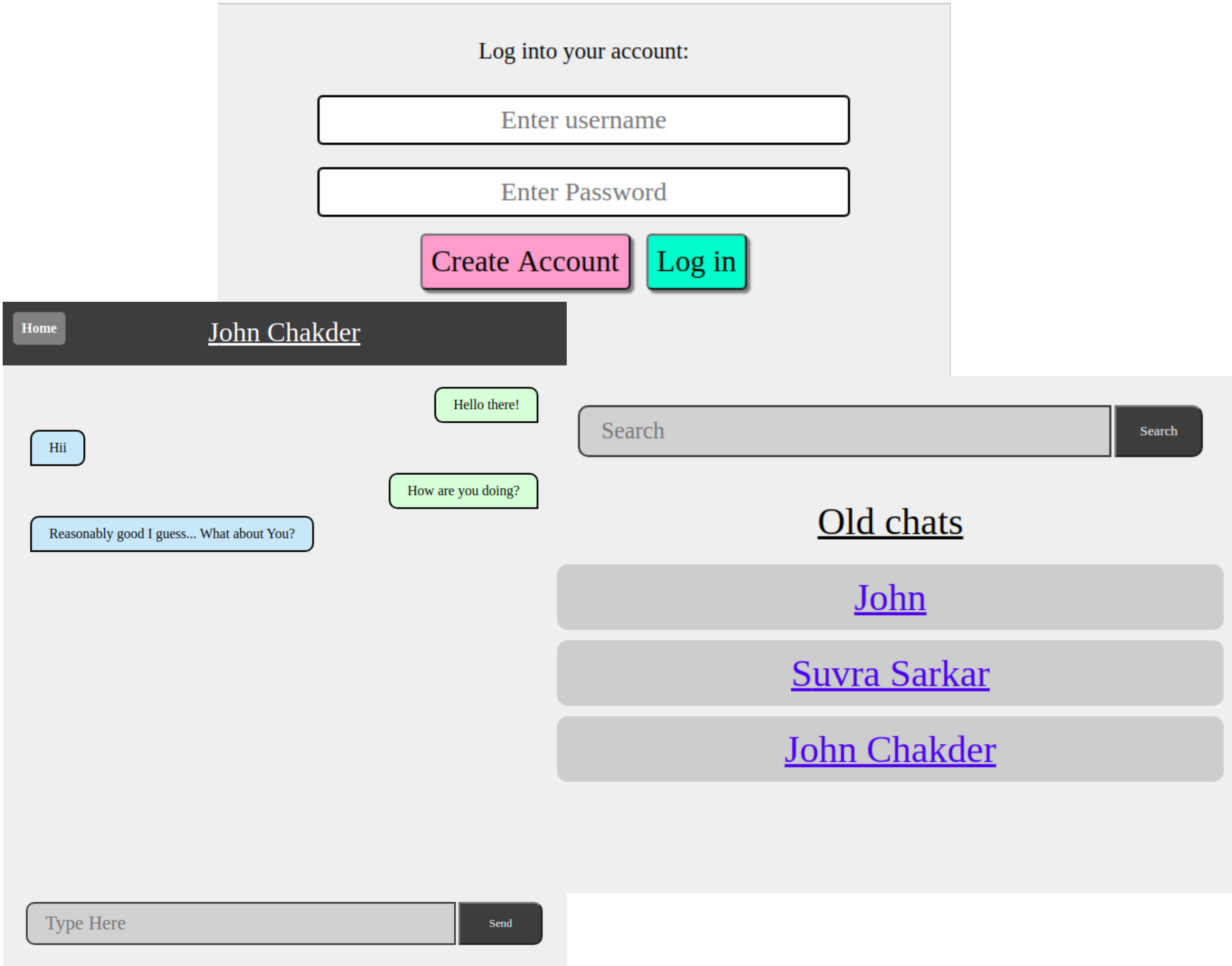
CS547 Assignment 1

Name- Daipayan Chakder. Roll No.- 2011CS02

Part 1

Security services are processing or communication services that improve the security of a system or organization. For this assignment, we have been asked to analyse any readily available open source software or make our own. Taking an open source software and analysing it’s code in this short interval is very challenging for one, so I decided to make a web based software and analyse it’s security aspects.

