CS 261, Fall 2008 Computer Security

Instructor:
David Wagner (daw@cs, 629 Soda Hall, 642-2758)

Lectures:
Tu-Th, 11:00-12:30, 310 Soda

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)

Prerequisites: CS 162 or equivalent. Familiarity with basic concepts in operating systems and networking.

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. Object capabilities.

Cryptography

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

Intro to Network security

TCP/IP. Attacks on network protocols: address spoofing, hijacking, DNS attacks, routing vulnerabilities. Firewalls: packet filtering, application proxying.

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. Phishing and cybercrime. 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.

Grading

Class project: 40%
Problem sets: 35%
Scribe notes: 15%
Paper summaries and class discussion: 10%

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. You will present your work at a poster session and prepare a conference-style paper describing your work.

You are encouraged to start thinking of topics of interest early. Be ambitious! I expect that the best papers will probably lead to publication (with some extra work).

Problem Sets

There will be approximately two to four homework assignments throughout the semester, to appear on the course webpage as they are assigned.

Turn in your homeworks on paper at the beginning of class on the day they are due. Due dates will be enforced strictly. Late homeworks will not be accepted.

Work on your own when doing homeworks. You may use any source you like (including other papers or textbooks), but if you use any source not discussed in class, you must cite it.

Scribe notes

You will be expected to write scribe notes for one lecture. Email me an PDF file with your scribe notes within one week after the lecture you are assigned to scribe.

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 below.

You will be required to write a brief summary of each paper you read. Submit your summary, on paper, before the beginning of the class when the reading is due. Your summary should list:

- the one or two or three most significant new insights you took away from the paper; and,
- the paper's two or three most significant flaws or weaknesses (e.g., methodology, vulnerabilities, relevance), or how the paper could be improved; and,
- the topics you would most like to see discussed in class, if any.

Your summary does not need to be formal (you may use bullet lists, incomplete sentences, etc.), and it may be brief, but it should reflect a thoughtful critical assessment of the paper.

Ethics

From time to time, we may discuss vulnerabilities in widely-deployed computer systems. This is *not* intended as an invitation to go exploit those vulnerabilities without informed consent of all involved parties. If it is not clear where to draw the line, please talk to me first.

David Wagner, daw@cs.berkeley.edu, http://www.cs.berkeley.edu/~daw/.