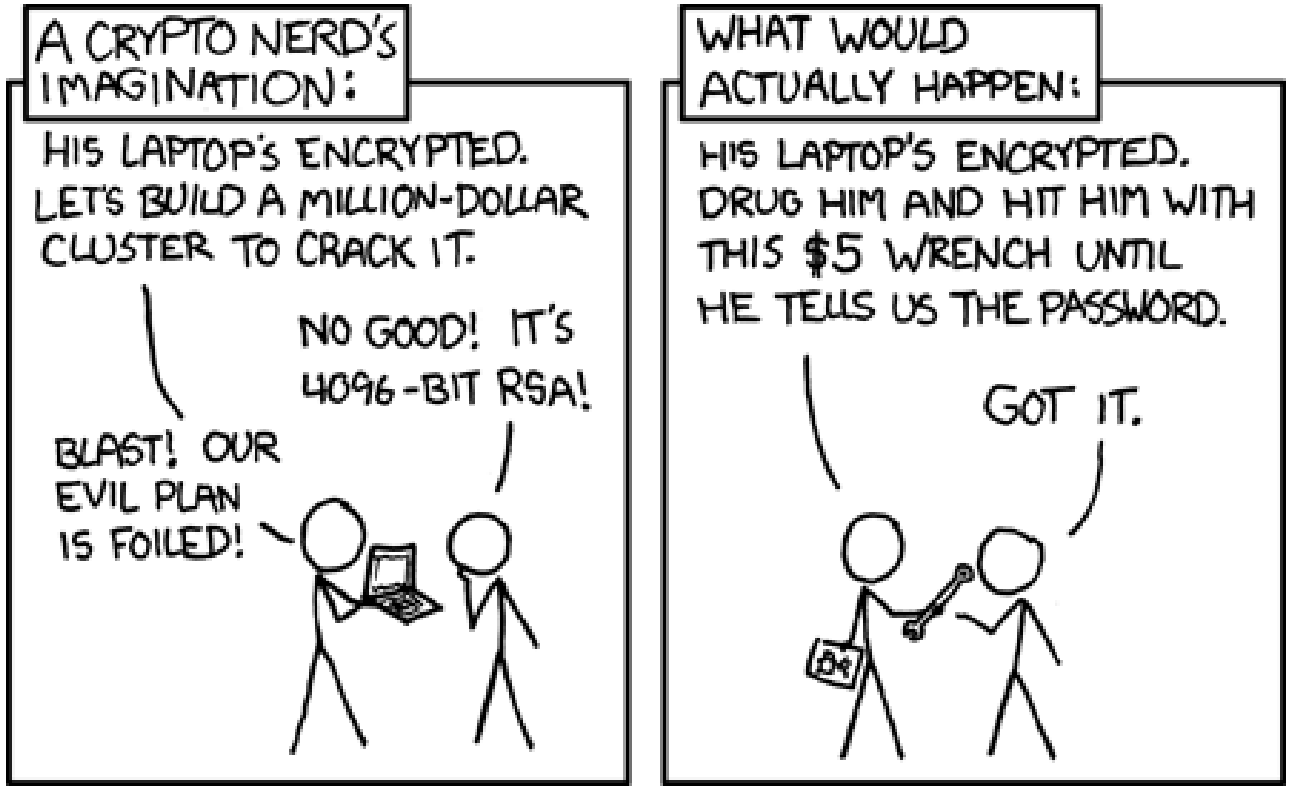
Threat Modelling

An Introduction by Kevin Denver


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.

What if?

Wouldn’t it be better to find security issues before you write a line of code?

Ways to find Security Issues

- Static Analysis
- Fuzzing
- Penetration Test
- Vulnerability Disclosures
- Bug Bounty Programe

..and Threat Modelling

- Think about security issues **early**!
- Understand your requirements better
- Prevent bugs before writing a single line of code

> Shift Left is a practice intended to find and prevent defects early in the software delivery process. devopedia: Shift Left

What is Threat Modelling?

Threat modelling works to **identify**, **communicate** and **understand threats and mitigations** within the context of protecting something of **value**.