The software I end up making is a web based messaging application. It is pretty basic chat-app where user can sign-up and search for his/her friend on the platform. One can message others and everything is real-time. (Software can be found at https://github.com/N7K5/CS547_ass01)



Log into your account:

Hello there!

Hii

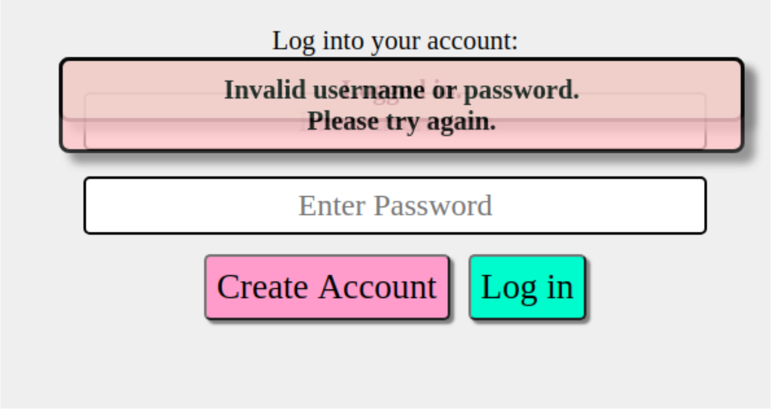
How are you doing?

Reasonably good I guess... What about You?

Image: Screenshots of the software

Though it was made in this short period of time, It was made with some security concerns in mind. for example-

Security- Software security is the idea of making a software such that it continues to function correctly under malicious attack. In the messaging software one need to start his session by logging into the system to send message. security this can be thought as a security feature.

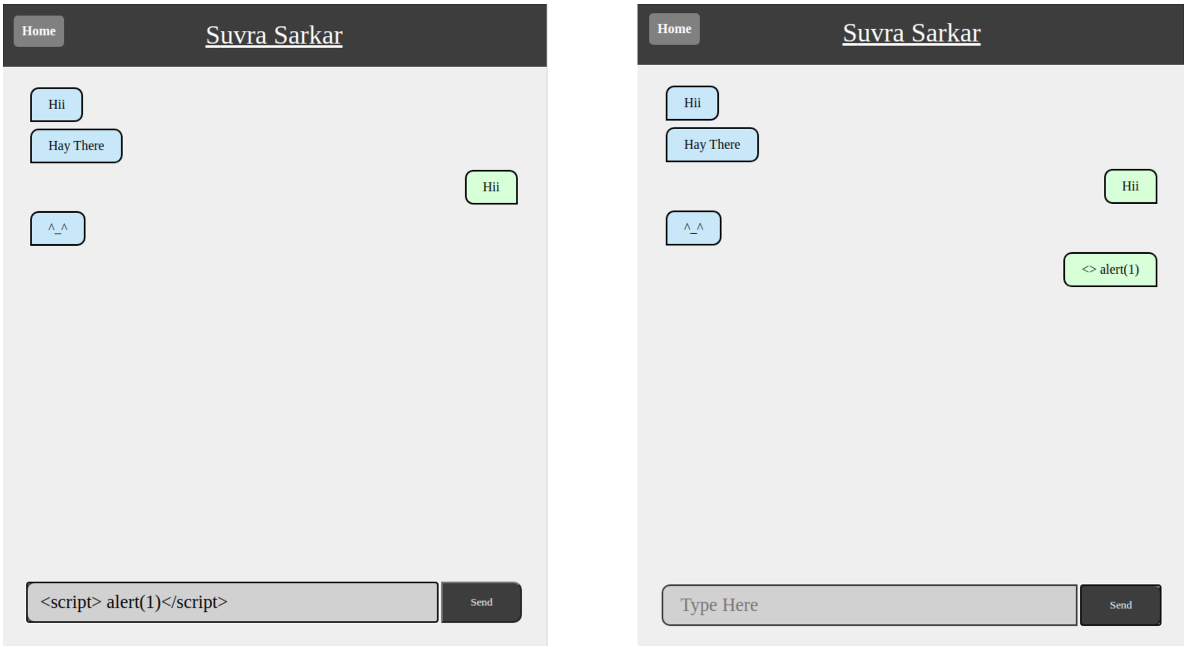


Log into your account:

Invalid username or password.
Please try again.

Enter Password

Threat- Software threats are malicious pieces of code that can damage the computer. Cross site scripting or XSS is one of the biggest threat in web based application where un-modified user data can be seen by other users. The messaging app has handled malicious script by removing any "<script>" from user message. So the user still can use html elements to modify their text while avoiding any un-trusted script execution.



