Abschlussprüfung Sommer 2019

Fachinformatiker für Systemintegration

Dokumentation zur betrieblichen Projektarbeit

Webserverkonfiguration

Konzeption & Konfiguration von Webservern für die browserbasierte Erreichbarkeit verschiedener Anwendungen

Prüfungsausschuss: FISI_09
Abgabedatum: Berlin, den 04.06.2019

Prüfungsbewerber

Sven Grothe
Höhndorfstraße 9
12101 Berlin

Ausbildungsbetrieb

taskit GmbH
Groß-Berliner Damm 53
12101 Berlin

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Abkürzungsverzeichnis

AES	Advanced Encryption Standard
CBC	Cipher Block Chaining
CLI	Command Line Interface
DH	Diffie-Hellman-Schlüsselaustausch
DMZ	Demilitarized Zone
DNS	Domain Name System
ECDH	Elliptic curve Diffie-Hellman
GCM	Galois/Counter Mode
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IP	Internet Protocol
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
MUA	Mail User Agent
NAT	Network Address Translation
OS	Operating System
POODLE	Padding Oracle On Downgraded Legacy Encryption
PHP	PHP: Hypertext Preprocessor (rekursives Akronym)
RFC	Request for Comments
RSA	Rivest, Shamir, Adleman (asymmetrisches kryptographisches Verfahren benannt nach den Erfindern)
SHA	Secure Hash Algorithm
SSH	Secure Shell
SSL	Secure Sockets Layer
TLS	Transport Layer Security
WAN	Wide Area Network
WLAN	Wireless Local Area Network
XSS	Cross-Site Scripting

1. Einleitung

Diese Dokumentation wurde im Rahmen der betrieblichen Projektarbeit zur Berufsausbildung zum Fachinformatiker für Systemintegration erstellt. Das Projekt beinhaltet die Konzeption und Konfiguration von Webservern für die browserbasierte Erreichbarkeit verschiedener Anwendungen unter Berücksichtigung technischer Maßnahmen zur Datensicherheit. Ziel des Projektes ist es, die Lerninhalte der Ausbildung zu vertiefen und ein realitätsnahes Projekt umzusetzen. Dabei sollen sowohl fachliche, als auch soziale Kompetenzen gefördert werden.

2. Projektbeschreibung

2.1. Projektumfeld & -motivation

Die Firma *taskit GmbH* ist ein mittelständischer Dienstleister im IT-Bereich, die auf den Vertrieb von Hard- und Softwarelösungen eingebetteter Systeme spezialisiert ist. [1]

Am Firmenstandort Groß-Berliner Damm in Berlin arbeiten rund 20 Mitarbeiter an der Produktion von eingebetteten Hardwaresystemen, an der softwaretechnischen Realisierung von kundenspezifischen Anforderungen der eingebetteten Produkte, sowie an Drittmittel-Förderprojekten zur Entwicklung von neuen Systemen im praxisnahen Umfeld.

Für die Optimierung der Arbeitsabläufe innerhalb der Firma soll die interne Infrastruktur ausgebaut werden. Dafür soll eine bestehende Server-Architektur so erweitert werden, dass mehrere Anwendungen im internen Firmenumfeld durch einen geeigneten Webbrowser erreichbar werden, sowie ein Remotezugriff von außerhalb des Firmennetzes über den Webbrowser auf diese Anwendungen realisiert werden kann.

2.2. Projektziele

Sachziel: Für drei verschiedene Anwendungen sollen Webserverlösungen zur browserbasierten Erreichbarkeit konzeptioniert und konfiguriert werden:

- Ein internes Kommunikationsprogramm soll über einen geeigneten Webbrowser sowohl im LAN, als auch im WAN erreichbar sein
- Ein internes Wiki soll über einen geeigneten Webbrowser im LAN, jedoch nicht im WAN erreichbar sein
- Ein webbasierter MUA soll über einen geeigneten Webbrowser sowohl im LAN, als auch im WAN erreichbar sein

Die über das WAN erreichbaren Anwendungen sollen unter Berücksichtigung technischer Maßnahmen zur Datensicherheit konfiguriert werden. Die Lösung muss gängigen Sicherheitsstandards entsprechen und die aktuellen Sicherheitsprotokolle unterstützen.

Kostenziel: Für die Realisierung der Projektziele soll die bestehende Serverarchitektur erweitert werden. Es soll eine kostenneutrale Lösung entworfen werden, die abgesehen von

Kosten für aufgewendete Arbeitszeit und Sachmittel, sowie den gegenwärtigen Fixkosten des Serverbetriebs keine weiteren Mehrkosten verursacht.

Zeitziel: Für das Projekt sind unter Nichtberücksichtigung der Prozessschnittstellen 35 Arbeitsstunden anberaumt, eingenommen sieben Stunden für die Dokumentation.

Qualitätsziel: Die vollständige und fehlerfreie Umsetzung der Sachziele sollen durch Testfälle abgedeckt werden. Dabei sind sowohl die vollständige Funktionalität, als auch die sicherheitstechnischen Maßnahmen zu testen.

2.3. Ist-Zustand

Das Unternehmensnetz besitzt für die interne Infrastruktur mehrere Server:

- Server *clara* nutzt als Betriebssystem *Debian 9 (Stretch)* und beinhaltet dabei *rocket-chat* als Kommunikationsprogramm, welches über einen eigenen Client genutzt werden kann, sowie *xwiki* als Wiki, welches bereits einen vorkonfigurierten Apache Tomcat-Webserver auf Port 8080 nutzt.
- Server *mail* ist ein autarker Mail Server mit einem MUA auf Basis der Skriptsprache PHP und nutzt als Betriebssystem ebenfalls *Debian 9 (Stretch)*.

Alle genannten Anwendungen verfügen über eine Nutzerauthentifizierung am zentralen LDAP-Server. Alle Server stehen im selben LAN-Subnetz hinter einem Router der Firma Mikrotik, dessen öffentliche IP-Adresse statisch ist. Im Besitz der Firma sind ferner mehrere Domains, darunter unter anderem *taskit.de* und *ledato.de*. Ein Netzplan des Unternehmensnetzes kann den Anlagen unter *A.2 Netzplan* entnommen werden.

2.4. Soll-Konzept

Ziel ist es funktional, dass alle genannten Anwendungen über geeignete Webbrowser erreichbar sind. Zur Realisierung der funktionalen Ziele sollen beide Server um eine Webserver-Komponente erweitert werden. Dies kann durch die Umsetzung folgender Schritte realisiert werden:

- Die Anwendungen sollen über Subdomains von *taskit.de* und *ledato.de* aufrufbar sein. Dafür ist es notwendig für den Zugriff aus dem WAN die DNS-Namensauflösung der Domains beim Provider der Domains (setzen von A-, CNAME- und TXT-Records), sowie beim Zugriff aus dem LAN die DNS-Namensauflösung der Domains im Standardgateway des Unternehmensnetz einzurichten.
- Am Router sind Firewall- und NAT-Regeln so einzurichten, dass die Subdomains korrekt auf die jeweiligen Server auflösen.
- Es ist ein geeigneter Webserver für die Server zu evaluieren (beispielsweise durch eine Nutzwertanalyse).
- Der gewählte Webserver soll installiert und für alle drei Anwendungen konfiguriert werden.

Für die Umsetzung der technischen Maßnahmen zur Datensicherheit sind für die öffentlich zugänglichen Anwendungen folgende Sicherheitsaspekte zu beachten:

- Verbindungen aus dem WAN sollen ausschließlich verschlüsselt (über HTTPS) erfolgen. Dazu ist es notwendig für die genutzten Subdomains SSL-Zertifikate aus vertrauenswürdigen Quellen auszustellen und die Aktualisierung der Zertifikate vor ihrem Ablaufdatum sicherzustellen.
- Es ist sicherzustellen, dass ausschließlich Verschlüsselungsprotokolle in ihrer aktuellen Version verwendet und unterstützt werden (SSL, TLS). Der Bedarf nach veralteten Versionen dieser Protokolle ist zunächst zu evaluieren.
- Die Cipher Suites (Chiffrensammlung) sind so zu konfigurieren, dass für den Schlüsselaustausch, die Authentifizierung, die Verschlüsselung und die Hashfunktion aktuelle Algorithmen für den Aufbau der gesicherten Datenverbindung verwendet werden.

Für die Qualitätssicherung sind alle Konfigurationen hinsichtlich ihrer Funktionalität und der Sicherheitsaspekte zu testen und die Ergebnisse zu dokumentieren (siehe auch *3.5 Planung QS-Maßnahmen*).

2.5. Abweichungen gegenüber dem Projektantrag

Inhaltlich gibt es keine Abweichungen vom eingereichten Projektantrag. Lediglich die Zeitplanung des Projektes, sowie der Titel des Projektes wurden geringfügig angepasst um das Projekt besser abzubilden.

2.6. Projektabgrenzung

Die Konfiguration der genannten Anwendungen wurde in separaten Projekten realisiert und ist nicht Teil dieses Projektes. Ebenso das Realisieren einer etwaigen DMZ, da die Server nach Abschluss des Projektes auf gewissen Ports öffentlich zugänglich sind ist kein Teil dieses Projektes, sondern als ein mögliches Folgeprojekt zu betrachten. Weiterhin ist die Nutzerauthentifikation der Anwendungen am zentralen LDAP-Server von diesem Projekt abzugrenzen.

2.7. Prozessschnittstellen

Personen

Tabelle 1: Auflistung der Prozessschnittstellen

Name	Firma	Abteilung	Beschreibung
[redacted]	taskit GmbH	Geschäftsführung	Ausbilder / Auftraggeber
[redacted]	taskit GmbH	Softwareentwicklung	Projektansprechparter

Nähere Beschreibung

[redacted] war maßgeblich Ansprechpartner bei der Konzeption und Umsetzung des Projektes. Er unterstützte bei der Erarbeitung der Projektziele und kann durch sein tiefgreifendes Fachwissen Anregungen bei der Entscheidungsfindung, sowie Hinweise auf mögliche Lösungswege bei der Projektumsetzung geben.

