Introduction to Cyber Security

Fall 2017 | Sherman Chow | CUHK IERG 4130

Chapter 3
System Security

This Lecture

- Physical Security
- Password
- Privilege Control
- Preventive Measures
- Proactive Security Measures

Physical Security

- Protecting the hardware, restrict access to system console, operating room
- Safeguard backup tapes
- Encrypted file-systems (*e.g.*, consider loss laptop)
  - "Data at rest"
- Bring your own device (BYOD)
- Look for leaks in *Security Perimeter*, *e.g.*
  - Rogue/Un-authorized Wireless Access Point
  - Simultaneous connections to Intranet and Internet by a Work-from-Home computer/user
- Beware of keystroke loggers, both software and hardware-based

Keyboard Acoustic Emanations Revisited

- First paper: [Asonov-Agrawal@S&P'04]
- "Taking as input a 10-minute sound recording of a user typing English text using a keyboard,
- and then recovering up to 96% of typed characters
- … can even recognize random text such as passwords
- … 80% of 10-character passwords can be generated in fewer than 75 attempts."
- [Zhuang-Zhou-Tygar@CCS'05] (a team from UC Berkeley)

Sniffing Keystrokes w/ Lasers/Voltmeters

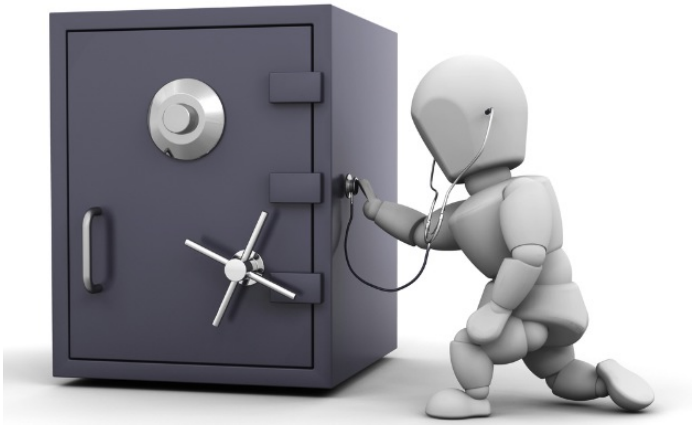
- “Attack 1: Power Line Leakage detection against wired PS/2 keyboards
- Attack 2: Optical Sampling of Mechanical Energy against laptop keyboards
- Unconventional side channel attacks
- Relatively cheap hardware
- As always....more important: girls will melt when you show this...”
- [Barisani-Bianco@BlackHat’09]

Blind Recognition of Touched Keys on Mobile Devices

- "We … analyze the shadow formation around the fingertip, apply … computer vision techniques to locate the touched points.
- We address both cases of tapping with one finger and tapping with multiple fingers and two hands.
- The per-character (or per-digit) success rate is over 97% while the success rate of recognizing 4-character passcodes is more than 90%."
- [Yue *et al.* @ CCS '14.]

Attack, Defense, …

Target not only on (Weak) Password
but also (Strong) Cryptographic Key



Side-Channel Attacks

- Side Channels are abundant:
  - running-time
  - power consumption
  - electromagnetic radiation
  - sound, *etc.*
- Cold-boot attack
  - side-channel attack announced in 2008 from a team in Princeton
  - retrieve secret key from a running operating system
  - compressed air cool/slow down the memory degradation

Against Side-Channel Leakage of Secret

- We have specific countermeasures, but
  - financially expensive, *e.g.*, tamper-resistant hardware module
    - aim to make extraction of the secret key from the hardware difficult
    - *e.g.*, consider TV box
  - or computationally expensive
    - traditional cryptographic schemes relies on perfect secrecy of the key
    - leakage-resilient schemes remain secure even if "part" of the secret is revealed, but they are more complicated
  - or not foolproof
    - cf. formal security guarantee by leakage-resilient cryptography
  - or attacks that are yet to be known, *etc*.

Dictionary attack against Password

- Let's talk about attack first
- In "dictionary attack", the adversary is trying to guess the victim's password by trying all possibilities from a dictionary.
- Consider the password is an English word, *e.g.*, "obscurity", when the attacker hits that entry in the dictionary, it guess the password correctly.
- Generalizing, the attacker can prepare a "passwords dictionary" listing all possibilities of passwords according to the rule
  - *e.g.*, with at least one numeric character and one symbol, etc.

