CSE 544
Advanced Systems Security

Trent Jaeger
Systems and Internet Infrastructure Security (SIIS) Lab
Computer Science and Engineering Department
Pennsylvania State University
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?
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
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 that content attacks an 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-s13/](http://www.cse.psu.edu/~tjaeger/cse544-s13/)
  ‣ 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
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 (55%)
  - Midterm (25%)
    - Take home – do the readings
  - Final (30%)
    - In class

- Projects (35%)
  - One programming project
  - Course Project

- Participation (10%)
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 of 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

• Program Security
  ‣ 1. Safe Execution 2. Program Analysis and Testing

• Practical System Integrity
  ‣ 1. System Integrity Models 2. Decentralized Label Model 3. Data/Control Flow Integrity

• Special Topics
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