HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

COMPASS SECURITY

Studienarbeit

Abteilung Informatik
Hochschule für Technik Rapperswil

Herbstsemester 2013

Autor(en): Dominique Sorg, Benjamin Kehl
Betreuer: Walter Sprenger

Compass Security AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

T +41 55 214 41 60
F +41 55 214 41 61
team@csnc.ch
www.csnc.ch

Aufgabenstellung

ICS Threat Map

Benjamin Kehl / Dominique Sorg

11. September 2013

Name des Dokuments:	Aufgabenstellung_ICS_ThreatMap_v0.9.docx
Version:	v1.0
Projektnummer:	30015
Autor(en):	Walter Sprenger, Compass Security AG
Referenzen:	-
Lieferungsdatum:	11. September 2013
Klassifikation:	PUBLIC

Aufgabenstellung – ICS Threat Map – v1.0

Inhaltsverzeichnis

Compass Security AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

T +41 55 214 41 60
F +41 55 214 41 61
team@csnc.ch
www.csnc.ch

1 Einführung

Ein Ziel der Cyber-Security-Strategie des Bundes ist es, die kritischen Infrastrukturen der Schweiz gegen Hacker-Angriffe besser zu schützen. Zu den kritischen Infrastrukturen gehören zum Beispiel Energie- und Wasserversorgung, Transport- und Gesundheitswesen. Viele dieser Anlagen werden mittlerweile über das Internet überwacht und bedient.

Diverse Untersuchungen und Medienberichte haben aufgezeigt, dass einige kritische Anlagen ungeschützt im Internet erreichbar sind und so von einem böswilligen Angreifer manipuliert werden könnten.

2 Aufgabe

In dieser Arbeit soll eine Web-Applikation entwickelt werden, welche mindestens folgende Funktionen zur Verfügung stellt:

- Suchen von ungeschützten ICS in der Schweiz und aufzeigen von neuen ICS
- ICS in einer Datenbank speichern und mit zusätzlichen Informationen anreichern
- Identifizieren des Anlagen-Standortes und des Anlagen-Betreibers
- Klassifizieren der Anlage (Typ der Anlage, Bedrohungsausmass, Vernetzung)
- Integration eines Trouble-Ticket-Systems, mit welchem die Aktionen bezüglich eines ICS dokumentiert werden
- Eine Landkarte, auf welcher die ICS und die damit verbundenen Risiken visualisiert werden

Das Ziel der Web-Applikation besteht darin, die aktuelle und historische Gefahr im Bereich ICS der Schweiz zu visualisieren. Zudem soll der Nutzer der Web-Applikation (z.B. ICS-CERT, MELANI, Bund, Security Firma) in der Lage sein, den Betreibern der ICS die Gefahr aufzuzeigen und dokumentieren, welche Aktionen er getroffen hat, dass die Bedrohung für die Schweiz minimiert werden kann.

Compass Security AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

T +41 55 214 41 60
F +41 55 214 41 61
team@csnc.ch
www.csnc.ch

3 Vorgehen

Im Rahmen der allgemeinen Richtlinien zur Durchführung von Studien-&Bachelorarbeiten gemäss eigenem Projektmanagementplan. Dieser Projektmanagementplan ist als Erstes von den Studenten zu erstellen und enthält insbesondere:

- Die Beschreibung des dem Projektcharakter angepassten Vorgehensmodells.
- Eine erste Aufteilung der Aufgabe in gemeinsam und einzeln zu bearbeitende Teile unter Berücksichtigung der vorgegebenen Teilaspekte. Die genaue Aufteilung muss spätestens nach der Anforderungsanalyse erfolgen.
- Den Projektplan (Zeitplan) und die Meilensteine.

4 Randbedingungen

- Wo möglich sollten Open Source Produkte eingesetzt werden
- Die entwickelten Serverteile sollten Plattform-Unabhängig betrieben werden können

5 Infrastruktur

Die Arbeiten werden auf den zugeteilten Rechnern an der HSR mit der Standardinstallation durchgeführt. Zusätzlich benötigte Software oder Hardware wird bei Bedarf und nach Rücksprache mit Compass Security zur Verfügung gestellt.

6 Erwartete Resultate

6.1 Was muss abgegeben werden

In elektronischer Form:

- Installationskit (alle Dateien für eine Installation und Installationsanweisung)
- kompletter Source Code
- komplettes Klassen Modell
- alle Dokumente
- alle Protokolle

Compass Security AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

T +41 55 214 41 60
F +41 55 214 41 61
team@csnc.ch
www.csnc.ch

Auf Papier:

Gemäss der Anleitung der HSR:
https://www.hsr.ch/Allgemeine-Infos-Diplom-Bach.4418.0.html

Es muss aus den abgegebenen Dokumenten klar hervorgehen, wer für welchen Teil der Arbeit und der Dokumentation verantwortlich war.

7 Termine

7.1 Start/Ende

Gemäss Vorgabe der HSR:
https://www.hsr.ch/Termine-Bachelor-und-Studiena.5142.0.html

7.2 Zeitplan und Meilensteine

Zeitplan und Meilensteine für das Projekt sind von den Studenten selber zu erarbeiten und zusammen mit dem Projektmanagementplan abzuliefern. Die Meilensteine sind bindend. Der erste Meilenstein ist vorgegeben. Mit den Betreuern werden regelmässige Sitzungen zur Fortschrittskontrolle durchgeführt.

Abgabetermin Projektmanagementplan mit Zeitplan: 7. Oktober 2013

Compass Security AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

T +41 55 214 41 60
F +41 55 214 41 61
team@csnc.ch
www.csnc.ch

8 Betreuung

Die Arbeiten werden von durch Walter Sprenger betreut.

8.1 Kontakt

Walter Sprenger
Compass Security AG
Werkstrasse 20
8645 Jona
e-mail: walter.sprenger@csnc.ch

9 Referenzen

- Vorgaben HSR
  https://www.hsr.ch/Allgemeine-Infos-Diplom-Bach.4418.0.html
- Links / Demos
  - http://www.scadacs.org/iram.html

10 Unterschriften

Rapperswil, 16. September 2013

Walter Sprenger Benjamin Kehl Dominique Sorg

Compass Security AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

T +41 55 214 41 60
F +41 55 214 41 61
team@csnc.ch
www.csnc.ch

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

Erklärung

Wir erklären hiermit,

- dass ich die vorliegende Arbeit selber und ohne fremde Hilfe durchgeführt habe, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit dem Betreuer schriftlich vereinbart wurde,
- dass ich sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben habe.
- das ich keine durch Copyright geschützten Materialien (z.B. Bilder) in dieser Arbeit in unerlaubter Weise genutzt habe.

Rapperswil, 17.12.2013

Benjamin Kehl

Dominique Sorg

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

Kurzfassung der Studienarbeit

Abteilung	**Informatik**
Name[n] der Studierenden	**Benjamin Kehl** **Dominique Sorg**
Studienjahr	**HS 2013**
Titel der Studienarbeit	**ICS ThreatMap**
Examinatorin / Examinator	**Walter Sprenger**
Themengebiet	**Internet-Technologien und Sicherheit**
Institut	**Institut für Software**

Diese Studienarbeit beschäftigt sich mit der Entwicklung einer Webapplikation namens ICS ThreatMap, welche ICS/SCADA Kontrollsysteme von kritischen Infrastrukturen in der Schweiz erfasst und auf einer Karte visualisiert. Zu den kritischen Infrastrukturen gehören zum Beispiel Energie- und Wasserversorgungen, Transport sowie Gesundheitswesen. Eine Bedrohung für die Schweiz entsteht dann, wenn die Kontrollsysteme ungeschützt ans Internet angeschlossen werden.

Da ICS ThreatMap vertrauliche Informationen verwaltet, ist es nur einem engen Nutzerkreis wie MELANI, ICS-CERT oder Sicherheitsfirmen vorbehalten. Die Nutzer sind in der Lage nach ICS/SCADA zu suchen und diese mit weiteren Informationen anzureichern. Mit einem selbstentwickelten Trouble Ticket System werden die Bedrohungen abgearbeitet. Die betroffenen Betreiber werden kontaktiert und über die Verletzlichkeit Ihrer Systeme aufgeklärt. Die Fortschritte werden im jeweiligen Ticket dokumentiert und die getätigten Massnahmen laufend überprüft. Ziel ist es, die betroffenen Betreiber auf die Gefahren aufmerksam zu machen und sie zu bewegen, die Kontrollsysteme besser zu schützen oder sie aus dem Netz zu nehmen.

