THREAT MODELING

PROACTIVE RISK IDENTIFICATION

STRIDE

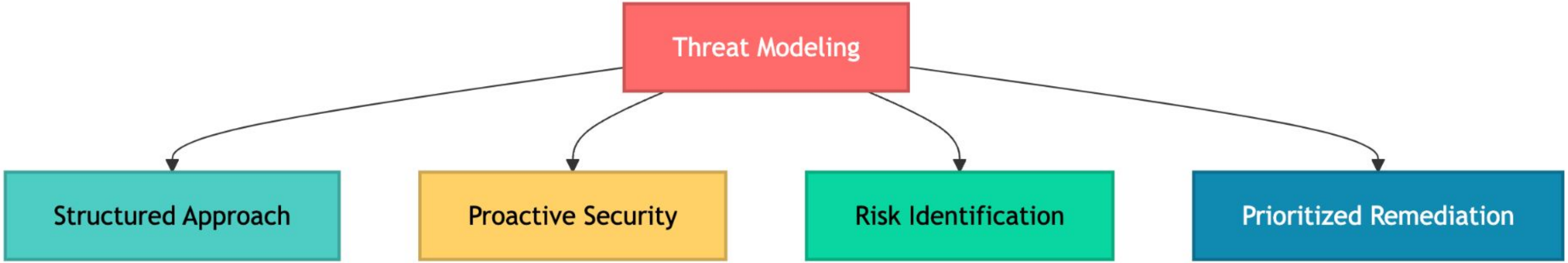
A Systematic Approach to Identify Security Risks

By Saravana Priya
09 March 2025

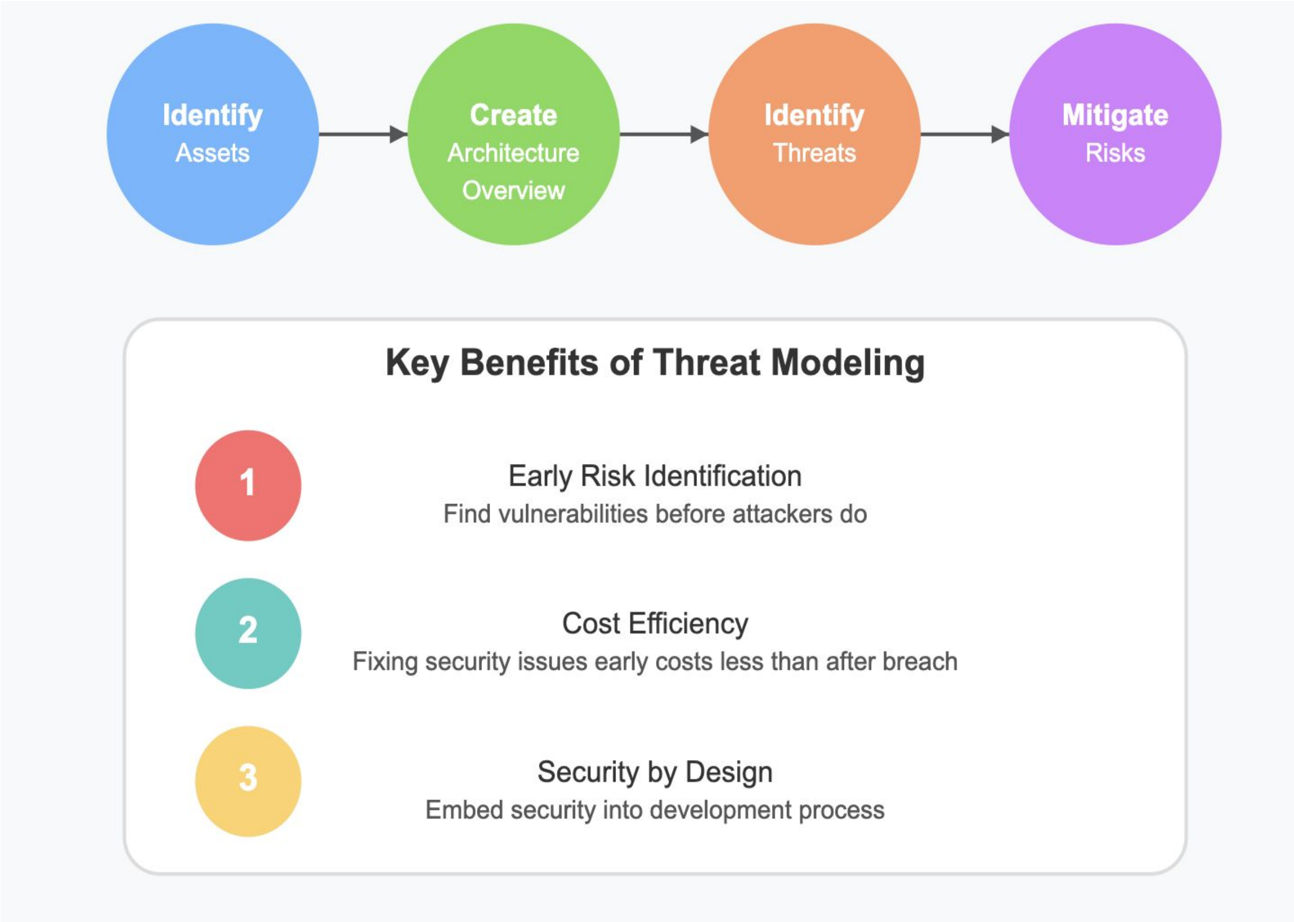
Agenda

- Introduction to Threat Modeling
- Why Threat Modeling ?
- Threat Modeling Frameworks
- Threat Modeling Process
- Threat Modeling - User Registration Module
- Tools & Resources
- Quiz

What is Threat Modeling ?



Why Threat Modeling ?



Key Benefits of Threat Modeling

1 Early Risk Identification

Find vulnerabilities before attackers do

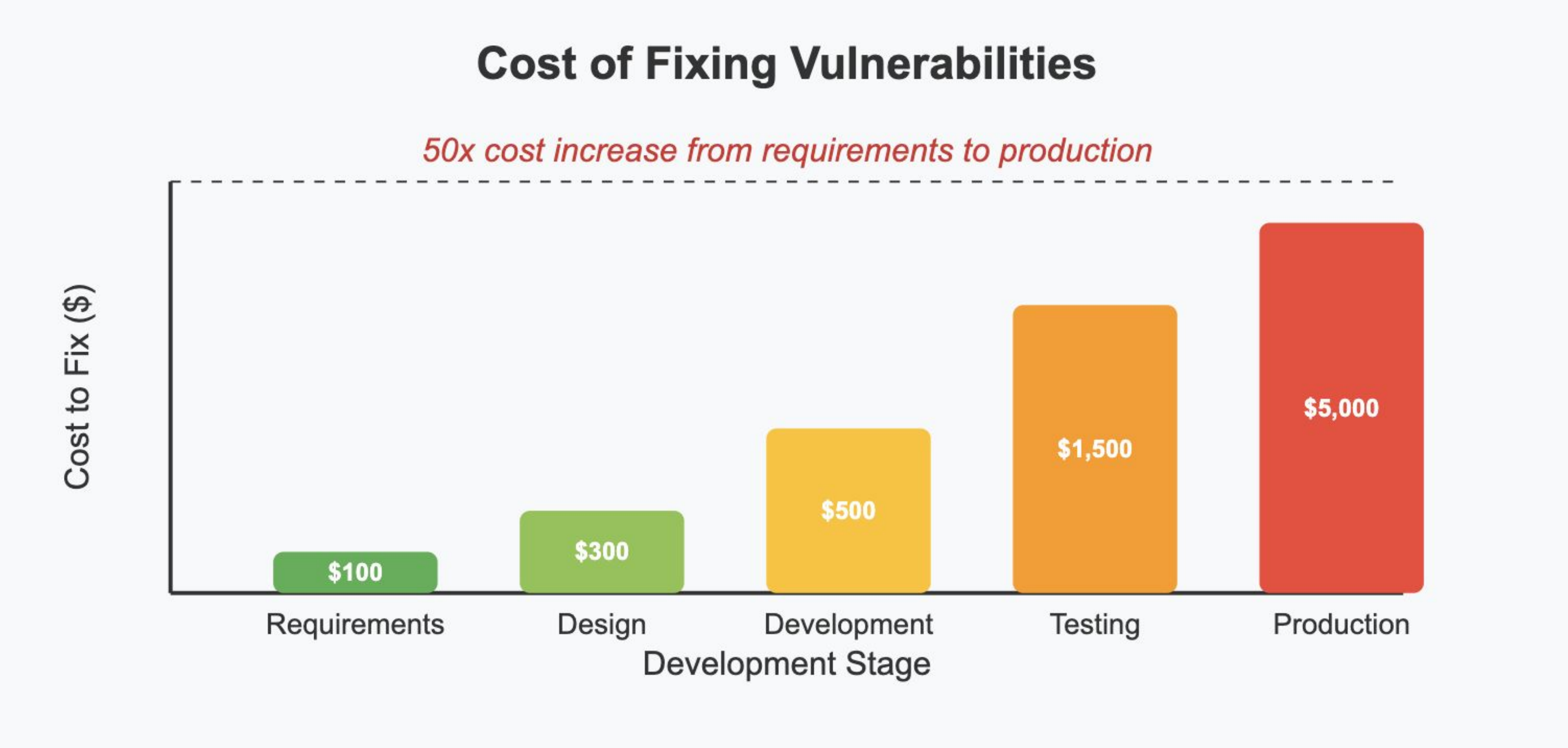
2 Cost Efficiency

Fixing security issues early costs less than after breach

3 Security by Design

Embed security into development process

Why Threat Modeling ?



