15. Firewalls unter UNIX

Gliederung

- Allgemeines
- Linux: iptables
- OpenBSD, FreeBSD: PF Toolkit
- BSD, Solaris: IPF Toolkit
- **Löcher in Firewalls – Virtuelle Private Netze**

Humboldt Universität Informatik
Systems Architecture Group
http://sar.informatik.hu-berlin.de

15.4 Firewall - Virtuelle private Netze

Was ist ein VPN?

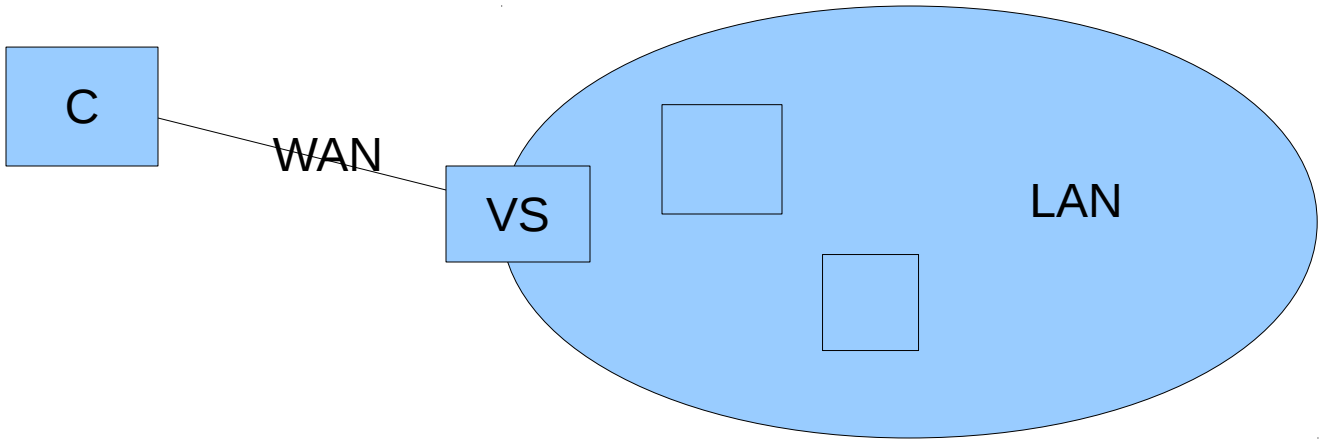
- ein Verfahren zur sicheren Kommunikation zwischen Rechnern an zwei oder mehreren Standorten.
- Alle beteiligten Rechner „bilden sich“ ein, in einem lokalen Rechnernetz zu sein.
- Die Kommunikation erfolgt verschlüsselt. Verschiedene kryptographische Verfahren werden unterstützt.
- Die Kommunikationspartner müssen sich authentifizieren.
- Nachfolger von direkten statischen WAN-Verbindungen, die sehr teuer waren und sind.

15.4 Firewall - Virtuelle private Netze

Anwendungsmöglichkeiten:

