Threat Modeling & Remediation in AWS

Raghav Bijjula

https://www.linkedin.com/in/raghavbijjula

https://twitter.com/BijjulaRaghav

Agenda

- Intro
- Threat Modeling Methodologies
- Components of Threat Modeling
- Mitigation Strategies
- Demo
- Q&A

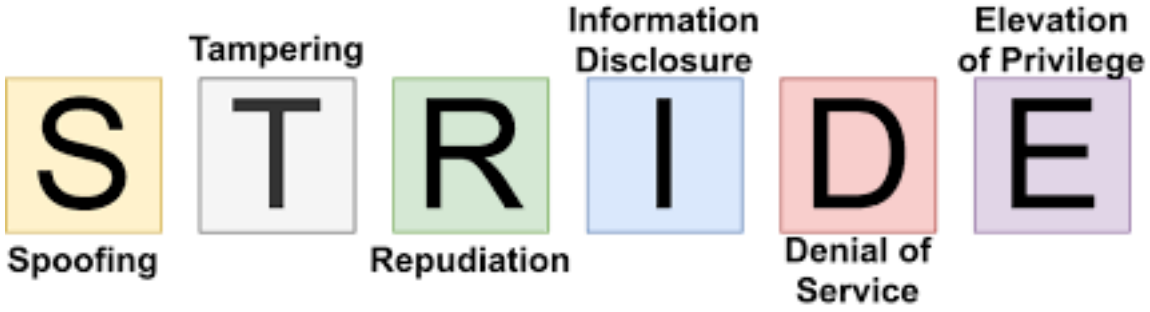
What is Threat Modeling ?

Threat modeling is a structured approach used in cybersecurity and software development to identify, assess, and mitigate security threats and vulnerabilities in a system or application. The primary goal of threat modeling is to proactively anticipate and address potential security issues before they can be exploited by malicious actors. It helps organizations make informed decisions about security controls and resource allocation to protect their assets effectively.

Threat modeling is a process of identifying, analyzing, and mitigating potential threats to a system or asset. It is a systematic way of thinking about security and helps to identify vulnerabilities that could be exploited by attackers.

The goal of threat modeling is to reduce the risk of a security breach by identifying and mitigating threats early in the development or design process. This can help to prevent costly and disruptive security incidents.

Threat Modeling Methodologies



PASTA

Process for Attack Simulation & Threat Analysis

DREAD

- **D**AMAGE
- **R**EPRODUCIBILITY
- **E**XPLOITABILITY
- **A**FFECTED USERS
- **D**ISCOVERABILITY

Components of Threat Modeling

- Asset Identification (e.g., data, software, hardware)
- Threat Enumeration - Sources of Threats (e.g., external attackers, insiders)
- Attack Surface Analysis - Identifying Entry Points and Interfaces
- Threat Assessment - Evaluating Likelihood and Impact
- Mitigation Strategies - Security Controls
- Risk Analysis - Balancing Threat Severity with Mitigation Efforts
- Documentation - Documenting Threat Models

Mitigation Strategies

- **Access Control**: RBAC, MFA, least privilege principle
- **Encryption**: encryption to protect data both in transit and at rest
- **Input Validation**: sanitize user inputs to prevent common attacks like SQL injection and cross-site scripting (XSS)
- **Patch Management**: Regularly update and patch software and systems to fix known vulnerabilities.
- **Security Testing**: security testing, including vulnerability scanning, penetration testing, and code reviews
- **Firewalls and Intrusion Detection/Prevention Systems (IDS/IPS)**: filter incoming and outgoing network traffic.
- **Security Training and Awareness**: Educate employees and users about security best practices and potential threats
- **Secure Development Practices**: Follow secure coding guidelines and best practices during software development
- **Logging and Monitoring**: Implement comprehensive logging to track system activities and potential security incidents.
- **Incident Response Plan**: Develop and maintain an incident response plan to handle security incidents effectively.
- **Redundancy and Backup**: Establish redundancy for critical components to ensure system availability.
- **Security Policies and Procedures**: Develop and enforce security policies and procedures that align with industry standards and regulatory requirements.
- **Continuous Monitoring:** Implement continuous security monitoring to identify and respond to evolving threats.

DEMO

Q&A

References

- https://aws.amazon.com/blogs/security/how-to-approach-threat-modeling/
- https://github.com/awslabs/threat-composer
- https://www.iriusrisk.com/threat-modeling-methodologies
- Microsoft Threat Modeling Tool 2016

Thank you