

IT Sicherachitekturen

**Einrichtung einer internen Firewall von
Firma-a und Firma-b, sowie
LAN-to-LAN-VPN von Firma-a
(Server) nach Firma-b(Client)**

Paul Drautzburg Georg Mohr

HTWG Konstanz, Sommersemester 2018

Inhaltsverzeichnis

1	Motivation	1
2	Problemstellung	1
2.1	Laborumgebung und Versuchsaufbau	1
2.2	Aufgabenstellung	3
3	Theoretische Grundlagen	4
3.1	Netzwerkarchitekturen	4
3.1.1	Die Begriffe Intranet, DMZ und Internet	4
3.2	Netzwerkprotokolle/Kommunikationsprotokolle	6
3.2.1	Aufbau eines Datenpakets	7
3.3	Virtuelles Betriebssystem (Debian GNU/Linux 7.11)	8
3.4	IP-Tables	9
3.4.1	Allgemein	9
3.4.2	Funktionsweise	10
3.4.3	Tabellen	10
3.4.4	Chains	10
3.4.5	Filterregeln	11
3.4.6	Policies	11
3.5	Public Key Infrastructure	12
3.6	OpenVPN	12
4	Lösungsskizze	13
4.1	Firewall	13
4.2	VPN	17
4.2.1	Serverzertifikate	17
4.2.2	Server Firma A	18
4.2.3	Client Firma B	22
5	Auswertung	23
6	Fazit	24

Abbildungsverzeichnis

1	Schematischer Aufbau der Laborumgebung im IT-Sicherheitslabor [ND18]	3
2	Aufbau eines Datenpakets [Hea]	8

Tabellenverzeichnis

1	OSI-Schichtenmodell	6
2	IP-Tables: Tabellen	10
3	IP-Tables: Chains	11
4	IPTABLES Optionen und Beschreibung	15
5	Befehle zur Erstellung eines Privaten Schlüssels mittels OpenSSL	18

1 Motivation

Im Zeitalter der Digitalisierung steigt der Grad der Vernetzung immer weiter an. Unternehmen, welche ihre Standorte über die ganze Welt verteilt haben, wollen Wege und Möglichkeiten haben ihre Arbeit sicher und Problemlos zu verrichten. Dafür ist es unerlässlich, dass verschiedene Standorte auf Ressourcen des anderen zugreifen können. Für solche Szenarien gibt es in der Netzwerktechnik und Netzwerkarchitektur verschiedene Lösungsansätze. Jedoch darf dabei ein wichtiger Aspekt nicht vernachlässigt werden, denn jedes Unternehmen hat Betriebsmittel und Informationen, welche es nicht Gefahren von außen aussetzen will. An diesem Punkt kommt der Begriff IT-Sicherheit ins Spiel und genau dieses Umfeld um den Begriff „IT-Sicherheit“ wird im Rahmen des Praktikums zur Lehrveranstaltung „IT-Sicherheitsarchitekturen“ untersucht. In den folgenden Kapiteln wird eine Problemstellung unter Laborbedingungen skizziert, welche ein Szenario widerspiegelt, mit dem sich Unternehmen täglich konfrontiert sehen. Anschließend wird werden die Laborbedingungen und der Versuchsaufbau beschrieben. Aus der Problemstellung und dem Versuchsaufbau wird eine Lösung auf Grundlage, der damit zu Grunde liegenden Theorie, erarbeitet. Diese Lösung wird abschließend in die Laborumgebung implementiert und auf ihre Akzeptanz gegenüber der Problemstellung untersucht und in einem Fazit bewertet.

2 Problemstellung

In diesem Kapitel wird die Laborumgebung eingeführt und beschrieben, sowie die damit verbundene Aufgaben-/Problemstellung skizziert.

2.1 Laborumgebung und Versuchsaufbau

Im Labor wird die IT-Landschaft zweier Unternehmen „A“ und „B“ simuliert. Jedes Unternehmen besitzt ein lokales LAN, an denen die pot. Mitarbeiter angeschlossen sind, zudem gibt es mehrere Server mit Hilfe derer verschiedene Dienste angeboten werden sollen. Es folgt eine Liste an Diensten welche Angeboten werden sollen:

- Internetzugang von den Mitarbeiter-Computern.
- Zugang über HTTPS der Webseiten beider Unternehmen.

-
- Eine Virtuelle Kopplung der beiden Unternehmen mit Hilfe einer sog. Site-to-Site Verbindung.
 - Jeder Mitarbeiter der Unternehmen soll die Möglichkeit haben, E-Mails über einen sicheren Mailserver zu verschicken.

Eine weitere wichtige Komponente in dieser IT-Landschaft sind die externen und internen Firewalls, da diese den Grundstein für eine sichere Umgebung legen. In dieser IT-Landschaft gibt es insgesamt 4 Firewalls, jedes Unternehmen hat jeweils eine interne und externe Firewall. Hinter den externen Firewalls beider Unternehmen liegt jeweils die sog. DMZ – Demilitarisierte Zone. In dieser Zone stehen üblicherweise die Unternehmensserver, diese werden nach außen hin von der externen Firewall und nach innen von der internen Firewall geschützt. Jede Firewall läuft auf einem Server, somit besteht die IT-Landschaft insgesamt aus 4 Servern auf welchen die Firewalls laufen, 2 Servern, welche die jeweiligen Unternehmensserver simulieren und einem Server welcher außerhalb der externen Firewall das Internet simulieren soll (vgl. Abb. xxx).

Internetzugang von den Mitarbeiter-Computern. Zugang über HTTPS der Webseiten beider Unternehmen. Eine Virtuelle Kopplung der beiden Unternehmen mit Hilfe einer sog. Site-to-Site Verbindung. Jeder Mitarbeiter der Unternehmen soll die Möglichkeit haben, E-Mails über einen Mailserver zu verschicken. Eine weitere wichtige Komponente in dieser IT-Landschaft sind die externen und internen Firewalls, da diese den Grundstein für eine sichere Umgebung legen. In dieser IT-Landschaft gibt es insgesamt 4 Firewalls, jedes Unternehmen hat jeweils eine interne und externe Firewall. Hinter den externen Firewalls beider Unternehmen liegt jeweils die sog. DMZ – Demilitarisierte Zone. In dieser Zone stehen üblicherweise die Unternehmensserver, diese werden nach außen hin von der externen Firewall und nach innen von der internen Firewall geschützt. Jede Firewall läuft auf einem Server, somit besteht die IT-Landschaft insgesamt aus 4 Servern auf welchen die Firewalls laufen, 2 Servern, welche die jeweiligen Unternehmensserver simulieren und einem Server welcher außerhalb der externen Firewall das Internet simulieren soll (vgl. Abb. 1).

