ESOF 422:

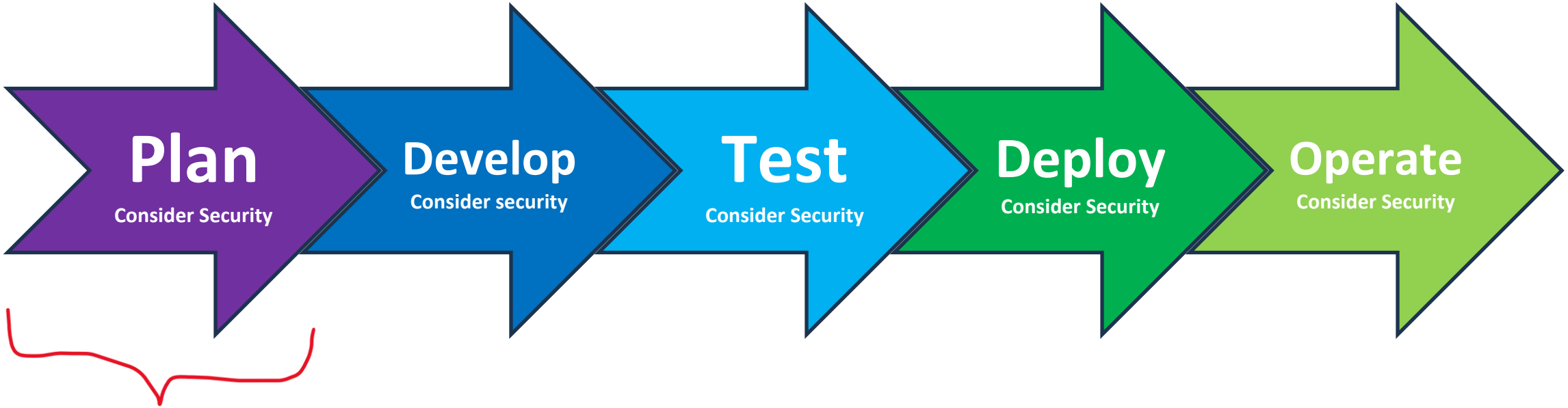
Advanced Software Engineering: Cyber Practices

Threat Modeling, Mitigating Threats, Security Reviews

Reese Pearsall

Spring 2025

During the secure software development lifecycle, we consider security at each step of the development process



During the Planning/Design face, we need to consider design flaws and potential vulnerabilities of the system

Threat Modeling is a formal approach for analyzing the security and potential **risks** of some system

- Risk- The potential for loss and damage when the threat occurs
- Vulnerability- a weaknesses that exposes an organization to a threat
- Exploit- some type of attack/approach to take advantage of a vulnerability

Must think from the perspective of an **attacker**, and identify potential vulnerabilities

Threat modeling allows you to identify and (hopefully) eliminate design issues in your system prior to implementation

It is an industry best practice to validate and implement the defenses that were derived from the threat model

Threat modeling isn't a super complex process, but it is an *incredibly valuable* skill in the cyber world

How do we actually “do” threat modeling?

Threat modeling boils down to answer four essential questions:

1. What are we building? (Provides **scope** and **assets** of threat modeling)
2. What can go wrong? (Identify potential vulnerabilities)
3. What are we going to do about it? (Provide countermeasure for Q2)
4. Did we do a good job? (Reflect on answers, and go back to Q1 if needed)

Other interesting questions worth answering:

- Who are the adversaries?
- What is their motive?

