CSE345/545: Homework 2

Mihir Chaturvedi (2019061)

Question 1

Yes, breaches of encrypted data must be notified by companies. Reasons for this include:

- If the company is using poor encryption standards and unsafe encryptions practices, the unencrypted data can easily be recovered by malicious hackers.
- Many companies poorly store encryption keys, and if hackers find a way to access the encryption key (private key), they can easily decrypt the data.
- Even if there is a breach of encrypted data, there is a breach - which indicates lapses and vulnerabilities in the company's data storage practices and strategies, which customers should be aware of.

Real-world examples of encrypted data breaches include:

- June 2012: League of Legends: Emails, usernames and "encrypted" passwords of more than 32 million registered accounts were stolen. Due to **weak encryption**, the passwords were decrypted and their plaintexts were leaked in 2018. [source, source]
- October 2013: Adobe - encrypted customer credit card records, as well as login data were stolen of more than 3 million users. But since password hints, which were unencrypted plaintext, were also leaked, they revealed a lot of information about the password. [source, source]
- October 2019: GateHub - Email addresses, Encrypted keys, Mnemonic phrases, Passwords of 1.4 million accounts. [source]

Question 2

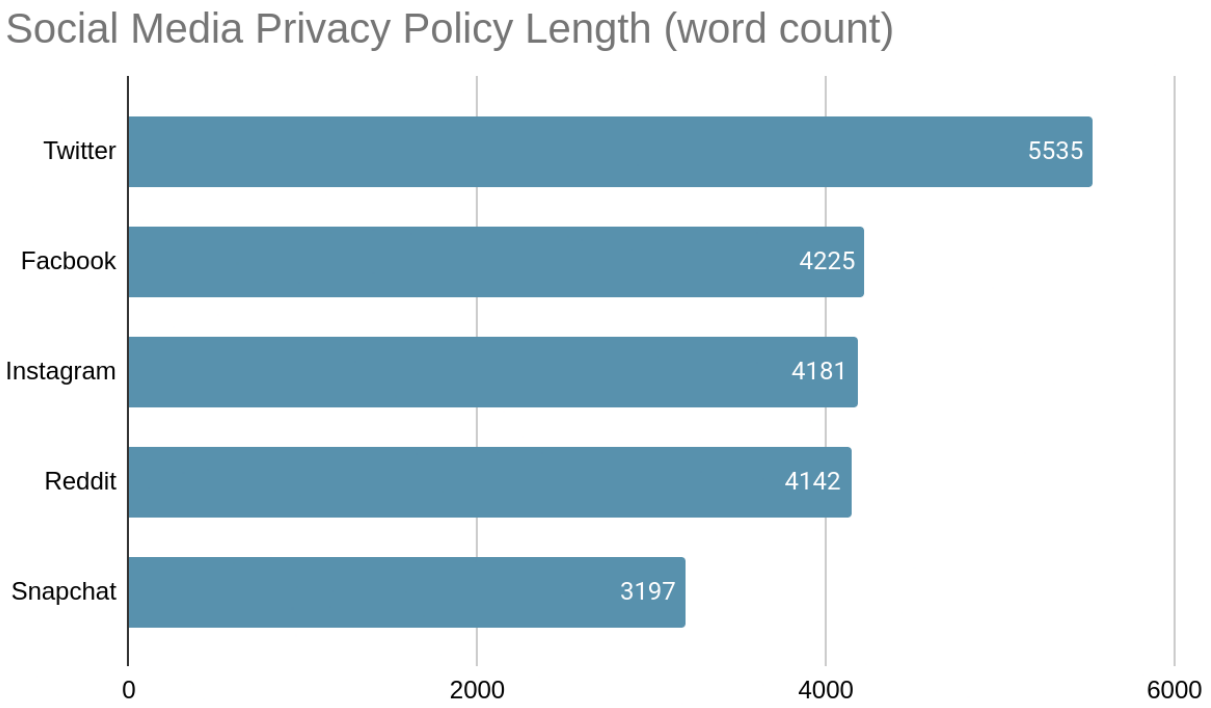
Key differences between Anonymization and Pseudonymization are:

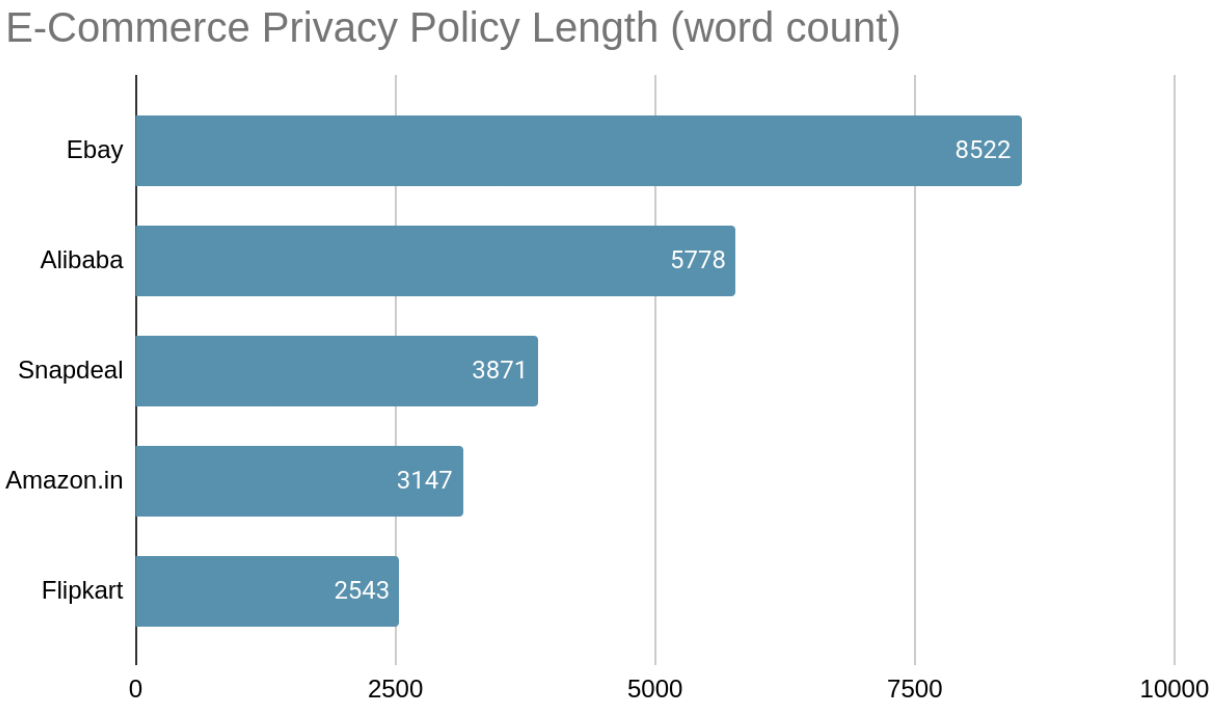
- Anonymization implies the identity of an individual is completely hidden, and it cannot be linked reliably to any particular individual in the real-world.
- Pseudonymization means hiding the real identity of an individual by replacing their name and other identifying information by different values. For example, someone can adopt a pseudonym to prevent others from knowing their real name while doing business.
- With pseudonymization, the activities of the pseudonymized individual can still be tracked, but their real identity still remains hidden.
- With anonymization, one cannot reliably track an individual's activity.

Anonymization: Patient *** *** has a gamma-GT value of 83U/L
Pseudonymization: Patient **Jake Ross** has a gamma-GT value of 83U/L

Question 3

a.


Social Media Privacy Policy Length (word count)
Twitter
5535
Facbook
4225
Instagram
4181
Reddit
4142
Snapchat
3197
0
2000
4000
6000


E-Commerce Privacy Policy Length (word count)
Ebay
8522
Alibaba
5778
Snapdeal
3871
Amazon.in
3147
Flipkart
2543
0
2500
5000
7500
10000

b.

Website	Comments on Privacy Policy
Twitter	• Presented in a readable and accessible website-type format with slides • Important points are highlighted and links are present • Sentences are short and crisp
Facebook	• Present in a readable webpage format with clickable sections • Long sentences with legal jargon
Instagram	• Long, text heavy document • Slightly difficult to read with long sentences • Has many links
Reddit	• Most information present in tabular format for quick readability • Still contains complex language
Snapchat	• Relatively short privacy policy and sentence lengths • Clear descriptive text and easy to read
Ebay	• Very unreadable and a small font • Classic legal jargon used
Alibaba	• Long but clear language • Neatly presented
Snapdeal	• Long wall of text, not great to read • Point-wise info helps a bit
Amazon.in	• Long but clearer language with descriptive headings • Point-wise info helps a bit
Flipkart	• Short but uneasy to read • Legal jargon

c. For the websites above, we note:

i. All privacy policies and notices are very long and detailed, as expected

ii. Most of the privacy policies are filled with legal-heavy vocabulary that is often meant to be properly understood by lawyers only.

iii. Most websites do not display the important points in prominence, and such points are hidden in the mess of the text.

iv. In general, the Social Media websites' policies are easier to read and better presented than the E-commerce websites.

v. Companies use long and convoluted sentences to reduce the chance of regular users reading them and realising the amount of information being stored and tracked, while also getting legal protection for the same.

Question 4

a.

```
~/Desktop
❯ exiftool 4_img.jpg | grep GPS
GPS Version ID                  : 2.3.0.0
GPS Latitude Ref                : South
GPS Longitude Ref               : East
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
```

- Longitude: 174 deg 46' 32.89" E
- Latitude: 36 deg 50' 30.18" S
- Position: 36°50'30.2"S 174°46'32.9"E
- City: Auckland
- Country: New Zealand
- Location: Ports of Auckland

b. Coordinates of IIIT Delhi: 28.545563, 77.273062

```
~/Desktop
❯ exiftool -GPSLongitude="77.273062"  -GPSLatitude="28.545563"  -GPSLongitudeRef="East" -GPSLatitudeRef="North" 4_img.jpg
    1 image files updated

~/Desktop
❯ exiftool 4_img.jpg | grep GPS
GPS Version ID                  : 2.3.0.0
GPS Latitude                    : 28 deg 32' 44.03" N
GPS Longitude                   : 77 deg 16' 23.02" E
GPS Latitude Ref                : North
GPS Longitude Ref               : East
GPS Position                    : 28 deg 32' 44.03" N, 77 deg 16' 23.02" E
```

c. MD5 checksum of original image: 50bb962c9c4dbd36efcc06a5c9c6462d
MD5 checksum of image after (b): 41b609f84f7889af301b250ae554b979

```
~/Desktop
❯ md5sum 4_img_original.jpg
50bb962c9c4dbd36efcc06a5c9c6462d  4_img_original.jpg

~/Desktop
❯ md5sum 4_img.jpg
41b609f84f7889af301b250ae554b979  4_img.jpg
```

GPS location data is stored in the metadata portion of the image file. When the md5 checksum is calculated, the entirety of the file is taken as input as a binary and its checksum is outputted. Thus, changing the metadata will change the md5 checksum.

d. No, the extension of a file cannot be trusted to correctly represent the real file type of the contents. The extension of a file is a file-system level feature enabled by the OS that provides a convenient way to assess the possible file type. It also allows applications to provide “sorting” mechanisms by extension (and thereby filetype).

   But, the real file type is determined by applications using the “magic number” present in the metadata of the file. Each file format has a different, unique and identifying “magic number” that corresponds to the file format. Along with the magic number, the metadata also contains the file mime type that contains information about the type of file (image, document, etc.) and format of the type (jpeg, png, etc.).

