Fachinformatiker für Anwendungsentwicklung

Dokumentation zur schulischen Projektarbeit im Fach P/LZ

Aufbau einer DMZ

in einem mittelständischen Unternehmen

Arbeitsgruppe 9: Rico Krüger, Andreas Biller

Abbildung 1: DMZ zwischen Nord- und Südkorea

Abgabetermin: Berlin, den 25.06.2017

Oberstufenzentrum Informations- und Medizintechnik
Haarlemer Str. 23-27, 12359 Berlin

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Abkürzungsverzeichnis

DMZ	Demilitarisierte Zone
FA54	Klassenbezeichnung am OSZ IMT)
ITS	Informationstechnische Systeme
P/LZ	Projekt/Linux-Zertifizierung
API	Application Programming Interface
CSV	Comma Separated Value
PHP	Hypertext Preprocessor
SVN	Subversion

1 Einleitung

1.1 Projektumfeld

Unternehmen: "Das OSZ IMT 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 Berufsfachschule, 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 Klassenbezeichnung am OSZ IMT) (FA54).

Auftraggeber: Als angehende Fachinformatiker für Anwendungsentwicklung am OSZ IMT sollen wir nun im Rahmen des Faches Projekt/Linux-Zertifizierung (P/LZ) ein auf mittelständige Unternehmen anwendbares IT-Sicherheitskonzept entwickeln. Dazu werden wir im Verlauf des Projektunterrichtes eine Demilitarisierte Zone (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 OSZ IMT 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 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] Pressemappe, "Porträt des OSZ IMT"**?**

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 das NATen 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 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: Grundlegende Motivation ist wohl für jeden Bereiligten an diesem Projekt seine ganz eigene Sache. Der Auftraggeber ist daran interresiert, ein fertiges, funktionierendes System zu erhalten, welches seine Wünsche und Anforderungen erfüllt, aber er und auch wir können darüber hinaus uns und uns gegenseitig an greifbaren Indikatoren bezüglich unserer 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-PCs 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 OSZ IMT 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 Zeitplanung für die einzelnen Projektphasen:

Projektphase	Geplante Zeit
Analysephase	6 h
Entwurfsphase	9 h
Implementierungsphase	12 h
Abnahmetest der Fachabteilung	1 h
Erstellen der Dokumentation	14 h
Pufferzeit	84 h
Gesamt	**126 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 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 Bildschirmarbeitzplatz, Windows 7, Adminrechte, zwei physikalische Netzwerkinterfaces, Anschluß an Labornetzwerk und Internet, die Software VMWare Player, Debian Images auf einem Netzlaufwerk sowie ein Webserver.

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 Routingeinstellungen 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 wwerden, 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 Upscaling 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 €.

[2] Räumlichkeiten, Arbeitsplatzrechner etc.

Vorgang	Zeit	Kosten pro Stunde	Kosten
Entwicklungskosten	42 h	4,62 €$x2 = 9{,}24$ €	388,08 €
			388,08 €

Tabelle 2: Kostenaufstellung

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 sich spätestens zur IHK-Prüfung und bei der Anfertigung der Dokumentation des IHK-Abschlussprojektes auszahlen.

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 Anwendungsfälle

.....

Beispiel Ein Beispiel für ein Use Case-Diagramm findet sich im Anhang **??**: **??** auf Seite ??.

3.5 Qualitätsanforderungen

Der Webserver soll von Außen (über die öffentliche IP des Outside-Routers) und Innen erreichbar, aber vor potentiellen Angreifern bestmöglich mit den zur Verfügung stehenden Mitteln geschützt sein. Es muss sichergestellt werden, dass kein unberechtigter Dritter Zugriff auf die Geräte und deren Konfiguration hat. Dabei ist darauf zu achten, dass die Mitarbeiter weiterhin wie gewohnt Zugriff auf das Internet und den Webserver haben.

3.6 Fachkonzept

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.

3.7 Zwischenstand

Tabelle 3 zeigt den Zwischenstand nach der Analysephase.

Vorgang	Geplant	Tatsächlich	Differenz
1. Analyse des Ist-Zustands	3 h	4 h	+1 h
2. „Make or buy“-Entscheidung und Wirtschaftlichkeitsanalyse	1 h	1 h	
3. Erstellen eines „Use-Case“-Diagramms	2 h	2 h	
4. Erstellen des Lastenhefts	3 h	3 h	

Tabelle 3: Zwischenstand nach der Analysephase

4 Entwurfsphase

Da unsere Hard- und Software von unserem Auftraggeber gestellt und vorgegeben wird, erübrigt seine ausführliche Begründung, weshalb wir diese Materialien verwendet haben. Zudem wird so sichergestellt, dass während unserer Projektzeit alle 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 VMs wird mit 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 SSH- und 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

Abbildung 2 zeigt die grundsätzliche IP-Adressverteilung in den geplanten Netzwerken. Unser Konzept teilt sich grundsätzlich in das Labornetz (hier symbolisch für den Rest der Welt), das interne Netz (mit den Windows-Clients unseres Kunden) und das von der Außenwelt abgeschottete DMZ-Netzwerk, welches nur über spezielle Berechtigungen zu erreichen und für spezielle Dienste (Webserver) zu verwenden ist.

