



PennState

CSE 443: Introduction to Computer Security

Module: Security Basics

Prof. Syed Rafiul Hussain
Department of Computer Science and Engineering
The Pennsylvania State University

Acknowledgements: Some of the slides have been adopted from
Trent Jaeger (Penn State), Patrick McDaniel (Penn State), Ninghui Li (Purdue) and William Enck (NCSU)

What is security?

What does it mean for a system to be secure?
Write your own definition?

- Garfinkel and Spafford (1991)
 - ▶ “A computer is secure if you can depend on it and its software **to behave as expected.**”
- Harrison, Ruzzo, Ullman (1978)
 - ▶ “Prevent access by **unauthorized users**”
- Not really satisfactory – does not truly capture that security speaks to the behavior of others
 - ▶ Expected by whom?
 - ▶ Under what circumstances?
 - ▶ What are the **risks**?



A Meta definition

A system is secure if it can maintain well-specified properties in spite of the actions of well-specified adversaries.

- The set of properties we assume to be correct is called the **trust model**
- The set of adversaries (and their capability) is called the **threat model**
- Trust Model + Threat Model = Security Model
- The art and science of secure systems lies in properly identifying these properties, adversaries, and designing mechanisms that achieve this goal.

- **At-risk** valued resources that can be misused
 - ▶ Monetary
 - ▶ Data (loss or integrity)
 - ▶ Time
 - ▶ Confidence
 - ▶ Trust
- **What does being misused mean?**
 - ▶ Confidentiality
 - ▶ Integrity
 - ▶ Availability
 - ▶ Privacy (personal)
- **Q:What is at stake in your life?**



Security Goals/Properties (C, I, A)



- **Confidentiality (secrecy, privacy)**
 - ▶ only those who are authorized to know can know
- **Integrity (also authenticity in communication)**
 - ▶ only modified by authorized parties and in permitted ways
 - ▶ do things that are expected
- **Availability**
 - ▶ those authorized to access can get access

- Prevent “unauthorized disclosure of information” (Stallings)
- Examples:
 - ▶ Keep Alice from reading Bob’s files without permission
 - ▶ Keep Bob from knowing Alice has a file called ILoveBob.txt
 - ▶ Prevent Eve from reading Alice’s network traffic
 - ▶ Prevent Steve from knowing whether Alice is a patient at a clinic

- Prevent unauthorized modification of data (also config, code!)
- Examples:
 - ▶ Keep Alice from changing Bob's files without permission
 - ▶ Keep Bob from deleting Alice's file
 - ▶ Prevent Mallory from modifying Alice's network traffic
 - ▶ <http://www.goat-simulator.com> -> <http://www.nastygoatsite.com>
 - ▶ Prevent Bob from changing an important system binary
 - ▶ ls -> sl

- Prevent “disruption of access to or use of information or information system” (Stallings)
- Examples:
 - ▶ Keep Bob from deleting Alice’s files
 - ▶ Prevent Mallory from crashing `csc.ncsu.edu`
 - ▶ Prevent Dave from flooding Bob’s computer with network requests

Assets: What we protect

- Assets are the items that we are trying to protect
 - ▶ Hardware Resources
 - ▶ Network Access
 - ▶ Operating Systems
 - ▶ Software
 - ▶ Data
 - ▶ Users
 - ▶ User Time
 - ▶ Money managed by system
 - ▶ Reputation

- **At-risk** valued resources that can be misused
 - ▶ Monetary
 - ▶ Data (loss or integrity)
 - ▶ Time
 - ▶ Confidence
 - ▶ Trust
- **What does being misused mean?**
 - ▶ Confidentiality
 - ▶ Integrity
 - ▶ Availability
 - ▶ Privacy (personal)
- **Q:What is at stake in your life?**



