VPN-Laborübung Einführung

Diese Seite lässt sich als PDF[1)] und als OpenOffice-Dokument[2)] herunterladen. Nutzen Sie das OpenOffice-Dokument, um Ihre Ergebnisse festzuhalten.

Allgemeine Hinweise für die Durchführung unter Linux

Im Meta-Paket `ipsec-tools` befinden sich die notwendigen Programme, um die SAD[3)] und SPD[4)] zu konfigurieren. Das entscheidende Programm lautet **`setkey`** und kann mittels einer Konfigurationsdatei beeinflusst werden. Für beide Seiten wird eine eigene Konfigurationsdatei benötigt. Zunächst werden mittels der beiden flush-Befehle die SAD und SPD gelöscht, damit es keine Beeinflussung mit alten Einträgen gibt. Im Anschluss werden zunächst die SAs für beide Richtungen angelegt und zum Schluss die Policies für die entsprechenden Richtungen (hier sind Source und Destination sowie die Richtung von Bedeutung).

1. Aufgabe: IPsec im manuell konfigurierten Transport-Modus (End-to-End)

Das Szenario umfasst ein vereinfachtes Modell, bei dem das öffentliche Internet über einen Router (eigener PC) repräsentiert wird. Das folgende Schaubild zeigt den prinzipiellen Aufbau. Die beiden Kommunikationspartner (Amy und Berny) sollen die VPN-Verbindung aufbauen. Als IPsec Protokoll soll AH im Transport-Modus verwendet werden. Nach erfolgreichem Aufbau soll auf dem Router die Kommunikation zwischen den beiden Endstationen mit geschnitten werden oder über einen weiteren Hub.

hilfreiche Hinweise und Beispiele

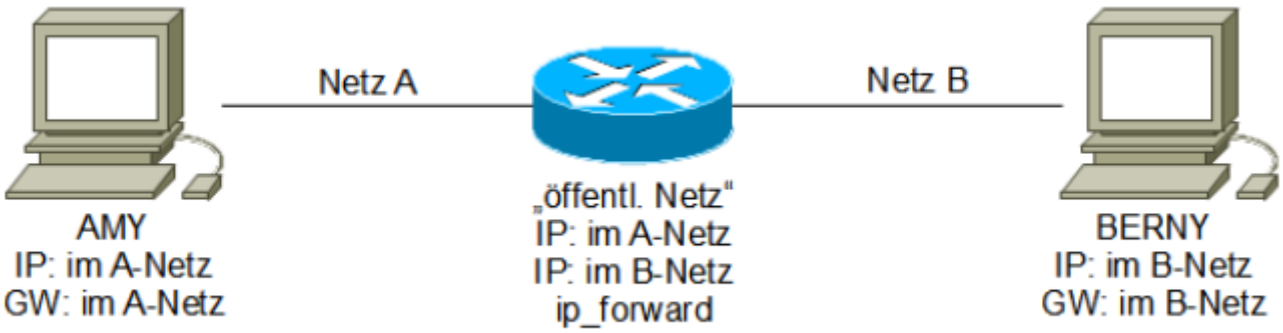


Abbildung 1: VPN: End-to-End im Transport-Modus (manuell konfiguriert)

2. Aufgabe: IPsec im manuell konfigurierten Tunnel-Modus (End-to-End)

Der Aufbau selbst bleibt unverändert (s. Abbildung in Aufgabe 1). Allerdings wird nun zusätzlich die

Verschlüsselung und der Tunnel-Modus per ESP aktiviert. Aktivieren Sie in dieser Aufgabe ESP sowie den Tunnel-Modus und protokollieren Sie wieder auf dem Router. Zeichnen Sie erneut auf und notieren Sie die Änderung zu Aufgabe 1.

