Linux FreeS/WAN

Thomas Hungenberg

<th@demonium.de>

24. Januar 2000

Inhaltsverzeichnis

1 Was ist FreeS/WAN?

1.1 Entstehung

John Gilmore[1] setzte sich für 1996 das Ziel, mindestens 5% des Internetverkehrs vor dem passiven Abhören zu schützen. Aus diesem Anlaß rief er ein Projekt zur Implementierung eines *S/WAN* (*Secure Wide Area Network*)[2] ins Leben. Da es sich bei diesem Projekt um die Erstellung von *freier* Software handelt, bekam es den Namen *FreeS/WAN*[3], um es deutlich von den verschiedenen kommerziellen Implementierungen zu unterscheiden. Der Ausdruck S/WAN wurde ursprünglich von RSA Data Security[4] geprägt, die die Koordination zwischen verschiedenen S/WAN-Projekten übernommen haben, um eine Interoperabilität zu garantieren.

Wie sich schnell herausstellte, war das von Gilmore angestrebte Ziel weit zu ehrgeizig gesetzt. Nicht zuletzt aus dem Grund, daß die zugrundeliegenden Standards noch nicht sehr ausgereift waren. Aus diesem Grund galten die angestrebten 5% weiterhin als „Meßlatte“ für die folgenden Jahre.

1.2 Entwicklungsziele

Eines der Hauptziele der Entwicklung von FreeS/WAN ist es, eine Möglichkeit zu schaffen, mittels einfacher „encryption boxes“ einen wachsenden Anteil der Kommunikation im Internet gegen das passives Abhören durch private, vor allem aber durch regierungsnahe Organisationen, zu schützen. Diese „encryption boxes“ sollen mittels handelsüblicher PCs, auf denen freie (Linux-)Software läuft, realisiert werden.

Als Einstieg dazu soll der Einsatz von FreeS/WAN auf *VPN-Gateways*[11] zur Kopplung von unverschlüsselt kommunizierenden Subnetzen über das (dann zur verschlüsselten Übertragung genutzte) Internet dienen. In der aktuellen Version schließt dies auch eine erste Lösung zur vertrauenswürdigen Ankopplung von mobilen PCs (ggf. auch mit dynamischen IP-Adressen) an entfernte VPN-Gateways ein. Dieses Szenario wird vom FreeS/WAN Entwickler-Team „Road Warrior“ genannt.

Bei der Entwicklung von FreeS/WAN wurde eine möglichst genaue Einhaltung des von der *IP Security Working Group* der *Internet Engineering*

Task Force (IETF)[6] entwickelten IPsec-Standards[7] angestrebt, welcher auch Bestandteil der nächsten Version des Internet Protokolls IPv6[9] sein wird.

Außerdem wird streng darauf geachtet, daß der Programmcode von FreeS/WAN *frei exportierbar* bleibt und nicht irgendwelchen Export-Restriktionen (z. B. den US-amerikanischen) unterliegt. Aus diesem Grund wird beispielsweise keinerlei Unterstützung (Quelltext oder auch nur Konfigurationshilfe) aus den USA oder von US-Bürgern[1] akzeptiert.

Selbstverständlich kommt in einem solchen Projekt nur *starke* Verschlüsselung zum Einsatz. Für eine detaillierte Beschreibung der verwendeten Authentifizierungs- und Verschlüsselungsalgorithmen siehe Kapitel 2.

1.3 Linux-Implementierung von FreeS/WAN

Im April 1999 wurde die Version 1.0 der Linux[5] Implementierung von FreeS/WAN veröffentlicht. Diese funktionierte ausschließlich mit 2.0.x Kernel-Versionen (empfohlen wurde die damals aktuelle Version 2.0.36). Am 20. Dezember 1999 wurde die derzeit aktuelle Version 1.2 von Linux FreeS/WAN veröffentlicht, die nun sowohl mit dem derzeit aktuellen Kernel der 2.0er Reihe (2.0.38) als auch mit Kerneln der 2.2er und 2.3er Reihe zusammenarbeitet.