- **Principals** are expected system subjects
 - ▶ Computers, agents, people, enterprises, ...
 - ▶ Depending on context referred to as: servers, clients, users, entities, hosts, routers, ... - and some may be adversarial
 - ▶ Security is defined with respect to these subjects
 - Implication: every principal may have unique view
- **A trusted third party**
 - ▶ Trusted by all principals for some set of actions
 - ▶ Often used as introducer or arbiter



- An **adversary** is any entity trying to circumvent the security infrastructure
 - ▶ The curious and otherwise generally clueless (e.g., script-kiddies)
 - ▶ Casual attackers seeking to understand systems
 - ▶ Venal people with an axe to grind
 - ▶ Malicious groups of largely sophisticated users (e.g, chaos clubs)
 - ▶ Competitors (industrial espionage)
 - ▶ Governments (seeking to monitor activities)



Are users adversaries?

- Have you ever tried to circumvent the security of a system you were authorized to access?
- Have you ever violated a security policy (knowingly or through carelessness)?

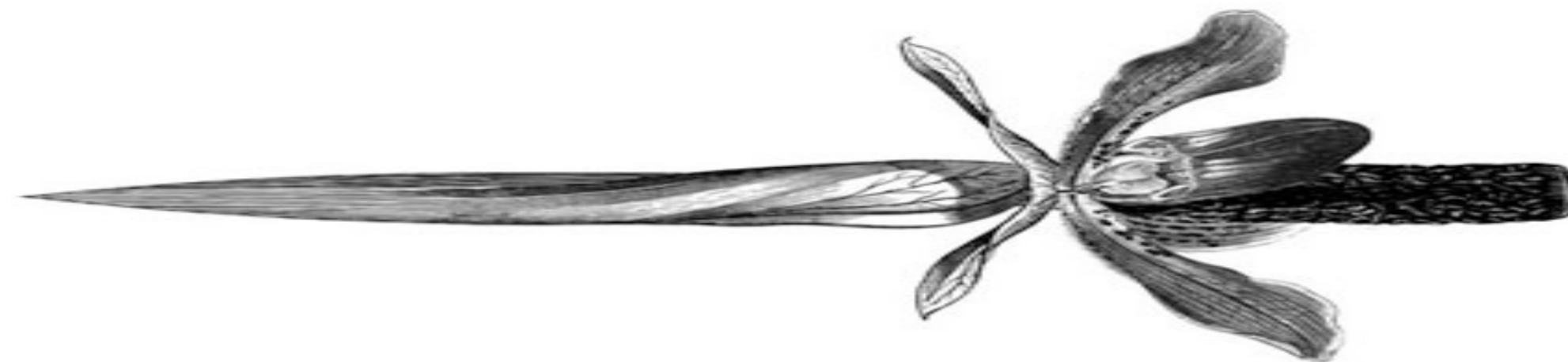
This is known as the insider adversary!

Thinking Like an Adversary

- Computer security experts think like an attacker all the time
 - ▶ “What can go wrong?”
 - ▶ “How can it go wrong?”
 - ▶ “What assumptions might not be correct?”
 - ▶ “How can I exploit the system?”



- A **threat** is a specific means by which an adversary can put a system at risk
 - ▶ An ability/goal of an adversary (e.g., eavesdrop, fraud, access denial)
 - ▶ Independent of what can be compromised
- A **threat model** is a collection of threats that deemed important for a particular environment
 - ▶ A collection of adversary(ies) abilities
 - ▶ E.g., a powerful adversary can read and modify all communications and generate messages on a communication channel
- Q: What were risks/threats in the introductory examples?
 - ▶ Slammer
 - ▶ Yale/Princeton
 - ▶ Estonia

