
Internet Security [1]

VU 184.216

Security and Networking Basics

Paolo Milani Comparetti

pmilani@seclab.tuwien.ac.at

Clemens Kolbitsch

ck@seclab.tuwien.ac.at

News from the Lab

Int. Secure Systems Lab
Vienna University of Technology

- Online registration has started last week
 - Registration possible until 31.03.2010
 - First registration 3 minutes after system went online
 - 2 minutes *before* the TUWIS announcement
 - that's what I call dedication :-) keep it up!!
- Lab starts in two weeks
 - 24.03.2010
 - Challenge 1 will be announced (sniffing, network tools)
- If you have problems, contact
 - inetsec@seclab.tuwien.ac.at

Outline -Today

Int. Secure Systems Lab
Vienna University of Technology

- Introduction and Motivation
- Security Threats
- Open Systems Interconnection (OSI) - Reference Model
 - comparison with TCP/IP protocol suite
- Internet Protocol
 - structure, attributes
 - IP on local networks
 - LAN and fragmentation attacks

Basic terminology

Int. Secure Systems Lab
Vienna University of Technology

- Who is a “hacker“ and who is a “cracker“?
- What is a script kiddie?
- Why do people hack into systems?

Basic terminology

- Who is a “hacker“ and who is a “cracker“?
 - *How to become a hacker (CCC)*
 - * Kündigung deines AOL/T-Online Account.
 - * Besorge dir einen echten IP-Zugang.
 - * Besorge dir ein richtiges UNIX-Betriebssystem (z.B. Linux, *BSD ...).
 - * Lösche Windows.
 - * Lies die Installationsanleitung deines Betriebssystems. Lies sie noch mal. Und noch ein drittes Mal, zur Sicherheit.
 - * Du sollst die Anleitung KOMPLETT lesen!
 - * Installiere das UNIX auf deinem Rechner.

Basic terminology

- Who is a “hacker“ and who is a “cracker“?
– *How to become a hacker (CCC)*

(cont.)

- * Arbeite dich durch die Kernel-Docs/FAQ/HOWTO. Lies alles ein zweites Mal langsamer.
- * Kompiliere dir einen neuen Kernel. Du weist nicht wie? Gehe einen Schritt zurück
- * Hast du einen neuen Kernel übersetzt, installiert und gebootet? Glückwunsch, du hast 5% des Wegs aus Lamerland geschafft.
- * ...

Basic terminology

Int. Secure Systems Lab
Vienna University of Technology

- Who is a “hacker“ and who is a “cracker“?
 - maybe, all a little absurd
 - hackers want to *understand* things...
 - ... down to the last tiny bit

 - applicable to all fields of technology

Basic terminology

- Who is a “hacker“ and who is a “cracker“?
 - *What is a hacker? (Eric S. Raymond)*

There is another group of people who loudly call themselves hackers, but aren't. These are people (mainly adolescent males) who get a kick out of breaking into computers and phreaking the phone system. Real hackers call these people ‘crackers’ and want nothing to do with them. Real hackers mostly think crackers are lazy, irresponsible, and not very bright, and object that being able to break security doesn't make you a hacker any more than being able to hotwire cars makes you an automotive engineer.

Basic terminology

Int. Secure Systems Lab
Vienna University of Technology

- Who is a “hacker“ and who is a “cracker“?
 - *What is a hacker? (Eric S. Raymond)*

The basic difference is this: hackers build things, crackers break them.

Basic terminology

- Who is a “hacker“ and who is a “cracker“?
- What is a script kiddie?
 - *How to become a script-kiddie (CCC)*

q: "Wie lerne ich hacken?"

a: "Lies das Hacker-werden-HOWTO ."

q: "Aach, ich bin zu faul zum lesen / lernen, wie werde ich ein Script-Kiddy?"

a: "Hm, ich glaube ich kann Dir helfen, aber für die Informationsverarbeitung bist Du ganz alleine verantwortlich."

Basic terminology

Int. Secure Systems Lab
Vienna University of Technology

- Who is a “hacker“ and who is a “cracker“?
- What is a script kiddie?
 - *How to become a script-kiddie (CCC)*

```
----- t0p s3kr3t 0nly llnux klddyZ c4n r3ad bel0w th1z
lln3 -----
/* top secret hamstuh encryption */
JLKADJFLK;ASDFJLKSA;DJFLASK;DFJSLAKFJLAKSDFJLASKFJDLSKDJF
* t00lZ *
  exploit code
  named remote expliot code
  ICQ bomber & flooder source code
  Denial Of Service code
  BitchX War Scriptz
* t00lZ EOF *
```

Basic terminology

Int. Secure Systems Lab
Vienna University of Technology

- Who is a “hacker“ and who is a “cracker“?
- What is a script kiddie?
- Why do people hack into systems?
 - Recognition
 - Admiration
 - Curiosity
 - Revenge
 - Power & Gain
 - M.O.N.E.Y

The biggest problems

Int. Secure Systems Lab
Vienna University of Technology

- Software engineering is still considered easy
 - anyone can do it
 - copy&paste code snippets (including vulnerabilities)
- System and network administrators are not prepared
 - insufficient resources
 - lack of training
- Intruders are now leveraging the availability of broadband connections
 - many connected home computers are vulnerable
 - collections of compromised home computers are “good” weapons (e.g., for DDOS, Spam, etc.).

The biggest problems

Int. Secure Systems Lab
Vienna University of Technology

- Typical users are not aware of the problems
- Security is still not part of the development process
 - often, it is applied “in the end”
 - what if a system is insecure *by design*?
- Security is difficult to measure
 - how likely is the abuse of a vulnerability?
 - how much would it cost if it actually occurs?

Number of Reported Incidents

Int. Secure Systems Lab
Vienna University of Technology

Year	1988	1989
Incidents	6	132

Year	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
Incidents	252	406	773	1,334	2,340	2,412	2,573	2,134	3,734	9,859

Year	2000	2001	2002	2003
Incidents	21,756	52,658	82,094	137,529

- Statistics from www.cert.org
 - Given the widespread use of automated attack tools, attacks against Internet-connected systems have become so commonplace **that counts of the number of incidents reported provide little information with regard to assessing the scope and impact of attacks.** Therefore, **we stopped providing this statistic** at the end of 2003.