[redacted] war Ansprechpartner bei Fragen zur technischen Umsetzung des Projektes,

sowie direkter Ansprechpartner für das Projekt. Er unterstützte zudem bei der Umsetzung, sofern notwendige Berechtigungen des Auszubildenden fehlen (Zugang zu den Firmendomains beim Provider zum Setzen der DNS-Einträge)

2.8. Projektablaufplan

Ein Projektablaufplan kann der Anlage *A.1 Projektablaufplan* entnommen werden.

3. Projektplanung

3.1. Personalplanung

Tabelle 2: Personalplanung

Name	Tätigkeit	Zeitaufwand
Sven Grothe	Projektumsetzung	35 Stunden
[redacted]	Projektdefinition, Projektübergabe, Projektunterstützung	4 Stunden
[redacted]	Ansprechpartner, Projektübergabe, Projektumsetzung bei fehlenden Berechtigungen	3 Stunden

3.2. Sachmittelplanung

Zur Realisierung des Projektes benötigte Mittel:

Tabelle 3: Sachmittelplanung – Hardware

Hardware	Beschreibung
zwei Server	genutzte Server zur Projektumsetzung (Installation und Konfiguration der Webserver)
MikroTik-Router	Gateway zwischen WAN und LAN des Unternehmens, sowie für die Firewall-, NAT- und DNS-Konfiguration
Arbeitsplatz-PC	System zur Projektumsetzung

Tabelle 4: Sachmittelplanung – Software

Software	Beschreibung
Debian 9 (Stretch)	wird als Betriebssystem zum Betreiben der beiden Server verwendet
nginx	ausgewählter Webserver (Die Entscheidungsfindung dazu ist unter *4.1 Entscheidungsfindung* erläutert)
MikroTik RouterOS 6.44	Betriebssystem des Routers
Windows 10 Pro	Betriebssystem für den Arbeitsplatz-PC
PuTTY 0.71	Software für die Remote-Konfiguration der Server via SSH
Google Chrome 74	gewählter Webbrowser für die Ausführung funktionaler Tests

Zusätzlich zu den genannten Sachmitteln standen für die Projektumsetzung die Räumlichkeiten des Unternehmens zur Verfügung.

3.3. Kostenplanung

Für die Kostenplanung wurden die Hardware-Sachmittel nicht beachtet, da es sich hierbei um bestehende Systeme handelt, die erweitert werden und nicht um Neuanschaffungen dieser im Rahmen des Projektes. Bei den Software-Sachmitteln handelt es sich, abgesehen von Windows 10 Pro auf dem Arbeitsplatz-PC, welches durch die weiter unten genannte Pauschale abgedeckt ist, um frei verfügbare OpenSource-Software.
Für den Auszubildenden im dritten Lehrjahr wurde ein Bruttogehalt von 12.000 € pro Jahr bei 150 Arbeitstagen angenommen (Abzug von je zwei Tagen Berufsschule für 35 Wochen).
Für den Mitarbeiter in der Softwareentwicklung wurde ein Bruttogehalt von 40.000 € pro Jahr, für die Geschäftsleitung ein Bruttogehalt von 100.000 € pro Jahr bei jeweils 220 Arbeitstagen angenommen.
Für die Verwendung der Räumlichkeiten und der Ressourcen zur Ausführung des Projektes (u.a. Arbeitsplatz-PCs, Büromaterialien) wurden pauschal 15 € pro Stunde für jeden Mitarbeiter angesetzt. Damit ergeben sich folgende Stundensätze:

$$\frac{12.000\,€/\text{Jahr}}{(150 \cdot 8)\,\text{h/Jahr}} + 15\,€/\text{h} = 25{,}00\,€/\text{h} \tag{1}$$

$$\frac{40.000\,€/\text{Jahr}}{(220 \cdot 8)\,\text{h/Jahr}} + 15\,€/\text{h} = 37{,}73\,€/\text{h} \tag{2}$$

$$\frac{100.000\,€/\text{Jahr}}{(220 \cdot 8)\,\text{h/Jahr}} + 15\,€/\text{h} = 71{,}82\,€/\text{h} \tag{3}$$

Somit ergeben sich folgende Gesamtkosten für das Projekt:

Tabelle 5: Kostenplanung

Beschreibung	**Stunden**	**Stundensatz**	**Gesamtkosten**
Personalkosten Sven Grothe	35 h	25,00 €/h	875,00 €
Personalkosten [redacted]	4 h	71,82 €/h	287,28 €
Personalkosten [redacted]	3 h	37,73 €/h	113,19 €
		Summe:	**1.275,47 €**

3.4. Wirtschaftlichkeitsanalyse

Das Projekt dient in erster Linie der Optimierung von Arbeitsabläufen im Unternehmen. Durch die browserbasierte Erreichbarkeit der Anwendungen entfallen somit separate Clients (im Fall des Kommunikationsprogramms, sowie des MUA), sowie ein umständlicher Aufruf des Wikis über die Angabe der IP-Adresse und des Ports. Ferner können auf das Kommunikationsprogramm, sowie der MUA von außerhalb des Firmennetzes zugegriffen werden, welches die Arbeitsabläufe im Außendienst optimiert.

3.4.1. Amortisationsdauer

Angenommen der Anwender spart durch diese Optimierung 5 min Arbeitszeit pro Tag für jeden der 20 Mitarbeiter. Dann ergibt sich eine Zeitersparung von:

$$20 \cdot 220\,\text{Tage/Jahr} \cdot 5\,\text{min/Tag} = 22.000\,\text{min/Jahr} = 366,67\,\text{h/Jahr} \tag{4}$$

Bei einem durchschnittlichen Bruttoeinkommen von 30.000 € im Jahr bei 220 Arbeitstagen folgt damit eine jährliche Kosteneinsparung von:

$$\frac{30.000\,€/\text{Jahr}}{(220 \cdot 8)\,\text{h/Jahr}} \cdot 366,67\,\text{h/Jahr} = 6.250{,}06\,€/\text{Jahr} \tag{5}$$

Die Amortisationszeit beträgt demnach $\frac{1.275{,}47\,€}{6.250{,}06\,€/\text{Jahr}} = 0,204\,\text{Jahre} \approx 74\,\text{Tage}$.

3.5. Planung QS-Maßnahmen

Für die Sicherstellung der Funktionalität des Endproduktes wurden folgende Maßnahmen geplant:

- Testen der Webserver-Konfiguration auf syntaktische Fehler
- Feststellung, dass der Webserver den Systemprozess fehlerfrei startet.
- manuelle Konnektivitäts-Tests der Weboberfläche des Kommunikationsprogramms, des Wikis und des MUA über geeignete Webbrowser aus dem LAN.
- Feststellung der Traceroute zum Webserver aus dem LAN, insbesondere Sicherstellung, dass die Pakete bei Nutzung aus dem LAN nicht durch das WAN geleitet werden.
- manuelles Testen der Weboberfläche des Kommunikationsprogramms und des MUA über geeignete Webbrowser aus dem WAN.
- Überprüfung, dass das Wiki nicht aus dem WAN erreichbar ist.

Für die Sicherstellung der korrekten Umsetzung technischer Maßnahmen zur Datensicherheit werden folgende Maßnahmen geplant:

- Überprüfung des Sicherheitszertifikats auf den öffentlich zugänglichen Subdomains
- Überprüfung der Aktualisierung des Sicherheitszertifikats vor seinem Ablaufdatum
- Überprüfung der unterstützten Sicherheitsprotokolle, Cipher Suits und Protokolle für den Schlüsselaustausch
- TLS-Handshake-Simulation für unterschiedliche Kombinationen von Betriebssystemen und Browser
- Simulation verschiedener bekannter Sicherheitslücken und Attacken gegen den Webserver

3.6. Zeitplanung

Die folgende Tabelle zeigt die Soll-Zeitplanung des Projektes:

Tabelle 6: detaillierte Soll-Zeitplanung

1	**Definitionsphase**	**2 h**
1.1	Definition der Projektziele	2 h
2	**Planungsphase**	**6 h**
2.1	Projektplanung und -konzeption	3 h
2.2	Evaluation verschiedener Webserver für die aktuelle Problemstellung	2 h
2.3	Planung QS-Maßnahmen	1 h
3	**Durchführungsphase**	**12 h**
3.1	Setzen der DNS-Records	0,5 h
3.2	Routerkonfiguration	1 h
3.3	Installation der Webserver	0,5 h
3.4	Konfiguration der Webserver bzgl. Funktionalität	5 h
3.5	Konfiguration der Webserver bzgl. der Maßnahmen zur Datensicherheit	4 h
3.6	Einbinden der Sicherheitszertifikate mit automatisiertem Update	1 h
4	**Qualitätssicherung**	**6 h**
4.1	Überprüfung der Konfiguration auf syntaktische Fehler	1 h
4.2	Testen der funktionalen Konfiguration	2 h
4.3	Testen der Maßnahmen zur Datensicherheit	1 h
4.4	Fehlerbehebung und Optimierung	2 h
5	**Projektabschluss und -übergabe**	**2 h**
6	**Projektdokumentation**	**7 h**
	Summe:	**35 h**

4. Projektdurchführung

4.1. Entscheidungsfindung

Zunächst galt es einen geeigneten Webserver für Debian 9 (Stretch) zu evaluieren. Die beiden am weitesten verbreiteten Webserver sind mit Abstand *Apache* und *nginx*, andere Webserver sind nur schwer auszumachen, so dass sich die Entscheidungsfindung auf diese beiden Softwareprodukte beschränkt. [2] Untersucht wurden die Produkte in den Aspekten Performance, Sicherheit, Flexibilität, Dokumentation und Support:

Tabelle 7: Entscheidungsmatrix bezüglich der Wahl der Webserver-Software

		Apache		**nginx**	
Kriterium	**Gewichtung**	Bewertung	Ergebnis	Bewertung	Ergebnis
Performance	30%	2	0,6	5	1,5
Sicherheit	30%	4	1,2	5	1,5
Flexibilität	10%	5	0,5	2	0,2
Dokumentation	20%	5	1,0	4	0,8
Support	10%	5	0,5	4	0,4
Summe	100%	—	4,0	—	4,4

kurze Erläuterung zur Bewertung der einzelnen Kriterien: [3]

- nginx ist für statische Inhalte deutlich performanter als Apache, da bei Apache für jeden Nutzer ein eigener Prozess auf dem Server geöffnet wird.
- Beide Projekte haben ein exzellentes Sicherheitstracking für ihre Codebase. Die nginx-Codebase ist jedoch deutlich kleiner als die von Apache, was im Sinne der Sicherheit positiv zu bewerten ist.
- Die Webserversoftware kann durch Module angepasst werden. Während Apache dynamisches Einbinden seit langer Zeit unterstützt ist dies bei nginx bei vielen Modulen nicht der Fall.
- Beide Produkte verfügen über eine exzellente Internet-Dokumentation. Einziges Manko ist, dass die nginx-Dokumentation nicht in deutscher Sprache vorliegt.
- Beide Produkte haben eine große Community und viele Hilfeforen. Apache ist jedoch etwas weiter verbreitet und länger am Markt.

