Remote Network Analysis

- I know what you know -

Torsten Höfler

htor@cs.tu-chemnitz.de

Outline

1. Introduction
2. Passive Analysis
3. Active Analysis
4. Advanced Scanning
5. Prevention
6. Questions

Introduction

Motivation

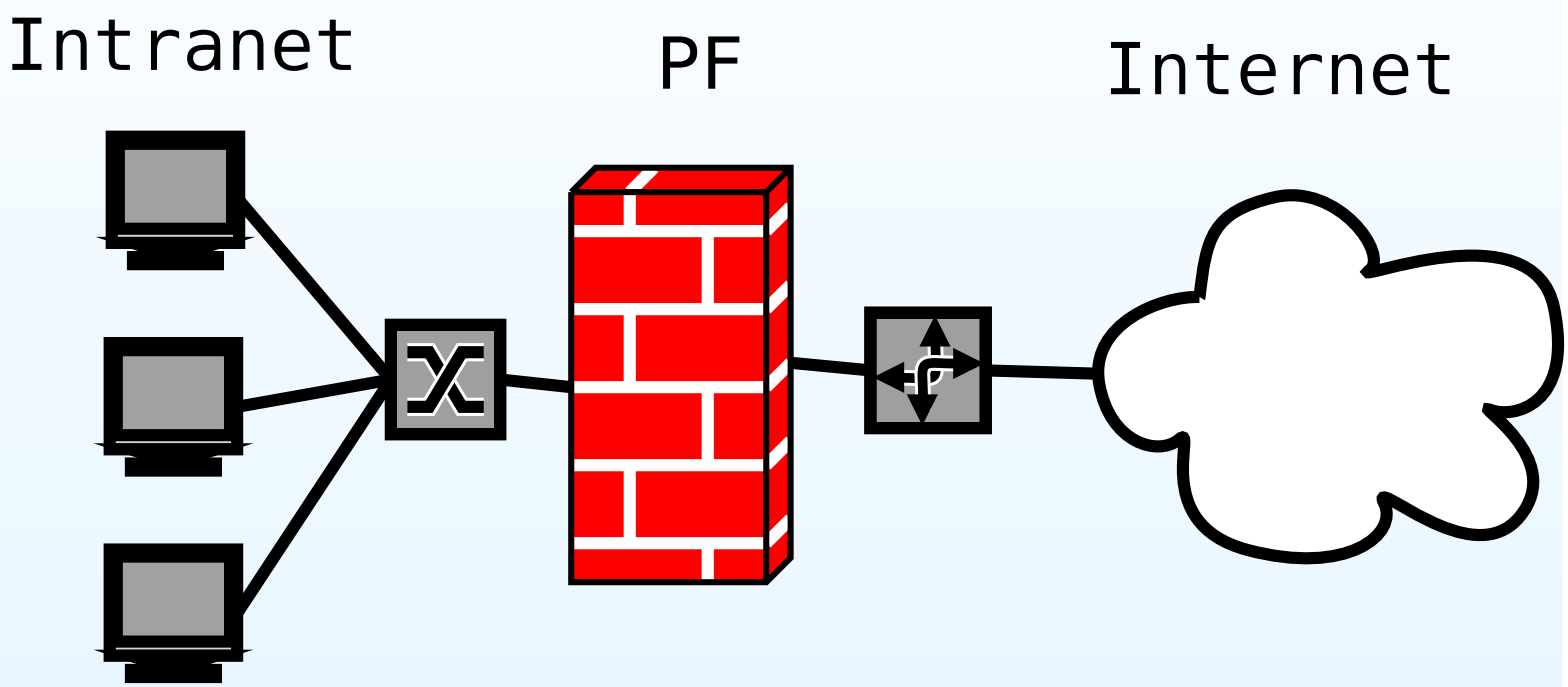
- play instinct :o)
- explore a remote network
- find backdoors
- check weaknesses
- prepare an attack
- fool IDS systems
- see which software your bank runs
- ...

Typical Targets

- Router / Firewalls / Packetfilter
- Intrusion Detection Systems
- Loghosts (to hide traces)
- servers - from the outside accessible (DMZ?)
- Client-Systems / Workstations
- Hardware-Systems (e.g. Access Points, Routers ...)