Vulnerabilities reported

Int. Secure Systems Lab
Vienna University of Technology

1995-1999

Year	1995	1996	1997	1998	1999
Vulnerabilities	171	345	311	262	417

2000-2003

Year	2000	2001	2002	2003
Vulnerabilities	1,090	2,437	4,129	3,784

2004-2008

Year	2004	2005	2006	2007	2008
Vulnerabilities	3,780	5,990	8,064	7,236	6,058

www.cert.org

A little bit of history

- “Hacking”, actually, has been around for decades.
 - 1870s: teenagers were playing around with the “new” phone system
 - 1940s: breaking enemy encryption during WW2 (e.g., *Enigma*)
 - 1960s: mainframe computers like the MIT’s Artificial Intelligence Lab became staging ground for hackers. Hacker was a neutral term.
 - 1960s and 70s: hackers start tampering with phones (the largest network back then). “phreaks” emerge (phone hackers)
 - *Blue Boxing*: 2600 Hz impulse, used by telephone switchboards
 - remember the film *Hackers*?
 - Early 1980s: The term “cyberspace” is coined in film *Neuromancer*. First hacker arrests are made. Two hacker groups form: Legion of Doom (US) and Chaos Computer Club (DE)

A little bit of history...

Int. Secure Systems Lab
Vienna University of Technology

- Late 1980s: Computer Fraud and Abuse Act, CERT (Computer Emergency Response Team) is formed, Kevin Mitnick is arrested
- Early 1990s: AT&T long distance service crashes, crackdown on hackers in the US, hackers break into Griffith Air Force Base, NASA, etc.
- Late 1990s: Hackers deface many government web sites, Defense Department computers receive 250,000 attacks in one year
- 2000s: Number of attacks keep rising, “new” attacks emerge (e.g., phishing)

Changing nature of the threat

Int. Secure Systems Lab
Vienna University of Technology

- Intruders are more prepared and organized
 - similar structures to mafia organizations
 - *Underground Economy* joins together experts of different domains
- Internet attacks are easy, low-threat and difficult to trace
- Intruder tools are increasingly sophisticated and easy to use (e.g., by kiddies)
- Source code is *not* required to find vulnerabilities
- Increasing complexity of Internet-related applications and protocols – also our dependency on them

Security threats

Int. Secure Systems Lab
Vienna University of Technology

Information Domain

- Leakage
 - acquisition of information by unauthorized recipients. e.g. Password sniffing
- Tampering:
 - unauthorized alteration/creation of information (including programs)
 - e.g. change of electronic money order, installation of a rootkit

Security threats

Int. Secure Systems Lab
Vienna University of Technology

Operation Domain:

- Resource stealing
 - (ab)use of facilities without authorization (e.g. Use a high-bandwidth infrastructure to issue DDOS attacks)
- Vandalism
 - interference with proper operation of a system without gain (e.g. flash bios with 0x0000)

Methods of attacking

Int. Secure Systems Lab
Vienna University of Technology

- Eavesdropping
 - getting copies of information without authorization
- Masquerading
 - sending messages with other's identity
- Message tampering
 - change content of message

Methods of attacking

Int. Secure Systems Lab
Vienna University of Technology

- Replaying
 - store a message and send it again later, e.g. resend a payment message
- Exploiting
 - using bugs in software to get access to a host
- Combinations
 - Man in the middle attack
 - emulate communication of both attacked partners (e.g., cause havoc and confusion)

Social engineering

- Before we get into technical stuff – let’s look at a popular non-technical attack method
 - “The art and science of getting someone to comply to your wishes”
 - Security is all about trust. Unfortunately, the weakest link, the user, is often the target (i.e., “Hit any user to continue” 😊)
 - Social engineering by phone
 - Dumpster Diving
 - Reverse social engineering
- According to reports, secret services often use social engineering techniques for intrusion

Social engineering

Int. Secure Systems Lab
Vienna University of Technology

- Semi-technical attacks
 - more or less technically sophisticated attacks
 - hard to fight with “technical means”
 - imagine how much information you can retrieve from used devices
 - buy hard drive on e-bay and undelete data (if necessary at all)
 - used/stolen/lost cell phones, PDAs, laptops, etc.
 - phishing email
 - spear phishing
 - social networks / platforms
 - tons of confidential data
 - applications have *unrestricted access* to all data (once installed)

Social engineering

Int. Secure Systems Lab
Vienna University of Technology

- *Only one* possible (real) solution
 - educate users
 - increase awareness
- Large companies now start to “attack” their own employees
 - e.g., Microsoft
 - targeted phishing attacks

Choosing a good password

Int. Secure Systems Lab
Vienna University of Technology

- Retina checks are currently not possible, so guard your password ;-)
 - **NEVER give your password to anyone**
 - *Not even your girl(boy-)friend*
 - **Make your password something you can remember**
 - **Make your password difficult for others to guess**
 - **DO NOT change your password because someone told you to (e.g., via e-mail)**
- Crackers might crack the following passwords:
 - Words in *any* dictionary, your user name, your name, names of people you know, substituting some characters (a 0 (zero) for an o, or a 1 for an l)
 - <http://www.openwall.com/john/> (John, passwd cracker)

