Fachinformatiker für Anwendungsentwicklung

Dokumentation zur betrieblichen Projektarbeit

Automatisierte Accounterstellung via AMQP-Messaging-System

mit Konsolidierung der Datenquellen (Übergabeprotokoll und Angebotssystem)

Auszubildender: Andreas Biller

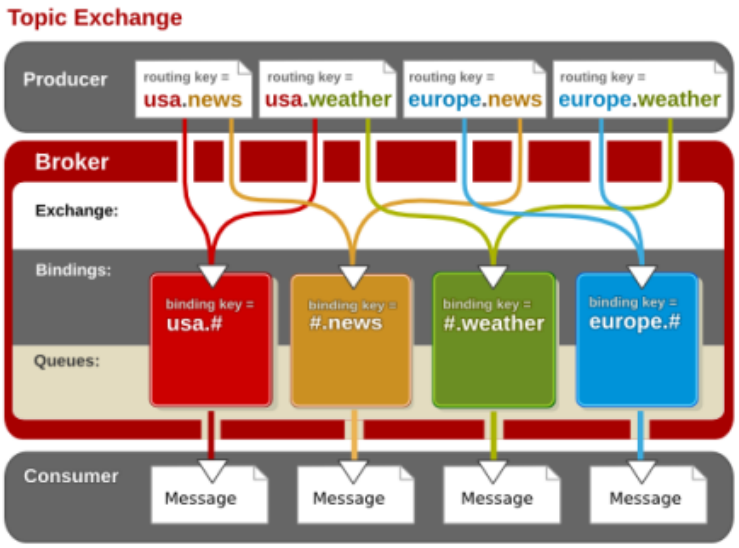


Abbildung 1: Versand von Nachrichten mittels AMQP

Abgabetermin: Berlin, den 24.05.2018

Doctena Germany GmbH
Urbanstr. 116, 10967 Berlin

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Abkürzungsverzeichnis

AMQP	Advanced Message Queuing Protocol
BASH	Bourne Again Shell
CAS	Column Access Strobe
CL	Column Access Strobe (CAS) latency
CLI	Command Line Interface
DDR4	Double Data Rate 4th-Generation
DHCP	Dynamic Host Configuration Protocol
DIMM	Dual In-line Memory Module
DMZ	Demilitarisierte Zone
DNS	Domain Name Server
FA	Fachinformatik für Anwendungsentwicklung
FTP	File Transfer Protokoll
GB	Giga Byte
GHz	Giga Hertz
GUI	Graphical User Interface
HTML	Hypertext Markup Language
ICMP	Internet Control Message Protocol
ID	Identification
IHK	Industrie- und Handelskammer
IP	Internet Protokoll
IT	Informationstechnik
ITS	Informationstechnische Systeme
LAN	Local Area Network
MB	Mega Byte
NAT	Network Adress Translation
NTP	Network Time Protocol

OSZIMT	Oberstufenzentrum Informations- und Medizintechnik
PC	Personal Computer
P/LZ	Projekt/Linux-Zertifizierung
RAM	Random Access Memory
SSD	Solid State Drive
SSH	Secure Shell
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VLAN	Virtual Local Area Network (LAN)
VM	Virtual Machine

1 Einleitung

1.1 Projektumfeld

Unternehmen: "Das Oberstufenzentrum Informations- und Medizintechnik (OSZIMT) in der Haarlemer Straße in Berlin-Britz im Bezirk Neukölln ist eines von 36 Oberstufenzentren in Berlin. Es vereint das Berufliche Gymnasium, die Berufsoberschule, die Fachoberschule, die schule, die Fachschule und die Berufsschule. (...) [An ihm] arbeiten etwa 160 Lehrkräfte und nichtpädagogisches Personal in Laboren, Werkstätten, Lernbüros und allgemeinen Unterrichtsräumen. (...) [Es] hat rund 3000 Schüler (...) [und] ist die größte Schule Berlins für Informationstechnik und Deutschlands größte Schule für Medizintechnik."[1] Wir besuchen dort seit 2 bzw. 1.5 Jahren den Unterricht der Klasse Fachinformatik für Anwendungsentwicklung (FA) 54.

Auftraggeber: Als angehende Fachinformatiker für Anwendungsentwicklung am OSZIMT sollen wir nun im Rahmen des Faches Projekt/Linux-Zertifizierung (P/LZ) ein auf mittelständige Unternehmen anwendbares Informationstechnik (IT)-Sicherheitskonzept entwickeln. Dazu werden wir im Verlauf des Projektunterrichtes eine DMZ unter Verwendung des zuvor in Informationstechnische Systeme (ITS) erlernten Wissens über Netzwerktechnik einrichten. Gleichzeitig erarbeiten wir uns Anhand eines Online-Kurses der Cisco-Networking-Academy die für das Projekt benötigten Grundkenntnisse im Umgang mit Linux. Verantwortlicher Auftraggeber und unser Ansprechpartner für dieses Projekt ist **Herr Ralf Henze**, Netzwerktechniker und Lehrer am OSZIMT in den Unterrichtsfächern ITS und P/LZ.

1.2 Projektziel

Projekthintergrund: Neben dem offensichtlichen Ziel dieses Projektes, ein DMZ-Netzwerk unter Linux einzurichten, will es uns als Teil des Berufsschulunterrichtes natürlich vor allem etwas beibringen. So ist die eigentliche Projektarbeit durchzogen von unterschwelligem Langzeitnutzen für unsere berufliche Entwicklung. Das Wissen, wie und wo man jederzeit Befehle nachschlagen kann, die beneidenswerten Möglichkeiten mit grep, pipes und kleinen Tools wie xargs erstaunlich komplizierte Probleme lösen zu können. Auch die bewusst schon fast aufs Niveau der Industrie- und Handelskammer (IHK) angehobenen Anforderungen an die Projektdokumentation und das Nahelegen, für deren Erstellung mit einer Sprache wie LaTeX zu arbeiten, anstelle dies mit gängigen Office Paketen zu tun, waren eine gute Vorbereitung und hervorragende Übung. So konnte Gelerntes durch praktisches Anwenden gefestigt und Neues sinnvoll ausprobiert werden.

[1] Porträt des OSZIMT, www.oszimt.de [2017]

Ziel des Projekts: Die eigentliche Kernaufgabe des Projektes ist die Planung und praktische Umsetzung eines grundlegenden IT-Sicherheitskonzeptes mit Hilfe eines DMZ-Netzwerkes und dessen Absicherung durch das Setzen bzw. Löschen von Firewall-Regeln über ein Shell-Script. Die demilitarisierte Zone soll zwischen den Windows-Clients des Kunden im internen Netz und den potentiell schädlichen Anfragen der restlichen Welt aus dem externen Netzwerk liegen. Hier steht auch der Windows-Webserver des Kunden, welcher sowohl von Innen (zur Wartung) wie auch von Außen (für Besucher) erreichbar sein muss. Zwei virtuelle Linuxmaschinen sollen als Router zwischen den Netzen konfiguriert werden, wobei der Äußere sowohl NAT als auch die Funktion der Firewall übernehmen soll. Planung und Umsetzung sollen umfassend Dokumentiert werden. Jedes Gruppenmitglied soll ein Kompetenzprtfolio führen, in dem er seine Kenntnisse, Gelerntes und Probleme vor, während und nach den Aufgaben der Projektarbeit sammelt und kritisch analysiert.

1.3 Projektbegründung

Nutzen des Projekts: Neben dem bereits mehrfach erwähnten Lerneffekt für uns als Schüler, sowohl in den Grundlagen der IT-Sicherheit, des Arbeitens auf dem Linux-Filesystem mit Hilfe der Command Line Interface (CLI), wie auch der Wiederholung der Befehle zur Konfiguration von Netzwerken und Schnittstellen in einer neuen leicht anderen Syntax, liegt der Projektnutzen wohl vor Allem auf dem Verstehen der Arbeitsweise von Access-Control-Listen, der Bedeutung der drei Chains sowie eines besseren Einblicks in die Welt der Linux-Distributionen, deren Stärken und Schwächen sowie deren Konfiguration. Und da das Projekt den Auftraggeber faktisch nichts kostet, uns aber fachlich weiter bringt, ist dessen Durchführung für beide Seiten ein Win-Win-Geschäft.

Motivation: Unser Auftraggeber ist daran interessiert, ein fertiges, funktionierendes System zu erhalten, welches seine Wünsche und Anforderungen erfüllt, aber er und auch wir können uns selbst an greifbaren Indikatoren unsere bisher erworbene Fachkompetenz bewerten. Wir stellen uns somit einer solchen Aufgabe, um etwas neues zu lernen, etwas zu wiederholen und uns zu verbessern. Oder einfach, weil wir es können. Manchmal auch, um uns auf eine Zertifizierung vorzubereiten.

1.4 Projektschnittstellen

Technisch gesehen interagieren in unserem Projekt zwei oder mehrere Windows-Rechner, welche über das Labornetzwerk des Raumes 3.1.01 verbunden sind. Auf beiden läuft jeweils eine Linux Debian Distribution in einer virtuellen Umgebung durch den VMWare Player. Die Schnittstellen der virtuellen Linuxdistributionen wiederum sind über den Bridged Modus in den Netzwerkeinstellungen des VMWare Players mit einer der physikalischen Netzwerkschnittstelle des Host-Personal Computer (PC)s verbunden. Über das Labornetz kann Verbindung zu den Rechnern der anderen Gruppen aufgenommen werden. Die Unterrichtszeit für das Projekt, sowie die Infrastruktur (Pro Gruppe 2 Rechner + benötigte Peripherie, 2 virtuelle Maschinen und alle sonst benötigten Ressourcen, Zugang zum Internet und ins Labornetz) und alles weitere wird uns im Rahmen des P/LZ-Unterrichtes zur

Verfügung gestellt. Dank der theoretischen Natur des Projektes sind die einzigen Benutzer unseres Projektes wir, evtl. unsere Mitschüler während des Erfahrungsaustausches untereinander, sowie unser Auftraggeber, Herr Henze, der sich immer wieder über den aktuellen Stand informiert und auch die finale Abnahme des Projektes übernimmt. Zur finalen Abnahme durch den Kunden sollen sowohl die Funktionalität der Firewall-Regeln nachweislich testbar sein, als auch die Projektdokumentation inkl. einer Kopie des verwendeten Firewall-Scriptes, den tabellarisch erfassten Testresultaten sowie je eines Kompetenzportfolios pro Gruppenmitglied zur Abgabe vorliegen.

1.5 Projektabgrenzung

Was dieses Projekt nicht bietet: Dieses Projekt will auf keinen Fall den Anspruch erheben, durch die verwendeten Techniken ein Netzwerk oder System perfekt und allumfassend vor unbefugtem Eindringen schützen zu können. Es vermittelt nur Einblicke in die Grundlagen der Netzwerktechnik und IT-Sicherheit. Ein perfektes und vor allen schädlichen Einflüssen geschütztes System kann es nicht geben. Weiterführende Informationen zur Verbesserung der Systemsicherheit können aber der im Quellverzeichnis angegebenen Literatur entnommen werden.

2 Projektplanung

Da unser Projekt über die Dauer eines ganzen Schuljahres angelegt ist und wir die Unterrichtszeit zum Teil mit dem Erlernen von Fertigkeiten im Umgang mit Linux verbringen werden, muss der Ablauf genau geplant werden. Im folgenden erläutern wir die einzelnen Projektphasen, welche Ressourcen genutzt wurden und wann die Durchführung von der Planung abgewichen ist.

2.1 Projektphasen

Im Rahmen des P/LZ Unterrichts erhalten wir in jeder Schulwoche meist Freitags für je zwei Blöcke a 90 Minuten Zugang zum Labor 3.1.01 am OSZIMT in Berlin. Das Schuljahr umfasst 14 Schulwochen in denen das Projekt durchgeführt werden muss. Außerhalb der Schulzeit können wir Private Ressourcen nutzen und planen pro Schulwoche jeweils 6 Stunden Freizeit am Wochenende als zusätzliche Pufferzeit ein. Die 42 Laborstunden und die Pufferzeit von 84 Stunden ergeben eine Gesamtzeit von 126 Stunden bis zur Projektabgabe. Wir gehen davon aus die grundlegende Planung und Analyse in den ersten beiden Schulwochen durchzuführen, die nächsten drei Schulwochen sollte das Netzwerk entworfen und erstellt werden. Anschließend wollen wir mit der Implementierung der Firewall beginnen, wofür wir ca. vier Schulwochen einplanen. Die Restliche Schulzeit wird für die Erstellung der Dokumentation und eine Stunde für die Abnahme durch den Kunden verplant. Je nach Bedarf kann die Pufferzeit zu weiterer Recherche zuhause genutzt werden.

2.2 Zeitplanung

Tabelle 1 zeigt unsere grobe Zeitplanung für die jeweils bevorstehenden Projektphasen:

Projektphase	Geplante Zeit
Analyse	6 h
Entwurf	11 h
Implementierung	39 h
Abnahme und Deployment	5 h
Dokumentation	9 h
Gesamt	**70 h**

Tabelle 1: Zeitplanung

2.3 Abweichungen vom Projektantrag

Aufgrund unserer Unerfahrenheit im Umgang mit LaTeX gestaltet sich die Erstellung der Projektdokumentation leider schwieriger als vermutet. Zudem konnten die Funktionstests an unserer Firewall nicht bis zum Ende des letzten Unterrichtsblockes abgeschlossen werden, worauf Herr Krüger viel Zeit damit verbracht hat, eine zweite Testumgebung für unser Firewall-Script mit Windows Server 2016 zu virtualisieren, deren Installation und Konfiguration im Anhang dokumentiert wurde. Deshalb erbaten wir eine kurzzeitige Verlängerung der Abgabefrist und konnten nur die während des Unterrichtes erstellte und benutzte Dokumentation einsenden, zu finden im Anhang A.1: Schritt-für-Schritt Anleitung auf Seite i.

2.4 Ressourcenplanung

Für die Durchführung im Labor werden benötigt: 2 Rechner mit Windows (und einem Benutzeraccount mit Adminrechten), die Software VMWare Player, eine Distribution von Debian für die virtuelle Maschine, Zugang zum Labornetz, ein Webserver und ein Editor zum Bearbeiten von Hypertext Markup Language (HTML), Zugang zum Internet für Recherche, Software zum Festhalten der Ergebnisse, Software zum Durchführen von Tests. Zusätzlich bedarf es der Unterstützung durch fachkundige Mitschüler wie den Herren Habekost, Schernekau und Mahnke sowie Hilfe durch Herrn Henze bei schwereren Problemen. Für die Arbeit außerhalb der Schule haben wir zur Recherche und für weitere Versuche sowohl Rechner mit Ubuntu 14.04 als auch Rechner mit Windows 7 und 10 und eigene Heimnetzwerke mit Internetanbindung. Auch die benötigte Software sowie LaTeX und Editoren um die Dokumentation anzufertigen sind vorhanden. Dank einer während des Projektes angelegten Schritt-für-Schritt Anleitung zum Einrichten des Netzwerks, sowie der Möglichkeit virtuelle Maschinen zu kopieren bzw. das Versuchsnetzwerk selbst zu virtualisieren, kann auch zuhause gearbeitet werden.

2.5 Entwicklungsprozess

Um unser Projekt durchzuführen benutzen wir einen auf dem Wasserfallmodel basierenden Entwicklungsprozess und den üblichen Stufen Anforderung, Entwurf, Implementation, Überprüfung und Wartung.

3 Analysephase

Im Nachfolgenden verzichten wir auf einen Großteil der üblichen Berechnungen zur Wirtschaftlichkeit des Projektes, da dieses zum Großteil unserer fachlichen Kompetenzbildung dienen soll. Darüber hinaus wäre für ein fiktives mittelständisches Unternehmen ein bereits existierendes Produkt sowohl vom zu erwartenden Arbeitsaufwand wie auch finanziell deutlich günstiger. Es wird daher lediglich eine Beispielhafte Kostenberechnung für die Umsetzung der Planung durch uns erstellt und dafür ein größeres Augenmerk auf Anforderungen und Nutzen des Projekts gelegt.

3.1 Ist-Analyse

Was ist vorhanden: Im Labor sind für jedes Gruppenmitglied vorhanden: ein Bildschirmarbeitsplatz, Windows 7, Admin-Rechte, zwei physikalische Netzwerkinterfaces, Anschluss an Labornetzwerk und Internet, die Software VMWare Player, Debian Images auf einem Netzlaufwerk sowie ein portabler Miniwebserver.

