Overview

Course Goals
Course Style

Course Content
- System and Software Security Foundations
- Vulnerability Analysis and Exploit Generation
- System Defense
- Reverse Engineering

Course Project

Course Policy

Homework
Course Goals

The **general goal** is to understand the state-of-the-art

**Offense/Attack**
- Find memory vulnerability
- Develop exploits
- Create malware

**Defense/Protection**
- Find vulnerability
- Stop exploits
- Analyze malware
- Ensure availability

Offense/Attack

**Mission**
Break the system to gain resources with respect to **confidentiality** (something you are not supposed to access) and **integrity** (something you are not supposed to modify)

**Techniques**
- Find memory vulnerability
  - Memory vulnerability
    - Buffer overflow, integer overflow, and format string
  - Logic vulnerability or new web vulnerability (SQL injection)
- Develop exploits
  - Memory exploits, shell code, ROP, heapspray
- Create malware (Obfuscation/Packing)
  - Packing (encryption)
  - Translation, virtualization

Defense/Protection

**Mission**
Protect information and system resources with respect to **confidentiality**, **integrity**, and **availability** (defending such as Denial of service attack)

**Techniques**
- Find vulnerability (Penetration Testing)
- Stop exploits
  - Architecture, hardware
  - Operating System, Loader, Linker
  - Compiler
- Analyze malware (Reverse Engineering)
  - Unpacking (decryption)
  - De-transformation, De-virtualization
- Ensure availability, preventing DoS

Outline

1. Overview
   - Course Goals
   - Course Style
2. Course Content
   - System and Software Security Foundations
   - Vulnerability Analysis and Exploit Generation
   - System Defense
   - Reverse Engineering
3. Course Project
4. Course Policy
5. Homework
This course is taught in both a seminar and a regular-course style. Each student will be expected to:

- **Read** one of the most recent papers given by the instructor thoroughly.
- **Present** the key techniques in the paper within 12 minutes, with additional 3 minutes question.
- **Attend** all the lectures.
- **Scribe** one lecture (write notes).
- **Perform**
  - An individual research project, or as
  - A team for an engineering project (5K LOC), with 3 members.

### Outline

1. **Overview**
   - Course Goals
   - Course Style
2. **Course Content**
   - System and Software Security Foundations
     - Vulnerability Analysis and Exploit Generation
     - System Defense
     - Reverse Engineering
3. **Course Project**
4. **Course Policy**
5. **Homework**

### Understanding OS Kernels

**Topics**

- **Process Management**
  - Creation/Running/Exit
  - Process address space
  - Context switch
- **Virtual Memory**
  - Paging
- **File System and Disk Data Management**
  - EXT2/EXT3, NTFS
  - Proc

**Topics**

- **Process Management**
  - Creation/Running/Exit
  - Process address space
  - Context switch
- **Virtual Memory**
  - Paging
- **File System and Disk Data Management**
  - EXT2/EXT3, NTFS
  - Proc
Exploits

Topics
- Shell Code
- Code Injection
- Return-into-libc
- Reliable Shell Code
  - HeapSpray (ASLR)
  - Return-oriented programming

Outline

1. Overview
   - Course Goals
   - Course Style

2. Course Content
   - System and Software Security Foundations
   - Vulnerability Analysis and Exploit Generation
   - System Defense
   - Reverse Engineering

3. Course Project

4. Course Policy

5. Homework

Architecture, OS Perspective

Topics
- Address Space Randomization (ASR), DEP, NX-bit
- Instruction Set Randomization (ISR)
- Data Randomization (DR)
- Operating System Interface Randomization, RandSys
- N-Variant System, Reverse Stack Execution
- System Call Interposition
- Library Extension (Libsafe/LibsafePlus/LibsafeXP)
- Virtual Machine Introspection

Compilation Extension, Code Transformation, Runtime Verification

Topics
- Bounds Checking, Type Checking
- Diehard (heap protection), exTerminator
- Binary Rewriting, SFI/XFI/CFI/DFI
- Program Shepherding
- Sandboxing, NativeClient
- Taint Checking, Blocking Bad Input
Course Goals
- System and Software Security Foundations
- Vulnerability Analysis and Exploit Generation
- System Defense
- Reverse Engineering

Course Style
- Compilation Extension
- Code Transformation
- Runtime Verification

Course Project
- Implement a dynamic binary code analysis plug-in, using a most recent PIN, with a data flow analysis (taint analysis at byte level) capability
- Roughly code size 3K LOC

Sample Projects
- Vulnerability analysis
- Penetration testing
- Reverse engineering
Outline

Overview
- Course Goals
- Course Style

Course Content
- System and Software Security Foundations
- Vulnerability Analysis and Exploit Generation
- System Defense
- Reverse Engineering

Course Project

Course Policy

Homework

Grading Policy
- 20% In-Class Presentations
- 10% Class participation
- 20% Scribe a lecture (write a note)
- 50% Class Project
- Exceptional work will be rewarded appropriately

Prerequisites

System Skill Set
Solid programming/development skills (Assembly, C, C++, Unix). "Operating System", "Compilers", and "Computer Security", are the least prerequisites for this class. I am training "academic hackers"

UTD
- CS 3340 Computer Architecture
- CS 3376 C/C++ Programming in a UNIX Environment
- CS 4348 Operating Systems Concepts
- CS 4393 Computer and Network Security
- CS 4394 Implementation of Modern Operating Systems

Leave vs. Stay

char code[] = "\xb0\x01\x31\xdb\xcd\x80";

08048080 <_start>:
08048080:  b0 01 mov $0x1,%al
08048082:  31 db xor %ebx,%ebx
08048084:  cd 80 int $0x80

char code[] = "\x31\xc0\xb0\x46\x31\xdb\xcd\x80\xe8\xe5\xff\xff\xff\x2f\x62\x6e\x2f\x73\x68\x58\x41\x41\x41\x41\x42\x42\x42\x42";
### Overview Course Content Course Project Course Policy Homework

#### Other Policy

**Late Policy**
No late submission

**Collaboration Policy**
Encouraged, but limit the team member to at most three students.

**Cheating Policy**
Strictly follow the university policy on cheating and plagiarism

### Outline

1. **Overview**
   - Course Goals
   - Course Style

2. **Course Content**
   - System and Software Security Foundations
   - Vulnerability Analysis and Exploit Generation
   - System Defense
   - Reverse Engineering

3. **Course Project**

4. **Course Policy**

5. **Homework**

### Homework-0: Due next Monday

**Paper Presentation Sign-Up**
Selecting the paper (date implicitly selected) to present

**Scribe Sign-Up**
Selecting the date for to-be-scribed lecture Two students are allowed to scribe the same lecture if there is no slot

**Engineering Project**
- Forming your team with at most 3 members
- Starting to get familiar with PIN

**Research Project**
- Decide the research project
- Talk to the instructor