Choosing a good password

*Int. Secure Systems Lab
Vienna University of Technology*

```
Jul 17 20:21:07 server sshd[27362]: Address 218.95.240.222 maps to 222.240.95.218.broad.xn.qh.dyn
amic.163data.com.cn, but this does not map back to the address - POSSIBLE BREAK-IN ATTEMPT!
Jul 17 20:21:07 server sshd[27362]: pam_unix(sshd:auth): authentication failure; logname= uid=0 e
uid=0 tty=ssh ruser= rhost=218.95.240.222 user=root
Jul 17 20:21:09 server sshd[27362]: Failed password for root from 218.95.240.222 port 43813 ssh2
Jul 17 20:21:12 server sshd[27390]: Address 218.95.240.222 maps to 222.240.95.218.broad.xn.qh.dyn
amic.163data.com.cn, but this does not map back to the address - POSSIBLE BREAK-IN ATTEMPT!
Jul 17 20:21:12 server sshd[27390]: Invalid user stud from 218.95.240.222
Jul 17 20:21:12 server sshd[27390]: pam_unix(sshd:auth): check pass; user unknown
Jul 17 20:21:12 server sshd[27390]: pam_unix(sshd:auth): authentication failure; logname= uid=0 e
uid=0 tty=ssh ruser= rhost=218.95.240.222
Jul 17 20:21:14 server sshd[27390]: Failed password for invalid user stud from 218.95.240.222 por
t 44610 ssh2
Jul 17 20:21:17 server sshd[27418]: Address 218.95.240.222 maps to 222.240.95.218.broad.xn.qh.dyn
amic.163data.com.cn, but this does not map back to the address - POSSIBLE BREAK-IN ATTEMPT!
Jul 17 20:21:17 server sshd[27418]: Invalid user trash from 218.95.240.222
Jul 17 20:21:17 server sshd[27418]: pam_unix(sshd:auth): check pass; user unknown
Jul 17 20:21:17 server sshd[27418]: pam_unix(sshd:auth): authentication failure; logname= uid=0 e
uid=0 tty=ssh ruser= rhost=218.95.240.222
Jul 17 20:21:19 server sshd[27418]: Failed password for invalid user trash from 218.95.240.222 po
rt 45382 ssh2
Jul 17 20:21:21 server sshd[27509]: Address 218.95.240.222 maps to 222.240.95.218.broad.xn.qh.dyn
amic.163data.com.cn, but this does not map back to the address - POSSIBLE BREAK-IN ATTEMPT!
```

Choosing a good password

Int. Secure Systems Lab
Vienna University of Technology

- Guidelines...
 - a password that is at least eight characters long
 - a good password will have a mix of lower- and upper-case characters, numbers, and punctuation marks, and should be at least eight characters long
 - take a phrase and try to squeeze it into eight characters
 - e.g., this is an interesting lecture oh yeah== ***tiailo***
 - throw in a capital letter, a punctuation mark, number or two
→ ***0Tiailo******y4***
 - Something that no one but you would ever think of. Use your imagination!
 - Remember a few passwords for different levels of importance, reaching from forum access to your online banking account

Password examples

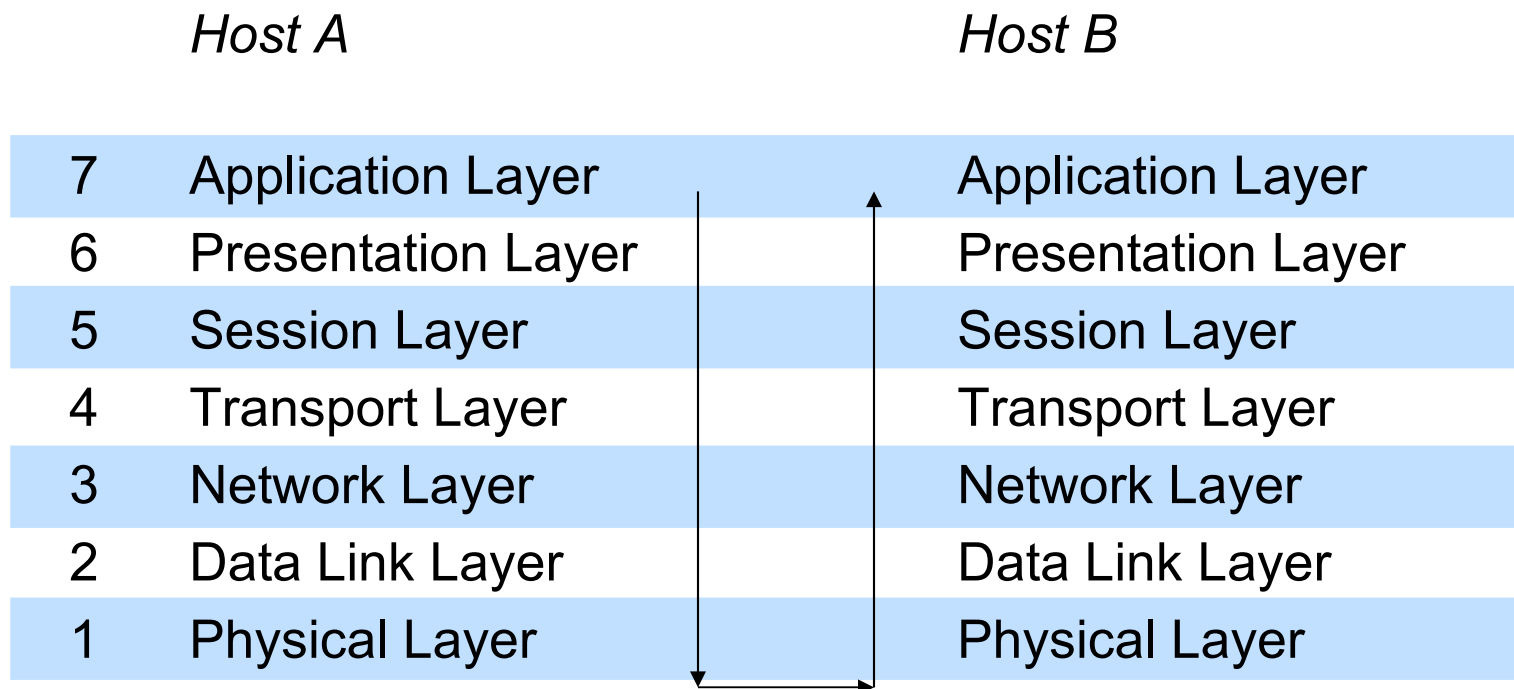
Int. Secure Systems Lab
Vienna University of Technology

- The “Bad”
 - acmilan1
 - mymusic2
 - bermuda6
 - Konrad4868

- The “Good”
 - #bdiBuM1a
 - Qa56Fge(/
 - sdFOiKqw”=

OSI reference model

- Developed by the ISO to support **open systems interconnection**
 - layered architecture, level n uses service of $(n-1)$



OSI reference model

Int. Secure Systems Lab
Vienna University of Technology

- Physical Layer (1)
 - connect to channel / used to transmit bytes (= network cable)
 - repeater, hub
- Data Link Layer (2)
 - error control between adjacent nodes
 - bridge, switch
- Network Layer (3)
 - transmission and routing across subnets
 - router
- Transport Layer (4)
 - ordering
 - multiplexing
 - correctness

