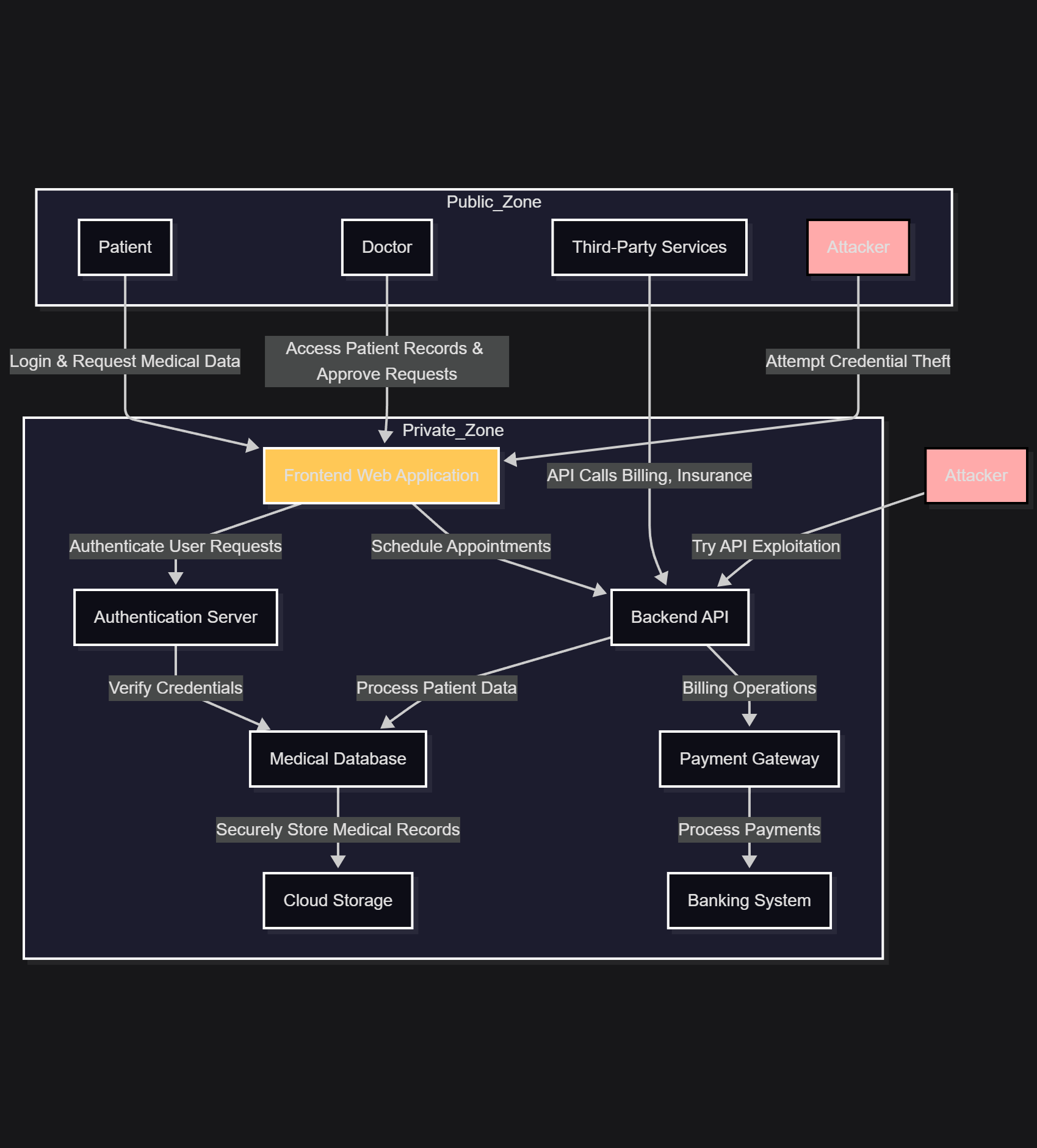
**Complete Lab Exercise: Applying STRIDE to a Cloud-Based Healthcare Portal**

**Scenario: Cloud-Based Healthcare Portal Security**

**MedSecure** is a cloud-based healthcare platform that allows patients to access medical records, schedule appointments, and communicate with doctors online. To ensure security, students will apply **STRIDE threat modeling** to analyze **potential risks and mitigation strategies** for the system.



**🛠 Lab Overview**

**Objectives**

🔹 Understand **STRIDE threat modeling** and how to apply it.  
🔹 Define **trust boundaries and isolate non-critical components** from scope.  
🔹 Create a **Data Flow Diagram (DFD)** using **draw.io desktop**.  
🔹 Identify threats and **prioritize security measures**.

**Tools Required**

✅ **draw.io desktop** – For creating DFDs  
✅ **Microsoft Threat Modeling Tool** (optional) – For automated analysis  
✅ **Web browser** – For referencing security best practices

**📍 Step 1: Define System Scope & Trust Boundaries**

**System Components**

✔ **Frontend Web Application** – Patient portal for login and appointment booking.  
✔ **Backend APIs** – Handles medical data retrieval and doctor interactions.  
✔ **Database Storage** – Stores sensitive patient records and prescriptions.  
✔ **Third-Party Services** – External integrations like payment processing for billing.

**Trust Boundaries**

🔹 **Public vs. Private Zones:**

* Separate **public patient interface** from **internal hospital systems**.
* Restrict direct access to **backend database and admin dashboards**.  
  🔹 **API Access & Authentication:**
* **Secure API endpoints** for appointment scheduling and data retrieval.
* **Enforce authentication layers** (OAuth, MFA).

