

**dn**

---

*Systems*

# Effiziente Filter gegen Kinderpornos und andere Internetinhalte

Lukas Grunwald

DN-Systems GmbH

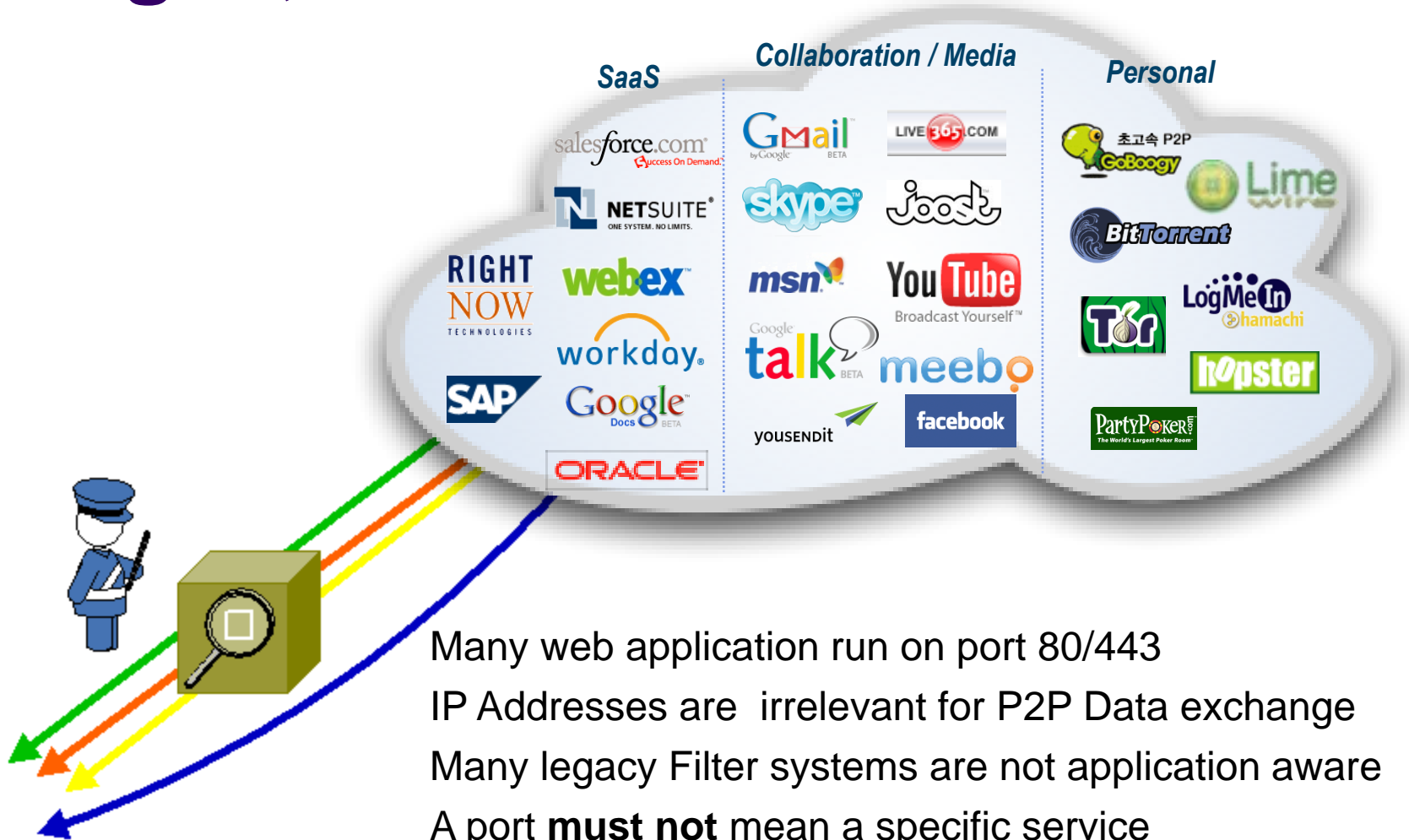
CeBIT 2010- Heise Forum

2010 Hannover

# Why Filtering

- Slow down distributed denial of service attacks (dDoS)
- Filter dangerous attacks from the networks (Ping of Death, XMAS-Scans, Conflicker...)
- Filter illegal content
  - Against child pornography
  - Protection of minors
- Fight against SPAM, other Malware