Threat Modeling Frameworks

1. **STRIDE** (Microsoft)

2. **PASTA** (Process for Attack Simulation and Threat Analysis)
    - Business-oriented and risk-centric approach
    - Seven-stage process from defining objectives to residual risk analysis

3. **OCTAVE** (Operationally Critical Threat, Asset, and Vulnerability Evaluation)
    - Focuses on organizational risk assessment
    - Well-suited for enterprise-level systems

Threat Modeling Frameworks

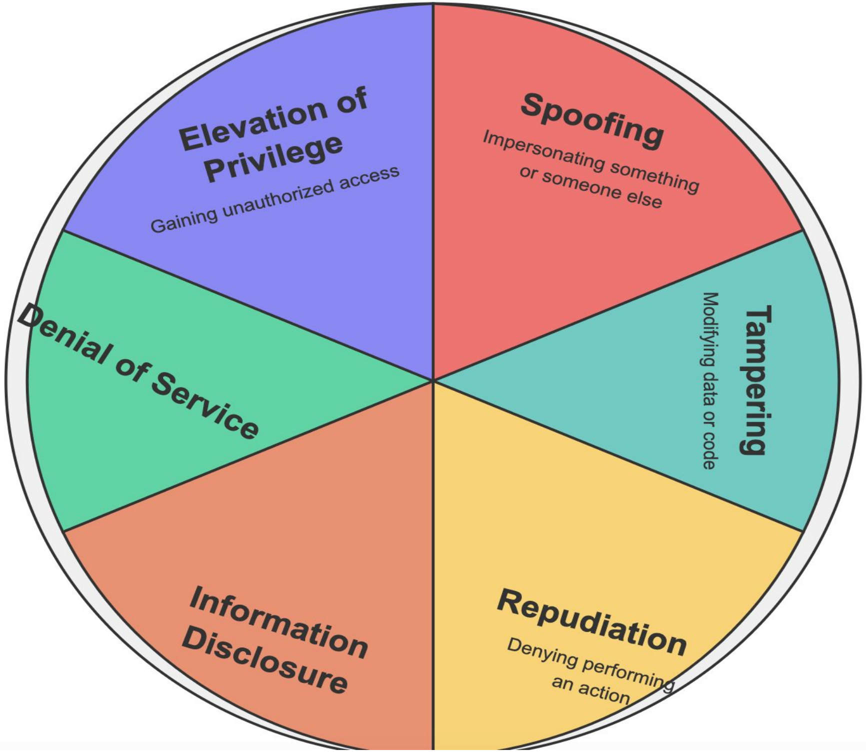
4. VAST (Visual, Agile, and Simple Threat modeling)

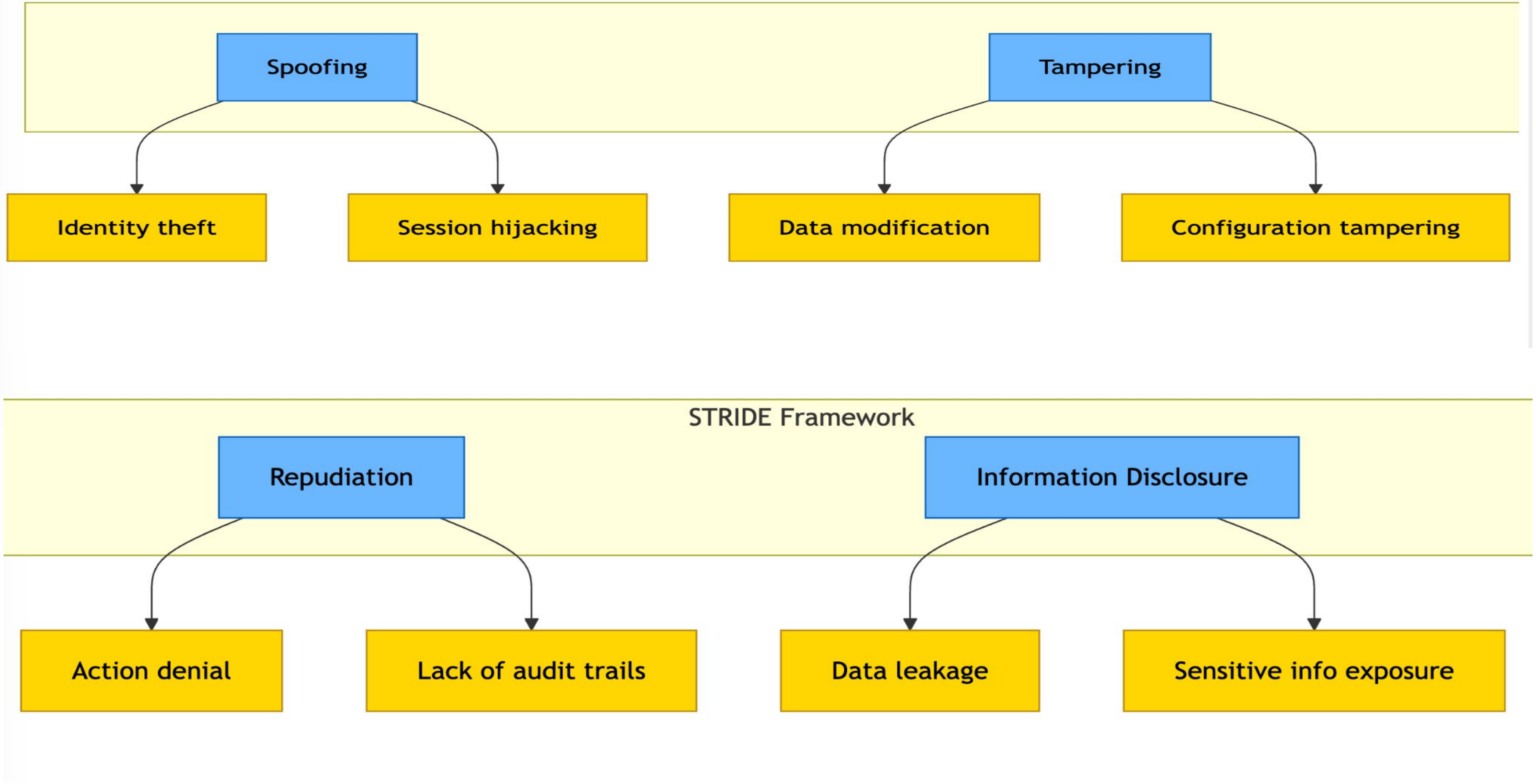
- ○ Designed for scalable threat modeling in Agile environments
- ○ Uses visual tools and automation

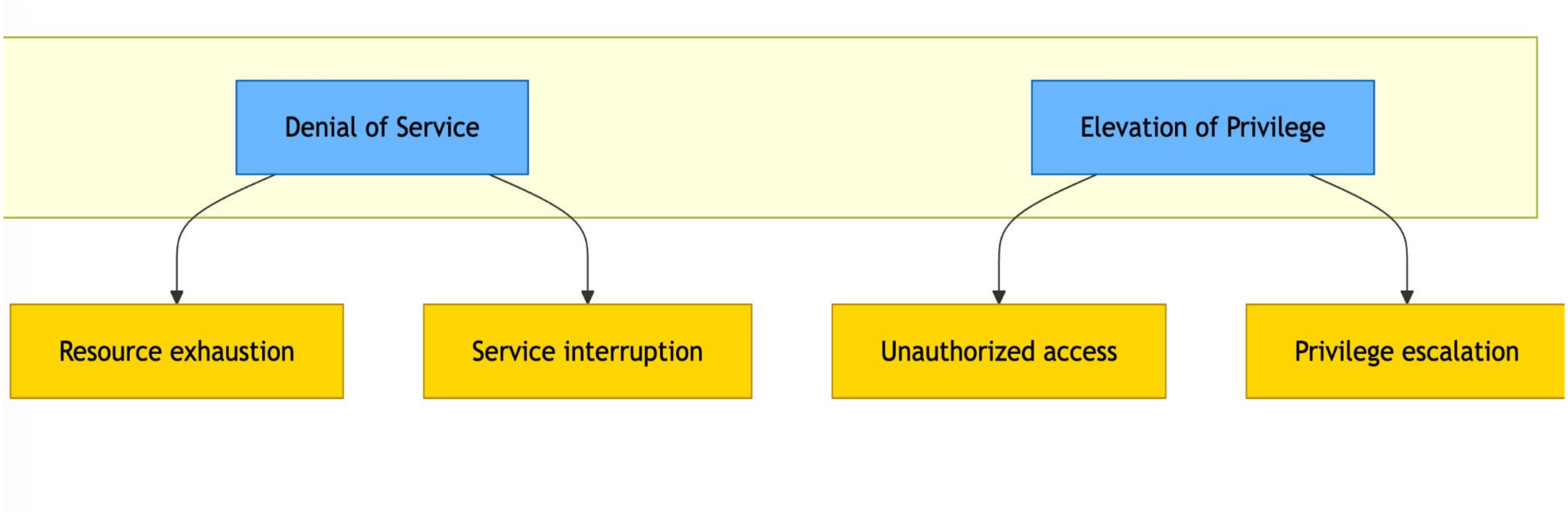
5. DREAD (Microsoft)

