

# Realisierungsbericht

<b>Status</b>	Abgeschlossen
<b>Projektname</b>	Network Snapshot Appliance (NSA)
<b>Projektleiter</b>	Joel Meier
<b>Auftraggeber</b>	Stephan Zigerli
<b>Autoren</b>	Micha Simon, Laxushan Yogalingam, Joel Meier, Sven Trachsel
<b>Verteiler</b>	Anna Stettler

## Änderungskontrolle, Prüfung, Genehmigung

Version	Datum	Beschreibung, Bemerkung	Name oder Rolle
0.0.1	10.11.2018	Erste Einträge in die Dokumentation	Micha Simon
0.0.2	20.11.2018	Systemdesign	Micha Simon, Laxushan Yogalingam, Sven Trachsel
0.0.3	27.11.2018	Systemdokumentation, Systemtest, Weiterführung der Projektplanung	Micha Simon, Laxushan Yogalingam, Joel Meier, Sven Trachsel

## Definitionen und Abkürzungen

Begriff / Abkürzung	Bedeutung

## Referenzen

Referenz	Titel, Quelle
[1]	
[2]	
[3]	

**Inhaltsverzeichnis**

1	Zusammenfassung .....	3
2	Technische Detailspezifikation .....	3
2.1	Systemdesign .....	3
2.1.1	Struktur .....	3
2.1.2	Beschreibung der Elemente .....	3
2.2	Schnittstellendefinitionen .....	4
2.3	Sicherheit (ISDS) .....	4
2.4	Anforderungszuordnung .....	5
3	Systemdokumentation .....	5
3.1	Konfigurations-Dokumentation .....	5
3.2	Benutzerhandbuch .....	6
3.2.1	Systemübersicht .....	6
3.2.2	Anwenderfunktionalität .....	6
3.3	Suppothandbuch .....	9
3.3.1	Massnahmen bei Benutzerproblemen .....	9
3.3.2	Massnahmen bei technischen Problemen .....	9
4	Systemtest .....	10
4.1	Testspezifikation .....	10
4.1.1	Kritikalität der Funktionseinheit .....	10
4.1.2	Testanforderungen .....	10
4.1.3	Testverfahren .....	10
4.1.4	Testkriterien .....	11
4.1.5	Testfälle .....	11
4.2	Testprozedur .....	11
4.2.1	Vorbereitung und Voraussetzungen .....	11
4.2.2	Durchführung .....	11
4.2.3	Nachbearbeitung .....	12
4.3	Testprotokoll .....	12
4.3.1	Testobjekt .....	12
4.3.2	Testresultate .....	13
4.3.3	Testauswertung .....	13
5	Weiterführung der Projektplanung .....	13
5.1	Ableich von Planung und tatsächlichem Verlauf der Phase Konzept .....	13
5.2	Aktualisierung der Risikosituation .....	14
5.3	Planung der nächsten Phase .....	15
6	Projektfreigabe .....	16

