Threat Modeling

- Threat Matrix
- Trust Boundaries
- Security Controls
- STRIDE Framework
    - **S**poofing
    - **T**ampering
    - **R**epudiation
    - **I**nformation disclosure
    - **D**enial of service
    - **E**levation of privilege
- MITRE ATT&CK Framework
- Excellent talk on "Defense Against the Dark Arts" by Lilly Ryan (contains *many* Harry Potter spoilers)

Threat Matrix

A Threat Matrix is **a structured framework or table used to systematically assess, prioritize, and map various security threats to an organization, system, or specific asset**. It helps security teams categorize and understand potential risks based on different threat actors, attack vectors, and the potential impact on the organization. **The Threat Matrix is a valuable tool in threat modeling, which is the process of identifying and evaluating threats to systems and developing strategies to mitigate or respond to them**.

Key Features of a Threat Matrix:

1. Structured **Threat Categorization**:

- A threat matrix organizes different threats into categories, allowing security teams to systematically evaluate how different types of attacks may occur and what their potential impact might be. Categories often include:
    - **Threat Actors**: The individuals or entities that may carry out the attack (e.g., nation-states, cybercriminals, insiders).
    - **Attack Vectors**: The methods or techniques an attacker might use to compromise a system (e.g., phishing, brute force, malware).
    - **Assets**: The systems, data, or resources that may be targeted.
    - **Impact**: The potential damage or disruption that a threat could cause if successfully executed (e.g., data theft, system downtime, financial loss).

2. **Prioritization** of Threats:

- The matrix can help **prioritize threats based on the likelihood of an attack and the severity of its impact**. By organizing threats in this way, security teams can focus their efforts on the most critical areas that need mitigation or monitoring.

3. **Mapping Attack Scenarios**:

- A Threat Matrix often includes **mapping specific attack scenarios to relevant vulnerabilities**. This helps teams understand which vulnerabilities are most likely to be exploited by certain types of attacks, allowing them to focus on addressing the highest-risk areas.

4. Real-World Examples:

- One well-known example of a threat matrix is **the MITRE ATT&CK Matrix, which organizes various techniques and tactics that attackers use during different stages of an attack**. It is widely used by security professionals to model adversary behavior and map defensive measures.

Example: MITRE ATT&CK Matrix

The MITRE ATT&CK Matrix is a specific and widely adopted threat matrix framework that categorizes techniques and tactics based on real-world observations of cyberattacks. It **breaks down attacks into stages (called tactics)** such as initial access, execution, privilege escalation, lateral movement, and more. Within each tactic, the matrix **lists different techniques** that adversaries can use.

- Key Components of MITRE ATT&CK:
    - **Tactics**: The goals that adversaries aim to achieve at different **stages** of an attack (e.g., gaining initial access, maintaining persistence).
    - **Techniques**: The specific **methods** used to accomplish a tactic (e.g., spear-phishing, command injection).
    - **Procedures**: The **steps** attackers take to execute the techniques in real-world scenarios.

Why a Threat Matrix Is Important in Threat Modeling:

- **Risk Prioritization**: A Threat Matrix helps identify which threats are the most relevant to an organization, allowing security teams to prioritize mitigation efforts accordingly.
- **Understanding Attack Paths**: By mapping out potential attack paths in a structured way, a threat matrix helps security teams see how different vulnerabilities might be exploited and which defenses are most effective.
- **Facilitates Communication**: **The structured approach of a threat matrix makes it easier for security teams to communicate risks and threats to other stakeholders**, such as management or IT teams, helping guide security planning and resource allocation.

Summary:

A Threat Matrix is a vital tool in threat modeling that organizes and prioritizes potential security threats based on factors like attack vectors, threat actors, and the impact of successful attacks. It helps security teams understand, categorize, and mitigate risks systematically. Frameworks like the MITRE ATT&CK Matrix offer real-world examples of how threat matrices are used to map adversary techniques and tactics, providing valuable insights for defense and detection strategies.

Trust Boundaries

Trust boundaries are **conceptual lines within a system or network architecture that separate areas of different trust levels**. **Crossing a trust boundary generally requires authentication, authorization, or inspection to ensure that data and operations are permitted**. By defining trust boundaries, organizations **can identify where sensitive operations or data interactions occur and apply appropriate security controls to protect against unauthorized access or abuse**.

