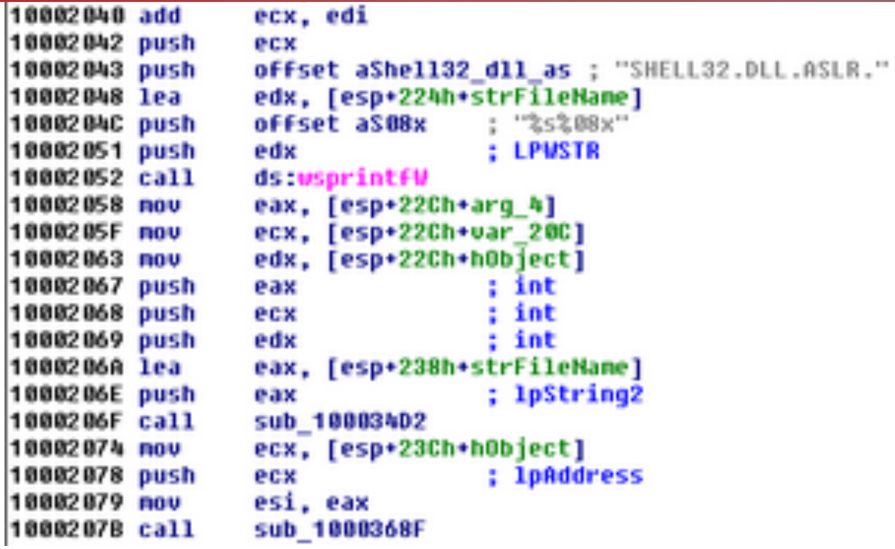
Lecture 1: Introduction to Security

CS 181S

Fall 2020

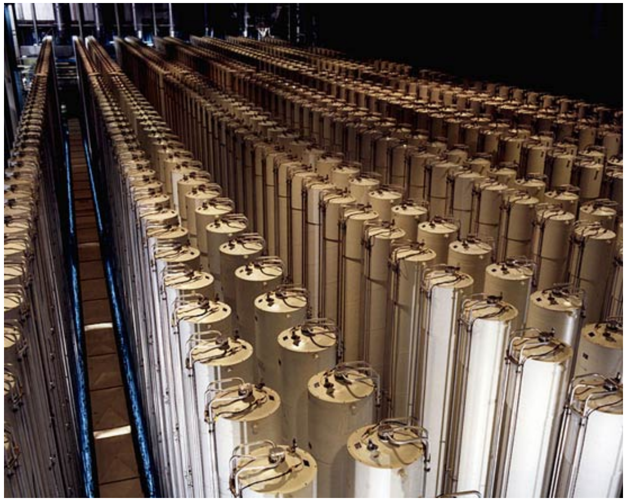
```
static report_breakin(arg1, arg2)                   /* 0x2494 */
{
    int s;
    struct sockaddr_in sin;
    char msg;

    if (7 != random() % 15)
          return;

    bzero(&sin, sizeof(sin));
    sin.sin_family = AF_INET;
    sin.sin_port = REPORT_PORT;
    sin.sin_addr.s_addr = inet_addr(XS("128.32.137.13"));
```