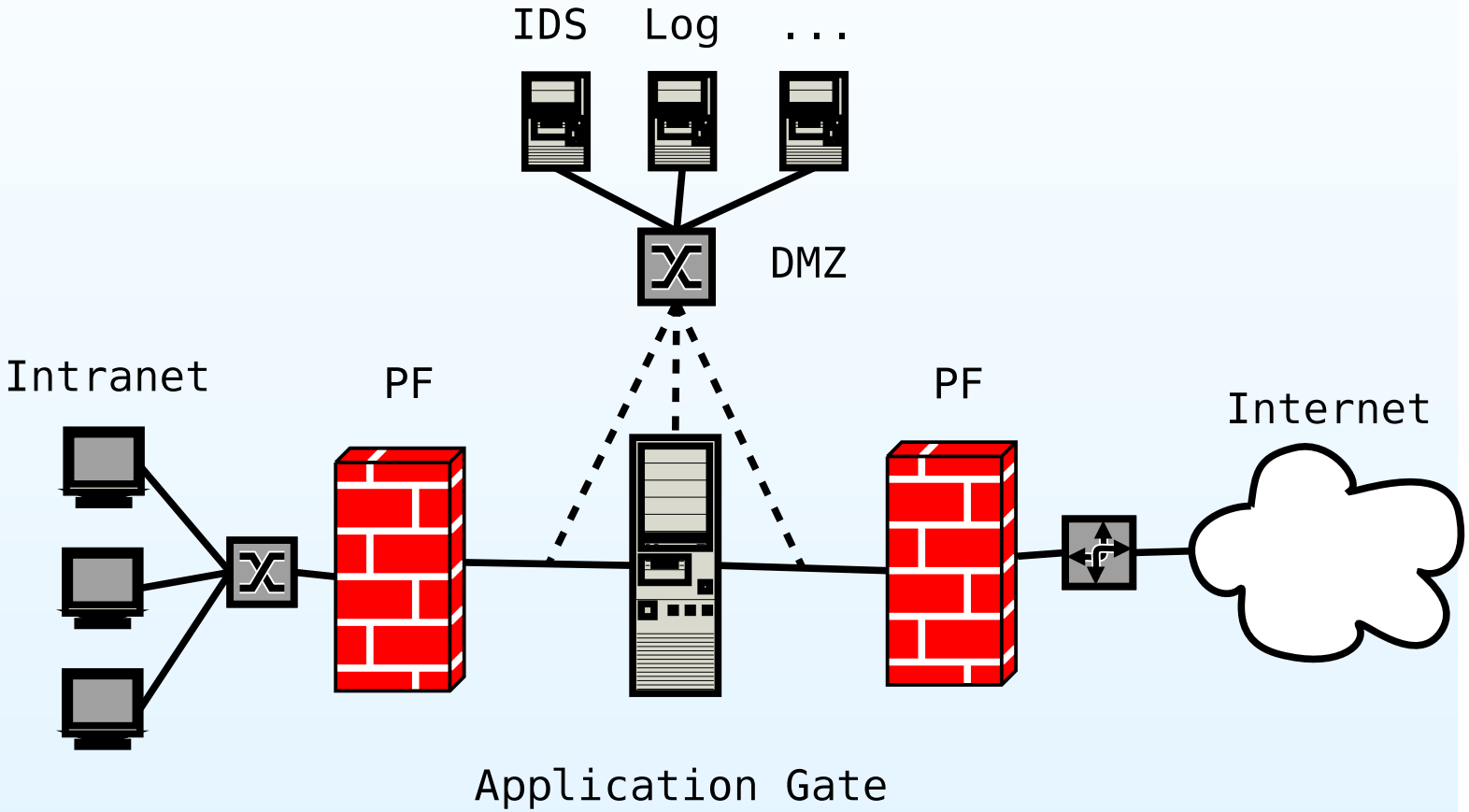
Structure of FW Systems

easy layout:



Structure of FW Systems

more complex layout(s):



Possible Attacks

$\Rightarrow$ attacker (we) located in the Internet
$\Rightarrow$ attacks performed from outside

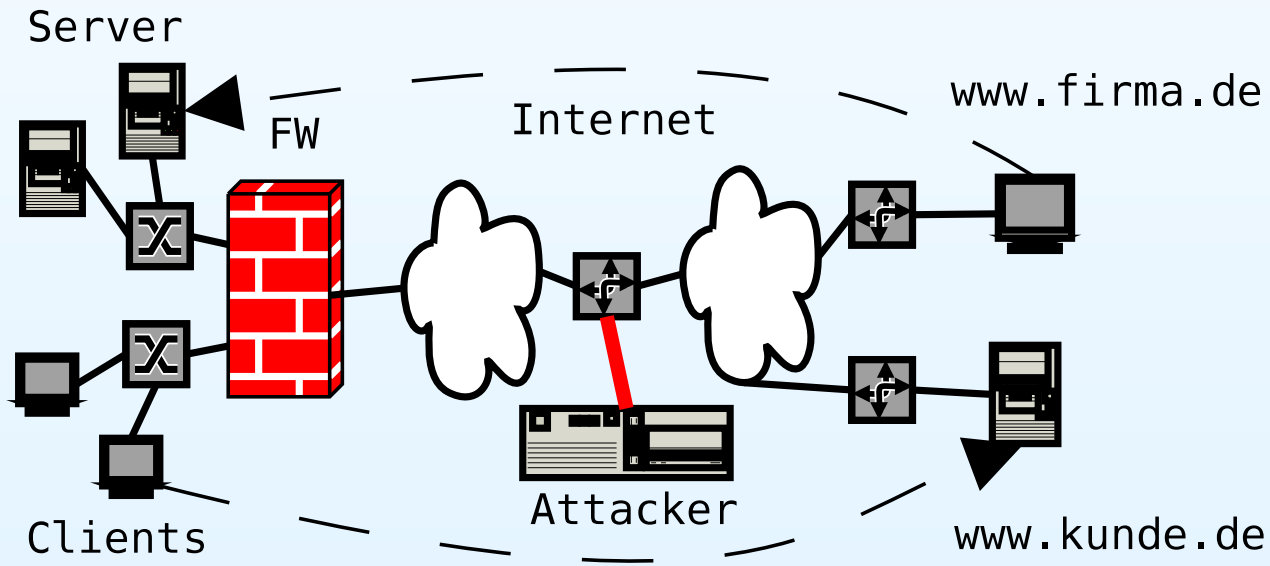
- passive analysis (e.g. sniffing)
- noticeable active analysis (e.g. scanning)
- hidden active analysis (e.g. fingerprinting)
- analysis of topology (e.g. firewalking, tracing)
- social engineering

Passive Analysis

Layer 2/3/4

⇒ different possibilities:

- passive fi ngerprinting (without sending anything)
  - Layer 4 (versions of used software products)
    - Payload Analysis (not widely used, no tools available)
  - Layer 2/3 (OS's TCP/IP implementation)
    - Header-Analysis (widely used, tools available)