# Applications have changed, Filter has not ..



Many web application run on port 80/443  
IP Addresses are irrelevant for P2P Data exchange  
Many legacy Filter systems are not application aware  
A port **must not** mean a specific service  
**Proxy and tunneling** runs independent from ports

# Different Applications

- Unwanted functions and features
  - P2P-Filesharing (mostly illegal content)
  - Malware (Conflicker, Port-Scans, dDoS attacks)
- Unwanted content
  - SPAM, HTTP-Floods, Website Mirrors
- Illegal Content
  - Child Pornography
  - National Propaganda
- Tunnel and Proxy to bypass filter

# Filter that won't work

- Just block the DNS Query of an URL / Website
  - An other DNS Server or a local lookup
    - /etc/hosts
      - filtered-host.com 1.2.3.4
    - Own secure DNS-Server
  - Filter by Obscurity
  - IP-Address direct access needs to be blocked too
  - Access via IP 1.2.3.4

# Filter that won't work

- Just block an UDP / TCP port
  - Popular with US-ISPs with large subscriber networks
  - Blocking port 25 did not resolve the spam problem
  - Alternative mail server run now of different ports
  - Filter by Obscurity
  - Mail can still be injected by HTTP / Web-Services

# Basis Knowledge

| bit offset        | 0-3                              | 4-7           | 8-15                    | 16-18           | 19-31           |
|-------------------|----------------------------------|---------------|-------------------------|-----------------|-----------------|
| 0                 | Version                          | Header length | Differentiated Services | Total Length    |                 |
| 32                | Identification                   |               |                         | Flags           | Fragment Offset |
| 64                | Time to Live                     |               | Protocol                | Header Checksum |                 |
| 96                | Source Address                   |               |                         |                 |                 |
| 128               | Destination Address              |               |                         |                 |                 |
| 160               | Options ( if Header Length > 5 ) |               |                         |                 |                 |
| 160<br>or<br>192+ | DATA                             |               |                         |                 |                 |

# TCP-Protocol

|   |   |   |   |   |          |   |   |   |             |             |             |             |             |             |             |             |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|---|---|---|---|---|----------|---|---|---|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| B<br>i<br>t<br>o<br>f<br>f<br>s<br>e<br>t | 0   | 1 | 2 | 3 | 4        | 5 | 6 | 7 | 8           | 9           | 10          | 11          | 12          | 13          | 14          | 15          | 16               | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|   | Source port   |   |   |   |          |   |   |   |             |             |             |             |             |             |             |             | Destination port |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 3<br>2                                    | Sequence number                                     |   |   |   |          |   |   |   |             |             |             |             |             |             |             |             |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 6<br>4                                    | Acknowledgment number                               |   |   |   |          |   |   |   |             |             |             |             |             |             |             |             |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 9<br>6                                    | Data offset   |   |   |   | Reserved |   |   |   | C<br>W<br>R | E<br>C<br>E | U<br>R<br>G | A<br>C<br>K | P<br>S<br>H | R<br>S<br>T | S<br>Y<br>N | F<br>I<br>N | Window Size      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 1<br>2<br>8                               | Checksum  |   |   |   |          |   |   |   |             |             |             |             |             |             |             |             | Urgent pointer   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 1<br>6<br>0<br>..<br>.                    | APPLICATION DATA (only Application Header and Data) |   |   |   |          |   |   |   |             |             |             |             |             |             |             |             |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

● Src/Dst port are not evident

● User can define ports for any application

● Only well known ports 1-1024 are common sense

● Application data tells it all

# Deep Packet Inspection

- Deep Packet Inspection (DPI)
- Look inside the payload / application data
- Does not rely on header information