OSI reference model

Int. Secure Systems Lab
Vienna University of Technology

- Session Layer (5)
 - support for session-based interaction
 - e.g. communication parameters/communication state
- Presentation Layer (6)
 - standard data representation
- Application Layer (7)
 - application specific protocols

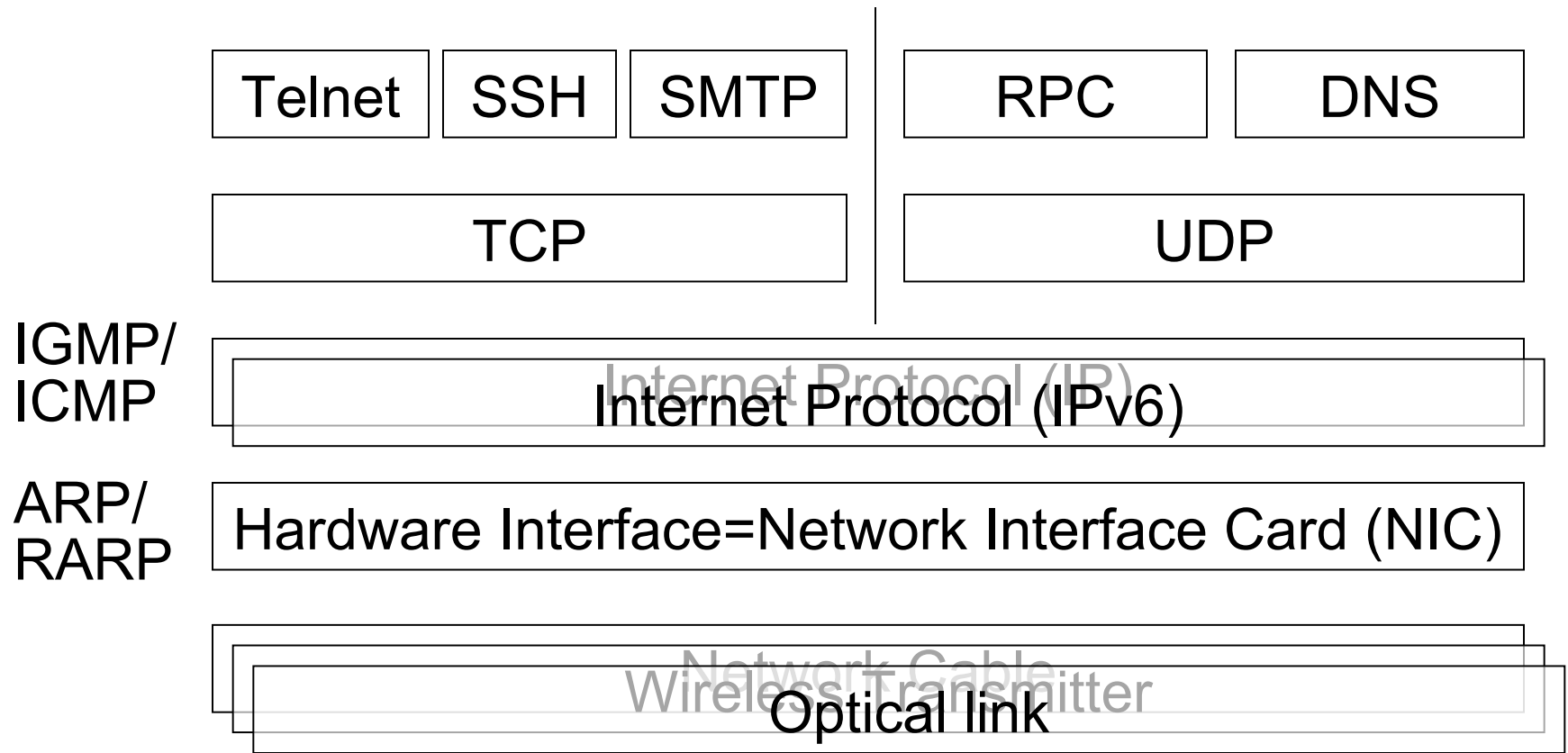
Why layering?

Int. Secure Systems Lab
Vienna University of Technology

- Openness
 - as long as upper layers are the same heterogenous networks can interact
- Fertilizes compatibility of systems
- Allows vendor-specific devices
- Allows vendor-specific protocols
- Provides independence from one manufacturer
- OSI implementation
 - *MAP* (Manufacturing Automation Protocol) – GM
 - Token Ring

TCP-IP layering

*Int. Secure Systems Lab
Vienna University of Technology*



Mapping

*Int. Secure Systems Lab
Vienna University of Technology*

TCP/IP

Telnet

SMTP

TCP

Internet Protocol (IP)

