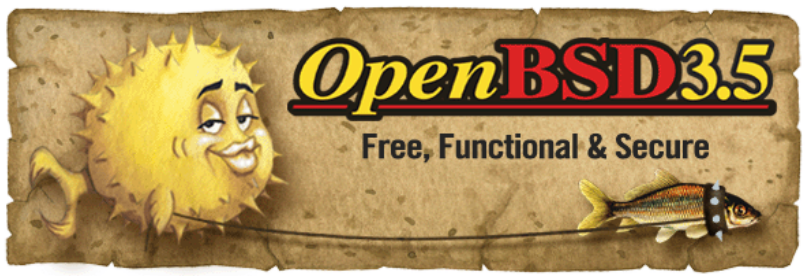
Themenabend C3D2 13.08.2004



OpenBSD Packet Filter PF

Tibor Varkonyi ~ tibyr@c3d2.de

Über was sprechen wir?

- Konfiguration von PF
- Aufbau der Datei pf.conf
- Erweiterte Features von PF
- Firewall Failover mit pfsync und CARP
- Dokumentationen

Der OpenBSD Packet Filter

- Bestandteil des generischen Kernels
- Filtert, normalisiert und konditioniert TCP/IP Fluss
- Network Adress Translation (NAT)
- Bandbreitenkontrolle / Queuing
- Paketpriorisierung
- Keep State Funktion (stateful inspection)
- Logging (pflogd → /var/log/pflog)

Aktivierung von pf

- pf=YES in /etc/rc.conf einfügen (Neustart)
- pfctl –e (enable) / pfctl –d (disable) im laufenden Betrieb
- PF rules liegen in /etc/pf.conf
- Forwarding in /etc/sysctl.conf aktivieren
- → net.inet.ip.forwarding=1
- → net.inet6.ip6.forwarding=1

pf.conf

- wird von Anfang bis Ende geparsed (Reihenfolge wichtig !)
- besteht aus sieben Teilen:
- **Makros:** Benutzedefinierte Variablen, die IP Adressen, Schnittstellennamen, usw. enthalten
- **Tabellen:** Eine Tabelle dient dazu, Listen von IP Adressen zu speichern
- **Optionen:** Zahlreiche Möglichkeiten um zu steuern wie der PF arbeitet
- **Scrub:** Paketaufbereitung zur Normalisierung oder zur Defragmentierung
- **Queueing:** Stellt Bandbreitenkontrolle und Paketpriorisierung zur Verfügung
- **Translation:** Regelt NAT und Paketweiterleitung
- **Filter Regeln:** Erlauben das selektive Filtern oder Blockieren von Paketen während sie eine der Schnittstellen passieren

Beispiel für pf.conf

```
### VARIABLEN ###
Ext = "tun0" # Device an dem das Internet angeschlossen ist
Int = "<internes_device>" # Device an dem das interne Netz haengt
IntNet = "192.168.1.0/24" # Adressraum des internen Netzes
RouterIP = "192.168.1.1" # IP Adresse des Routers
Loop = "lo0" # Loopback Device
# NAT aktivieren
nat on $Ext from $IntNet to any -> $Ext static-port
# Generelle Block Regel
block on $Ext
# Freiwillig machen wir keinen Mucks ;)
block return log on $Ext
# Wir wollen kein IPv6.0
block quick inet6
# Loopback Device darf alles
pass quick on $Loop
```

pf.conf Teil 2

```
# Erschwert scannen mit nmap und co.
block in log quick on $Ext inet proto tcp from any to any flags FUP/FUP
block in log quick on $Ext inet proto tcp from any to any flags SF/SFRA
block in log quick on $Ext inet proto tcp from any to any flags /SFRA
block in log quick on $Ext os NMAP
# IP Spoofing verhindern
block in log quick on $Ext inet from <NoRoute> to any
block in log quick on $Ext inet from any to <NoRoute>
# Active FTP erlauben
pass in quick on $Ext inet proto tcp from any to any port > 49151 user proxy flags S/SAFR keep state
# Ping akzeptieren
pass in quick on $Ext inet proto icmp all icmp-type 8 code 0 keep state
# Ports nach aussen oeffnen
pass in quick on $Ext inet proto tcp from any to any port $InServicesTCP flags S/SAFR keep state label
      ServicesTCP
anchor passin
# Raus darf (fast) alles
pass out quick on $Ext keep state queue (q_def,q_pri)
```