Optional steps of Threat Modeling

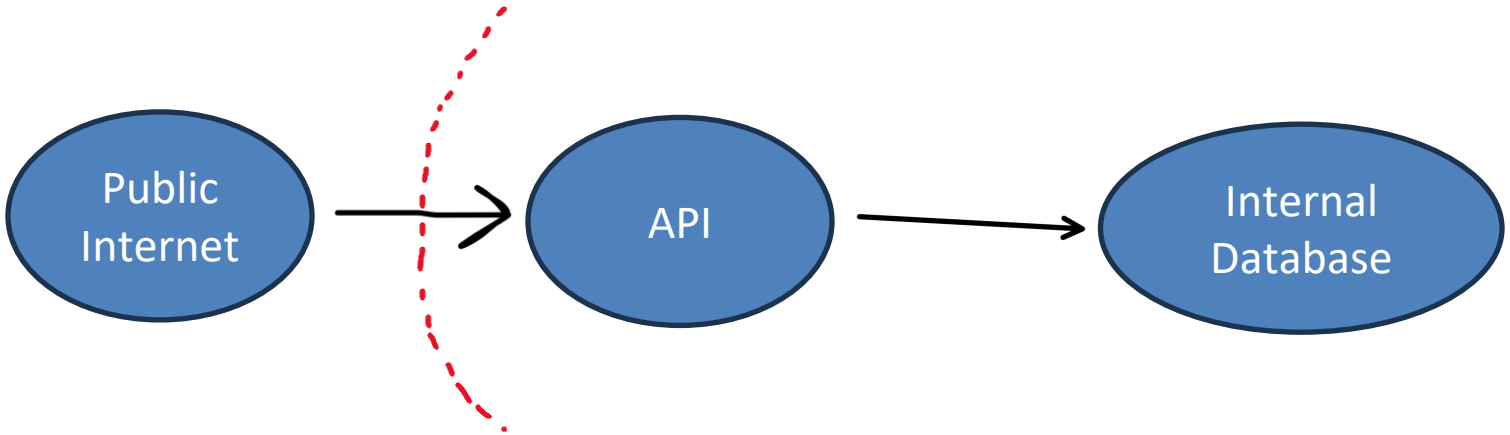
It is helpful to define the *trust boundaries* of the system

Trust Boundary- a point where a system's level of trust changes from untrusted to trusted

Attacks and vulnerabilities typically occur at the trust boundary

Most web applications are exposed to the public internet, which is an untrusted zone

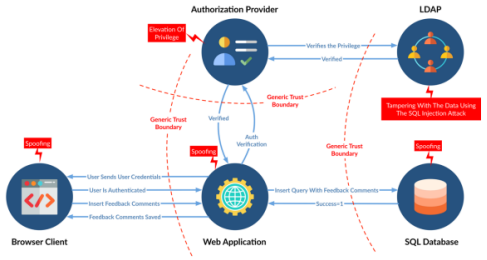
Trusted zones
- Internal networks
- Authenticated information
- Input sanitized



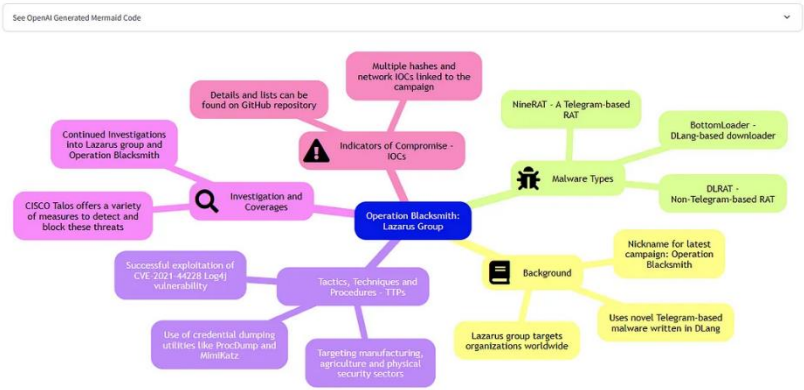
Answering Questions

1. What are we building?
2. What can go wrong?
3. What are we going to do about it?
4. Did we do a good job?

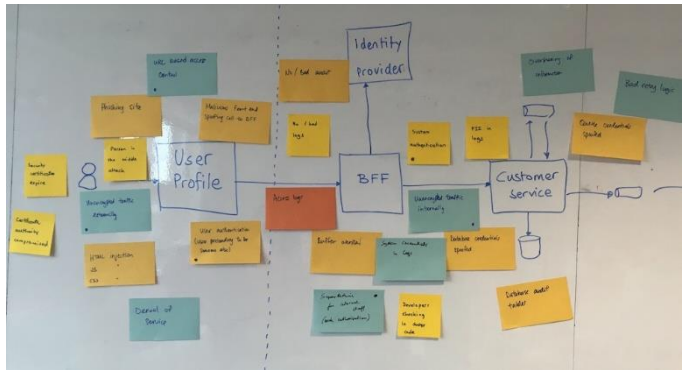
We answering questions, we typically start with a high-level **diagram** of the system and list of requirements



Mind Map



- Brainstorm, Answer in Writing, Whiteboard
- Mind map or Diagram
- Research similar systems
- Generate scenarios
- Use a **structured** threat modeling approach