# Ngrep – simple DPI

- ngrep (Network grep)
  - is a simple open-source tool to do DPI on clear text protocols
  - HTTP, HTTP-Proxy, SMTP, POP3, IMAP ....
  - <http://ngrep.sourceforge.net/>

# Example DPI HTTP

```
# ngrep -d eth0 port 1080
GET /example.jpg HTTP/1.1..Referer: http://www.unsecure.info/index.htm
  l..Connection: Keep-Alive..Host: www.unsecure.de..User-Agent: Mozilla/4.5
  (compatible; HTTrack 3.0x; Windows 98)..Accept
  : image/png, image/jpeg, image/pjpeg, image/x-xbitmap, image/svg+xml,
  image/gif;q=0.9, */*;q=0.1..Accept-Language: en, en
  , *..Accept-Charset: iso-8859-1, iso-8859-*;q=0.9, utf-8;q=0.66, *;q=0.33..Accept-
  Encoding: gzip, identity;q=0.9....
```

```
T 78.8.17.65:3129 -> 193.108.181.189:1080 [AP]
```

```
  GET /index.html HTTP/1.1..Host: www.unsecure.de..User-Agent: Mozilla/5.0
  (Windows; U; Windows NT 5.1; pl; rv:1.8.1.20) Gecko/20081217 Firefox/2.0.0.20
  (.NET CLR 3.5.30729)..Accept: image/png,*/*;q=0.5..Accept-Language: pl,en-
  us;q=0.7,en;q=0.3..Accept-Encoding: gzip,deflate..Accept-Charset: ISO-8859-2,utf-
  8;q=0.7,*;q=0.7..Keep-Alive
  : 300..Connection: keep-alive..Referer: http://www.unsecure.de/index.html....
```

# Example SMTP DPI

```
# ngrep -d eth0 port 25
```

```
T 193.108.181.246:25 -> 189.33.16.223:22530 [AP]
```

```
  220-mail.unsecure.de ESMTP DN-SystemsMailer.el Fri, 19 Feb 2010 13:03:33  
+0100..220-(c) 2008,2009 by L. Grunwald..220-
```

```
  RFC<822,1149,1413,1738,2255,2487,2505,2822>..
```

```
T 189.33.16.223:22530 -> 193.108.181.246:25 [AP]
```

```
  HELO bruno..
```

```
T 193.108.181.246:25 -> 189.33.16.223:22530 [AP]
```

```
  250 mail.unsecure.de Hello bd2110df.virtua.com.br [189.33.16.223].
```

```
T 189.33.16.223:22530 -> 193.108.181.246:25 [AP]
```

```
  MAIL From: <vdah_wmol_r_f_i@x-paste.de>..
```

```
T 193.108.181.246:25 -> 189.33.16.223:22530 [AP]
```

```
  250 OK..
```

```
T 189.33.16.223:22530 -> 193.108.181.246:25 [AP]
```

```
  RCPT TO: <mmcClendonmy@x-paste.de>..
```

# OpenDPI

- A software based Open-Source engine
  - <http://code.google.com/p/opendpi/>
  - Google Summer of Code project
  - Most classifiers are packet length based
- Works with PCAP format

# Example: OpenDPI Surf

```

rxvt
[usr/local/src/openspi-1.1.1/src/examples]/- \> ./dpi_pcap -f surf.pcap

WARNING: packet capture size is smaller than packet size, DETECTION MIGHT NOT WORK CORRECTLY
OR EVEN CRASH

pcap file contains
  ip packets: 7329          of 7398 packets total
  ip bytes:   6285737
  unique ids: 11
  unique flows: 90

detected protocols:
  unknown      packets: 165      bytes: 11770      flows: 0
  DNS          packets: 140      bytes: 18013      flows: 35
  HTTP        packets: 1020     bytes: 485270     flows: 46
  Flash       packets: 5994     bytes: 5768210    flows: 7
  Windowsmedia packets: 10       bytes: 2474       flows: 2

{bash}-(Sat Feb 20 13:26:28)-<root@ophelia>
[usr/local/src/openspi-1.1.1/src/examples]/- \>

