FCS Assignment 3

Name: Meetakshi Setiya
Roll no: 2019253

Question 1.

Yes, even if the stolen data was encrypted, the organisation must notify the security breach. This is because:

- Everyone in the organisation has to be alerted and it has to be made sure that any further damages or data loss is prevented.
- It is very important to discuss the severity and the scope of the breach along with the legal repercussions.
- The organisation also has to make sure that the exploited vulnerability is fixed and that any compromised system is shut down or suspended. This should be done to make sure that no further unauthorized access is possible.
- Besides, legislatures like The General Data Protection Regulation (GDPR) require the organisations to mandatorily notify the affected users that the security of their data has been compromised. Anything otherwise might lead to legal ramifications for the organisation.
- There is always a possibility that even if encrypted data is stolen, and there is no ample security measure taken, the attacker can attempt to steal the encryption key too and in that case, the harm will be irreversible.

Example: In 2021, a hacker used data scraping techniques to exploit LinkedIn's API before dumping an information data set containing details of 500M of its customers. LinkedIn argued that no private data was exposed, the hacker posed a scraped data sample containing phone numbers, geolocation records, gender, social media details of the users. This was plenty of information for malicious parties to start social engineering and other attacks on the victims of the breach. In November 2015, the database of the website Abandonia was breached, giving the attackers access to the data (IP address, username and encrypted passwords) of more than 700k users registered on the website. The passwords were salted and encrypted using MD5 which were easily cracked. The breach was not even made known to its 700k+ victims until sometime later.

Question 2.

Both anonymization and pseudonymisation are practices used to secure sensitive or private data about individuals and prevent exposing the said data to unauthorized personnel.

Anonymization: Anonymization is the process of protecting private or sensitive information by encryption or complete removal of identifiers (like IP addresses, devices etc.) that can be used to identify an individual. Since anonymization involves permanent replacement of the selected data with unrelated characters, the data once anonymized cannot be re-identified directly. Anonymizing data however, can destroy the value that the data holds for an organization. Additionally, GDPR does not see anonymized data as personal data anymore.

Pseudonymization: Pseudonymization is the process of switching sensitive details, like name, email etc with false information- like a pseudonym or an alias such that it cannot be attributed to a specific subject without the use of additional information. A good example would be the replacement of names with temporary IDs. Pseudonymization is a reversible process, unlike anonymization, which means that even though the data was changed for the purpose of its concealment/de-identification, it can still be re-identified if necessary (using a proper key). According to GDPR, a pseudonym is still considered to be personal data because of its reversible nature.

Given data: Patient John Doe has a gamma-GT value of 83U/L

Anonymized data: A patient has has a gamma-GT value of 83U/L

Pseudonymised data: Patient 1221B has a gamma-GT value of 83U/L
(Key mapping table: Patient 1221B → John Doe)

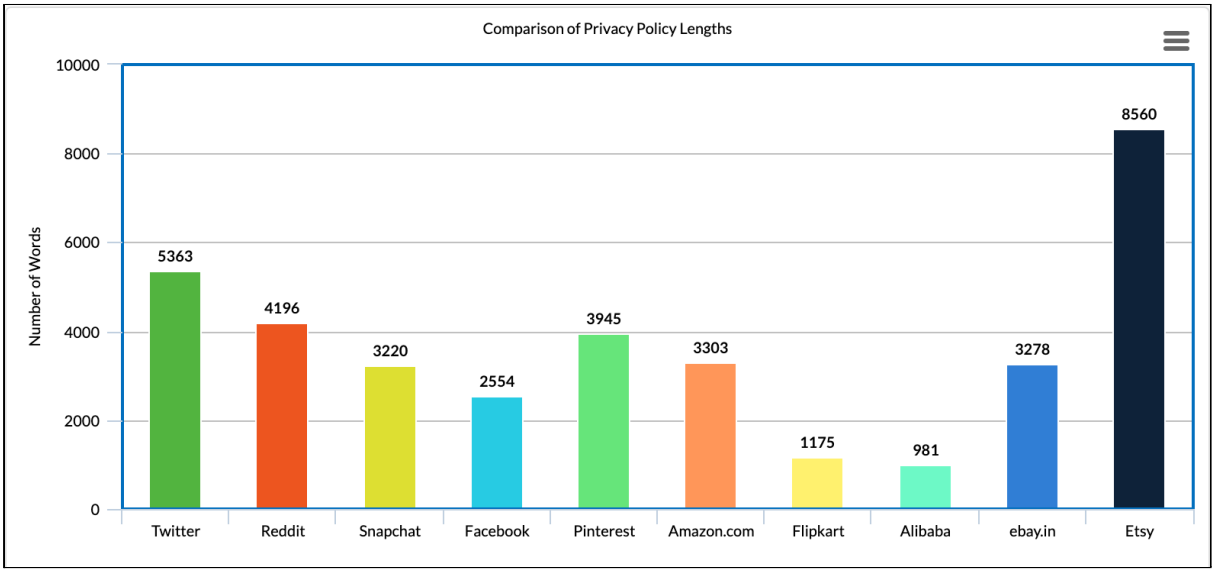
Question 3.

The social media websites I chose are- Twitter, Reddit, Snapchat, Facebook and Pinterest.

The e-commerce websites I chose are- Amazon.com, ebay.in, Flipkart, Alibaba and Etsy.

a.

Bar chart showing the lengths of privacy policy for 5 different social media websites and 5 e-commerce websites is given below:



b.

I used Webfx to get the text complexity metrics of the various privacy policies. The major ones are summarised below:

Website	**No. of words**	**No. of complex words**	**Average words per sentence**	**Flesch Kincaid Reading Ease***	**Flesch Kincaid Grade Level#**
----------	----------	**Social media**	**websites**	----------	----------
Twitter	5363	840	23.12	46.3	12.5
Reddit	4196	793	21.52	39.6	13.1
Snapchat	3220	512	19.75	50.5	11.1
Facebook	2554	406	21.11	49.5	11.6
Pinterest	3945	661	21.68	47.7	12
----------	----------	**E- commerce**	**websites**	----------	----------

Amazon.com	3303	670	23.43	37.2	13.9
Flipkart	1175	212	18.65	47.1	11.3
Alibaba	981	197	23.93	35.5	14.3
ebay.in	3278	657	26.02	35.6	14.8
Etsy	8560	1603	25.86	37.2	14.5

*The Flesch Kincaid readability scores are the most widely used measures of readability. The score is based on a ranking scale of 0-100, and the higher the score the better. For most businesses, 65 is a good target.
#The Flesch Kincaid readability scores are the most widely used measures of readability. The score is based on the American school grade one would need to be in to comprehend the material on a page.

c.

Observations:

- ➔ Every privacy policy scores low on the readability ease front and thus requires some scrutiny and effort to go through them. Average complexity of words used (total number of words/complex words) also bordered to a ratio of roughly 6:1.
- ➔ On comparing the complexity of vocabulary used, the ratio of total number of words to complex words in the text was on average higher for the social media websites (nearing approx 6) than the e-commerce websites.
- ➔ Out of the social media websites, Snapchat scored the highest in readability score and had a lesser average number of words per sentence. It was also comparatively shorter than the privacy policies of its counterparts (except facebook). Interestingly, Snapchat is also reported to be a safer social media app than its counterparts.
- ➔ Out of the social networking sites, Pinterest's privacy policy had the least ratio of total words to complex words - 5.3:1. On going through the document, I found it comparatively easier to comprehend on a first glance.
- ➔ Facebook is reportedly the most used social media platform, even in 2021, yet its privacy policy had the least word count of the rest of the social media platforms'. The complexity ratio of the text however, was on the higher end- ie. 6.29.

- ➔ Etsy's privacy policy was the longest and the hardest to go through because of long sentences and the sheer length of the document.
- ➔ Out of the ecommerce websites, Alibaba's privacy policy text was the shortest. It was, in fact, the shortest privacy policy text in the table. The ratio of total words to complex words was also the least: around 4.9.
- ➔ Flipkart's privacy policy text has the least sentence length out of every privacy policy text present in the table. The complexity ratio is about 5.5. The text was also on the shorter end of the range- a mere 1175 words long.
- ➔ Amazon.com is reportedly the biggest e-commerce website by traffic. Its privacy policy was neither too long, nor too short, by word count. Nor was the complexity ratio as high as that of some of the other websites in this table. In fact it was about 4.92.
- ➔ For Amazon.com, Flipkart and Alibaba, the order of popularity is the same as the order of privacy policy's length, is the same as the enlisted order.
- ➔ Additionally, ebay.in scored the highest in the Flesch Kincaid Grade Level readability test scale. Snapchat scored the highest in Flesch Kincaid reading ease scale, and Flipkart had the lowest average number of words per sentence.

Question 4.

a.

Given below is a screenshot of the metadata of the image 4_img.jpg as shown by exiftool.

Location- 36 deg 50' 30.18" S, 174 deg 46' 32.89" E
City- Auckland
Country- New Zealand

```
1573 → ~/desktop  ❯❯ exiftool 4_img.jpg
ExifTool Version Number         : 12.34
File Name                       : 4_img.jpg
Directory                       : .
File Size                       : 83 KiB
File Modification Date/Time     : 2021:11:02 14:22:09+05:30
File Access Date/Time           : 2021:11:02 14:22:20+05:30
File Inode Change Date/Time     : 2021:11:02 14:22:19+05:30
File Permissions                : -rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Exif Byte Order                 : Big-endian (Motorola, MM)
Image Description               : Malana
X Resolution                    : 1
Y Resolution                    : 1
Resolution Unit                 : None
Y Cb Cr Positioning             : Centered
GPS Version ID                  : 2.3.0.0
GPS Latitude Ref                : South
GPS Longitude Ref               : East
Image Width                     : 719
Image Height                    : 496
Encoding Process                : Progressive DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Image Size                      : 719x496
Megapixels                      : 0.357
GPS Latitude                    : 36 deg 50' 30.18" S
GPS Longitude                   : 174 deg 46' 32.89" E
GPS Position                    : 36 deg 50' 30.18" S, 174 deg 46' 32.89" E
1574 → ~/desktop  ❯❯
```

b.

IIIT Delhi's GPS coordinates are: 28.5455536926139, 77.27307041485886
The command that can be used to change the location of the image to IIIT Delhi is:

```
exiftool -GPSLatitude="28.5455536926139"
-GPSLongitude="77.27307041485886" 4_img.jpg
```

Here is a screenshot showing the changed GPS location:

```
[1579 → ~/desktop  ›› exiftool -GPSLatitude="28.5455536926139" -GPSLongitude="77.27307041485886" 4_img.jpg
    1 image files updated
[1580 → ~/desktop  ›› exiftool 4_img.jpg
ExifTool Version Number         : 12.34
File Name                       : 4_img.jpg
Directory                       : .
File Size                       : 83 KiB
File Modification Date/Time     : 2021:11:02 14:49:59+05:30
File Access Date/Time           : 2021:11:02 14:50:01+05:30
File Inode Change Date/Time     : 2021:11:02 14:49:59+05:30
File Permissions                : -rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Exif Byte Order                 : Big-endian (Motorola, MM)
Image Description               : Malana
X Resolution                    : 1
Y Resolution                    : 1
Resolution Unit                 : None
Y Cb Cr Positioning             : Centered
GPS Version ID                  : 2.3.0.0
GPS Latitude Ref                : South
GPS Longitude Ref               : East
Image Width                     : 719
Image Height                    : 496
Encoding Process                : Progressive DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Image Size                      : 719x496
Megapixels                      : 0.357
GPS Latitude                    : 28 deg 32' 43.99" S
GPS Longitude                   : 77 deg 16' 23.05" E
GPS Position                    : 28 deg 32' 43.99" S, 77 deg 16' 23.05" E
1581 → ~/desktop  ››
```

c.

I used the ‘md5’ command on macOS to find the md5 checksums of the image before and after changing its location.

The command I used was :

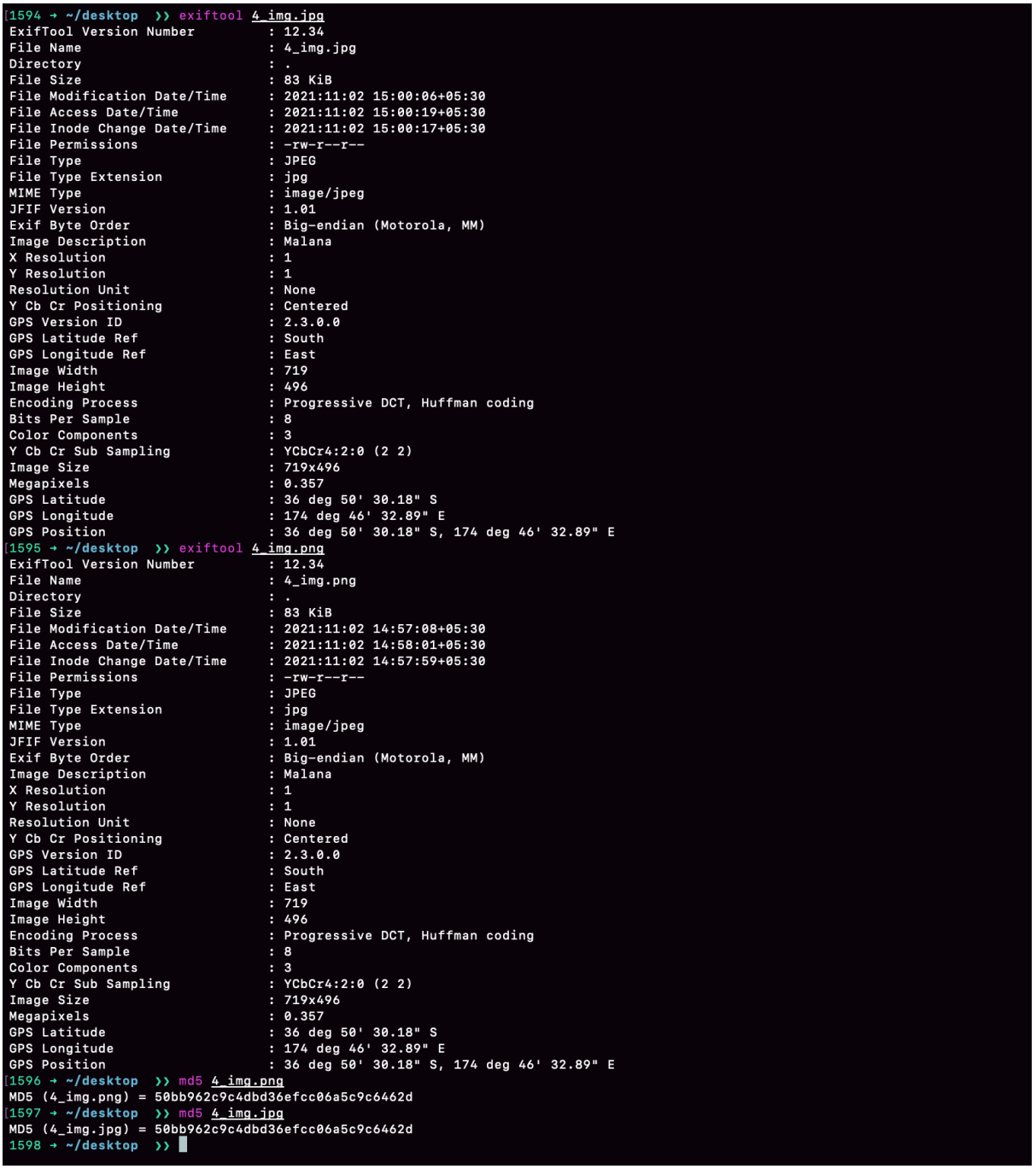
```
md5 4_img.jpg
```

```
[1575 → ~/desktop  ›› md5 4_img.jpg
MD5 (4_img.jpg) = 50bb962c9c4dbd36efcc06a5c9c6462d
[1576 → ~/desktop  ›› exiftool -GPSLatitude="28.5455536926139" -GPSLongitude="77.27307041485886" 4_img.jpg
    1 image files updated
[1577 → ~/desktop  ›› md5 4_img.jpg
MD5 (4_img.jpg) = 7f659256b4a414b5f30180c53ff433d9
1578 → ~/desktop  ››
```

The two checksums are different. This is because md5 hash is a hash of the complete binary image file. For two md5 hashes to match, the images as well as their metadata must be completely identical. Hence any change to the file metadata will produce a new hash value as is visible in the screenshot.

d.

I changed the extension of the image from jpg to png. On comparing the metadata of the two variants, I found them to be exactly equal. Also, the md5 hash for both formats was the same as well. The screenshot is given below:



No, we cannot trust the extension of a file for forensic purposes. We should always parse the file to make sure that the content is exactly what the file extension specifies. File extensions are only used so that the operating system can call an appropriate program to open the file. Changing the file extension will not change the file type. Someone can, for example, upload a file called “file.mp4” which actually is a malicious script that could harm the service/system.

Question 5.

DNS traffic monitoring can be used to deanonymize the users of Tor network. Monitoring DNS traffic requests and combining it with various fingerprinting techniques can be used to create a new type of "DNS-enhanced website fingerprinting attack." These kinds of passive attacks can not only uncover the IP address of the client but also the physical location of servers, sometimes. These attacks work particularly well against websites that do not have frequent traffic since their DNS records stand out. For example, an attacker could be monitoring the DNS traffic and if a user A visits a sparingly-visited website on tor, they can look up the exit node traffic on the said website and identify the IP address that requested the service of that website.

A research team also discovered that roughly one-third of DNS requests sent through Tor exit relays are routed through Google's public resolvers. Therefore, resolving DNS queries through Tor service ensures a significantly higher level of anonymity than making the requests directly because of DNS leaks. This also prevents the ISP and the resolved from knowing which domain was resolved by the user.