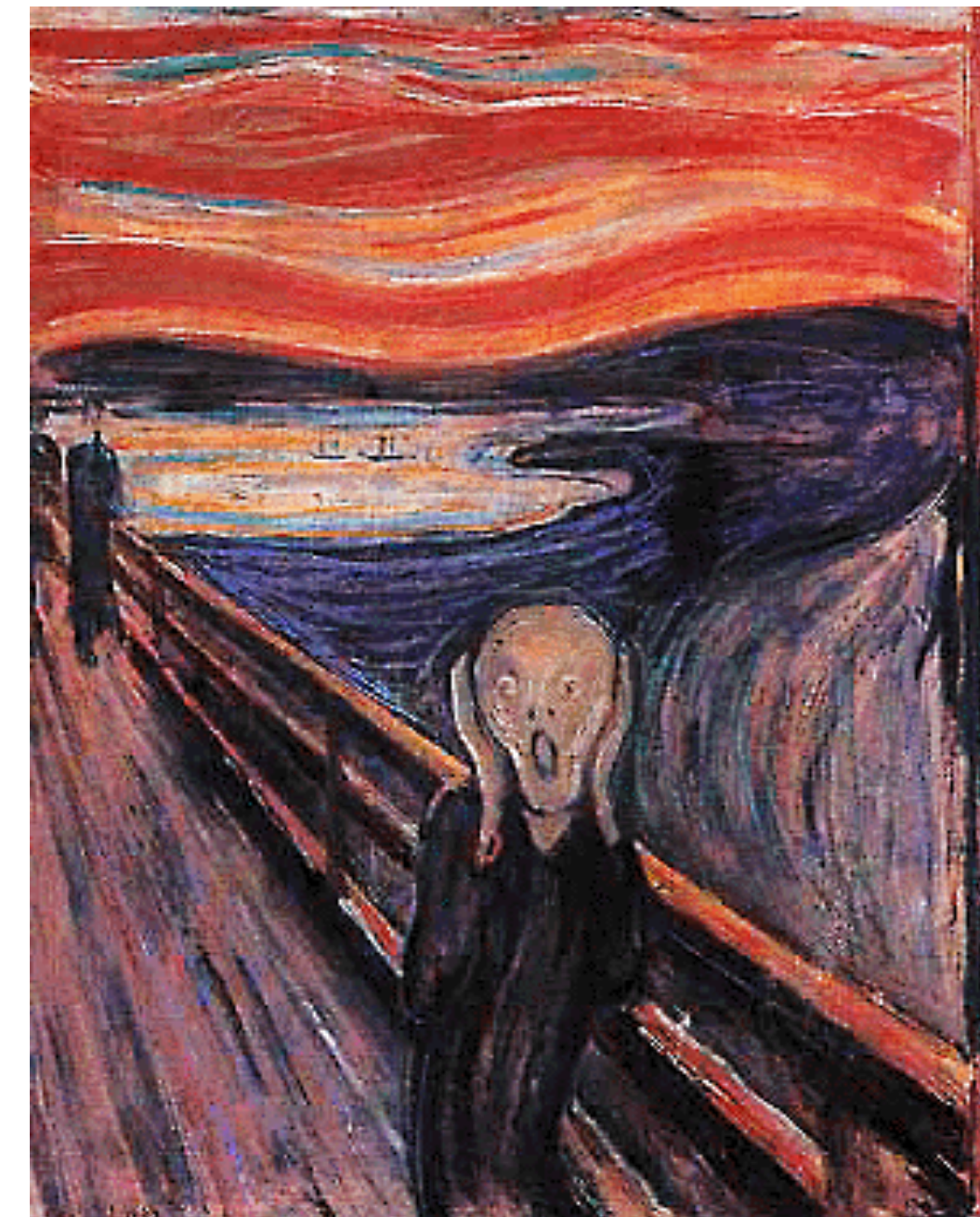


Threat Modeling Approaches

- **Diagram-driven**
 - ▶ Architectural diagram
 - ▶ Data flow diagram
 - ▶ User workflow
 - ▶ Ask: what could go wrong?
- **Attack Tree**
 - ▶ Attacker goal at top
 - ▶ Branches are ways to get to the goal
- **Checklists**
 - ▶ From past experiences
- **STRIDE**
 - ▶ Spoofing
 - ▶ Tampering
 - ▶ Repudiation
 - ▶ Information disclosure
 - ▶ Denial of service
 - ▶ Escalation of privilege

