29.06.2017

E-PORTFOLIO

M239

ROGER ZÜND / JEROME ACKERMANN

INHALTSVERZEICHNIS

1 ANFORDERUNGEN AUFNEHMEN

SICHERHEIT:

Die Sicherheit muss sowohl von aussen, als auch von Innen gewährleistet sein. Sowohl die Hardware, wie auch die Software müssen geschützt werden. Hauptsächlich geht es bei der Sicherheit um den Schutz der Daten einer Person oder Firma.

Bei der Hardware konzentriert sich der physische Schutz auf den Server. Dieser sollte in einem separaten, abgeschlossenen Raum stehen, zu dem nur berechtigte Personen Zugang haben. Es sollte eine Zugangskontrolle stattfinden (zb. mittels elektronischer Schlösser), um jederzeit feststellen zu können, wer wann auf die Hardware des Servers Zugriff hatte. Gegen Schäden des Servers durch Unfälle, Naturkatastrophen etc. sollte der Raum entsprechend geschützt sein, ausserdem sollte die Hardware des Servers redundant ausgelegt sein mit einem entsprechenden RAID und regelmässigen Backups der Daten.

Um das interne Netzwerk einer Firma vor Eindringlingen zu schützen, sollte vor allem eine gute Firewall an den Router gehängt werden, die nur durchlässt, was benötigt wird und alles andere ablehnt. Die Kommunikation nach aussen (Email etc.) sollte verschlüsselt geschehen. Der/die Server der Firma sollten in einer separaten Zone des Netzwerkes stehen am besten mit einer DMZ dazwischen. Damit wird ein unerlaubter Zugriff schon einmal extrem erschwert.

Sämtliche Zugriffe auf den Server sollten in detailreichen Logdateien festgehalten werden, um so im Nachhinein genau nachvollziehen zu können, wer wann etwas getan hat und von wo aus. Die Rechte der Nutzer des Servers sollten so weit wie möglich eingeschränkt werden, so dass die Nutzer tun können was sie wollen, aber keinen weiteren Zugriff haben.

LASTPROFIL

Um herauszufinden, wie sehr ein Server durch Zugriffe von Aussen belastet ist, sollten diese mit einem entsprechenden Programm gemessen und protokolliert werden. Die so gesammelten Daten werden dann zu einem Lastprofil zusammengefasst, in dem ersichtlich ist, wie viele Zugriffe es gab, wie lange die durchschnittliche Antwortzeit war, wieviele Daten versendet wurden und ob und wie viele Ausfälle es gab. Mit dem Lastprofil kann man schnell erkennen, wie es mit der Auslastung des Servers steht und ob es entsprechenden Handlungsbedarf gibt, d. h. ob die Serverinfrastruktur eventuell angepasst werden muss.

DATENVOLUMEN

Durch die Messungen weiss man, wie gross das Datenvolumen ist, das der Server durchschnittlich bearbeiten und versenden bzw. empfangen muss. Dadurch kann man erkennen, ob an der Infrastruktur (Kabel, Router, Switches, Firewall etc.) etwas geändert werden muss, um ein grösseres Volumen zu ermöglichen.

VERFÜGBARKEIT

Die Verfügbarkeit beschreibt, wie lange ein Gerät, in diesem Fall der Server im Netzwerk verfügbar, d. h. erreichbar war. Da ein nicht erreichbarer Server natürlich eine Katastrophe für jede Firma ist, sollte die Verfügbarkeit entsprechend hoch sein. Die Verfügbarkeit des Servers setzt sich aus mehreren Verfügbarkeiten zusammen, nämlich der des Servers und jedes Knotenpunktes davor wie z. B. die Firewall. Sollte z. B. die Firewall ausfallen ist der Server nicht mehr erreichbar, auch wenn er selbst eigentlich problemlos läuft. Da man auf das Internet keinen Einfluss nehmen kann, beschränkt sich die Beeinflussung der Verfügbarkeit auf das eigene Intranet. Man kann mit dem Provider einen Vertrag mit einer möglichst hohen Verfügbarkeit aushandeln und dann auch meist davon ausgehen, dass er sich auch daran hält.

Die Verfügbarkeit wäre im Idealfall 100%, allerdings ist das nicht garantier bar. Jedoch sollte die durchschnittliche Verfügbarkeit so nahe wie möglich an den 100% dran sein. 99% Verfügbarkeit klingt auf den ersten Blick gut. Jedoch bedeutet das, dass in einer von hundert Stunden der Server ausfallen würde. Auf ein Jahr gerechnet würden 99% eine durchschnittliche Ausfallzeit von ca. 87 ½ Stunden bedeuten, also mehr als 3 ½ Tage im Jahr. Das ist natürlich viel zu viel. Eine sehr gute Verfügbarkeit wäre z. B. 99.9999%. Das wären unter 6 Minuten Ausfallzeit im Jahr.

Um die Verfügbarkeit des eigenen Servers zu erhöhen, sollte er immer genug Kapazität haben, sowohl bei der Datenübertragung (Kabel, Firewall etc.) wie auch bei der Speicherung und Verarbeitung der Daten im Server (Festplatten, Prozessor etc.) Ausserdem sollte die Hardware des Servers nach Möglichkeit Redundant ausgelegt sein, sodass ein Ausfall eines Teils nicht direkt zum Absturz des Servers führt.

BEISPIELSZENARIO:

AUSGANGSLAGE:

Die Firma We like to move IT ist eine kleine Firma mit fünf Mitarbeitern, die verschiedene Dienstleistungen im Bereich Support und Lösungen der IT für KMU und Privatpersonen anbietet. Die Firma hat einst als Ein-Mann-Projekt des Chefs angefangen und damals wurde auch das Netzwerk entsprechend aufgebaut. Nun sind es bereits fünf Mitarbeiter und da die Firma grossen Erfolg hat ist die Tendenz stark steigend. Die Firma besitzt für jeden Mitarbeiter einen Laptop, der an den Router des Hauses angeschlossen ist. Es gibt einen Webserver, sowie einen Datenbank- und einen Dateiserver. Die Server stehen in einem offenen Raum in der Firma, jeder hat unkontrollierten Zugriff darauf. Auf alle Server kann von der Website aus zugegriffen werden. Vor allem der Dateiserver benötigt z. T. eine Menge Bandbreite. Die Firma ist über einen Router (Swisscom) mit dem Internet verbunden. Eine Firewall gibt es nicht.

In letzter Zeit hatte die Firma mehrere Server Ausfälle zu beklagen, die zum Teil Stunden dauerten. Es ist nicht klar, ob die Server einfach überlastet waren, oder angegriffen wurden. Da die Firma vermutlich weiterwachsen wird, will der CEO rechtzeitig das Netzwerk verbessern.

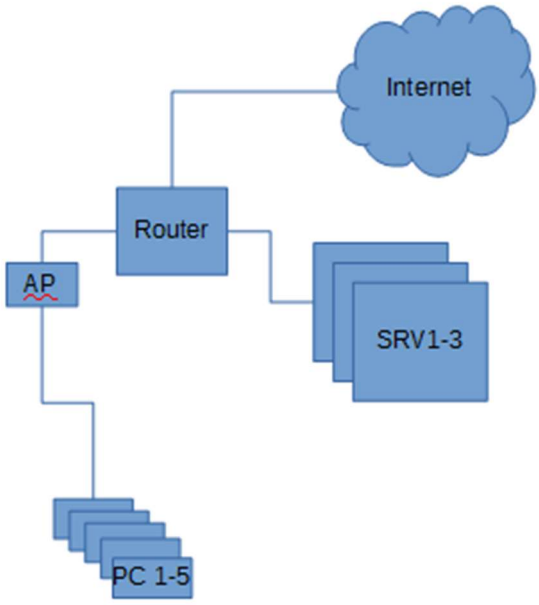
ZIELE:

Alle Dienste der Server müssen funktionieren

- Die Umgebung ist jederzeit ohne Einschränkung ausbaubar.
- Der Preis muss so günstig wie möglich ausfallen.

IST-SITUATION:

NETZWERKPLAN:



Es sind im Moment 5 Laptops über einen Access Point an das Netzwerk angeschlossen, mit denen die Mitarbeiter und der CEO arbeiten. Die Server hängen ebenfalls direkt am Router. Von der Swisscom hat die Firma eine öffentliche IP, die vom Router in Private IPs umgewandelt werden. Der verwendete IP Range ist 192.168.1.X

SICHERHEIT:

Es gibt keine Firewall. Dadurch ist der Schutz gegen Hackerangriffe absolut nicht gegeben, das Tor steht quasi für jeden weit offen.

1. Es gibt keinerlei Benutzererkennung wie etwa durch eine Domäne mit Benutzersystem.
2. Die Server sind nicht in einer separaten Zone untergebracht, eine DMZ existiert nicht.
3. Der Physische Server kann von jedem beliebigen Menschen erreicht werden, ohne dass es bemerkt wird.