Suvra Sarkar

Hii

Hay There

Hii

^_^

<script> alert(1)</script>

Suvra Sarkar

Hii

Hay There

Hii

^_^

<> alert(1)

Type Here

Image: Removing the Script tag

Vulnerability- Vulnerability in a security application is any bug, flaw, weakness in the code which can be explored by attackers and may lead to failure. In case of the messaging software there can be many combination of input for which the software is vulnerable.

Adversary- Adversary is a malicious entity whose primary goal is to prevent users of a system to use the cryptosystem and therefore preventing the primary goals of cryptosystem such as privacy, integrity or availability of the data. In case of the messaging application, The user's session is is stored as a cookie in the browser. an adversary attacker can steal/change the cookie and getting full access of user-account.

Attack- A software attack is the process of using a system in an un-intended way. The type of attack can vary in a wide range from whole-system attack to social engineering attack like phishing attack. Most of the web based applications are vulnerable to phishing attack and our messaging app is no difference.

Countermeasure- Countermeasure is any action taken by the designer/user to reduce the risk of an attack. Different countermeasures have been taken in our messaging app which includes session maintaining and changing session keys rapidly and un-trusted script execution block.

Risk- Risk encompasses the probability of occurrence for uncertain events and their potential for loss within the software.

Security policy- The security policy is a high-level document that defines an organization's vision concerning security, goals, needs, scope, and responsibilities.

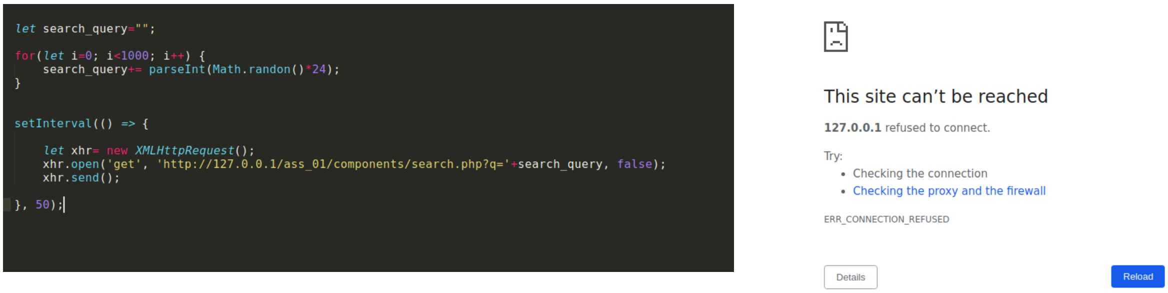
Asset- Asset is any physical/digital component which is owned by an organization. Our messaging software can be an asset of an organization and the may charge to use the software.

Part 2

The application discussed above is made in a very short duration of time and when searching for vulnerabilities / attacks possible, I have come across different attacks which includes -

DoS-

Attack- Denial-of-Service is one of the simplest yet most effective attack to this system. I have write this simple JavaScript code which requests for the page in a 50ms interval. After parallel execution of multiple instances of this code, The site got unresponsive and I had to restart the PHP server to resume the service.



```
let search_query="";

for(let i=0; i<1000; i++) {
    search_query+= parseInt(Math.randon()*24);
}


setInterval(() => {

    let xhr= new XMLHttpRequest();
    xhr.open('get', 'http://127.0.0.1/ass_01/components/search.php?q='+search_query, false);
    xhr.send();

}, 50);
```

This site can't be reached

127.0.0.1 refused to connect.

Try:

- Checking the connection
- Checking the proxy and the firewall

ERR_CONNECTION_REFUSED

Image: DoS code and it's effect

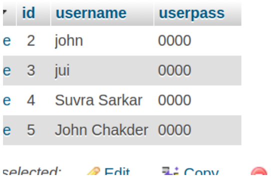
Prevention- This attack can be prevented by using firewall to prevent unwanted request or using powerful server.

Password leak-

Attack- Password leak is another one of the biggest problem of current world. Most of people uses same password for multiple services. Some of the services doesn't Store the hash of the password but the raw password itself. So if one database is compromised, the attacker can access accounts from other services using the passwords.