Vulnerabilities (attack vectors)

- A vulnerability is a **flaw** that is **accessible to an adversary** who can **exploit that flaw**
- E.g., buffer overflow, file open w/ adversary name
- What is the source of a vulnerability?
 - ▶ Bad software (or hardware)
 - ▶ Bad design, requirements
 - ▶ Bad policy/configuration
 - ▶ System Misuse
 - ▶ Unintended purpose or environment
 - E.g., student IDs for liquor store

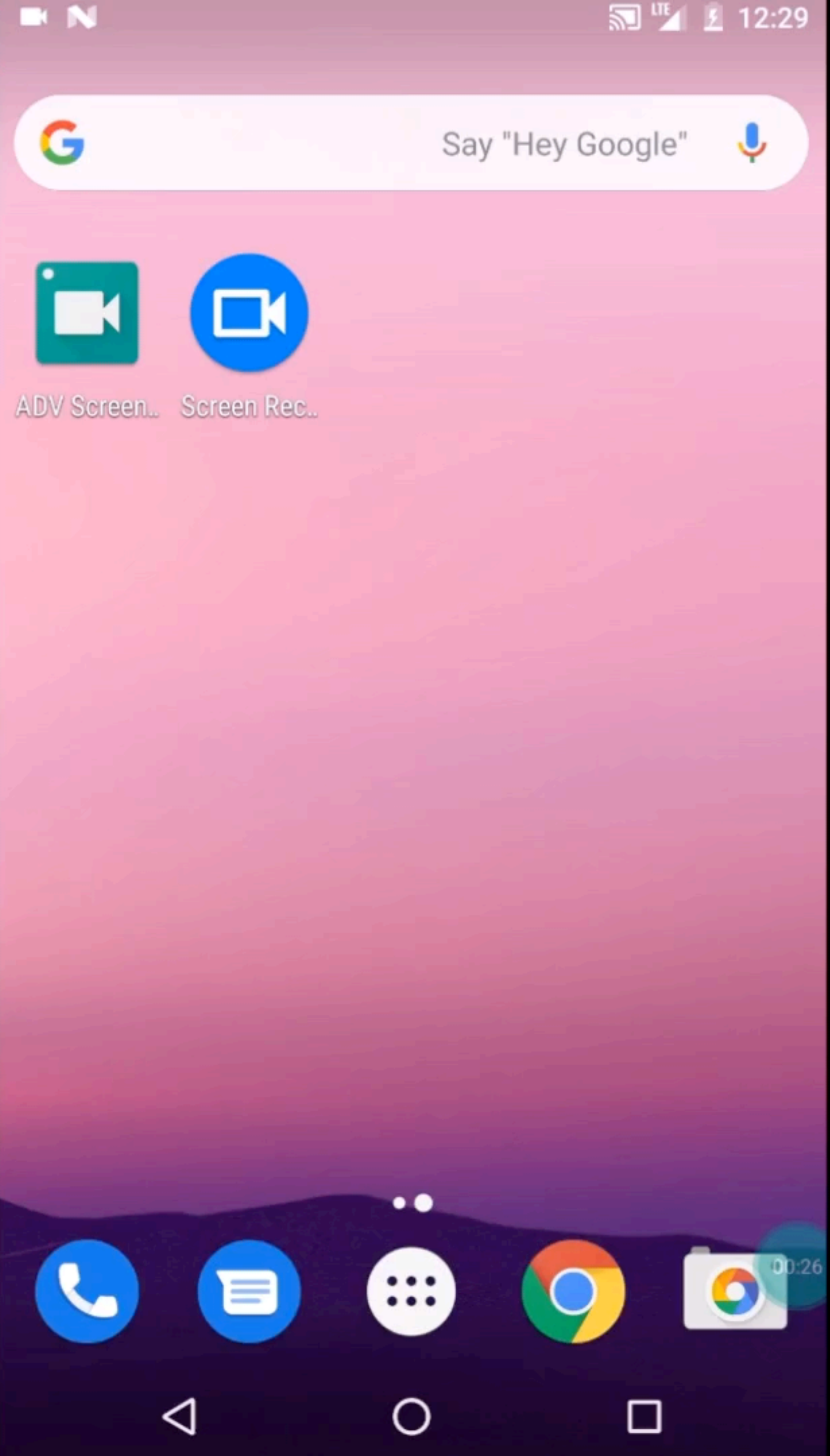


- An **attack** occurs when an adversary attempts to **exploit** a vulnerability
- Kinds of attacks
 - ▶ Passive (e.g., eavesdropping)
 - ▶ Active (e.g., password guessing)
 - ▶ Denial of Service (DOS)
 - Distributed DOS – using many endpoints
- A **compromise** occurs when an attack is successful
 - ▶ Typically associated with taking over/altering resources



```
(base) cyber2slab@cyber2slab-ThinkPad-T480:~/Documents  
/src$
```