hilfreiche Hinweise und Beispiele

3. Aufgabe: IPsec mit automatisch ausgehandelten Schlüsseln

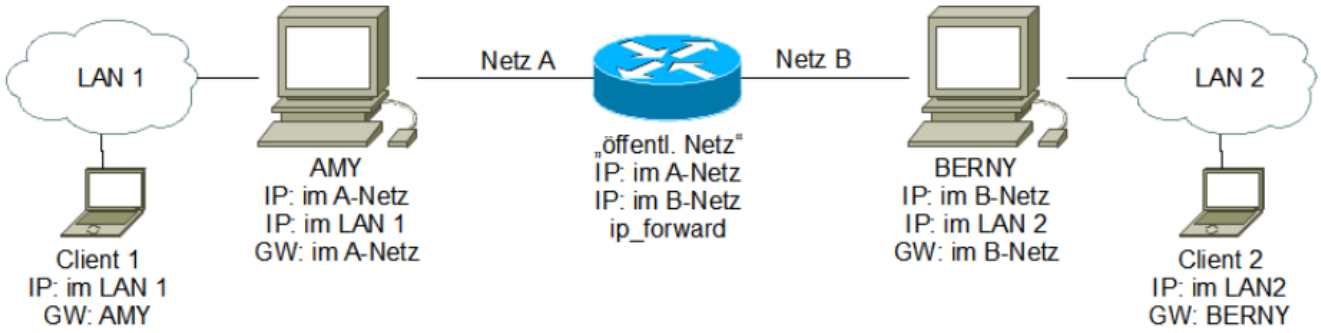


Abbildung 2: VPN: Site-to-Site im Tunnel-Modus (automatisch konfiguriert)

Für die automatische Aushandlung der Schlüssel können unterschiedliche Produkte zum Einsatz kommen. Allen gemeinsam ist das `IKE` bzw. das darin verwendete `ISAKMP`. In dieser Laborübung wird racoon eingesetzt. StrongSWAN oder OpenSWAN 5 sind Alternativen hierzu. Weiterhin soll in dieser Aufgabe eine Site-to-Site-Verbindung zwischen zwei LANs über zwei VPN-Gateways aufgebaut werden. Protokollieren Sie auf der Strecke zwischen **Amy** und **Berny** (auf dem Router) die `ISAKMP`-Nachrichten sowie die Nachrichten, die zwischen Client 1 und Client 2 transportiert werden.

hilfreiche Hinweise und Beispiele

Gesamte Aufgabenstellung für die Laborübung in PDF-Form: VPN IPsec Laborübung

Die Grundlagen zu VPN werden auf dieser Seite zusammengefasst.

Hinweise und Howtos

Da Sie keinen direkten Zugriff auf den Ordner `/etc` haben, müssen Sie die ipsec-tools mit entsprechenden Parametern aufrufen, damit Sie Ihre Konfiguration nutzen können. IPsec-Tools-Aufruf-Parameter

---

Manuell verschlüsselte Verbindungen setkey

Mittels `setkey` lassen sich manuell verschlüsselte Verbindungen konfigurieren. `setkey` selbst baut dabei nicht die Verbindung auf, sondern dient lediglich zur Verwaltung der Verbindungen, die in den unterschiedlichen Datenbanken (s.u. Tabelle) hinterlegt werden. Die Verbindungen werden bei

Bedarf, also sobald ein Paket die Kriterien erfüllt, durch den IP-Protokollstack aufgebaut.

Folgende Begriffe werden dabei verwendet:

Begriff	Bedeutung	Kommentar
SA	Security Association	eigentliche Verbindung zwischen zwei Endpunkten
SPD	Security Policy Database	Datenbank in der die SA-Konfigurationen verwaltet werden
SAD	Security Associations Database	Datenbank in der die Zustände [5)] der SAs verwaltet werden

Hier ist ein englisches Howto zu finden, in dem die einzelnen Konfigurationsschritte erklärt sind:
Ubuntu Howto zu IPsec

Erklärung der notwendigen Schritte für manuelle Verbindung

1. Anlegen der beiden `setkey`-Konfigurationen (s. setkey-Konfiguration)
2. Routing im Router aktivieren (s. Router-Konfiguration)
3. Gateway auf den Endteilnehmern eintragen
4. Starten der Verbindung (s. setkey-Aufrufparameter)
5. Überprüfen der Verbindung mittels `ping` und Mitschnitt mit `wireshark` an geeigneter Stelle.