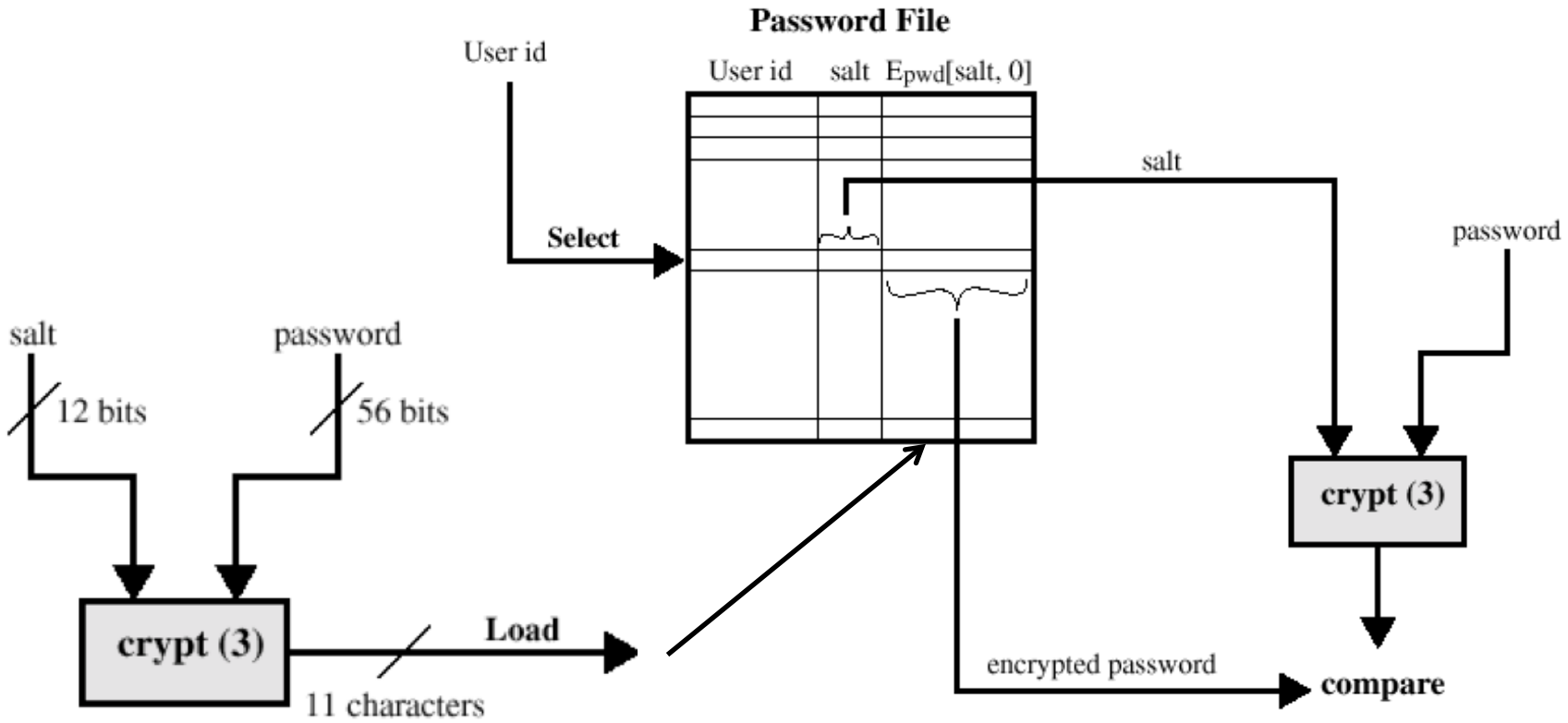
Password

- For your online account (*e.g.*, e-banking), it might be locked if you have mistyped your password for a threshold # of times
  - Prevent online dictionary attack
  - Security vs. usability (it's you who are testing the few possibilities?)
- The password is stored somewhere anyway
  - What if the system administrator is malicious?
  - Or the system is compromised
- Encrypt the password?
- Offline dictionary attack (try all possible decryption key to decrypt the encrypted password)
  - Key space is supposed to be large(r than that of password space)?

