

---

# Internet Security [1]

## VU 188.366

# Security and Networking Basics

Paolo Milani Comparetti, Christian Platzer, Gilbert Wondracek,  
Markus Huber and Edgar Weippl

[inetsec@iseclab.org](mailto:inetsec@iseclab.org)

# Administration

---

- Online registration started today
  - Registration possible until 06.04.2011
    - 96 fellow aspirants so far
  - First registration 8 minutes after system went online
    - → that's Dedication!
- Lab starts next week
  - 24.03.2011
  - Challenge 1 will be announced (sniffing, network tools)
- If you have problems, contact
  - [inetsec@seclab.tuwien.ac.at](mailto:inetsec@seclab.tuwien.ac.at)

# Outline

---

- 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

---

- 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)
    - Get rid of your T-Online account and get a real IP-Access.
    - Get a Unix OS (Linux, BSD...).
    - Delete Windows
    - Read the OS Manual
    - Install the OS
    - Dig through the Kernel Docs/FAQ/HOWTO.
    - Compile your own kernel
    - Grats! You're 5% through
    - .....

# Basic terminology

---

- Who is a “hacker“ and who is a “cracker“?
  - Ok, it’s a little absurd, but the essence is:
    - Hackers want to understand things...
    - ... down to the last detail
  
  - Applies to almost any field 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

---

- Who is a “hacker” and who is a “cracker“?
- What is a script kiddie?

```
-------- t0p s3kr3t 0nly llnux klddyZ c4n r3ad bel0w thlz  
  lln3 -----  
/* top secret hamstuh encryption */  
JLKADJFLK;ASDFJLKSA;DJFLASK;DFJSLAKFJLAKSDFJLASKFJDLSKDJF  
* t001Z *  
  exploit code  
  named remote exploit code  
  ICQ bomber & flooder source code  
  Denial 0f Service code  
  BitchX War Scriptz  
* t001Z EOF *
```

# Basic terminology

---

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

# The biggest problems

---

- Software development is perceived as
  - being easy (anyone can do it)
  - a matter of copying and pasting 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

---

- Typical users are not aware of possible problems
- Security is not part of the development process
  - Security fixes on a “on-demand-basis”
  - Insecurity by design
  - Fixing bugs is more important than closing possible security holes
- Security is hard to measure
  - How likely is an abuse of a vulnerability?
  - How much does it cost when it happens?
  - How much would it cost to tackle it right away?

# Number of reported incidents

---

## 1988-1989

Year	1988	1989
Incidents	6	132

## 1990-1999

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

## 2000-2003

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

[www.cert.org/stats](http://www.cert.org/stats)

# Vulnerabilities reported

---

## 1995-1999

Year	1995	1996	1997	1998	1999
<b>Vulnerabilities</b>	171	345	311	262	417

## 2000-2003

Year	2000	2001	2002	2003
<b>Vulnerabilities</b>	1,090	2,437	4,129	3,784

## 2004-2008

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

[www.cert.org/stats](http://www.cert.org/stats)

# 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.
  - 1970s: hackers start tampering with phones (the largest network back then). “phreaks” emerge (phone hackers)
    - Blue Boxing
  - 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...

---

- 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

---

- Intruders are more prepared and organized
  - Similar structures to organized crime (e.g. mafia)
  - Monetary possibilities form a *Underground Economy*
- 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
- The complexity of Internet-related applications and protocols are increasing – and so is our dependency on them

# Security threats

---

## 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

---

## 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

---

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

# Methods of attacking

---

- 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
  - Remember the film “Die hard 4”?
  - “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

# Social engineering

---

- 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

---

- Only one solution
  - User education
  - Raising awareness
- Large companies now start to “attack” their own employees
  - e.g. Microsoft
  - Targeted phishing attacks

# Choosing a good password

---

- 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 tells you to***
- 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