Was ist zu erstellen: Zuerst muss nun von jeder Gruppe ein Netzplan erstellt werden. Dann gilt es, die Debian 7 (Wheezy) Linux-Images in virtuellen Maschinen auf beiden Rechnern mit Hilfe des VMWare Players aufzusetzen. Diese werden zu einem Outside- und einem Inside-Router konfiguriert und die geplanten Netzwerk- und Routing-Einstellungen müssen sowohl an den virtuellen wie auch physikalischen Schnittstellen durchgeführt werden. Auf dem Rechner des Outside-Routers muss ein Webserver eingerichtet werden, wofür NAT und Port-Forwarding nötig sind. Zwischendurch wird es immer wieder der gezielten Recherche bedürfen. Um schließlich Zugriffe von außen zu regulieren, muss eine Firewall mit entsprechenden Regeln erstellt werden, die per Skript an- und abschaltbar ist. Die Funktionalität muss getestet werden und Projekt und Tests sind zu dokumentieren. Unser Lernfortschritt ist in einem Kompetenzportfolio niederzuschreiben. Gleichzeitig sind Laborübungen und Tests zu Linux-Kenntnissen zu absolvieren.

3.2 Wirtschaftlichkeitsanalyse

Wie bereits Anfänglich erwähnt, lohnt sich das Projekt für ein fiktives mittelständisches Unternehmen nur bedingt.

3.2.1 „Make or Buy“-Entscheidung

Die Kosten für eine qualifizierte Kraft zur ständigen Wartung des Servers, die durch Dauerbetrieb anfallenden Stromkosten sowie die zusätzlichen Hardwarekosten bei einem zukünftigen Up-Scaling übersteigen bei weitem die Kosten für einen fachkundig und sicher Administrierten Server bei einem seriösen Hosting-Anbieter. Da unsere Empfehlung an den Kunden ein Produkt eines anderen Anbieters wäre, wird das Projekt nur zu unserem Nutzen und der Erfahrung willen, die wir damit gewinnen, umgesetzt.

3.2.2 Projektkosten

Da es sich nur um ein fiktives Projekt handelt, verzichten wir auf eine detaillierte Berechnung mit Stromkosten innerhalb des Labors, den Gehältern der Lehrkräfte oder etwaiger Lizenzgebühren. Wir beschränken uns auf eine fiktive Beispielrechnung mit unserem Stundenlohn während der Projektdauer.

Beispielrechnung (verkürzt): Die realen Kosten für die Durchführung des Projekts setzen sich sowohl aus Personal-, als auch aus Ressourcenkosten zusammen. Wir rechnen hier lediglich mit dem fiktiven Gehalt eines Auszubildendem im zweiten Lehrjahr von ca. 800 € Brutto pro Monat.

$$3 \cdot 800\,€/\text{Monat} \div 13 \div 40\ \text{h/Monat} \approx 4{,}62\,€/\text{h} \tag{1}$$

Es ergibt sich also ein Stundenlohn von 4,62 €. Die Durchführungszeit des Projekts beträgt 42 Stunden. Die Nutzung von Ressourcen[2] sowie die Kosten durch andere Mitarbeiter werden hier nicht mit eingerechnet. Eine Aufstellung der Kosten befindet sich in Tabelle 2 und sie betragen insgesamt 388,08 € für zwei Entwickler bei 42 h/Monat Arbeitszeit und je einem Gehalt von 800 € monatlich.

Vorgang	Zeit	Kosten pro Stunde	Kosten
Entwicklungskosten	42 h	4,62 € · 2 = 9,24 €	388,08 €
			388,08 €

Tabelle 2: Kosten für 2 Entwickler/Monat

3.2.3 Amortisationsdauer

Aufgrund unserer „Make or Buy“-Entscheidung und da das Projekt nur zu Lernzwecken umgesetzt wird verzichten wir hier auf die Berechnung eines fiktiven Rentabilitätszeitpunktes. Das gelernte wird spätestens zur IHK-Prüfung und bei der Anfertigung der Dokumentation des IHK-Abschlussprojektes auszahlen.

[2] Räumlichkeiten, Arbeitsplatzrechner etc.

3.3 Nutzwertanalyse

Durch den Aufbau einer DMZ können wir die Zugriffe auf unsere Server, in diesem Fall ein einfacher Webserver, von Außen und Innen reglementieren. So wird über den Routern mit einer konfigurierten Firewall ein sicherer Zugang zu unserem Webserver ermöglicht. Die Aufteilung in unterschiedliche Netzwerke ermöglicht den Administratoren eine einfachere Verwaltung der Berechtigungen für die Mitglieder des Firmennetzes.

3.4 Qualitätsanforderungen

Der Webserver soll von Außen (über die öffentliche Internet Protokoll (IP)-Adresse des Outside-Routers) und Innen erreichbar, aber vor unbefugten Zugriffen potentieller Angreifer mit den uns zur Verfügung stehenden Mitteln geschützt werden. Es muss also sichergestellt werden, dass kein unberechtigter Dritter administrativen Zugriff auf die Geräte und deren Konfiguration hat. Dabei ist darauf zu achten, dass die Mitarbeiter mit entsprechender Berechtigung (also zum Beispiel von einem Admin-PC aus dem inneren Netz aus) weiterhin Zugriff auf das Internet und den Webserver in der DMZ haben.

3.5 Lastenheft

Einen genaueren Überblick über die festgestellten Anforderungen an die einzelnen Teile der DMZ findet sich in unserem ausführlichen Lastenheft im Anhang A.2: Lastenheft auf Seite vi. Im groben gliedern sich die Anforderungen jedoch in folgende generelle Bereiche:

Die Mitarbeiter sollen untereinander, mit dem Webserver und dem Internet kommunizieren können, dabei jedoch bestmöglich geschützt werden.

Die Administrator sollen zusätzlich die Möglichkeit haben, die Server und Router aus der Ferne zu warten. Dabei sollte es unerheblich sein, wie viele Clients und Server sich im internen bzw. DMZ-Netz befinden.

3.6 Zwischenstand

Tabelle 3 zeigt den Zwischenstand nach der Analysephase.

Vorgang	Geplant	Tatsächlich	Differenz
1. Analyse des Ist-Zustands	3 h	3 h	
2. Zeit- und Ressourcenplanung	2 h	3 h	+1 h
3. Wirtschaftlichkeit und „Make or buy“-Entscheidung	2 h	1 h	-1 h
4. Qualitätsanforderungen und Lastenheft	3 h	3 h	
5. Beginn Dokumentation und Kompetenzportfolio	2 h	3 h	+1 h
Analysephase	**12 h**	**13 h**	**+1 h**

Tabelle 3: Zwischenstand nach der Analysephase

4 Entwurfsphase

Da Hard- und Software von unserem Auftraggeber gestellt und vorgegeben wird, erübrigt seine ausführliche Begründung, weshalb wir diese Materialien verwendet haben. So wird sichergestellt, dass während unserer Projektzeit allen die gleichen benötigten Mittel zur Verfügung stehen.

4.1 Zielplattform

Hardware: Die uns zur Verfügung stehenden Desktop PCs bleiben unverändert. Die Leistungsdaten derer genügen für den Aufbau einer einfachen DMZ.

Software: Für die Implementation eines Routers als virtuelle Maschine nutzen wir den vorinstallierten VMWare Player. Dieser ist kostenlos und berechtigt uns zum Virtualisieren einer Linux Distribution. Des Weiteren werden wir auch das beigefügte Debian benutzen. Auf den Virtual Machine (VM)s wird mit Bourne Again Shell (BASH) und Linux-Befehlen gearbeitet, da wir nur kleinere Konfigurationen und Scripts schreiben. Um die Konfiguration zu testen, die Router per Remote zu konfigurieren und eventuell Dateien auszutauschen, wird noch Secure Shell (SSH)- und File Transfer Protokoll (FTP)-Client-Software benötigt. Dafür werden wir Putty und winscp verwenden. Diese Tools sind kompakt und beeinträchtigen nicht die Leistung der Hosts.

4.2 Netzwerkplan

Die im Anhang A.4: Netzpläne auf Seite viii zu findenden Netzpläne zeigen die grundsätzliche IP-Adressverteilung in den geplanten Netzen unseres Projektes. Der zweite Netzplan zeigt die erweiterte Testumgebung die wir gegen Ende des Projekts zuhause einrichten mussten, um die Tests an der Firewall zu beenden. Unser Netz teilt sich gleichfalls jeweils in das Labornetz (hier auch symbolisch für die Cloud, das Internet, etc.. stehend), das von der Außenwelt abgeschottete interne Netz (mit den Windows-Clients und dem Admin-Rechner unseres Kunden) und das als Pufferzone dazwischen liegende DMZ-Netzwerk, welches zur Absicherung des internen Netzes nur über spezielle Berechtigungen zu erreichen und für spezielle Dienste (Webserver) zu verwenden ist.

4.3 Maßnahmen zur Qualitätssicherung

Bei jeder Veränderungen der Konfiguration werden Funktionstests durchgeführt. Diese sollen gewährleisten, dass die Anforderungen aus dem Lastenheft eingehalten werden. Vorgenommene Änderungen an der Firewall und der Systemkonfiguration werden in unserer vorläufigen Dokumentation, zu finden im Anhang A.1: Schritt-für-Schritt Anleitung auf Seite i, notiert und das Firewall-Script wird separat auf einem externen Datenträger gespeichert. So wird sichergestellt, dass auch bei einem Defekt eines der virtuellen Linux-Router die ursprüngliche Konfiguration schnell wieder von Null auf herstellbar ist und möglichst keine Downtime bei der Arbeit entsteht.

4.4 Pflichtenheft

Die aus den zuvor im Lastenheft gesammelten Punkte hervorgehenden Anforderungen werden im Pflichtenheft genauer in bevorstehende Aufgaben übersetzt. Dieses ist im Anhang A.3: Pflichtenheft auf Seite vii zu finden.

4.5 Zwischenstand

Tabelle 4 zeigt den Zwischenstand nach der Entwurfsphase.

Vorgang	Geplant	Tatsächlich	Differenz
1. Erstellen des Netzwerkplanes	1 h	1 h	
2. Qualitätssicherung	1 h	2 h	+1 h
3. Erstellen des Pflichtenhefts	1 h	4 h	+3 h
4. Dokumentation erweitern	2 h	1 h	-1 h
Entwurfsphase	**5 h**	**8 h**	**+3 h**

Tabelle 4: Zwischenstand nach der Entwurfsphase

5 Implementierungsphase

Dank der von der Schule zur Verfügung gestellten Hard- und Software im Labor 3.1.01 erfolgt die im folgenden genauer dargestellte Installation der virtuellen Router relativ problemlos. Dies muss jedoch immer wieder durch Testen des bisherigen Fortschritts verifiziert werden.

5.1 Implementierung der Virtuellen Maschinen

Eine Debian Distribution als virtuelle Maschine ist bereits auf beiden Rechnern vorhanden. Diese wird kopiert und dann mit dem VMWare Player gestartet. Wir überbrücken die physischen Netzwerkadapter der Windows-Hosts auf die virtuellen Adapter der Linux Distribution. So haben die designierten Router über die physischen Interfaces Zugriff auf das Netzwerk.

5.2 Konfiguration der Router

Über dem VMWare Player auf den Windows Hosts verbinden wir uns auf die Router und können diese dann über das Terminal konfigurieren. Die Passwörter, die wir vom Kunden erhalten haben, lassen wir unverändert. Als erstes werden die Hostnamen angepasst. Dazu ersetzt man den alten Namen in den Dateien `/etc/hostname` und `/etc/hosts`. Danach sollte die Maschine neu gestartet werden. Diese und alle weiteren von uns benötigten Dateien lassen sich über einen vorinstallierten Editor öffnen und bearbeiten, z. B. mit vi:

```
vi /etc/hostname
```

5.2.1 Konfiguration der Interfaces

Für die Konfiguration der Interfaces halten wir uns an den erstellten Netzpläne. Um die Interfaces zu konfigurieren editieren wir jeweils deren Konfiguration in der Datei `/etc/network/interfaces`.

Inside-Router Für den Inside-Router tragen wir neben den IP-Adressen seiner Schnittstellen als Standard-Gateway das Interface des Outside-Routers ein, welsches sich in der DMZ befinden soll. (Siehe Anhang InideRouterInt.png)

Outside-Router Der Outside-Router erhält zusätlich zu seinen IP-Adressen als Gateway die IP-Adresse 192.168.200.1 (Standard-Gateway Labornetz). (Siehe Anhang OutsideRouterInt.png)

5.2.2 Konfiguration der statischen Routern

Wir benötigen zwei statische Routen auf dem Outside-Router, eine für die DMZ und eine für das LAN. (Siehe Anhang OuutsideRouterInt.png)

5.2.3 Konfiguration von NAT und Port-Forwarding

Weiterhin konfigurieren wir in der *interfaces* Datei vom Outside-Router NAT für die DMZ und das LAN sowie Port-Forwarding zu unserem Webserver ein. (Siehe Anhang OuutsideRouterInt.png) Um jedoch NAT und Port-Forwarding auf beiden Routern nutzen zu können, müssen wir dies erst aktivieren. Dies geschieht mit dem Befehl `echo 1 > /proc/sys/net/ipv4/ip_forward}`.

Dies ist jedoch nur eine temporäre Lösung und geht nach einem Neustart verloren. Damit der Prozess mit dem Systemstart geladen wird, setzen wir den Wert von `#net.ipv4.ip_forward` in der Datei `/etc/sysctl.conf` auf `1` und entfernen den Kommentar vom Anfang dieser Zeile.

5.2.4 Konfiguration des DNS-Servers

In der Datei `/etc/resolv.conf` tragen wir für beide die IP-Adresse der von unserem Auftraggeber bereitgestellten DNS-Server ein.

```
nameserver 192.168.200.40
nameserver 192.168.200.41
```

5.2.5 Konfiguration des Zeitserver

Um einen Zeitserver angeben und nutzen zu können, installieren wir mit `apt-get install ntp` den Network Time Protocol (NTP)-Dienst. Danach fügen wir die IP-Adresse des bereitgestellten NTP-Servers (Standard-Gateway) in die Datei `/etc/ntp.conf` ein: `server 192.168.200.1 iburst`. (Siehe ntp.conf)

5.3 Implementierung der physischen Hosts

Bevor die Schnittstellen auf die Router angepasst werden, werden noch evtl. benötigte Dateien und Programme (webserver, notepad++, putty, winscp) heruntergeladen. Im Gegensatz zu Router-Konfiguration wird hier fast ausschließlich mit der Graphical User Interface (GUI) gearbeitet.

5.3.1 Konfiguration der Interfaces

Für die IP-Adressierung halten wir uns ebenfalls an den Netzpläne.

Admin-PC Der für die spätere Verwaltung der Router und des Webserver zuständige Host, befindet sich im LAN und erhält als Gateway den Inside-Router (Siehe Anhang AdminPCInt.png)

Webserver Der Webserver befindet sich in der DMZ und erhält als Gateway den Outside-Router. (Siehe Anhang WebserverInt.png)

5.3.2 Konfiguration des Webservers

Auf dem Host in der DMZ wird ein einfacher Webserver, welcher über Port 80 kommuniziert, ausgeführt. Durch das Anpassen der Datei *index.html* wird die Website entsprechend des Kundenwunsches angepasst.

5.3.3 Konfiguration der Windows-Firewall

Um auf den Hosts die Firewall testen und einen DNS-Server nutzen zu können muss die Windows-Firewall noch dementsprechend angepasst werden. Dazu ist es nötig die Anpassungen für sowohl die ein- als auch ausgehenden Regeln vorzunehmen. Damit wir einen *ping*-Befehl absetzen können, ist es nötig die Regel für die *Datei- und Druckerabfrage* für Internet Control Message Protocol (ICMP)v4 zu aktivieren. Für die Kommunikation zum DNS-Server erstellen wir zwei Regeln, je eine für das Transmission Control Protocol (TCP)- bzw. das User Datagram Protocol (UDP)-Protokoll. Darin erlauben wir die Kommunikation über die Ports 53 und 853.

5.3.4 Konfiguration des Zeitservers

Die IP-Adresse des Zeit-Server tragen wir in den "Datum und Uhrzeiteinstellungenünter der Registerkarte Ïnternetzeitëin.