Packet Filtering

- **block** Das Paket wird nicht durchgelassen, sondern verworfen.
- **pass** Das Paket darf passieren.
- trifft keine Regel zu, so gilt die Standard Regel (pass).
- erste Regel sollte daher alles blocken !

Anchors

- im laufenden Betrieb können Regelsätze hinzugefügt werden
- Reset der Firewall nicht nötig
- pf rules werden in /etc/proggie.redirect bzw. in /etc/proggie.passin eingefügt
- pf.conf → z.B. anchor redirect / passin

Packet Scrubbing

- normalisiert alle ankommenden Pakete
- hilft Zweideutigkeiten zu vermeiden
- ungültige Flagkombinationen werden bereinigt
- fragmentierte Pakete werden wieder zusammengesetzt
- pf.conf → scrub in all

Queuing / Rate Limiting

- Pakete können an Warteschlangen (QUEUE) angehangen werden
- Bandbreitensteuerung möglich
- pf.conf → altq on dc0 cbq bandwidth 5Mb queue f std, http, mail, ssh g

Redirection

- Verändert bei eingehenden Paketen den Port und/oder die Ziel-Adresse
- Pakete können an einen anderen Rechner weitergeleitet werden
- sinnvoll, wenn das interne Netz über eine externe Adresse mit dem Internet verbunden ist
- pf.conf → rdr on tl0 proto tcp from any to any port 80 -> 192.168.1.20

NAT

- verbindet ein Netz über nur eine IP-Adresse mit einem anderen
- privates Netz kann so z.B. mit dem Internet verbunden werden
- Von Aussen ist das private Netz über eine einzige IP-Adresse erreichbar
- pf.conf → nat on tl0 from dc0/24 to any -> (tl0)

OS-Fingerprinting

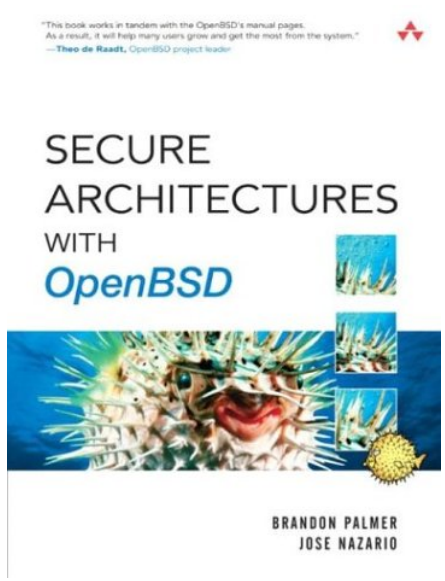
- PF erkennt Remote-Betriebssysteme über Merkmale im TCP SYN Paket
- Vergleich mit Fingerprintfile (/etc/pf.os)
- Verbindungsanfragen von unerwünschten OS können abgewiesen werden
- pf.conf → pass in on $ext_if any os OpenBSD keep state
  block in on $ext_if any os "Windows 2000"

Anti Spoofing

- Angreifer fälschen ursprüngliche IP-Adresse
- Vortäuschen eines trusted Hosts wäre möglich
- PF kennt verschiedene Methoden zur Erkennung
- pf.conf → antispoof for fxp0 inet

Miscellaneous

- PF ist Bestandteil des generischen Kernels
- OpenBSD Homepage dokumentiert hervorragend
- www.openbsd.org/faq/pf/de/index.html
- www.countersiege.com/doc/pfsync-carp/
- www.fmi.uni-passau.de/~grafj/openbsd/3.5/index.html
- Buch-Tipp:



pfsync – you see?