Jede dieser angesprochenen Komponenten muss entsprechen konfiguriert werden, damit die angesprochenen Dienste fehlerfrei funktionieren können. Technisch gesehen wird jeder dieser Server auf Grundlage einer virtuellen Maschine abgebildet. Jede dieser Maschinen hat Linux als Betriebssystem installiert und vorkonfiguriert, jedoch ohne zusätzliche Pakete, nur die reine Grundinstallation. Da die Implementierung und Installation aller dieser Dienste und der damit verbundenen Komponenten den Rahmen dieses Praktikums

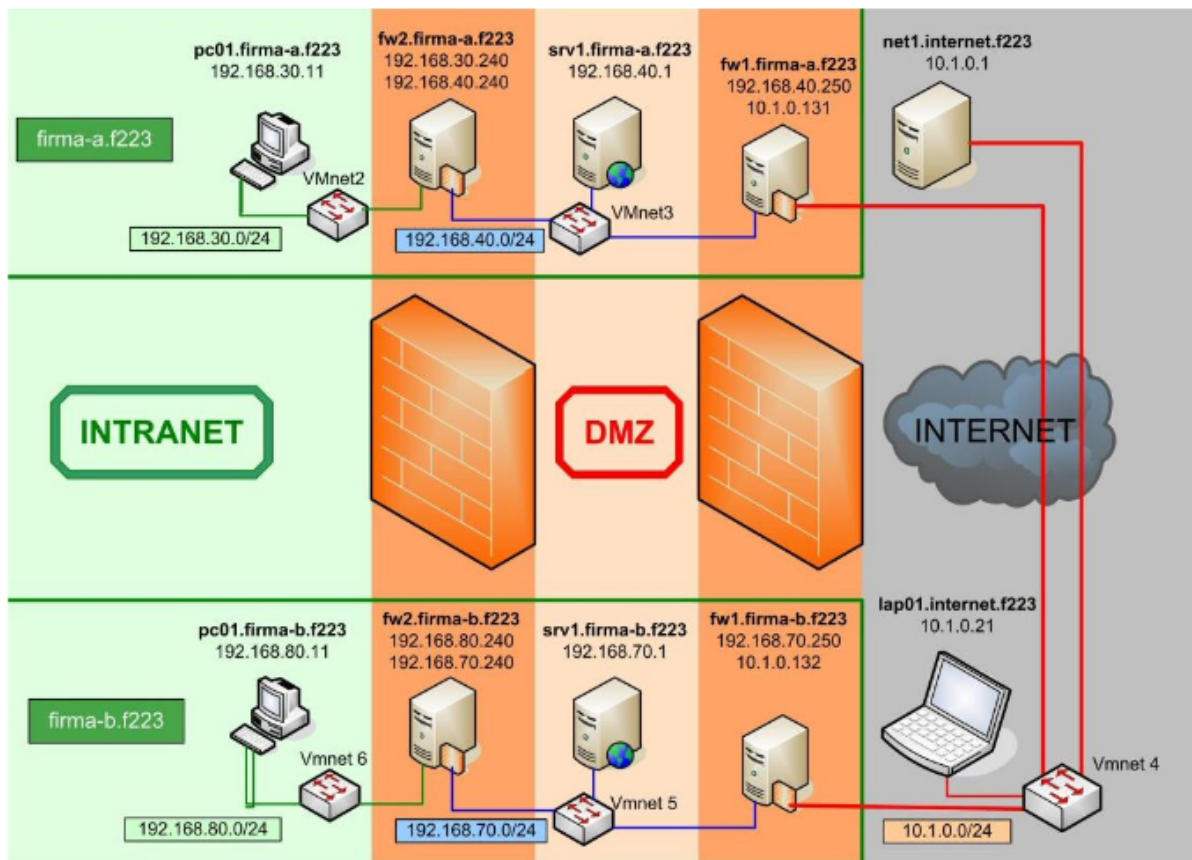


Abbildung 1: Schematischer Aufbau der Laborumgebung im IT-Sicherheitslabor [ND18]

sprengen würde, werden die Aufgabenstellungen zuvor runter gebrochen. Die detaillierte Aufgabenstellung wird im nächsten Abschnitt eingeführt.

2.2 Aufgabenstellung

Wie schon im vorhergehenden Abschnitt beschrieben, gibt es mehrere Dienste, welche in der IT-Landschaft angeboten werden sollen. In diesem Bericht, wird die Virtuelle Koppelung der Unternehmen „A“ und „B“ durch eine „Site-to-Site“ oder auch „LAN-to-LAN-“ genannt -Verbindung implementiert. Die Problemstellung gilt als gelöst, wenn es möglich ist, dass Unternehmen A eine gesicherte Verbindung zu Unternehmen B aufbauen kann

und die jeweiligen internen Firewalls entsprechend konfiguriert sind. Die Konfiguration der externen Firewalls kann in dieser Aufgabe vernachlässigt werden, hierfür sollen die bereitgestellten internen Firewalls verwendet werden. //TODO Verweis auf Bilder, Begriffe zuordnen.

3 Theoretische Grundlagen

In diesem Kapitel werden die theoretischen Grundlagen eingeführt, welche im Kontext der Aufgabestellung benötigt werden. Dies beschränkt sich auf drei Bereiche, die Netzwerkarchitekturen, Debian GNU Linux 7.11 und IP-Tables. Jede dieser genannten Bereiche wird innerhalb dieses Kapitels eingeführt und in den Aufgabenkontext eingeordnet.

3.1 Netzwerkarchitekturen

Der Bereich Netzwerkarchitekturen ist ein sehr breites Gebiet, deshalb muss dieser zuvor anhand der Anforderungen der Aufgabenstellung abgegrenzt werden.

3.1.1 Die Begriffe Intranet, DMZ und Internet

Das Netzwerk des Labors besteht netzwerktechnisch aus drei Bereichen dem Intranet, der Demilitarisierten Zone und dem Internet. Diese Begriffe werden folgend für den Rahmen dieses Berichts definiert und in den Kontext der Problemstellung eingeordnet.

”Intranet” Nach dem Gabler Wirtschaftslexikon wird ein Intranet als ” ein unternehmensinternes Kommunikationsnetz, in dem Daten auf der Basis der Protokollfamilie TCP/IP übertragen werden” definiert. In dieser Arbeit wird der Aspekt der Protokollfamilie ”TCP/IP” von vorrangiger Bedeutung sein. Der Punkt des ”unternehmensinternen Kommunikationsnetzes”, kann für diese Arbeit in den Hintergrund gestellt werden, da sich das Laborpraktikum auf die technischen Umsetzung beschränkt. Ein weiterer wichtiger Aspekt des Intranets ist, dass es nur autorisierten Benutzern gestattet werden soll zuzugreifen.

