What is this Course about?

- Network & Computer (in)security
- Network security — protect the network infrastructure, and secure the end-to-end communications
- Not entirely true — we also focus on security of networked applications

Why Network Security?

- Touches every aspect of network and system design and implementation
- Different mentality from other disciplines
  - “Does it work?” vs “Can it be broken?”
  - “Is the fix going to break something else?”
- Learn to think differently :-)

Importance of network security
How to Think About Insecurity

Course Objectives:

- Network Security
- Network (in)Security
- Course Outline

Administrivia

Course Overview
What is this Course about?
Why Network Security?
Importance of network security
How to Think About Insecurity
Course Objectives
Administrivia
Network Security
Network (in)Security
Course Outline
Importance of network security

- Increasingly large deployments of networked computers
- Sensitive information/resources are coming online
- Personal information
- Financial services
- Military operations
- Critical Infrastructure
- Enormous number of users, vast amount of money
- Cyber-attacks can cause significant economic damage

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 Objectives

- Learn how to design secure networked systems
- Quantify the cost and tradeoffs of security
- Determine where to apply/use cryptography (Cryptography not a prerequisite!)
- Appreciate the role of correct software
- Prevent?/Mitigate/Limit the security threats that step bad software
- Get hands-on knowledge practicing on real systems in the lab!

Administrivia
### Course Location and Time
- Always check the page website for new material: [http://ise.gmu.edu/~astavrou/isa656_F07.html](http://ise.gmu.edu/~astavrou/isa656_F07.html)
- Time: Tuesday 7:20 pm - 10:00pm
- Room: Robinson Hall A, room A247
- Lab: Science and Technology I 128
- Lab Meeting: Scheduled the same time as the class, usually every third lecture (you will be notified in advance)
- We will always meet at Robinson Hall A, room A247 and then proceed to labs

### Course Structure
- Lectures and Laboratory Sessions
- Approximately five homework assignments, all with programming and non-programming components
- Group Project or Midterm and a Final

### Prerequisites
- CS 555, or General 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
- ISA 562 or understanding of network protocols
- Understand how to use “make”, the compiler, etc.
- Programming in either C or Java

### Readings
- Research papers and reference manuals (RFCs etc.) (Provided on the class web site)
Grading

Midterm/Project  20%
Final  25%
Homeworks  50%
Class Participation  5%

In addition: extra credit assignments (why?)

Exams will be open book having part of the exam in the lab.

Office Hours & TAs

Instructor: Angelos Stavrou <astavrou@gmu.edu>
Office: 441 Science & Technology II
Hours: Monday 7 - 9pm & by appointment

TA: Ahmed K Alazzawe <aalazza1@gmu.edu>
Office: Adjunct office, Science & Technology II
Hours: Friday 7 - 9pm & by appointment

Grading 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 astavrou@ise.gmu.edu . . .
■ The TA should be your first contact point but you can also contact me with any questions or problems related to the class (or security in general).

Contacting Me

■ You don’t need to be in trouble to talk with me . . .
■ You can always arrange an appointment with me via email
■ We will also have Q&A sessions outside the class hours
■ But — I also travel to conferences...
Class & Lab Lectures

- I will prepare slides for each class, and upload them on the web site ahead of time (usually 2-3 weeks)
- Well, occasionally they’re uploaded shortly before class...
- For the Laboratory Sessions, you need to come prepared (read the material posted on the web) before the lab starts
- If you miss a class make sure that you read the lecture notes and come see us at our office hours

Homeworks

- A lot of it...
- As noted, approximately five homework assignments
- Homeworaks 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.

Programming Assignments

- 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 (will be posted online)
- 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
- Looking or Copying other people’s work is not allowed
- Zero tolerance for cheating or “outsourced homework”
- See the University academic honesty policy: http://www.gmu.edu/catalog/apolicies/#Anchor12. You are responsible for following it
- ALWAYS reference your source of information

The Ethics of Security

- 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
- No, I’m not joking

Responsibility

- You’re all adults
- You’re all responsible for your own actions
- Ask the TA or me if you are in doubt!

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 Security Lab

- We would like you to bring with you a USB key of at least 512MB
- As an alternative, you can bring your own laptop
- If we are more than 30, we will split into two groups
- No food or drink in the Security lab

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!

Differences from systems security

- Attacks can come from anywhere, at any time
- Highly automated (scripts)
- Physical security measures are inadequate
- Wide variety of applications, services, protocols
- Complexity
- Different constraints, assumptions, goals
- No single "authority"/administrator
- Somehow at odds with concept of networking
Network Security: A layered approach

Type of security mechanisms

- Pro-active try to keep the bad guys out
  - Passwords
  - Smartcards
  - Encrypted login protocols
  - Armed Marines
- Reactive mechanisms try to detect and contain an attack
  - Intrusion detection
  - DoS push-back
  - Flood the enemy
  - Attack using physical forces

Network Stack

- Application Layer: SSH, passwords
- Transport Layer: SSL
- Network Layer: I Pike, Link encryptors
- Link Layer: Pressurized cables, guards
- Physical Layer

Security-aware System Design

- Cost/benefit tradeoffs
- Threat model
- Trust model
- Available mechanisms
- Security is not only cryptography
- Security often conflicts with other goals: Fault tolerance, debugging & monitoring, sharing, etc.

Reactive mechanisms - problems

- No "strike-back" mechanisms widely in use
- Air Force Caller-ID program
- RIAA anti-P2P work
- It involves legal, moral, and practical issues
Failures of security mechanisms

- Failures of security mechanisms
- Failure to understand the threat model
- Failure to understand what a mechanism protects against
- No (or wrong) mechanism/tool used
- Bad design
- Implementation fault
- Mis-configuration

More failures...

- Bad user interface
- Complexity (inherent in "systems")
- Emergent properties vs. bugs
- Theory vs. practical implementation

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 userIDs and privileges
- As a consequence, notions of privilege are lacking

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

Observations about Networks

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

Trust Nothing

- A host can trust nothing that comes over the wire
- Any desired protections have to be supplied explicitly
- Perhaps there’s a middle-ware layer supplying the protection — but such middle-ware 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

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

Protocol Design

- Leave room for crypto and authentication
- Make sure all sensitive fields are protected
- Make authentication bilateral
- Figure out the proper authorization
- Defend against eavesdropping, modification, deletion, replay, and combinations thereof
Buggy Software

■ Most network security holes are due to buggy code
■ A buggy network-connected program is an insecure one
■ Correct coding counts for a lot

Network Availability

■ Attacks and threats
■ Firewalls & VPNs
■ Intrusion Detection
■ Network scans
■ Worms
■ Denial of service
■ Network infrastructure Design

Authentication & Secure Protocols

■ Cryptography overview
■ Network authentication and key management
■ Kerberos
■ SSL
■ IPsec
■ Protocol design
Applications

- Web security
- Email security and phishing
- Voice over IP (VoIP) security
- Network storage
- Trust Management