Internet Security

Prof. Anja Feldmann, Ph.D.
anja@net.t-labs.tu-berlin.de
http://www.net.t-labs.tu-berlin.de/

Prof. Dr. Jean-Pierre Seifert
jpseifert@sec.t-labs.tu-berlin.de
http://www.sec.t-labs.tu-berlin.de/

General information

- Area: BKS – Hauptstudium Vertiefer
  - Will be integrated into a Module system of SECT and INET
- Time
  - Thursday: 14:00 – 16:00
- Room
  - MA 043
- Language
  - English (questions can be asked in German!)
- Web site
  - http://www.net.t-labs.tu-berlin.de/teaching/ss09/is_ss09/
- Mailing list
  - see Web page
General information

- Exam
  - For those that need it 😊
  - Oral or written exam after semester end (depends on # of participants)

- Prerequisite: some knowledge of
  - How the Internet works
  - How operating systems work
  - Little bit of undergraduate math for cryptography

- Additional contact persons:
  - Gregor Maier (INET) and Collin Mulliner (SECT)

What is this course about?

- Network security? Not quite!

- Focus:
  - Security of networked **applications**
    - Security of Web browsers
  - Protection of network **infrastructure**
Topics

- Basics of secure network protocol design
  - Using cryptography (not a cryptography class!)
  - The role of correct software

- Practical focus
  - This is not a pure academic-style course
  - You’ll see real security holes
  - A lot of (in)security is about doing the unexpected
  - “Think sideways”

How to think about insecurity

- Bad guys don’t follow rules
- Need to understand what sort of attacks are possible to compromise a system
  - Prerequisite to understand what to protect in a system!
- This is not the same as actually launching them!
  - Taking a 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
Reading


- ... (see Web)

- Research papers (see Web)

Network security

Overview
**Dichotomy: hosts**

- Is (or can be) well-controlled
- There are well-developed authentication and authorization models
- Strong notion
  - Of „privileged“ state
  - What programs can use/do

**Dichotomy: networks**

- None of the above
- 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 OS have different – or no – notions of userIDs and privileges

=> notions of privilege is missing
Networking

- Networks interconnect
- Networks always interconnect
- Interconnections happen everywhere 😊 but mainly at the edges

Failures

- Benign failures
  - Most network failures are benign
  - Programs allow for such failures
    - Data corruption
    - Timeouts
    - Dead hosts
    - Routing problems
    - ...

- Rule of thumb:
  - Anything that can happen by accident can happen malicious
  - -> much more dangerous!
**Principle: trust nothing**

- A host can/should trust **nothing** that comes over the wire!
- Any desired protections have to be explicitly supplied
- There may be help from lower layers that supply protection
  - Yet those layers have to be based on the same principle!
  - Research on such lower layer protection is a very hot topic and far from being solved!
**Attitude question**

- **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
- Protocol analysis by formal verification
- *Paranoid design!*
Protocol design

- Heavy use of crypto and authentication
- Ensure that 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!
Course overview

- Introduction
  - Attacks and threats, cryptography overview
  - Authentication (Kerberos, SSL)
- Applications
  - Web, browser, email, ssh
- Lower layer network security
  - NAT, (IPsec), firewalls
- Monitoring / information gathering
  - Intrusion detection, network scans
- Availability
  - Worms, denial of service, network infrastructure