LASTPROFIL:

Nach Messungen auf allen drei Servern wurde ein Lastprofil für die Server erstellt.

Beschreibung	Ergebnisse		
Uhrzeiten	00.01 - 08.00 Uhr	08.01 - 16.00 Uhr	16.01 - 00.00 Uhr
Anzahl Anfragen	200	800	500
Datenvolumen	800 MB	3500 MB	2000 MB
Latenzzeit	150ms	250ms	170ms
Anzahl Ausfälle	1	3	2

Was einem direkt an dem Ergebnis auffällt, sind die relativ hohen Latenzzeiten, sowie die Ausfälle, die passiert sind. Beides ist ein Problem, die Server sind eindeutig überlastet und müssen in ihrer Kapazität erweitert werden.

DATENVOLUMEN:

Auch wenn das eigentlich mögliche Datenvolumen noch nicht erreicht ist, ist der Server und das Netzwerken zu Spitzenzeiten ausgelastet, was zu den Abstürzen führt. Vor allem das Netzwerk muss deutlich verbessert werden.

VERFÜGBARKEIT:

An einem durchschnittlichen Tag 6 Serverausfälle zu haben, ergibt eine ziemlich schlechte Verfügbarkeit. Der Server sollte dringend aufgerüstet und das Netzwerk erweitert werden.

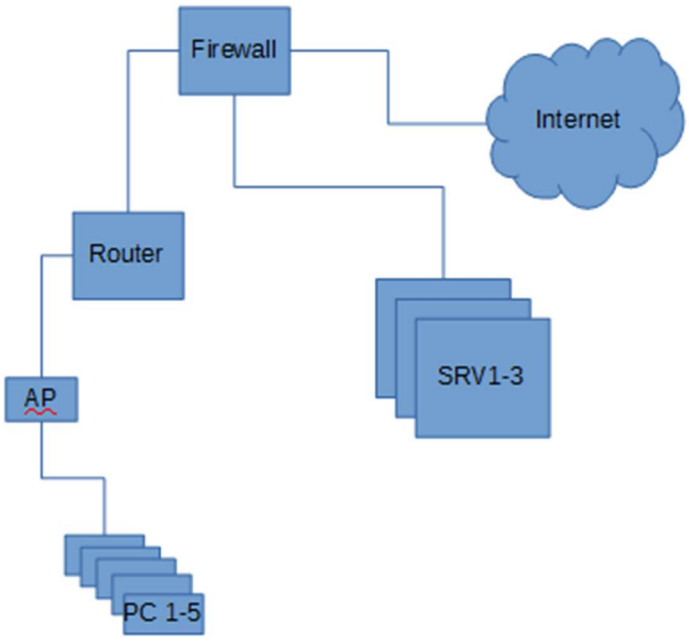
APPLIKATIONEN:

Im Moment laufen auf einem Hardware Server drei verschiedene Server Funktionen, die eine Menge Ressourcen benötigen. Diese sollten eventuell aufgespaltet und überarbeitet werden.

SOLL-SITUATION:

NETZWERKPLAN:

So sollte das neue Netzwerk aussehen:



Die Laptops bleiben über den Access Point mit dem Router verbunden. Dieser ist jedoch neu durch eine Firewall vom Internet und den Servern getrennt. Die Server befinden sich in einer Einstufigen DMZ, in einem separaten IP Range.

SICHERHEIT:

1. Da nun eine Firewall mit strengen Einstellungen aktiv ist, ist die direkte Gefahr durch Hacker eingedämmt.
2. Auf dem Server wird zusätzlich eine Benutzererkennung eingerichtet. (Z. B. im Falle eines Windows Servers eine Domäne mit Active Directory).
3. Eine DMZ wurde erstellt, sodass der Server nun vom Rest des Netzes getrennt ist.
4. Der Physische Server wird in einen abschliessbaren Raum verlegt, zu dem nur der CEO bzw. der zuständige Systemadmin zutritt hat.

LASTPROFIL:

Mit der neuen Konfiguration wird erneut die Belastung gemessen.

Das Ziel ist es, die Ausfälle bei annähernd 0 zu halten, was auch gelingen sollte.

Beschreibung	Ergebnisse		
Uhrzeiten	00.01 - 08.00 Uhr	08.01 - 16.00 Uhr	16.01 - 00.00 Uhr
Anzahl Anfragen	210	770	550
Datenvolumen	850 MB	3200 MB	2400 MB
Latenzzeit	<10ms	<20my	<15ms
Anzahl Ausfälle	0	0	0

VERFÜGBARKEIT:

Auch wenn die Verfügbarkeit durch weniger Ausfälle kurzfristig besser geworden ist, kann nur eine Redundanz des Systemes und Netzwerkes sie langfristig verbessern.

APPLIKATIONEN:

Die Ausführung der drei Hauptaufgaben (Webserver, Datenbankserver, Dateiserver) liegt weiterhin an einem Server, das sollte geändert werden. Zusätzlich kam noch ein Active Directory Domänendienst für die Benutzererkennung hinzu. In Zukunft wird wohl auch noch der Mailserver hinzukommen.

2 BESTEHENDE INFRASTRUKTUR ANPASSEN / ERWEITERN

TECHNISCHE SPEZIFIKATIONEN EINES INTERNETSERVERS

Je nach dem, was auf dem Internet Server für Dienste ausgeführt werden müssen, braucht man verschiedene Hard und Software Anforderungen

WEBSERVER

Wenn man einen ganz normalen Webserver aufsetzen möchte mit einer kleinen Webseite ohne grosse Applikationen dahinter, reicht ein kleiner Raspberry Pi. Der hat 512MB Ram und einen schwachen Prozessor.
Wenn man aber grössere Webseiten mit Datenbanken, Login und Diversen Applikationen möchte, reicht ein solch kleiner Webserver nicht mehr. Dann wird je nach Anwendung empfohlen. Gut dabei ist man sicher, wenn man 8GB RAM und ab einem Intel Core i3 Prozessor im Webserver verbaut hat. Auf die Netzwerkschnittstelle und Internetleitung kommt es natürlich auch immer noch an. Bestenfalls hat man Glasfaser Anschluss mit entsprechendem Abonnement und eventuell auch noch Porttrunking.

MAILSERVER – EXCHANGE

Bei einem Exchange Mail Server braucht man mehr Rechenpower. Es kommt immer auf die Rollen darauf an, damit man sagen kann, was für Komponenten benötigt werden. Man sollte einen 64BIT Prozessor, 8GB RAM und mindestens eine 30GB grosse Festplatte haben.

NAMENSAUFLÖSUNG / DNS

DNS heisst Domain Name System. Jedes Gerät im Internet erhält eine eindeutige IP Adresse. Diese kann sich je nach dem auch wieder ändern. Die IP Adresse wird benötigt, um eine Verbindung zu einem Server aufbauen zu können. DNS dient zur Auflösung von Computernamen in IP-Adressen und umgekehrt. Diese Informationen sind auf vielen Tausenden Nameservern verteilt, die auf der ganzen Welt verteilt stehen.
Möchte man zum Beispiel auf die Webseite www.google.ch, dann fragt der Browser einen DNS-Server. In der Regel ist das der Router des Internet-Zugangs. Wenn der Router die Adresse nicht kennt, fragt er einen anderen DNS Server. Dies geht so lange, bis ein DNS Server die Adresse weiss. DNS greift auf die Datei hosts zurück. Deren Inhalt dient zur Namensauflösung. Der Nachteil der Datei ist, dass diese durch einen Virus manipuliert werden kann, und so auf falsche Webseiten umgeleitet werden kann. DNSSEC ist ein Verfahren um zu prüfen, ob eine DNS Antwort (DNS-Response) vertrauenswürdig erscheint und ob der Transport unverfälscht erfolgt ist.
Ein Domain Name ist dazu verantwortlich, kaum merkbare IP Adressen richtige Namen zu geben. Ein solcher Domain Name ist zum Beispiel: google.ch. Hinter dieser Adresse befindet sich nichts weiteres als eine öffentliche IP Adresse: 172.217.16.131. Diese Namen sind hierarchisch unterteilt.
Häufig sind diese Namen Teil einer URL (Uniform Resource Locator). URL beginnt mit einem Kürzel den man auch als Dienst bezeichnet → www. / ftp. / …
Die Struktur für einen Domain-Name ist so aufgebaut:

www.	google.	ch
Host oder Dienst	Second Level Domain	Top Level Domain

Ein Domain-Name wird immer von hinten nach vorne gelesen. Dort beginnt die Adresse mit der **Top-Level-Domain (TLD).** Man unterscheidet zwischen zwei Typen von Top-Level-Domains. Geografische Top-Level-Domains, die Ländercodes. Dann gibt es noch die organisatorischen oder generischen Top-Level-Domains (Generic Top-Level-Domain, gTLD).
Die **Second-Level-Domain** kann von einer Person oder einer Firma beantragt und eingesetzt werden. Der Name kann frei ausgewählt werden. Ein Beispiel wäre google.
Für weitere Unterteilungen existiert noch eine **Third-Level-Domain**, die auch als Sub-Level-Domain oder Subdomain bezeichnet wird. Ganz am Ende der Kette (am Anfang des Domain-Namens) wird dann der optionale Hostname des Computers eingesetzt.
Eine so zusammengesetzte Adresse (zum Beispiel www.google.ch) ist ein sogenannter Fully Qualified Domain Name (FQDN).