1. Anbindung eines externen Clienten an ein Firmennetz (Loch im Firewall)

   VPN-Server im Firewall oder dahinter

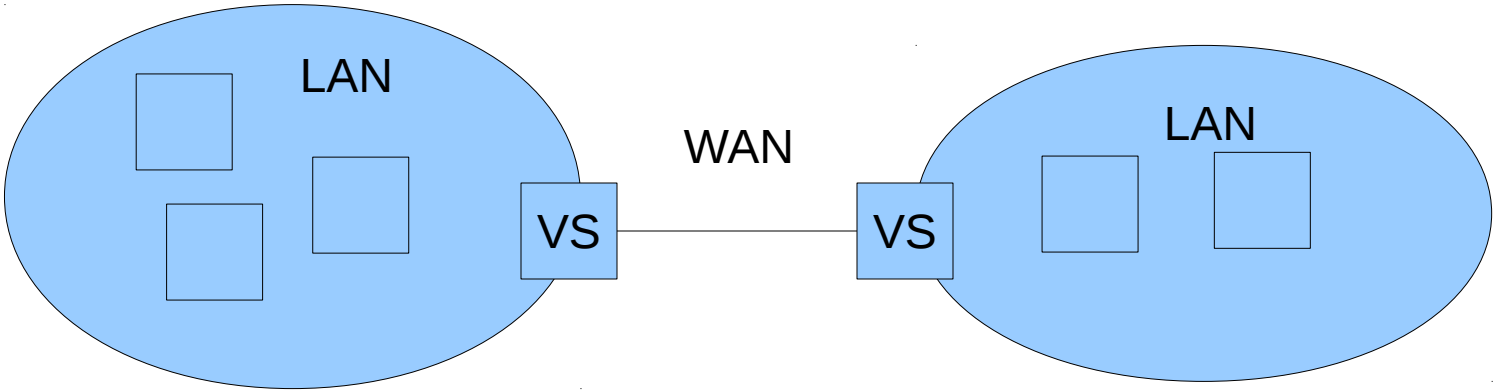


Systems Architecture Group
http://sar.informatik.hu-berlin.de

15.4 Firewall - Virtuelle private Netze

Anwendungsmöglichkeiten:

2. Verbindung eines Teils eines Firmennetzes mit einem anderen Teil eines Firmennetz (VPN-Server im Firewall)

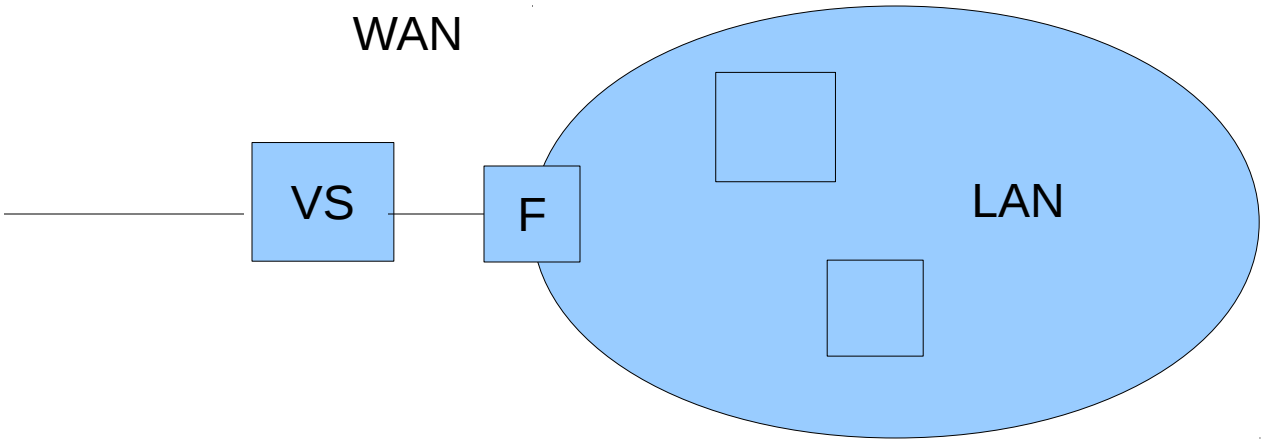


Systems Architecture Group
http://sar.informatik.hu-berlin.de

15.4 Firewall - Virtuelle private Netze

Anwendungsmöglichkeiten:

3. Alle Rechner eines Firmennetzes haben keinen Zugang zum Internet. Der Zugang zum Internet darf erst nach Authentifizierung freigegeben werden.



Systems Architecture Group
http://sar.informatik.hu-berlin.de

15.4 Firewall - Virtuelle private Netze

Ziele und Methoden

Ziele

- Vertraulichkeit
- Integrität
- Authentizität
- Verifizierbarkeit

Methoden

- symmetrische und asymmetrische Schlüssel
- digitale Signaturen
- Nutzerverifizierung mittels Zertifikaten oder Nutzername/Passwort
- Message Digest (Einweg-Hash-Funktionen)

15.4 Firewall - Virtuelle private Netze

Historisches(1)

1998: IPSEC RFCs: 4301(2401), 4302(2402), 4303(2406), 4306(2407,...), 3830

„Security Architecture for the Internet Protocol“

IKE – Internet Key Exchange

AH – Authentication Header

ESP – Encapsulated Security Payload

Probleme:

- Export Kontrolle (USA)
- Hohe CPU-Leistung für alle beteiligten Komponenten (Rechner und Router) notwendig
- Komplexität, Kernelintegration

Historisches(2)