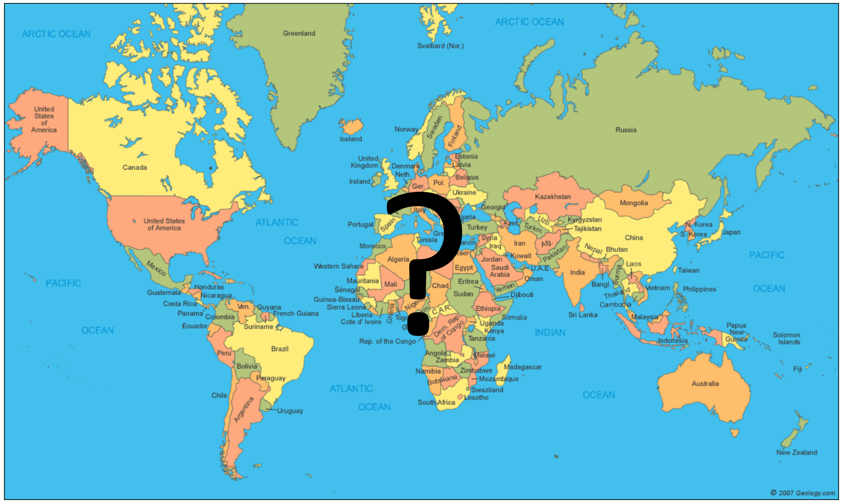
November 2, 1988



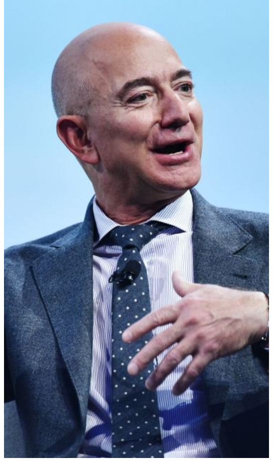
```
10002040 add     ecx, edi
10002042 push    ecx
10002043 push    offset aShell32_dll_as ; "SHELL32.DLL.ASLR."
10002048 lea     edx, [esp+224h+strFileName]
1000204C push    offset aS08x     ; "%s%08x"
10002051 push    edx              ; LPWSTR
10002052 call    ds:wsprintfW
10002058 mov     eax, [esp+22Ch+arg_4]
1000205F mov     ecx, [esp+22Ch+var_20C]
10002063 mov     edx, [esp+22Ch+hObject]
10002067 push    eax              ; int
10002068 push    ecx              ; int
10002069 push    edx              ; int
1000206A lea     eax, [esp+238h+strFileName]
1000206E push    eax              ; lpString2
1000206F call    sub_100034D2
10002074 mov     ecx, [esp+23Ch+hObject]
10002078 push    ecx              ; lpAddress
10002079 mov     esi, eax
1000207B call    sub_1000368F
```

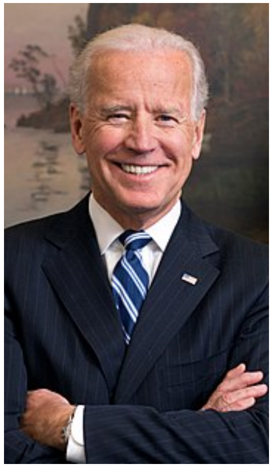
June 1, 2012





July 15, 2020





INTERESTING

HARD

Today

FUN

IMPORTANT

Defining security

"This tops the list of recommendations for upgrading your online security."

Functional Requirements

- **Security** = **does what it should** + nothing more
- "As a *user* I can *action* so that *purpose*"
  - e.g., As a professor, I can create a new assignment by specifying its name, number of possible points, and due date.
  - e.g., As a student, I can upload a file as a solution to an assignment.
  - e.g., As a professor, I can assign grades to student solutions.

Functional requirements should specify **what** not **how**

- Should be **testable**: a 3rd party could determine whether requirement is met
- These user stories reveal system **assets**

Security Goals

- **Security** = does what it should + **nothing more**
- "The system shall prevent/detect *action* on/to/with *asset*."
  - e.g., "The system shall prevent students from accessing assignments that are not theirs"
  - e.g., "The system shall prevent grades from being changed by anyone but the professor"

Security goals should specify **what** not **how**

- Poor goals:
  - "the system shall use encryption to prevent reading of messages"
  - "the system shall use authentication to verify user identities"
  - "the system shall resist attacks"
- If a system enforces a goal, it is called a **security property**

C I A

CIA

Confidentiality

Integrity

Availability

Confidentiality Properties

Protection of assets from unauthorized disclosure

i.e., which principals are allowed to learn what

Examples:

- Keep contents of a file from being read (*access control*: more later)
- Keep information secret (*information flow*: more later)
  - value of variable secret
  - behavior of system
  - information about individual

Privacy

Privacy concerns information about individuals (people, organizations, etc.)

- Often construed as legal right
- *Privacy* is not a synonym for confidentiality or for secrecy

Integrity Properties

Protection of assets from unauthorized modification

i.e., what changes are allowed to system and its environment, including inputs and outputs

Examples:

- Output is correct according to (mathematical) specification
- No exceptions thrown
- Only certain principals may write to a file (access control)
- Data are not corrupted or tainted by downloaded programs (information flow)

Availability Properties

Protection of assets from loss of use

i.e., what has to happen when/where

Examples:

- Operating system accepts inputs periodically
- Program produces output by specified time
- Requests are processed fairly (order, priority, etc.)

Denial of service (DoS) attacks compromise availability

Label each property as C/I/A

1. Students can always log into their accounts
2. The grade for an assignment is available only to the student who submitted that assignment.
3. The professor can see all submitted assignments and grades.
4. If your course grade changed, then the professor made that change.
5. If your course grade changed, you see the updated grade.
6. Requests to the grading server are processed in the order they were received.

