Introduction
What is this Course?

- Network security
- Mostly not true — primary focus is security of networked applications
- Some true network security — protect the network infrastructure
Secure network protocol design
Using cryptography (COMS W4261 not a prerequisite!)
The role of correct software
How to Think About Insecurity...

- The bad guys don’t follow the rules
- To understand how to secure a system, you have to understand what sort of attacks are possible
- Note that that is *not* the same as actually launching them...
Course Structure

- Lectures
- Approximately five homework assignments, all with programming and non-programming components
- Midterm, final
Prerequisites

- COMS W4119 — Networking
  - Network layers
  - Basics of TCP/IP
  - Difference between IP, ICMP, TCP, and UDP
  - Port numbers and sequences numbers
  - Some understanding of the TCP flags

- COMS W3137 or W3139
- Understand how to use “make”, the compiler, etc.
- C or Java
Grading

Midterm 20%
Final 30%
Homeworks 50%

Exams will be open book.
Yes, I curve.


Occasional papers
Logistics

- For grading issues, approach the TA within two weeks; if you don’t receive a satisfactory answer, contact me.
- For issues relating to *this class*, email smb+4180@cs. . .
- That lets me auto-sort class-related mail and keep better track of things
- My office hours are posted; I try to note (too frequent) changes because of my travel schedule
Talking to Me

- Drop by, just to talk
- You don’t need to be in trouble to talk with me...
- If my office door is open, c’mon in
- But — I travel too much
TAs

- Elli Androulaki <elli@cs...>
- TBA
Lectures

- I prepare slides for each class, and upload them shortly before class time.
- Slides (and other information) is uploaded both to Courseworks and to my web page.
- Well, occasionally they’re uploaded shortly after class...
- Because the class is being recorded for CVN, you’ll be able to watch any lectures you’ve missed.
- General access to the videos starts after the add/drop period ends.
Homeworks

- A lot of it...
- As noted, approximately five homework assignments
- Homeworks are designed for practice, teaching, and evaluation
- Homeworks must be submitted electronically by the start of class
- Homeworks received later that day lose 5%, the next day 10%, two days late 20%, three days late 30%; after that, zero credit
- Exceptions granted only for *unforeseeable* events. Workload, day job, etc., are quite foreseeable.
All programming assignments must be done in C or Java.
Assignments will involve socket programming and use of cryptographic libraries — see HW0.
All inputs must be checked for validity and proper values and lengths — bugs are the major source of security problems.
Homework 0

- Simple socket exercise
- Not collected, not graded, completely optional
- But — it will be a useful base for another assignment
- It’s also a refresher exercise for you on socket programming
Co-operation versus Dishonesty

- Discussing homework with others is encouraged.
- All programs and written material must be individual work unless otherwise instructed.
- Please use appropriate file permission mechanisms to protect your homework. (Looking at other people’s work is not allowed.)
- Zero tolerance for cheating or “outsourced homework”
- See the department’s academic honesty policy: http://www.cs.columbia.edu/education/honesty. You are responsible for following it.
Taking a computer security class is *not* an excuse for hacking

“Hacking” is any form of unauthorized access, including exceeding authorized permissions

The fact that a file or computer is not properly protected is no excuse for unauthorized access

*If* the owner of a resource invites you to attack it, such use is authorized

For more details, see


Absolutely no Trojan horses, back doors, or other malicious code in homework assignments

No, I’m not joking
Responsibility

- You’re all adults
- You’re all responsible for your own actions
- If there’s something missing, you have to tell me
Practical Focus

- This is not a pure academic-style OS course
- You’ll be experimenting with real security holes
- A lot of (in)security is about doing the unexpected
- The ability to “think sideways” is a big advantage
The CLIC Lab

- All programs *must* run on the CLIC machines
- Programs that don’t compile *on those machines* receive zero credit
- You need a CS account to use CLIC; see https://www.cs.columbia.edu/~crf/accounts/
- Some of the CLIC machines are for in-person use; others can only be accessed remotely
- New policy: no food or drink in the CLIC lab
Network Security

Introduction

Administrivia

Network Security
Goals
Dichotomy
Anarchic Networks
Bellovin’s Laws of Networking
Benign Failures
Trust Nothing
Unproductive Attitudes
Better Attitudes
Network Security Tools
Protocol Design
Buggy Software

Course Outline
Goals

- Usual security trinity: confidentiality, integrity, availability
- Must ensure these in two domains: over-the-wire and on the host (for network-connected applications)
- Strategies are very different!
Dichotomy

- The host is (or can be) well-controlled
- There are well-developed authentication and authorization models
- There is a strong notion of “privileged” state, as well as what programs can use it
- None of that is true for the network
Anarchic Networks

- More or less anyone can (and does) connect to the network
- Connectivity can only be controlled in very small, well-regulated environments, and maybe not even then
- Different operating systems have different — or no — notions of user IDs and privileges
- As a consequence, notions of privilege are lacking
Bellovin’s Laws of Networking

1. Networks interconnect
2. Networks *always* interconnect
3. Interconnections happen at the edges, not the center
Benign Failures

- On top of all that, most network failures are benign.
- You have to program allowing for such failures: data corruption, timeouts, dead hosts, routing problems, etc.
- Rule of thumb: anything that can happen by accident can happen by malice — only more so.
A host can trust *nothing* that comes over the wire

Any desired protections have to be supplied explicitly

Perhaps there’s a middleware layer supplying the protection — but such middleware is based on the same principles
Unproductive Attitudes

- “Why would anyone ever do *that*?”
- “That attack is too complicated”
- “No one knows how this system works, so they can’t attack it”
Better Attitudes

- “Programming Satan’s Computer” (Ross Anderson)
- “Assume that serial number 1 of any device is delivered to the enemy
- “You hand your packets to the enemy to deliver; you receive all incoming packets from the enemy
Network Security Tools

- Cryptography
- Network-based access control (firewalls and more)
- Monitoring
- Paranoid design
Protocol Design

- Leave room for crypto and authentication
- Make sure all sensitive fields are protectable
- Make authentication bilateral
- Figure out the proper authorization
- Defend against eavesdropping, modification, deletion, replay, and combinations thereof
Most network security holes are due to buggy code

A buggy network-connected program is an insecure one

Correct coding counts for a lot
Course Outline
Introduction

- Attacks and threats
- Cryptography overview
- Network authentication and key management
- Kerberos
- SSL
Applications

- Web security
- Email security and phishing
- Network storage
- Secure shell
Lower Layers

- IPsec
- Firewalls
- Wireless
- Protocol design
Information

- Intrusion Detection
- Network scans
- Privacy
Availability

- Worms
- Denial of service
- Network infrastructure