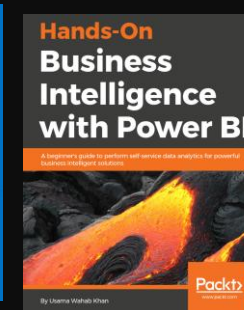
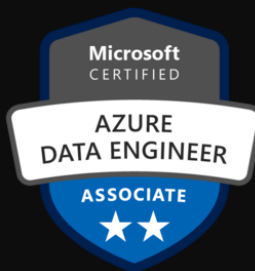




Introduction to Threat Modeling & Methodologies

Understanding frameworks for software
security





Usama Wahab Khan


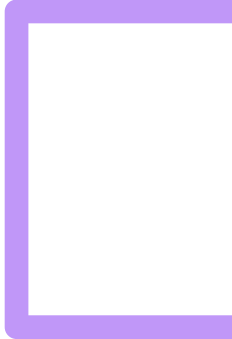
- Father, data Scientist, Developer/Nerd, Traveler

Twitter : @usamawahabkhan
LinkedIn : Usamawahabkhan





Workshop Agenda

- Welcome and Introduction
 - Overview of Threat Modeling
 - Introduction to Threat Modeling Frameworks
 - Case Study: Network Management and Operations
 - Day two : Hands-On Lab
 - Hands-On Exercise: Basic Threat Model for a Sample Application
 - Group Discussion and Q&A
- 
- 



Welcome and Introduction



Overview of Workshop Objectives

Understanding Threat Modeling

Participants will gain a solid understanding of what threat modeling is and its importance in cybersecurity.

Methodologies in Threat Modeling

The workshop will cover various methodologies used in threat modeling, helping participants choose the right approach for their needs.

Real-World Applications

Participants will learn how to apply threat modeling in real-world scenarios, effectively identifying and analyzing potential security threats.



Introduction to threat modeling

Engineering teams can enhance an application's security by actively identifying potential risks and creating mitigation strategies. This involves developing threat models through the evaluation and verification of the software architecture prior to code deployment.

Importance of Threat Modeling in Software Development

Early Threat Identification

Threat modeling allows developers to identify potential security threats at the beginning of the development process, ensuring proactive mitigation.

Minimizing Risks

By addressing security threats early, threat modeling minimizes risks associated with software vulnerabilities and potential breaches.

Improving Security Posture

Implementing threat modeling enhances the overall security posture of software applications, leading to safer and more reliable products.



[This Photo](#) by Unknown Author is licensed under [CC BY-SA-NC](#)

Threat modeling manifesto

we ask four key 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 enough job?

Threat modeling manifesto

- Principles

- We follow these principles:
- The best use of threat modeling is to improve the security and privacy of a system through early and frequent analysis.
- Threat modeling must align with an organization's development practices and follow design changes in iterations that are each scoped to manageable portions of the system.
- The outcomes of threat modeling are meaningful when they are of value to stakeholders.
- Dialog is key to establishing the common understandings that lead to value, while documents record those understandings, and enable measurement.

- Values

- We have come to value:
- A culture of finding and fixing design issues over checkbox compliance.
- People and collaboration over processes, methodologies, and tools.
- A journey of understanding over a security or privacy snapshot.
- Doing threat modeling over talking about it.
- Continuous refinement over a single delivery.



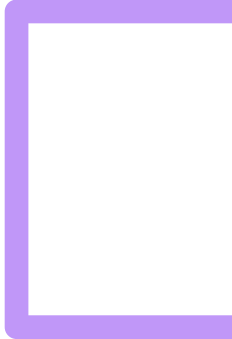

Why Threat Modeling Is Critical in Secure Software Development

Understanding Threats

Threat modeling provides a framework for organizations to identify and understand potential threats specific to their applications.

Prioritizing Security Efforts

By identifying risks, threat modeling enables teams to prioritize their security efforts more effectively, focusing on the most significant vulnerabilities.



THE PROCESS OF THREAT MODELING

Define the Scope

- Identify the system or process for analysis.
- Determine which assets require protection.
- Establish trust boundaries and security needs.

Identify Threats

- Use frameworks like STRIDE or OCTAVE to categorize threats.
- Consider possible attack vectors and adversaries.
- Analyze potential entry points for exploitation.

Create a Model

- Develop Data Flow Diagrams (DFDs) to visualize data movement.
- Identify security controls and vulnerabilities.
- Map threats to system components.

Assess Risks

- Prioritize threats by impact and likelihood.
- Use techniques to evaluate threat severity.
- Focus on the most critical vulnerabilities.

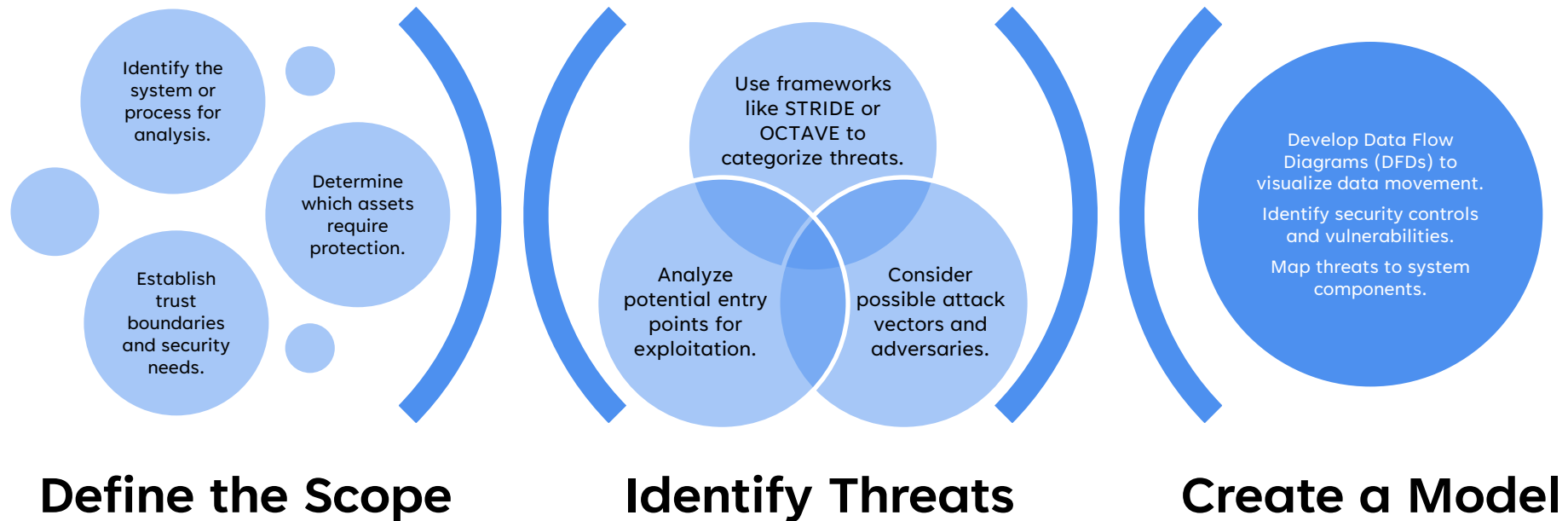
Implement Mitigation Strategies

- Develop security controls for identified threats.
- Apply encryption and access control measures.
- Utilize tools like Microsoft Threat Modeling Tool for efficiency.

Validate & Iterate

- Test the effectiveness of security measures.
- Update the threat model as the system changes.
- Conduct regular security reviews to address new threats.

THE PROCESS OF THREAT MODELING



Decomposing a Large Application

Identify Core Components

Divide the application into **major sections**:

- **Frontend** (User Interface, Browser Interactions)
- **Backend** (APIs, Business Logic, Database Access)
- **Data Storage** (Databases, File Storage, Cloud Services)
- **Network Layer** (Servers, Firewalls, Load Balancers)
- **Example:** An online banking system contains:
 - ◆ **Login Page** (Frontend) → **Authentication Service** (Backend) → **Database** (User Accounts)

Map Data Flow Across the System

Use Data Flow Diagrams (DFDs) to show how data moves between components.

- Example: User submits login → Authentication service checks credentials → Database responds with user info → Session starts
- Helps identify entry points and attack surfaces.

Identify Assets & Trust Boundaries

- **Assets:** Sensitive data (e.g., customer financial details, login credentials).
 - ◆ **Trust Boundaries:** Areas where data moves between trusted and untrusted zones (e.g., from user devices to servers).

Identify Assets & Trust Boundaries

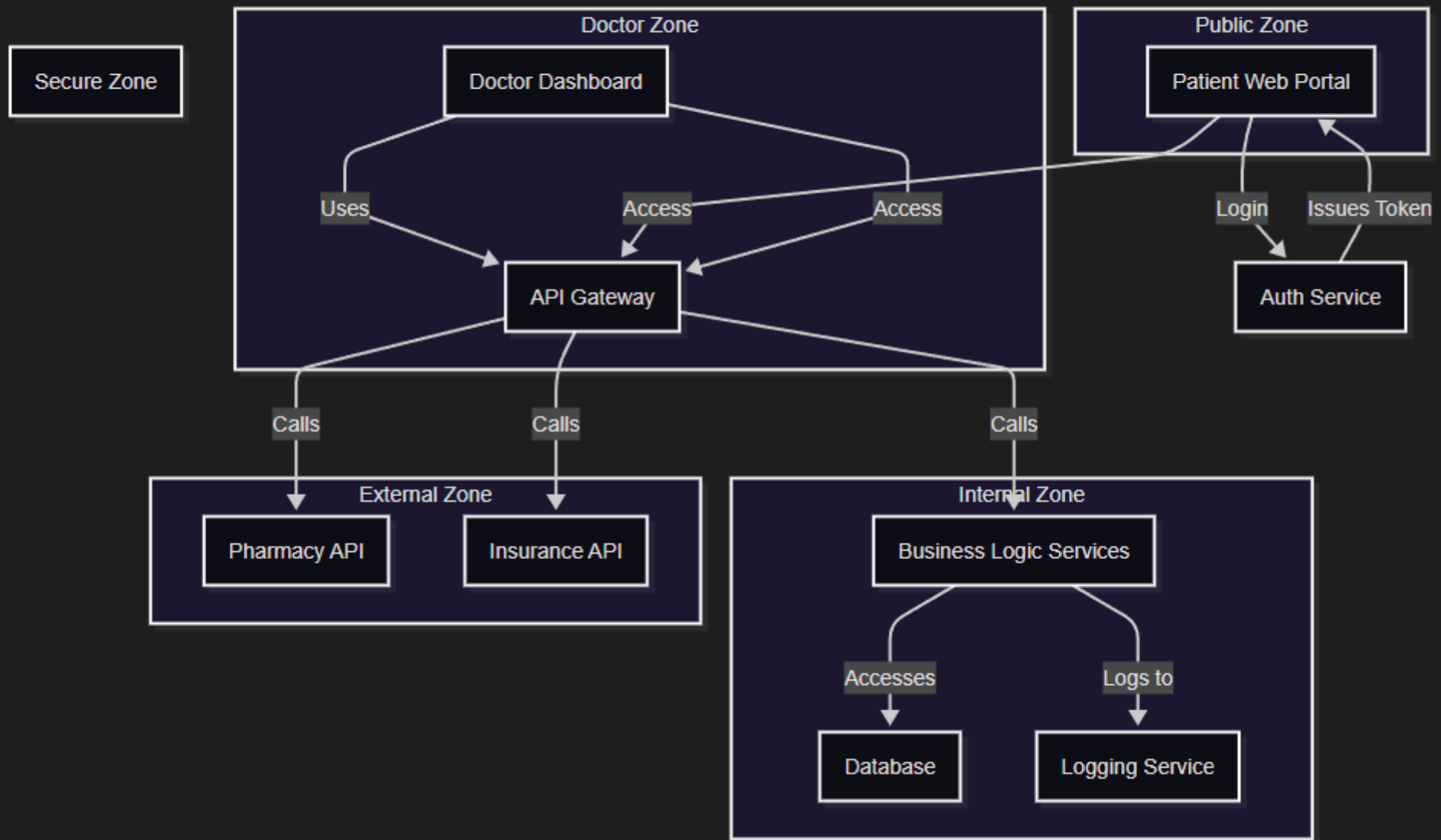
What Are Trust Boundaries?

