Iptables

Ein Paketfilter auf Linux-Basis

Andreas Leibrock

UnixAg - FH ZW

25.03.2006

Unix& Linux DAY

Begriffe

Begriffe

- Firewall.

Begriffe

- Firewall: Konzept.

Begriffe

- Firewall: Konzept.
- Paketfilter

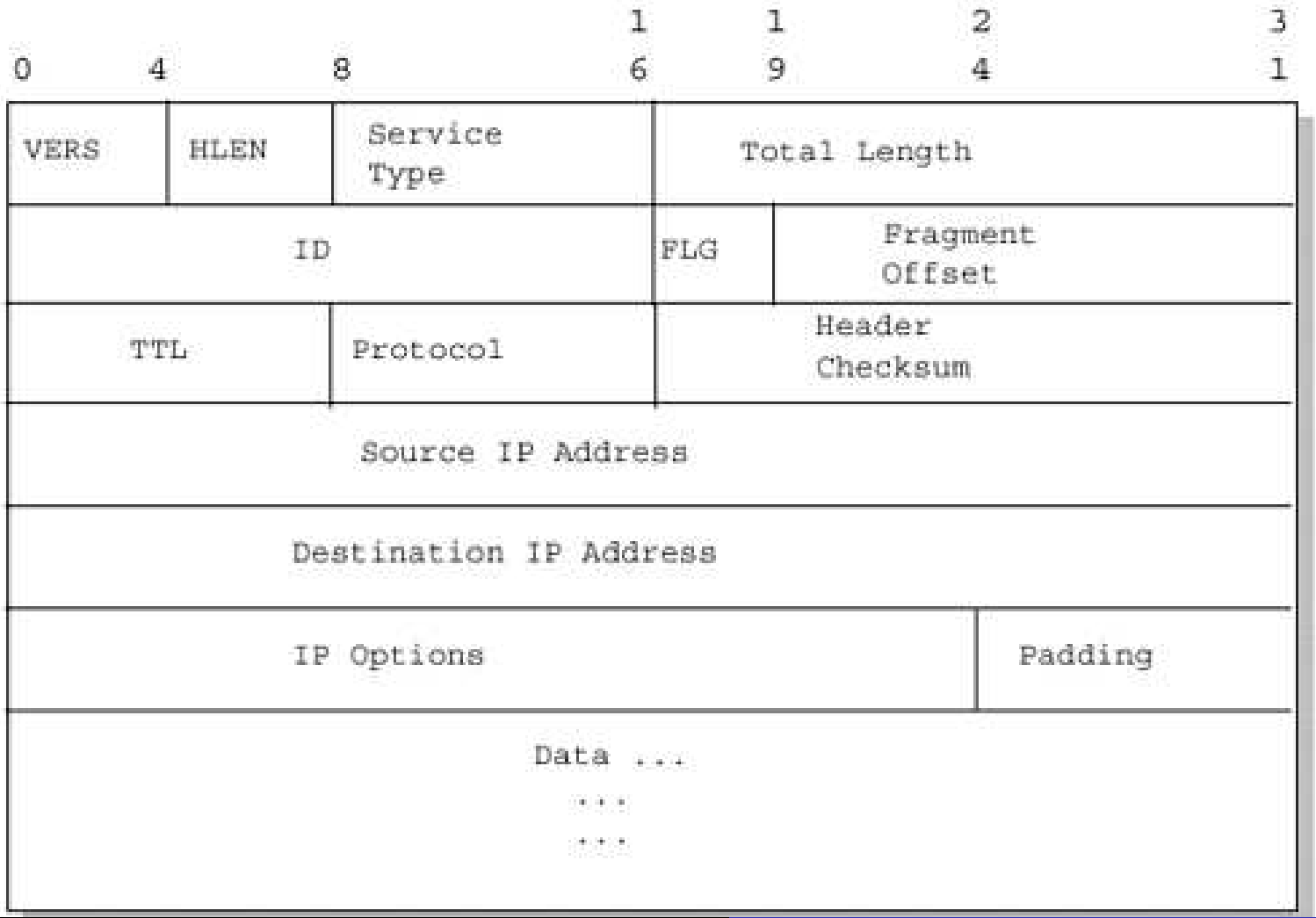
Begriffe

- Firewall: Konzept.
- Paketfilter
- Application Level Filter

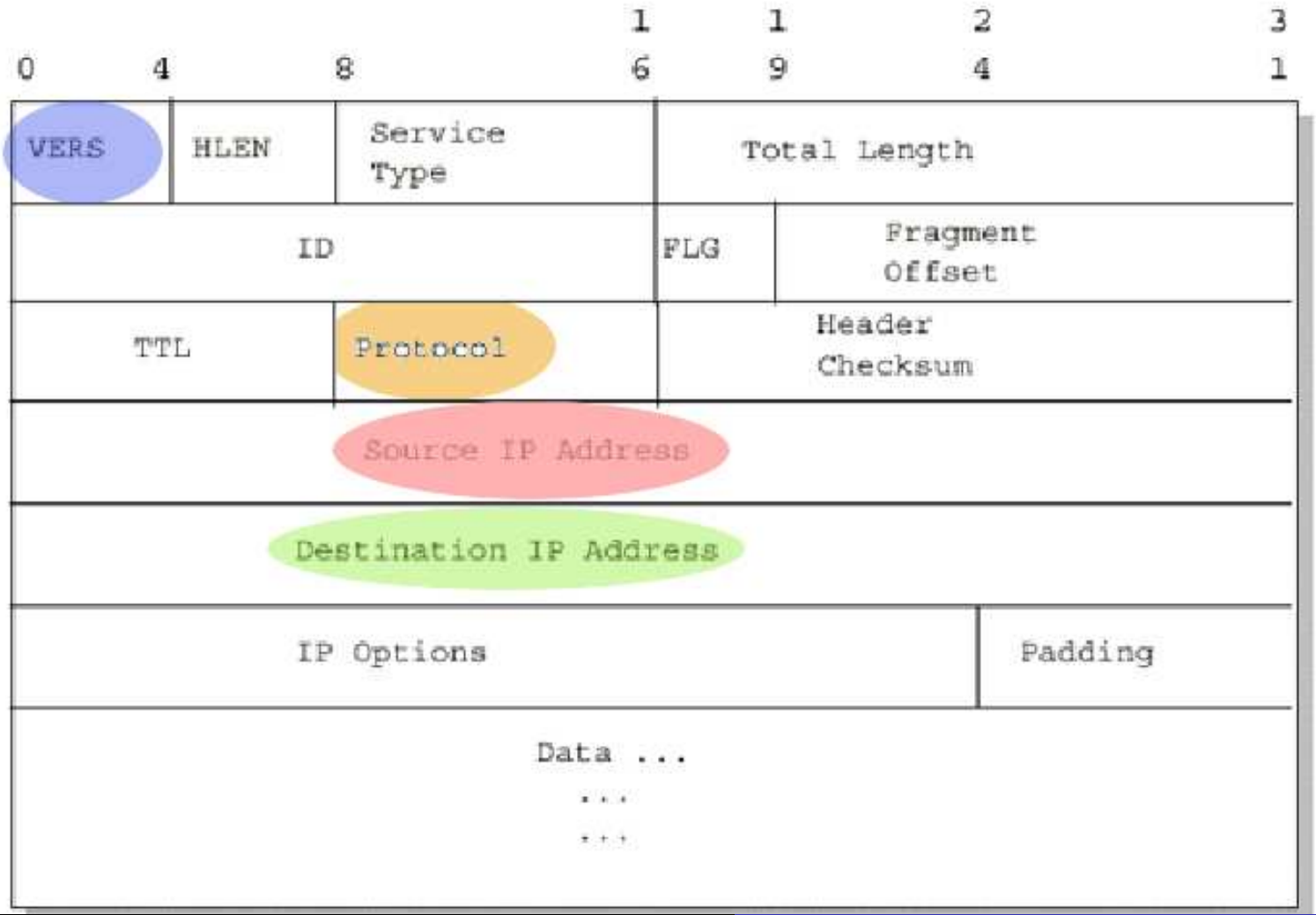
Begriffe

- Firewall: Konzept.
- Paketfilter
- Application Level Filter: Proxy