**Abbildungsverzeichnis**

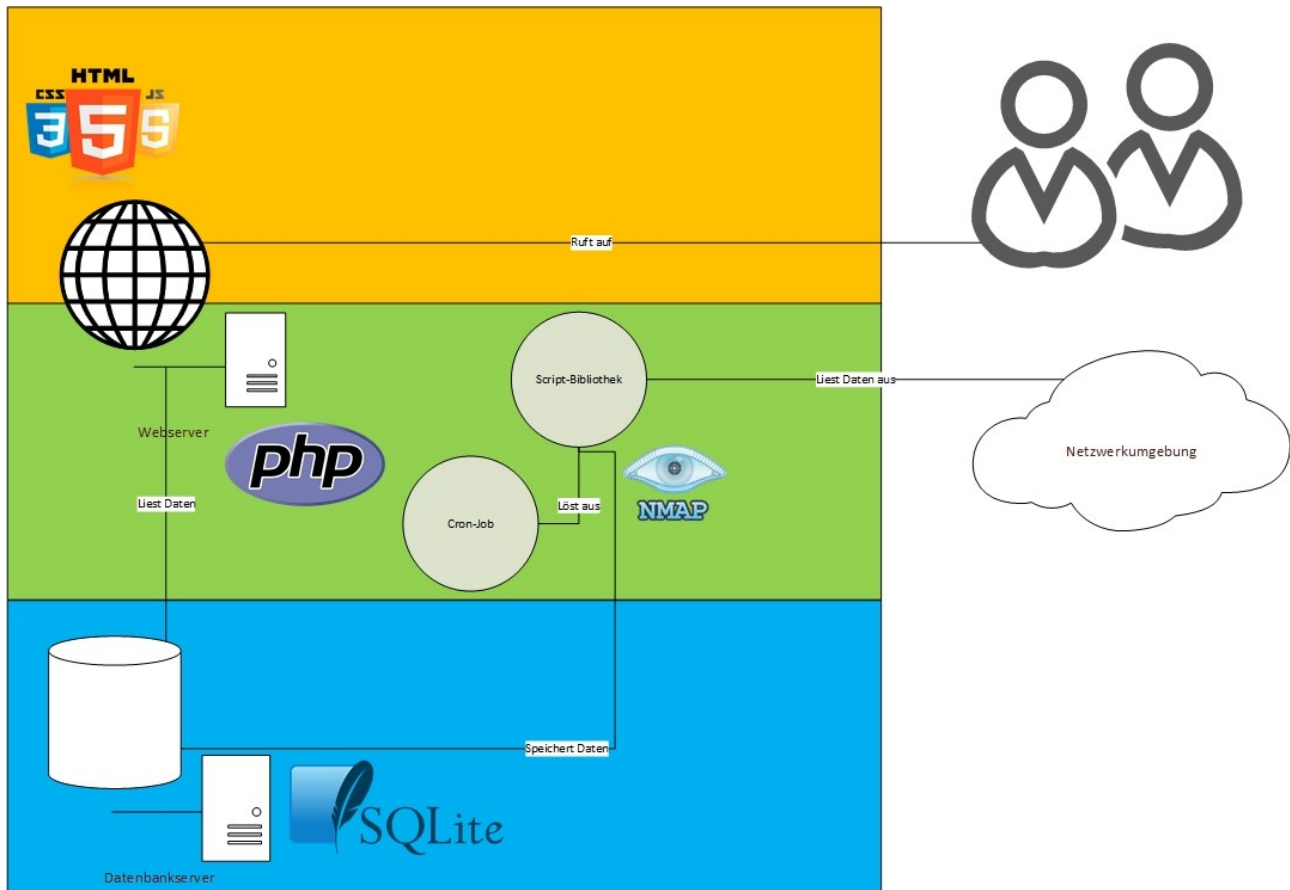
## 1 Zusammenfassung

## 2 Technische Detailspezifikation

### 2.1 Systemdesign

Beim Umsetzen des Projektes wurde die Systemarchitektur gemäss Konzeptbericht berücksichtigt.

#### 2.1.1 Struktur



#### 2.1.2 Beschreibung der Elemente

Nr.	Modul	Teile	Beschreibung
1	Frontend (Webseite)	Darstellung	Die Darstellung wird mittels HTML, CSS und JavaScript dargestellt.
		Logik	Die Logik des Frontend, wird mit PHP gesteuert und von JavaScript Assentiert. Der Logikteil bezieht die Daten aus der S2 und übergibt die dem Darstellungsteil, der sie dann darstellt.
2	Datenbeschaffung	Script-Bibliothek	Die Script-Bibliothek deren Scripts meistens NMAP benutze, wird von Server ausgeführt und fragt die verschiedenen Clients und Server im Netzwerk ab und gibt die der S1 zur Verfügung.
3	Datenspeicherung	Datenbank (Diagramm im Anhang)	Die Datenbank bekommt ihre Daten von der Datenbank Schnittstelle und stellt die Daten über die S2 zur Verfügung.

Auf der Darstellungsebene werden HTML, CSS und PHP verwendet, als Webserver verwenden wir einen Apache Webserver. Zur Datenbeschaffung benutzen wir eine Scriptbibliothek basierend auf Python mit NMAP-Commands. Diese wird von einem Cron-Job gestartet. Als Datenbank wird ein SQLite3 Server benutzt.

## 2.2 Schnittstellendefinitionen

Nr.	Intern / Extern	Schnittstelle	Beschreibung	Daten
S1	Intern	Schnittstelle von Modul Datenbeschaffung zu Datenspeicherung	Die Scripts parsen am Ende die erhaltenen Daten und schreiben sie dann über SQLite3 in die Datenbank speichert.	Resultate der abfragen der Clients im Netzwerk. Die zuvor von der Bibliothek zurechtgeschnitten wurden.
S2	Intern	Schnittstelle von Modul Datenspeicherung zu Frontend	Die Schnittstelle besteht aus dem Programm PHP das Abfragen auf der Datenbank macht und die Daten dann der Webseite übergibt.	Die Daten die in der Datenbank gespeichert sind.
S3	Extern	Schnittstelle von Modul Datenbeschaffung ins Netzwerk	Die Schnittstelle ist die Netzwerkkarte bzw. das Tool NMAP	Angaben über den Angefragten Client oder Server
S4	Extern	Schnittstelle von Darstellung zum User	Die Schnittstelle ist eigentlich die Webseite, die der Benutzer sieht.	Die dargestellte Webseite und Kommandos.