- ○ Damage
- ○ Reproducibility
- ○ Exploitability
- ○ Affected users
- ○ Discoverability
- ○ Used for risk rating after threats are identified

STRIDE Threat Modeling Framework




Spoofing
Tampering
Identity theft
Session hijacking
Data modification
Configuration tampering
STRIDE Framework
Repudiation
Information Disclosure
Action denial
Lack of audit trails
Data leakage
Sensitive info exposure


Denial of Service

Resource exhaustion

Service interruption

Elevation of Privilege

Unauthorized access

Privilege escalation

Things to consider - STRIDE Considerations

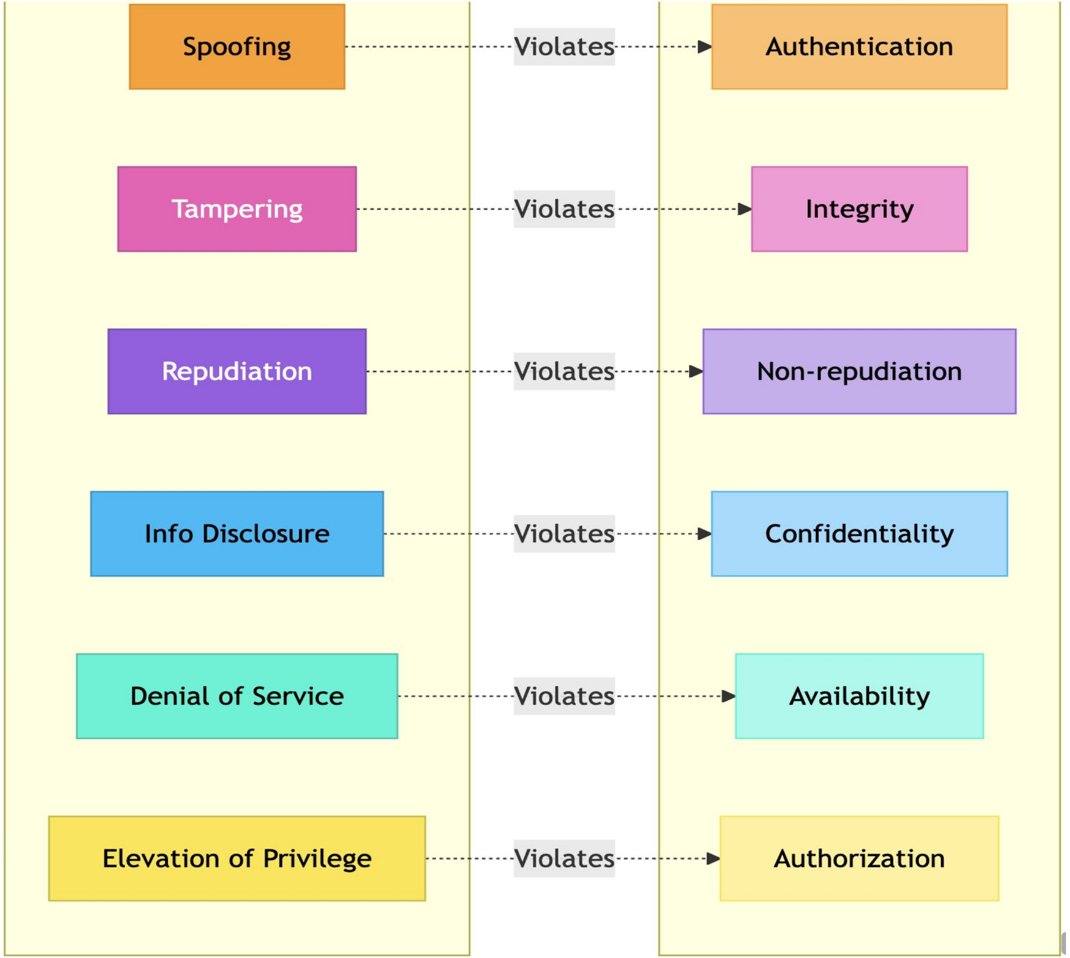
"**STRIDE-per-Element -** that not all threats apply to all system elements.

- **Data flows**
- **Data stores**
- **Processes**
- **External entities**

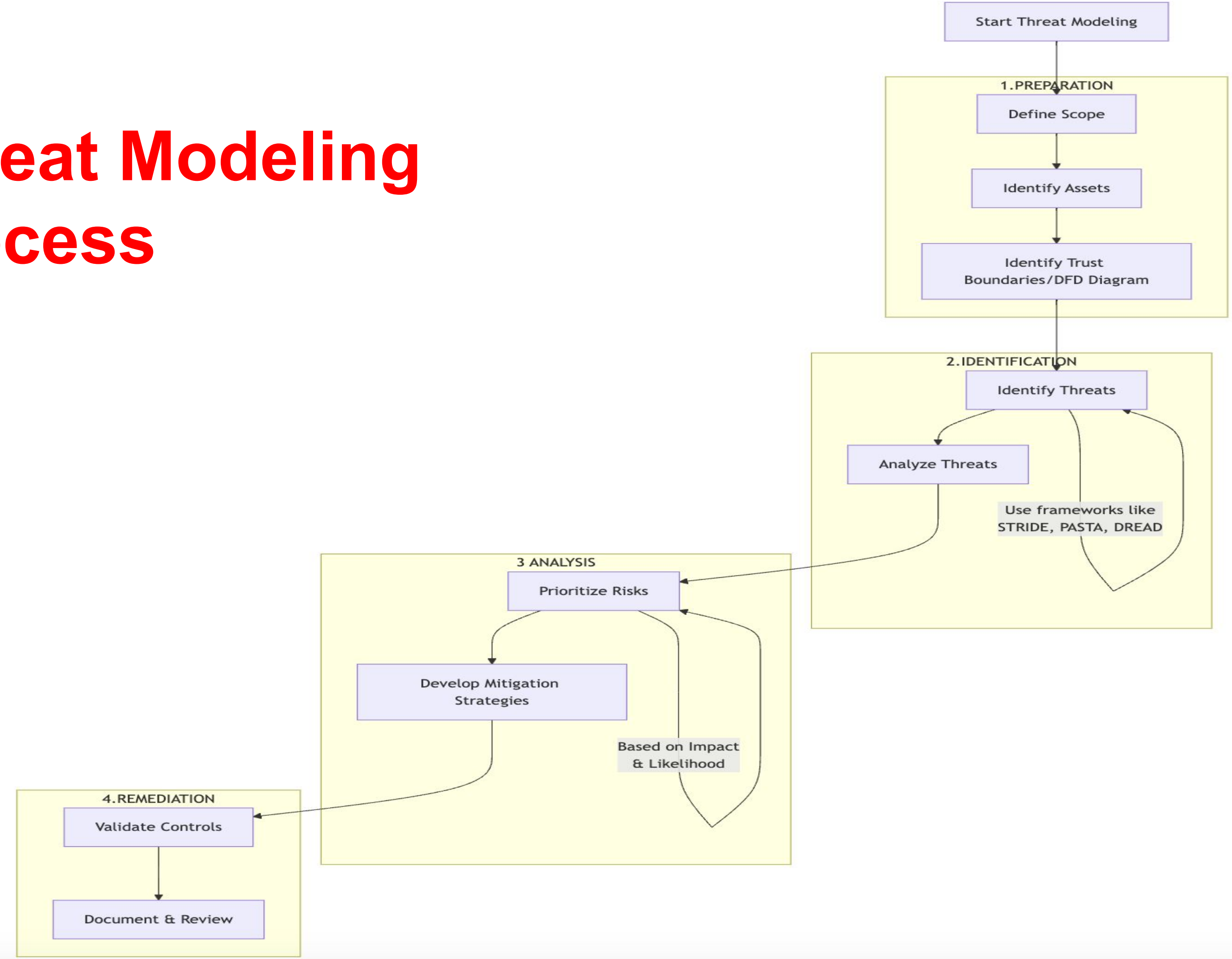
Things to consider - STRIDE Considerations

- **Data flows** - vulnerable to Tampering, Information Disclosure, and Denial of Service
- **Data stores** - vulnerable to Tampering, Information Disclosure, Repudiation, and Denial of Service
- **Processes** - vulnerable to all six STRIDE categories
- **External entities** are primarily concerned with Spoofing and Repudiation

This refinement helps focus the threat modeling effort more efficiently.