5.4 Konfiguration der Firewall

Dass durch den Auftraggeber vorgegebene Script wird entsprechend der in sich befindlichen Vorlage auf beiden Routern angepasst und die DMZ somit von beiden Seiten abgeschottet. Entsprechend des übergebenen Parameters (`start`, `stop`) wird das BASH-Script gestartet bzw. geschlossen. Wird die Outside-Firewall gestoppt, existiert eine uneingeschränkte Verbindung zwischen dem Labornetz und der DMZ. Das interne Netz ist weiterhin durch den Inside-Router geschützt. Ist die Inside-Firewall gestoppt, sind die Netze weiterhin durch den Outside-Router geschützt. Der Inside-Router ist nun jedoch aus dem internen Netz frei erreichbar. Des weiteren schreibt die Firewall ihre Einstellungen zum jeweiligen Zustand, wenn Sie gestartet bzw. gestoppt wird in eine Log-File. Diese befindet sich im Ordner `/var/log/firewall/firewallConfig`. Genauere Angaben zu den finalen Firewall-Scripten finden sich im Anhang B.2.1: firewall.sh (auf dem Outside-Router) auf Seite xviii sowie im Anhang B.2.2: firewall.sh (auf dem Inside-Router) auf Seite xxvi.

5.5 Zwischenstand

Tabelle 5 zeigt den Zwischenstand nach der Implementierungsphase.

Vorgang	Geplant	Tatsächlich	Differenz
1.Einrichten der VMs	1 h	1 h	
2. Konfiguration der Router	4 h	3 h	-1 h
3. Einrichtung von NAT und Portforwarding	6 h	4 h	-2 h
4. Einrichtung physische Hosts	1 h	1 h	
5. Einrichten des Webservers	3 h	1 h	-2 h
6. Erstellen der Webserver-Startseite	1 h	2 h	+1 h
7. Konfiguration der Firewall	30 h	28 h	-2 h
8. Qualitätssicherung	3 h	1 h	-2 h
9. Dokumentation erweitern	12 h	30 h	+18 h
Implementierungsphase	**61 h**	**71 h**	**+10 h**

Tabelle 5: Zwischenstand nach der Implementierungsphase

6 Abnahmephase

Der Zugang zum Webserver ohne aktivierte Firewall konnte hier bereits zum Halbjahr bei Abnahme der Funktionalität des zugrundeliegenden Netzwerkes durch unseren Auftraggeber festgestellt werden. Eine HTML-Seite mit Stand des aktuellen Projektfortschritts wurde mit Bootstrap selbst für mobile Endgeräte optimiert. Sie zeigte neben verschiedenen Gruppeninformationen auch den Netzplan und die vorläufige Dokumentation zusammen mit einer einfachen Liste aus roten und grünen Buttons für jede Projektanforderung. Somit war daraus einfach ersichtlich, welche der Aufgaben bereits erfüllt werden konnten. Da es nach der nur bei einigen Gruppen stichprobenartig durchgeführten finalen Abnahme durch Herrn Henze nur noch die Abgabe der Dokumentation vor Ende des Projektes gibt, jedoch keinen real existierenden Kunden, bei dem die entworfene DMZ umgesetzt werden soll, wird die Einführungsphase aus der weiteren Projektbeschreibung entfallen. Eine Beispielhafte Implementierung kann jedoch auch der Testdokumentation entnommen werden.

Aufbau einer virtuellen Testumgebung:

Da die Originalmaschinen zum Testzeitpunkt nicht mehr verfügbar waren, wurde hierzu eine eigene Testumgebung mittels HyperV nachgestellt. Genauere Angaben über die Teststellung finden sich im zweiten Netzplan A.4, den ausführlichen Testprotokollen B.1.5 und einer Dokumentation des virtuellen Testsystems B, alles zusammen zu finden im Anhang B: Testdokumentation auf Seite xii.

6.1 Zwischenstand

Tabelle 6 zeigt den Zwischenstand nach der Abnahmephase.

Vorgang	Geplant	Tatsächlich	Differenz
1. Abnahmetest der Fachabteilung	1 h	0 h	-1 h

Tabelle 6: Zwischenstand nach der Abnahmephase

7 Dokumentation

Da unser Auftraggeber bereits früh im Projekt seine Vorliebe nach einer, den IHK-Richtlinien entsprechend umgesetzten Dokumentation Ausdruck verlieh, beschlossen wir uns, seinem Wunsch zu entsprechen. So wurde die finale Dokumentation in LaTeX erstellt. Das Ergebnis mag sich zwar sehen lassen, dennoch schlägt die Bearbeitung der Dokumentation dank der aufgetretenen Schwierigkeiten im Umgang mit LaTeX mit einem zu hohen Anteil des Zeitbudgets zu Buche. Nichtsdestotrotz hier das beschriebene Resultat. Wir hoffen, es war die Mühen wert. Zu Ihrer Erstellung wurden zusätzlich folgende Webseiten zu Hilfe gezogen: texstudio.sourceforge.net [2017], www.latex tutorial.com [2015] und vor allem www.debian.org [2015]

Entwicklerdokumentation: Die der neben der Konfiguration angelegte Entwicklerdokumentation befindet sich im Anhang A.1: Schritt-für-Schritt Anleitung auf Seite i. Sie wurde als Schritt-für-Schritt-Anleitung zum Wiederherstellen des bereits erreichten Zustandes im Fall eines technischen Versagens geführt. Sie wurde basierend auf Informationen aus folgenden Webseiten erstellt: www.debian.org [2017a], www.debian.org [2017b], www.netfilter.org [2002b], www.frozentux.net [2006] und www.netfilter.org [2002a]

7.1 Zwischenstand

Tabelle 7 zeigt den Zwischenstand nach der Dokumentation.

Vorgang	Geplant	Tatsächlich	Differenz
1. Erstellen des Kompetenzportfolios	3 h	3 h	
2. Erstellen der Projektdokumentation	12 h	40 h	+28 h
3. Erstellen der Testdokumentation	12 h	20 h	+6 h
Dokumentation	**27 h**	**63 h**	**+34 h**

Tabelle 7: Zwischenstand nach der Dokumentation

8 Fazit

Obwohl die Tests der Firewall am letzten Projekttag im Labor 3.1.01 nicht mehr rechtzeitig durchgeführt werden konnten, und die Erstellung der Dokumentation mit LaTeX sich als schwieriger und langwieriger als Angenommen darstellte, sind wir mit unserem Ergebnis durchaus zufrieden. Die Dokumentation ist noch zu umfangreich und Stellenweise nicht ganz ausgearbeitet, doch haben wir gerade durch sie einiges gelernt. Dies zeigt sich auch im folgenden Überblick des finalen Standes.

8.1 Soll-/Ist-Vergleich

Durch ein Firewall-Script sowohl auf dem Inside- wie auch auf dem Outside-Router der DMZ schließen wir zusätzliche Sicherheitslücken in unserem System. Und durch das nachträgliche Testen in einer mit Windows Server 2016 virtualisierten Netzwerkumgebung haben wir die im Projekt erlernten Fähigkeiten erfolgreich auf ein weiteres System portiert und so hoffentlich auch gleich gefestigt. So haben wir viel über Netzwerke, Firewall-Regeln, NAT und Port-Forwarding mithilfe von iptables, Linux im allgemeinen, dessen grundsätzliche Verzeichnisstruktur, dem Arbeiten im Terminal, sowie zusätzlich den Umgang mit LaTeX zur Anfertigung einer Projektdokumentation, die von der IHK geforderten Richtlinien dazu, sowie dem Arbeiten mit virtualisierten Netzwerken gelernt. Leider konnten wir dies nicht ganz im Rahmen des gegebenen Zeitbudgets tun, daher wissen wir noch nicht, wie zufrieden unser Auftraggeber mit den erbrachten Leistungen sein wird, oder ob die Überschreitung der Abgabefrist sich als schlechte Schulnote widerspiegeln wird. Uns ist durchaus bewusst, das in einem nicht akademischen Umfeld eine Verzögerung des Projektes zusätzliche Kosten bedeutet hätte. Allerdings sind wir auch der Meinung, eine entsprechend große Gegenleistung an Wissen und Erfahrung durch dieses Projekt gewonnen zu haben, um die Abweichung vom vorher geplanten Projektrahmen zu rechtfertigen. Wie in Tabelle 8 noch einmal genau zu erkennen ist, konnte die Zeitplanung bis auf wenige Ausnahmen, einige davon jedoch aus bereits unter Dokumentation erwähnten Gründen mit gravierend abweichenden Zeiten, eingehalten werden (falls der Auftraggeber bei unserer verspäteten Abgabe nochmal beide Augen zudrückt).

Phase	Geplant	Tatsächlich	Differenz
Analysephase	12 h	13 h	+1 h
Entwurfsphase	5 h	8 h	+3 h
Implementierungsphase	61 h	71 h	+10 h
Abnahmetest der Fachabteilung	1 h	0 h	-1 h
Erstellen der Dokumentation	27 h	63 h	+34 h
Pufferzeit	20 h	0 h	-2 h
Gesamt	**126 h**	**155 h**	**+45 h**

Tabelle 8: Soll-/Ist-Vergleich

8.2 Lessons Learned

Wir haben jedenfalls gelernt uns eine realistischere Zeitplanung für kommende Projekte zu erstellen und wissen nun auch, welch ein enormer Aufwand eine Dokumentation im Rahmen der IHK-Richtlinien darstellt. Wir haben auch den Vorteil einer schon während der Arbeit vorhandenen Doku zu schätzen gelernt und haben nun sowohl unter Linux wie auch Windows einiges über die Konfiguration von Netzwerken und Firewalls sowie das Arbeiten mit virtuellen Netzen verinnerlicht.

8.3 Ausblick

Obwohl das Projekt beendet ist, können wir die virtualisierte Testumgebung benutzen, um weitere Übungen daran durchzuführen. So wollen wir zum Beispiel in Zukunft unser Testnetzwerk noch ausbauen und es um Domain-Controller, Dynamic Host Configuration Protocol (DHCP)-, DNS-, FTP- und Exchange-Server erweitern. Ein individueller, ausführlicherer Ausblick auf unsere weiteren Vorhaben kann jeweils unserem Kompetenzportfolio entnommen werden, welche im Anhang A.5: Kompetenzportfolios auf Seite ix zu finden sind.

Literaturverzeichnis

blog.workinghardinit.work 2017

BLOG.WORKINGHARDINIT.WORK, Install Windows Server 2016 Network D. HowTo: *Didier Van Hoye, Installing Intel I211, I217V, I218V and I219V drivers on Windows Server 2016 with EUFI boot.* 2017. – https://blog.workinghardinit.work/2017/06/19/installing-intel-i211-i217v-i218v-i219v-drivers-windows-server-2016-eufi-boot/, Aufgerufen 2017-06-27

texstudio.sourceforge.net 2017

TEXSTUDIO.SOURCEFORGE.NET, User M. Manual: *TexStudio Manual, User Manual.* 2017. – http://texstudio.sourceforge.net/manual/current/usermanual_en.html, Aufgerufen 2017-06-07

www.latex tutorial.com 2015

TUTORIAL.COM, Bibtex www.latex Tutorial: *Bibliography in LaTeX with Bibtex/Biblatex.* 2015. – https://www.latex-tutorial.com/tutorials/beginners/latex-bibtex/, Aufgerufen 2017-06-30

www.debian.org 2017a

WWW.DEBIAN.ORG, Chapter 1. Manual: *Debian Manual, Chapter 12: Programming.* 2017. – https://www.debian.org/doc/manuals/debian-reference/ch12.en.html, Aufgerufen 2017-06-10

www.debian.org 2017b

WWW.DEBIAN.ORG, Chapter 5. Manual: *Debian Manual, Chapter 5: Network Setup.* 2017. – https://www.debian.org/doc/manuals/debian-reference/ch05.en.html, Aufgerufen 2017-06-10

www.debian.org 2015

WWW.DEBIAN.ORG, Securing D. HowTo: *Debian Manual, Detailed user guide for securing and hardening of the default Debian installation.* 2015. – http://www.debian.org/doc/manuals/securing-debian-howto/, Aufgerufen 2017-06-12

www.frozentux.net 2006

WWW.FROZENTUX.NET, Iptables Tutorial: *Oskar Andreasson, Iptables Tutorial 1.2.1.* 2006. – https://www.frozentux.net/iptables-tutorial/chunkyhtml/index.html, Aufgerufen 2017-06-14

www.netfilter.org 2002a

WWW.NETFILTER.ORG, NAT HowTo: *Rusty Russell, Linux 2.4 NAT HOWTO.* 2002. – http://www.netfilter.org/documentation/HOWTO/NAT-HOWTO.html, Aufgerufen 2017-06-14

www.netfilter.org 2002b

WWW.NETFILTER.ORG, Packet F. HowTo: *Rusty Russell, Linux 2.4 Packet Filtering HOW-*

TO. 2002. – http://www.netfilter.org/documentation/HOWTO/packet-filtering-HOWTO-7.html, Aufgerufen 2017-06-14

www.oszimt.de 2017
WWW.OSZIMT.DE, Porträt des Ṗressemappe: *Porträt des OSZIMT (Pressemappe)*. 2017. – http://www.oszimt.de/ueber-uns/presse/pressemappe/portraet.html, Aufgerufen 2017-06-11

Eidesstattliche Erklärung

Ich, Andreas Biller, versichere hiermit, dass ich meine **Dokumentation zur betrieblichen Projektarbeit** mit dem Thema

> ***Automatisierte Accounterstellung via AMQP-Messaging-System***
> *mit Konsolidierung der Datenquellen (Übergabeprotokoll und Angebotssystem)*

selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, wobei ich alle wörtlichen und sinngemäßen Zitate als solche gekennzeichnet habe. Die Arbeit wurde bisher keiner anderen Fachkraft vorgelegt und auch nicht veröffentlicht.

Berlin, den 24.05.2018

_____

Andreas Biller

A Anhang

A.1 Schritt-für-Schritt Anleitung

FA54 P / LZ Herr Henze Gruppe 9 Andreas Biller, Rico Krüger

Thema: Aufbau einer DMZ

1. Aufsetzen der virtuellen Maschinen

Auf zwei Clients je eine virtuelle Maschine mit Linux-OS (Debian) aufsetzen (mit VM-Ware Player). Falls VM bereits vorhanden, diese in eigenen Benutzer-Ordner kopieren. Sonst über Linux mit VM-Ware Player installieren.

Rolle	Name	Passwort
Benutzer	user	oszimt
Administrator	root	osz

2. Änderung des Modus der Netzwerkschnittstellen

Wir öffnen VM-Ware Player und starten Linux. Dann versetzen wir in den Einstellungen die Netzwerkschnittstellen in den **Bridge-Modus**.

3. Erstellung Netzwerkplan

Wir erstellen einen Netzplan und vergeben die benötigten IP-Adressen.

4. Konfiguration Schnittstellen und NAT der Linux-VMs als Router

Die Schnittstellen werden auf beiden Debian-Systemen in der Datei ***„/etc/network/interfaces“*** konfiguriert.

4.1. Konfiguration Inside-Router

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
address 10.0.9.1
netmask 255.255.255.0

# The second interface
allow-hotplug eth1
iface eth1 inet static
address 172.16.9.2
netmask 255.255.255.0
gateway 172.16.9.1
```

4.2. Konfiguration Outside-Router

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
address 172.16.9.1
netmask 255.255.255.0

# second interface
allow-hotplug eth1
iface eth1 inet static
address 192.168.200.109
netmask 255.255.255.0
gateway 192.168.200.1

### static routing ###
post-up route add -net 10.0.9.0 netmask 255.255.255.0 gw 172.16.9.2
pre-down route del -net 10.0.9.0 netmask 255.255.255.0 gw 172.16.9.2

### NAT and Port-Forwarding ###
```

FA54 P / LZ Herr Henze Gruppe 9 Andreas Biller, Rico Krüger

Thema: Aufbau einer DMZ

```
post-up iptables -A FORWARD -o eth1 -s 172.16.9.0/24 -m conntrack --ctstate NEW -j ACCEPT
post-up iptables -A FORWARD -o eth1 -s 10.0.9.0/24 -m conntrack --ctstate NEW -j ACCEPT
post-up iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE

post-up iptables -A PREROUTING -t nat -i eth1 -p tcp --dport 80 -j DNAT --to-destination 172.16.9.3:80
post-up iptables -A FORWARD -p tcp -d 172.16.9.3 --dport 80 -j ACCEPT
post-up iptables -A POSTROUTING -t nat -s 172.16.9.3 -o eth1 -j MASQUERADE
```

5. Aktivierung IP-Forwarding

Temporäre Aktivierung:
Ausführen des Befehls: `echo „1“ > /proc/sys/net/ipv4/ip_forward`

Permanente Aktivierung:
In der Datei *„/etc/sysctl.conf“* den Wert von ***„#net.ipv4.ip_forward“*** auf ***1*** setzen und die Auskommentierung aufheben: `net.ipv4.ip_forward=1`

6. Neustarten der Schnittstellen zum Übernehmen der Konfiguration

Dafür werden folgende Befehle nacheinander ausgeführt:

```
ifdown eth0
ifdown eth1
ifup eth0
ifup eth1
```

7. Konfiguration der physikalischen Netzwerk-Schnittstellen der Windows-Clients

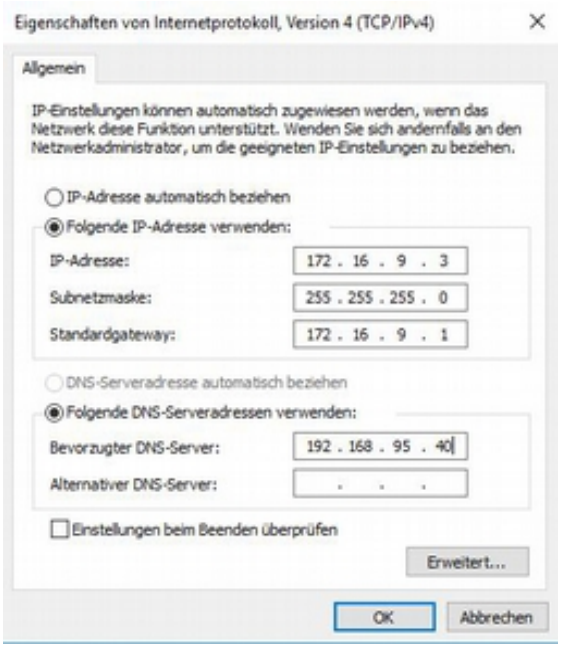
Die physikalischen Schnittstellen der Hosts von den beiden Linux-VMs werden über *„Systemsteuerung“ -> „Netzwerk- und Freigabecenter“ -> „Adaptereinstellungen ändern“ -> „Ethernet-Adapter“ -> Eigenschaften -> „Internetprotokoll, Version 4 (TCP/IPv4)“ -> „Eigenschaften“* geändert.

