CS 493/693 - Intrusion Detection

Meeting time: 1-2pm Room 104 Chapman Building University of Alaska Fairbanks	CS F493-F01 #36664 CS F693-F01 #36665 3.0 Credits, Spring 2006 Prerequisite: CS 321 (OS)	Instructor: Dr. O. Lawlor ffosl@uaf.edu, 474-7678 Office: 210C Chapman Hours: 2-3 MWF or by appointment

Required Textbooks: *Intrusion Detection* by Rebecca Gurley Bace *Network Intrusion Detection (3rd Edition)* by Northcutt and Novak	ADA Compliance: Will work with Office of Disabilities Services (203 WHIT, 474-7043) to provide reasonable accomodation to students with disabilities.	Course Website (& links to Blackboard): **http://www.cs.uaf.edu/2006/spring/cs493** Machines: ASSERT lab, nanook.uaf.edu, Chapman lab, or Linux CDs available

Course Goals and Requirements

Intrusion Detection Systems (IDS) are an essential component of a computer security strategy. This course will focus on the reasons why IDS technology is important; the origin and resolution of common security holes; cryptographic and network approaches to IDS implementation; and legal, ethical, and privacy issues involved with IDS use. The course will illustrate the general principles of IDS design by examining specific cases on both Windows and Linux systems. Because many exploits and intrusion detection mechanisms relate to the subtle details of the hardware and OS, students will need both CS 321 (Operating Systems) and its prerequisite CS 301 (Assembly Language).

Calendar

Last day to drop: February 3. Spring break: March 11-19. Last day to withdraw: March 24. Midterm exam will be held at 1:00pm on Wednesday, March 8. Final exam will be held at 1pm on Wednesday, May 10.

Student Resources

Information Security Resources: National Information Assurance Training and Education Center, Snort IDS, Security Focus, Bruce Schneier.

Academic Help: Google, Rasmuson Library, Academic Advising Center (509 Gruening, 474-6396), Math Lab (Chapman Room 305), English Writing Center (801 Gruening Bldg, 478-5246).

Grading

Your work will be evaluated on correctness, rationale, and insight, not on successful regurgitation of random trivia. Grades for each assignment and test may be curved. Your grade is then computed based on four categories of work:

1. **HW:** Homeworks and machine problems, to be distributed through the semester.
2. **PROJ:** A substantial software development project related to intrusion detection, together with a short presentation of your results. Example projects: build a cryptographically secure filesystem change monitoring system; build an OS kernel integrity checking tool; build a self-monitoring system for an installed binary; build a secure network message passing protocol.
3. **MT:** Midterm Exam.
4. **FINAL:** Final Exam (comprehensive).

The final score is then calculated as:

TOTAL = 20% **HW** + 25% **PROJ** + 25% **MT** + 30% **FINAL**

Letter grades are then assigned at the usual 90/80/70 (etc) cutoffs. At my discretion, I may round your grade up if it is

near a grading boundary.

Homeworks are due by 5pm on the day they are due. Late homeworks will receive no credit. At my discretion, I may allow late assignments without penalty when due to circum stances beyond your control. Major assignments that are slightly late may be accepted at a 50% grade penalty (e.g., on-time grade: 86%; late grade: 43%). Everything you turn in must be your own work--violations of the UAF Honor code will result in a minimum penalty equal to THAT ENTIRE SECTION OF YOUR GRADE (e.g., one plagarized homework question will negate a perfect grade on all homeworks). However, even substantial reuse of other people's work is fine (and not plagarism) if it is clearly cited; you'll be graded on what you've added to others' work. Group work on substantial assignments (not homeworks, not tests) is acceptable if you clearly label who did what work; but I do expect a two-person group project to represent twice as much work as a one-person project. Department policy does not allow tests to be taken early; but in extraordinary circumstances may be taken late.

The homeworks, tests, and projects will almost all include extra work or requirements for students enrolled in the graduate section, CS 693.

Course Outline (Tentative)

Overview

- Necessity of Intrusion Detection.
- The history of Intrusion Detection.
- The difficulty of Intrusion Detection: you can't trust anything a compromised machine says or does.
- Commonly exploited intrusion sources:
  - Buffer overflow/invalid input
  - Race conditions
  - Excess priviledge
- Common post-intrusion changes:
  - New network services
  - Rootkits, spyware, trojans
  - Backdoored executables & password collection

Intrusion Detection Basics

- Cryptography & trust
  - Authentication vs. Encryption
  - Cryptographic hash
  - Public-key encryption basics
- Secure network protocols
  - SSL, SSH, SCP, SFTP
  - Secure protocol design
- Account security & permissions
  - Passwords
  - Encrypted keys
  - Biometrics
- Filesystem security & permissions
  - The beauty of read-only media

Intrusion Detection Techniques

- Log & registry analysis
  - What is "normal"?
  - Automating analysis
- Network traffic analysis
- Network exposure testing (self portscan)
- User-level instrumentation tools:
  - Cryptographic file checksums
  - Secure/remote logging
- Kernel-level instrumentation:
  - System call interception
  - Filesystem access logging
  - Process separation & jails
  - Virtual machines

Intrusion Response

- Legal issues:
  - Pre-intrusion privacy concerns
  - Evidence collection (disk imagers, network traceback)
  - Countermeasures & the dangers of automated response or vigilante justice
- Evaluation of intrusion scope
- Issues for production machines/servers
  - Data backups: they may be bad too.
- Remove-bad-files or format-and-reinstall?
- Post-intrusion recovery