   As we can see here, changing the extension to `png` has not affected the file itself.

```
~/Desktop
❯ mv 4_img.jpg 4_img.png

~/Desktop
❯ exiftool 4_img.png
ExifTool Version Number         : 11.88
File Name                       : 4_img.png
Directory                       : .
File Size                       : 86 kB
File Modification Date/Time     : 2021:10:24 16:35:51+05:30
File Access Date/Time           : 2021:10:24 16:43:24+05:30
File Inode Change Date/Time     : 2021:10:24 16:43:23+05:30
File Permissions                : rw-rw-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
XMP Toolkit                     : Image::ExifTool 11.88
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                      : 719×496
Megapixels                      : 0.357
GPS Latitude                    : 28 deg 32' 44.03" N
GPS Longitude                   : 77 deg 16' 23.02" E
GPS Latitude Ref                : North
GPS Longitude Ref               : East
GPS Position                    : 28 deg 32' 44.03" N, 77 deg 16' 23.02" E
```

Question 5

To resolve a hostname to its destination, we have two options:

1. Directly resolve the hostname and connect to the server using Tor
2. Ask Tor to resolve the hostname, and continue connection to the server using Tor

We must always use option 2: request DNS resolution over Tor. Reasons:

- With option 1, directly requesting for a resolution from a DNS server will expose our IP address to the resolver.
- With option 1, if our client is not using DNS-over-HTTPS, the hostname that we are requesting a resolution for will be visible to our ISP and the DNS resolver.
- With option 2, a Tor exit node will perform the DNS resolution for us and supply us with the resolved server details. Since a Tor node is performing the resolution, our IP address will not be compromised.

Question 6

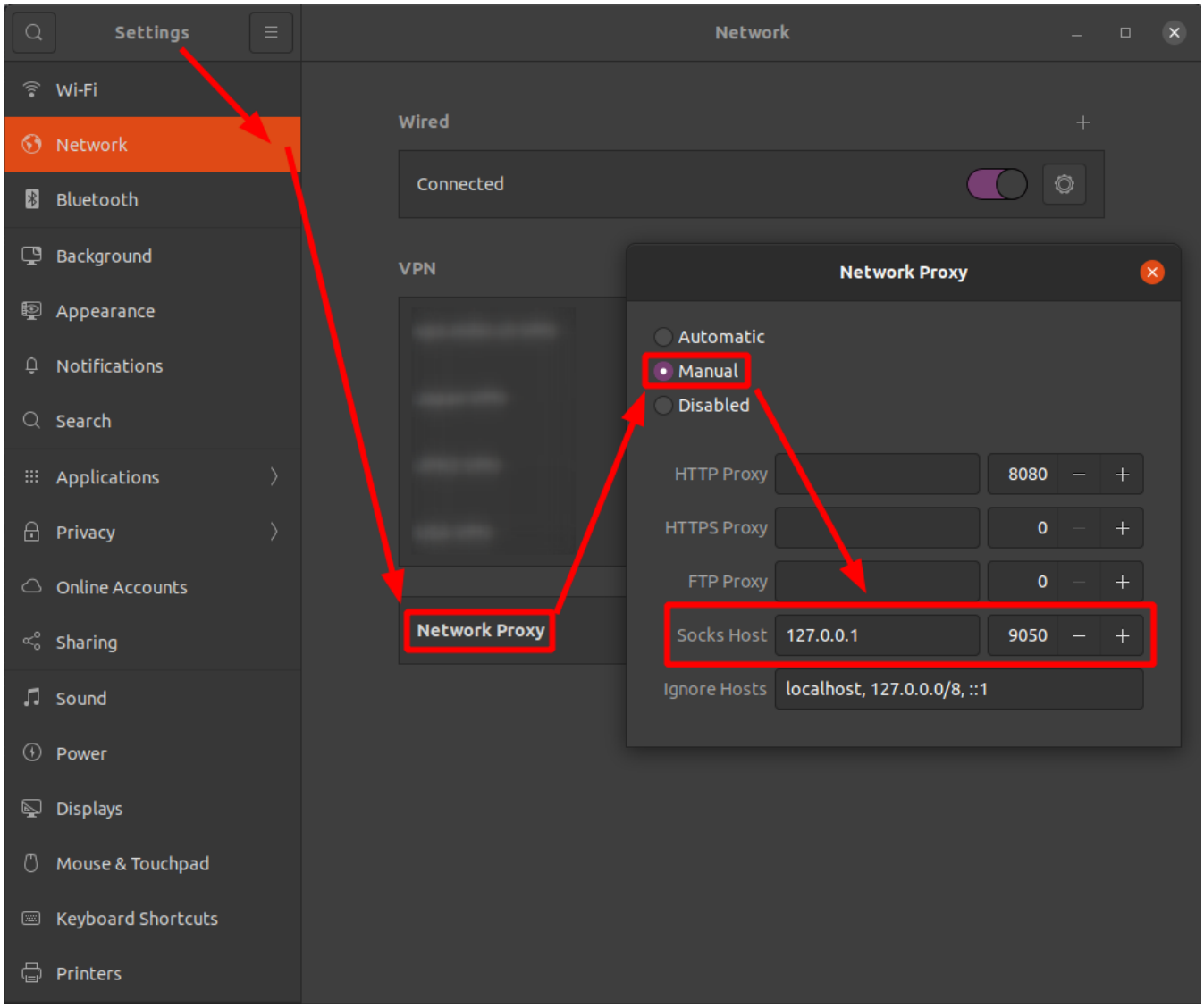
1. Install the `tor` package from `apt`: `$ sudo apt install tor`.
   Note that the `torsocks` package is installed alongside it.

```
~ 13s
❯ sudo apt install tor
Reading package lists ... Done
Building dependency tree
Reading state information ... Done
The following packages were automatically installed and are no longer required:
  intltool libllvm11 libllvm11:i386 libqt5designer5 libqt5help5 libqt5test5 libxdamage1:i386
  python3-distutils-extra python3-pyqt5 python3-sip shim
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  tor-geoipdb torsocks
Suggested packages:
  mixmaster torbrowser-launcher socat tor-arm apparmor-utils obfs4proxy
The following NEW packages will be installed:
  tor tor-geoipdb torsocks
0 upgraded, 3 newly installed, 0 to remove and 110 not upgraded.
Need to get 2,439 kB of archives.
After this operation, 13.5 MB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 tor amd64 0.4.2.7-1 [1,410 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 torsocks amd64 2.3.0-2 [61.5 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 tor-geoipdb all 0.4.2.7-1 [968 kB]
Fetched 2,439 kB in 0s (8,403 kB/s)
Selecting previously unselected package tor.
(Reading database ... 447869 files and directories currently installed.)
Preparing to unpack .../tor_0.4.2.7-1_amd64.deb ...
Unpacking tor (0.4.2.7-1) ...
Selecting previously unselected package torsocks.
Preparing to unpack .../torsocks_2.3.0-2_amd64.deb ...
Unpacking torsocks (2.3.0-2) ...
Selecting previously unselected package tor-geoipdb.
Preparing to unpack .../tor-geoipdb_0.4.2.7-1_all.deb ...
Unpacking tor-geoipdb (0.4.2.7-1) ...
Setting up tor (0.4.2.7-1) ...
Setting up torsocks (2.3.0-2) ...
Setting up tor-geoipdb (0.4.2.7-1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for systemd (245.4-4ubuntu3.11) ...
```

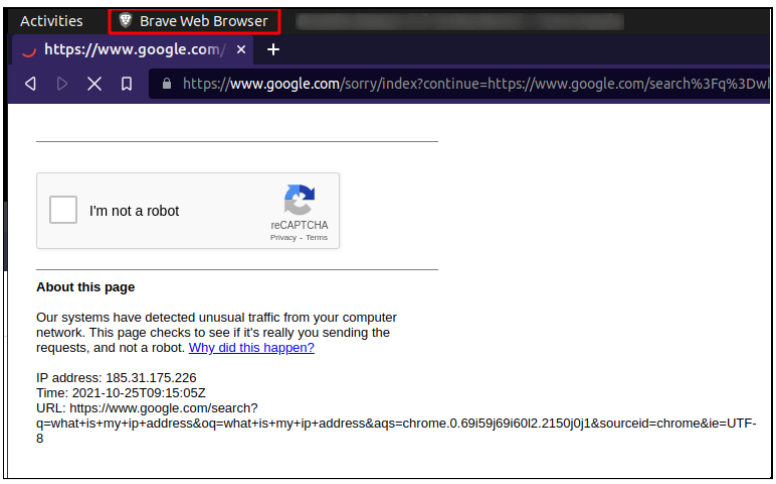
2. Run `tor` in the terminal: `$ tor`. This will start the `tor` process and open a SOCKs proxy on port 9050 of localhost.

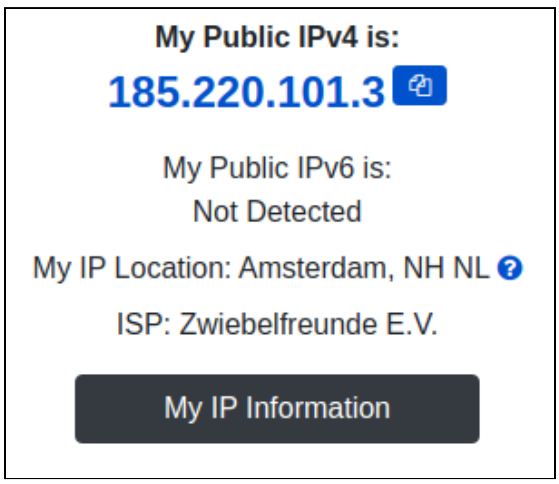
```
~
❯ tor
Oct 25 14:40:04.840 [notice] Tor 0.4.2.7 running on Linux with Libevent 2.1.11-stable, OpenSSL 1.1.1f, Zlib 1.2.11, Liblzma 5.2.4, and Libzstd 1.4.4.
Oct 25 14:40:04.840 [notice] Tor can't help you if you use it wrong! Learn how to be safe at https://www.torproject.org/download/download#warning
Oct 25 14:40:04.840 [notice] Read configuration file "/etc/tor/torrc".
Oct 25 14:40:04.841 [notice] Opening Socks listener on 127.0.0.1:9050
Oct 25 14:40:04.841 [notice] Opened Socks listener on 127.0.0.1:9050
Oct 25 14:40:04.000 [notice] Parsing GEOIP IPv4 file /usr/share/tor/geoip.
Oct 25 14:40:04.000 [notice] Parsing GEOIP IPv6 file /usr/share/tor/geoip6.
Oct 25 14:40:05.000 [notice] Bootstrapped 0% (starting): Starting
Oct 25 14:40:05.000 [notice] Starting with guard context "default"
Oct 25 14:40:06.000 [notice] Bootstrapped 5% (conn): Connecting to a relay
Oct 25 14:40:06.000 [notice] Bootstrapped 10% (conn_done): Connected to a relay
Oct 25 14:40:06.000 [notice] Bootstrapped 14% (handshake): Handshaking with a relay
Oct 25 14:40:07.000 [notice] Bootstrapped 15% (handshake_done): Handshake with a relay done
Oct 25 14:40:07.000 [notice] Bootstrapped 75% (enough_dirinfo): Loaded enough directory info to build circuits
Oct 25 14:40:07.000 [notice] Bootstrapped 90% (ap_handshake_done): Handshake finished with a relay to build circuits
Oct 25 14:40:07.000 [notice] Bootstrapped 95% (circuit_create): Establishing a Tor circuit
Oct 25 14:40:07.000 [notice] Bootstrapped 100% (done): Done
```

3. Go to Ubuntu's system settings and enable the Network Proxy.
    a. Configure the Network Proxy to use the "Manual mode"
    b. Add 127.0.0.1:9050 as a SOCKs host for the proxy



4. The system is now configured to route internet traffic through Tor. As we can see, my IP address has changed to that of a server in Amsterdam. Note: I am using the Brave browser and not Tor Browser.