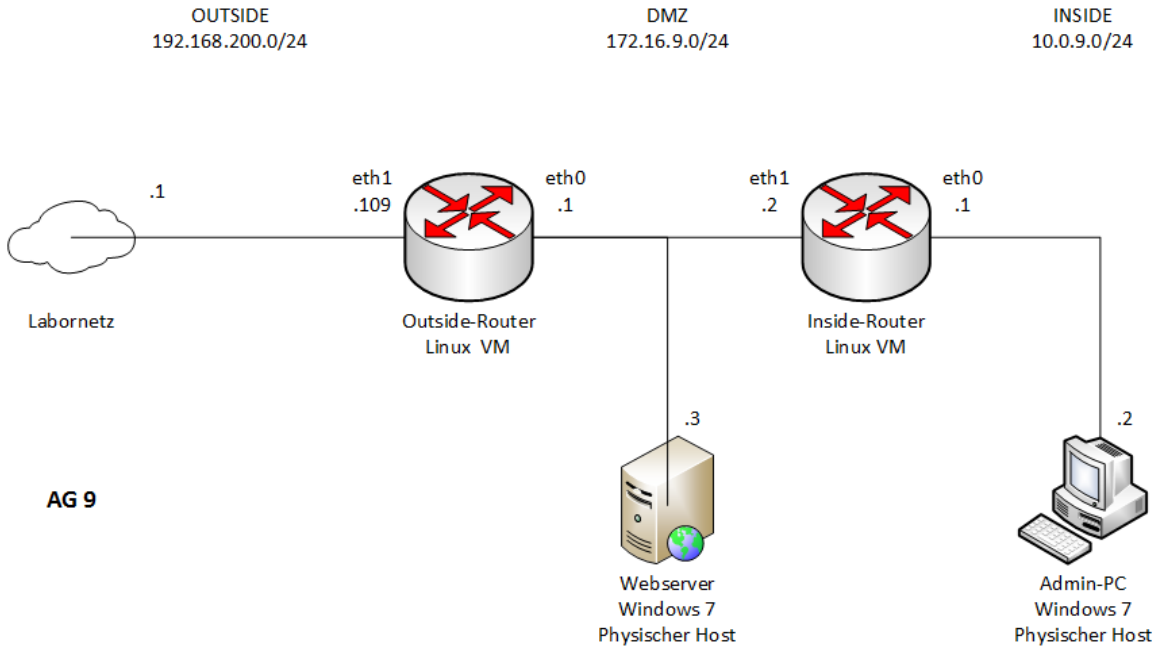


Abbildung 2: Netzplan DMZ Arbeitsgruppe 9

4.3 Maßnahmen zur Qualitätssicherung

Bei jeder Veränderungen der Konfiguration werden Tests durchgeführt. Diese sollen gewährleisten, dass das Fachkonzept eingehalten wird. Vorgenommene Konfigurationen werden notiert und das Firewall-Script wird zusätzlich auf einen externen Datenträger kopiert. So wird sichergestellt, dass bei einem Defekt die ursprüngliche Konfiguration schnell wieder verfügbar ist.

4.4 Pflichtenheft/Datenverarbeitungskonzept

1. Musskriterien
    - Das DMZ-Netz erhält die Netzmaske 172.16.9.0/24
    - Das intere Netz erhält die Netzmaske 10.0.9.0/24

- Die öffentliche Schnittstelle des Outside-Router erhält die IP 192.168.200.109
- Der Outside-Router erhält als Standard-Gateway die IP 192.168.200.1
- Der Outside-Router erhält eine statische Route für das interne und DMZ-Netz
- Der Inside-Router erhält als Standard-Gateway das Interface des Outside-Routers, welches in die DMZ zeigt
- Der Webserver ist über die öffentliche IP des Outside-Routers über HTTP/S von außen erreichbar
- Der Webserver ist über die lokale IP 172.16.9.3 über HTTP/S aus dem internen Netzwerk erreichbar
- Die Router und Windows-Clients bekommen als DNS-Server die IPs 192.168.95.40 und 192.168.95.41
- Die Router und Windows-Clients bekommen als NTP-Server die IP 192.168.200.1
- Die Firewall verhindert unrechtmäßigen Datentransfer zwischen den Netzen und auf den Routern
- Der Admin-PC mit der IP 10.0.9.2 ist berechtigt mittels SSH auf die Router zuzugreifen

2. Kannkriterien

- Die Firewall lässt sich mit den Optionen ßtartünd ßtopän- bzw. ausschalten
- Die Firewall-Scripts der Router befinden sich im Verzeichnis /root/bin
- Die Veränderung der Firewall-Konfiguration befindet sich jeweils im Verzeichnis /var/log/-firewall
- Der Admin-PC mit der IP 10.0.9.2 ist berechtigt mittels RDP auf den Webserver zuzugreifen

4.5 Zwischenstand

Tabelle 4 zeigt den Zwischenstand nach der Entwurfsphase.