setkey-Konfiguration

Erklärung der Abschnitt einer `setkey`-Konfigurationsdatei anhand eines Beispiels:

amy.conf

```
# WICHTIG: Jede Zeile muss mit einem Semikolon ";" abschliessen.
# Beispieldatei fuer AMY
# Alles Loeschen SAD und SPD
flush;
spdflush;

# Amy 3.0.0.3
# Berny 2.0.0.2

# Beispiel: manuelle Parameter fuer AH-SAs
# syntax: add src dst proto spi -A authalgo key;
add 3.0.0.3 2.0.0.2 ah 700 -A hmac-md5
0xbf9a081e7ebdd4fa824c822ed94f5226;
add 2.0.0.2 3.0.0.3 ah 800 -A hmac-md5
0xbf9a081e7ebdd4fa824c822ed94f5226;

# Beispiel: manuelle Parameter fuer ESP-SAs
# syntax: add src dst proto spi -E encalgo key;
add 3.0.0.3 2.0.0.2 esp 701 -E 3des-cbc
0x3f0b868ad03e68acc6e4e4644ac8bb80ecea3426d3d30ada;
add 2.0.0.2 3.0.0.3 esp 801 -E 3des-cbc
0x3f0b868ad03e68acc6e4e4644ac8bb80ecea3426d3d30ada;
```

```
# Richtlinien (Policies) fuer SAs anlegen
# ACHTUNG: die Richtung (in/out) ist wichtig, diese Datei kann nur fuer
EINE Seite verwendet werden!!
# syntax: spdadd src-range dst-range upperspec policy;

# Einrichten der Policy für ESP UND AH
spdadd 2.0.0.2 3.0.0.3 any -P in ipsec esp/transport//require
ah/transport//require;
spdadd 3.0.0.3 2.0.0.2 any -P out ipsec esp/transport//require
ah/transport//require;

# Einrichten von AH OHNE ESP
#spdadd 2.0.0.2 3.0.0.3 any -P in ipsec ah/transport//require;
#spdadd 3.0.0.3 2.0.0.2 any -P out ipsec ah/transport//require;
```

setkey-SAs

Der Befehl `add` fügt eine neue SA in die SAD ein. Es ist darauf zu achten, dass je eine SA pro Richtung konfiguriert werden muss. D.h. für eine vollständige Verbindung (beide Richtungen) werden ZWEI SAs benötigt. Diese beiden SAs sind für beide Partner identisch. Im obigen Beispiel wurden zwei SAs für AH (SPI: 700 und 800) und zwei SAs für ESP (SPI: 701 und 801) konfiguriert. Pro Richtung können unterschiedliche Schlüssel verwendet werden.

setkey-Policies

Der Befehl `spdadd` für das Einrichten einer Policy ist etwas komplexer, daher hier eine kurze Erklärung der einzelnen Parameter. Die Policy legt die Regeln fest, wann IPsec anzuwenden ist und mit welchen Protokollen.

syntax: `spdadd src-range dst-range upperspec policy;`

Parameter	**Erklärung**
src-range	Quell-IP-Adressbereich
dst-range	Ziel-IP-Adressbereich
upperspec	mögliche Werte für die Richtung [`in` / `out`]
	mögliche Werte für die Behandlung [`ipsec` / `discard` / `none`]
policy	Aufbau: (`protocol/mode/src_dst/level`)
	mögliche Werte für `protocol` [`ah / esp`]
	mögliche Werte für `mode` [`transport / tunnel`]
	für `src_dst` kann die Angabe entfallen, wenn dieselben Werte wie fuer SAs verwendet werden.
	mögliche Werte für `level` [`use / require`]; `use`=Verschlüsselung erwünscht; `require`=Verschlüsselung zwingend erforderlich

ACHTUNG: In einer `policy` kann sowohl AH, als auch ESP angegeben werden. Die Reihenfolge ist vorgeschrieben: erst ESP dann AH!

setkey-Aufrufparameter

Der Befehl `setkey` benötigt administrative Rechte, daher muss `sudo` vorangestellt werden.