> *„I have made enough money from several successful startup companies, that for a while I don't have to work to support myself. I spend my energies and money creating the kind of world that I'd like to live in and that I'd like my (future) kids to live in. Keeping and improving on the civil rights we have in the United States, as we move more of our lives into cyberspace, is a particular goal of mine."* – John Gilmore

[1] Das Gehirn eines US-Bürgers gilt nach US-amerikanischem Recht als US-Territorium – selbst wenn es (bzw. die dazugrhörige Person) sich außerhalb der USA befindet.[10]

2 Derzeit implementierte Teile der IPsec-Spezifikation

2.1 IPsec-Transformationen

Linux FreeS/WAN unterstützt *AH-Header*[13] und *ESP-Header*[14] mit den Enkapsulierungsmodi *Transport-* und *Tunnelmodus* (optional ESP mit integriertem AH).

2.2 Schlüsselvereinbarung

Die automatische Schlüsselvereinbarung geschieht im *IKE Main Mode*[15] mittels des *Diffie-Hellman(DH)* Schlüsselaustausch-Protokolls (768, 1024 oder 1512 Bit). Die Authentifizierung läuft dabei entweder über lokal gespeicherte *Preshared Secrets* (in Form von ASCII-Zeichenketten) oder in der aktuellen Version auch über RSA-Signaturen. Optional ist ebenfalls eine manuelle Schlüsselvereinbarung möglich.

2.3 Datenverschlüsselung

Derzeit ist *Triple-DES* mit einer Schlüssellänge von 168 Bit der einzige in Linux FreeS/WAN nutzbare Algorithmus zur Datenverschlüsselung. Der für *(Single-)DES* nötige Algorithmus ist zwar im Source-Code vorhanden (da er für die Implementiertung von *Triple-DES* nötig ist), steht jedoch nicht für die Benutzung zur Verfügung[2]. Außerdem ist die vom IPsec-Standard geforderte „null encryption“[16] (unverschlüsselte Übertragung) implementiert.

2.4 Message-Digest Algorithmen

Für die Benutzung in AH- und ESP-Headern sind die *MAC*-Algorithmen *HMAC-MD5* (96 Bit)[17] und *HMAC-SHA-1* (160 Bit)[18] implementiert.

[2] Die IETF stufte im März 1999 *Single-DES* als *unsicher* für die Benutzung in IPsec ein.

3 Praktischer Test

Der praktische Test von Linux FreeS/WAN wurde in einer Testreihe im Rahmen der Veranstaltung „Virtual Private Networks (VPN)“ von Prof. Dr. Hartmut Pohl im Wintersemester 1999/2000 an der Fachhochschule Rhein-Sieg[19] durchgeführt. Aus diesem Grund kam neben Linux auch das Betriebssystem *Microsoft Windows NT* zum Einsatz.

Die Performance- und Angriffstests wurden in Zusammenarbeit mit den Kommilitonen Stephan Baum und Swen Sommermeyer durchgeführt.

3.1 Testszenario

Der Einsatz von Linux FreeS/WAN zur Realisierung eines *Site-to-Site Virtual Private Network (VPN)* wurde in einer Versuchsanordnung entsprechend Abbildung 1 getestet.

3.1.1 Hardware

Die beiden Gateway-Rechner waren mit einem Intel Pentium II 300 MHz und 128 MByte RAM ausgestattet. Die beiden Workstations und der „Angriffsrechner“ liefen mit einem Intel Pentium 200 MHz und 64 MByte RAM. Als Netzwerkkarten kamen die Modelle 900-Combo und 905B-TX der Firma 3Com zum Einsatz. Die Rechner in Ethernet 2 waren netzwerktechnisch über einen 10Base-T HUB (Modell AT-MR820TR von Allied Telesyn International) verbunden. Die Workstations waren jeweils über ein CrossOver-Kabel an das entsprechende Gateway angeschlossen.

3.1.2 Software

Auf beiden Gateways wurde ein *S.u.S.E. Linux 6.0*[20] Basissystem installiert. Auf den anderen Rechner lief das Betriebssystem *Microsoft Windows NT 4.0 Workstation* (Build 1381, ServicePack 3). Für die Performancetests (siehe Abschnitt 3.3) wurde das Benchmark-Tool *Netperf*[21] auf beiden Workstations eingesetzt. Für die Angriffstests kamen das Netzwerk-Monitoring-Tool *NetXRay* (Dual, Intern. Version 3.0.2) unter Windows sowie die von Diskette bootbare Linux-Distribution *Trinux*[22] zum Einsatz.