Key Concepts of Trust Boundaries:

1. **Different Levels of Trust**:

- Trust boundaries **separate areas with different security requirements**. For example, a public-facing web server is typically less trusted than an internal database containing sensitive data, so a trust boundary would exist between them.
- Examples of trust boundaries include:
    - **Internal vs. External Networks**: Separating internal corporate networks from the internet.
    - **Application Layers**: Dividing the application's front-end (e.g., user input forms) from its back-end (e.g., databases).
    - **User Roles**: Separating areas accessible to regular users from those accessible only to administrators.

2. **Identification of Sensitive Interactions**:

- Trust boundaries **help identify where sensitive interactions take place in a system**. These interactions require stricter controls, such as encryption, authentication, and access restrictions.
- Example: Between a web application's front end and a database where user credentials are stored, a trust boundary would exist, as unauthorized access to the database could expose sensitive information.

3. **Security Controls at Trust Boundaries**:

- To secure trust boundaries, security controls are implemented based on the sensitivity of the resources and the level of trust required.
- Common controls include:
    - **Firewalls**: Separate trusted internal networks from untrusted external networks.
    - **Access Control**: Role-based access to limit interactions across boundaries to authorized users.
    - **Encryption**: Protects data as it crosses boundaries, such as between client devices and servers.
    - **Input Validation**: Ensures data integrity when crossing boundaries, such as filtering user input to prevent SQL injection.

4. Trust Boundary Violations:

- Attackers often try to exploit trust boundaries to gain access to restricted areas of a system. By crossing a trust boundary without proper authorization, attackers can potentially access sensitive data or services.

- Example: A compromised web server might allow attackers to access the internal database if there aren't strict boundary controls.

5. Examples of Trust Boundaries in Different Environments:

- Web Applications: **Between the client and server**; for instance, user input fields (client) are separated from the database (server) by a trust boundary.
- Network Segmentation: **Separates user devices from sensitive networks**, such as corporate or server networks.
- Microservices and Containers: In microservices architectures, **each service might have a trust boundary to isolate sensitive services and prevent unauthorized data access between them**.

Why Trust Boundaries Matter in Threat Modeling:

In threat modeling, **defining trust boundaries helps identify where security risks are the greatest**. By understanding where untrusted or partially trusted entities interact with sensitive data or services, security teams can:

- **Map Attack Surfaces**: Understand which parts of the system are exposed to potential attacks and focus on protecting these boundaries.
- **Apply Appropriate Controls**: Decide on the right level of security controls based on the sensitivity of data or services crossing each boundary.
- **Prevent Privilege Escalation**: Prevent attackers from moving laterally across boundaries and gaining unauthorized access to restricted areas.

Summary:

Trust boundaries define the separation of areas with different trust levels within a system or network. They highlight sensitive areas where security controls are essential to prevent unauthorized access, data leakage, or privilege escalation. Properly managing trust boundaries is crucial for securing interactions in systems and networks, helping to prevent attackers from exploiting boundaries to gain unauthorized access to sensitive information or systems.

Security Controls

Security controls are **measures implemented to reduce risk, protect assets, and ensure the integrity, confidentiality, and availability of information systems**. They are **used to prevent, detect, mitigate, and respond to threats and vulnerabilities** in a system or network. Security controls can be **technical, physical, or administrative in nature, and they work together to create a layered defense** that safeguards an organization's data and resources.

Types of Security Controls:

1. **Preventive** Controls:

- Purpose: Prevent security incidents by stopping unauthorized access or actions before they occur.
- Examples:
    - **Firewalls**: Block unauthorized access to network resources.
    - **Access Control**: Limits access based on roles, permissions, or attributes to ensure only authorized users can perform specific actions.
    - **Encryption**: Protects data in transit or at rest, making it unreadable to unauthorized users.
    - **Multi-Factor Authentication (MFA)**: Requires additional verification factors for access, reducing the risk of unauthorized login.