Header-Analysis

- gives information about deployed topology:
  - TTL: OS usually starts with ”typical” values (255, 128, 64 ...) -> difference equals Hop-Count
  - be aware of exceptions (e.g. traceroute)!
- offered or used services e.g.:
  - analyse source or/and destination port

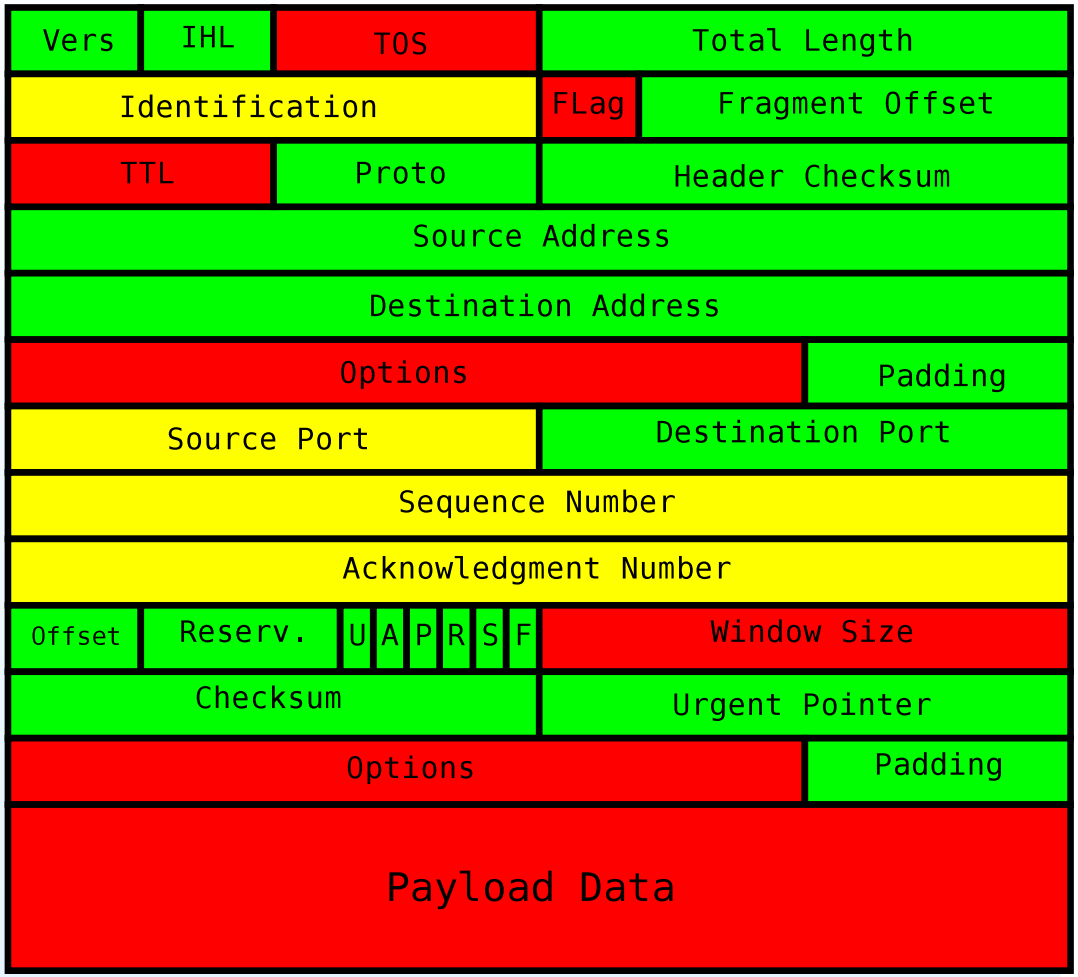
Header-Fields

information contained
in TCP/IP packets
(useable for analysis)

