CSE 598i
Verification Methods for Security

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

August 22, 2011

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: by appointment
- Office: 346A IST Bldg
- Email: tjaeger@cse.psu.edu

Motivation

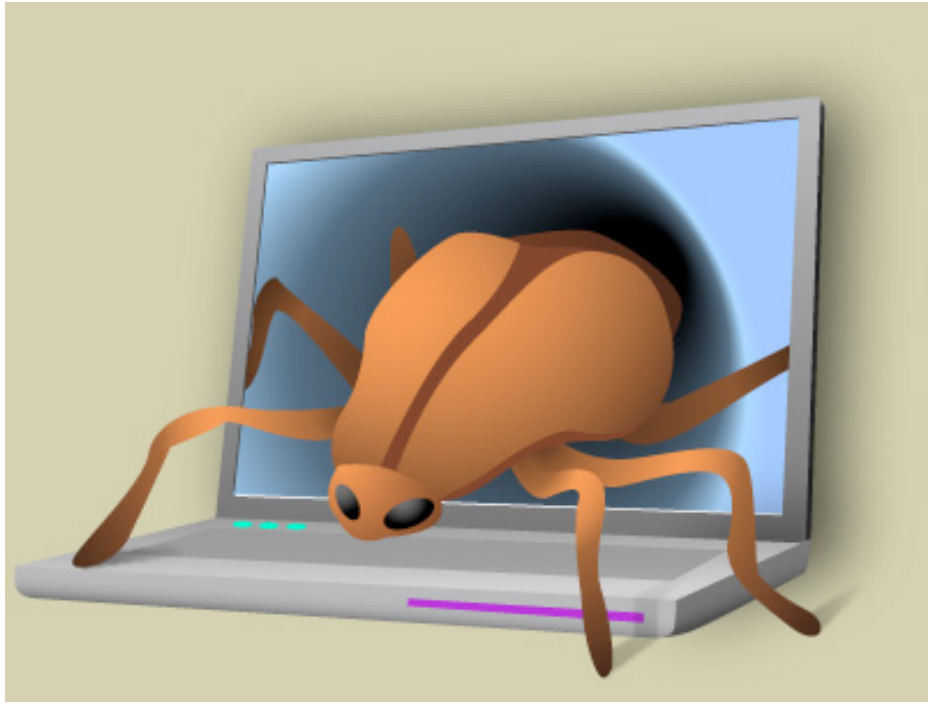
Security mechanisms and policies have been implemented at several system layers (app, OS, VM, network)

Are we now secure?