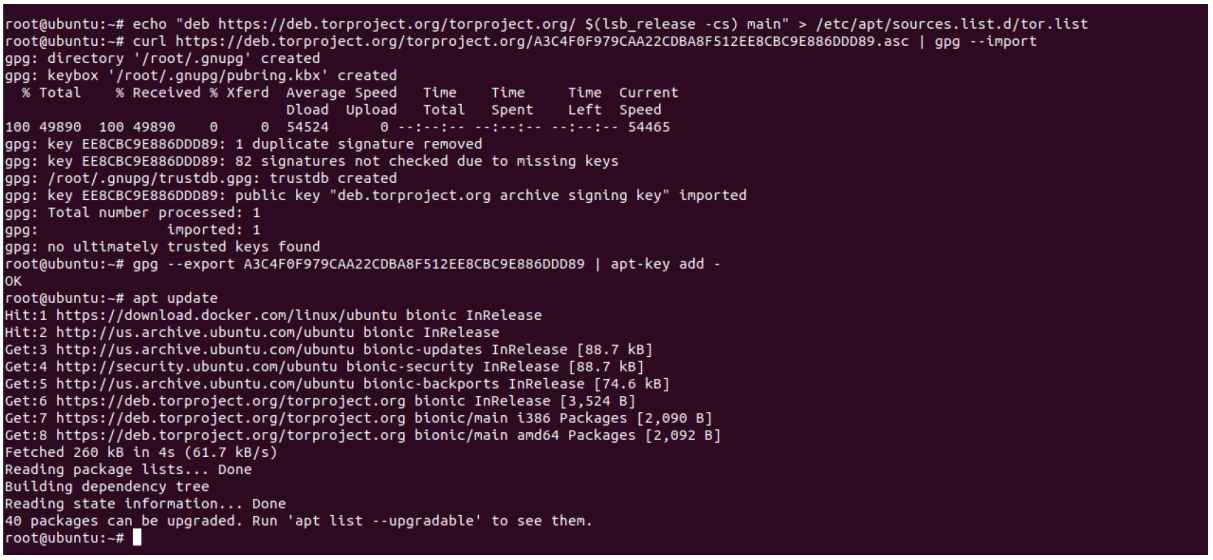
Question 6.

I will be installing Tor on an Ubuntu 18.04 Virtual Machine.
First, install apt-transport -https to use https-repositories and curl to download the repo key.

```
sudo apt install apt-transport-https curl
```

Then, add Tor repo and its key using these commands:

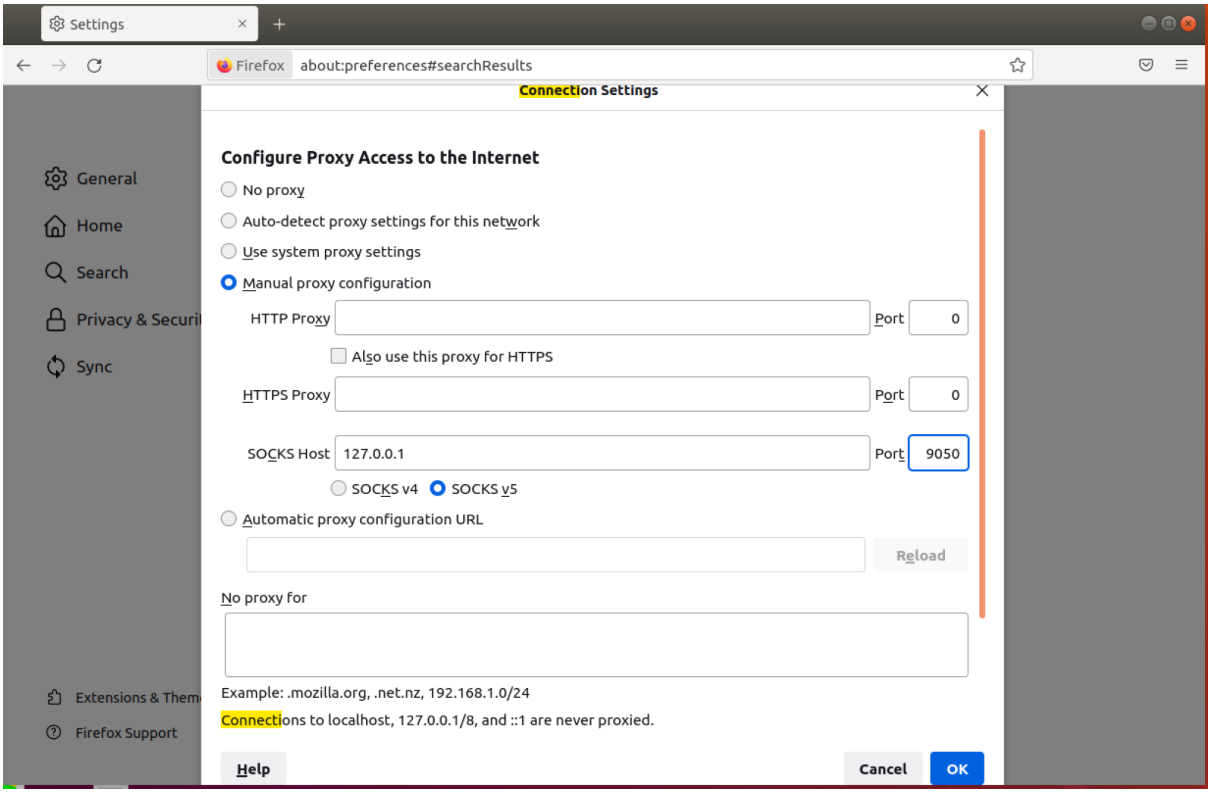
```
sudo -i
echo "deb https://deb.torproject.org/torproject.org/ $(lsb_release
-cs) main" > /etc/apt/sources.list.d/tor.list
curl
https://deb.torproject.org/torproject.org/A3C4F0F979CAA22CDBA8F512
EE8CBC9E886DDD89.asc | gpg --import
apt update
```



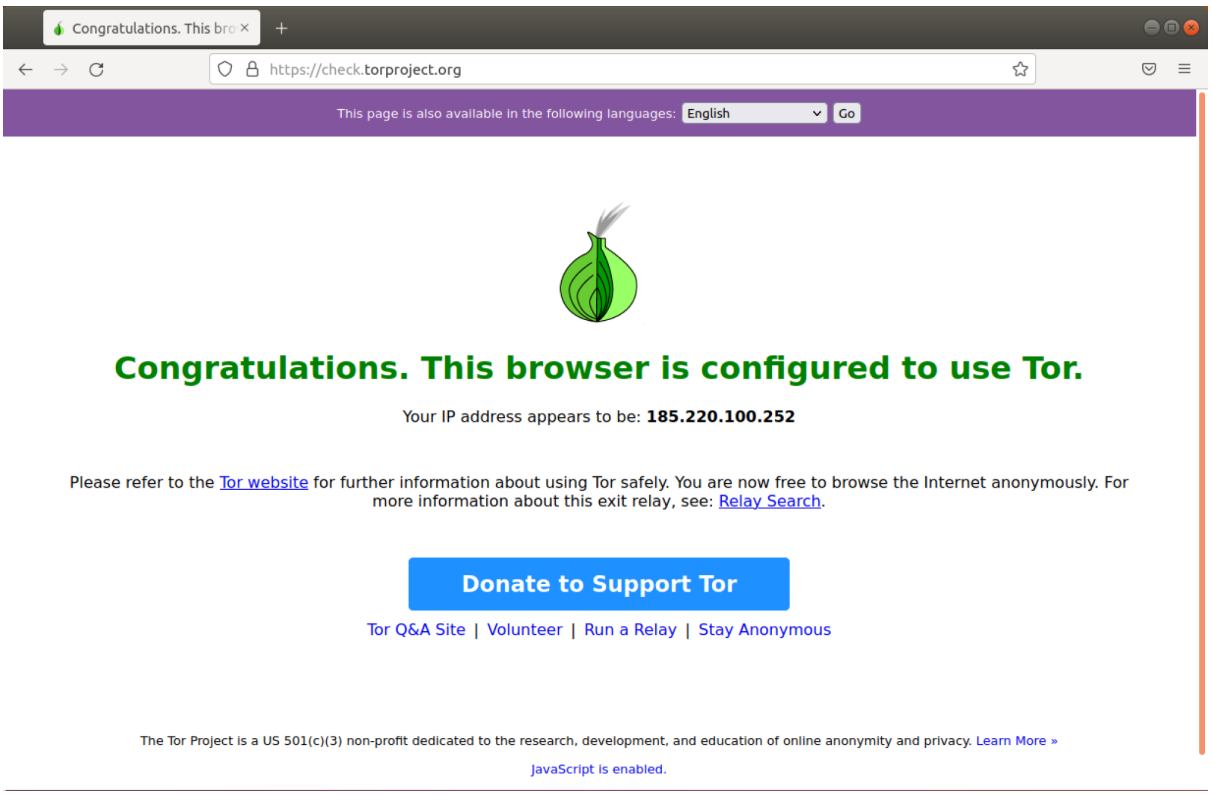
Now, install torsocks, tor-geoipdb and deb.torproject.org-keyring along with tor.

```
sudo apt install tor tor-geoipdb torsocks
deb.torproject.org-keyring
```

Once tor is installed, configure Firefox to connect to Tor SOCKS5 proxy on localhost:9050



Now, Tor can be successfully run on Firefox.



To get the hostname of the onion service on ubuntu, I followed these steps:

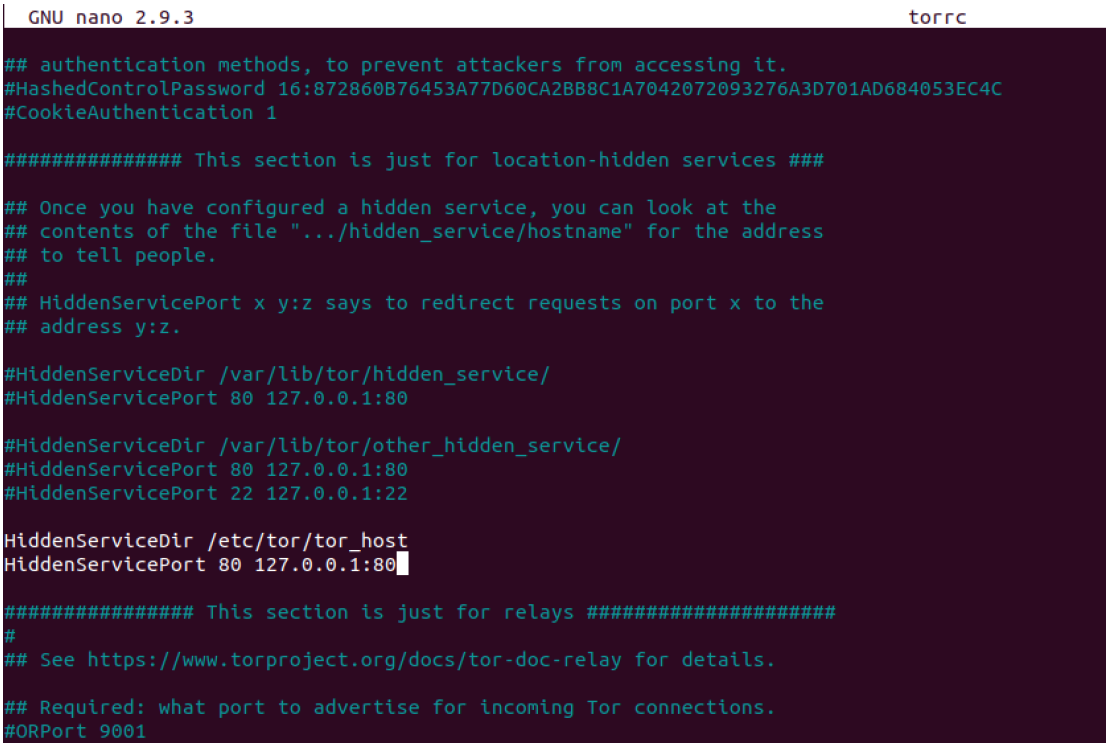
First, navigate to /etc/tor and create a directory tor_host using these commands:

```
sudo -i
cd /etc/tor
mkdir tor_host
```

Now, modify the permissions of the tor_host directory using the given command:

```
chmod 700 tor_host
```

The next step is to open the torrc file and do the appropriate configurations to set up an onion service. I then put these two lines in the location-hidden services section:
HiddenServiceDir /etc/tor/tor_host
HiddenServicePort 80 127.0.0.1:80
According to the Tor project,
The HiddenServiceDir line specifies the directory which should contain information and cryptographic keys for the onion service and the HiddenServicePort line specifies a *virtual port*.



Now, after saving the torrc file, restart the Tor service using the command `tor`.

Now, when I checked the contents of the tor_host directory, the file hostname contained the hostname autogenerated by tor.

```
vasilisa@ubuntu:/etc/tor$ sudo ls tor_host
authorized_clients  hostname  hs_ed25519_public_key  hs_ed25519_secret_key
vasilisa@ubuntu:/etc/tor$ sudo cat tor_host/hostname
vwojiogorepq2at5t5neu4b4skjsa3n5267fd6rdhvwovcpv33cgrtid.onion
vasilisa@ubuntu:/etc/tor$
```

The auto generated hostname is :
vwojiogorepq2at5t5neu4b4skjsa3n5267fd6rdhvwovcpv33cgrtid.onion

Question 7.

a.
One way of doing password less authentication is using SSH private-public key pairs.
First, use ssh-keygen to generate the RSA key pair. I also used a passphrase to secure my private key.

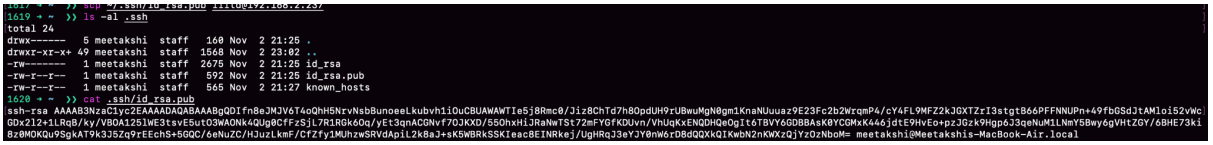
```
ssh-keygen -t rsa
```

```
1614 ➜ ~  ❯❯ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/meetakshi/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/meetakshi/.ssh/id_rsa.
Your public key has been saved in /Users/meetakshi/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:rrsRsACs/MRrEzJ12x7BKjLrv/ePT/4lW9E+U5wyARE meetakshi@Meetakshis-MacBook-Air.local
The key's randomart image is:
+---[RSA 3072]----+
|o     .    Eo    |
| o . . o    .    |
|o + o + .    .   |
|.* * = o      o..|
|  O = o S    + oo|
| . =   +      = .|
|. . . . .. . o + |
| .   . o+   =   o|
|  .oo ==o+.o     |
+----[SHA256]-----+
```

Now, check the generated keys and view the public key using cat.

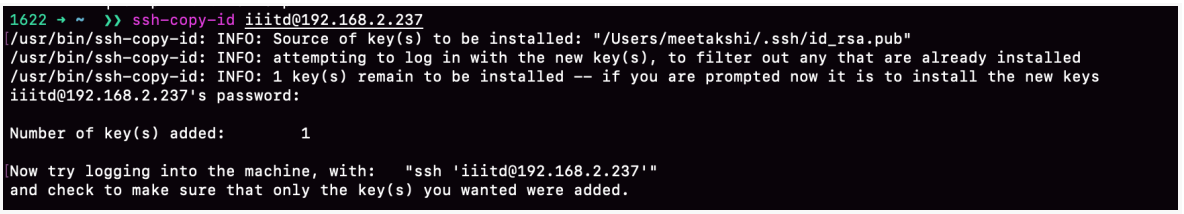
```
ls -al .ssh
```

```
cat .ssh/id_rsa.pub
```

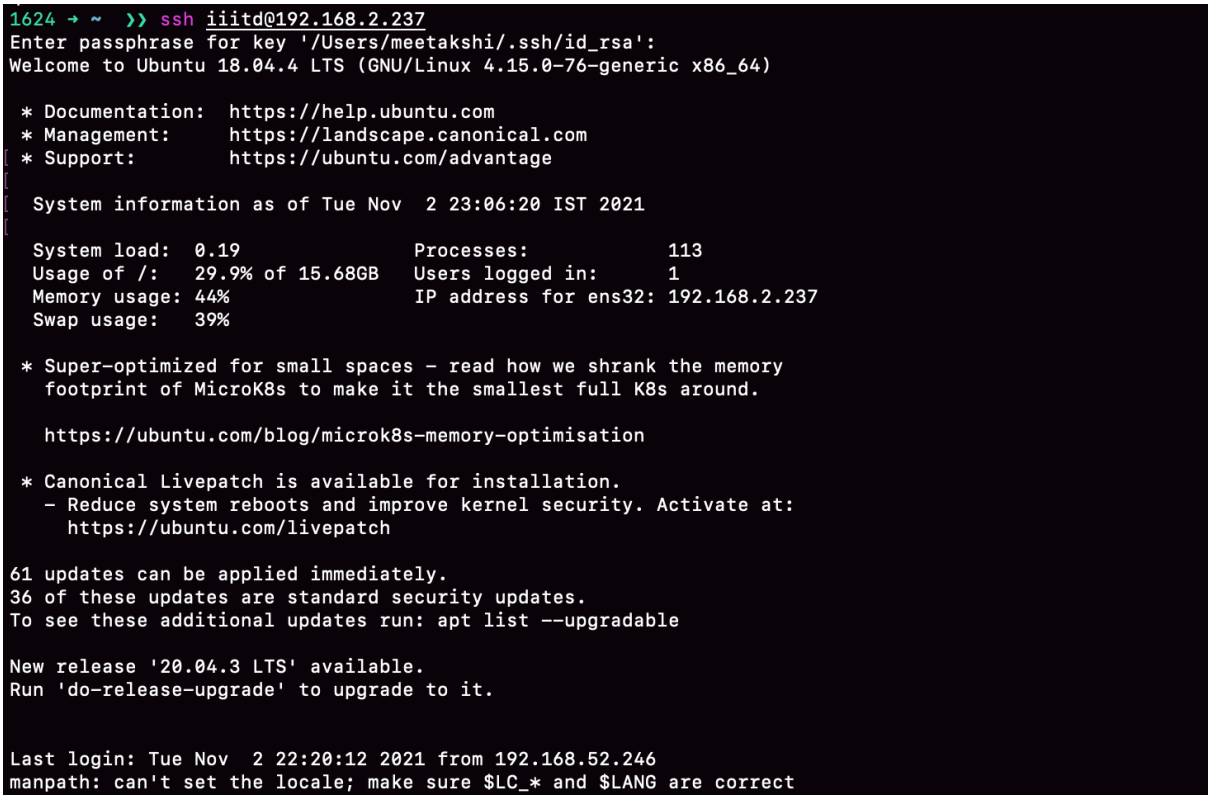


Now, use ssh-copy-id to propagate the public key to the VM since password auth is currently enabled.

```
ssh-copy-id iiitd@192.168.2.237
```



Now, as the prompt says, ssh into the VM. A prompt for passphrase appears. I entered the passphrase I used to secure my keys during generation.



