CS 261: Computer Security

Overview

CS 261 is a graduate class on computer security offered in the Fall 2000 semester.

Instructor: David Wagner.
Time: 9:00--10:30am, Mondays and Wednesdays.
Location: 320 Soda.
Prerequisites: CS 162 or equivalent. Familiarity with basic concepts in operating systems and networking.
Web page: `http://www.cs.berkeley.edu/~daw/teaching/cs261-f00/`

Course description

CS261: Security in Computer Systems. Prerequisite: CS162. Graduate survey of modern topics in computer security, including: protection, access control, distributed access control, Unix security, applied cryptography, network security, firewalls, secure coding practices, safe languages, mobile code, and case studies from real-world systems. May also cover cryptographic protocols, privacy and anonymity, and/or other topics as time permits. Term paper or project required. Three hours of lecture per week. (3 units)

Course topics

An approximate list of course topics (subject to change; as time permits):

Basic concepts
: Trust, trusted computing base, trusted path, transitive trust. Reference monitors. Policy vs. mechanism. Assurance. Lessons from the Orange Book.

Access control
: Authorization, policy, access matrix. Subjects and objects. ACLs, capabilities. Rings, lattices. Revocation. Groups. The role of crypto. Distributed access control. Mandatory vs. discretionary access control, compartmentalization, covert channels.

Protection
: Traditional OS centralized protection: address spaces, uids, resource management. The Unix security model: file permissions, the super-user, setuid programs, system calls, password security. How networks change the problem space.

Secure coding
: Design principles: code structure, least privilege, small security kernels, small interfaces. Tools: language support, type-safe languages, static checking. Common vulnerabilities: buffer overruns, setuid programs, the confused deputy, race conditions, improper canonicalization.

Cryptography
: Symmetric key, public key, certificates. Choosing an algorithm. Protocols. Integrity, authenticity confidentiality, availability. Non-repudiation.

Network security

TCP/IP. Attacks on network protocols: address spoofing, hijacking, source routing, SYN floods, smurfing, etc. DNS attacks, routing vulnerabilities. Attacks on network daemons. The Internet Worm. TCP wrappers. Intrusion detection.

Firewalls

Philosophy, benefits. Styles: packet filter, application proxying, stateful inspection. Performance, scalability. Fail-safety, assurance. Techniques. Do's and don'ts.

Confining untrusted code

Motivation: the mobile code problem, implementing least privilege. Mechanisms: signed code, interpreted code, software fault isolation, proof-carrying code, virtualization, extensible reference monitors. Practical experience: ActiveX, Java, Javascript.

Case studies

Kerberos. PGP and the web of trust. SSL and centralized certification authorities. SSH. IPSEC. Cellphones. Therac-25. Practical issues: risk management, key management, smartcards, copy protection systems, social engineering.

Extra topics

Privacy: Anonymity and traffic analysis; remailers and rewebbers; practical experience. Cryptographic protocols: protocol failures, design principles; logics of authentication; Formal methods. Others as time permits and according to student interest.

Project presentations

Grading

Class project: 45%
Problem sets: 40%
Paper summaries and class discussion: 15%

There will be no final or midterm.

Projects

There will be a term project. You will do independent research in small groups (e.g., teams of 2--3). Projects may cover any topic of interest in systems security, interpreted broadly (it need not be a topic discussed in class); ties with current research are encouraged. A conference-style report and a project presentation on your results will be due at the end of the semester.

Readings

There is no required textbook. All reading will be from papers. Whenever possible, handouts and papers will be placed online on the web page; papers not available online will be handed out in class. A schedule of assigned readings is available online.

You will be required to write a *very brief* summary of each paper you read, listing the two most important contributions of the paper and its most significant flaw, to be handed in at the beginning of the class when the reading is due.