A **trust boundary** is a point in a system where the level of trust changes.
This could be:

- Between a user and a web application
- Between a web server and a database
- Between internal and external networks
- Between different microservices with varying levels of access

Examples of Trust Boundaries

| Trust Boundary | Description | Example Threats |
|-----------------------------|-------------------------------|-------------------------------------|
| User ↔ Web App | User input enters the system | XSS, SQL Injection |
| Web App ↔ DB | App sends queries to DB | SQL Injection, privilege escalation |
| Internal ↔ External Network | Data crosses firewall | DDoS, spoofing |
| Microservice A ↔ B | Services with different roles | Unauthorized access, data leakage |



| From Zone | To Zone | Trust Boundary Reason | Key Controls |
|-----------|--------------|---|--------------------------------------|
| Public | Internal | Untrusted user accessing internal services | Auth, validation, HTTPS |
| Doctor | Internal | Semi-trusted user accessing internal services | RBAC, logging, token validation |
| Internal | Secure | Accessing sensitive data/logs | Encryption, access control |
| Internal | External | Calling third-party APIs | API auth, data filtering, monitoring |
| Public | Auth Service | User authentication | MFA, brute-force protection |

Break Down Threats per Component Using STRIDE

Frontend → Spoofing, cross-site scripting (XSS).

Backend → API injections, unauthorized data access.

Database → SQL injection, privilege escalation.

Network → Man-in-the-middle (MITM) attacks, denial-of-service (DoS).

Example: Attackers try to inject malicious code in an API endpoint to access financial data.



Apply Security Controls

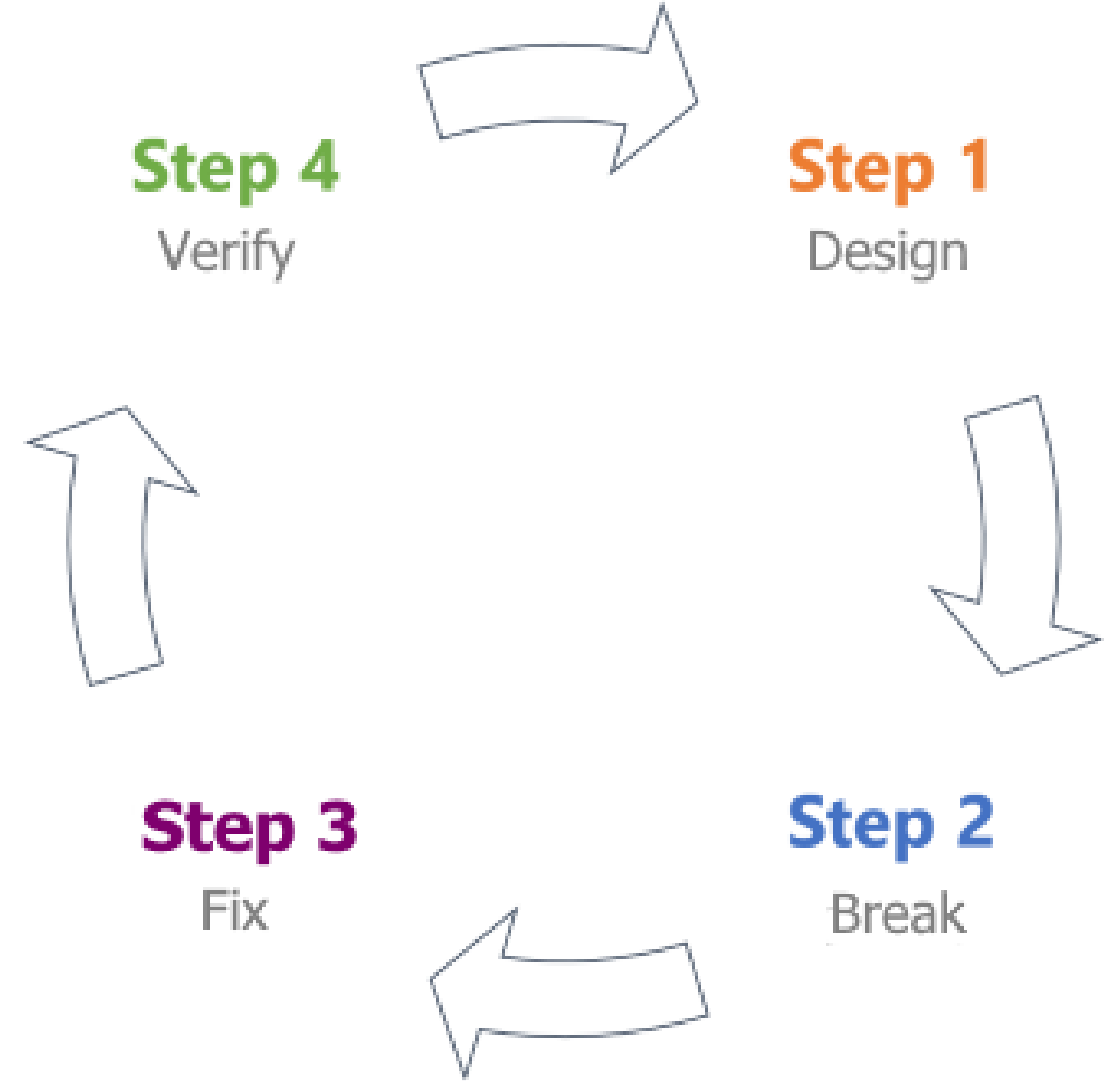
- Encrypt sensitive data for protection.
- Use multi-factor authentication (MFA) for user security.
- Implement firewalls, IDS, and secure API gateways for network security.

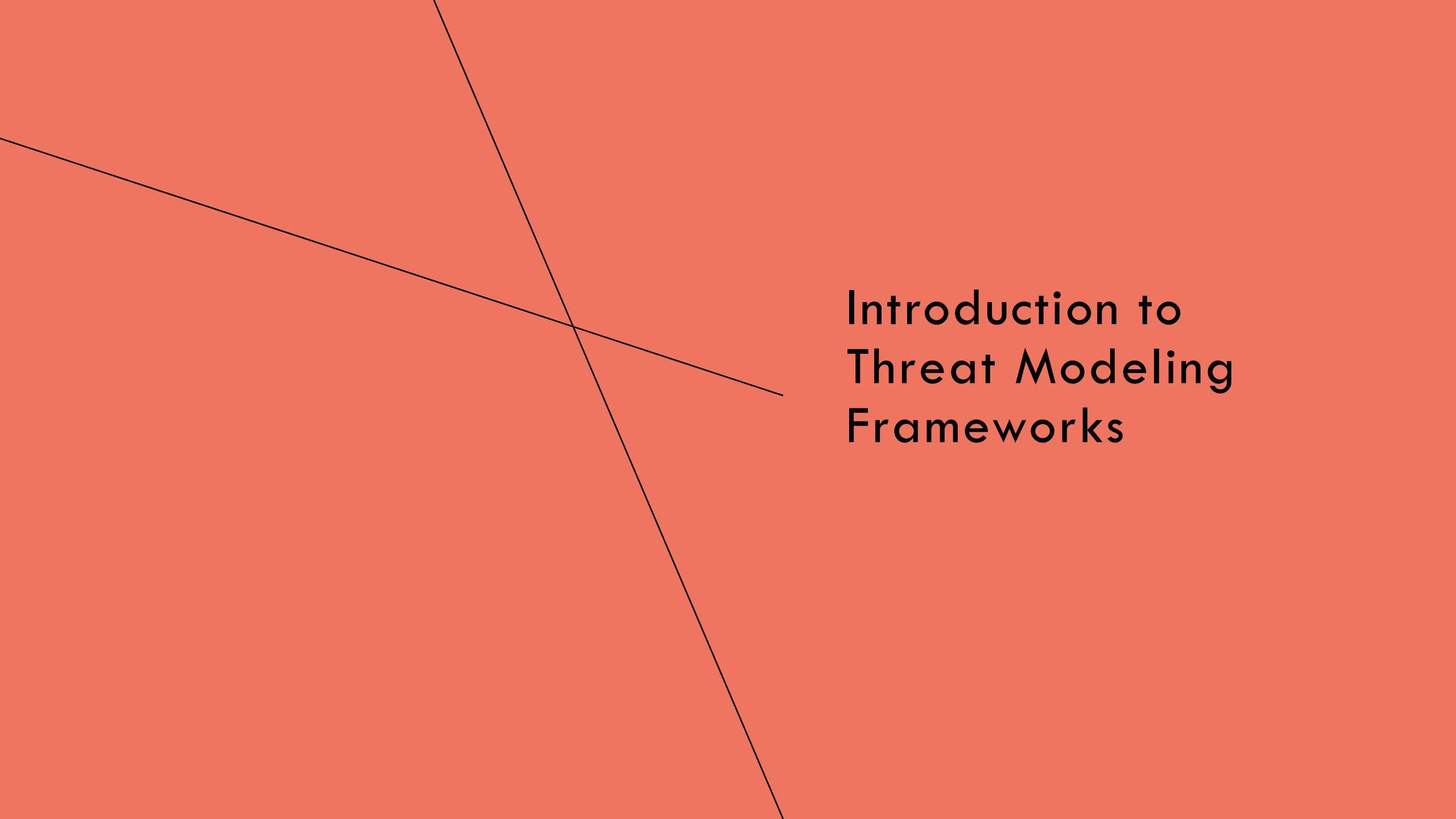
Outcome:

By decomposing a large application, organizations gain visibility into security risks, attack surfaces, and ways to mitigate vulnerabilities.

