

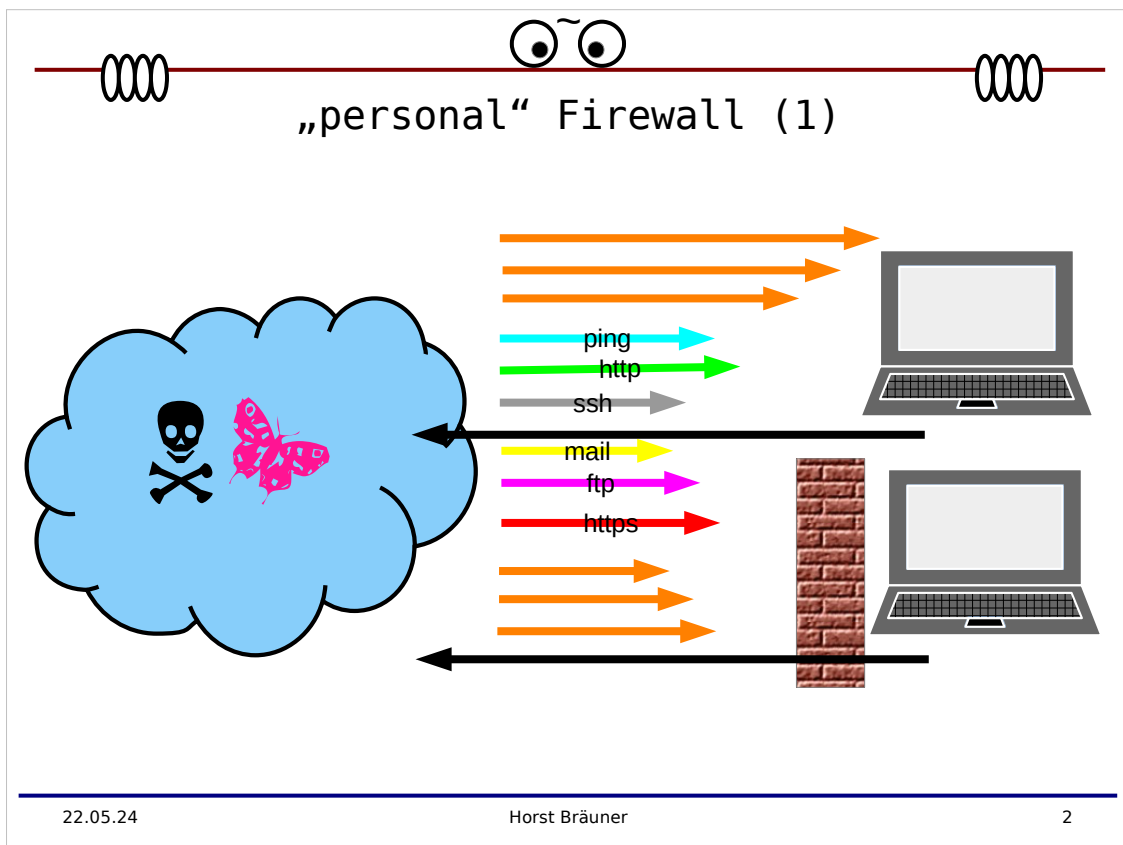
Praktische Datenverarbeitung

- Firewall
 - Beispiel personal Firewall
 - Beispiel mehrstufige Firewall / DMZ
 - http / https
 - mail
- iptables-Beispiele
 - keine Regeln
 - icmp ssh http https
 - icmp/ssh/http/https/cifs
- fail2ban
 - Beispiel
- nmap - Portscanner
- wireshark - Protokollanalyse

„Security“

Aufwand etwa 3-4 UE

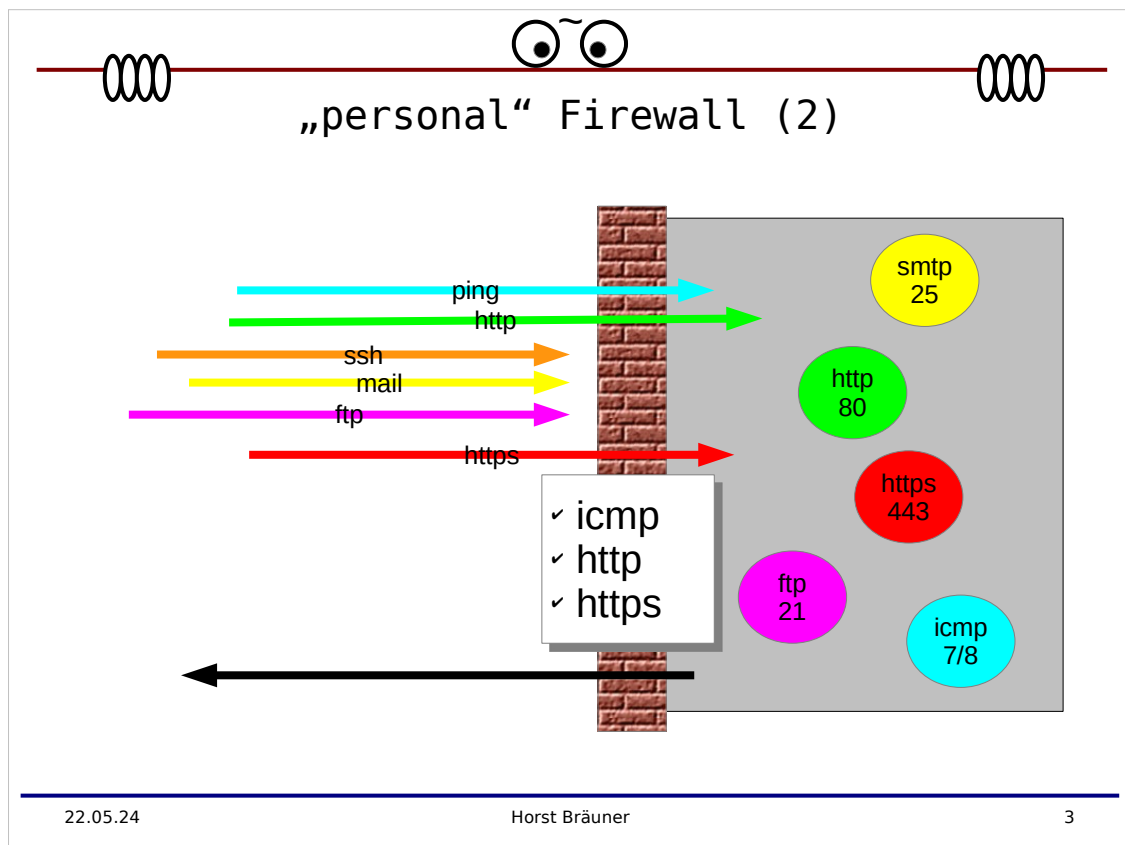
Ein wenig zum Thema Sicherheit und Sicherheitskonzepte. Die Vorlesung soll nur einen ersten Eindruck vermitteln und ein paar gängige Begriffe beleuchten, die Ihnen begegnen werden, sofern Sie in der Praxis später einmal Netzwerke administrieren.