Entropy of Password

- The password should have high-entropy to withstand this attack
- *n*-bit of entropy, 2^n possibilities
- A fair coin gives you 1 bit of entropy
- If you chose a 4-character ASCII string uniformly at random, entropy:
  - $\log_2(256^4) = 4\log_2(256) = 4 \times 8 = 32$

Unix Password Scheme depicted



Generation of Record to be Stored

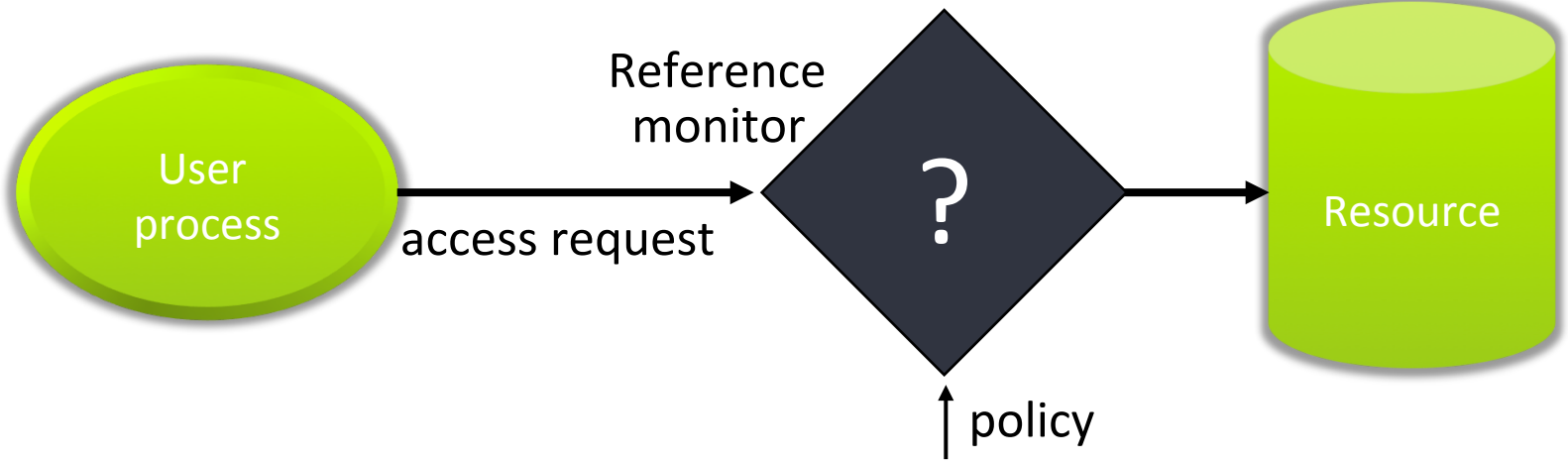
Verification against Stored Record

Unix Password Scheme

- Salting:
  - Each salt is randomly generated (w/ high prob. diff. user have diff. salt)
  - Slow down dictionary attacks
  - Prevents duplicate passwords to be noticeable
  - Effectively increases the length of the password
- The “crypt” function is a minor variant of DES (data encryption standard)
  - Prevent attackers to use hardware DES accelerator for cracking
  - DES is a symmetry-key block-cipher, see the “Crypto” part of the course
- UNIX passwords were kept in a publicly readable file
  - Located at etc/passwords
  - Now they are kept in a “shadow” directory and only visible by “root”.

(Software-based) Access Control

- System knows who the user is via Authentication
  - *e.g.*, name, password, or other credentials
- Access requests pass through gatekeeper
  - System must not allow monitor to be bypassed
- Trust on the system, it must not lie about who the user is.