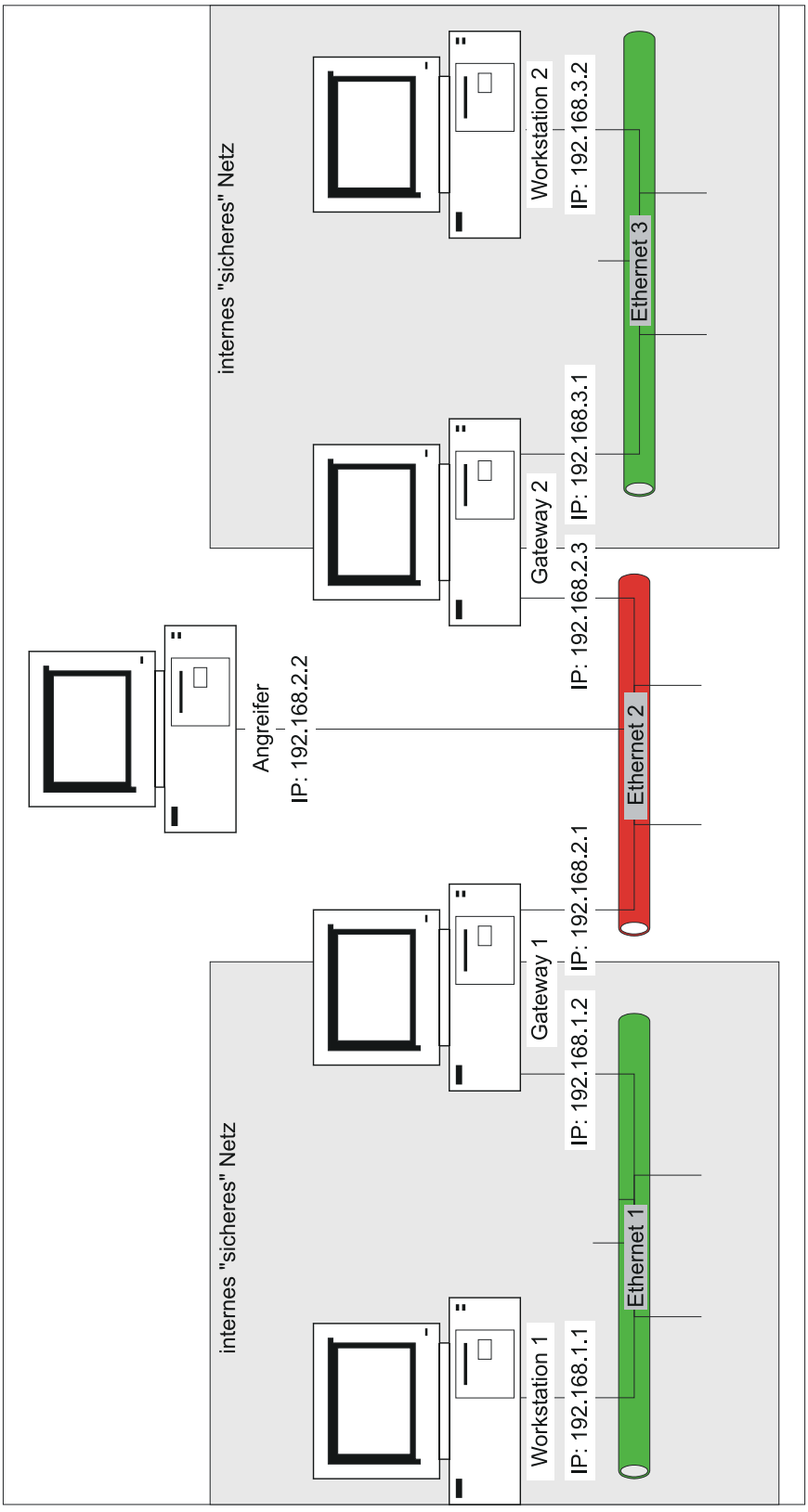

Abbildung 1: Versuchsanordnung

3.2 Installation und Konfiguration

3.2.1 Basissystem

Auf beiden Gateways wurde nach der Installation des Linux-Basissystems der Quellcode des in der aktuellen FreeS/WAN Distribution empfohlenen aktuellen Kernelreleases der 2.0er Reihe (2.0.38) im Verzeichnis `/usr/src/` entpackt und ein entsprechender symbolischer Link `/usr/src/linux` auf das neue Verzeichnis erstellt. Danach wurde der neue Kernel konfiguriert, kompiliert, installiert und getestet (was notwendig ist, da die anschließende Installations-Prozedur von FreeS/WAN auf die Ergebnisse der Kernelkompilierung zurückgreift). Anschließend wurde das Netzwerk (Interfaces und Routing) entsprechend des Szenarios konfiguriert, so daß eine Kommunikation zwischen allen Rechnern möglich war.