I



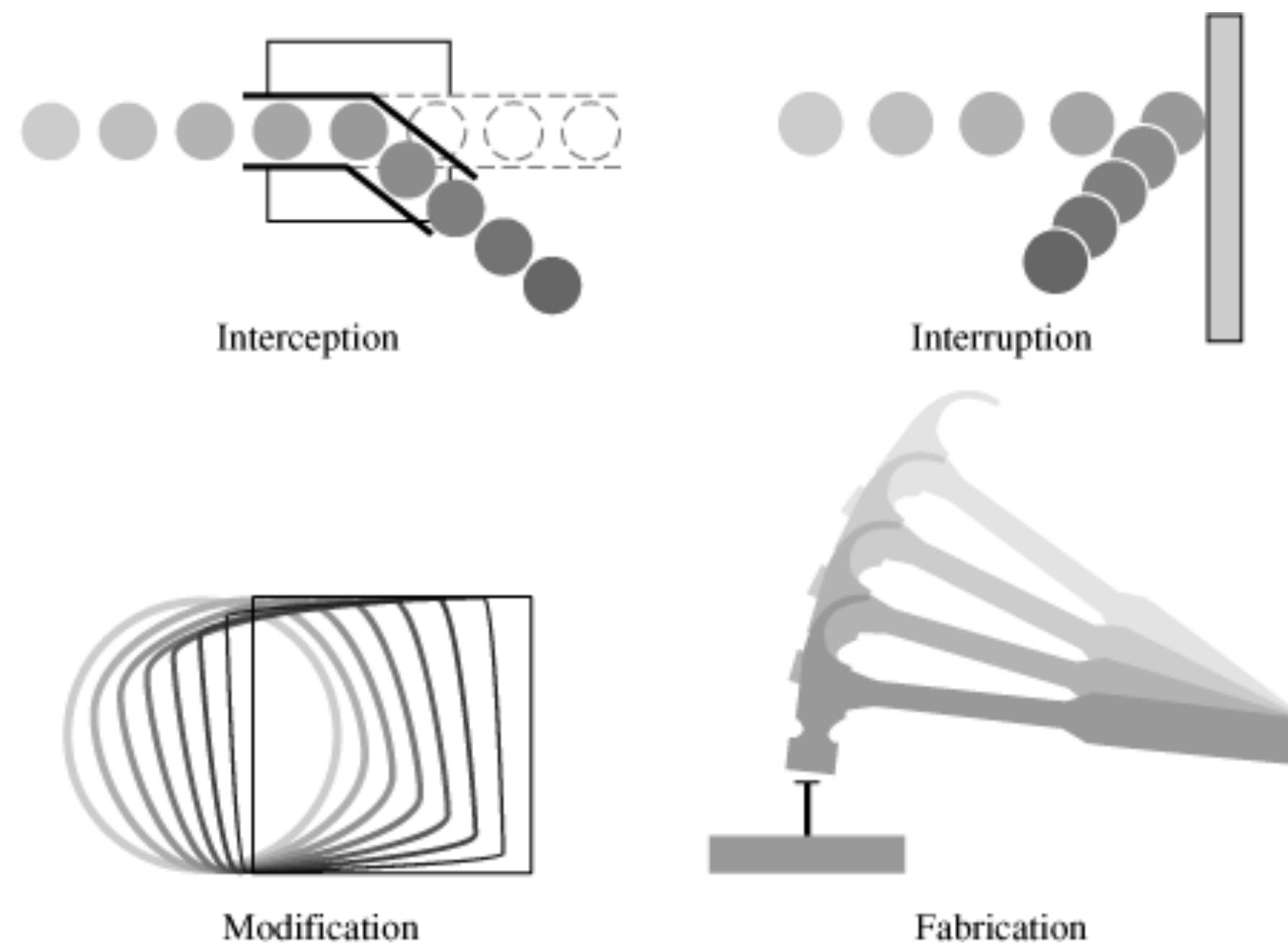
Attack Archetypes

Attacks will typically fall into one of four “Archetypes”:

- **Interception** — unauthorized access to an asset
- **Modification** — unauthorized changes to an asset
- **Fabrication** — creation of fake objects
 - ▶ Files, Messages, etc.
- **Interruption** — asset is “lost, unavailable, unusable”

Attach Archetypes

- Are these “archetypes” mutually exclusive?
- Does modification require interception?
- Can a victim detect interception?



Thinking Like a Defender

- Security Policy
- Threat Model
- Risk Assessment
- Countermeasures



- Measures taken to reduce the potential or impact of an attack
- AKA “Controls” or “Countermeasures”
- There are Five Defense Archetypes:
 - ▶ Prevention — “block the attack or close the vulnerability” (P&P)
 - ▶ Deterrence — Make attack harder (but not impossible)
 - ▶ Deflection — Make target less desirable than others
 - ▶ Detection — Detect attack in progress (and try to do something about it)
 - ▶ Recovery — Assume attack and just plan to fix things later

- Is prevention feasible?
- Can deflection be an effective strategy?
