
Threat
Modeling
AI Engineering
Recitation 10

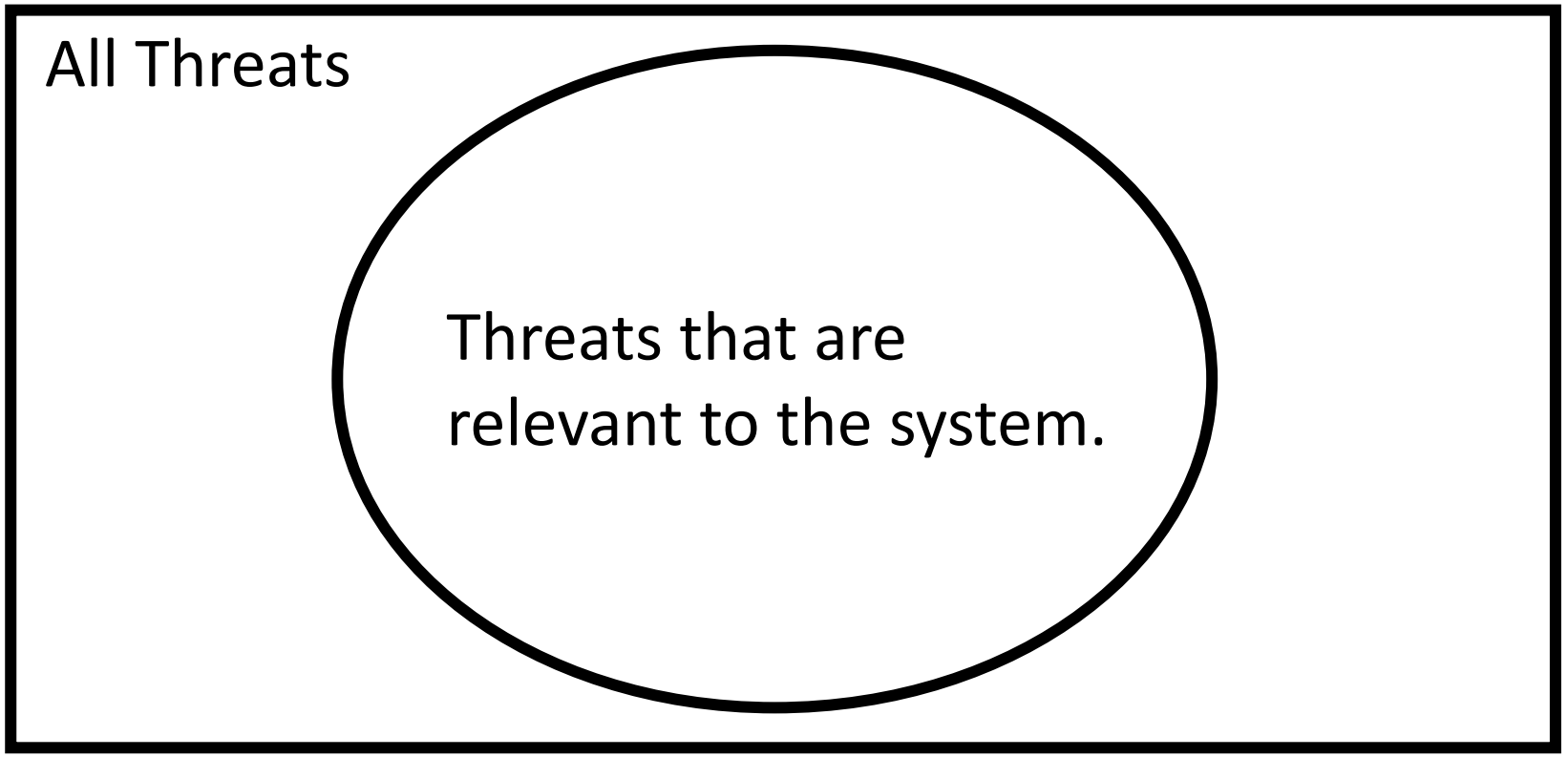
Why threat modelling?

- Almost any software system can come under attack.
- How can a development team ensure that their system is safe?
- Threat modelling is a structured way of doing this.

Threat Modelling in Context

- Policy: Set of security concerns you want to ensure (subset of requirements)
- Threat model: Assumptions about the adversary & how they could attach the system.
- Mechanism: Software/hardware/system that ensures the policy is followed so long as the adversary follows the threat model.