Um gezielt nach ungeschützten Kontrollsysteme zu suchen, wurden in der ersten Phase geeignete Suchfilter für Shodan oder Google gesammelt und erarbeitet. Mit einem täglich laufenden Programm werden die Suchabfragen mittels eines CronJobs ausgeführt und die als JSON zurückgelieferten Daten in die MySQL-Datenbank gespeichert. Die in der zweiten Phase entwickelte Webapplikation verarbeitet die Daten weiter. ICS ThreatMap wurde mit Zend Framework 2 realisiert. Für die Darstellung der ICS auf einer Karte wurde GoogleMaps und für Statistiken jqPlot verwendet.

Im Ergebnis wird deutlich, dass die Webapplikation laufend mit neuen Suchfiltern aktuell gehalten werden muss, da ständig neue Produkte für ICS/SCADA auf den Markt kommen. Diese Produkte weisen oft Muster im Protokoll-Header auf, nach denen gezielt gesucht werden kann. Auf viele dieser Kontrollsysteme lässt sich ohne Authentisierung oder nur mit Standard Login zugreifen. Dabei handelt es sich bei den Betreibern meist um keine IT-Fachleute, welche primär froh sind, wenn ihre Anlagen problemlos laufen.

Ziel der nationalen SKI-Strategie (Schutz Kritischer Infrastrukturen) des Bundes ist es, die kritischen Infrastrukturen der Schweiz gegen Cyber-Angriffe zu schützen. Der Betrieb von ICS ThreatMap wird zeigen, ob die Sicherheit der Kontrollsysteme in der Schweiz mittels der Webapplikation ICS ThreatMap verbessert werden kann.

ICS ThreadMap - v1.0

Management Summary

Benjamin Kehl
Dominique Sorg

Änderungsgeschichte

Datum	Version	Änderungen	Autor
10.12.2013	0.1	Erstellung Einleitung	Dominique Sorg
16.12.2013	0.2	Überarbeitung	Benjamin Kehl
18.12.2013	0.3	Korrekturen	Dominique Sorg

Inhalt

1. Ausgangslage

Die Schweiz verfügt über kritische Infrastrukturen, die essentielle Güter wie Verkehr, Kommunikation oder Energie sicherstellen. Grossflächige Ausfälle von kritischen Infrastrukturen haben schwerwiegende Folgen für die Bevölkerung und Wirtschaft. Es entsteht ein sogenannter Dominoeffekt. So könnte ein grossflächiger Stromausfall Auswirkungen auf die Wasserversorgung, Telekommunikation und den Schienenverkehr haben. Obwohl die Schweiz in einzelnen Bereichen über ein hohes Schutzniveau verfügt und von einem stabilen Umfeld profitiert, sind schwerwiegende Ausfälle bisher selten und wenn nur von kurzer Dauer gewesen. Nichts desto trotz hat sich das Risiko durch die Zunahme von Cyber-Angriffen, Naturkatastrophen oder der Veraltung von technischen Systemen erhöht. Um die Widerstandfähigkeit der kritischen Infrastrukturen in der Schweiz zu verstärken, hat der Bundesrat am 27. Juni 2012 die *Nationale Strategie zum Schutz kritischer Infrastrukturen* (kurz SKI-Strategie) verabschiedet.

Ein Ziel der SKI-Strategie ist es unter anderem, die industrielle Kontrollsysteme (ICS[1])/SCADA[2] von kritischen Infrastrukturen der Schweiz gegen Cyber-Angriffe besser zu schützen. Diverse Untersuchungen und Medienberichte haben aufgezeigt, dass einige Kontrollsysteme ungeschützt im Internet erreichbar sind und von einem böswilligen Angreifer manipuliert werden könnten. Der Artikel der Sonntags-Zeitung am 01. Dezember 2013 zeigt einen aktuellen Fall in der Schweiz[3]. Dabei lag das Schliesssystem des ST. Jakob-Parks in Basel während eines Jahres für Hacker offen. Durch das massive Medieninteresse werden die Kontrollsysteme zu beliebten Angriffszielen. Unter anderem wurden erhebliche Sicherheitslücken in Systemen von unterschiedlichen Herstellern gefunden. Geschweige denn, dass viele dieser Systeme offen, ungeschützt und öffentlich mit dem Internet verbunden sind.

Die Compass Security AG möchte deshalb die ICS/SCADA innerhalb der Schweiz aufsuchen, um die Sicherheit der Gesellschaft zu optimieren. Um das zu erreichen, soll durch eine Studienarbeit von der HSR eine Webapplikation namens ICS ThreatMap entwickelt werden.
Mit dieser Studienarbeit haben wir (Dominique Sorg und Benjamin Kehl) eine funktionierende Webapplikation erstellt, in der die Kontrollsysteme auf einer Landkarte visualisiert werden. Dort kann man ihre Gefahr und die damit verbundenen Risiken überprüfen. Anschliessend können die Nutzer aus einem Zielpublikum wie der Bund, ICS-Cern oder die Sicherheitsfirmen die Suchresultate bearbeiten und bewerten. Mit einem selbstentwickelten Trouble Ticket System werden die Bedrohungen verringert. Dabei werden die betroffenen Firmen kontaktiert und über die Sicherheitslücken ihrer Kontrollsysteme aufgeklärt. Die Fortschritte werden im jeweiligen Ticket dokumentiert und die getätigten Massnahmen laufend überprüft.

Das Ziel von ICS ThreatMap ist die aktuelle und historische Gefahr im Bereich kritischen Infrastrukturen der Schweiz zu visualisieren und die Betreiber auf ihre gefährdeten Anlagen aufmerksam zu machen, damit die Bedrohung für die Schweiz minimiert werden kann.

[1] ICS: Industrial Control System
[2] SCADA: Supervisory Control and Data Acquisition
[3] Imbach, F., & Alexandre, H.(01.12.2013). SonntagsZeitung. Von http://www.sonntagszeitung.ch/fokus/artikel-detailseite/?newsid=268454 (12.12.2013)

2. Vorgehen & Technologien

2.1 Vorgehen

Aus den Bedürfnissen von Compass Security AG leiteten wir entsprechend priorisierte Arbeitspakete ab, die mit Priorität 1 oder 2 bezeichnet wurden. Als Vorgehensmodell wurde Rational Unified Process (Abk.: RUP) angewendet.

Für die kritischen Arbeitspakete wurden bereits früh Prototypen erstellt, da diese die Basis der Webapplikation darstellen.

Mittels Usability Tests wurde überprüft, ob die Bedienung von ICS ThreatMap einfach und intuitiv ist. Gegen Ende der Arbeit haben wir diese Tests durchgeführt, um Aufschluss über fehlende Funktionen und unlogische Abläufe innerhalb der Applikation zu bekommen.

2.2 Technologien

Für die Verwendung von Technologien legten wir sehr viel Wert auf Open-Source-Produkte. Dazu zählt das Betriebssystem Ubuntu, Eclipse als Entwicklungsumgebung und die Entwicklung selbst mit einem objektorientierten PHP-Framework namens Zend Framework 2.

3. Erreichte Ziele

ICS ThreatMap war für jedes Teammitglied das erste grosse Webprojekt mit einer sehr hohen Komplexität. Uns ist es gelungen, einen funktionstüchtigen Prototyp zu entwickeln, der viele der gesetzten Ziele mit erster Priorität erreicht hat. Der Zeitaufwand für Optimierungen darf nicht überschätzt werden. Dies ist aber sehr wichtig für die Entwicklung einer qualitativ hochwertigen Webapplikation. Durch die Applikation können die gefundenen ICS mit individuellen Suchfiltern bei externen Quellen wie Shodan gesammelt, bearbeitet und in der Datenbank abgelegt werden. Die Web Applikation ist einerseits eine Wissensbasis von gesammelten ICS und andererseits eine Suchmaschine, um nach solchen Systemen in der Schweiz zu suchen. ICS können hinzugefügt, bearbeitet, entfernt oder klassifiziert werden. Des Weiteren hilft ein selbstentwickeltes Trouble Ticket System die Abarbeitung von kritischen Anlagen.

ICS ThreatMap bildet eine gute ausbaubare Grundlage für die Weiterentwicklung. Zu den gegebenen Anforderungen wurden folgende Ergebnisse erzielt:

Datenbeschaffung
Die Datenbeschaffung konnte mit Shodan realisiert werden. Durch Suchfilter von Shodan können neue ICS/SCADA gefunden werden. Neue ICS werden durch ein automatisches Skript zu der Datenbank hinzugefügt und bestehende ICS werden aktualisiert. Shodan liefert bereits umfangsreiche Daten wie Geodaten und Adressdaten, auch wenn diese nur mit Vorsicht zu geniessen sind.