2. **Detective** Controls:

- Purpose: Identify and detect incidents or suspicious activities in real time, allowing for a quick response.
- Examples:
    - **Intrusion Detection Systems (IDS)**: Monitors network traffic to detect potential threats, such as malware or port scanning.
    - **SIEM** Systems: Collects and analyzes log data from various sources to detect security incidents and generate alerts.
    - **File Integrity Monitoring**: Tracks changes to critical files, alerting security teams of unexpected modifications.
    - **Security Audits**: Regular audits to detect non-compliance with security policies and procedures.

3. **Corrective** Controls:

- Purpose: Respond to and fix issues after they have been detected, minimizing the impact of an incident.
- Examples:
    - **Patching**: Applies updates to software and systems to fix vulnerabilities after they've been identified.
    - **Backup and Restore**: Restores data or systems to an operational state following a compromise, such as ransomware.
    - **Incident Response**: Steps taken by a security team to contain, investigate, and mitigate the effects of an attack.

4. **Deterrent** Controls:

- Purpose: Discourage attackers from attempting to breach systems by increasing the perceived difficulty of an attack.
- Examples:
    - **Security Awareness Training**: Educates employees on security best practices, reducing the likelihood of social engineering attacks.
    - **Warning Signs and Legal Notices**: Visible signs indicating that unauthorized access is monitored, deterring potential attackers.
    - **Physical Security Measures**: Security cameras, guards, and signage that deter unauthorized access.

5. **Compensating** Controls:

- Purpose: Provide alternative protections when primary controls cannot be implemented.
- Examples:
    - **Network Segmentation**: Divides the network into isolated segments to limit access to sensitive areas.
    - **Application Whitelisting**: Allows only trusted applications to run, providing security when full application control isn't possible.
    - **Access Logging and Monitoring**: Used as a compensating control when strict access control mechanisms are unavailable.

6. **Physical** Controls:

- Purpose: Protect the physical infrastructure where data and systems are stored, limiting access to authorized personnel.
- Examples:
    - **Locks**: Secures doors, cabinets, or equipment from unauthorized access.
    - **Security Cameras**: Monitor physical spaces to detect and deter unauthorized access.
    - **Biometric Scanners**: Use physical characteristics, like fingerprints or retina scans, to verify identity.

7. **Administrative** Controls:

- Purpose: Implement policies, procedures, and guidelines to manage security within an organization.
- Examples:
    - **Security Policies**: Define acceptable use, data protection, and access control policies.
    - **Incident Response Plan**: Provides steps for responding to and recovering from security incidents.
    - **Risk Assessment**: Evaluates and prioritizes security risks within the organization to focus on critical areas.
    - **Employee Training and Awareness**: Educates staff on security protocols and how to recognize potential threats.

Importance of Layered Security Controls:

A strong security posture relies on a combination of these security controls. Using multiple types of controls in a layered or **defense-in-depth** approach creates redundancies, so if one control fails, others are still in place to protect assets. For instance, even if a firewall (preventive) is bypassed, an intrusion

detection system (detective) can still alert security teams of suspicious activity, and backup systems (corrective) can recover lost data.

Summary:

Security Controls are critical measures to protect information and systems from security threats. They can be **preventive, detective, corrective, deterrent, compensating, physical, or administrative**. Each type serves a unique role in securing an organization, and **together they provide a robust defense strategy to detect, prevent, and respond to potential security incidents**.

STRIDE Framework

The STRIDE framework is **a threat modeling approach developed by Microsoft that helps identify potential security threats in a system**. It categorizes threats based on the types of attacks or vulnerabilities that could be exploited, providing a structured way to assess and mitigate risks. STRIDE stands for **Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege**. Each category represents a specific type of threat.

Components of STRIDE:

1. **Spoofing**:

- Definition: Spoofing occurs when an **attacker impersonates another user or device**, usually by falsifying identity credentials, such as usernames, passwords, or IP addresses.
- Example: An attacker uses stolen credentials to log in to a system as an authorized user, gaining access to restricted resources.
- Mitigation: Implement **strong authentication mechanisms** (e.g., multi-factor authentication), secure password policies, and digital certificates to verify identity.

2. **Tampering**:

- Definition: Tampering involves **maliciously modifying data or code**. This can **occur in transit, at rest, or even within an application**, compromising data integrity and potentially leading to unauthorized actions.
- Example: An attacker intercepts and modifies network traffic to inject malicious commands or modify data in a database.
- Mitigation: **Use encryption** to protect data in transit, **implement integrity checks** (e.g., digital signatures or checksums), and control file permissions to **restrict data access**.