Eine Firewall („Brandmauer“) soll einen Rechner oder ein Netzwerk vor ungewollten Zugriffen schützen.

Grundsätzlich wollen wir Dienste im Internet nutzen. Gleichzeitig versuchen möglicherweise andere uns zu „beeinflussen“. Das kann eine bloße Kontrolle sein, wie/womit wir uns im Internet bewegen. Das kann jedoch auch in der Absicht sein, mit der uns Geschäfte zu machen oder auch uns zu schaden.

Eine „personal“ Firewall dient dem Schutz unserer Person auf unserem Desktop-Rechner vor ungewollten Kontakten.



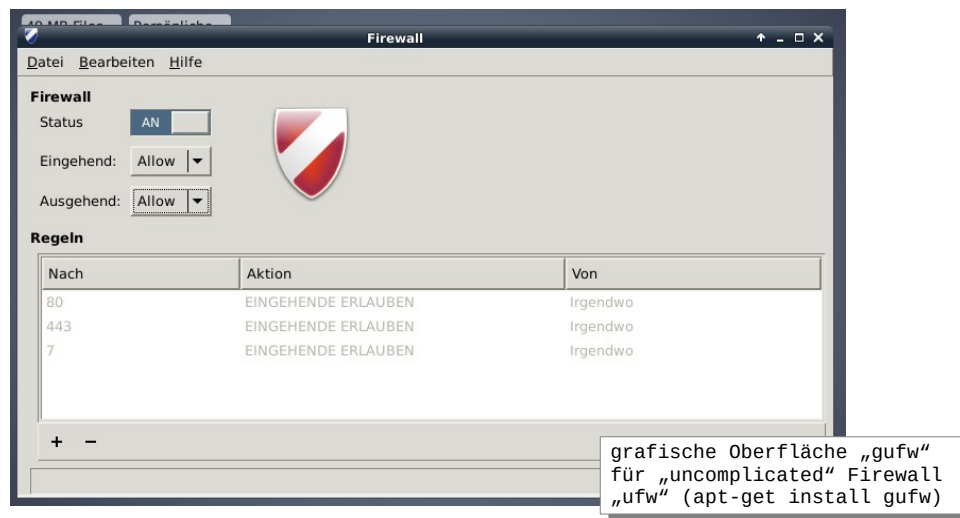
Eine personal Firewall („Brandmauer“) soll einen Rechner oder ein daran angeschlossenes Netzwerk vor ungewollten Zugriffen schützen.

Das Prinzip von „personal“ Firewalls ist meist einfach:

Außen „Feind“ innen „Freund“, das heißt, ich darf nach draußen alle Dienste und Services in Anspruch nehmen. Zum Beispiel Surfen, E-Mail senden, Dateien in eine Cloud kopieren usw. aber von Außen darf niemand auf meinen Rechner zugreifen, es sei denn, ich erlaube es explizit.

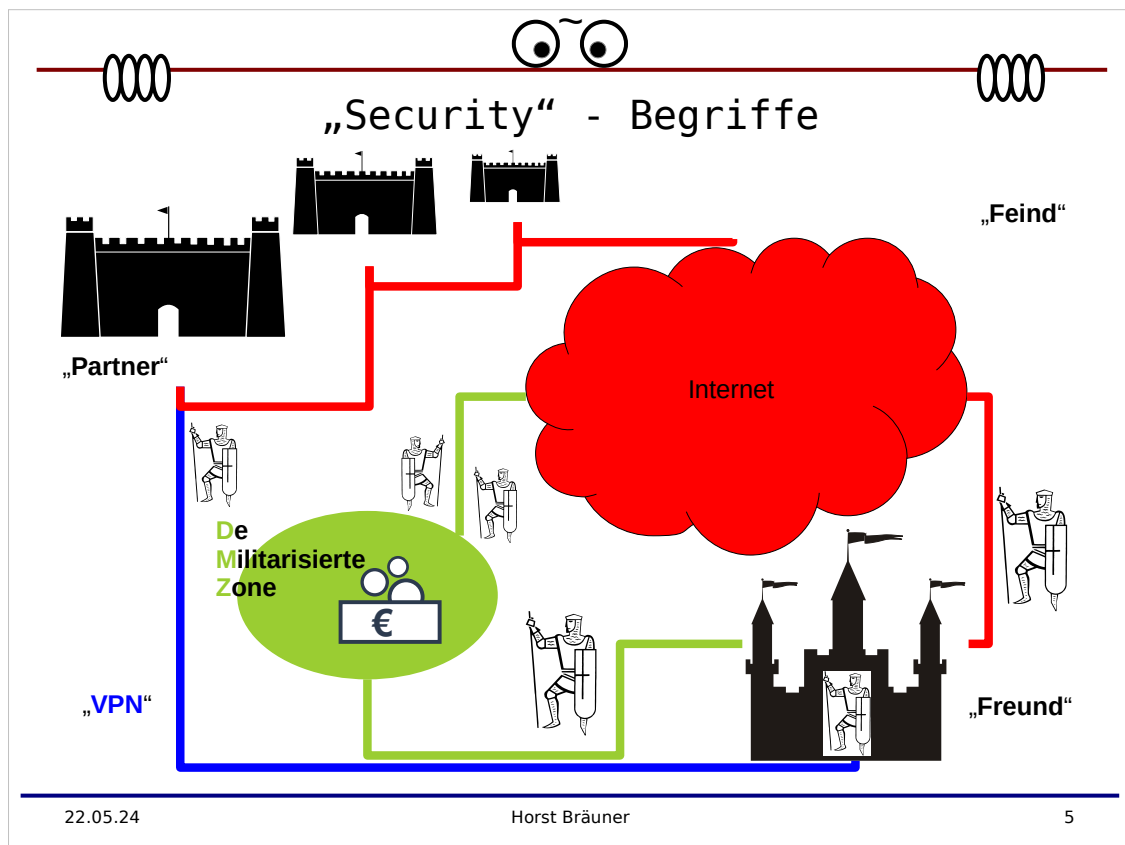
Im obigen Beispiel ist der Schutz so eingestellt, dass von Außen auf meinen Webserver auf meinem Rechner per http zugegriffen werden kann und mein Rechner ange“ping“t werden darf. Alles andere ist verboten. Ich darf ebenfalls nur per http/https im Internet surfen und andere an“ping“en.

„personal“ Firewall (3)



Eine personal Firewall unter Linux kann zum Beispiel mit „gufw“ konfiguriert werden. Sie sehen oben, dass die Firewall „An“ ist und Zugriffe von „irgendwo“ per http, https und ping auf meinen Rechner erlaubt.