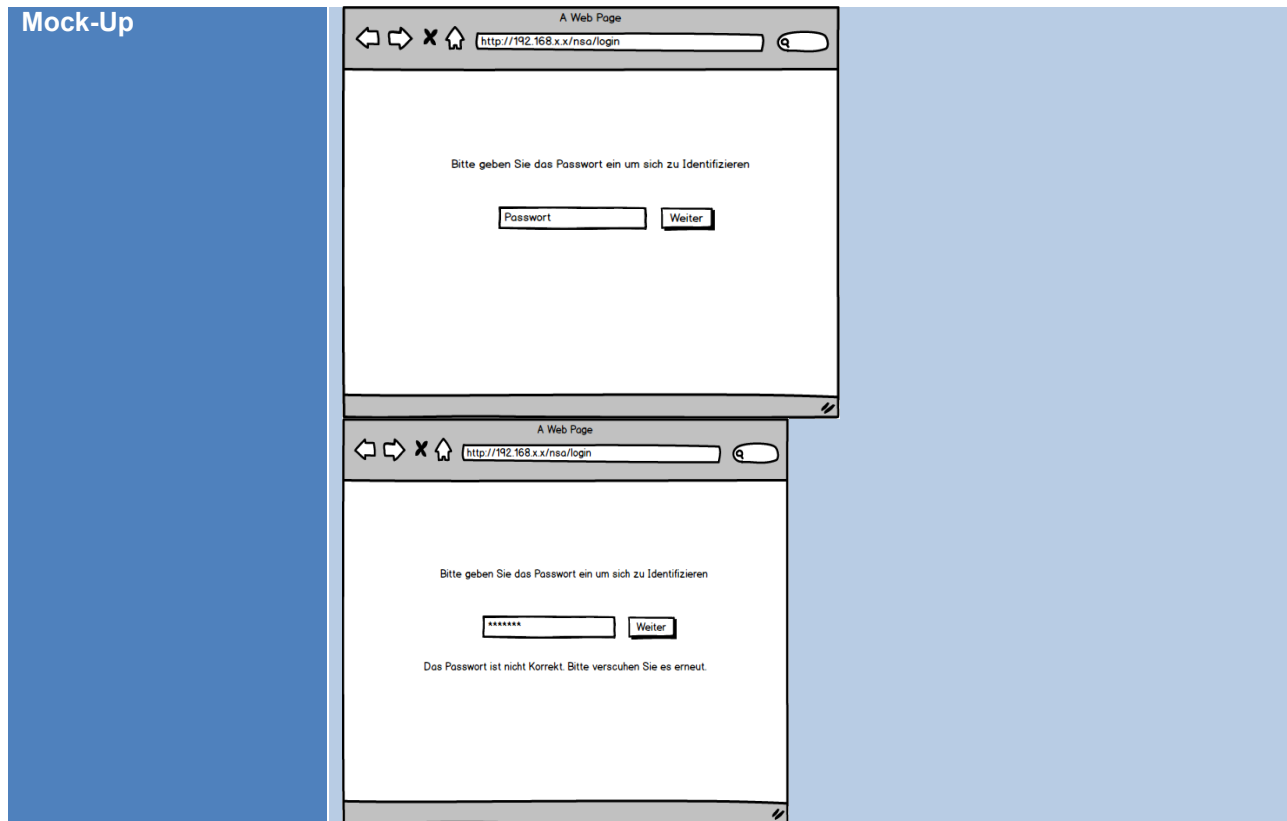
Python Scripts: Liest die verschiedenen Geräte im Netzwerk au und schreibt sie in die Datenbank

Datenbank: Wird verwendet um die vom Script ausgelesenen Daten zu speichern

Website: Das Frontend von dem aus das Script ausgeführt werden kann und die Daten der Datenbank angezeigt wird.

## 2.3 Sicherheit (ISDS)

ID	SAF-01	Titel	Einloggen
User Story	<b>Rolle:</b>		
	<ul style="list-style-type: none"> <li>Benutzer</li> </ul>		
	<b>Ziel:</b>		
	<ul style="list-style-type: none"> <li>Ein Passwort wird benötigt, wenn man auf die Webseite geht.</li> </ul>		
	<b>Grund:</b>		
	<ul style="list-style-type: none"> <li>Um die Netzwerkkdaten vor Unbefugten zu schützen.</li> </ul>		
Akzeptanz	<ul style="list-style-type: none"> <li>Beim ersten aufrufen wird ein Feld mit der Erklärung und einem Passwortfeld sowie einem Knopf angezeigt.</li> <li>Falls der User das Passwort falsch eingibt, wird eine passende Meldung angezeigt.</li> <li>Falls der User das Passwort richtig eingibt, wird für ihn eine Session erstellt und die Hauptseite der Appliance wird angezeigt.</li> </ul>		



## 2.4 Anforderungszuordnung

AFo.-Nr.	Anforderung (Stichwort)	Webseite	Scripts	Datenbank
AF-01	Automatisierte Ausführung		X	
AF-02	Genauigkeit		X	
AF-03	freundliche Benutzeroberfläche	X		
AF-04	benutzerfreundliche Erweiterbarkeit	X	X	X
AF-05	Freeware	X	X	X
SAF-01	Einloggen	X		