7.1. Konfiguration Host Inside-Router

FA54	P / LZ	Herr Henze	Gruppe 9	Andreas Biller, Rico Krüger

Thema: Aufbau einer DMZ

7.2. Konfiguration Host Outside-Router



8. Deaktivierung der Windows-Firewall

Firewall auf den Windows-Clients deaktivieren.

9. Bereitstellung des Webservers

Auf dem physischen Host des Outside-Routers wird ein einfacher Webserver auf Port 80 gestartet.
Index.htm in das Root-Verzeichnis des Webservers kopieren / aktualisieren.

10. Testen der Konfigurationen

- Zugriff auf das Internet vom Client aus dem Inside-Netz testen.
- Zugriff auf das Internet vom Client aus dem Outside-Netz testen
- Zugriff auf den Webserver aus dem Inside- und Labornetz (192.168.200.0/24) testen.

11. Einrichten der Firewall

Outside-Router:
Wir erstellen mit `mkdir /root/bin` den Ordner, wechseln dorthin und erstellen `touch firewall.sh` im Ordner **/root/bin/** als root folgendes **firewall.sh** Script und machen dieses mit `chmod 700 firewall.sh` ausführbar:

```
#!/bin/sh
case "$1" in
stop)
  echo
  echo "Stopping Firewall..."
  echo
  iptables -F
  iptables -P INPUT ACCEPT
```

FA54 P / LZ Herr Henze Gruppe 9 Andreas Biller, Rico Krüger

Thema: Aufbau einer DMZ

```
  iptables -P FORWARD ACCEPT
  iptables -P OUTPUT ACCEPT
  ;;
start)
  echo
  echo "Starting Firewall..."
  echo
  iptables -A OUTPUT -p icmp --icmp-type 8 -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT
  iptables -A INPUT -p icmp --icmp-type 0 -m state --state
ESTABLISHED,RELATED -j ACCEPT
  iptables -P INPUT DROP
  iptables -P FORWARD DROP
  iptables -P OUTPUT DROP
  ;;
*)
  echo "Es wurde kein oder ein falscher Parameter übergeben"
  echo "start: Zum Starten der Firewall."
  echo "stop: Zum Beenden der Firewall."
esac
iptables -L
```

Dann fügen wir den Ordner **/root/bin** zur PATH-Variablen hinzu, um das Script von überall ausführbar zu machen:

```
PATH=$PATH:/root/bin
```

Inside-Router:
Wir erstellen mit `mkdir /root/bin` den Ordner, wechseln dorthin und erstellen `touch firewall.sh` im Ordner **/root/bin/** als root folgendes **firewall.sh** Script und machen dieses mit `chmod 700 firewall.sh` ausführbar:

```
#!/bin/bash
if [ -z "$1" ]; then
  echo ""
  echo "enter \"start\" or \"stop\" as an argument to start or stop the
firewall"
  echo "enter \"show\" as an argument to display the current configuration"
  echo ""
  exit 1
else
  if [ "$1" = "start" ]; then
    echo ""
    echo "starting firewall..."
    echo ""
    # set default policy to drop everything
    iptables -P INPUT DROP
    iptables -P FORWARD DROP
    iptables -P OUTPUT DROP
    # flush all filter table rules
    iptables -F
    # flush all user defined filter table rules
    # iptables -X
    # allow outgoing ping request
    iptables -A OUTPUT -p icmp --icmp-type 8 -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT
```

FA54 P / LZ Herr Henze Gruppe 9 Andreas Biller, Rico Krüger

Thema: Aufbau einer DMZ

```
    iptables -A INPUT -p icmp --icmp-type 0 -m state --state
ESTABLISHED,RELATED -j ACCEPT
    # allow incomming ping request
    iptables -A INPUT -p icmp --icmp-type 8 -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT
    iptables -A OUTPUT -p icmp --icmp-type 0 -m state --state
ESTABLISHED,RELATED -j ACCEPT
  elif [ "$1" = "stop" ]; then
    echo ""
    echo "stopping firewall..."
    echo ""
    # allow everything
    iptables -P INPUT ACCEPT
    iptables -P FORWARD ACCEPT
    iptables -P OUTPUT ACCEPT
    # flush all filter table rules
    iptables -F
  elif [ "$1" = "show" ]; then
    echo ""
    echo "showing iptables:"
    echo ""
    iptables -L
  else
    echo ""
    echo "unrecognized argument: $1"
    echo "exiting script..."
    echo "enter \"start\" or \"stop\" as argument to start or stop the
firewall"
    echo ""
    exit 1
  fi
  # show iptables
  iptables -L
  echo ""
  echo "Good job! All done."
  echo ""
  exit 0
fi
```

Dann fügen wir den Ordner **/root/bin** zur PATH-Variablen hinzu, um das Script von überall ausführbar zu machen:

```
PATH=$PATH:/root/bin
```

TODO: allow `ssh` for using puTTY and xming through **firewall.sh,** DNS mit NAMESERVER ip-dns-labornetz (inside und outside) in die /etc/resolv.conf

A.2 Lastenheft

Es folgt unser Lastenheft mit Fokus auf den Anforderungen:

Die Umsetzung muss folgende Anforderungen erfüllen:

1. DMZ
   - 1.1. Die DMZ soll aus zwei virtuellen, zu Routern konfigurierten Linux-Distributionen bestehen, welch die Netze INSIDE, OUTSIDE und das DMZ-Netz miteinander verbinden.
   - 1.2. Die Router sollen entsprechend des Netzplanes eingerichtet und konfiguriert werden.
   - 1.3. Die DMZ soll Zugriffe auf den Webserver erlauben, aber Zugriffe auf das INSIDE-Netz verhindern. Hierzu soll auf dem Outside-Router NAT, Portforwarding und eine Firewall laufen.
   - 1.4. Die Router sollen nur vom Client-Rechner her fernadministrierbar sein.
2. Client-Rechner
   - 2.1. Der Client-Rechner im INSIDE-Netz nutzt das Betriebssystem Windows.
   - 2.2. Der Webserver soll eine Webseite mit dem aktuellen Stand der Gruppe anzeigen.
3. Webserver
   - 3.1. Der Webserver nutzt das Betriebssystem Windows. Er wird über das Tool Mini-Webserver vom Auftraggeber bereitgestellt.
   - 3.2. Der Webserver im DMZ-Netz muss vom OUTSIDE-Netz über Port 80 erreichbar sein. Hierzu soll auf dem Outside-Router NAT und Port-Forwarding eingerichtet werden.
   - 3.3. Der Webserver soll eine Webseite mit dem aktuellen Stand der Gruppe anzeigen.
4. Firewall
   - 4.1. Die Firewall soll den Webserver in der DMZ über Port 80 erreichbar sein lassen.
   - 4.2. Die Firewall soll SSH nur vom Admin-PC zulassen.
   - 4.3. Die Firewall soll ICMP zulassen.
   - 4.4. Die Firewall soll DNS zulassen.
   - 4.5. Die Firewall soll RDP zulassen.
   - 4.6. Die Firewall soll per Script an- und ausschaltbar sein. Hierzu muss an diversen Stellen per Script die Linux-Systemkonfiguration verändert werden
5. Sonstige Anforderungen
   - 5.1. Das Projekt soll unter Berücksichtigung der von der IHK ausgegebenen Richtlinien für eine Projektdokumentation dokumentiert werden.
   - 5.2. Es soll ein logischer Netzplan in Papierform erstellt und der Dokumentation angefügt werden.

5.3. Pro Person soll ein ausführliches Kompetenzportfolio erstellt werden, welches einen kritischen Überblick über unsere individuellen Kompetenzstände vor, während und nach dem Projekt liefert. Diese sollen der Dokumentation angehängt werden.

5.4. Die Funktionalität der Firewall soll getestet und die Ergebnisse in zwei Testprotokollen festgehalten werden. Diese sind der Dokumentation anzuhängen.

A.3 Pflichtenheft

Unser aus den Anforderungen des Lastenheftes erstelltes Pflichtenheft:

1. Musskriterien
   1.1. Das DMZ-Netz erhält die Netzmaske 172.16.9.0/24
   1.2. Das intere Netz erhält die Netzmaske 10.0.9.0/24
   1.3. Die öffentliche Schnittstelle des Outside-Router erhält die IP 192.168.200.109
   1.4. Der Outside-Router erhält als Standard-Gateway die IP 192.168.200.1
   1.5. Der Outside-Router erhält eine statische Route für das interne und DMZ-Netz
   1.6. Der Inside-Router erhält als Standard-Gateway das Interface des Outside-Routers, welches in die DMZ zeigt
   1.7. Der Webserver ist über die öffentliche IP des Outside-Routers über HTTP/S von außen erreichbar
   1.8. Der Webserver ist über die lokale IP 172.16.9.3 über HTTP/S aus dem internen Netzwerk erreichbar
   1.9. Die Router und Windows-Clients bekommen als DNS-Server die IPs 192.168.95.40 und 192.168.95.41
   1.10. Die Router und Windows-Clients bekommen als NTP-Server die IP 192.168.200.1
   1.11. Die Firewall verhindert unrechtmäßigen Datentransfer zwischen den Netzen und auf den Routern
   1.12. Der Admin-PC mit der IP 10.0.9.2 ist berechtigt mittels SSH auf die Router zuzugreifen
2. Kannkriterien
   2.1. Die Firewall lässt sich mit den Optionen ßtartünd ßtopän- bzw. ausschalten
   2.2. Die Firewall-Scripts der Router befinden sich im Verzeichnis /root/bin
   2.3. Die Veränderung der Firewall-Konfiguration befindet sich jeweils im Verzeichnis /var/log/-firewall
   2.4. Der Admin-PC mit der IP 10.0.9.2 ist berechtigt mittels RDP auf den Webserver zuzugreifen

A.4 Netzpläne

Der Netzplan unserer DMZ in der Projektumgebung im Labor 3.1.01:

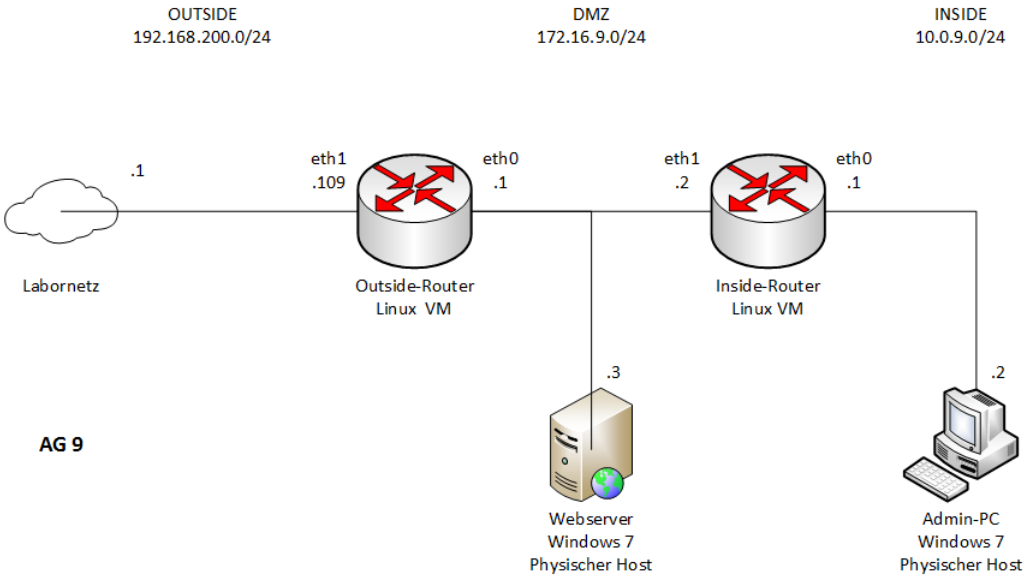


Abbildung 2: Netzplan der DMZ in Raum 3.1.01 (Arbeitsgruppe 9)

Der Netzplan unserer DMZ in der virtualisierten Testumgebung:

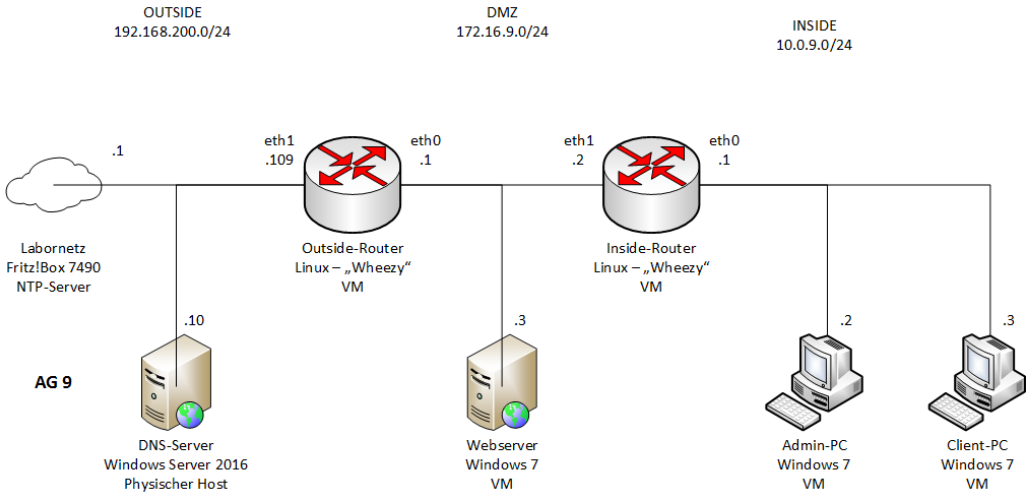


Abbildung 3: Netzplan der erweiterten DMZ in unserer virtuellen Testumgebung

A.5 Kompetenzportfolios

FA54 | **P / LZ** | **Herr Henze** | **Gruppe 9** | **Andreas Biller, Rico Krüger**

Kompetenzportfolios (Thema: Aufbau einer DMZ)

Kompetenzportfolio - Andreas Biller:

Themen / Inhalte	Kenntnisse & Fertigkeiten			Ausblick / Fazit
	Vor dem Projekt	Während dem Projekt	Nach dem Projekt	
Linux (Basics): Command Line Interface, Navigation im Dateisystem, Benutzer und Gruppen, Man Pages, Berechtigungen, usw.	Im Betrieb wird mit Linux gearbeitet und ich bin u. a. für Installation und den technischen Support zuständig. Daher arbeite ich täglich auf der Kommandozeile im Linux Dateisystem, lege Benutzer an und vergebe und ändere Berechtigungen. Sich wiederholende Tätigkeiten werden mit Ansible-Playbooks oder Shell-Skripten erledigt, um Fehler zu vermeiden. In Man-Pages schaue ich regelmäßig, um benötigte Parameter für seltener benutzte Befehle herauszusuchen. Die Rechner im Betriebsnetz lassen sich remote über ssh administrieren. Mit grep, cut, pipes, xargs, wget und anderen Befehlen werden hier z. B. aus csv-Dateien urls aus Spalten ausgelesen, mit anderen Kriterien verglichen oder korrigiert. So können durch einfache Bash-Befehlsketten unter Linux sehr kreative Ergebnisse erzielt werden, für die andere schon mal Excel und dessen Sortierfunktionen benutzen müssen, um Daten aus csv-Dateien auswerten zu können.	Da ich im Betrieb jeden Tag auf der Linux-Kommandozeile arbeite gibt hier nur bedingt neues für mich zu lernen. Die Grundlagen werden eher durch Wiederholung und Anwendung aufgefrischt, wenn in einem Lab Befehle geübt werden, die im Arbeitsalltag nicht so häufig Verwendung finden. So habe ich z. B. dank der guten Erklärung des netacad-Labs die verschiedenen Eigenschaften bei gesetzten Sticky-Bits besser verstanden. Auch die Suche nach Dateien mit find fand ich erfrischend einfach, da ich im Betrieb selten nach Datein an sich, sondern eher nach Text in Dateien mit Befehlen wie grep und Regulären Ausdrücken suche. Gerade das Arbeiten in der Bash schätze ich zunehmend dank dem schnellen Wiederholen von Befehlen durch die „Pfeil hoch“- bzw. „Pfeil runter“-Tasten oder dem interaktiven, rekursiven Suchen in der shell-History mit Strg+R.	Ich kenne mich dank vieler Aufgaben im Betrieb bereits gut im Linux-Dateisystem aus, deswegen wusste ich bereits, wo z. B. ausführbare Dateien liegen (bin), wie man die Path-Variable für neu installierte Befehle erweitert, wo Konfigurations- (etc) oder Log-Dateien (var) abgelegt werden. Ich kann dank den netacad-Labs und Chaptern vieles wiederholen und Skripte, Exit-Codes, verschiedene Outputvarianten mit Umgebungsvariablen, Pipes und Befehlen zu komplexen Anwendungen verknüpfen und weiß wo und wie ich Hilfe zu den benutzten Programmen finde.	Ich werde weiterhin auf Linux arbeiten und auch zuhause nuze ich in meiner Freizeit mindestens einen Rechner, auf dem Linux installiert ist. Grundwissen wird also immer wieder angewendet werden und sich festigen und weiterentwickeln, bis ich hoffentlich nicht mehr bei so vielen der seltener benutzten Befehlen in die Anleitung schauen muss.
Linux (Advanced): Shell-Skripte & Befehle, Umgebungsvariablen, Editoren (vi, nano, etc.), Konfigurationsdateien, usw.	Andere Aufgaben im Betrieb werden durch Skripte (.sh oder .rb) zu wiederholbaren Prozessen, CSV-Listen mit Befehlen wie grep, cut, xargs und dem Pipe-Operator ausgewertet. Auch Webseiten für Kunden erstelle ich dort über Commandline-Tools indem Markdown-Dateien mit jekyll und liquid templates zu html umgewandelt wird. Umgebungsvariablen benutzen wir in der Entwicklung und Produktion um die verwendeten Passwörter aus unserem Programmcode herauszuhalten. Zum Editieren benutze ich in der bash meist nano, ansonsten benutze ich der zusätzlichen Fuktionalität wegen lieber einen graphischen Editor wie gedit oder sublime.	Hier war der Nutzen schon größer, da ich im Betrieb aufgrund der für unsere Platform genutzten Programmiersprache meine Skripte hier meist in ruby schreibe und bash-Skripte nur dann verwende, wenn ich umbedingt einige der normalen Linux-Befehle für etwas Spezielles benötige. Aber auch hier ist vieles Wiederholung, gehört doch die Arbeit im CLI mit Umgenungsvariablen, Skripten, allgemeinen Befehlen wie git oder eher ruby-spezifischen wie rvm, bundle, rails, rubocop, etc. zu meinen täglichen Aufgaben. Zur Konfiguration editiere ich auch schon mal Dateien wie sudoers.d um Benutzern sudo-Rechte zu entziehen. Das Arbeiten in der shell fällt mir in der Arbeit dank um eigene Aliase erweiterter bash.rc einfacher als in einem unmodifizierten Debian wie dem genutzten	Ich habe viele Dinge wiederholt, die im Betrieb auch schon relevant waren, einige Sachen besser verstanden und gerade beim Skripten in der Shell Dinge angewendet, die im Betrieb wegen der Präferenz ruby zu benutzen häufig nicht in Shell-Skripten umgesetzt werden. Obwohl es mich sehr gereizt hätte, besser mit vi umgehen zu lernen, hat mir auch dieser kurze Versuch damit editieren zu wollen gezeigt, warum ich hier wenn möglich immer noch zu nano oder anderen, weniger komplexen Editoren greife.	Auch hoffe ich irgendwann die Zeit aufzubringen mich länger und intensiver mit vi oder vim zu beschäftigen. Da unser Betrieb stetig wächst und ich einen Linux-Rechner nach dem anderen konfiguriere, werde ich auch hier weitere Fertigkeiten entwickeln und mit dem gelernten Wissen Scripte schreiben und in Übung bleiben.
Netzwerk: Planung & Darstellung, Dienste, Routingtabellen, Statische Routen,	Netzwerke kenne ich hauptsächlich aus dem privaten Bereich, z. B. von der Konfiguration des Routers bzw. Netzwerkdruckers zuhause oder von kleinen Netzwerken über Hubs auf LAN-Parties. Mit ping habe ich bereits das eine oder andere mal die Funktionalität von LAN- und Internetverbindungen getestet, auch wenn ich nicht wußte, was dort alles genau passiert.	Dank der vorherigen Konfiguration von Netzwerken im ITS-Unterricht mit Packet-Tracer in der cisco-Syntax, auf Windows-Rechnern in deren Dialekt und nun dem Übersetzen bzw. Wiederholen der bereits bekannten Befehle nach Linux lerne ich hier gefühlt am meisten, da ich bei Befehlen wie ip route sehe, wie ähnlich sich die unterschiedlichen Systeme sind und dass sich meistens nur einige Schlüsselwörter oder die Schreibweise der Parameter	Ich habe ein besseres Verständnis davon erhalten, wie die Kommunikation in einem bzw. die Konfiguration eines Netzwerkes an sich funktioniert, egal auf welchem System man diese vornimmt. Dennoch fehlt mir noch einiges an Erfahrung, da ich im Fall von Problemen immer noch lange Suchen muß, bis ich das Problem eingrenzen kann.	Auch im Bereich Netzwerk kann ich das eine oder andere Gelernte aus diesem Projekt im Betrieb bestimmt noch praktisch umsetzen, da durch unseren momentanen Wachstum auch das verwendete Netzwerk immer wieder erweitert werden muss.

FA54 **P / LZ** **Herr Henze** **Gruppe 9** **Andreas Biller, Rico Krüger**

Kompetenzportfolios (Thema: Aufbau einer DMZ)

Testen (ping), usw.		unterscheiden, die zugrundeliegenden Prinzipien jedoch gleich sind.		
NAT: Webserver aufsetzen, Schnittstellen konfigurieren, Konfiguration speichern, usw.	Ich habe schon eigene Webserver (apache2) konfiguriert und betrieben, um Webseiten lokal zu testen und mein altes Portfolio selbst zu hosten, bzw. Portforwarding im Router eingerichtet um online mit Freunden spielen zu können. Den Router sowie die Schnittstellen habe ich bisher über graphische Oberflächen (unter Windows) eingerichtet.	Da ich bereits eigene Erfahrungen mit dem Aufsetzen und Konfigurieren von apache2 sowohl unter Windows wie unter Linux habe, ist das miniwebserver Tool keine Quelle neuer Lernerfahrungen (wenn auch einfach zu benutzen). NAT und Portforwarding sind dank dem gezielten Suchen nach Informationen im Internet und dem in ITS Erlernten schnell in die benötigten Konfigurationsdateien (/proc/sys/net/ipv4/ip_forward) geschrieben und über den auch für die Firewall benutzten Befehl iptables in der FORWARD-Chain mit dem Parameter MASQUERADE eingerichtet.	Auch im Betrieb arbeite ich unter anderem mit einem lokalen apache2 Webserver, um zu bearbeitende Webseiten in html oder php über localhost anzeigen zu lassen. Auch jekyll beim Erstellen von Webseiten sowie das „ruby on rails“-Framework bietet neben einer CLI- auch eine über den Browser geservte Entwicklungsumgebung, in der einiges konfiguriert werden muss, bis alles läuft wie es soll. Das erlangte Wissen aus dem Projekt wird hier in der einen oder anderen Form.sicher immer wieder auftauchen.	NAT muss ich betrieblich nicht wirklich einrichten, allerdings kann ich erlerntes Wissen bestimmt auf die eine oder andere Art bei der Arbeit an bzw. der Konfiguration von unserer bei Heroku gehosteten Webapp weiter verwenden.
Firewall: Skript erstellen bzw. anpassen, iptables verstehen & benutzen (Tables, Chains, Rules), usw.	Im Rahmen mehrerer Weiterbildungsmaßnahmen des Jobcenters durfte ich bereits vor der Arbeit im heutigen Betrieb als Aushilfe im Server. und IT-Bereich an der Rixdorfer Grundschule für einige Zeit Grundlagen in der PC- und Linux-Administration erlernen, unter anderem auch die prinzipielle Arbeitsweise und das Erstellen von Filterregeln für die einzelnen Chains bei einer Firewall, allerdings in einer graphischen Umgebung (webmin). Und leider musste ich diese Fähigkeiten dann nie wirklich praktisch anwenden und hatte das meiste davon bis heute wieder vergessen.	Da wir mit den Linux-Grundlagen und dem Erstellen des Netzwerkes die erste Hälfte des Projektes beschäftigt waren und auch in ITS erst zum zweiten Halbjahr etwas zu Access-Control-Lists gelernt haben, hatte ich nur dank dem beim NAT verwendeten Befehl iptables etwas zur Firewall gelernt, ohne dies jedoch zu wissen.	Nach dem Projekt habe ich die Bedeutung der einzelnen Chains für den Routing-Prozess verstanden und weiß wieder wie die iptable Regeln zusammen mit den Standard-Policies als Firewall verwendet werden können und wie ich diese mit Hilfe unseres Scriptes in der post-up bzw. pre-down Sequenz der interfaces speichern und so sowohl bei einem Neustart durch Befehle wie ifup/ifdown sowie beim Booten des Systems aktivieren bzw. auch wieder deaktivieren kann.	Das Wissen um die Funktionsweise von Firewall-Regeln und Grundlegende IT-Sicherheit sind wichtig, aber die Übung im Erstellen einer Dokumentation nach den IHK-Richtlinien für das bevorstehende Prüfungsprojekt sind dagegen nahezu unbezahlbar und werden mir von all dem Gelernten mit Sicherheit am ehesten noch unschätzbare Dienste leisten.

Kompetenzportfolio - Rico Krüger:

Themen / Inhalte	**Kenntnisse & Fertigkeiten**			**Ausblick / Fazit**
	Vor dem Projekt	**Während dem Projekt**	**Nach dem Projekt**	
Linux (Shell: Navigation, Befehle, Scripte, Editor)	Auf der Arbeit oder zu Hause arbeite ich selten in der Konsole oder mit Linux. Über die Verzeichnisstruktur und wichtige Dateien weiß ich kaum etwas. Die meisten Befehle und deren Optionen sind mir nicht geläufig. Ich habe in Linux noch kein Script geschrieben. Ich nutzte bisher stets einen grafischen Editor.	Ich nutze anfangs viel Google um nach Befehlen oder Dateien zu suchen. Die Befehle werden geläufiger und ich versuche mich an die man pages für Kommandos zu gewöhnen. Um Dateien schnell zu finden ist find / -name [name] sehr hilfreich. Anfangs wechsle ich noch in den Ordner um Dateien zu öffnen. Das ist nicht nötig. Alle Dateien lassen sich von überall her ansprechen. Das ist bei grafischen Oberflächen nicht gegeben. Ein Script wird stets mit !#[Path][Shell] eingeleitet. Um ein Script auszuführen muss man die Berechtigung mit chmod +x [Path][Script] ändern. Zum Editieren wechsle ich zwischen nano und vi, welcher mich an das less Kommando erinnert. Aufgerufen wird ein Script über ./[Path][Script]. Kommandos, die man aufruft werden nicht gespeichert, solange die Änderung	Ich fühle mich in der Shell wesentlich wohler und finde mich in der Verzeichnisstruktur zurecht. Das Arbeiten in der Konsole bringt viele Vorteile mit sich. Getätigte Befehle lassen schnell wiederholen und man muss sich nicht lange durch irgendwelche Fenster und Verzeichnisse navigieren. Viele Programme haben *.dotfiles*, mithilfe man diese konfigurieren kann. Einige Befehle haben sich eingeprägt. Um mir Optionen anzeigen zu lassen benutze ich --help und für genauere Informationen man. Nichtsdestotrotz greife ich noch, vor allem bei mir noch unbekannten Befehlen, auf Google zurück. Bevor man ein Script mit mehreren Befehlen schreibt, kann man die einzelnen Befehle erstmal problemlos in der Shell testen und hier auch nach Hilfe suchen. Alle Befehle haben Standardkanäle für Ausgabe(0), Eingabe(1) und Fehler(2).	Ich würde gerne weiter mit Linux arbeiten um meine Fähigkeiten im Scripting und Nutzen des Shell zu verbessern und so auch meine Produktivität zu steigern. Zudem ist Linux „sauberer“ als Windows, wo schon eine schier unendliche Anzahl an Diensten, Programmen und Bibliotheken vorinstalliert ist und man von vornherein jeglichen Überblick verloren hat, welche Dienste und Programme schon vorinstalliert sind und was diese eigentlich schon alles definieren

FA54 **P / LZ** **Herr Henze** **Gruppe 9** **Andreas Biller, Rico Krüger**

Kompetenzportfolios (Thema: Aufbau einer DMZ)

		nicht in eine Datei geschrieben werden. Die Interfaces konfiguriere ich in /etc/interfaces. Hier kann man auch DSN, statische Routen und NAT konfigurieren und speichern. Diese Datei wird beim Booten von Linux geladen.	Dieses Verhalten lässt sich mittels [x]> ändern. Ich bevorzuge vi nano, da ich so auch gleich die Navigation mit less verinnerliche und ich beim Editieren die *home row* nicht verlassen muss. Die DNS-Server trägt man in der /etc/resolv.conf ein. Um das mittels eines Scripts zu lösen kann man >> benutzen. Für Routen und NAT erstellt man am besten ein Script.	und ausführen. Mir gefällt die Logik von vi und würde gerne die diversen Eingaben aus dem Effeff beherrschen. So könnte ich wesentlich schneller arbeiten ohne auch nur die Tastatur verlassen zu müssen.
Netzwerk (NAT, Portforwarding, Statische Routen)	Ein Netzwerk zu konfigurieren haben ich bisher nur unter Windows mithilfe einer GUI gemacht. Wofür Statische Routen, NAT usw. gebraucht werden, wusste ich zwar jedoch kannte ich nicht den genauen Inhalt und wie ich diese unter Linux konfiguriere.	Das wichtigste ist erstmal, dass ich lerne dass ich ohne weiteres ein Linux-System als Router konfigurieren kann. Um IP-Forwarding zu aktivieren nutze ich echo "1" > /proc/sys/net/ipv4/ip_forward, für die statische Route ip route add -net [Netz] netmask [Netzmaske] gw [gw]. Um NAT zu konfigurieren für die Netzwerke nutze ich iptables –A FORWARD –o eth1 –s 172.16.9.0/24 --ctstate NEW –j ACCEPT. NAT konfiguriere ich mittels iptables –t nat –A POSTROUTING –o eth1 –j MASQUERADE. Danach muss ich ggf. die Interfaces über ifup / ifdown neustarten. Speicher ich diese Befehle in interfaces kann ich mit post-up / pre-down dafür sorgen, dass diese Konfiguration automatisch beim Booten geladen wird	Ich habe diese Befehle in mein Firewall.sh Script geschrieben und bin jetzt in der Lage, diese Befehle automatisiert ausführen zu lassen. Mithilfe von NAT werden interne Adressen in eine öffentliche Adresse des Routers gewandelt. Der Router speichert diese Zuordnung in eine Tabelle. Dort werden die Anfragen mit Hilfe von Portnummern gespeichert um diese wieder dem Client und dem zugehörigen Dienst zuordnen zu können.	Ich habe mein Wissen definitiv gesteigert und werde es auch noch weiter. Ich finde es hat Spaß gemacht und hilft mir auch auf der Arbeit. Ich werde mir mal privat ein kleines Netzwerk erstellen und dafür ein rasperry pi nutzen.
Firewall	Wofür eine Firewall gut ist, war mir schon bewusst, jedoch nicht nach welchem Prinzip sie arbeitet. Auch fand jegliche Konfiguration stets über eine GUI statt.	Ich lerne, dass solche Befehle in einer Liste abgearbeitet werden. Diese werden mit iptables –A angehängt. Dabei unterscheidet der Router in der INSIDE, OUTSIDE und FORWARD-chain. Man kann hier die Ports, das Protokoll sowie die Quell- und Zieladressen definieren. Dabei spielt es immer eine Rolle aus welcher Richtung die Anfrage kommt und an wen sie gerichtet ist. Diese Regeln werden von oben nach unten abgearbeitet.	Ich habe ein Firewall.sh Script. In diesen sich jetzt die NAT-Regeln und ein paar weitere Berechtigungen für DNS, HTTP/S und SSH. Ich denke, ich habe das Prinzip der Abarbeitung der ACLs verstanden und kann die verschiedenen chains auseinanderhalten.	Mein Überblick hat sich erhöht und ich kann das Prinzip gut nachvollziehen. Inwieweit ich diese Thematik jedoch noch vertiefe kann ich zum jetzigen Zeitpunkt noch nicht sagen, könnte mir aber vorstellen mal für zu Hause mir eigene Firewall aufzusetzen.
VMs	Ich habe vorher schon mit VMware und HyperV gearbeitet.	Ich habe VMWare Player genommen um Linux auszuführen.	Ich habe über Virtualisierung nichts Neues gelernt.	Ich werde wohl mehr mit HyperV, aufgrund der kostenlosen Lizenz arbeiten. So kann ich auch ein komplettes virtuelles Netzwerk errichten.