VORSICHT: ufw, als Basis der grafischen Oberfläche gufw, wird im **Verzeichnis /etc/ufw/** konfiguriert. Wenn Regeln dort über die Konfigurationsdateien geändert werden, sind diese nicht immer in der Grafik sichtbar. Zudem arbeitet die Grafik mit „Templates“/„Profilen“. Diese setzen die Regeln, die über die Konfigurationsdateien eingestellt wurden außer Kraft.



Für ganze Netzwerke ist der Schutz aufwändiger. Eine Firewall („Brandmauer“) muss eine Vielzahl an Rechnern und/oder ein Netzwerk vor ungewollten Zugriffen schützen. Gleichzeitig möchte jedoch eine Vielzahl an Personen/Rechnern Dienste im Internet nutzen.

Zum Bild: Grundsätzlich soll mir kein „Feind“ schaden...

... jedoch sollen Dienste außerhalb meines Bereichs (meiner „Burg“) genutzt werden.

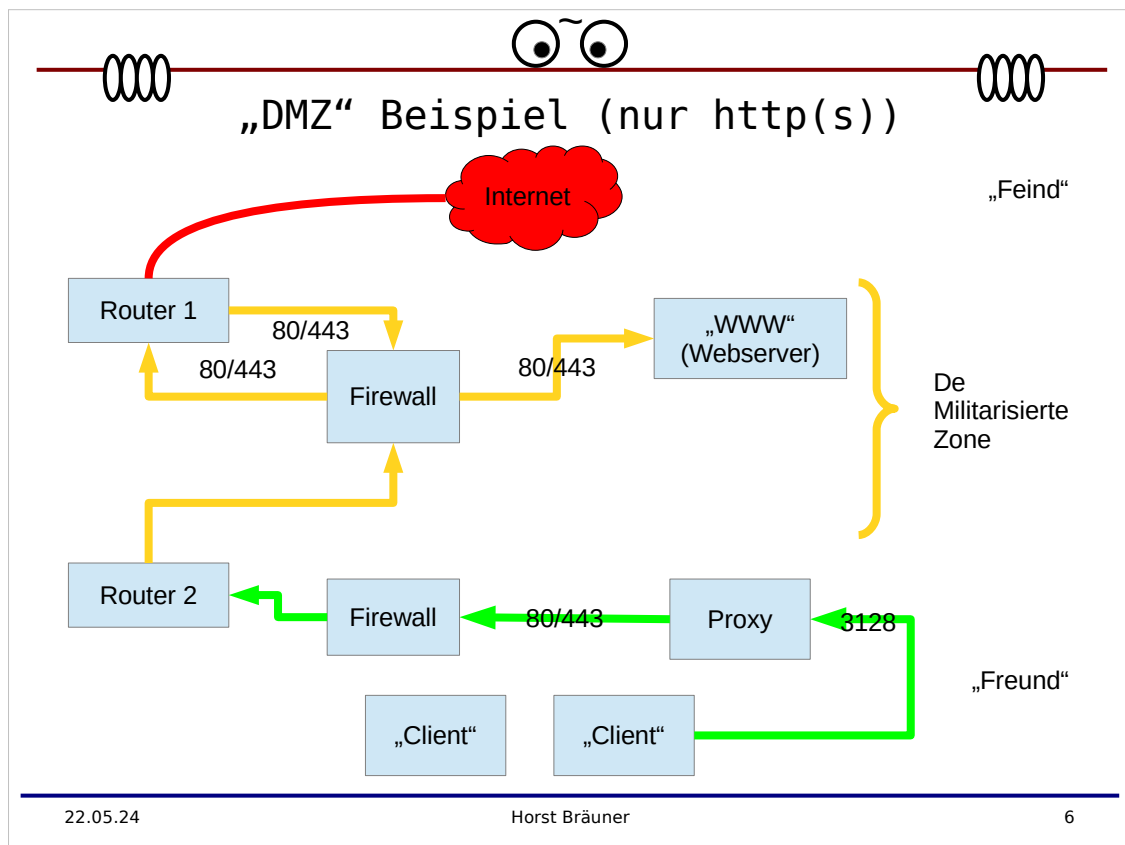
Eine Marktplatz mit Umgangsregeln und „Waffenverbot“ ist dafür **eine** Möglichkeit. Dinge / Informationen können hier für beide Seiten gefahrlos ausgetauscht werden.

Waffenverbot heißt, es wird eine **De Militarisierte Zone** eingerichtet.

Es gibt Handelspartner, mit denen eine „intimere“ Beziehung / ein Vertrauensverhältnis besteht. Diese Beziehungen gehen über den bloßen Austausch hinaus. Die Grenzen sollen nahezu aufgehoben werden und beispielsweise Ressourcen gemeinsam genutzt werden.

Es wird ein deshalb ein **Virtuelles Privates Netzwerk** eingerichtet.

An jedem Übergang zu einem anderen Bereich werden „Wächter“ platziert.



Im Gegensatz zu personal Firewalls, die grundsätzlich nur einen Rechner schützen, ist die Herausforderung in der Praxis hingegen, ganze Netzwerke zu schützen und dennoch den Austausch von Informationen mit „Feinden“ zu ermöglichen.

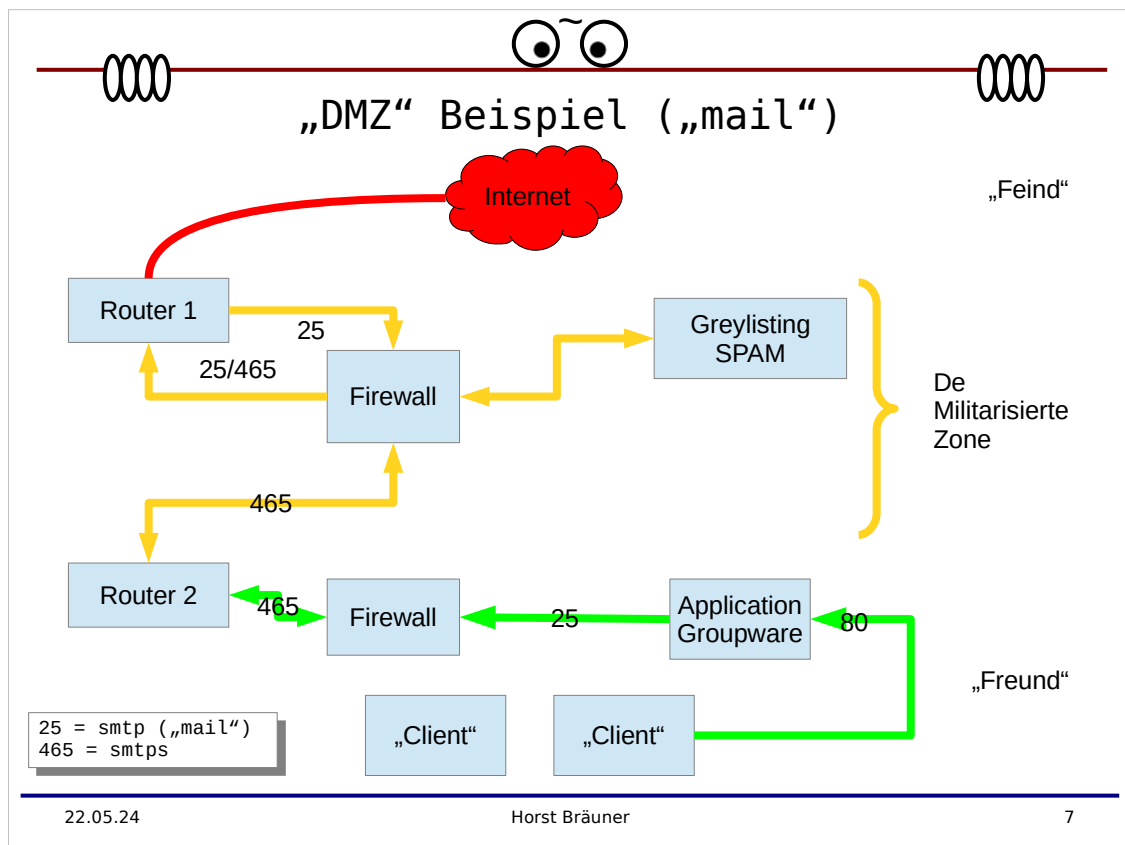
Ein Beispiel für den Schutz eines Netzwerks durch eine mehrstufige Firewall sehen Sie abgebildet.