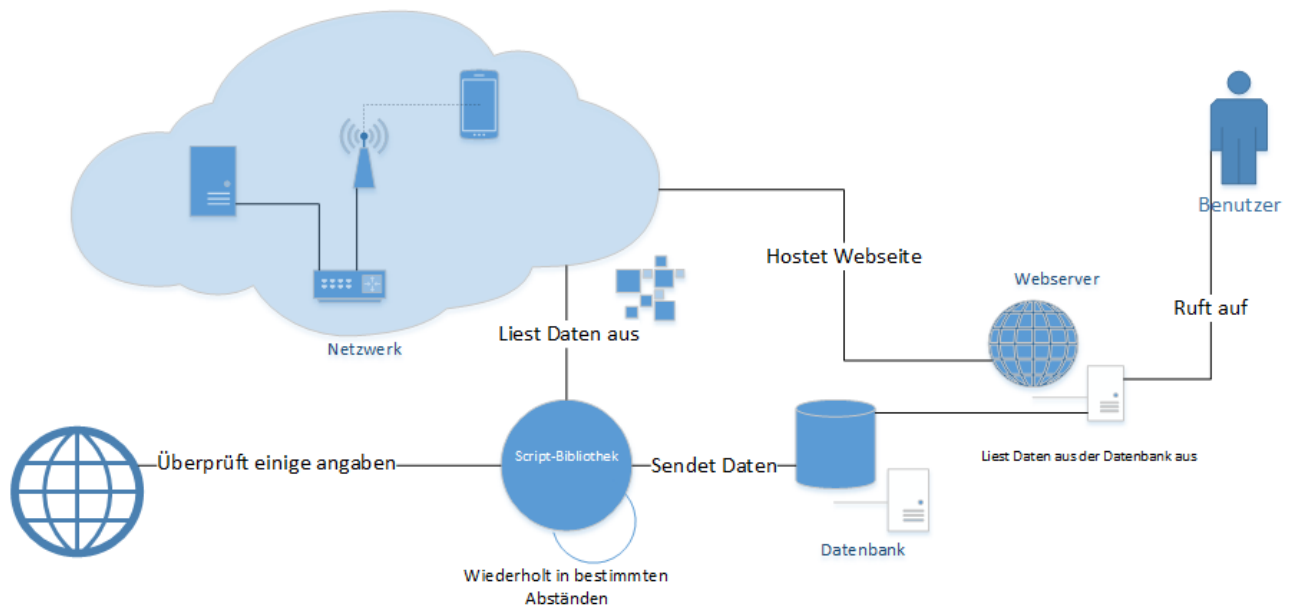
## 3 Systemdokumentation

### 3.1 Konfigurations-Dokumentation

Netzwerk- oder Systemdokumentation mit allen Konfigurationsdaten, geordnet nach Elementen.

## 3.2 Benutzerhandbuch

### 3.2.1 Systemübersicht



## Network Snapshot Appliance

**Beschrieb:** Wenn der Benutzer die Webseite auf dem Webserver aufruft, liest der Webserver die benötigten Daten aus der Datenbank aus.

In bestimmten Abständen, die noch zu definieren sind, führt die Script Bibliothek sich selber aus und liest die Daten aus dem Netzwerk mit abfragen aus. Einige Angaben, zum Beispiel die der DNS anfragen, werden mit Anfragen aus dem Internet verglichen. Die Daten werden dann in die Datenbank eingetragen. Die Webseite liest diese Daten aus und stellt sie dar.

### 3.2.2 Anwenderfunktionalität

#### Installation

```
joco@BudgieX:~$ git clone https://github.com/Jocomol/NSA.git
```

Um die Dateien der Appliance zu kommen muss man zuerst das Repository Clonen, wie das gemacht wird ist im Bild oben ersichtlich.

```
joco@BudgieX:~$ cd NSA/Network_Snapshot_Appliance/System/ && sudo bash install.sh
```

Um die Appliance zu Installieren oder Änderungen anzuwenden, muss man das Installation Script «install.sh» mit Root rechten ausführen.

```
joco@BudgieX:~$ sudo nsacnf
```

Um die Appliance zu konfigurieren muss man das Kommando «nsacnf» mit Rootrechten ausführen. Dies ist nach jeder Neu-Installation erforderlich.

```
joco@BudgieX:~$ sudo nsaconf
Enter the following requirements, if you want to keep the old jsut press enter
Please enter the Interface which IP should be used: enp0s31f6
Please enter how many times the DNS-Discovery Script should be repeated: 3
Please enter how many times the DHCP-Discovery Script should be repeated: 3
Please enter how often the scriptlib should be executed (cronjob) in minutes: 3
joco@BudgieX:~$ s
```

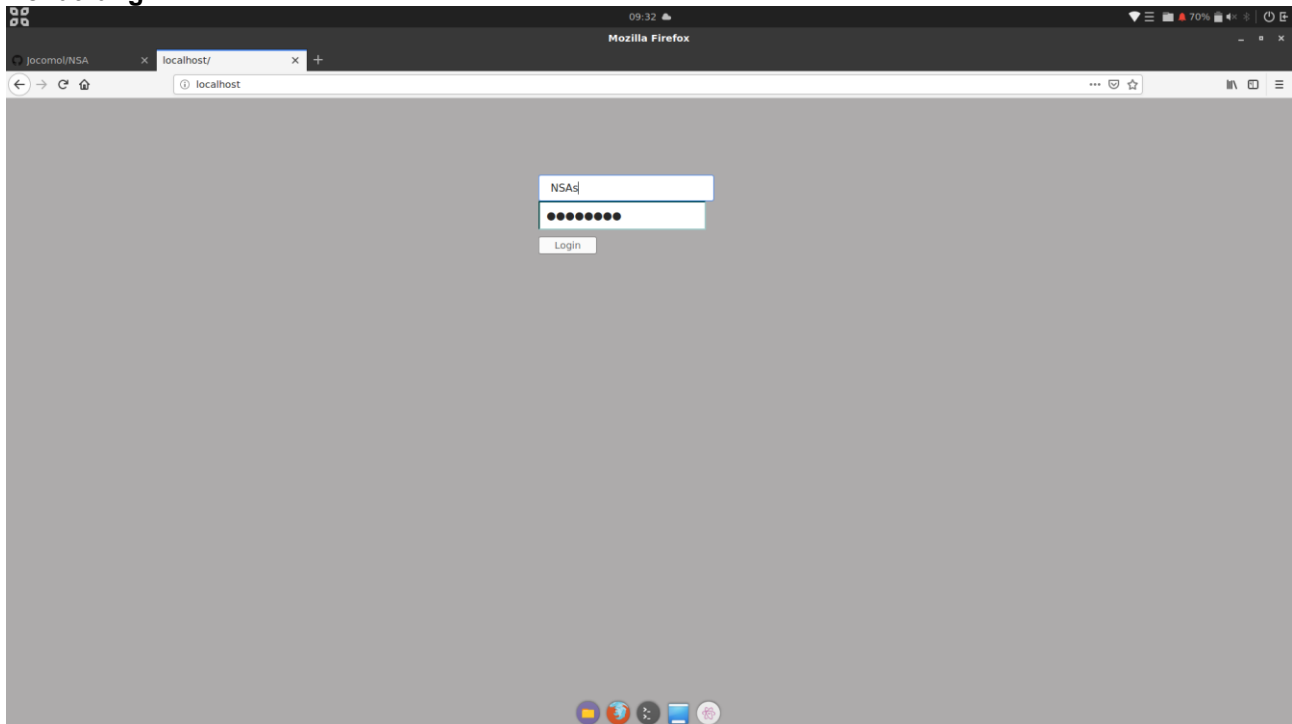
Man muss angeben auf welchen Network Interface die Abfragen ausgeführt werden sollen. Dazu muss man auch angeben wie häufig die Discovery Scripts ausgeführt werden sollten. Dies ist erforderlich, wenn man mehrere DNS oder DHCP Server im Netzwerk hat. Zuletzt muss man angeben wie häufig die Appliance anfragen machen soll (in Minuten).

