Class Information

What?
- It's about computer and network security
- Textbook:
  - *Introduction to Computer Security* by Matt Bishop
  - Does everybody have the book?
- Reference book:
  - *Network Security: Private Communication in a Public World* by Charlie Kaufman, Radia Perlman, and Mike Speciner
- Web page:
  - [http://www.cs.uoregon.edu/classes/05F/cis410cnsec/](http://www.cs.uoregon.edu/classes/05F/cis410cnsec/)

Prerequisites
- CIS 432 or instructor approval

Who?
- Instructor:
  - Jun Li: lijun@cs.uoregon.edu
  - Office: Deschutes 334

When?
- Class hours:
  - Mondays, Wednesdays 16 - 17:20 a.m.
  - 203 Chapman Hall
- Office hours:
  - TBA soon
  - 334 Deschutes

Grading Policy
- Class participation: 10%
- In-class quiz: 5%
- Homework: 25%
- Final: 60%
Quiz and Homework

- **Quiz**:
  - To check your learning quality
  - No announcement in advance
  - Often in 5 minutes

- **Homework**:
  - Usually due in one week
  - No late submission will be accepted

Exams

- **Final**:
  - December 5, Monday, 15:15-17:15, 203 Chapman

On using emails

- Subject line preferred to be in the format of
  - CIS 410: <issue>, or
  - CIS 510: <issue>

- We will create a mailing list of everyone in the class
  - Based on your registered email address at Duckweb
  - If you do not receive a test email by this Friday, contact me
  - Some important announcements will use this mailing list

An Overview of Computer Security

- **Confidentiality**
  - The concealment of information or resources
    - “need to know” principle
  - Mechanisms to support confidentiality
    - Cryptography
    - System-dependent mechanisms
      - E.g. File permission handling

Basic Components

- **Confidentiality**
- **Integrity**
- **Availability**
Integrity

- Trustworthiness of data or resource
- Includes:
  - Data integrity
  - Origin integrity
  - Source authentication to ensure credibility
- Mechanisms to support integrity:
  - Prevention
    - Block outsiders and insiders
  - Detection
    - Analyze system events and data

Availability

- The ability to use the info or resource as desired
- Mechanisms to support availability:
  - ?

Threat and Attack

- Threat: a potential violation of security
  - Need not actually occur
  - E.g. data loss
- Attack: an action that can cause a threat to become real
  - E.g. wire eavesdropping
  - Interchangeable in the following text

Four Classes of Threat

<table>
<thead>
<tr>
<th></th>
<th>Detection</th>
<th>Integrity</th>
<th>Availability</th>
<th>Mechanism</th>
<th>Auditability</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Goals of Security

- Prevention
- Detection
- Recovery

- Based on the security policy defined
- Using the security mechanism
Assumptions

- How to determine whether or not a security policy is correctly defined?
  - Depending on the specific assumptions made to a specific system in a specific environment
- Two assumptions often made of a security policy
  - The policy leads to two types of states: secure and nonsecure
  - The security mechanisms prevent the system from entering into a nonsecure state

Mechanisms: secure, precise, or broad

- Denote
  - P: the set of all possible states
  - Q: the set of secure states as specified by security policy
  - R: the set of states that a system can enter with the security mechanisms provided
- A security mechanism is
  - Secure: if $R \subseteq Q$
  - Precise: if $R = Q$
  - Broad: if $\exists$ state $r \in R$ but $r \notin Q$

Assurance

- How much can one believe that a system is secure?
- Look at three aspects:
  - Specification
    - formally or informally states the desired functions, and security policies
  - Design
    - translates the specifications into components
  - Implementation
    - creates a system that satisfies the design
- Assurance: what one must trust in order to believe the security of a system

Operational Issues

- Useful security policy and mechanism must consider cost-benefit tradeoff
- Risk analysis is often the key
  - In order to know whether and to what level to protect a system
  - A function of environment, time, service provided, etc.
- Laws and Customs

Human Issues

- Organizational problems
  - Who is responsible, where are necessary resources, what to protect, etc.
- People problems
  - Heart of any security problem is people
  - Human intervention can often bypass security mechanisms
  - Outsiders vs. insiders
  - Social engineering
  - Configuration errors
  - etc.