Zwischen Freund und Feind ist eine demilitarisierte Zone eingerichtet über die die Kommunikation stattfindet. Es gibt keine direkten Zugriff ins und vom Internet.

Der interne „Client“ darf im Internet surfen, aber nur über den „Proxy“, den er per Port 3128 (hier: „squid“) kontaktieren darf. Der Proxy darf per http und https über den inneren Router 2 auf die Firewall zugreifen, die den Verkehr von http/https über den äußeren Router 1 ins Internet erlaubt und zusätzlich den Verkehr **in** die DMZ erlaubt.

Von **außen** darf ein „Kunde“ per http/https auf die Firewall zugreifen, die wiederum diesen Verkehr auf den Webserver in der DMZ erlaubt.

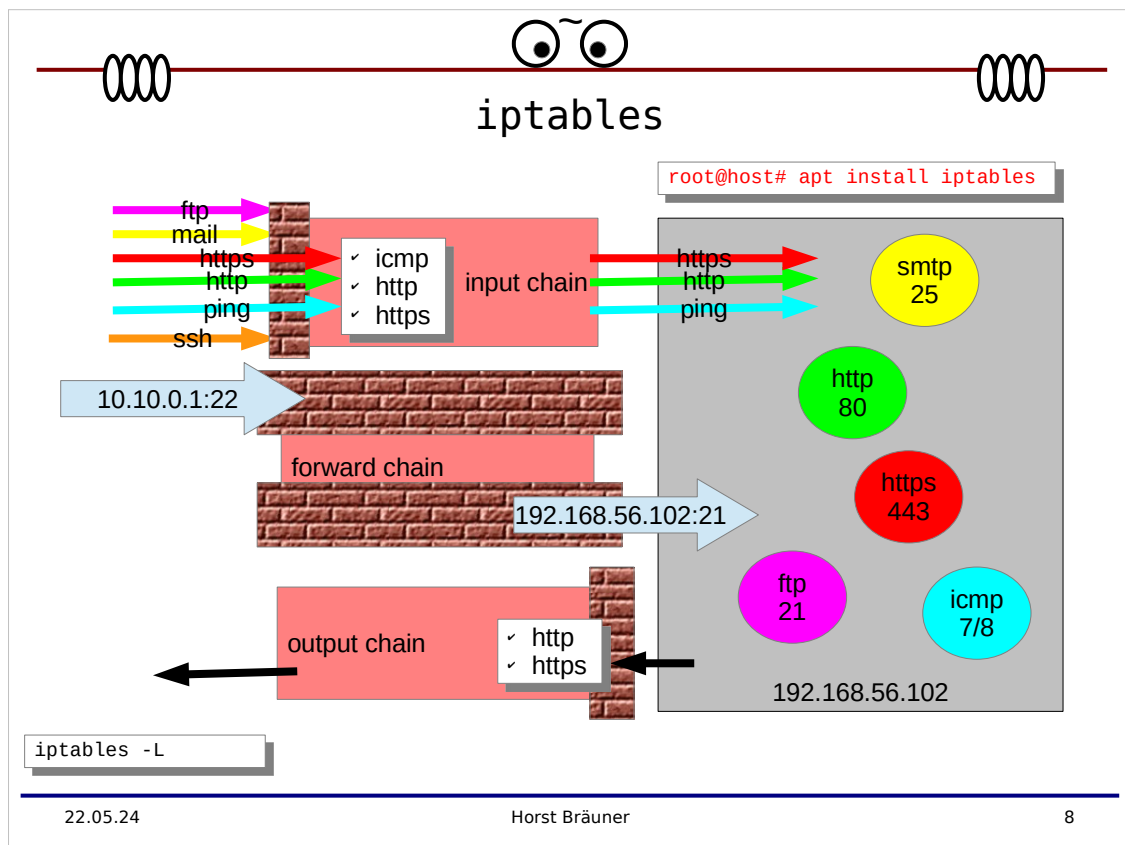
Der direkte Kontakt zwischen „Freund“ und „Feind“ ist damit unterbunden, jedoch können Daten über die DMZ ausgetauscht werden.



Dasselbe Konzept für den E-Mail Verkehr.

Der Client kann über seine Application Groupware (hier: ein Webmailer) per smtp (unverschlüsselt) Mail versenden. Die innere Firewall schickt die Mail mit Transportverschlüsselung (smtps, 465) über den Router 2 zur äußeren Firewall. Die wiederum kann, je nach Empfänger-Server, per smtp oder smtps die Mail weiter leiten. Unterstützt der empfangende, externe MTA Verschlüsselung, erfolgt der Transport per smtps.

Von außen kann Mail per smtp/smtps empfangen werden. Smtps wird in diesem Fall dann eingeschaltet, wenn der externe Sender Verschlüsselung unterstützt. Die Mail wird dann über eine Sicherheitslösung in der DMZ beispielsweise auf SPAM überprüft und im negativen Fall über die innere Firewall wieder dem MDA auf der Application Groupware zugestellt.



Die nachfolgend beschriebene Firewall „iptables“ kennt für solche Regeln 3 sogenannte „chains“ („Ketten“).

Die input chain behandelt Zugriffe **von** außen.




Die output chain behandelt Zugriffe **nach** außen.

Die forward chain modifiziert Zugriffe von außen nach innen. Mit der forward chain können Sie zum Beispiel einstellen, dass Zugriffe auf eine öffentliche IP-Adresse Ihres Netzwerks im Internet auf eine interne Adresse in Ihrer DMZ / Ihrem LAN **umgeleitet** werden (und umgekehrt). Das heißt, der von außen Zugreifende sieht Ihre Infrastruktur nicht.