4.2. Beschreibende Arbeitsschritte

4.2.1. Setzen der DNS-Einträge und NAT-Regeln

Zunächst wurden die DNS-Einträge gesetzt. Dabei lösen folgende Subdomains auf folgende Anwendungen auf:

- *chat.taskit.de* auf das Kommunikationsprogramm
- *wiki.taskit.de* auf das Wiki
- *mail.ledato.de* auf die MUA-Weboberfläche

Dazu mussten beim Provider DNS-A-Records gesetzt werden, die jeweils auf die öffentliche statische IP des Standardgateways (87.138.117.152) des Unternehmensnetz auflösen. Für das Wiki entfällt dieser Eintrag, da das Wiki nicht aus dem WAN genutzt werden soll:

```
1 chat.taskit.de 86400 IN A 87.138.117.152
2 mail.ledato.de 86400 IN A 87.138.117.152
```

Ferner müssen die am Standardgateway ankommenden Pakete noch auf die Webserver aufgelöst werden. Da ein Router in der Regel nicht in der Lage ist den HTTP-Header zu demanglen, und die verschiedenen Subdomains verschiedenen Hosts zuzuweisen, werden alle Pakete auf dem HTTP-Port (80) und dem HTTPS-Port (443) zunächst an den Server *clara* (10.1.100.48) weiter geleitet. Dazu wurden folgende NAT-Regeln erstellt, die die Destination IP im IP-Header ersetzen und die Pakete weiter leiten:

```
1 /ip firewall nat add chain=dstnat dst-address=87.138.117.152 dst-port=80 action=dst-nat
      to-address=10.1.100.48 to-port=80
2 /ip firewall nat add chain=dstnat dst-address=87.138.117.152 dst-port=443 action=dst-nat
       to-address=10.1.100.48 to-port=443
```

Damit beim Benutzen der Subdomains aus dem LAN diese nicht unnötig durch das Internet gerouted werden, wurden im Standardgateway weiterhin statische DNS-Einträge hinzugefügt, die direkt auf den jeweiligen Host auflösen. Hier folgt nun auch die Namensauflösung für das Wiki:

```
1 /ip dns static add address=10.1.100.48 name=chat.taskit.de
```

```
/ip dns static add address=10.1.100.48 name=wiki.taskit.de
/ip dns static add address=10.1.100.56 name=mail.ledato.de
```

4.2.2. Installation und funktionale Konfiguration der Webserver

Die Installation der Webserver kann bei Debian über die zentrale Paketverwaltung vorgenommen werden:

```
sudo apt install nginx
```

Zunächst galt es den Werbserver so zu konfigurieren, dass ausschließlich HTTPS verwendet wird. Dazu wurde eingehender HTTP-Traffic auf HTTPS umgeleitet und der Statuscode 301 (moved permanently) zurückgegeben:

```
listen 80;
listen [::]:80;
server_name mail.ledato.de chat.taskit.de 10.1.100.48;
return 301 https://$server_name$request_uri;
```

Für die Verwendung von HTTPS benötigen die Subdomains ein ausgestelltes Sicherheitszertifikat. Für Debian-Betriebssysteme können Zertifikate aus der Quelle *Let's Encrypt Authority X3* mittels dem CLI-Tool certbot ausgestellt werden. [4] Das Zertifikat wird auf die einzelnen Subdomains begrenzt und bevorzugt über den HTTPS-Port angefordert:

```
sudo certbot certonly --standalone --preferred-challenges tls-sni -d chat.taskit.de -d
    mail.ledato.de
```

Im Folgenden galt es das ausgestellte Zertifikat in die nginx-Konfiguration einzubinden:

```
listen 443;
listen [::]:443;
server_name 10.1.100.48 chat.taskit.de;

ssl on;
ssl_certificate /.../fullchain.pem;
ssl_certificate_key /.../privkey.pem;
```

Im nächsten Schritt wurden die Proxy-Einstellungen des Webservers programmiert, damit die eingehenden HTTP/S-Pakete auf die einzelnen Anwendungen weitergeleitet werden. Dazu sind Pakete aus der Quelle *chat.taskit.de* auf den designierten Port für das Kommunikationsprogramm (3000), aus der Quelle *wiki.taskit.de* auf den Apache Tomcat-Server vom Wiki (8080) und aus der Quelle *mail.ledato.de* auf den HTTPS-Port des autarken Mail Servers (10.1.100.56:443) weiterzuleiten. Ferner wurden u.a. die IP-Adresse des Clients und das Protokoll übergeben. Beispielhaft die Konfiguration für die Weboberfläche des Mail Servers:

```
proxy_pass https://10.1.100.56:443;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forward-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forward-Proto https;
proxy_set_header X-Nginx-Proxy true;
```

Funktional sollten die einzelnen Anwendungen nun über den Webbrowser sowohl aus dem LAN, als auch aus dem WAN (ausgenommen das Wiki) erreichbar sein.

4.2.3. Konfiguration der Webserver in Bezug auf die Datensicherheit

Im nächsten Schritt wurden noch die Parameter für das Verschlüsselungsprotokoll in Bezug auf die Datensicherheit optimiert. Mit der RFC 8446 wurde im August 2018 das Verschlüsselungsprotokoll TLS 1.3 dokumentiert, welches den aktuellen Standard darstellt. [5] TLS 1.2 sollte von der Serverseite definitiv weiterhin unterstützt werden, da eine Vielzahl von Clients noch nicht auf den neuesten Standard operiert, die Unterstützung von TLS 1.1 ist optional, ältere Versionen (TLS 1.0, SSL 3.0) gelten als unsicher und sollten daher nicht mehr unterstützt werden. TLS 1.3 wurde im Rahmen des Projektes nicht benutzt, die Gründe dafür sind unter *4.4 Begründungen für Abweichungen* aufgeführt.
Die RFC 7525 (Mai 2015) beschreibt die aktuelle *Best Current Practice* und gibt Empfehlungen für den sicheren Einsatz von TLS. [6] Als weitere Erweiterung zu TLS 1.2 wurde zudem die RFC 7905 (Juni 2016) veröffentlicht, welche den Einsatz von Chacha20-Poly1305 in TLS behandelt. [7] Auf Basis dieser Dokumente folgt für die verwendeten Cipher Suites:

- Ephemeral Diffie Hellman für den Schlüsselaustausch in Kombination mit RSA für die Authentifizierung (`ECDH_RSA`, bzw. `DH_RSA`)
- ChaCha20 in Kombination mit Poly1305, sowie AES im Galois/Counter Mode für die Verschlüsselung (`CHACHA20_POLY1305`, bzw. `AES_128_GCM` und `AES_256_GCM`)
- SHA256, bzw. SHA384 als Hashfunktion

Für die Abwärtskompatibilität wurde zudem RSA für den Schlüsselaustausch, sowie AES im CBC-Modus für die Verschlüsselung zugelassen.
RFC 8270 (Dezember 2017) empfiehlt für den DH-Schlüsselaustausch eine Schlüssellänge von mindestens 2048 bits. [8] Da OpenSSL für Debian 9 (Stretch) standardmäßig DH-Parameter mit 1024 bits zur Verfügung stellt, wurden diese zunächst über das CLI neu generiert. Dabei wurde direkt eine höhere Schlüssellänge von 4096 bits gewählt:

```
1 openssl dhparam -out /.../dhparam.pem 4096
```

Anschließend folgt in nginx das Einbinden des Verschlüsselungsprotokolls, der Cipher Suites und der DH-Parameter. Zudem wird nur die Verwendung der Cipher Suite des Servers in der vom Server vorgegebenen Reihenfolge unterstützt:

```
1 ssl_dhparam /.../dhparam.pem;
2 ssl_protocols TLSv1.1 TLSv1.2;
3 ssl_ciphers 'ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-
    AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-
    RSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:ECDHE-
    ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-
    SHA384:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA:ECDHE-
    RSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA:DHE-RSA-AES256-SHA256:DHE-
    RSA-AES256-SHA:ECDHE-ECDSA-DES-CBC3-SHA:ECDHE-RSA-DES-CBC3-SHA:EDH-RSA-DES-CBC3-SHA:
    AES128-GCM-SHA256:AES256-GCM-SHA384:AES128-SHA256:AES256-SHA256:AES128-SHA:AES256-
    SHA:DES-CBC3-SHA:!DSS';
4 ssl_prefer_server_ciphers on;
```

Im letzten Schritt der Webserver-Konfiguration wurden nun noch Maßnahmen gegen bekannte Techniken von Computerhackern eingebunden. Dazu werden dem HTTP-Header Parameter hinzugefügt, die diese Techniken unterbinden: [9]

- Clickjacking ist eine bekannte Technik, bei der eine Webseite mit einem iFrame überlagert wird (und dadurch den User fälschlicherweise durch den überlagerten Content eine vom Hacker definierte Aktion auslöst). Das Einbinden von iFrames auf fremden Domains wird unterbunden um diesem Vorgehen vorzubeugen.
- XSS ist eine der bekanntesten Methoden schadhaften Code durch Dritte in eine bekannte Seite einzuspeisen. Dabei wird z.B. schadhafter Code in Eingabefeldern abgelegt, der dann unbewusst vom Server ausgeführt wird. Moderne Browser verwenden jedoch XSS-Filter, die durch HTTP-Header-Optionen aktiviert werden können.
- Content sniffing ist eine Methode, bei der der Datenstream in Byteform ausgelesen wird um darüber Metadaten zu erfassen. Der HTTP-Header wird dahingehend angepasst, dass moderne Webbrowser dies unterbinden.