Threat Modelling involves identifying the threats that are relevant to a system.

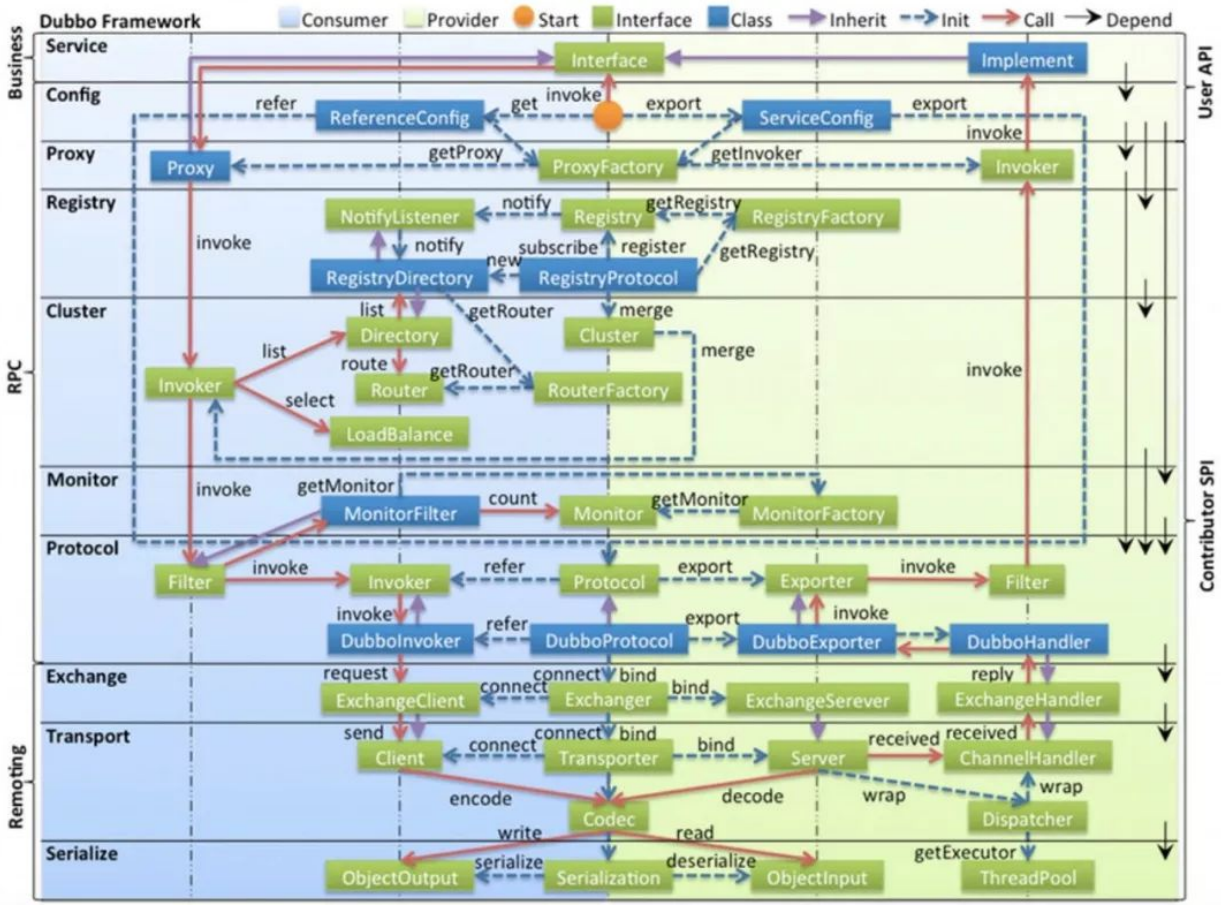


Threat Modelling (Partial) Example

- System: Online gradebook
- Policy: Only the Faculty and TAs assigned to a specific course should be able to edit the gradebook for that course.
- Threat Model: Students who know the web/IP address for the gradebook service's servers and can send an arbitrary stream of bytes at those servers
- Mechanism: The authentication/input processing subsystem of the online gradebook.

When in the software product development cycle should we do threat modelling?

For production-scale systems, threat modelling is a formal process.



There are many such processes.

- STRIDE
- PASTA
- LINDDUN
- CVSS
- Attack Trees
- And many more! See https://insights.sei.cmu.edu/blog/threat-modeling-12-available-methods/ for more information on even more threat modelling methods.