Einträge in einer DNS-Zone bzw. Zonendatei
Die Einträge in einer DNS-Zone werden als Ressource Records bezeichnet. Jeder Ressource Record bezieht sich auf einen anderen Record-Type, der eine bestimmte Information enthält. Zum Beispiel eine IP-Adresse oder die Mailserver-Adresse eines Domain-Namens bzw. der Zone.
Record-Type / Eintrag
A / IPv4-Adresse
AAAA / IPv6-Adresse
CNAME / Verweis, Weiterleitung oder Alias
MX / zuständiger Mailserver für die Zone (Mail Exchange)
NS / zuständiger Nameserver für die Zone
SRV / Server für einen Dienst im Windows-AD
TXT / liefert einen Text zurück
SOA / Ansprechpartner und Parameter zur abgefragten Zone (SOA: engl. für Start of Authority)
Ebenfalls gibt es noch verschiedene DNS Server die für andere Verantwortungsbereiche zuständig sind.
DNS Root Server

- Sie beantworten Anfrage zur Root-Zone oder geben eine Liste alternativen Namensservern für eine bestimmte Top-Level-Domain zurück (ch, de, com)

Autoritativer Namensserver

- Ist für Zonen zuständig und beantwortet auch nur Anfragen für diese Zonen. Bedeutet die Informationen dieses Namensservers gelten als verbindlich.

Nicht autoritativer Namensserver

- Ist nicht selbst für eine DNS-Zone verantwortlich und muss daher Informationen aus zweiter oder dritter Hand mittels einer DNS Abfrage ermitteln.

DNS Protokoll
DNS ist auf der Anwendungsschicht des OSI-Schichtenmodells angeordnet. Deshalb nutzt es zur Übertragung TCP und UDP auf dem Port 53.

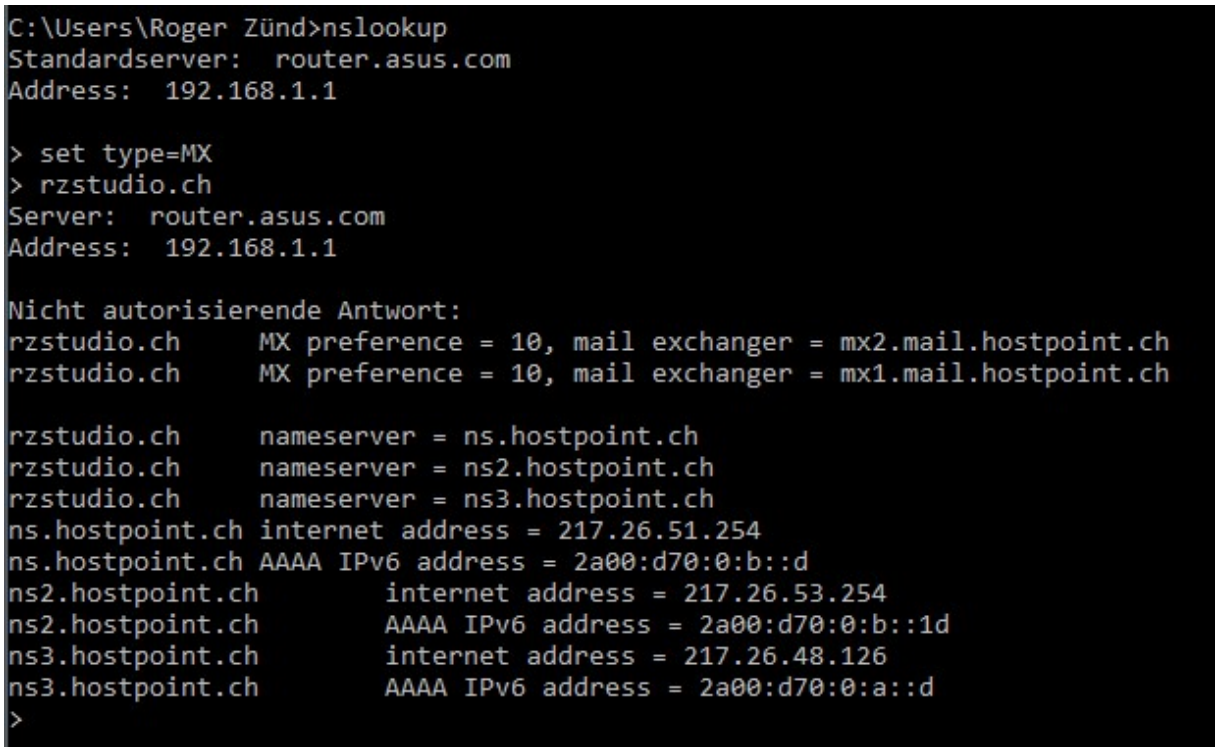
VIRTUAL HOSTS

Virtual Hosts sind dazu da, um auf einem Webserver mehrere Webseiten mit unterschiedlichen Domains anzuzeigen. Der Webserver wird so konfiguriert, dass er für verschiedene Hostnamen verschiedene Inhalte liefert. Jede dieser Webseiten ist danach ein Virtual Host.

SERVER HERAUSFINDEN

Um zum Beispiel einen Mail Server herauszufinden von einer Domain gibt es einen einfache Befehl den man im CMD ausführen muss.

1. Man gibt im CMD nslookup ein
2. Man gibt danach set type=MX ein (MX sind die DNS Einträge für den Mailserver)
3. Man gibt die Domain an rzstudio.ch



PROXY

Wenn man eine Normale Internet Anfrage startet, zum Beispiel auf www.rzstudio.ch, dann verbindet man sich direkt vom PC aus auf den entsprechenden Webserver. Hat man aber einen Proxy Server aktiv, verbindet man sich zuerst auf den Proxy Server, diesen macht dann die Anfrage auf www.rstudio.ch mit seiner IP Adresse, bekommt eine Antwort und schickt mir diese wieder zurück. Ein Vorteil davon ist, dass man seine IP Adresse verschleiern kann. Man kann den PC der die Anfrage über den Proxy Server gemacht hat nicht mehr so einfach zurückverfolgen.
Ein weiterer Vorteil davon ist Caching. Hierbei werden Daten vom Proxy zwischengespeichert um wiederkommende Anfragen schneller zu liefern.
Noch ein Vorteil ist das Filtern. Proxy-Server können auch durch Filter die eingehenden Anfragen auswerten und so nur bestimmte Anfragetypen durchlassen. Dies ist oft in Firmen-Netzwerken der Fall, in denen bestimmte Dienste, wie beispielsweise Filesharing- oder Porno-Websites, geblockt werden.

BETRIEBS- UND SICHERHEITSEINSTELLUNGEN

BETRIEBSKONZEPT

BACKUP

Backups müssen regelmässig ausgeführt werden um den Internetserver in Betrieb zu halten. Um den Server in auf Betrieb zu halten, empfiehlt sich die Festplatten via RAID so zu betreiben, dass die Festplatten gespiegelt werden, und so ein reibungsloser Auswechseln von einer Festplatte betreiben zu können.
Das Backup sollte mehrmals vorhanden sein an verschiedenen Standorten

UPDATES

Updates müssen regelmässig ausgeführt werden. Bestenfalls testet man die Updates vor dem Patchen des Systems und erstellt zuvor ein Backup des Servers. Ansonsten müssen Sie immer so schnell wie möglich installiert werden.

USV

Um einen Ständigen Betrieb zu gewährleisten und den Server zu schützen, sollte der Server an einer USV, einer Batterie, angeschlossen werden. So verhindert man den sofortigen Absturz des Servers durch einen Stromausfall oder kann kürzere Stromausfälle überbrücken.

ERSATZTEILE

Um einen Server möglichst lange in Betrieb zu halten, sollte man zuvor abklären, ob auch noch in ein paar Jahren Ersatzteile vorhanden sind. Festplatten die mit RAID Aufgesetzt wurden nützen einem nichts ohne den zugehörigen RAID Controller.

BETRIEBSDOKUMENTATION

Es sollte eine Betriebsdokumentation erstellt werden, die für den Administrator des Systems ist. Dort drin wird das Notfallkonzept wie auch alle anderen Daten die für den Betrieb der Server benötigt werden. Konfigurationen und Einstellungen die an den Server vorgenommen wurden werden dort drin beschrieben.
Um den Server Warten zu können werden ebenfalls die dazu benötigten Hilfsprogramme und Befehle in die Dokumentation mit ein bezogen.
Diese Betriebsdokumentation muss stets auf dem neusten Stand gehalten werden, ansonsten nützt diese nicht viel.