3. **Repudiation**:

- Definition: Repudiation refers to the ability of **users or attackers to deny actions they have performed**. Without proper logging and accountability, it can be difficult to trace activities back to the responsible party.
- Example: A user performs a transaction but later denies it, and there is no log or evidence to prove the action.
- Mitigation: **Implement audit logs with timestamps and user identifiers to track actions**. Ensure logs are **secure and tamper-resistant for accountability**.

4. **Information Disclosure**:

- Definition: Information disclosure involves **unauthorized access to sensitive information, exposing it to unintended recipients**. This compromises confidentiality.
- Example: Sensitive data like personally identifiable information (PII) is accidentally exposed via an unsecured API endpoint or a public-facing server.
- Mitigation: **Use encryption** for data storage and transmission, control access with **role-based permissions**, and **conduct regular audits to detect potential data leaks**.

5. **Denial of Service (DoS)**:

- Definition: Denial of Service occurs when **an attacker overwhelms a system, network, or application, causing it to become unavailable to legitimate users**.
- Example: An attacker sends a massive number of requests to a server, exhausting its resources and making it inaccessible to other users.
- Mitigation: **Implement rate limiting, load balancing, and redundancy**. Employ DDoS protection measures, such as web application firewalls (WAFs) and traffic filtering.

6. **Elevation of Privilege**:

- Definition: Elevation of privilege occurs when **an attacker gains higher access levels than intended**, allowing them to perform actions beyond their normal permissions.
- Example: A regular user exploits a vulnerability to gain administrator access and modify system configurations.
- Mitigation: Use the principle of **least privilege**, regularly **patch** systems to fix vulnerabilities, and implement **strict access control** policies.

Summary of STRIDE:

Threat Type	Definition	Example	Mitigation
Spoofing	Impersonating another user or system	Using stolen credentials	Multi-factor authentication, secure passwords
Tampering	Modifying data or code maliciously	Modifying database entries	Encryption, integrity checks
Repudiation	Denying an action or transaction	Denying a transaction with no audit log	Secure audit logging
Information Disclosure	Unauthorized access to sensitive information	Exposing PII via unsecured API	Encryption, access control
Denial of Service	Making a system or service unavailable by overwhelming it	Sending excessive requests to exhaust server resources	Rate limiting, redundancy, DDoS protection
Elevation of Privilege	Gaining unauthorized access to higher privileges	Exploiting a vulnerability to gain admin rights	Least privilege, regular patching

Purpose of STRIDE in Threat Modeling:

The STRIDE framework helps security teams systematically identify potential threats and vulnerabilities, enabling them to develop countermeasures for each specific threat type. It is often used during system design to assess security risks and ensure that appropriate protections are in place to defend against different types of attacks.

MITRE ATT&CK Framework

The MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) Framework is **a comprehensive knowledge base that categorizes the tactics, techniques, and procedures (TTPs) used by attackers throughout the lifecycle of a cyberattack**. Developed by MITRE Corporation, ATT&CK helps cybersecurity professionals **understand and defend against adversarial behavior by mapping real-world attack techniques to the different phases of an attack**. The framework is widely used in threat modeling, detection, and incident response.

Key Components of the MITRE ATT&CK Framework:

1. **Tactics**:

- Tactics represent the goals or objectives an attacker aims to achieve at each **stage of an attack**. Each tactic aligns with a specific phase in the attack lifecycle, providing a high-level view of the attacker's goals.
- Examples of Tactics:
    - Initial Access: Gaining entry into the target network.
    - Persistence: Maintaining a foothold within the system.
    - Privilege Escalation: Obtaining higher access levels to perform unauthorized actions.
    - Lateral Movement: Moving through the network to find additional targets.
    - Exfiltration: Stealing data from the system.