Spoofing
Violates
Authentication
Tampering
Violates
Integrity
Repudiation
Violates
Non-repudiation
Info Disclosure
Violates
Confidentiality
Denial of Service
Violates
Availability
Elevation of Privilege
Violates
Authorization

Threat Modeling Process



Start Threat Modeling

1.PREPARATION

Define Scope

Identify Assets

Identify Trust Boundaries/DFD Diagram

2.IDENTIFICATION

Identify Threats

Analyze Threats

Use frameworks like STRIDE, PASTA, DREAD

3 ANALYSIS

Prioritize Risks

Develop Mitigation Strategies

Based on Impact & Likelihood

4.REMEDIATION

Validate Controls

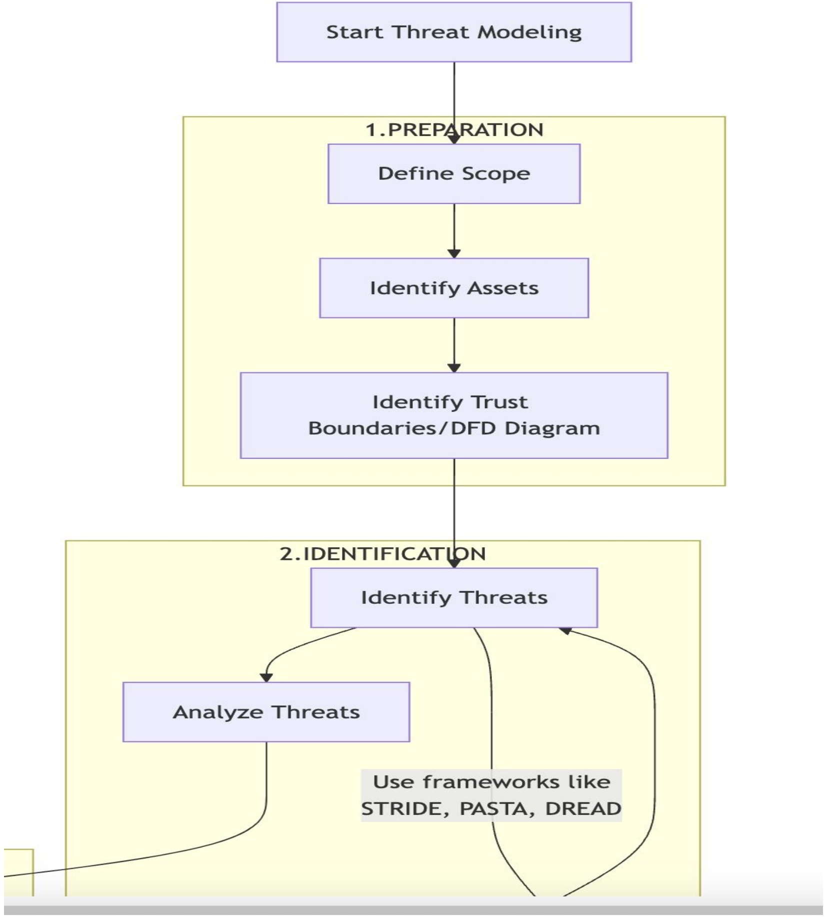
Document & Review

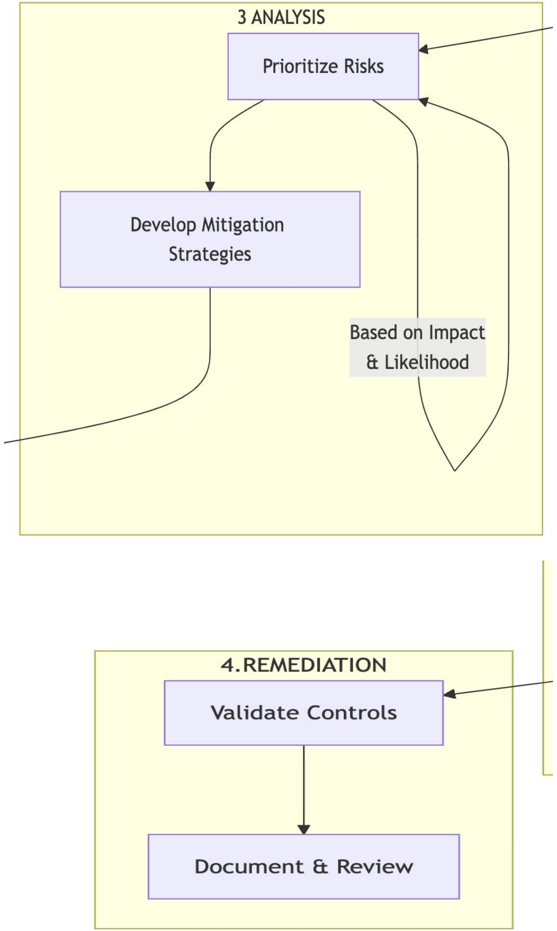
Threat Modeling Process

Phases of Threat modeling

- Preparation Phase
- Identification Phase
- Analysis Phase
- Remediation Phase

Threat Modeling Process





Threat Modeling Process

Step 1: Define the Scope

Define the Scope

- Identify the specific system components to be evaluated
- For our user registration system: registration form, email verification, password management, authentication, and user dashboard
- Document system boundaries and external dependencies

Step 2: Identify Assets **(**Determine what needs protection)

- User credentials and personal information
- Authentication tokens and session data

The Threat Modeling Process

Step 3: Identify Trust Boundaries

- Mark points where data crosses different trust zones

Step 4: Identify Threats using any Frameworks

- Apply threat frameworks (STRIDE, PASTA, etc.)
- Consider different attacker motivations and capabilities

Threat Modeling Process

Step 5: Analyze and Prioritize Threats

- Assess likelihood and impact
- Calculate risk scores
- Prioritize based on business context

Step 6: Mitigate Risks

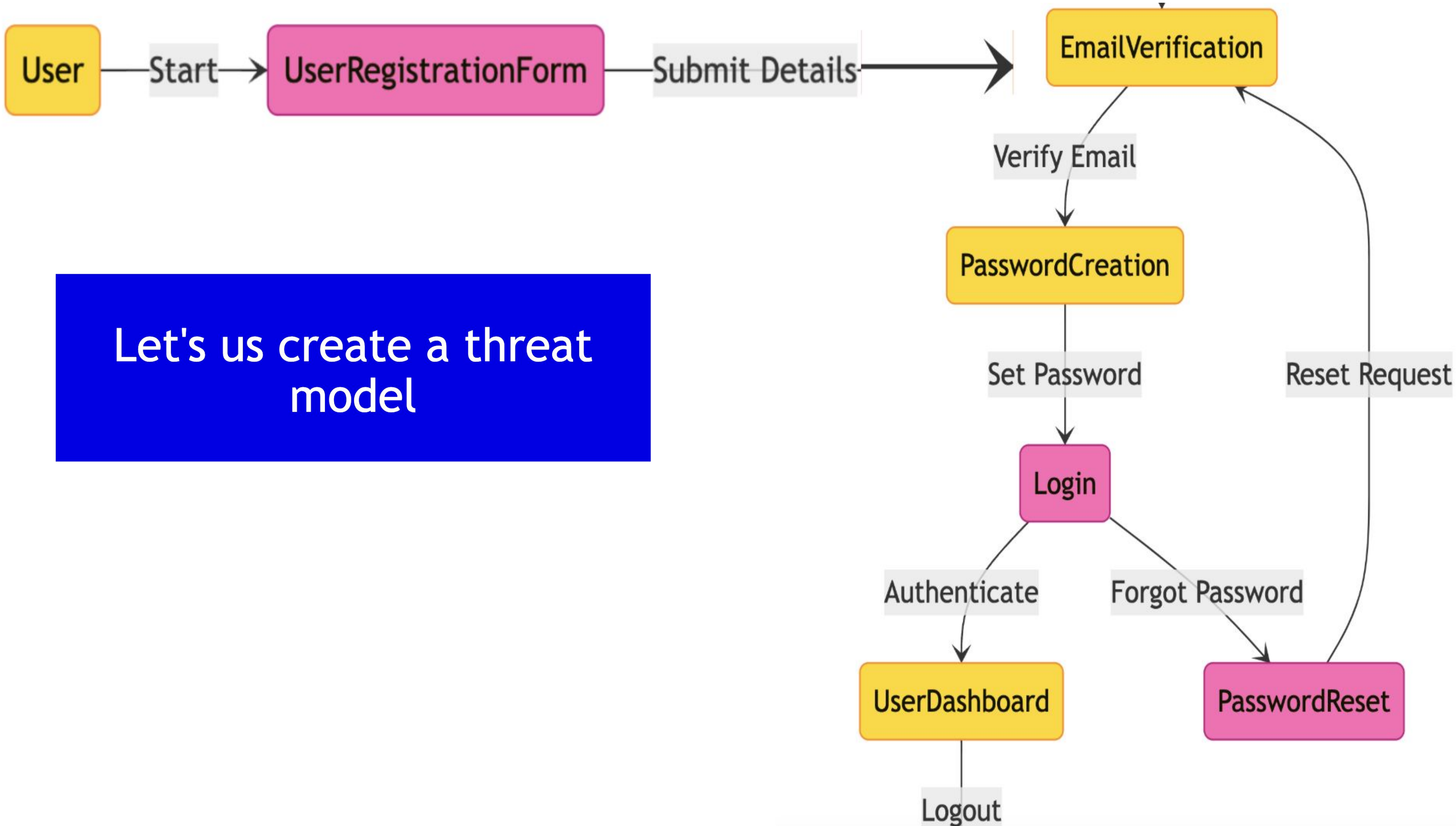
- Define countermeasures and controls
- Implement security requirements
- Validate effectiveness