ip


0
4
8
16
19
24
31
VERS
HLEN
Service Type
Total Length
ID
FLG
Fragment Offset
TTL
Protocol
Header Checksum
Source IP Address
Destination IP Address
IP Options
Padding
Data ...
...
...

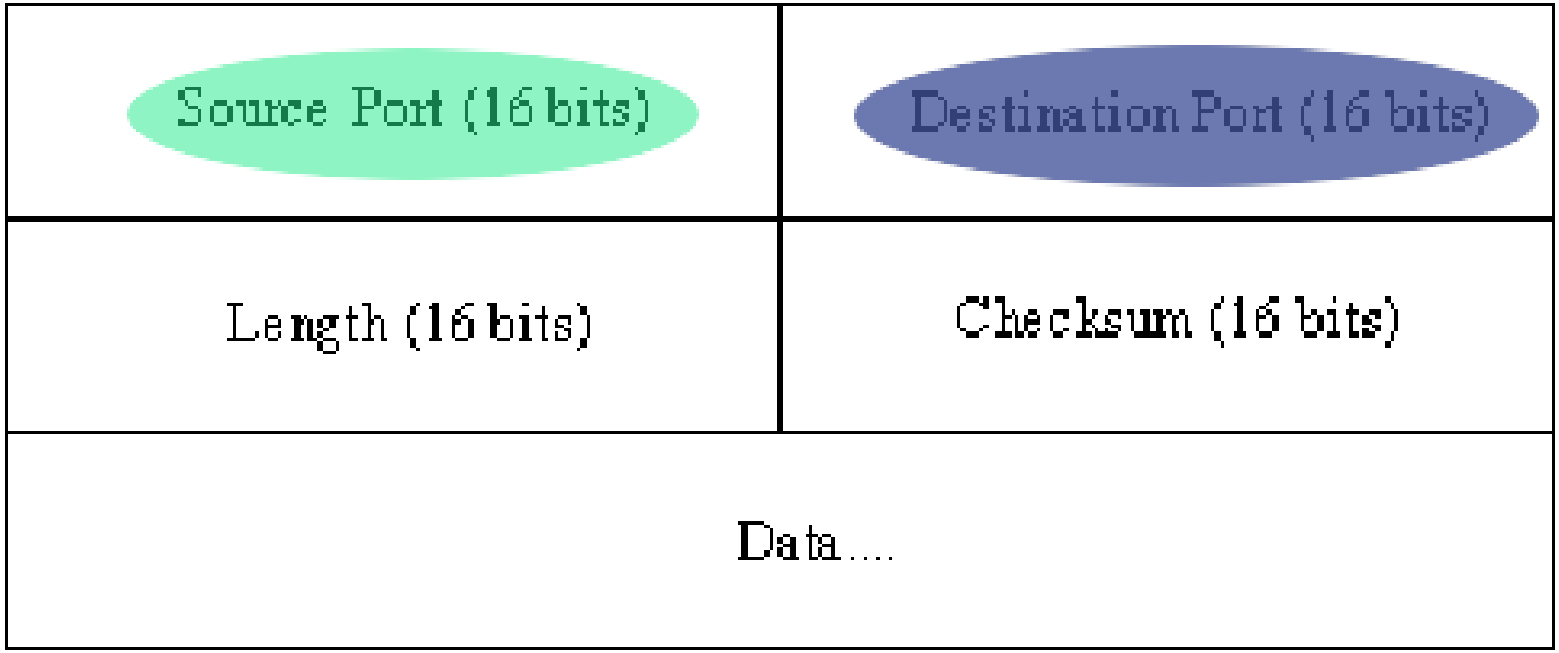
ip


0
4
8
16
19
24
31
VERS
HLEN
Service Type
Total Length
ID
FLG
Fragment Offset
TTL
Protocol
Header Checksum
Source IP Address
Destination IP Address
IP Options
Padding
Data ...
...
...

