

# CN ASSIGNMENT 2

Group members:

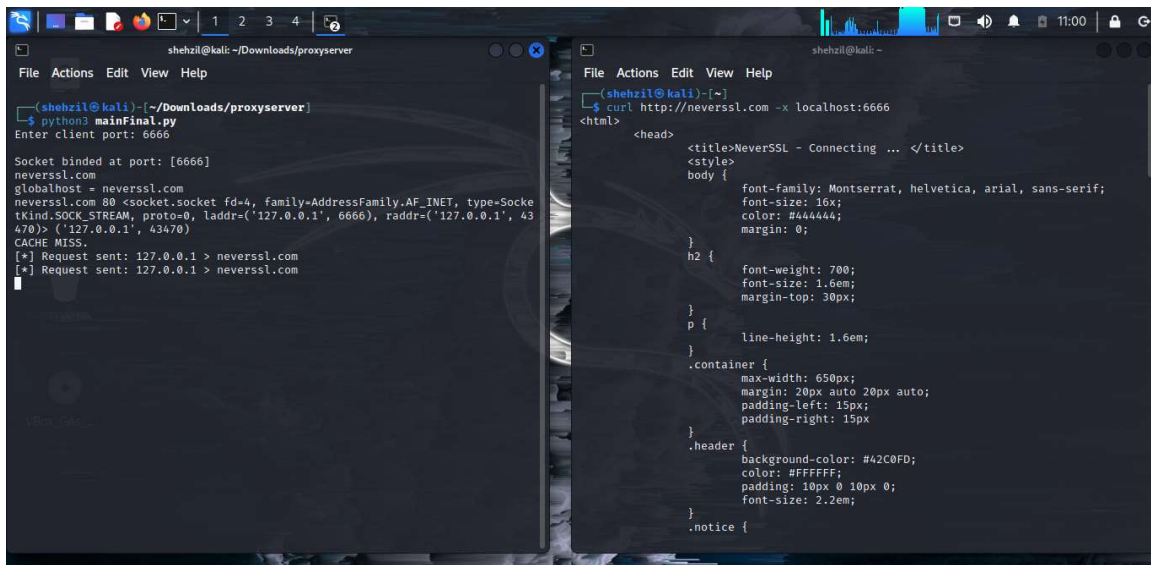
Shehzil Ali(20k-0336)

Shamikh Ali(20k-0301)

---

## WORKING DEMO SCREENSHOTS:

->Cache missed because we have searched this website for the first time



The image shows two terminal windows side-by-side. The left window is titled 'shehzil@kali: ~/Downloads/proxyserver' and shows the execution of a Python script 'mainFinal.py'. The script sets up a socket listener on port 6666, binds it to the local IP 127.0.0.1, and then receives a request from 'neverssl.com'. The output shows 'Socket binded at port: [6666]', 'neverssl.com', and 'CACHE MISS.' followed by two request logs. The right window is titled 'shehzil@kali: ~' and shows a curl command being executed: 'curl http://neverssl.com -x localhost:6666'. The output is an HTML document with a title 'NeverSSL - Connecting ...' and various CSS styles.

```
shehzil@kali: ~/Downloads/proxyserver
File Actions Edit View Help
[shehzil@kali] ~/Downloads/proxyserver
$ python2 mainFinal.py
Enter client port: 6666

Socket binded at port: [6666]
neverssl.com
globalhost = neverssl.com
neverssl.com 80 <socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 6666), raddr=('127.0.0.1', 43470)> ('127.0.0.1', 43470)
CACHE MISS.
[*] Request sent: 127.0.0.1 > neverssl.com
[*] Request sent: 127.0.0.1 > neverssl.com

shehzil@kali: ~
File Actions Edit View Help
[shehzil@kali] ~/
$ curl http://neverssl.com -x localhost:6666
<html>
  <head>
    <title>NeverSSL - Connecting ... </title>
    <style>
      body {
        font-family: Montserrat, helvetica, arial, sans-serif;
        font-size: 16x;
        color: #444444;
        margin: 0;
      }
      h2 {
        font-weight: 700;
        font-size: 1.6em;
        margin-top: 30px;
      }
      p {
        line-height: 1.6em;
      }
      .container {
        max-width: 650px;
        margin: 20px auto 20px auto;
        padding-left: 15px;
        padding-right: 15px;
      }
      .header {
        background-color: #42C0FD;
        color: #FFFFFF;
        padding: 10px 0 10px 0;
        font-size: 2.2em;
      }
      .notice {
```

->Now cache is hit when we searched it again because now the webpage is cached in a cache file.

The screenshot shows two windows from a Kali Linux desktop. The left window is a terminal with the following output:

```
shehzil@kali: ~/Downloads/proxyserver
python2 mainFinal.py
Enter client port: 6666

Socket binded at port: [6666]
neverssl.com
globalhost = neverssl.com
neverssl.com 80 <socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 6666), raddr=('127.0.0.1', 43470)> ('127.0.0.1', 43470)
CACHE MISS.
[*] Request sent: 127.0.0.1 > neverssl.com
[*] Request sent: 127.0.0.1 > neverssl.com
neverssl.com
globalhost = neverssl.com
neverssl.com 80 <socket.socket fd=5, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 6666), raddr=('127.0.0.1', 43472)> ('127.0.0.1', 43472)
CACHE HIT.
```