---

```
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

---

- 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
  - take a phrase and try to squeeze it into eight characters
    - e.g., this is an interesting lecture oh yeah == ***tiailoy***
    - Throw in a capital letter and a punctuation mark or a number or two (== ***0Tiailoy4***)
  - 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

---

- The “Bad”
  - acmilan1
  - mymusic2
  - bermuda6
  - Konrad4868
  - Master
  - God
  
- 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)

#	Host A	Host B
7	Application Layer	Application Layer
6	Presentation Layer	Presentation Layer
5	Session Layer	Session Layer
4	Transport Layer	Transport Layer
3	Network Layer	Network Layer
2	Data Link Layer	Data Link Layer
1	Physical Layer	Physical Layer

# OSI reference model

---

- 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

---

- 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

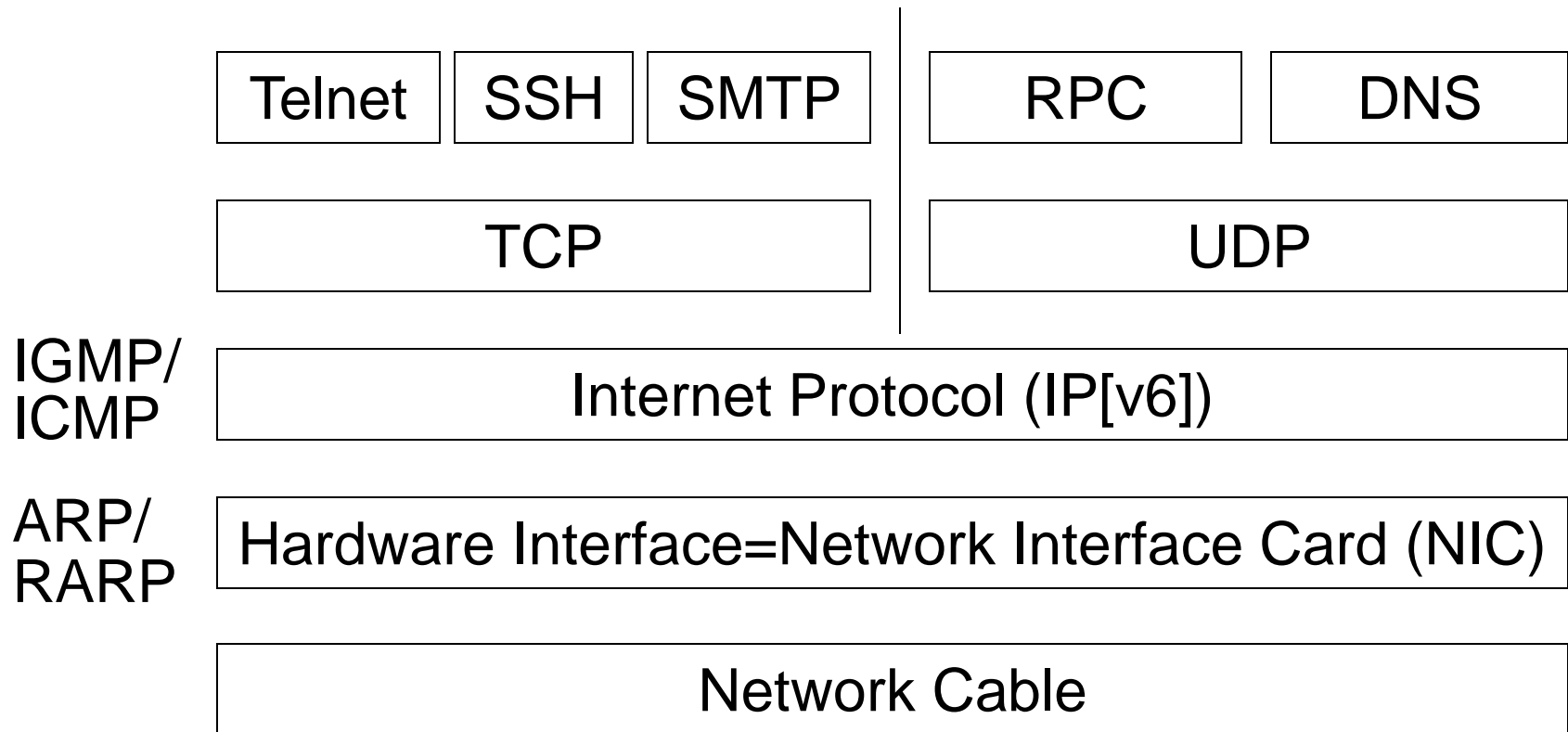
# Why layering?

---

- 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)

# TCP-IP layering

---



# Mapping

---

## TCP/IP

Telnet

SMTP

TCP

Internet Protocol (IP)

Ethernet Packet

NIC

## OSI-Reference

Application

Transport

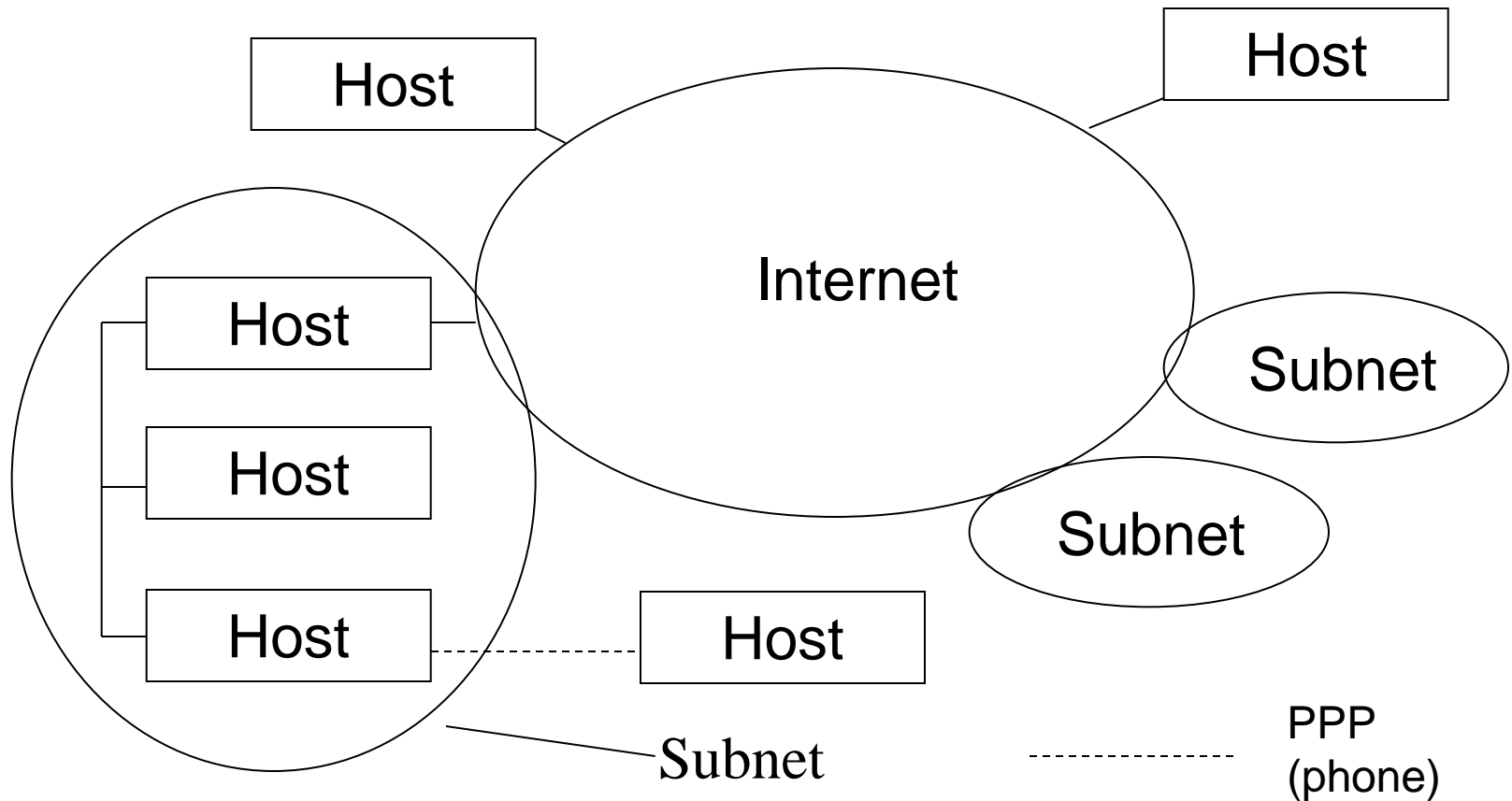
Network

Data Link Layer

Physical Layer

# The Internet

---



# 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

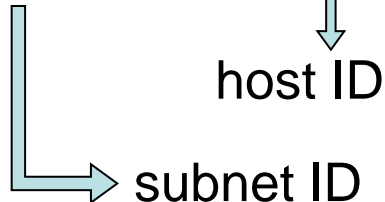
---

- 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



=> 16 hosts within every subnet

=> 16 subnets within this 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
  - netid <> only 1s, hostid only 1s
    - net directed broadcast to netid