Privilege

- Ability to access or modify a resource
  - a.k.a.: access right: permission, *etc*.
- Principle of Least Privilege
  - "A system module should only have the minimal privileges needed for its intended purposes"

Pokémon GO

Version 0.37.1 may request access to

Camera
- take pictures and videos

Contacts
- find accounts on the device

Location
- approximate location (network-based)
- precise location (GPS and network-based)

Storage
- modify or delete the contents of your SD card
- read the contents of your SD card

Other
- activity recognition
- prevent phone from sleeping
- receive data from Internet
- access Bluetooth settings
- control vibration
- view network connections
- full network access
- Google Play billing service
- pair with Bluetooth devices

Two Ideas of Implementations

- Access control list (ACL)
  - Store column of matrix with the resource
  - More widely used
  - User can be generalized to groups / roles
  - Usually apply on “low level” objects
- Capability
  - User holds a “unforgeable ticket” for each resource
  - Store row of matrix with user, under OS control

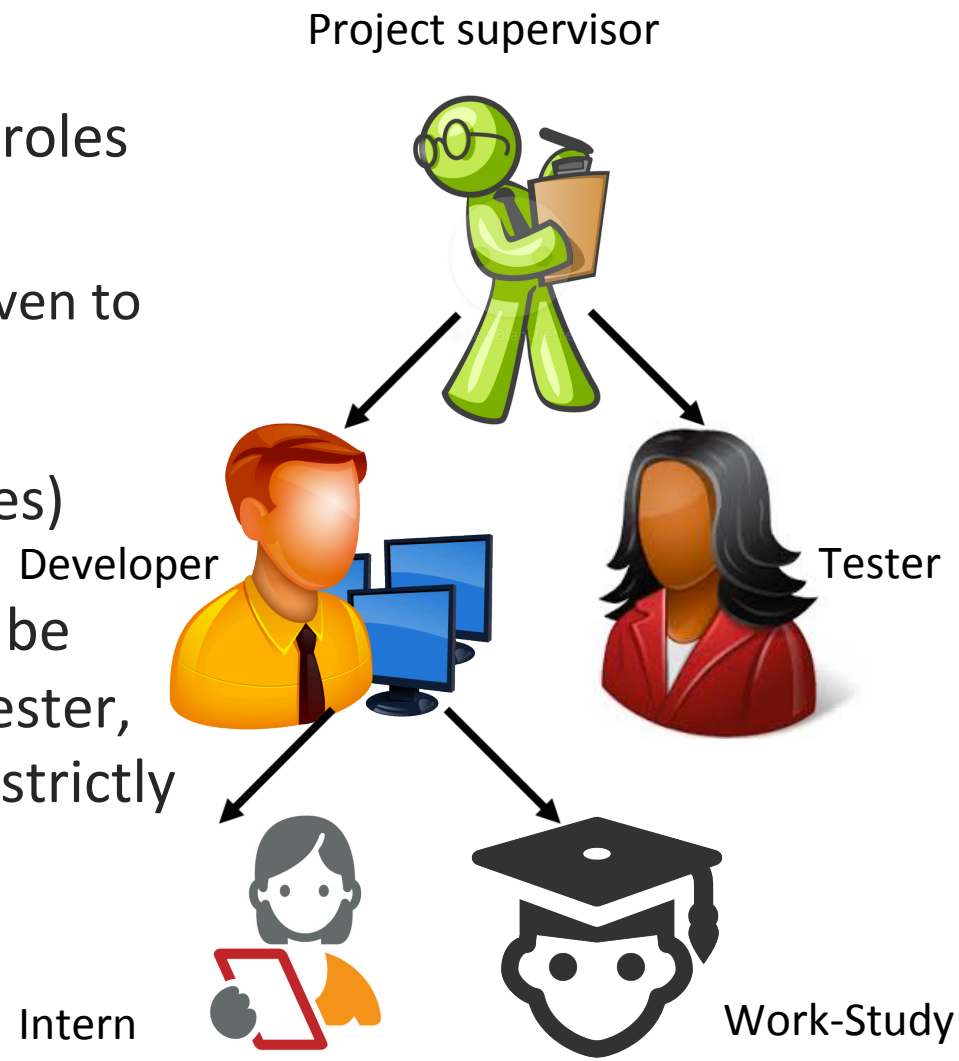
	File 1	File 2	…
User 1	read	write	-
User 2	write	write	-
User 3	-	-	read
…			
User *m*	read	write	write