Introduction to STRIDE

- A threat modelling process developed by Microsoft
- Each letter in STRIDE represents a different method of attack.
  - Spoofing
  - Tampering
  - Repudiation
  - Information Disclosure
  - Denial of Service
  - Elevation of Privilege
- STRIDE is also associated with a systematic process that

Source: https://docs.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-threats

Spoofing

- Acting as a valid user (such as illegally accessing and then using another user’s username and password)
- Example: An employee is socially engineered into disclosing his account credentials to an unauthorized user.

Tampering

- Malicious modification of data
- Example: A hacker intercepts unprotected packets and changes the destination address.

Repudiation

- Malicious users deny performing an action they performed with the service provider unable to prove otherwise
- Example: A customer claims they did not receive a package that they did and demands compensation.

Information Disclosure

- Exposure of information to parties who are not supposed to have access to it
- Example: Leaked credit card data

Denial of Service

- Renders the service unavailable to valid users
- Example: A web server is overloaded with so many requests that it can’t handle them all.

Amazon says it mitigated the largest DDoS attack ever recorded

An attack with a previously unseen volume of 2.3 Tbps

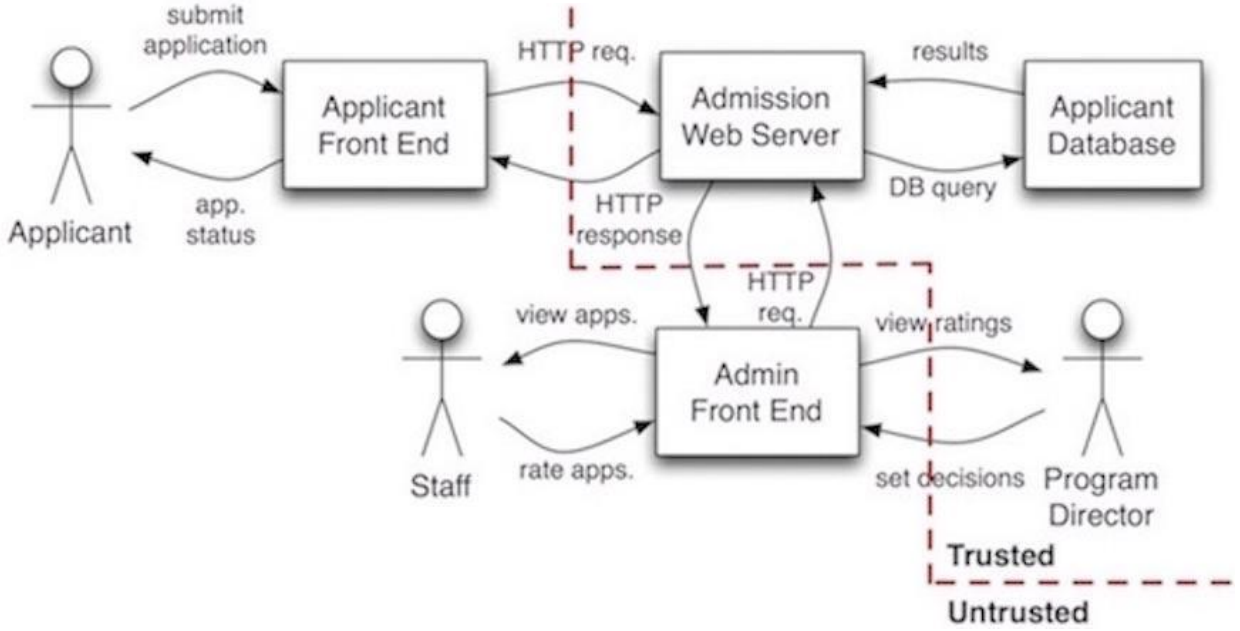
Elevation of Privilege

- Someone gains more power over the system than they are supposed to
- Example: an authentication bug allows a disgruntled employee to have root access on a company's servers