## Neukonfiguration

```
joco@BudgieX:~$ sudo nsaconf
[sudo] password for joco:
Current configuration:
Interface: enp0s31f6
DNS Repetation: 3
DHCP Repetation: 3
Cronjob Repetation: 3
-----
Enter the following requirements, if you want to keep the old jsut press enter
Please enter the Interface which IP should be used: 
```

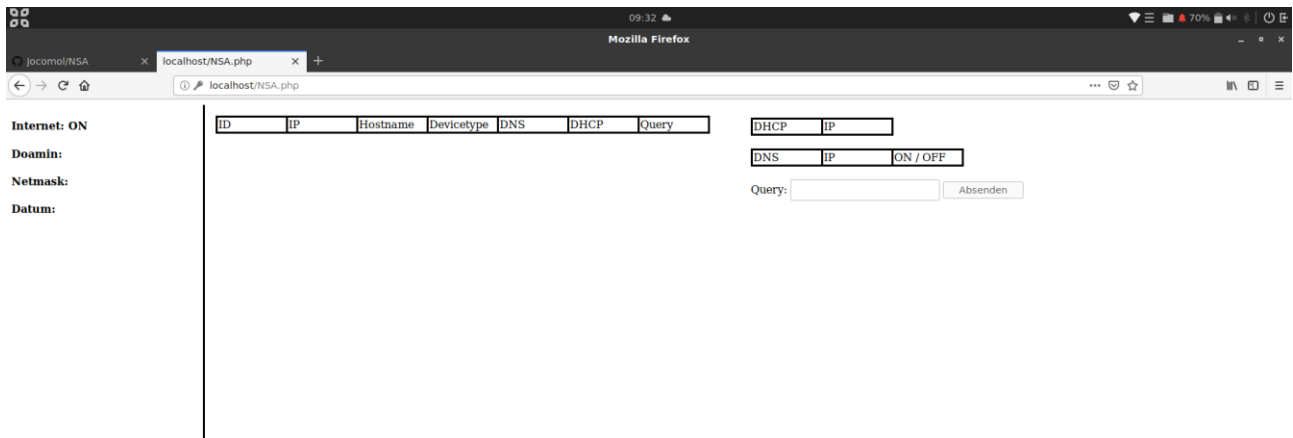
Um die Appliance zu Konfigurieren muss man das Kommando «nsaconf» mit Root rechten ausführen. Falls eine Konfiguration schon vorliegt, wird sie ausgegeben. Man kann nun die Appliance neukonfigurieren, dies funktioniert genau gleich wie bei der Erstkonfiguration, jedoch kann man dieses Mal die Entertaste betätigen falls man die Alte Konfiguration beibehalten will.

## Benutzung



Die Webseite der Appliance ist unter <http://localhost> zu finden. Um sich anzumelden geben Sie folgende Informationen ein:

- Benutzername: NSA
- Passwort: sml12345



Internet: ON

Doamin:

Netmask:

Datum:

ID	IP	Hostname	Devicetype	DNS	DHCP	Query
----	----	----------	------------	-----	------	-------

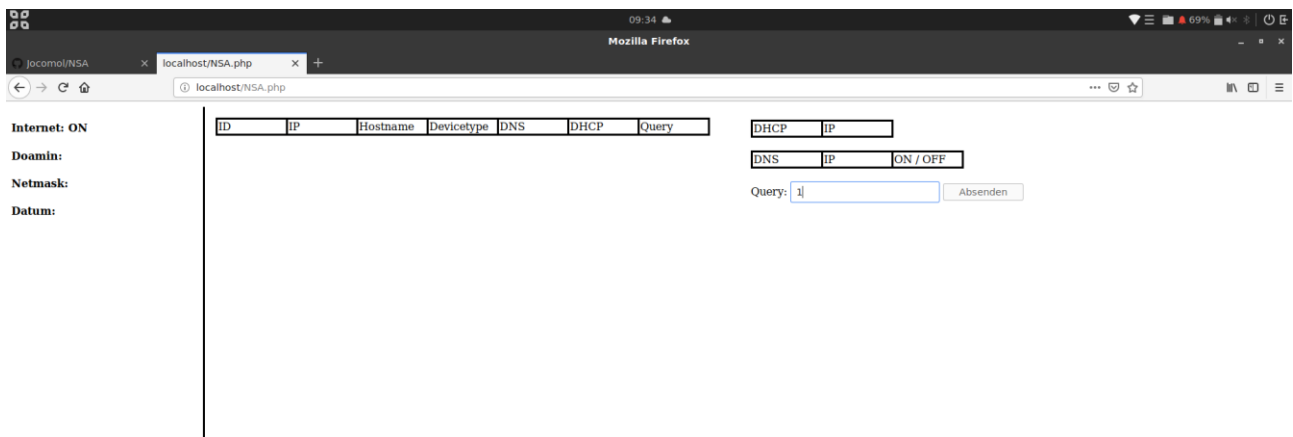
DHCP IP

DNS IP ON / OFF

Query:  Absenden



Dann kommt man auf die bisher leere Webseite.