”Demilitarisierte Zone” Der Begriff demilitarisierte Zone kommt ursprünglich aus dem militärischen Umfeld und beschreibt eine Zone oder Bereich in der sich keine militärischen Streitkräfte gegenüberstehen dürfen, quasi ein neutraler Bereich. In der Netzwerktechnik wird dieser Begriff benutzt, um eine Zone zu beschreiben welche sich zwischen zwei Schutzeinrichtungen befindet. Bei diesen Schutzeinrichtungen handelt es sich meistens um eine externe Firewall und eine interne Firewall. Der Hintergrund für die Einrichtung einer DMZ ist es die Sicherheit der Komponenten und Teilnehmer eines Intranets zu sichern, falls es einen Angriff auf die Komponenten innerhalb der DMZ gibt und diese korumpiert werden sollten. Die Komponenten innerhalb einer DMZ werden als potentielle Opfer oder ”Victims” bezeichnet. Es handelt sich hier meistens um Server welche nach einem Schadensfall einfach durch einen Reboot wiederhergestellt werden können. Als Alternative könnte statt der DMZ ein sog. Application Layer Gateway (APL) eingesetzt werden. Dieser zeichnet sich dadurch, dass er den Netzwerkverkehr komplett auftrennt und sich nach allen Seiten als Kommunikationspartner verhält. Zudem wäre auch eine Lösung aus DMZ und APL[int16]. Im Rahmen des Praktikums wird jedoch lediglich eine DMZ eingesetzt.

”Internet” Das Internet ist ”ein weltumspannendes, heterogenes Computernetzwerk, das auf dem Netzwerkprotokoll TCP/IP basiert. Über das Internet werden zahlreiche Dienste wie z.B. E-Mail, FTP, World Wide Web (WWW) oder IRC angeboten” [Gab]. Ein wichtiger Punkt im Rahmen dieses Praktikums ist, dass das Internet innerhalb der Laborumgebung nur simuliert wird, es besteht also kein richtiger Zugang zum Internet. Dieser Fakt stellt sicher, dass der Versuchsaufbau bei falscher Konfiguration, von außen beschädigt werden könnte. Zudem wird somit sichergestellt, dass bei der Konfiguration der virtuellen Maschinen nur die Versionen der zu Grunde liegenden Images verwendet werden können.

Nachdem die Begriffe Intranet, DMZ und Internet im Kontext der Laborumgebung und Problemstellung definiert und eingrenzt wurden müssen noch weitere Begriffe eingeführt werden. Im folgenden Abschnitt werden die Netzwerkprotokolle oder auch Kommunikationsprotokolle eingeführt, welche im Rahmen des Praktikums verwendet werden.

3.2 Netzwerkprotokolle/Kommunikationsprotokolle

Um eine Netzwerkkommunikation verschiedener Komponenten innerhalb eines Netzwerks zu gewährleisten müssen Netzwerkprotokolle eingesetzt werden.

Im Rahmen des Laborpraktikums werden verschiedene Netzwerkprotokolle eingesetzt, diese lassen sich am besten der entsprechenden Layern im OSI-Schichtenmodell beschreiben. In der nachfolgenden Tabelle werden anhand der Schichten die einzelnen Protokolle eingeordnet, anschließend werden die für dieses Praktikum relevanten Protokolle gesondert beschrieben.

#	OSI-Schicht	Einordnung	Protokolle	Kopplungselemente
7	Anwendungen(Application)	Anwendungsorientiert	HTTP, FTP, HTTPS, SMTP, DNS, LDAP	Gateway
6	Darstellung(Presentation)			Content-Switch
5	Sitzung(Session)			Proxy
4	Transport(Transport)	Transportorientiert	TCP, UDP, SCTP, SPX	Layer-4-7-Switch
3	Vermittlung-/Paket(Network)		ICMP, IGMP, IP, IPsec, IPX	RouterLayer-3-Switch
2	Sicherung(Data Link)		Ethernet, Token Ring, FDDI	BridgeLayer-2-Switch
1	Bitübertragung(Physical)			Netzwerkkabel Repeater Hub

Tabelle 1: OSI-Schichtenmodell

Die bereits beschriebene Aufgabenstellung (vgl. 2.2 Aufgabenstellung) soll eine site-to-site VPN eingerichtet werden. Da es um eine die Verbindung geht, kann der Fokus auf die Transport-orientierten Schichten der OSI-Modells 1-4 gerichtet werden. Für die Umsetzung einer VPN-Verbindung und Konfiguration der internen Firewalls sind die folgenden Protokolle vordergründig

”ICMP” Das Internet Control Message Protocol, wird dazu verwendet in Rechnernetzen einen Austausch von Fehlermeldungen durchzuführen.

”IP” Das Internet Protocol ist ein zustands- und verbindungsloses Protokoll, welches die Implementierung der Vermittlungsschicht (3) widerspiegelt.

”IPsec” Das Internet Protocol Security ist eine Erweiterung des IP-Protokolls und soll eine gesicherte Kommunikation über unsichere IP-Netze ermöglichen.

”TCP” Das Transmission Control Protocol wird von allen modernen Betriebssystemen genutzt um, zu definieren, wie die Daten zwischen verschiedenen Netzwerkkomponenten ausgetauscht werden sollen

”UDP” Das User Datagram Protocol zeichnet sich dadurch aus, dass es verbindungslos und ungeschützt ist. Es kann also nicht gesichert werden, ob ein gesendetes Datenpaket richtig (nicht verfälscht von Dritten) oder überhaupt ankommt. [KR08]

Diese beschriebenen Protokolle sind der Lösung der Aufgabe von großer Bedeutung. Sie werden bei der Erstellung der Lösungsskizze wieder aufgegriffen. Um die Beschreibung Netzwerkprotokolle zu komplettieren muss noch auf eine Ebene tiefer geschaut werden, auf die Datenpakete. Im nächsten Abschnitt wird der Aufbau und die wichtigsten Eigenschaften eines Datenpakets beschrieben.

3.2.1 Aufbau eines Datenpakets

Im vorhergehenden Abschnitt wurden die wichtigsten Netzwerkprotokolle für den Rahmen des Laborpraktikums beschrieben. In diesem Abschnitt, wird das Zusammenspiel eines Netzwerksprotokoll und Datenpaketen gezeigt.