<u>I have seen the same in the .230 server of our college (the student information server). When I click on the forget password button. It shows the raw password itself.</u>

Our software has the same problem as the password is stored as raw text in the database.



id	username	userpass
2	john	0000
3	jui	0000
4	Suvra Sarkar	0000
5	John Chakder	0000

Image: Un-hashed Password in Database

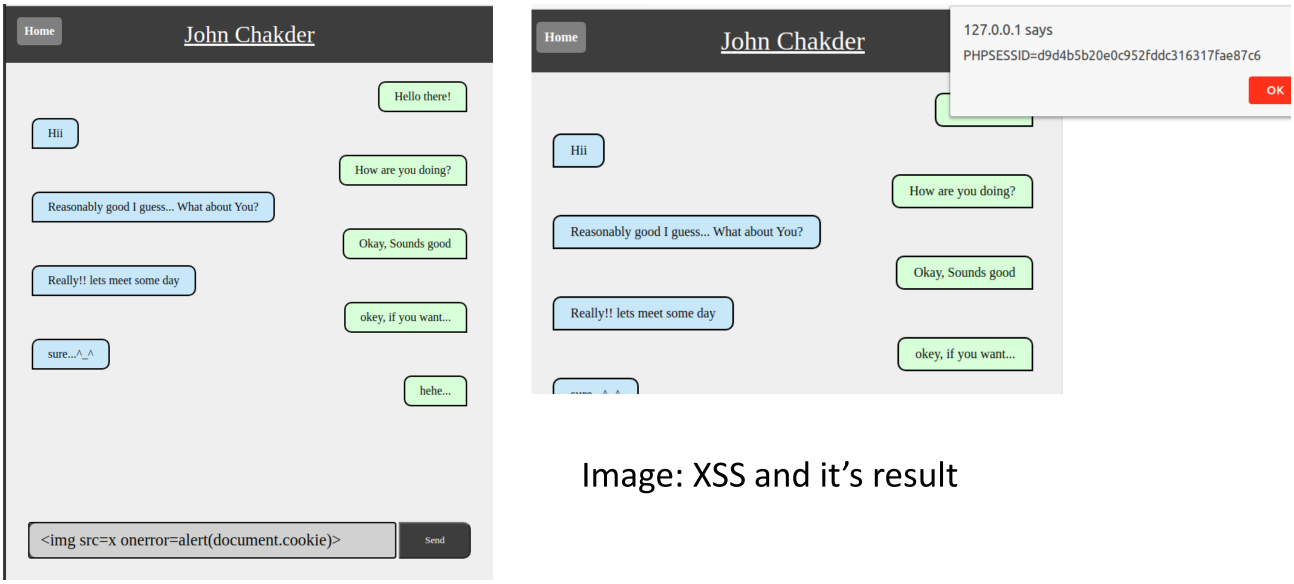
Prevention- Password leak can be easily fixed bt storing the hash value of the password and at the time of varify, farify with the hash instead. Even hash algorithms like md5, SHA1 can be broken by using rainbow tables available in the internet. So, Salting the password before hashing is considered better.

XSS-

Attack- Cross-site Scripting is one of the most well known attack and was popular in social media till early 20s. In this attack, an attacker tries to store his<img src=x onerror=alert(document.cookie)> stript in the database so that it is executed in some other user's machine. Even in 2019, Google have been found to be vulnerable to XSS and have been fixed in their github commit - `c79ab48e8e962fee57e68739c00e16b9934c0ffa`. In case of the messaging app although I have written the code such that no script tag is stored in the server, there are other ways to execute scripts without the script keyword.

<img src=x onerror=alert(document.cookie)>

The above line tries to load an image to the page, and when it does not find the image, it executes the onerror which is a JavaScript function.



Hello there!

Hii

How are you doing?

Reasonably good I guess... What about You?

Okay, Sounds good

Really!! lets meet some day

okey, if you want...

sure...^_^

hehe...

<img src=x onerror=alert(document.cookie)>

127.0.0.1 says

PHPSESSID=d9d4b5b20e0c952fddc316317fae87c6

Hii

How are you doing?

Reasonably good I guess... What about You?

Okay, Sounds good

Really!! lets meet some day

okey, if you want...

Image: XSS and it’s result

Prevention- To prevent XSS, The system has to ensure no style/script related tag is ever in the database. Ensuring this can be tricky but general ideas include - not allowing any '<' or '>' or exchange them with their html equivalent.