3.2.2 Installation von FreeS/WAN

Nach der Installation der Basissysteme wurde auf beiden Gateways die aktuelle FreeS/WAN Distribution (`freeswan-1.2.tar.gz`)[23] im Verzeichnis `/usr/src` entpackt und der ebenfalls auf der Distributions-Seite erhältliche Patch (`freeswan-1.2.patch1.gz`) wie in der Installationsanleitung (Textdatei `INSTALL` im Basisverzeichnis der Distribution) angegeben angewandt.

Danach wurde im FreeS/WAN-Hauptverzeichnis das Kommando `make menugo` aufgerufen, welches die Kernelkonfiguration (analog zu `make menuconfig`) mit den um die IPsec-Features erweiterten Netzwerk-Optionen startet. Neben den neuen IPsec-Parametern werden dabei auch andere Netzwerk-Optionen automatisch auf sinnvolle Default-Werte gesetzt, die bei unserer Konfiguration so beibehalten wurden. Insbesondere die Optionen „IP: forwarding/gatewaying“ und „Kernel/ User network link driver“ dürfen hier nicht deaktiviert werden. Optional kann jedoch noch der Parameter „IP: optimize as router not host“ aktiviert werden.

Nach dem Verlassen der Kernelkonfiguration wurde nun automatisch die Kompilierung des neuen Kernels und der FreeS/WAN-Quellen eingeleitet. Nachdem auch die Kompilierung der Module abgeschlossen war, wurden der neue Kernel und die Module mittels `make install` und

`make modules_install` installiert und beide Gateways neu gestartet.

Die Installations-Prozedur von FreeS/WAN hatte automatisch im Verzeichnis `/sbin/init.d/` ein Startup-Skript für das IPsec-Subsystem installiert und eine entsprechende RunLevel-Konfiguration vorgenommen, so daß dieses beim Neustart der Rechner automatisch geladen wurde.

3.2.3 Konfiguration von FreeS/WAN

Die RunTime-Konfiguration von Linux FreeS/WAN befindet sich in der Datei `/etc/ipsec.conf`. Für unseren Test wurden nur wenige Änderungen an der vorinstallierten Beispielkonfiguration vorgenommen:

- Mittels `interfaces="ipsec0=eth1"` wurde die IPsec-Verschlüsselung an das Netzwerk-Interface (Netzwerkkarte) gebunden, über welches die Gateways jeweils mit dem „öffentlichen Netz“ (Ethernet 2) verbunden waren.
- Die Parameter in der „Beispiel-Verbindung“ (`sample connection`) wurden auf folgende Werte gesetzt:
  - `left=192.168.2.1`
  - `leftsubnet=192.168.1.0/24`
  - `right=192.168.2.3`
  - `rightsubnet=192.168.3.0/24`
  - `leftnexthop` und `rightnexthop` wurden nicht gesetzt, da sich beide Gateways in einem Subnetz befanden und die Angabe dieser Parameter in diesem Fall laut Dokumentation nicht nötig ist.

Nachdem die Konfiguration auf beiden Gateays angepaßt worden war, wurde in der Datei `/etc/ipsec.secrets` noch ein gemeinsames *Preshared Secret* in Form einer ASCII-Zeichenkette für die Authentifizierung während der Schlüsselvereinbarung eingetragen.

Nach einem Neustart der IPsec-Komponenten auf beiden Gateways mittels des Kommandos `/sbin/init.d/ipsec restart` war der VPN-Tunnel aktiv und die beiden Workstations konnten über diesen miteinander

	Übertragungszeit (Sekunden)	***Durchsatz***
Set 1 (große Dateien)		
ohne Verschlüsselung	4	768,0 kb/s
mit Verschlüsselung	5	614,4 kb/s
Set 2 (kleine Dateien)		
ohne Verschlüsselung	212	14,5 kb/s
mit Verschlüsselung	213	14,4 kb/s

Tabelle 1: Ergebnisse der Dateiübertragungstests

kommunizieren. Das Mitschneiden des Netzverkehrs mittels *NetXRay* auf dem Angriffsrechner zeigte, daß die gesamte Kommunikation zwischen den beiden Workstations im „öffentlichen Netz“ nun verschlüsselt war.