ACL vs. Capabilities

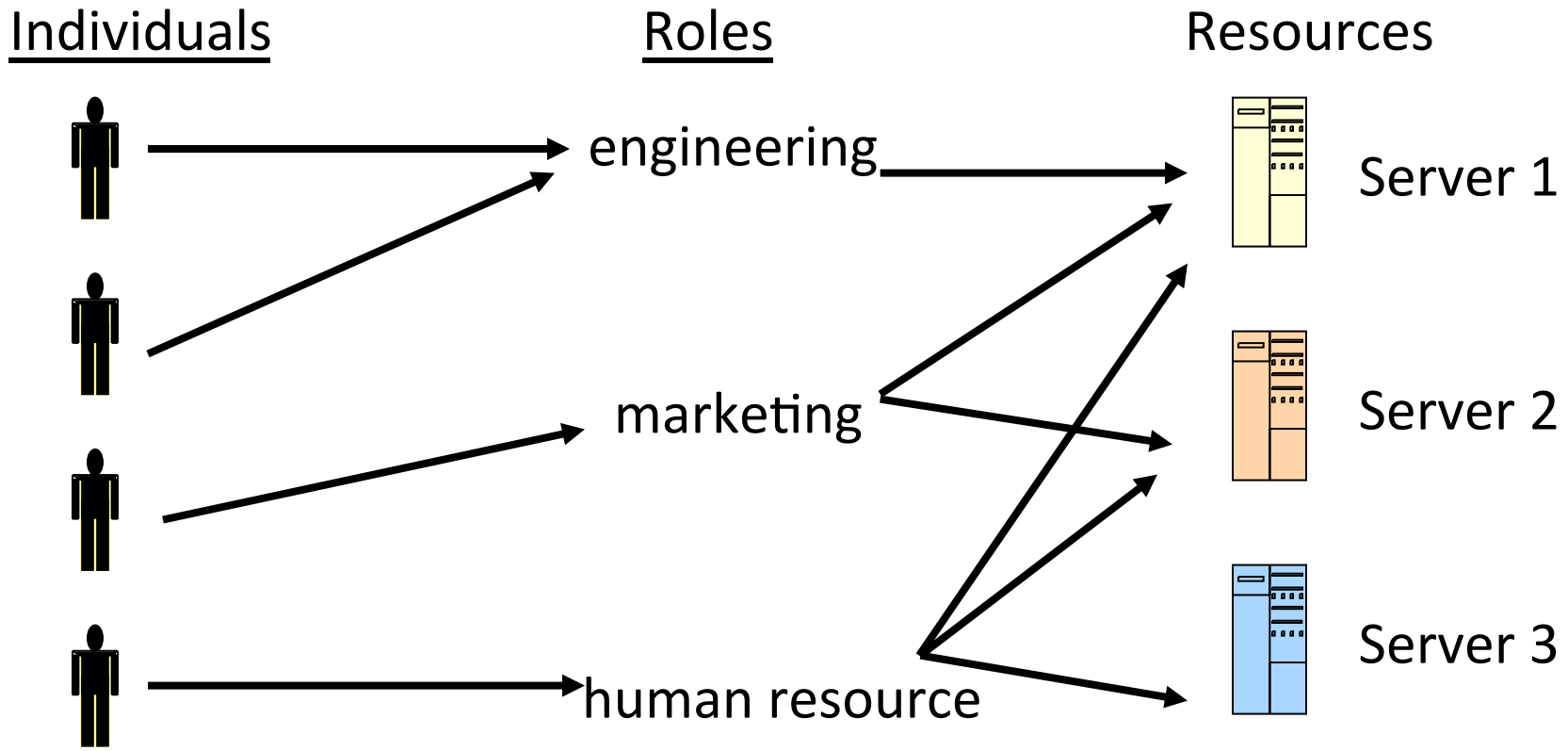
- Access control list
  - Associate list with each object
  - Check user/group against list
  - Relies on authentication: need to know user
- Capabilities
  - Reference monitor *checks* capabilities in the form of tickets
  - Ticket is random bit sequence, or managed by OS
  - Can be passed from one process to another
  - Doesn't need to know the identity of user/process for *checking*

Role-Based Access Control (RBAC)

- Each role gets permissions of roles below
  - List only new permissions given to each role
- Partial order (hierarchy of roles)
- Developer's permissions may be "incomparable" to those of Tester, *i.e.*, We cannot say if one has strictly more permissions than other