In einem Protokoll wird der Aufbau eines Datenpakets definiert, zudem enthält es wichtige Informationen über den Datenaustausch. Es definiert,

- wer der Absender und Empfänger eines Datenpaketes sein soll,
- von welchem Typ ein Datenpaket ist, ob es für den Verbindungsaufbau, Verbindungsabbau oder für Nutzdaten genutzt wird,
- die Größe des Datenpaket, welches beim Empfänger ankommen soll.

Wenn es sich um eine mehrteilige Kommunikation handelt, muss zusätzlich noch die laufende Nummer und die Anzahl der Pakete definiert werden. Zuletzt folgt noch eine Prüfsumme, mit dessen Hilfe der Empfänger prüfen kann, ob die Datenpakete fehlerfrei angekommen sind. Alle dieser beschriebenen Informationen werden dem ”Header” eines Datenpaketes vorne oder hinten angehängt, dieser angehängte Bereich wird auch ”Trailer genannt”. Anhand der folgenden Abbildung 2 wird der Aufbau eines Datenpakts und der einzelnen Bestandteile noch verdeutlicht. Nach dem nun die wichtigsten Bestandteile eines auch Datenpakets eingeführt wurden, sind alle Grundlagen aus dem Bereich der Netzwerkarchitekturen und Netzwerktechnik beschrieben, welche im Rahmen des Praktikums benötigt werden. Es wurde die Begriffe DMZ, Intranet und Internet eingeführt, sowie die wichtigsten Netzwerkprotokolle und ihr Zusammenhang zu den Datenpaketen. Im folgenden Kapitel wird das zugrunde liegende Betriebssystem ”Debian GNU Linux 7.11” und die Besonderheiten in der Laborumgebung in kürze beschrieben.

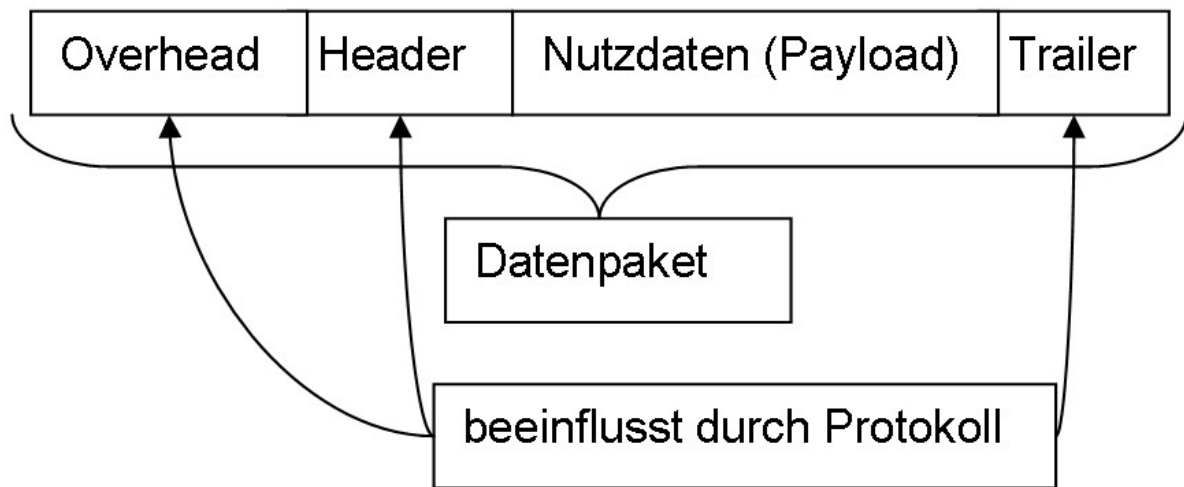


Abbildung 2: Aufbau eines Datenpakets [Hea]

3.3 Virtuelles Betriebssystem (Debian GNU/Linux 7.11)

Debian GNU Linux 7.11 ist ein kostenfreies gemeinschaftliches entwickeltes Betriebssystem. Die Version 7.11 welche im Rahmen des Praktikums eingesetzt wird, gehört zu einer älteren Version und wird von Softwareentwicklern nicht mehr unterstützt. Es unterstützt verschieden Rechnerarchitekturen wie z.B. x86(i386 & amd64).

Die Besonderheit im Laborpraktikum ist die, dass mit virtuellen Maschinen auf einem zugrunde liegenden Windows 10 System gearbeitet wird. Dies bringt einige Probleme mit sich, zum einen müssen die virtuellen Maschinen im Bereich der Netzwerkadapter einwandfrei konfiguriert sein, zum anderen kann mit den Maschinen nicht auf das "WWW" zugegriffen werden. Dies stellt ein wesentliches Problem da, da ohne Verbindung ins WWW Paketdienste wie Aptitude nicht funktionieren. Aus letzterem Grund muss zwingend bei der Installation von neuen Paketen das virtuelle Image in die Maschine "eingelegt" werden. Eine andere Möglichkeit wäre es, die Pakete einzelnen mit Hilfe des zugrunde liegenden Systems zu Downloaden und einzeln einzuspielen. Diese Variante stellte sich aber als sehr zeitintensiv und kompliziert heraus, da jedes Paket Abhängigkeiten mitbringt und diese zusätzlich an Versionen gebunden sind. So kann es passieren, dass dieser Prozess der Installation eine neue Problemkette im Betriebssystem schafft. Somit wurden alle Pakete, welche für die Lösung der Aufgabe benötigt wurden, mit Hilfe der bereitliegenden Linux-Images gemacht.

Eine weitere Besonderheit ist die, dass es keine grafische Benutzeroberfläche gibt, dies setzt die Arbeit "auf Konsole" voraus. Ein Problem was daraus resultiert ist, die Arbeit mit den Linux internen Texteditoren wie z.B. dem "VI" oder "Nano", dies stellt insofern ein Problem da, da Linux in sich hochgradig "case sensitiv" reagiert. Dies bedeutet im Umkehrschluss, wenn ein Leerzeichen oder Komma falsch gesetzt wird, terminiert bspw. ein Skript nicht so wie der Nutzer es vorgesehen hat. Jedoch ist es zwingend empfohlen sämtliche Skripte auf dem virtuellen System zu schreiben, da sonst durch das Kopieren von bspw. einem Windows Betriebssystem illegale Steuerzeichen im Skript auftreten könnten[edi].