Die Implementierung mit Google konnte leider aus Zeitgründen nicht realisiert werden. Der Prototyp von Google zeigt, dass diverse externe Dienste wie Geolocation genutzt werden müssen, um die Daten ähnlich anzureichen.

Registration
Neue Benutzer können sich auf ICS ThreatMap erfolgreich registrieren. Die Informationen können aus Sicherheitsgründen nicht an beliebige Nutzer veröffentlicht werden, deshalb wird der Account vorerst deaktiviert. Der Administrator kann dem Benutzer den Zugriff berechtigen.

Benutzerverwaltung
Jeder registrierte Benutzer kann sein eigenes Profil verwalten. Er kann dies mit zusätzlichen Daten erweitern oder sein Passwort ändern.

Der Administrator ist in der Lage jeden Benutzer zu bearbeiten. Sein Passwort zurückzusetzen oder zu löschen.

Benutzer aktivieren & deaktivieren
Der Administrator kann jeden beliebigen Benutzer aktivieren oder deaktivieren.

Login
Um Zugang zu der Applikation und zu den ICS Daten zu erhalten, müssen sich die Benutzer anmelden.

Filterverwaltung
Die Filterverwaltung ermöglicht den Benutzer, neue Filter zu erstellen, bestehende zu bearbeiten oder bestehende zu entfernen. Die Datenbeschaffung holt alle Filter aus der Filterverwaltung um die gewünschten ICS Daten bei Shodan zu holen.

ICS hinzufügen, bearbeiten & löschen
Selbst gefundene ICS können manuell hinzugefügt werden. Bestehende ICS, unabhängig ob sie manuell oder durch eine Engine hinzugefügt worden sind, können bearbeitet oder entfernt werden. Beim Entfernen wird die ICS in der Datenbank archiviert.

ICS suchen & ansehen
Eine Suchfunktion ermöglicht es nach ICS zu suchen. Es kann nach der IP Adresse, Port, Datum, Organisation, Titel, Produkt, Kategorie oder nach der Bedrohung gesucht werden. Gefundene Suchergebnisse verweisen auf eine Detailansicht. In der Detailansicht werden die ICS Daten angezeigt.

ICS Filterauflistung
Zu einem ICS System werden die Suchfilter angezeigt, der die ICS gefunden hat.

ICS History
Die History zeigt alle Änderungen der Benutzer mit Datums- und Zeitangaben an.

ICS Klassifizieren und auf der Karte anzeigen
ICS können nach deren Bedrohung klassifiziert werden. Die Systeme können nach unterschiedlichen Merkmalen untersucht werden (z.B. Schweregrad, Hersteller, Systemtyp). Diese Merkmale, wurden in verschiedene ICS Kategorien unterteilt. Der Schwergrad auf der Map wurde durch Farben hervorgehoben, damit die Schwachstellen der Systeme möglichst schnell ersichtlich werden.

Trouble Ticket System
Ein Trouble Ticket System ermöglicht es, Tickets zu einem ICS zu erstellen. Die Tickets können hergestellt, bearbeitet oder entfernt werden. Beim Entfernen von Tickets werden diese in der Datenbank archiviert. Eine History protokolliert die Tätigkeiten am Ticket (wiederum mit Datums- und Zeitangaben).

Emailbenachrichtigung
ICS ThreatMap ist in der Lage Emailbenachrichtigungen zu verschicken. Wenn sich ein Benutzer registriert, wird der Administrator per Mail informiert. Wird der Benutzer durch den Administrator aktiviert, erfolgt eine automatische Emailbestätigung. Wird die Datenbank durch das automatische

Skript aktualisiert, wird eine Statistik der letzten Änderungen gesendet. (vgl. Datenbeschaffung) .
Zudem wird eine Mailsoftware benötigt, die die Emails an den Mailserver der HSR weiterleitet.

4. Ausblick

Durch unsere Studienarbeit konnten wir eine gute Grundlage für die Weiterentwicklung des ICS ThreatMap schaffen. Durch die intensive Arbeit und das Umsetzen unserer Prototypen wurde eine Applikation entwickelt, die über die essentiellen Funktionen verfügt, um einen überschaubaren Überblick über die Sicherheitslücken der industriellen Systeme zu gewinnen.
Das Projekt kann durch weitere Funktionen erweitert und optimiert werden. Diese möchten wir in diesem Kapitel erläutern.

Preview Ansicht Filter
In der Filterverwaltung können neue Suchfilter hinzugefügt werden. Damit die Gewissheit besteht, dass ein Suchfilter neue Anlagen zurückliefert, wäre eine Preview-Ansicht hilfreich.

ICS gruppieren
Viele Systeme haben Ähnlichkeiten wie von welcher Organisation sie stammen oder wo sie sich befinden. Eine Zusammenfassung oder Gruppierung könnte eine bessere Übersicht für solche Systeme geben.

Bevölkerungspopulation
Die Bevölkerungsdichte hat einen zusätzlichen Einfluss auf den Schweregrad eines ICS. In einer Stadt sind mehr Personen betroffen als beispielsweise auf dem Land. So könnte die ICS zusätzlich zu ihrem Bedrohungsgrad in der Berechnung auch die Bevölkerungsdichte miteinbeziehen.

Whitelist/Blacklist führen
Mit zunehmenden Daten wächst auch das Bedürfnis eine Blacklist einzuführen, der die Filter aus der Filterliste global eingrenzt. Das Update Script holt sich die Filter und negiert die Begriffe in der Blackliste auf den Filter.

Verfügbarkeit der ICS überprüfen
Die bestehenden ICS in der Web Applikation sollten überprüft werden, ob diese in der Engine noch verfügbar sind. Veraltete Daten sollen damit archiviert werden.

Inhaltsverzeichnis I

Inhaltsverzeichnis II

Technischer Bericht (TEB)

Glossar (GLO)

Literaturverzeichnis (LIV)

Technologie- & Filteranalyse (TFA)

Projektplan (PRP)

Risikomanagement

Anforderungsspezifikation (SAS)

Software Architektur Dokument (SAD)

User Interface (UI)

Prototypen (SPR)

Software Qualitätsmanagement (SQM)

Anhang

Zeitungsartikel Fahrlässig Durchlässig

ICS ThreadMap - v1.0

Technischer Bericht (TEB)

Benjamin Kehl
Dominique Sorg

Änderungsgeschichte

Datum	Version	Änderungen	Autor
10.12.2013	0.1	Erstellung Einleitung	Dominique Sorg
12.12.2013	0.2	Anpassungen und Ergänzungen	Benjamin Kehl
12.12.2013	0.3	Was sind ICS, Wo kommen ICS vor, Was ist die Gefahr bei ICS	Dominique Sorg
12.12.2013	0.4	Ergebnisse, Schlussfolgerung	Dominique Sorg
13.12.2013	0.5	Überarbeitung Einführung	Benjamin Kehl
18.12.2013	0.6	Korrekturen Dokument	Benjamin Kehl
18.12.2013	0.7	Korrekturen Dokument	Dominique Sorg
19.12.2013	0.8	Ergänzungen Ergebnisse	Benjamin Kehl

Inhalt

1. Einführung

Die Sicherheit von ICS (Industrial Control Systems) Kontrollsysteme in Industrien und wie man sich gegen online Angriffe bestmöglich schützt, erhielt in den letzten Jahren immer mehr an Aufmerksamkeit. Das Streben nach technischem Fortschritt, treibt die Automatisierung von Geschäftsprozessen in allen Bereichen voran. Damit sich heutzutage ein Unternehmen auf dem Markt bewähren kann, sind Überlegungen für den Einsatz von ICS Systemen unerlässlich. Sie finden ihren Einsatz beispielsweise in Energie- und Wasserversorgungen, Transport- sowie im Gesundheitswesen aber auch in Produktionen oder in Pharmaindustrien. Automatisierungslösungen sind nichts Neues, sondern das gibt es schon seit Jahrzehnten. Jedoch war die Sicherheit damals noch nicht so wichtig, wie es heute der Fall ist.

Durch die Entdeckung von Malware wie Stuxnet erlangt die Frage nach Sicherheit von ICS/SCADA Systemen immer mehr an Wichtigkeit. Unter Malware versteht man eine Software, die einen unerwünschten oder schädlichen Code ausführt. Durch das massive Medieninteresse werden ICS/SCADA Systeme zu beliebten Angriffszielen. Unter anderem wurden erhebliche Sicherheitslücken in Systemen von unterschiedlichen Herstellern gefunden. Geschweige denn, dass viele dieser Systeme offen, ungeschützt und öffentlich im Netz verfügbar sind. Diese kritischen Systeme werden aus Komfortgründen, aufgrund günstiger Internetlösungen oder durch mangelnde Schulung an das Internet veröffentlicht. Die Steueranlagen stehen damit für Hacker ohne Authentisierung offen zur Verfügung. Es ist ein weltweites Problem, das mittlerweile auch die Schweiz betrifft.