5 Implementierungsphase

. . . immer testen usw.

Vorgang	Geplant	Tatsächlich	Differenz
1. Prozessentwurf	2 h	3 h	+1 h
2. Datenbankentwurf	3 h	5 h	+2 h
3. Erstellen von Datenverarbeitungskonzepten	4 h	4 h	
4. Benutzeroberflächen entwerfen und abstimmen	2 h	1 h	-1 h
5. Erstellen eines UML-Komponentendiagramms	4 h	2 h	-2 h
6. Erstellen des Pflichtenhefts	4 h	4 h	

Tabelle 4: Zwischenstand nach der Entwurfsphase

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 Netzplan (Siehe ??). Um die Interfaces zu konfigurieren, wird die Datei `/etc/network/interfaces` geöffnet.

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 ïnterfaces"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 Losung und geht nach einem Neustart verloren. Damit der Prozess mit dem Systemstart geladen wird, tragen wir (, nachdem unsere Tests erfolgreich waren,) setzen wir den Wert in der Datei `/etc/sysctl.conf` von `#net.ipv4.ip_forward` auf `1` und kommentieren diese Zeile aus.

5.2.4 Konfiguration des DNS-Server

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 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 GUI gearbeitet.

5.3.1 Konfiguration der Interfaces

Für die IP-Adressierung halten wir uns ebenfalls an den Netzplan(Siehe Netzplan Produktionsumgebung).

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 ïndex.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 ëingehendenäls auch äusgehenden"Regeln vorzunehmen.

Damit wir einen "pingBefehl absetzen können, ist es nötig die Regel für die "Datei- und Druckerabfrage"für ICMPv4 zu aktivieren.

Für die Kommunikation zum DNS-Server erstellen wir zwei Regeln, je eine für das TCP- bzw. UPD-Protokoll. Darin erlauben wir die Kommunikation über die Ports 53 und 853.

5.3.4 Konfiguration des Zeitservers

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

5.4 Konfiguration der Firewll

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 loggt die Firewall die, wann Sie gestartet / gestoppt und wie ihre Einstellungen zum derzeitigen Stand sind / waren. Das Log-File befindet sich im Ordner `/var/log/firewall/firewallConfig`.

Genauere Angaben finden sich im angefügten Firewall-Script.

5.5 Zwischenstand

Tabelle 5 zeigt den Zwischenstand nach der Implementierungsphase.

Vorgang	Geplant	Tatsächlich	Differenz
1. Anlegen der Datenbank	1 h	1 h	
2. Umsetzung der HTML-Oberflächen und Stylesheets	4 h	3 h	-1 h
3. Programmierung der PHP-Module für die Funktionen	23 h	23 h	
4. Nächtlichen Batchjob einrichten	1 h	1 h	

Tabelle 5: Zwischenstand nach der Implementierungsphase

6 Abnahmephase

Da die Originalmaschinen zum Testzeit nicht mehr verfügbar waren, wurde hierzu eine eigene Testumgebung mittels HyperV nachgestellt. Genauere Angaben über die Teststellung und ein ausführlicher Test befinden sich im Anhang B.

6.1 Zwischenstand

Tabelle 6 zeigt den Zwischenstand nach der Abnahmephase.

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

Tabelle 6: Zwischenstand nach der Abnahmephase

7 Einführungsphase

- Welche Schritte waren zum Deployment der Anwendung nötig und wie wurden sie durchgeführt (automatisiert/manuell)?
- Wurden ggfs. Altdaten migriert und wenn ja, wie?
- Wurden Benutzerschulungen durchgeführt und wenn ja, Wie wurden sie vorbereitet?

7.1 Zwischenstand

Tabelle 7 zeigt den Zwischenstand nach der Einführungsphase.

Vorgang	Geplant	Tatsächlich	Differenz
1. Einführung/Benutzerschulung	1 h	1 h	

Tabelle 7: Zwischenstand nach der Einführungsphase

8 Dokumentation

- Wie wurde die Anwendung für die Benutzer/Administratoren/Entwickler dokumentiert (z. B. Benutzerhandbuch, API-Dokumentation)?
- Hinweis: Je nach Zielgruppe gelten bestimmte Anforderungen für die Dokumentation (z. B. keine IT-Fachbegriffe in einer Anwenderdokumentation verwenden, aber auf jeden Fall in einer Dokumentation für den IT-Bereich).

Beispiel Ein Ausschnitt aus der erstellten Benutzerdokumentation befindet sich im Anhang **??**: **??** auf Seite ??. Die Entwicklerdokumentation wurde mittels PHPDoc[3] automatisch generiert. Ein beispielhafter Auszug aus der Dokumentation einer Klasse findet sich im Anhang **??**: **??** auf Seite ??.

8.1 Zwischenstand

Tabelle 8 zeigt den Zwischenstand nach der Dokumentation.

