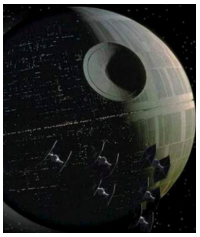
CSci 4271W
Development of Secure Software Systems
Day 4: More threats, and mitigation

Stephen McCamant (he/him)
University of Minnesota, Computer Science & Engineering

Based in large part on slides originally by Prof. Nick Hopper

Threat modeling

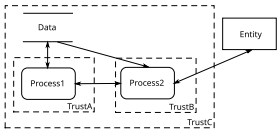


Star Wars TM and (C) Lucasfilm, Ltd.

- What are we building?
- What could go wrong?
- What are you doing about it?
- How did you do?

What could go wrong

- A good way to start thinking about the security of a system is to by describing how it works.



- Flows that cross trust boundaries are a good place to think about what could go wrong…

What could go wrong?

- S.poofing
- T.ampering
- R.epudiation
- I.nformation Disclosure
- D.enial of Service
- E.levation of Privilege

Beyond STRIDE

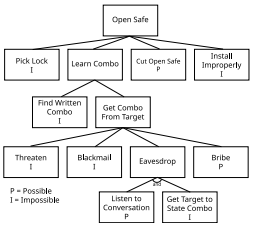
- STRIDE is useful as a starting point. Other useful ways to think about "what could go wrong":
  - Attack trees
  - Attacker profiles
  - Attack libraries
- These often focus on the goal of the attacker rather than the goal of the developer/operator.

Attack trees

1. Create an attack tree
   a. Find a goal
      i. Steal from textbook
         1. Reach chapter 4
         2. Look at examples
      ii. Steal from Internet
   b. Find subgoals
   c. Draw on slide…

Attack trees

1. Create an attack tree
   a. Find a goal
      i. Steal from textbook
         1. Reach chapter 4
         2. Look at examples
      ii. Steal from Internet
   b. Find subgoals
   c. Draw on slide…



https://www.schneier.com/academic/archives/1999/12/attack_trees.html

Attacker profiles

- Spy
- Terrorist
- Thief
- Vandal
- Insider
- …

See also: Shostack appendix C

Outline

More threat modeling perspectives

Announcements break

More threat modeling perspectives, cont'd

Revisiting diagram examples

Homework 1

- Now open for submission on Gradescope (linked from Canvas)
- Due Tuesday 2/4, by 11:59pm
- May do in groups of up to 3 students
- Be careful of the following on Gradescpe:
  - Include the names of your other group members
  - Provide the right range of pages for each answer

Outline

More threat modeling perspectives

Announcements break

More threat modeling perspectives, cont'd

Revisiting diagram examples

Attack libraries

- Knowing different kinds of attacks can also help with the question "what can go wrong?"
- Examples:
  - CAPEC (`https://capec.mitre.org/`)
  - ATT&CK (`https://attack.mitre.org/`)
  - OWASP Top 10 (`https://owasp.org/www-project-top-ten/`)

Mechanism-based

Another way to categorize possible attacks is by the mechanism they use:

- Misconfiguration
- Incomplete validation
- Memory corruption (all about this next week)
- Interpreters
- Social engineering (last unit of course)

Misconfiguration

Are there settings that should prevent an attack but don't?

- Default passwords
- Unnecessary network services (Telnet, SMTP, chargen, finger)
- Incorrect access-control settings (world read/writeable logs, open password files…)

Incomplete validation

Inputs that cross trust boundaries should be validated for purpose. Some pitfalls:

- Allowlist vs. blocklist (deny list)
- TOCTOU
- Non-canonicalization (directory traversal, DNS name vs. IP address)

Interpreters

Inputs that can have code:

- Javascript (in HTML, PDFs, emails, …)
- Macros (e.g. in MS Office documents)
- Anything passed to command shell, SQL, shell script…
- JSON, XML, YAML, and object serialization formats…
- Format strings (`printf("This is actually code.");`)
- Compressed files/strings (zip, xz, bzip2, …)

https://googleprojectzero.blogspot.com/2021/12/a-deep-dive-into-nso-zero-click.html

What to do about threats

- Mitigate: add a defense, which may not be complete
- Eliminate: such as by removing functionality
- Transfer functionality: let someone else handle it
- Transfer risk: convince another to bear the cost
- Accept risk: decide that the risk (probability · loss) is sufficiently low

Mitigations

What are we doing about it? How did we do?

- Spoofing: authentication (OS), crypto, canonicalization

Mitigations

What are we doing about it? How did we do?

- Spoofing: authentication (OS), crypto, canonicalization
- Tampering: OS controls (access control, isolation), crypto

Mitigations

What are we doing about it? How did we do?

- Spoofing: authentication (OS), crypto, canonicalization
- Tampering: OS controls (access control, isolation), crypto
- Repudiation: logging and audits

Mitigations

What are we doing about it? How did we do?

- Spoofing: authentication (OS), crypto, canonicalization
- Tampering: OS controls (access control, isolation), crypto
- Repudiation: logging and audits
- Information Disclosure: OS controls, crypto

Mitigations

What are we doing about it? How did we do?

- Spoofing: authentication (OS), crypto, canonicalization
- Tampering: OS controls (access control, isolation), crypto
- Repudiation: logging and audits
- Information Disclosure: OS controls, crypto
- Denial of Service: OS controls, rate limits/throttling

Mitigations

What are we doing about it? How did we do?

- Spoofing: authentication (OS), crypto, canonicalization
- Tampering: OS controls (access control, isolation), crypto
- Repudiation: logging and audits
- Information Disclosure: OS controls, crypto
- Denial of Service: OS controls, rate limits/throttling
- Elevation of Privilege: memory mitigation, OS controls, sandboxes/containers, input validation

Outline

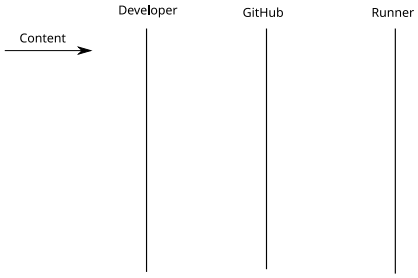
More threat modeling perspectives

Announcements break

More threat modeling perspectives, cont'd

Revisiting diagram examples

GitHub CI swim lanes



Example: LMS (Canvas, etc.)