Die Schweiz verfügt über kritische Infrastrukturen, die essentielle Güter wie Verkehr, Kommunikation oder Energie sicherstellen. Grossflächige Ausfälle von kritischen Infrastrukturen haben schwerwiegende Folgen für die Bevölkerung und Wirtschaft. Aus diesem Grund ist ein Ziel der Cyber-Security-Strategie des Bundes, die kritischen Infrastrukturen gegen Hacker Angriffe besser zu schützen.

Der Artikel der SonntagsZeitung am 01. Dezember 2013 von Florian Imbach und Alexandre Haederli zeigt einen aktuellen Fall in der Schweiz[1]. Dabei lag das Schliesssystem des ST. Jakob-Parks in Basel während eines Jahres für Hacker offen. Durch das massive Medieninteresse werden die kritischen Infrastrukturen zu beliebten Angriffszielen. Unter anderem wurden erhebliche Sicherheitslücken in Systemen von unterschiedlichen Herstellern gefunden. Geschweige denn, dass viele dieser Systeme offen, ungeschützt und öffentlich mit dem Internet verbunden sind.

1.1 Übersicht

Dieses Dokument gibt eine Einführung in die Problematik von kritischen Infrastrukturen in der Schweiz. Es wird erklärt was unter ICS/SCADA Systeme verstanden wird, wo sie ihren Einsatz finden und wo die Gefahren lauern. Mit der gegebenen Aufgabenstellung wird nach einer Lösung für die gegebene Problemstellung gesucht. Dieses Dokument beschreibt das Vorgehen und die entstandenen Ergebnisse. Die Schlussfolgerung bewertet die Ergebnisse der Studienarbeit und zeigt weiterführende Möglichkeiten auf.

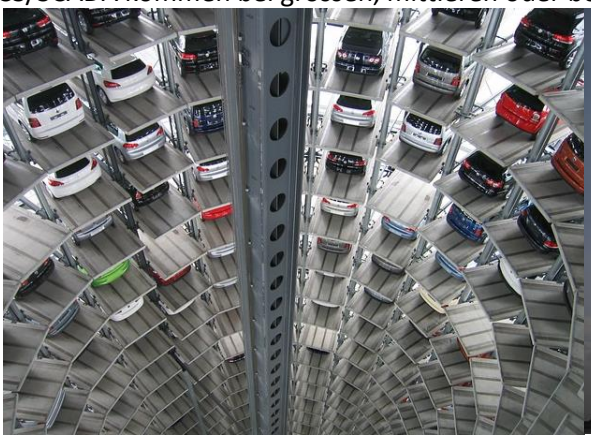
In einem externen Dokument wird die Technologie- und Filteranalyse detailliert beschrieben. Für Software Interessierte helfen die Software Engineering Dokumente weiter.

[1] Imbach, F., & Alexandre, H.(01.12.2013). SonntagsZeitung. Von http://www.sonntagszeitung.ch/fokus/artikel-detailseite/?newsid=268454 (12.12.2013)

1.2 Was sind ICS/SCADA

Wenn man über ICS (Industrial Control System) spricht sind verschiedene Kontrollsysteme wie SCADA, UPS, BMS, ERP, HMI, PDU, PLC, PLCND gemeint. Diese Kontrollsysteme finden Anwendungen in industriellen Umgebungen. Darunter fallen typischerweise Elektrizitätswerke, Wasserversorgungen, Öl-/Gasplattformen, usw. Oft wird SCADA (Supervisory Control and Data Acquisition) im Bezug zu ICS genannt. Darunter versteht man das Überwachen und Steuern von technischen Prozessen mittels eines Computersystems. Damit möchte man eine Gesamtübersicht des Arbeitsworkflows innerhalb eines Unternehmen erhalten und diese an einer zentralen Stelle bedienen sowie verwalten können. Jedoch deckt SCADA nur einen kleinen Teil von ICS ab. Die folgenden Bilder visualisieren Einsatzorte von SCADA Steueranlagen.

ICS/SCADA kommen bei grossen, mittleren oder bereits auch in kleinen Industrien zum Einsatz.




In Anlagen wie Motor-Produktionen, in Atomkraftwerken, ..


..in Wasserversorgungsanlagen oder auch in Staudämmen werden Kontrollsysteme genutzt[2].

[2] Pixabay. (kein Datum). Autos Technik Vw Parkhaus Lagerhalle Fahrzeuge. doi:Public Domain CC0 https://creativecommons.org/publicdomain/zero/1.0/deed.de
Pixabay. (kein Datum). Kernkraftwerk Zentrale Dampf Energie Non. doi:Lizenz Public Domain CC0 https://creativecommons.org/publicdomain/zero/1.0/deed.de
Pixabay. (kein Datum). Segovia Dam Ponton Hochwasserentlastung Bauarbeiten. doi:Public Domain CC0 https://creativecommons.org/publicdomain/zero/1.0/deed.de
Pixabay. (kein Datum). Solarzellen Energie Strom Umweltfreundlich. doi:Public Domain CC0 https://creativecommons.org/publicdomain/zero/1.0/deed.de

Des Weiteren werden in der Schweiz kritische Anlagen auch in folgenden Bereichen genutzt:

Sektor	Teilsektoren
Behörde	Parlament, Regierung, Justiz, Verwaltung, Kulturgüter, Ausländische Vertretungen
Energie	Erdgasversorgung, Erdölversorgung, Stromversorgung
Entsorgung	Abfall, Abwasser
Finanzen	Banken
Gesundheit/Medizin	Spitäler, Labors
Industrie	Forschungseinrichtungen
Information und Kommunikation	Informationstechnologie, Medien, Telekommunikation
Nahrung	Lebensmittelversorgung, Wasserversorgung
Öffentliche Sicherheit	Armee, Blaulichtorganisation, Zivilschutz
Verkehr	Luftverkehr, Postverkehr, Schienenverkehr, Schiffsverkehr, Strassenverkehr (darunter gehören auch Rotlichtüberwachungsanlagen oder Geschwindigkeitskameras)
Gebäude	Klimaanlagen, Lüftungen, Heizungen, Schliesssysteme
Sicherheit	Überwachungskamerasysteme, Webcams

ICS/SCADA Systeme werden aus verschiedenen Kontrollsystemen zusammengebaut. Die folgenden Bilder zeigen, wie solche Kontrollsysteme (PLC)[3] aussehen können.

ICS/SCADA Systeme werden in einem Dashboard visualisiert. Dabei wurden Screenshots von einer online verfügbaren HMI DEMO[4] verwendet.

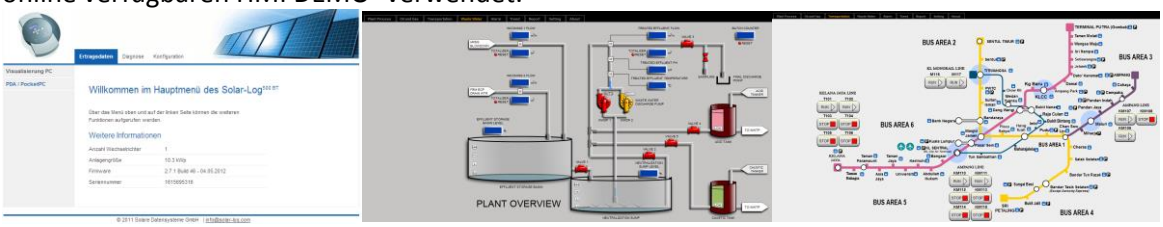


Abbildung 1 ICS/SCADA Dashboard (Solar Log, Wasserversorgung HMI, Transport HMI)

ICS/SCADA werden in verschiedensten Variationen und Bereichen genutzt und machen für den Betreiber das Leben *einfacher*.

[3] Mixabest. (7. March 2008). File:MITSIBISHI PLC Panel.jpg. *PLC*. doi:This file is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported license.
Mixabest. (18. 12 2012). File:MITSIBISHI PLC 04.png. *PLC*. doi:This file is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported license.