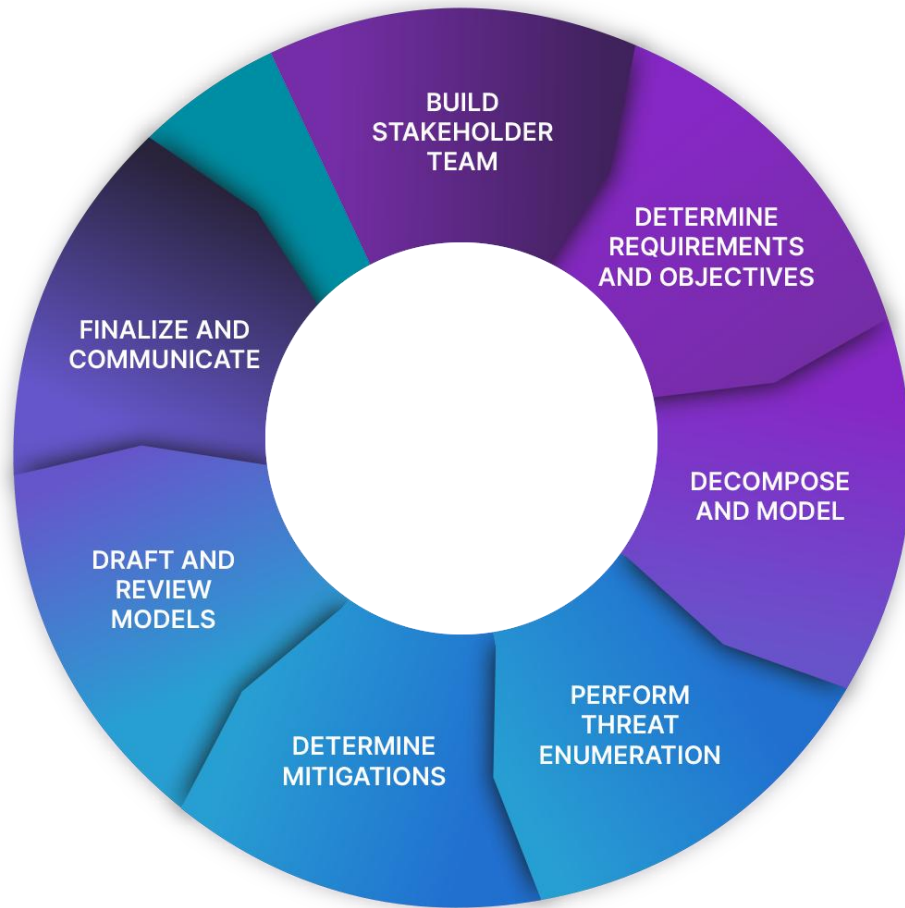
THREAT MODELING PHASES

- 1 Design Capture all requirements for your system and create a data-flow diagram.
- 2 Break Apply a threat-modeling framework to the data-flow diagram and find potential security issues.
- 3 Fix Decide how to approach each issue with the right combination of security controls.
- 4 Verify Verify requirements are met, issues are found, and security controls are implemented.



The background is a solid salmon color. Two thin black lines intersect diagonally. One line runs from the top-left towards the bottom-right, and the other runs from the top-right towards the bottom-left. They cross each other in the upper-left quadrant of the image.

Introduction to Threat Modeling Frameworks



Threat Models Frameworks

Threat modeling frameworks offer organized approaches to identify, evaluate, and reduce security risks in software and systems. Notable examples include STRIDE (Microsoft), PASTA, OCTAVE, Trike, and VAST, each utilizing various techniques like risk analysis and attack simulation to enhance security and address vulnerabilities.

Factors in Choosing a Threat Model



Industry-Specific Risks

Identify risks unique to the industry the organization operates in to tailor the threat model accordingly.

Security Team Size

Consider the size and capabilities of the security team when choosing a threat model.

Organizational Structure

Evaluate the organizational structure and involved stakeholders to ensure the threat model aligns with the organization's needs.

Resource Availability

Assess the available resources, including technology and budget, to implement the chosen threat model effectively.

Threat Model and Factors Table

Here's a structured table outlining key threat modeling frameworks and the factors influencing threat modeling decisions:

| Threat Model | Description | Best Use Case |
|--------------------|---|--|
| STRIDE (Microsoft) | Categorizes threats into Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. | Software & system security |
| PASTA | A seven-stage risk-based framework focusing on attacker simulation. | Enterprise security & risk analysis |
| OCTAVE | A risk-driven approach for organizing asset security evaluations. | Organizational threat assessment |
| MITRE ATT&CK | Maps real-world attack techniques to adversary tactics for threat hunting. | Cybersecurity operations & threat intelligence |
| VAST | A scalable approach designed for agile environments using data flow diagrams. | DevOps & cloud-based security |

Factors Influencing Threat Modeling

| Factor | Influence on Threat Modeling |
|--------------------------|---|
| Industry-Specific Risks | Determines unique attack vectors (e.g., financial fraud in banking, data breaches in healthcare). |
| Security Team Size | Defines the scope & complexity of the threat model based on available personnel. |
| Organizational Structure | Ensures alignment with stakeholder priorities & security policies. |
| Resource Availability | Impacts the use of automated tools, security controls, and mitigation strategies. |

Practical Considerations in Threat Modeling

Employees and Internal Teams

Assessing the security practices and behavior of employees and internal teams is crucial in threat modeling.

Devices and System Architecture

Evaluating the security of devices and the overall system architecture helps identify potential vulnerabilities.

Code Deployment Process

Analyzing the code deployment process ensures that potential threats are mitigated before new code goes live.

Third-Party Integrations

Reviewing third-party integrations is essential to ensure that external components do not introduce new risks.

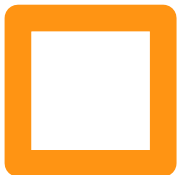





MITRE ATT&CK framework

Key Components of MITRE ATT&CK

- ✓ Tactics – The goals attackers aim to achieve (e.g., Initial Access, Privilege Escalation).
- ✓ Techniques – The methods used to accomplish tactics (e.g., Phishing, Credential Dumping).
- ✓ Procedures – The specific implementations of techniques used by threat actors.
- ✓ Threat Groups – Profiles of known cyber adversaries and their attack patterns

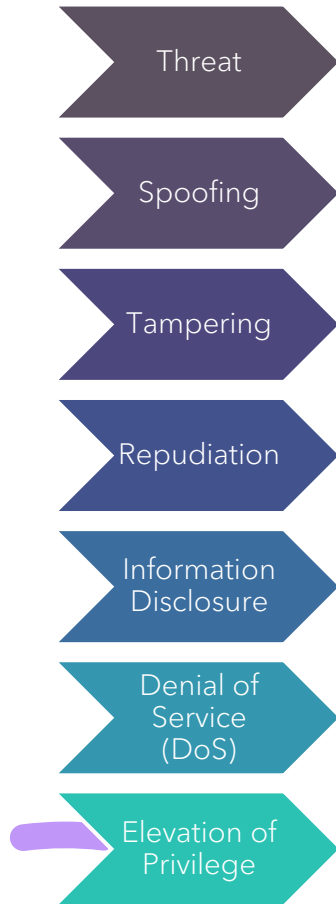
The MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) framework is a globally accessible knowledge base that categorizes cyber threats, attack techniques, and adversary behaviors based on real-world observations.





| Tactic | Example Technique | Description |
|-------------------|---------------------------------|---|
| Initial Access | Phishing (T1566) | Trick users into clicking malicious links |
| Execution | PowerShell (T1059.001) | Run malicious scripts |
| Persistence | Registry Run Keys (T1547) | Maintain access after reboot |
| Credential Access | Credential Dumping (T1003) | Steal user credentials |
| Exfiltration | Exfiltration Over HTTPS (T1041) | Send stolen data out via HTTPS |

STRIDE (Microsoft)



| Threat | Property | Definition |
|-------------------------|-----------------|--|
| Spoofing | Authentication | Impersonating something or someone else |
| Tampering | Integrity | Modifying data or code |
| Repudiation | Non-repudiation | Claiming to have not performed the action |
| Information Disclosure | Confidentiality | Exposing information to someone not authorized to see it |
| Denial of Service (DoS) | Availability | Deny or degrade service to users |
| Elevation of Privilege | Authorization | Gain capabilities without proper authorization |



Microsoft STRIDE Example

The Case of SecureAuth: A STRIDE-Based Approach

Scenario: SecureAuth Inc., a fictional web application providing authentication services for banking customers, faces security threats due to increasing cyberattacks



Step 1: Identifying the Threats Using STRID

As SecureAuth analyzes its authentication system, it discovers several vulnerabilities:

Spoofing: Hackers attempt to impersonate legitimate users.

Tampering: Data alterations in transit can compromise transactions.

Repudiation: Users deny performing actions, creating compliance issues.

Information Disclosure: Sensitive customer details may be exposed.

Denial of Service (DoS): Attackers attempt to overwhelm login servers.

Elevation of Privilege: Unauthorized users gain admin access.



Step 2: Applying Security Measures

Multi-Factor Authentication (MFA) to prevent spoofing. Cryptographic Hashing to protect against data tampering.

Audit Logs for accountability and dispute resolution.

Data Encryption to prevent leaks in transmission.

Rate Limiting & Firewall Protections for DoS mitigation.

Role-Based Access Controls (RBAC) to prevent unauthorized privilege escalation.