To get our auto-generated hostname, we follow the following steps:

1. Edit the `torrc` file and create a new hidden service by pointing to its location:

```
############### This section is just for location-hidden services ###

## Once you have configured a hidden service, you can look at the
## contents of the file ".../hidden_service/hostname" for the address
## to tell people.
##
## HiddenServicePort x y:z says to redirect requests on port x to the
## address y:z.

HiddenServiceDir /home/plibither8/tor_test
HiddenServicePort 80 127.0.0.1:80
```

2. Create the directory and limit its permissions to make it less accessible:

```
~
❯ pwd
/home/plibither8

~
❯ mkdir tor_test

~
❯ chmod 700 tor_test
```

3. Run the tor service using `tor`, as we did previously
4. Now, our directory `/home/plibither8/tor_test`, should be populated with auto-generated files of our new hidden onion service, with the hostname and public and private keys.

```
~
❯ \ls tor_test
authorized_clients  hostname  hs_ed25519_public_key  hs_ed25519_secret_key

~
❯ cat tor_test/hostname
jvkbxk5gjw2gnvlaxqb6ghv22jjpotjcjtonidig3vcid4ncdpzr63yd.onion
```

5. The auto-generated hostname is:
   jvkbxk5gjw2gnvlaxqb6ghv22jjpotjcjtonidig3vcid4ncdpzr63yd.onion

Question 7

a. We follow the following steps:
   i. One our personal machine, we generate a 2048 size RSA private key: `$ openssl genrsa -out ~/.ssh/fcs 2048`

```
~/.ssh
❯ openssl genrsa -out ~/.ssh/fcs 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....................................................+++++
.....................................................+++++
e is 65537 (0×010001)
```

   ii. Get the public key from the private key: `$ ssh-keygen -f ~/.ssh/fcs -y > ~/.ssh/fcs.pub`. Now we have both the private and public keys in the `.ssh` directory:

```
~/.ssh
❯ ls ~/.ssh/fcs*
-rw------- 1 plibither8 plibither8 1679 Oct 25 15:26 /home/plibither8/.ssh/fcs
-rw-rw-r-- 1 plibither8 plibither8  381 Oct 25 15:28 /home/plibither8/.ssh/fcs.pub
```

   iii. Inside our VM, we edit the `~/.ssh/authorized_keys` file and add our public key:

```
➜  ~ echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDG/n4OG1YVRJwugcwIVyugMNMmlJZv2QI75cu6ReZ3Z275373PUWXv
OUOLGC28W4A2FErquqtCLfbW6YfNP7RLjx2d18dSpN7y7RncMD2I+VJS+dM18LfhwKHJt49n4sv8FV4IoahYE/lOOo7pUt9qT1ITq5x
FZrPm1LarQujngRnfbnG/skT6ZCRyVoiGK5UsF1BXa66nrJUGuAwBtwOx+Mvn2l3VuEt6L6NHnqK5IE+whQOOr3730bOuHSwJUgHAjf
O85DNqy9NisEerGXRf7D9kIvJCWkOVM5ndh1TOJzriR5c2Tj1u+jyQfIYCN8OCOlGqxjHr/JadqGAS8Tp1" > ~/.ssh/authorized
_keys
```

   iv. After exiting our VM, we can now SSH into the VM without having to enter our password.

b. Both have their pros and cons, but a public key authentication is a lot safer in many ways than password-authentication. Some of the reasons:
   i. Passwords are usually small, insecure and easily guessable or computable. Private keys are very long, and cannot be guessed, memorized or computed through brute-force.
   ii. Many times, passwords are directly entered into the command line and are saved into the shell history file, which makes them visible in plaintext.
   iii. Access through public keys can be easily revoked by deleting the line in the `authorized_keys` section without affecting access to the other users of the machine.
   iv. When using password-authentication, the password is transmitted over the network and compared, whereas private keys are stored client-side only and their contents are not transmitted.

c. We create the text files using the `seq` command and stream the output into 3 respective files:

```
~/iiitd/padhai/semester-5/cse545_fcs/homeworks/2/texts main*
❯ seq 1 100 > 100.txt

~/iiitd/padhai/semester-5/cse545_fcs/homeworks/2/texts main*
❯ seq 1 100000 > 100000.txt

~/iiitd/padhai/semester-5/cse545_fcs/homeworks/2/texts main*
❯ seq 1 100000000 > 100000000.txt

~/iiitd/padhai/semester-5/cse545_fcs/homeworks/2/texts main*
❯ ls
total 868648
drwxrwxr-x 2 plibither8 plibither8      4096 Oct 25 16:03 .
drwxrwxr-x 4 plibither8 plibither8      4096 Oct 25 15:59 ..
-rw-rw-r-- 1 plibither8 plibither8 888888898 Oct 25 16:03 100000000.txt
-rw-rw-r-- 1 plibither8 plibither8    588895 Oct 25 16:03 100000.txt
-rw-rw-r-- 1 plibither8 plibither8       292 Oct 25 16:03 100.txt

~/iiitd/padhai/semester-5/cse545_fcs/homeworks/2/texts main*
❯ wc -l *
100000000 100000000.txt
   100000 100000.txt
      100 100.txt
100100100 total
```

i. RC4: Since this is a symmetric encryption, let the password be the contents of the private key generated in (a):

`$ openssl rc4 -pass file:../keys/fcs -in 100.txt -out 100.enc -pbkdf2`

```
~/iiitd/padhai/semester-5/cse545_fcs/homeworks/2/texts main*
❯ openssl rc4 -pass file:../keys/fcs -in 100.txt -out 100.enc -pbkdf2

~/iiitd/padhai/semester-5/cse545_fcs/homeworks/2/texts main*
❯ cat 100.enc
Salted__VOS%4
             /ə;k>Jh    3\b\Gu5
                                Wn(cL%sO
                                         1.}wWĵñ}07{%L\X{6}g^bzo{G>
h,MCNs  #&—us%WUD1h:[<.Im6! P3J2HB8
{ƶlUt̲d15CvZ3w'a<Ep`)8o=V#d      6t~+
```

Similarly, we do this for the 100000.txt and 100000000.txt files.

ii. AES-256: Again, since AES-256 is also a symmetric encryption scheme, we will be encrypting with the private key as the password.

`$ openssl enc -aes-256-cbc -pass file:../keys/fcs -in 100.txt -out 100.enc -pbkdf2`

```
~/iiitd/padhai/semester-5/cse545_fcs/homeworks/2/texts main*
❯ openssl enc -aes-256-cbc -pass file:../keys/fcs -in 100.txt -out 100.enc -pbkdf2

~/iiitd/padhai/semester-5/cse545_fcs/homeworks/2/texts main*
❯ cat 100.enc
A1M\(Jj:ZW|L\əiPJ C^<cE—≺'~oGRB5p⁂R;hy
                        0GKIXf7B4LB
                                   fv*4B\gtVL+35D{ų'5d 4{!#ˆVP Ck[Jr—1`iφA'#!5F>>HO`M"
```

Similarly, we do this for the 100000.txt and 100000000.txt files.

iii. RSA: Since RSA is an asymmetric encryption algorithm, it does not support encryption of large files. For large files, the more appropriate method for encryption is by using symmetric encryption schemes like AES-256 shown above. In any case, the following command would have been used to encrypt the file, but it causes an error since the input data is too large:

`$ openssl rsautl -encrypt -inkey ../keys/fcs.pub -pubin -in 100.txt -out 100.enc`

```
~/iiitd/padhai/semester-5/cse545_fcs/homeworks/2/texts main*
❯ openssl rsautl -encrypt -inkey ../keys/fcs.pub -pubin -in 100.txt -out 100.enc
RSA operation error
140035559028032:error:0406D06E:rsa routines:RSA_padding_add_PKCS1_type_2:data too large for key size:cr
ypto/rsa/rsa_pk1.c:125:
```

Similarly, the data in 100000.txt and 100000000.txt files are too large for RSA encryption to handle.

Question 8

a. Before disabling ICMP echo we can ping successfully to our own machine (loopback / 127.0.0.1 / localhost) and other servers:

```
~
❯ ping localhost
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.045 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.049 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.041 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.036 ms
64 bytes from localhost (127.0.0.1): icmp_seq=5 ttl=64 time=0.036 ms
^C
--- localhost ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4074ms
rtt min/avg/max/mdev = 0.036/0.041/0.049/0.005 ms

~
❯ ping google.com
PING google.com (142.250.77.46) 56(84) bytes of data.
64 bytes from bom07s26-in-f14.1e100.net (142.250.77.46): icmp_seq=1 ttl=117 time=25.6 ms
64 bytes from bom07s26-in-f14.1e100.net (142.250.77.46): icmp_seq=2 ttl=117 time=25.5 ms
64 bytes from bom07s26-in-f14.1e100.net (142.250.77.46): icmp_seq=3 ttl=117 time=25.5 ms
64 bytes from bom07s26-in-f14.1e100.net (142.250.77.46): icmp_seq=4 ttl=117 time=25.8 ms
64 bytes from bom07s26-in-f14.1e100.net (142.250.77.46): icmp_seq=5 ttl=117 time=25.9 ms
^C
--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 25.532/25.677/25.906/0.149 ms
```

Using iptables, we disable a response to ICMP requests, but still allow outgoing ICMP requests to other servers:

`$ iptables -I INPUT -p icmp --icmp-type echo-request -j DROP`

```
~
❯ sudo iptables -I INPUT -p icmp --icmp-type echo-request -j DROP


~
❯ ping localhost
PING localhost (127.0.0.1) 56(84) bytes of data.
^C
--- localhost ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3055ms


~
❯ ping google.com
PING google.com (142.250.77.46) 56(84) bytes of data.
64 bytes from bom07s26-in-f14.1e100.net (142.250.77.46): icmp_seq=1 ttl=117 time=25.6 ms
64 bytes from bom07s26-in-f14.1e100.net (142.250.77.46): icmp_seq=2 ttl=117 time=26.0 ms
64 bytes from bom07s26-in-f14.1e100.net (142.250.77.46): icmp_seq=3 ttl=117 time=25.5 ms
64 bytes from bom07s26-in-f14.1e100.net (142.250.77.46): icmp_seq=4 ttl=117 time=25.8 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 25.506/25.726/25.987/0.192 ms
```

b. Since I'm on the IIIT Delhi network, the three devices that I'm using that are connected to the IIIT internal network are:
   i. My laptop (IP: 192.168.65.227)
   ii. My phone (IP: 192.168.166.185)
   iii. VM (IP: 192.168.2.237)

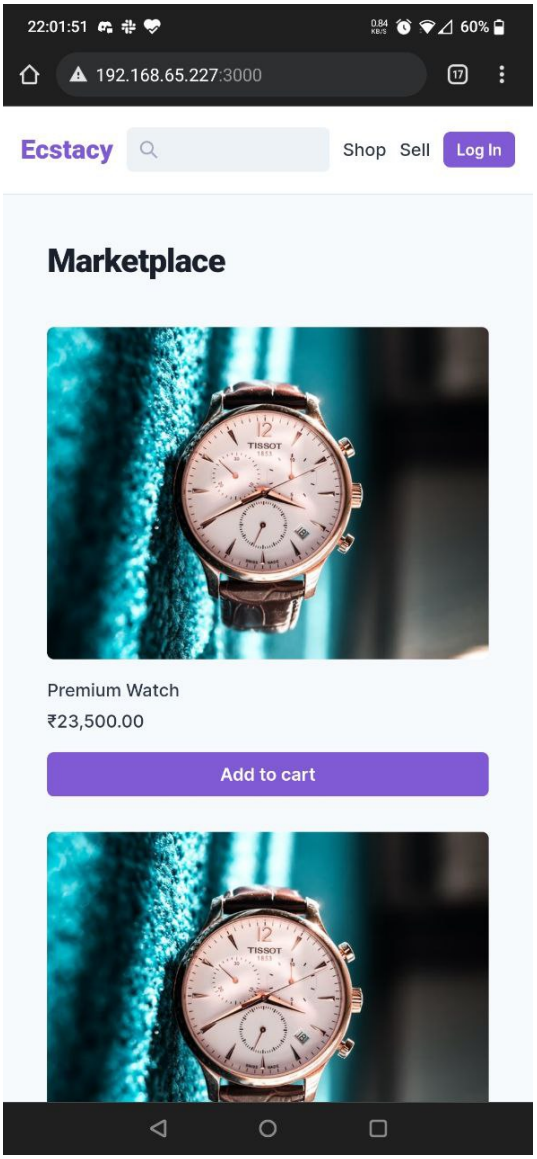
   I am hosting a website at port 3000 of my laptop's localhost (http://localhost:3000).
   The aim is to make this website accessible from my own machine (of course), my phone but no other machine on the IIIT network (which includes the VM).