Kommerzielle Firewall Lösungen, beispielhaft:

- Cisco ASA / Meraki
- Juniper SRX
- Fortinet Fortigate
- Sonicwall
- Sophos NGFW
- Barracuda NGF
- Watchguard
- Checkpoint NGF
- Microsoft Azure FW

...



iptables „managen“ - live

- Regeln anzeigen
 - iptables -L
 - iptables -L -n -v
- Regeln speichern
 - iptables-save > **aktuelle.regeln**
- Regeln anpassen
 - cp **aktuelle.regeln** **neue.regeln**
 - vi/nano neue.regeln
- Regeln aktivieren
 - iptables-restore < **neue.regeln**
 - (service iptables restart)
- alte Regeln wieder herstellen
 - iptables-restore < **aktuelle.regeln**
 - (service iptables restart)

22.05.24

Horst Bräuner




9

Die Regeln werden über Dateien eingestellt.

Die aktuell gültigen Regeln können Sie sich mit iptables -L anzeigen lassen. Mit iptables-save und Umleitung per „>“ in eine Datei (siehe „Erste Schritte“) speichern Sie die aktuellen Regeln.

Mit iptables-restore und Umleitung per „<“ aus einer Datei (siehe „Erste Schritte“) aktivieren Sie gespeicherte Regeln.

VORSICHT: Stellen Sie Regeln nur dann ein, wenn Sie sicher sind, dass Sie sich nicht vom System aussperren. Also am Besten immer an der lokalen Konsole Ihres Systems arbeiten und nicht über Netzwerk ;-) Falls eine lokale Konsole nicht möglich ist, erstellen Sie sich zur Sicherheit immer eine Regel, die es Ihnen erlaubt mindestens per ssh zuzugreifen.



iptables „managen“ - permanent

- Regeln anzeigen
 - iptables -L
 - iptables -L -n -v
- Regeln speichern
 - iptables-save > **aktuelle.regeln**
- Regeln anpassen
 - cp **aktuelle.regeln** **neue.regeln**
 - vi/nano neue.regeln
- Regeln **beim Start des Netzwerks** aktivieren **DEPRECATED!**
 - vi /etc/network/if-pre-up.d/**iptables**
 - chmod +x /etc/network/if-pre-up.d/iptables
 - service networking restart
 - ln -s meineregeln iptables.up.rules
- **ALTERNATIV / NEU:**
 - apt install iptables-persistent
 - Dateien /etc/iptables/rulesv4 bzw. /etc/iptables/rules.v6 anpassen

```
root@host# cat /etc/network/if-pre-up.d/iptables
#!/bin/sh
/sbin/iptables-restore < /etc/iptables.up.rules
alternativ: iptables-apply /etc/iptables/rules.v4
```

```
root@host# /sbin/iptables-restore < /etc/iptables/rules.v4
Oder
reboot
```

22.05.24

Horst Bräuner

10

Wenn Sie die Firewall-Regeln bereits beim Start Ihres Systems aktivieren möchten, können Sie dafür ein eigenes Script in die Netzwerkkonfiguration stellen oder alternativ das Paket iptables-persistent installieren. Das Paket iptables-persistent fragt Sie bei der Installation, in welchen Dateien Sie die Regeln verwalten möchten. Die Regeln für IP-Version 4 verwalten Sie dann beispielsweise in der Datei /etc/iptables/rules.v4, analog die Regeln für IP-Version 6 in /etc/iptables/rules.v6. Das ist die Standardeinstellung.

Nachfolgend Beispiele für Konfigurationen mit unterschiedlichen Zugriffen.



iptables Beispiel (1)

```
root@debs01:~#  
root@debs01:~# iptables -L  
Chain INPUT (policy ACCEPT)  
target      prot opt source      destination  
  
Chain FORWARD (policy ACCEPT)  
target      prot opt source      destination  
  
Chain OUTPUT (policy ACCEPT)  
target      prot opt source      destination  
root@debs01:~# █
```

erlaubt: alles = Firewall „ausgeschaltet“

Standardeinstellung Firewall „aus“



iptables Beispiel (2)

```
root@debs01:~/firewall# cat iptables.up.nocifs.rules
# Generated by iptables-save v1.4.8 on Sun Mar  3 23:29:51 2013
*filter
:INPUT ACCEPT [1:40]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -d 127.0.0.0/8 ! -i lo -j REJECT --reject-with icmp-port-unreachable

-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 443 -j ACCEPT
-A INPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A INPUT -m limit --limit 5/min -j LOG --log-prefix "iptables denied: " --log
-level 7
-A INPUT -j REJECT --reject-with icmp-port-unreachable
-A FORWARD -j REJECT --reject-with icmp-port-unreachable
-A OUTPUT -j ACCEPT
COMMIT
# Completed on Sun Mar  3 23:29:51 2013
erlaubt eingehend: ping(8), ssh(22), http(80), https
(443) und alle ausgehenden Pakete (OUTPUT).
Alle anderen eingehenden Versuche erhalten „unreachable“
```

Von Außen ist ssh, http und https erlaubt. Von innen nach außen ist alles erlaubt.



iptables Beispiel (3)

```
root@debs01:~/firewall# cat iptables.up.cifs.rules
# Generated by iptables-save v1.4.8 on Sun Mar  3 23:29:51 2013
*filter
:INPUT ACCEPT [1:40]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -d 127.0.0.0/8 ! -i lo -j REJECT --reject-with icmp-port-unreachable

-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 139 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 443 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 445 -j ACCEPT
-A INPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A INPUT -m limit --limit 5/min -j LOG --log-prefix "iptables denied: " --log
-level 7
-A INPUT -j REJECT --reject-with icmp-port-unreachable
-A FORWARD -j REJECT --reject-with icmp-port-unreachable
-A OUTPUT -j ACCEPT
COMMIT
```

erlaubt **eingehend**: ping(8), ssh (22), http (80), https (443), cifs
(139, 445)
alle anderen Versuche erhalten „unreachable“

... von außen ist der Zugriff via ping, ssh, http, https und cifs erlaubt. Cifs („Common Internet File System“) sind Zugriffe auf Netzwerkfreigaben.

siehe https://de.wikipedia.org/wiki/Server_Message_Block#CIFS

iptables Beispiel (4)

```
# Generated by iptables-save v1.4.21 on Sun May 21 16:47:12 2017
*filter
:INPUT ACCEPT [47:5279]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [48:5074]
-A INPUT -i lo -j ACCEPT
-A INPUT -d 127.0.0.0/8 ! -i lo -j REJECT --reject-with icmp-port-unreachable
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
### ICMP
-A INPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A OUTPUT -p icmp -m icmp --icmp-type 0 -j ACCEPT
### ssh / http
-A OUTPUT -p tcp -m state --state NEW,ESTABLISHED -m tcp --dport 22 -j ACCEPT
-A OUTPUT -p tcp -m state --state NEW,ESTABLISHED -m tcp --dport 80 -j ACCEPT
-A INPUT -m limit --limit 5/min -j LOG --log-prefix "iptables denied: " --log-level 7
-A INPUT -j REJECT --reject-with icmp-net-unreachable
-A FORWARD -j REJECT --reject-with icmp-port-unreachable
-A OUTPUT -j REJECT --reject-with icmp-admin-prohibited
COMMIT
# Completed on Sun May 21 16:47:12 2017
```

erlaubt **ausgehend**: ping(0), ssh (22), http (80) und die **Antwort**-
Pakete (ping=8) dazu
alle anderen Versuche erhalten „unreachable“