A threat model is a **structured representation** of all the information that affects the security of an application.

In essence, it is a view of the application and its environment through the **lens of security**.

What does a Threat Model consist of?

A threat model typically includes:

- Description of the subject to be modelled
- Assumptions that can be checked or challenged in the future as the threat landscape changes
- Potential threats to the system
- Actions that can be taken to mitigate each threat
- A way of validating the model and threats, and verification of success of actions taken

When do you do a Threat Model?

Threat modelling is best applied **continuously** throughout a software development project!

Updating threat models is advisable after events such as:

- A new feature is released
- Security incident occurs
- Architectural or infrastructure changes

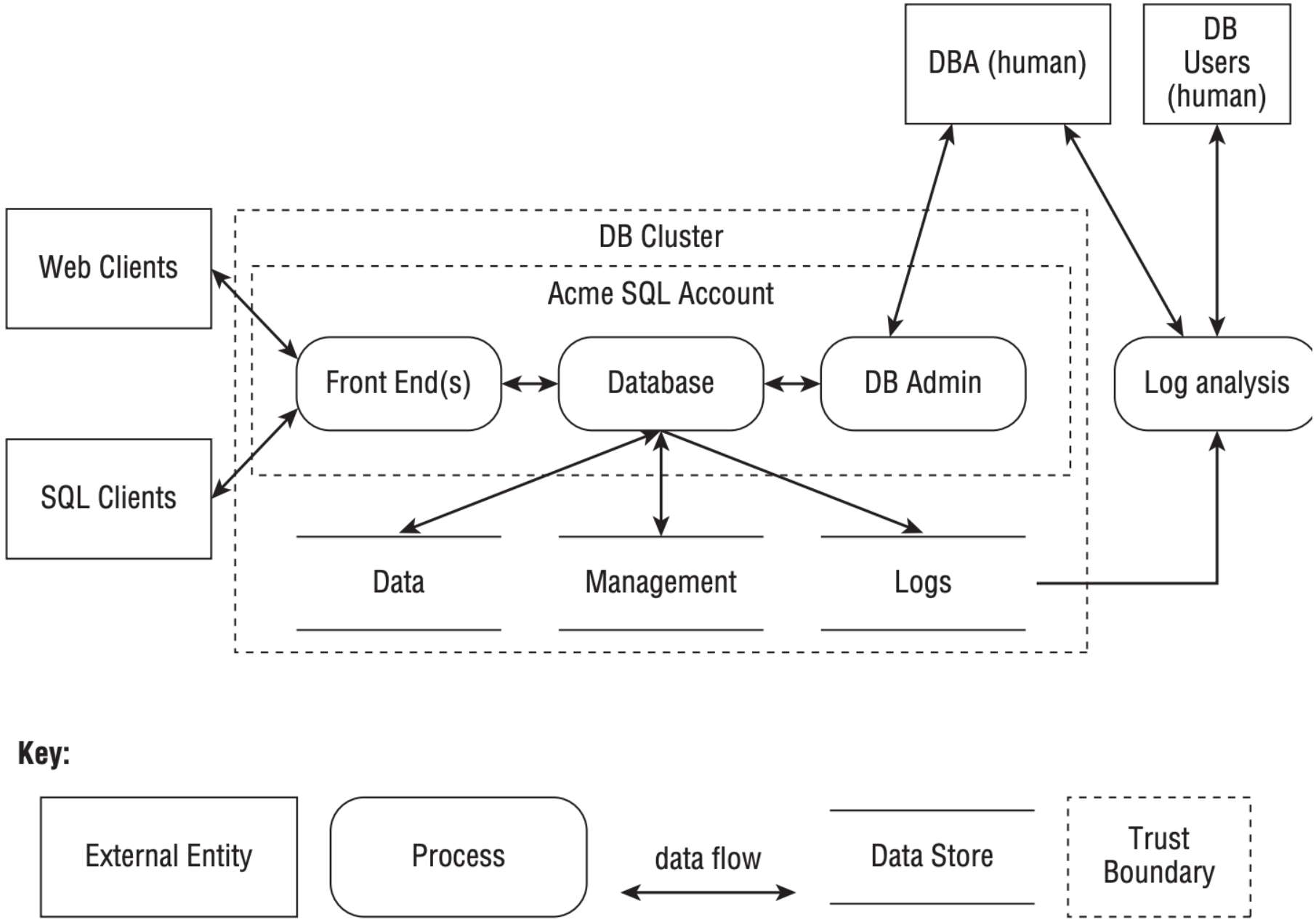
How to Threat Model

1. What are you building?
2. What can go wrong?
3. What are you going to do about it?

What are you building?

