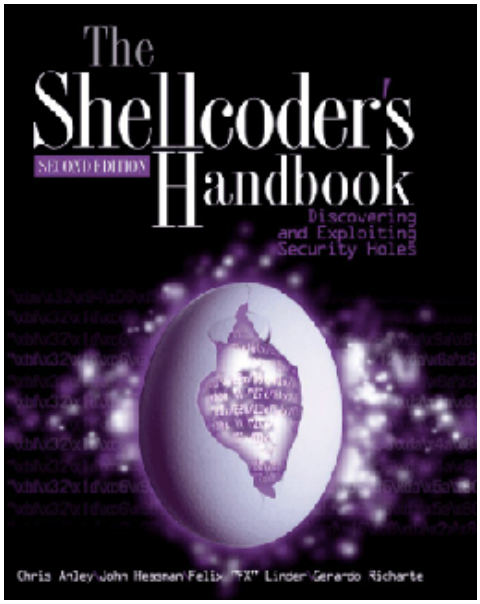
CNIT 127: Exploit Development



37711 Thu 6:10 - 9:00 PM SCIE 200

Spring 2017 Sam Bowne

Catalog Description

Learn how to find vulnerabilities and exploit them to gain control of target systems, including Linux, Windows, Mac, and Cisco. This class covers how to write tools, not just how to use them; essential skills for advanced penetration testers and software security professionals.

Advisory: CS 110A or equivalent familiarity with programming

Upon successful completion of this course, the student will be able to:

A. **Read and write basic assembly code routines**
B. **Read and write basic C programs**
C. **Recognize C constructs in assembly**
D. **Find stack overflow vulnerabilities and exploit them**
E. **Create local privilege escalation exploits**
F. **Understand Linux shellcode and be able to write your own**
G. **Understand format string vulnerabilities and exploit them**
H. **Understand heap overflows and exploit them**
I. **Explain essential Windows features and their weaknesses, including DCOM and DCE-RPC**
J. **Understand Windows shells and how to write them**
K. **Explain various Windows overflows and exploit them**
L. **Evade filters and other Windows defenses**
M. **Find vulnerabilities in Mac OS X and exploit them**
N. **Find vulnerabilities in Cisco IOS and exploit them**

Student Learning Outcomes (measured to guide course improvements)

1. Read and write basic assembly code routines
2. Find stack overflow vulnerabilities and exploit them
3. Evade filters and other Windows defenses

Textbook

"The Shellcoder's Handbook: Discovering and Exploiting Security Holes ", by Chris Anley, John Heasman, Felix Lindner, Gerardo Richarte; ASIN: B004P5O38Q Buy from Amazon

Quizzes

The quizzes are multiple-choice, online, and open-book. However, you may not ask other people to help you during the quizzes. You will need to study the textbook chapter before the lecture covering it, and take the quiz before that class. Each quiz is available for one week, up till 8:30 am Saturday. Each quiz has 5 questions, you have ten minutes to take it, and you can make two attempts. If you take the quiz twice, the second score is the one that counts, not necessarily the higher score.

To take quizzes, log in to CCSF's online class site here:

https://ccsf.instructure.com

Schedule (may be revised)

Date	Quiz	Topic
Thu 1-19		**Ch 1: Before you Begin**
Thu 1-26		**Ch 2: Stack overflows on Linux**
Thu 2-2		**Ch 3: Shellcode**
Fri 2-3	***Last Day to Add Classes***	
Thu 2-9	**Ch 1 Quiz due before class** **Ch 4 Quiz due before class** **Proj 0-2 due**	**Ch 4: Introduction to format string bugs**
Thu 2-16	**Ch 2 Quiz due before class** **Ch 5 Quiz due before class** **Proj 3-4 due**	**Ch 5: Introduction to heap overflows**
Thu 2-23	**Ch 3 Quiz due before class** **Ch 6 (Part 1) Quiz due before class** **Proj 5-6 due**	**Ch 6: The Wild World of Windows (Part 1)**
Thu 3-2	**Ch 6 (Part 2) Quiz due before class** **Proj 7 due**	**Ch 6: The Wild World of Windows (Part 2)**
Thu 3-9	**No Quiz** **Proj 8-9 due**	**Lecture 7: Intro to 64-Bit Assembler (Not in book)**
Thu 3-16	**Ch 8 (Part 1) Quiz due before class** **Proj 10 due**	**Ch 8: Windows overflows (Part 1)**
Thu 3-23	**No Quiz** **No Proj Due**	**Guest Speaker TBA**
Thu 3-30	***Holiday - No Class***	
Thu 4-6	**Ch 8 (Part 2) Quiz due before class** **Proj 11-12 due**	**Ch 8: Windows overflows (Part 2)**
Thu 4-13	**Ch 14 Quiz due before class** **Proj 13 due**	**Ch 14: Protection Mechanisms**
Thu 4-20	***Holiday - No Class***	
Thu 4-27	**Ch 16+17+18 Quiz due before class** **Proj 14-15 due**	**Ch 16: Fault Injection** **Ch 17: The Art of Fuzzing** **Ch 18: Source Code Auditing**
Thu 5-4	**No Quiz** **Proj 16 due**	**Hopper Debugger**
Thu 5-11	**No Quiz** **Proj 17-18 due** **All Extra Credit Projects Due**	**Last Class: TBA**
Thu 5-18		***Final Exam***