... **ausgehend** ist **NUR** ping, ssh und http erlaubt, **eingehend** nichts. Für die Antwortpakete auf den ausgehenden ping (Port 0) muss der eingehende Port 8 „offen“/erlaubt sein.


iptables Beispiel (5)

```
# Generated by iptables-save v1.4.21 on Sun May 21 16:47:12 2017
*filter
:INPUT ACCEPT [47:5279]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [48:5074]
-A INPUT -i lo -j ACCEPT
-A INPUT -d 127.0.0.0/8 ! -i lo -j REJECT --reject-with icmp-port-unreachable
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
### ICMP Ping mit Antwort
-A INPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A OUTPUT -p icmp -m icmp --icmp-type 0 -j ACCEPT
### FTP
-A OUTPUT -p tcp --dport 21 -m state --state NEW,ESTABLISHED -j ACCEPT
-A INPUT -p tcp --sport 21 -m state --state ESTABLISHED -j ACCEPT
### PASV FTP
-A OUTPUT -p tcp --dport 20 -m state --state ESTABLISHED -j ACCEPT
-A INPUT -p tcp --sport 20 -m state --state ESTABLISHED,RELATED,NEW -j ACCEPT
### FTP High-Ports
-A OUTPUT -p tcp --sport 1024:65535 --dport 1024:65535 -m state --state ESTABLISHED,RELATED,NEW -j ACCEPT
-A INPUT -p tcp --sport 1024:65535 --dport 1024:65535 -m state --state ESTABLISHED -j ACCEPT
### outgoing ssh / http / https
-A OUTPUT -p tcp -m state --state NEW,ESTABLISHED -m tcp --dport 22 -j ACCEPT
-A OUTPUT -p tcp -m state --state NEW,ESTABLISHED -m tcp --dport 80 -j ACCEPT
-A OUTPUT -p tcp -m state --state NEW,ESTABLISHED -m tcp --dport 443 -j ACCEPT
-A INPUT -m limit --limit 5/min -j LOG --log-prefix "iptables denied: " --log-level 7
-A INPUT -j REJECT --reject-with icmp-net-unreachable
-A FORWARD -j REJECT --reject-with icmp-port-unreachable
```

erlaubt **ausgehend**: ping(0), ftp (20,21,+Highports), ssh (22), http (80), https (443) und die **Antwort**-Pakete (ping=8) dazu
alle anderen Versuche erhalten „unreachable“

... erlaubt **ausgehend**: ping(0), ftp (20,21,+Highports), ssh (22), http (80), https (443) und die Antwort-Pakete (ping=8) dazu. Alle anderen Versuche erhalten „unreachable“

Für FTP (File Transfer Protokoll) sind mehrere Ports notwendig, da die Daemons auf verschiedenen Ports kommunizieren.



iptables Beispiel (6)

```

... Auszug /etc/iptables/roules.v4 ..., s. Extra Dokumentation iptablesFORWARDsp.pdf
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
...
-A FORWARD -i enp0s9 -o enp0s8 -p tcp --syn --dport 80 -m conntrack --ctstate NEW -j ACCEPT
-A FORWARD -i enp0s9 -o enp0s8 -p tcp -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
-A FORWARD -i enp0s8 -o enp0s9 -p tcp -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

-A FORWARD -j REJECT --reject-with icmp-port-unreachable

COMMIT

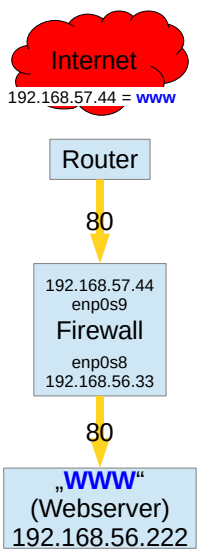
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]

-A PREROUTING -i enp0s9 -p tcp --dport 80 -j DNAT --to-destination 192.168.56.222
-A POSTROUTING -d 192.168.56.222 -o enp0s8 -p tcp --dport 80 -j SNAT --to-source 192.168.56.33

COMMIT

```

FORWARD-CHAIN: Zugriffe per http werden an einen internen Webserver weiter gegeben, der geNATet ist



22.05.24
Horst Bräuner
16

... erlaubt **aus- und eingehend**: ping(0) und die Antwort-Pakete (ping=8) dazu. Alle anderen Versuche erhalten „unreachable“

Für den „geschützten“ Zugriff auf den Webserver, intern oder in der DMZ, ist ein FORWARD-CHAIN eingerichtet, der die Zugriffe auf den Webserver durch die Firewall routet.

EXTERN ist im DNS der Webserver, hier im Beispiel, als 192.168.57.44 bekannt.



iptables ... -reject-with

Falls vom Kernel unterstützt, kann statt „Drop“ eine „Warum?“ Meldung mit gegeben werden

```
A [chain] -j REJECT --reject-with ...
```

```
icmp-net-unreachable ... Netzwerk nicht erreichbar
```

```
icmp-host-unreachable ... Host nicht erreichbar
```

```
icmp-port-unreachable ... Port nicht erreichbar
```


```
icmp-proto-unreachable ... Protokoll nicht unterstützt
```

```
icmp-net-prohibited ... Zugriff auf Netz verboten
```

```
icmp-host-prohibited or ... Zugriff auf Host verboten
```

```
icmp-admin-prohibited ... administrativ verboten
```

Falls Ihr System es unterstützt können Sie die Antworten auf vergebliche Kontaktversuche je Regel anpassen. Dafür steht Ihnen eine begrenzte Auswahl an Antworten zur Verfügung.



iptables ... -logging

Standard-Logfile /var/log/syslog bzw. nach log-Level oder

```
root@host:~# cat /etc/rsyslog.d/10-iptables.conf
:msg,contains,"nicht rein" /var/log/iptables.log
```

Logging **VOR** dem DROP/REJECT aktivieren

```
...
-A INPUT ... [z.B. -p tcp state --state -dport 80 -j ACCEPT]
-A INPUT -m limit --limit 5/min -j LOG --log-prefix „nicht rein“ --log-level 7
-A INPUT -j REJECT ... [z.B. --reject-with icmp-port-unreachable]

-A FORWARD -m limit --limit 5/min -j LOG --log-prefix „nicht rein“ --log-level 7
-A INPUT -j REJECT ... [z.B. --reject-with icmp-port-unreachable]

-A OUTPUT ... [z.B. -p tcp state --state -sport 80 -j ACCEPT]
-A OUTPUT -m limit --limit 5/min -j LOG --log-prefix „nicht rein“ --log-level 7
-A OUTPUT -j REJECT ... [z.B. --reject-with icmp-port-unreachable]
...
```

limit: 5 Einträge / Minute
log-level: 0=emergency -> 7=debug, s. /var/log/rsyslog.conf

22.05.24

Horst Bräuner




18

Wenn Sie die vergeblichen Kommunikationsversuche protokollieren möchten, können Sie das Standard-Logging (siehe „Systemverwaltung“) nutzen oder für jedes Log-Level andere Protokolldateien erstellen (lassen).