Security Life Cycle

- Threats
- Policy
- Specification
- Design
- Implementation
- Operation and Maintenance
**Malicious Logic**

- **Malicious logic**: a set of instructions that cause a site’s security policy to be violated
- Typically by assuming an authorized user’s identity
- We will talk Trojan horses, viruses, worms, etc.

**Introduction**

- **Trojan Horse**: a program with an overt effect and a covert effect
  - Overt effect: documented or known effect
  - Covert effect: undocumented or unknown effect
- **Propagating Trojan horse**: a Trojan horse that creates a copy of itself
  - Also called replicating Trojan horse

**Trojan Horse**

- An example:
  A Unix script called “ls” in directory /foo, created by EVE:
  ```bash
cp /bin/sh /tmp/.xxsh
chmod o+s, o+x /tmp/.xxsh
rm /ls
ls $*
```
  If user VICTOR is cheated to run the above “ls” instead of the original “ls” he will create a /tmp/.xxsh file. EVE then can run /tmp/.xxsh, a shell program that allows EVE to enjoy a shell running environment for VICTOR!

- This “ls” by EVE does do what the original “ls” does
  - But it also does something else: creating a shell program that can give EVE the access rights of VICTOR!

**Computer Viruses**

- **Computer virus**: a program that inserts itself into one or more programs and then performs some action
  - A Trojan horse only propagates itself
  - A virus infects others

- Note: some experts regard virus as a type of Trojan horse, where the infected program gives the overt effect and the virus code leads to covert effect.

**Virus Mechanism**

```bash
beginvirus:
if spread-condition then begin
for some set of target files do begin
  if target is not infected then begin
    1. determine where to place virus instructions
    2. copy virus instructions from beginvirus to endvirus into target
    3. alter target to execute added instructions
  end
end
perform some action(s)
goto beginning of infected program
endvirus
```
There are many many viruses . . .

- And there will be many many of them!

Types:
- Boot sector infectors
- Executable infectors
- Multipartite viruses
- TSR viruses
- Stealth viruses
- Encrypted viruses
- Polymorphic viruses
- Macro viruses

Boot Sector Infectors

- Boot sector: the part of a disk used to bootstrap the system
  - Or mount a disk
- Boot sector is executed when the system “sees” the disk for the first time
- Boot sector infector: a virus that inserts itself into the boot sector of a disk
  - Loaded into memory when invoked

Executable Infectors

- Executable infectors: a virus that infects executable programs (or applications)
  - Also called COM or EXE viruses
  - The virus prepends or appends itself

Multipartite Viruses

- Multipartite virus: a virus that can infect either boot sectors or applications
  - Often has two parts, one for each type
    - Depending on what to infect

TSR Viruses

- Terminate and stay resident (TSR) virus: a virus that stays active (resident) in memory after the application (or bootstrapping, or disk mounting) has terminated
  - Can be boot sector infectors or executable infectors

Stealth Viruses

- Stealth virus: a virus that conceal the infection of files.
- For example:
  - If you try to read an infected program to check the integrity of the program, the virus will return you the original program
  - But, the virus will still get executed when the program is executed
Encrypted Viruses

- **Encrypted virus**: a virus that uses enciphered code
  - In order to prevent virus detection

Macro Viruses

- **Macro virus**: a virus that composed of a sequence of instructions that is interpreted, rather than executed directly
  - Can infect either executable applications, or data files
  - The latter called **data virus**

Polymorphic Viruses

- **Polymorphic virus**: a virus that changes its form each time it inserts itself into another program
  - For example, using different machine instructions with the same effects
  - Can be automated
  - An encrypted virus can evolve into a polymorphic virus by varying the decipherment routine

Computer Worms

- **Computer worm**: a program that copies from one computer to another
  - A variant of virus
  - Many worms have occurred
    - Father Christmas worm, CodeRed worm, Slammer worm, etc.
    - The Slammer worm spread worldwide in ~5 minutes

Rabbits and Bacteria

- **Rabbit or bacterium**: a program that absorbs all or some class of resource
  - Creating denial-of-service attack
- For example
  ```plaintext
  while (true) do
    mkdir x
    chdir x
  done
  ```

Logic Bomb

- **Logic bomb**: a program that performs a malicious action when some external event occurs