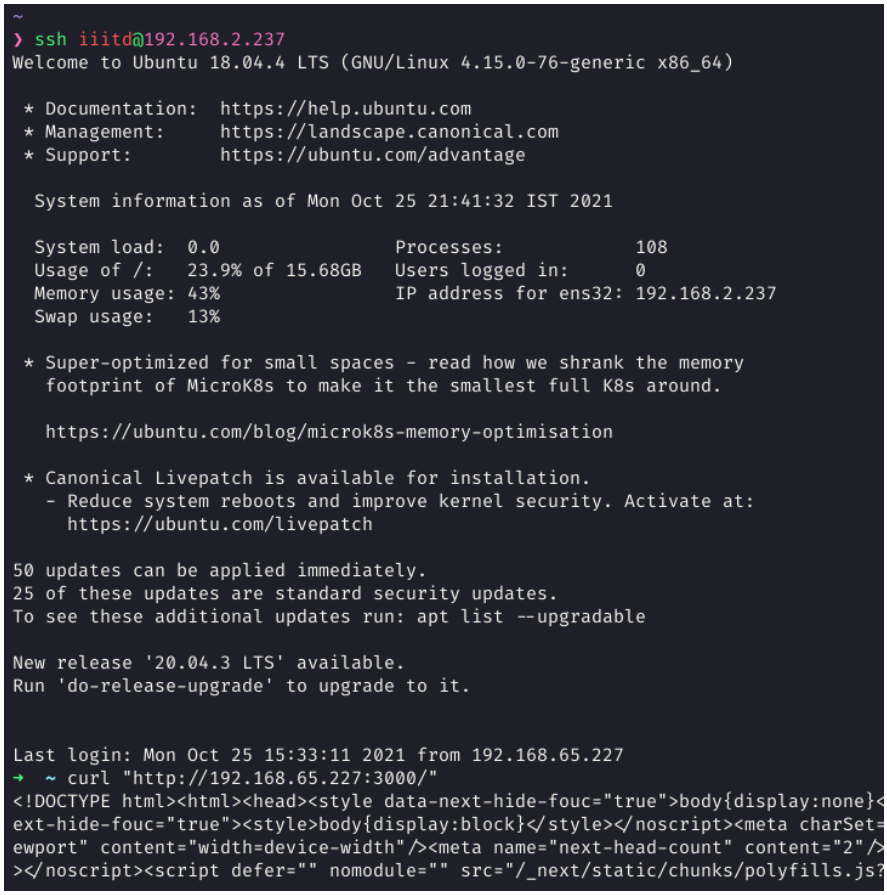
   Before:

   - The IP Table of my laptop is empty of any rules in any chains:

```
~
❯ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain DOCKER (0 references)
target     prot opt source               destination

Chain DOCKER-ISOLATION-STAGE-1 (0 references)
target     prot opt source               destination

Chain DOCKER-ISOLATION-STAGE-2 (0 references)
target     prot opt source               destination

Chain DOCKER-USER (0 references)
target     prot opt source               destination

Chain LIBVIRT_FWI (0 references)
target     prot opt source               destination

Chain LIBVIRT_FWO (0 references)
target     prot opt source               destination

Chain LIBVIRT_FWX (0 references)
target     prot opt source               destination

Chain LIBVIRT_INP (0 references)
target     prot opt source               destination

Chain LIBVIRT_OUT (0 references)
target     prot opt source               destination
```

- My laptop, phone and the VM can access the site on http://192.168.65.227:3000:





After:

- We add the following ip table rules:
  - `$ iptables -A INPUT -p tcp --dport 3000 -s 192.168.166.185 -j ACCEPT`
    - Accept incoming TCP connections from my mobile to port 3000
  - `$ iptables -A INPUT -p tcp --dport 3000 -i lo -j ACCEPT`
    - Accept incoming TCP connection from my loopback interface (own machine) to port 3000

- `$ iptables -A INPUT -p tcp --dport 3000 -j DROP`
  - Reject all other connection request to port 3000

```
~
❯ sudo iptables -A INPUT -p tcp --dport 3000 -s 192.168.166.185 -j ACCEPT

~
❯ sudo iptables -A INPUT -p tcp --dport 3000 -j DROP

~
❯ sudo iptables -D INPUT 2

~
❯ sudo iptables -A INPUT -p tcp --dport 3000 -i lo -j ACCEPT

~
❯ sudo iptables -A INPUT -p tcp --dport 3000 -j DROP

~
❯ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  192.168.166.185      anywhere             tcp dpt:3000
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:3000
DROP       tcp  --  anywhere             anywhere             tcp dpt:3000

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain DOCKER (0 references)
target     prot opt source               destination

Chain DOCKER-ISOLATION-STAGE-1 (0 references)
target     prot opt source               destination

Chain DOCKER-ISOLATION-STAGE-2 (0 references)
target     prot opt source               destination

Chain DOCKER-USER (0 references)
target     prot opt source               destination

Chain LIBVIRT_FWI (0 references)
target     prot opt source               destination

Chain LIBVIRT_FWO (0 references)
target     prot opt source               destination

Chain LIBVIRT_FWX (0 references)
target     prot opt source               destination

Chain LIBVIRT_INP (0 references)
```

- My laptop and mobile can still access the site on the given URL, but the VM's curl command fails:

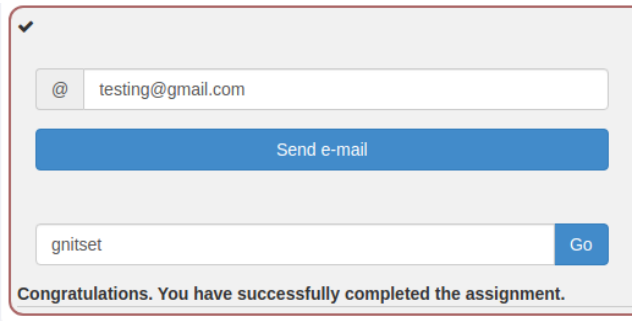
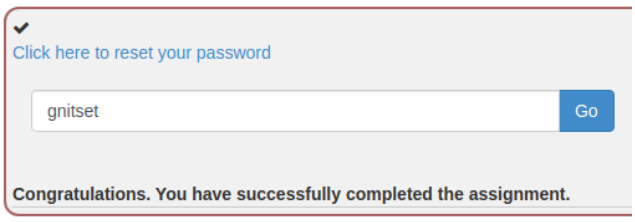
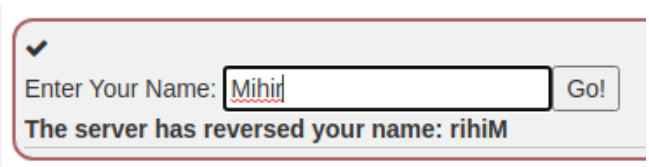
```
➜  ~ curl "http://192.168.65.227:3000/"
curl: (7) Failed to connect to 192.168.65.227 port 3000: Connection timed out
➜  ~
```

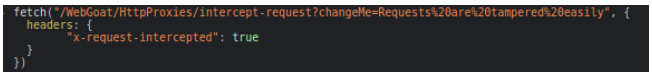
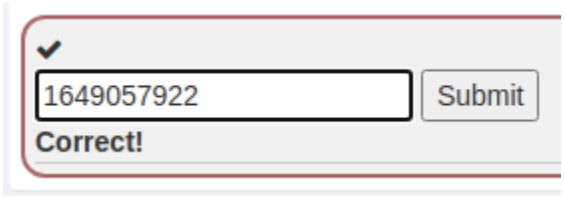
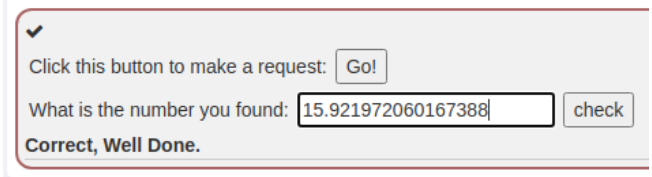
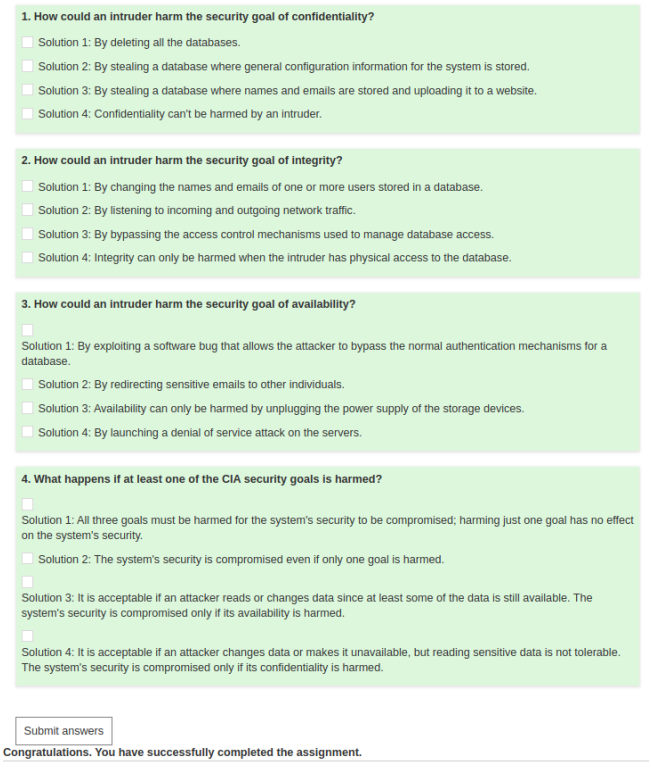
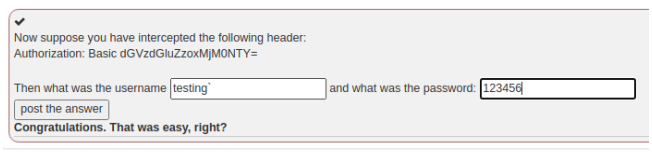
Question 9