Step 7: Validate and Review

- Review findings with stakeholders

Threat modeling - User Registration Module

Let's us create a threat model



Threat Modeling - User registration

Step 1: Define the Scope

We are now analyzing the security threats for a **user registration flow**

1. User Registration → Email Verification → Password Creation
2. Login → Dashboard → Logout
3. Forgot Password → Email Verification → Password Reset

Threat Modeling - User registration

Step 2: Identify Assets

1. **User Credentials**
   - Username/email addresses
   - Passwords (hashed)
   - Security questions/answers
   - Multi-factor authentication secrets
2. **Personal Information**
   - Names
   - Email addresses
   - Profile information
   - Activity logs

3. Authentication Tokens

- Session tokens
- Email verification tokens
- Password reset tokens
- Remember-me tokens

4. System Components

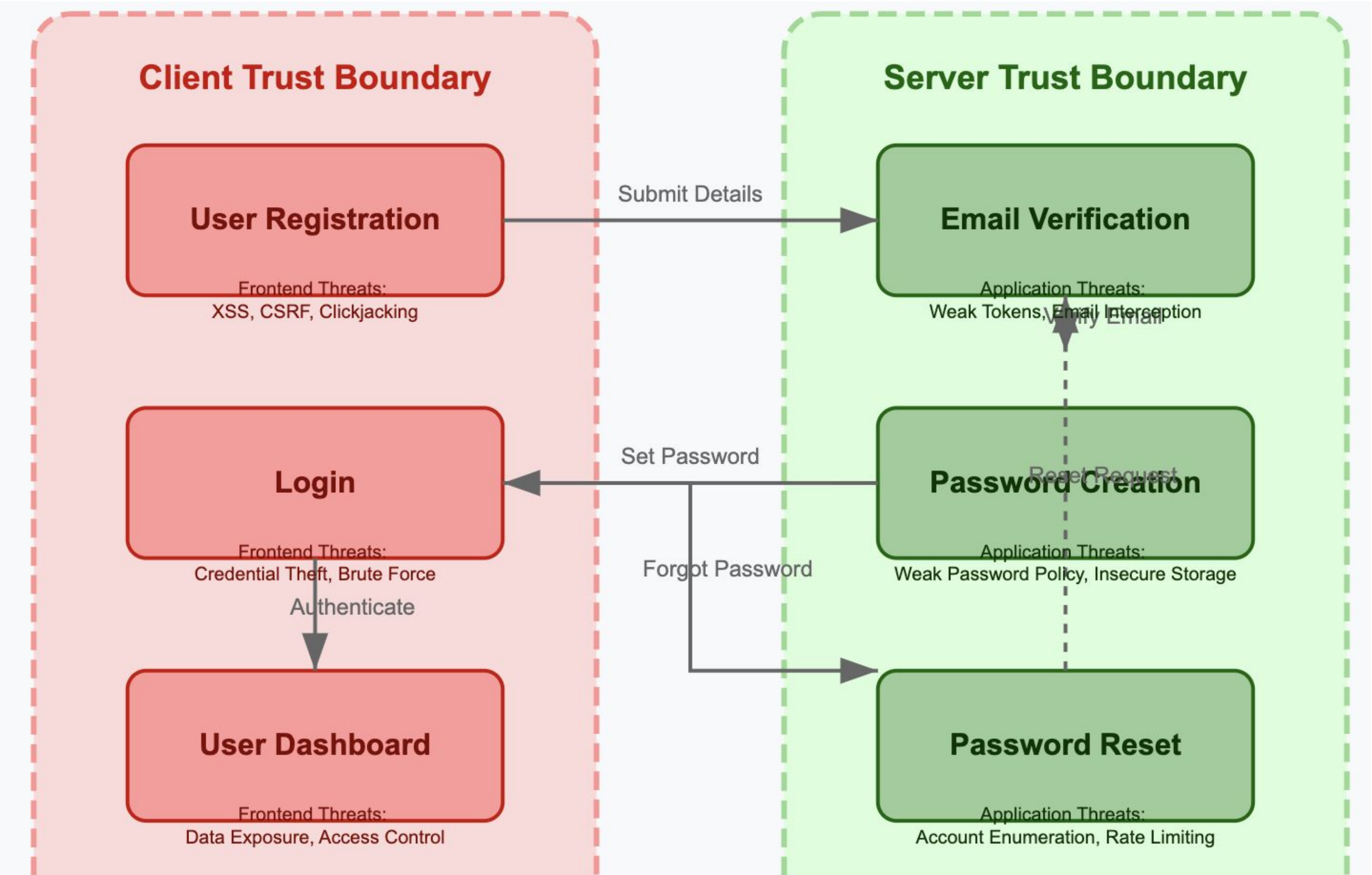
- Authentication servers
- User database
- Email verification service
- Session management system