Ein wichtiger Punkt im Zusammenspiel der Skripte und dem Betriebssystem ist die Wahl der Shell mit welcher es ausgeführt werden soll. Linux bietet viele verschiedene Shells an, im Rahmen des Praktikums wurde ausschließlich mit der sog. "bash" gearbeitet, da sie unter Linux die "Standard-Login" Shell darstellt[bas]. In sämtlichen Skripten wurde in der ersten Zeile der sog. "Shebang", welcher aus "# !" besteht, eingefügt. Gefolgt von diesem wird der Ort des Interpreter angegeben mit welchem das Skript ausgeführt werden soll. In diesem Fall sieht diese erste Zeile eines Skripts immer wie folgt aus, "#!/bin/bash" [she].

Damit wurden alle wichtigen Besonderheiten des Betriebssystem im Rahmen des Praktikums eingeführt. Im nächsten Kapitel wird das "IP-Tables" Paket von Linux eingeführt, da dieses einen Grundstein für die Lösung der Aufgabenstellung im Bezug auf die internen Firewalls darstellt.

3.4 IP-Tables

3.4.1 Allgemein

Das Paket "IP-Tables" ist Standardmäßig in der Linux Installation enthalten. Es kann aber auch einfach über den Paketmanager mit Hilfe des Befehls "apt-get install iptables" installiert werden. In diesem Fall war das Paket schon installiert und musste nicht mehr manuell installiert werden. Grundslegend stellt das IP-Tables einen Paketfilter dar, welcher Pakete auf IP-Ebene auf Basis zuvor definierten Regeln filtert. Empfangene Pakete werden überprüft bevor sie weitergeleitet werden. Umgekehrt werden ausgehenden Pakete geprüft, bevor sie den Rechner verlassen. Es kann aber auch sein, dass der Rechner als Router fungiert, so müssen alle Pakete geprüft werden während diese weitergeleitet werden sollen.

3.4.2 Funktionsweise

Die Paketprüfung wird dreistufig in einer Hierarchie durchgeführt. Folgende drei Stufen gibt es

- Tabellen,
- Chains (Ketten),
- Filterregeln.

Wird eine definierte Regel in einer Chain oder Tabelle gefunden wird die hinterlegte Aktion ausgeführt. Wenn keine Regel definiert wurde, wird eine allgemein definierte sog. Policy ausgeführt.

3.4.3 Tabellen

In einer Tabellen werden die Filterregeln zu Gruppen zusammengefasst und nach der grundlegenden Aufgabe sortiert. Insgesamt gibt es vier Tabellen in denen eine Regel eingeordnet werden kann. In der folgenden Tabelle werden diese vier Tabellen und ihre Eigenschaften beschrieben[ipt].

Tabelle	Beschreibung
filter	Standardtabelle, hier werden nur die grundlegenden Regeln hinterlegt.
nat	(Network Address Translation) Tabelle für die Adressumsetzung und für das Port-Forwarding.
mangle	Tabelle bei gewünschter Manipulation von Paketen.
raw	Tabelle für Ausnahmen beim Connection tracking.

Tabelle 2: IP-Tables: Tabellen

3.4.4 Chains

In jeder Tabelle sind sog. "Chains" enthalten, welche festlegen wann ein Paket geprüft wird, bspw. bevor es verschickt werden soll. In der folgenden Tabelle werden alle fünf verschiedenen möglichen "Chains" beschrieben[ipt].

Chain	Tabelle	Beschreibung
INPUT	filter, mangle	Relevant für alle Pakete, welche einen lokalen Prozess gerichtet sind.
OUTPUT	filter, nat, mangle, raw	Relevant für alle Pakete, welche von einem lokalen Prozess abstammen.
FORWARD	filter, mangle	Relevant für alle Pakete, welche geroutet werden.
PREROUTING	nat, mangle, raw	Relevant für alle Pakete, bevor sie geroutet werden.
POSTROUTING	nat, mangle	Relevant für alle Pakete, nachdem sie geroutet werden sollen.

Tabelle 3: IP-Tables: Chains

3.4.5 Filterregeln

Sobald ein Paket auf eine Filterregel trifft muss definiert werden, wie mit dem Paket umgegangen werden soll. Dafür gibt es vier Optionen, welche am häufigsten Anwendung finden [ipt]. Diese sind

- ACCEPT: Das Paket wird akzeptiert und angekommen,
- DROP: Das Paket wird, ohne Nachricht an den Absender, abgelehnt,
- REJECT: Das Paket wird, mit Nachricht an den Absender, abgelehnt,
- LOG: Die Daten des Paket werden in den System Log geschrieben und es wird mit der nächsten Regel der Chain fortgefahren.

Für dieses Laborpraktikum sind die beschriebenen vier Filterregeln ausreichend, im Kontext des IP-Tables Pakets bietet Debian/Linux noch weitere Optionen an.

3.4.6 Policies

Da sich aus den beschriebenen Möglichkeiten eine große Anzahl an Möglichkeiten für Filterregeln für die einzelnen Pakete ergibt, gibt es sog. "Policies". Diese "Policies" setzen sich aus der Filterregel, der Chain und Aktion zusammen [ipt]. Exemplarisch könnte also so ein Befehl, für Input, Drop oder Forward wie folgt aussehen:

Auflistung 1: Policies

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

Alle beschriebenen Möglichkeiten von IP-Tables wurden im Rahmen des Praktikums für die Lösung der Problemstellung eingesetzt. Dies ist jedoch nur ein Teil der Möglichkeiten von IP-Tables und spiegelt nicht den kompletten Funktionsumfang ab. Im folgenden Kapitel wird eine Einführung in die Grundlagen einer Public Key Infrastructure gegeben, wie sie auch für das Praktikum benötigt wird.

3.5 Public Key Infrastructure

3.6 OpenVPN

4 Lösungsskizze

In diesem Kapitel wird sich mit dem Thema befasst, wie sich eine Side-to-Side VPN Verbindung, die die Firmen A und B wie in Kapitel 2.1 beschrieben verbindet. Dazu waren zu Beginn mehrere Vorüberlegungen zu tätigen. Dies sind zum einen, wie wird das innere Netz der Firmen A und B geschützt, dann welcher VPN Dienst kann für diesen Zweck verwendet werden und welche weiteren Strategien braucht es bei der Umsetzung der Security Policies der Firmen.