[4] Integraxor. (17. 12 2013). *Demo HMI Integraxor*. Abgerufen am 17. 12 2013 von http://www.integraxor.com/demo.htm

1.3 Worin bestehen die Gefahren von ICS/SCADA

Das Internet bringt heutzutage viele Vorteile mit sich. Auch wir Schweizer nutzen die Vorteile des Internets, wo immer auch möglich. Im geschäftlichen Bereich kann es genutzt werden, um schnell an Informationen zu gelangen, um Informationen auszutauschen, um zu kommunizieren und um Arbeitsprozesse abzuwickeln. Trotz den Vorteilen des Internets, müssen wir auch die Schattenseiten in Betracht ziehen: Internetseiten können einfach von Betrügern und Kriminellen ausgenutzt werden. Recherchen und Gespräche, die vor allem im Kapitel Fachsymposium beschrieben sind, zeigen, dass es bei den Betreibern von ICS Systemen nicht immer um IT-Fachleute handelt. Meist handelt es sich um Automatiker, die eher für die Funktionalität des Kontrollsystems verantwortlich sind. Sie sind sich aber nicht bewusst, dass ihre Anlangen öffentlich im Internet zugänglich sind. Dabei stellen die angeschlossenen Kontrollsysteme am Internet eine ernstzunehmende Bedrohung für die Schweiz dar.

Die Produkte von ICS Herstellern weisen an diversen Stellen wie beispielsweise am Web Interface oder in den Protokoll Headerdaten Muster auf, nach denen gezielt gesucht werden kann. Dabei scannt die Suchmaschine Shodan, IP-Adressen- und Port-Kombination und speichert die Antwort Headerdaten in seiner Datenbank ab. Für ICS sind dabei TCP/UDP Ports wie SSH, Telnet, FTP, HTTP, HTTPS oder SNMP relevant. Am 10.12.2013 liefert Shodan insgesamt 1‘182‘766 potentiell angreifbare Systeme (alle Resultate) als Suchresultate aus der Schweiz. Auch über die Suchmaschine Google lassen sich ICS finden. Google sucht im Gegensatz zu Shodan, nach Titeln oder Inhalten von Webseiten.

Ein Hacker könnte damit passive Information-Gathering betreiben und dann zuschlagen, wenn die Gelegenheit günstig ist. Denn einen wesentlichen Nachteil bringt das Kontrollsystem nebst den vielen Vorteilen mit sich. Durch die zentrale Verwaltung der Sensoren gibt es auch einen zentralen Ausschaltknopf. Man möchte sich nicht vorstellen, was passieren könnte, wenn unbefugte Hacker unsere Wasserversorgungen oder Elektrizitätswerke manipulieren oder abschalten würden und was für Auswirkungen dies auf die Schweizer Wirtschaft und Bevölkerung hätte.

1.4 Wie funktionieren ICS/SCADA

SCADA Systeme sind heutzutage fest in Geschäftsprozesse integriert. Die untere Pyramide visualisiert, wie die Komponenten miteinander in Verbindung stehen.

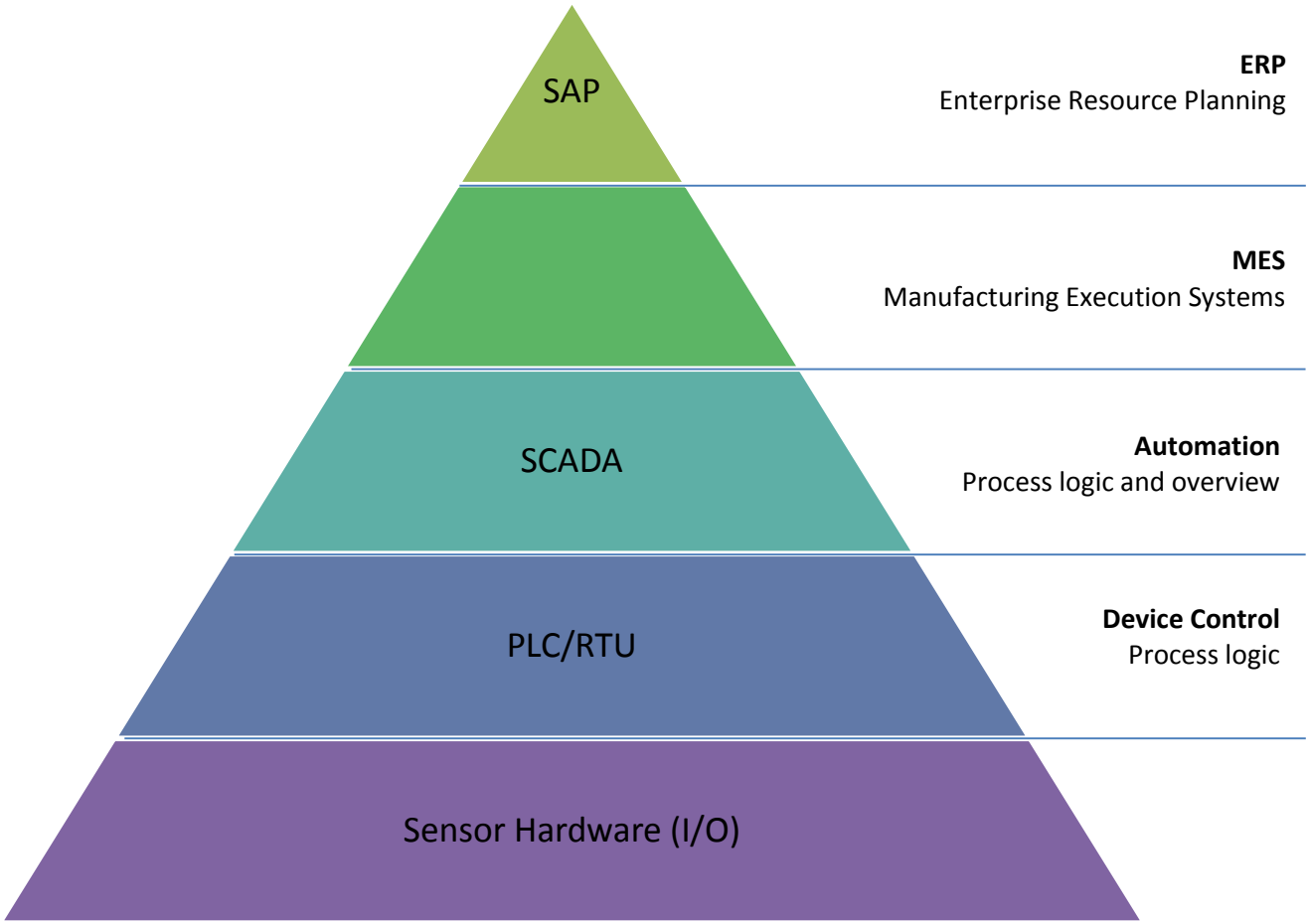


Abbildung 2 ICS Pyramide

Zuunterst sind die verschiedenen Sensoren, Pumpen, etc. die als Hardware installiert werden. Die Device Control Schicht ist die Programmier- Schnittstelle der Sensoren. Diese Device Controls werden alle in einem SCADA System zusammengeführt, um die Verwaltung aller Sensoren zu vereinfachen. Anschliessend können die SCADA Systeme an SAP Lösungen angehängt werden.

2. Problemstellung

„Der Schutz von kritischen Infrastrukturen ist an sich nichts Neues. Gefehlt hat aber lange Zeit eine sektorübergreifende Koordination und ein einheitliches Vorgehen auf nationaler Ebene.“ heisst es in einem Artikel vom Bund[5]. Die Schweiz ist auf funktionierende kritische Infrastrukturen angewiesen. Diese stellen unverzichtbare Dienstleistungen wie Energie, Wasser, Kommunikation oder Verkehr sicher. Wenn diese Infrastrukturen gestört werden, hat das schwerwiegende Auswirkungen auf die Bevölkerung und Wirtschaft. Auch grossflächige Ausfälle wie ein Stromausfall, können auf andere Infrastrukturen wie Wasserversorgung oder Schienenverkehr übergreifen.

Die momentane Situation der Schweiz zeigt aber, dass eine nationale Überwachung und der Schutz von kritischen Anlagen noch nicht möglich sind. Aus diesem Grund rief der Bund die SKI (*Nationale Strategie zum Schutz kritischer Infrastrukturen)* Strategie ins Leben. Mit der SKI Strategie soll ein Weg gefunden werden, um kritische Infrastrukturen in der Schweiz besser zu schützen.

Eine Arbeit die dieses Problem in Angriff nimmt ist IRAM. IRAM analysiert und zeigt unsichere ICS/SCADA Systeme auf einer Karte. IRAM wurde von der SCADACS an der Universität Berlin entwickelt. Ihr Ziel ist es Strategien zu analysieren, um die kritischen Systeme sicherer zu machen. IRAM zeigt eine grosse Menge von ungeschützten ICS/SCADA Systemen weltweit auf.