SICHERHEITSKONZEPT

HÄRTEN DES SYSTEMS

Das System sollte gehärtet werden. Das Heisst es sollte am Anfang alles verboten werden. Nur die Sachen die man braucht wollte man in einer Ausnahmeregelung einfügen. Dies benötigt eine Einarbeitungszeit, ist jedoch sehr effektiv.

ANTIVIRENSCHUTZ

Um den Computer auch vor Viren und Malware schützen zu können, benötigt es ein gutes Antiviren Programm, wie den Microsoft Defender.

BENUTZERSCHULUNG

Ebenfalls gehört eine Benutzerschulung in das Sicherheitskonzept. Dort müssen die Benutzer über die alltäglichen Gefahren (Wie E-Mails mit Malware) aufgeklärt werden und Ihnen eine Ansprech-Person zur Verfügung stellen.

BENUTZERRICHTLINIEN

Zu den Benutzerrichtlinien gehört es, dass man die Benutzer einschränkt. Die Benutzer müssen in sinnvollen Abständen Ihre Passwörter ändern, sowie die Rechte auf Daten eingestellt werden. Niemand arbeitet mit einem Admin-Konto.

3 PROTOKOLLE

HTTP (HYPERTEXT TRANSFER PROTOCOL)

Das Hypertext transfer protocol ist ein Zustandsloses Protokoll zur Übertragung von Daten auf der Anwendungsschicht über ein Netzwerk. Hauptsächlich wird es verwendet um Webseiten und andere Dokumente aus dem WWW für einen Webbrowser bereitzustellen. Neben HTTP gibt es noch HTTPS (secure) welches mittels SSL verschlüsselt ist.

Wie bereits erwähnt ist HTTP zustandslos, das heisst es kann keine Daten speichern und sieht jede Anfrage als separat an. Es kann nicht von selbst Anfragen verknüpfen. Nur mit Cookies im Header kann ein Nutzer erkannt werden.

HTTP funktioniert relativ simpel. Es gibt eigentlich nur Anfragen (requests) und Antworten (responses). Jede Nachricht besteht aus zwei Teilen, dem Header und dem Body. Der Header enthält zuerst die URL, die erreicht werden möchte, sowie die HTTP Version. Dann kommen zusätzliche Informationen. Im Body sind die effektiv angefragten Daten.

Der Standard Port für HTTP ist 80, der für HTTPS ist 443.

HTTP kann standardisierte Fehlermeldungen zurückgeben.

1xx = Informationen

2xx = Erfolge

3xx = Umleitungen (siehe Beispiel unten)

4xx = Client-Fehler (z.B. 404, file not found)

5xx = Server-Fehler

GET / HTTP/1.1

Host: www.wikipedia.ch

User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: de,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1

HTTP/1.1 302 Found
Date: Thu, 08 Jun 2017 07:12:21 GMT
Server: Apache
Location: http://portal.wikimedia.ch/wikipedia
Content-Length: 286
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=iso-8859-1

Der Browser fragt nach wikipedia.ch. Nach dem herausfinden der IP mittels DNS schickt er die ANfrage. Vom Server erhält er die Antwort, in der steht, dass die Adresse falsch ist, es steht auch direkt die richtige Adresse drin, mit der der Browser dann eine neue Anfrage starten kann. Da es eine Umleitung ist, gibt der Server den Fehlercode 302 zurück.

METHODEN:

HTTP unterstützt mehrere Anfragemethoden die andere auswirkungen haben. Die wichtigsten beiden sind GET und POST.

GET

Get ist eine der gebräuchlichsten Anfragemethoden für HTTP. Mit ihr wird eine Ressource unter Angabe einer URL vom Server gefordert. In der URL können Argumente mit übertragen werden. (DIese sieht man dann z. B. im Browser in der Adresszeile). Theoretisch kann man also mit GET Daten an den Server senden, allerdings sollte es nur zur Abfrage verwendet werden.

POST

Post kann je nach ausstattung des Servers quasi unbegrenzt Daten an den Server übermitteln. Diese sollen meist vom Server weiterverarbeitet werden. Meistens sind die übertragenen Daten Formulardaten die mit einer Datenbank abgeglichen werden sollen. (Z. B. Name und Passwort für eine Anmeldung).

HEAD

Head weist den Server an, die gleichen HTTP-Header wie bei GET zu senden, jedoch ohne den eigentlichen Inhalt der Datei (also ohne Body). SO kann man z. B. die gültigkeit einer Datei im Browser Cache überprüfen ohne sie nochmals komplett zu laden.

PUT

Mit Put kann man eine Datei unter Angabe einer URL auf den Server hochladen. Sollte schon eine Datei gleichen Namens bestehen, wird diese ersetzt.

PATCH

Patch macht dasselbe wie Put, allerdings ohne eine Datei zu ersetzen. Patch wurde nach Post dem Standard hinzugefügt.

DELETE

Delete löscht eine Ressource vom Server.

TRACE

Trace liefert eine Anfrage genau so zurück, wie der Server sie empfangen hat. Es wird genutzt um zu prüfen, ob die Anfrage verändert wurde.

OPTIONS

Options liefert eine Liste der vom Server unterstützten Methoden.

SMTP (SIMPLE MAIL TRANSFER PROTOCOL)

SMTP wird verwendet um E-Mails in Netzwerken zu versenden. (Meist im Internet). SMTP kann keine Mails empfangen, dafür werden POP3 oder IMAP verwendet. Der Standardport von SMTP ist 25. SMTP kann auch den Port 587 nutzen. SMTP wird von Mailservern verwendet (Mail Submission Agents und Mail Transfer Agents), sowie von einem Mail Client (Mail User Agent) wie z. B. Microsoft Outlook.

Das SMTP hat einen standardisierten ablauf für die Übermittlung einer Mail Nachricht. Da man mittels z. B. Telnet auch "von Hand" eine Mail schreiben kann, kann man auch den Absender beliebig ändern. Die Authentizität eines ABsenders ist also nicht sicher.

MÖGLICHER ABLAUF EINER VERBINDUNG:

Client	Server	Erklärung
telnet mail.example.ch 25		Client ruft Server
	220 service ready	Server meldet sich
HELO client.example.net		Client authentifiziert sich
	250 ok	Server bestätigt
MAIL FROM:<sender@example.ch>		Client nennt Absenderadresse
	250 ok	Server bestätigt
RCPT TO:<receiver@example.ch>		Client nennt Empfängeradresse
	250 ok	Server bestätigt
DATA		Client kündigt Inhalt an
	354 start mail input	Server meldet bereitschaft
From: <sender@example.ch> To: <receiver@example.ch> Subject: Beispielmail Date: Thu, 08 Jun 2017 10:44:55 +0100 Dies ist der Inhalt des Beispielmails.		Client sendet Inhalt der Mail Im Header die Angaben, im Body die Nachricht
	250 ok	Server bestätigt

QUIT		Client fordert trennung der Verbindung
	221 closing channel	Server kündigt trennung an

Der Server nutzt standardisierte Nummern und passende kurze Erklärungen zur Kommunikation. (250 steht z. B. für okay, bestätigt)

1xx = Anforderung akzeptiert aber noch nicht bestätigt

2xx = Anforderung ausgeführt

3xx = Anforderung verstanden, mehr Informationen benötigt

4xx = Temporärer fehler festgestellt

5xx = Fataler fehler festgestellt, verarbeitung nicht möglich

POP (POST OFFICE PROTOCOL)

POP ist ein Protokoll, das zur Übertragung von E-Mail Nachrichten verwendet wird. Es kann E-Mails vom Server auf den Client übertragen. Im Moment wird POP3 verwendet.

POP3 ist sehr einfach strukturiert. Es erlaubt nur das Auflisten, Abholen und löschen von E-Mails auf Servern. Für weitere Funktionen muss zwangsläufig IMAP verwendet werden.

Der Standardport von POP3 ist 110 bzw. 995 für verschlüsselte Nachrichten.

VOR- UND NACHTEILE VON POP3:

Vorteil	Nachteil
• Es ist keine permanente Verbindung zum Mailserver erforderlich. • Die Verbindung wird bei Bedarf vom Client aufgebaut und beendet • Nach der Anmeldung werden alle E-Mails vom Server heruntergeladen.	• Es findet keine Synchronisierung zwischen Server und CLient statt. Wenn man z. B. eine Mail auf dem Server löscht, wird dies nicht auf den Client übertragen.

Folgende Standardkommandos werden von einem POP3 Server unterstützt:

USER xxx
wählt den Benutzernamen auf dem Mailserver.

PASS xxx
übergibt das Passwort (im Klartext!)

STAT
liefert den Status der Mailbox, u. a. die Anzahl aller E-Mails im Postfach und deren Gesamtgröße (in Byte).