4.1 Firewall

Wenn eine Firewall aufgebaut werden soll, muss sich zu Beginn überlegt werden, welche allgemeine Strategie mit der Firewall gefahren werden soll. Das heißt sollen alle Verbindungen Standardmäßig freigegeben werden und nur explizit nicht erlaubt Verbindungen geblockt werden (Default-Allow-Strategy), oder sollen alle Verbindungen blockiert werden und nur die die explizit erlaubt wurden, geöffnet werden (Default-Deny-Strategy). Im Standardfall wird in Unternehmen, die Strategie Default-Deny verwendet, weshalb auch im Versuchsaufbau diese Strategie gewählt wurde. Eine Firewall wird mit Hilfe des Userspace-Programmes IPTABLES (siehe Kapitel 3.4) unter Linux realisiert. Dafür wird eine Rules Datei (nachfolgend Skript genannt) angelegt, in dieser werden die Einstellungen und Regeln für die Firewall definiert. Zu Beginn des Skriptes wird der Interpreter für den Code angegeben. Dann wird begonnen die durch die Default Strategie vorgegebenen Default Policies umzusetzen (siehe Auflistung 2). Dabei sagt das -P, dass die Policy angesprochen werden soll, das INPUT, OUTPUT und FORWARD bezieht sich auf die im Kapitel 3.4 vorgestellten Chains. DROP sagt dabei aus, dass die eintreffenden Pakete, wenn keine weiteren Regeln definiert oder zutreffen nicht weitergeleitet und “fallengelassen” werden sollen.

Auflistung 2: Default Policy IPTABLE

```
#Verwenden der Tabelle FILTER
*filter
-P INPUT DROP
-P OUTPUT DROP
-P FORWARD DROP
```

Nach dem die Default Policy erfolgreich eingeführt wurde, werden die feingranulareren Regeln definiert. Dies setzt die Verwendung des Befehles -F voraus. Dieser löscht alle bisherigen Filterregeln, um die nach folgenden neu definierten Regeln einzuführen. Bei Linux ist dabei zu beachten das ein Teil der Interprozesskommunikation über das interne Netzwerk läuft. Dafür ist es nötig diese Kommunikation zuzulassen, dies geschieht wie in der nachfolgenden Auflistung 3 zusehen ist.

Auflistung 3: Interprozesskommunikation zulassen

```
# Interprozesskommunikation Verbindungen erlauben
-A INPUT -i lo -j ACCEPT
-A OUTPUT -o lo -j ACCEPT
```

Dabei sagt das -A an welche Chain diese Regel angehängt werden soll. Das -i ist dabei die Option über welches Netzwerkinterface das Paket eingegangen ist, beziehungsweise -o für das Paket versenden. In diesem Fall das "lo" Netzwerkinterface. Wenn alle Prüfregeln auf das Paket zutreffen wird mit -j entschieden wie mit dem Paket verfahren werden soll, in diesem Beispiel soll es mit ACCEPT akzeptiert werden.

Auflistung 4: Weitere allgemeine Firewallregeln

```
# Erlaube ICMP Befehle
-A INPUT -p icmp -j ACCEPT
-A OUTPUT -p icmp -j ACCEPT

# Erlaube SSH Verbindung
-A INPUT -p tcp --dport 22 -j ACCEPT

# Alle Verbindungen von innen nach aussen zulassen
-A FORWARD -i eth0 -o eth1 -m state --state NEW -j ACCEPT

# Erlaube nur bereits aufgebaute
# Verbindungen von aussen nach innen
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
```

Die in Auflistung 4 dargestellten Regeln sind allgemeine Regeln für die Firewall, diese erlauben das empfangen von Pinganfragen, den Fernzugriff mittels SSH, alle neuen Verbindungen von innen nach außen und alle bereits aufgebauten Verbindungen beziehungsweise alle Verbindungen die einen Bezug auf eine andere Verbindung besitzen. In

der nachfolgenden Tabelle 4 werden die hier verwendeten Optionen nochmals näher beschrieben. Nach den allgemeine Regeln müssen nun die VPN spezifischen Regeln definiert

Optionen	Beschreibung
-p	Gibt das Protokoll an welches verwendet werden soll hier icmp(Ping), tcp
-dport	Gibt den Destinationport an, auf den zugegriffen werden soll hier 22
eth0,eth1	bezieht sich auf das verwendete Netzwerkinterface
-m state	die enttreffenden Pakete sollen auf den Status überprüft werden
-state	Status der eintreffenden Pakete, hier NEW, RELATED,ESTABLISHED

Tabelle 4: IPTABLES Optionen und Beschreibung

werden. Dazu muss man die auf den VPN Ports eintreffenden und ausgehenden Pakete betrachten. Diese Ports sind entweder 1194 oder 1195. In der Auflistung 5 sind VPN Regeln zusehen. Eine Besonderheit bei VPN ist das zum einen alle verbindungen die über das Netzwerkinterface tun0 eintreffenden Pakete ohne Überprüfung an das Inteface eth0 übergeben werden. Des weiteren wird noch die Tabelle NAT benötigt, was das ”*nat” angibt. In dieser Tabelle gibt es die Chain POSTROUTING, über diese kann nachträglich der Verkehrsheader eines Paketes verändert werden. MASQUERADE hat dabei die Funktion das wenn ein Paket versendet wird die Source-IP-Adresse so verändert wird das nur noch die IP-Adresse des Firewallservers ersichtlich ist. Dies hat den Grund das von außen nicht ersichtlich wird, was sich für IP-Adressen hinter der Firewall befinden und es so Angreifern erschwert wird diese anzugreifen.

Auflistung 5: Weitere VPN Firewallregeln

```
# Erlaube alle Verbindungen auf den VPN Ports
-A INPUT -i eth1 -p udp --dport 1194 -m state --state NEW -j ACCEPT
-A INPUT -i eth1 -p udp --dport 1195 -m state --state NEW -j ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i tun0 -j ACCEPT

#Forwarding fuer die VPN Verbindungen
-A FORWARD -i tun0 -o eth0 -m state --state NEW -j ACCEPT
-A FORWARD -i eth0 -o tun0 -m state --state NEW -j ACCEPT
-A FORWARD -i tun0 -o eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -i eth0 -o tun0 -m state --state RELATED,ESTABLISHED -j ACCEPT
COMMIT

#Verwenden der Tabelle NAT
*nat
```

```
#Loeschen der vorhandenen Regeln
-F

-A POSTROUTING -o eth1 -j MASQUERADE
COMMIT
```

Nach dem erstellen des Skriptes, muss dieses nun eingespielt werden. Dazu wird der Befehl "iptables-restore" verwendet. Dieser wird exemplarisch in der Auflistung 6 veranschaulicht, dabei liegt das Skript im Ordner "etc".