Internet: ON

Doamin:

Netmask:

Datum:

ID	IP	Hostname	Devicetype	DNS	DHCP	Query
----	----	----------	------------	-----	------	-------

DHCP IP

DNS IP ON / OFF

Query: 1 Absenden



Man kann nach der Ersten erfolgreichen Ausführung im Feld oben rechts die Nummer der Ausführung eingeben, um die Resultate anzuschauen.



**Internet: ON**

**Doamin: None**

**Netmask: 255.255.255.240**

**Datum: 2018-11-27 09:33:25.788651**

ID	IP	Hostname	Devicetype	DNS	DHCP	Query
1	172.20.10.3	BudgieX		1		1
2	8.8.8.8	google-public-dns-a.google.com		2		1
3	8.8.4.4	google-public-dns-b.google.com		3		1
4	172.20.10.1	gateway		1	1	

**DHCP**

DHCP	IP
1	172.20.10.1
4	

**DNS**

DNS	IP	ON / OFF
1	172.20.10.3	ON
2	8.8.8.8	ON
3	8.8.4.4	ON

Query:  Absenden

## Überwachung

Um zu überwachen ob die Appliance ausgeführt wird kann man die Systemlogs anzeigen lassen

```
Netmask: 255.255.255.240
Nov 27 09:33:28 BudgieX dbus-daemon[1691]: [session uid=1000 pid=1691] Successfully activated service 'org.gnome.Screenshot'
Nov 27 09:33:34 BudgieX gnome-screenshot[1547]: Unable to use GNOME Shell's builtin screenshot interface, resorting to fallback X11.
Nov 27 09:33:36 BudgieX org.gnome.Nautilus[1691]: ('Handling file: ', 'file:///home/joco/Pictures/Screenshot%20from%202018-11-27%2009-24-48.png')
Nov 27 09:33:36 BudgieX org.gnome.Nautilus[1691]: ('file scheme: ', 'file')
Nov 27 09:33:36 BudgieX org.gnome.Nautilus[1691]: RuntimeError: object at 0x7f7b583315f0 of type FolderColorMenu is not initialized
Nov 27 09:33:36 BudgieX org.gnome.Nautilus[1691]: ('Handling file: ', 'file:///home/joco/Pictures/Screenshot%20from%202018-11-27%2009-24-48.png')
Nov 27 09:33:36 BudgieX org.gnome.Nautilus[1691]: ('file scheme: ', 'file')
Nov 27 09:33:36 BudgieX org.gnome.Nautilus[1691]: RuntimeError: object at 0x7f7b583315f0 of type FolderColorMenu is not initialized
Nov 27 09:33:01 BudgieX CRON[15464]: (root) CMD: python3 /etc/NSA/script/controller.py
Nov 27 09:33:03 BudgieX system-resolved[587]: message repeated 2 times: [ Server returned error NXDOMAIN, mitigating potential DNS violation DVE-2018-0001, retrying transaction with reduced feature level UDP.]
Nov 27 09:33:13 BudgieX dbus-daemon[1691]: [session uid=1000 pid=1691] Activating service name='org.gnome.Screenshot' requested by ':1.54' (uid=1000 pid=1997 comm='budgie-panel' label='unconfined')
Nov 27 09:33:18 BudgieX gnome-screenshot[15546]: Unable to select area using GNOME Shell's builtin screenshot interface, resorting to fallback X11.
Nov 27 09:33:23 BudgieX gnome-screenshot[15546]: Unable to use GNOME Shell's builtin screenshot interface, resorting to fallback X11.
Nov 27 09:33:23 BudgieX dbus-daemon[761]: [system] Activating via systemd: service name='org.freedesktop.hostname1' unit='dbus-org.freedesktop.hostname1.service' requested by ':1.299' (uid=1000 pid=15546 comm='r/bin/gnome-screenshot' label='unconfined')
Nov 27 09:33:23 BudgieX system[1]: Starting Hostname Service...
Nov 27 09:33:23 BudgieX dbus-daemon[761]: [system] Successfully activated service 'org.freedesktop.hostname1'
Nov 27 09:33:23 BudgieX system[1]: Started Hostname Service.
Nov 27 09:33:24 BudgieX system-resolved[587]: message repeated 2 times: [ Server returned error NXDOMAIN, mitigating potential DNS violation DVE-2018-0001, retrying transaction with reduced feature level UDP.]
Nov 27 09:33:29 BudgieX system-resolved[587]: message repeated 2 times: [ Server returned error NXDOMAIN, mitigating potential DNS violation DVE-2018-0001, retrying transaction with reduced feature level UDP.]
```

Rot: Die Abfrage wird ausgeführt.

Blau: Die Abfrage wurde Beendet.

## 3.3 Supporthandbuch

### 3.3.1 Massnahmen bei Benutzerproblemen

Bei jedem Problem sollte man zuerst folgende Checkliste durchprüfen:

- Stimmt die Linux Version: **Ubuntu 18.10**
- Wurde die Appliance korrekt konfiguriert: **nsaconf**
- Ist die Neuste Version der NSA Installiert: **git pull im Repository und neu Installieren.**
- Hat die Scriptlib ein Problem: **python3 /etc/NSA/script/controller.py** und Output analysieren
- Meldung an das Entwicklerteam über Github: **Issue erstellen mit Fehlerbeschreibung sowie output**

### 3.3.2 Massnahmen bei technischen Problemen

#### Appliance stürzt nach Ausführung ab:

Wenn die Appliance nicht korrekt konfiguriert wurde wird sie nicht gestartet bzw. stürzt nach dem Start ab.