Now, if I check the list of authorized keys using cat,

My ssh public key is listed here.

b.

Using (ssh) public keys is a safer method than using passwords. Some of the reasons why this is true are:

- ➔ Since keys do not have to be memorized or are human generated, they are long and complex and thus, are more difficult to brute-force or guess than passwords.
- ➔ SSH keys are not sent to the server, unlike passwords. If an attacker takes over the server, they can easily view the password if proper security measures are not followed. In ssh, the private key remains on the client side, and no secret value is sent to the server. Hence, keys are in this way more robust than passwords.
- ➔ Where a user can use multiple systems during password authentication, authentication using a public key can only come from a system where the private key resides.
- ➔ Keys can even be password protected to provide another layer of security.

All in all, keys provide a method of maintaining a degree of entropy and manageability while also providing a method of access that doesn't impede with the actions of a legitimate user.

c.

My public ssh key is :

AAAAB3NzaC1yc2EAAAADAQABAAABgQDIfn8eJMJV6T4oQhH5NrvNsbBunoeeLkubvh1iOuCBUAWAWTIe5j8Rmc0/Jiz8ChTd7h8OpdUH9rUBwuMgN0gm1KnaNUuuaz9E23Fc2b2WrqmP4/cY4FL9MFZ2kJGXTZrI3stgtB66PFFNNUPn+49fbGSdJtAMloi52vWcGDx2l2+1LRqB/ky/VBOA125lWE3tsvE5utO3WAONk4QUg0CfFzSjL7R1RGk6Oq/yEt3qnACGNvf7OJKXD/55OhxHiJRaNwTSt72mFYGfKDUvn/VhUqKxENQDHQeOgIt6TBVY6GDBBAsK0YCGMxK446jdtE9HvEo+pzJGzk9Hgp6J3qeNuM1LNmY5Bwy6gVHtZGY/6BHE73ki8z0MOKQu9SgkAT9k3J5Zq9rEEchS+5GQC/6eNuZC/HJuzLkmF/CfZfy1MUhzwSRVdApiL2k8aJ+sK5WBRkSSKIeac8EINRkej/UgHRqJ3eYJY0nW6rD8dQQXkQIKwbN2nKWXzQjYzOzNboM=

I used the yes command to generate the text file having the letter M in each line.I will use it to encrypt the text files.

```
[1628 → ~/desktop  ›› yes M | head -n 10 > out10.txt
[1629 → ~/desktop  ›› yes M | head -n 100000 > out10k.txt
[1630 → ~/desktop  ›› yes M | head -n 100000000 > out10m.txt
 1631 → ~/desktop  ››
```

1. For AES encryption. The passphrase was my public ssh key.

```
1631 → ~/desktop  ›› openssl aes-256-cbc -salt -a -e -in out10.txt -out aesout10.txt
enter aes-256-cbc encryption password:
1633 → ~/desktop  ›› openssl aes-256-cbc -salt -a -e -in out10.txt -out aesout10.txt
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
1634 → ~/desktop  ›› openssl aes-256-cbc -salt -a -e -in out100k.txt -out aesout100k.txt
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
1635 → ~/desktop  ›› openssl aes-256-cbc -salt -a -e -in out100m.txt -out aesout100m.txt
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
1636 → ~/desktop  ››
```

2. For RC4 encryption. The passphrase was my public ssh key.

```
1641 → ~/desktop  ›› openssl rc4 -in out100k.txt -out rc4out100k.txt
enter rc4 encryption password:
Verifying - enter rc4 encryption password:
1642 → ~/desktop  ›› openssl rc4 -in out10.txt -out rc4out10.txt
enter rc4 encryption password:
Verifying - enter rc4 encryption password:
1643 → ~/desktop  ›› openssl rc4 -in out100m.txt -out rc4out100m.txt
enter rc4 encryption password:
Verifying - enter rc4 encryption password:
1644 → ~/desktop  ››
```

3. RSA gave error with the key

Question 8.

a.

I configured the firewall settings using iptables - Linux's built-in firewall. The INPUT chain is for all packets meant for the host computer i.e. packets going into local sockets. It controls the behaviour of incoming connections. Since the ping to my machine had to fail (disabled), I added a rule to the INPUT chain to drop all icmp packets from the incoming traffic. I did this using the command-

```
sudo iptables -A INPUT -p icmp --icmp-type echo-request -j REJECT
```

The screenshot below shows that pinging the localhost (loopback interface) failed but my machine could still ping google.com

```
vasilisa@ubuntu:~$ sudo iptables -A INPUT -p icmp --icmp-type echo-request -j REJECT
vasilisa@ubuntu:~$ ping -c 1 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
From 127.0.0.1 icmp_seq=1 Destination Port Unreachable

--- 127.0.0.1 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

vasilisa@ubuntu:~$ ping -c 1 www.google.com
PING www.google.com (142.250.192.164) 56(84) bytes of data.
64 bytes from del11s11-in-f4.1e100.net (142.250.192.164): icmp_seq=1 ttl=117 time=13.5 ms

--- www.google.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 13.575/13.575/13.575/0.000 ms
vasilisa@ubuntu:~$
```

Additionally, here is what the firewall configuration looks like now, with a new rule added to the INPUT chain:

```
vasilisa@ubuntu:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
REJECT     icmp --  anywhere             anywhere             icmp echo-request reject-with icmp-port-unreachable
```

b.

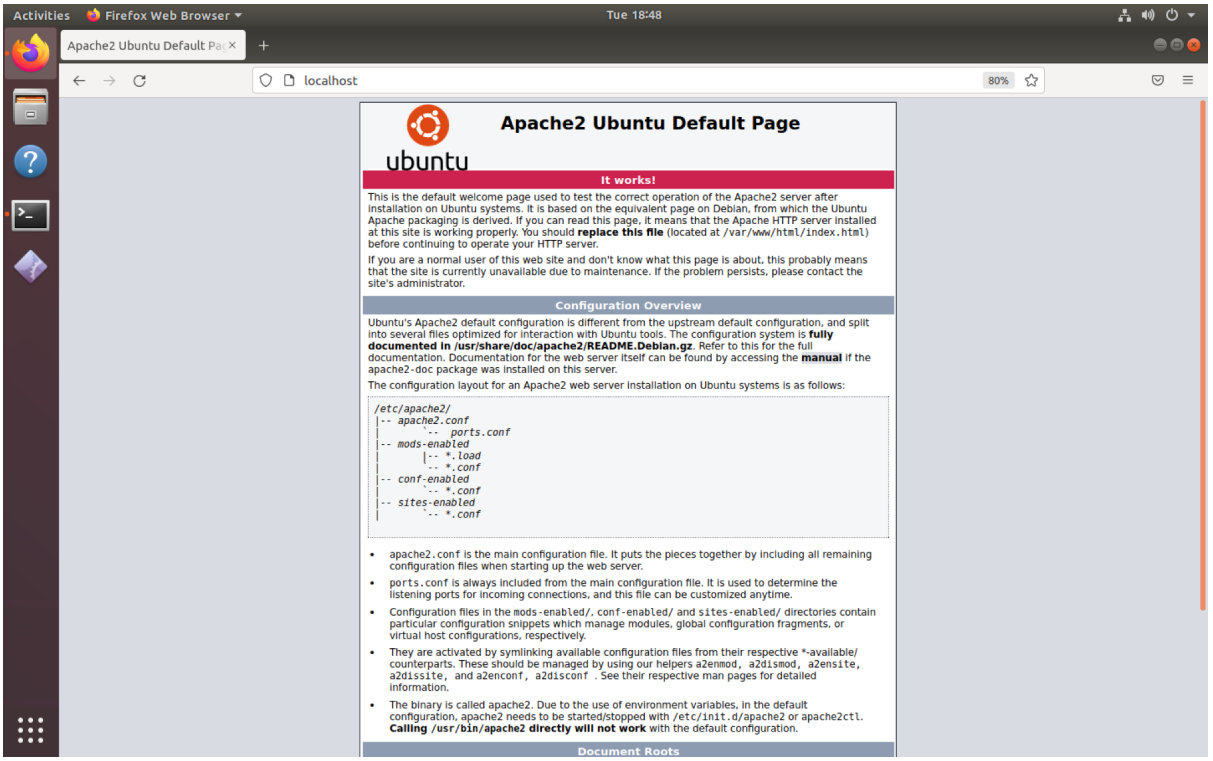
I will host the apache web server on my machine. Apache can be installed using this command:

```
sudo apt-get install apache2
```

Start the server using this command:

```
sudo service apache2 start
```

By default, apache listens on HTTP port 80. If we go to localhost:80 on the host machine, we see the apache default page.



Since my virtual machine and host machine both are in a private network, I can directly access the webpage from my host machine (without doing any port forwarding or such). Now, to only allow my mobile phone to access the web page, I will configure the firewall of my virtual machine using iptables to drop all connections to port 80 of the virtual machine except the one from my mobile phone.

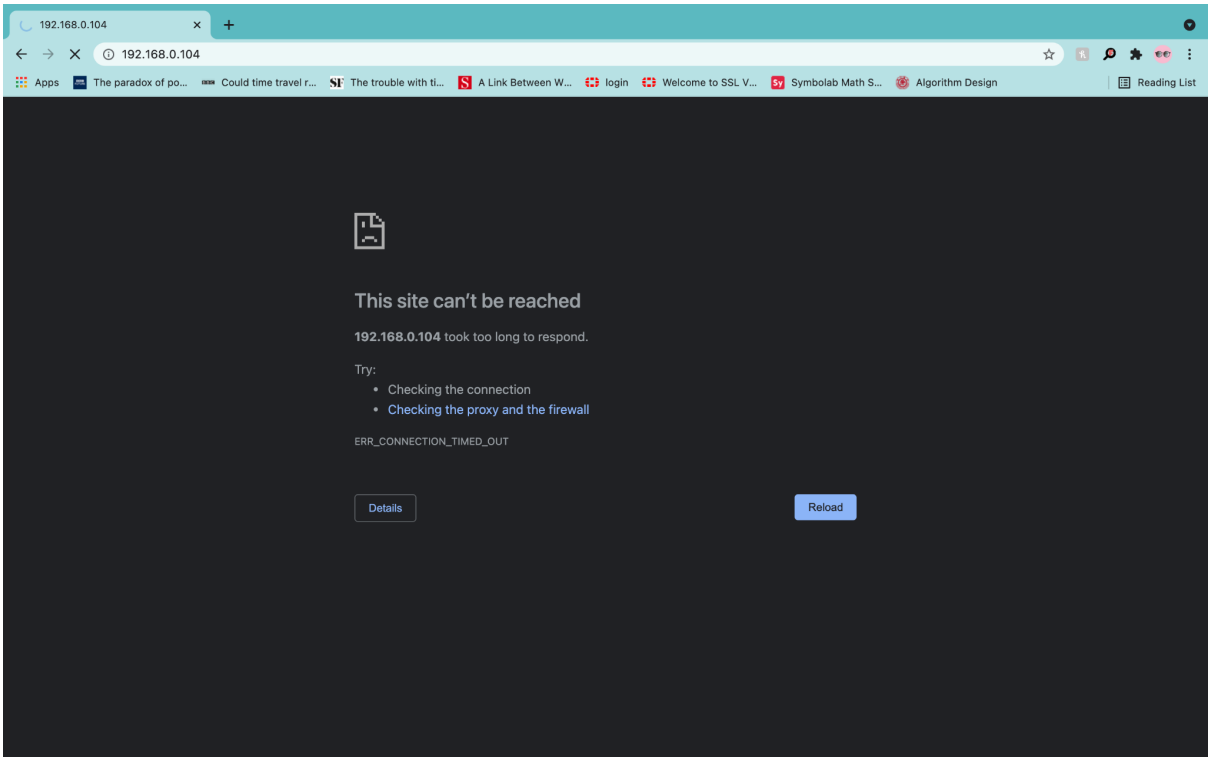
I used the following commands to update the INPUT chain policy of the firewall:

```
sudo iptables -A INPUT -p tcp --dport 80 -s <phone ip address> -j
ACCEPT
sudo iptables -A INPUT -p tcp --dport 80 -j DROP
```

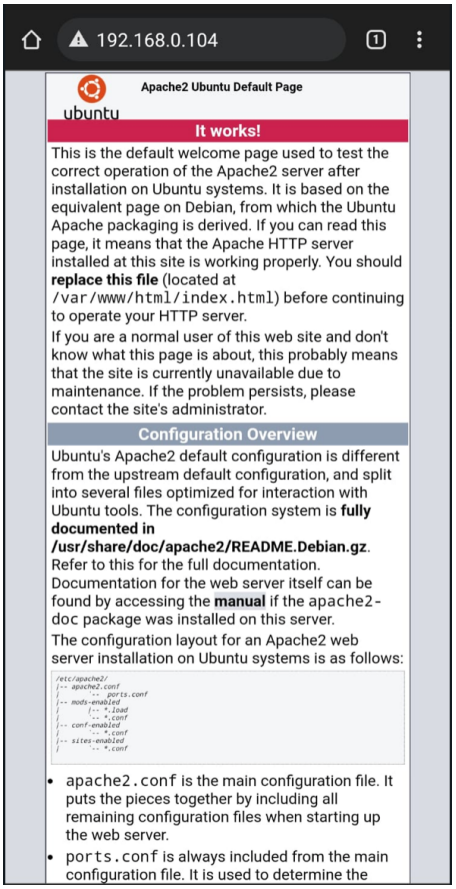
A screenshot of the updated rules-

```
vasilisa@ubuntu:~$ sudo iptables -A INPUT -p tcp --dport 80 -s 192.168.0.103 -j ACCEPT
vasilisa@ubuntu:~$ sudo iptables -A INPUT -p tcp --dport 80 -j DROP
vasilisa@ubuntu:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  192.168.0.103        anywhere             tcp dpt:http
DROP       tcp  --  anywhere             anywhere             tcp dpt:http
```

Now, if I access the webpage using my laptop, I am not able to.



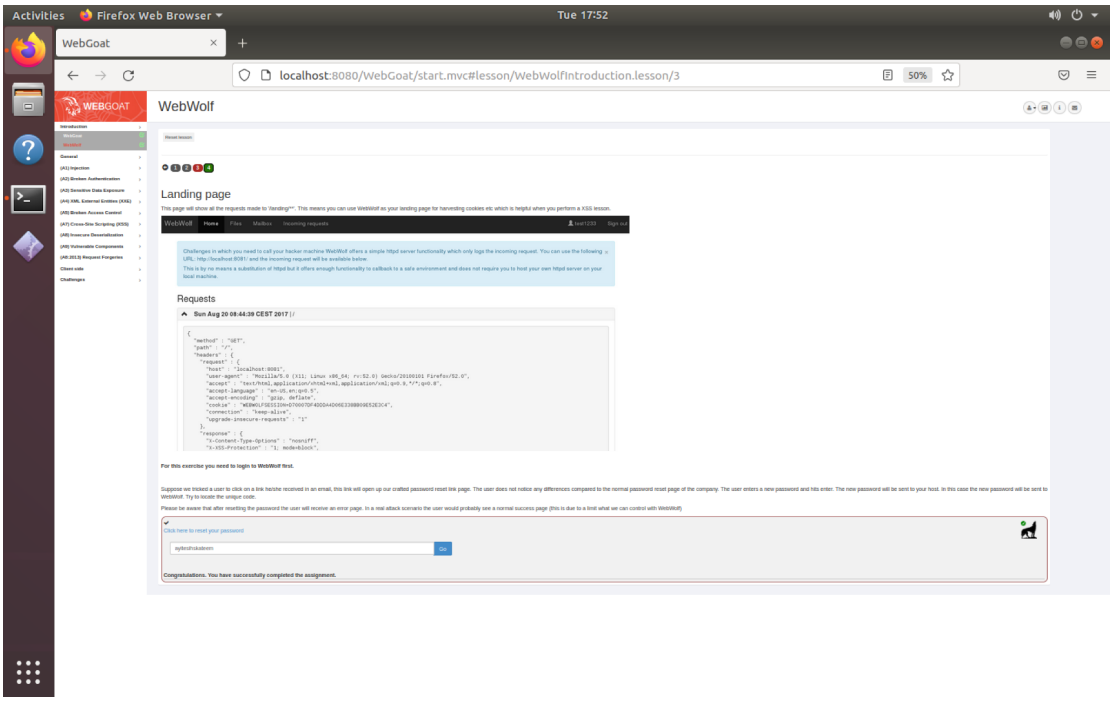
Whereas, on my mobile, I see the default apache web page.