- 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)

---

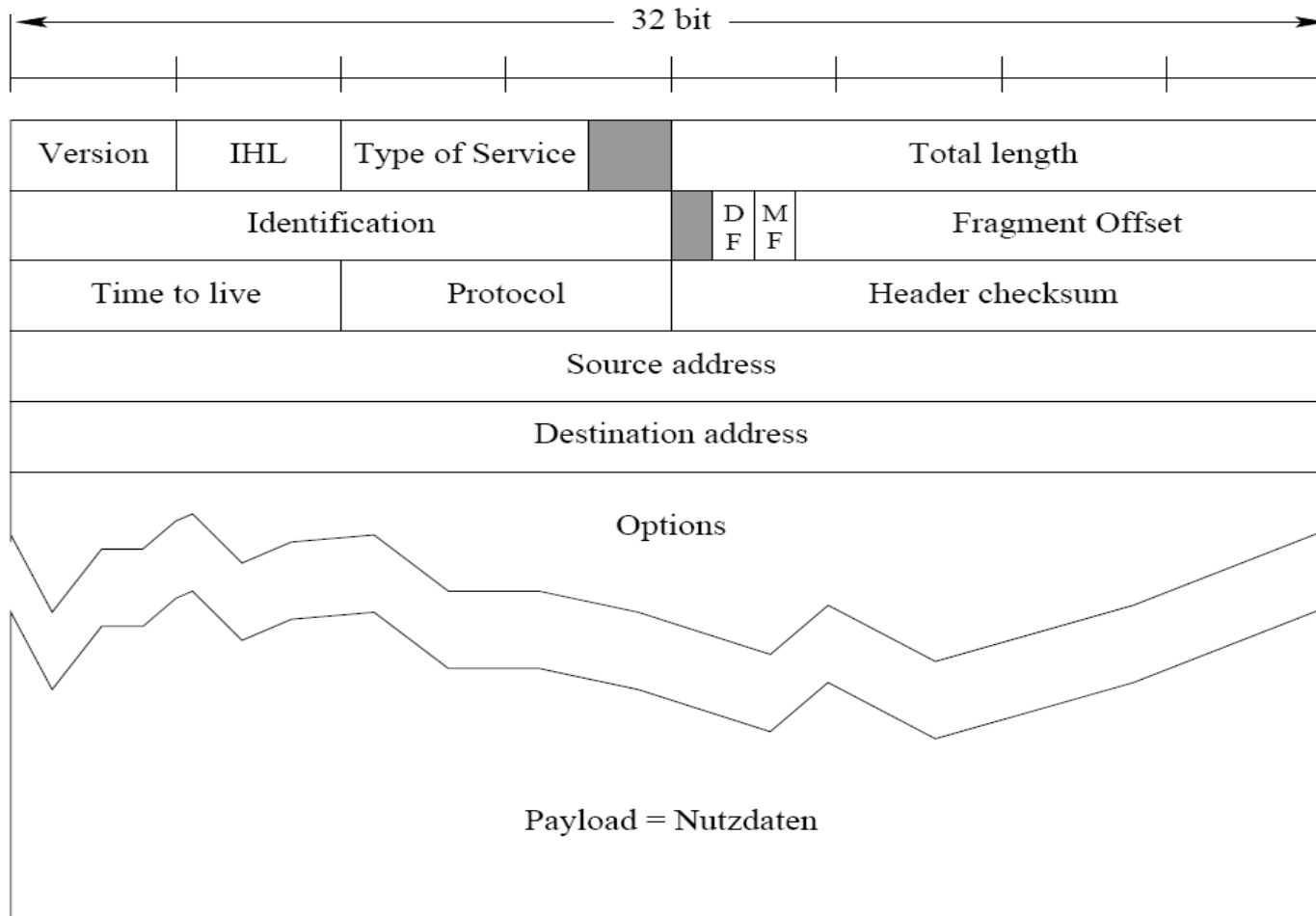
- Is the glue between hosts of the Internet
- Standardized in RFC 791
- Attributes of delivery
  - Connectionless
  - unreliable best-effort datagram
    - delivery, integrity, ordering, non-duplication are NOT guaranteed
    - i.e., they can be dropped, tampered with, replayed, spoofed, etc. (at least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

# Internet Protocol (IP)

---

- 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



# 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 (8bits):
  - 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

---

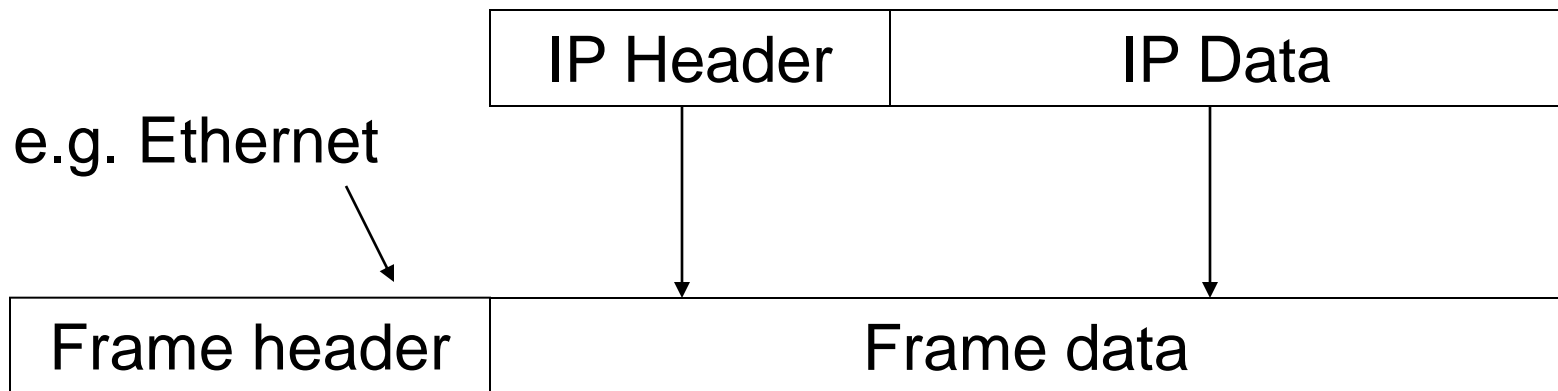
- Variable length
- Identified by first byte
  - security and handling restrictions:
  - Record route: ip addresses of routers are stored
  - Time stamp: each router records its timestamp