3.3 Performance-Tests

Um die Auswirkungen des VPN-Einsatzes auf die Performance von Netzwerkübertragungen zu prüfen, wurden verschiedene Tests durchgeführt. Um Dateiübertragungstests zu ermöglichen, wurde auf Workstation 2 (siehe Abb. 1) ein FTP-Server installiert (*Microsoft Internet Information Server 2.0*). Desweiteren wurden TCP-Streaming-Tests und TCP-Request-Response-Tests mit dem Tool *Netperf* durchgeführt, wozu auf Workstation 2 die dazugehörige Server-Komponente installiert wurde.

3.3.1 Dateiübertragungstests

Es wurden zwei Dateisets zusammengestellt: Set 1 bestand aus 3 Dateien mit einem Gesamtvolumen von 3 MB. In Set 2 befanden sich 1059 Dateien, deren Größe zwischen 1 KB und 5 KB lag und die zusammen ebenfalls ein Volumen von 3 MB aufwiesen. Eine Gegenüberstellung der durchschnittlich gemessenenen Übertragungszeiten ohne und mit Verschlüsselung (in der FreeS/WAN-Standardkonfiguration) zeigt Tabelle 1.

Die Verschlüsselung innerhalb des VPN führt (naturgemäß) zu einer Senkung der maximalen Übertragungsrate. Dies ist erkennbar an dem Performance-Verlust bei der Übertragung großer Dateien. Bei den kleinen

Dateien bewegte sich der Verlust an Performance in der Größenordnung von Meßfehlern. Hier ist der Aufwand auf FTP-Server- und -Client-Seite für das Handling der über 1000 Dateien (Dateianforderung, -bestätigung und -speicherung jeder einzelnen Datei) so groß, daß die verfügbare Netzbandbreite bei weitem nicht ausgenutzt wird.

3.3.2 Netperf-Tests

Mit *Netperf* wurden TCP-Stream- und Request-Response-Tests durchgeführt. Die Stream-Tests untersuchen die maximal erreichbare Übertragungsrate, während mit den Request-Response-Tests festgestellt wird, wieviele Transaktionen pro Sekunde durchgeführt werden können (interaktiver Netzverkehr).

Um den Einfluß verschiedener, in der Konfigurationsdatei `/etc/ipsec.conf` angegebenen Parameter auf die Netzperformance zu überprüfen, wurden alle Tests zunächst mit deaktiviertem VPN und anschließend mit der FreeS/WAN-Standardkonfiguration durchgeführt. Danach wurden, jeweils ausgehend von der Standardkonfiguration, folgende Parameter verändert:

- **esp:** Verschlüsselungs- bzw. Authentifizierungsalgorithmus bei Verwendung des ESP-Headers.
- **ah:** Authentifizierungsalgorithmus bei Verwendung des AH-Headers.
- **keylife:** Gültigkeitsdauer der Security-Associations (SA) und damit auch der Schlüssel.
- **rekeymargin:** Zeit vor dem Ablauf der Schlüsselgültigkeitsdauer, in der das Aushandeln neuer Schlüssel beginnen soll.
- **rekeyfuzz:** maximaler Prozentsatz zur zufälligen Vergrößerung des Wertes *rekeymargin*, wichtig zur Lastverteilung bei VPN-Gateways mit vielen VPN-Verbindungen.

veränderter Konfigurations-parameter	*relative Performance*	*durchschnittlich gemessene Datenrate (MBit/s)*	*minimal gemessene Datenrate (MBit/s)*	*maximal gemessene Datenrate (MBit/s)*
keiner (VPN deaktiviert)	100,0%	9,09	9,07	9,10

keiner (Standardkonfiguration)
`esp=3des-md5-96`
`(ah deaktiviert)`
`keylife=8.0h`
`rekeymargin=9m`
`rekeyfuzz=100%`
	72,8%	6,62	6,59	6,65

`esp=3des`	72,9%	6,63	6,55	6,66
`esp=3des-sha1-96`	73,2%	6,65	6,63	6,67
`ah=hmac-md5-96`	72,6%	6,60	6,57	6,63