Whiteboard w/ Sticky Notes

STRIDE

1. What are we building?
2. What can go wrong?
3. What are we going to do about it?
4. Did we do a good job?

STRIDE is a model made by Microsoft for identifying security threats during threat modeling
→ Widely-used in industry
→ Classifies attacks under 6 different categories

Spoofing – illegally accessing and then using another user's authentication information or pretending to be someone else

Ex. Identify theft, spoofing packets to do something malicious

STRIDE

1. What are we building?
2. What can go wrong?
3. What are we going to do about it?
4. Did we do a good job?

STRIDE is a model made by Microsoft for identifying security threats during threat modeling
→ Widely-used in industry
→ Classifies attacks under 6 different categories

Tampering – malicious modification of data (Data integrity violation)

Ex. Database modifications, packet tampering

STRIDE

1. What are we building?
2. What can go wrong?
3. What are we going to do about it?
4. Did we do a good job?

STRIDE is a model made by Microsoft for identifying security threats during threat modeling
→ Widely-used in industry
→ Classifies attacks under 6 different categories

Repudiation – users deny performing a malicious action without parties having any way to prove otherwise

Ex. No logging or auditing done on system

STRIDE

1. What are we building?
2. What can go wrong?
3. What are we going to do about it?
4. Did we do a good job?

STRIDE is a model made by Microsoft for identifying security threats during threat modeling
→ Widely-used in industry
→ Classifies attacks under 6 different categories

Information Disclosure – Exposing data to unauthorized users

Ex. password leaks, API key hardcoded, error message leaking

STRIDE

1. What are we building?
2. What can go wrong?
3. What are we going to do about it?
4. Did we do a good job?

STRIDE is a model made by Microsoft for identifying security threats during threat modeling
→ Widely-used in industry
→ Classifies attacks under 6 different categories

Denial of service– Denying service to a user (Availability violation)

Ex. DDOS attacks

STRIDE

1. What are we building?
2. What can go wrong?
3. What are we going to do about it?
4. Did we do a good job?

STRIDE is a model made by Microsoft for identifying security threats during threat modeling
→ Widely-used in industry
→ Classifies attacks under 6 different categories

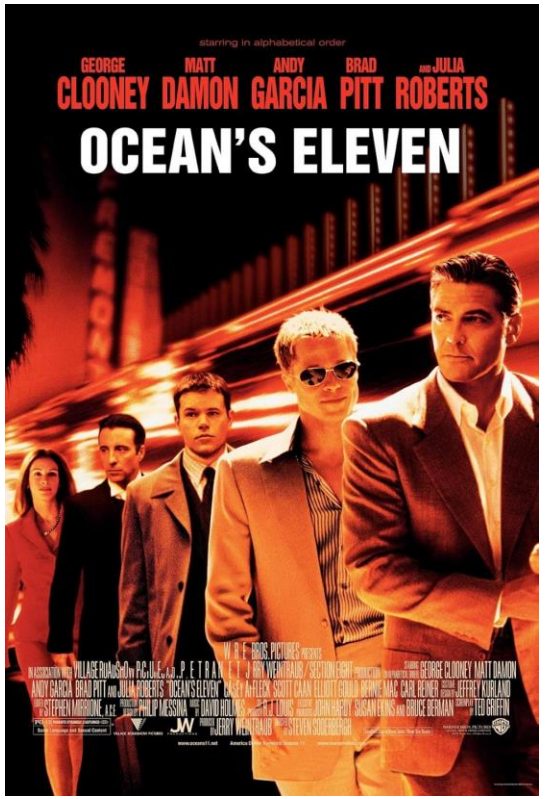
Elevation of Privilege– An unprivileged user gains access and does an action they should not be able to do

Ex. SET-UID program exploit to gain root access

STRIDE

Spoofing
Tampering
Repudiation
Information Disclosure
Denies of Service
Elevation of Privilege

Let’s apply STRIDE threats to the famous heist movie ***Ocean’s Eleven (2001)***



The story follows friends Danny Ocean (Clooney) and Rusty Ryan (Pitt), who plan a heist of $160 million from casino owner Terry Benedict (García), the lover of Ocean's ex-wife Tess (Roberts).