Auflistung 6: Einspielen des Firewallskriptes der Firma a
iptables-restore < etc/iptables.firewall-a.rules

Nach dem das Skript eingespielt wurde, muss dieses noch so konfiguriert werden, dass es nach einem Serverneustart automatisch neu eingespielt wird. Dazu könne zwei Arten verwendet werden. Zum einen mit Hilfe des "if-pre-up.d" Directories. Dabei wird eine neue Datei angelegt mit dem Namen "iptables" angelegt. In der Auflistung 7 wird der Inhalt, der neu angelegten Datei gezeigt. Nun muss die Datei noch Ausführbar gemacht werden, dies geschieht mit Hilfe des Befehls "chmod +x"

Auflistung 7: Automatisches Laden des Iptableskriptes bei Serverneustart

```
#!/bin/bash
/sbin/iptables-restore < /etc/iptables.fwirewall-a.rules
```

Die zweite Möglichkeit des automatischen Neuladens der Firewall ist mit Hilfe des Tools "iptables-persistent". Dieses Tool muss mit "aptitude" installiert werden. Nach dem das Tool installiert wurde, können mit dem Befehl "iptables-save > /etc/iptables/rules.v4" die Regeln für jeden Neustart automatisch geladen werden.

Eine weitere Konfiguration die vorgenommen werden muss, ist das Akzeptieren des Forwardings. Dazu muss im Verzeichnis "/proc" die Datei "/proc/sys/net/ipv4/ip_forward" die Zeile in der Auflistung 8 eingefügt werden. Zu Beginn steht in dieser Datei lediglich eine "0" diese muss ersetzt werden durch eine "1" dies aktiviert das Forwarding.

Auflistung 8: Forwarding aktivieren

```
/bin/echo "1" > /proc/sys/net/ipv4/ip_forward
```

Wurden all diese Einstellungen vorgenommen, kann begonnen werden mit der Konfiguration der Site-toSite VPN Verbindung, was im nachfolgenden Kapitel beschrieben wird.

4.2 VPN

Nach dem die Firewall auf den Servern der Firma A und B eingerichtet wurde, muss nun die VPN Verbindung erstellt werden. Dazu dient der Firewallserver der Firma A als Server und der Firewallserver der Firma B als Client. Diese Einrichtung wird in diesem Kapitel näher beschrieben.

Dafür muss zu Beginn auf beiden Servern das in Kapitel 3.6 vorgestellte Tool OpenVPN installiert werden. Dies geschieht wie schon im Kapitel 4.1 erwähnt mit dem Tool "aptitude". Nach der Installation müssen wie nachfolgend erklärt für den Server der Firma A und B ein Serverzertifikat erstellt werden.

4.2.1 Serverzertifikate

Wie schon in Kapitel 3.5 erklärt wird für die Einrichtung eines VPN's Zertifikate benötigt. Diese werden verwendet um eine Verbindungsverschlüsselung zu realisieren. In diesem Kapitelteil wird anhand der Firma A erklärt, wie ein solches Zertifikat in der Laborumgebung erstellt wird, um ein Zertifikat für Firma B zu erstellen, ist der gleiche Prozess notwendig.

Für das Erstellen eines Zertifikates ist das Tool OpenSSL notwendig. Über OpenSSL wird ein öffentlicher und privater Schlüssel erzeugt. Nach dem diese erstellt wurden, muss von der Zertifizierungsstelle die Konfigurationsdatei geladen werden. Mit Hilfe dieser Konfigurationsdatei des privaten Schlüssels und des Tools OpenSSL wird nun eine CSR (Certificate Signing Request) Datei erzeugt. Nach dem diese Datei erzeugt wurde, muss diese zur Zertifizierungsstelle hochgeladen werden. Nach dem Upload der Datei (Zertifikat), muss diese nun Signiert werden. Dies wird im Labor mit einem Shellskript realisiert. Nach dem Signieren kann nun das fertige Zertifikat geladen und auf dem Server abgelegt werden. In der Auflistung 9 werden die einzelnen Befehle zur Erstellung der Schlüssel und der CSR Datei gezeigt und die einzelnen Befehle in der Tabelle 5 näher

beschrieben.

Auflistung 9: Erzeugen eines privaten Schlüssels mit OpenSSL

```
#Privaten Schluessel erzeugen
openssl genrsa -out firma-A.key 2048

#Request erstellen - CSR-Datei
openssl req -new -key firma-A.key -config req.cnf -reqexts v3_req_srv -o
```

Befehle	Beschreibung
openssl	Tool zur Erstellung von privaten Schlüsseln
genrsa	Befehl zum erstellen des Schlüssels mittels RSA
-out	Ausgabe des Schlüssels in angegebene Datei
firma-A.key	Name der Schlüsseldatei
2048	Schlüssellänge in Bits
req -new	Erzeugen eines neuen Requests
-config	laden der Konfigurationsdatei req.cnf
-reqexts	Erweiterung des X.509 Formates

Tabelle 5: Befehle zur Erstellung eines Privaten Schlüssels mittels OpenSSL

4.2.2 Server Firma A

Nach dem die Zertifikate wie im vorherigen Kapitel erstellt wurden, kann nun mit dem Konfigurieren des VPN-Servers begonnen werden. Dazu wird zuerst mittels des Root-Users das Verzeichnis `"/etc/opnevpn"` erstellt. Nach dem dieses erstellt wurde werden zwei Unterverzeichnisse `"/etc/openvpn/certs"` und `"/etc/openvpn/keys"` angelegt. In das Unterverzeichnis `"certs"` werden das Serverzertifikat `"fw2-firma-A.crt"` und die Zertifikatschain (von der Zertifizierungstelle) kopiert. In das Unterverzeichnis `"keys"` wird der private Schlüssel verschoben. Da der private Schlüssel geheimgehalten werden soll, muss die Berechtigung für das Unterverzeichnis noch geändert werden. Dies wird mit dem Befehl `"chmod -R 600 /etc/openvpn/keys/"` bewerkstelligt. Der nächste Schritt ist das erstellen des Diffie-Hellman Schlüssels. Dieser Schlüssel wird später benötigt um später die Verbindung symmetrisch zu verschlüsseln. Der dafür notwendige Befehl wird in der Auflistung 10 gezeigt.

Auflistung 10: Erzeugen des Diffie-Hellman Schlüssels

```
openssl dhparam -out dh2048.pem 2048
```

Eine besondere Eigenschaft des Site-to-Site VPN's ist das man für jeden einzelnen Clienten eine besondere Konfiguration "pushen" kann. Dies bringt den Vorteil, dass wie hier verwendet, ein Client die Infrastruktur hinter der Firewall so verwenden kann, als wäre sie ein Teil seiner eigenen Infrastruktur. Dazu muss ein weiteres Unterverzeichnis mit dem Namen "client_configs" erstellt werden. Darin muss eine Datei angelegt werden, dass den gleichen Namen trägt wie das Clientzertifikat. In dieser Datei werden die Konfigurationsdaten für den Clienten geschrieben, wie in Auflistung 13 zusehen ist.