`keylife=2s`
`rekeymargin=1s`
`rekeyfuzz=0%`
	63,8%	5,80	5,57	6,16

`keylife=10s`
`rekeymargin=3s`
`rekeyfuzz=100%`
	72,9%	6,63	6,60	6,67

Tabelle 2: Ergebnisse der TCP-Stream-Tests

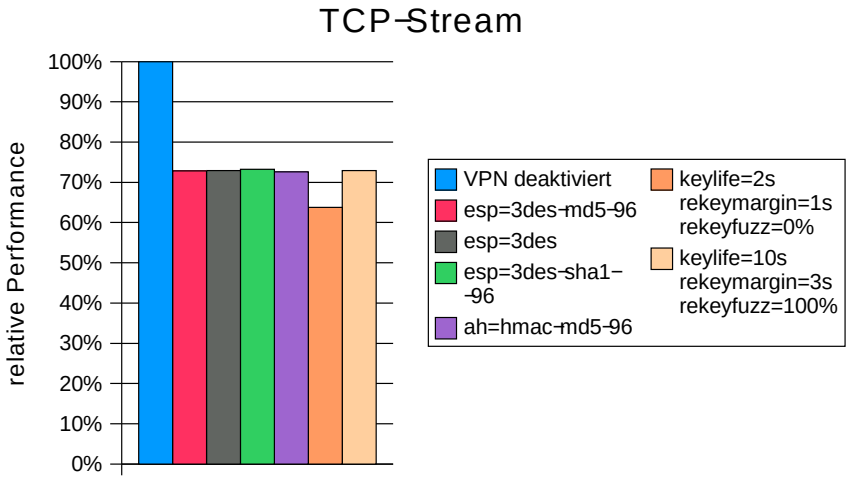

Abbildung 2: Ergebnisse der TCP-Stream-Tests

Ergebnisse der TCP-Stream-Tests

Aus den in Tabelle 2 und Abbildung 2 dargestellten Ergebnissen lassen sich insbesondere folgende Aussagen bzgl. der maximalen Datenrate ablesen:

- Die Auswahl eines zur Authentifizierung benutzten Hash-Algorithmus innerhalb des ESP-Headers hat praktisch keinen Einfluß auf die erreichbare Datenrate. Die Unterschiede liegen in der Größenordnung von Meßfehlern.
- Die zusätzliche Verwendung des AH-Headers mindert die in unserer Testumgebung maximal erreichbare Datenrate nur unwesentlich.
- Ein wesentlicher Performance-Rückgang tritt lediglich bei *extrem* kurzen Schlüssel-Lebensdauern auf. Bei der in der Tabelle gezeigten Konfiguration (`keylife=2s`) liegt dies insbesondere daran, daß in diesem speziellen Fall bei jedem Schlüsselwechsel eine kurze Unterbrechung des VPN auftrat, da kein neuer Schlüssel vereinbart werden konnte, bevor der alte Schlüssel seine Gültigkeit verlor.
- Schon durch die Wahl einer etwas längeren Gültigkeitsdauer für die verwendeten Schlüssel (z. B. `keylife=10s`) konnte der o. g. Performance-Rückgang vermieden werden.

Ergebnisse der Request-Response-Tests

Die in Tabelle 3 und Abbildung 3 dargestellten Ergebnisse der Request-Response-Tests bestätigen die zuvor getroffenen Aussagen. Erwartungsgemäß hat der Einsatz des FreeS/WAN-VPN größeren Einfluß auf die relative Request-Response-Performance als auf die relative TCP-Stream-Performance. Ursache dafür sind die durch die Verschlüsselung bedingten erhöhten Latenzzeiten bei der Paketübertragung.

3.3.3 Weitere Informationen zur Performance

Weitere Aussagen zur Performance von Linux FreeS/WAN sind bei [24] zu finden.

veränderter Konfigurations-parameter	*relative Performance*	*durchschnittlich gemessene Transaktionsrate (Trans./s)*	*minimal gemessene Transaktionsrate (Trans./s)*	*maximal gemessene Transaktionsrate (Trans./s)*
keiner (VPN deaktiviert)	100,0%	1140,9	1134,2	1146,6
keiner (Standardkonfiguration)	65,1%	743,2	737,1	746,8
`esp=3des-md5-96 (ah deaktiviert) keylife=8.0h rekeymargin=9m rekeyfuzz=100%`				
`esp=3des`	64,6%	737,4	735,5	738,9
`esp=3des-sha1-96`	65,1%	743,0	743,1	744,0
`ah=hmac-md5-96`	64,6%	736,7	732,0	739,2
`keylife=2s rekeymargin=1s rekeyfuzz=0%`	54,7%	624,3	612,1	642,3
`keylife=10s rekeymargin=3s rekeyfuzz=100%`	62,6%	714,0	695,8	737,8