- Can we prevent/deter attacks that we can't detect?



Authentication

- Authentication means “Who are you?”
- Why is this an important security question?
- Why is it alone not enough?



- Authorization means "What are you allowed to do?"
- Physical world examples?
- Can we have access control without authentication?

- **Trust** refers to the degree to which a principal is expected to behave
 - ▶ What the principal not expected to do?
 - E.g., not expose password
 - ▶ What the principal is expected to do (obligations)?
 - E.g., obtain permission, refresh
- A **trust model** describes, for a particular environment, who is trusted to do what?
- **Note: you make trust decisions every day**
 - ▶ Q: What are they?
 - ▶ Q: Whom do you trust?

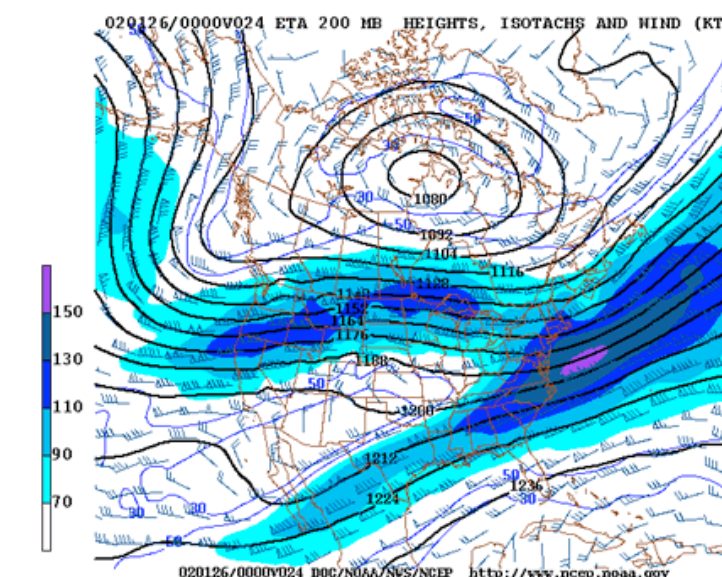


Trusted vs. Trustworthy

- Trusted and trustworthy are often confused
- “A **trusted** system or component is one whose failure can break the security policy.” (Anderson)
 - ▶ **Trusted Computing Base (TCB)** is the parts of the system you trust.
- “A trustworthy system or component is one that won’t fail.” (Anderson)