Andernfalls kann es sein das die Dependencies nicht installiert sind.

Massnahme: Neu konfigurieren oder install.sh ausführen.

**Nach Korrekter Ausführung werden keine Resultate angezeigt.**

Wenn das Network Interface falsch bei der Appliance konfiguriert wurde wird nichts abgefragt

Massnahme: nsaconf und richtig konfigurieren.

## 4 Systemtest

### 4.1 Testspezifikation

#### 4.1.1 Kritikalität der Funktionseinheit

Funktionseinheit	Beschreibung	Kritikalität	Begründung
Frontend / Webseite	Benutzerinterface, darüber greift der Benutzer auf unsere Applikation zu	Hoch	Für den Benutzer ist die Applikation nicht mehr erreichbar, wenn diese Funktionseinheit ausfällt
Scriptbibliothek	Datenbeschaffung, neue Daten werden hiermit beschafft	Mittel	Es können keine neuen Daten mehr beschafft werden, Alte werden aber immer noch angezeigt
Datenbank	Datenspeicher, hier liegen alle Daten	Hoch	Beschaffte Daten können nicht mehr gespeichert werden, das Frontend kann nichts mehr auslesen. Die Applikation würde nicht mehr funktionieren.

#### 4.1.2 Testanforderungen

Die Tests sollen unter Normalbedingungen mit normalen Werten durchgeführt werden. Das heisst, dass die Appliance in einem funktionierenden Netzwerk mit einem Router, einem DNS- und einem DHCP-Server durchgeführt werden sollen.

#### 4.1.3 Testverfahren

Test	Vorbereitung	Durchführung	Auswertung
T-001: Cron-Job	<ol style="list-style-type: none"><li>1. Auf Server einloggen</li><li>2. Cron-Job starten und einige Zeit warten, er sollte mehrmalig durchlaufen</li></ol>	<ol style="list-style-type: none"><li>1. <code>sudo nsaconf</code> ausgeführt</li><li>2. <code>cat /etc/crontab */5 * * * * root python3 /etc/NSA/script/controller.py</code></li><li>3. <code>tail -f /var/log/syslog</code> nach Cron-Ausführungen geschaut</li></ol>	Der Cron-Job sollte regelmässig gelaufen sein. Das wird mit den vorher aufgeführten Befehlen sichtbar.
T-002: Website-Login	Webseite aufrufen	<ol style="list-style-type: none"><li>1. Falsches Passwort eingeben</li><li>2. Richtiges Passwort eingeben</li></ol>	Beim falschen Passwort sollte man nicht weitergeleitet werden, wenn das richtige eingegeben wird schon.
T-003: Script	Webseite aufrufen	<ol style="list-style-type: none"><li>1. Scripts ausführen</li></ol>	Eine Fehlermeldung dies bestätigt das die zwei Scripts

		<ol style="list-style-type: none"><li>Bestimmen in welchem Intervall der Zweite Durchlauf gestartet wird</li><li>Vor dem zweiten Durchlauf eine Veränderung am Netzwerk durchführen</li></ol>	zusammen verglichen werden.
--	--	---	-----------------------------

#### 4.1.4 Testkriterien

Abdeckungsgrad:

Da die Appliance schlussendlich Automatisch und selbständig laufen muss, wurde so breit wie möglich getestet um Fehler zu vermeiden.

Checklisten:

Die Checklisten sind im Kapitel 4.3 zu finden.

Ende-Kriterien:

Ebenfalls unter Kapitel 4.3 zu finden.

#### 4.1.5 Testfälle

- Zu testen sind: Ob das Script das Netzwerk scannt, Ob das Script die Daten in die Datenbank schreibt, Ob die Website die Daten auslesen kann.
- Um die Website zu testen muss eine DB mit Testdaten erstellt werden. Das Script zu testen sollte es fertig sein (man kann auch nur die einzelnen kleinen Funktionen des Scripts testen, wenn diese bereits geschrieben wurden) und es sollten Geräte im Netzwerk sein.
- Auf der Website kann gibt es einen Button der das Script ausführt, wenn nach dem betätigen dieses Buttons Clients aufgeführt werden ist der Test erfolgreich
- Clients (ID, IP, Hostname, Devicetype, DNS, DHCP, Query), DHCP (Name, IP) und DNS(Name, IP, ON/OFF)

## 4.2 Testprozedur

### 4.2.1 Vorbereitung und Voraussetzungen

Die Grundvoraussetzung ist ein Linux Betriebssystem

#### **Script**

- Die einzelnen Scriptteile müssen fertig sein.
- Es müssen Geräte mit dem Netzwerk verbunden sein.
- Eine DB sollte erstellt sein um zu testen ob das Script hineinschreibt.

#### **Website**

- Eine DB mit Testdaten sollte erstellt sein um zu testen ob das PHP die Daten ausliest.

#### **Appliance**

- Und die Appliance zu testen sollte das Script und die Website fertig sein.