Hundreds of Millions of Dell Users at Risk from Kernel-Privilege Bugs

https://threatpost.com/dell-kernel-privilege-bugs/165843/

Using STRIDE - College admission



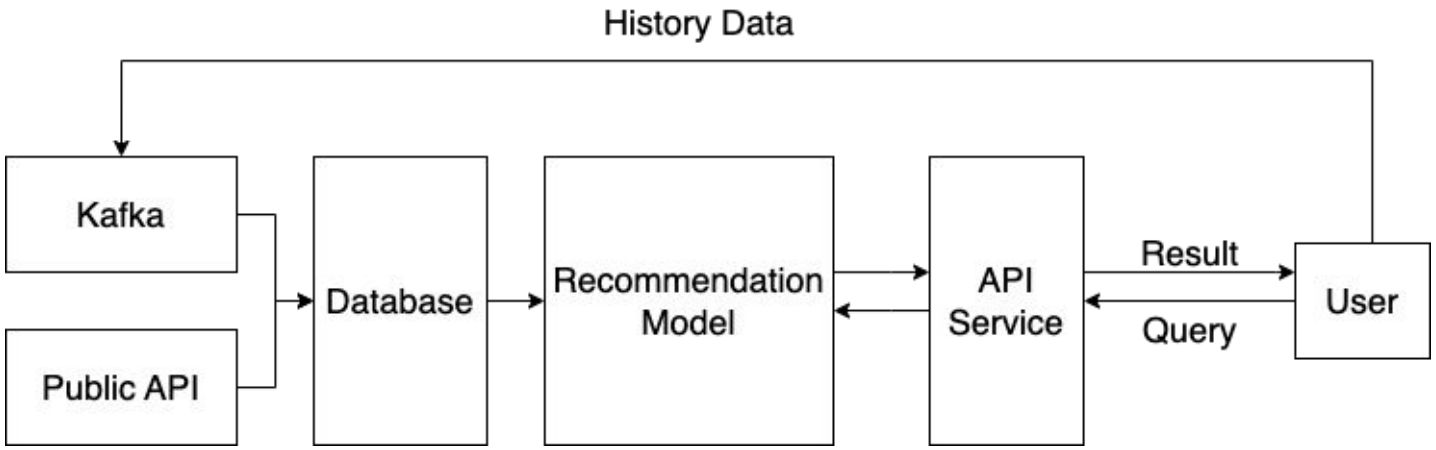
- Proofing: Attacker pretends to be another applicant by logging in
- Tampering: Attacker modifies applicant info using browser exploits
- Information disclosure: Attacker intercepts HTTP requests from/to server to read applicant info
- Denial of service: Attacker creates a large number of bogus accounts and overwhelms system with requests

STRIDE: Example Mitigations

- Spoofing: Attacker pretends to be another applicant by logging in
  - -> **Require stronger passwords**
- Tampering: Attacker modifies applicant info using browser exploits
  - -> **Add server-side security tokens**
- Information disclosure: Attacker intercepts HTTP requests from/to server to read applicant info
  - -> **Use encryption (HTTPS)**
- Denial of service: Attacker creates many bogus accounts and overwhelms system with requests
  - -> **Limit requests per IP address**

Group activities - Using STRIDE

- Using the system architecture, decompose your system into individual components.
- For each component, ask “How could an attacker spoof/tamper with/… this component”? and talk about mitigations



Security for ML-based systems

- Systems with machine learning components are vulnerable to unique attacks, often involving the data.
- Poisoning the training data:
  - If the training data is collected publicly and an attacker figures out the collection policy, they can create malicious training data which will influence the model.
  - Example: A language model is trained on data from social media sites. Attackers create many pages with fake data, which is then used to train the model.

Security for ML-based systems

- Eliciting training data from the model:
  - If the training data consists of confidential information, carefully crafted inputs can be used to extract training data from a model.
  - Example: An automated chatbot designed to help users with their accounts trained on past conversations with employees is used to reveal information.

Security for ML-based systems

- Adversarial inputs:
  - Inputs are modified such that a human wouldn’t change their decision about that input but the model’s decision does change
  - Example: a small amount of noise is added to a photograph so that facial recognition software labels the image as a different person

Security for ML-based systems

- Attacking the data is a way to attack an ML-based system
- Remember, the model serves as an “encoded” representation of the training data

  Some very sophisticated techniques exist for extracting information from ML models.

- This is in addition to all other methods of attack.

Security Strategies

Non-ML method

- Encryption
- Firewalls and VPNs
- User Account
- Permissions

ML Method

- Classifiers to learn malicious content: Spam filters, virus detection
- Anomaly detection: Identify unusual/suspicious activity, eg. credit card fraud, intrusion detection
- Game theory: Model attacker costs and reactions, design countermeasures