Engineering Secure Software

THREAT MODELING

Uses of Risk Thus Far

- Start with the ***functionality***
  - Use cases → abuse/misuse cases
  - p(exploit), p(vulnerability)
- Start with ***what to protect***
  - Goals → High-level Risks → Indicators → Tests
  - Assets
  - Domain, domain, domain
- Today: start with ***threats***

STRIDE

- Spoofing — `I am Spartacus.`
- Tampering — `Looks like Johnny got an A!`
- Repudiation — `Didn't Johnny have a B?`
- Information disclosure — `Johnny's SSN is…`
- Denial of Service — `Please try again later.`
- Elevation of privilege — `sudo rm –rf /home/johnny`

STRIDE ~> Security Properties

- *Kind of* the inverse of security properties, but not fully
  - **T**ampering → Integrity violation
  - **R**epudiation → Integrity of the *history* violation
  - **I**nformation Disclosure → Confidentiality violation
  - **D**enial of service → Availability violation
- Spoofing
  - Violating *authentication*
  - You are not who you say you are
    (e.g. session hijacking, guessing passwords)
- Elevation of privilege
  - Violating *authorization*
  - You can access things you should not be allowed to access
    (e.g. permissions, network access)

Repudiation

- A threat to the ***belief*** that integrity was preserved
  `Didn't Johnny have a B?`
- Provenance
  - Logs
  - Hash (digest) algorithms
  - Third-party verification
  - e.g. artwork, copyright registration
- Ultimately, another type of integrity violation
  - …but not exactly a tampering threat
  - Protect against tampering? Filter access, etc.
  - Protect against repudiation? Keep a reliable history

Architectural Risk Analysis

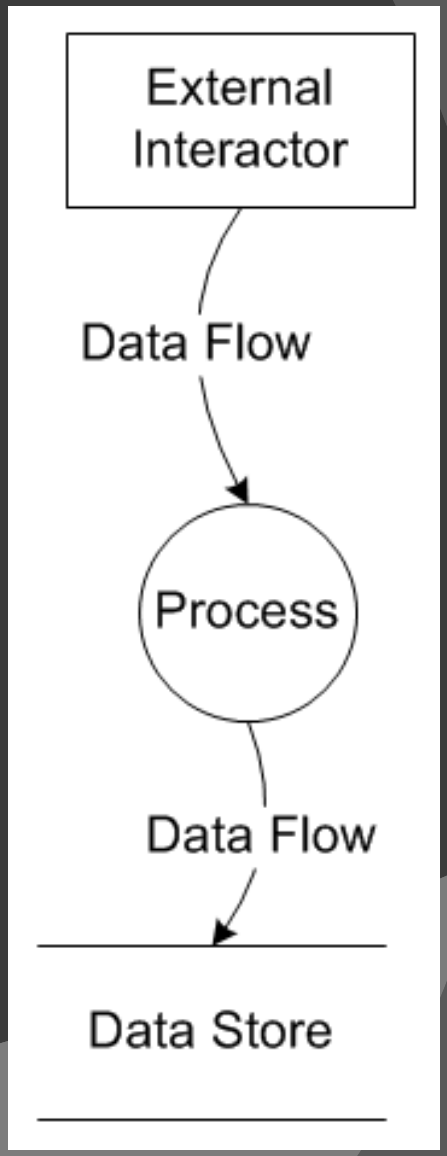
- Discuss security risk once most of the architecture is settled
- Motivation: a few good early decisions goes a long way
  - e.g. incorporating encryption
  - e.g. authentication & access control concerns
  - e.g. choice of technologies used
- Must-haves vs. Nice-to-haves at the design level
- Emphasis of design flaws over code-level vulnerabilities
- Note: “Risk Analysis” is not necessarily “Modeling”

Threat Modeling

- Architectural risk analysis tool
  - Built at Microsoft, on top of Visio
  - STRIDE concept
- Methodology:
  - Define architecture elements
    - Processes
    - External interactors
    - Data store
  - Connect with data flows
  - Define trust & machine boundaries
  - Map STRIDE to each element & relationship

Primitives

- External interactors
  e.g. clients, other systems, dependencies
- Process
  Architecture-centered functionality
  e.g. dispatcher, input validator
- Data store
  e.g. database, file system
- Data flow
  Domain & Design-specific explanation of data
  e.g. “HTTP Login Requests”