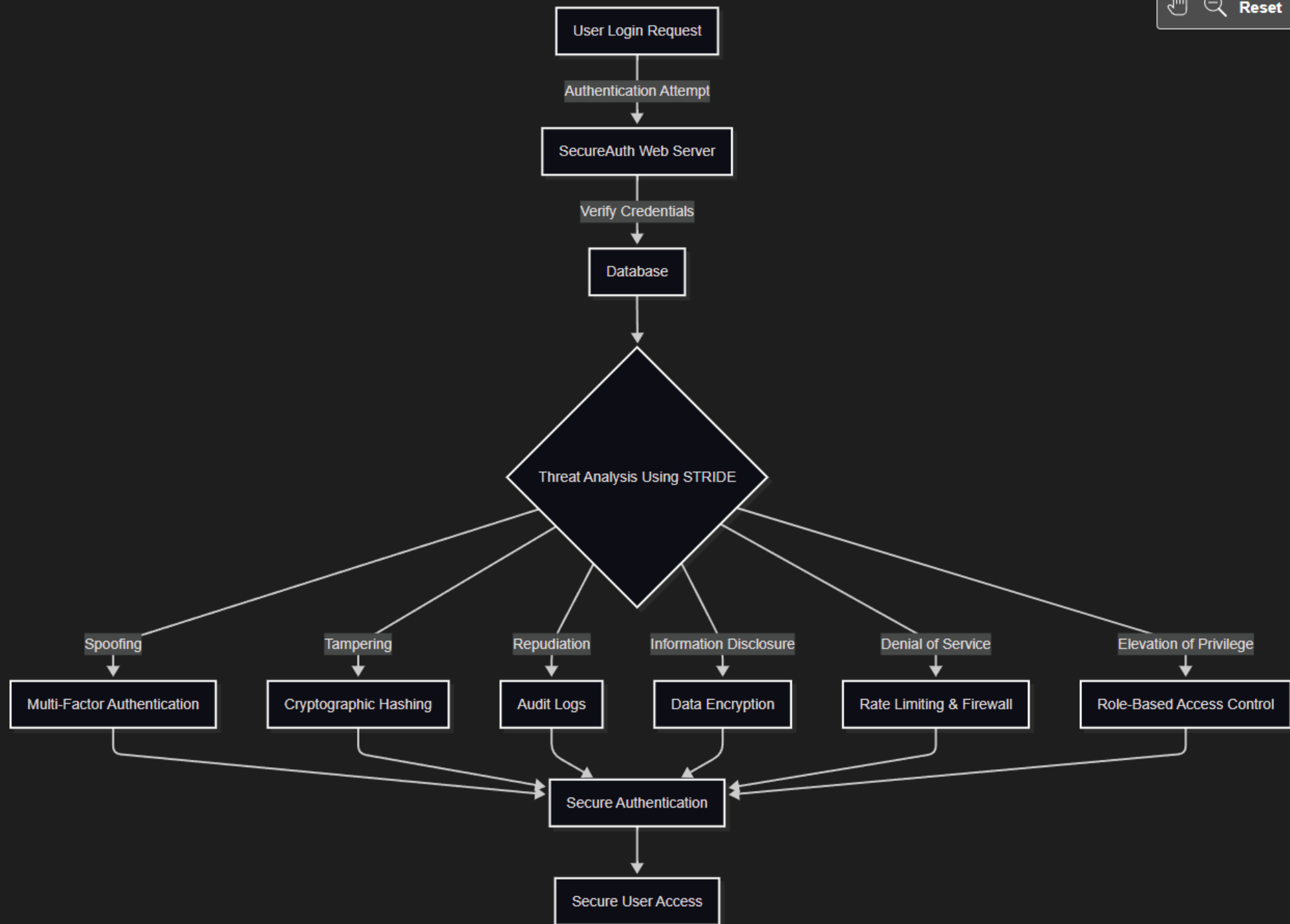


Step 3: Testing & Monitoring

SecureAuth conducts penetration testing to simulate attack attempts and verify security measures. Continuous monitoring with automated tools ensures threats are detected and mitigated before causing damage.

Outcome:


With STRIDE-based security, SecureAuth significantly **reduces cyber risks, protects user data, and enhances trust** among its banking customers.



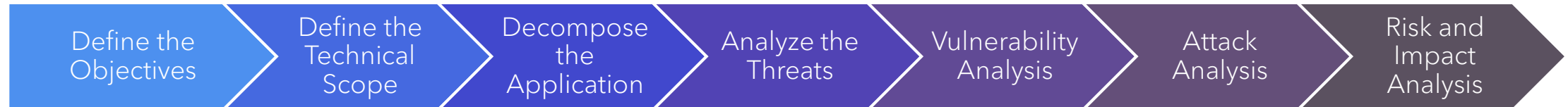
DREAD

- **Damage** – The potential impact of an attack.
- **Reproducibility** – How easily the attack can be repeated.
- **Exploitability** – The effort required to exploit the vulnerability.
- **Affected users** – The number of people or systems impacted.
- **Discoverability** – How easy it is to find the vulnerability.

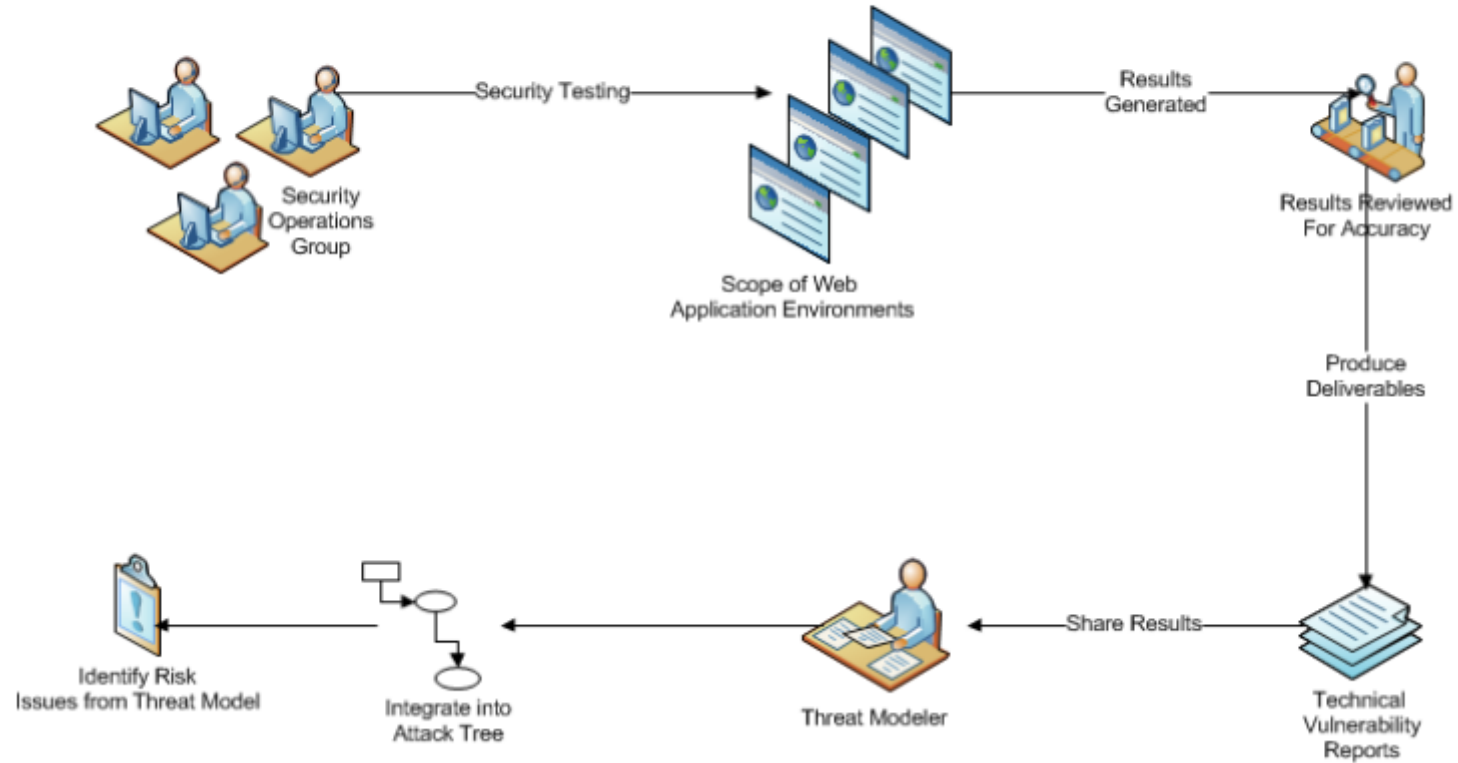
PASTA (Process for Attack Simulation & Threat Analysis)

- Overview
- PASTA stands for Process for Attack Simulation and Threat Analysis (PASTA). It is a risk-centric threat modeling method, meaning that risk plays a central role and the focus is on the highest and most relevant risks that can affect your business. After all, IT (such as applications, systems, etc.) serve business, and that is their reason for existing.
- Risk-centric threat modeling methodology
- Co-founded by Tony UcedaVélez & Marco M. Morana
- Highly scalable for businesses of all sizes
-  Key Features
- Cross-team collaboration between technical teams & decision-makers