Tabelle 3: Ergebnisse der Request-Response-Tests

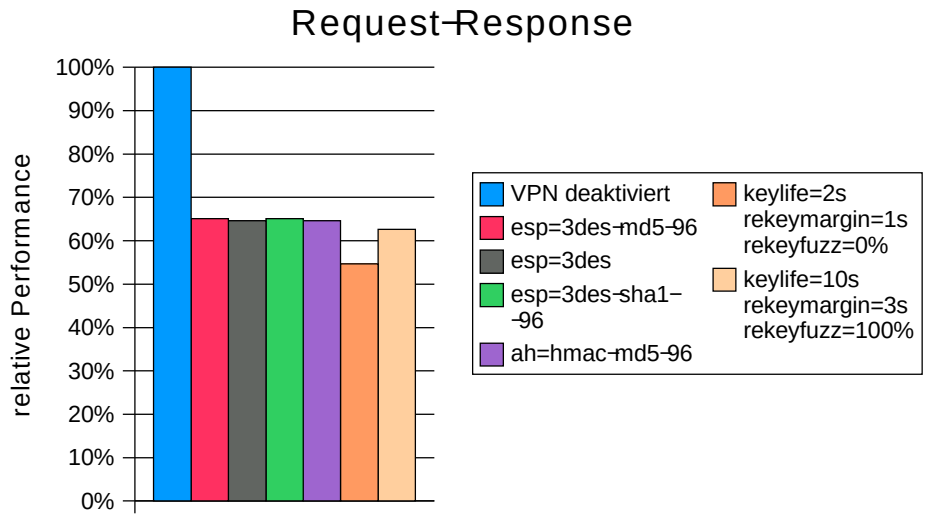

Abbildung 3: Ergebnisse der Request-Response-Tests

3.4 Angriffs-Tests

3.4.1 Replay-Angriff

Mittels des Netzwerk-Sniffers *NetXRay* wurden Pakete aus dem Datenverkehr zwischen den beiden Gateways aufgezeichnet. Anschließend wurden diese Pakete immer wieder in den Netzverkehr eingefügt. Dieser Angriffsversuch führte jedoch, abgesehen von der teilweisen Belegung der Netzbandbreite, zu keiner weiteren Beeinträchtigung der Kommunikation.

Auch Replay-Attacken mit Paketen aus der Phase des IPsec-Verbindungsaufbaus oder des Schlüsselwechsels waren nicht erfolgreich. Auffällig war lediglich die bei diesem Angriff festzustellende Aktivität der Festplatten, deren Ursache darin lag, daß von FreeS/WAN in der Datei `/var/log/messages` der Empfang doppelter (also alter) Pakete vermerkt wurde.

3.4.2 Angriff mit manipulierten Paketen

FreeS/WAN benutzt für die Schlüsselvereinbarung den UDP-Port 500. Da UDP im Gegensatz zu TCP ein nicht-verbindungsorientiertes Protokoll ist, konnte kein Angriff durch Pakete mit gesetztem SYN-Bit erfolgen. Das Senden von aufgezeichneten UDP-Paketen aus der Schlüsselwechselphase (vgl. Replay-Angriff) an Port 500 führte zu keiner Beeinträchtigung des VPN und auch mit manipulierten Paketen konnte kein erfolgreicher Angriff durchgeführt werden.

3.4.3 Weiterleitung unverschlüsselter Pakete

Dieses unter Umständen kritische Problem der aktuellen FreeS/WAN-Version ist in der in der Distribution enthaltenen Datei `BUGS` beschrieben.

Empfängt Gateway 2 (vgl. Abbildung 1) ein *unverschlüsseltes* Paket, das als Absenderadresse die IP-Adresse von Workstation 1 enthält und das an Workstation 2 gerichtet ist, so wird dieses Paket weitergeleitet. Da sich die Absenderadresse eines Paketes mit entsprechenden Tools beliebig fälschen läßt, wäre durch Ausnutzung dieses Fehlers z. B. ein Denial-Of-Service-Angriff denkbar. Der Angreifer muß dazu jedoch die IP-Adresse des anzugreifenden Rechners kennen und von „außen“ Pakete an diese Adresse

schicken können.