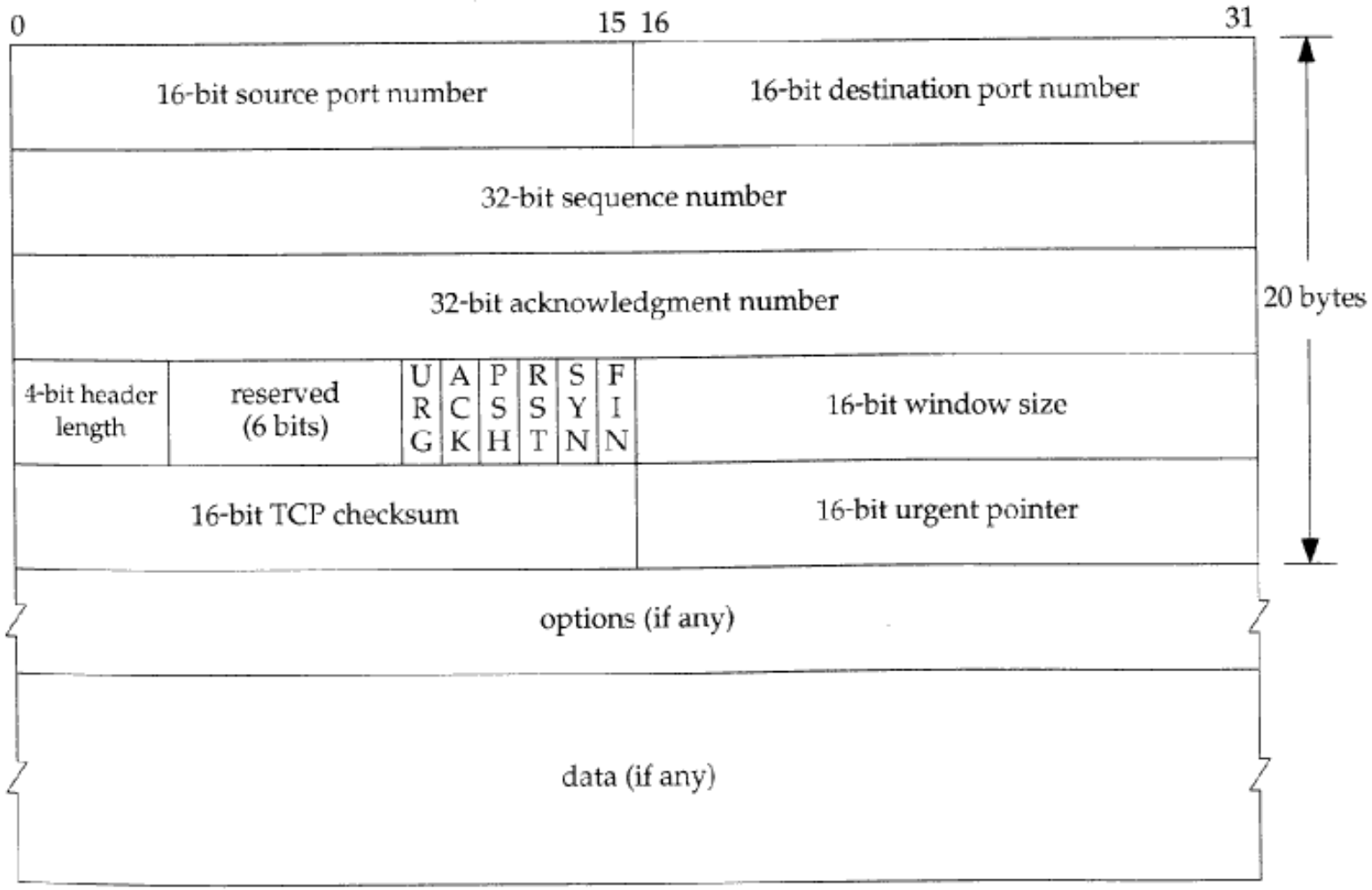
udp

Source Port (16 bits)	Destination Port (16 bits)
Length (16 bits)	Checksum (16 bits)
Data....	