Für die nginx-Konfiguration folgt damit:

```
1 add_header X-Frame-Options SAMEORIGIN;
2 add_header X-Content-Type-Options nosniff;
3 add_header X-XSS-Protection "1; mode=block";
```

Damit wäre die Webserver-Konfiguration abgeschlossen. Als letzten Schritt galt es nun die Konfiguration für die automatisierte Erneuerung des Sicherheitszertifikats vorzunehmen. Mit Hilfe des Cron-Daemon kann unter Debian eine zeitbasierte Ausführung eines Prozesses gesteuert werden um wiederkehrende Aufgaben in Form von Cronjobs zu automatisieren. Die Zertifikatserneuerung findet wie bereits die Zertifikatserstellung mit Hilfe von certbot statt. Zusätzlich muss nginx für die Zeit der Zertifikatserneuerung seinen Prozess beenden, da der Webserver den HTTPS-Port nutzt, der auch von certbot für die Zertifikatserneuerung genutzt wird. Daraus resultiert folgender cronjob:

```
1 @weekly sudo certbot renew --pre-hook "systemctl stop nginx" --post-hook "systemctl
      start nginx"
```

Die komplette Konfiguration kann dem Anhang unter *A.3 Webserver-Konfiguration* entnommen werden.

4.3. Qualitätssicherung

4.3.1. Testen der funktionalen Konfiguration

Für die Qualitätssicherung wurde nun zunächst die syntaktische Konfiguration der Webserver überprüft. Mit nginx kann über das CLI die Konfiguration auf Syntaxfehler überprüft werden. Zudem werden alle Abhängigkeiten (DH-Parameter und SSL-Zertifikate), die in der Konfiguration verwendet wurden, testweise geöffnet:

```
1 nginx -t
```

Nach der Behebung syntaktischer Fehler wurde im Folgenden der Systemprozess von nginx gestartet:

```
systemctl start nginx
```

Auftretende Fehler beim Starten des Systemprozesses können mit `journalctl -xe` über die Kommandozeile untersucht werden. Im Normalfall startet der Prozess nach der Behebung syntaktischer Fehler ohne auftretende Probleme.

Für die funktionale Korrektheit der Webserver-Konfiguration wurden die Subdomains zunächst aus dem LAN getestet. Dazu wurde von einem Windows-Client mittels traceroute festgestellt, dass die Pakete direkt auf den jeweiligen Host auflösen und nicht durch das Internet geroutet werden:

```
C:\>tracert wiki.taskit.de
Routenverfolgung zu wiki.taskit.de [10.1.100.48]
über maximal 30 Hops:
  1    1 ms    1 ms    8 ms  clara [10.1.100.48]
Ablaufverfolgung beendet.

C:\>tracert chat.taskit.de
Routenverfolgung zu chat.taskit.de [10.1.100.48]
über maximal 30 Hops:
  1    1 ms    1 ms    1 ms  clara [10.1.100.48]
Ablaufverfolgung beendet.

C:\>tracert mail.ledato.de
Routenverfolgung zu mail.ledato.de [10.1.100.56]
über maximal 30 Hops:
  1    2 ms    5 ms    1 ms  mail.ledato.de [10.1.100.56]
Ablaufverfolgung beendet.
```

Damit ist die Korrektheit der DNS-Konfiguration im Standardgateway sichergestellt.

Im nächsten Schritt werden über einen Webbrowser die Webseiten unter den jeweiligen Subdomains aufgerufen und überprüft, ob der Inhalt korrekt dargestellt wird. Dies wurde erfolgreich für alle Domains durchgeführt.

Anschließend wurde die funktionale Konfiguration über das WAN getestet. Dazu wurde der Arbeitsplatzrechner in ein WLAN von einem mobilen Hotspot versetzt und die jeweiligen Subdomains über ein ICMP-Echo-Request angesprochen. An dieser Stelle wurde für die Subdomains *chat.taskit.de* und *mail.ledato.de* eine Antwort von der öffentlichen IP des Standardgateways erwartet und für die Subdomain *wiki.taskit.de* eine Fehlermeldung, dass der Host nicht gefunden werden konnte:

```
C:\>ping mail.ledato.de
Ping wird ausgeführt für mail.ledato.de [87.138.117.152] mit 32 Bytes Daten:
Antwort von 87.138.117.152: Bytes=32 Zeit=43ms TTL=54
[...]

C:\>ping chat.taskit.de
Ping wird ausgeführt für chat.taskit.de [87.138.117.152] mit 32 Bytes Daten:
Antwort von 87.138.117.152: Bytes=32 Zeit=50ms TTL=54
[...]

```

```
11 C:\>ping wiki.taskit.de
12 Ping-Anforderung konnte Host "wiki.taskit.de" nicht finden. Überprüfen Sie den Namen,
       und versuchen Sie es erneut.
```

Damit ist die Korrektheit der DNS-Konfiguration beim Provider sichergestellt. Um die funktionale Konfiguration der Webserver, sowie der NAT-Einträge zu testen wurden die jeweiligen Subdomains sowohl über den HTTP-Port, als auch über den HTTPS-Port aufgerufen und überprüft. Dies wurde erfolgreich für beide Domains durchgeführt und der Inhalt wurde korrekt dargestellt.

4.3.2. Testen der Maßnahmen zur Datensicherheit

Für das Testen der implementierten Maßnahmen zur Datensicherheit wurde maßgeblich das Tool *SSL Labs* verwendet und die Ergebnisse untersucht. *SSL Labs* untersucht dabei das Zertifikat, die unterstützten Verschlüsselungsprotokolle und die Cipher Suite. [10] Ferner wird eine Handshake-Simulation für verschiedene Betriebssysteme und Webbrowser durchgeführt und gegen bekannte Sicherheitslücken getestet (u.a. POODLE, Bleichenbacher, Heartbleed). [11, 12, 13] Nach mehrmaligem Optimieren der Webserver-Konfiguration wurden final für beide öffentlich zugänglichen Subdomains folgende Resultate erzielt:

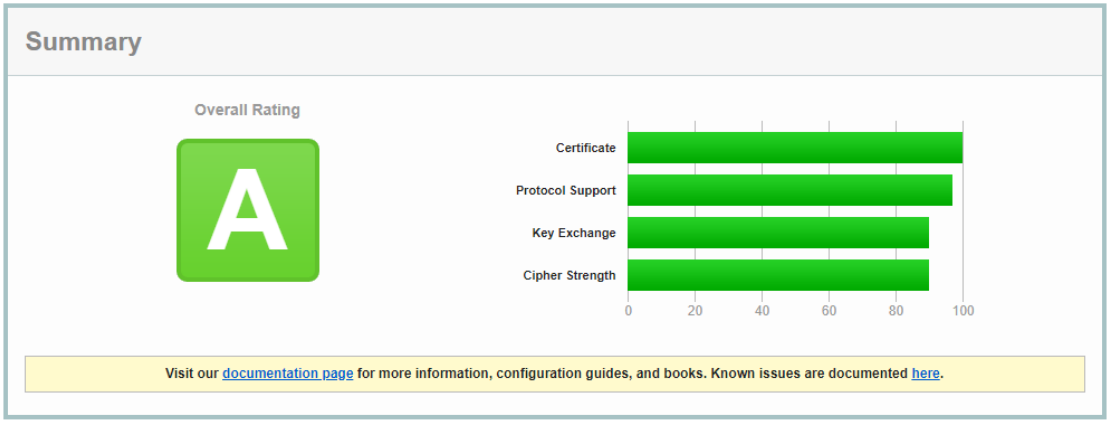


Abbildung 1: Zusammenfassung der Sicherheitstests durch SSL Labs

Die ausführlichen Testergebnisse können dem Anhang unter *A.4 SSL Labs ausführliche Testdokumentation* entnommen werden.

Weiterhin wurde die automatisierte Zertifikatserneuerung getestet. Dafür kann *certbot* in einem Probelauf gestartet werden:

```
1 sudo certbot renew --pre-hook "systemctl stop nginx" --post-hook "systemctl start nginx"
      --dry-run
```

Der Probelauf war erfolgreich, so dass vor dem Ablaufdatum der Zertifikate diese über den Cronjob automatisiert erneuert werden sollten.

4.4. Begründungen für Abweichungen

In der Webserver-Konfiguration wurde auf TLS in seiner aktuellen Version 1.3 verzichtet. Dies hängt maßgeblich mit dem Zeitaufwand und dem Update-Zyklus von Debian zusam-

men. OpenSSL unterstützt auf Debian 9 (Stretch) kein TLS 1.3, da Debian 9 sich seit dem 5. Februar 2017 im full-freeze befindet und TLS 1.3 erst 2018 durch eine RFC dokumentiert wurde. Aktuellere Versionen von OpenSSL führen zu Konflikten mit dem Betriebssystem, so dass man hier auf Alternativen angewiesen ist. Infolgedessen müssten zunächst Alternativen recherchiert werden und die nginx-Codebase danach manuell gegen die TLS-Software gebaut werden. Dies würde die Zeitaufwendung für das Projekt unplanmäßig deutlich verlängern. Da sich Debian 10 (Buster) zudem bereits im full-freeze befindet, das die aktuelle OpenSSL-Version unterstützt und der Release von Debian 10 (Buster) für Mitte diesen Jahres geplant ist, überwiegt hier die Entscheidung den Support für TLS 1.3 nach einem Update des neuen Betriebssystems nachzurüsten.

5. Projektabschluss

5.1. Projektübergabe

Das Projekt wurde in einem Abschlussmeeting vorgestellt und an den Auftraggeber des Projektes übergeben. Dabei wurde insbesondere die korrekte Funktionalität, sowie die Maßnahmen in Bezug auf die Datensicherheit besprochen. Insbesondere die Notwendigkeit des stetigen Updates der Verschlüsselungsprotokolle und der Cipher Suites wurde besprochen, insbesondere mit dem Bezug auf das Einbinden von TLS 1.3 nach Update des Betriebssystems auf Debian 10 (Buster). Das vollständige Übergabeprotokoll kann dem Anhang unter *B.2 Abnahmeprotokoll* entnommen werden.