Ethernet Packet

NIC

OSI-Reference

Application

Transport

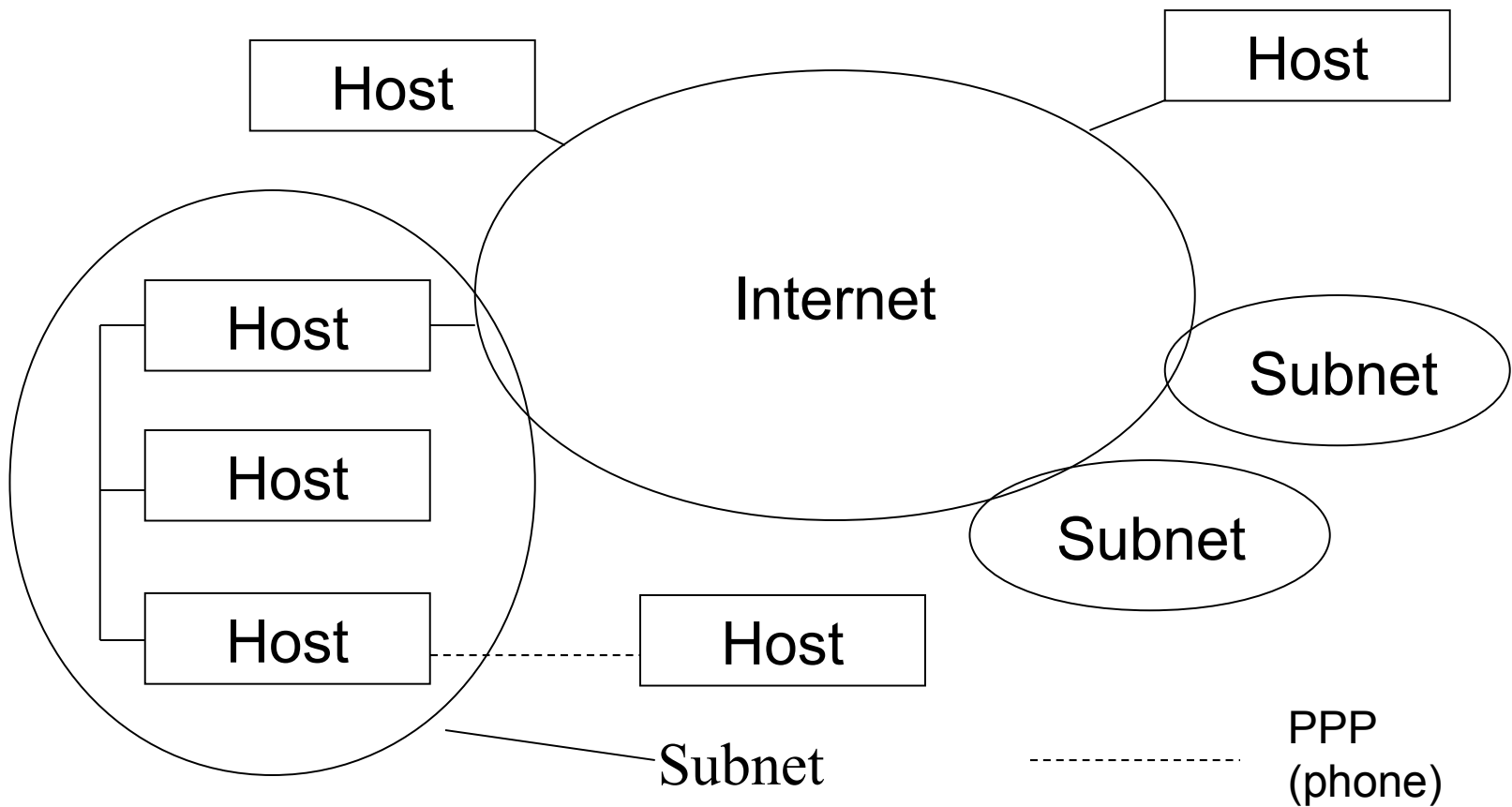
Network

Data Link Layer

Physical Layer

The Internet

*Int. Secure Systems Lab
Vienna University of Technology*



IP addresses

- IP addresses in IPv4 are 32 bit numbers
 - ([class]+net+host id)
- Each host has a unique IP address for each NIC
- Represented as dotted-decimal notation:
 - 10000000 10000011 10101100 00000001 = 128.131.172.1
- Classes: <starts with> <netbits> <hostbits> <#of possible hosts>
- Class A: 0 7 24 16,777,216
- Class B: 10 14 16 65,536
- Class C: 110 21 8 256
- Class D: 1110 *special meaning: 28 bit multicast address*
- Class E: 1111 *reserved for future use*

IP subnetting

- It is unrealistic to have networks with so many hosts
 - divide the hostbits into subnet ID and host ID
 - saves address space

- Example: Class C normally has 24 netbits

Class C network with subnet mask 255.255.255.240

240 = 1111 0000

| host ID → 16 hosts within every subnet

subnet ID → 16 subnets within network

Special IP addresses

- As source and destination address
 - loopback interface (127.0.0.1)
- As destination address
 - all bits set to 1: local broadcast
 - net ID <> only 1s, host ID only 1s
 - net directed broadcast to net ID
- Reserved addresses (RFC 1597) - non routable
 - 10.0.0.0 - 10.255.255.255
 - 172.16.0.0 - 172.131.255.255
 - 192.168.0.0 - 192.168.255.255

Internet Protocol (IP)

Int. Secure Systems Lab
Vienna University of Technology

- Is the glue between hosts of the Internet
- Standardized in RFC 791
- Attributes of delivery
 - connectionless
 - unreliable, best-effort datagram
 - delivery, integrity, ordering, non-duplication are NOT guaranteed
 - i.e., they can be dropped, tampered with, replayed, spoofed, etc. (at least in IPv4)