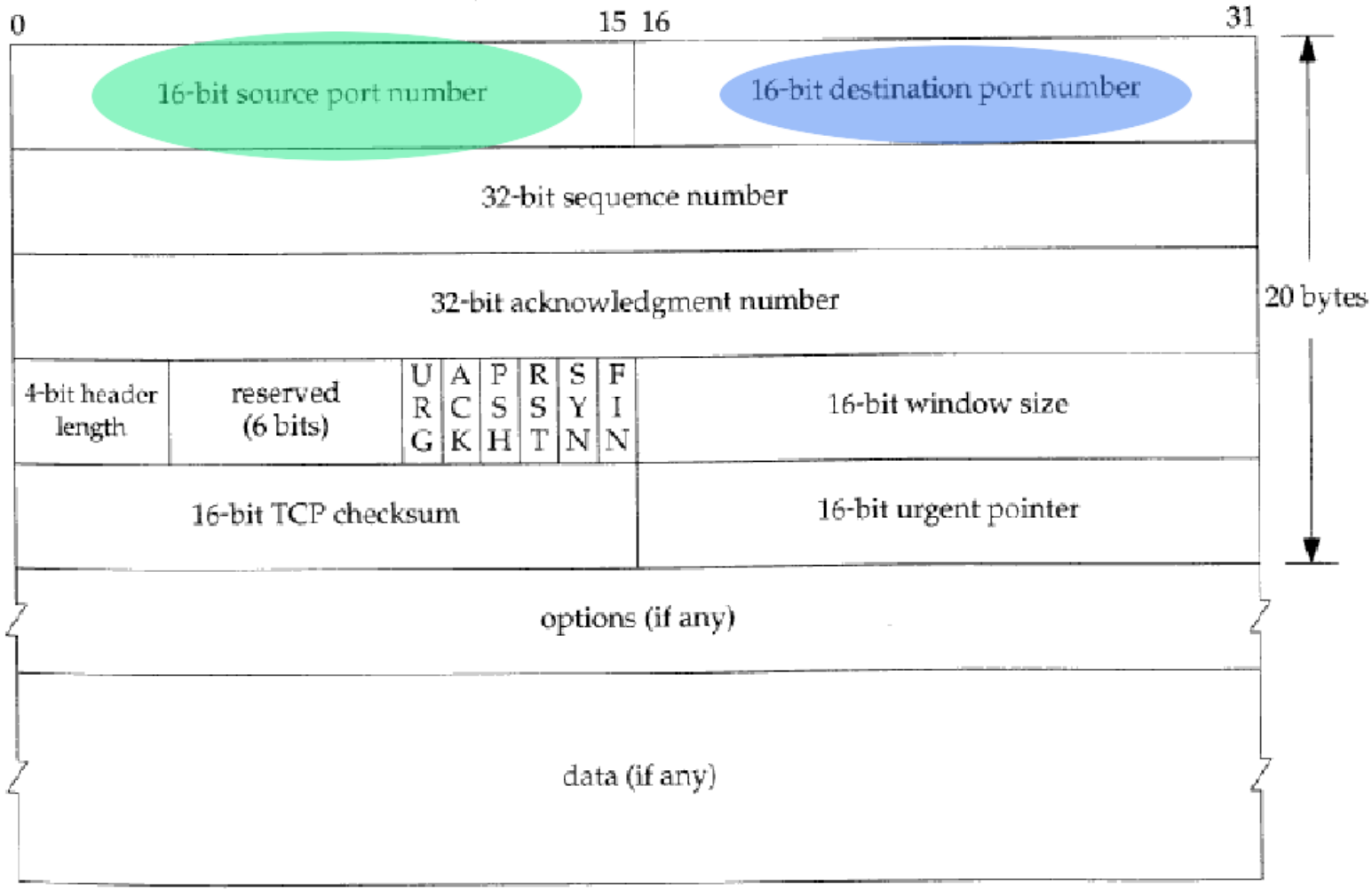
udp


Source Port (16 bits)
Destination Port (16 bits)
Length (16 bits)
Checksum (16 bits)
Data....

tcp


0
15 16
31
16-bit source port number
16-bit destination port number
32-bit sequence number
32-bit acknowledgment number
20 bytes
4-bit header length
reserved (6 bits)
URG
ACK
PSH
RST
SYN
FIN
16-bit window size
16-bit TCP checksum
16-bit urgent pointer
options (if any)
data (if any)

tcp


0
15 16
31
16-bit source port number
16-bit destination port number
32-bit sequence number
32-bit acknowledgment number
20 bytes
4-bit header length
reserved (6 bits)
URG
ACK
PSH
RST
SYN
FIN
16-bit window size
16-bit TCP checksum
16-bit urgent pointer
options (if any)
data (if any)

iptables - was ist das

iptables - was ist das

- packetfilter

iptables - was ist das

- packetfilter
- netfilter.org

iptables - was ist das

- packetfilter
- netfilter.org
- Harald Welte

iptables - basisregeln

iptables - basisregeln

Ansatz

iptables - basisregeln

Ansatz

- Alles erlauben

iptables - basisregeln

Ansatz

- Alles erlauben - Gezielt verbieten.

iptables - basisregeln

Ansatz

- Alles erlauben - Gezielt verbieten.
- Alles verbieten

iptables - basisregeln

Ansatz

- Alles erlauben - Gezielt verbieten.
- Alles verbieten - Gezielt erlauben.

iptables - basisregeln

Ansatz

- Alles erlauben - Gezielt verbieten.
- Alles verbieten - Gezielt erlauben.
- Default Policy.

iptables - basisregeln

Ansatz

- Alles erlauben - Gezielt verbieten.
- Alles verbieten - Gezielt erlauben.
- Default Policy. Alles verbieten

iptables - basisregeln

Ansatz

- Alles erlauben - Gezielt verbieten.
- Alles verbieten - Gezielt erlauben.
- Default Policy. Alles verbieten

Hilfreiches

iptables - basisregeln

Ansatz

- Alles erlauben - Gezielt verbieten.
- Alles verbieten - Gezielt erlauben.
- Default Policy. Alles verbieten

Hilfreiches

- Am Ende alles loggen.

iptables - basisregeln

Ansatz

- Alles erlauben - Gezielt verbieten.
- Alles verbieten - Gezielt erlauben.
- Default Policy. Alles verbieten

Hilfreiches

- Am Ende alles loggen.
- Am Ende alles rejecten.

iptables - syntax

iptables - syntax

- Chains

iptables - syntax

- Chains
- INPUT,OUTPUT,FORWARD

iptables - syntax

- Chains
- INPUT,OUTPUT,FORWARD

Chain handling

iptables - syntax

- Chains
- INPUT,OUTPUT,FORWARD

Chain handling

- Default Policy.

iptables - syntax

- Chains
- INPUT,OUTPUT,FORWARD

Chain handling

- Default Policy.

```
iptables -P $CHAIN $TARGET
```

iptables - syntax

- Chains
- INPUT,OUTPUT,FORWARD

Chain handling

- Default Policy.

  ```
  iptables -P $CHAIN $TARGET
  ```

- Chains flushen.

iptables - syntax

- Chains
- INPUT,OUTPUT,FORWARD

Chain handling

- Default Policy.

```
iptables -P $CHAIN $TARGET
```

- Chains flushen.

```
iptables -F $CHAIN
```

iptables - syntax

- Chains
- INPUT,OUTPUT,FORWARD

Chain handling

- Default Policy.

```
iptables -P $CHAIN $TARGET
```

- Chains flushen.

```
iptables -F $CHAIN
```

- Chains anlegen.

iptables - syntax

- Chains
- INPUT,OUTPUT,FORWARD

Chain handling

- Default Policy.

```
iptables -P $CHAIN $TARGET
```

- Chains flushen.

```
iptables -F $CHAIN
```

- Chains anlegen.

```
iptables -N $NEWCHAIN
```

Unix& Linux DAY

iptables - syntax

- Chains
- INPUT,OUTPUT,FORWARD

Chain handling

- Default Policy.

```
iptables -P $CHAIN $TARGET
```

- Chains flushen.

```
iptables -F $CHAIN
```

- Chains anlegen.

```
iptables -N $NEWCHAIN
```

- Chains löschen.