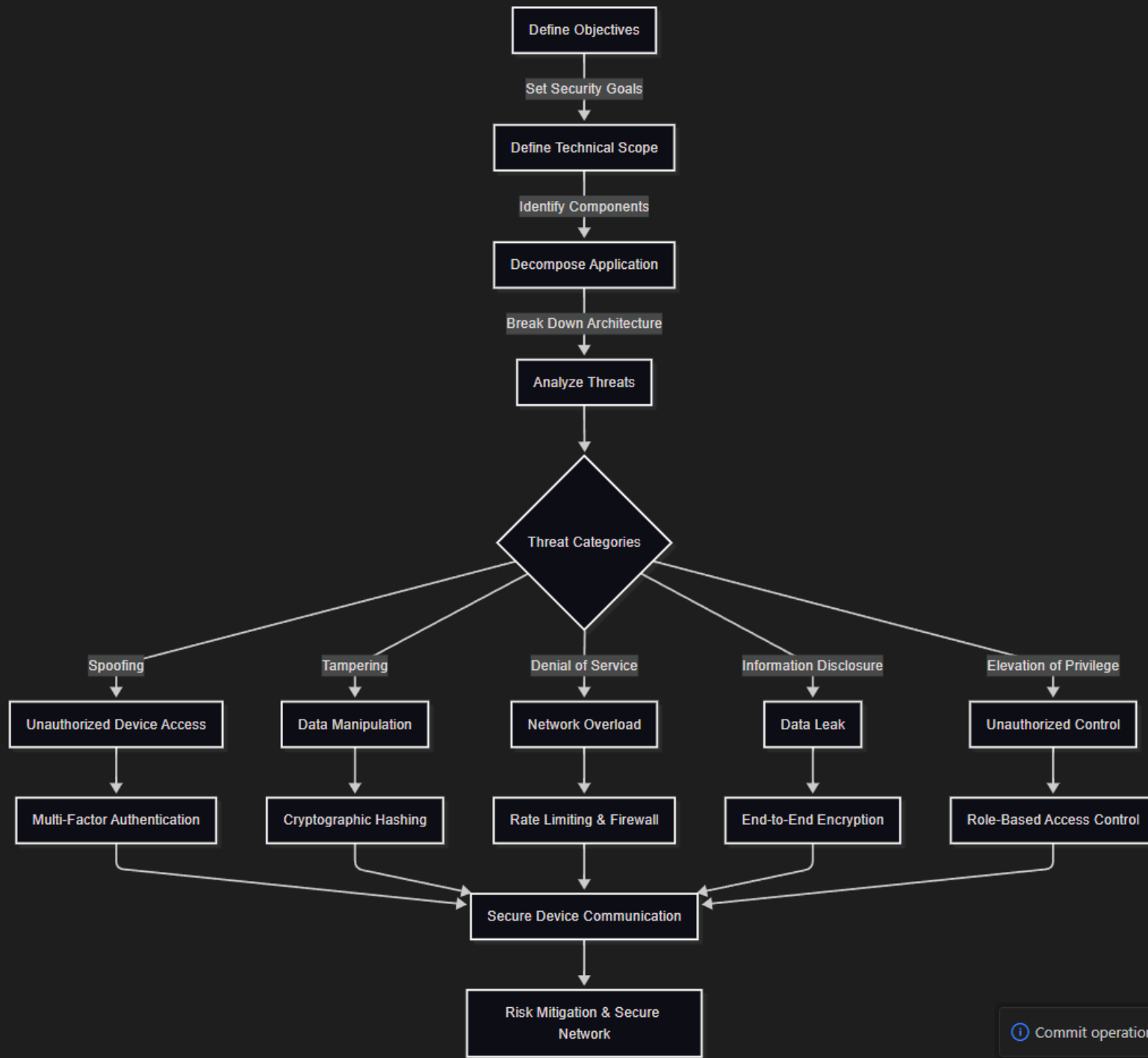
The seven stages of PASTA threat modeling:



PASTA & Collaboration :: Integrative Process







Risk-Based Threat Modeling Frameworks

Trike: Integrates security auditing

- Focuses on acceptable risk levels

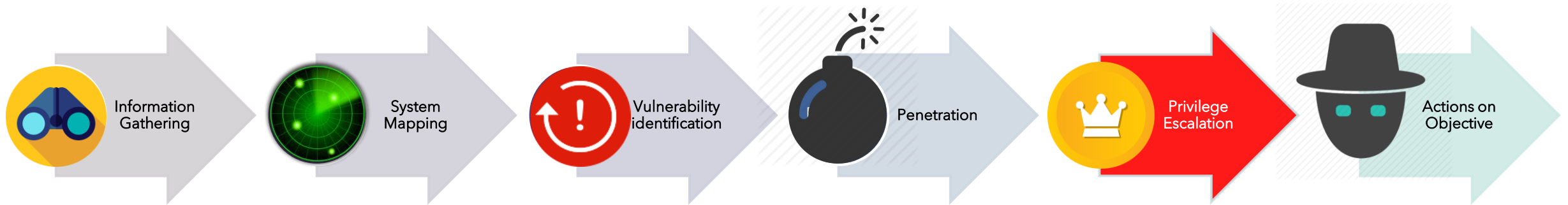
OCTAVE: Developed by Carnegie Mellon University

- Risk-centric framework for assessments
- Evaluates critical assets, vulnerabilities, threats

VAST: Designed for scalability and automation

- Integrates with Agile and DevOps workflows

KILL CHIAN



Stages of the Cyber Kill Chain:



```
graph TD; A[Stages of the Cyber Kill Chain:] --> B[Reconnaissance - Attackers gather information about the target (e.g., scanning networks, identifying vulnerabilities).]; B --> C[Weaponization - They create malware or exploit tools tailored to the target's weaknesses.]; C --> D[Delivery - The malicious payload is sent via phishing emails, infected websites, or other attack vectors.]; D --> E[Exploitation - The malware is executed, exploiting system vulnerabilities.]; E --> F[Installation - The attacker installs backdoors or persistence mechanisms to maintain access.]; F --> G[Command & Control (C2) - The compromised system connects to an attacker-controlled server for remote control.]; G --> H[Actions on Objectives - The attacker achieves their goal, such as data theft, system disruption, or espionage.];
```

Reconnaissance - Attackers gather information about the target (e.g., scanning networks, identifying vulnerabilities).

Weaponization - They create malware or exploit tools tailored to the target's weaknesses.

Delivery - The malicious payload is sent via phishing emails, infected websites, or other attack vectors.

Exploitation - The malware is executed, exploiting system vulnerabilities.

Installation - The attacker installs backdoors or persistence mechanisms to maintain access.

Command & Control (C2) - The compromised system connects to an attacker-controlled server for remote control.

Actions on Objectives - The attacker achieves their goal, such as data theft, system disruption, or espionage.

assets, attack surfaces, and entry point

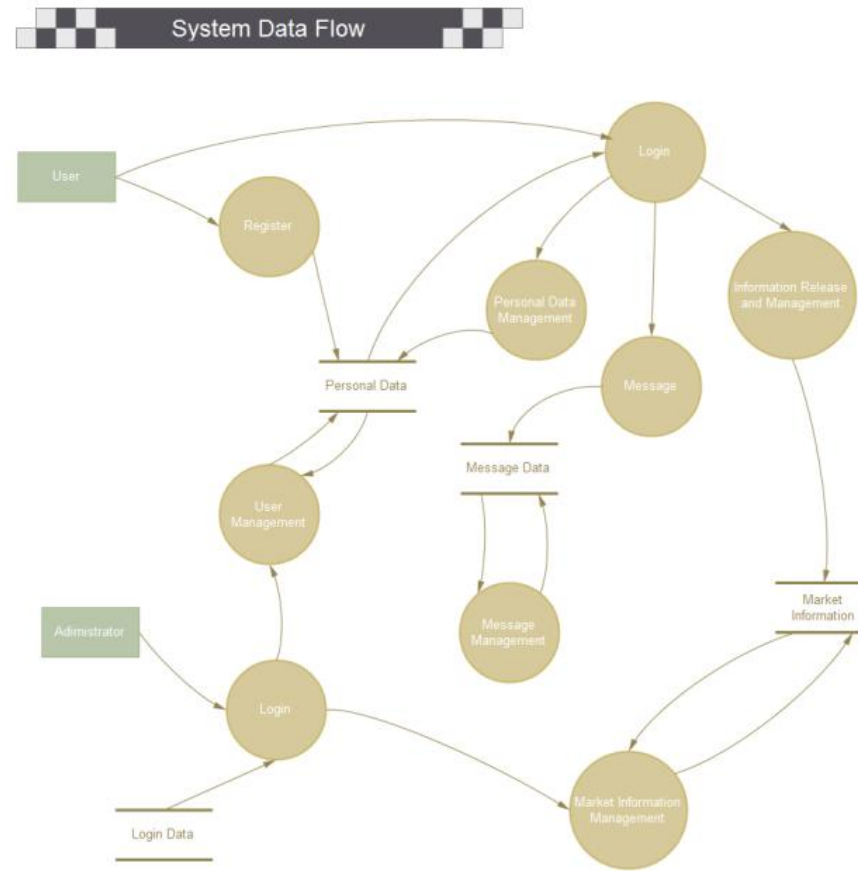
Assets: Hardware, software, data, people—all critical elements requiring protection.

Attack Surfaces: External (APIs, websites), internal (databases, employee systems), physical (IoT devices, USB ports).

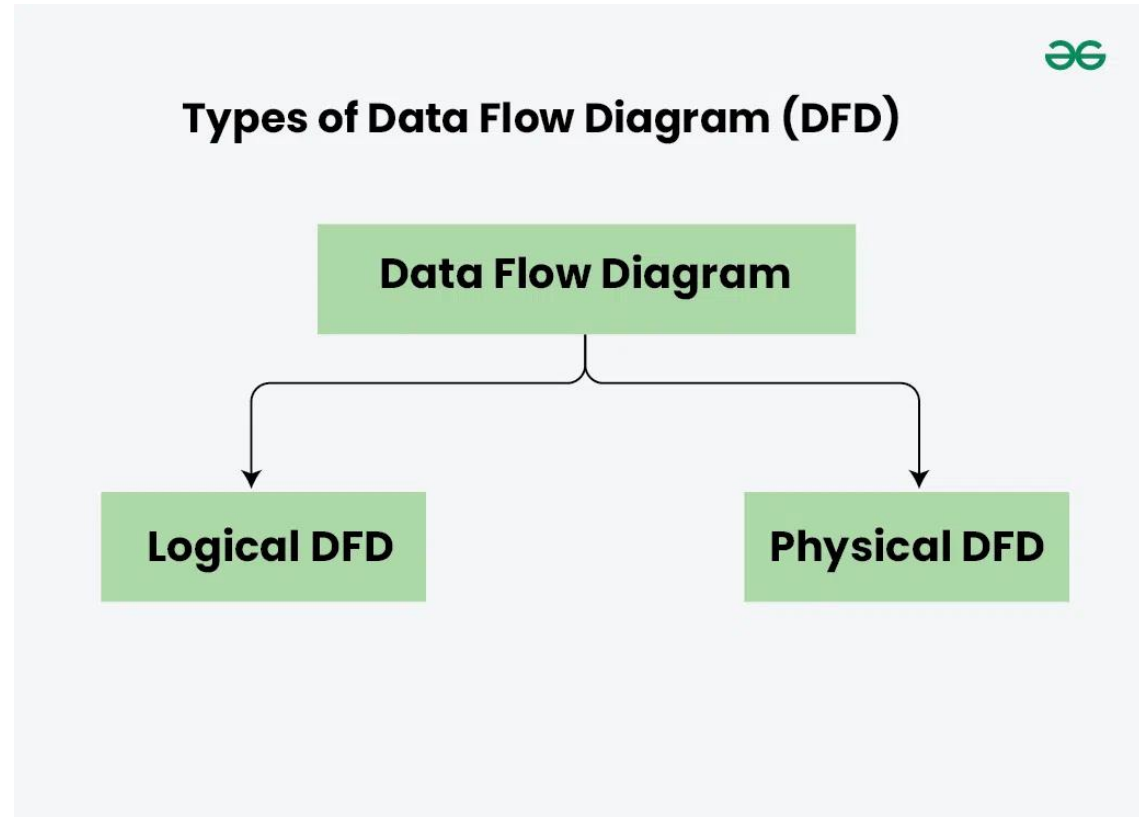
Entry Points: Network (open ports, Wi-Fi), applications (login pages, APIs), human-based (phishing, weak passwords).