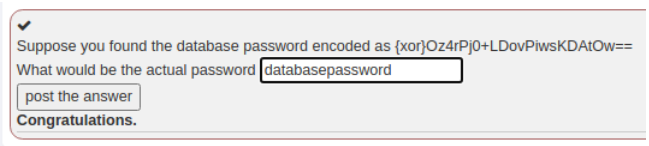
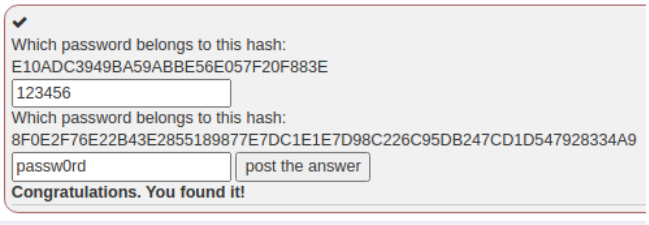
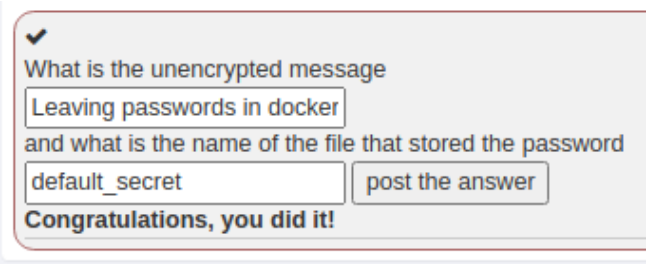
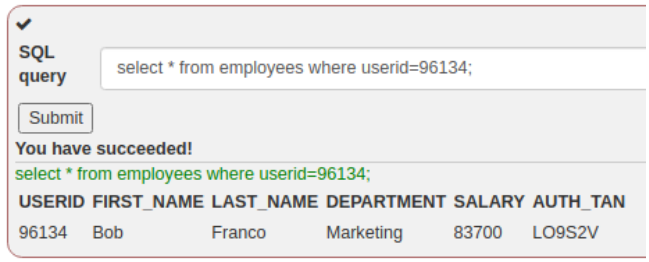
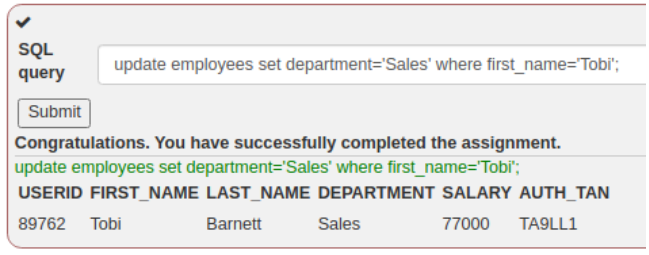
Notes:

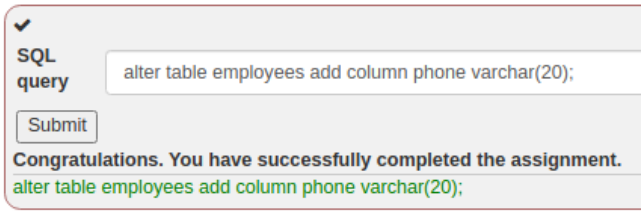
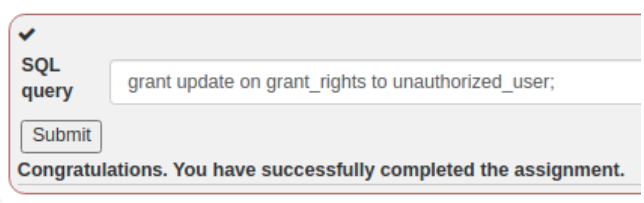
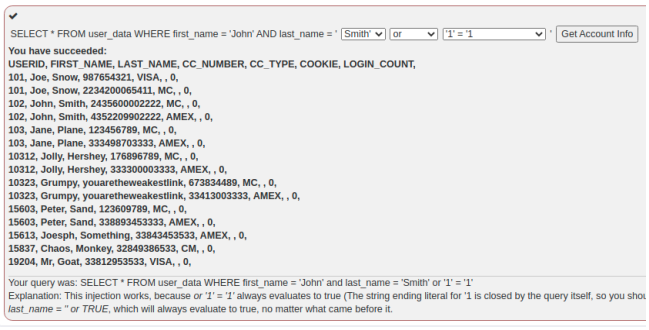
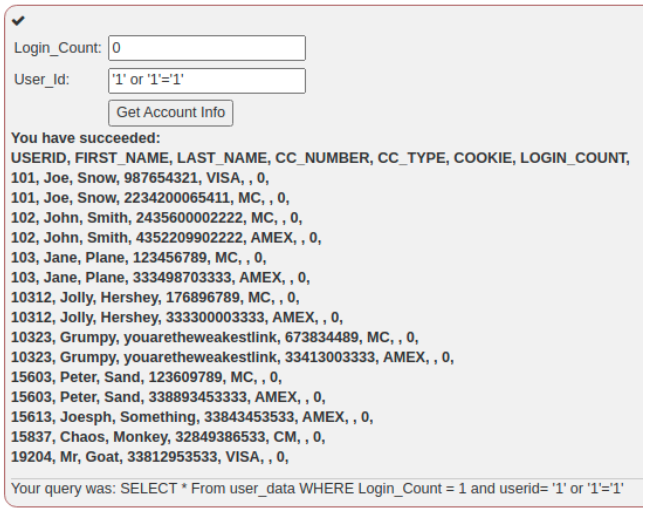
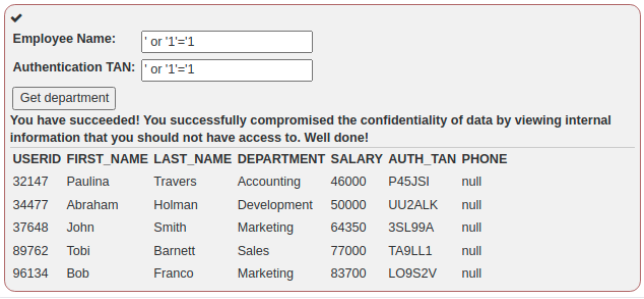
- "(A9) Vulnerable Components" lesson, "Exploiting XStream" assignment has a bug in it's current version. I have reported it on GitHub, and it has been acknowledged by the maintainer. Thus, this lesson cannot be completed.
- "(A8) Insecure Deserialization" lesson, Assignment 5 has a bug in its current version. I have reported it on GitHub, and it has been acknowledged by the maintainer. Thus, this lesson cannot be completed.
- "Cross-Site Request Forgeries", Assignment 8 does not show a green tick even though it is completed, thus the lesson is marked as "unsolved" even though it is solved.
- "SQL Injection (advanced)", does not show a green tick even though all its assignments are completed, thus the lesson is marked as "unsolved" even though it is solved.

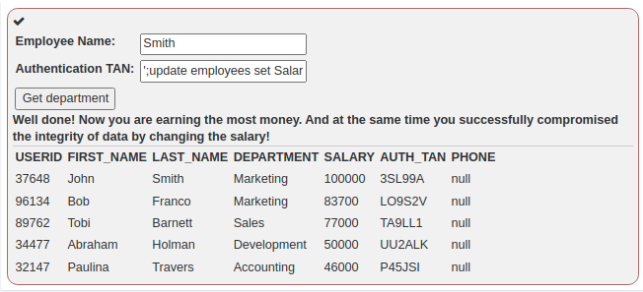
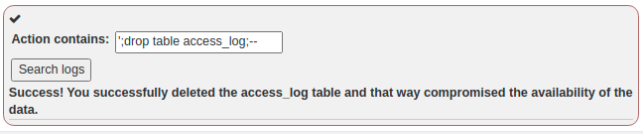
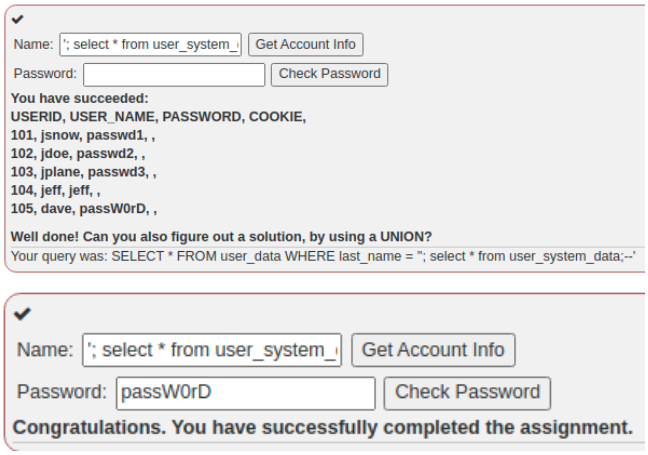
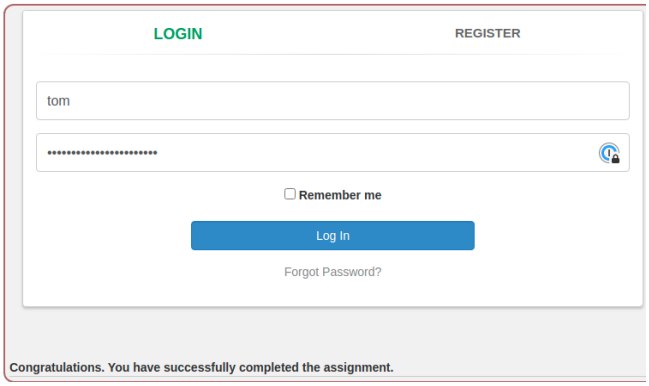
The lessons have been numbered in the order they appear on the left nav bar:

Lesson	Assignment	Screenshot	Observations
1	--	--	Assignment-less introduction to WebGoat
2	3		Using WebGoat we send an email, which is captured by WebWolf through which we get the secret.
	4		On WebWolf we see the incoming requests where the GET request's query param has the secret
3	2		The server sends a POST request which the server parses, and send a response

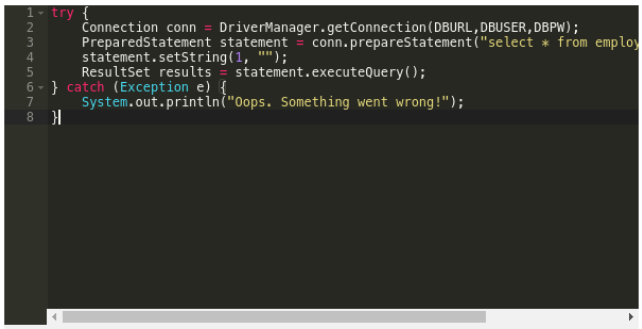
	3		Inspecting the http request in the network tab we see that its a POST request and the payload holds the magic number
4	6		Using this command we make a request with the specified configurations to complete the assignment. Response: “Well done, you tampered the request as expected”
5	4		Running the function in dev tool’s console logs the string where the phone number is found
	6		‘networkNum’ in FormData
6	5		Simple quiz with simple answers
7	2		Using `atob()` function in devtools, decoded to “testing:123456”

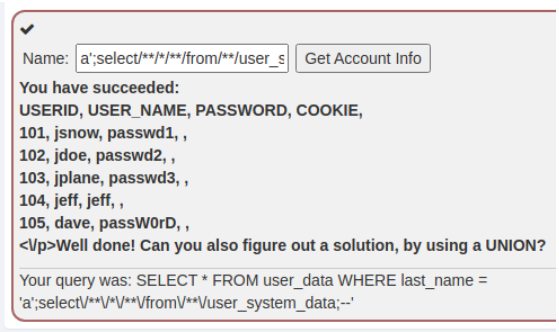
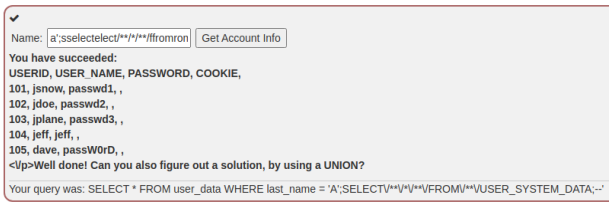
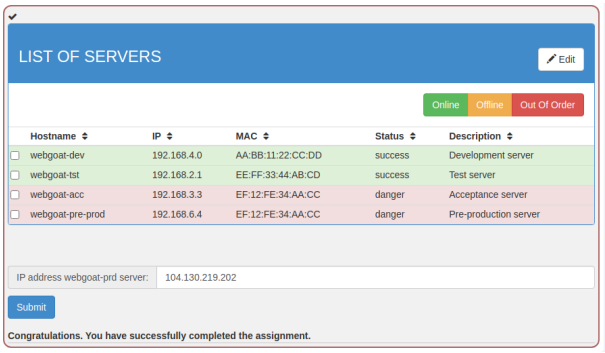
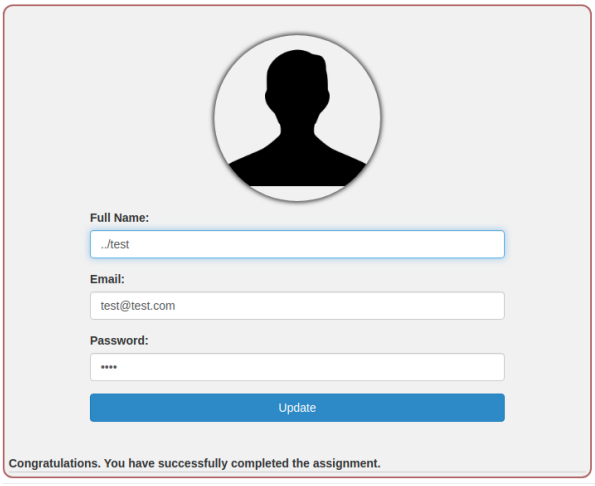
	3		Used online service to decrypt the XOR encoding
	4		Simple google search for the hashes resulted in these.
	6		Created a short Java program to decode the result: https://replit.com/@plibither8/TalkativeFirmBrackets
	8		• Run the docker container • Run `docker exec -ti -u 0 {container_id}` • File is /root/default_secret • Run the given command and see the password
8	6		`private final String secretValue = "secr37Value";`
9	2		Simple SQL query
	3		Simple SQL query