Current Security Problems

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

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



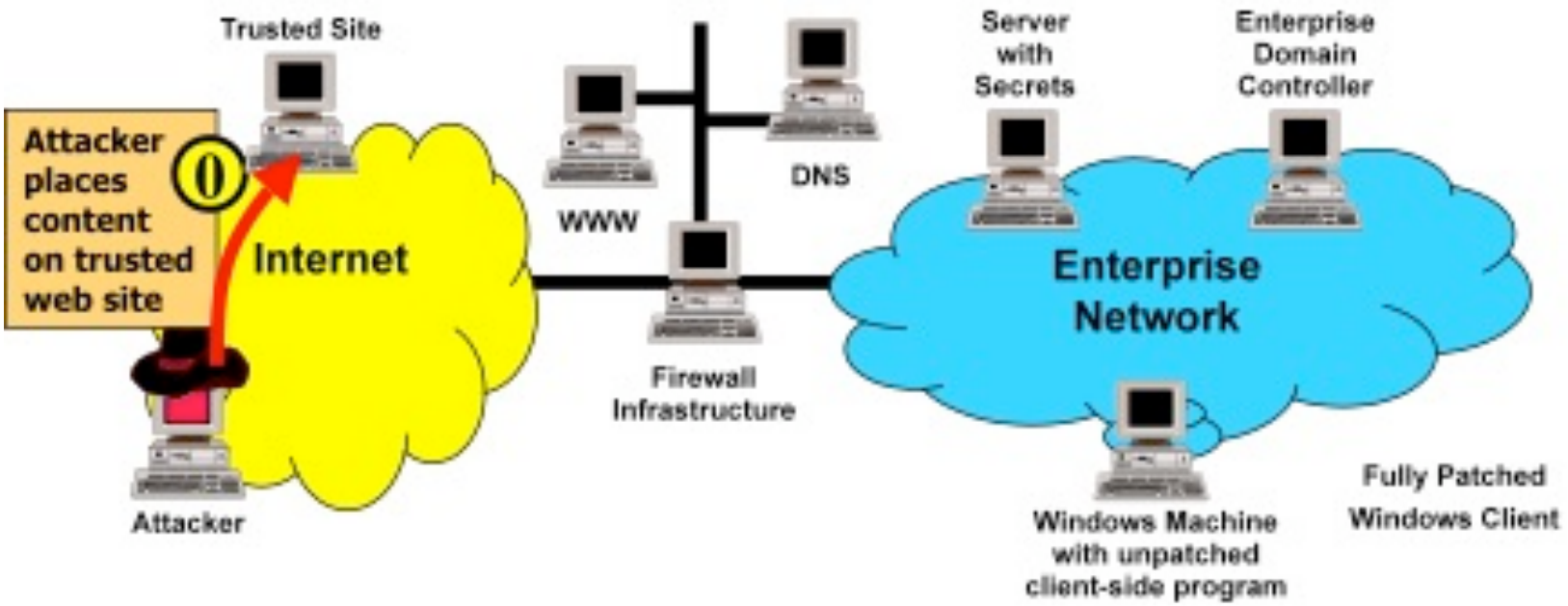
Security State

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

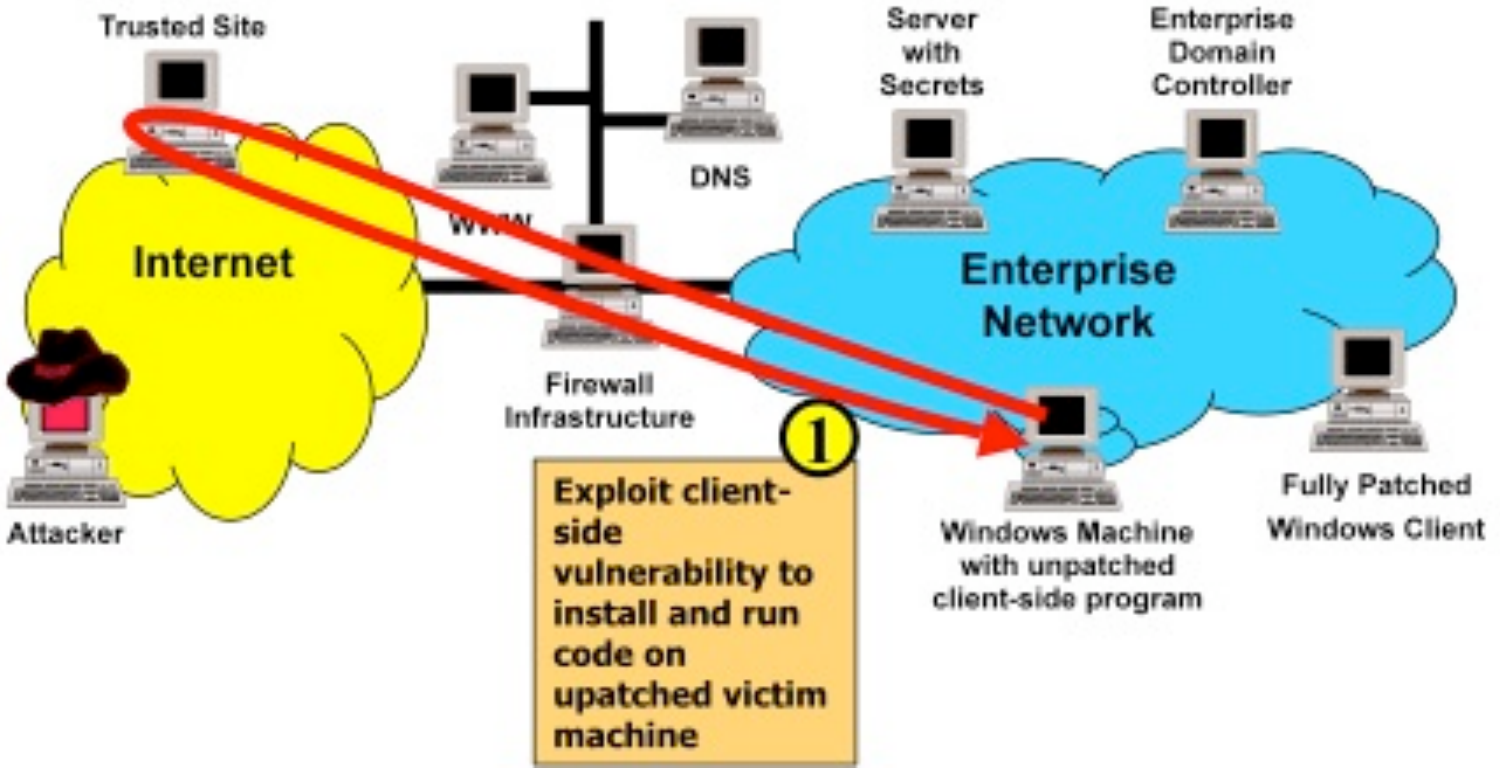
- *Client-side software is unpatched (apps patched slower)*
- *Web servers are vulnerable (XSS are 80%)*
- *Application vulnerabilities exceed OS vulnerabilities*
- *Attacks on Mac systems (QuickTime)*
- *US is the major attack target (30:1)*
- *Still buffer (and heap) overflows*