STRIDE

Spoofing
Tampering
Repudiation
Information Disclosure
Denial of Service
Elevation of Privilege

Danny violates his parole and flies out to meet his partner in crime, Rusty

With low permissions (parole) he is able to escape and fly across the country → **Elevation of Privilege**

STRIDE

Spoofing
Tampering
Repudiation
Information Disclosure
Denial of Service
Elevation of Privilege

Danny and Rusty meet with a casino insider, who provides them with sensitive operational details of the victim casino → **Information Disclosure**

STRIDE

Spoofing
Tampering
Repudiation
Information Disclosure
Denial of Service
Elevation of Privilege

Before the heist, Danny is apprehended by security, which gives him a perfect alibi → **Repudiation** of guilt

STRIDE

Spoofing
Tampering
Repudiation
Information Disclosure
Denial of Service
Elevation of Privilege

Danny and his team end up stealing half of money from the vault later that night (**Tampering** of integrity of vault)

STRIDE

Spoofing
Tampering
Repudiation
Information Disclosure
Denial of Service
Elevation of Privilege

Rusty threatens to blow up the entire vault if the casino doesn't allow them to steal half of the money (a very expensive **Denial of Service**)

STRIDE

Spoofing
Tampering
Repudiation
Information Disclosure
Denial of Service
Elevation of Privilege

The team impersonates as the Las Vegas SWAT to extract the money → **Spoofing and Elevation of Privilege**

Mitigating Risks for Threat Modeling

1. What are we building?
2. What can go wrong?

3. What are we going to do about it?

- **Mitigate** the risk by either redesigning or add defense (most common)
- **Remove** the threatened asset entirely, or reduce exposure
- **Transfer** the risk to a third party, usually in exchange for compensation
- **Accept** the risk and accept consequences

4. Did we do a good job?

Let’s get some hands-on practice

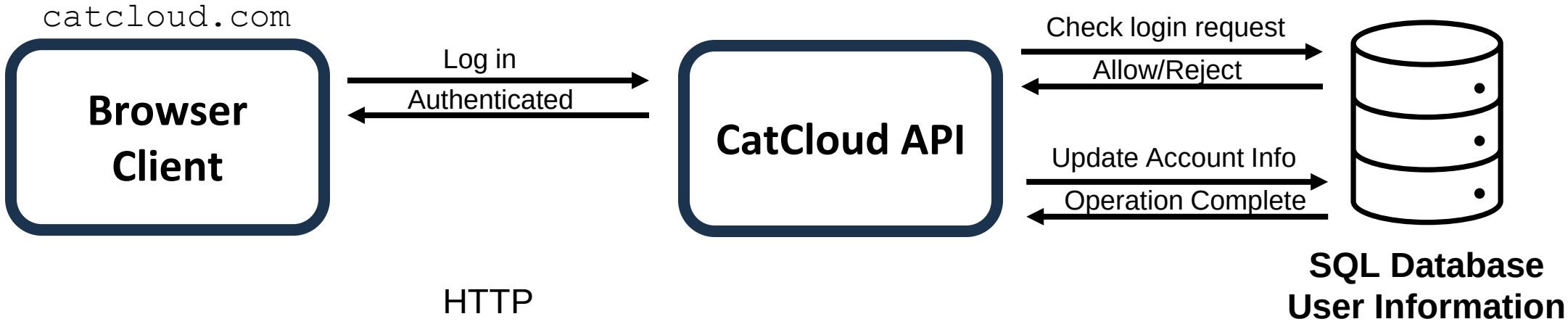
CatCloud if a file-sharing web application that allows users to upload and share files that are stored on some remote server

CatCloud if a file-sharing web application that allows users to upload and share files that are stored on some remote server

