Advanced Systems Security

Trent Jaeger
Systems and Internet Infrastructure Security (SIIS) Lab
Computer Science and Engineering Department
Pennsylvania State University

January 14, 2010

About Me

- *Trent Jaeger* (PhD, University of Michigan)
- Associate Professor, CSE -- after 9 years at IBM Research
- Research: Operating System Security
- Example Projects
  - L4 Microkernel -- minimal, high performance OS
  - Linux -- Open source, UNIX variant
  - Xen hypervisor -- Open source, virtual machine platform
- Office Hours: Tu 4-5, W 1-2, or by appointment
- Office: 346A IST Bldg
- Email: tjaeger@cse.psu.edu

Motivation

Security mechanisms and policies have been implemented at several system layers (app, OS, VM, network)

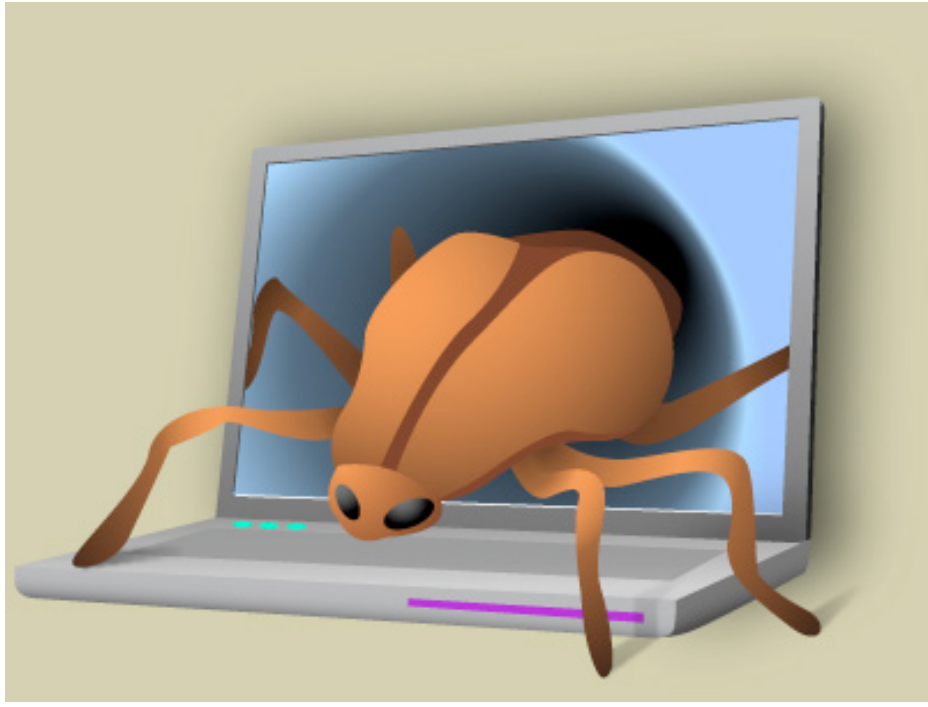
Are we now secure?

Current Security Problems

PENNSTATE 1855

Most current security problems are based on the failure of people to deploy hosts securely

Botnets
Rootkits
Web attacks: XSS, SQL Inject, …
Worms (Conficker)
Password Guessing
Buffer Overflows
Arbitrary App Flaws



Security State

SANS Top Security Risks
http://www.sans.org/top-cyber-security-risks/

- *Client-side software is unpatched (apps patched slower)*
- *Web servers are vulnerable (XSS are 80%)*
- *Application vulnerabilities exceed OS vulnerabilities*
- *Attacks on Mac systems (QuickTime)*
- *US is the major attack target (30:1)*
- *Still buffer (and heap) overflows*

We will study the structure of attacks on hosts and a general procedure for their prevention

Cross-site Scripting (XSS)

- Aim: *Get a client to run an attackers' code at higher privilege (Privilege Escalation)*
- Attack:
  - Attacker places content on trusted site
  - Client downloads content and attacks unpatched client program (e.g., media player)
  - Attacker can run as client user
  - Install reverse shell backdoor (outbound HTTPS)
  - Download local privilege escalation program (again unpatched client code)
  - Attack other machines – Windows domain controller

Security Mythology

- Claim: *All these problems were solved in Multics*
- Is this claim true?
- Why not just use it?
- What is necessary?
- By whom?
- Can we make it happen?