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

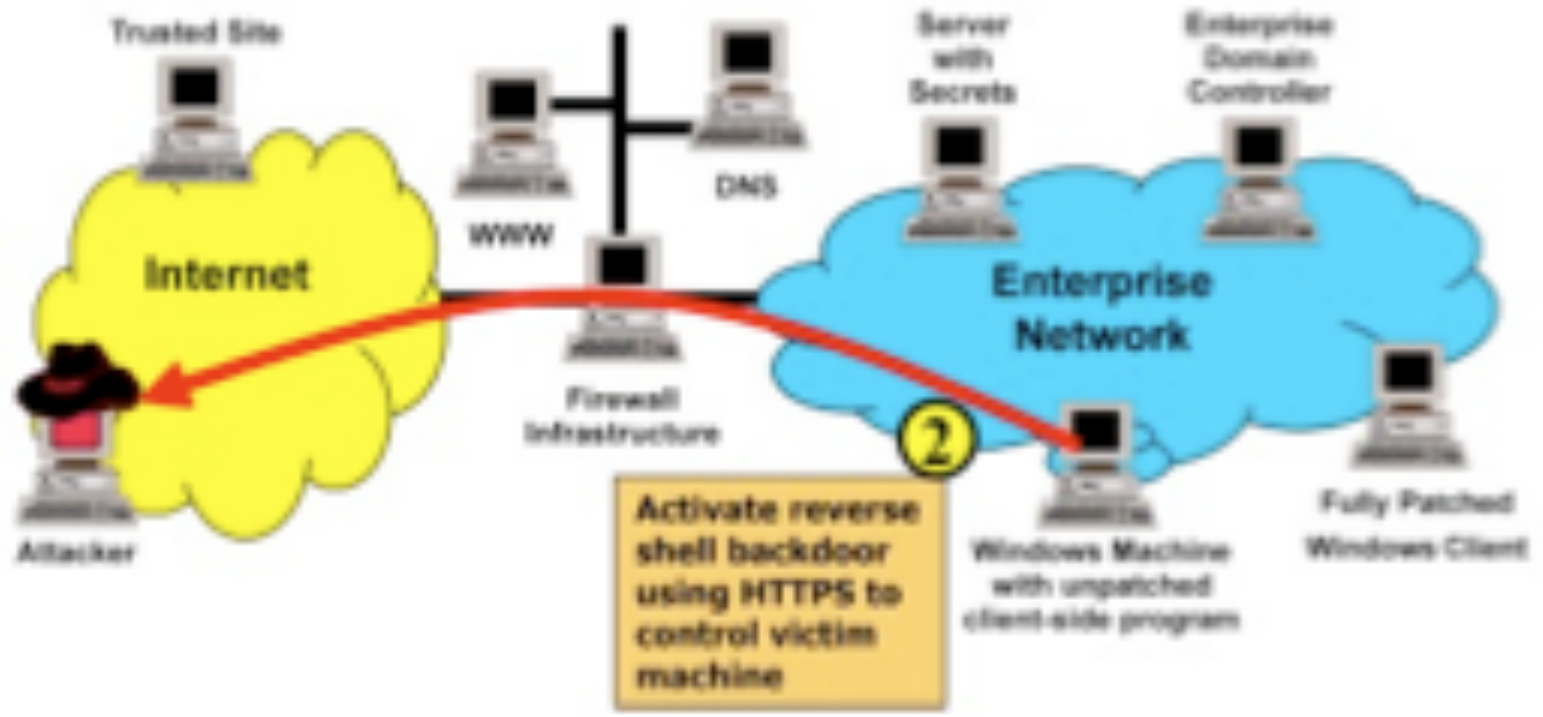
SANS Example



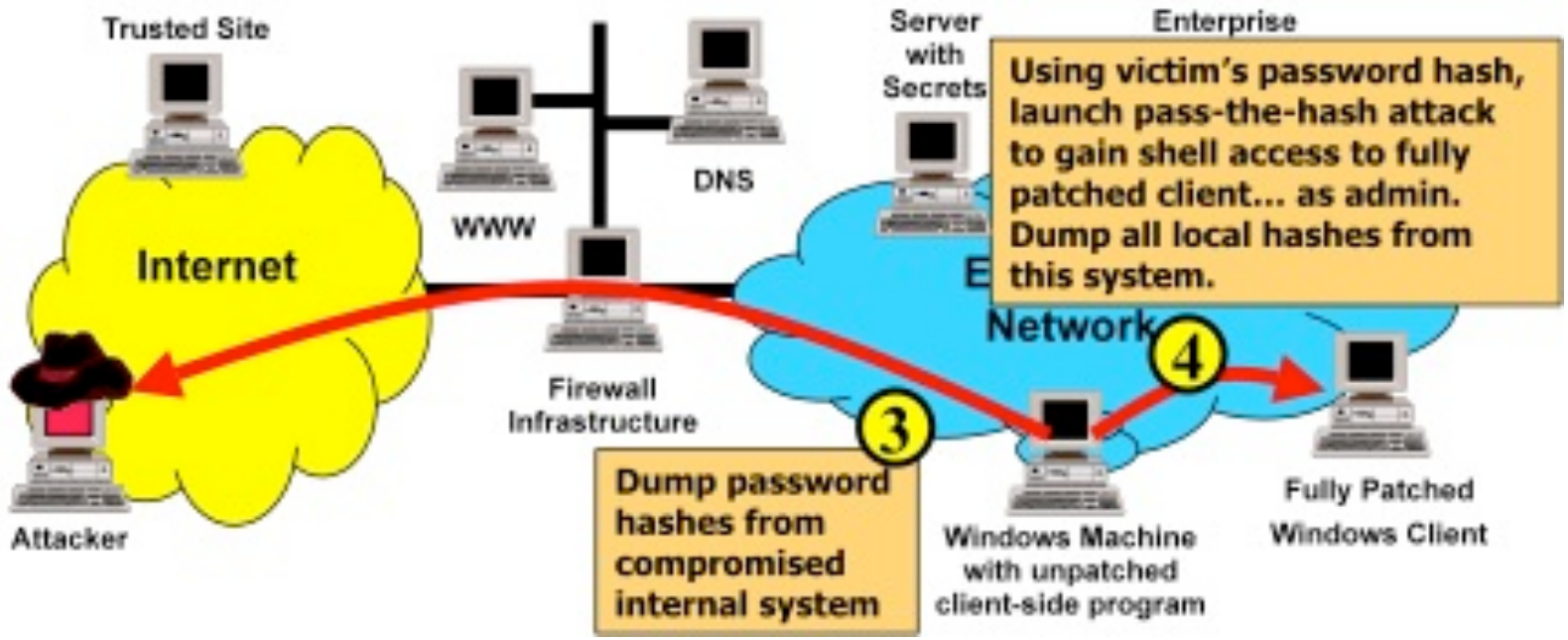
SANS Example



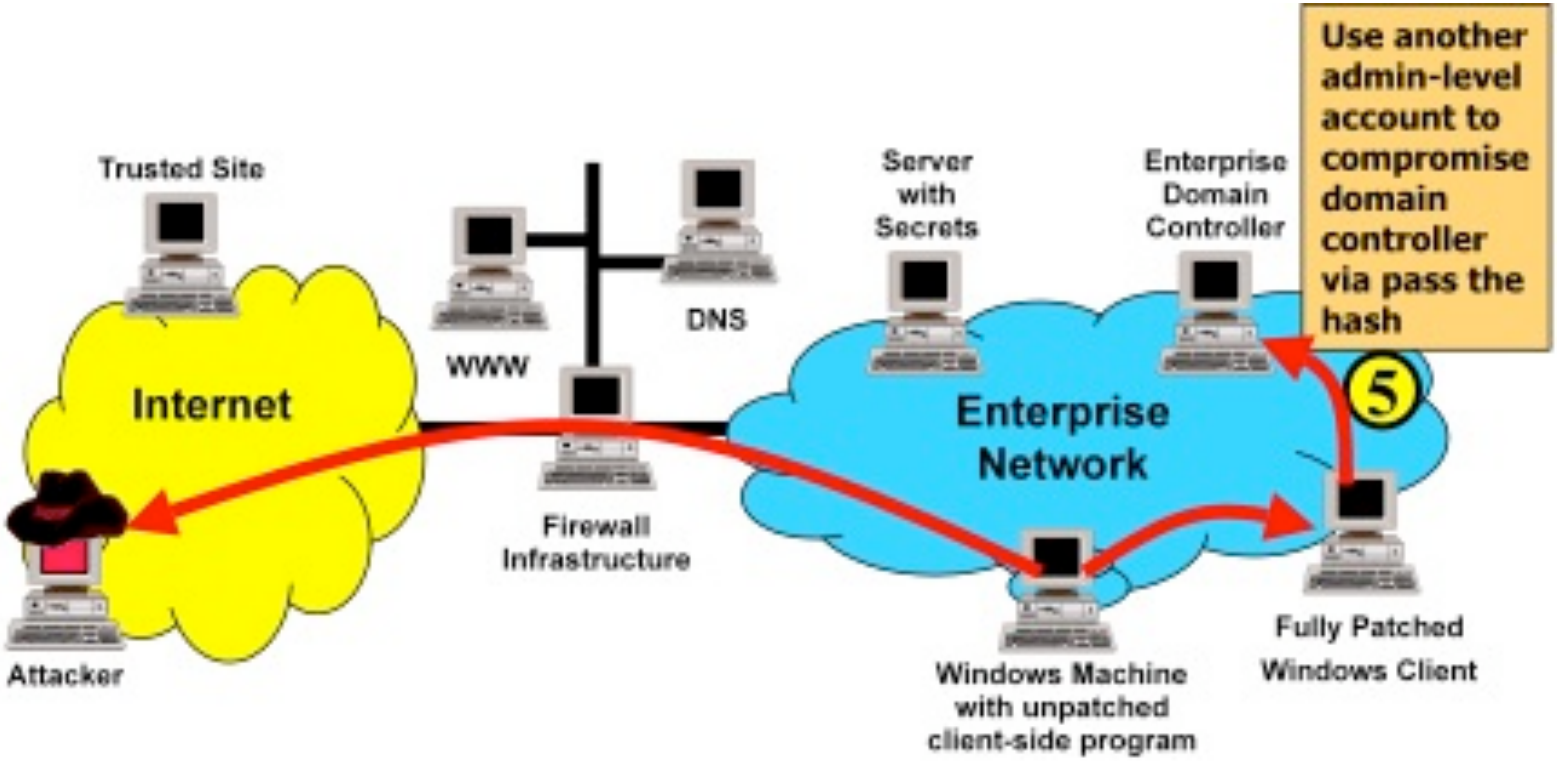
SANS Example



SANS Example

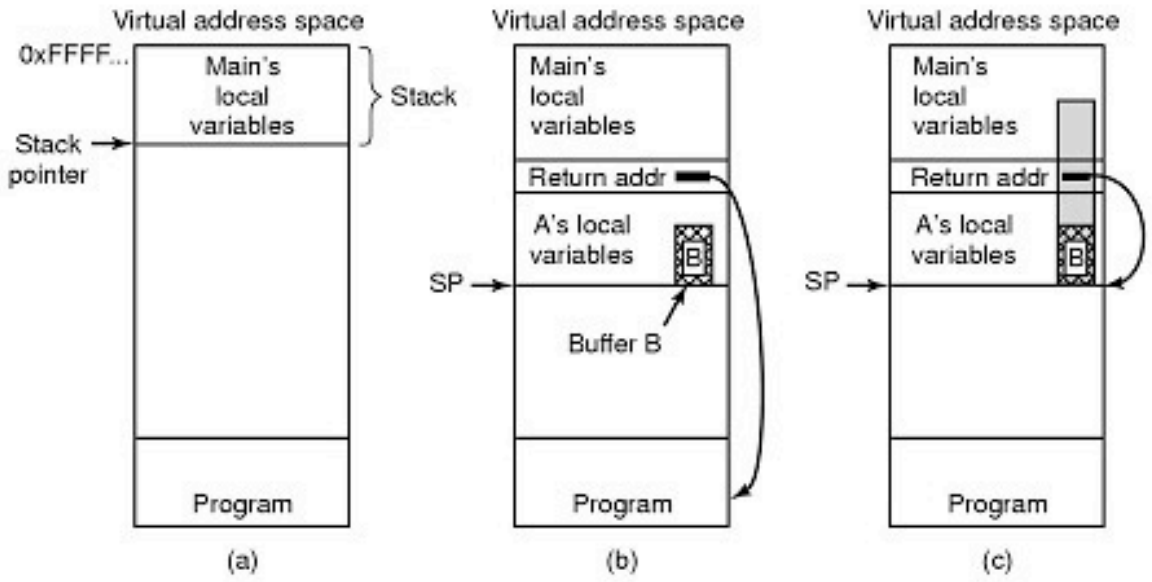


SANS Example



Causes?

Buffer Overflow



- (a) Situation when main program is running
- (b) After program *A* called
- (c) Buffer overflow shown in gray

Lec 19 Fig 1

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*

Answer?

- *Analysis Tools for systems and programs*
  - 1980s – 90s: formal verification methods
  - 2000s: Bug finding
  - 2010s: tools to find and fix security bugs?
  - Beyond??
- Problem: what bugs should be discovered?
- Problem: soundness and completeness
- Problem: how should analysis impact software development and system deployment?

Who Has a Role?

- Programmers (may be multiple groups)
- OS Distributors
- Administrators
- Users
- Service Providers
- Content Providers
- **Challenge**: Must consider the balance between function and security

This course….

- Is a software course that teaches principles and techniques for verifying security properties
  - Lots of techniques have been developed, but we need to figure out how to use/extend them to improve systems security
  - **Topics**: What should "secure" mean in systems? How to find violations of security in programs and systems? How to fix such violations of security automatically? How to make such techniques tractable and practical?