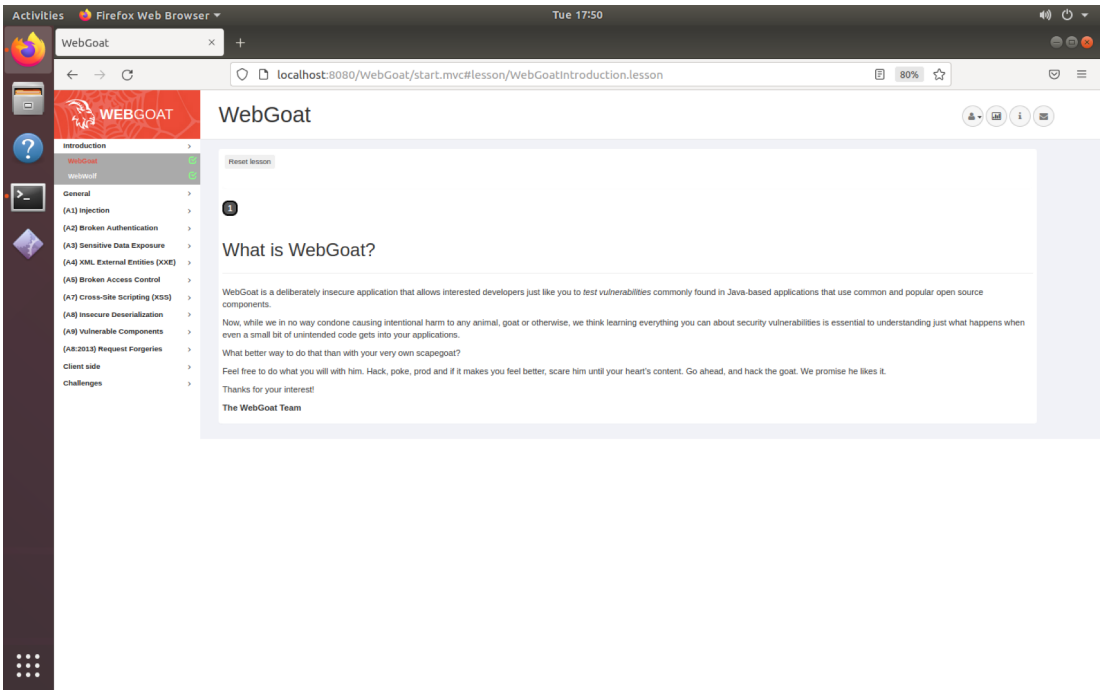
Question 9.

Screenshots of WebGoat assignments:

1. Introduction
    1.1. WebGoat

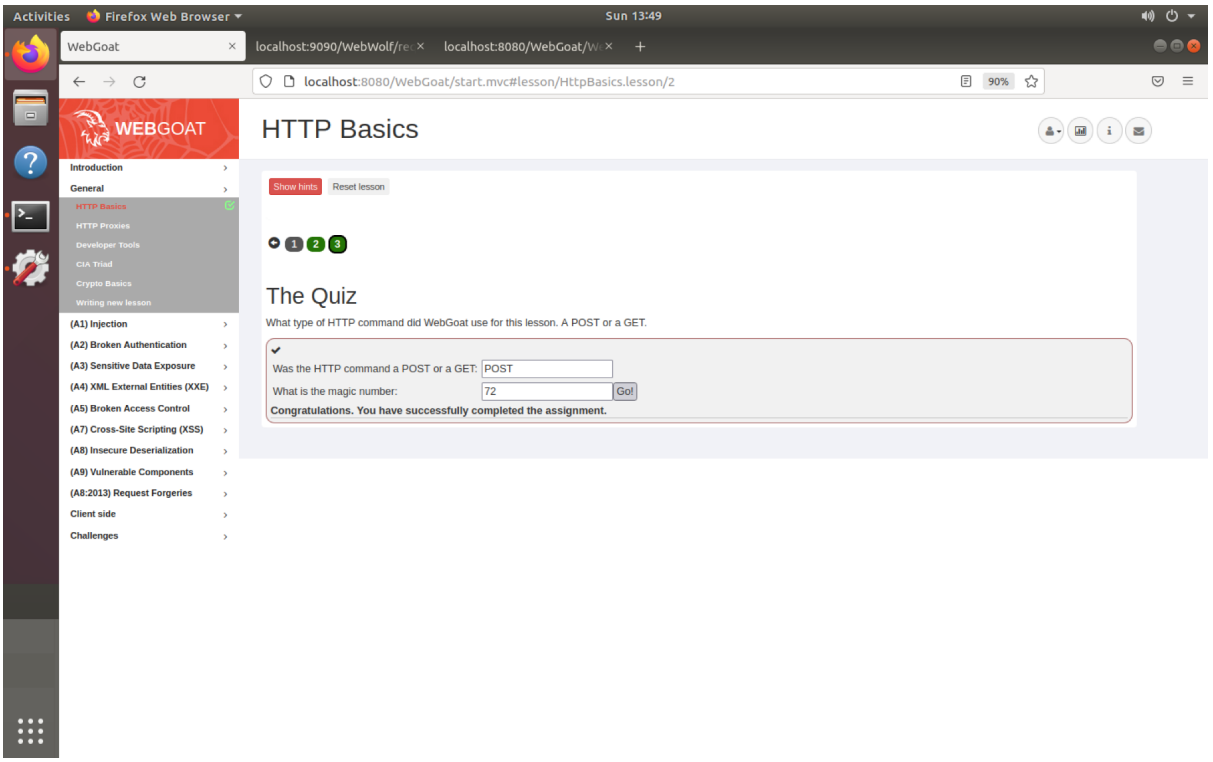


    1.2. WebWolf

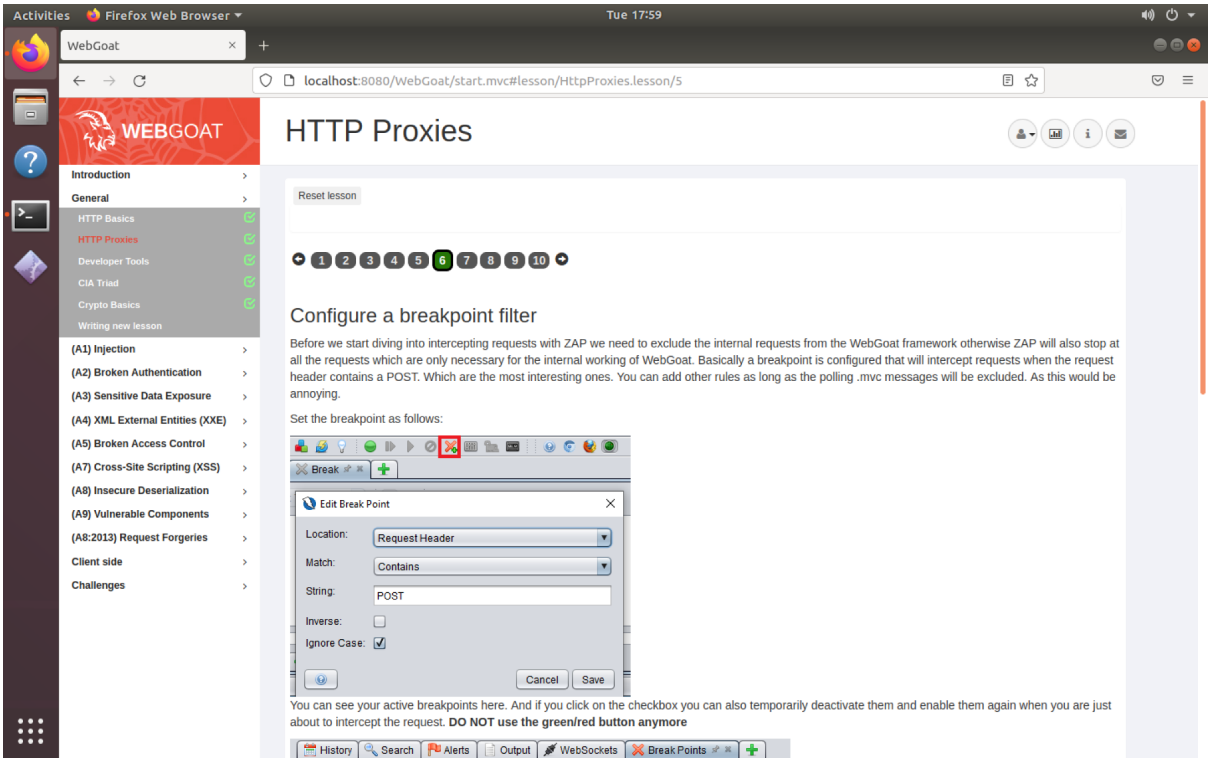


2. General

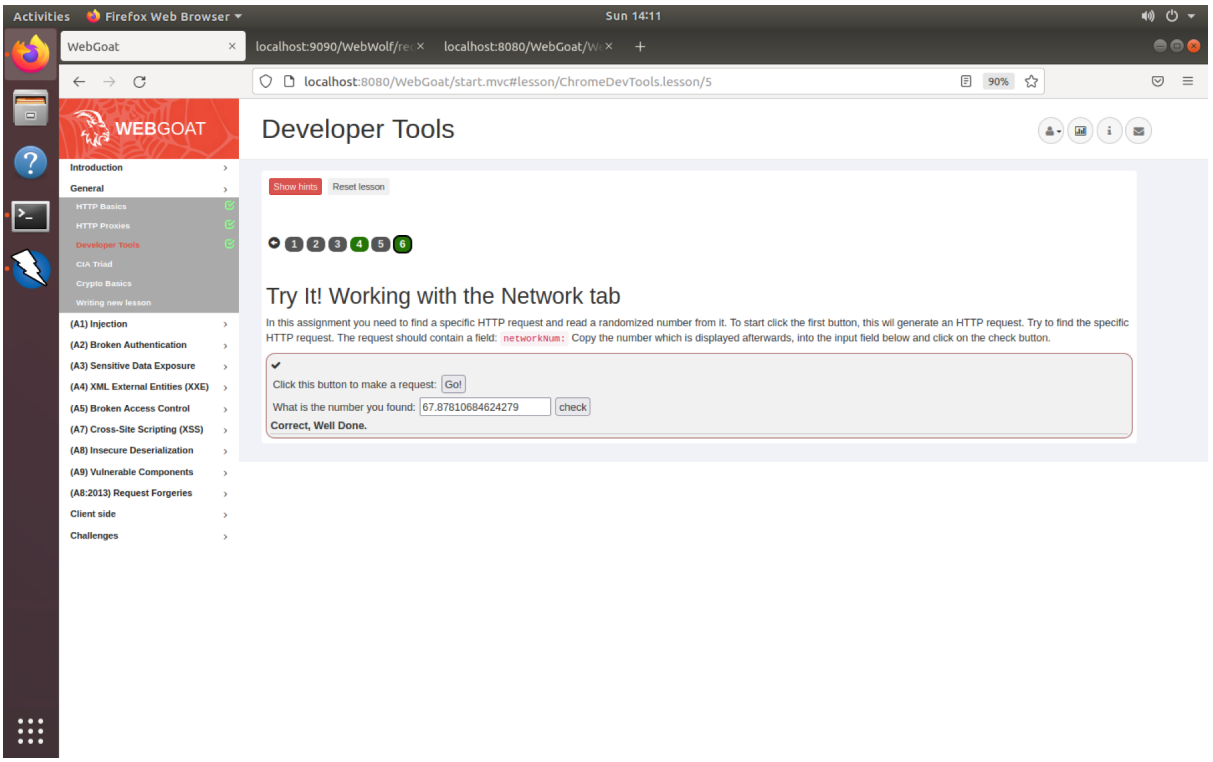
2.1. HTTP Basics



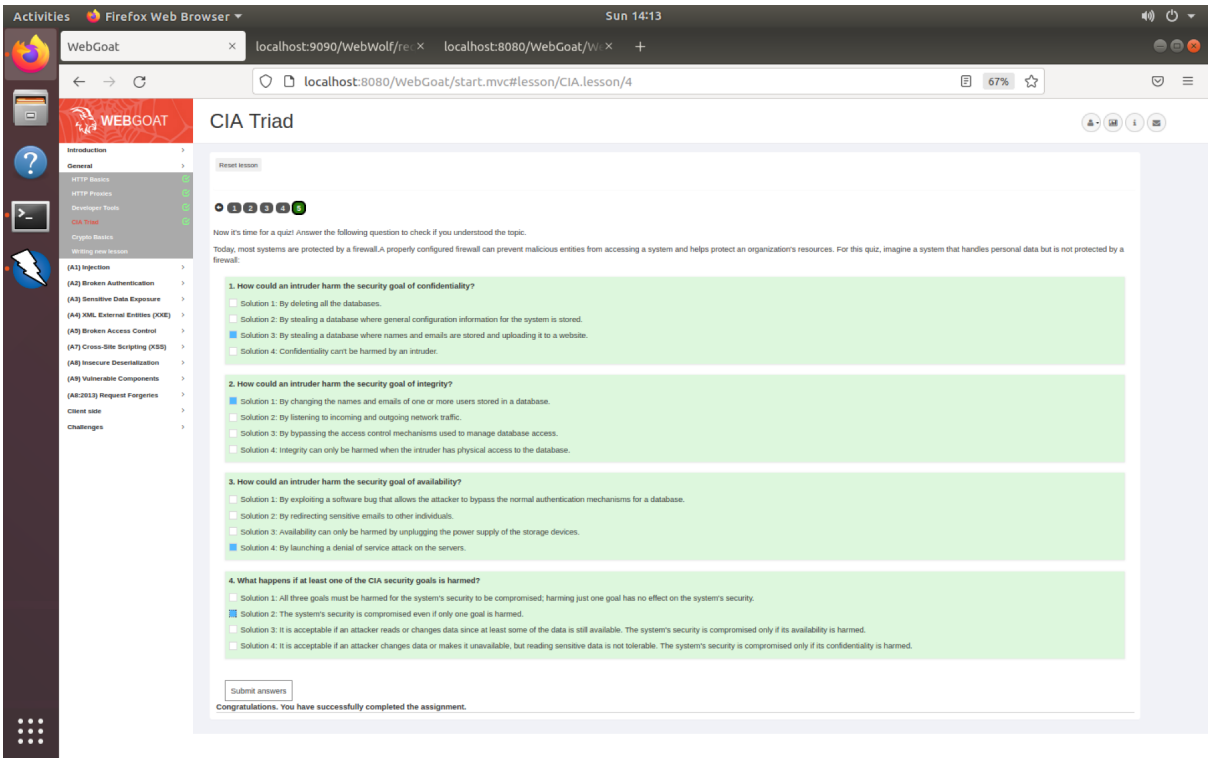
2.2. HTTP Proxies



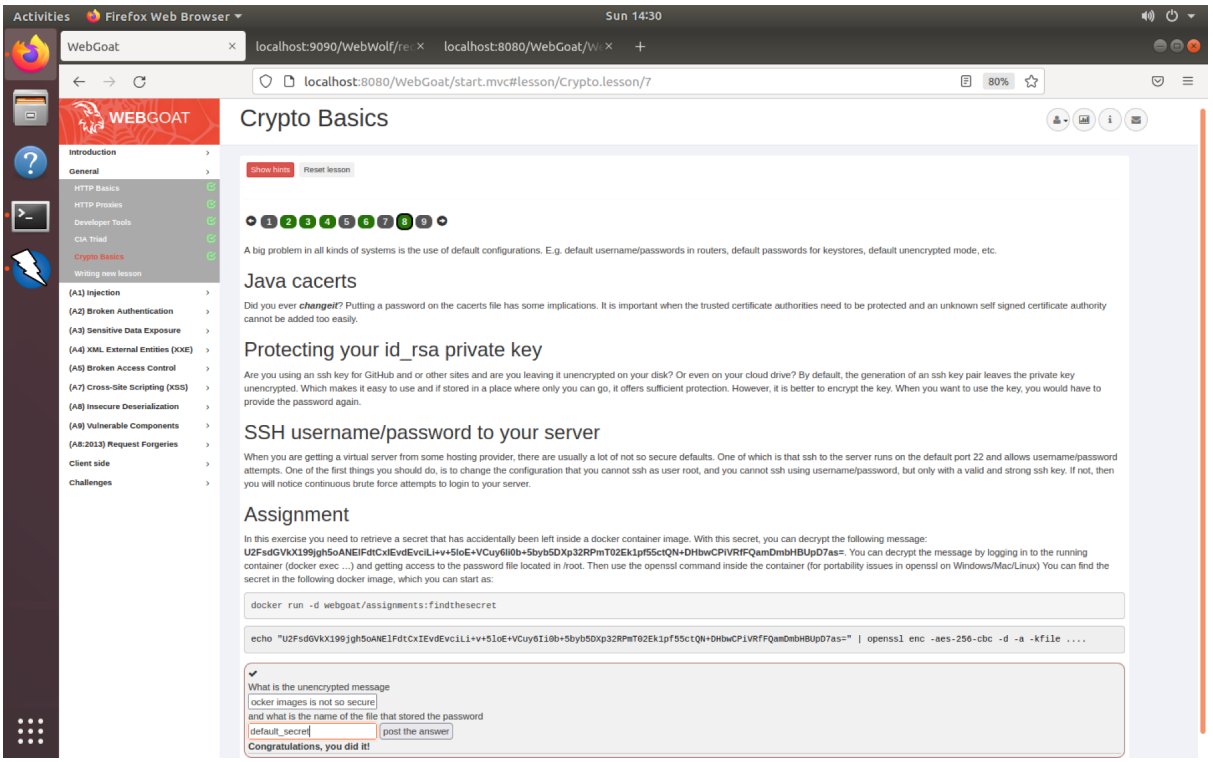
2.3. Developer Tools



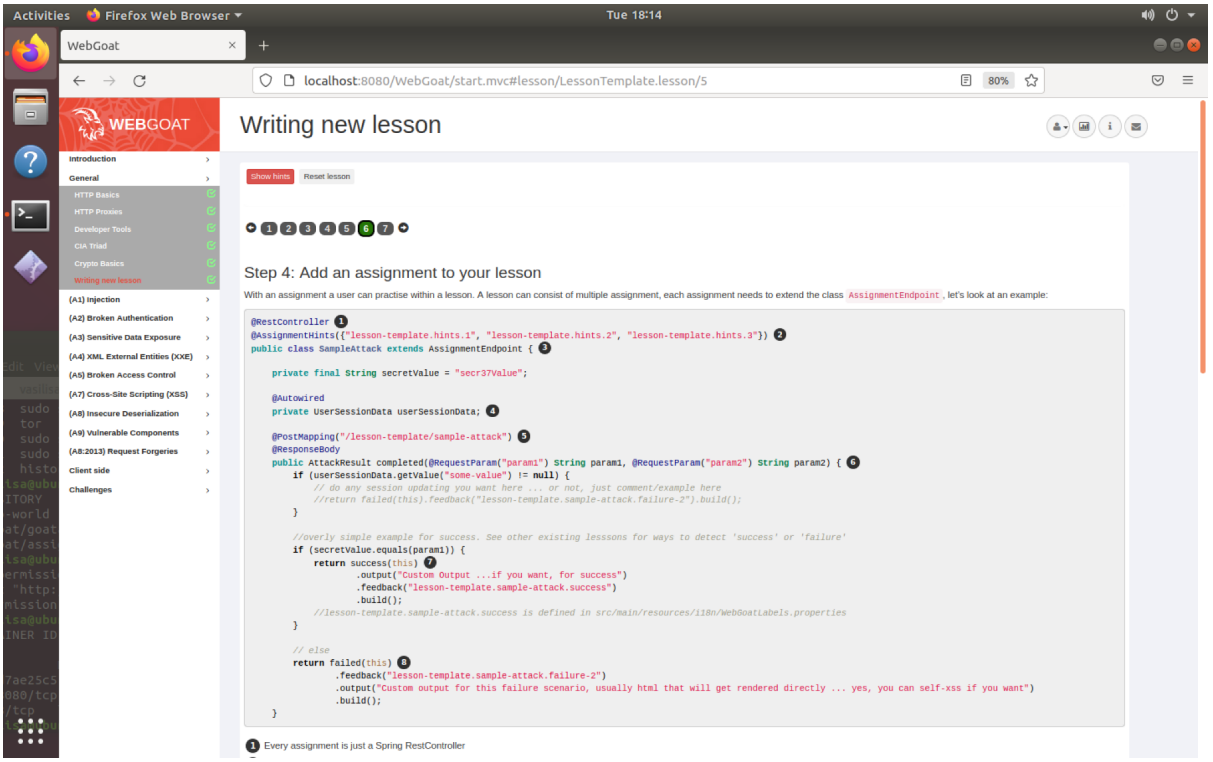
2.4. CIA Triad



2.5. Crypto basics

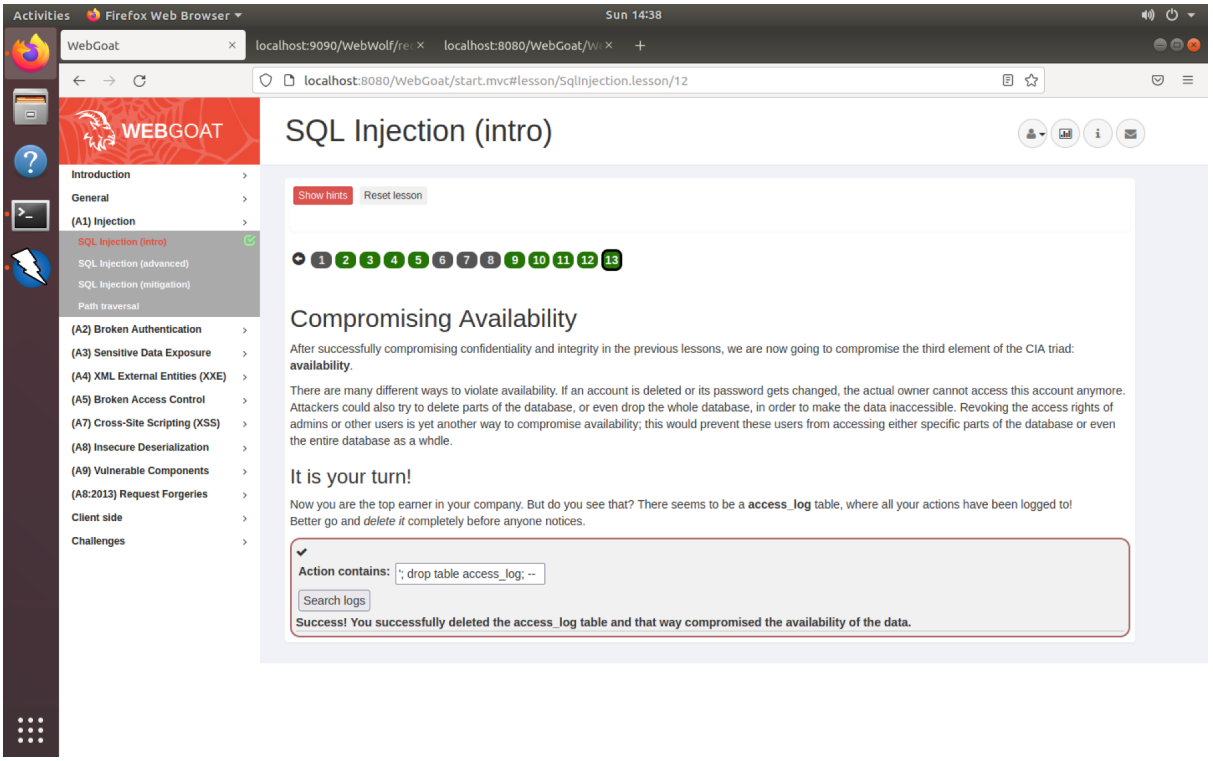


2.6. Writing a new lesson

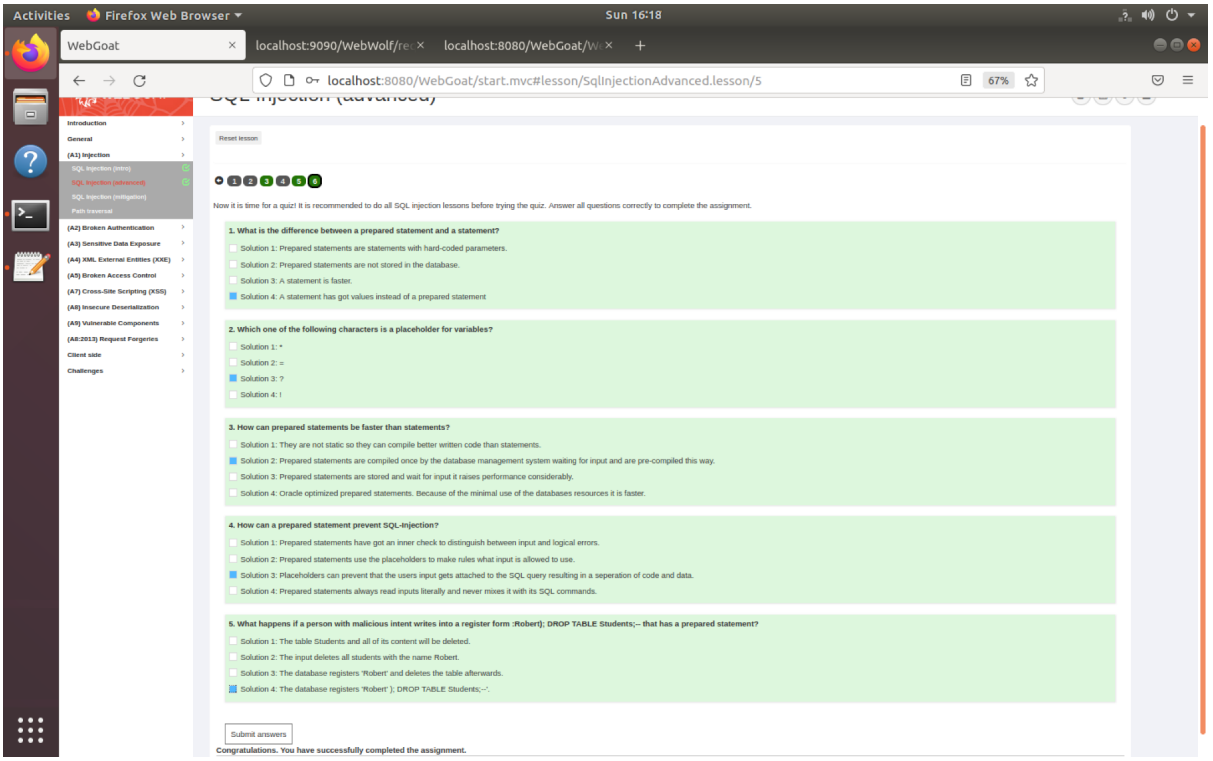


(A1) Injection

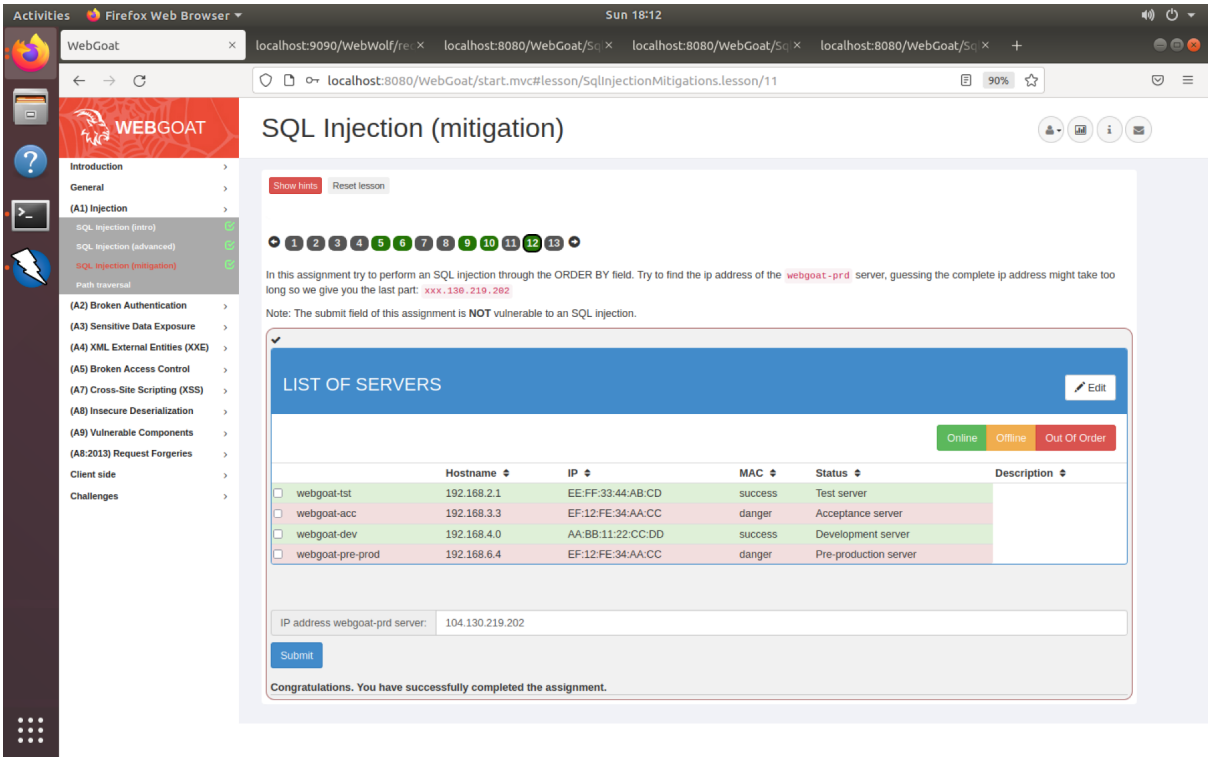
1. SQL Injection (intro)



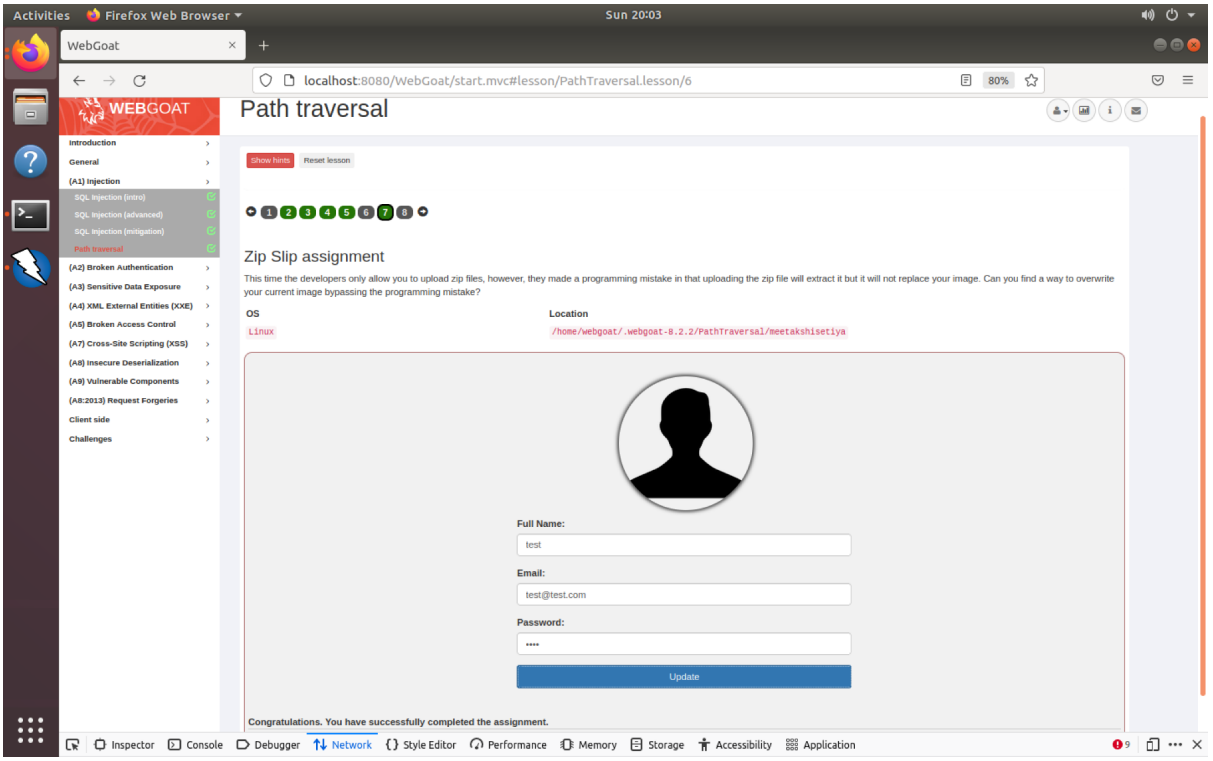
2. SQL Injection (advanced)