- Betriebsmodi von IPSEC:
  - Tunnelmodus
    - Kryptographie wird zwischen zwei Gateways durchgeführt, es werden neue Header erzeugt.
    - für LAN zu LAN Verbindungen eingesetzt
  - Transportmodus
    - Kryptographie wird für die Nutzerdaten durchgeführt
    - für Client zu Client Verbindungen eingesetzt
- Verfügbarkeit:
  - Windows, Linux, Unix

Historisches(3)

Implementierungen für VPNs:

- IPSEC: IPSEC, Openswan, FreeS/WAN, strongSwan
- TLS/SSL – End to End VPNs
- ViPNet
- PPTP – ohne Verschlüsselung
- L2TP – ohne Verschlüsselung
- PPP – ohne Verschlüsselung
- SSH-Tunnel – umständlich, risikobehaftet

Allgemeine Nachteile: Komplexität, Herstellergeheimnisse, Kompatibilität, vielfach Kerneleingriffe notwendig

OpenVPN

Funktionsweise

Grundidee: Zwei virtuelle Interfaces (tun bzw. tap) werden durch eine reale TCP oder UDP Verbindung miteinander gekoppelt. Der Datentransport auf dieser Verbindung erfolgt verschlüsselt. Der Zugriff auf das virtuelle Interface ist identisch zu einem physischen Interface – open, close, read, write, ioctrl, d.h. es können die gleichen Protokolle (UDP, TCP, ICMP,...) wie bei einem physischen Interface benutzt werden.

Virtuelle Interfaces:

- tun: Stellt für den Rechner eine Punkt zu Punkt Verbindung dar. Ein normales Programm kann dieses Interface öffnen und anschließend IP-Pakete schreiben und lesen.
- tap: Stellt für den Rechner ein Ethernet-Adapter dar. Ein normales Programm kann dieses Interface nicht öffnen und aber bei einem geöffneten Interface IP-Pakete schreiben und lesen (Broadcast möglich).

15.4 Firewall - Virtuelle private Netze

OpenVPN

Entwicklung

Author: James Yonan

- Version 1.0 23.3.2002 - TLS Unterstützung, Schlüsselaustausch
- Version 1.2 22.5.2002 - Solaris, OpenBSD, MacOSX
- Version 1.3 7.10.2002 - IPV6 für tun-Device
- Version 1.4 7.5.2003
- Version 1.5 20.11.2004 - TCP
- Version 2.0 17.4.2005 - Server mode, Managment Int., auth-pam
- Version 2.0.9 1.10.2006 - Windows Vista
- Version 2.3.2 31.3.2013 - Aktuelle Version

15.4 Firewall - Virtuelle private Netze

OpenVPN

Verfügbarkeit

- Solaris
- AIX
- OpenBSD
- FreeBSD
- NetBSD
- MacOSX
- Windows: 2000, 2003, XP, Vista, Windows 7
- Linux
-

Problem für die Weiterverbreitung: Driver für tun- und tap-Devices

Humboldt Universität Informatik
Systems Architecture Group
http://sar.informatik.hu-berlin.de

15.4 Firewall - Virtuelle private Netze

OpenVPN

Vorteile für OpenVPN:

basierend auf SSL:

- kennen alle
- wird von vielen benutzt
- viele vertrauen auf SSL
- sehr weit verbreitet
- einfach
- robust
- gut getestet
- vielfach durchleuchtet

Open Source

Nur ein externer Port notwendig – geht leicht durch Wände

15.4 Firewall - Virtuelle private Netze

OpenVPN

Client-Server Architektur von OpenVPN

Server

- Wird bei Systemstart aktiviert
- bearbeitet Requests für den Tunnelaufbau zu einem Clienten einschließlich der Authentifizierung des Clienten
- Modifiziert Routen des Rechners und des Firewalls

Client

- Initiiert den Tunnelaufbau, sendet Request und Authentifizierungs-Informationen an den Server
- Verarbeitet Optionen des Servers (routing, dhcp)

15.4 Firewall - Virtuelle private Netze

OpenVPN

Client-Server-Architektur von OpenVPN

Features:

- Verbindung über TCP oder UDP
- Variable MTU-Size
- Ein Server kann mehrere Clienten bedienen
- NAT-Unterstützung
- re-connect nach Neustart
- Multithreaded, chroot, userspace
- Kompremierung
- Plugin-Scripte möglich