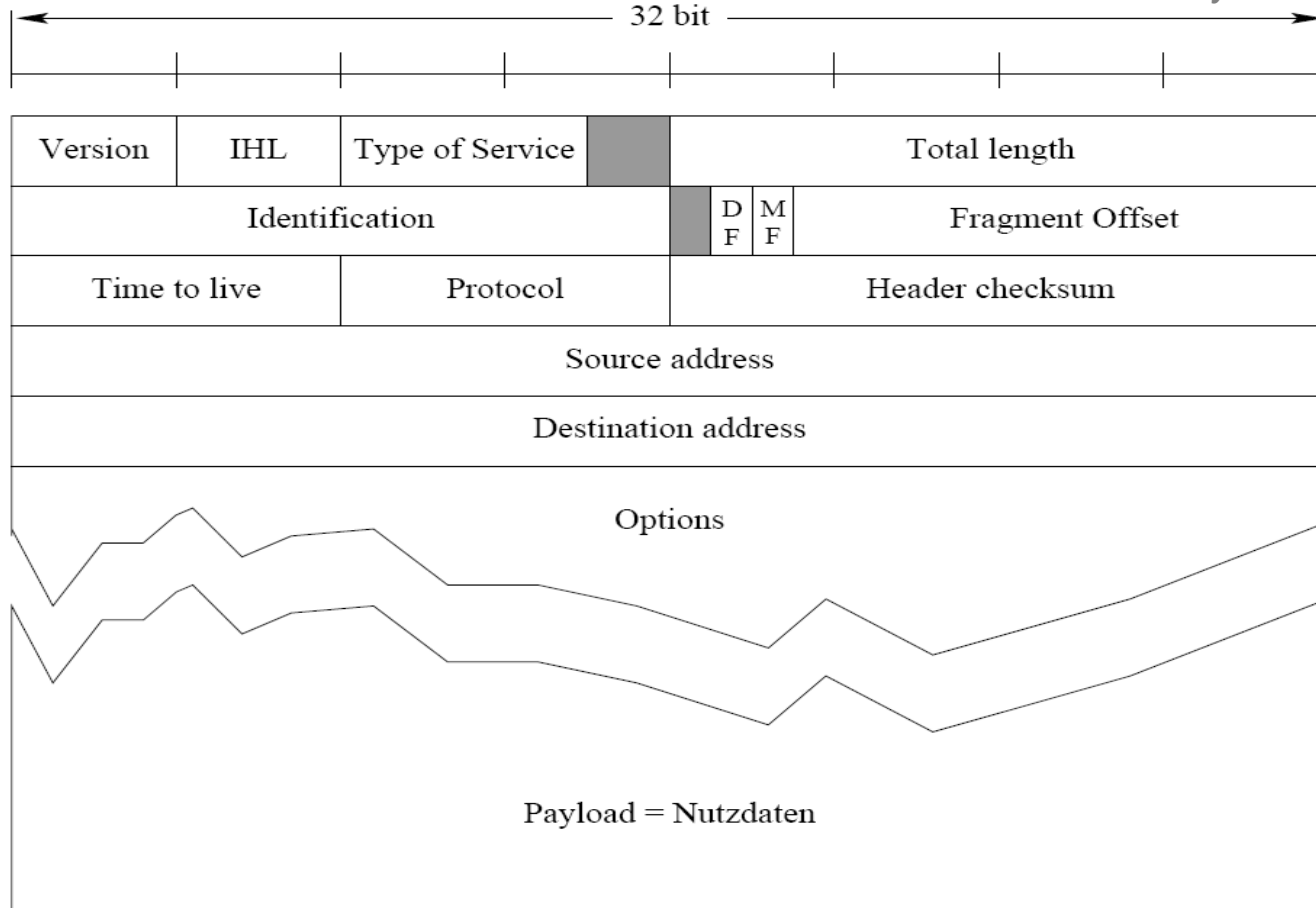
Internet Protocol (IP)

Int. Secure Systems Lab
Vienna University of Technology

- IP packets (datagrams) can be exchanged by any two nodes that are set up as IP nodes
- For direct communication IP is tunneled through lower level protocols like
 - Ethernet
 - Token Ring
 - FDDI (optical)
 - PPP, etc.

IP Datagram

Int. Secure Systems Lab
Vienna University of Technology



IP Header

- Normal size: 20 bytes
- Version (4 bits):
 - current value = 4 (IPv4)
- Header length (4 bits):
 - number of 32 bit words in the header, including IP options
- Type of service
 - priority (3 bits), QOS(4), unused bit
- Total length: total size of the IP header and data
- Identifier (16): datagram identification
 - +1 incremented

IP Header

- Flags (3) and Offset (13 bits)
 - used for fragmentation of datagrams
- Time To Live (8 bits):
 - Allowed number of hops in the delivery process. Initially meant to entitle seconds between hops.
- Protocol (8 bits):
 - specifies the type of protocol which is encapsulated in the datagram (TCP, UDP)
- Header checksum (16):
 - checksum calculated over the IP header.
- Addresses (32+32 bits)
 - specify source and destination

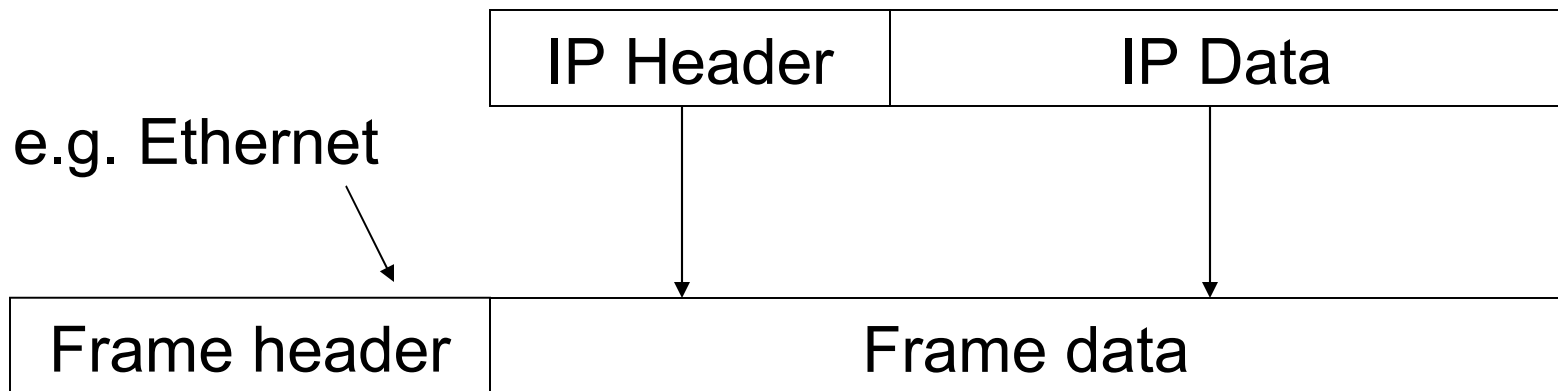
IP Options

- Variable length
- Identified by first byte
 - security and handling restrictions
 - record route: ip addresses of routers are stored
 - time stamp: each router records its timestamp
 - source route:
 - specifies a list of IP addresses that the datagram has to traverse
 - loose: prefer these hosts
 - strict: only use the specified hosts (route)