3/3

B Testdokumentation

Bauteil	Spezifikation
Prozessor	Intelel® Core™ i7-6700K Prozessor 8 MB Cache, 4.20 GHz
RAM	2x16GB DDR4-2400 DIMM CL15 Dual
Speicher	1x250GB SSD + 1x500GB SSD
Betriebssystem	Windows Server 2016 Datacenter

tab:SysteminformationSysteminformation.tex

Tabelle 9: Hardwaredetails des Testsystems

B.1 Aufbau der Testumgebung

Die zur Umsetzung dieses Kapitels benötigten Informationen entstammen unter anderem den hilfreichen Artikeln folgender Webseiten: BLOG.WORKINGHARDINIT.WORK [2017]

B.1.1 Implementierung der Virtuellen Maschinen

Im Server-Manager fügen wir über *Verwalten > Rollen und Features hinzufügen* den Hyper-V-Manager hinzu indem wir dem Assistenten folgen. Dieser gestattet es virtuelle Maschinen und Netzwerke zu installieren. Als nächstes wird eine neue virtuelle Linux (Debian 7.1) Maschine (Generation 1) aus einem Image erstellt. Dies geschieht mit Hilfe eines Assistenten. Sie bekommt einen virtuellen Prozessor und 1 GB Arbeitsspeicher. Des weiteren wird bei der Installation eine 5 GB große Festplatte für die Maschine erstellt und ihr zugewiesen. Als virtuellen Switch weisen wir ihr vorläufig den Netzwerkadapter des Hosts zu. Somit besitzt unsere Linux-VM Internet. Um sie zu installieren, startet man nun die Maschine und verbindet sich zu ihr. Danach folgt man wie gewohnt den Installationsschritten wie bei einer physischen Maschine. Danach installieren wir den NTP-Service. Ist die Grundkonfiguration fertig, wird die Maschine ausgeschaltet. Die Installation der Windows 7 VM erfolgt analog zu die der Linux VM. Wir vergeben jedoch 4 GB RAM und erstellen eine mindestens 30 GB große virtuelle Festplatte. Nach der Installation wird die Firewall wie in Implementierungsphase eingerichtet. Zusätzlich werden noch nützliche Software wie putty oder winscp heruntergeladen. Nach der Grundkonfiguration der beiden VMs können diese nun dupliziert werden. Dazu muss man die virtuelle Maschine erst exportieren, um sie danach wieder zu importieren. Beim Import sollte man darauf achten, dass man *eine neue eindeutige Identification (ID)* erstellt. Nachdem starten der importierten Maschine wird als erstes der Hostname geändert, um sie von der Originalen zu unterscheiden und um DNS-Konflikte zu vermeiden.

B.1.2 Implementierung des virtuellen Netzwerkes

Virtuelle Netzwerke werden über das Hinzufügen virtueller Switche an den Netzwerkadaptern der virtuellen Maschine erstellt. Auf diesen lassen sich auch Virtual LAN (VLAN)s einrichten. Die Installation eines solchen Switch wird ebenfalls vom Hyper-V-Manager mit einem Assistenten bereitgestellt. Für Testzwecke werden 2 *private* Switche erstellt, da diese die direkte Kommunikation mit dem Host unterbinden und somit nicht die Router umgangen werden. Diese erhalten den Namen DMZ- bzw. LAN-Switch. Ein *öffentlicher* Switch ist bereits vorhanden. Mit diesem ist der physische Netzwerkadapter des Hosts verbunden. Diese werden dann den VMs entsprechend des Netzpläne zugeordnet. Für die Linux-VMs, die als Router fungieren, muss evtl. noch ein zweiter Netzwerkadapter hinzugefügt werden. Nun können die Router und Clients (Siehe Bild und Implementierung) konfiguriert werden.

B.1.3 Implementierung des DNS-Servers

Der DNS-Server wird ebenfalls über den Server-Manager (unter *Rollen und Features hinzufügen*) installiert. Diesen kann man nun über den DNS-Manager verwalten. Es genügt eine *Forward-Lookup*-Zone zu erstellen. Als Zonennamen wählen wir *fritz.box* da bereits das Standard-Gateway darauf verweist. Dies ist die Domäne bzw. das DNS-Suffix. Dieses Suffix wird auf den Windows-VMs in den IPv4-Einstellungen des Netzwerkadapters nachgetragen. Auf den Linux-VMs tragen wir dies zusätzlich in die `/etc/resolv.conf` vor unserem DNS-Server ein. **siehe resolv.conf oder selber schreiben]**
Über den DNS-Manager werden im Anschluss noch in der Zone *fritz.box* unsere VMs (A-Record) mit Namen und IP-Adressen eingetragen. Siehe DNSManager.png.

B.1.4 Testen der Firewall

Nachdem das Firewall-Script auf die Router kopiert und die DNS-Server angepasst wurden, kann mit den Tests begonnen und die Firewall ggf. angepasst werden. Dazu speichern wir den Verlauf der erstellten Regeln als Log-Ausgabe in `/var/log/firewall/firewallConfig` ab. Die Ergebnisse unserer Tests finden sich als Übersicht in den folgenden Tabellen der Testprotokolle:

B.1.5 Testprotokolle

Service	Command	Source-IP	Destination-IP	Soll	Ist
ICMP	ping	10.0.9.2	10.0.9.1	ja	ja
ICMP	ping	10.0.9.3	10.0.9.1	ja	ja
ICMP	ping	10.0.9.2	172.16.9.1	ja	ja
ICMP	ping	10.0.9.3	172.16.9.1	ja	ja
ICMP	ping	172.16.9.3	172.16.9.1	ja	ja
ICMP	ping	172.16.9.3	172.16.9.2	ja	ja
ICMP	ping	10.0.9.3	192.168.200.10	ja	ja
ICMP	ping	192.168.200.10	10.0.9.3	ja	ja
ICMP	ping	192.168.200.10	172.16.9.3	ja	ja
ICMP	ping	192.168.200.10	192.168.200.109	ja	ja
ICMP	ping	10.0.9.3	8.8.8.8	ja	ja
HTTP	http://172.16.9.3	192.168.200.10	192.168.200.109	ja	ja
HTTP	http://172.16.9.3	192.168.200.10	172.16.9.3	ja	ja
HTTP	http://172.16.9.3	10.0.9.3	192.168.200.109	ja	ja
HTTP	http://172.16.9.3	10.0.9.3	172.16.9.3	ja	ja
NTP	w32tm /stripchart /computer:192.168.200.1	10.0.9.2	192.168.200.1	ja	ja
NTP	w32tm /stripchart /computer:192.168.200.1	172.16.9.3	192.168.200.1	ja	ja
NTP	ntpq -p	192.168.200.109	192.168.200.1	ja	ja
RDP	mstsc.exe	10.0.9.2	172.16.9.3	ja	ja
RDP	mstsc.exe	10.0.9.3	172.16.9.3	ja	ja
SSH	putty	10.0.9.2	10.0.9.1	ja	ja
SSH	putty	10.0.9.2	172.16.9.1	ja	ja
SSH	putty	10.0.9.3	10.0.9.1	ja	ja
SSH	putty	10.0.9.3	172.16.9.1	ja	ja
SSH	putty	172.16.9.3	172.16.9.1	ja	ja
SSH	putty	172.16.9.3	172.16.9.2	ja	ja
DNS	nslookup 172.16.9.1	10.0.9.3	192.168.200.10	ja	ja
DNS	nslookup Inside-Router	172.16.9.3	192.168.200.10	ja	ja
DNS	nslookup 10.0.9.1	172.16.9.2	192.168.200.10	ja	ja
DNS	nslookup Client-PC	192.168.200.109	192.168.200.10	ja	ja

Tabelle 10: Aus - Aus

Service	Command	Source-IP	Destination-IP	Soll	Ist
ICMP	ping	10.0.9.2	10.0.9.1	ja	ja
ICMP	ping	10.0.9.3	10.0.9.1	nein	nein
ICMP	ping	10.0.9.2	172.16.9.1	ja	ja
ICMP	ping	10.0.9.3	172.16.9.1	nein	nein
ICMP	ping	172.16.9.3	172.16.9.1	ja	ja
ICMP	ping	172.16.9.3	172.16.9.2	nein	nein
ICMP	ping	10.0.9.3	192.168.200.10	ja	ja
ICMP	ping	192.168.200.10	10.0.9.3	nein	nein
ICMP	ping	192.168.200.10	172.16.9.3	ja	ja
ICMP	ping	192.168.200.10	192.168.200.109	ja	ja
ICMP	ping	10.0.9.3	8.8.8.8	ja	ja
HTTP	http://172.16.9.3	192.168.200.10	192.168.200.109	ja	ja
HTTP	http://172.16.9.3	192.168.200.10	172.16.9.3	ja	ja
HTTP	http://172.16.9.3	10.0.9.3	192.168.200.109	nein	nein
HTTP	http://172.16.9.3	10.0.9.3	172.16.9.3	ja	ja
NTP	w32tm /stripchart /computer:192.168.200.1	10.0.9.2	192.168.200.1	ja	ja
NTP	w32tm /stripchart /computer:192.168.200.1	172.16.9.3	192.168.200.1	ja	ja
NTP	ntpq -p	192.168.200.109	192.168.200.1	ja	ja
RDP	mstsc.exe	10.0.9.2	172.16.9.3	ja	ja
RDP	mstsc.exe	10.0.9.3	172.16.9.3	nein	nein
SSH	putty	10.0.9.2	10.0.9.1	ja	ja
SSH	putty	10.0.9.2	172.16.9.1	ja	ja
SSH	putty	10.0.9.3	10.0.9.1	nein	nein
SSH	putty	10.0.9.3	172.16.9.1	ja	ja
SSH	putty	172.16.9.3	172.16.9.1	ja	ja
SSH	putty	172.16.9.3	172.16.9.2	nein	nein
DNS	nslookup 172.16.9.1	10.0.9.3	192.168.200.10	ja	ja
DNS	nslookup Inside-Router	172.16.9.3	192.168.200.10	ja	ja
DNS	nslookup 10.0.9.1	172.16.9.2	192.168.200.10	ja	ja
DNS	nslookup Client-PC	192.168.200.109	192.168.200.10	ja	ja

Tabelle 11: Aus - An

Service	Command	Source-IP	Destination-IP	Soll	Ist
ICMP	ping	10.0.9.2	10.0.9.1	ja	ja
ICMP	ping	10.0.9.3	10.0.9.1	ja	ja
ICMP	ping	10.0.9.2	172.16.9.1	ja	ja
ICMP	ping	10.0.9.3	172.16.9.1	nein	nein
ICMP	ping	172.16.9.3	172.16.9.1	nein	nein
ICMP	ping	172.16.9.3	172.16.9.2	ja	ja
ICMP	ping	10.0.9.3	192.168.200.10	ja	ja
ICMP	ping	192.168.200.10	10.0.9.3	nein	nein
ICMP	ping	192.168.200.10	172.16.9.3	nein	nein
ICMP	ping	192.168.200.10	192.168.200.109	nein	nein
ICMP	ping	10.0.9.3	8.8.8.8	ja	ja
HTTP	http://172.16.9.3	192.168.200.10	192.168.200.109	ja	ja
HTTP	http://172.16.9.3	192.168.200.10	172.16.9.3	ja	ja
HTTP	http://172.16.9.3	10.0.9.3	192.168.200.109	nein	nein
HTTP	http://172.16.9.3	10.0.9.3	172.16.9.3	ja	ja
NTP	w32tm /stripchart /computer:192.168.200.1	10.0.9.2	192.168.200.1	ja	ja
NTP	w32tm /stripchart /computer:192.168.200.1	172.16.9.3	192.168.200.1	ja	ja
NTP	ntpq -p	192.168.200.109	192.168.200.1	ja	ja
RDP	mstsc.exe	10.0.9.2	172.16.9.3	ja	ja
RDP	mstsc.exe	10.0.9.3	172.16.9.3	nein	nein
SSH	putty	10.0.9.2	10.0.9.1	ja	ja
SSH	putty	10.0.9.2	172.16.9.1	ja	ja
SSH	putty	10.0.9.3	10.0.9.1	ja	ja
SSH	putty	10.0.9.3	172.16.9.1	ja	ja
SSH	putty	172.16.9.3	172.16.9.1	nein	nein
SSH	putty	172.16.9.3	172.16.9.2	ja	ja
DNS	nslookup 172.16.9.1	10.0.9.3	192.168.200.10	ja	ja
DNS	nslookup Inside-Router	172.16.9.3	192.168.200.10	ja	ja
DNS	nslookup 10.0.9.1	172.16.9.2	192.168.200.10	ja	ja
DNS	nslookup Client-PC	192.168.200.109	192.168.200.10	ja	ja

Tabelle 12: An - Aus

Service	Command	Source-IP	Destination-IP	Soll	Ist
ICMP	ping	10.0.9.2	10.0.9.1	ja	ja
ICMP	ping	10.0.9.3	10.0.9.1	ja	ja
ICMP	ping	10.0.9.2	172.16.9.1	ja	ja
ICMP	ping	10.0.9.3	172.16.9.1	nein	nein
ICMP	ping	172.16.9.3	172.16.9.1	nein	nein
ICMP	ping	172.16.9.3	172.16.9.2	ja	ja
ICMP	ping	10.0.9.3	192.168.200.10	ja	ja
ICMP	ping	192.168.200.10	10.0.9.3	nein	nein
ICMP	ping	192.168.200.10	172.16.9.3	nein	nein
ICMP	ping	192.168.200.10	192.168.200.109	nein	nein
ICMP	ping	10.0.9.3	8.8.8.8	ja	ja
HTTP	http://172.16.9.3	192.168.200.10	192.168.200.109	ja	ja
HTTP	http://172.16.9.3	192.168.200.10	172.16.9.3	ja	ja
HTTP	http://172.16.9.3	10.0.9.3	192.168.200.109	nein	nein
HTTP	http://172.16.9.3	10.0.9.3	172.16.9.3	ja	ja
NTP	w32tm /stripchart /computer:192.168.200.1	10.0.9.2	192.168.200.1	ja	ja
NTP	w32tm /stripchart /computer:192.168.200.1	172.16.9.3	192.168.200.1	ja	ja
NTP	ntpq -p	192.168.200.109	192.168.200.1	ja	ja
RDP	mstsc.exe	10.0.9.2	172.16.9.3	ja	ja
RDP	mstsc.exe	10.0.9.3	172.16.9.3	nein	nein
SSH	putty	10.0.9.2	10.0.9.1	ja	ja
SSH	putty	10.0.9.2	172.16.9.1	ja	ja
SSH	putty	10.0.9.3	10.0.9.1	ja	ja
SSH	putty	10.0.9.3	172.16.9.1	ja	ja
SSH	putty	172.16.9.3	172.16.9.1	nein	nein
SSH	putty	172.16.9.3	172.16.9.2	ja	ja
DNS	nslookup 172.16.9.1	10.0.9.3	192.168.200.10	ja	ja
DNS	nslookup Inside-Router	172.16.9.3	192.168.200.10	ja	ja
DNS	nslookup 10.0.9.1	172.16.9.2	192.168.200.10	ja	ja
DNS	nslookup Client-PC	192.168.200.109	192.168.200.10	ja	ja

