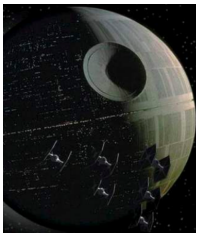
CSci 4271W
Development of Secure Software Systems
Day 3: "What Could Go Wrong?" with STRIDE

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

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

Threat modeling

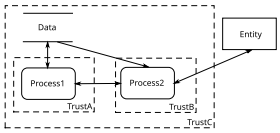


Star Wars TM and (C) Lucasfilm, Ltd.

- 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

Spoofing

- Pretending to be something/someone you're not.
- Examples:
  - Email / SMS spoofing (in phishing, etc.)
  - Network spoofing (more later...)
  - Password guessing and default passwords
  - Credential stuffing
- Victims: processes, external entities, people

Local spoofing

- Often involves file names or symbolic links.

```
if filename.startswith("/writedir/"):
  f = open(filename, "w")
  # ... do something with f ...
```

- Sometimes there is a Time of Check/Time of Use (TOCTOU) problem, or a race condition:

```
if not os.path.exists(tmpfile):
  f = open(tmpfile, "w")
  # link and write to attacker's favorite location
```

Tampering

- Modifying data in memory, on disk, over the network
- Examples:
  - Unprotected files
  - Ad replacement
  - Longer-than-expected input writes over memory
- Victims: data stores, data flows, processes

Process tampering

Modifying data in memory:

```
void vulnerable() {
    char msg[512];
    char email[80];
    char name[16];
    strncat(msg,"Dear ", 6);
    strncat(msg, name, 16);
    /* add some other stuff to msg... */
    gets(email);
    /* do some other stuff... */ }
```

Process tampering (2)

Modifying data in memory:

```
void vulnerable() {
  int a[4];
  long i;
  cout << "Enter option to change: ";
  cin >> i;
  if (i < 4) {
    cout << "Enter new value: ";
    cin >> a[i];
  } }
```

Repudiation

- Claiming you didn't do something
- Examples:
  - "I didn't order that pizza!"
  - "My 4K TV never arrived!"
  - The fire alarm was not caused by a faulty sensor
- Victims: processes

Outline

STRIDE

Announcements break

STRIDE, cont'd

Revisiting diagram examples

Homework 1

- Available now on the public course web site
- Due a week from today, Tuesday 2/4, by 11:59pm
- Today's lecture is the last material for the homework
- May do in groups of up to 3 students
- Submission will be via Gradescope (not available yet)

Lab sections

- Monday mornings in Walter B28 (basement)
- Gives you hands-on experience using tools
- Important to come to the labs in person: benefit from staff and other students
- Partial-credit makeup available if you missed this week

Now available on the website

- Links to lab and homework instructions
- Schedule overview for rest of the semester

Outline

STRIDE

Announcements break

STRIDE, cont'd

Revisiting diagram examples

Information disclosure

- Providing data (or metadata) to unauthorized entities
- Examples:
  - Inadequate file protections
  - File names: `Li Wang severance.doc`
  - Unencrypted network traffic
  - Error handling: "unrecognized user"
  - Heartbleed bug
- Victims: processes, data stores, data flows

Denial of service

- Consuming resources needed to provide service
- Examples:
  - Network DoS: Morris worm, chargen, Mirai
  - Process DoS: persistent crash bug, e.g. Crowdstrike
  - CPU DoS: Hashtable collisions, Evil Regexes
  - User DoS: account freeze
  - Disk DoS: fill up the log
- Victims: processes, data stores, data flows

Elevation of privilege

- Allowing someone to execute code in an unauthorized context
- Examples:
  - Remote client allowed to execute code
  - Normal user allowed to execute code as `root`
  - User-space program executes code as kernel

Pass-thru elevation of privilege

Executing commands from inputs across a trust boundary

```
filename = input()
os.system("/usr/bin/nano " + filename)
# if filename is "; rm -rf ~"...

ord_num = input()
crsr.execute("SELECT * FROM orders where number = "+ord_num+";")
ans = crsr.fetchall()
for i in ans:
  # do stuff
```

Order number "`1; DROP TABLE billing`"...

Combinations

Some attacks could come in combination, e.g.,

- Spoofing default user…
- To tamper with access list…
- To elevate privileges…
- To repudiate log entry

Outline

STRIDE

Announcements break

STRIDE, cont'd

Revisiting diagram examples

Example: GitHub CI

GitHub CI swim lanes