IP Encapsulation

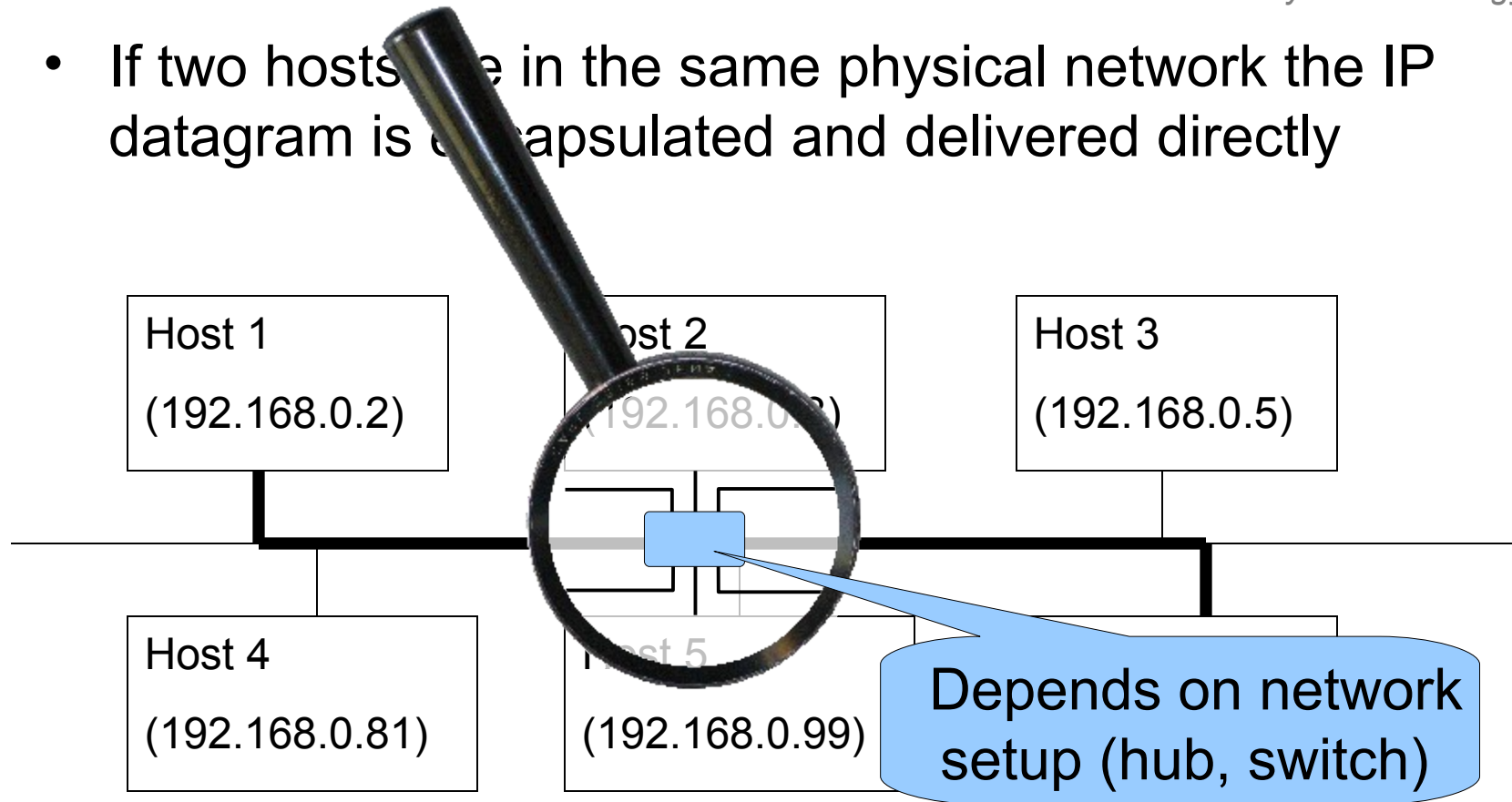
- How are IP datagrams transferred over a LAN?
- Can't be done directly because of different formats.
RFC 894, 826 explain IP over Ethernet

Solution: Encapsulation + direct delivery



Direct IP delivery

- If two hosts are in the same physical network the IP datagram is encapsulated and delivered directly



Fragmentation

- Used if encapsulation in lower level protocol demands to split the datagram into smaller portions
 - when datagram size is larger than data link layer MTU
 - (=Maximum Transmission Unit)
- Performed at
 - the source host
 - or in an intermediate step
- Reassembling
 - = rebuilding the IP packet
 - is ONLY performed at the destination
- Each fragment is delivered as a separate datagram

Fragmentation

- Adapted IP header is sent in every fragment
- Controlled using 3 bits IP-flags + 13 bits offset
 1. reserved
 2. don't fragment bit: set if datagram shouldn't be fragmented
 3. more fragments bit: set if this is not the last fragment of an IP datagram
- If fragmentation would be necessary, but don't fragment bit is set -> Error message (ICMP) is sent to sender
- If one fragment is distorted or lost, the entire datagram is discarded

Fragmentation attacks

Old trick: Ping of death:

violate maximum IP datagram size

- ping is an IP based service: are hosts up and reachable?
- Normally uses 64 bytes payload.
- With fragmentation an IP packet with size > 65535 could be sent

Offset of the last segment is such that the total size of the reassembled datagram is bigger than the maximum allowed size: a static kernel buffer is overflowed causing a kernel panic (worked with Windows, Mac, Linux 2.0.x)

Fragmentation attacks

Old trick: TCP overwrite:

fool the firewall

- IP datagram containing TCP traffic is fragmented
- TCP header contains allowed port (e.g. 80)
- => firewall lets this packet pass
- data is sent fragmented
- one packet contains frag-offset=1: ports will be overwritten (e.g. new port = 23).
- after packet has been reassembled completely, it will be delivered to the new port

Ethernet

*Int. Secure Systems Lab
Vienna University of Technology*

dest (48 bits)	src (48 bits)	type (16)	data	CRC (32)
----------------	---------------	-----------	------	----------

0x0800	IP Datagram
--------	-------------

0x0806	ARP	PAD
--------	-----	-----

0x8035	RARP	PAD
--------	------	-----

< 28 bytes >

< 18 bytes >

Ethernet

Int. Secure Systems Lab
Vienna University of Technology

- Widely used link layer protocol
- Carrier Sense, Multiple Access, Collision Detection
- Addresses: 48 bits (e.g. 00:38:af:23:34:0f), mostly
 - hardwired by the manufacturer
- Type (2 bytes): specifies encapsulated protocol
 - IP, ARP, RARP
- Data:
 - min 46 bytes payload (padding may be needed), max 1500 bytes
- CRC (4 bytes)