- Claim: *We are still trying to solve the same security problems since Multics*

Who Has a Role?

- We want to examine what all the interested parties do/do not want for security/function and what they can/cannot know to resolve conflicts
- Programmers (may be multiple groups)
- OS Distributors
- Administrators
- Users
- Service Providers
- Content Providers

This course….

- Is a systems course that teaches principles for building a secure system and techniques for implementing those principles
  - Caveat: We are still trying to figure out the latter
  - Topics: What makes a system secure (mechanisms and policies); Example implementations of such principles (at OS, VMM, and application); Tools to assist in such implementations; How do we put it together; Recent research in secure systems design

Background

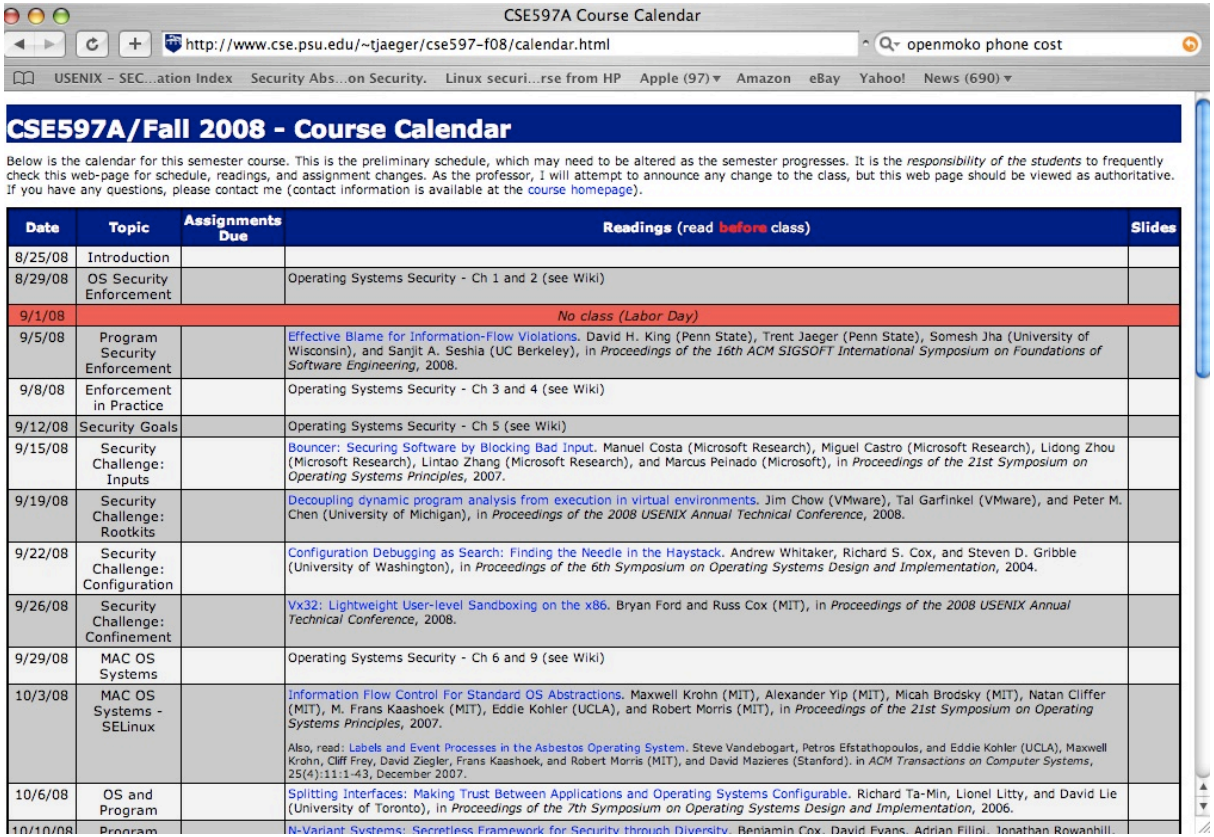
- Required:
  - CSE 543
- Expected:
  - Solid OS and PL background
- Additional:
  - Willingness to read
    - We are going to read a lot of systems security papers
  - Willingness to program
    - We are going to have some OS programming assignments (Linux)