iptables - syntax

- Chains
- INPUT,OUTPUT,FORWARD

Chain handling

- Default Policy.

```
iptables -P $CHAIN $TARGET
```

- Chains flushen.

```
iptables -F $CHAIN
```

- Chains anlegen.

```
iptables -N $NEWCHAIN
```

- Chains löschen.

```
iptables -X $NEWCHAIN
```

iptables - syntax II

iptables - syntax II

- Regel hinzufügen:

iptables - syntax II

- Regel hinzufügen:

```
iptables -A $CHAIN $REGEL
```

iptables - syntax II

- Regel hinzufügen:

  ```
  iptables -A $CHAIN $REGEL
  ```

- Regel löschen:

iptables - syntax II

- Regel hinzufügen:

```
iptables -A $CHAIN $REGEL
```

- Regel löschen:

```
iptables -D $CHAIN $REGEL
```

iptables - syntax II

- Regel hinzufügen:

  ```
  iptables -A $CHAIN $REGEL
  ```

- Regel löschen:

  ```
  iptables -D $CHAIN $REGEL
  ```

Regeln

iptables - syntax II

- Regel hinzufügen:

  `iptables -A $CHAIN $REGEL`
- Regel löschen:

  `iptables -D $CHAIN $REGEL`

Regeln

- Protokoll

iptables - syntax II

- Regel hinzufügen:

  ```
  iptables -A $CHAIN $REGEL
  ```

- Regel löschen:

  ```
  iptables -D $CHAIN $REGEL
  ```

Regeln

- Protokoll: tcp,udp,icmp

iptables - syntax II

- Regel hinzufügen:

  ```
  iptables -A $CHAIN $REGEL
  ```

- Regel löschen:

  ```
  iptables -D $CHAIN $REGEL
  ```

Regeln

- Protokoll: tcp,udp,icmp,esp

iptables - syntax II

- Regel hinzufügen:

  `iptables -A $CHAIN $REGEL`
- Regel löschen:

  `iptables -D $CHAIN $REGEL`

Regeln

- Protokoll: tcp,udp,icmp,esp -p

iptables - syntax II

- Regel hinzufügen:

  ```
  iptables -A $CHAIN $REGEL
  ```

- Regel löschen:

  ```
  iptables -D $CHAIN $REGEL
  ```

Regeln

- Protokoll: tcp,udp,icmp,esp -p
- Quell-Adresse

iptables - syntax II

- Regel hinzufügen:

  ```
  iptables -A $CHAIN $REGEL
  ```

- Regel löschen:

  ```
  iptables -D $CHAIN $REGEL
  ```

Regeln

- Protokoll: tcp,udp,icmp,esp -p
- Quell-Adresse: ip-adresse,dns-name

iptables - syntax II

- Regel hinzufügen:

  ```
  iptables -A $CHAIN $REGEL
  ```

- Regel löschen:

  ```
  iptables -D $CHAIN $REGEL
  ```

Regeln

- Protokoll: tcp,udp,icmp,esp -p
- Quell-Adresse: ip-adresse,dns-name -s

iptables - syntax II

- Regel hinzufügen:

  ```
  iptables -A $CHAIN $REGEL
  ```

- Regel löschen:

  ```
  iptables -D $CHAIN $REGEL
  ```

Regeln

- Protokoll: tcp,udp,icmp,esp -p
- Quell-Adresse: ip-adresse,dns-name -s
- Ziel-Adresse

iptables - syntax II

- Regel hinzufügen:

  ```
  iptables -A $CHAIN $REGEL
  ```

- Regel löschen:

  ```
  iptables -D $CHAIN $REGEL
  ```

Regeln

- Protokoll: tcp,udp,icmp,esp -p
- Quell-Adresse: ip-adresse,dns-name -s
- Ziel-Adresse: ip-adresse,dns-name

iptables - syntax II

- Regel hinzufügen:

  ```
  iptables -A $CHAIN $REGEL
  ```

- Regel löschen:

  ```
  iptables -D $CHAIN $REGEL
  ```

Regeln

- Protokoll: tcp,udp,icmp,esp -p
- Quell-Adresse: ip-adresse,dns-name -s
- Ziel-Adresse: ip-adresse,dns-name -d

iptables - syntax II

- Regel hinzufügen:

  ```
  iptables -A $CHAIN $REGEL
  ```

- Regel löschen:

  ```
  iptables -D $CHAIN $REGEL
  ```

Regeln

- Protokoll: tcp,udp,icmp,esp -p
- Quell-Adresse: ip-adresse,dns-name -s
- Ziel-Adresse: ip-adresse,dns-name -d
- Quell-Port

Unix& Linux DAY

iptables - syntax II

- Regel hinzufügen:

  ```
  iptables -A $CHAIN $REGEL
  ```

- Regel löschen:

  ```
  iptables -D $CHAIN $REGEL
  ```

Regeln

- Protokoll: tcp,udp,icmp,esp -p
- Quell-Adresse: ip-adresse,dns-name -s
- Ziel-Adresse: ip-adresse,dns-name -d
- Quell-Port: port-Nummer, port-name

Unix& Linux DAY

iptables - syntax II

- Regel hinzufügen:

  ```
  iptables -A $CHAIN $REGEL
  ```

- Regel löschen:

  ```
  iptables -D $CHAIN $REGEL
  ```

Regeln

- Protokoll: tcp,udp,icmp,esp -p
- Quell-Adresse: ip-adresse,dns-name -s
- Ziel-Adresse: ip-adresse,dns-name -d
- Quell-Port: port-Nummer, port-name –sport

Unix& Linux DAY

iptables - syntax II

- Regel hinzufügen:

  ```
  iptables -A $CHAIN $REGEL
  ```

- Regel löschen:

  ```
  iptables -D $CHAIN $REGEL
  ```

Regeln

- Protokoll: tcp,udp,icmp,esp -p
- Quell-Adresse: ip-adresse,dns-name -s
- Ziel-Adresse: ip-adresse,dns-name -d
- Quell-Port: port-Nummer, port-name –sport
- Ziel-Adresse