Konfiguration:

### 4.2.2 Durchführung

Beim Script müssen die Verschiedenen Scriptteile ausgeführt werden

Bei der Website muss Die Website geöffnet werden und der Scanbutton gedrückt werden und schon sollten die Geräte angezeigt werden

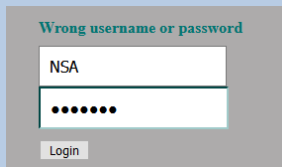
#### 4.2.3 Nachbearbeitung

Die Testdaten wurden automatisch beim ausführen in der Datenbank gespeichert.

### 4.3 Testprotokoll

#### 4.3.1 Testobjekt

ID / Bezeichnung	T-001 Cron-Job
Beschreibung	Überprüfung des Cron-Jobs in den Cron-Einstellungen
Testvoraussetzung	Der Cron-Job sollte verfügbar sein.
Testschritte	<ol style="list-style-type: none"><li>1. Sudo nsacnf ausgeführt</li><li>2. cat /etc/crontab */5 * * * * root python3 /etc/NSA/script/controller.py</li><li>3. tail -f /var/log/syslog nach Cron-Ausführungen geschaut</li></ol>
Erwartetes Ergebnis	Die Scripts werden regelmässig ausgeführt
Tester	Joel Meier
Datum Testdurchführung	27.11.2018

ID / Bezeichnung	T-002 Webseite Login
Beschreibung	Kontrolle des Logins von der Webseite
Testvoraussetzung	Webseite sollte erstellt sein mit einem Login Formular
Testschritte	<ol style="list-style-type: none"><li>3. Webseite aufrufen</li><li>4. Falsches Passwort eingeben</li></ol>  <ol style="list-style-type: none"><li>5. Kontrollieren ob es funktioniert hat</li><li>6. Richtiges Passwort eingeben</li></ol>
Erwartetes Ergebnis	Login nur mit dem korrektem Passwort
Tester	Laxushan Yogalingam
Datum Testdurchführung	20.11.2018

ID / Bezeichnung	T-003	Script
Beschreibung	Script sollte die zwei Resultate vergleichen und einen Fehler melden.	
Testvoraussetzung	Netzwerk	
Testschritte	4. Scripts ausführen 5. Bestimmen in welchem Intervall der Zweite Durchlauf gestartet wird 6. Vor dem zweiten Durchlauf eine Veränderung am Netzwerk durchführen	
Erwartetes Ergebnis	Eine Fehlermeldung dies bestätigt das die zwei Scripts zusammen verglichen werden.	
Tester	Joel Meier	
Datum Testdurchführung	20.11.2018	

Test Ergebnis zusammengefasst:

ID	Erfolgreich	Gescheitert
T-001		
T-002		
T-003		

#### 4.3.2 Testresultate

Die Testresultate sind bei uns Positiv ausgegangen. Wir haben schon in der Entwicklung unser System schon angefangen mit dem Testing. Mit der Vorgehensweise konnten wir am Schluss die Tests erfolgreich Testen.

#### 4.3.3 Testauswertung

Die Tests waren bei uns alle erfolgreich. Die meisten Tests waren für die Scripts ob diese wirklich das machen was von uns erwartet wird. Dies haben aus dem Grund gemacht, wenn die Scripts nicht richtig würde funktionieren oder falsche Werte ausgibt dann wäre unser Komplette Appliance nichts wert, weil es falsche Informationen über das Netzwerk ausgibt

## 5 Weiterführung der Projektplanung

### 5.1 Abgleich von Planung und tatsächlichem Verlauf der Phase Konzept

Aufgabe Nr.	Aufgabenname	Dauer	Anfangsdatum	Abschlussdatum
1	Vorarbeiten	6	14.08.2018 08:00	21.08.2018 16:00
2	Initialisierung	16	21.08.2018 08:00	11.09.2018 16:00
3	Konzept inkl. Ferien	31	11.09.2018 08:00	23.10.2018 16:00

4	Datenbank Realisiert	0	30.10.2018 12:00	30.10.2018 16:00
5	Scripts Realisiert	0	13.11.2018 12:00	23.11.2018 16:00
6	Realisierung	21	23.10.2018 08:00	27.11.2018 16:00
7	Einführung	11	20.11.2018 08:00	04.12.2018 16:00
8	Projektabschluss	11	04.12.2018 08:00	18.12.2018 16:00
9	Präsentationen	6	08.01.2019 08:00	15.01.2019 16:00

Eine Grafische Darstellung ist im Anhang zu finden.

## 5.2 Aktualisierung der Risikosituation

Grün = Niedriges Risiko

Gelb = Mittleres Risiko

Rot = Hohes Risiko

Eintrittswahrscheinlichkeit	W5		1	8	
	W4				
	W3		5	4 & 3	
	W2		9	6 & 7	2
	W1				
		S1	S2	S3	S4
	Schadensausmass				

Abbildung 1 Risikograph

Risiken Nr.	
1	Krankheit, Unfall
2	Datenverlust
3	Abgabetermin verpassen
4	Allgemeine Netzwerkprobleme
5	Fehler in der Konzeptphase
6	Hardware Ausfall (gibbix)
7	Script Fehler (Patch)
8	Ausfall von Auftraggeber
9	Ausfall von Gibb Netzwerk

## **5.3 Planung der nächsten Phase**

In der nächsten Phase, der Einführung, werden wir den Schlussbericht schreiben. Dabei gehen wir das Dokument durch und werden uns die Zeit laufend einplanen.

## **6 Projektfreigabe**

Hiermit bestätigt der Auftraggeber die Freigabe des Projekts:

---

Der Auftraggeber

(Ort, Datum, Unterschrift)

---

Der Projektleiter

(Ort, Datum, Unterschrift)