Background

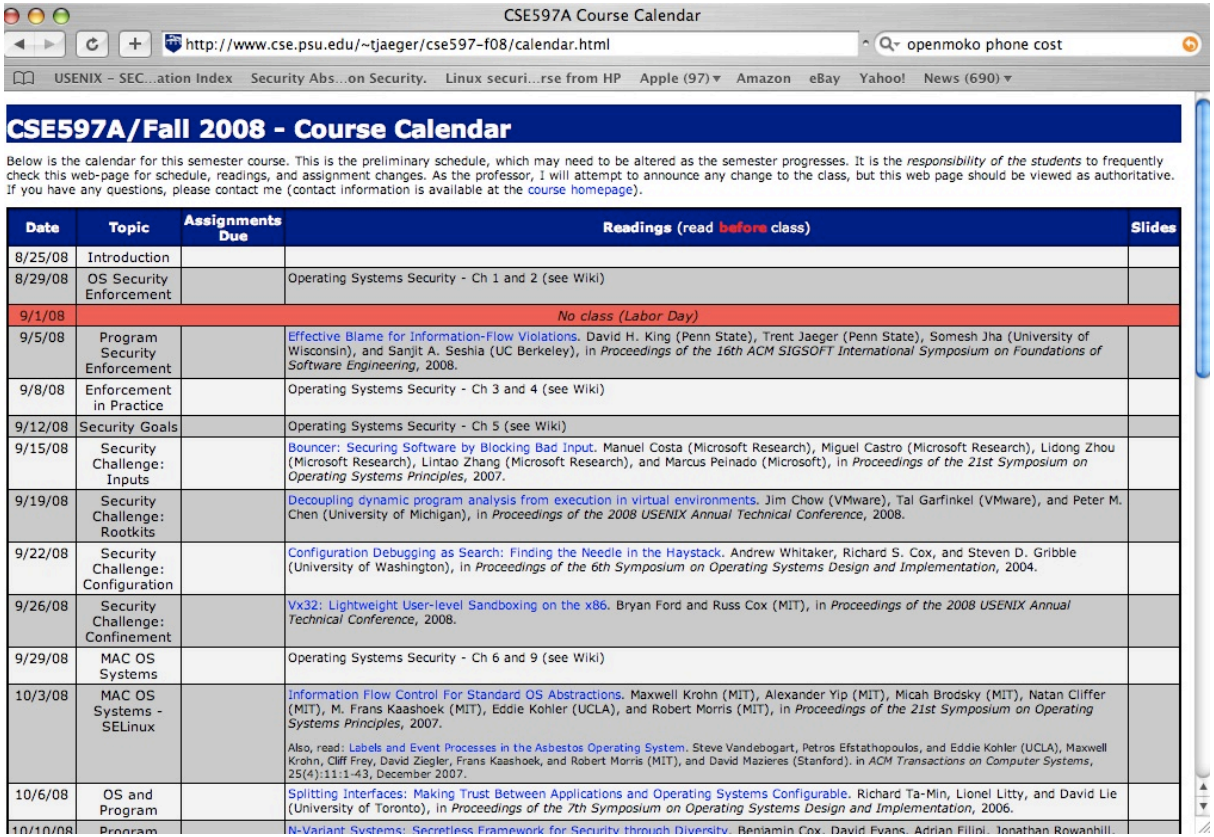
- Required:
  - CSE 543
- Expected:
  - Solid OS and PL background
- Additional:
  - Willingness to read
    - We are going to read a lot of papers on security and analysis techniques
  - Willingness to program
    - We are going to have some programming assignments

Course Materials

- Website
  - http://www.cse.psu.edu/~tjaeger/cse598-f11/
  - Course assignments, slides, etc. will be placed here
    - Check back often -- I may change some of the papers/assignments
- Readings
  - Book: *Analysis Techniques for 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/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 “598” in the subject

Grading

- Exams (55%)
  - Midterm (25%)
    - Take home – do the readings
  - Final (25%)
    - In class
- Projects (35%)
  - 2 programming projects
  - Final project
- Presentations (15%)

Projects

- We are going to have three project deliverables
  - Per person
- Topics
  - Security analysis for programs
  - Security analysis for systems (policies)
- Final Project
  - Per my approval

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
  - Current Attacks
  - System Security Basics
- Static Analysis Techniques
  - Foundations
  - Detecting Bugs in Programs and Systems Policies
  - Constraint Solving and Compiler Infrastructure
- More Advanced Problems
  - Namespaces
  - Attack Graphs
- More Advanced Analysis Topics
  - Summary Functions, Runtime Analysis, Put It Together

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