LIST (n)
liefert die Anzahl und die Größe der (n-ten) E-Mail(s).

RETR n
holt die n-te E-Mail vom E-Mail-Server.

DELE n
löscht die n-te E-Mail am E-Mail-Server.

NOOP
keine Funktion, der Server antwortet mit +OK.

RSET
setzt alle DELE-Kommandos zurück.

QUIT
beendet die aktuelle POP3-Sitzung und führt alle DELE-Kommandos durch.
Optionale Kommandos (serverabhängig):

TOP n x
ruft den Header und die ersten x Zeilen der n-ten Mail ab.

UIDL n
zeigt die eindeutige ID der E-Mail an.

MÖGLICHER ABLAUF EINER VERBINDUNG:

Client	Server	Erklärung
öffnet Verbindung		Client öffnet Verbindung
	+OK example.ch POP3-Server	POP3 Server bestätigt Verbindung
USER testuser@example.ch		Client authentifiziert sich
	+OK mailbox locked and ready	POP3 bestätigt und lädt Mailbox
STAT		Client fragt nach Serverstatus
	+OK 1 236	Server antwortet. 1 = Alle Mails im Posteingang, 236 = Gesamtgrösse des Posteinganges in Bytes
LIST		Client fragt nach einer Liste aller E-Mails
	+OK mailbox has 1 massages (236 octets) 1 236 .	Server antwortet (selbe Angaben wie bei STAT)
RETR 1	+OK message follows	Client verlangt nach der ersten Mail im Posteingang Server übergibt Mail, zuerst den Header mit den ANgaben, dann der Body mit dem Inhalt.

	Date: Mon, 05 Jun 2017 09:12:40 +0100 From: beispieluser <beispieluser@example.ch> To: testuser@example.ch Subject: Test E-Mail Content-Type: text/plain; charset=ascii; format=flowed Content-Transfer-Encoding: 7bit Dies ist der Inhalt der Test Email. .	
DELE 1		Client befiehlt die erste Mail im Posteingang zu löschen
	+OK massage marked for delete	Server bestätigt (führt aber noch nicht aus)
QUIT		Client beendet verbindung
	+OK bye	Server bestätigt Danach löscht er die E-Mail

IMAP (INTERNET MESSAGE ACCESS PROTOCOL)

IMAP ist ein Protokoll, das dazu verwendet wird E-Mail Nachrichten zu empfangen. Es ist eine weiterentwicklung des POP, die erstellt wurde, um dem Nutzer mehr Möglichkeiten zu bieten. Der Nutzer kann Mails, Ordnerstrukturen und einstellungen nun direkt auf dem Server speichern und dort belassen. Die Clients greifen direkt Online auf die Funktionen zu. Sie laden nur noch kopien lokal herunter. Alles was der Nutzer abfragt wird direkt vom Server geladen und angezeigt. Die Daten verbleiben auf dem Server. Wenn man nun mit einem anderen Gerät auf den Server zugreift, erhält man exakt das gleiche ergebnis.

Mit IMAP sind verschiedene Sachen möglich, wie z. B. mit mehreren Benutzern eine Mail zu bearbeiten und dazu entsprechende Berechtigungen zu verteilen. Der Server kann Mails auch direkt filtern und entsprechend einteilen. Auch Push-Nachrichten bei erhalt einer Mail sind möglich. (So muss der Client nicht immer wieder nachfragen, was den Datenverkehr verkleinert). Der einzige Nachteil von IMAP ist, das man Online sein muss, um auf die Mails zuzugreifen. Einige Clients lösen dieses Problem, indem sie lokale Kopien der Mails erstellen.

MÖGLICHER ABLAUF EINER VERBINDUNG:

Client	Server	Erklärung
	* OK IMAP4rev1 Service Ready	Server begrüsst Client
a001 login mrc secret		Client meldet sich an
	a001 OK LOGIN completed	Server bestätigt anmeldung
a002 select inbox		Client wählt inbox als aktiven Ordner
	* 18 EXISTS * FLAGS (\Answered \Flagged \Deleted \Seen \Draft) * 2 recent * OK (UNSEEN 17) Message 17 is the first unseen message a002 OK [READ-WRITE] SELECT completed	18 Mails vorhanden definierte Flags 2 mails sind als dringlich markiert Mail Nummer 17 ist noch nicht gelesen.

		Client darf änderungen an Mails durchführen
a003 fetch 12 full		Client fordert Informationen zu Mail nummer 12
	* 12 FETCH (FLAGS (\Seen) INTERNALDATE "17-May-2017 03:44:25 -0100" RFC822.SIZE 4286 ENVELOPE ("Wed, 17 May 2017 03:40:25 -0100 (PDT)" "IMAP4rev1 WG Beispielbetreff" (("J. Ackermann" NIL "ackermann" "test.example.ch")) (("J. Ackermann" NIL "ackermann" "test.example.ch")) (("J. Ackermann" NIL "ackermann" "test.example.ch")) ((NIL NIL "imap" "test.example.ch")) (("Roger Zuend" NIL "zuend" "test.ch")) NIL NIL <B27397-0100000@test.example.ch>") BODY ("TEXT" "PLAIN" ("CHARSET" "ASCII") NIL NIL "7BIT" 302892)) a003 OK FETCH completed	Mail wurde bereits gelesen Am 17 Mai 2017 zugestellt Über 4kb gross Mail header: Datum Betreff Absender Absender Antwort-an Empfänger Kopie

		Infos
a005 store 12 +flags \deleted		Mail Nr. 12 als gelöscht markieren
	* 12 FETCH (FLAGS (\Seen \Deleted)) a005 OK +FLAGS completed	Server passt Flags an (markiert Mails für löschung)
a006 logout		Client meldet sich ab
	* BYE IMAP4rev1 server terminating connection a006 OK LOGOUT completed	Server beendet verbindung

VOR- UND NACHTEILE VON IMAP:

Vorteil	Nachteil
• Nachrichten werden separat auf dem Server gespeichert • Schneller erster Zugriff auf den Briefkasten • Der Inhalt des Briefkastens ist immer auf dem neuesten Stand	• Für jede ungelesene Nachricht muss eine Verbindung zum Server hergestellt werden • Um die Kopie einer gesendeten Nachricht zu speichern, muss diese ein zweites Mal hochgeladen werden • Höhere Serverbelastung - insbesondere beim Suchen und Sortieren

4 SOFTWARE INSTALLIEREN / KONFIGURIEREN

BETRIEBSSYSTEM

Als Betriebssystem haben wir uns für Debian in der Version 8.8 entschieden. Sicherheitstechnisch ist man schon auf einem sehr guten Zweig, wenn man mit Linux unterwegs ist, da es schlicht weniger Viren für dieses Betriebssystem gibt. Als weiteres Sicherheits und Leistung Feature haben wir keine Oberfläche installiert. Das steigert die Effizienz des Betriebssystems, spart Leistung und bietet eine kleinere Angriffsfläche des Systems

WEBSERVER

Als Webserver haben wir uns für Apache 2.4.10 entschieden.

INSTALLATION

Informationen zur Installation findet man auf folgender Webseite: https://wiki.ubuntuusers.de/Apache_2.4/. Wir haben den Webserver so wie es in der Anleitung steht installiert. Die Installation ging schnell und einfach von statten.

KONFIGURATION

An der Konfiguration des Webservers haben wir nicht viel geändert. Den Webserver erreicht man über den Port 80 und unter dem Verzeichnis `/var/www/html/` werden die Webseitendaten abgelegt. Wir haben keine Zusatzmodule installiert. Hier kann man die Fertigen HTML Daten direkt hochladen. Den Server haben wir so konfiguriert, dass ein SSH Zugang möglich ist. So kann man beispielsweise mit einem SSH Client wie FileZilla oder WinSCP auf den Server zugreifen um diesen so von der Ferne zu steuern oder Daten zu übertragen. Ebenfalls ist der Server mit HTTPS verschlüsselt.

DIENSTE

Der Webserver Dienst heisst apache2.

PROTOKOLLIERUNG

Unter folgenden Verzeichnissen kann man die Logs anschauen.

var/log/apache	→ Protokollierung vom Apache Dienst
var/log/access.log	→ Log von den Zugriffen
dmesg	→ Log vom Betriebssystem, Debian
/var/log/sys.log	→ Log vom System

ZUGRIFFSBERECHTIGUNGEN

Auf die Apache relevanten Daten hat der Benutzer root und die Benutzergruppe Sudo Zugriff. Der System Admin arbeitet nicht mit dem Root Benutzer, sondern nur als Sudo, eine weitere Sicherheitsmassnahme.

ZERTIFIKATE

Als Zertifikat wurde hier das Standard SSL Zertifikat, SnakeOil verwendet. Da wir schon ein eigenes Zertifikat für den Mailserver gemacht haben, haben wir nun jediglich die Standard SSL Zertifikate verwendet. Um dies zu konfigurieren musste man in der Datei apache2.conf SSL auf YES einstellen und alle Dienste neu starten.