Jedoch kann auch IRAM die Erwartungen, kritische Anlagen zu schützen, nicht vollumfänglich erfüllen. Mit IRAM ist es zwar möglich diverse Anlagen anzusehen, jedoch können diese nicht abgearbeitet werden, um die Sicherheit innerhalb der Schweiz zu verbessern. Des Weiteren können keine Statistiken oder Trends festgestellt werden, ob sich die Situation in der Schweiz verbessert oder verschlechtert hat. Die Anlagen können auch nicht bearbeitet oder nach deren Schweregrad klassifiziert werden.

Daraus leitet sich die Studienarbeit ICS ThreatMap ab. In dieser Studienarbeit soll eine Webapplikation entwickelt werden, die sich vor allem auf die Abarbeitung von kritischen Anlagen spezialisiert.

3. Aufgabenstellung

Diese Studienarbeit beschäftigt sich mit der Entwicklung einer Webapplikation namens ICS ThreatMap, welche kritische ICS/SCADA Kontrollsysteme in der Schweiz erfasst und auf einer Karte visualisiert bzw. darstellt. Zu den kritischen ICS/SCADA Systemen gehören zum Beispiel Energie- und Wasserversorgungen, Transport und Gesundheitswesen, die ungeschützt im Internet erreichbar sind.

Da ICS ThreatMap sensible Informationen enthält, ist es nur einem engen Nutzerkreis wie MELANI, ICS-CERT oder Sicherheitsfirmen zugänglich. Die Nutzer sind in der Lage nach ICS/SCADA zu suchen und diese mit weiteren Informationen anzureichern. Ein Mehrwert der Webapplikation ist das selbstentwickelte Trouble Ticket System. Mit dem Trouble Ticket System werden die Bedrohungen abgearbeitet. Die betroffenen Betreiber werden kontaktiert und über die Verletzlichkeit Ihrer Systeme aufgeklärt. Die Fortschritte werden im jeweiligen Ticket dokumentiert und die getätigten Massnahmen laufend überprüft. Ziel ist es, die betroffenen Betreiber auf die Gefahren aufmerksam zu machen und sie zu bewegen, die Kontrollsysteme besser zu schützen oder sie aus dem Netz zu entfernen.

[5] BABS, B. f. (2013). *Schweizer Eidgenossenschaft.* Abgerufen am 12. 10 2013 von http://www.bevoelkerungsschutz.admin.ch/internet/bs/de/home/themen/ski.html

4. Ziele

Das Ziel der Studienarbeit ist daher, eine Web Applikation bezüglich dieser Aufgabenstellung zu erstellen. Die Web Applikation soll dem Bund, ICS-Cern und den Sicherheitsfirmen bei der Abarbeitung von kritischen Kontrollsystemen unterstützen. Es soll möglich sein jederzeit ein ICS/SCADA System zu verfolgen und bereits getätigten Massnahmen laufend zu kontrollieren. Dadurch sollen den betroffenen Firmen die Gefahren aufgezeigt werden, um sie dazu zu bewegen ihre Kontrollsysteme besser zu schützen oder aus dem Netz zu entfernen.

5. Vorgehensweise

Als Arbeitsprozess wurde RUP gewählt. Der Grund, warum wir dieses Vorgehensmodell gewählt haben, liegt an den bereits gesammelten Erfahrungen im SE2-Projekt. Ausserdem können mittels RUP die Aufteilungen der Arbeit in einzelne Phasen und Iterationen mit Meilensteinen definiert werden.

Die Anfangsphase, auch Inception-Phase genannt, dauerte eine Woche und sie wurde dazu genutzt um die Aufgabenstellung zu klären und das Projekt zu starten.

Ab der zweiten Woche startete die Elaboration-Phase welche 4 Wochen lang dauerte. In den ersten zwei Wochen definierten wir die Risiken und erstellten den Projektmanagementplan.
Wir begannen auch mit der Einarbeitung für Zend Framework 2 und Google Maps.
In der zweiten Phase der Elaboration wurde vermehrt Zeit für die Erstellung der Prototypen in Zend Framework 2 und Google Maps investiert. Auch ein erster Entwurf für das Datenbankmodell wurde erstellt und die Anforderungen in Use Cases ausgearbeitet.

Die Construction-Phase, dauerte 7 Wochen und wurde in 4 einzelne Phasen in je zwei Wochen unterteilt. Die letzte Woche diente für das Refactoring und Testing.
In der ersten Phase wurde die Datenbank aufgesetzt, welche als Basis für die Webapplikation dient. Zusätzlich wurde die Suchfunktion und Anzeige für ein ICS implementiert.
Die zweite Phase diente dem Frontend Design, das heisst der GoogleMap- und der Statistik-Anzeige. Auch mit dem Konfigurationsbereich haben wir in dieser Phase begonnen, damit die Suchfilter auf der Seite verwaltet werden können. In der dritten Phase mussten neue ICS hinzugefügt und bestehende bearbeitet werden. Um bestimmte ICS zu dokumentieren, wurde auch ein Trouble Ticket System implementiert. Die vierte und letzte Phase der Construction-Phase diente für das Bugfixing, Refactoring und Testing.

In der Schlussphase, auch Transition-Phase, wurden die Dokumente aktualisiert und für die Abgabe vorbereitet.

5.1 Verhaltensregeln

Während dieser Arbeit definierten wir uns Verhaltensregeln bei einem Fund von ICS/SCADA Systemen. Es kann durchaus vorkommen, dass wir auf nicht geschützte Steueranlagen stossen.

- Wir wollen die Systeme für unsere Web Applikation sammeln. Wir interagieren mit keiner Steueranlage ausser für das Betrachten der Protokoll Headerdaten und um zu prüfen, ob sich hinter einem Befund ein ICS System befindet. Manipulationen oder Änderungen werden nicht vorgenommen.

- Wir melden uns nicht absichtlich in Steuersysteme an. Diese Logins werden abgebrochen. Jedoch werden diese Systeme in unsere Applikation als Funde miteinbezogen.

- Wir scannen keine Domänen im Internet. Wir verwenden öffentlich zugängliche Quellen wie Shodan oder Google.

- Gefundene Anlagen werden nicht an Dritte weitergegeben und werden mit HTTPS Verschlüsselung und Benutzer Authentisierung geschützt.

5.2 Fachsymposium Anlagesicherheit

Damit wir uns besser in die Thematik unserer Studienarbeit einarbeiten konnten, besuchten wir am 05.11.2013 das Fachsymposium der HIMA in Bern. Der Fokus der Veranstaltung beruhte sich auf funktionale Sicherheit (Safety) und Cyber Security auf Automatisierungslösungen von ICS bzw. SCADA, PLC, etc. Funktionale Sicherheit dient dem Schutz von Mensch, Maschine und Umwelt. Cyber Security hingegen zielt auf Verfügbarkeit, Integrität und Vertraulichkeit von Daten und Programmen. Anhand von Live Demos und Vorträgen konnten wir einen Einblick erhalten, wie die Systeme eingesetzt werden und wo die Systeme evtl. verletzlich sind. Des Weiteren konnten wir interessante Gespräche mit HIMA Verantwortlichen führen. Dabei konnten folgende neue Erkenntnisse gesammelt werden:

- Die Systeme werden von Kunden selbst eingerichtet und konfiguriert.

- Die Systeme sollten auf keinen Fall, egal ob mit oder ohne Login-Bereich im Internet verfügbar sein. Der Zugriff auf die betroffenen Systeme sollte über VPN getunnelt werden, was auch von dem jeweiligen Hersteller empfohlen wird. HIMA selber, tunnelt ihre Protokolle für den Betrieb der Systeme über VPN.

- Die Hersteller sind noch am „Lernen“, um Muster in ihren Anlagen zu vermeiden. Damit Suchmaschinen wie Shodan diese nicht allzu leicht auffinden können. Dabei geht man in die Richtung nur noch das nötigste im Web anzuzeigen.

- Bei HIMA kann der Web Zugriff in den neusten Produkten, direkt abgestellt und blockiert werden. Bei älteren Produkten oder bei anderen Herstellern ist der Webzugriff ohne Wissen des Nutzers aktiviert.

6. Ergebnisse

6.1 Erreichte Ziele

Als Ergebnis entstand eine erste Version der Web Applikation ICS ThreatMap. Die Web Applikation ist in der Lage mit einem täglich laufenden Programm, die ICS Daten von externen Suchmaschinen zu holen und in unsere Web Applikation abzulegen. Die ICS Daten werden mit weiteren Informationen ergänzt und für die Benutzer aufbereitet.