2. **Techniques**:

- Techniques describe **the specific methods or actions attackers use to achieve their objectives within each tactic**. Techniques provide detailed descriptions of how adversaries can compromise systems, evade detection, and achieve their goals.
- Examples of Techniques:
    - Phishing (Initial Access): Using email to deceive users into downloading malware or disclosing credentials.
    - Credential Dumping (Credential Access): Extracting stored passwords or hashes from a system to facilitate unauthorized access.
    - Remote File Copy (Lateral Movement): Transferring files from one host to another to establish control over new systems.
    - Data Compression (Exfiltration): Compressing files to reduce their size before exfiltrating data out of the target network.

3. **Sub-Techniques**:

- Sub-techniques break down each technique into **more specific actions, providing a more granular level of detail**. Sub-techniques allow organizations to understand the exact methods attackers might use under each main technique.
- Example of a Technique with Sub-Techniques:
    - Phishing (Initial Access) has sub-techniques like Spear Phishing Attachment (sending malicious attachments), Spear Phishing Link (sending malicious links), and Spear Phishing via Service (using a third-party service to deliver phishing messages).

4. **Procedures**:

- Procedures describe **specific, real-world implementations of techniques**. They document how different threat actors or malware families use certain techniques in actual attacks, providing context and examples of the techniques in action.
- Example: APT28 (a known threat group) using spear-phishing emails with malicious attachments to gain initial access to a target organization.

MITRE ATT&CK Matrices:

The MITRE ATT&CK framework provides multiple matrices that organize tactics and techniques according to different environments, such as:

- Enterprise: Focused on attacks against enterprise networks and systems, including Windows, macOS, Linux, and cloud environments.
- Mobile: Contains tactics and techniques specific to mobile platforms, including both Android and iOS.
- ICS (Industrial Control Systems): Focused on attacks against industrial and critical infrastructure systems, such as SCADA (Supervisory Control and Data Acquisition) and OT (Operational Technology) environments.

Each matrix is a structured table where the rows represent techniques and the columns represent tactics. This structure allows analysts to see how each technique aligns with a specific tactic in the attack lifecycle.

Common Use Cases for the MITRE ATT&CK Framework:

1. **Threat Detection and Response**:

- Security teams use ATT&CK to map observed attacker behavior to known techniques and tactics. By comparing suspicious activity to the framework, they can better understand the attack's progression and prioritize responses based on the attack's lifecycle stage.

2. **Threat Intelligence**:

- Threat intelligence teams use ATT&CK to profile threat actors by associating their known TTPs with tactics and techniques in the framework. This helps security teams anticipate future actions based on the attacker's profile.

3. **Incident Investigation and Forensics**:

- Investigators use ATT&CK to map out all the observed behaviors during an incident. This helps analysts understand how an attack was conducted and identify which techniques may have been used but weren't detected, highlighting potential gaps in defense.

4. **Red and Blue Teaming**:

- Red teams (offensive security teams) use ATT&CK to simulate realistic attack scenarios by mimicking techniques that real-world adversaries use. Blue teams (defensive teams) use the framework to prepare defenses and monitor for specific techniques in their environment.

5. **Security Gap Analysis**:

- Organizations can use ATT&CK to evaluate their current security controls and see if they are adequately covering known techniques. This helps in identifying potential gaps and implementing additional security measures to cover them.

Example of an ATT&CK Matrix Row:

Here's a simplified example of a row in the MITRE ATT&CK Enterprise Matrix, focused on the "Initial Access" tactic:

Tactic	Technique	Sub-Technique	Procedure (Example)
Initial Access	Phishing	Spear Phishing Attachment	APT28 uses malicious attachments in emails to gain access to target networks
Initial Access	Exploit Public-Facing Application	N/A	Attackers exploit unpatched vulnerabilities in web servers to gain entry
Initial Access	Drive-by Compromise	N/A	APT32 uses compromised websites to deliver malware to unsuspecting visitors

This row shows how techniques and sub-techniques are mapped to the "Initial Access" tactic, along with real-world examples of threat actors using these methods.

Summary:

The MITRE ATT&CK Framework is a widely used threat intelligence and defense framework that categorizes the tactics, techniques, and procedures (TTPs) used by attackers. The framework is structured into matrices that map tactics (the goals attackers aim to achieve) to techniques (the specific methods they use). It is used by cybersecurity professionals for threat detection, threat intelligence, incident response, and gap analysis, providing a common language and structured approach for understanding and defending against cyber threats.