high

average

low/on information



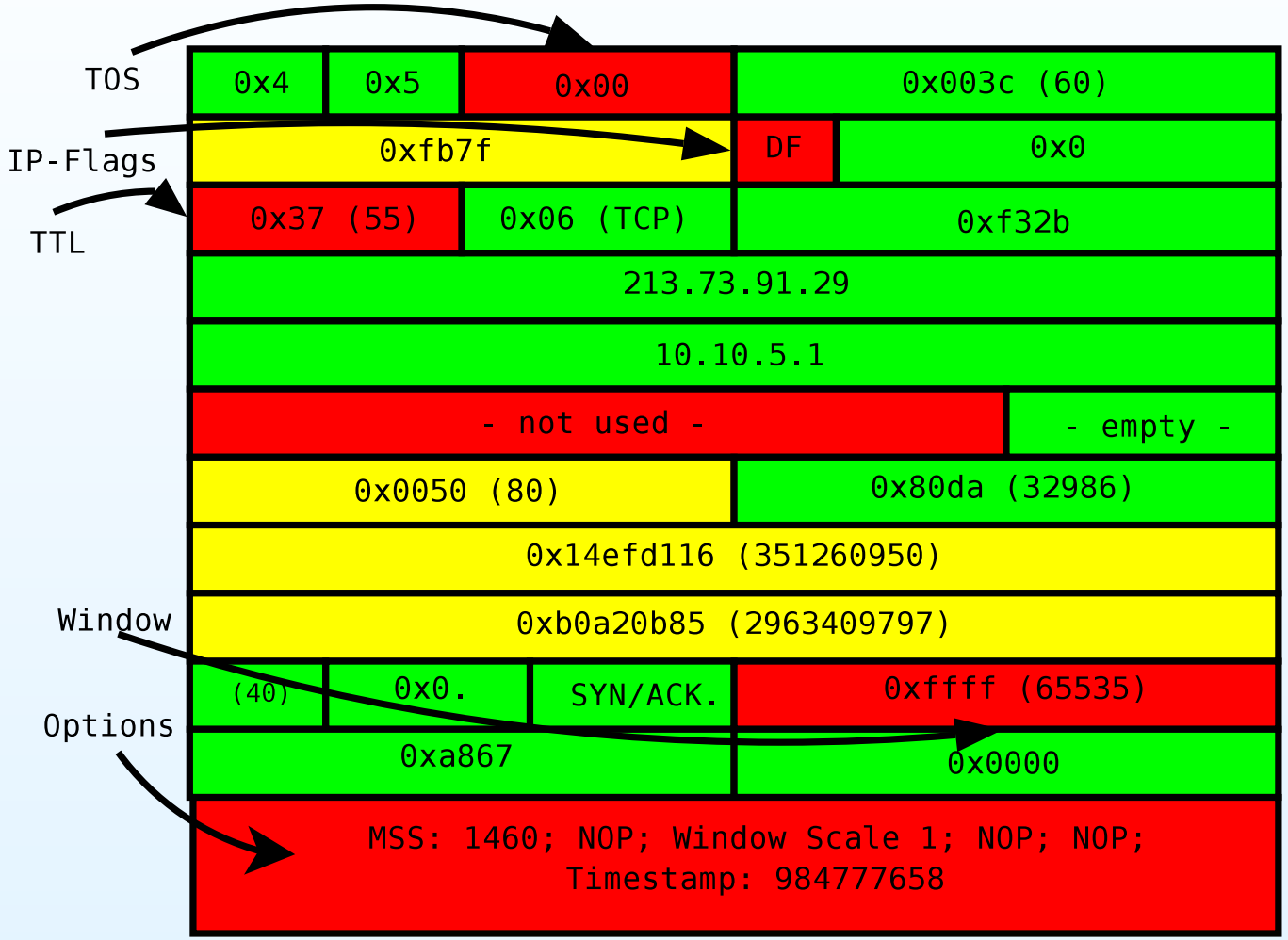
Header-Information

much information can be gained from the header fi elds:

Field	Location	Tools?	What?
TTL	IP	x	OS + Topology
Fragmentation	IP	x	OS + Topology
Header Length	IP	x	OS
TOS	IP	-	OS
ID	IP	-	OS + Traffi c
Source Port	TCP	-	OS + Traffi c
Window Size	TCP/Opt	x	OS
Max. Segmentsz.	TCP/Opt	x	OS
...	...	-	OS

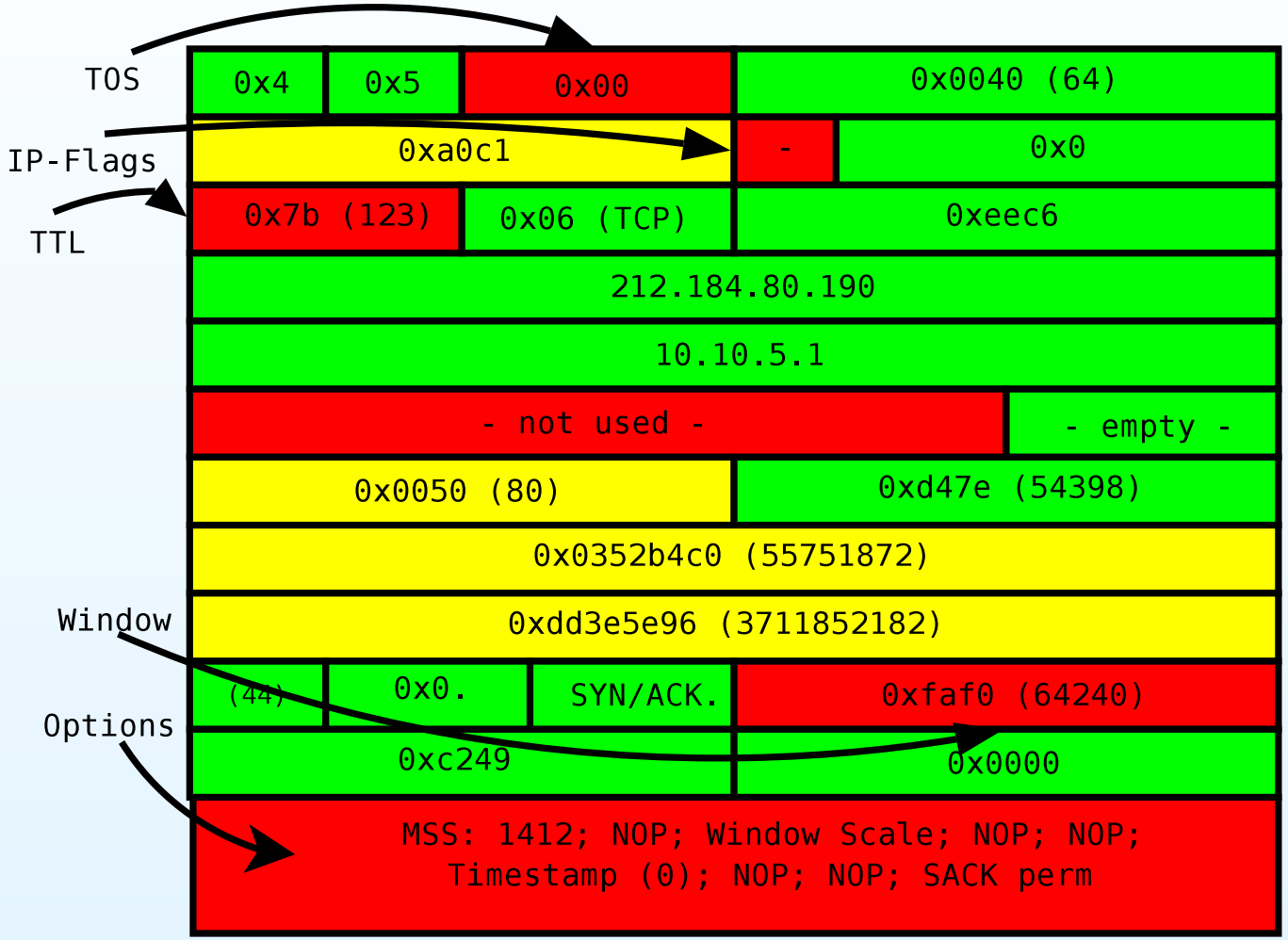
Header-Analysis (example)