MAILSERVER

Als Mailserver haben wir uns für eine Zusammenarbeit zwischen Dovecot und Postfix entschieden.

DOVECOT

Dovecot ist ein Open Source IMAP und POP3 E-Mail Server. Er ist dafür verantwortlich, dass die E-Mails empfangen werden können, über die Protokolle POP3 oder IMAP.

POSTFIX

Posfix ist ein Postausgansserver der die E-Mails über das Protokoll SMTP verschickt.

INSTALLATION

Link zur Installation von **Postfix:** https://wiki.ubuntuusers.de/Postfix/
Link zur Installation von **Dovecot:** https://wiki.ubuntuusers.de/Dovecot_2/

Wir haben die Installation von beiden Servern nach den obenstehenden Anleitungen durchgeführt. Die Installation dauerte ca. 3.5h bis alles funktioniert hatte. Wenn es benötigt wird haben wir auch ein Video erstellt, während der ganzen Installation der Server.

KONFIGURATION

DOVECOT

Die Haupt Konfigurationsdatei von Dovecot ist in folgendem Verzeichnis abgelegt:
`/etc/dovecot/dovecot.conf`

Ebenfalls kann man sich die Ganze Konfiguration mit dem Befehl `doveconf -n` anzeigen lassen.
Die Konfigurationsdateien von **POP3** und **IMAP** befinden sich in folgendem Pfad:
/etc/dovecot/conf.d/20-pop3.conf und /etc/dovecot/conf.d/20-imap.conf.

1. Neuen User Hinzufügen. Dieser hat ein Home Verzeichnis unter /home/virtual/mailuser, damit wir eine Übersicht haben. Er kommt in die Gruppe Mail, kann sich nicht am Server selber anmelden (Debian Maschiene), hat kein Passwort und bekommt keinen Zugriff auf die Shell
   ```
   adduser kaelin --home /home/virtual/mailuser --ingroup mail --
   disabled-login --disabled-password --shell /bin/false$
   ```
2. Benutzer in die Liste unter /etc/dovecot/users eintragen
3. Mit dem Befehl `doveadm pw` Passwort für Benutzer setzen. Danach wird ein Hash Wert ausgegeben den man in die Liste von Punkt 2 eintragen muss.
   Ein solcher Eintrag sieht dann so aus;

kaelin@tbz.ch:{CRAM-MD5}ab44804364e3e6cd2476855a7f51c4da8fb155ed6064179273045ec03b5e6b86:1001:8:/bin/false:/home/virtual/kaelin

4. Danach wird ein sasl User erstellt. Dort wird zum ersten mal die Domain (tbz.ch) angegeben
   ```
   sasl user erstellen mit saslpasswd2 -c -u tbz.ch kaelin
   ```
5. Nun muss die Ordnerstruktur angepasst werden.
   /var/maildirs/tbz.ch/kaelin erstellen
   mkdir /var/maildirs/tbz.ch/
   mkdir /var/maildirs/tbz.ch/kaelin
   chgrp mail /var/maildirs/tbz.ch
   chown kaelin:mail /var/maildirs/tbz.ch/kaelin
   chmod -R 750 /var/maildirs/tbz.ch/
   ```
   (Root kann alles, Kälin kann alles in kälin, Gruppe Mail kann lesen
   im TBZ und kaelin und alle andern können nichts, keine Berechtigung)
   ```
6. Danach die Dienste neu starten und überprüfen
   ```
   service dovecot restart && service postfix restart && service
   saslauthd restart
   service dovecot status && service postfix status && service saslauthd
   status
   ```
7. Danach kann man den Benutzer im Mail Client einrichten. Das Konto muss man mit IMAP einrichten, als Stammordnerpfad INBOX eingeben, Port 143 mit TLS Verschlüsselung, Port 587 mit TLS Verschlüsselung und Authentifizierung einschalten.
8. Damit mehrere Benutzer mit einer Domäne existieren können müssen gewisse Einträge noch geändert werden in `/etc/dovecot/dovecot.conf`

mail_location = maildir:/var/maildirs/%d/%n
So können mehrere Benutzer einer einzigen Domain exisieren. Die Mailboxen bleiben dabei getrennt mi tdem wert %d (domain Name) %n (benutzer). Das sieht dann so aus ->
/var/maildirs/domain.ch/test: mail_location = maildir:/var/maildirs/%d/%n

9. Nun erstellen wir ein Self Signed Zertifikat
   ```
   openssl genrsa -des3 -out mail.srvm239.local.key 4096
   openssl req -new -key mail.srvm239.local.key -out
   mail.srvm239.local.csr
   ```

```
openssl rsa -in mail.srvm239.local.key -out
mail.srvm239.local.key.nopass
mv mail.srvm239.local.key.nopass mail.srvm239.local.key
```

10. Danach wird noch ein CA Key erstellt, damit unser Zertifikat auch authentifiziert wird
```
CA Key erstellen openssl req -new -x509 -extensions v3_ca -
keyout cakey.pem -out cacert.pem -days 3650
```

11. Postfix Submission starten. So ist danach der Prot 587 via TLS erreichbar. Config dazu: /etc/postfix/master.cf

```
submission inet n       -       -       -       -       smtpd
-o syslog_name=postfix/submission
-o smtpd_tls_security_level=encrypt
-o smtpd_sasl_auth_enable=yes
```

12. SSL Zertifikate installieren in Dovecot. config dazu `/etc/dovecot/conf.d/10-ssl.conf`

```
# SSL/TLS support: yes, no, required. <doc/wiki/SSL.txt>
ssl = yes

# PEM encoded X.509 SSL/TLS certificate and private key. They're opened before
# dropping root privileges, so keep the key file unreadable by anyone but
# root. Included doc/mkcert.sh can be used to easily generate self-signed
# certificate, just make sure to update the domains in dovecot-openssl.cnf
ssl_cert = </etc/ssl/certs/dovecot.pem
ssl_key = </etc/ssl/private/dovecot.pem

# PEM encoded trusted certificate authority. Set this only if you intend to use
# ssl_verify_client_cert=yes. The file should contain the CA certificate(s)
# followed by the matching CRL(s). (e.g. ssl_ca = </etc/ssl/certs/ca.pem)
ssl_ca = </etc/ssl/certs/cacert.pem

ssl v2 und v3 verbieten (nur tls nutzen)
# SSL protocols to use
ssl_protocols = !SSLv2 !SSLv3 # !=nicht
```

DIENSTE

Dovecot → Posteingang Dienst
Posftix → Post Ausgang
Saslauthd → Für Authentifizierung von SASL Username und Passwort an Postfix

PROTOKOLLIERUNG

/var/log/mail.log → Log des Mail Dienstes
/var/log/access.log → Log des Zugriffes

ZUGRIFFSBERECHTIGUNGEN

Auf Systemrelevante Daten hat nur der Benutzer root vollen Zugriff und Mitglieder der Gruppe sudo. Ansonsten haben Mail Benutzer wie oben beschreiben kein Systemzugriff. Sie haben nur auf Ihre eigene Mailbox Zugriff.

ZERTIFIKATE

Wie oben erklärt haben wir ein eigenes Zertifikat erstellt und dieses zur SSL Authentifizierung verwendet.

FTP SERVER

Als FTP Server haben wir uns für vsftp entschieden.

INSTALLATION

Mit dem Befehl `apt-get install vsftpd` kann der Server einfach installiert werden.

KONFIGURATION

`/etc/vsftpd.chroot_list` erstellt benötigt eine Liste um Verzeichnisse einzuschränken
`/etc/vsftpd.user_lis` erstellt, alle die erlaubt sind um auf FTP zuzugreifen
`/etc/vsftpd.conf` bearbeitet:
`write_enable=YES` --> Einschalten damit man schreibzugriff gewähren kann
`chroot_local_user=YES` --> User läuft in einem CHROOT, kann nicht ausbrechen
`local_root=/home/virtual/$USER` --> Wo seine Dateien gespeichert werden und er auch Zugriff erhält
`user_sub_token=$USER` --> braucht es damit die Variabel erkennt wird
`chroot_list_file=/etc/vsftpd.chroot_list` --> CHROOT liste ob man CH Rooten darf
`userlist_deny=NO` --> alle die nicht drinn sind nicht zugreiffen dürfen
`userlist_enable=YES` --> userliste aktivieren
`userlist_file=/etc/vsftpd.user_list` --> Userliste wo alle erlaubten benutzer drinn stehehn

Neuen Benutzer anlegen mit:
adduser zuend --home /home/virtual/zuend --ingroup mail --disabled-login --shell /bin/false
passwd zuend
mkdir /home/virtual/zuend/upload
chmod 550 /home/virtual/zuend
chmod 740 /home/virtual/zuend/upload
chown zuend:ftp /home/virtual/zuend/upload
nano /etc/vsftpd.user_list --> zuend in Liste eintragen
service vsftpd restart

DIENSTE

VSFTPD -> FTP Dienst

PROTOKOLLIERUNG

/var/log/vsftp → Log vom FTP Dienst

ZUGRIFFSBERECHTIGUNGEN

Wie oben geschrieben erhalten die erstellten Benutzer nur Zugriff auf Ihr erstelltes Verzeichnis und können nicht ausbrechen. Heruafgeladene Daten können nur gelesen werden und bearbeitet, nicht ausgeführt.