[3] Vgl. **?**

Vorgang	Geplant	Tatsächlich	Differenz
1. Erstellen der Benutzerdokumentation	2 h	2 h	
2. Erstellen der Projektdokumentation	6 h	8 h	+2 h
3. Programmdokumentation	1 h	1 h	

Tabelle 8: Zwischenstand nach der Dokumentation

9 Fazit

9.1 Soll-/Ist-Vergleich

- Wurde das Projektziel erreicht und wenn nein, warum nicht? ja, viel über Netzwerk, Firewall(iptables), Linux(grundsätzliche Struktur, Terminal) aber nicht in der Zeit ne
- Ist der Auftraggeber mit dem Projektergebnis zufrieden und wenn nein, warum nicht?
- Wurde die Projektplanung (Zeit, Kosten, Personal, Sachmittel) eingehalten oder haben sich Abweichungen ergeben und wenn ja, warum? zeit, somit kosten auch aufgrund von Krankheit, begrenztem Zugang testabweichung
- Hinweis: Die Projektplanung muss nicht strikt eingehalten werden. Vielmehr sind Abweichungen sogar als normal anzusehen. Sie müssen nur vernünftig begründet werden (z. B. durch Änderungen an den Anforderungen, unter-/überschätzter Aufwand).

Beispiel (verkürzt) Wie in Tabelle 9 zu erkennen ist, konnte die Zeitplanung bis auf wenige Ausnahmen eingehalten werden.

Phase	Geplant	Tatsächlich	Differenz
Entwurfsphase	19 h	19 h	
Analysephase	9 h	10 h	+1 h
Implementierungsphase	29 h	28 h	-1 h
Abnahmetest der Fachabteilung	1 h	1 h	
Einführungsphase	1 h	1 h	
Erstellen der Dokumentation	9 h	11 h	+2 h
Pufferzeit	2 h	0 h	-2 h
Gesamt	70 h	70 h	

Tabelle 9: Soll-/Ist-Vergleich

9.2 Lessons Learned

- Was hat der Prüfling bei der Durchführung des Projekts gelernt (z. B. Zeitplanung, Vorteile der eingesetzten Frameworks, Änderungen der Anforderungen)?

Zeitplanung, Linux, Firewall, zusätzlich (Netzwerk-)Virtualisierung

9.3 Ausblick

- Wie wird sich das Projekt in Zukunft weiterentwickeln (z. B. geplante Erweiterungen)?

Netzwerk ausbauen, domain controller, DHCP, DNS, FTP, Exchange (über Windows und / oder Linux)

Eidesstattliche Erklärung

Wir, Rico Krüger und Andreas Biller, versichern hiermit, dass wir unsere **Dokumentation zur schulischen Projektarbeit im Fach P/LZ** mit dem Thema

Aufbau einer DMZ in einem mittelständischen Unternehmen

selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben, wobei wir alle wörtlichen und sinngemäßen Zitate als solche gekennzeichnet haben. Die Arbeit wurde bisher keinem anderen Lehrer vorgelegt und auch nicht veröffentlicht.

Berlin, den 25.06.2017

_____

Andreas Biller, Rico Krüger

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

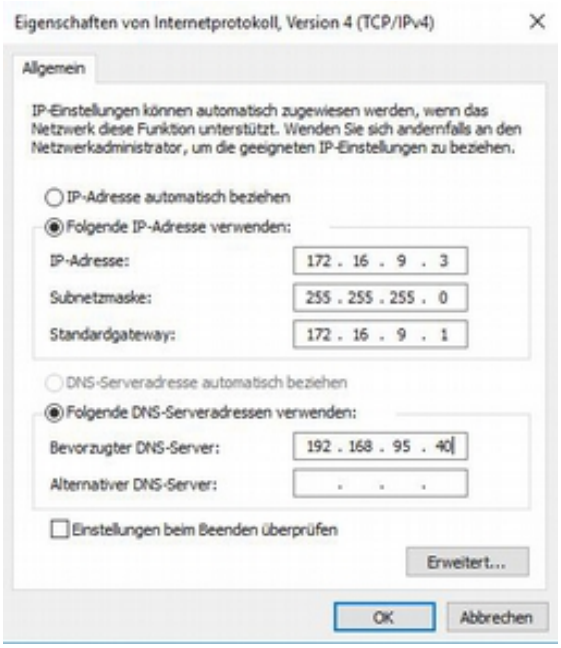
Die physikalischen Schnittstellen der Hosts von den beiden Linux-VMs werden über *„Systemsteuerung“ -> „Netzwerk- und Freigabecenter“ -> „Adaptereinstellungen ändern“ -> „Ethernet-Adapter“ -> Eigenschaften -> „Internetprotokoll, Version 4 (TCP/IPv4)“ -> „Eigenschaften“* geändert.