Course Materials

- Website
  - http://www.cse.psu.edu/~tjaeger/cse544-s10/
  - Course assignments, slides, etc. will be placed here
    - Check back often -- I may change some of the papers/assignments
- Course Textbook
  - My book: *Operating Systems Security*
  - Augmented with research papers

Course Calendar

- The course calendar has all the details
- Links to online papers for readings
- Links to projects
- Please check the calendar frequently
  - it's the real-time state of the course



CSE597A Course Calendar

http://www.cse.psu.edu/~tjaeger/cse597-f08/calendar.html

openmoko phone cost

USENIX – SEC...ation Index | Security Abs...on Security. | Linux securi...rse from HP | Apple (97) | Amazon | eBay | Yahoo! | News (690)

CSE597A/Fall 2008 - Course Calendar

Below is the calendar for this semester course. This is the preliminary schedule, which may need to be altered as the semester progresses. It is the *responsibility of the students* to frequently check this web-page for schedule, readings, and assignment changes. As the professor, I will attempt to announce any change to the class, but this web page should be viewed as authoritative. If you have any questions, please contact me (contact information is available at the course homepage).

Date	Topic	Assignments Due	Readings (read before class)	Slides
8/25/08	Introduction			
8/29/08	OS Security Enforcement		Operating Systems Security - Ch 1 and 2 (see Wiki)	
9/1/08	*No class (Labor Day)*			
9/5/08	Program Security Enforcement		Effective Blame for Information-Flow Violations. David H. King (Penn State), Trent Jaeger (Penn State), Somesh Jha (University of Wisconsin), and Sanjit A. Seshia (UC Berkeley), in *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, 2008.	
9/8/08	Enforcement in Practice		Operating Systems Security - Ch 3 and 4 (see Wiki)	
9/12/08	Security Goals		Operating Systems Security - Ch 5 (see Wiki)	
9/15/08	Security Challenge: Inputs		Bouncer: Securing Software by Blocking Bad Input. Manuel Costa (Microsoft Research), Miguel Castro (Microsoft Research), Lidong Zhou (Microsoft Research), Lintao Zhang (Microsoft Research), and Marcus Peinado (Microsoft), in *Proceedings of the 21st Symposium on Operating Systems Principles*, 2007.	
9/19/08	Security Challenge: Rootkits		Decoupling dynamic program analysis from execution in virtual environments. Jim Chow (VMware), Tal Garfinkel (VMware), and Peter M. Chen (University of Michigan), in *Proceedings of the 2008 USENIX Annual Technical Conference*, 2008.	
9/22/08	Security Challenge: Configuration		Configuration Debugging as Search: Finding the Needle in the Haystack. Andrew Whitaker, Richard S. Cox, and Steven D. Gribble (University of Washington), in *Proceedings of the 6th Symposium on Operating Systems Design and Implementation*, 2004.	
9/26/08	Security Challenge: Confinement		Vx32: Lightweight User-level Sandboxing on the x86. Bryan Ford and Russ Cox (MIT), in *Proceedings of the 2008 USENIX Annual Technical Conference*, 2008.	
9/29/08	MAC OS Systems		Operating Systems Security - Ch 6 and 9 (see Wiki)	
10/3/08	MAC OS Systems - SELinux		Information Flow Control For Standard OS Abstractions. Maxwell Krohn (MIT), Alexander Yip (MIT), Micah Brodsky (MIT), Natan Cliffer (MIT), M. Frans Kaashoek (MIT), Eddie Kohler (UCLA), and Robert Morris (MIT), in *Proceedings of the 21st Symposium on Operating Systems Principles*, 2007. Also, read: Labels and Event Processes in the Asbestos Operating System. Steve Vandebogart, Petros Efstathopoulos, and Eddie Kohler (UCLA), Maxwell Krohn, Cliff Frey, David Ziegler, Frans Kaashoek, and Robert Morris (MIT), and David Mazieres (Stanford). in *ACM Transactions on Computer Systems*, 25(4):11:1-43, December 2007.	
10/6/08	OS and Program		Splitting Interfaces: Making Trust Between Applications and Operating Systems Configurable. Richard Ta-Min, Lionel Litty, and David Lie (University of Toronto), in *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*, 2006.	
10/10/08	Program		N-Variant Systems: Secretless Framework for Security through Diversity. Benjamin Cox, David Evans, Adrian Filipi, Jonathan Rowanhill,	

