Threat Modeling
(Architectural Risk Analysis)

Threat Model

- The **threat model** **makes explicit the adversary's assumed powers**
  - Consequence: The threat model must match reality, otherwise the risk analysis of the system will be wrong

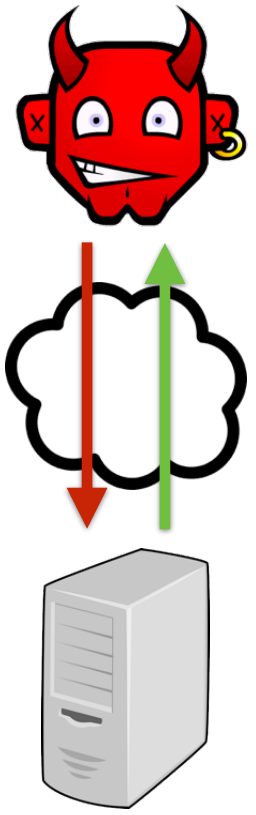
- The threat model is **critically important**
  - If you are not explicit about what the attacker can do, how can you assess whether your design will repel that attacker?

- This is part of **architectural risk analysis**

Example: **Network User**

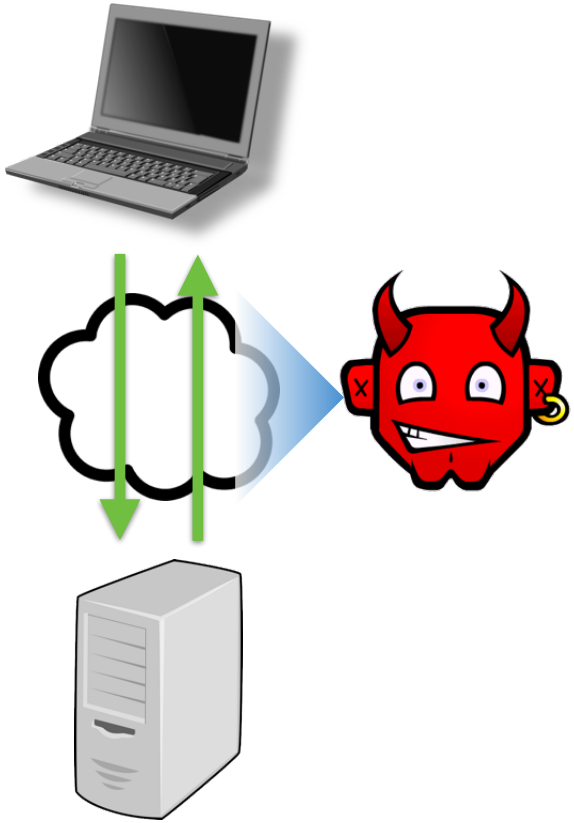
- An (anonymous) user that can connect to a service via the network
- Can:
  - **measure** the size and timing of requests and responses
  - run **parallel sessions**
  - provide **malformed inputs**, **malformed messages**
  - **drop or send extra messages**

- **Example attacks**: SQL injection, XSS, CSRF, buffer overrun/ROP payloads, …



Example: **Snooping User**

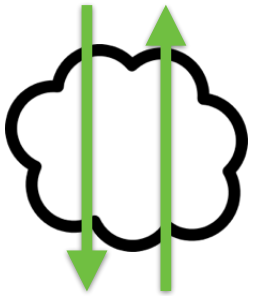
- Internet user **on the same network** as other users of some service
  - For example, someone connected to an unencrypted Wi-Fi network at a coffee shop
- Thus, can additionally
  - **Read/measure** others' **messages**,
  - **Intercept, duplicate,** and **modify messages**
- **Example attacks**: **Session hijacking** (and other data theft), **privacy-violating side-channel attack**, **denial of service**



Example: **Co-located User**

- Internet user **on the same machine** as other users of some service
  - E.g., **malware** installed on a user's laptop

- Thus, can additionally
  - **Read/write** user's **files** (e.g., cookies) and **memory**
  - **Snoop keypresses** and other events
  - Read/write the user's **display** (e.g., to **spoof**)



- **Example attacks**: **Password theft** (and other credentials/secrets)

Threat-driven Design

- Different threat models will elicit different responses

- **Network-only attackers** implies **message** traffic **is safe**
  - No need to encrypt communications
  - This is what `telnet` remote login software assumed

- **Snooping attackers** means **message** traffic **is visible**
  - So use encrypted wifi (link layer), encrypted network layer (IPsec), or encrypted application layer (SSL)
    - Which is most appropriate for your system?

- **Co-located attacker** can **access local files, memory**
  - Cannot store unencrypted secrets, like passwords

Bad Model = Bad Security

- Any **assumptions** you make in your model are potential **holes that the adversary can exploit**

- E.g.: Assuming no snooping users **no longer valid**
  - *Prevalence of wi-fi networks in most deployments*
- Other mistaken assumptions
  - **Assumption**: Encrypted traffic carries no information
    - Not true! By analyzing the size and distribution of messages, you can infer application state
  - **Assumption**: Timing channels carry little information
    - Not true! Timing measurements of previous RSA implementations could be used eventually reveal a remote SSL secret key

Finding a good model

- **Compare against similar systems**
  - What attacks does their design contend with?
- **Understand past attacks and attack patterns**
  - How do they apply to your system?
- **Challenge assumptions in your design**
  - What happens if an assumption is untrue?
    - What would a breach potentially cost you?
  - How hard would it be to get rid of an assumption, allowing for a stronger adversary?
    - What would that development cost?