3. SQL injection (mitigation)

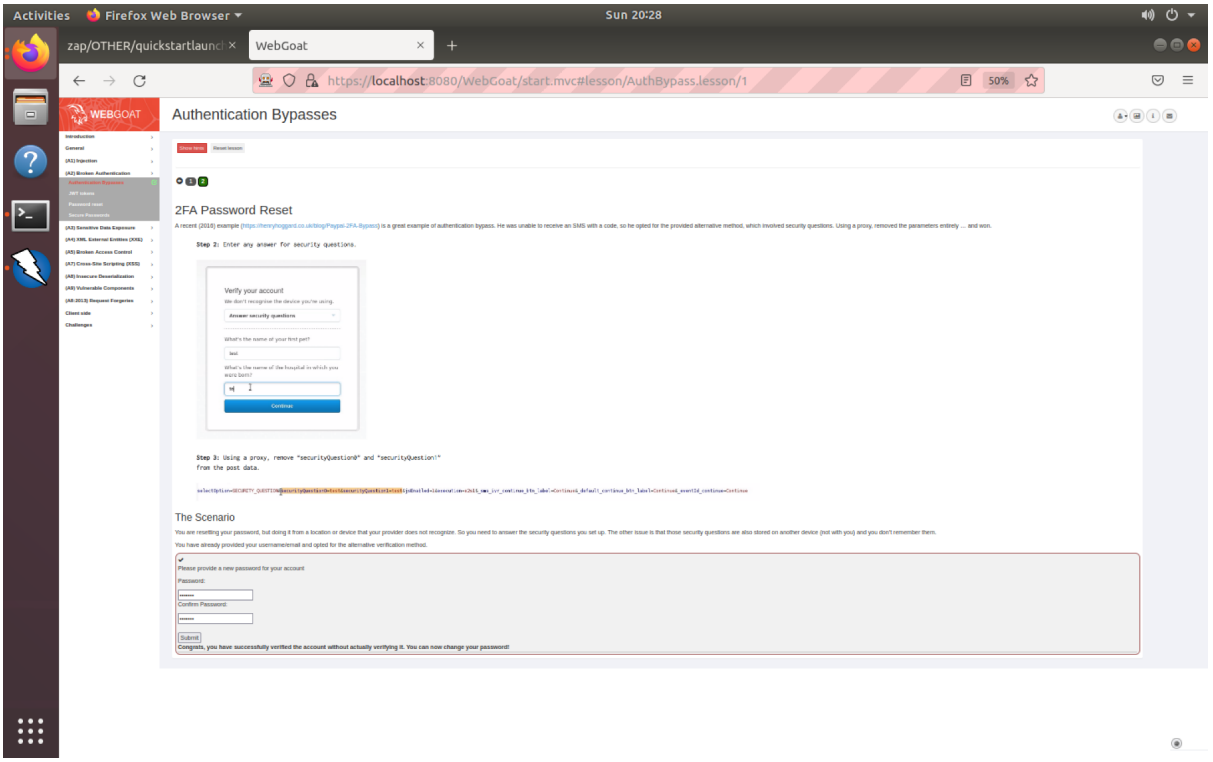


4. Path Traversal

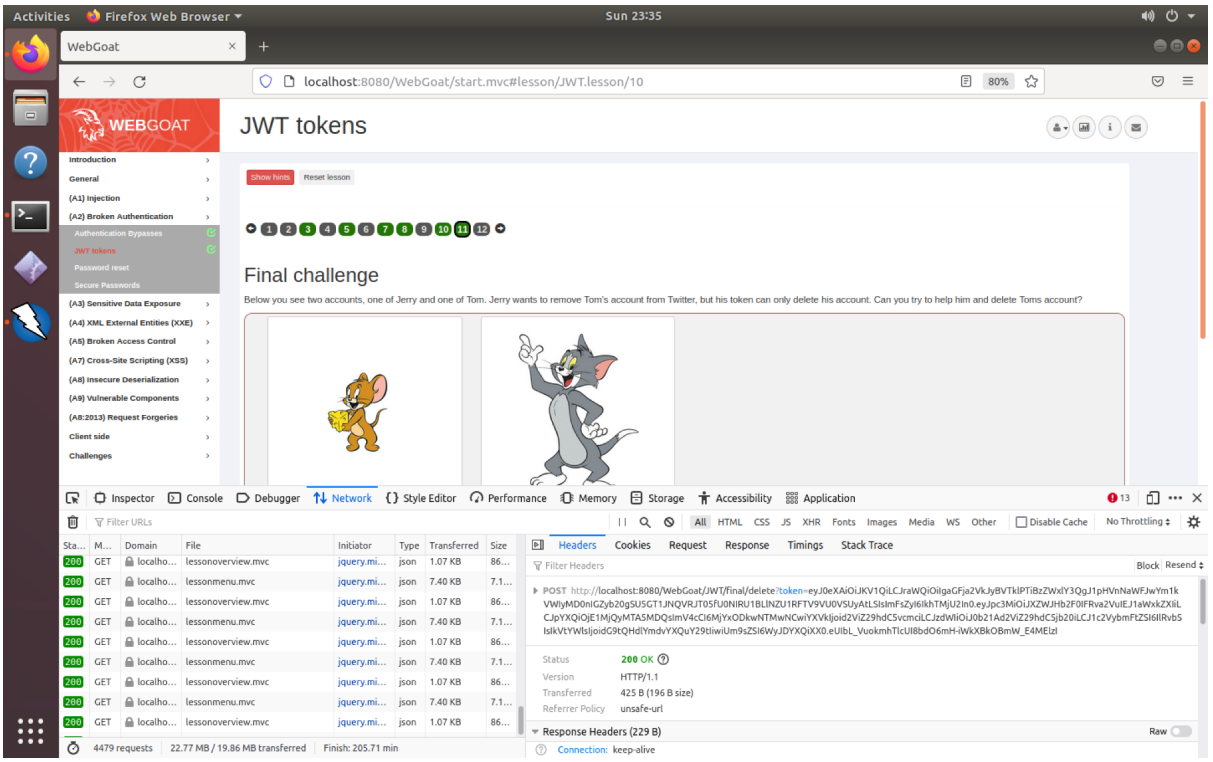


(A2) Broken Authentication

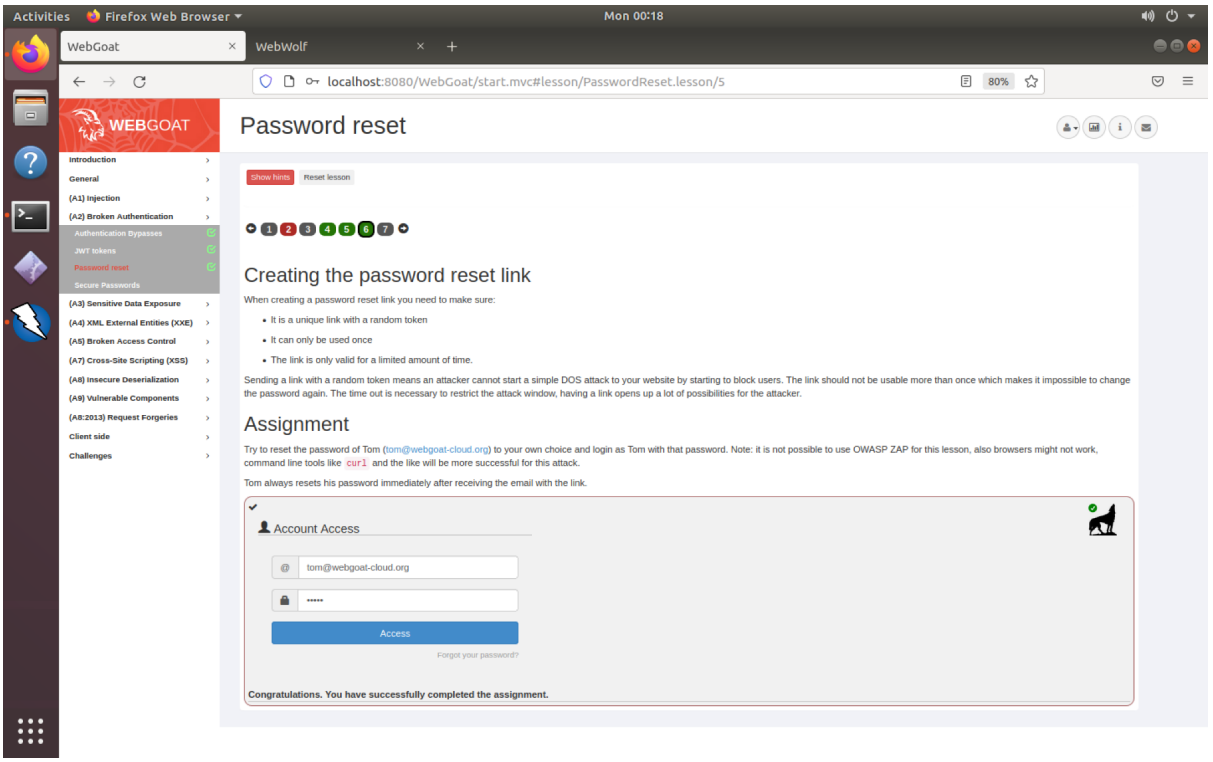
1. Authentication Bypass



2. JWT Tokens



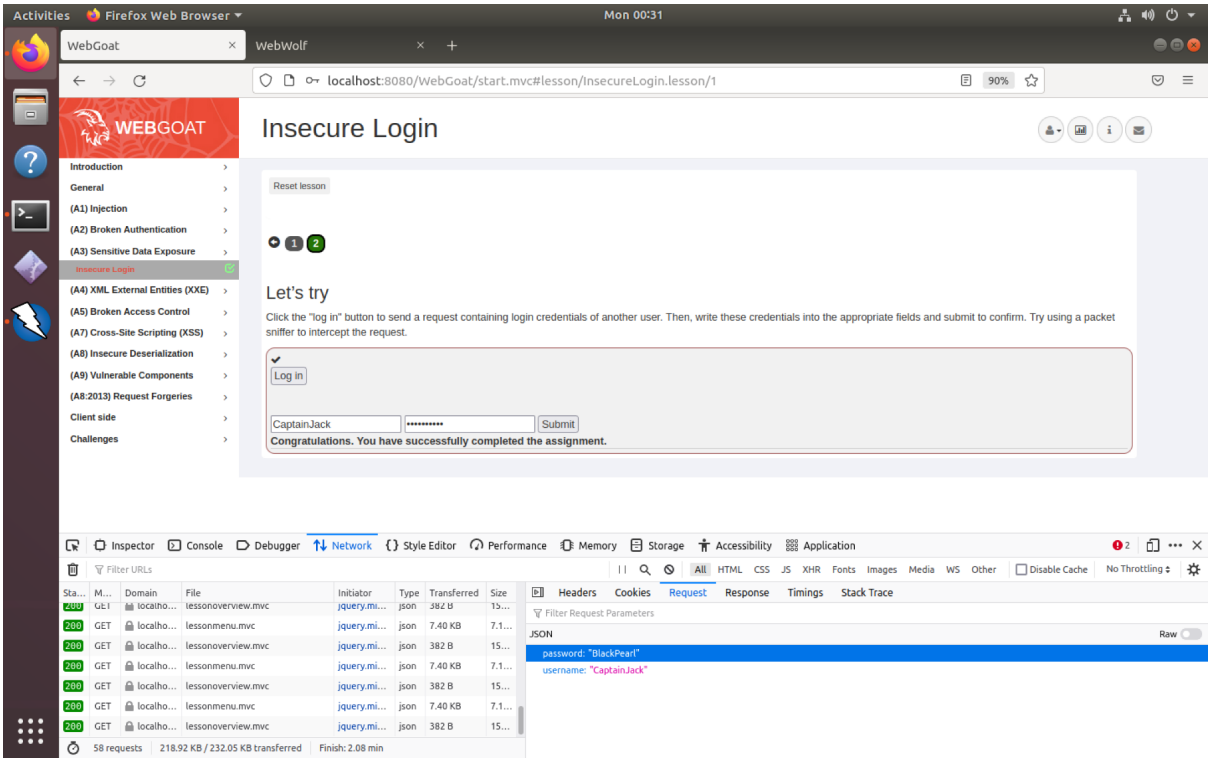
3. Password Reset



4. Secure passwords

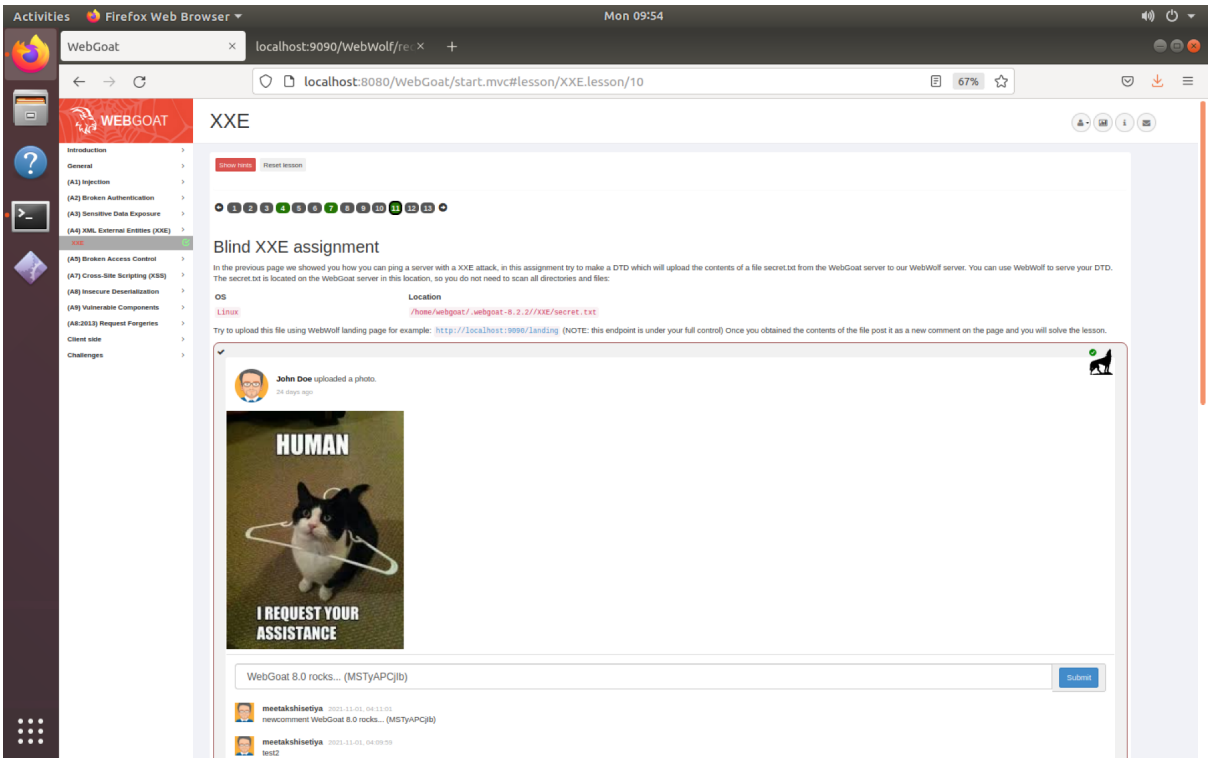
(A3) Sensitive Data Exposure

1. Insecure Login



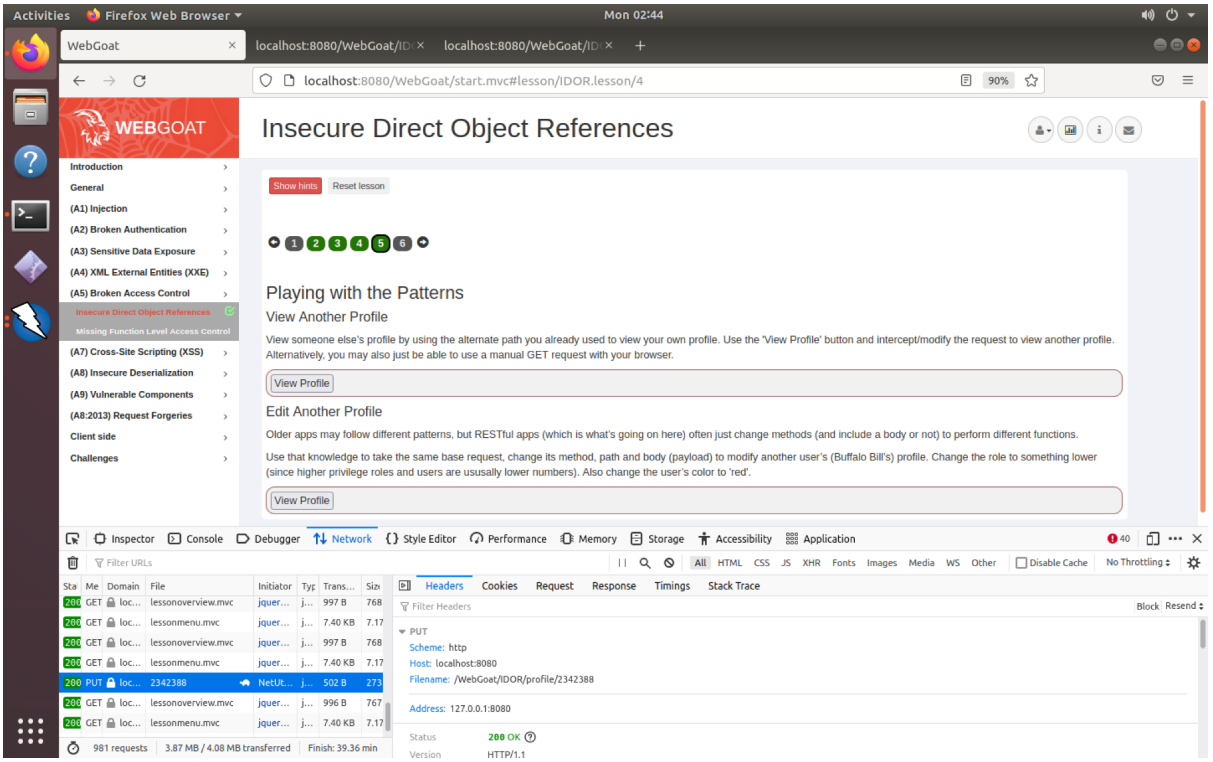
(A4) XML External Entities

1. XXE

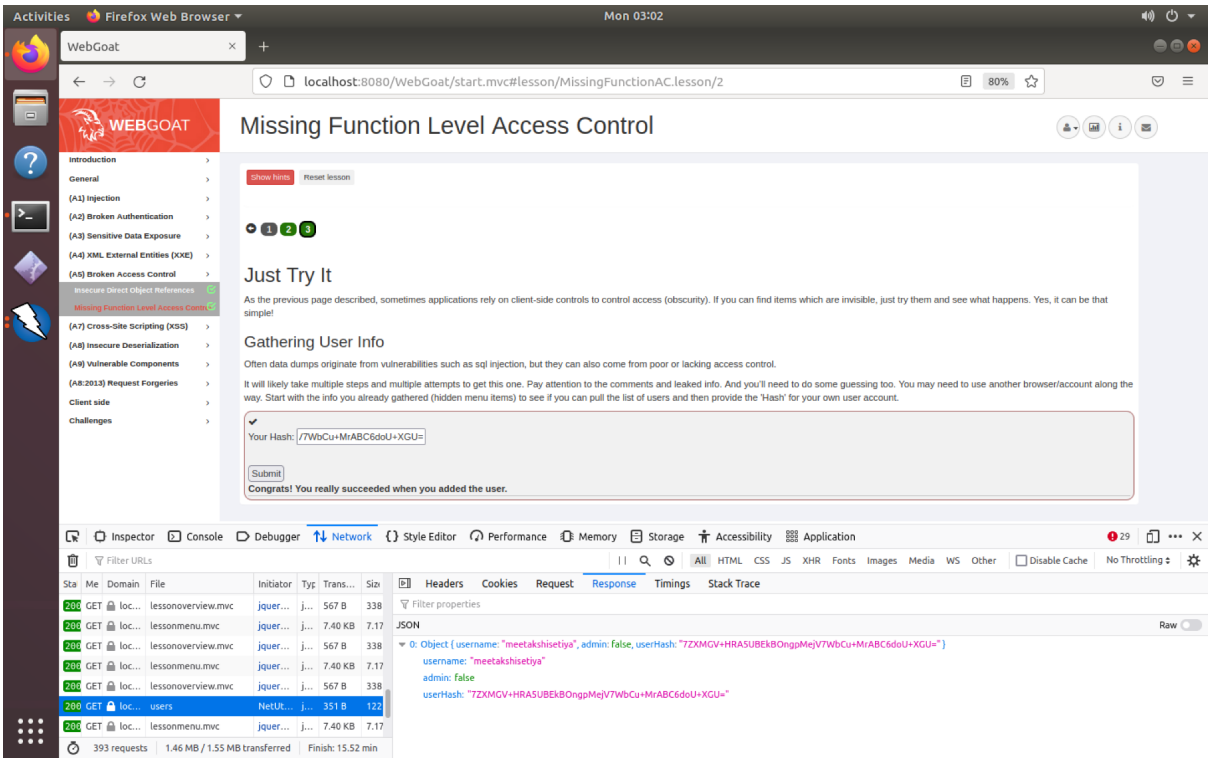