Die FreeS/WAN-Entwickler stufen dieses Problem als schwerwiegend ein und planen daher auch, zukünftige Versionen gegen Angriffe dieser Art abzusichern.

4 Fazit

Linux FreeS/WAN machte insgesamt gesehen einen sehr guten Eindruck. Der Vergleich mit den anderen im Rahmen des Produkttests geprüften VPN-Implementierungen zeigte, daß sich Linux FreeS/WAN als *freie* Software keinesfalls hinter kommerziellen Produkten verstecken muß.

Der Vorteil des Einsatzes *freier* Software liegt auch im Bereich von VPN-Produkten auf der Hand: Der Quellcode ist für jedermann einsehbar und somit z. B. auch auf Schwachstellen hin prüfbar. Außerdem werden bekannte Schwachstellen der Implementierung in der Dokumentation veröffentlicht und nicht ggf. vom Hersteller geheimgehalten.

Einer der wesentlichen Schwachpunkte der aktuellen Version von Linux FreeS/WAN ist die fehlende Unterstützung für Industriestandard-Zertifikate. Außerdem müssen die *Preshared Secrets* in Form von ASCII-Zeichenketten bzw. die RSA Public-Keys im Vorfeld über einen vertrauenswürdigen Kanal „von Hand“ ausgetauscht werden. Von einer komfortablen Schlüsselverteilung im Sinne einer (lokalen) PKI kann hier also keine Rede sein.

Literatur

[1] John Gilmores Homepage,
`http://www.cygnus.com/~gnu/`

[2] RSA Labs FAQ - What is S/WAN,
`http://www.rsasecurity.com/rsalabs/faq/5-1-3.html`

[3] FreeS/WAN Project Home Page,
`http://www.freeswan.org/`

[4] RSA Data Security,
`http://www.rsasecurity.com/`

[5] Linux - A Free Unix-Type Operating System,
`http://www.linux.org/`

[6] The Internet Engineering Task Force (IETF),
`http://www.ietf.org/`

[7] IP Security Protocol (IPsec) Charter,
`http://www.ietf.cnri.reston.va.us/html.charters/ipsec-charter.html`

[8] Labouret, Ghislaine: „IPsec: a technical overview“,
`http://www.hsc.fr/ressources/veille/ipsec/papier/papier.html.en`

[9] IP Next Generation (IPng, IPv6),
`http://playground.sun.com/pub/ipng/html/ipng-main.html`

[10] FreeS/WAN Dokumentation, Kapitel „Cryptography Export Restrictions“

[11] Schmidt, Beate et al.: „Virtual Private Networks - Das Internet als Corporate Network“, Connector Ausgabe 15, Heidelberg, August 1998.

[12] Virtual Private Network Consortium,
http://www.vpnc.org

[13] Kent, S. et al.: „IP Authentication Header“, Request For Comments (RFC) 2402, November 1998

[14] Kent, S. et al.: „IP Encapsulation Header“, Request For Comments (RFC) 2406, November 1998

[15] Harkins, D. et al.: „Internet Key Exchange“, Request For Comments (RFC) 2409, November 1998

[16] Glenn, R. et al.: „The NULL Encryption Algorithm and Its Use With IPsec“, Request For Comments (RFC) 2410, November 1998

[17] Madson, C. et al.: „The Use of HMAC-MD5-96 within ESP and AH“, Request For Comments (RFC) 2403, November 1998

[18] Madson, C. et al.: „The Use of HMAC-SHA-1-96 within ESP and AH“, Request For Comments (RFC) 2404, November 1998

[19] Fachhochschule Rhein-Sieg,
http://www.fh-rhein-sieg.de/

[20] S.u.S.E., Gesellschaft für Software- und Systementwicklung mbH,
http://www.suse.de/

[21] Netperf,
http://www.netperf.org/

[22] Trinux: A Linux Security Toolkit,
http://www.trinux.org/

[23] ftp://ftp.xs4all.nl/pub/crypto/freeswan/

[24] Performance of the FreeS/WAN IPSEC Implementation,
http://tsc.llwybr.org.uk/public/reports/SWANTIME/