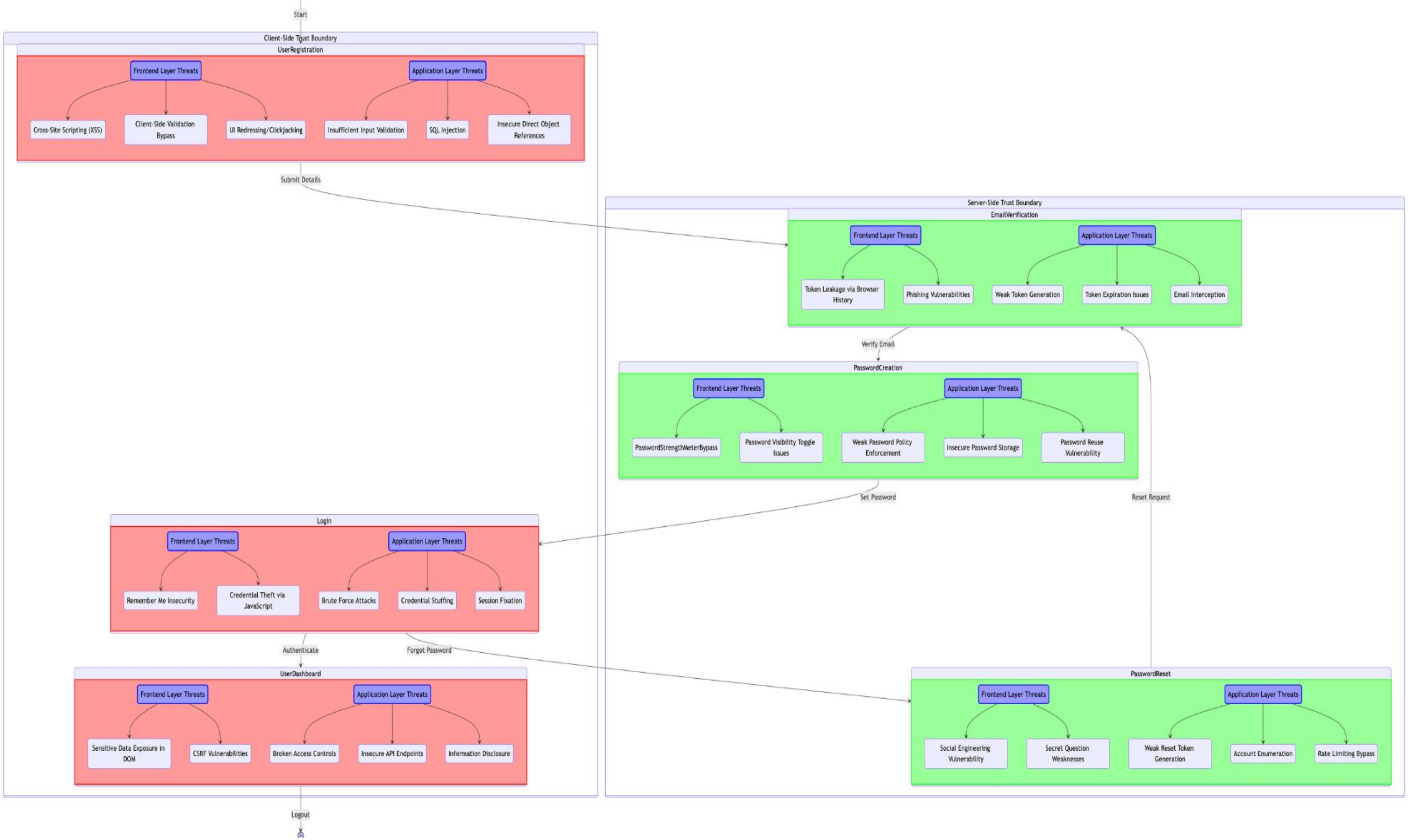
Threat Modeling - User registration

Step 3: Identify Trust Boundaries

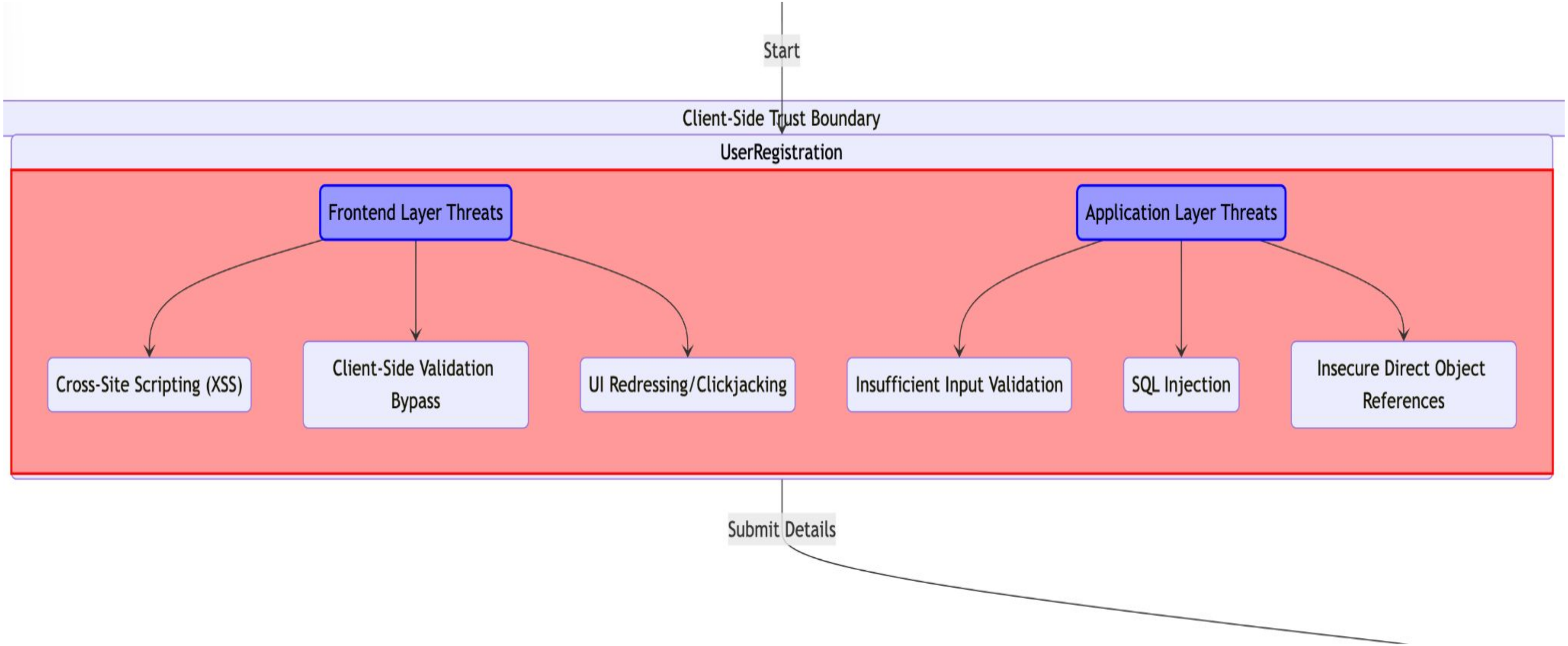
- User-to-Application Boundary
- Password Management Boundary
- Authentication Service Boundary
- Email System Boundary
- Session Boundary
- Data Access Boundary