(A5) Broken Access Control

1. Insecure Direct Object References

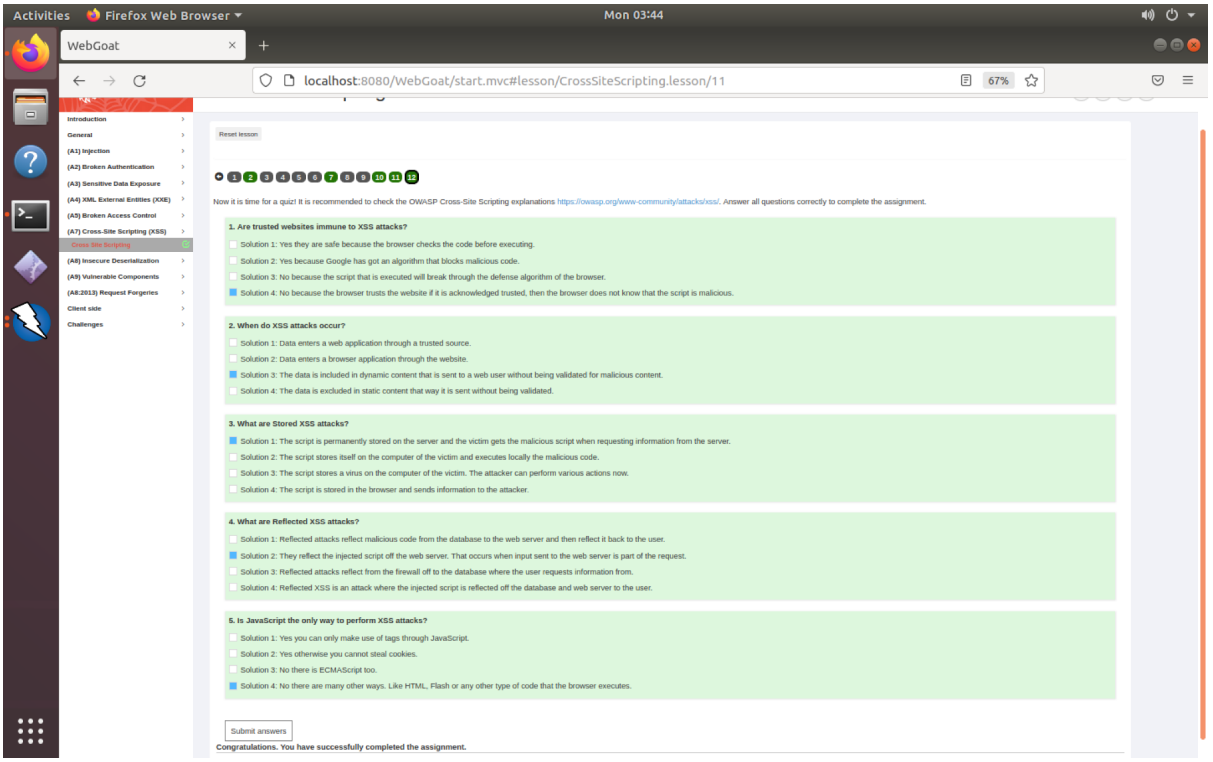


2. Missing Function Level Access Control



(A7) Cross-Site Scripting (XSS)

1. Cross-Site Scripting



(A8) Insecure Deserialization

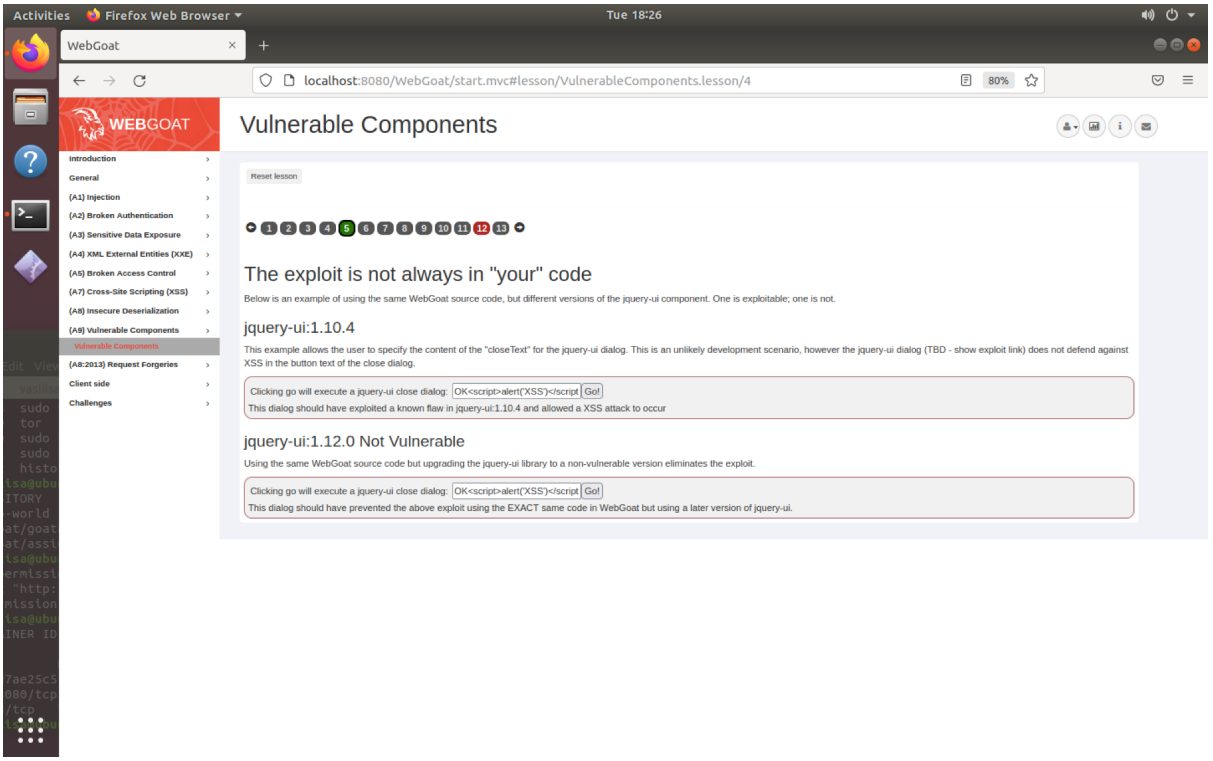
There is an application error in this exercise:

https://github.com/WebGoat/WebGoat/issues/1129

(A9) Vulnerable Components

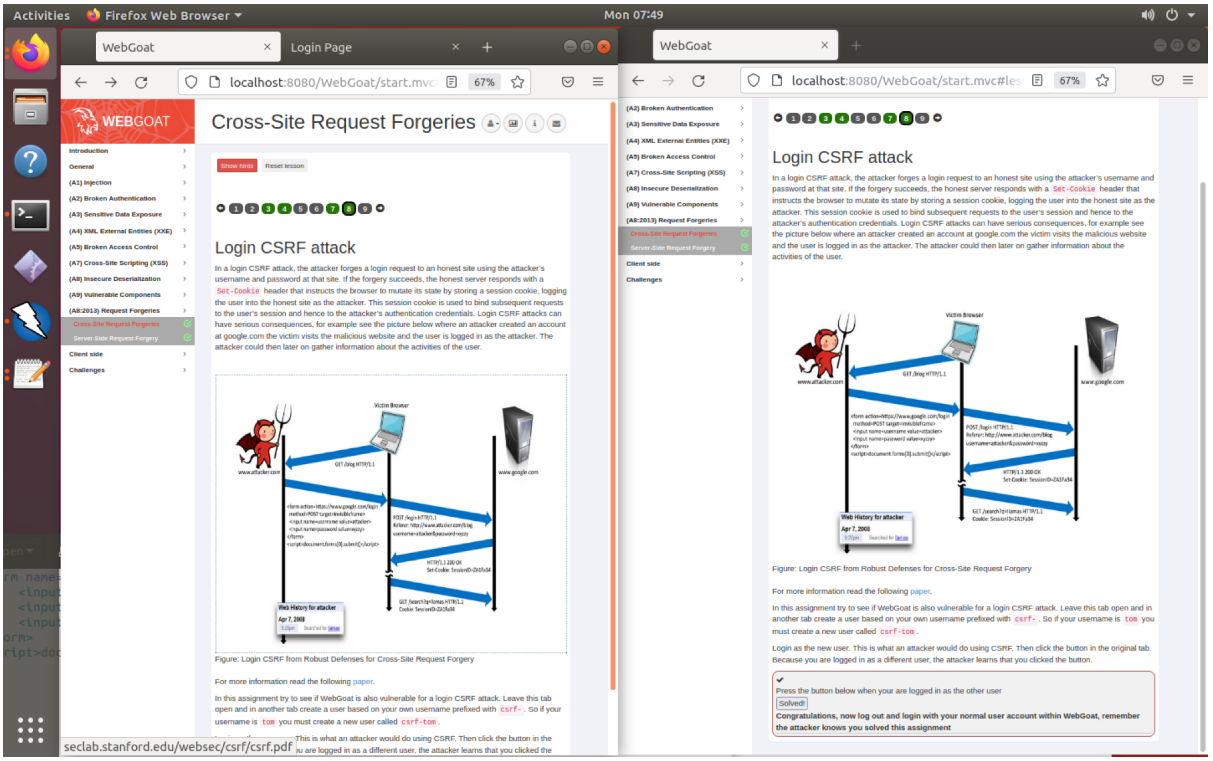
There is an application error in part 12 of this exercise. I just did part 5.