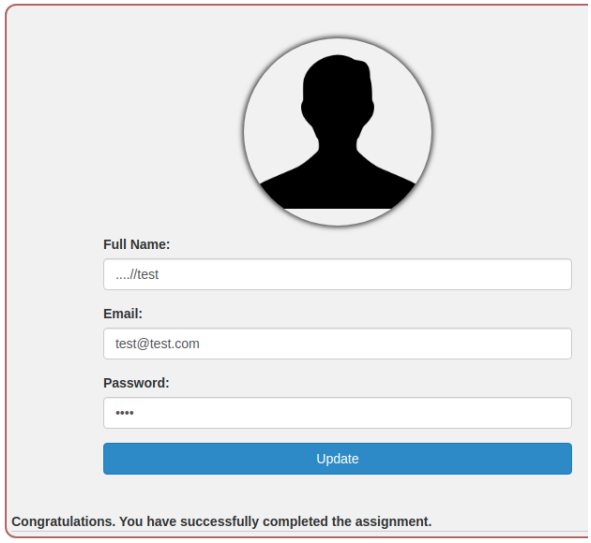
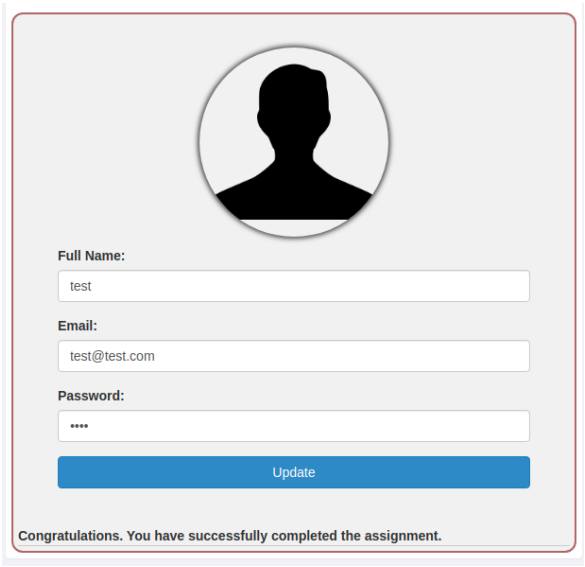
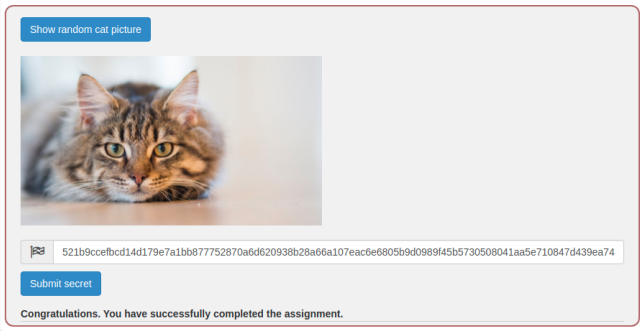
	4		Simple SQL query
	5		Simple SQL query
	9		Here we learn about the string SQL injection and how to manipulate single quotes
	10		Creating an sql injection using single quotes
	11		Viewing hidden table info using sql injection

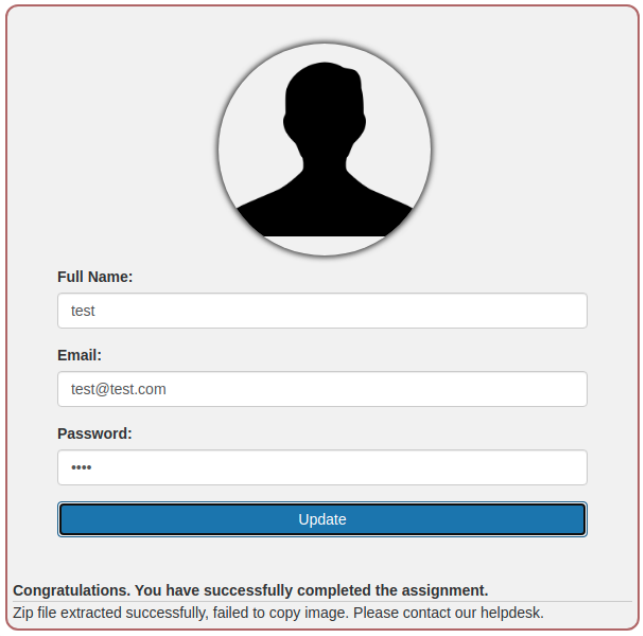
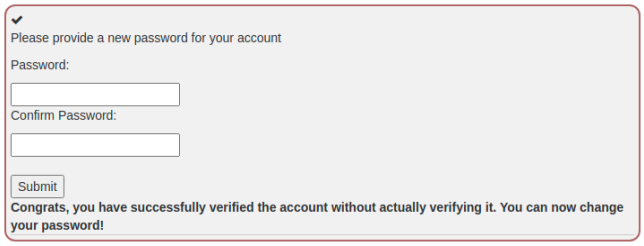
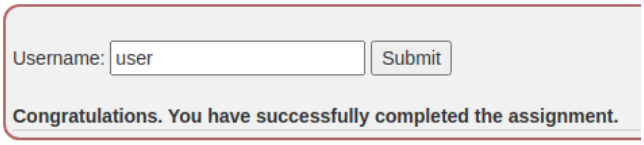
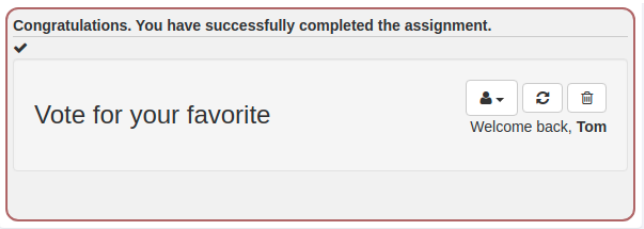
	12		Injection string used in 'Authentication TAN': `';update employees set Salary=100000 where last_name='Smith';--`
	13		Deleted the entire table thereby compromising availability of the database
10	3		First we find look at the user_system_data table, find Dave's password in plaintext, and complete the assignment
	5		Here we bruteforce and find the password for tom by trying out repeatedly, letter-by-letter the possible password that tom has.

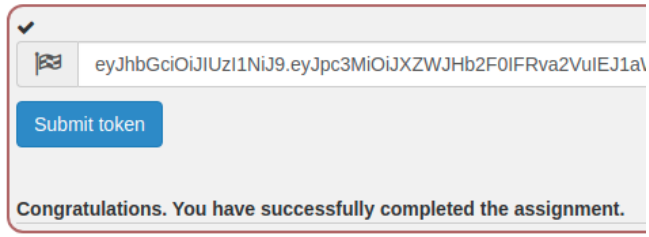
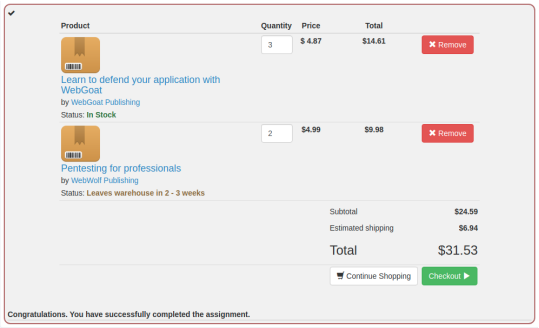
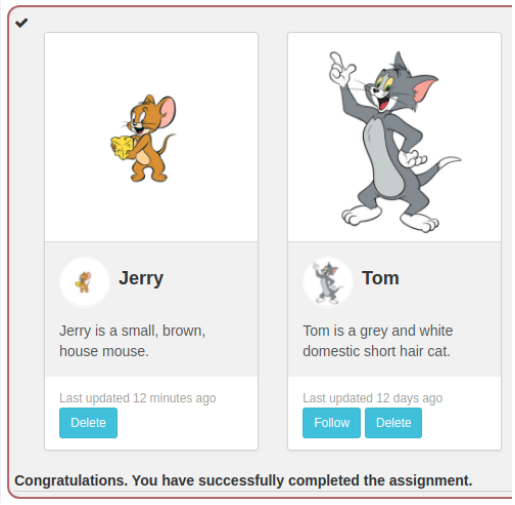
	6	**1. What is the difference between a prepared statement and a statement?** Solution 1: Prepared statements are statements with hard-coded parameters. Solution 2: Prepared statements are not stored in the database. Solution 3: A statement is faster. Solution 4: A statement has got values instead of a prepared statement **2. Which one of the following characters is a placeholder for variables?** Solution 1: * Solution 2: = Solution 3: ? Solution 4: ! **3. How can prepared statements be faster than statements?** Solution 1: They are not static so they can compile better written code than statements. Solution 2: Prepared statements are compiled once by the database management system waiting for input and are pre-compiled this way. Solution 3: Prepared statements are stored and wait for input it raises performance considerably. Solution 4: Oracle optimized prepared statements. Because of the minimal use of the databases resources it is faster. **4. How can a prepared statement prevent SQL-Injection?** Solution 1: Prepared statements have got an inner check to distinguish between input and logical errors. Solution 2: Prepared statements use the placeholders to make rules what input is allowed to use. Solution 3: Placeholders can prevent that the users input gets attached to the SQL query resulting in a seperation of code and data. Solution 4: Prepared statements always read inputs literally and never mixes it with its SQL commands. **5. What happens if a person with malicious intent writes into a register form :Robert); DROP TABLE Students;-- that has a prepared statement?** Solution 1: The table Students and all of its content will be deleted. Solution 2: The input deletes all students with the name Robert. Solution 3: The database registers 'Robert' and deletes the table afterwards. Solution 4: The database registers 'Robert'); DROP TABLE Students;--'. Submit answers **Congratulations. You have successfully completed the assignment.**	Simple quiz
11	5	✔ Connection conn = DriverManager.getConnection (DBURL, DBUSER, DBPW); PreparedStatement statemer = conn.prepareStatement ("SELECT status FROM users WHERE name= ? AND mail= ? "); statement.setString(1, "") ; statement.setString(2, "") ; Submit **Congratulations. You have successfully completed the assignment.**	Taken from the sample Java code that was there in the previous assignment