5.2. Soll/Ist Vergleich

Im Hinblick auf die Zielsetzung des Projektes wurden die Projektziele mit geringen Abweichungen zur vollen Zufriedenheit des Auftraggebers erreicht:

Sachziel: Die funktionalen Anforderungen wurden in vollem Maße umgesetzt. Einzig bei den Maßnahmen zur Datensicherheit kommt beim Verschlüsselungsprotokoll aus den unter *4.4 Begründungen für Abweichungen* beschriebenen Gründen nicht TLS 1.3 momentan nicht zum Einsatz, das jedoch nach einem Betriebssystemupdate noch 2019 eingebunden wird.

Kostenziel: Das Kostenziel wurde erreicht. Für die gesamte Projektumsetzung wurde auf kostenfreie OpenSource-Software gesetzt, so dass mit Ausnahme des personellen Aufwandes keine Kosten für Lizenzgebühren oder Hardwarebeschaffungen auftraten

Zeitziel: Die Zeitplanung konnte mit wenigen Ausnahmen eingehalten werden. Die geringen Abweichungen sind mit unterschätztem bzw. überschätztem Aufwand zu begründen:

Tabelle 8: Zeitplanung Soll-/Ist-Vergleich

Position	Beschreibung	Geplant	Tatsächlich	Differenz
1	Definitionsphase	2 h	2 h	
2	Planungsphase	6 h	7 h	+1 h
3	Durchführungsphase	12 h	10 h	−2 h
4	Qualitätsmanagement	6 h	7 h	+1 h
5	Projektabschluss & -übergabe	2 h	1 h	−1 h
6	Dokumentation	7 h	8 h	+1 h

Qualitätsziel: Alle Konfigurationen wurden im Rahmen des Projektes bezüglich ihrer Funktionalität und den Maßnahmen zur Datensicherheit getestet.

5.3. Projektfazit/Reflexion

Das Projekt war meines Erachtens eine sehr gute Aufgabe im Hinblick auf die Thematiken der vergangen 18 Monate während der Ausbildung. Während des Projektes mussten viele Themenbereiche im praktischen Umfeld angewendet werden (Routerkonfiguration, DNS, NAT, zeitgesteuerte Prozesse, bash, Datenschutz und Datensicherheit). Insbesondere die Umsetzung technischer Maßnahmen zur Datensicherheit um den Schutz personenbezogenen Daten gewährleisten zu können ist ein sehr modernes und komplexes Thema, dessen Umsetzung immer wieder Optimierungspotenzial bietet. Das Projekt gab einen Einblick in die Vielzahl sicherheitsrelevanter Aspekte und wie diese ausgehebelt werden können, was ich gerne thematisch noch viel weiter vertieft hätte. Alles in allem bin ich mit dem Projektergebnis sehr zufrieden und hoffe, dass ich mich in Zukunft durch weitere Aufgaben im Bereich des Datenschutzes und der Datensicherheit weiter in der Thematik vertiefen kann.

Literatur

[1] *Webpräsenz des Unternehmens taskit GmbH*. https://www.taskit.de,

[2] *Usage of Web servers*. https://w3techs.com/technologies/overview/web_server/all, Juni 2019

[3] *NGINX vs. Apache (Pro/Con Review, Uses, & Hosting for Each)*. https://www.hostingadvice.com/how-to/nginx-vs-apache/,

[4] *Kette des Vertrauens*. https://letsencrypt.org/de/certificates/,

[5] RESCORLA, E.: *RFC 8446: The Transport Layer Security (TLS) Protocol Version 1.3*. https://tools.ietf.org/html/rfc8446, August 2018

[6] SHEFFER, Y. ; HOLZ, R. ; SAINT-ANDRE, P.: *RFC 7525: Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)*. https://tools.ietf.org/html/rfc7525, Mai 2015

[7] LANGLEY, A. ; CHANG, W. ; MAVROGIANNOPOULOS, N. ; STROMBERGSON, J. ; JOSEFSSON, S.: *RFC 7905: ChaCha20-Poly1305 Cipher Suites for Transport Layer Security (TLS)*. https://tools.ietf.org/html/rfc7905, Juni 2016

[8] VELVINDRON, L. ; BAUSHKE, M.: *RFC 8270: Increase the Secure Shell Minimum Recommended Diffie-Hellman Modulus Size to 2048 Bits*. https://tools.ietf.org/html/rfc8270, Dezember 2017

[9] *HTTP headers*. https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers,

[10] *SSL Server Test*. https://www.ssllabs.com/ssltest/,

[11] SHEFFER, Y. ; HOLZ, R. ; SAINT-ANDRE, P.: *RFC 7457: Summarizing Known Attacks on Transport Layer Security (TLS) and Datagram TLS (DTLS)*. https://tools.ietf.org/html/rfc7457, Februar 2015

[12] SEGGELMANN, R. ; TUEXEN, M. ; WILLIAMS, M.: *RFC 6520: Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension*. https://tools.ietf.org/html/rfc6250, Februar 2012

[13] SEGGELMANN, R. ; TUEXEN, M. ; WILLIAMS, M.: *RFC 8447: IANA Registry Updates for TLS and DTLS*. https://tools.ietf.org/html/rfc8447, August 2018