SYN/ACK Header from www.ccc.de:80



Header-Analysis (example)

SYN/ACK Header from www.microsoft.de:80



Example

practical values:

OS	TOS	DF	TTL	Window	Options
Win2000	0	1	128	65535	tsval=0, SACK
Win98	0	1	128	8760	SACK
Linux 2.2	0	1	64	32210	tsval>0, SACK
Linux 2.4	0	1	64	5792	tsval>0, SACK
Linux 2.6	0	1	64	5792	tsval>0, SACK
FreeBSD 4.6	0	1	64	57344	tsval>0
FreeBSD 5.0	0	1	64	65535	tsval>0
OpenBSD 2.x	16	0	64	17520	tsval=0, SACK
...	...	...	...	...	...

More Examples

examples (p0f - SYN/ACK analysis):

- www.metro.de - Windows 2000 SP4
- www.ebay.de - unknown
- www.heise.de - NetApp Data OnTap 6.x
- www.microsoft.de:80 - Windows 2000 (SP1+) (fi rewall!)
- www.openbsd.org:80 - Solaris 7 (up: 2533 hrs)
- www.freebsd.org:80 - FreeBSD 4.6-4.8 (up: 9 hrs)
- www.mcafee.com:80 - Windows 2000 SP4
- www.georgewbush.com:80 - Windows 2000 SP4
- www.bundeskanzler.de:80 - Linux recent 2.4 (1) (up: 11405 hrs)
- www.nsa.gov:80 - Linux recent 2.4 (1) (up: 5664 hrs)
- www.dod.gov:80 - Linux recent 2.4 (up: 2804 hrs)
- ...

Summary

fi ngerprinting without sending any data

- utilizes imprecise standard defi nitions ...
- ... or deviations of OSes from standards (RFC)
- cumulative analysis of different header fi elds
- manually nearly impossible (huge information databases)
- $\Rightarrow$ automated tools (ettercap, siphon, p0f)
- BUT: very slow / imprecise! $\Rightarrow$ active analysis is more accurate
- new techniques (AI / Fuzzy Match) improve accurancy

Active Analysis

Layer 4 (Application Level)

sending packets and analysing the response

- ”classical manual banner grabbing”
  - e.g. FTP, HTTP, POP, IMAP, SMTP, SSH, NNTP, Finger ...
- binary analysis
  - e.g. /bin/ls from FTP server (which binary format (ELF, COFF) $\rightarrow$ OS)
- well known ports
  - e.g. 80 $\rightarrow$ HTTP, 22 $\rightarrow$ SSH, ...
- $\Rightarrow$ easy to prevent/fake
  - e.g. 222 $\rightarrow$ SSH (ipcop)
- $\rightarrow$ application fi ngerprinting (sending special requests, evaluate (error) responses)
  - automated tools: thc-amap, nmap (-sV)

Layer 2/3 (OS Level)

- send ”special” crafted IP packets and analyse the response
- $\Rightarrow$ easy to detect
  - e.g. IDS notifi es attempts (see portscan)
- $\Rightarrow$ fi rewall can block results
  - e.g. stateful fi rewalls block connectionless FIN packets
- $\Rightarrow$ fi rewall can modify results
  - e.g. change TTL, TOS or fi lter out Options with iptables

OS Detection Tools

- de-facto standard: nmap from Fyodor
- a lot of active fi ngerprinting techniques (FIN to open port, ISN Sampling, ICMP Tests, TCP Options, Fragmentation Handling ...)
- is recognized by many IDS or packetfi lters and can be fi ltered easily
- nmap needs one opened and one closed TCP-Port + one closed UDP-Port (often not possible $\rightarrow$ fi rewall)
- $\Rightarrow$ other metrics have to be found
- others:
  - xprobe2 - fuzzy logic, similar to nmap
  - queso - no further development

Advanced Methods

Old Techniques

Inverse Mapping

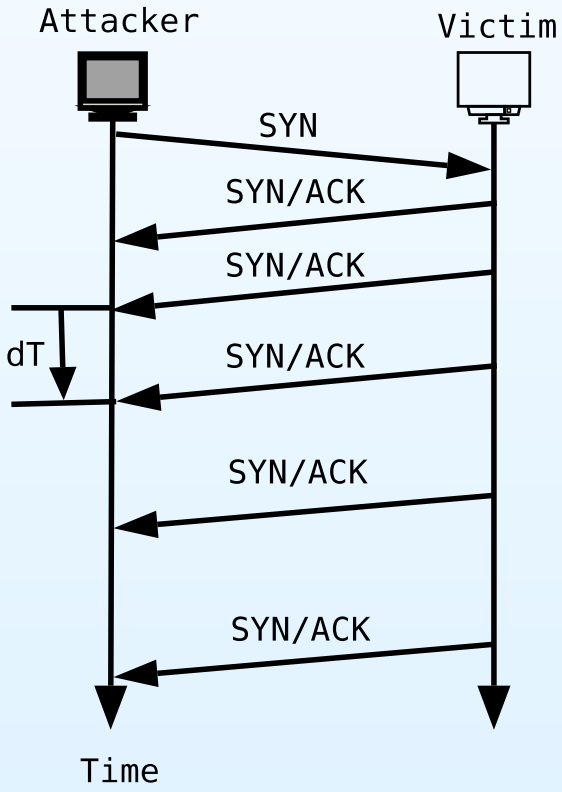
- normal ”inconspicuous” packets to different IP’s
- e.g. FIN, ACK, DNS-Reply
- $\longrightarrow$ only for stateless fi rewalls / IDS
- non existing hosts: router sends ICMP host unreachable -> attacker concludes network structure / used addresses