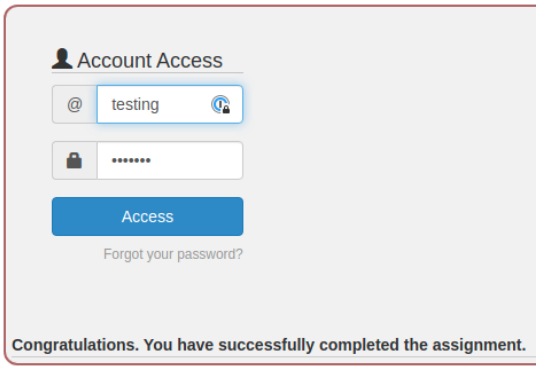
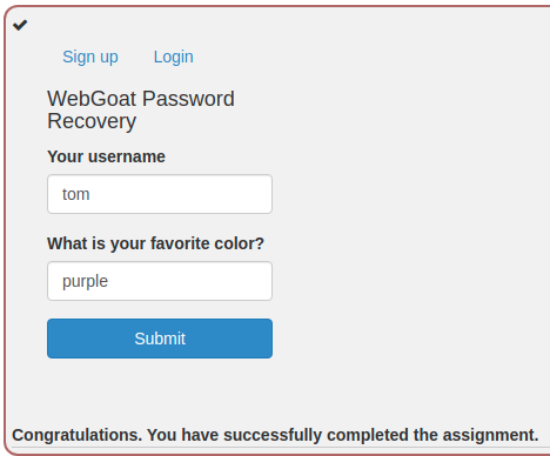
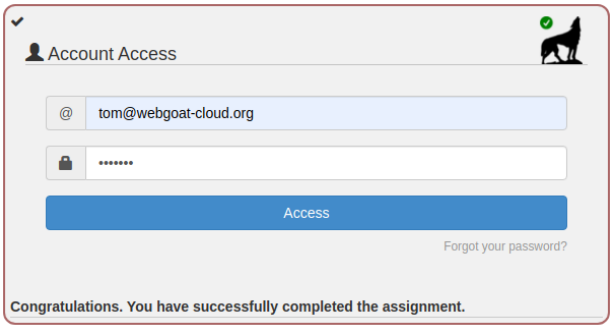
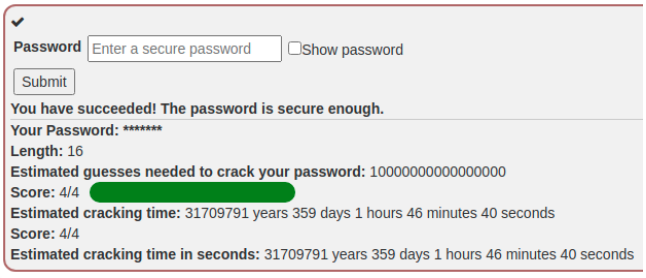
	6	
		
		Submit
		You did it! Your code can prevent an SQL injection attack!
			`Connection conn = DriverManager.getConnection(DBURL,DBUSER,DBPW);`
			`PreparedStatement statement = conn.prepareStatement("select * from employees where userid = ?;");`
			`statement.setString(1, "");`
			`ResultSet results = statement.executeQuery();`

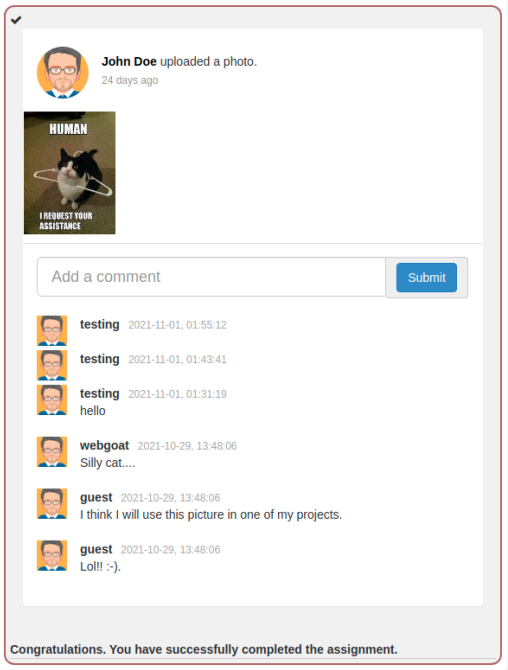
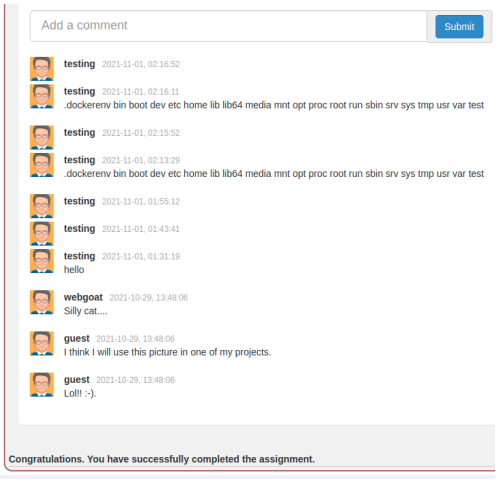
	9		Comments like /**/ can replace whitespaces
	10		Bypassing the poor “removal” of “SELECT”/”FROM” tokens by inserting them in the middle
	12		Sending an sql command through the query parameters
12	2		Simple relative path Input: ../test

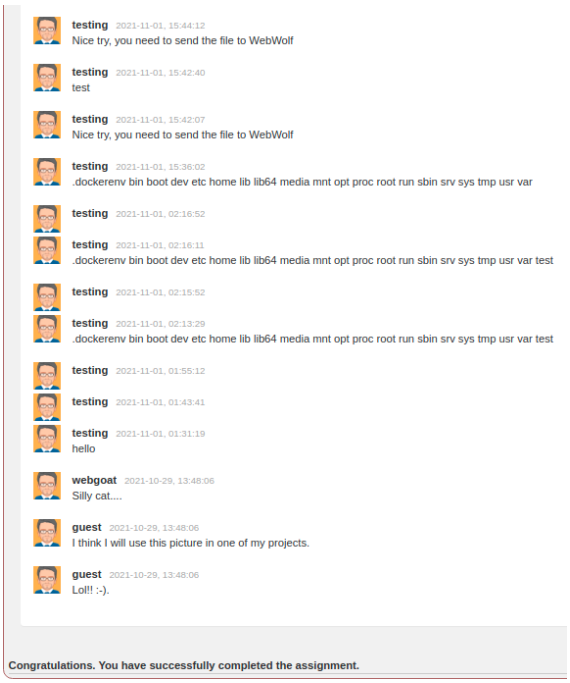
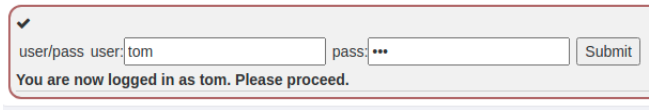
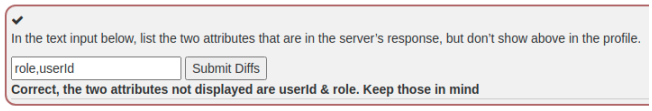
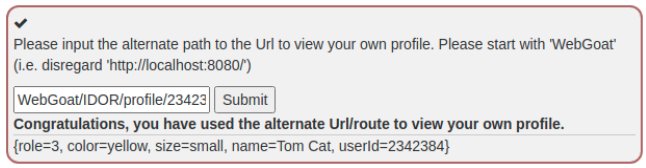
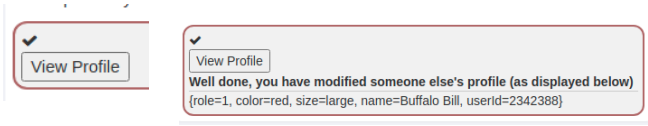
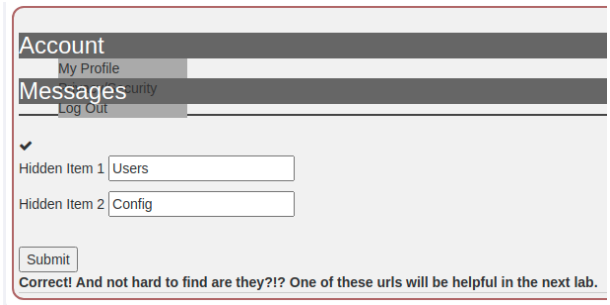
	3		Relative path Input: ....//test
	4		Intercepted the image and changed the filename to “../test”
	5		Found the secret image that said to encode our username using SHA-512

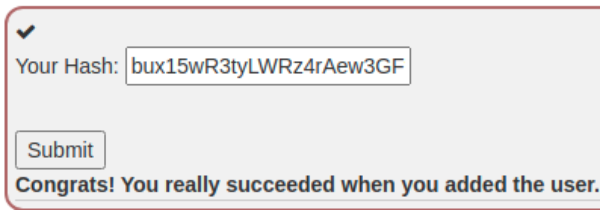
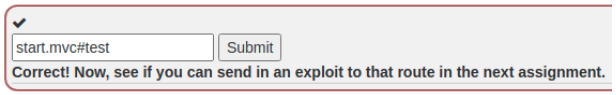
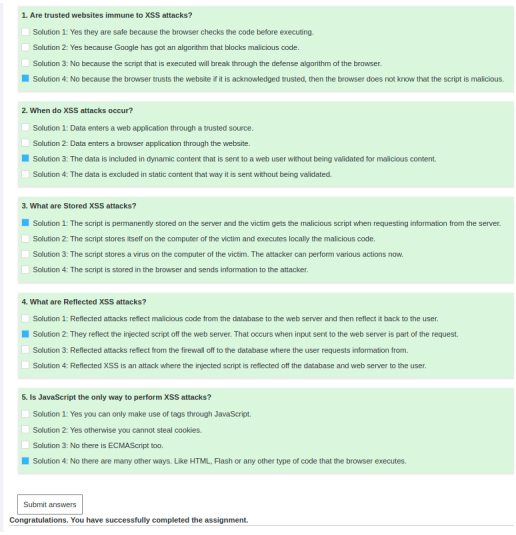
	7		Uploading a zip with an attack file
13	2		Change the names of the form inputs of question fields by inspecting the element and submit
14	3		Using https://jwt.im
	5		Intercepted request using Burp and changed admin: true and alg: none

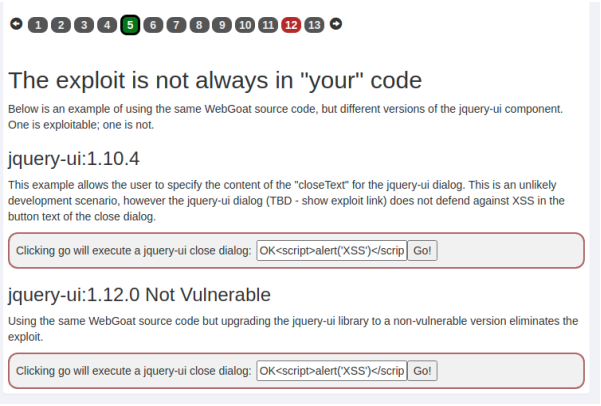
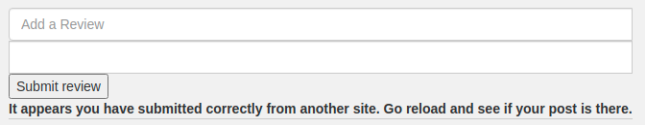
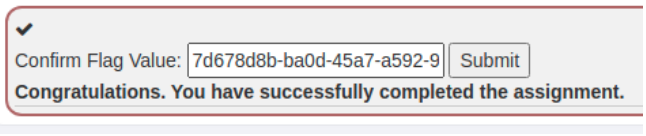
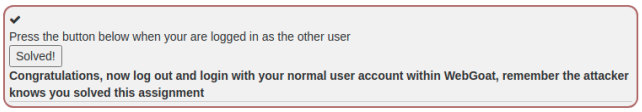
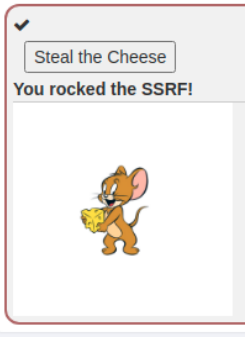
	7	Congratulations. You have successfully completed the assignment. **1. What is the result of the first code snippet?** ☑ Solution 1: Throws an exception in line 12 ☐ Solution 2: Invoked the method removeAllUsers at line 7 ☐ Solution 3: Logs an error in line 9 **2. What is the result of the second code snippet?** ☐ Solution 1: Throws an exception in line 12 ☐ Solution 2: Invoked the method removeAllUsers at line 7 ☑ Solution 3: Logs an error in line 9 Submit answers Congratulations. You have successfully completed the assignment.	In 1 we are only parsing claims, but in 2 we are parsing the token. Also, the claims have admin:true
	8		"shipping" was the signing secret, used jwt.io to change the payload and claims
	10		Requested new refresh token for myself Used refresh token to refresh leaked JWT Used JWT to authorize purchase
	12		Intercepted the request, changed the "kid", changed "jerry" to "tom" and re-signed using "deletingTom"