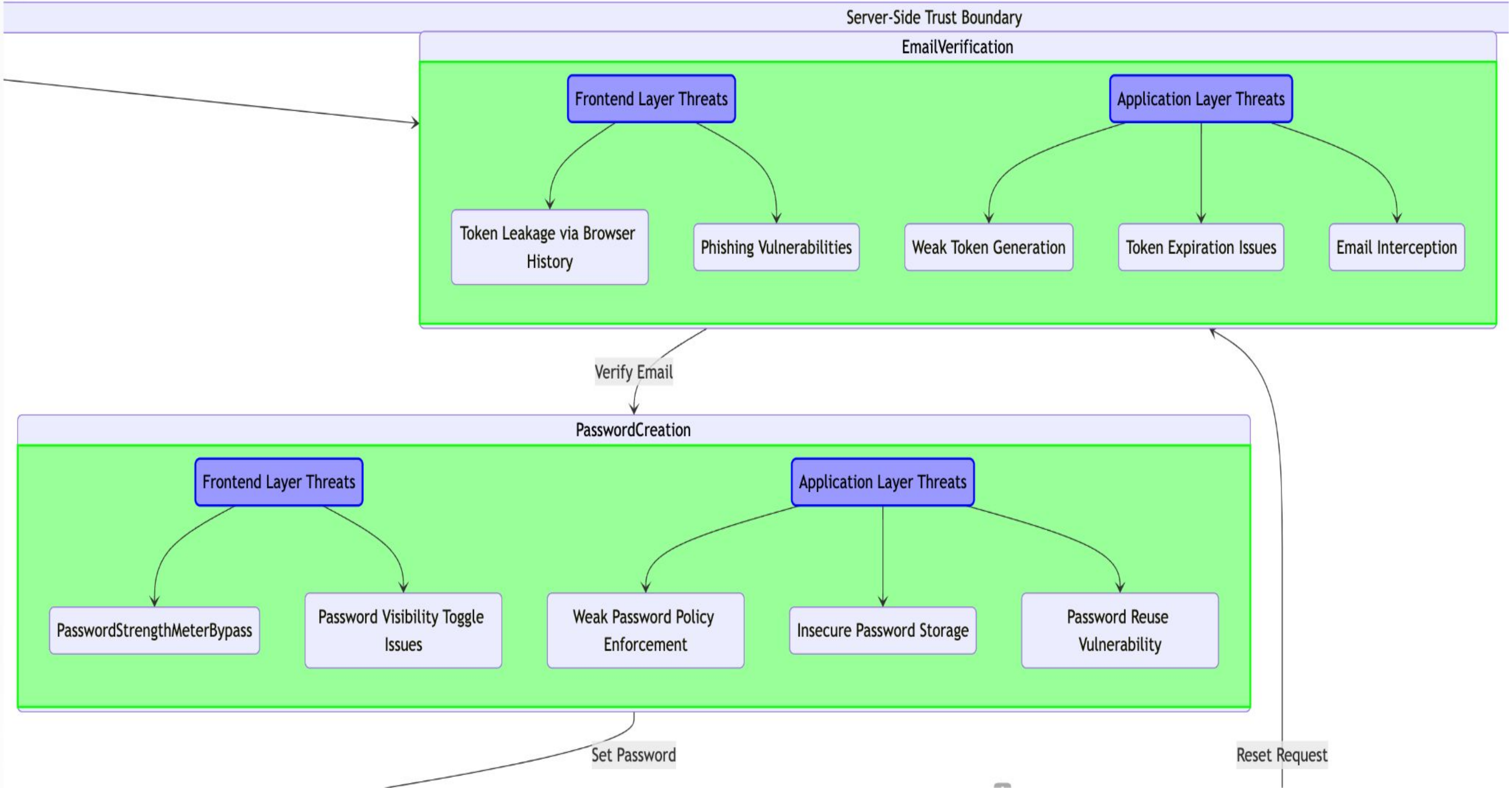
Threat Modeling - User registration

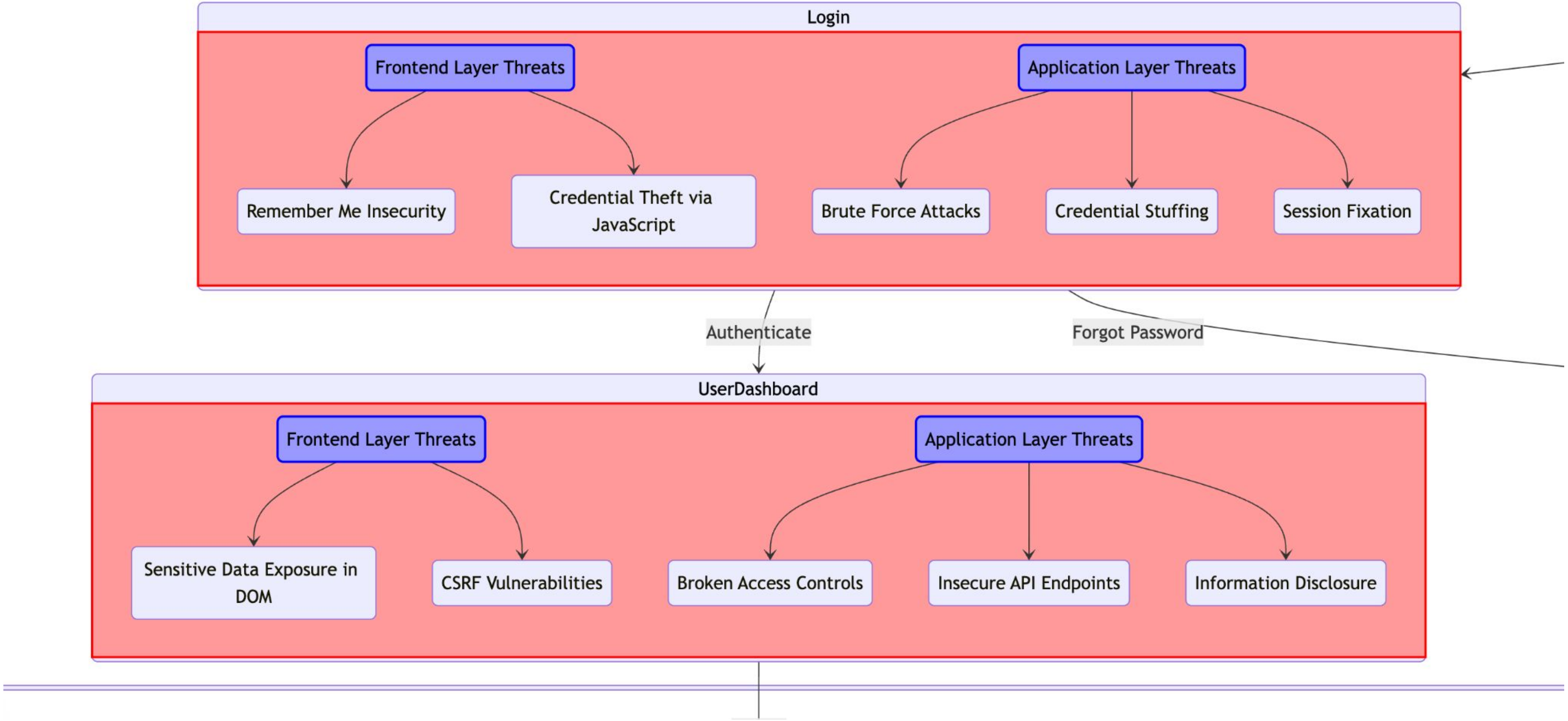


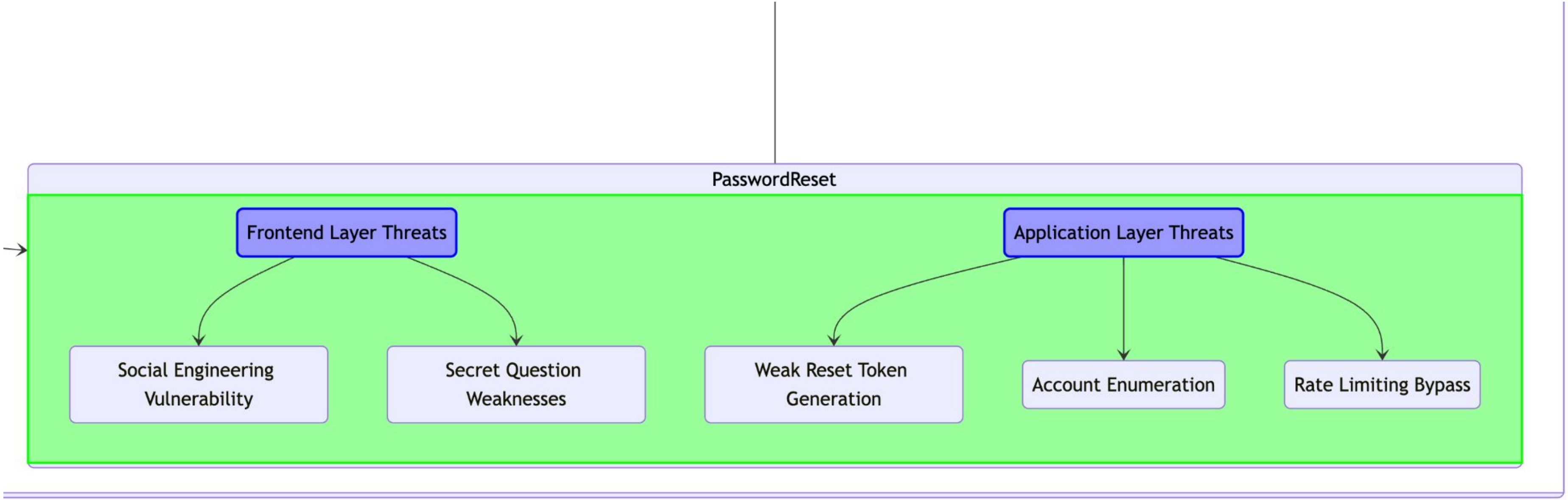

Start
Client-Side Trust Boundary
UserRegistration
Frontend Layer Threats
Application Layer Threats
Cross-Site Scripting (XSS)
Client-Side Validation Bypass
UI Redressing/Clickjacking
Insufficient Input Validation
SQL Injection
Insecure Direct Object References
Submit Details
Server-Side Trust Boundary
EmailVerification
Frontend Layer Threats
Application Layer Threats
Token Leakage via Browser History
Phishing Vulnerabilities
Weak Token Generation
Token Expiration Issues
Email Interception
Verify Email
PasswordCreation
Frontend Layer Threats
Application Layer Threats
PasswordStrengthMeterBypass
Password Visibility Toggle Issues
Weak Password Policy Enforcement
Insecure Password Storage
Password Reuse Vulnerability
Set Password
Reset Request
Login
Frontend Layer Threats
Application Layer Threats
Remember Me Insecurity
Credential Theft via JavaScript
Brute Force Attacks
Credential Stuffing
Session Fixation
Authenticate
Forgot Password
UserDashboard
Frontend Layer Threats
Application Layer Threats
Sensitive Data Exposure in DOM
CSRF Vulnerabilities
Broken Access Controls
Insecure API Endpoints
Information Disclosure
PasswordReset
Frontend Layer Threats
Application Layer Threats
Social Engineering Vulnerability
Secret Question Weaknesses
Weak Reset Token Generation
Account Enumeration
Rate Limiting Bypass
Logout

Identify the trust boundaries


Start
Client-Side Trust Boundary
UserRegistration
Frontend Layer Threats
Application Layer Threats
Cross-Site Scripting (XSS)
Client-Side Validation Bypass
UI Redressing/Clickjacking
Insufficient Input Validation
SQL Injection
Insecure Direct Object References
Submit Details


Server-Side Trust Boundary
EmailVerification
Frontend Layer Threats
Application Layer Threats
Token Leakage via Browser History
Phishing Vulnerabilities
Weak Token Generation
Token Expiration Issues
Email Interception
Verify Email
PasswordCreation
Frontend Layer Threats
Application Layer Threats
PasswordStrengthMeterBypass
Password Visibility Toggle Issues
Weak Password Policy Enforcement
Insecure Password Storage
Password Reuse Vulnerability
Set Password
Reset Request


Login
Frontend Layer Threats
Application Layer Threats
Remember Me Insecurity
Credential Theft via JavaScript
Brute Force Attacks
Credential Stuffing
Session Fixation
Authenticate
Forgot Password
UserDashboard
Frontend Layer Threats
Application Layer Threats
Sensitive Data Exposure in DOM
CSRF Vulnerabilities
Broken Access Controls
Insecure API Endpoints
Information Disclosure


PasswordReset
Frontend Layer Threats
Application Layer Threats
Social Engineering Vulnerability
Secret Question Weaknesses
Weak Reset Token Generation
Account Enumeration
Rate Limiting Bypass

Threat Modeling - User registration

Step 4: Applying Threat Framework (STRIDE) - Component wise breakdown

User Registration Component

Threat Category	Description	Examples
Spoofing	Attackers may register with fake identities or impersonate others	• Creating accounts using someone else's personal information • Registering with fraudulent credentials
Tampering	Manipulation of registration data during transmission	• Modifying registration requests via man-in-the-middle attacks • Tampering with HTML forms
Repudiation	Users denying they created an account	• Claiming unauthorized account creation • Denying registration actions