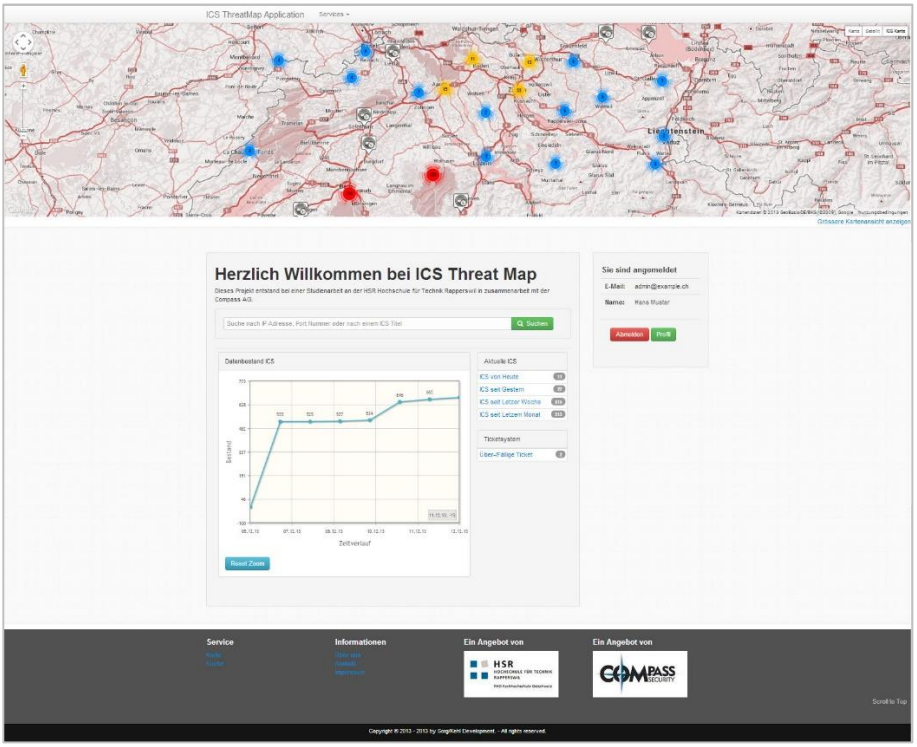


Abbildung 3 Startseite ICS ThreatMap

Die ICS Daten werden in einer Karte nach deren Schweregrad visualisiert (Abbildung 4). Detailinformationen sind für eingeloggte Benutzer zugänglich. Für die Öffentlichkeit werden die Systeme nicht alle Informationen dargestellt.

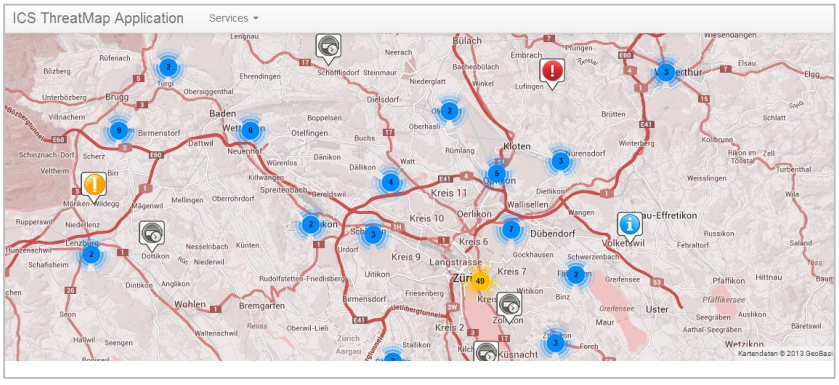


Abbildung 4 ICS Karte

Ein Dashboard visualisiert den Trend von neuen ICS, die täglich hinzugefügt werden (Abbildung 5).

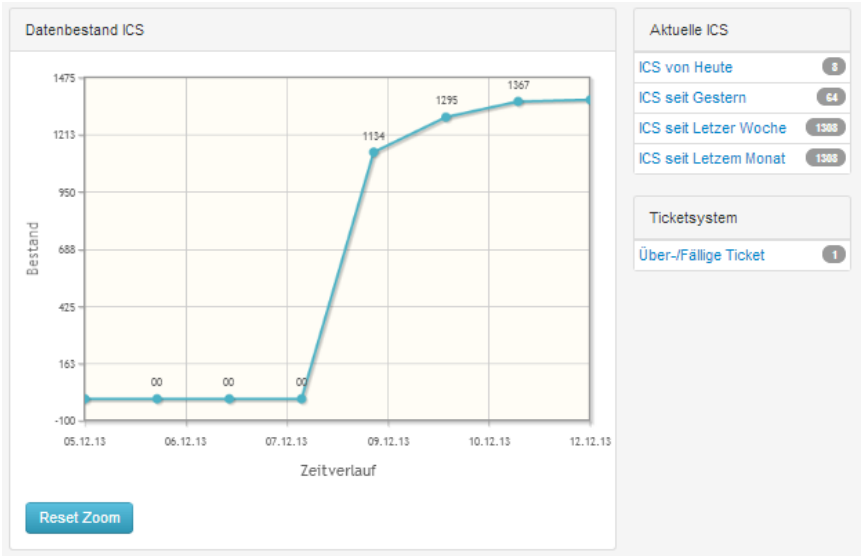


Abbildung 5 Dashboard

Eine Benutzerverwaltung ermöglicht nur einem kontrollierten Kundenkreis Zugang zu der Applikation zu erteilen (Abbildung 8). Registrierte Benutzer werden standartmässig vorerst deaktiviert. Der Administrator kann dem Benutzer den Zugriff berechtigen (Abbildung 9) und er kann eine von drei möglichen Berechtigungsstufen auswählen. Darunter zählt das Mitglied, das nur Leserecht auf der Webapplikation hat, der Operator mit Schreibrechte und der Administrator mit zusätzlichen Rechte wie die Benutzerverwaltung. Auf der Profilseite können sich angemeldete Nutzer ihre Informationen ansehen oder bearbeiten.

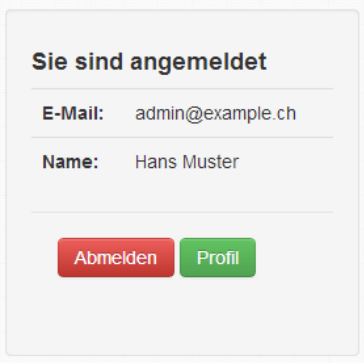


Abbildung 6 Login/Logout-Widget

Abbildung 7 Profilseite

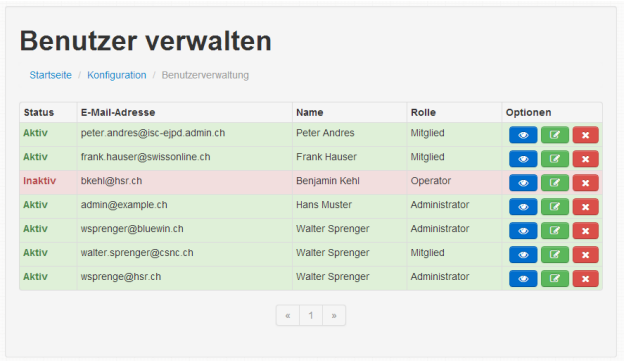


Abbildung 8 Benutzerverwaltung

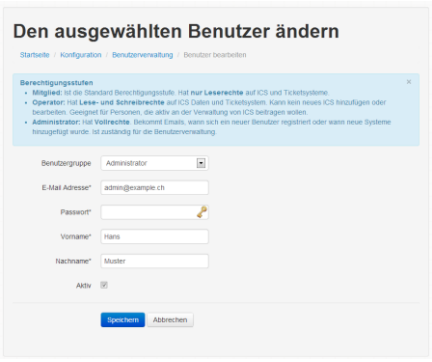


Abbildung 9 Benutzeraktivieurng und Rollenvergabe

Die Filterverwaltung ermöglicht neue Filter hinzuzufügen, bestehende Filter zu bearbeiten, unerwünschte Filter zu entfernen oder einfach eine Liste aller Filter anzuzeigen.