Aspects of security

- **Confidentiality:** protection of assets from unauthorized disclosure
- **Integrity:** protection of assets from unauthorized modification
- **Availability:** protection of assets from loss of use

Ex 1

- **Attack:** John copies Mary's homework

- What is a **security goal** this attack would violate?

- Which **aspect** of security does that policy address?

Ex 2

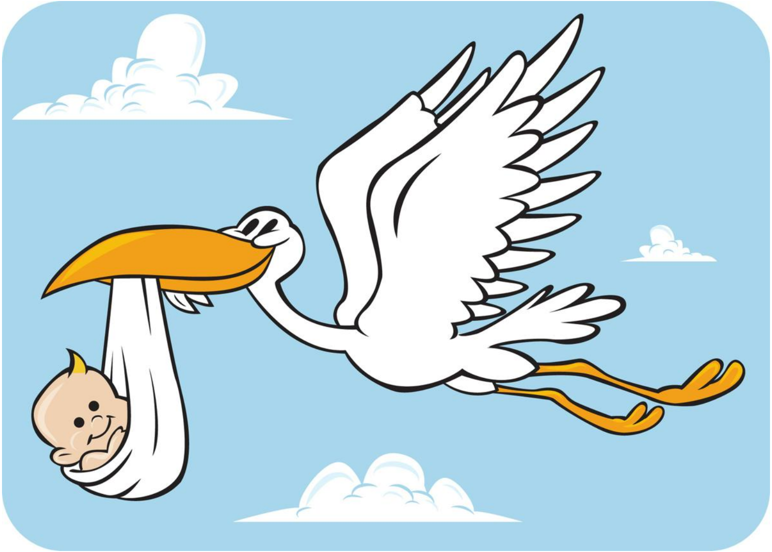
- **Attack:** Paul causes Linda's system to freeze
- **Goal?**
- **Aspect?**

EXERCISE: SECURITY GOALS

Stork Baby Delivery

The *stork baby delivery system* allows an autonomous aircraft (a *stork*) to deliver a payload (a *baby*) to a geographic location prespecified by some higher authority (*providence*). Prior to take-off, providence programs a stork with the geographic location describing where the baby should be delivered. Throughout the mission, the stork transmits back to providence a video of the landscape (labeled with geographic location coordinates) that the stork flies over. While a stork is in flight, providence may issue commands to that stork and change the location for the delivery, alter the path being followed to that location, or abort the mission.

Threat model: The adversary desires to prevent baby deliveries. The adversary has access to radio equipment that transmits and receives on the same frequencies that providence uses for communication with a stork. The adversary also controls weapons systems that can destroy a stork in flight.



The Bigger Picture

Attacks
are perpetrated by
threats
that inflict
harm
by exploiting
vulnerabilities
which are controlled by
countermeasures.

LOGISTICS

Course Logistics

Prof. Eleanor Birrell

Research in security and privacy
OH: M 2-4pm PT + TBA

- **Lecture Videos:**
  - 2 per week, allow ~75mins for each video
  - Published on EdPuzzle
  - Must be completed before class on Monday

- **Class Meetings:**
  - Monday and Wednesday, 12:45-2pm PT on Zoom
  - Attendance is required