Unix& Linux DAY

iptables - syntax II

- Regel hinzufügen:

  ```
  iptables -A $CHAIN $REGEL
  ```

- Regel löschen:

  ```
  iptables -D $CHAIN $REGEL
  ```

Regeln

- Protokoll: tcp,udp,icmp,esp -p
- Quell-Adresse: ip-adresse,dns-name -s
- Ziel-Adresse: ip-adresse,dns-name -d
- Quell-Port: port-Nummer, port-name –sport
- Ziel-Adresse: port-nummer, port-name

Unix& Linux DAY

iptables - syntax II

- Regel hinzufügen:

  ```
  iptables -A $CHAIN $REGEL
  ```

- Regel löschen:

  ```
  iptables -D $CHAIN $REGEL
  ```

Regeln

- Protokoll: tcp,udp,icmp,esp -p
- Quell-Adresse: ip-adresse,dns-name -s
- Ziel-Adresse: ip-adresse,dns-name -d
- Quell-Port: port-Nummer, port-name –sport
- Ziel-Adresse: port-nummer, port-name –dport

Unix& Linux DAY

iptables - syntax II

- Regel hinzufügen:

  ```
  iptables -A $CHAIN $REGEL
  ```

- Regel löschen:

  ```
  iptables -D $CHAIN $REGEL
  ```

Regeln

- Protokoll: tcp,udp,icmp,esp -p
- Quell-Adresse: ip-adresse,dns-name -s
- Ziel-Adresse: ip-adresse,dns-name -d
- Quell-Port: port-Nummer, port-name –sport
- Ziel-Adresse: port-nummer, port-name –dport
- Ziel

iptables - syntax II

- Regel hinzufügen:

  ```
  iptables -A $CHAIN $REGEL
  ```

- Regel löschen:

  ```
  iptables -D $CHAIN $REGEL
  ```

Regeln

- Protokoll: tcp,udp,icmp,esp -p
- Quell-Adresse: ip-adresse,dns-name -s
- Ziel-Adresse: ip-adresse,dns-name -d
- Quell-Port: port-Nummer, port-name –sport
- Ziel-Adresse: port-nummer, port-name –dport
- Ziel: ACCEPT,REJECT,DROP,LOG,$NEWCHAIN

Unix& Linux DAY

iptables - syntax II

- Regel hinzufügen:

```
iptables -A $CHAIN $REGEL
```

- Regel löschen:

```
iptables -D $CHAIN $REGEL
```

Regeln

- Protokoll: tcp,udp,icmp,esp -p
- Quell-Adresse: ip-adresse,dns-name -s
- Ziel-Adresse: ip-adresse,dns-name -d
- Quell-Port: port-Nummer, port-name –sport
- Ziel-Adresse: port-nummer, port-name –dport
- Ziel: ACCEPT,REJECT,DROP,LOG,$NEWCHAIN -j

Einsatz

Einsatz

host based

Einsatz

host based

- einfachster Fall

Einsatz

host based

- einfachster Fall
- keine Dienste

Einsatz

host based

- einfachster Fall
- keine Dienste

router

Einsatz

host based

- einfachster Fall
- keine Dienste

router

- Dienste?

Einsatz

host based

- einfachster Fall
- keine Dienste

router

- Dienste?
- Weiterleitung von Verbindungen

Einsatz

host based

- einfachster Fall
- keine Dienste

router

- Dienste?
- Weiterleitung von Verbindungen
- NAT/Masquerading.

Einsatz

host based

- einfachster Fall
- keine Dienste

router

- Dienste?
- Weiterleitung von Verbindungen
- NAT/Masquerading.

server

Einsatz

host based

- einfachster Fall
- keine Dienste

router

- Dienste?
- Weiterleitung von Verbindungen
- NAT/Masquerading.

server

- Dienste

Einsatz

host based

- einfachster Fall
- keine Dienste

router

- Dienste?
- Weiterleitung von Verbindungen
- NAT/Masquerading.

server

- Dienste
- Verbindungen von und nach aussen

Host based I

Eingehende Verbindungen

Host based I

Eingehende Verbindungen

- Nur Antworten auf eigene Anfragen.

Host based I

Eingehende Verbindungen

- Nur Antworten auf eigene Anfragen.
- keine Dienste

Host based I

Eingehende Verbindungen

- Nur Antworten auf eigene Anfragen.
- keine Dienste

Alternative 1

- Jeden Port und Peer freigeben.

Host based I

Eingehende Verbindungen

- Nur Antworten auf eigene Anfragen.
- keine Dienste

Alternative 1

- Jeden Port und Peer freigeben.
- `iptables -A INPUT -p udp -s $DNS1 --sport 53 --dport 1024: -j ACCEPT`

Host based I

Eingehende Verbindungen

- Nur Antworten auf eigene Anfragen.
- keine Dienste

Alternative 1

- Jeden Port und Peer freigeben.
- `iptables -A INPUT -p udp -s $DNS1 --sport 53 --dport 1024: -j ACCEPT`
- Antworten?

Host based I

Eingehende Verbindungen

- Nur Antworten auf eigene Anfragen.
- keine Dienste

Alternative 1

- Jeden Port und Peer freigeben.
- `iptables -A INPUT -p udp -s $DNS1 --sport 53 --dport 1024: -j ACCEPT`
- Antworten?

```
iptables -A INPUT -p tcp ! --syn --dport 1024: -j ACCEPT
```

Unix& Linux DAY

Host based II

Host based II

Alternative 2

Host based II

Alternative 2

- Stateful filtering.