https://github.com/WebGoat/WebGoat/issues/1134

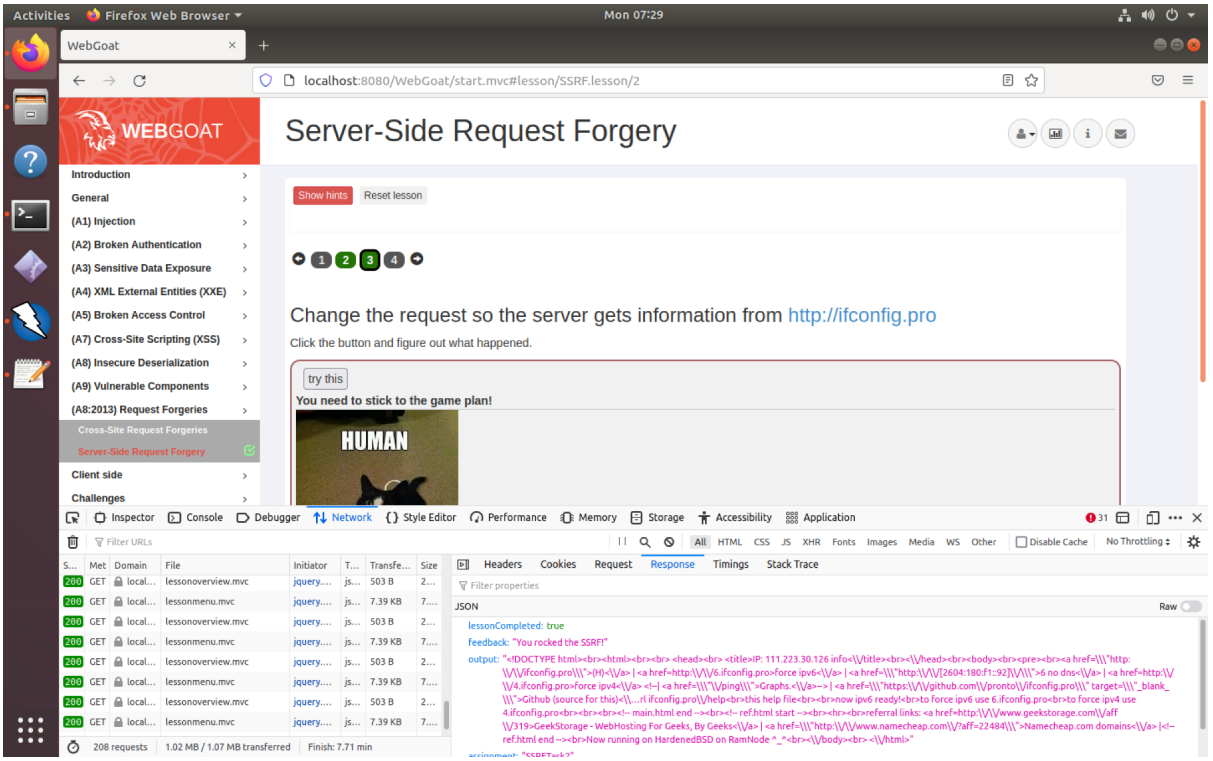


(A8) Request Forgeries

1. Cross-site Request Forgeries

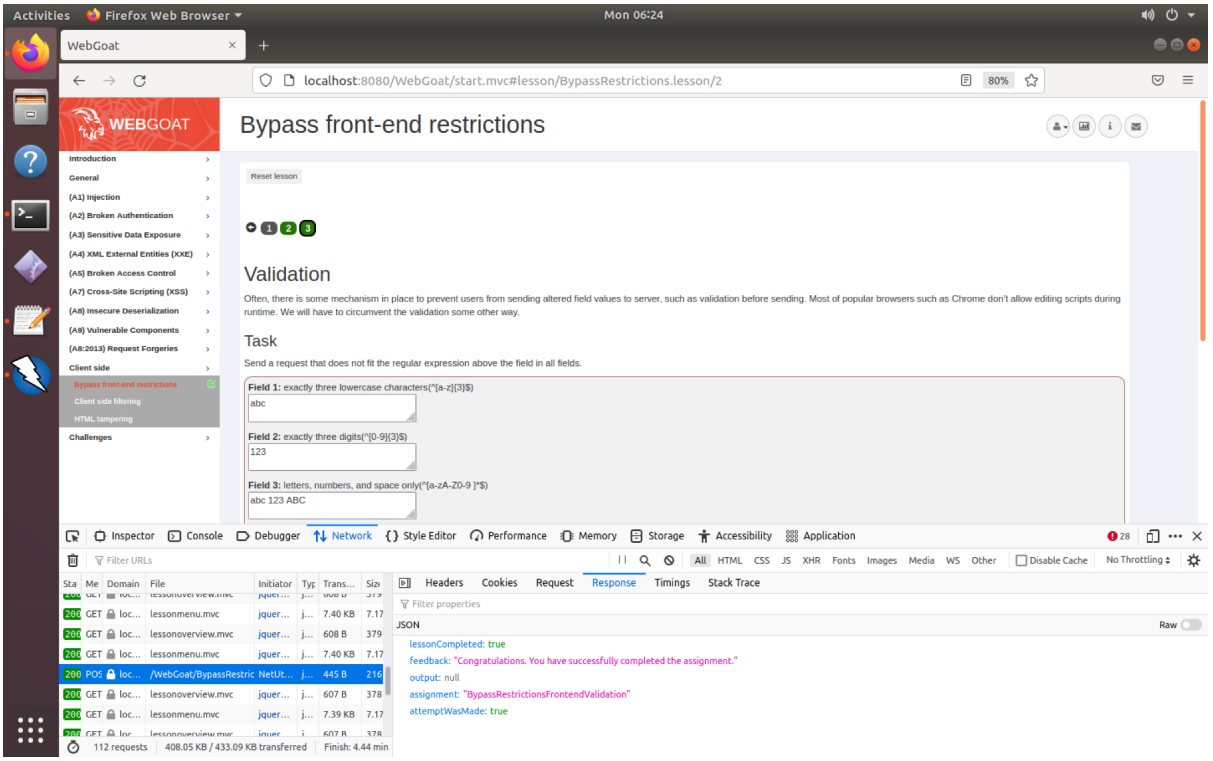


2. Server-side Request Forgeries

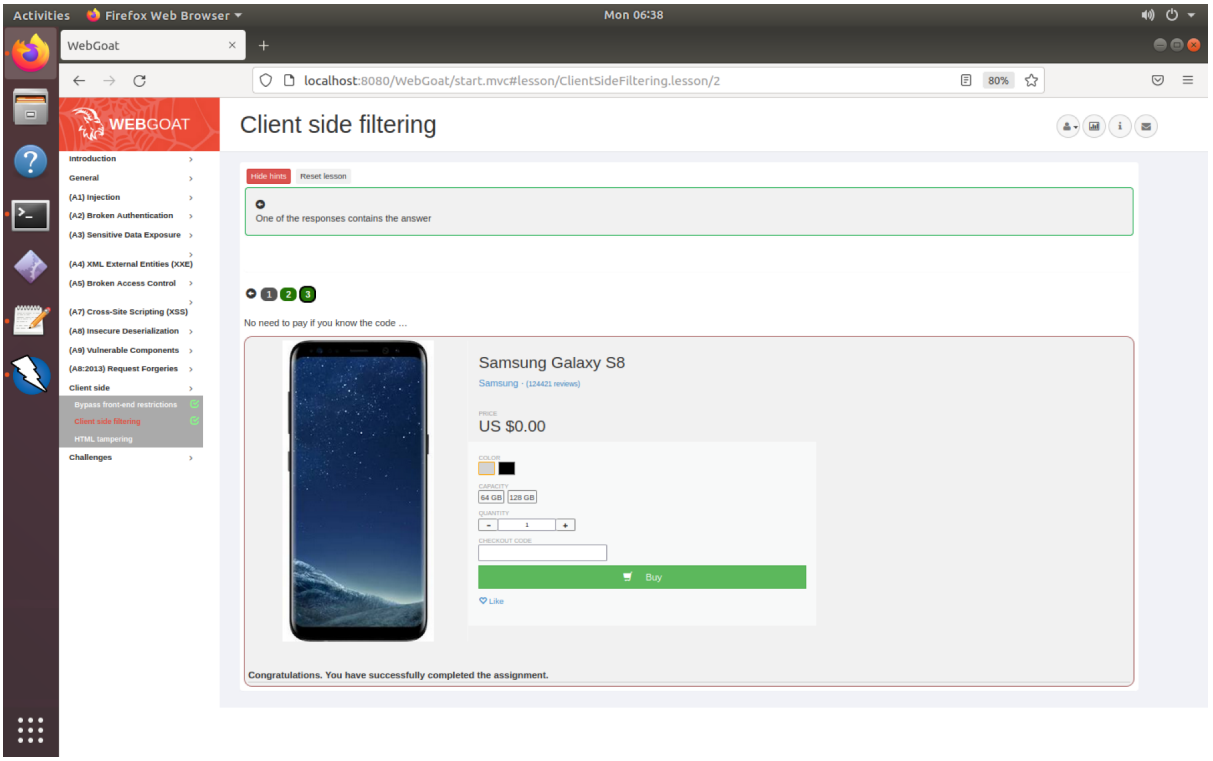


Client Side

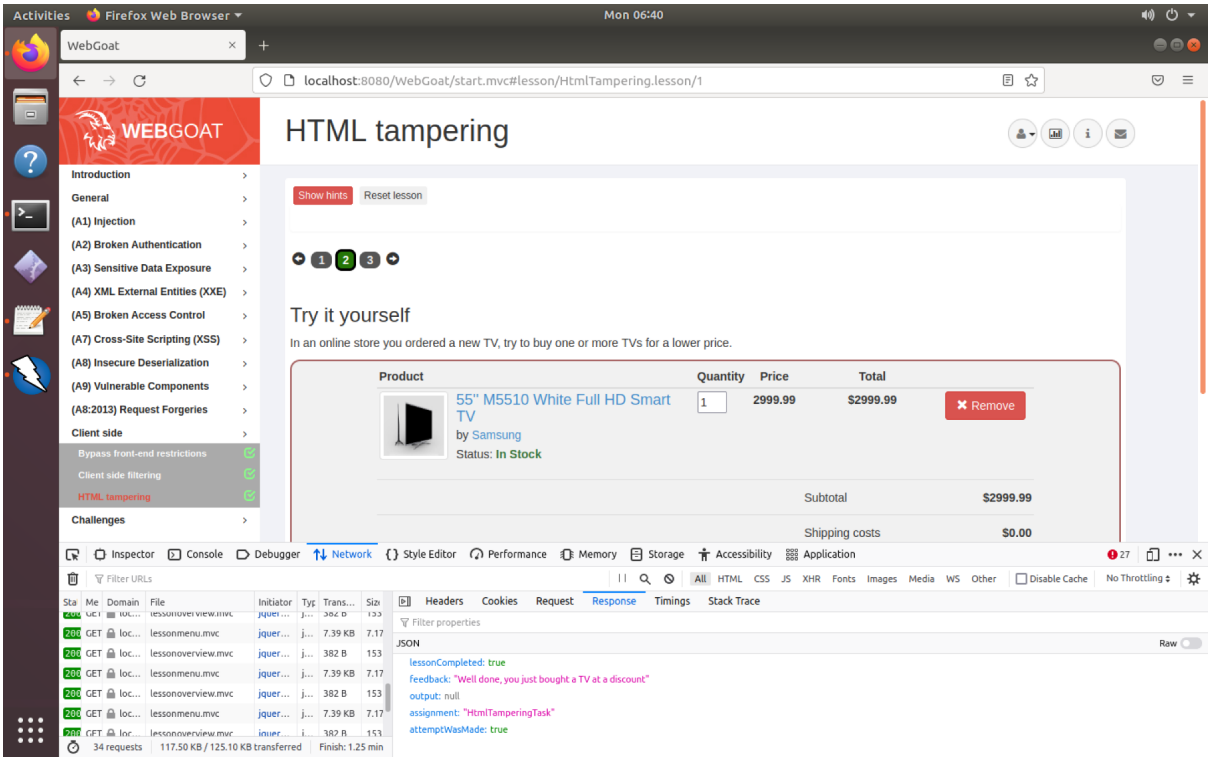
1. Bypass Front End Restrictions



2. Client-Side Filtering

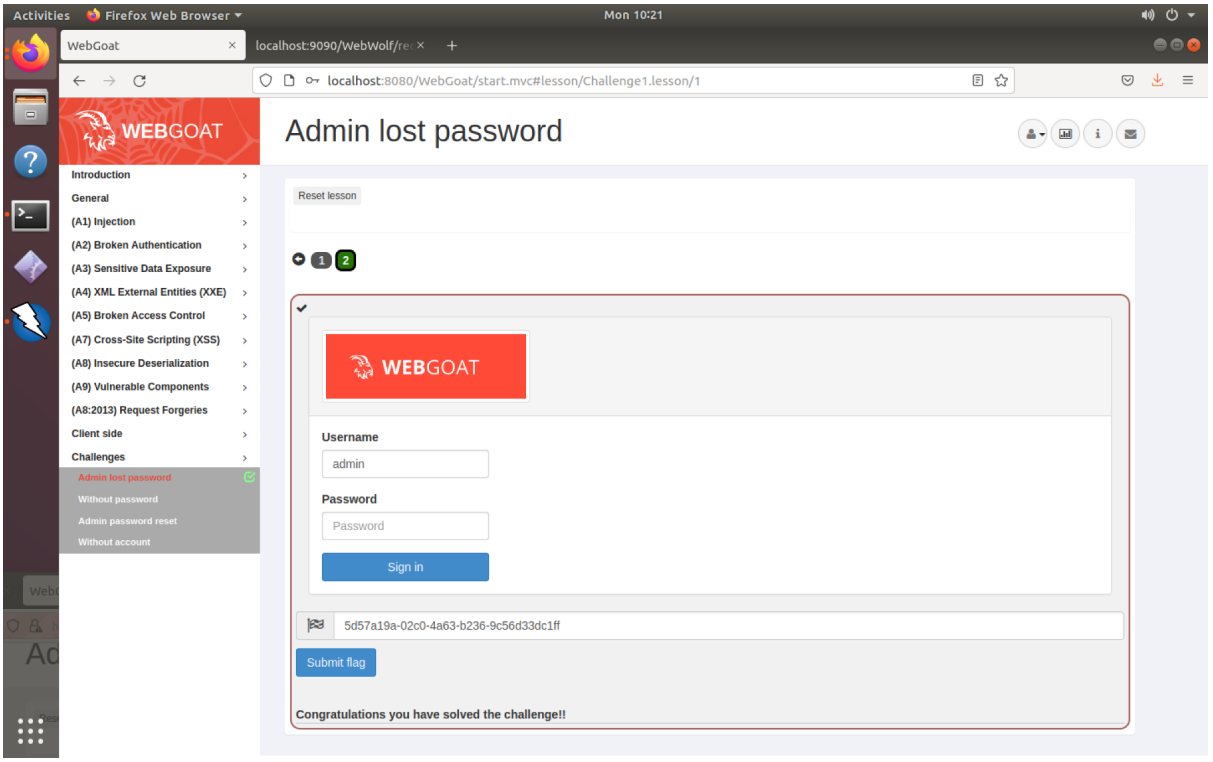


3. HTML Tampering



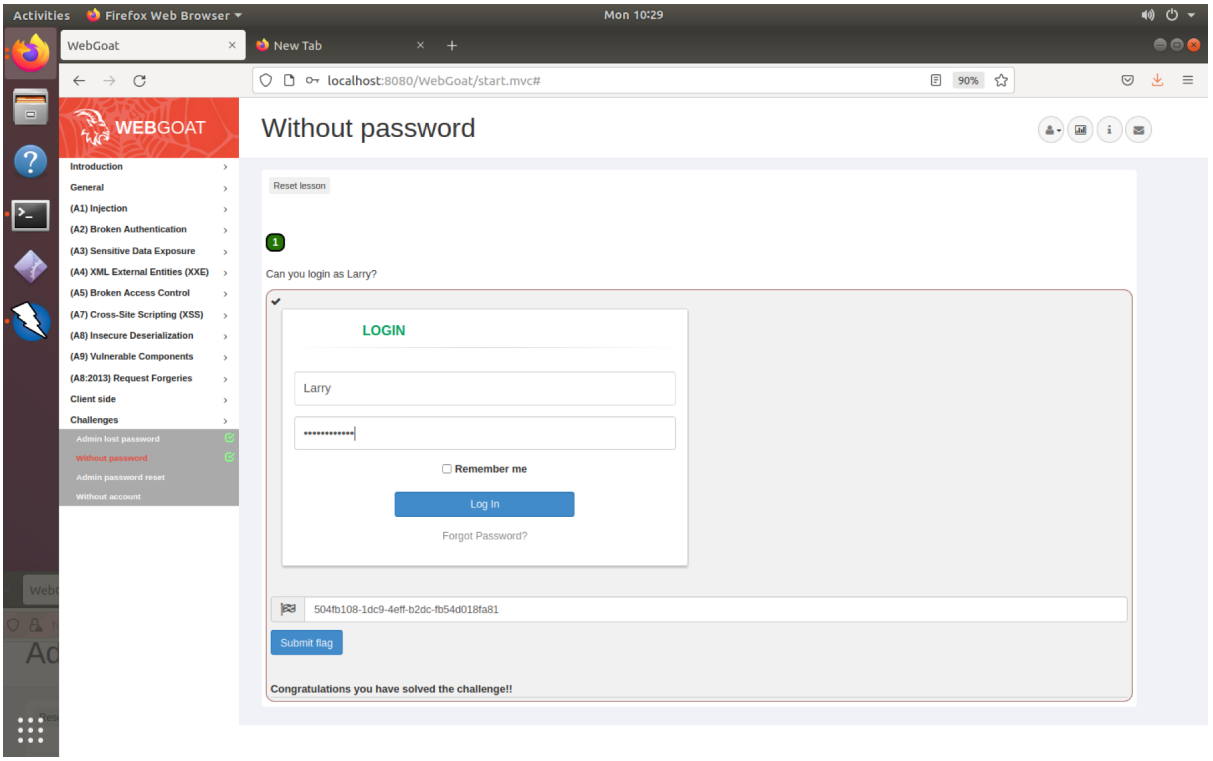
Challenges

1. Admin lost password



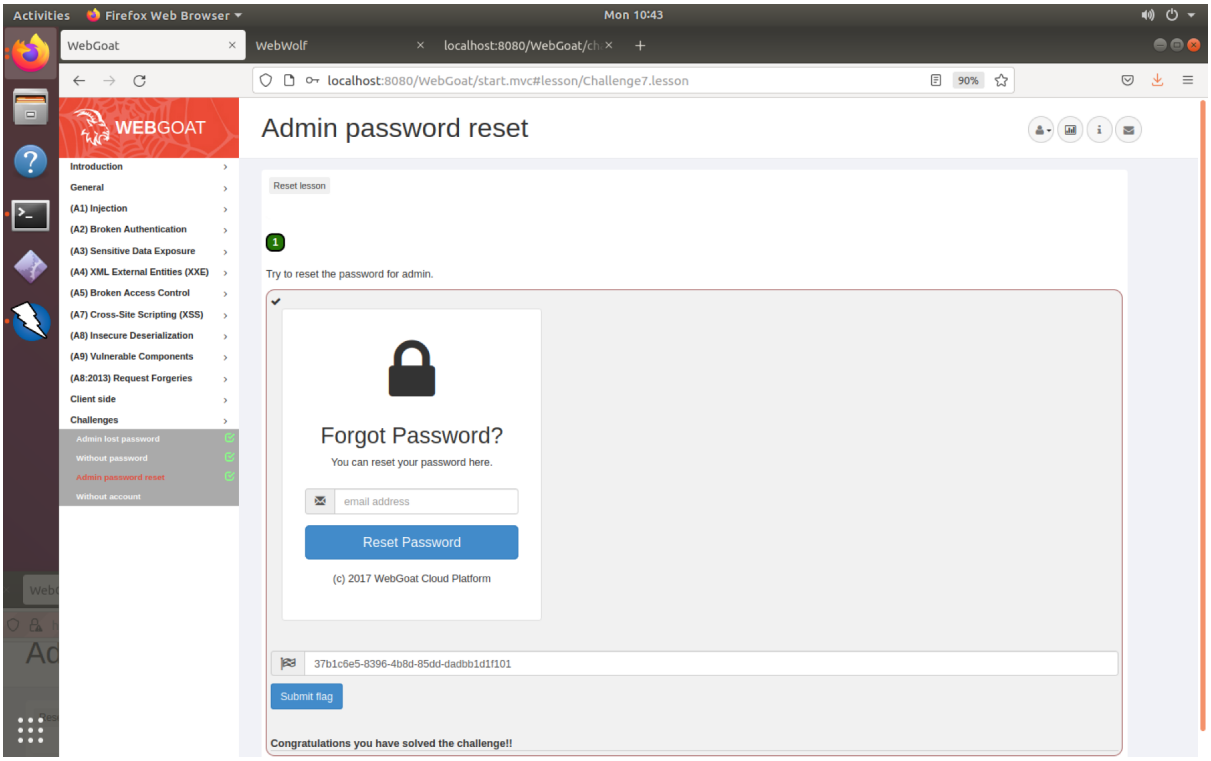
Observation: The password was in the logo file. I downloaded the logo shown and opened it in a text editor.

2. Without password



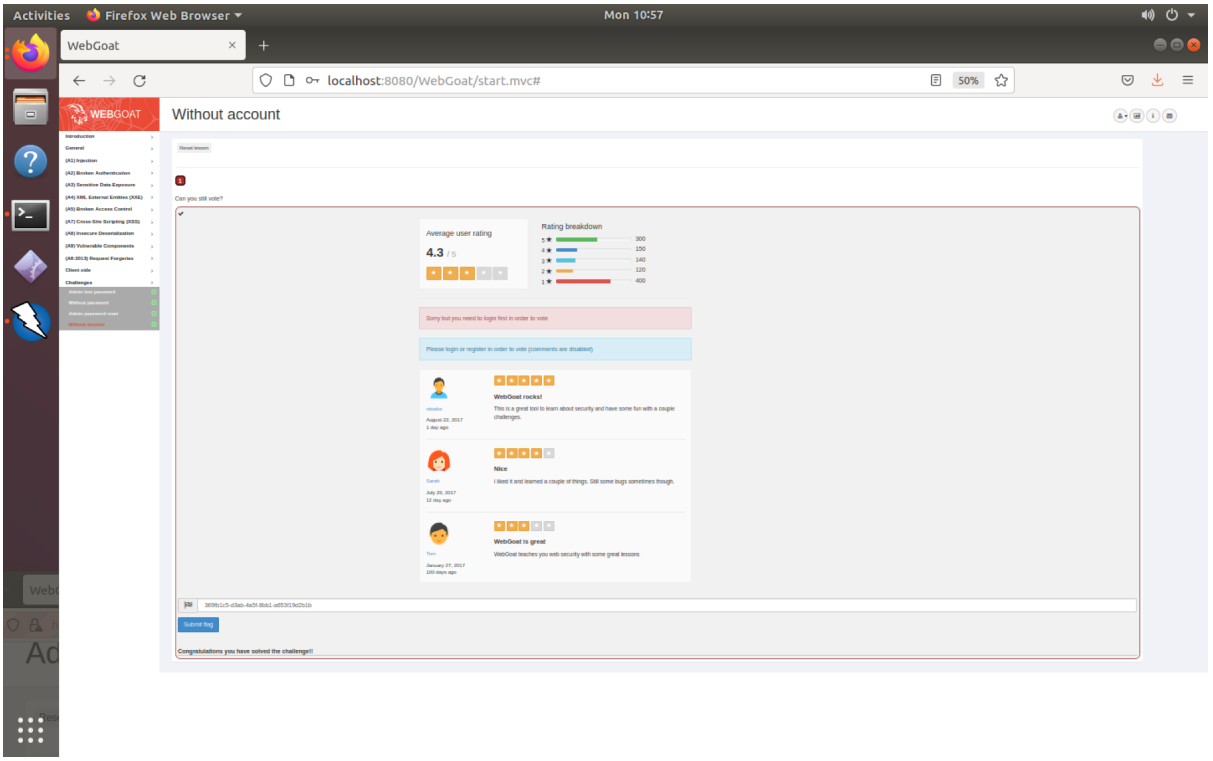
Observation: Sending an invalid request showed the SQL query being used. A simple SQL injection in the password field (p' or '1'='1) completed the assignment.

3. Admin Password reset



Observation: I obtained the admin-password-reset constant token from https://github.com/WebGoat/WebGoat/blob/develop/webgoat-lessons/challenge/src/main/java/org/owasp/webgoat/challenges/SolutionConstants.java
After this, I changed the token of the link in mail box to the admin password reset link and made a new link: http://localhost:8080/WebGoat/challenge/7/reset-password/375afe1104f4a487a73823c50a9292a2

4. Without Account



Observation: An OPTIONS request at the /vote/{num} endpoint showed that only HEAD/GET/OPTIONS request was valid. HEAD request at the /vote/{num} endpoint did the deed.