Course Work

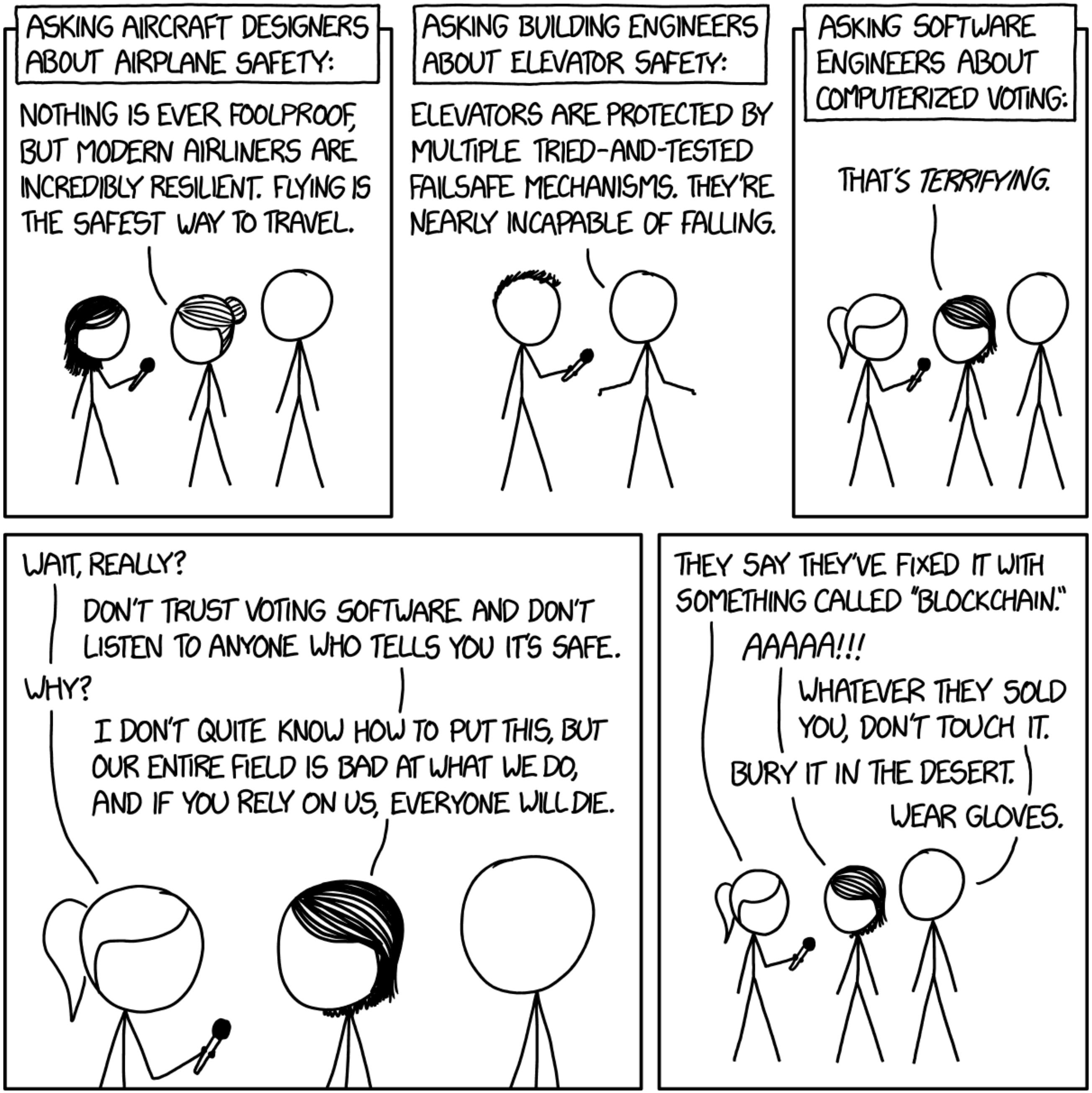
- 7-8 assignments (60%)
  - Mix of theory assignments and programming assignments
- Course project (30%)
  - Design and build a secure system
  - Done in groups of 3-4
- Participation (10%)
  - Watching video lectures + doing exercises on time
  - Attending and participating in synchronous classes
    - this requirement can be waived

- All assignments will be due Mondays at 11:59pm PT

Course website

http://www.cs.pomona.edu/classes/cs181s/2020fa/

- All information is on the course website
- Various reading materials: slides, notes, links to online readings, pointers to text book chapters
  - Optional? Yes. But...
    - the more of these you read, the more you will get out of the course
    - assignments are often inspired by this material
  - Lectures are the ground truth for material we cover


ASKING AIRCRAFT DESIGNERS ABOUT AIRPLANE SAFETY:
NOTHING IS EVER FOOLPROOF, BUT MODERN AIRLINERS ARE INCREDIBLY RESILIENT. FLYING IS THE SAFEST WAY TO TRAVEL.
ASKING BUILDING ENGINEERS ABOUT ELEVATOR SAFETY:
ELEVATORS ARE PROTECTED BY MULTIPLE TRIED-AND-TESTED FAILSAFE MECHANISMS. THEY'RE NEARLY INCAPABLE OF FALLING.
ASKING SOFTWARE ENGINEERS ABOUT COMPUTERIZED VOTING:
THAT'S *TERRIFYING.*
WAIT, REALLY?
DON'T TRUST VOTING SOFTWARE AND DON'T LISTEN TO ANYONE WHO TELLS YOU IT'S SAFE.
WHY?
I DON'T QUITE KNOW HOW TO PUT THIS, BUT OUR ENTIRE FIELD IS BAD AT WHAT WE DO, AND IF YOU RELY ON US, EVERYONE WILL DIE.
THEY SAY THEY'VE FIXED IT WITH SOMETHING CALLED "BLOCKCHAIN."
AAAAA!!!
WHATEVER THEY SOLD YOU, DON'T TOUCH IT.
BURY IT IN THE DESERT.
WEAR GLOVES.