ZERTIFIKATE

Als SSL Zertifikat haben wir das Standard Zertifikat SnakeOil verwendet. Um dies einzurichten musste in der Konfigurationsdatei vsftpd.conf der Eintrag SSL auf YES gestellt werden.

5 TESTFÄLLE

Service	Was wird getestet?	Erwartetes Ergebnis	Ergebnis	OK / NOK	Massnahme
Apache	- Aufrufen der Webseite via SSL, https://192.168.1.201	Webseite wird aufgerufen über die sichere SSL Verbindung	Die Webseite wird via SSL aufgerufen und so eine sichere Verbindung aufgebaut	OK	Funktioniert
Apache	Aufrufen der Webseite ohne SSL, http://192.168.1.201	Webseite wird aufgerufen	Webseite wird aufgerufen und darauf hingewiesen das es keine Sichere Seite ist	OK	Nichts unternehmen, Fehlermeldung ist gewollt
Apache	Eine neue HTML Seite kann hochgeladen werden	HTML Seite wird richtig hochgeladen und dargestellt	Die neue HTML Seite wird richtig angezeigt und aufgerufen	OK	Funktioniert
Apache	Versuchen über URL auf Struktur des Servers zu geraten (https://192.168.1.201/var)	Es wird eine Fehlermeldung ausgeben, dass die Seite nicht gefunden wird oder dann man keine Berechtigung darauf hat	Fehlermeldung wird angezeigt, es findet die Seite nicht	OK	Funktioniert
VSFTP	Man kann einen neuen Benutzer anlegen	Benutzer kann angelegt werden und er kann sich via SSL mit einem Client verbinden	Neuer Benutzer konnte angelegt werden, er konnte sich via File Zilla eine SSL Verschlüsselte Verbindung zum FTP Server aufbauen	OK	Funktioniert
VSFTP	Es können Daten auf den FTP Server hochgeladen werden	Daten können in seiner Order hochgeladen werden	Daten können hochgeladen werden und Ordner erstellt	OK	Funktioniert
VSFTP	Daten löschen / bearbeiten	Daten können wieder gelöscht oder bearbeitet werden auf dem FTP Server	Die Daten können wieder gelöscht werden und man kann sie auch bearbeiten.	OK	Funktioniert
VSFTP	Benutzer kommt nicht in andere Verzeichnisse	Benutzer kann nur in seinem Ordner bleiben.	Benutzer kann nur in seinem Order bleiben und kommt in keine anderen Verzeichnisse.	OK	Funktioniert

Mail	Neuer Benutzer kann angelegt werden	Benutzer kann angelegt werden	Der neue Benutzer konnte angelegt werden	OK	Funktioniert
Mail	Benutzer kann sein Konto auf einem Mail Client einrichten	Benutzer kann sein Konto einrichten	Konto konnte via IMAP, SSL, Authentifizierung, Port TLS 443 und TLS 587, Stammorderpfad INBOX eingerichtet werden	OK	Funktioniert
Mail	Benutzer kann sein Konto auf Mail Client als POP3 einrichten	Benutzer kann kein POP3 Konto einrichten	Server verweigert die Anmeldung	OK	Funktioniert
Mail	Benutzer kann sich ohne TLS Authentifizieren	Es wird eine Fehlermeldung geben	Benutzer kann sich nicht ohne TLS anmelden, Fehlermeldung wird ausgegeben	OK	Sicherheitstechnisch, Funktioniert
Mail	Passwörter können aus den Config Files heraus gelesen werden	Passwort kann nicht heraus gelesen werden	Passwörter können nicht heraus gelesen werden, lediglich der Hash wert	OK	Funktioniert
Mail	Mails können versendet werden	Mails werden verschickt übers Internet	Die E-Mails werden korrekt verschickt. Wenn man die Mails ohne Domain nur mit meiner Lokalen IP Adresse verschickt, kommen die E-Mails je nach Provider nicht beim Empfänger an	OK	Funktioniert
Mail	Mails können untereinander versendet werden	Mails können versendet werden	Mails werden korrekt versendet	OK	Funktioniert
Mail	Mails werden übers Internet empfangen	Mails können nicht empfangen werden	Die Mails werden nicht empfangen, da ich nur von einer Pseudo Domain aus geschrieben habe. Wenn man darauf antwortet geht die E-Mail an den Mailserver dieser Domain	OK	Ist korrekt
Debian	Protokolle können abgerufen werden	Protokolle können immer abgerufen werden	Die Protokolle waren immer zugänglich	Ok	Funktioniert
Debian	Mail Benutzer können sich an OS anmelden	Erstellte Mail Benutzer können sich nicht direkt am Server anmelden	Die erstellten Mailbenutzer können sich nicht direkt am Betriebssystem anmelden	OK	Funktioniert

Debian	Debian lässt sich updaten	Updates könne im laufendem Betrieb gemacht werden	Updates können jederzeit durchgeführt werden	OK	Funktioniert

LAST TEST

Um einen Last Test auszuführen, begebe ich mich an einen zweiten Rechner mit Linux oder Bash darauf installiert. Um den Apache Webserver zu testen, gibt es ein Tool. Das Tool heisst ab und ist ein Apache Server Benchmarking Tool. Standardmässig ist dies gleich auch installiert.

Nun wird der folgende Befehl ausgeführt:

```
sudo ab -c 10000 -t 100  http://192.168.1.201/
```

Der Befehl wird mit `sudo` Rechten ausgeführt, ab ist das Tool das gestartet wird, `-c 1000` bedeutet Nummer der Anfragen die der Rechner machen soll, `-t 100` bedeutet er soll 100 Sekunden zeit haben um diese 10000 Request durchzuführen und am Schluss wird noch das Ziel angegeben. Um Den Server maximal zu belasten habe ich diesen Befehl gleichzeitig auf den Server abgesetzt.

Um die Belastung des Servers mit an zu sehen führt man den Befehl top aus. Dort ist ersichtlich wieviel Leistung der Server bezieht und von welchen Diensten.

```
top - 14:12:20 up 15:24,  2 users,  load average: 0.00, 0.00, 0.00
Tasks:  91 total,   1 running,  90 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.3 us,  0.0 sy,  0.0 ni, 99.3 id,  0.3 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:   2044836 total,   316620 used,  1728216 free,    29976 buffers
KiB Swap:  1113084 total,        0 used,  1113084 free.   168072 cached Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
   74 root      20   0       0      0      0 S   0.3  0.0   0:26.68 kworker/0:2
    1 root      20   0   28624   4840   3176 S   0.0  0.2   0:01.29 systemd
    2 root      20   0       0      0      0 S   0.0  0.0   0:00.00 kthreadd
    3 root      20   0       0      0      0 S   0.0  0.0   0:10.47 ksoftirqd/0
    5 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/0:+
    7 root      20   0       0      0      0 S   0.0  0.0   0:02.21 rcu_sched
```

Wichtig hierbei ist die load average. Diese Zahl sollte nicht höher als 1 gehen, da dies die Anzahl der CPU Cores ist und nicht überschritten werden sollte. Danach ist die Leistung des Servers überschritten. Mit obigem Test ist dies der Fall gewesen. Die load average ist über 1.4 gesprungen. Danach sind nur noch Fehlermeldungen gekommen, many open files. So einen Test wollten wir genau verursachen. So steigert sich die load average immer weiter hoch, da der Server nun waits machen muss, eine Art Warteliste, die regelt was er nun der Rheie nach abarbeiten muss. Der Server blieb aber standhaft und ist nicht abgestürzt.
Mit dem Befehl sudo ab -c 1000 -t 100 http://192.168.1.201/ Schaffte es der Server gerade noch knapp.

Mit dem Floodping Befehl `ping -f 192.168.1.201` schaffte er ca. 200 requests pro Sekunde. Somit war der Server in die Wand gefahren. Der Crashtest ist bestanden.

THEORIE

Tests eines Systems dienen ganz einfach dazu es auf verschiedene Dinge zu testen, bevor man das System wirklich verwendet. So kann man Fehler und Grenzen schon im Vorhinein finden und bei Bedarf verbessern. Jeglicher Test muss natürlich genau festgehalten werden und er muss von einer anderen Person replizierbar sein, ansonsten ist das Ergebnis des Tests wertlos.