RBAC depicted



- Users change more frequently than roles
- An individual may also have multiple roles

Problem in Access Control

- There are other models, *e.g.*, Context-based Access Control
- Complex mechanisms require complex input
- Difficult to configure and maintain
  - Roles, other organizing ideas try to simplify problem
  - Hierarchy for resources, *e.g.*, If user has read access to directory, user has read access to every file in directory
  - But still…
- Still a major research area
- ACM Symposium on Access Control Models and Technologies (SACMAT)
  - 22nd Edition in 2017

ACL in Unix/Linux

- ACL of a resource determines who can have what type of access of it
  - Who: specified in form of UserID (u), GroupID (g), other (o), etc
  - Type: read, write, execute, from local console, from network, etc
- Can use chmod command to change access rights
  - *E.g.*, chmod o-rwx [filename], or chmod 770 [filename]
  - this command will disable [o]thers to read/write/execute the file
  - but allow user "*UserID*" and his/her group to have full (4+2+1) access
  - [r]ead: 4 [w]rite: 2, e[x]ecute: 1
- In an networked environment, each user (program) carries a credential showing his/her UserID as well as Group membership info

Privilege of a running program

- A running program/process "typically" inherits the access rights of the login-account through which a program is run
- Instead of inheriting the rights of the program's runner, Unix is based on "Setuid" which may "inherit" the rights of the program's owner
  - *E.g.*, mkdir command in UNIX changes file-system data-structure
    - *i.e.*, need "root" or superuser privilege,
  - Thus, mkdir is owned by root but executable by users.
  - If a user runs mkdir, its "effective userid" is switched to "root"
  - Setuid-programs are especially "dangerous" because if there is a flaw in such programs, attacker can exploit it to gain superuser privilege!
    - *e.g.*, sendmail http://www.cis.syr.edu/~wedu/Teaching/cis643/LectureNotes_New/Set_UID.pdf

Android Application Sandbox for Isolation

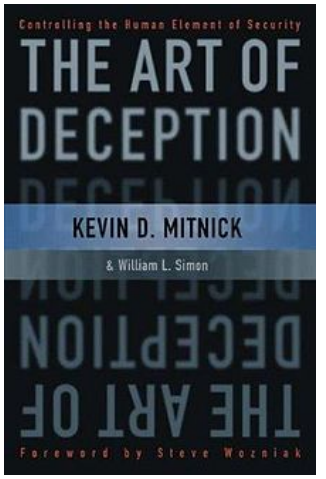
- Each application runs with its own UID in own Virtual Machine
  - Provides memory protection
  - Communication protected using Unix domain sockets
  - Only ping, zygote (spawn another process) run as root
- Interaction: reference monitor checks permissions on inter-component communication
- Least Privilege: Applications announces permission
  - User grants access at install time

Basic System Security Measures

- Hold everyone accountable for Security
- Always Set a Password, Make it Complex and Change it Often
  - (not too overdone though)
- Keep Up with Vendor Patches – Diligently and Promptly
  - (well, there could also be some problematic patches…)
- Block or Disable Everything that is not Explicitly allowed
- Authorize All Access Using Least Privilege
- Limit Trust
- Be Paranoid with External Access
  - Dial up, Networking, Terminal-server, Shared files, *etc*.
- Defense in Depth – Compartmentalization

More Basic Measures

- Perform Real-World Risk Assessments; Independent Penetration tests (aka hiring the Tiger-teams)
- Educate Users
  - (against) Social Engineering
  - Technology implications
- Learn better than your Enemies
  - your platforms, technologies and applications
- Building Secure Software if this is under your control
  - *e.g.* http://research.microsoft.com/en-us/um/redmond/events/swsecinstitute



Proactive System Security Measures

- Security Vulnerability Analysis/Scanners:
  - System Scanners
  - Network Scanners
- Intrusion Detection System (IDS)
- System Hardening
  - Turn-off unused/unneeded services, accounts
  - Tightening default configurations
- Logging activities and perform Log file analysis
- File-system Integrity Checks

Next Lecture: Web Security

- Domain Name Server (DNS), Time-to-Live (TTL)
- Cookies
- SQL Injection vs. Input Validation
- Session Management
- JavaScript (and its Security)
- Same Origin Policy (SOP)