Security Model

- A **security model** is the combination of a trust and threat models that address the set of perceived risks
 - ▶ The “security requirements” used to develop some cogent and comprehensive design
 - ▶ Every design must have security model
 - LAN network or global information system
 - Java applet or operating system
- This class is going to talk a lot about security models
 - ▶ What are the security concerns (risks)?
 - ▶ Who are our adversaries?
 - ▶ What are the threats?
 - ▶ Who do we trust and to do what?
- Systems must be explicit to be secure.

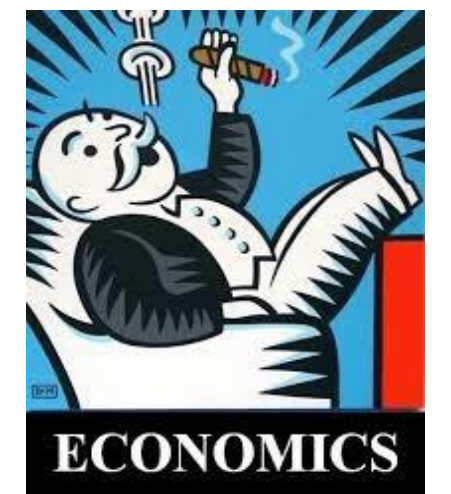


A Security Model Example

- Assume we have a University website that hosts courses through the web (e.g., Canvas)
 - ▶ Syllabus, other course information
 - ▶ Assignments submissions
 - ▶ Online grading
- In class: elements of the security model
 - ▶ Principals (Trusted)
 - ▶ Adversaries
 - ▶ Risks
 - ▶ Threats



- Security is practically always a cost
- This is one factor why computer systems are not as secure as they technologically could be
- Security potentially adds another point of failure in complex systems
- What are the costs of computer security?



- Performance

- ▶ Usability

- ▶ Time

- Developers, Sys Admins, Educators, Users

- Capital

- ▶ HW Firewalls, Replace outdated devices, more servers to support load of encryption