Tabelle 13: An - An

B.2 Firewall-Skripte

B.2.1 firewall.sh (auf dem Outside-Router)

```
1  #!/bin/bash
2  # Bourne- Again Shell#
3
4  # ======================================================
5  # === Aufgabenstellung ================================
6  # ======================================================
7
8  # 1. Datei in " firewall .sh" umbenennen
9  # 2. Datei ausfuehrbar machen: Auf der Kommandozeile das Skript starten mit: ./firewall.sh ENTER
10 # 3. Wenn Fehlermeldung (das Skript laeuft gar nicht) Konvertierung mit "dos2unix Dateiname"
11 #------------Aufgabenstellung
       ------------------------------------------------
12 # 1.Passen Sie dieses Firewall-Skript an die folgende Aufgabenstellung an.
13 # 2.Ihre unter Linux laufenden Rechner (Router/Firewalls) sollen mindestens folgendermassen konfiguriert sein:
14 # a) Jeder Rechner (Webserver, Host, zwei Linux-Router) Ihrer Arbeitsgruppe muss die eigene Zeit mit einem
       Zeitserver
15 #  synchronisieren koennen. Nehmen Sie auf jeden Fall den schulinternen Zeitserver (Standardgateway:192.168.200.1)
       , da die
16 #  externen evt. nicht erreichbar sind.
17 # b) Ihr Webserver soll von ueberall (eigenes LAN und fremde Netzwerke) nur auf Port 80 erreichbar sein.
18 # c) Ping (echo-request) soll fuer alle Rechner des eigenen Netzes (Intern) erlaubt sein und auch echo-reply
       Antworten
19 #  aus dem Inernet erhalten. (z.B. ping 141.1.1.1, ping 8.8.8.8)
20 # d) Die Wartung der Linux Router mittels 'ssh' soll nur von einem ausgezeichneten Rechner Ihres eigenen LANs
       erlaubt sein.
21 #   Die Linux-Router sind vor allen anderen Zugriffen zu schuetzen!!
22 # e) Der/die Rechner des eigenen LANs sollen per "http" in das Internet (google, gmx etc.) kommen koennen.
23 # f) Die "Default Policy" der Firewalls muss auf "DROP" stehen. (Alles was nicht explizit erlaubt ist, ist verboten
       !!)
24 # g) Darueber hinaus lassen Sie sich in Ihrer Kreativitaet nicht einschraenken.
25 #
26 # 3.Tipp: Sie sollten sich ein zweites, kurzes Skript schreiben, das die Firewall komplett oeffnet und alle Regeln
       loescht, um
27 # jederzeit testen zu koennen, ob Ihr Netzwerk noch steht.
28 #
29 #------Ende--Aufgabenstellung
       ------------------------------------------------
30
31
32 # ======================================================
33 # === Part 1: Variablen ===============================
34 # ======================================================
35 echo " - Variablen werden gesetzt"
36
37 # Pfad zu iptables
38 IPTABLES=/sbin/iptables
```

```
39
40 # Macht Linux-Maschine zu einem Router
41 echo "1" > /proc/sys/net/ipv4/ip_forward
42
43 # Interfaces
44 iINT=eth0
45 iEXT=eth1
46
47 # Definition DNS
48 DNS=("192.168.95.40/32 192.168.95.41/32")
49
50 # Timeserver: hier Standardgateway
51 TimeSrv=192.168.200.1
52
53 # Der Rechner, auf dem die Firewall (Inside) laufen soll, hier die VMWare
54 LinuxInside_in=10.0.9.1
55 LinuxInside_dmz=172.16.9.2
56
57 # Der Rechner, auf dem die Firewall (Outside) laufen soll, hier die VMWare
58 LinuxOutside_out=192.168.200.109
59 LinuxOutside_dmz=172.16.9.1
60
61 # Rechner fuer Fernwartung z.B. mit ssh, hier der Windowswirt (XP, Win7 o.ae.)
62 AdminPC=10.0.9.2
63
64 # Webserver
65 Webserver=172.16.9.3
66
67 # Das DMZ-Netz
68 DMZ=172.16.9.0/24
69
70 # Das LAN-Netz
71 LAN=10.0.9.0/24
72
73 # Protokolle
74 protocols=("tcp" "udp")
75
76 # DNS Ports
77 dnsPorts=("53" "853")
78
79 # HTTP/S Port
80 webPorts=("80" "443")
81
82 # ntp Port
83 ntpPort=123
84
85 # rdp Port
86 rdpPort=3389
87
88 # Pfad zur aktuellen Firewall Konfiguration
89 lopPath="/var/log/firewall/ firewallconfig "
```

```
90
91 #------Ende--Variablen setzen
       ------------------------------------------------
92
93
94 # =========================================================
95 # =========================================================
96 # === Starten / Stoppen / Hilfe ========================
97 # =========================================================
98 # =========================================================
99 case "$1" in
100
101
102 # =========================================================
103 # =========================================================
104 # === Firewall stoppen ================================
105 # =========================================================
106 # =========================================================
107 stop)
108
109 # =========================================================
110 # === Part 2: Default Policy setzen ====================
111 # =========================================================
112
113 # ****************** Alles erlauben und alle Regeln loeschen
114 echo " - do: Policy and flush"
115 # Default policy setzen (Alles erlauben)
116 $IPTABLES -P INPUT ACCEPT
117 $IPTABLES -P FORWARD ACCEPT  # Bei 2 Interfaces (Router)
118 $IPTABLES -P OUTPUT ACCEPT
119
120 # Loesche alle Filterregeln
121 $IPTABLES -F  # flush aller chains (Tabelle filter )
122 $IPTABLES -t nat -F # flush aller chains (Tabelle nat)
123 $IPTABLES -X  # delete all userdefined chains (Tabelle  filter )
124
125 # ***** ENDE ******* NAT und Port-Forwarding **********
126 echo " - done: Policy and flush"
127
128
129 # =========================================================
130 # === Part 3: NAT und Port-Forwarding implementieren ===
131 # =========================================================
132
133 # ****************** NAT und Port-Forwarding aktivieren
134 echo " - do: NAT und Port-Forwarding"
135 # Hier die Zeilen schreiben, die
136 # a) NAT auf dem Outside-Router implementiert und
137 $IPTABLES -A FORWARD -o $iEXT -s $DMZ -m conntrack --ctstate NEW -j ACCEPT
138 $IPTABLES -A FORWARD -o $iEXT -s $LAN -m conntrack --ctstate NEW -j ACCEPT
139 $IPTABLES -t nat -A POSTROUTING -o $iEXT -j MASQUERADE
```

```
140
141 # b) das Port-Forwarding von ausserhalb zu dem Webserver aktivieren
142 $IPTABLES -A PREROUTING -t nat -i $iEXT -p tcp --dport 80 -j DNAT --to-destination $Webserver:80
143 $IPTABLES -A FORWARD -p TCP -d $Webserver --dport 80 -j ACCEPT
144 $IPTABLES -A POSTROUTING -t nat -s $Webserver -o $iEXT -j MASQUERADE
145
146 # ***** ENDE ******* NAT und Port-Forwarding aktivieren
147 echo " - done: NAT und Port-Forwarding"
148
149
150 # ========================================================
151 # === Ausgabe ============================================
152 # ========================================================
153
154 # ****************** Konfiguration in DAtei umleiten
155 echo " - do: Schreibe Konfiguration in $lopPath"
156 echo -e "\n\n================================================" >> $lopPath
157 date >> $lopPath
158 echo "Firewall gestoppt" >> $lopPath
159 echo -e "================================================\n" >> $lopPath
160 $IPTABLES -L -v -n >> $lopPath
161
162 # ***** ENDE *********** Konfiguration in DAtei umleiten
163 echo " - done: Schreibe Konfiguration in $lopPath"
164
165 # ========================================================
166 #*******ENDE************ Firewall stoppen **************
167 # ========================================================
168 ;;
169
170
171 # ========================================================
172 # ========================================================
173 # === Firewall starten ===================================
174 # ========================================================
175 # ========================================================
176 start )
177
178 # ========================================================
179 # === Part 2: Default Policy setzen ======================
180 # ========================================================
181
182 # ****************** Alles verbieten und alle Regeln loeschen
183 echo " - do: Policy and flush"
184
185 # Default Policy: Alles verbieten
186 $IPTABLES -P INPUT DROP
187 $IPTABLES -P FORWARD DROP  # Bei 2 Interfaces (Router)
188 $IPTABLES -P OUTPUT DROP
189
190 # Loesche alte Filterregeln
```

```
191 # chain (engl. Kette, Folge, Befehlsfolge)
192 $IPTABLES -F  # flush aller chains (Tabelle  filter )
193 $IPTABLES -t nat -F # flush aller chains (Tabelle nat)
194 $IPTABLES -X  # delete all userdefined chains (Tabelle  filter )
195
196 # ***** ENDE ******* Alles verbieten und alle Regeln loeschen
197 echo " - done: Policy and flush"
198
199
200 # ========================================================
201 # === Part 3: NAT und Port-Forwarding implementieren ===
202 # ========================================================
203
204 # ***************** Loopback erlauben ****************
205 echo " - do: NAT und Port-Forwarding"
206 # Hier die Zeilen schreiben, die
207 # a) NAT auf dem Outside-Router implementiert und
208 $IPTABLES -A FORWARD -o $iEXT -s $DMZ -m conntrack --ctstate NEW -j ACCEPT
209 $IPTABLES -A FORWARD -o $iEXT -s $LAN -m conntrack --ctstate NEW -j ACCEPT
210 $IPTABLES -t nat -A POSTROUTING -o $iEXT -j MASQUERADE
211
212 # b) das Port-Forwarding von ausserhalb zu dem Webserver aktivieren
213 $IPTABLES -A PREROUTING -t nat -i $iEXT -p tcp --dport 80 -j DNAT --to-destination $Webserver:80
214 $IPTABLES -A FORWARD -p TCP -d $Webserver --dport 80 -j ACCEPT
215 $IPTABLES -A POSTROUTING -t nat -s $Webserver -o $iEXT -j MASQUERADE
216
217 # ******ENDE ******* Alles verbieten und alle Regeln loeschen
218 echo " - done: NAT und Port-Forwarding"
219
220
221 # ========================================================
222 # === Part 4: Aufgabenstellung umsetzen ================
223 # ========================================================
224
225 # ***************** Loopback erlauben ****************
226 echo " - do: Loopback erlauben"
227 $IPTABLES -A INPUT -i lo -j ACCEPT
228 $IPTABLES -A OUTPUT -o lo -j ACCEPT
229
230 #****** ENDE ******* Loopback erlauben *****************
231 echo " - done: Loopback erlauben"
232
233
234 # ***************** ssh-Zugriff vom AdminPC auf Router sicherstellen
235 echo " - do: SSH-Zugang fuer AdminPC"
236 # fuer den Outside-Router
237 $IPTABLES -A INPUT -p TCP -s $AdminPC --dport ssh -j ACCEPT
238 $IPTABLES -A OUTPUT -p TCP -d $AdminPC --sport ssh -j ACCEPT
239
240 # fuer den Inside-Router
241 $IPTABLES -A FORWARD -p TCP -s $AdminPC -d $LinuxInside_dmz --dport ssh -j ACCEPT
```

```
242 $IPTABLES -A FORWARD -p TCP -s $LinuxInside_dmz -d $AdminPC --sport ssh -j ACCEPT
243 $IPTABLES -A FORWARD -p TCP -s $AdminPC -d $LinuxInside_in --dport ssh -j ACCEPT
244 $IPTABLES -A FORWARD -p TCP -s $LinuxInside_in -d $AdminPC --sport ssh -j ACCEPT
245
246 # ***** ENDE ******* ssh-Zugriff vom AdminPC auf Router sicherstellen
247 echo " - done: SSH-Zugang fuer AdminPC"
248
249
250 # ****************** Verbindung zu einem Zeitserver erlauben
251 echo " - do: NTP Ports oeffnen"
252 # fuer diesen (den Outside-) Router erlauben
253 $IPTABLES -A INPUT -p udp -s $TimeSrv --sport $ntpPort -j ACCEPT
254 $IPTABLES -A OUTPUT -p udp -d $TimeSrv --dport $ntpPort -j ACCEPT
255
256 # fuer DMZ-Netz erlauben
257 $IPTABLES -A FORWARD -p udp -s $DMZ -d $TimeSrv --dport $ntpPort -j ACCEPT
258 $IPTABLES -A FORWARD -p udp -d $DMZ -s $TimeSrv --sport $ntpPort -j ACCEPT
259
260 # fuer LAN-Netz erlauben
261 $IPTABLES -A FORWARD -p udp -s $LAN -d $TimeSrv --dport $ntpPort -j ACCEPT
262 $IPTABLES -A FORWARD -p udp -d $LAN -s $TimeSrv --sport $ntpPort -j ACCEPT
263
264 # ***** ENDE ******* Konfiguration fuer Zeitsynchronisation
265 echo " - done: NTP Ports oeffnen"
266
267
268 echo " - do: Zugang fuer Webserver"
269 # ****************** Verbindung zum Webserver zulassen *
270 $IPTABLES -A FORWARD -p TCP -s $Webserver --sport 80 -j ACCEPT
271 $IPTABLES -A FORWARD -p TCP -d $Webserver --dport 80 -j ACCEPT
272
273 # ***** ENDE *********** Konfiguration fuer Zugriff auf Webserver
274 echo " - done: Zugang fuer Webserver"
275
276
277 # ***************** ICMP Erlauben **********************
278 echo " - do: Ping erlauben"
279 # ICMP-ECHO Request und ICMP-ECHO Reply fuer den Outside-Router durch AdminPC erlauben
280 $IPTABLES -A INPUT -p icmp --icmp-type 8 -s $AdminPC -j ACCEPT
281 $IPTABLES -A OUTPUT -p icmp --icmp-type 0 -d $AdminPC -j ACCEPT
282
283 # ICMP-ECHO Request und ICMP-ECHO Reply fuer das DMZ-Netz erlauben
284 $IPTABLES -A FORWARD -p icmp --icmp-type 8 -s $DMZ -j ACCEPT
285 $IPTABLES -A FORWARD -p icmp --icmp-type 0 -d $DMZ -j ACCEPT
286
287 # ICMP-ECHO Request und ICMP-ECHO Reply fuer das LAN-Netz
288 $IPTABLES -A FORWARD -p icmp --icmp-type 8 -s $LAN -j ACCEPT
289 $IPTABLES -A FORWARD -p icmp --icmp-type 0 -d $LAN -j ACCEPT
290
291 # ***** ENDE *********** Konfiguration ICMP ***********
292 echo " - done: Ping erlauben"
```

```
293
294
295 # ****************** Konfiguration DNS HTTP HTTPS ******
296 echo " - do: DNS erlauben"
297 ## DNS durchlassen fuer DMZ und LAN
298 for port in ${dnsPorts[@]}
299 do
300   for protocol in ${protocols[@]}
301   do
302     $IPTABLES -A FORWARD -p "$protocol" -s $DMZ --dport "$port" -j ACCEPT
303     $IPTABLES -A FORWARD -p "$protocol" -d $DMZ --sport "$port" -j ACCEPT
304     $IPTABLES -A FORWARD -p "$protocol" -s $LAN --dport "$port" -j ACCEPT
305     $IPTABLES -A FORWARD -p "$protocol" -d $LAN --sport "$port" -j ACCEPT
306   done
307 done
308
309 ## Bestimmte DNS-Server fuer Router
310 for port in ${dnsPorts[@]}
311 do
312   for protocol in ${protocols[@]}
313   do
314     for dnsSrv in ${DNS[@]}
315     do
316       $IPTABLES -A INPUT -p "$protocol" -s "$dnsSrv" -d $LinuxOutside_out --sport "$port" -j ACCEPT
317       $IPTABLES -A OUTPUT -p "$protocol" -s $LinuxOutside_out -d "$dnsSrv" --dport "$port" -j ACCEPT
318     done
319   done
320 done
321
322 # ***** ENDE *********** Konfiguration DNS *************
323 echo " - done: DNS erlauben"
324
325
326 # ****************** Konfiguration HTTP HTTPS **********
327 echo " - do: HTTP/S erlauben"
328 ## HTTP/S fuer LAN und DMZ erlauben
329 for port in ${webPorts[@]}
330 do
331   $IPTABLES -A FORWARD -p TCP -s $DMZ --dport "$port" -j ACCEPT
332   $IPTABLES -A FORWARD -p TCP -d $DMZ --sport "$port" -j ACCEPT
333   $IPTABLES -A FORWARD -p TCP -s $LAN --dport "$port" -j ACCEPT
334   $IPTABLES -A FORWARD -p TCP -d $LAN --sport "$port" -j ACCEPT
335 done
336
337 # ***** ENDE *********** Konfiguration DNS HTTP/S ******
338 echo " - done: HTTP/S erlauben"
339
340
341 # ****************** Konfiguration RDP *****************
342 echo " - do: RDP erlauben"
343 # RDP Zugang fuer DMZ-Server
```

```
344 for protocol in ${protocols[@]}
345 do
346   $IPTABLES -A FORWARD -p "$protocol" -s $Webserver -d $AdminPC --sport $rdpPort -j ACCEPT
347   $IPTABLES -A FORWARD -p "$protocol" -s $AdminPC -d $Webserver --dport $rdpPort -j ACCEPT
348 done
349
350 # ***** ENDE ********** Konfiguration RDP *************
351 echo " - done: RDP erlauben"
352
353
354 # ======================================================
355 # === Ausgabe ==========================================
356 # ======================================================
357
358 # ***************** Konfiguration in DAtei umleiten ***
359 echo " - do: Schreibe Konfiguration in $lopPath"
360 echo -e "\n\n==================================================" >> $lopPath
361 date >> $lopPath
362 echo "Firewall gestartet" >> $lopPath
363 echo -e "==================================================\n" >> $lopPath
364 $IPTABLES -L -v -n >> $lopPath
365
366 # ***** ENDE ********** Konfiguration in DAtei umleiten
367 echo " - done: Schreibe Konfiguration in $lopPath"
368
369 # ======================================================
370 # ***** ENDE ********** Firewall starten *************
371 # ======================================================
372 ;;
373
374
375 # ======================================================
376 # === Firewall Parameter anzeigen ======================
377 # ======================================================
378 *)
379
380 # ***************** Anzeige Fehlermeldung und Hilfe
381 echo "Falscher oder kein Parameter uebergeben!"
382 echo "stop - Stoppt die Firewall."
383 echo "start - Startet die Firewall."
384
385 # ***** ENDE ********** Anzeige Fehlermeldung und Hilfe
386
387 # ======================================================
388 # ***** ENDE ********** Eingabeoptionen anzeigen ******
389 # ======================================================
390 ;;
391
392
393 esac
```