◆ **Mitigation:** Regular assessments, encryption, MFA, firewalls, and intrusion detection systems (IDS).

Data flow diagrams






Types of Data Flow Diagram (DFD)

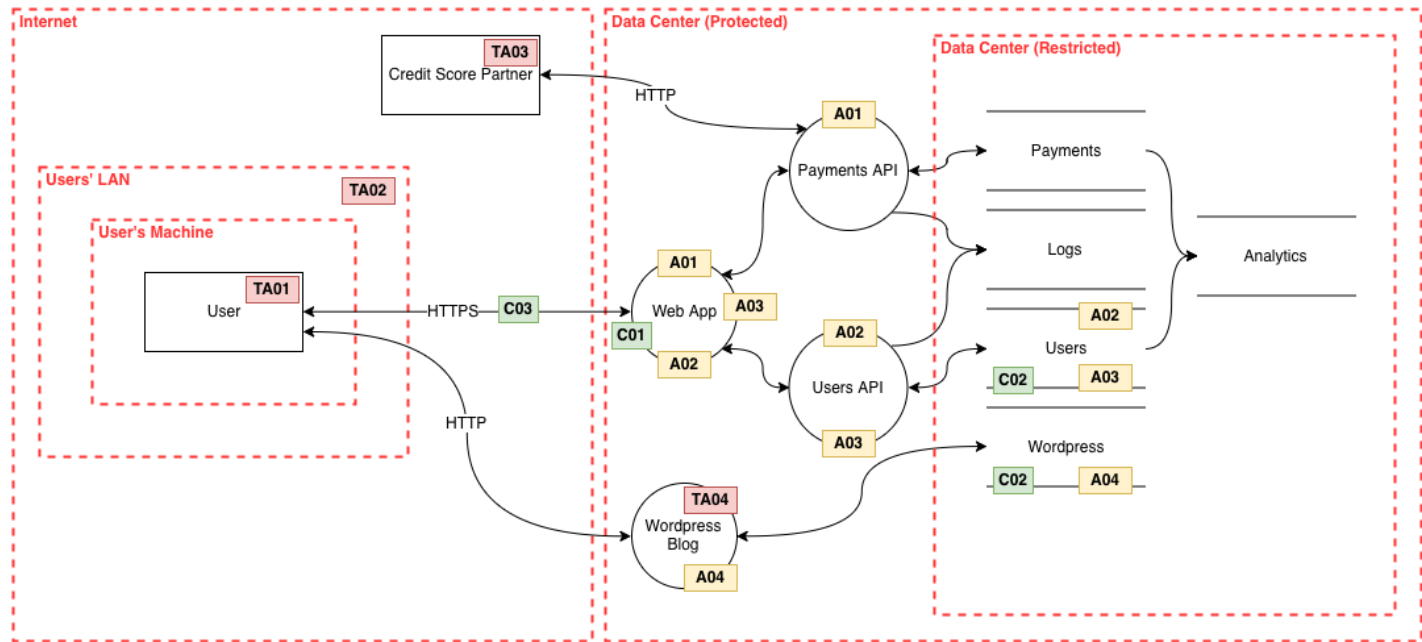




Data-flow diagram elements

| Component | Yourdon and Coad symbol |
|----------------------------|---|
| <div>External entity</div> |  |
| <div>Process</div> |  |
| <div>Data store</div> |  |
| <div>Data flow</div> |  |





| Assets | |
|--------|--------------------------------|
| ID | Description |
| A01 | Credit card data |
| A02 | User PII |
| A03 | User credentials |
| A04 | Blog administrator credentials |

| Security Controls | |
|-------------------|------------------|
| ID | Description |
| C01 | Authentication |
| C02 | Password hashing |
| C03 | TLS |

| Threat Actors | |
|---------------|--|
| ID | Description |
| TA01 | Malicious user |
| TA02 | LAN Man-In-The-Middle |
| TA03 | Malicious/compromised credit score partner |
| TA04 | Compromised Wordpress blog |

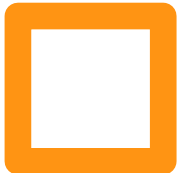
The image features a solid salmon-colored background. Two thin, black lines intersect diagonally. One line runs from the top-left towards the bottom-right, and the other runs from the top-right towards the bottom-left. They cross each other in the upper-left quadrant of the image.

Hands-On Lab

Identifying Potential Threats in Network Access Points

SecureNet Solutions is a global enterprise IT company specializing in network security, cloud infrastructure, and managed services. The company operates data centers across multiple regions and provides secure connectivity solutions for businesses.

SecureNet Solutions uses HPE Aruba Networking Central On-Premises alongside other networking and security solutions to ensure high-performance, secure, and scalable network management.



The image features a solid salmon-colored background. Two thin, black lines intersect diagonally. One line runs from the top-left towards the bottom-right, and the other runs from the top-right towards the bottom-left. They cross each other in the upper-left quadrant of the image.

Hands-On Lab



Agenda for the day

Threat Dragon Threat Analysis

1. Investigate the threat posed by the dragon
2. . Develop a model for its creation
3. . Document the Data Flow Diagram (DFD)
4. Apply STRIDE threat analysis methodology
5. . Generate a report
6. Hand-On lab

OWASP Threat Dragon





What is Threat Dragon?

Threat Dragon is a free, open-source threat modeling application designed to help security teams identify and mitigate potential risks in software systems. It allows users to create data-flow diagrams to visualize how information moves through a system, pinpoint security threats, and document necessary remediations.

Key Features:

Cross-platform: Works across different operating systems.

Threat modeling: Supports structured frameworks for identifying security risks.

Diagram-based analysis: Uses visual models to map out potential vulnerabilities.



What is Threat Dragon?

Supported Frameworks:

Threat Dragon incorporates several established threat modeling methodologies:

STRIDE: Focuses on six security threats: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege.

LINDDUN: Specialized for privacy threat modeling.

CIA: Evaluates threats based on Confidentiality, Integrity, and Availability.

DIE: Distributed security model emphasizing security durability.

PLOT4ai: Security considerations tailored for AI-driven systems.

Main screen



Main screen - Online



Welcome screen - Online



Threat Dragon v2.4.1-latest

English ▾

Logged in as local-user



Welcome!

You're ready to start making your application designs more secure. You can open an existing threat model or create a new one by choosing one of the options below.



Open an existing threat model



Create a new, empty threat model



Explore a sample threat model



Creating a Threat Model

Threat Model
Edit Page



The Title field is required, while the others are optional but offer valuable context for future reference. Click the **Edit** button to modify the threat model details.

Title - required, other fields optional

Owner - usually a development team or individual

Reviewer - currently limited to one

High-level system description - adds context to your model

Contributors - acknowledges those involved

Threat Model Edit Page

Editing: New Threat Model

Title

New Threat Model

Owner

Reviewer

High level system description

Contributors

Start typing to add a contributor

Diagrams

+ Add a new diagram...

Save

Reload

Cancel

Threat Model Edit Page

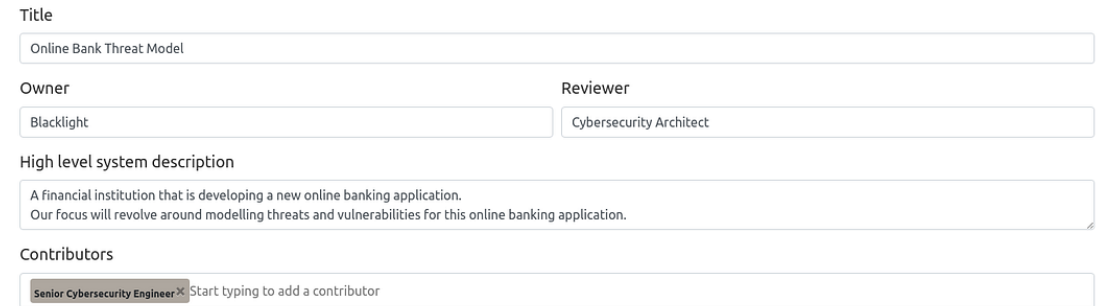
"Title" – Online Bank Threat Model

"Owner" – Blacklight

"Reviewer" – Cybersecurity Architect

"High Level System Description" – A financial institution that is developing a new online banking application. Our focus will revolve around modelling threats and vulnerabilities for this online banking application.

"Contributors" – Senior Cybersecurity Engineer



The screenshot shows a web form for editing a threat model. It contains five main sections: Title, Owner, Reviewer, High level system description, and Contributors. Each section has a text input field. The Title field contains 'Online Bank Threat Model'. The Owner field contains 'Blacklight'. The Reviewer field contains 'Cybersecurity Architect'. The High level system description field contains a paragraph about a financial institution developing a new online banking application. The Contributors field contains a list of contributors, with 'Senior Cybersecurity Engineer' selected and highlighted in orange. The form is styled with a light blue background and rounded corners.

| Title | |
|--|-------------------------|
| Online Bank Threat Model | |
| Owner | Reviewer |
| Blacklight | Cybersecurity Architect |
| High level system description | |
| A financial institution that is developing a new online banking application. Our focus will revolve around modelling threats and vulnerabilities for this online banking application. | |
| Contributors | |
| Senior Cybersecurity Engineer Start typing to add a contributor | |

Add Diagram

Title

Online Bank Threat Model

Owner

Blacklight

Reviewer

Cybersecurity Architect

High level system description

A financial institution that is developing a new online banking application.
Our focus will revolve around modelling threats and vulnerabilities for this online banking application.

Contributors

Senior Cybersecurity Engineer Start typing to add a contributor

Diagrams

STRIDE

e-bank

e-bank STRIDE

Remove

+ Add a new diagram...