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
```

```
  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 Detaillierte Zeitplanung

Analysephase			**9 h**
1. Analyse des Ist-Zustands		3 h	
1.1. Fachgespräch mit der EDV-Abteilung	1 h		
1.2. Prozessanalyse	2 h		
2. „Make or buy“-Entscheidung und Wirtschaftlichkeitsanalyse		1 h	
3. Erstellen eines „Use-Case“-Diagramms		2 h	
4. Erstellen des Lastenhefts mit der EDV-Abteilung		3 h	
Entwurfsphase			**19 h**
1. Prozessentwurf		2 h	
2. Datenbankentwurf		3 h	
2.1. ER-Modell erstellen	2 h		
2.2. Konkretes Tabellenmodell erstellen	1 h		
3. Erstellen von Datenverarbeitungskonzepten		4 h	
3.1. Verarbeitung der CSV-Daten	1 h		
3.2. Verarbeitung der SVN-Daten	1 h		
3.3. Verarbeitung der Sourcen der Programme	2 h		
4. Benutzeroberflächen entwerfen und abstimmen		2 h	
5. Erstellen eines UML-Komponentendiagramms der Anwendung		4 h	
6. Erstellen des Pflichtenhefts		4 h	
Implementierungsphase			**29 h**
1. Anlegen der Datenbank		1 h	
2. Umsetzung der HTML-Oberflächen und Stylesheets		4 h	
3. Programmierung der PHP-Module für die Funktionen		23 h	
3.1. Import der Modulinformationen aus CSV-Dateien	2 h		
3.2. Parsen der Modulquelltexte	3 h		
3.3. Import der SVN-Daten	2 h		
3.4. Vergleichen zweier Umgebungen	4 h		
3.5. Abrufen der von einem zu wählenden Benutzer geänderten Module	3 h		
3.6. Erstellen einer Liste der Module unter unterschiedlichen Aspekten	5 h		
3.7. Anzeigen einer Liste mit den Modulen und geparsten Metadaten	3 h		
3.8. Erstellen einer Übersichtsseite für ein einzelnes Modul	1 h		
4. Nächtlichen Batchjob einrichten		1 h	
Abnahmetest der Fachabteilung			**1 h**
1. Abnahmetest der Fachabteilung		1 h	
Einführungsphase			**1 h**
1. Einführung/Benutzerschulung		1 h	
Erstellen der Dokumentation			**9 h**
1. Erstellen der Benutzerdokumentation		2 h	
2. Erstellen der Projektdokumentation		6 h	
3. Programmdokumentation		1 h	
3.1. Generierung durch PHPdoc	1 h		
Pufferzeit			**2 h**
1. Puffer		2 h	
Gesamt			**70 h**

A.3 Lastenheft (Auszug)

Es folgt ein Auszug aus dem Lastenheft mit Fokus auf die Anforderungen:

Die Anwendung muss folgende Anforderungen erfüllen:

1. Verarbeitung der Moduldaten
   1.1. Die Anwendung muss die von Subversion und einem externen Programm bereitgestellten Informationen (z.B. Source-Benutzer, -Datum, Hash) verarbeiten.
   1.2. Auslesen der Beschreibung und der Stichwörter aus dem Sourcecode.
2. Darstellung der Daten
   2.1. Die Anwendung muss eine Liste aller Module erzeugen inkl. Source-Benutzer und -Datum, letztem Commit-Benutzer und -Datum für alle drei Umgebungen.
   2.2. Verknüpfen der Module mit externen Tools wie z.B. Wiki-Einträgen zu den Modulen oder dem Sourcecode in Subversion.
   2.3. Die Sourcen der Umgebungen müssen verglichen und eine schnelle Übersicht zur Einhaltung des allgemeinen Entwicklungsprozesses gegeben werden.
   2.4. Dieser Vergleich muss auf die von einem bestimmten Benutzer bearbeiteten Module eingeschränkt werden können.
   2.5. Die Anwendung muss in dieser Liste auch Module anzeigen, die nach einer Bearbeitung durch den gesuchten Benutzer durch jemand anderen bearbeitet wurden.
   2.6. Abweichungen sollen kenntlich gemacht werden.
   2.7. Anzeigen einer Übersichtsseite für ein Modul mit allen relevanten Informationen zu diesem.
3. Sonstige Anforderungen
   3.1. Die Anwendung muss ohne das Installieren einer zusätzlichen Software über einen Webbrowser im Intranet erreichbar sein.
   3.2. Die Daten der Anwendung müssen jede Nacht bzw. nach jedem SVN-Commit automatisch aktualisiert werden.
   3.3. Es muss ermittelt werden, ob Änderungen auf der Produktionsumgebung vorgenommen wurden, die nicht von einer anderen Umgebung kopiert wurden. Diese Modulliste soll als Mahnung per E-Mail an alle Entwickler geschickt werden (Peer Pressure).
   3.4. Die Anwendung soll jederzeit erreichbar sein.
   3.5. Da sich die Entwickler auf die Anwendung verlassen, muss diese korrekte Daten liefern und darf keinen Interpretationsspielraum lassen.
   3.6. Die Anwendung muss so flexibel sein, dass sie bei Änderungen im Entwicklungsprozess einfach angepasst werden kann.