Slow Scan

- packet-rate < 1 packet/hour
- very hard to detect automatically

RING

- $\Rightarrow$ RING (**R**emote **I**dentifi cation **N**ext **G**eneration)
- TCP retransmissioncount and -time is used!
- deviations from RFC2988 (defi nes a retransmission algorithm)
- tools: snacktime, Cron-OS, tbit

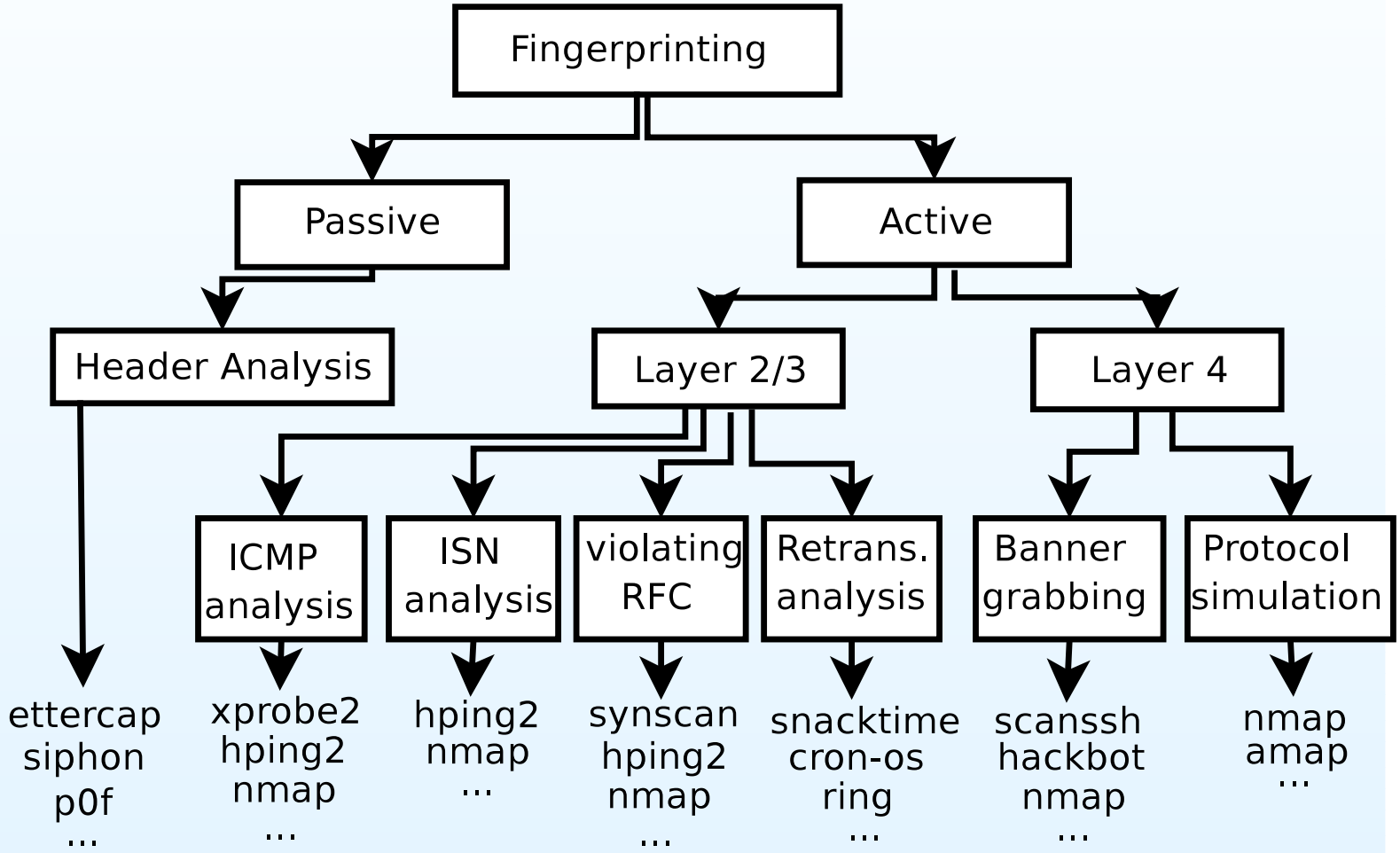


RING - Examples

snacktime evaluation:

- www.metro.de:80 - Windows_2000_Server_SP3
- www.ebay.de:80 - Windows_XP_Professional
- www.heise.de:80 - no retransmission
- www.microsoft.de:80 - no retransmission
- www.openbsd.org:80 - Linux_2.4.9_Alpha (???)
- www.freebsd.org:80 - Generic_BSD_Stack
- www.mcafee.com:80 - no retransmission
- www.georgewbush.com:80 - RST after fi rst retrans!
- www.bundeskanzler.de:80 - Linux_2.4.18
- www.nsa.gov:80 - no retransmission
- www.dod.gov:80 - Linux_2.4.18
- ...