  - Source route:
    - specifies a list of IP addresses that the datagram has to traverse
      - loose: prefer 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

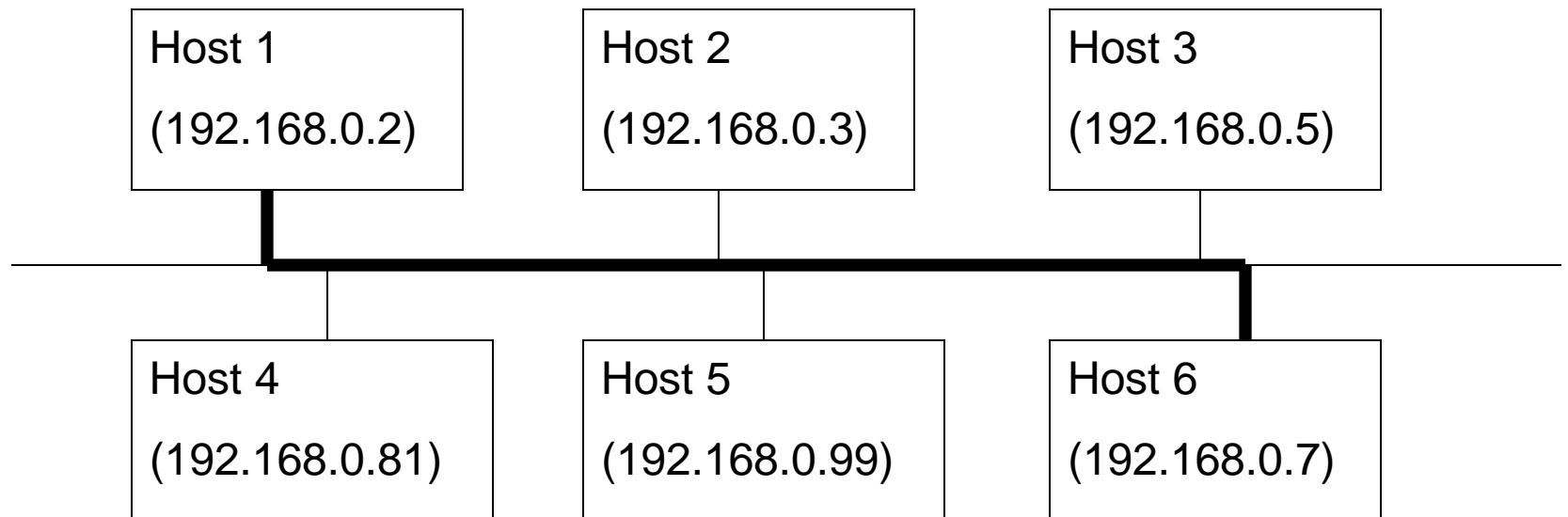
Solution:      Encapsulation + direct delivery



# 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

---

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

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

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

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

< 18 bytes >

< 28 bytes >

# Ethernet

---

- 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)

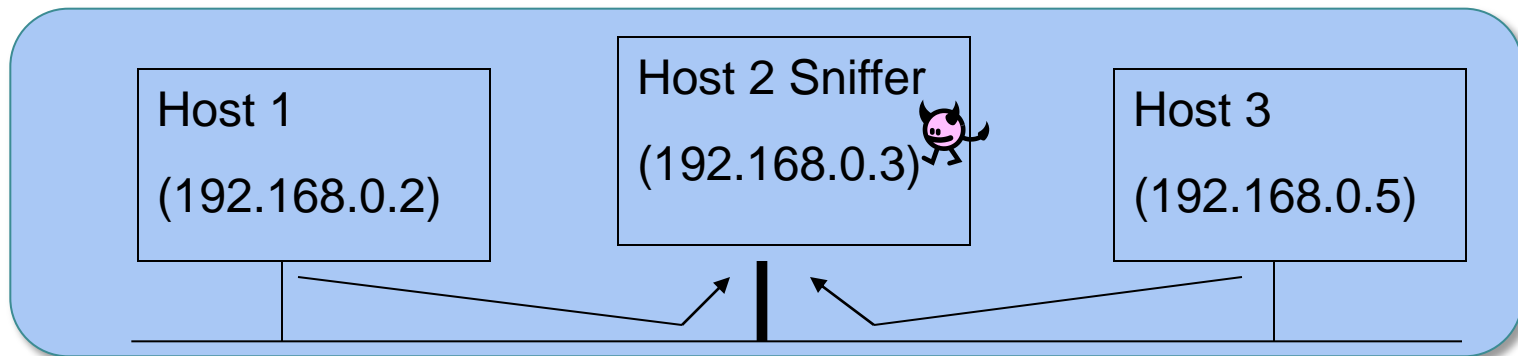
# LAN Attacks

---

- Goals:
  - Information Recovery
  - Impersonate Host
  - Tamper with delivery mechanisms
- Methods:
  - Sniffing
  - IP Spoofing (next lectures)