A.4 Pflichtenheft (Auszug)

Zielbestimmung

1. Musskriterien
   1.1. Modul-Liste: Zeigt eine filterbare Liste der Module mit den dazugehörigen Kerninformationen sowie Symbolen zur Einhaltung des Entwicklungsprozesses an
      - In der Liste wird der Name, die Bibliothek und Daten zum Source und Kompilat eines Moduls angezeigt.
      - Ebenfalls wird der Status des Moduls hinsichtlich Source und Kompilat angezeigt. Dazu gibt es unterschiedliche Status-Zeichen, welche symbolisieren in wie weit der Entwicklungsprozess eingehalten wurde bzw. welche Schritte als nächstes getan werden müssen. So gibt es z. B. Zeichen für das Einhalten oder Verletzen des Prozesses oder den Hinweis auf den nächsten zu tätigenden Schritt.
      - Weiterhin werden die Benutzer und Zeitpunkte der aktuellen Version der Sourcen und Kompilate angezeigt. Dazu kann vorher ausgewählt werden, von welcher Umgebung diese Daten gelesen werden sollen.
      - Es kann eine Filterung nach allen angezeigten Daten vorgenommen werden. Die Daten zu den Sourcen sind historisiert. Durch die Filterung ist es möglich, auch Module zu finden, die in der Zwischenzeit schon von einem anderen Benutzer editiert wurden.

   1.2. Tag-Liste: Bietet die Möglichkeit die Module anhand von Tags zu filtern.
      - Es sollen die Tags angezeigt werden, nach denen bereits gefiltert wird und die, die noch der Filterung hinzugefügt werden könnten, ohne dass die Ergebnisliste leer wird.
      - Zusätzlich sollen die Module angezeigt werden, die den Filterkriterien entsprechen. Sollten die Filterkriterien leer sein, werden nur die Module angezeigt, welche mit einem Tag versehen sind.

   1.3. Import der Moduldaten aus einer bereitgestellten CSV-Datei
      - Es wird täglich eine Datei mit den Daten der aktuellen Module erstellt. Diese Datei wird (durch einen Cronjob) automatisch nachts importiert.
      - Dabei wird für jedes importierte Modul ein Zeitstempel aktualisiert, damit festgestellt werden kann, wenn ein Modul gelöscht wurde.
      - Die Datei enthält die Namen der Umgebung, der Bibliothek und des Moduls, den Programmtyp, den Benutzer und Zeitpunkt des Sourcecodes sowie des Kompilats und den Hash des Sourcecodes.
      - Sollte sich ein Modul verändert haben, werden die entsprechenden Daten in der Datenbank aktualisiert. Die Veränderungen am Source werden dabei aber nicht ersetzt, sondern historisiert.

   1.4. Import der Informationen aus SVN. Durch einen „post-commit-hook“ wird nach jedem Einchecken eines Moduls ein PHP-Script auf der Konsole aufgerufen, welches die Informationen, die vom SVN-Kommandozeilentool geliefert werden, an NatInfo übergibt.

1.5. Parsen der Sourcen

- Die Sourcen der Entwicklungsumgebung werden nach Tags, Links zu Artikeln im Wiki und Programmbeschreibungen durchsucht.
- Diese Daten werden dann entsprechend angelegt, aktualisiert oder nicht mehr gesetzte Tags/Wikiartikel entfernt.

1.6. Sonstiges

- Das Programm läuft als Webanwendung im Intranet.
- Die Anwendung soll möglichst leicht erweiterbar sein und auch von anderen Entwicklungsprozessen ausgehen können.
- Eine Konfiguration soll möglichst in zentralen Konfigurationsdateien erfolgen.

Produkteinsatz

1. Anwendungsbereiche
   Die Webanwendung dient als Anlaufstelle für die Entwicklung. Dort sind alle Informationen für die Module an einer Stelle gesammelt. Vorher getrennte Anwendungen werden ersetzt bzw. verlinkt.
2. Zielgruppen
   NatInfo wird lediglich von den Natural-Entwicklern in der EDV-Abteilung genutzt.
3. Betriebsbedingungen
   Die nötigen Betriebsbedingungen, also der Webserver, die Datenbank, die Versionsverwaltung, das Wiki und der nächtliche Export sind bereits vorhanden und konfiguriert. Durch einen täglichen Cronjob werden entsprechende Daten aktualisiert, die Webanwendung ist jederzeit aus dem Intranet heraus erreichbar.