- Users log in to website with a username and password

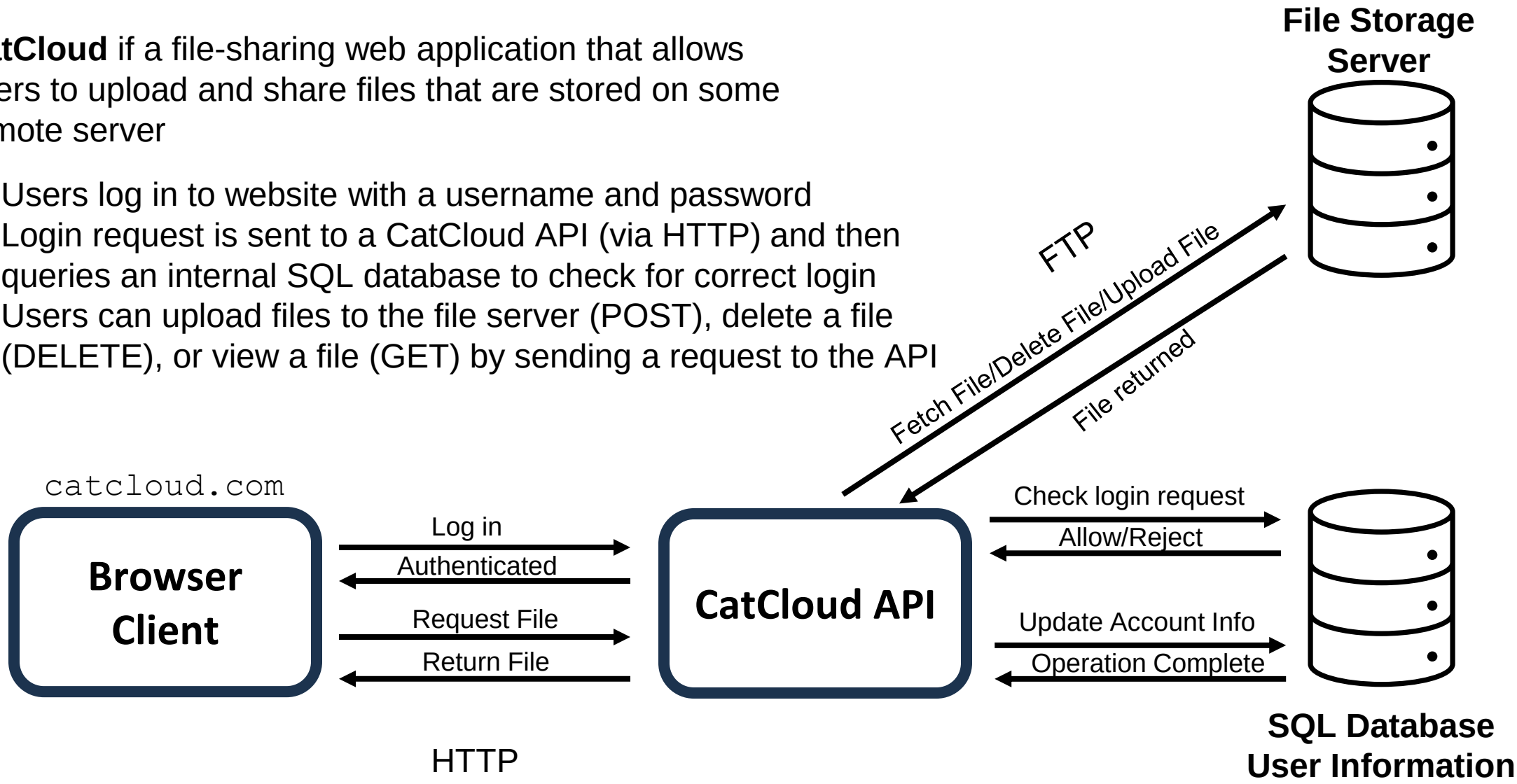
CatCloud if a file-sharing web application that allows users to upload and share files that are stored on some remote server

- Users log in to website with a username and password
- Login request is sent to a CatCloud API (via HTTP) and then queries an internal SQL database to check for correct login



CatCloud if a file-sharing web application that allows users to upload and share files that are stored on some remote server

- Users log in to website with a username and password
- Login request is sent to a CatCloud API (via HTTP) and then queries an internal SQL database to check for correct login
- Users can upload files to the file server (POST), delete a file (DELETE), or view a file (GET) by sending a request to the API




```
GET catcloudapi.com/retrieve?filename=meatball&userid=12053754

DELETE catcloudapi.com/delete?filename=embarrassingpicture&userid=12053754
```

Group Activity Time !

Go to a whiteboard with a group of (2-4) students (or you can work alone) and threat model this system!

1. **What are the assets of CatCloud ?**
2. **What can go wrong?**

- There are *many answers here.* Step into the shoes of an evil hacker
- This could be generally risk, or a specific type of attack
- Think about common weaknesses, lack of requirements, and stuff from CSCI 476, 466, 460, 440
- For each weakness state what kind of CIA violation it is
- You can just do bullet points, mind maps, or STRIDE

3. **What can we do about it?**

- For each item from Q2, propose a countermeasure that would mitigate or remove the threat

Please write your name(s) next to your threat model ☺