15.4 Firewall - Virtuelle private Netze

OpenVPN

Client-Server-Architektur von OpenVPN

Authentifizierungsmöglichkeiten:

- X.509 Zertifikate
  - openssl
  - CA notwendig
  - aufwendig
- Pre-Shared Key
  - einfach
  - nicht sicher
- Plugins
  - PAM
  - LDAP

15.4 Firewall - Virtuelle private Netze

OpenVPN

Bestandteile

- Programm: openvpn
- Konfigurationsfile: openvpn.conf
- Driver: tun/tap
- Startscript für Openvpn-Server
- X.509 Zertifikate (Server, CA)
- eventuell Preshared Key
- Scripte zur Authentifizierung
- eventuell Firewallregeln

Dokumentation: openvpn.8, http://openvpn.net/man.html

Alle openvpn-Optionen können im Konfigurationsfile stehen!

15.4 Firewall - Virtuelle private Netze

OpenVPN

Beispiel 1 (1)

Szenarium: LAN durch Firewall geschützt. Clienten aus dem WAN sollen einen Netzzugang zum LAN erhalten. Die Clienten sollen sich über ein Zertifikat beim Server authentifizieren. Die Clienten sind sicher, dass sie mit dem Server verbunden sind. openvpn-Server wird auf dem Firewall installiert. Betriebssystem OpenBSD

Arbeitsschritte allgemein:

- Zertifikate bereitstellen – CA-Zertifikat, Server-Zertifikat für denFirewall, Clienten-Zertifikat für jeden Clienten, CRL
- Preshared Key erzeugen: openvpn --genkey --secret sar.key
  (für Client und Server notwendig bei TLS-Nutzung)

15.4 Firewall - Virtuelle private Netze

OpenVPN

Beispiel 1 (2)

Arbeitsschritte Server:

- CA-Zertifikat, CRL, Preshared Key, eventuell Server-Zertifikat
- Diffie Hellman Parameter erzeugen:
  openssl dhparam -out dh1024.pem 1024
- openvpn.conf auf dem Firewall bereitstellen
- Konfigurationsfile des Firewall pf.conf aktualisieren und laden
- openvpn auf dem Firewall als Server starten

15.4 Firewall - Virtuelle private Netze

OpenVPN

Beispiel 1 (3)

Arbeitsschritte Client:

- CA-Zertifikat, Preshared Key, Client-Zertifikat
- openvpn.conf auf dem Clienten anpassen
- openvpn auf Client starten, eventuell Passphrase eingeben

OpenVPN

Beispiel 1 (4)

Server: Sar/openvpn.conf:

```
#############################################################
# Sample OpenVPN 2.0 config file for          #
# multi-client server.
# Which local IP address should OpenVPN
# listen on? (optional)
# local a.b.c.d
port 1194       # standard
proto upd       # oder tcp
dev tun0
pkcs12 /etc/openvpn/sar.informatik.hu-berlin.de.p12    # CA-Zertifikat
crl-verify /etc/openvpn/crl.pem
```

OpenVPN

Beispiel 1 (5)

```
# Diffie hellman parameters
dh /etc/openvpn/dh1024.pem
server 192.168.10.0 255.255.255.0
keepalive 10 60          # ping alle 10 Sekunden, nach 60 Sekunden down
tls-auth /etc/openvpn/sar.key 0          # preshared key
comp-lzo                    #  Kompression, auch auf dem Clienten angeben
user nobody
group nobody
persist-key                          # kein lesen der keys bei reset
persist-tun                          # beibehalten des tun-Devices bei reset
status /etc/openvpn/openvpn-status.log
```

15.4 Firewall - Virtuelle private Netze

OpenVPN

Beispiel 1 (6)

```
log-append  /var/log/openvpn.log
verb 4
float                   # IP-Adresse des Clienten kann sich ändern
```

OpenVPN

Beispiel 1 (7)

Client Sar/sar.conf

```
##############################################
# Sample client-side OpenVPN 2.0 config file #
# for connecting to multi-client server.
script-security 2
client
daemon
dev tun
proto udp
remote sar.informatik.hu-berlin.de 1194
```

OpenVPN

Beispiel 1 (8)

```
resolv-retry infinite
nobind
pkcs12 /etc/openvpn/Jan-Peter_Bell.p12
tls-auth /etc/openvpn/sar.key 1
comp-lzo
log /var/log/openvpn-sar
verb 3
redirect-gateway
route 192.168.2.0 255.255.255.0
route 192.168.3.0 255.255.255.0
```