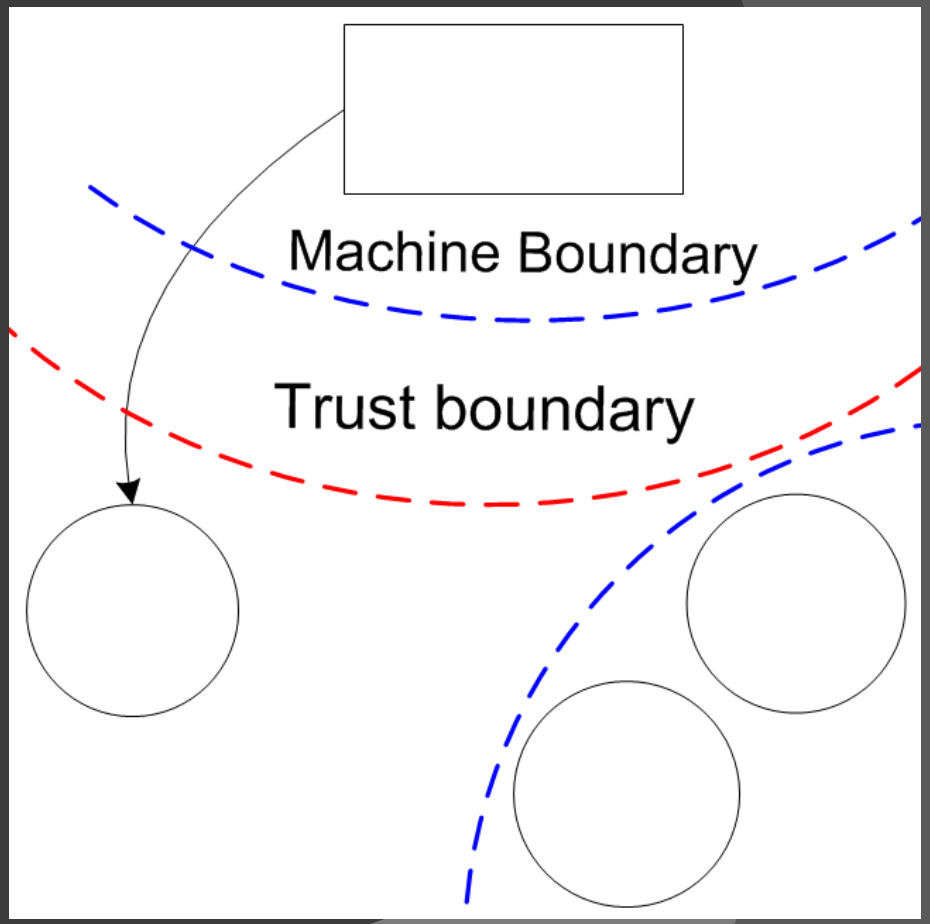
Boundaries

- Machine boundaries

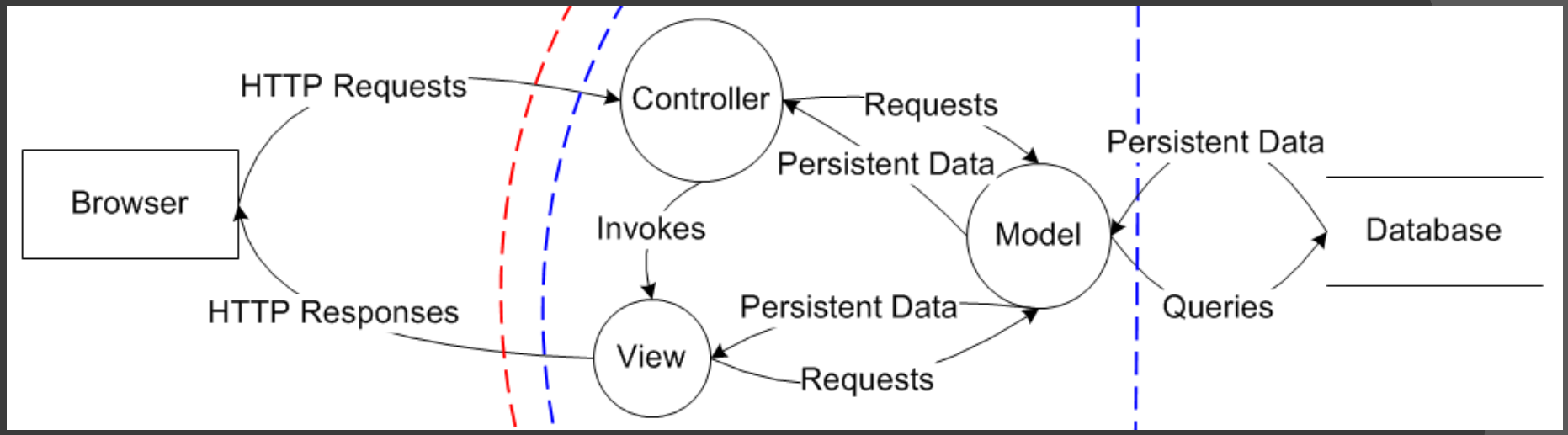
  Same physical machine

- Trust boundaries

  If the input can be trusted



e.g. Generic Webapp with MVC

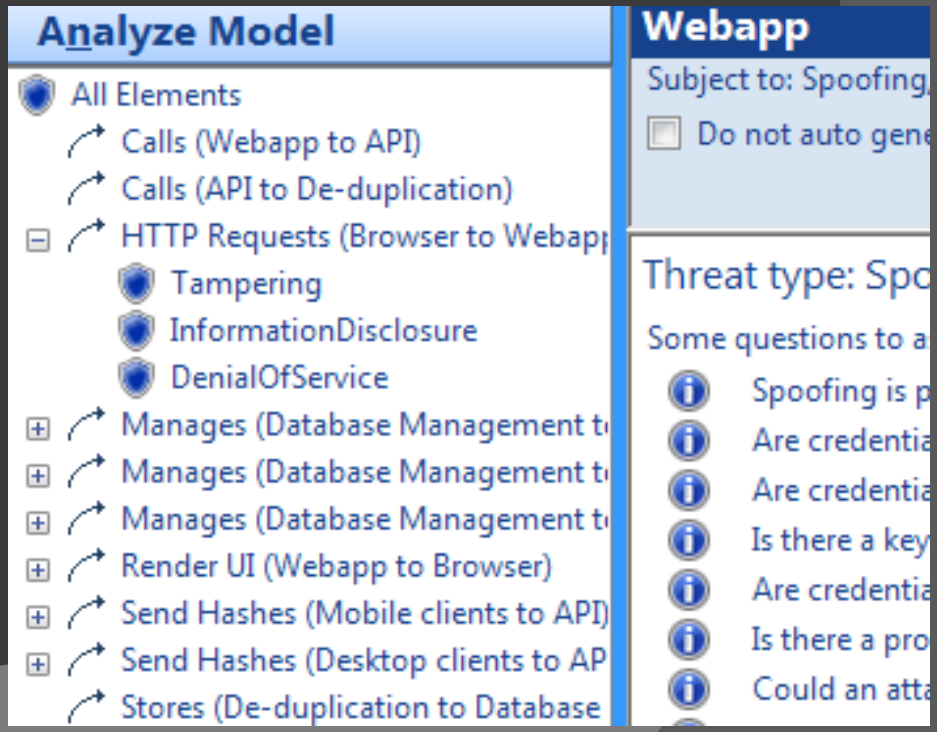


- Note: example is typically more domain-specific

- e.g. How will we prevent **spoofing** of the **browser**?
- e.g. How will we prevent **tampering** of **queries**?
- e.g. How do we avoid p**ersistent data from the DB** being **disclosed**?
- E.g. How will we avoid **repudiation** of the **database**?

Analysis

- What the tool does…
  - Eliminates categories of threats
  - Forces you describe mitigations
  - Helps record assumptions
  - Go directly to file a bug
- Threats arise when…
  - Flows cross boundaries
  - More processes
  - Forgetting what to trust



Tips for Threat Modeling

- Be honest with the process
  - Make sure the model represents reality (or what you really believe reality will be)
  - Consider *all* types of threats – code-level vulns are just a "for example"
- As with all modeling, use appropriate complexity
  - Overly-simplified?
    - Departs from reality
    - You get exactly what you put into it – no new knowledge
  - Overly-complicated?
    - Too much to analyze
    - "Check it off the list" syndrome
- Test your model

  Think of a specific security concern, then try to see where it fits in your threat model

Today's Activity

- Groups of 2-3
  - Go through the Threat Modeling activity
  - Tool: c:\Program Files (x86)\Microsoft\SDL Threat Modeling\SDLTM.exe
- For attendance, find me to check you off for today before leaving