```

# OpenDPI: Fail with Skype

```

rxvt
{bash}-(Sat Feb 20 13:27:22)-<root@ophelia>
[/usr/local/src/openspi-1.1.1/src/examples]/- \> ./dpi_pcap -f skype.pcap

pcap file contains
  ip packets:      701          of 729 packets total
  ip bytes:        106196
  unique ids:      194
  unique flows:    207

detected protocols:
  unknown          packets: 662          bytes: 100324      flows: 194
  DNS              packets: 4            bytes: 645         flows: 1
  HTTP            packets: 7            bytes: 1717        flows: 1
  SSDP            packets: 8            bytes: 1396        flows: 4
  ICMP            packets: 20           bytes: 2114        flows: 7

{bash}-(Sat Feb 20 13:27:23)-<root@ophelia>
[/usr/local/src/openspi-1.1.1/src/examples]/- \>

```

# Filter only on the content

- Collect all relevant meta-data on layer 2-5
  - Login / Logout time, subscriber ID, Radius ID, phone number, called peer
- Collect User-Agent ID and Fingerprint ID
  - Browser Request-String
  - OS-Fingerprint, end station polling
- Collect full user data of evidence rich application
  - e.g. full EMAIL Header and Body, all downloads,..

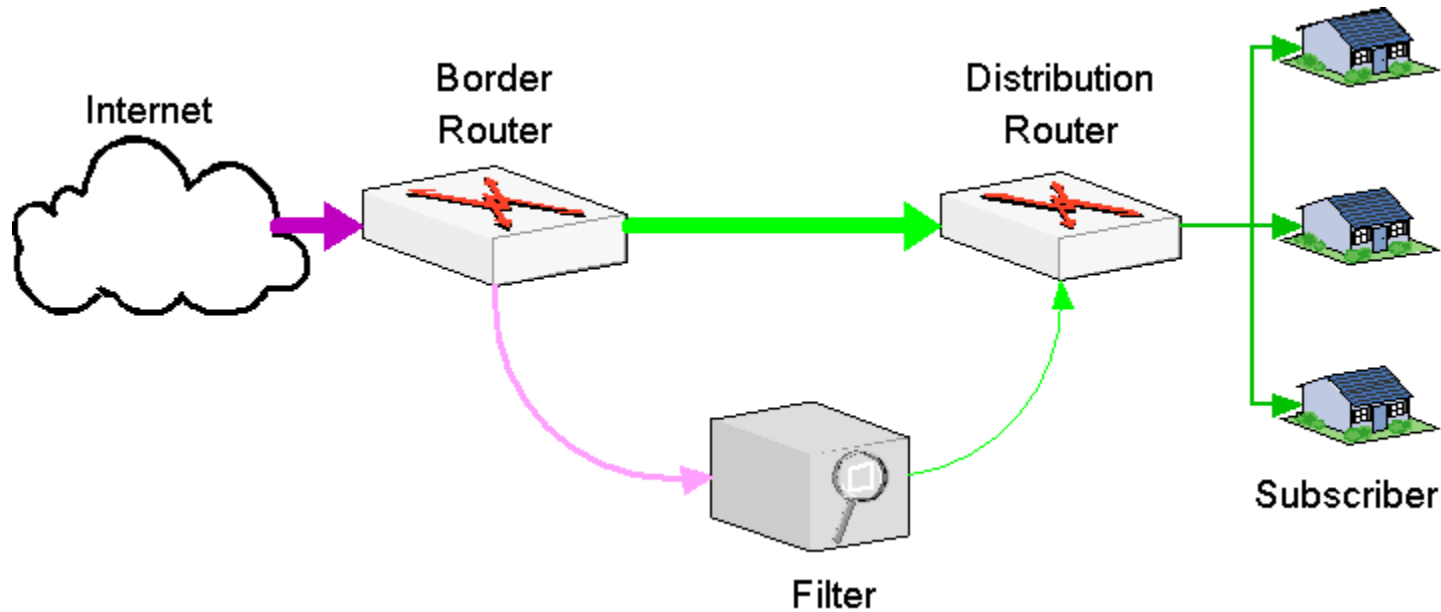
# Hardware Based Solution

- Most DPI implementations are software solutions
- Software is too slow for real time switching
- An intelligent hardware switch is needed for real time DPI decision
- Hardware can filter unwanted and malicious protocols, and split packets from the different clearance levels to different probes