Hier im Beispiel wird das logging über eine selbst erstellte Datei /etc/rsyslog.d/10-iptables.conf für das Schlagwort „nicht rein“ in die Datei /var/log/iptables.log geschrieben.

Das Schlagwort kann frei gewählt werden. Werden für INPUT-, FORWARD oder OUTPUT-Chain unterschiedliche Schlagwörter gewählt, müssen dies selbstverständlich alle in der Konfiguration 10-iptables.conf eingetragen sein.

Denken Sie daran: Logging muss **VOR** dem Verwerfen der Kommunikation, also vor dem „REJECT“ aktiviert werden ;-)



fail2ban

- Verbindungsversuche AKTIV blockieren

```
root@host: apt-get install fail2ban
```

```
cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
```

- Konfiguration über /etc/fail2ban/jail.local
- Aufheben der Blockade mit fail2ban-client

```
fail2ban-client set [jail] unbanip [x.x.x.x]
```

z.B.

```
fail2ban-client set sshd unbanip 192.168.56.1
```

- nutzt/ergänzt iptables um HOSTS zu blockieren
 - fail2ban-chains wirken REAKTIV und nutzen die Protokolldateien der Daemons um „Angriffe“ zu erkennen


22.05.24

Horst Bräuner


19


Eine weitere Möglichkeit Ihren Server oder Ihr Netzwerk zu schützen ist „fail2ban“. Das ist eine aktive Methode ungewollte Kommunikation zu unterbinden. Fail2ban überwacht Systemprotokolle und erkennt zum Beispiel fehlgeschlagene Login-Versuche per ssh oder http/https oder E-Mail usw.

Je nach Konfiguration können Sie einstellen, dass nach dem X-ten Versuch der Zugreifende (besser: dessen IP) aktiv in der Firewall blockiert wird und weitere Versuche mit „Host unreachable“ von vornherein abgelehnt werden. Sie können ebenfalls einstellen, wie lange der „vergeblich Zugreifende“ blockiert wird.



fail2ban






```
# Generated by iptables-save v1.6.0 on Sun Jun  3 17:05:49 2018
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:f2b-proftpd - [0:0]
:f2b-sshd - [0:0]
-A INPUT -p tcp -m multiport --dports 21,20,990,989 -j f2b-proftpd
-A INPUT -p tcp -m multiport --dports 22 -j f2b-sshd
-A INPUT -i lo -j ACCEPT
-A INPUT -d 127.0.0.0/8 ! -i lo -j REJECT --reject-with icmp-admin-prohibited
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -m limit --limit 5/min -j LOG --log-prefix "nix in: " --log-level 7
-A INPUT -j REJECT --reject-with icmp-port-unreachable
-A FORWARD -m limit --limit 5/min -j LOG --log-prefix "nix fw: " --log-level 7
-A FORWARD -j REJECT --reject-with icmp-port-unreachable
-A OUTPUT -p tcp -m state --state ESTABLISHED -m tcp --sport 22 -j ACCEPT
-A OUTPUT -m limit --limit 5/min -j LOG --log-prefix "nix out: " --log-level 7
-A OUTPUT -j ACCEPT
-A f2b-proftpd -j RETURN
-A f2b-sshd -j RETURN
COMMIT
# Completed on Sun Jun  3 17:05:49 2018
```

fail2ban - chains
KEIN ACCEPT / REJECT

In diesem Beispiel wurde Fail2ban so konfiguriert, dass der FTP-Daemon überwacht wird.



fail2ban

```

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination
LOG        all  --  anywhere               anywhere
REJECT     all  --  anywhere               anywhere
limit: avg 5/min burst 5 LOG level
reject-with icmp-port-unreachable

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
ACCEPT     tcp  --  anywhere              anywhere
state ESTABLISHED tcp spt:ssh
limit: avg 5/min burst 5 LOG level

Chain f2b-proftpd (1 references)
target     prot opt source                destination
REJECT     all  --  192.168.56.152         anywhere
RETURN     all  --  anywhere              anywhere
reject-with icmp-port-unreachable

Chain f2b-sshd (1 references)
target     prot opt source                destination
RETURN     all  --  anywhere              anywhere
root@depp2:~#
  
```

fail2ban - chains
 NACH einem BAN
 (iptables -L)

Chain f2b-proftpd (1 references)
 target prot opt source destination
 REJECT all -- 192.168.56.152 anywhere
 RETURN all -- anywhere anywhere

22.05.24
Horst Bräuner
21

... wenn nun ein „Kunde“ zu oft das falsche Passwort eingibt, blockiert fail2ban die IP des „Kunden“ und passt iptables dafür dynamisch an.



fail2ban – Beispiel (1)

- Fehlerhafte ssh- und ftp- Logins „bannen“

```
root@host: apt-get install proftpd-basic
```

installiert proftpd, einen ftp-Server

```
vi /etc/fail2ban/jail.local
```

```
...
```

```
[sshd]
```

```
enabled = true
```

aktiviert den „jail“ sshd (alternativ über /etc/fail2ban/jail.d/defaults-debian.conf)

```
port     = ssh
```

```
logpath  = %(sshd_log)s
```

```
backend  = %(sshd_backend)s
```

```
...
```

```
[proftpd]
```

```
enabled = true
```

aktiviert den „jail“ proftpd

```
port     = ftp,ftp-data,ftps,ftps-data
```

```
logpath  = %(proftpd_log)s
```

```
backend  = %(proftpd_backend)s
```

```
...
```

Die Konfiguration von fail2ban am Beispiel ssh und ftp (Daemon „proftpd“).



fail2ban – Beispiel (2)

- Beobachten von „Angriffen“

root@host: **tail -f /var/log/fail2ban.log**

```
2018-06-03 17:28:52,100 fail2ban.filter [2317]: INFO [proftpd] Found
192.168.56.152
2018-06-03 17:28:57,909 fail2ban.filter [2317]: INFO [proftpd] Found
192.168.56.152
2018-06-03 17:28:58,708 fail2ban.actions [2317]: NOTICE [proftpd] Ban 19
2.168.56.152
```

„BAN“ wegen proftpd

```
.168.56.152
2018-06-03 17:35:17,495 fail2ban.filter [2317]: INFO [sshd] Found 192
.168.56.152
2018-06-03 17:35:18,567 fail2ban.filter [2317]: INFO [sshd] Found 192
.168.56.152
2018-06-03 17:35:18,575 fail2ban.actions [2317]: NOTICE [sshd] Ban 192.1
68.56.152
```


„BAN“ wegen sshd

```
root@ollie:~# ssh -l lll depp2
lll@depp2's password:
```

„BAN“ auf der Client-Seite ist ein
„normaler“ connection refused

```
root@ollie:~# ssh -l lll depp2
ssh: connect to host depp2 port 22: Connection refused
root@ollie:~# _
```

... und der „ban“ wegen zu vielen Fehlversuchen.