Host based II

Alternative 2

- Stateful filtering.
- `iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT`

Host based III

Host based III

ausgehende Verbindungen

Host based III

ausgehende Verbindungen

- Anfragen von uns

Host based III

ausgehende Verbindungen

- Anfragen von uns
- Genauer: Anfragen von unserem Rechner

Host based III

ausgehende Verbindungen

- Anfragen von uns
- Genauer: Anfragen von unserem Rechner: Trojanische Pferde

Host based III

ausgehende Verbindungen

- Anfragen von uns
- Genauer: Anfragen von unserem Rechner: Trojanische Pferde

Regeln

Host based III

ausgehende Verbindungen

- Anfragen von uns
- Genauer: Anfragen von unserem Rechner: Trojanische Pferde

Regeln

- `iptables -A OUTPUT -p tcp -d $PROXY --dport 80 -j ACCEPT`

Host based III

ausgehende Verbindungen

- Anfragen von uns
- Genauer: Anfragen von unserem Rechner: Trojanische Pferde

Regeln

- `iptables -A OUTPUT -p tcp -d $PROXY --dport 80 -j ACCEPT`
- `iptables -A OUTPUT -p tcp --dport 22 -j ACCEPT`

Unix& Linux DAY

router

router

Ein- und ausgehende Verbindungen

router

Ein- und ausgehende Verbindungen

- Fast wie bei host based.

router

Ein- und ausgehende Verbindungen

- Fast wie bei host based.
- Anfragen von den Rechnern im inneren Netz annehmen.

router

Ein- und ausgehende Verbindungen

- Fast wie bei host based.
- Anfragen von den Rechnern im inneren Netz annehmen.

```
iptables -A INPUT -s $LOCALNET -j ACCEPT
```

router

Ein- und ausgehende Verbindungen

- Fast wie bei host based.
- Anfragen von den Rechnern im inneren Netz annehmen.

```
iptables -A INPUT -s $LOCALNET -j ACCEPT
```

- Antworten zulassen.

router

Ein- und ausgehende Verbindungen

- Fast wie bei host based.
- Anfragen von den Rechnern im inneren Netz annehmen.

```
iptables -A INPUT -s $LOCALNET -j ACCEPT
```

- Antworten zulassen.

```
iptables -A OUTPUT -m --state ESTABLISHED,RELATED -j ACCEPT
```

router

Ein- und ausgehende Verbindungen

- Fast wie bei host based.
- Anfragen von den Rechnern im inneren Netz annehmen.

```
iptables -A INPUT -s $LOCALNET -j ACCEPT
```

- Antworten zulassen.

```
iptables -A OUTPUT -m --state ESTABLISHED,RELATED -j ACCEPT
```

- IP-Forwarding.

router

Ein- und ausgehende Verbindungen

- Fast wie bei host based.
- Anfragen von den Rechnern im inneren Netz annehmen.

```
iptables -A INPUT -s $LOCALNET -j ACCEPT
```

- Antworten zulassen.

```
iptables -A OUTPUT -m --state ESTABLISHED,RELATED -j ACCEPT
```

- IP-Forwarding.

```
echo "1ÿ /proc/sys/net/ipv4/ip_forward
```

router

Ein- und ausgehende Verbindungen

- Fast wie bei host based.
- Anfragen von den Rechnern im inneren Netz annehmen.

```
iptables -A INPUT -s $LOCALNET -j ACCEPT
```

- Antworten zulassen.

```
iptables -A OUTPUT -m --state ESTABLISHED,RELATED -j ACCEPT
```

- IP-Forwarding.

```
echo "1" > /proc/sys/net/ipv4/ip_forward
```

NAT

router

Ein- und ausgehende Verbindungen

- Fast wie bei host based.
- Anfragen von den Rechnern im inneren Netz annehmen.

```
iptables -A INPUT -s $LOCALNET -j ACCEPT
```

- Antworten zulassen.

```
iptables -A OUTPUT -m --state ESTABLISHED,RELATED -j ACCEPT
```

- IP-Forwarding.

```
echo "1> /proc/sys/net/ipv4/ip_forward
```

NAT

- `iptables -A POSTROUTING -t nat -s $LAN -j MASQUERADE`

router

Ein- und ausgehende Verbindungen

- Fast wie bei host based.
- Anfragen von den Rechnern im inneren Netz annehmen.

```
iptables -A INPUT -s $LOCALNET -j ACCEPT
```

- Antworten zulassen.

```
iptables -A OUTPUT -m --state ESTABLISHED,RELATED -j ACCEPT
```

- IP-Forwarding.

```
echo "1" > /proc/sys/net/ipv4/ip_forward
```

NAT

- `iptables -A POSTROUTING -t nat -s $LAN -j MASQUERADE`
- `iptables -A PREROUTING -t nat -p tcp --dport 80 -j DNAT --to-destination $LOCALSERVER:80`

server

server

Ein- und ausgehende Verbindungen

server

Ein- und ausgehende Verbindungen

- Fast wie bei router.

server

Ein- und ausgehende Verbindungen

- Fast wie bei router.
- Nur kein Forwarding.

server

Ein- und ausgehende Verbindungen

- Fast wie bei router.
- Nur kein Forwarding.
- Anfragen für die Dienste zulassen.

server

Ein- und ausgehende Verbindungen

- Fast wie bei router.
- Nur kein Forwarding.
- Anfragen für die Dienste zulassen.

```
iptables -A INPUT -p tcp --dport $DIENSTPORT -j ACCEPT
```

Antworten

server

Ein- und ausgehende Verbindungen

- Fast wie bei router.
- Nur kein Forwarding.
- Anfragen für die Dienste zulassen.

```
iptables -A INPUT -p tcp --dport $DIENSTPORT -j ACCEPT
```

Antworten

- Antworten zulassen.

server

Ein- und ausgehende Verbindungen

- Fast wie bei router.
- Nur kein Forwarding.
- Anfragen für die Dienste zulassen.

```
iptables -A INPUT -p tcp --dport $DIENSTPORT -j ACCEPT
```

Antworten

- Antworten zulassen.

```
iptables -A OUTPUT -m --state ESTABLISHED,RELATED -j ACCEPT
```

Unix& Linux DAY

komplexere protokolle

komplexere protokolle

Conntrack

komplexere protokolle

Conntrack

- Connection tracking

komplexere protokolle

Conntrack

- Connection tracking
- `proc/net/ip_conntrack`

komplexere protokolle

Conntrack

- Connection tracking
- `proc/net/ip_conntrack`

FTP

komplexere protokolle

Conntrack

- Connection tracking
- `proc/net/ip_conntrack`

FTP

- `modprobe ip_conntrack_ftp`

komplexere protokolle

Conntrack

- Connection tracking
- `proc/net/ip_conntrack`

FTP

- `modprobe ip_conntrack_ftp`
- `iptables -A OUTPUT -p tcp --dport 20 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT`

