null, table_name from information_schema.tables #”* into the *User ID* text box and click Submit

Now we are displaying all the tables in the information_schema of database.

- The INFORMATION_SCHEMA is the information database, the place that stores information about all the other databases that the MySQL server maintains.
- There are a lot of tables.

4. Users Tables

We now have the table names. What we really want is the table with the usernames and passwords. Let’s reduce the number of table names we need to look, so we’ll just display all the ‘user’ tables in information_schema
Input: *“%' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#”* into the *User ID* Textbox and click Submit.

We are displaying all the tables that start with the prefix "user" in the information_schema database.

5. Columns with Critical Data

We want to know what columns of data are stored in the *User* table.
Input: *“%' and 1=0 union select null, concat(table_name,0x0a,column_name) from information_schema.columns where table_name = 'users' #”* into the User ID text box and click Submit.

- Take a screenshot and insert it into your report under the heading ‘**User Table Columns’**.

Now we are displaying all the columns in the **users** table. Notice there are user_id, first_name, last_name, user and **Password** columns.

6. Username and Passwords

Now for the gold… we want all the column field contents in the information_schema *User* table

- Input *“%' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #”* into the *User ID* text box and click Submit.

Now we have successfully displayed all the necessary authentication information in this database, you can just use your favourite cracking tool to decrypt the passwords.