[14] *nginx Wiki*. https://www.nginx.com/resources/wiki/,

A. Anhänge

A.1. Projektablaufplan

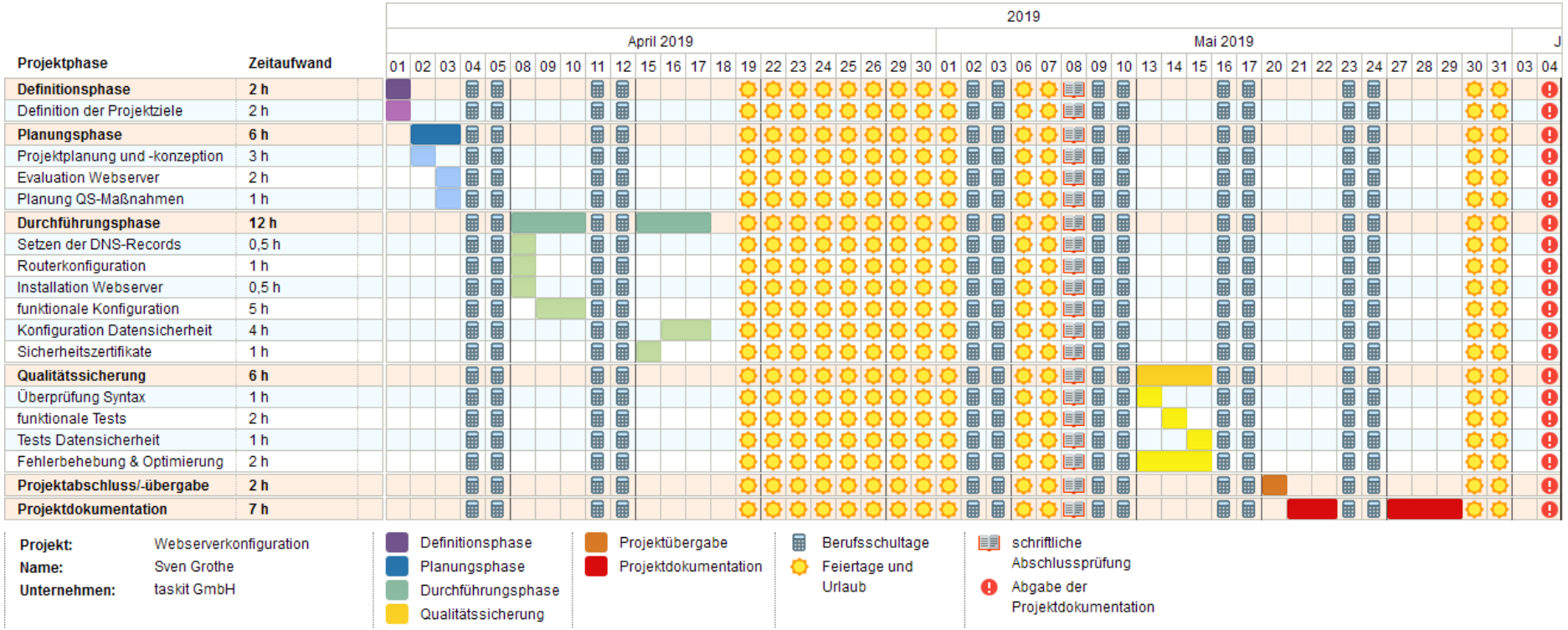


Abbildung 2: Projektablaufplan

A.2. Netzplan

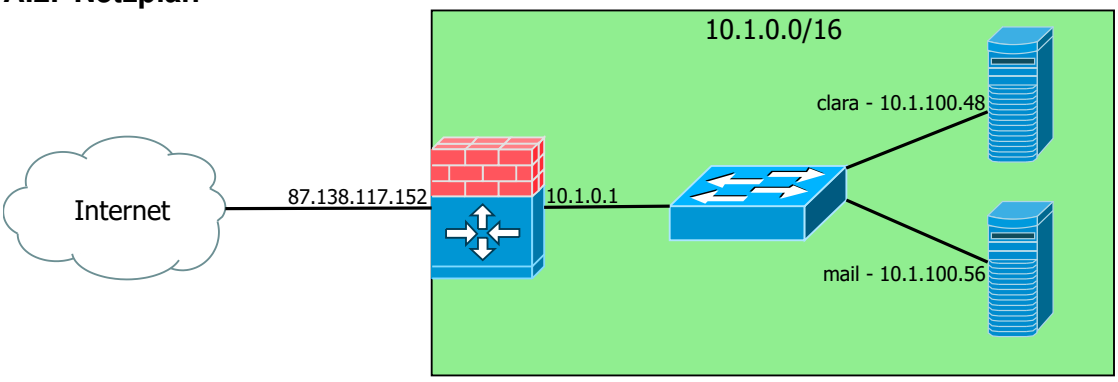


Abbildung 3: Netzplan des Unternehmensnetzes

A.3. Webserver-Konfiguration

A.3.1. Webserverkonfiguration für das Kommunikationsprogramm auf *clara*

```
# Upstreams
upstream backend {
    server 127.0.0.1:3000;
}

# Redirect Options
server {
  listen 80;
  listen [::]:80;
  server_name chat.taskit.de 10.1.100.48;
  # enforce https
  return 301 https://$server_name$request_uri;
}

# HTTPS Server
server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    server_name 10.1.100.48 chat.taskit.de;

    error_log /.../rocketchat.access.log;

    ssl on;
    ssl_certificate /.../fullchain.pem;
    ssl_certificate_key /.../privkey.pem;

    ssl_dhparam /etc/ssl/certs/dhparam.pem;
    ssl_ciphers 'ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-
        AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:
        ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:
        ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-SHA:ECDHE-
        RSA-AES256-SHA384:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-
        AES256-SHA:ECDHE-RSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA:DHE-RSA
        -AES256-SHA256:DHE-RSA-AES256-SHA:ECDHE-ECDSA-DES-CBC3-SHA:ECDHE-RSA-DES-CBC3-
```

```
        SHA:EDH-RSA-DES-CBC3-SHA:AES128-GCM-SHA256:AES256-GCM-SHA384:AES128-SHA256:
        AES256-SHA256:AES128-SHA:AES256-SHA:DES-CBC3-SHA:!DSS';
    ssl_prefer_server_ciphers on;

    ssl_protocols TLSv1.1 TLSv1.2;

    resolver 8.8.8.8;

    location / {
        proxy_pass http://10.1.100.48:3000/;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header Host $http_host;

        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forward-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forward-Proto $scheme;
        proxy_set_header X-Nginx-Proxy true;

        add_header X-Frame-Options "SAMEORIGIN";
        add_header X-XSS-Protection "1; mode=block" always;
        add_header X-Content-Type-Options "nosniff" always;

        proxy_redirect off;
    }
}
```

Listing 1: Webserverkonfiguration für das Kommunikationsprogramm auf *clara*

A.3.2. Webserverkonfiguration für den MUA auf *clara*

```
# Redirect Options
server {
  listen 80;
  listen [::]:80;
  server_name mail.ledato.de;
  # enforce https
  return 301 https://$server_name$request_uri;
}

server {
  listen 443 ssl http2;
  listen [::]:443 ssl http2;
  server_name mail.ledato.de;

  ssl on;
  ssl_certificate /.../fullchain.pem;
  ssl_certificate_key /.../privkey.pem;

  ssl_dhparam /.../dhparam.pem;
```

```
  ssl_ciphers 'ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-
      AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-
      RSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:ECDHE-
      ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-
      AES256-SHA384:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES256-
      SHA:ECDHE-RSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA:DHE-RSA-AES256-
      SHA256:DHE-RSA-AES256-SHA:ECDHE-ECDSA-DES-CBC3-SHA:ECDHE-RSA-DES-CBC3-SHA:EDH-RSA-
      DES-CBC3-SHA:AES128-GCM-SHA256:AES256-GCM-SHA384:AES128-SHA256:AES256-SHA256:
      AES128-SHA:AES256-SHA:DES-CBC3-SHA:!DSS';
  ssl_prefer_server_ciphers on;

  ssl_protocols TLSv1.1 TLSv1.2;

  location / {
    proxy_pass https://10.1.100.56:443;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forward-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forward-Proto $scheme;
    proxy_set_header X-Nginx-Proxy true;
  }
}
```

Listing 2: Webserverkonfiguration für den MUA auf *clara*

A.3.3. Webserverkonfiguration für den MUA auf *mail*

```
server {
    listen 80;
    listen [::]:80;
    server_name mail.ledato.de;
    # enforce https
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    server_name mail.ledato.de;

    ssl on;
    ssl_certificate /.../fullchain.pem;
    ssl_certificate_key /.../privkey.pem;

    ssl_dhparam /.../dhparam.pem;
    ssl_protocols TLSv1.1 TLSv1.2;

    ssl_ciphers 'ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-
        AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:
        ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:
        ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-SHA:ECDHE-
        RSA-AES256-SHA384:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-
        AES256-SHA:ECDHE-RSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA:DHE-RSA
```

```
        -AES256-SHA256:DHE-RSA-AES256-SHA:ECDHE-ECDSA-DES-CBC3-SHA:ECDHE-RSA-DES-CBC3-
        SHA:EDH-RSA-DES-CBC3-SHA:AES128-GCM-SHA256:AES256-GCM-SHA384:AES128-SHA256:
        AES256-SHA256:AES128-SHA:AES256-SHA:DES-CBC3-SHA:!DSS';
    ssl_prefer_server_ciphers on;

    resolver 8.8.8.8;

    root /var/www/html/;
    index index.html index.htm index.php;

    add_header X-Frame-Options "SAMEORIGIN";
    add_header X-XSS-protection "1; mode=block" always;
    add_header X-Content-Type-Options "nosniff" always;

    #für PHP7
    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/run/php/php7.0-fpm.sock;
    }
}
```