  - ARP attacks (next lectures)

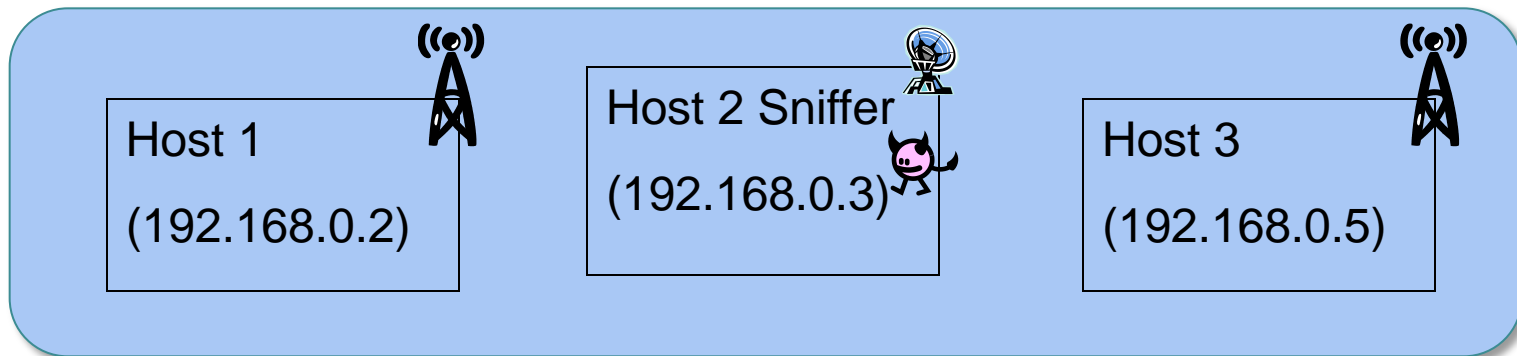
# Network sniffing

- 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 => collect username/password etc.
- Many tools available: tcpdump -x, dsniff etc.



# Network sniffing

- 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 (even multiple frequencies via channel hopping)
- Many tools to break encryption
  - e.g. aircrack-ng
  - do not use WEP
  - breaking 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

---

- Interface is in promiscuous 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

---

- Sending IP packet to a replying service (DNS, Telnet)
  - set the destination IP Address to that 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

---

- 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 and related attacks
  - More technical attack