Course Mailing List

- Via ANGEL
  - Use with care
- I will send a test email
  - Please reply if you do not receive by Fr
  - May need to forward to your CSE account
- Can use to email me
  - Please use “544” in the subject

Grading

- Exams (50%)
  - Midterm (25%)
    - Take home – do the readings
  - Final (25%)
    - In class
- Projects (40%)
  - 4 programming projects
  - OS and Source code analysis
- Participation (10%)

Projects

- We are going to have four project deliverables
  - Per person
- Topics
  - Buffer Overflow
  - Linux Security Module
  - Security Policy
  - Source Code Analysis for Security
- C programming required
  - Kernel debugging also

Lateness Policy

- Assignments and project milestones are assessed a 20% per-day late penalty, up to a maximum of 4 days. Unless the problem is apocalyptic, don't give me excuses. Students with legitimate reasons who contact the professor before the deadline may apply for an extension.
- You decide what you turn in

Ethics Statement

- This course considers topics involving personal and public privacy and security. As part of this investigation we will cover technologies whose abuse may infringe on the rights of others. As an instructor, I rely on the ethical use of these technologies. Unethical use may include circumvention of existing security or privacy measurements for any purpose, or the dissemination, promotion, or exploitation of vulnerabilities of these services. Exceptions to these guidelines may occur in the process of reporting vulnerabilities through public and authoritative channels. Any activity outside the letter or spirit of these guidelines will be reported to the proper authorities and may result in dismissal from the class.
- When in doubt, please contact the instructor for advice. **Do not** undertake any action which could be perceived as technology misuse anywhere and/or under any circumstances unless you have received explicit permission from Professor Jaeger.

Road Map

- Introduction
  - 1. What is security? 2. What are the fundamental principles if secure execution?
- Designing for Security and Protection
  - 1. Experiences with Multics and UNIX/Windows
- Mandatory Access Control
  - 1. Policy Models 2. Lattice Models in Depth
- Systems Security Architectures
  - 1. Security Kernels 2. Secure UNIX Variants 3. Capability Systems 4. VM Systems
- Assurance
  - 1. Common Criteria 2. Program Analysis
- Practical System Integrity
  - 1. System Integrity Models 2. Decentralized Label Model 3. Data/Control Flow Integrity
- Special Topics
  - 1. Trustworthy Computing 2. Device Security 3. Storage Security 4. Web Security

Review

- Are we speaking the same language?
- General Terms
  - Principals/Subjects and Adversaries/Attackers
  - Trust Model
  - Threat Model
  - Security Model
- We will develop (semi-)formal models for each