The right window is a web browser displaying a page with the following HTML content:

```
<p>Normally, that's a bad idea. You should always use SSL and secure encryption when possible. In fact, it's such a bad idea that most websites are now using https by default.</p>

<p>And that's great, but it also means that if you're relying on poorly-behaved wifi networks, it can be hard to get online. Secure browsers and websites using https make it impossible for those networks to send you to a login or payment page. Basically, those networks can't tap into your connection just like attackers can't. Modern browsers are so good that they can remember when a website supports encryption even if you type in the website name, they'll use https.</p>

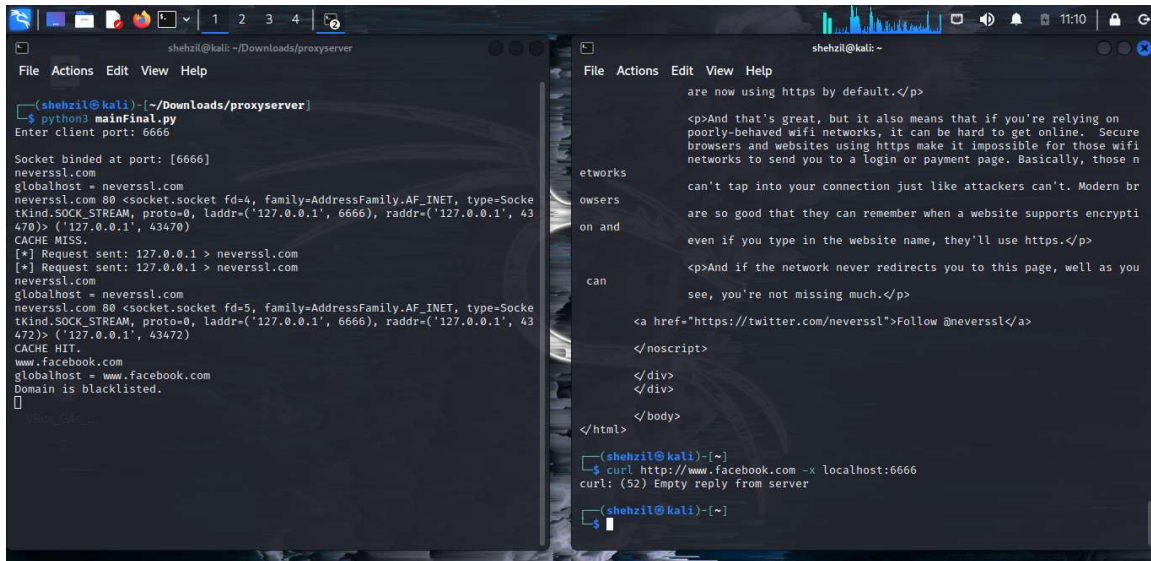
<p>And if the network never redirects you to this page, well as you see, you're not missing much.</p>

<a href="https://twitter.com/neverssl">Follow @neverssl</a>

</noscript>
</div>
</div>
</html>
```

->Now we have added a website '[www.facebook.com](https://www.facebook.com)' in blacklist so now when we try to access it, access will be denied as shown below:

The screenshot shows a web browser window titled '-/Downloads/proxyserver/blacklist/domain - Mousepad'. The address bar shows '1 www.facebook.com' and '2'. The main content area is empty, indicating that access to the website has been denied due to it being in the blacklist.



The image shows two terminal windows. The left window is titled 'shehzil@kali: ~/Downloads/proxyserver' and shows the execution of a Python script 'mainFinal.py'. The script sets up a socket listener on port 6666 and processes incoming requests. It shows a 'CACHE MISS' for 'neverssl.com' and a 'CACHE HIT' for 'www.facebook.com', indicating that the domain is blacklisted. The right window is titled 'shehzil@kali: ~' and shows a text file with HTML content. The content includes a paragraph about secure connections and a link to 'https://twitter.com/neverssl'. Below the text, a terminal command is shown: 'curl http://www.facebook.com -x localhost:6666', which returns '(52) Empty reply from server'.

```
shehzil@kali: ~/Downloads/proxyserver
File Actions Edit View Help
[shehzil@kali] ~/Downloads/proxyserver
$ python2 mainFinal.py
Enter client port: 6666

Socket binded at port: [6666]
neverssl.com
globalhost = neverssl.com
neverssl.com 80 <socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 6666), raddr=('127.0.0.1', 43470)> ('127.0.0.1', 43470)
CACHE MISS.
[*] Request sent: 127.0.0.1 > neverssl.com
[*] Request sent: 127.0.0.1 > neverssl.com
neverssl.com
globalhost = neverssl.com
neverssl.com 80 <socket.socket fd=5, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 6666), raddr=('127.0.0.1', 43472)> ('127.0.0.1', 43472)
CACHE HIT.
www.facebook.com
globalhost = www.facebook.com
Domain is blacklisted.
[]

shehzil@kali: ~
File Actions Edit View Help
are now using https by default.</p>

<p>And that's great, but it also means that if you're relying on poorly-behaved wifi networks, it can be hard to get online. Secure browsers and websites using https make it impossible for those wifi networks to send you to a login or payment page. Basically, those n
etworks
can't tap into your connection just like attackers can't. Modern br
owsers
are so good that they can remember when a website supports encrypti
on and
even if you type in the website name, they'll use https.</p>

<p>And if the network never redirects you to this page, well as you
see, you're not missing much.</p>

<a href="https://twitter.com/neverssl">Follow @neverssl</a>

</noscript>
</div>
</div>
</body>
</html>

[shehzil@kali] ~
$ curl http://www.facebook.com -x localhost:6666
curl: (52) Empty reply from server

[shehzil@kali] ~
```

->After configuring the web browser we can see same effect on the browser as well.Access to facebook is refused.