# Offload your Backbone

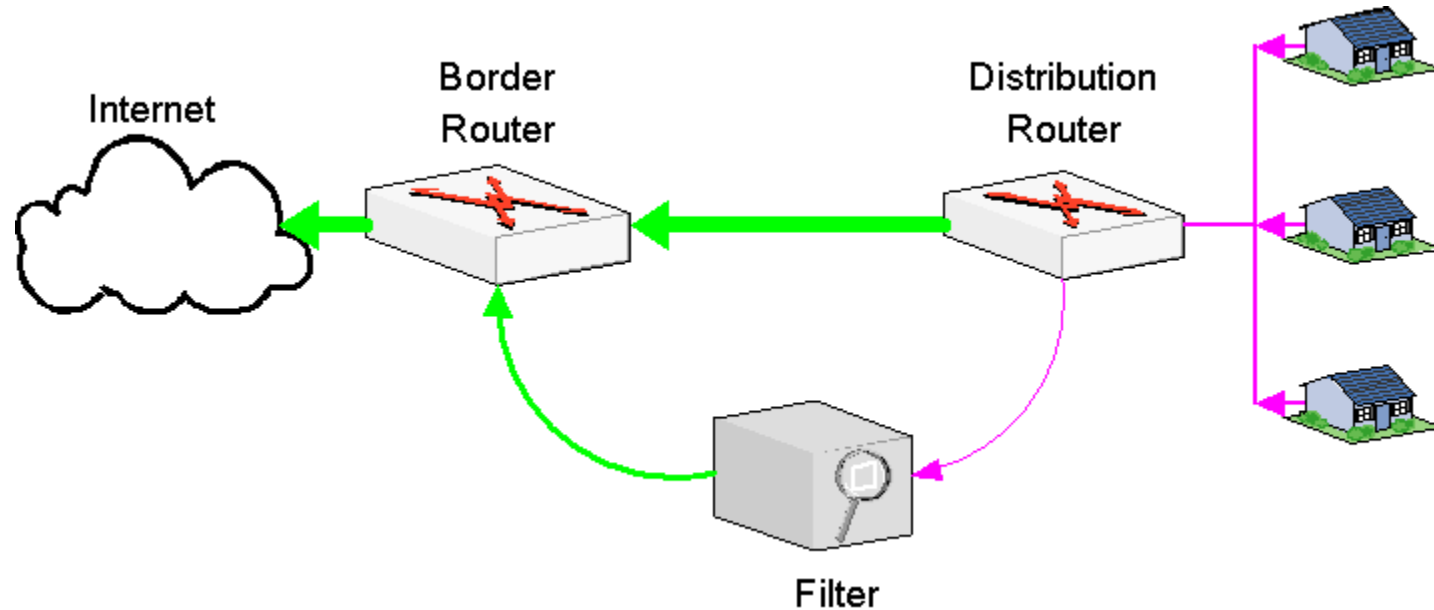
- A DPI Solution is very expensive for a Backbone 100GBit and more per Interface
- Border router don't have processing power / time for DPI and routing decision
- Filter decision could often be done by an IP-address
  - SPAM IP-Blacklists
  - Child pornography Host-IPs known from DNS

# Split good and bad packets



- Only suspicious packets are routed to an Hardware DPI Systems
- Other clear traffic passes inspection

# Filter Subscriber Traffic



- Reverse requests filter from Subscribers as well

# Conclusions

- Powerful filter can protect subscribers as well as services in the Internet
- Simple Filter (IP, Port, DNS) are not good enough for this purpose
- DPI and Policy Based Routing is needed for high traffic backbones
- dDoS and SPAM protection vs. censorship and rouge regime methods

**Thank you, keep in mind ...**



**You need the right tools ...**