Listing 3: Webserverkonfiguration für den MUA auf *mail*

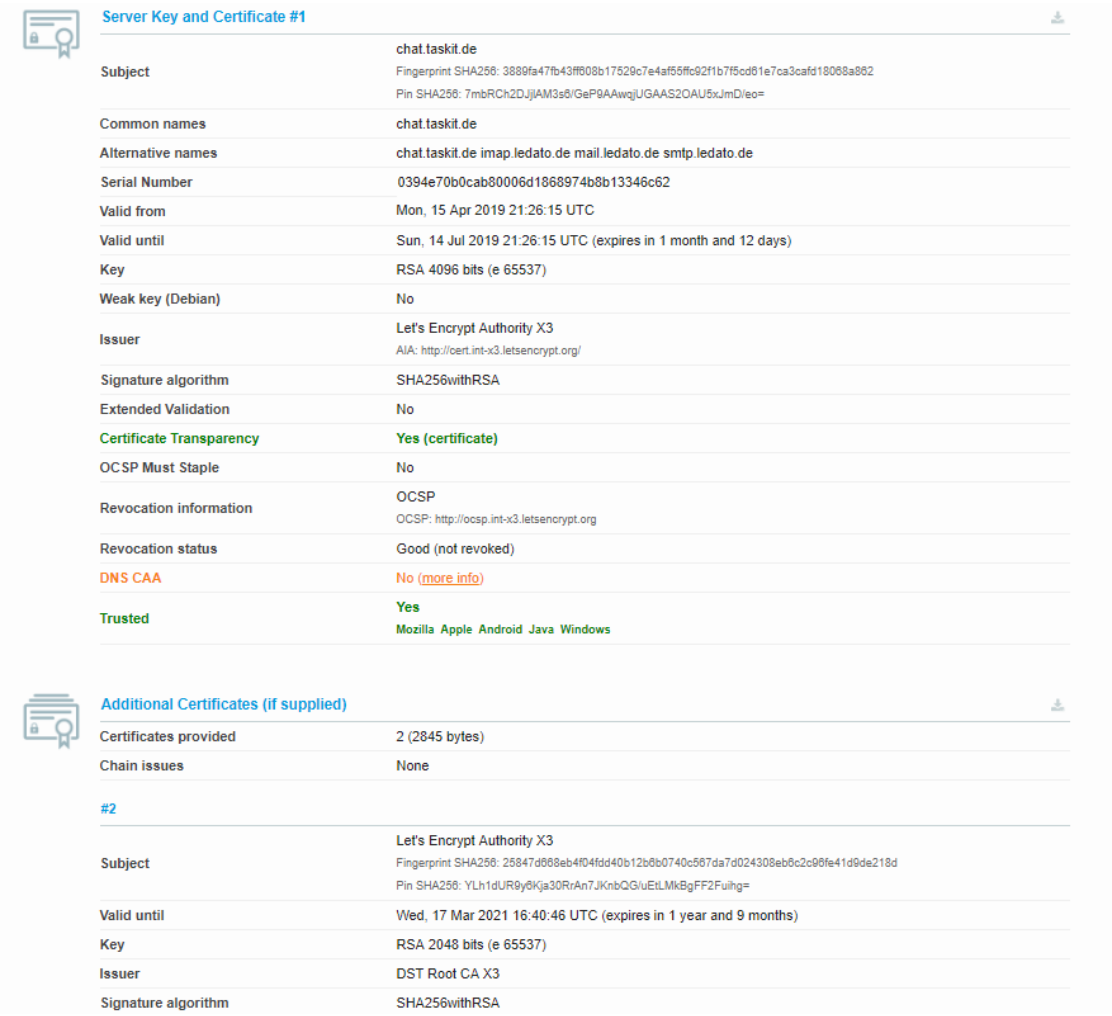
A.3.4. Webserverkonfiguration für das Wiki auf *clara*

```
server {
    listen      80;
    server_name wiki.taskit.de;

    root /var/www/html;

    location / {
        # All "root" requests will have /xwiki appended and redirected
        rewrite ^ $scheme://$server_name/xwiki$request_uri? permanent;
    }

    location ^~ /xwiki {
       # redirect to XWiki application in Tomcat
       proxy_pass http://localhost:8080;
       proxy_set_header X-Real-IP $remote_addr;
       proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
       proxy_set_header Host $http_host;
       proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

Listing 4: Webserverkonfiguration für das Wiki auf *clara*

A.4. SSL Labs ausführliche Testdokumentation

A.4.1. Zertifikatsdetails



Server Key and Certificate #1

Subject	chat.taskit.de Fingerprint SHA256: 3889fa47fb43ff608b17529c7e4af55ffc92f1b7f5cd61e7ca3cafd18068a862 Pin SHA256: 7mbRCh2DJjlAM3s6/GeP9AAwqjUGAAS2OAU5xJmD/eo=
Common names	chat.taskit.de
Alternative names	chat.taskit.de imap.ledato.de mail.ledato.de smtp.ledato.de
Serial Number	0394e70b0cab80006d1868974b8b13346c62
Valid from	Mon, 15 Apr 2019 21:26:15 UTC
Valid until	Sun, 14 Jul 2019 21:26:15 UTC (expires in 1 month and 12 days)
Key	RSA 4096 bits (e 65537)
Weak key (Debian)	No
Issuer	Let's Encrypt Authority X3 AIA: http://cert.int-x3.letsencrypt.org/
Signature algorithm	SHA256withRSA
Extended Validation	No
Certificate Transparency	Yes (certificate)
OCSP Must Staple	No
Revocation information	OCSP OCSP: http://ocsp.int-x3.letsencrypt.org
Revocation status	Good (not revoked)
DNS CAA	No (more info)
Trusted	Yes Mozilla Apple Android Java Windows

Additional Certificates (if supplied)

Certificates provided	2 (2845 bytes)
Chain issues	None
#2	
Subject	Let's Encrypt Authority X3 Fingerprint SHA256: 25847d668eb4f04fdd40b12b6b0740c567da7d024308eb6c2c96fe41d9de218d Pin SHA256: YLh1dUR9y6Kja30RrAn7JKnbQG/uEtLMkBgFF2Fuihg=
Valid until	Wed, 17 Mar 2021 16:40:46 UTC (expires in 1 year and 9 months)
Key	RSA 2048 bits (e 65537)
Issuer	DST Root CA X3
Signature algorithm	SHA256withRSA

Abbildung 4: Zertifikatsdetails

A.4.2. unterstützte Protokolle

Protocols

TLS 1.3	No
TLS 1.2	Yes
TLS 1.1	Yes
TLS 1.0	No
SSL 3	No
SSL 2	No

Abbildung 5: unterstützte Protokolle

A.4.3. Cipher Suites

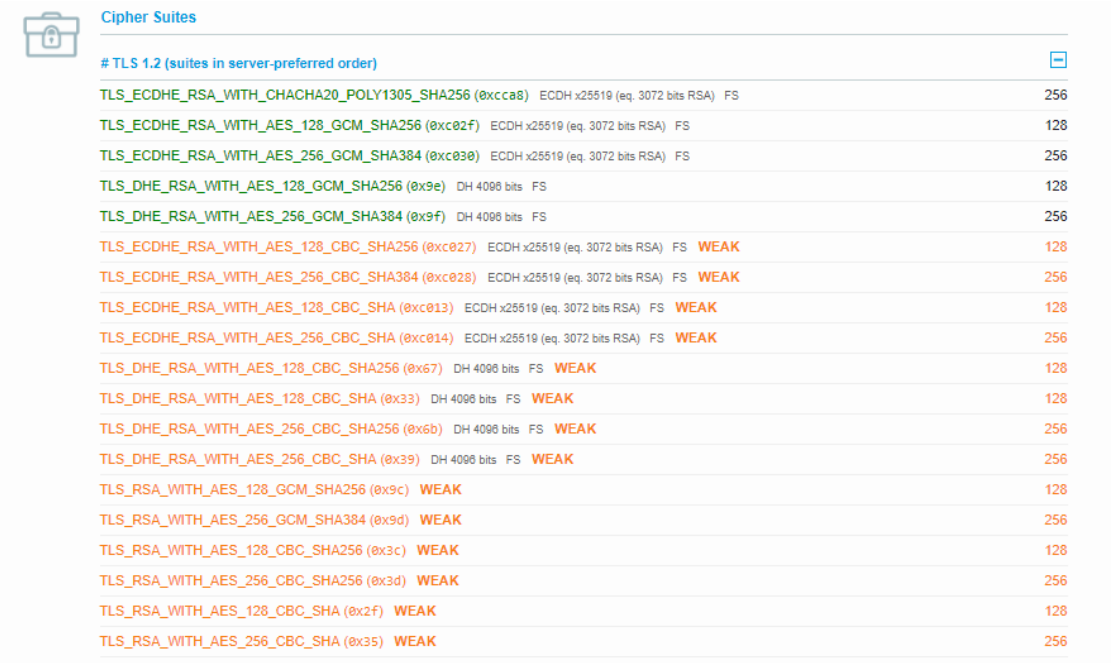
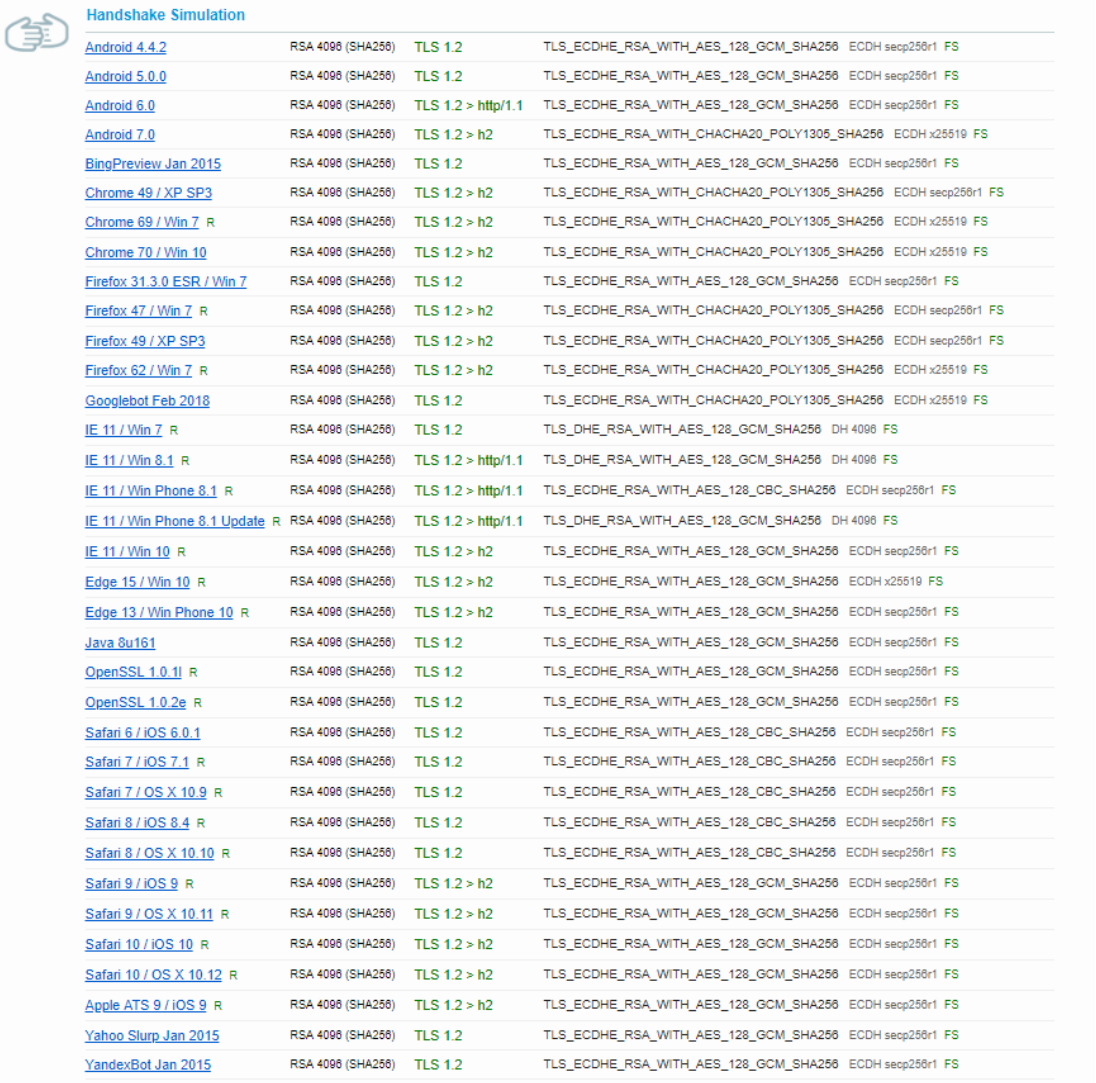


Abbildung 6: Cipher Suites

A.4.4. SSL Handshake Simulation

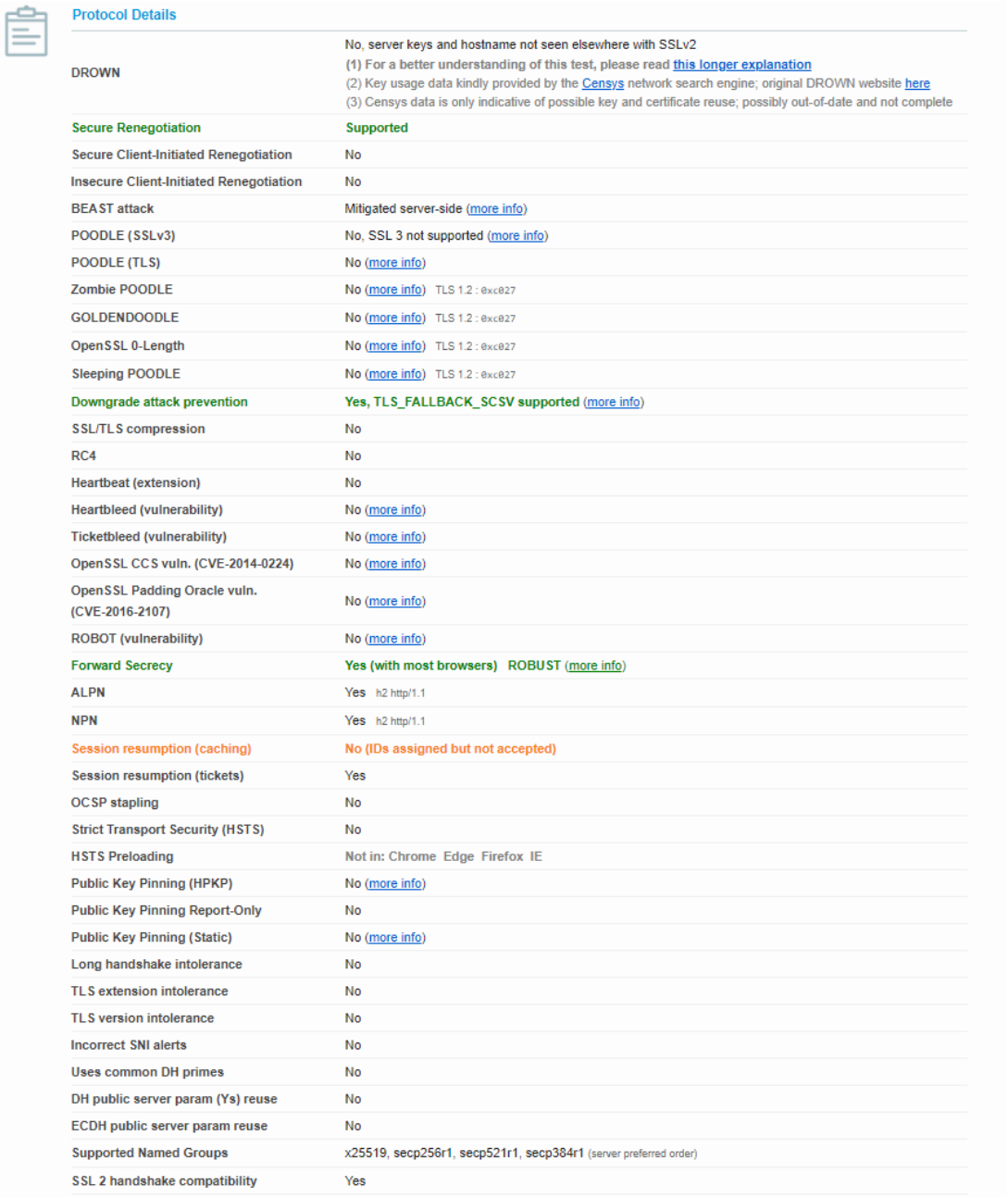


Handshake Simulation

Android 4.4.2	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Android 5.0.0	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Android 6.0	RSA 4096 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Android 7.0	RSA 4096 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 ECDH x25519 FS
BingPreview Jan 2015	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Chrome 49 / XP SP3	RSA 4096 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 ECDH secp256r1 FS
Chrome 69 / Win 7 R	RSA 4096 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 ECDH x25519 FS
Chrome 70 / Win 10	RSA 4096 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 ECDH x25519 FS
Firefox 31.3.0 ESR / Win 7	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Firefox 47 / Win 7 R	RSA 4096 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 ECDH secp256r1 FS
Firefox 49 / XP SP3	RSA 4096 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 ECDH secp256r1 FS
Firefox 62 / Win 7 R	RSA 4096 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 ECDH x25519 FS
Googlebot Feb 2018	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 ECDH x25519 FS
IE 11 / Win 7 R	RSA 4096 (SHA256)	TLS 1.2	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 DH 4096 FS
IE 11 / Win 8.1 R	RSA 4096 (SHA256)	TLS 1.2 > http/1.1	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 DH 4096 FS
IE 11 / Win Phone 8.1 R	RSA 4096 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 ECDH secp256r1 FS
IE 11 / Win Phone 8.1 Update R	RSA 4096 (SHA256)	TLS 1.2 > http/1.1	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 DH 4096 FS
IE 11 / Win 10 R	RSA 4096 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Edge 15 / Win 10 R	RSA 4096 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH x25519 FS
Edge 13 / Win Phone 10 R	RSA 4096 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Java 8u161	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
OpenSSL 1.0.1l R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
OpenSSL 1.0.2e R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Safari 6 / iOS 6.0.1	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 ECDH secp256r1 FS
Safari 7 / iOS 7.1 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 ECDH secp256r1 FS
Safari 7 / OS X 10.9 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 ECDH secp256r1 FS
Safari 8 / iOS 8.4 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 ECDH secp256r1 FS
Safari 8 / OS X 10.10 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 ECDH secp256r1 FS
Safari 9 / iOS 9 R	RSA 4096 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Safari 9 / OS X 10.11 R	RSA 4096 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Safari 10 / iOS 10 R	RSA 4096 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Safari 10 / OS X 10.12 R	RSA 4096 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Apple ATS 9 / iOS 9 R	RSA 4096 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Yahoo Slurp Jan 2015	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