🛠 **Lab Task:**

* Students will **define trust boundaries** using **draw.io**
* Label **restricted zones** vs. **public-facing components**

**Step 1: Identify Key Assets**

**Assets** are valuable system components that need security measures.

| **Asset Type** | **Description** | **Security Concern** |
| --- | --- | --- |
| **Patient Data** | Medical records, prescriptions, and diagnosis history. | **Privacy breaches, data theft**. |
| **User Credentials** | Login credentials for patients and healthcare staff. | **Spoofing, unauthorized access**. |
| **Appointment System** | Stores scheduled doctor visits and prescriptions. | **Tampering, denial of service**. |
| **Payment Details** | Billing information for healthcare services. | **Repudiation, fraud risks**. |
| **API Services** | Cloud-based APIs handling communication. | **API injections, data exposure**. |
| **Healthcare Provider Access** | Admin dashboards used by doctors and staff. | **Privilege escalation risks**. |

🎯 **Task:**  
🔹 **Label assets in your Data Flow Diagram (DFD) using draw.io**.  
🔹 **Define security risks per asset and assign priority levels**.

**📌 Step 2: Identify & Classify Threat Actors**

Threat actors are **individuals, groups, or automated processes** attempting to **exploit system vulnerabilities**.

| **Threat Actor** | **Potential Actions & Risks** | **Affected Component** |
| --- | --- | --- |
| **Malicious Hackers** | Attempt credential theft, exploit API weaknesses. | **Login system, API**. |
| **Insider Threats** | Staff with authorized access tampering with patient records. | **Medical database, admin panel**. |
| **Cybercriminal Groups** | Large-scale ransomware attacks, fraudulent medical billing. | **Payment gateway, patient data**. |
| **Competitor Attacks** | Data harvesting for competitive advantage. | **Medical research records**. |
| **Automated Bots** | Brute-force login attacks, flooding appointment systems. | **Authentication & scheduling services**. |

🎯 **Task:**  
🔹 **Review potential threats based on STRIDE classification**.  
🔹 **Update threat actors in your security model & DFD documentation**.

**📌 Expected Learning Outcomes**

✔ **Improved threat awareness** by identifying **key assets & attack vectors**.  
✔ **STRIDE-based threat classification** for **proactive risk mitigation**.  
✔ **Hands-on experience with draw.io & DFD threat mapping**.

🚀 **Would you like modifications or additional components?** 😊  
I can also **guide you on refining security measures for industry-specific compliance**!

**📍 Step 2: Create a Data Flow Diagram (DFD) in draw.io**

**How to Create a DFD Using draw.io:**

1️ Open **draw.io desktop** and create a **new blank diagram**.  
2️Add **shapes representing system components:**  
✔ **External Entities** (Patients, Doctors, Third-Party Services)  
✔ **Processes** (Authentication System, Appointment Scheduler)  
✔ **Data Stores** (Encrypted Patient Database)  
3️ **Draw arrows** showing data flow between components.  
4️ **Add trust boundaries** around restricted systems.

🔹 **Outcome:** A complete **DFD representing healthcare portal security architecture**.

Students **categorize threats for each system component** using **STRIDE methodology**:

| **STRIDE Category** | **Threat Example** | **Affected Component** | **User Decision: Prioritize or Ignore?** |
| --- | --- | --- | --- |
|  | Attackers impersonate doctors/patients for unauthorized access. | **Login System** | 🔲 Prioritize 🔲 Ignore |
|  | Modification of medical prescriptions in database. | **Medical Database** | 🔲 Prioritize 🔲 Ignore |
|  | Users claim they didn’t book an appointment. | **Appointment Scheduling System** | 🔲 Prioritize 🔲 Ignore |
|  | Patient records exposed due to API vulnerabilities. | **API Services & Cloud Storage** | 🔲 Prioritize 🔲 Ignore |
|  | Attackers flood appointment booking, preventing access. | **Frontend Web App** | 🔲 Prioritize 🔲 Ignore |
|  | Unauthorized admin access to modify hospital records. | **Admin Dashboard & Database** | 🔲 Prioritize 🔲 Ignore |

🎯 **Task:**  
🔹 Review each **STRIDE threat** and **prioritize the risks** based on impact & likelihood.  
🔹 Mark **critical areas in draw.io** using **warning symbols** (red borders).  
🔹 Provide **justifications for prioritizing or ignoring threats**

🛠 **Lab Task:**

* Students will **prioritize threats** based on **severity & impact**.
* Mark **high-risk areas** in **draw.io** using warning symbols.

**📍 Step 4: Implement Security Controls**

✅ **Frontend Security:** CSP headers, secure cookies, XSS prevention.  
✅ **Backend Security:** OAuth 2.0 authentication, input validation, rate limiting.  
✅ **Database Security:** Encrypted storage, parameterized queries, access control lists.  
✅ **Network Security:** TLS 1.3 encryption, VPN for internal hospital services.

🛠 **Lab Task:**

* Apply **security controls** in the **draw.io model**.
* Define **access restrictions for each component**.

**📍 Step 5: Generate Security Report**

📝 **Short Security Report – Cloud Healthcare Portal Security**

✔ **Scope Defined:** Protecting authentication, medical records, and appointment scheduler.  
✔ **Threats Addressed:** Spoofing, tampering, API vulnerabilities.  
✔ **Security Measures Applied:** MFA, API encryption, WAF, RBAC.  
✔ **Conclusion:** The threat model **ensures patient data security, compliance, and operational integrity**.

Sample report