- Services

- ▶ DoS protection, AV subscriptions, Monitoring

The Security Mindset

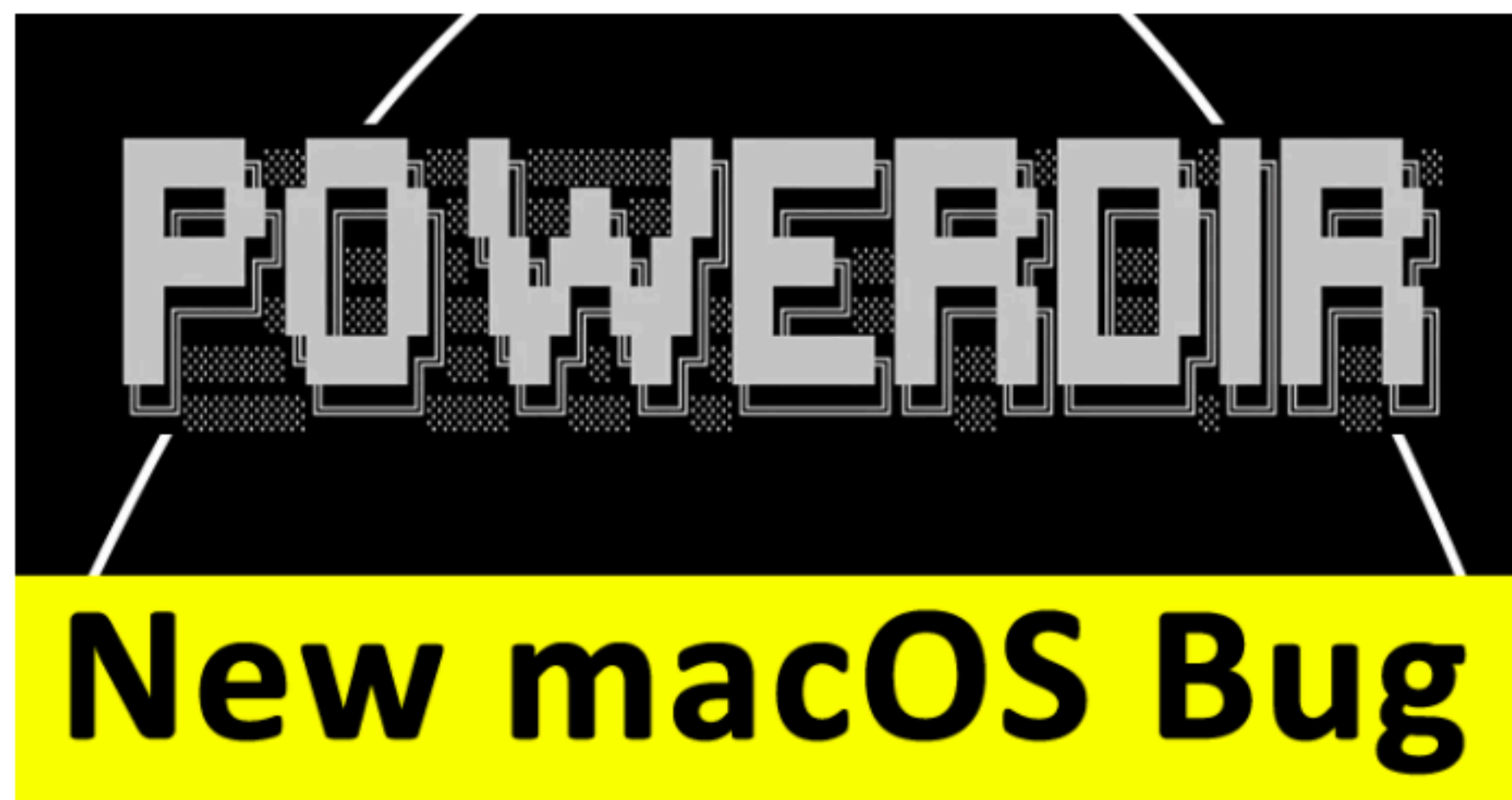
- Thinking like an attacker
 - ▶ Understanding how to circumvent security
 - ▶ Look for where security can fall down
- Thinking like a defender
 - ▶ What are you defending and from whom
 - ▶ Weigh benefits vs. costs: **No system is ever completely secure!**

Coming Attractions

- Symmetric Key Cryptography

'Powerdir' New macOS Bug Let Hackers Accessed Unauthorized User Data Access

By Balaji N - January 12, 2022 0



A new macOS vulnerability has been detected recently by the security team of Microsoft that is tracked as "powerdir," and this vulnerability is identified as [CVE-2021-30970](#).

This security flaw allows any threat actors to bypass one of the crucial technologies of macOS, Transparency, Consent, and Control (TCC).

Evading the Transparency, Consent, and Control (TCC) technology of macOS means gaining unauthorized access to the protected data of macOS users.



'Powerdir' New macOS Bug Let Hackers Accessed Unauthorized User Data Access

January 12, 2022



Heap-overflow Vulnerability Affects Multiple VMware Products

January 6, 2022



Beware of Fake RedLine Stealer That Distributed As Fake Omicron Stats Counter

January 13, 2022

Acknowledgements



Acknowledgement: Materials for the class lectures are taken from Trent Jaeger (Penn State), Ninghui Li (Purdue), Christina Garman (Purdue), and William Enck (NCSU).

My Research Group SyNSec (Systems and Network Security) is hiring!