YandexBot Jan 2015	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS

Abbildung 7: SSL Handshake Simulation

A.4.5. Testszenarien gegen das Verschlüsselungsprotokoll



Protocol Details

DROWN	No, server keys and hostname not seen elsewhere with SSLv2 (1) For a better understanding of this test, please read this longer explanation (2) Key usage data kindly provided by the Censys network search engine; original DROWN website here (3) Censys data is only indicative of possible key and certificate reuse; possibly out-of-date and not complete
Secure Renegotiation	Supported
Secure Client-Initiated Renegotiation	No
Insecure Client-Initiated Renegotiation	No
BEAST attack	Mitigated server-side (more info)
POODLE (SSLv3)	No, SSL 3 not supported (more info)
POODLE (TLS)	No (more info)
Zombie POODLE	No (more info) TLS 1.2 : 0xc027
GOLDENDOODLE	No (more info) TLS 1.2 : 0xc027
OpenSSL 0-Length	No (more info) TLS 1.2 : 0xc027
Sleeping POODLE	No (more info) TLS 1.2 : 0xc027
Downgrade attack prevention	Yes, TLS_FALLBACK_SCSV supported (more info)
SSL/TLS compression	No
RC4	No
Heartbeat (extension)	No
Heartbleed (vulnerability)	No (more info)
Ticketbleed (vulnerability)	No (more info)
OpenSSL CCS vuln. (CVE-2014-0224)	No (more info)
OpenSSL Padding Oracle vuln. (CVE-2016-2107)	No (more info)
ROBOT (vulnerability)	No (more info)
Forward Secrecy	Yes (with most browsers) ROBUST (more info)
ALPN	Yes h2 http/1.1
NPN	Yes h2 http/1.1
Session resumption (caching)	No (IDs assigned but not accepted)
Session resumption (tickets)	Yes
OCSP stapling	No
Strict Transport Security (HSTS)	No
HSTS Preloading	Not in: Chrome Edge Firefox IE
Public Key Pinning (HPKP)	No (more info)
Public Key Pinning Report-Only	No
Public Key Pinning (Static)	No (more info)
Long handshake intolerance	No
TLS extension intolerance	No
TLS version intolerance	No
Incorrect SNI alerts	No
Uses common DH primes	No
DH public server param (Ys) reuse	No
ECDH public server param reuse	No
Supported Named Groups	x25519, secp256r1, secp521r1, secp384r1 (server preferred order)
SSL 2 handshake compatibility	Yes

Abbildung 8: Testszenarien gegen das Verschlüsselungsprotokoll

B. Dokumente

Auf den folgenden Seiten sind ein Angebot, sowie das Abnahmeprotokoll abgebildet. Da dies eigenständige vollwertige Dokumente sind, sind diese nicht im Layout der übrigen Dokumentation gehalten. Inhalt und Layout sind dabei Eigenleistungen des Prüfungsbewerber, es wurde kein Corporate Design verwendet. Da es sich hierbei um ein internes Projekt handelt ist das Angebot als fiktiv zu betrachten.

taskit GmbH | Groß-Berliner Damm 37 | 12487 Berlin

Herr [redacted]

taskit GmbH
Groß-Berliner Damm 37
12487 Berlin

Angebotsdatum: 15.03.2019
Gültigkeitsdatum: 29.03.2019

Angebots-Nr.: DL 190315-1
Kunden-Nr.: 7-614638

Sehr geehrter [redacted],

vielen Dank für Ihr Interesse an unserer Dienstleistung. Bezugnehmend auf Ihre Anforderungen unterbreiten wir Ihnen folgendes Angebot:

Pos.	Bezeichnung	Menge/Std.	Stückpreis	Preis
1	Installation & Konfiguration (Webserverkonfiguration, Erstellung von Maßnahmen zur Datensicherheit, Qualitätssicherung, Dokumentation)	35 h	69,00 €	2.415,00 €

Summe netto:	2.415,00 €
Umsatzsteuer 19 %:	458,85 €
Gesamtkosten:	**2.873,85 €**

taskit GmbH
Groß-Berliner Damm 37
12487 Berlin
Geschäftsführer: [redacted]
Ust-ID: [redacted]

Kontakt
Tel: +49 (0)30 611 295 - 0
Fax: +49 (0)30 611 295 - 10
E-Mail: info@taskit.de
Support: support@taskit.de
Web: www.taskit.de

taskit

Das Angebot ist ab Erstelldatum 14 Tage freibleibend gültig.

Alle Preise verstehen sich zuzüglich der gesetzlichen Mehrwertsteuer und setzen die Erteilung einer Einzugsermächtigung voraus. Die Zahlung der monatlichen bzw. jährlichen Beträge erfolgt im Voraus.

Die Allgemeinen Geschäftsbedingungen der taskit GmbH sind Bestandteil dieses Auftrages und unter https://www.taskit.de/allgemeine-geschaeftsbedingungen.html einzusehen.

Hinweise zur Verarbeitung personenbezogener Daten können Sie unserer Datenschutzerklärung (https://www.taskit.de/datenschutzbestimmungen.html) entnehmen.

Mit freundlichen Grüßen,

[redacted]

Geschäftsführer taskit GmbH

Mit Ihrer Unterschrift wird aus diesem Angebot ein Auftrag:	**Hiermit bestätigen wir die Annahme Ihres Auftrages:**
	Berlin, den
(Ort/Datum)	
Unterschrift Auftraggeber	Unterschrift taskit GmbH

taskit GmbH
Groß-Berliner Damm 37
12487 Berlin
Geschäftsführer: [redacted]
Ust-ID: [redacted]

Kontakt
Tel: +49 (0)30 611 295 - 0
Fax: +49 (0)30 611 295 - 10
E-Mail: info@taskit.de
Support: support@taskit.de
Web: www.taskit.de

taskit

Abnahmeprotokoll

Projekt: Konzeption & Konfiguration von Webservern für die browserbasierte Erreichbarkeit verschiedener Anwendungen unter Berücksichtigung technischer Maßnahmen zur Datensicherheit

Zeitraum: 01.04.2019 – 29.05.2019

☒ Die Abnahme war erfolgreich.

☐ Die Abnahme war nicht erfolgreich. Folgende Arbeiten sind noch auszuführen:

_____

☐ Sonstiges:

_____

Durchgeführte Tätigkeiten	Abnahme erfolgreich
Evaluation der Software	☒
Installation der Software	☒
Anpassungen der Netzwerkkonfiguration	☒
Konfiguration der Software	☒
Durchführung von Maßnahmen zur Qualitätssicherheit	☒
Dokumentation und Einweisung	☒

taskit GmbH, Groß-Berliner Damm 37
D - 12487 Berlin
Tel. +49 (0)30 611 295 - 0 Fax: - 10
info@taskit.de www.taskit.de

3.6.19 [signature] 3.6.19 [signature]

Datum, Unterschrift Auftraggeber

Datum, Unterschrift Auftragnehmer