A.5 Netzpläne

Der Netzplan unserer DMZ in der Projektumgebung Der Netzplan unserer DMZ in der Testumgebung

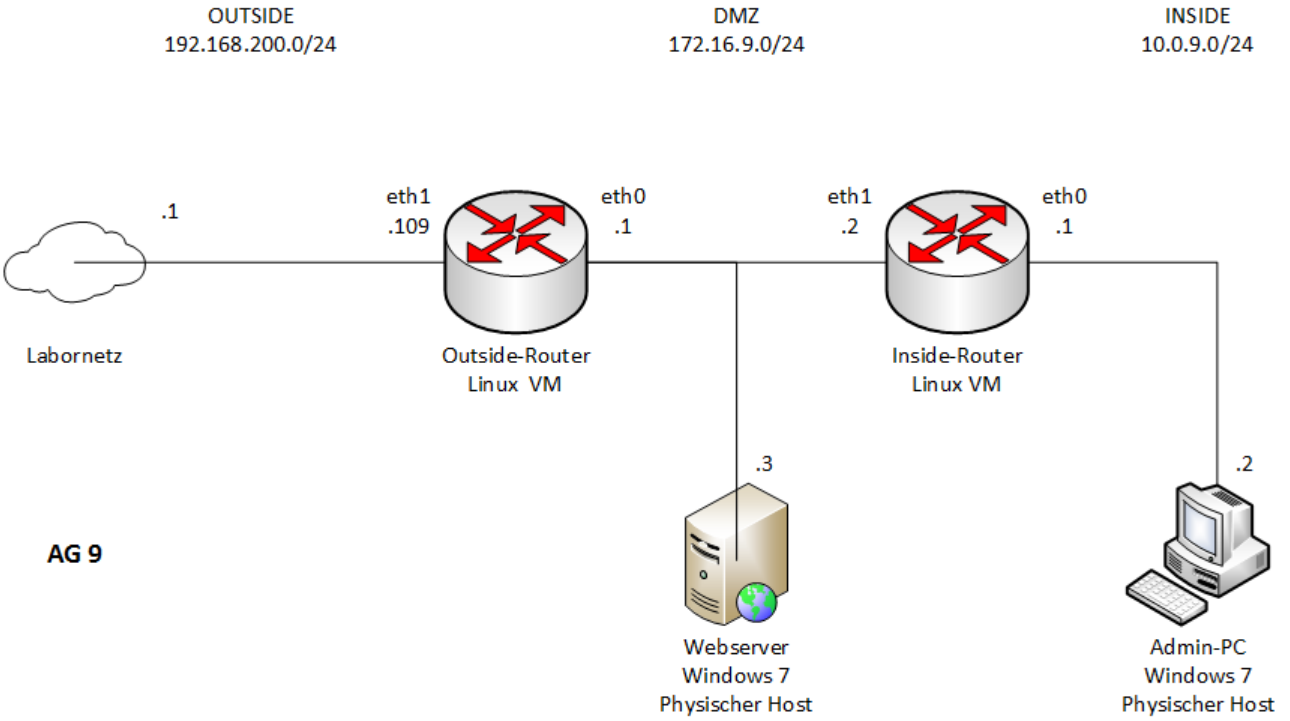


Abbildung 3: Netzplan der DMZ (Arbeitsgruppe 9)