Parameter	Steht für	Erklärung
`-D`	Dump	Ausgabe der SAD
`-DP`	Dump Policys	Ausgabe der SPD
`-F`	Flush	Löschen der SAD
`-FP`	Flush Policys	Löschen der SPD
`-f CONFIGFILE`	File	Angabe der Konfigurationsdatei; wird der Parameter weggelassen so wird `/etc/ipsec-tools.conf` verwendet

Beispielaufruf zum starten einer Verbindung:

```
setkey -f amy_ah.conf
```

Automatisch verschlüsselte Verbindungen mittels racoon

Das statische Anlegen der SAs (vgl. `add`-Befehl unter setkey-SAs) ist bei einer automatische Aushandlung überflüssig, da die notwendigen SAs über IKE[6] automatisch ausgehandelt werden. Das Tool `racoon` startet auf Anfrage eine IPsec-Verbindung mittels IKE. `racoon` benötigt zunächst eine `racoon`-eigene Konfigurationsdatei in der die Eckdaten für die **„Phase 1: IKE-SAs“** [7] als `proposal` angelegt und im zweiten Teil die eigentlichen **„Phase 2: Nutzdaten-SAs“** [8]. Da `racoon` nicht selbstständig den VPN-Tunnel etabliert, müssen noch die entsprechende Policies angelegt werden (s. setkey-Policies einrichten). Dies geschieht wie bereits bei dem statischen Aufbau mittels `setkey`.

ACHTUNG: `racoon` aktzeptiert nur sichere psk-Dateien, die bedeutet nur `root` darf diese Datei lesen/schreiben und `root` muss der Owner sein! Da `keinpasswort` keine Rechte hat eine solche Datei zu erstellen, muss für die Gateway-Rechner auf eine VMs ausgewichen werden. Dort hat der User die nötigen Rechte.

Erklärung der notwendigen Schritte für automatisch aufgebauter Verbindungen

1. Anlegen der beiden `setkey`-Konfigurationen (NUR SPD!)
2. Anlegen der beiden `racoon`-Konfigurationen (s. racoon-Konfiguration)
3. Routing im Router aktivieren (s. Router-Konfiguration)
4. Gateway auf den Endteilnehmern eintragen
5. Starten der Verbindung (s. racoon-Aufrufparameter)
6. Überprüfen der Verdingung mittels `ping` und Mitschnitt mit `wireshark` an geeigneter Stelle.

racoon-Konfiguration

Erklärung der Parameter anhand einer `racoon`-Beispiel-Konfiguration.

amy_ike.conf

```
# racoon
# local: amy 3.0.0.3  Netz: 10.0.0.0/8
# remote: berny 2.0.0.2 Netz: 20.0.0.0/8

# Pfad zur PSK-Datei
path pre_shared_key "/home/USER/vpn/ike/psk.txt";

# Proposal für Berny(2.0.0.2) als Gegenstelle einrichten
remote 2.0.0.2 {
    exchange_mode main;
    proposal {
        encryption_algorithm aes;
        hash_algorithm sha256;
        authentication_method pre_shared_key;
        dh_group modp1024;
    }
}

# Eigentliche SA für Nutzdatenverbindung Remote: 20.0.0.0/8 Local:
10.0.0.0/8
sainfo address 10.0.0.0/8 any address 20.0.0.0/8 any {
    pfs_group modp1536;
    encryption_algorithm aes 256;
    authentication_algorithm hmac_sha256;
    compression_algorithm deflate;
}
```

Die PSK-Datei enthält den gemeinsamen Key für die Verbindung. Es muss jeweils die IP-Adresse der Gegenstelle eingetragen werden. Beispiel einer psk.txt:

psk.txt

```
2.0.0.2     gemeinsamerkey
```

Diese Datei muss mit den folgenden Befehlen für `racoon` vorbereitet werden.

```
chown root psk.txt
chgrp root psk.txt
chmode 0600 psk.txt
```