Save

Reload

Close

After save

Example threat model

Owner:

Threat Dragon workshop
team

Reviewer:

Threat Dragon workshop
attendees

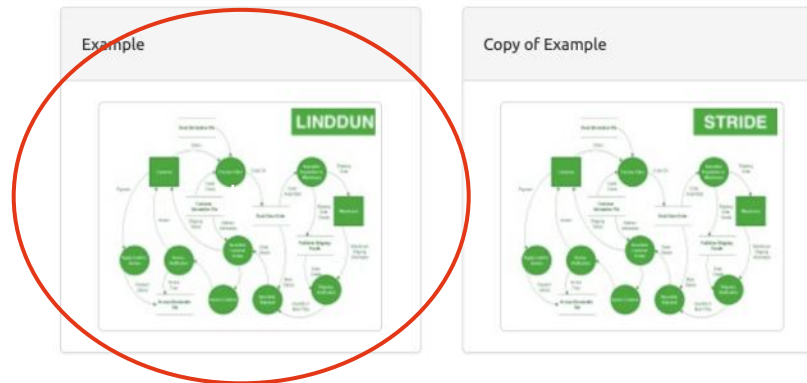
Contributors:

Workshop attendee #1; Workshop attendee #1

High level system description

This is an example model used for the PDX OWASP Training Day 2021
It is a threat model of Threat Dragon itself

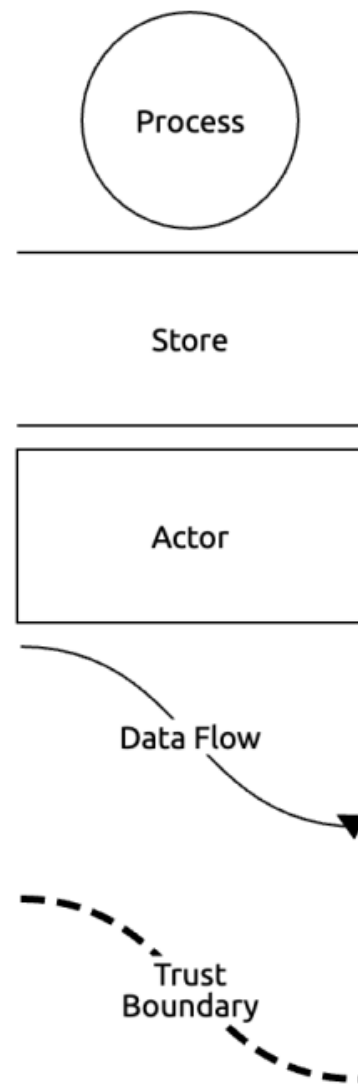
Select the
A diagram
to begin
constructi
ng your
model.



Diagrams

Threat, not system, perspective

- Process
- Store
- Actor
- Data flow
- Trust boundary



Process

Usually a component under our control

- Name
- Description
- Out of scope? Reasoning

Context properties

- Privilege level



Store

Data at rest, almost always within the system but can be external

- The usual Name, Description, Out of scope? & Reasoning

Context properties

- Is a log?
- Stores credentials?
- Is encrypted?
- Is signed?

This could be regarded as an asset

store

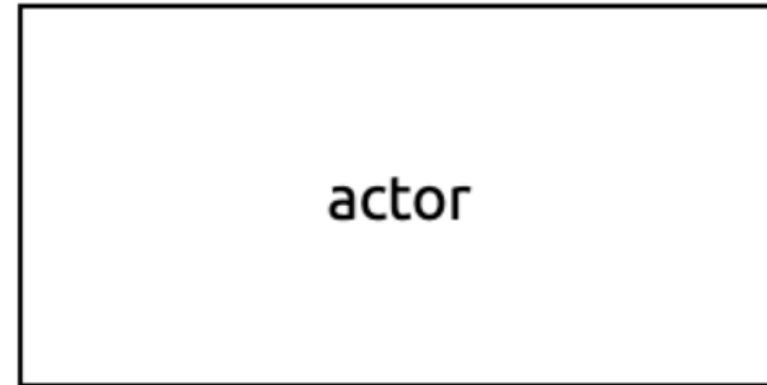
Actor

Commonly a component outside of our system

- The usual Name, Description, Out of scope? & Reasoning

Properties

- Provides authentication?



Data Flow

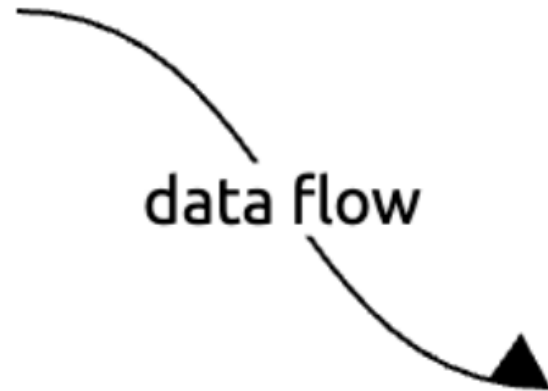
Data in transit, often cross trust boundaries

- The usual Name, Description, Out of scope? & Reasoning

Properties

- Protocol
- Is encrypted?
- Is over a public network?

Two ways to create data flow



Data Flow

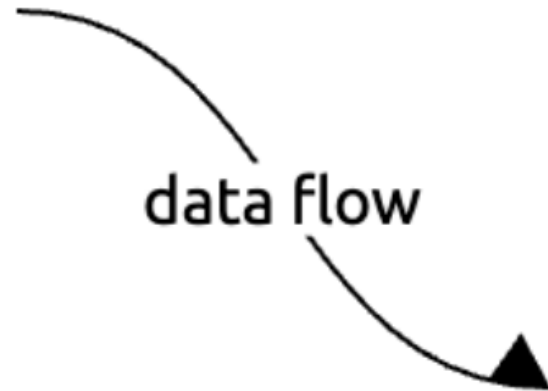
Data in transit, often cross trust boundaries

- The usual Name, Description, Out of scope? & Reasoning

Properties

- Protocol
- Is encrypted?
- Is over a public network?

Two ways to create data flow



Trust Boundary

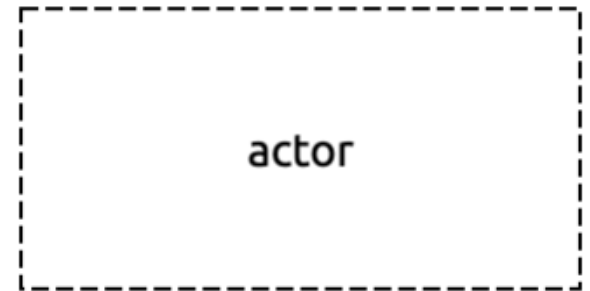
- Name is optional in this case
- No other properties
- It is not a box (yet)
- *The most important of the elements*



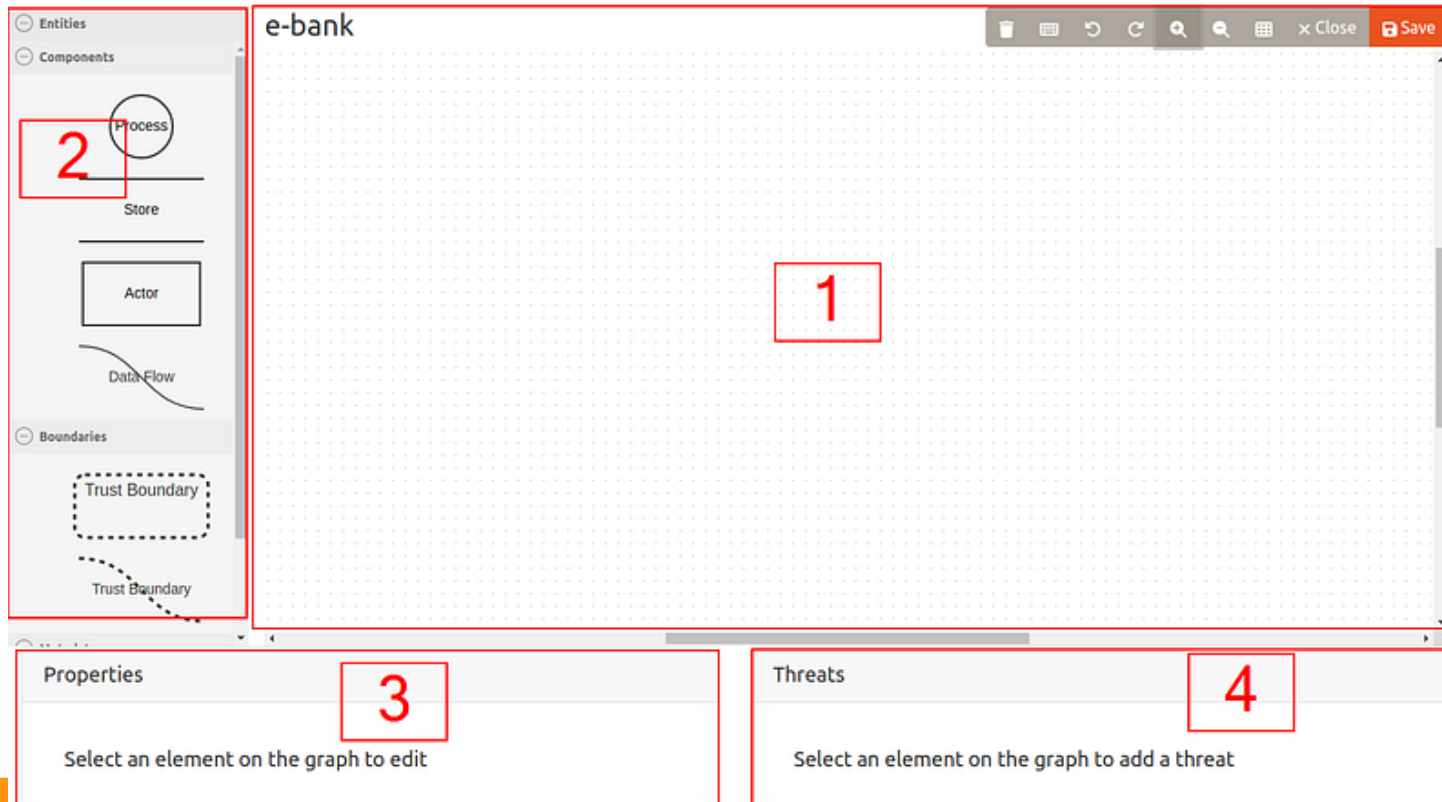
Scope

Scope for diagram components

- Components can be declared out of scope
- Useful for focussing on important components
- Boundaries never out of scope
- Try and give a reasoning
- *Helps incremental*



Diagram



1. The canvas where you'll construct the model
2. The **"entities"** pane where you can find the **"components"**, **"boundaries"** and **"metadata"**.
3. The **"properties"** pane where you can tweak the properties of entities including their names, descriptions and if they are out of scope.
4. The **"threats"** pane; where we'll add new threats to the entities