LAN Attacks

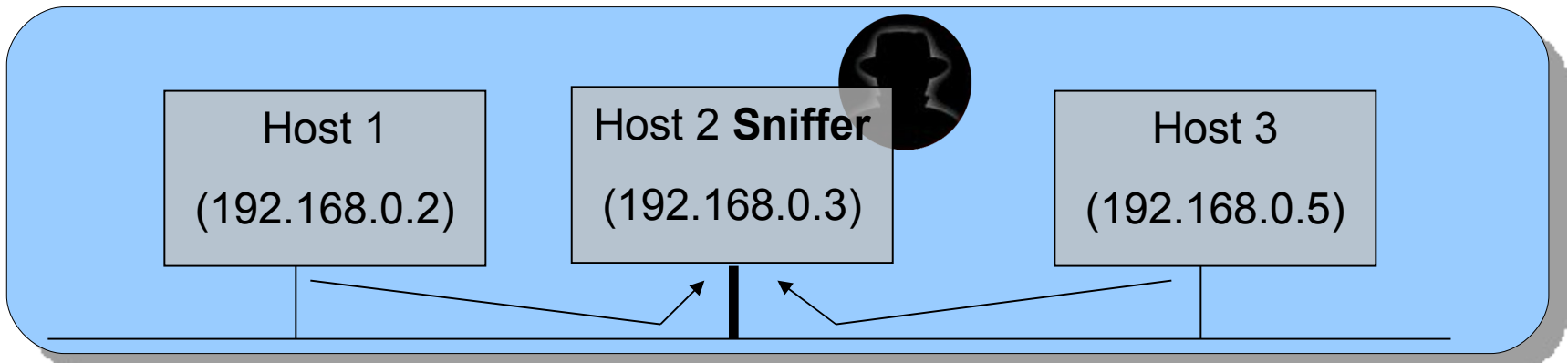
Int. Secure Systems Lab
Vienna University of Technology

- **Goals:**
 - information recovery
 - impersonate host
 - tamper with delivery mechanisms

- **Methods:**
 - sniffing
 - IP Spoofing (next lectures)
 - ARP attacks (next lectures)

Network sniffing

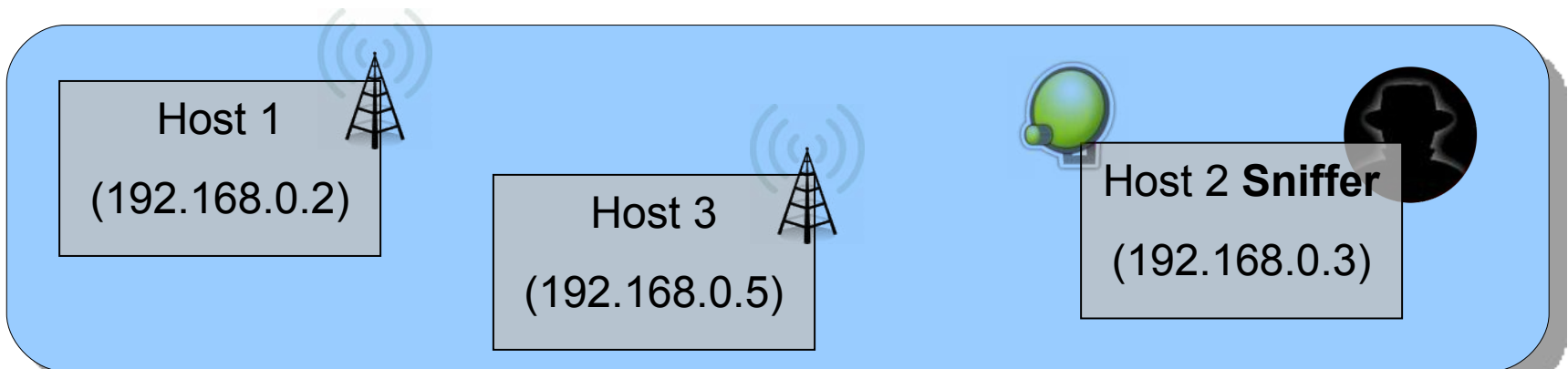
- Is the base for many attacks
 - attacker sets computer's NIC into **promiscuous mode**
 - NIC delivers all arriving packets to *IP* layer
 - can access all the traffic on the segment
- Many protocols transfer authentication information in cleartext
 - possible to collect username/password etc.
- Many tools available: tcpdump -x, dsniff etc.



Network sniffing

Int. Secure Systems Lab
Vienna University of Technology

- Particularly worry-some: Wireless networks
 - attacker sets computer's NIC into **monitor mode**
 - NIC delivers all arriving packets to *physical* layer
 - can access all the traffic on all networks (possibly even multiple frequencies via channel hopping)
- Many tools can also break encryption: e.g., aircrack-ng
 - do **not** use WEP
 - breaking it is a matter of **seconds!**



Network sniffing

Is sniffing also possible at switched Ethernet, where the switch only forwards the right packets to your host? YES!

- MAC flooding
 - switch maintains table with MAC address/port mappings
 - flooding switch with bogus MAC addresses will overflow table
 - switch will revert to hub mode
- MAC duplicating/cloning
 - you can buy NICs with reconfigurable MAC addresses
 - switch will record this in table and sends traffic to you

Detecting sniffers

Int. Secure Systems Lab
Vienna University of Technology

- Interface is in promiscuous / monitor mode
 - use programs like `/sbin/ifconfig` to find out state of NIC
 - for wireless NIC: `/sbin/iwconfig`
- Suspicious DNS lookups
 - sniffer attempts to resolve names associated with IP addresses
 - trap: generate connection from fake IP => detect DNS traffic

Detecting sniffers

Int. Secure Systems Lab
Vienna University of Technology

- Sending IP packet to a replying service (DNS, Telnet)
 - set the destination IP address to suspected sniffer host
 - set the MAC address to a non-existing one
 - host replies => all packets are delivered to the TCP/IP stack
- Latency
 - use ping to analyze response time of host A
 - generate huge amount of traffic to other hosts
 - analyze response time of host A
 - if in promiscuous mode: larger response time, because all the packets are analyzed

Conclusion

Int. Secure Systems Lab
Vienna University of Technology

- In this lecture, we looked at security and networking basics
 - security threats
 - social Engineering
 - OSI Reference Model and TCP/IP Protocol Suite
 - Ethernet, IP
 - LAN and Fragmentation attacks
- Next lecture:
 - we start looking at TCP/IP Protocol Suite
 - more technical attacks