Auflistung 11: Clientkonfigurationsdatei fw-firma-b

```
iroute 192.168.70.0 255.255.255.0  
iroute 192.168.80.0 255.255.255.0
```

Das "iroute" Statement gibt dem Client an welche IP-Adressen von ihm angesprochen werden können.

Nun muss als letztes die OpenVPN Konfigurationsdatei erstellt und angepasst werden. Dazu können die bei der Installation von OpenVPN mitgelieferten Beispieldateien verwendet werden und nach und nach an den Verwendungszweck angepasst werden. Für die Konfiguration für den Server wird die "server.conf" benötigt.

Auflistung 12: server.conf Datei der Firma A

```
script-security 3  
  
port 1194  
  
proto udp  
  
dev tun  
  
tls-server  
auth SHA1  
  
ca /etc/openvpn/cert/f223CA.chain.crt  
cert /etc/openvpn/cert/fw-firma-a.crt
```

```
# This file should be kept secret
key /etc/openvpn/keys/firma-A.key

dh /etc/openvpn/dh2048.pem

server 192.168.100.0 255.255.255.0

ifconfig-pool-persist ipp.txt

route 192.168.70.0 255.255.255.0
route 192.168.80.0 255.255.255.0
push "route 192.168.30.0 255.255.255.0"
push "route 192.168.40.0 255.255.255.0"

client-config-dir /etc/openvpn/client-configs

keepalive 10 120

cipher aes-256-cbc

comp-lzo

user nobody
group nogroup

persist-key
persist-tun

verb 3
```

Die Auflistung 12 zeigt die "server.conf" der Firma A. Dabei wird von oben nach unten die folgenden Konfigurationsbefehle abgearbeitet.

script-security 3 Beschreibt die Skript Sicherheit, das heißt hier wird definiert wie OpenVPN externe Programme und Skripte verwenden darf. Die drei erlaubt die Übergabe von Passwörtern an Skripte über Umgebungsvariablen.

port 1194 Definiert den Port über den der VPN Verkehr abläuft. Alternativport 1195

proto udp Definiert über welches Protokoll kommuniziert wird.

dev tun Erzeugt einen gerouteten IP-Tunnel.

tls-server auth SHA1 Authentifikationskonfiguration

ca Pfadangabe zur Zertifikatschain

cert Pfadangabe zum Serverzertifikat

key Pfadangabe zum Privaten Schlüssel

dh Pfadangabe zum Diffi-Hellman Key

server Spezifiziert den Server als Server und gibt den IP-Adress Raum des VPN Netzes an.

ifconfig-pool-persist Datei in dem IP-Adressen der Clients gespeichert werden, um beim nächsten Verbinden von dem selben Client diesem die gleiche IP-Adresse zu geben.

route Definiert die hinter dem Client liegenden IP-Adressen auf die von Serverseite aus zugegriffen werden kann.

push IP-Adressraum der an den Client "gepusht" werden, die der Client annehmen darf.

client-config-dir Pfadangabe zur Clientkonfiguration

keepalive Gibt an das nach 10 Sekunden Inaktivität ein Ping an den Client gesendet wird, nach 120 Sekunden Inaktivität wird ein weiterer Versuch unternommen sich mit dem Client zu verbinden.

cipher Kryptographische Chiffre, über die die Verbindung zwischen Client und Server verschlüsselt wird. Dies muss beim Client der gleiche Chiffre sein.

comp-lzo Aktiviert die Kompression der Verbindung, dies muss auch beim Client aktiviert sein.

user Nach der Initialisierung werden die Rechte des OpenVPN Nutzers auf nobody gesetzt, da zum starten eines VPN's Rootrechte benötigt und dies ist aber während der Sitzung nicht gewollt ist.

group Nach der Initialisierung werden die Rechte des OpenVPN Nutzers auf nogroup gesetzt, da zum starten eines VPN's Rootrechte benötigt und dies ist aber während der Sitzung nicht gewollt ist.

persist-key Sichert vor einem Keepalive Neustart die Schlüssel, um diese nicht nochmals erzeugen zu müssen.

persist-tun Ähnlich wie persist-key nur für das tun-interface.

verb 3 Logging Optionen, über diese Einstellung kann die Genauigkeit des Loggens eingestellt werden.

4.2.3 Client Firma B

Ähnlich wie beim Server der Firma A, wie im Kapitel-teil zuvor, wird der Client der Firma B eingerichtet. Allerdings wird hier die "client.conf" benötigt und es wird kein "client_conf" Verzeichnis benötigt. Nachfolgend wird in der Auflistung die Konfigurationsdatei gezeigt und die einzelnen Begriffe nochmals erklärt.

Auflistung 13: client.conf Datei der Firma B

```
port 1194

client

dev tun

proto udp

remote 10.1.0.131 1194

user nobody
group nogroup

persist-key
persist-tun

ca /etc/openvpn/f223CA.chain.crt
cert /etc/openvpn/fw2.firma-b.f223.crt
key /etc/openvpn/vpnb.key

pull
auth SHA1

cipher aes-256-cbc

comp-lzo

verb 3
```

client Spezifiziert den Client als Client

remote Gibt die IP-Adresse und den Port des VPN-Servers an, mit dem sich der Client verbinden soll.

tls-client Spezifiziert den Client als tls-Client

pull holt sich über diesen Befehl die von Server "gepushten" Konfigurationen

5 Auswertung

6 Fazit

Literatur

- [bas] <https://wiki.ubuntuusers.de/Bash/>
- [edi] <https://wiki.ubuntuusers.de/Editoren/>
- [Gab] <https://wirtschaftslexikon.gabler.de/definition/internet-37192>
- [Hea] <https://blog.milsystems.de/2011/11/netzwerktechnologie/>
- [int16] *DMZ, demilitarisierte Zone.* <https://www.iternas.com/dmz>. Version: 2016, Abruf: 20.07.2018
- [ipt] <https://wiki.ubuntuusers.de/iptables2/>
- [KR08] KUROSE, James F. ; ROSS, Keith W.: *Computernetzwerke: der Top-Down-Ansatz.* 4., aktual. Aufl. München [u.a.] : Pearson Studium, 2008 (IT - Informatik). – 896 Seiten S. <http://www.gbv.de/dms/ilmenau/toc/558366325.PDF>. – ISBN 978-3-8273-7330-4. – HTWG Konstanz
- [ND18] NEUSCHWANDER, Jürgen ; DÜSTERHÖFT, Sabine: *IT-Sicherheitsarchitekturen Aufgabenstellung zum Praktikum.* March 2018
- [she] https://wiki.ubuntuusers.de/Shebang_f%C3%BCr_Shellskripte/