Information Disclosure	Exposure of personal information during registration	• Leaking submitted data via error messages • Insecure storage of registration details
Denial of Service	Overwhelming registration system with automated requests	• Bot registrations • Form spamming
Elevation of Privilege	Gaining administrative access during registration	• Parameter manipulation to set admin privileges • Exploiting registration logic flaws

Email Verification Component

Threat Category	Description	Examples
Spoofing	Fake verification emails or falsifying verified status	• Phishing emails mimicking verification messages • Forged verification tokens
Tampering	Modifying verification tokens or URLs	• Altering verification links to bypass email confirmation • Tampering with verification cookies
Repudiation	Denying receipt of verification emails	• Claiming verification emails were never received • Disputing verification timestamps

Information Disclosure	Email addresses or verification status leaked	• Verification URLs exposing user information • Error messages revealing verification status	• One-time use tokens • Minimal information in verification links • Proper error handling
Denial of Service	Flooding verification systems with requests	• Mass verification email requests • Token validation overload	• Email sending quotas • Rate limiting verification attempts • Token expiration
Elevation of Privilege	Bypassing verification to gain verified status	• Token prediction attacks • Exploiting verification logic flaws	• Strong token generation • Proper verification workflow enforcement

Password Creation Component

Threat Category	Description	Examples
Spoofing	Setting passwords for other users' accounts	• Session hijacking during password creation • Cross-site request forgery
Tampering	Manipulating password requirements or creation requests	• Bypassing password strength requirements • Altering password hash
Repudiation	Denying password creation or changes	• Claiming unauthorized password setting • Disputing password change history

Information Disclosure	Exposure of password policies or storage mechanisms	• Error messages revealing password requirements • Leaking password hashing methods	• Generic error messages • Secure storage of password creation metadata
Denial of Service	Excessive resource consumption during password processing	• Submitting extremely long passwords • Triggering expensive password hashing operations	• Password length limits • Optimized password processing • Resource throttling
Elevation of Privilege	Manipulating password creation to gain unauthorized access	• Exploiting password reset flows • Injecting malicious code via password fields	• Input sanitization • Separation of password creation from privilege management

Step 5: Analyze and Prioritize Threats

Step 6: Mitigate Risks

Step 7: Validate and Review

Threat Modeling - Four Fundamental Questions

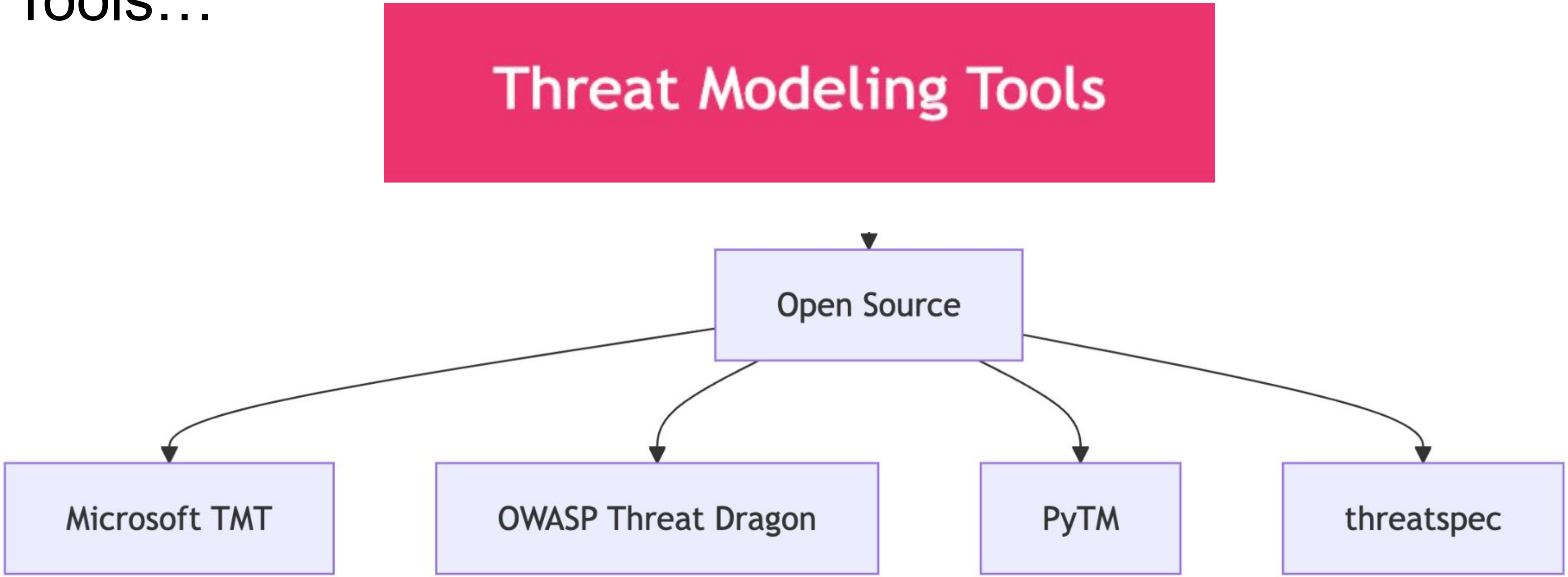
1. **What Are We Working On?**

2. **What Can Go Wrong?**

3. **What Are We Going to Do About It?**

4. **Did We Do a Good Job?**

Tools…



Threat Modeling Tools

Open Source

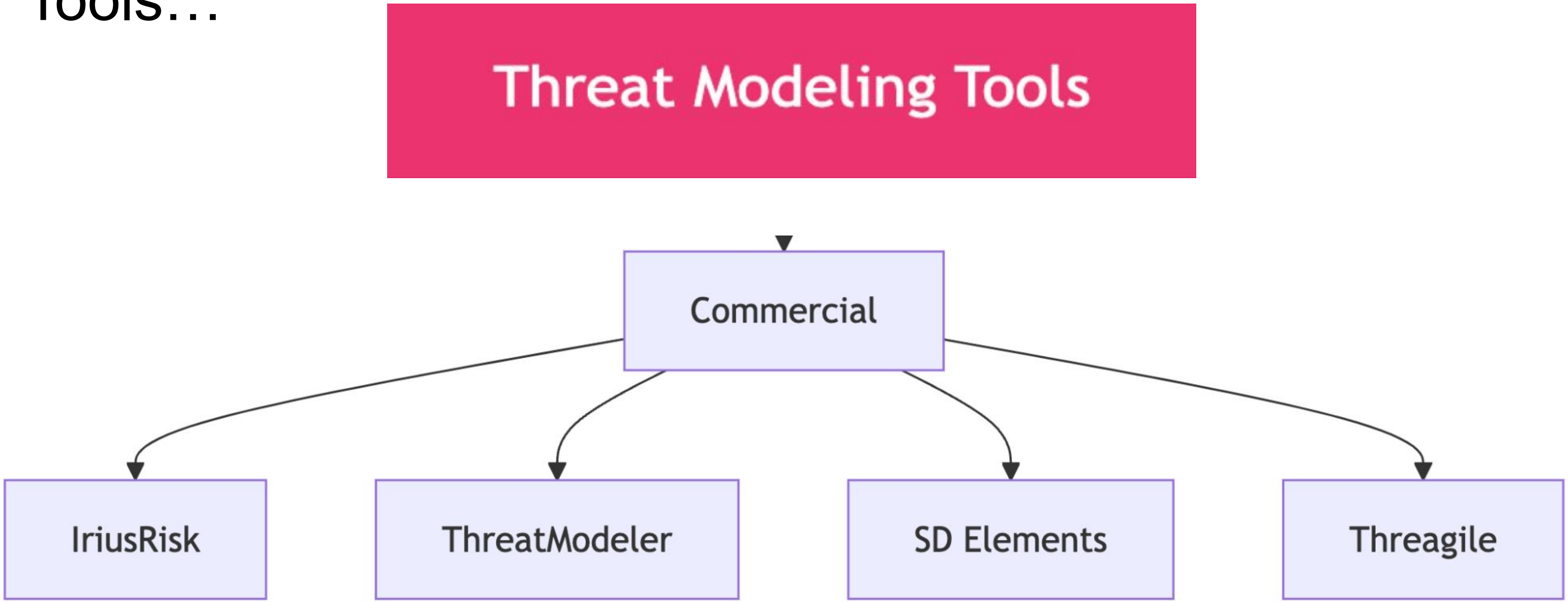
Microsoft TMT

OWASP Threat Dragon

PyTM

threatspec

Tools…



Threat Modeling Tools

Commercial

IriusRisk

ThreatModeler

SD Elements

Threagile

Resources

- OWASP Application Security Verification Standard (ASVS)
- MITRE ATTACK Framework
- NIST Cybersecurity Framework
- CWE/SANS Top 25 Most Dangerous Software Errors
- OWASP Top 10 Web Application Security Risks

https://quizizz.com/pro/join?gc=944825

Any

Questi n

THANK YOU!