Overview Fingerprinting

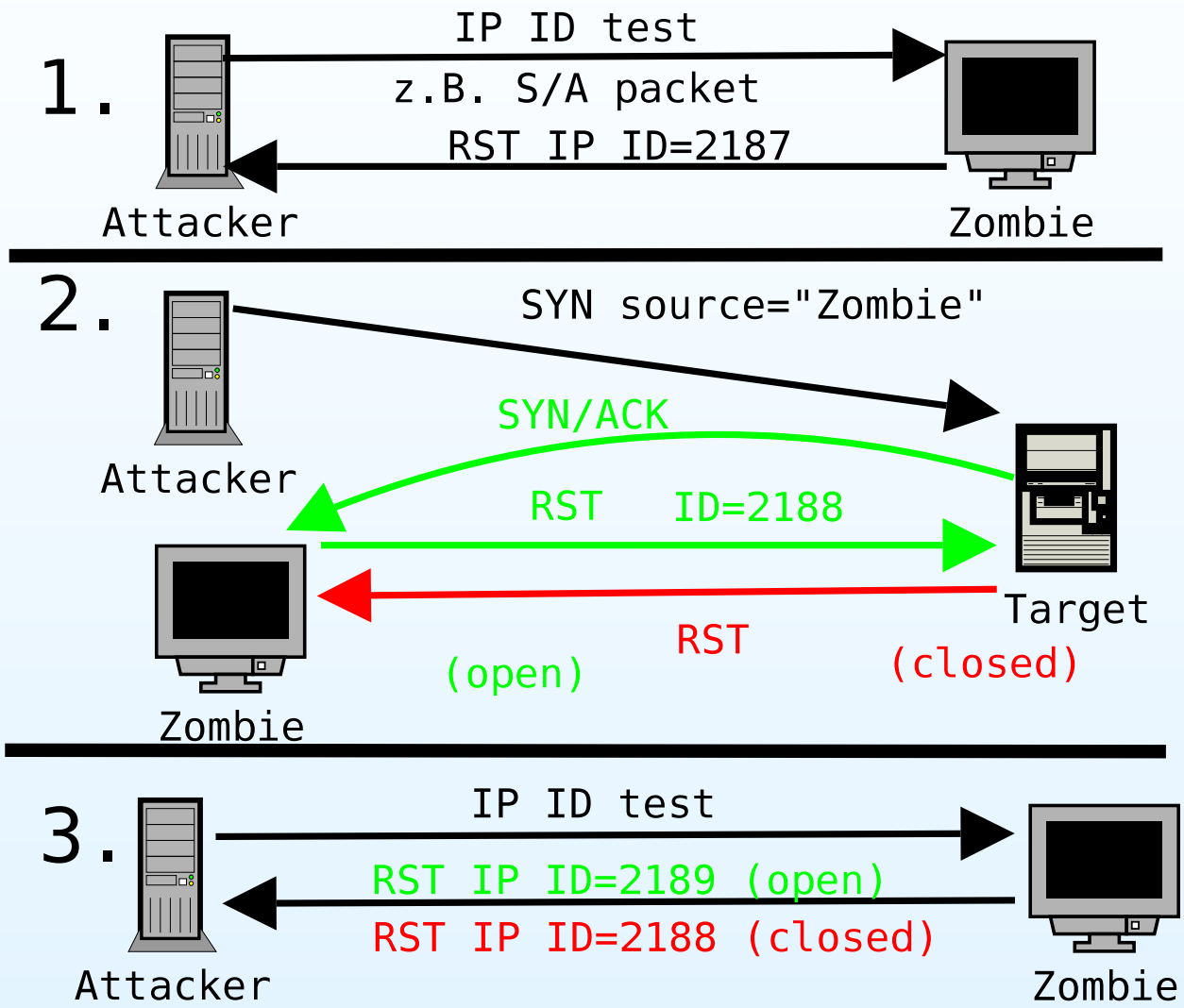


Idle Scan

- no packet sent directly → sending through ”zombie” hosts
- using predictable IP fragment-numbers
- suitable for testing IP-based fi lter-rules
- IDS sees ”zombie” as attacker
- tool: nmap (-D) - decoy scan

protect own hosts from being used as zombies:
- stateful fi rewall
- OS with unpredictable or constant fragment-numbers

Idle Scan - Example



Finding Zombies

chance to fi nd zombies is relatively high!

```
htor@archimedes: $ hping2 -SA -p 80 www.tu-dresden.de
len=46 ip=141.30.66.151 ttl=46 DF id=216 sport=80 ...
len=46 ip=141.30.66.151 ttl=46 DF id=217 sport=80 ...
len=46 ip=141.30.66.151 ttl=46 DF id=218 sport=80 ...
htor@archimedes: $ ping -c1 www.tu-dresden.de
htor@archimedes: $ hping2 -SA -p 80 www.tu-dresden.de
len=46 ip=141.30.66.151 ttl=46 DF id=220 sport=80 ...
```

$\rightarrow$ IP-ID counts up globally!

more useable zombies:

- www.tu-dresden.de
- 64.203.100.217 (hosting georgewbush.com :)
- mx2.freebsd.org
- www.openbsd.org
- ...