Unix& Linux DAY

komplexere protokolle

Conntrack

- Connection tracking
- `proc/net/ip_conntrack`

FTP

- `modprobe ip_conntrack_ftp`
- `iptables -A OUTPUT -p tcp --dport 20 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT`
- `iptables -A OUTPUT -p tcp --dport 21 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT`

Unix& Linux DAY

komplexere protokolle II

komplexere protokolle II

Erweiterungen

komplexere protokolle II

Erweiterungen

- http://www.netfilter.org/documentation/HOWTO/de/netfilter-extensions-HOWTO.html

komplexere protokolle II

Erweiterungen

- http://www.netfilter.org/documentation/HOWTO/de/netfilter-extensions-HOWTO.html
- http://www.netfilter.org/patch-o-matic/pom-extra.html

komplexere protokolle II

Erweiterungen

- http://www.netfilter.org/documentation/HOWTO/de/netfilter-extensions-HOWTO.html
- http://www.netfilter.org/patch-o-matic/pom-extra.html
- conntrack

komplexere protokolle II

Erweiterungen

- http://www.netfilter.org/documentation/HOWTO/de/netfilter-extensions-HOWTO.html
- http://www.netfilter.org/patch-o-matic/pom-extra.html
- conntrack allgemeine Regeln

komplexere protokolle II

Erweiterungen

- http://www.netfilter.org/documentation/HOWTO/de/netfilter-extensions-HOWTO.html
- http://www.netfilter.org/patch-o-matic/pom-extra.html
- conntrack allgemeine Regeln
- mport

komplexere protokolle II

Erweiterungen

- http://www.netfilter.org/documentation/HOWTO/de/netfilter-extensions-HOWTO.html
- http://www.netfilter.org/patch-o-matic/pom-extra.html
- conntrack allgemeine Regeln
- mport

```
iptables -A INPUT -p tcp -m mport --ports 20:23,80 -j DROP
```

komplexere protokolle II

Erweiterungen

- http://www.netfilter.org/documentation/HOWTO/de/netfilter-extensions-HOWTO.html
- http://www.netfilter.org/patch-o-matic/pom-extra.html
- conntrack allgemeine Regeln
- mport

```
iptables -A INPUT -p tcp -m mport --ports 20:23,80 -j DROP
```

- string

komplexere protokolle II

Erweiterungen

- http://www.netfilter.org/documentation/HOWTO/de/netfilter-extensions-HOWTO.html
- http://www.netfilter.org/patch-o-matic/pom-extra.html
- conntrack allgemeine Regeln
- mport

```
iptables -A INPUT -p tcp -m mport --ports 20:23,80 -j DROP
```

- string

```
iptables -A INPUT -m string --string 'cmd.exe' -j $QUEUE
```

komplexere protokolle II

Erweiterungen

- http://www.netfilter.org/documentation/HOWTO/de/netfilter-extensions-HOWTO.html
- http://www.netfilter.org/patch-o-matic/pom-extra.html
- conntrack allgemeine Regeln
- mport

```
iptables -A INPUT -p tcp -m mport --ports 20:23,80 -j DROP
```

- string

```
iptables -A INPUT -m string --string 'cmd.exe' -j $QUEUE
```

- rpc

komplexere protokolle II

Erweiterungen

- http://www.netfilter.org/documentation/HOWTO/de/netfilter-extensions-HOWTO.html
- http://www.netfilter.org/patch-o-matic/pom-extra.html
- conntrack allgemeine Regeln
- mport

```
iptables -A INPUT -p tcp -m mport --ports 20:23,80 -j DROP
```

- string

```
iptables -A INPUT -m string --string 'cmd.exe' -j $QUEUE
```

- rpc

```
-m rpc
```

fallen

fallen

- Droppen von icmp.

fallen

- Droppen von icmp.
- Verbindungen über localinterface (lo) verbieten.

fallen

- Droppen von icmp.
- Verbindungen über localinterface (lo) verbieten.
- SSH verbieten während man dran arbeitet.

fallen

- Droppen von icmp.
- Verbindungen über localinterface (lo) verbieten.
- SSH verbieten während man dran arbeitet.
- Chains flushen mit Default-Policy auf DROP.

fallen

- Droppen von icmp.
- Verbindungen über localinterface (lo) verbieten.
- SSH verbieten während man dran arbeitet.
- Chains flushen mit Default-Policy auf DROP.
- Beim flushen der Chains die NAT-table vergessen.

fallen

- Droppen von icmp.
- Verbindungen über localinterface (lo) verbieten.
- SSH verbieten während man dran arbeitet.
- Chains flushen mit Default-Policy auf DROP.
- Beim flushen der Chains die NAT-table vergessen.

weitere fallen

fallen

- Droppen von icmp.
- Verbindungen über localinterface (lo) verbieten.
- SSH verbieten während man dran arbeitet.
- Chains flushen mit Default-Policy auf DROP.
- Beim flushen der Chains die NAT-table vergessen.

weitere fallen

- auch iptables ist nur Software

fallen

- Droppen von icmp.
- Verbindungen über localinterface (lo) verbieten.
- SSH verbieten während man dran arbeitet.
- Chains flushen mit Default-Policy auf DROP.
- Beim flushen der Chains die NAT-table vergessen.

weitere fallen

- auch iptables ist nur Software $\Rightarrow$ fehleranfällig

fallen

- Droppen von icmp.
- Verbindungen über localinterface (lo) verbieten.
- SSH verbieten während man dran arbeitet.
- Chains flushen mit Default-Policy auf DROP.
- Beim flushen der Chains die NAT-table vergessen.

weitere fallen

- auch iptables ist nur Software $\Rightarrow$ fehleranfällig
- protokollkenntnisse

Abschluss

Vielen Dank für Ihre Aufmerksamkeit.

Fragen?

Abschluss

Vielen Dank für Ihre Aufmerksamkeit.

Fragen?

Kontakt:

Andreas Leibrock
unixag@leibi.net
http://www.leibi.net/

Unix& Linux DAY