OpenVPN

Beispiel 1 (9)

```
# DNS
dhcp-option DOMAIN sar.informatik.hu-berlin.de
dhcp-option DNS 192.168.2.20
# up   "/sbin/modify_resolvconf modify  -s openvpn -p /usr/sbin/openvpn -t 'Name
     Server modified by openvpn: /etc/openvpn/SAR-VPN.ovpn' -l 'sar.informatik.hu-
     berlin.de informatik.hu-berlin.de' -n '192.168.2.20 141.20.20.50 141.20.40.2';
     echo > /dev/null"
# down "/sbin/modify_resolvconf restore -s openvpn; echo > /dev/null"
```

15.4 Firewall - Virtuelle private Netze

OpenVPN

Beispiel 1 (10)

Firewall: pf.conf (OpenBSD)

```
ext_if="bge0"
ExtAddr="141.20.23.63"
VPN0="tun0"
vpn_port0 = "{ 1194 }"
vpn_net0 = "{ 192.168.10.0/24 }"
nat on $ext_if from $vpn_net0 to any -> $ExtAddr
pass in on $ext_if inet proto udp from any to ($ext_if) port $vpn_port0
pass in quick on $VPN0  all
pass out quick on $VPN0  all
```

15.4 Firewall - Virtuelle private Netze

OpenVPN

Beispiel 2 (1)

Szenarium: Server steht im WAN. Clienten aus dem WAN sollen eine statische IP-Adresse aus dem Adressbereich des Servers für weitere Verbindungen erhalten. Die Clienten sollen sich über Nutzeraccount und Passwort authentifizieren. Clienten sollen das Zertifikat des Servers überprüfen. Betriebssystem: Solaris 10. Besonderheit: 2 x Ethernet

Arbeitsschritte allgemein:

- Zertifikate bereitstellen – CA-Zertifikat, Server-Zertifikat, CRL
- Preshared Key erzeugen:
  openvpn –genkey –secret base1.key
  (für Client und Server notwendig bei TLS-Nutzung)

15.4 Firewall - Virtuelle private Netze

OpenVPN

Beispiel 2 (2)

Arbeitsschritte Server:

- CA-Zertifikat, CRL, Preshared Key, Server-Zertifikat (/usr/share/ssl/misc)
- Diffie Hellman Parameter erzeugen:
  openssl dhparm -out dh1024.pem 1024
- openvpn.conf auf dem Server bereitstellen
- Authentifizierungsscript bereitstellen
- Firewall-Regeln für den Server bereitstellen: /etc/ipf/ipf.conf und /etc/ipf/ipnat.conf – damit NAT funktioniert
- NAT aktivieren: routeadm -u -e ipv4-forwarding
- openvpn auf dem Firewall als Server starten

OpenVPN

Beispiel 2 (3)

Arbeitsschritte Client:

- CA-Zertifikat, Preshared Key, Nutzer auf Server
- openvpn.conf auf dem Client anpassen
- openvpn als Client starten, eventuell Nutzername und Passwort eingeben

OpenVPN

Beispiel 2 (4)

File: /etc/init.d/openvpn

```
#!/bin/sh
case $1 in
'start')
        ifconfig eri0 deprecated netmask 255.255.255.255
        /opt/csw/bin/openvpn –config /etc/csw/openvpn/openvpn.conf
        ;;
'stop')
        PID=`ps -efa | grep /opt/csw/bin/openvpn | grep -v grep | awk '{ print $2 ; }'`
        kill -TERM $PID
         route delete 192.168.199.0 192.168.199.2
        ifconfig tun0 unplumb
         ;;
esac
```

15.4 Firewall - Virtuelle private Netze

OpenVPN

Beispiel 2 (5)

Weitere Files Server:

Wlanin/auth_script	- Authentifizierungsscript
Wlanin/openvpn.conf	- Konfigurationsfile openvpn
Wlanin/ipf.conf	- Konfigurationsfile ipf
Wlanin/ipnat.conf	- Konfigurationsfile ipnat

Weitere Files Client:

Wlanin/wlanin.conf	- Konfigurationsfile openvpn

15.4 Firewall - Virtuelle private Netze

OpenVPN