Damit `racoon` vom Linux-Kernel angestoßen werden kann, werden noch entsprechende Policies benötigt. Diese können über `setkey` eingerichtet werden. Die einzelnen Parameter sind hier erklärt. Hier eine Beispiel-Datei

amy_SPD

```
# WICHTIG: Jede Zeile muss mit einem Semikolon ";" abschliessen.
```

```
# Datei fuer AMY
# Alles Loeschen SAD und SPD
flush;
spdflush;

# amy: 3.0.0.3 (Netz: 10.0.0.0/8)
# berny: 2.0.0.2 (Netz: 20.0.0.0/8)

# Richtlinien (Policies) fuer SAs anlegen
# ACHTUNG: die Richtung (in/out) ist wichtig, diese Datei kann nur fuer
EINE Seite verwendet werden!!
# syntax: spdadd src-range dst-range upperspec policy;
spdadd 20.0.0.0/8 10.0.0.0/8 any -P in ipsec
esp/tunnel/2.0.0.2-3.0.0.3/require;
spdadd 10.0.0.0/8 20.0.0.0/8 any -P out ipsec
esp/tunnel/3.0.0.3-2.0.0.2/require;
```

racoon-Aufrufparameter

Der Befehl `racoon` benötigt administrative Rechte, daher muss entweder `sudo` vorangestellt werden.

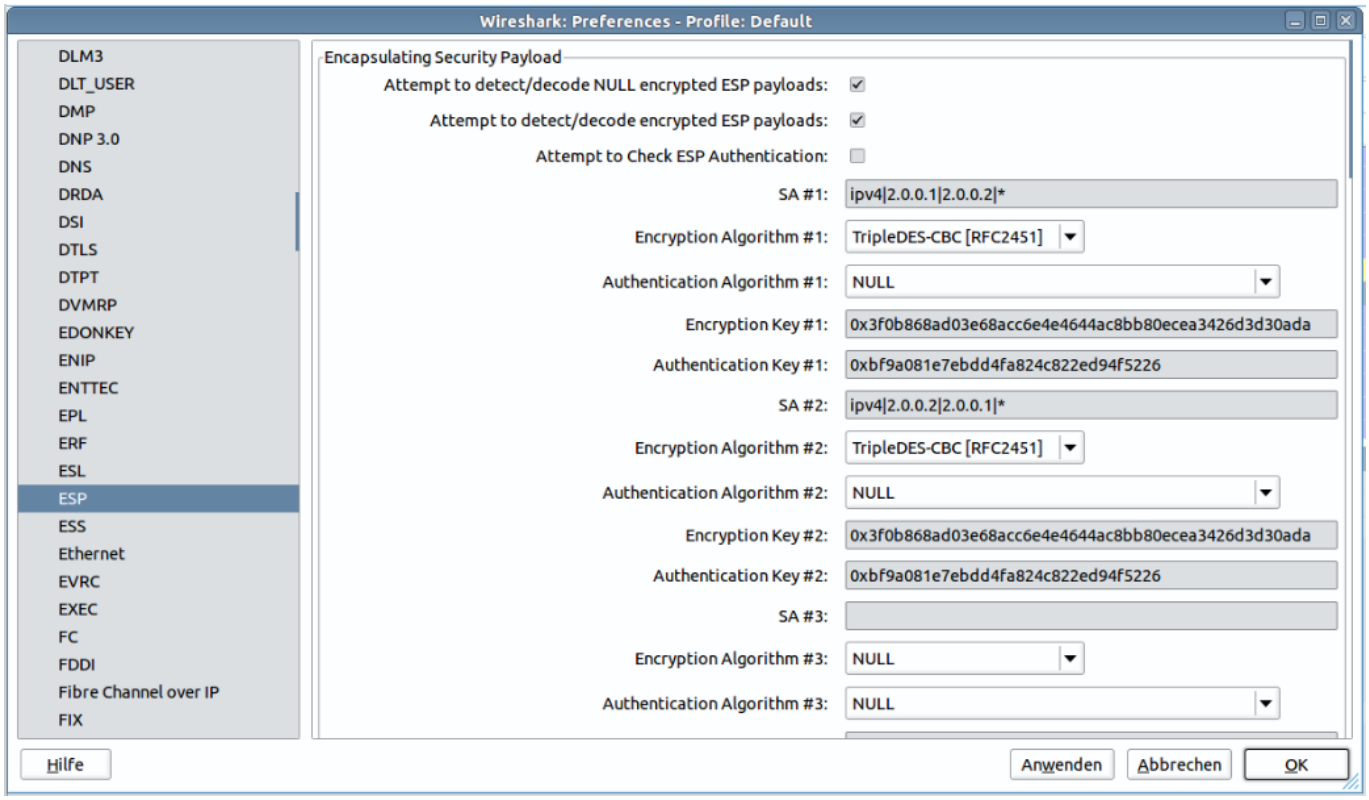
Parameter	Steht für	Erklärung
`-f CONFIGFILE`	File	Angabe des Speicherort der Konfigurationsdatei `CONFIGFILE`; wird der Parameter weggelassen so wird `/etc/racoon/racoon.conf` verwendet
`-F`	Foreground	Alle Ausgaben werden im Terminal ausgegeben. Sehr gut für Testzwecke
`-l LOGFILE`	Logfile	