MÖGLICHE TESTGEBIETE SIND:

Technische Tests
Dabei wird vor allem die Erreichbarkeit des Servers getestet sowie das Netz und die Hardware. Mögliche Testgebiete sind: Das Routing (Ist der Server problemlos erreichbar). Die Umgebung (Stromversorgung, Belüftung/Temperatur, Zugang zum Server) Hardware (Funktioniert die Hardware, auch unter längerer Belastung).

Applikationstests
Dabei werden die installierten Applikationen auf ihre fehlerfreie Funktion getestet. Bei Webservern zb. werden die Richtigen Seiten angezeigt? Werden Zugriffsberechtigungen berücksichtigt? Funktioniert die Verschlüsselung? Bei Mailservern: Können Mails gesendet und empfangen werden? Werden Mails akzeptiert? Bei FTP Servern: Ist der Server erreichbar? Wie viele Sitzungen schafft der Server gleichzeitig? Beim DNS Server: Werden Namen (und evtl. IPs) richtig aufgelöst?

Sicherheitstests
Sicherheitstest umfassen einen riesigen Bereich, Im Prinzip ist jede Einstellung die die Sicherheit betrifft für einen Test relevant. Insbesondere muss darauf geachtet werden, dass folgendes gewährleistet ist:

Verfügbarkeit (Wie verhält sich der Server unter Last? Ist er immer verfügbar?) Am einfachsten kann man dies mit einem automatischen Zweitsystem laufend testen.

Vertraulichkeit (Sind die Daten geschützt? Funktioniert die Verschlüsselung? Können nur berechtigte Nutzer auf Daten zugreifen? Welche Daten sind öffentlich sichtbar?)

Integrität (Ist das Zertifikat des Servers aktuell und gültig? Sind alle DNS Einträge korrekt?)

Im Internet gibt es eine Menge Tools, die bestimmte Einstellungen und gefährliche Lücken auf den Servern suchen und melden. Natürlich kann man auch manuell alles durchgehen.

Lasttests
Das Verhalten von Servern ist unter Last nicht einfach zu testen, da das Benutzerverhalten nicht vorherzusagen ist. Es gibt dennoch viele Tools um Benutzeranfragen zu simulieren. Eine andere Möglichkeit ist ein Betatest, bei dem echtes Userverhalten analysiert werden kann. Grundsätzlich muss einfach die Auslastung des Servers unter bestimmten Situationen gemessen werden.

Crashtests
Ein Crashtest ist eigentlich nichts weiter als ein Lasttest, bei dem man das System bis über das Limit drängt, um zu sehen, was der Server kann, oder auch eben nicht. Eine Möglichkeit ist es, mit Absicht eine DDOS Attacke durchzuführen und mittels Überwachung festzustellen wann der Server einknickt. Dies kann dann für die weitere Planung verwendet werden.

6 ZERTIFIKAT

THEORIE

Ein Zertifikat ist grundlegend eine Bestätigung einer Sache. In der Informatik geht es um digitale Zertifikate. Diese werden für die Sicherheit von Daten verwendet. Ein digitales Zertifikat bestätigt die Authentizität und Integrität von Daten. Durch kryptische Verfahren kann diese nachgewiesen werden. Die Ausstellung eines Zertifikates erfolgt meist über eine offizielle Zertifizierungsstelle. Die häufigsten verwendeten Zertifikate sind Public-Key-Zertifikate (Standard X.509) mit denen die Echtheit eines Schlüssels belegt wird.

Bei Webservern werden Zertifikate in Verbindung mit Verschlüsselung von Übertragungen, d. h. SSL bzw. TLS verwendet. Bei einer Anfrage auf einen Webserver mittels HTTP wird der Server normalerweise einfach die gewünschten Daten übermitteln. Bei HTTPS gibt es einen Zwischenschritt, der Server sendet sein Zertifikat, dass er von einer offiziellen Stelle bekommen hat. Der Client kann nun das Zertifikat bei eben dieser Stelle validieren und so die Echtheit des Servers feststellen. Nach erfolgreicher Identifizierung beginnt die verschlüsselte Kommunikation zwischen Server und Client.

SSL/TLS arbeitet mit einem Public Key und einem Private Key zur Sicherheit. Der öffentliche Schlüssel des Empfängers ist dem Sender bekannt, er nutzt diesen um die Daten zu verschlüsseln. Zur Entschlüsselung wird nun aber der Private Schlüssel benötigt. Dieser darf nur dem Empfänger bekannt sein.

Bevor der Sender jedoch etwas verschlüsselt übermittelt, muss er zweifelsfrei feststellen, ob der öffentliche Schlüssel wirklich dem Empfänger gehört. SSL/TLS bietet quasi keinen Schutz, wenn nicht sicher ist, dass der Schlüssel wirklich vom richtigen Server kommt. Hier kommt das Zertifikat ins Spiel, es bestätigt quasi die Echtheit des Schlüssels.

In dem Zertifikat steht unter anderem:
- Der öffentliche Schlüssel
- Der Domainname
- Ein Ablaufdatum
- Die Instanz, die das Zertifikat ausgestellt hat

Es gibt 3 Arten von Zertifikaten, diese unterscheiden sich hinsichtlich des Prüfaufwandes und entsprechend der bestätigten Echtheitsstufe:

- Domain-Validated-Zertifikat (DV-SSL)
- Organisation-Validation-Zertifikat (OV-SSL)
- Extended-Validation-Zertifikat (EV-SSL)

Während man die ersten beiden bereits sehr günstig bekommt (evtl. sogar gratis) kostet das letzte deutlich mehr, weil es mit einem deutlich grösseren Prüfungsaufwand verbunden ist, daher kann man dabei aber auch von der grössten Vertrauenswürdigkeit ausgehen.

Es gibt weltweit über 700 Zertifizierungsstellen, diese Prüfen die Webseiten und stellen Zertifikate aus, ausserdem validieren sie diese auf Anfrage eines Servers/Clients.

UMSETZUNG

MAILSERVER

Hierfür haben wir ein eigenes Zertifikat generiert, wie oben beim Mailserver erklärt wird. Dies ist ein Self Signed Zertifikat. Damit unser Zertifikat auch authentifiziert wird, haben wir noch einen CA Key erstellt. Nun kommt am Anfang bei der Einrichtung der Mailkonten lediglich eine Meldung, dass das Zertifikat selber erstellt wurde. Probleme bereitet dies aber nicht.

FTP SERVER

Als TLS Zertifikat haben wir das Standard Zertifikat SnakeOil verwendet. Um dies einzurichten musste in der Konfigurationsdatei vsftpd.conf der Eintrag SSL auf YES gestellt werden.

WEBSERVER

Als Zertifikat wurde hier das Standard TLS Zertifikat, SnakeOil verwendet. Da wir schon ein eigenes Zertifikat für den Mailserver gemacht haben, haben wir nun lediglich die Standard TLS Zertifikate verwendet. Um dies zu konfigurieren musste man in der Datei apache2.conf SSL auf YES einstellen und alle Dienste neu starten, sowie den Pfad zum Zertifikat angeben.

PROTOKOLLIERUNG

Um ein noch ausführlicheres Log File zu kreieren haben wir das Forensic Modul installiert (mod_log_forensic).

Zuerst wurde das Forensic modul aktiviert und das zugehörige mod_unique_id.so aktiviert, mittels a2enmod log_forensic und unique_id

Danach musste man in der Apache.conf den neuen Log Pfad hinzufügen.
ForenicLog /var/log/apache2/forensic_log. In diese Datei schreibt es nun die Log Daten.
In dieser Log Datei sehen wir den Request mit einer Unique ID beim Start, den Browser Agent mit all seinen Einstellungen und mit der zweiten gleichen ID beim Abschluss vom Request, so sieht man ob ein Request hängen bleibt und wie schnell ein Request anfängt und fertig ist.

Dies ist eine Möglichkeit um den Header zu protokollieren.

So könnte ein Mögliches Log File aussehen.

```
+WURAbX8AAQEAADR@UMEAAABG|GET / HTTP/1.1|Host:192.168.1.201|Connection:keep-alive|Upgrade-Insecure-Requests:1|User-Agent:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like G$
-WURAbX8AAQEAADR@UMEAAABG

+WURAmX8AAQEAADR9Rw8AAAAD|GET / HTTP/1.1|Host:192.168.1.201|Connection:keep-alive|Accept-Language:de-CH,de,en-US,en|Upgrade-Insecure-Requests:1|User-Agent:Mozilla/5.0 (Linux; Android 7.0; SAMSUNG $
-WURAmX8AAQEAADR9Rw8AAAAD

+WURAsH8AAQEAADR9RxIAAAAG|GET / HTTP/1.1|Host:192.168.1.201|Connection:keep-alive|Upgrade-Insecure-Requests:1|User-Agent:Mozilla/5.0 (Linux; Android 6.0.1; Redmi 4 Build/MMB29M; wv) AppleWebKit/53$
-WURAsH8AAQEAADR9RxIAAAAG
```