

# Internet Security

Prof. Anja Feldmann, Ph.D.

[anja@net.t-labs.tu-berlin.de](mailto:anja@net.t-labs.tu-berlin.de)  
<http://www.net.t-labs.tu-berlin.de/>

# General information

- ❑ Area: BKS – Hauptstudium Vertiefer
- ❑ Time
  - Wednesday: 10:00 – 12:00
- ❑ Room
  - Telefunkenhochhaus 20. Stock
- ❑ Language
  - English (questions can be asked in German!)
- ❑ Web site
  - [http://www.net.t-labs.tu-berlin.de/teaching/ss07/is\\_ss07/](http://www.net.t-labs.tu-berlin.de/teaching/ss07/is_ss07/)
- ❑ Mailing list
  - see Web page

# General information (2)

## □ 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

# What is this course?

- ❑ Network security? Not quite!
  
- ❑ Focus:
  - Security of networked applications
  - Protection of network infrastructure

# Topics

## ❑ 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 secure 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

- ❑ Kaufman, Perlman, and Spencer.  
Network Security: Private Communication in a  
Public World,  
Second Edition, Prentice Hall, 2002
- ❑ Cheswick, Bellovin, and Rubin.  
Firewalls and Internet Security:  
Repelling the Wily Hacker,  
Second Edition, Addison-Wesley Professional 2003
- ❑ 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 by malice – only more so!

# Principle: trust nothing

- ❑ A host can trust **nothing** that comes over the wire!
- ❑ Any desired protections have to be explicitly supplied
- ❑ There may be help from a middleware layer that supplies protection  
Yet the middleware has to be based on the same principle!

# 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
- ❑ Paranoid design!

# Protocol design

- ❑ Leave room for crypto and authentication
- ❑ Ensure that sensitive fields are protectable
- ❑ 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, email, ssh

## □ Lower layer network security

- IPsec, firewalls, wireless

## □ Monitoring / information gathering

- Intrusion detection, network scans

## □ Availability

- Worms, denial of service, network infrastructure