Names: PJ Sangvong , Luisa Escosteguy

STRIDE Based Threats and Attack Surfaces

Spoofing identity:

1. Threat: A hacker can make a phishing login webpage and send it to our users to trick them into thinking that they are logging into Tapirs Unlimited. With this, the hacker will be able to obtain their username and passwords.

   Mitigation: We can require users to enable 2-factor authentication on their accounts.

2. Threat: A hacker can pretend to be Tapirs Unlimited and make phone calls to our users, asking them for username and password.

   Mitigation: We can make warning signs/banners, or send emails, that our website will never ask for usernames and passwords.

Tampering with data:

1. Threat: A hacker can use the sign-in form to issue a SQL injection attack to delete some data from the database.

   Mitigation: User data validation in all forms/input fields.

2. Threat: A hacker can manually make POST requests by changing the URL of the website to try to modify the data that the hacker doesn't have access to.

   Mitigation: We need to make sure that requests made by HTTP(S) requests have permission to do what they are requesting to do, by checking the authorization header with the correct permissions.

Repudiation threats:

1. Threat: (Assuming that users don't have to log in to post something) A hacker can try a DDoS attack by spamming requests to create billions of Tapirs posts, overloading the traffic of the server without us being able to trace who posted those posts.

   Mitigation: We need to require log-in for users to be able to post anything so that we can ban users that intentionally try to attack our website.

Information disclosure:

1. Threat: (similar to example) Eavesdroppers can eavesdrop on packets sent between our users and our server through HTTP.

Mitigation: We can require all communications to be done through HTTPS so that eavesdroppers can’t “read” our communications.

2. Threat: A hacker can break into your home office and have physical access to the database, and read/modify users' data.

   Mitigation: Use a cloud provider (e.g. Azure, AWS) database service.

Denial of service:

1. Threat: A hacker can try to post posts that are billions of lines long/very very large images. This can overload the database server, causing data to be erased or the server to go down. If it does not go down, it will still be slower to run queries on, which will still deny service to some of the users.

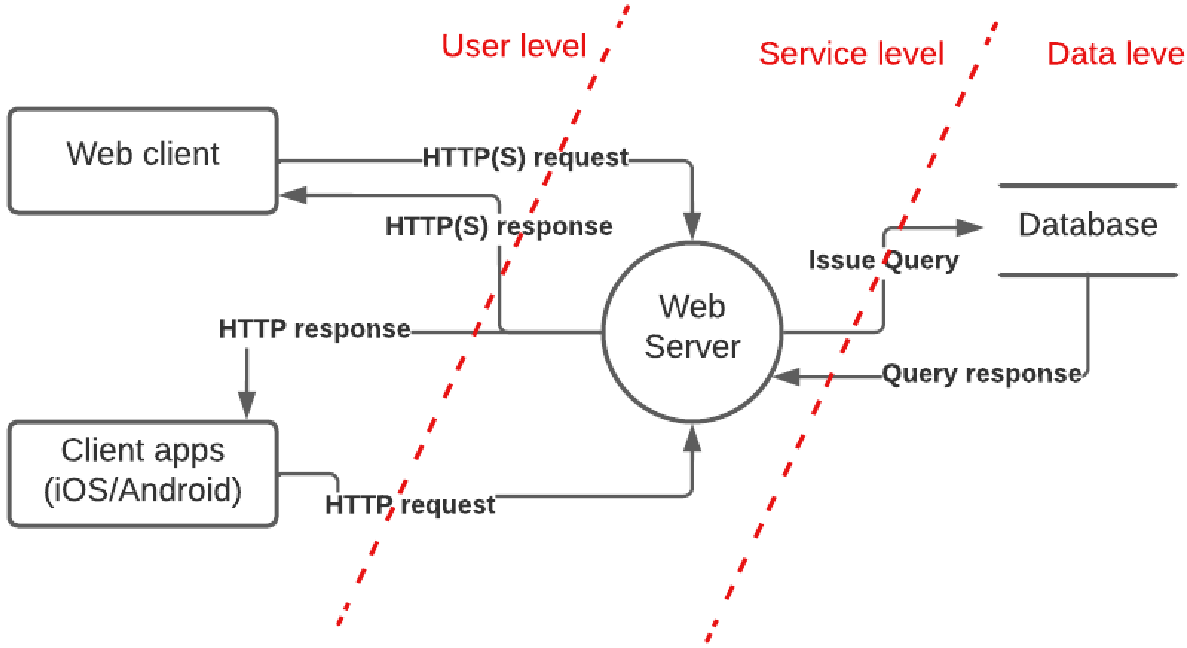
   Mitigation: We can limit the length of each post, and the size of images.

Elevation of privileges:

1. Threat: A user can send a SQL query that elevates its own privileges to admin level via SQL injection.

   Mitigation: User data validation in all forms/input fields.

Data Flow Diagram


User level
Service level
Data level
Web client
HTTP(S) request
HTTP(S) response
Web Server
Issue Query
Database
Query response
HTTP response
Client apps (iOS/Android)
HTTP request