Firewalking

= analyze the hosts behind a firewall
e.g. test if IP is up:

- SYN-packets are dropped by firewall
- SYN/ACK not (only stateful FWs)
- use SYN/ACK packets to scan IP's behind FW
- ruleset of the FW can be guessed

Firewalking - Example

e.g. portscan of wald.informatik.tu-chemnitz.de (134.109.184.40):
archimedes: # hping2 wald.informatik.tu-chemnitz.de -p 22 -A
HPING wald.informatik.tu-chemnitz.de (eth0 134.109.184.40): A set ...
len=46 ip=134.109.184.40 ttl=55 DF id=0 sport=22 flags=R seq=0 win=0 rtt=64.3 ms
len=46 ip=134.109.184.40 ttl=55 DF id=0 sport=22 flags=R seq=1 win=0 rtt=64.8 ms
⇒ port 22 (ssh) open

archimedes: # hping2 wald.informatik.tu-chemnitz.de -p 81 -A
HPING wald.informatik.tu-chemnitz.de (eth0 134.109.184.40): A set ...
ICMP Port Unreachable from ip=134.109.184.40 name=wald
ICMP Port Unreachable from ip=134.109.184.40 name=wald
⇒ port 81 closed

are the pool-computers switched on during the weekend?
HPING donau.hrz.tu-chemnitz.de (eth0 134.109.72.177): SA set ...
len=46 ip=134.109.72.177 ttl=55 DF id=0 sport=82 flags=R seq=0 win=0 rtt=62.5 ms
len=46 ip=134.109.72.177 ttl=55 DF id=0 sport=82 flags=R seq=1 win=0 rtt=65.4 ms
⇒ yes ;o)

Firewalking (2)

Cambridge Technology Partners: ”A Traceroute-Like Analysis of IP Packet Responses to determine Gateway Access Control Lists.”

- newer development
- phase 1: determine hop-count to FW (Gateway) = HC(FW)
- phase 2: packets with TTL=HC(GW)+1 for SYN scan
- if HC(target) > HC(GW)+1 $\rightarrow$ no packet reaches target
- open port: ICMP Time exceed
- closed Port: no answer (timeout)
- prevention:
  - drop outgoing ICMP time exceed packets
  - application proxy

Prevention

TCP/IP Stack Tuning

Linux (adjustment of kernel parameters):

- /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts = 1
- /proc/sys/net/ipv4/conf/*/accept_source_route = 0
- /proc/sys/net/ipv4/conf/*/rp_fi lter = 1 (prevent spoofi ng)
- /proc/sys/net/ipv4/ipfrag_high_thresh = ? (fragments will be dropped when this valua is reached - Rose Attacks?)
- /proc/sys/net/ipv4/ipfrag_low_thresh = ? (fragments will be accepted again under this level)
- /proc/sys/net/ipv4/conf/*/log_martians (log packets with unusual addresses)
- /proc/sys/net/ipv4/ip_default_ttl = ? (confuses simple OS detection)

$\Rightarrow$ appropriate values on other Operating Systems (e.g. sysctl with BSD)

Linux Kernel Modifications

⇒ Grsecurity

- larger entropy pools (better random numbers)
- randomized TCP Initial Sequence Numbers (confuses OS detection)
- randomized IP IDs (prevents ”zombie”-scans)
- randomized TCP source ports (confuses OS detection)

⇒ IP Personality

- changeable characteristics to pretend other TCP/IP stacks
- nice tool, but only up to kernel 2.4.18 :-(

⇒ own modifi cations in kernel sources

- z.B. no answer to illegal packets: /usr/src/linux/net/ipv4/*
- change the window size: /usr/src/linux/include/net/tcp.h (MAX_TCP_WINDOW)
- ...

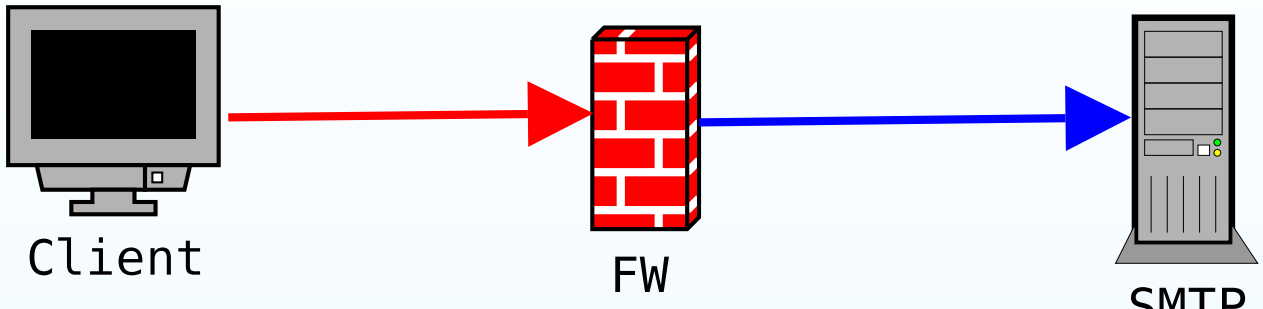
Deep Packet Inspection

Firewall Evolution

- Firewall - protect
- IDS - observe
- today: manual adding of fi rewall rules after notifi cation from IDS
- problem: fast attacks (Code Red, Nimda)
- $\longrightarrow$ deep packet inspection = FW + IDS coupled
- exploits and scan attempts can be interrupted automatically

e.g. layer 4 monitoring
(see PIX ”fi xup” command) $\Rightarrow$ next slide

DPI - Example



State Table:
SMTP: client:1025 - SMTP:25

Protokoll (SMTP):

Questions

Questions?

Thank You

for your attention!

Torsten Höfler

<htor@cs.tu-chemnitz.de>

Sources

- own experience :o)
- Laurent Joncheray: Simple Active Attack Against TCP, 1995
- Kevin Timm: Passive Network Traffi c Analysis, 2003
- Lance Spitzner: Passive Fingerprinting, 2000
- Fyodor: Remote OS detection via TCP/IP Stack FingerPrinting, 1998
- Intranode Research: RING - Full Paper, 2002
- Synnergy Networks: Advanced Host Detection, 2001
- Cambridge Technology Partners: Firewalking, 1998
- Ido Dubrawsky: Firewall Evolution - Deep Packet Inspection, 2003