B.2.2 firewall.sh (auf dem Inside-Router)

```
#!/bin/bash
# Bourne- Again Shell#

# ======================================================
# === Bemerkung ========================================
# ======================================================

# Sekundaere Firewall
# ...verhindert unbefugten Zugriff vom lokalem Netz auf lokales Interface des Routers.
# Wenn die Firewall gestoppt ist, wird auch NAT gestoppt. Dies sorgt dafuer,
# dass auch alle internen Anfragen an das DMZ-Netz ueber den Outside-Router laufen.
# Wenn die Firewall startet laeuft der Verkehr zwischen LAN und DMZ nur ueber den Inside-Router.
# Da NAT die IP des Admin-PCs uebersetzt, greifen die Zugriffsberechtigungen
# (ICMP, SSH) auf dem Outside-Router nicht. Daher wird wird er vom NAT ausgeschlossen.
#
#------Ende--Bemerkung--------------------------------


# ======================================================
# === Part 1: Variablen ================================
# ======================================================
echo " - Variablen werden gesetzt"

# Pfad zu iptables
IPTABLES=/sbin/iptables

# Macht Linux-Maschine zu einem Router
echo "1" > /proc/sys/net/ipv4/ip_forward

# Interfaces
iINT=eth0
iEXT=eth1

# Definition DNS
DNS=("192.168.95.40/32 192.168.95.41/32")

# Timeserver: hier Standardgateway
TimeSrv=192.168.200.1

# Der Rechner, auf dem die Firewall (Inside) laufen soll, hier die VMWare
LinuxInside_in=10.0.9.1
LinuxInside_dmz=172.16.9.2

# Der Rechner, auf dem die Firewall (Outside) laufen soll, hier die VMWare
LinuxOutside_out=192.168.200.109
LinuxOutside_dmz=172.16.9.1

# Rechner fuer Fernwartung z.B. mit ssh, hier der Windowswirt (XP, Win7 o.ae.)
AdminPC=10.0.9.2
```

```
50
51 # Webserver
52 Webserver=172.16.9.3
53
54 # Das DMZ-Netz
55 DMZ=172.16.9.0/24
56
57 # Das LAN-Netz
58 LAN=10.0.9.0/24
59
60 # Protokolle
61 protocols=("tcp" "udp")
62
63 # DNS Ports
64 dnsPorts=("53" "853")
65
66 # HTTP/S Port
67 webPorts=("80" "443")
68
69 # ntp Port
70 ntpPort=123
71
72 # rdp Port
73 rdpPort=3389
74
75 # Pfad zur aktuellen Firewall Konfiguration
76 logPath="/var/log/firewall/ firewallconfig "
77
78 # ***** ENDE ******* Variablen setzen ******************
79
80
81 # ========================================================
82 # ========================================================
83 # === Starten / Stoppen / Hilfe ==========================
84 # ========================================================
85 # ========================================================
86 case "$1" in
87
88 # ========================================================
89 # ========================================================
90 # === Firewall stoppen ===================================
91 # ========================================================
92 # ========================================================
93 stop)
94
95 # ========================================================
96 # === Part 2: Default Policy setzen ======================
97 # ========================================================
98
99 # ****************** Alles erlauben und alle Regeln loeschen
100 echo " - do: Policy and flush"
```

```
101 # Default policy setzen (Alles erlauben)
102 $IPTABLES -P INPUT ACCEPT
103 $IPTABLES -P FORWARD ACCEPT
104 $IPTABLES -P OUTPUT ACCEPT
105
106 # Loesche alle Filterregeln
107 $IPTABLES -F  # flush aller chains (Tabelle filter )
108 $IPTABLES -t nat -F # flush aller chains (Tabelle nat)
109 $IPTABLES -X  # delete all userdefined chains (Tabelle filter )
110
111 # ***** ENDE ******* NAT und Port-Forwarding
112 echo " - done: Policy and flush"
113
114
115 # ========================================================
116 # === Ausgabe ============================================
117 # ========================================================
118
119 # ****************** Konfiguration in DAtei umleiten
120 echo " - do: Schreibe Konfiguration in $lopPath"
121 echo -e "\n\n================================================" >> $lopPath
122 date >> $lopPath
123 echo "Firewall gestoppt" >> $lopPath
124 echo -e "================================================\n" >> $lopPath
125 $IPTABLES -L -v -n >> $lopPath
126
127 # ***** ENDE *********** Konfiguration in DAtei umleiten
128 echo " - done: Schreibe Konfiguration in $lopPath"
129
130 # ========================================================
131 # ***** ENDE *********** Firewall stoppen **************
132 # ========================================================
133 ;;
134
135
136 # ========================================================
137 # ========================================================
138 # === Firewall starten ===================================
139 # ========================================================
140 # ========================================================
141 start)
142
143 # ========================================================
144 # === Default Policy setzen und NAT ======================
145 # ========================================================
146
147 # ****************** Alles verbieten und alle Regeln loeschen
148 echo " - do: Policy and flush"
149
150 # Default Policy: Alles verbieten
151 $IPTABLES -P INPUT DROP
```

```
152 $IPTABLES -P FORWARD DROP
153 $IPTABLES -P OUTPUT DROP
154
155 # Loesche alte Filterregeln
156 # chain (engl. Kette, Folge, Befehlsfolge)
157 $IPTABLES -F  # flush aller chains (Tabelle filter )
158 $IPTABLES -t nat -F # flush aller chains (Tabelle nat)
159 $IPTABLES -X  # delete all userdefined chains (Tabelle filter )
160
161 # ***** ENDE ******* Alles verbieten und alle Regeln loeschen
162 echo " - done: Policy and flush"
163
164
165 # ****************** NAT aktivieren ********************
166 echo " - do: NAT"
167 # NAT auf dem Inside-Router implementieren, Admin-PC auschliessen
168 $IPTABLES -A FORWARD -o $iEXT -s $LAN -m conntrack --ctstate NEW -j ACCEPT
169 $IPTABLES -t nat -A POSTROUTING -m iprange --src-range 10.0.9.3-10.0.9.254 -o $iEXT -j
      MASQUERADE
170
171 # ***** ENDE ******* NAT aktivieren ********************
172 echo " - done: NAT"
173
174
175 # ============================================================
176 # === LO, NTP, ICMP, SSH, DNS, HTTPS, RDP ==============
177 # ============================================================
178
179 # ****************** Loopback erlauben *****************
180 echo " - do: Loopback erlauben"
181 $IPTABLES -A INPUT -i lo -j ACCEPT
182 $IPTABLES -A OUTPUT -o lo -j ACCEPT
183
184 # ***** ENDE ******* Loopback erlauben *****************
185 echo " - done: Loopback erlauben"
186
187
188 # ****************** Verbindung zu einem Zeitserver erlauben
189 echo " - do: NTP Ports oeffnen"
190 # fuer diesen (den Inside-) Router erlauben
191 $IPTABLES -A INPUT -p udp -s $TimeSrv --sport $ntpPort -j ACCEPT
192 $IPTABLES -A OUTPUT -p udp -d $TimeSrv --dport $ntpPort -j ACCEPT
193
194 # fuer LAN-Netz erlauben
195 $IPTABLES -A FORWARD -p udp -s $LAN -d $TimeSrv --dport $ntpPort -j ACCEPT
196 $IPTABLES -A FORWARD -p udp -d $LAN -s $TimeSrv --sport $ntpPort -j ACCEPT
197
198 # ***** ENDE ******* Konfiguration fuer Zeitsynchronisation
199 echo " - done: NTP Ports oeffnen"
200
201
```

```
202 # ****************** ssh-Zugriff vom AdminPC auf Router sicherstellen
203 echo " - do: SSH-Zugang fuer AdminPC"
204 # fuer den Inside-Router
205 $IPTABLES -A INPUT -p TCP -s $AdminPC --dport ssh -j ACCEPT
206 $IPTABLES -A OUTPUT -p TCP -d $AdminPC --sport ssh -j ACCEPT
207
208 # fuer den Outside-Router
209 $IPTABLES -A FORWARD -s $AdminPC -d $LinuxOutside_dmz -p TCP --dport ssh -j ACCEPT
210 $IPTABLES -A FORWARD -s $LinuxOutside_dmz -d $AdminPC -p TCP --sport ssh -j ACCEPT
211 $IPTABLES -A FORWARD -s $AdminPC -d $LinuxOutside_out -p TCP --dport ssh -j ACCEPT
212 $IPTABLES -A FORWARD -s $LinuxOutside_out -d $AdminPC -p TCP --sport ssh -j ACCEPT
213
214 # ***** ENDE ******* ssh-Zugriff vom AdminPC auf Router sicherstellen
215 echo " - done: SSH-Zugang fuer AdminPC"
216
217
218 # ***************** ICMP Erlauben **********************
219 echo " - do: Ping erlauben"
220 # ICMP-ECHO Request und ICMP-ECHO Reply fuer den Inside-Router durch AdminPC erlauben
221 $IPTABLES -A INPUT -p icmp --icmp-type 8 -s $AdminPC -j ACCEPT
222 $IPTABLES -A OUTPUT -p icmp --icmp-type 0 -d $AdminPC -j ACCEPT
223
224 # ICMP-ECHO Request und ICMP-ECHO Reply fuer das DMZ-Netz erlauben
225 $IPTABLES -A FORWARD -p icmp --icmp-type 8 -s $DMZ -j ACCEPT
226 $IPTABLES -A FORWARD -p icmp --icmp-type 0 -d $DMZ -j ACCEPT
227
228 # ICMP-ECHO Request und ICMP-ECHO Reply fuer das LAN-Netz
229 $IPTABLES -A FORWARD -p icmp --icmp-type 8 -s $LAN -j ACCEPT
230 $IPTABLES -A FORWARD -p icmp --icmp-type 0 -d $LAN -j ACCEPT
231
232 # ***** ENDE *********** Konfiguration ICMP ************
233 echo " - done: Ping erlauben"
234
235
236 echo " - do: Zugang fuer Webserver"
237 # ****************** Verbindung zum Webserver zulassen
238 $IPTABLES -A FORWARD -p TCP -s $Webserver --sport 80 -j ACCEPT
239 $IPTABLES -A FORWARD -p TCP -d $Webserver --dport 80 -j ACCEPT
240
241 # ***** ENDE *********** Konfiguration fuer Zugriff auf Webserver
242 echo " - done: Zugang fuer Webserver"
243
244
245 # ****************** Konfiguration DNS HTTP HTTPS ******
246 echo " - do: DNS erlauben"
247 ## DNS durchlassen fuer LAN
248 for port in ${dnsPorts[@]}
249 do
250   for protocol in ${protocols[@]}
251   do
252     $IPTABLES -A FORWARD -p "$protocol" -s $LAN --dport "$port" -j ACCEPT
```

```
253     $IPTABLES -A FORWARD -p "$protocol" -d $LAN --sport "$port" -j ACCEPT
254   done
255 done
256
257 ## Bestimmte DNS-Server fuer Router
258 for port in ${dnsPorts[@]}
259 do
260   for protocol in ${protocols[@]}
261   do
262     for dnsSrv in ${DNS[@]}
263     do
264       $IPTABLES -A INPUT -p "$protocol" -s "$dnsSrv" -d $LinuxInside_dmz --sport "$port" -j ACCEPT
265       $IPTABLES -A OUTPUT -p "$protocol" -s $LinuxInside_dmz -d "$dnsSrv" --dport "$port" -j ACCEPT
266     done
267   done
268 done
269
270 # ******ENDE********** Konfiguration DNS **********
271 echo " - done: DNS erlauben"
272
273
274 # ********** Konfiguration HTTP HTTPS **********
275 echo " - do: HTTP/S erlauben"
276 ## HTTP/S fuer LAN und DMZ erlauben
277 for port in ${webPorts[@]}
278 do
279   $IPTABLES -A FORWARD -p TCP -s $LAN --dport "$port" -j ACCEPT
280   $IPTABLES -A FORWARD -p TCP -d $LAN --sport "$port" -j ACCEPT
281 done
282
283 # ***** ENDE ********** Konfiguration DNS HTTP/S ******
284 echo " - done: HTTP/S erlauben"
285
286
287 # ********** Konfiguration RDP **********
288 echo " - do: RDP erlauben"
289 # RDP Zugang fuer DMZ-Server
290 for protocol in ${protocols[@]}
291 do
292   $IPTABLES -A FORWARD -p "$protocol" -s $AdminPC -d $Webserver --dport $rdpPort -j ACCEPT
293   $IPTABLES -A FORWARD -p "$protocol" -s $Webserver -d $AdminPC --sport $rdpPort -j ACCEPT
294 done
295
296 # ***** ENDE ********** Konfiguration RDP **********
297 echo " - done: RDP erlauben"
298
299
300 # ======================================================
301 # === Ausgabe ==========================================
302 # ======================================================
303
```

```
304 # ****************** Konfiguration in DAtei umleiten
305 echo " - do: Schreibe Konfiguration in $lopPath"
306 echo -e "\n\n=============================================" >> $lopPath
307 date >> $lopPath
308 echo "Firewall gestartet" >> $lopPath
309 echo -e "=============================================\n" >> $lopPath
310 $IPTABLES -L -v -n >> $lopPath
311
312 # ***** ENDE *********** Konfiguration in DAtei umleiten
313 echo " - done: Schreibe Konfiguration in $lopPath"
314
315 # ============================================================
316 # ***** ENDE *********** Firewall starten **************
317 # ============================================================
318 ;;
319
320
321 # ============================================================
322 # === Eingabeoptionen anzeigen ==========================
323 # ============================================================
324 *)
325
326 # ****************** Anzeige Fehlermeldung und Hilfe ***
327 echo "Falscher oder kein Parameter uebergeben!"
328 echo "stop - Stoppt die Firewall."
329 echo "start - Startet die Firewall."
330
331 # ***** ENDE *********** Anzeige Fehlermeldung und Hilfe
332
333 # ============================================================
334 # ***** ENDE *********** Eingabeoptionen anzeigen ******
335 # ============================================================
336 ;;
337
338
339 esac
```