Threats

The reason for the threat model

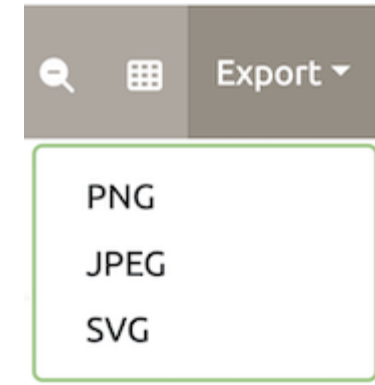
- STRIDE / CIA / LINDDUN
- You can mix and match
- Status: NA / Open / Mitigated
- Priority: Low / Medium / High
- Description of threat
- Mitigation or even prevention



Toolbar



1. Delete the selected element(s)
2. Configure the keyboard shortcuts from the defaults
3. Undo and Redo edits
4. Zoom In and Zoom Out
5. Toggle gridlines on/off, allowing for neater models
6. Close the diagram and return to the threat model details view
7. Save the threat model



Threat generation

Threats

+ New Threat

+ New Threat by Type

+ New Threat by Context

Edit Threat #1

Title

New STRIDE threat

Type

Elevation of privilege

Status

N/A Open Mitigated

Score

Severity

TBD Low Medium High Critical

Description

Provide a description for this threat

Mitigations

Provide remediation for this threat or a reason if status is N/A

Remove

Apply

Threat properties

- All threats have the following properties:
- **Title** is free form text, usually a short descriptive title
- **Type** is a category selection determined by the diagram type (STRIDE / LINDDUN / PLOT4ai / CIA / CIA-DIE / Generic)
- **Status** is one of N/A / Open / Mitigated
- **Score** contains a free text field, often used to score the threat from 0.0 to 10.0 but can be any text or CVSS score
- **Severity** is one of TBD / Low / Medium / High / Critical, similar to CVSS
- **Description** of the threat and possible impact
- **Mitigations** for the threat, probably a remediation from TAME (Transfer / Accept / Mitigate / Evade)

Threats by element type -Threats by context

The components on the diagram have type-specific properties,

for example the Actor component has a property 'Provides Authentication' via a check-box. These properties are used to determine context-specific threat suggestions using 'New Threat by Context'.

At present the suggestions are based on the OWASP Automated Threats to Web Applications, commonly known as [OATS](#). The threat suggestion can be accepted using **Apply** and cycle through the threats using the **Previous** and **Next** buttons. Use **Cancel** to exit the suggestion sequence.

New Threat #1

Title

Carding

Type

Information disclosure

Status

N/AOpenMitigated

Score

Severity

TBDLowMediumHighCritical

Description

See OWASP Automated Threat #1:
Lists of full credit/debit card data are tested against a merchant's payment processes to identify valid card details

Mitigations

Defences include control of interaction frequency, enforcement of a single unique a action and preventing abuse of functionality

PreviousNext

CancelApply

Threats by element type

The threat model can have different types of threats added to it according to the diagram type. Currently the supported types are STRIDE, LINDDUN, CIA, CIA-DIE and PLOT4ai; these are configured as part of the diagram attributes when editing the model. A 'Generic' type is provided so that you can select any type of threat from any of the categories.

New Threat #2

Title

New STRIDE threat

Type

Repudiation

Status

N/A

Open

Mitigated

Score

Severity

TBD

Low

Medium

High

Critical

Description

Provide a description for this threat

Mitigations

Provide remediation for this threat or a reason if status is N/A

Previous

Next

Cancel

Apply

Component by Category

| Component | Spoofing | Tampering | Repudiation | Information Disclosure | Denial of Service | Elevation of Privilege |
|-------------------|--|--|--|---|--|---|
| External Entities | ✓ Possible (User Impersonation, API Spoofing) | ✗ Rare | ✓ Possible (Lack of Audit Trails) | ✗ Rare | ✗ Rare | ✗ Rare |
| Processes | ✗ Rare | ✓ Common (Data Manipulation, Process Hijacking) | ✓ Possible (No Logging for Actions) | ✓ Common (Leaking Data via APIs) | ✓ Common (System Overload Attacks) | ✓ Critical (Privilege Escalation Exploits) |
| Data Stores | ✗ Rare | ✓ Common (Unauthorized Data Modifications) | ✓ Possible (Weak Logging) | ✓ Critical (Unencrypted Sensitive Data) | ✗ Rare | ✓ Possible (Unauthorized Access to Database) |
| Data Flows | ✗ Rare | ✓ Common (MITM Attacks, Packet Tampering) | ✗ Rare | ✓ Critical (Unsecured Data Transmission) | ✓ Common (Flooding API Calls, Network DDoS) | ✗ Rare |

Framework by Category

STRIDE threats by element

| | S | T | R | I | D | E |
|---------|---|---|---|---|---|---|
| ACTOR | X | | X | | | |
| STORE | | X | X | X | X | |
| PROCESS | X | X | X | X | X | X |
| FLOW | | X | | X | X | |


LINDDUN threats by element

| | L | I | N | D | D | U | N |
|---------|---|---|---|---|---|---|---|
| ACTOR | X | X | | | | X | |
| STORE | X | X | X | X | X | | X |
| FLOW | X | X | X | X | X | | X |
| PROCESS | X | X | X | X | X | | X |

PLOT4ai threats by element





| | T | A | I | S | S | U | E | N |
|---------|---|---|---|---|---|---|---|---|
| ACTOR | | X | X | X | X | X | X | |
| STORE | X | X | X | X | | | | X |
| FLOW | X | | X | X | | | | X |
| PROCESS | X | X | X | X | | | | X |

Report

Threat Dragon v2.4.1-latest

English ▾

Logged in as local-user



☒ Show model diagrams


☒ Show mitigated threats


☒ Show out of scope elements

☒ Show empty elements

☐ Threat Dragon logo

☐ Show element properties

 Print

 Close

Wed May 28 2025

Threat model report for Online Payments Processing Platform

Owner:
A development team

Reviewer:
A security architect

Contributors:
development engineers, product managers, security architects

Executive Summary

High level system description

This threat model has been provided by the OWASP Threat Model Cookbook: [threat-model-cookbook/Flow Diagram/payment](#)

Summary

| Metric | Total |
|--------------------------|-------|
| Total Threats | 0 |
| Total Mitigated | 0 |
| Total Open | 0 |
| Open / Critical Severity | 0 |
| Open / High Severity | 0 |
| Open / Medium Severity | 0 |



Hands-On Lab

Exercise: Threat Modeling with OWASP Threat Dragon



Group Discussion
and Q&A

Sharing Insights and Experiences



Collaborative Learning

Encouraging collaboration enhances understanding and creates a supportive environment for sharing knowledge.

Insights on Threat Modeling

Participants can share their insights on threat modeling to improve awareness and strategies in software security.

Experiences in Software Security

Sharing experiences related to software security fosters best practices and collective growth among participants.

Addressing Questions and Clarifications



Interactive Q&A Session

Participants are encouraged to ask questions to clarify any doubts about threat modeling discussed in the workshop.

Clarification of Topics

This time allows for a deeper understanding of the topics covered, ensuring everyone is on the same page.

Enhancing Understanding

The goal is to enhance understanding of threat modeling by addressing participants' specific questions.

A series of white, overlapping geometric lines and polygons on a black background, located on the left side of the slide.

Conclusion

Understanding Threat Modeling

Enhancing Software Security

Workshop Takeaways



Agenda

Introduction

Building confidence

Engaging the audience

Visual aids

Final tips & takeaways



The power
of communication


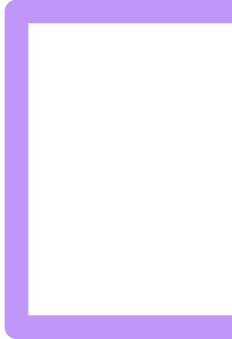


Overcoming nervousness

Confidence-building
strategies



Engaging the audience

- Make eye contact with your audience to create a sense of intimacy and involvement
 - Weave relatable stories into your presentation using narratives that make your message memorable and impactful
 - Encourage questions and provide thoughtful responses to enhance audience participation
 - Use live polls or surveys to gather audience opinions, promoting engagement and making sure the audience feel involved
- 
- 



Selecting visual aids

Enhancing your
presentation

Effective delivery techniques

This is a powerful tool in public speaking. It involves varying pitch, tone, and volume to convey emotion, emphasize points, and maintain interest.

- Pitch variation
- Tone inflection
- Volume control

Effective body language enhances your message, making it more impactful and memorable.

- Meaningful eye contact
- Purposeful gestures
- Maintain good posture
- Control your expressions

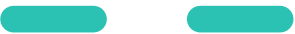


Navigating Q&A sessions

- Know your material in advance
- Anticipate common questions
- Rehearse your responses

Maintaining composure during the Q&A session is essential for projecting confidence and authority. Consider the following tips for staying composed:

- Stay calm
- Actively listen
- Pause and reflect
- Maintain eye contact



Speaking impact

Your ability to communicate effectively will leave a lasting impact on your audience

Effectively communicating involves not only delivering a message but also resonating with the experiences, values, and emotions of those listening



Dynamic delivery

Learn to infuse energy into your delivery to leave a lasting impression

One of the goals of effective communication is to motivate your audience

| Metric | Measurement | Target | Actual |
|-------------------------------|----------------|--------|--------|
| Audience attendance | # of attendees | 150 | 120 |
| Engagement duration | Minutes | 60 | 75 |
| Q&A interaction | # of questions | 10 | 15 |
| Positive feedback | Percentage (%) | 90 | 95 |
| Rate of information retention | Percentage (%) | 80 | 85 |

Final tips & takeaways

Consistent rehearsal

- Strengthen your familiarity

Refine delivery style

- Pacing, tone, and emphasis

Timing and transitions

- Aim for seamless, professional delivery

Practice audience

- Enlist colleagues to listen & provide feedback

Seek feedback

Reflect on performance

Explore new techniques

Set personal goals

Iterate and adapt



Speaking engagement metrics

| Impact factor | Measurement | Target | Achieved |
|-----------------------------|--------------------|--------|----------|
| Audience interaction | Percentage (%) | 85 | 88 |
| Knowledge retention | Percentage (%) | 75 | 80 |
| Post-presentation surveys | Average rating | 4.2 | 4.5 |
| Referral rate | Percentage (%) | 10 | 12 |
| Collaboration opportunities | # of opportunities | 8 | 10 |



Thank you

Brita Tamm

502-555-0152

brita@firstupconsultants.com

www.firstupconsultants.com