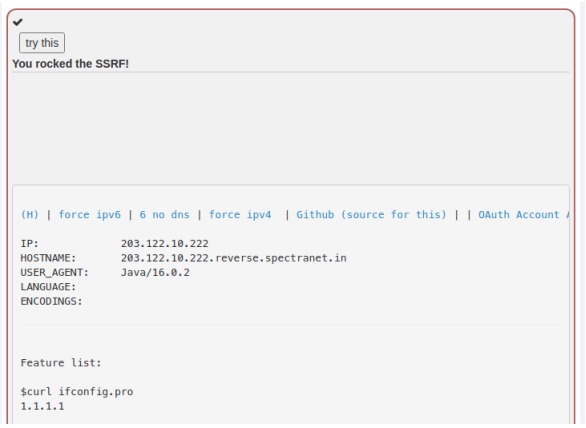
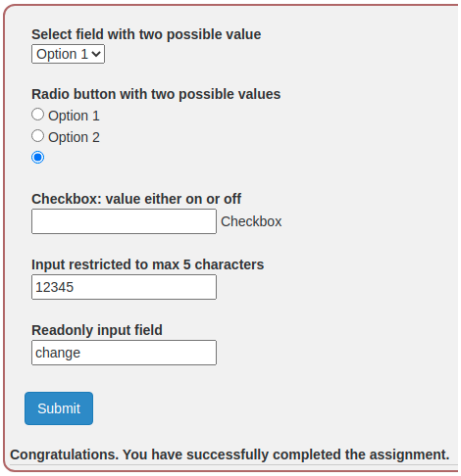
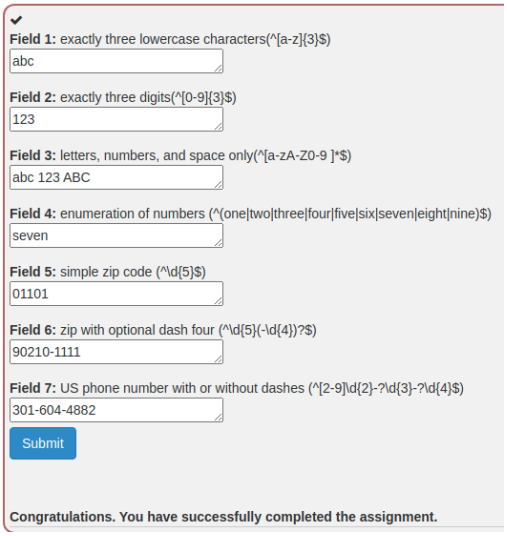
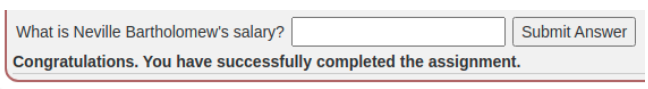
15	2		Simple assignment -- check webwolf for the received email and enter with updated password
	4		Trial and error through possible colors
	5		Describes details of why these security questions are bad
	6		Changing the host/referer to 127.0.0.1:9090 and intercepting the request, then using the password reset link to update tom's password
16	4		Entered password: *HZ99!hvXEPWkBp*

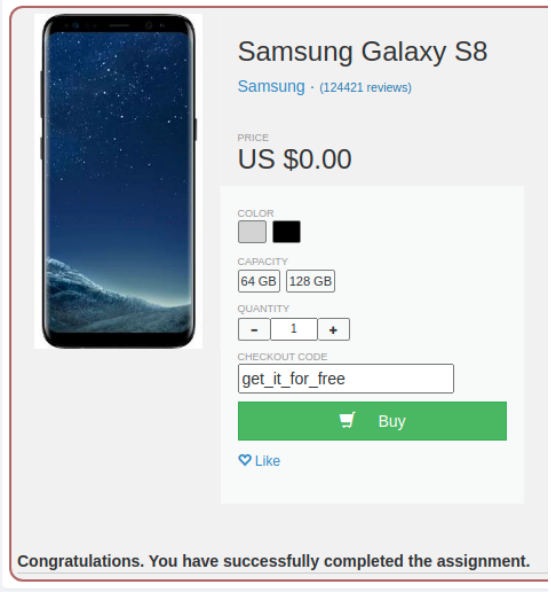
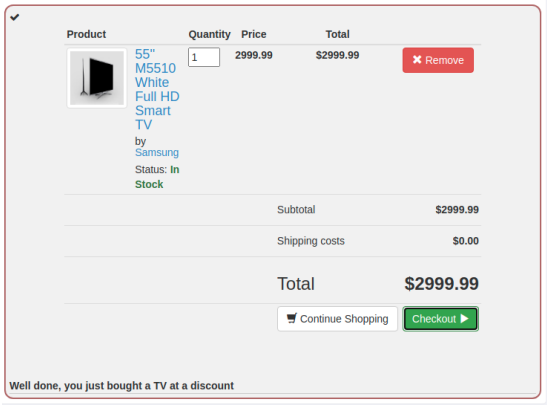
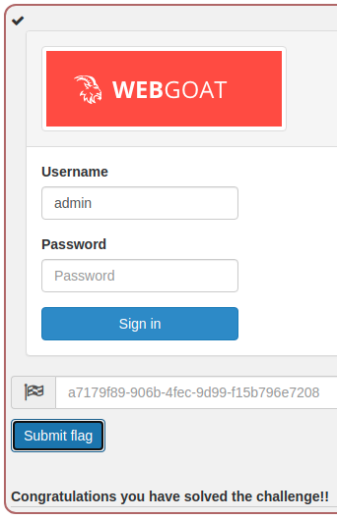
17	2		Username: CaptainJack Password: BlackPearl Had to modify the submit_secret_credentials() function and change the route slightly to intercept the request
18	4		Intercepted request using Burp and changed the request XML
	7		Intercepted request using Burp and changed the request XML and the content-type to XML

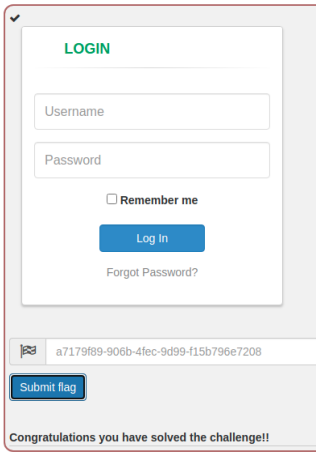
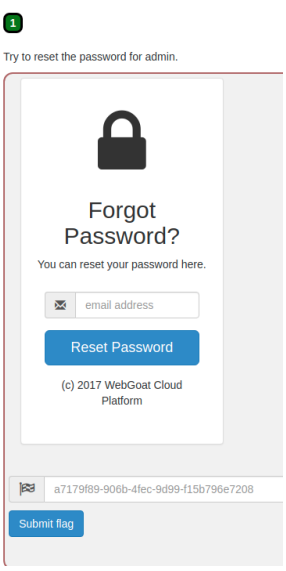
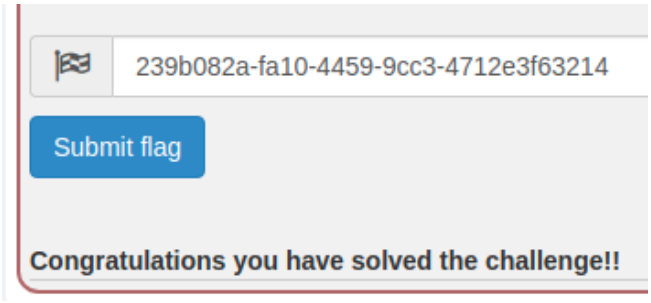
	11		Uploaded dtd attack file to WebWolf Got secret text from location to comment Modified XML request to use remote dtd
19	2		Credentials already provided
	3		Checking network requests and reading the response
	4		Finding and viewing our own profile
	5		Viewing details of someone else's profile and updating the details using PUT request
20	2		Using inspect element, found these links hidden using CSS

	3		Made GET request to /WebGoat/users with application/json Content-Type
21	2		Same host, same cookies!
	7		Simple XSS
	10		Finding hidden route by looking into the JavaScript source code using the console
	11		Running the function and retrieving the phone number
	12		Simple quiz
22	5	--	This assignment is broken for now, and I have created an issue for it on GitHub. https://github.com/WebGoat/WebGoat/issues/1129

23	5		
	12	--	This assignment is broken for now, and I have created an issue for it on GitHub. https://github.com/WebGoat/WebGoat/issues/1134
24	3		
	4		Intercepted and removed referer header
	7		Converted content-type to text/plain
	8		Created new user and used Session ID to make cross-site request Lesson does not turn green, so the report card does not show the lesson as completed
25	2		Modified the requested image input

	3		Modified the requested URL by intercepting and replaced with ifconfig
26	2		Manipulating the inputs using inspect element and then sending the request
	3		Failing all front-end regex, editing the JavaScript function that tests for this returning true early, and making the successful request
27	2		Checking the source code, filtering using inspect element

	3		Check network requests for '/coupons' and there is a "get_it_for_free" in the response list that does a discount of 100%
28	2		Again, change the value of the hidden input field in the html using inspect element and make the price 0
29	2		Download the webgoat logo Find the admin:!!... username-password combo, this changes every time. Enter the creds and get the flag

30	1		Similar to “Password reset” assignment
31	1		*Forgot to take a screenshot of this challenge when I completed it*
32	1		Change the request method to HEAD and look for the “X-Flag” header.

Report Card

OVERVIEW	
Total number of lessons	32
Total number of lessons solved	28
Total number of assignments	85
Total number of assignments solved	6

LESSON OVERVIEW

Lesson name	Solved	Number of attempts
Without password	true	4
Admin password reset	true	5
Admin lost password	true	24
Without account	true	2
Bypass front-end restrictions	true	16
Client side filtering	true	61
Crypto Basics	true	11
Cross Site Scripting	true	40
HTML tampering	true	4
HTTP Basics	true	4
HTTP Proxies	true	1
CIA Triad	true	1
Developer Tools	true	3
Insecure Direct Object References	true	11

Cross-Site Request Forgeries	false	14
Insecure Login	true	3
Insecure Deserialization	false	8
JWT tokens	true	49
Path traversal	true	60
SQL Injection (intro)	true	24
SQL Injection (mitigation)	true	143
SQL Injection (advanced)	false	13
Vulnerable Components	false	16
XXE	true	16
Authentication Bypasses	true	16
WebGoat	true	3
WebWolf	true	3
Missing Function Level Access Control	true	4
Password reset	true	45
Server-Side Request Forgery	true	3
Secure Passwords	true	2
Writing new lesson	true	2