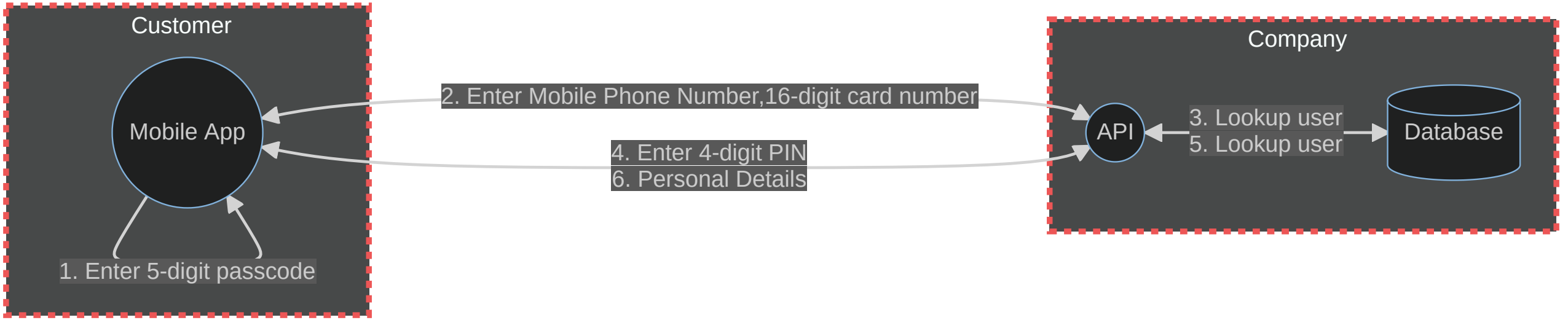
- Create a model of the system
  - Whiteboarding
  - Brainstorming
  - **Data Flow Diagrams** (preferred!)
- Focus on **Assets**: valuable things the business cares about
  - Something an attacker wants
  - Something you want to protect
  - A stepping stone?

> Use tools such as Mermaid or pytm rather than drawaing diagrams by hand


DBA (human)
DB Users (human)
DB Cluster
Web Clients
Acme SQL Account
Front End(s)
Database
DB Admin
Log analysis
SQL Clients
Data
Management
Logs
Key:
External Entity
Process
data flow
Data Store
Trust Boundary

What can go wrong?

- Fun to brainstorm
- Methodologies:
  - STRIDE (**S**poofing, **T**ampering, **R**epudiation, **I**nformation Disclosure, **D**enial of Service, **E**levation of Privileges)
    - Consider how each STRIDE threat could impact each part of the model!
    - MITRE Att&ck and D3fend can help if you get stuck
  - Attack Trees
- Structure helps get you to **completeness** and **predictability**
  - Need an engineering approach: Predictable, Reliable, Scalable
  - Can't be dependent on one brilliant person


Customer
Mobile App
1. Enter 5-digit passcode
2. Enter Mobile Phone Number,16-digit card number
4. Enter 4-digit PIN
6. Personal Details
Company
API
3. Lookup user
5. Lookup user
Database

References

- Lecture on Threat Modeling with STRIDE
- OWASP Threat Modelling
- Threat Modelling Cheat Sheet
- Threat Modelling Cookbook
- pytm: A Pythonic framework for threat modeling
- mermaid: Mermaid lets you create diagrams and visualisations using text and code
- MITRE ATT&CK: A knowledge base of adversary tactics and techniques
- MITRE D3FEND: A knowledge graph of cybersecurity countermeasures
- The STRIDE Threat Model