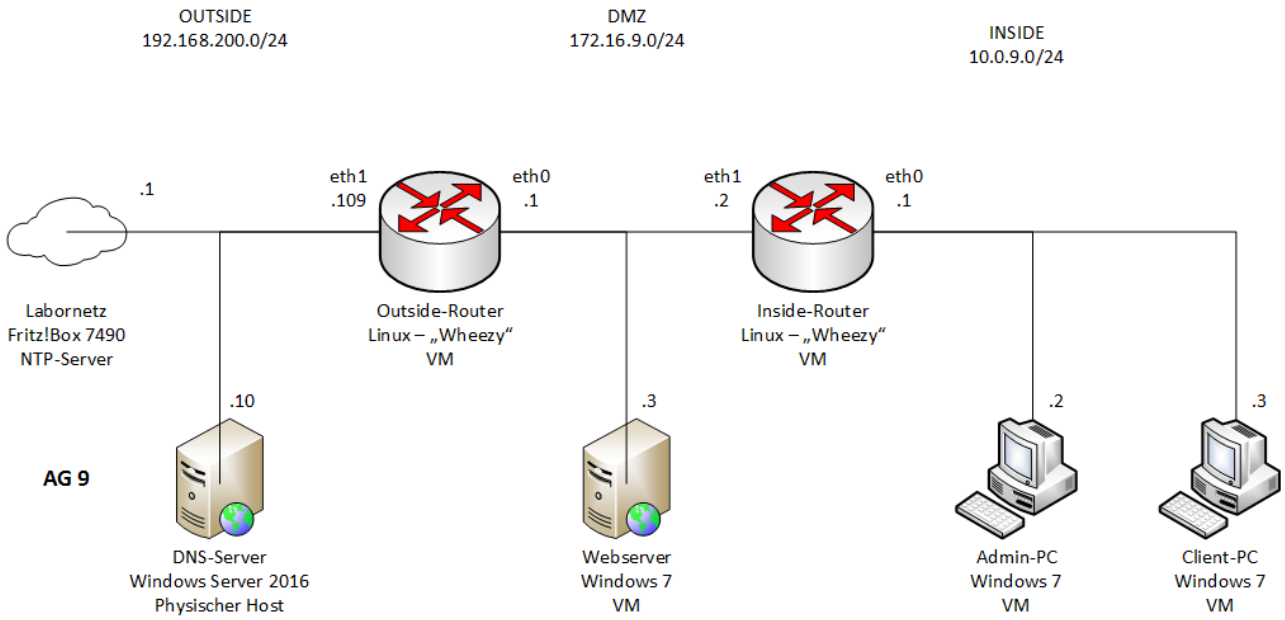


Abbildung 4: Netzplan der erweiterten DMZ in unserer Testumgebung

A.6 Testdokumentation

A.6.1 firewall.sh (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 ausfhrbar 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 berall (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 Kreativit 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 lschen
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 lschen
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 lschen
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 fr 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 fr 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 bergeben!"
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
```

A.6.2 firewall.sh (Inside-Router)

```
1  #!/bin/bash
2  # Bourne- Again Shell#
3
4  # ============================================================
5  # === Bemerkung ==============================================
6  # ============================================================
7
8  # Sekund Firewall
9  # ...verhindert unbefugten Zugriff vom lokalem Netz auf lokales Interface des Routers.
10 # Wenn die Firewall gestoppt ist, wird auch NAT gestoppt. Dies sorgt dafuer,
11 # dass auch alle internen Anfragen an das DMZ-Netz ber den Outside-Router laufen.
12 # Wenn die Firewall startet lt der Verkehr zwischen LAN und DMZ nur ueber den Inside-Router.
13 # Da NAT die IP des Admin-PCs uebersetzt, greifen die Zugriffsberechtigungen
14 # (ICMP, SSH) auf dem Outside-Router nicht. Daher wird wird er vom NAT ausgeschlossen.
15 #
16 #------Ende--Bemerkung--------------------------------
17
18
19 # ============================================================
20 # === Part 1: Variablen ======================================
21 # ============================================================
22 echo " - Variablen werden gesetzt"
23
24 # Pfad zu iptables
25 IPTABLES=/sbin/iptables
26
27 # Macht Linux-Maschine zu einem Router
28 echo "1" > /proc/sys/net/ipv4/ip_forward
29
30 # Interfaces
31 iINT=eth0
32 iEXT=eth1
33
34 # Definition DNS
35 DNS=("192.168.95.40/32 192.168.95.41/32")
36
37 # Timeserver: hier Standardgateway
38 TimeSrv=192.168.200.1
39
40 # Der Rechner, auf dem die Firewall (Inside) laufen soll , hier die VMWare
```

```
41 LinuxInside_in=10.0.9.1
42 LinuxInside_dmz=172.16.9.2
43
44 # Der Rechner, auf dem die Firewall (Outside) laufen soll, hier die VMWare
45 LinuxOutside_out=192.168.200.109
46 LinuxOutside_dmz=172.16.9.1
47
48 # Rechner fuer Fernwartung z.B. mit ssh, hier der Windowswirt (XP, Win7 o.ae.)
49 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 lopPath="/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 ===================================
```

```
# ====================================================
# ====================================================
stop)

# ====================================================
# === Part 2: Default Policy setzen ==================
# ====================================================

# ****************** Alles erlauben und alle Regeln loeschen
echo " - do: Policy and flush"
# Default policy setzen (Alles erlauben)
$IPTABLES -P INPUT ACCEPT
$IPTABLES -P FORWARD ACCEPT
$IPTABLES -P OUTPUT ACCEPT

# Loesche alle Filterregeln
$IPTABLES -F  # flush aller chains (Tabelle filter )
$IPTABLES -t nat -F # flush aller chains (Tabelle nat)
$IPTABLES -X  # delete all userdefined chains (Tabelle filter )

# ***** ENDE ******* NAT und Port-Forwarding
echo " - done: Policy and flush"


# ====================================================
# === Ausgabe ========================================
# ====================================================

# ****************** Konfiguration in DAtei umleiten
echo " - do: Schreibe Konfiguration in $lopPath"
echo -e "\n\n==================================================" >> $lopPath
date >> $lopPath
echo "Firewall gestoppt" >> $lopPath
echo -e "==================================================\n" >> $lopPath
$IPTABLES -L -v -n >> $lopPath

# ***** ENDE *********** Konfiguration in DAtei umleiten
echo " - done: Schreibe Konfiguration in $lopPath"

# ====================================================
# ***** ENDE *********** Firewall stoppen **************
# ====================================================
 ;;


# ====================================================
# ====================================================
# === Firewall starten ===============================
# ====================================================
# ====================================================
```

```
141 start)
142
143 # ======================================================
144 # === Default Policy setzen und NAT ====================
145 # ======================================================
146
147 # ****************** Alles verbieten und alle Regeln lschen
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 lschen
162 echo " - done: Policy and flush"
163
164
165 # ****************** NAT aktivieren ********************
166 echo " - do: NAT"
167 # NAT auf dem Inside-Router implementieren, Admin-PC auschlien
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 fr 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 fr 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 bergeben!"
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
```