Portscanner - nmap

- z.B. „nmap“ scannt alle Ports eines Hosts
 - graf. Frontend z.B. Zenmap, nmapsi4
 - Optionen mit `root@host: nmap`
 - z.B. scan ohne Ping(ICMP)-Test `root@host: nmap -p0 192.168.57.152`
 - z.B. scan eines Netzwerks nach aktiven Hosts (Ping-Test)
 - `root@host: nmap -sP 192.168.56.*`

```
hbraeuner@tardis:~$ nmap depp2

Starting Nmap 7.40 ( https://nmap.org ) at 2018-06-03 17:43 CEST
Nmap scan report for depp2 (192.168.56.102)
Host is up (0.00027s latency).
rDNS record for 192.168.56.102: katze
Not shown: 995 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
443/tcp   open  https


Nmap done: 1 IP address (1 host up) scanned in 0.16 seconds
hbraeuner@tardis:~$
```

22.05.24Horst Bräuner24

Um heraus zu finden, auf welchen Ports ein Rechner/Server ansprechbar ist, können Sie eine Portscanner verwenden.

Denken Sie daran: Ports scannen ist vergleichbar mit „um das Haus gehen und schauen, ob eine Tür oder ein Fenster geöffnet werden kann“. Normalerweise sieht der gescannte Server im Scan die Vorbereitung eines Angriffs. Je nach Konfiguration (siehe „fail2ban“) erfolgen bereits beim Portscan Abwehrmaßnahmen.

Sie sehen oben, dass der Server 192.168.56.102 (Name „katze“) auf den TCP-Ports 21, 22, 25, 80 und 443 erreichbar ist. Bei einer Standardkonfiguration der Daemons heißt dies: Sie können per FTP, SSH, SMTP, HTTP und HTTPS zugreifen.



Portscanner - Beispiel

```
hbraeuner@tardis:~$ nmap depp2
Starting Nmap 7.40 ( https://nmap.org ) at 2018-06-03 17:52 CEST
Warning: 192.168.56.102 giving up on port because retransmission cap hit (10).

hbraeuner@tardis:~$ nmap -p0 depp2
Starting Nmap 7.40 ( https://nmap.org ) at 2018-06-03 18:01 CEST
Nmap scan report for depp2 (192.168.56.102)
Host is up (0.0019s latency).
rDNS record for 192.168.56.102: katze
PORT STATE SERVICE
0/tcp closed unknown

Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
hbraeuner@tardis:~$
```

```
Starting Nmap 7.40 ( https://nmap.org ) at 2018-06-03 17:54 CEST
Nmap scan report for 192.168.56.1
Host is up (0.00060s latency).
Nmap scan report for katze (192.168.56.102)
Host is up (0.00093s latency).
Nmap scan report for ollie (192.168.56.152)
Host is up (0.0016s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 2.55 seconds
hbraeuner@tardis:~$
```

scan ohne Ping-Test, z.B. wegen blockierten Ports durch Firewall




scan eines Netzwerks nach aktiven Hosts

22.05.24Horst Bräuner25

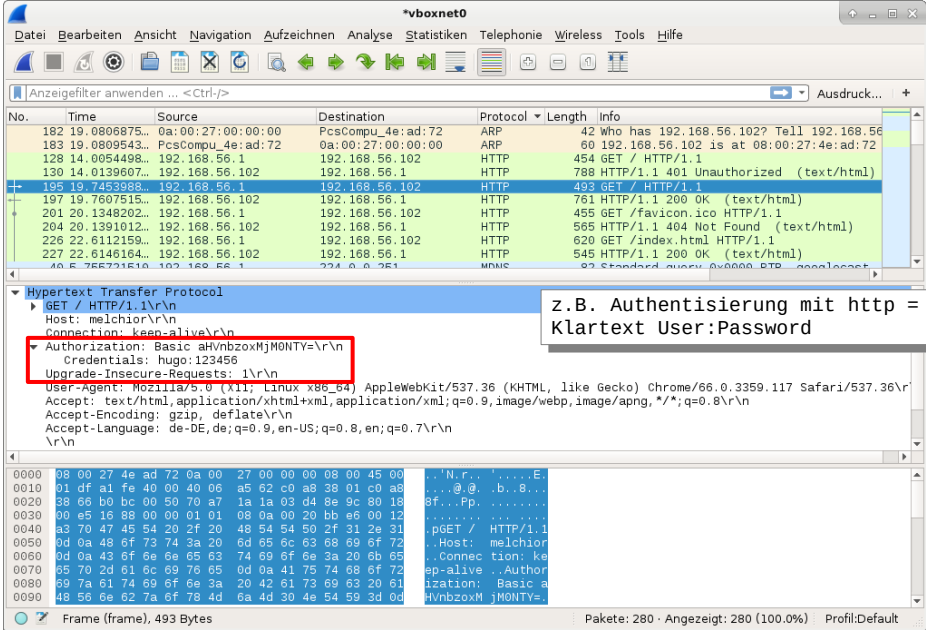
Ein Portscan kann auf verschiedene Art erfolgen. Ein einfacher „ping“ leitet den Scan standardmäßig ein. Ein erfolgloser „ping“ beendet den Scan.

Falls ein Rechner jedoch „ping“ blockiert, können Sie den „ping“-Test übergehen und weitere Ports dennoch scannen.

Ein reiner „ping“-Scan (-sP) ist eine Methode, ein ganzes Netzwerk (oder ~segment) nach aktiven Geräten abzusuchen.

wireshark - Protokollanalyse



The screenshot shows the Wireshark interface with a packet list on the left and a packet details pane on the right. The selected packet is an HTTP GET request. In the details pane, the 'Authorization' field is expanded, showing 'Basic aHvnb3oxMjMONTY='. A red box highlights this field, and a callout bubble points to it with the text: 'z.B. Authentisierung mit http = Klartext User:Password'.

22.05.24
Horst Bräuner
26

Zum Abschluss noch ein Screenshot des Programms Wireshark. Wireshark ist eine grafische Oberfläche um Netzwerkverkehr mitzuschneiden und zu analysieren.

Oben dargestellt der unverschlüsselte HTTP-Verkehr mit einem Webserver. Sie sehen eine protokollierte Anmeldung mit Benutzernamen und Passwort (= „Credentials“). Benutzer „hugo“ mit Passwort „123456“ ist im Klartext lesbar.