Beispielaufruf zum starten einer Verbindung:

```
racoon-f amy_ike.conf -F -l racoon.log
```

Entschlüsselung einer ESP-Verbindung

Die folgende Abbildung zeigt die notwendigen Einstellungen zur Entschlüsselung einer ESP gesicherten Verbindung.



Beispiel-Capture für Wireshark

Die folgende Datei enthält zwei einfache Beispiel-Capture Dateien, die einmal eine nur AH-gesicherte Verbindung zeigt und einmal eine ESP-gesicherte Verbindung zeigt. Beide Verbindungen sind End-to-End und im Transport-Modus. D.h. die eigentlichen Kommunikationspartner sind erkennbar und nicht verschlüsselt.

DOWNLOAD CAPTURE

Hinweis: Bei der ESP-Verbindung wurde der Schlüssel aus den obigen Beispiel-Dateien verwendet. Wie im vorangegangen Abschnitt gezeigt kann man diese Datenstrom also entschlüsseln.

Parameter ab Wireshark 2.0

Bearbeiten → Einstellungen → Protocols → ESP

Unter `Encapsulating Security Payload` nur bei `Attempt to detect/decode encrpyted ESP payloads` setzen.

ESP SAs Edit:

Protocol	Src IP	Dest IP	SPI	Encryption	Encryption Key	Authentication	Authentication Key
IPv4	192.168.0.244	192.168.0.77	*	TripleDES-CBC	0x3f0b868...	NULL	0xbf9a081...
IPv4	192.168.0.77	192.168.0.244	*	TripleDES-CBC	0x3f0b868...	NULL	0xbf9a081...

HINWEIS: Das Authentication Protocol (hier eigentlich HMAC-MD5-96) darf **NICHT** ausgewählt werden, da sonst Wireshark ein rot markiertes `Malformed Packet` ausgibt. Stattdessen wird der Wert `NULL` verwendet. Gleiches gilt für die SPIs. Hier wird per *[9] der SPI offen gelassen.

Nicht eingesetzte Variante: automatisch verschlüsselte Verbindungen OpenSWAN

Mittels OpenSWAN lassen sich automatisch verschlüsselte Verbindungen herstellen. Begriffsklärung: Begriffe rund um OpenSWAN[10].

OpenSWAN im Detail

1) , 2)

s. Symbol rechts

3)

SAD: **S**ecurity **A**ssociation **D**atabase

4)

SPD: **S**ecurity **P**olicy **D**atabase

5)

States

6)

Internet **K**ey **E**xchange

7)

Phase 1: Aufbau eines verschlüsselten Konfigurationskanals

8)

Phase 2: Aushandlung der Nutzdaten-SAs

9)

entspricht wildcard, also beliebig

10)

SWAN steht für **S**ecure **WAN**