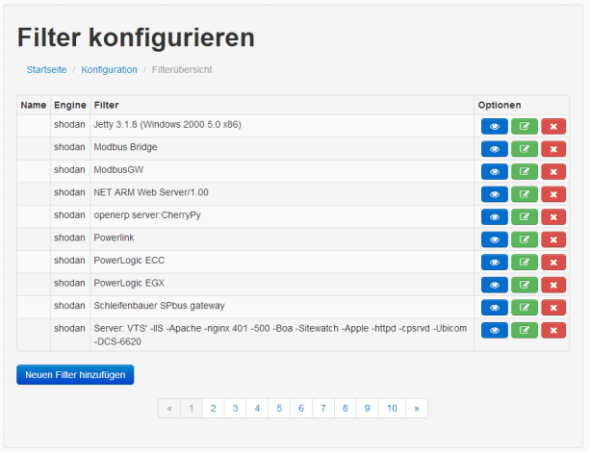


Abbildung 10 Filteransicht

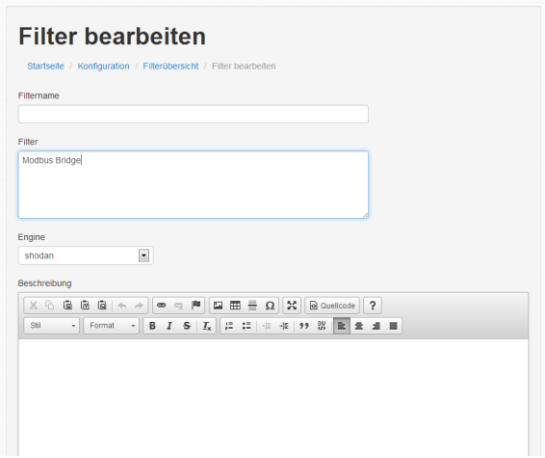


Abbildung 11 Filter bearbeiten

Mit der Suchfunktion kann spezifisch nach kritischen ICS Anlagen gesucht werden.

Abbildung 12 Suchfunktion Startseite

Eine erweiterte Suchfunktion bietet spezifischere Filtermöglichkeiten wie Bedrohung, Kategorie, Firma, Service, usw. an (Abbildung 13). Die Resultate werden anhand des Schweregrads mit einem Symbol dargestellt (Abbildung 14).

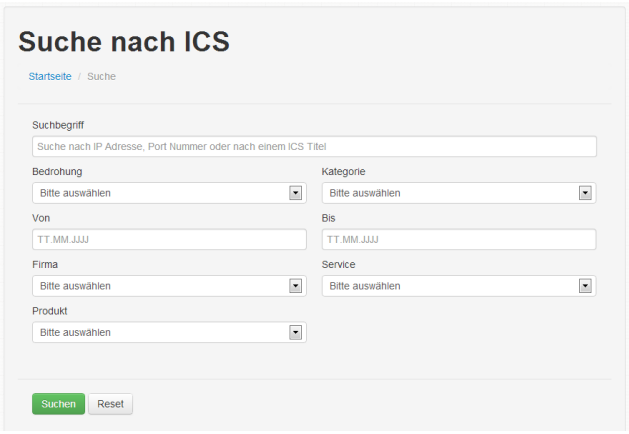


Abbildung 13 Erweiterte Suchfunktion

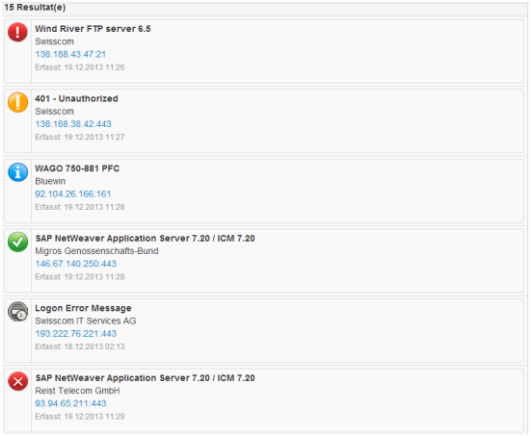


Abbildung 14 Suchresultate mit Schweregrad

Die ICS Daten werden für den Benutzer aufbereitet und strukturiert dargestellt (Abbildung 15Abbildung 15 ICS Detailansicht). Die Rubrik Suchfilter (Abbildung 16), verweist auf die Filter, die das ICS gefunden haben. Eine History (Abbildung 17) zeigt die letzten Änderungen mit Datum-, Zeit- und Benutzerangabe an einem ICS auf.

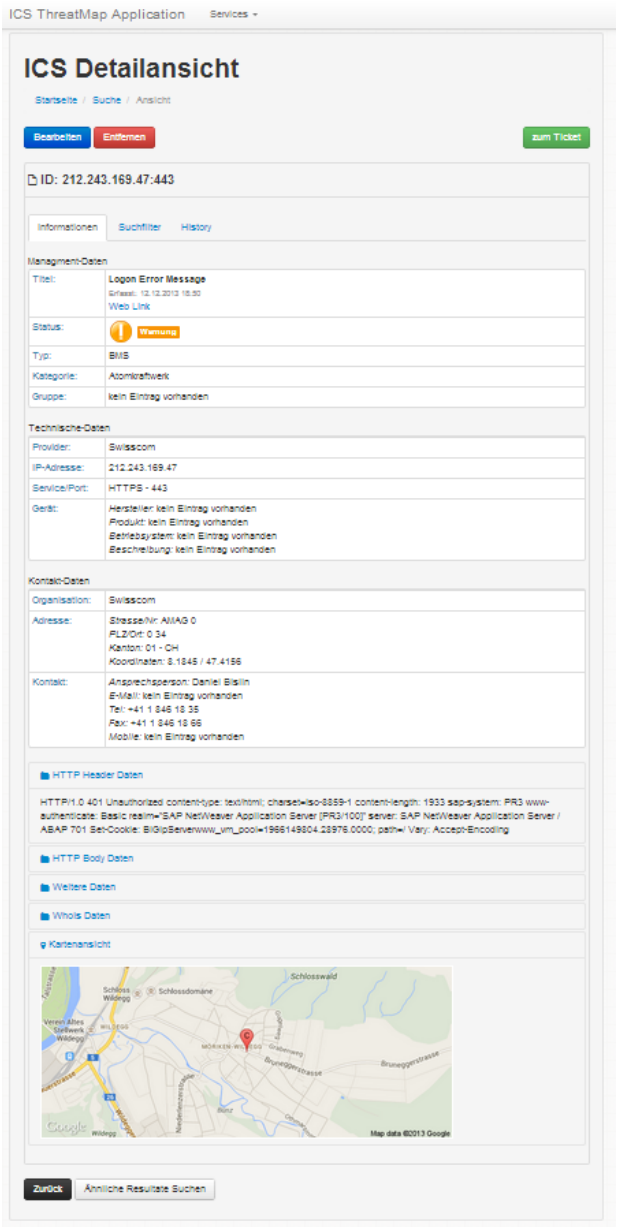


Abbildung 15 ICS Detailansicht

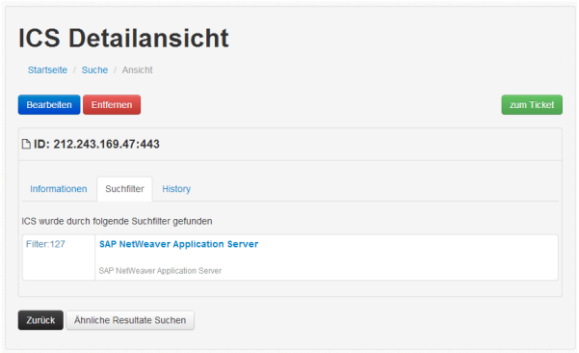


Abbildung 16 Suchfilter Anzeige

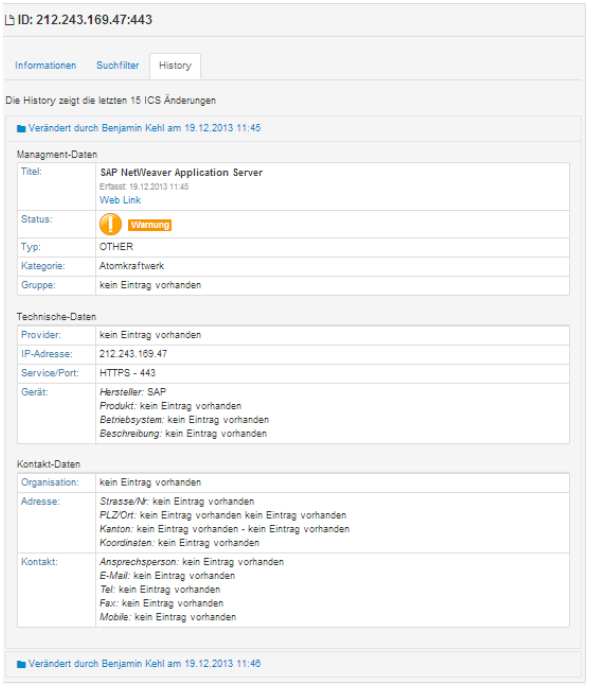


Abbildung 17 History Auflistung

Ein Trouble Ticket System hilft bei der Abarbeitung von kritischen ICS (Abbildung 18). Für ein ICS kann ein Ticket erstellt, bearbeitet oder entfernt werden. Überfällige Tickets werden in einer separaten Liste aufgeführt (Abbildung 19).

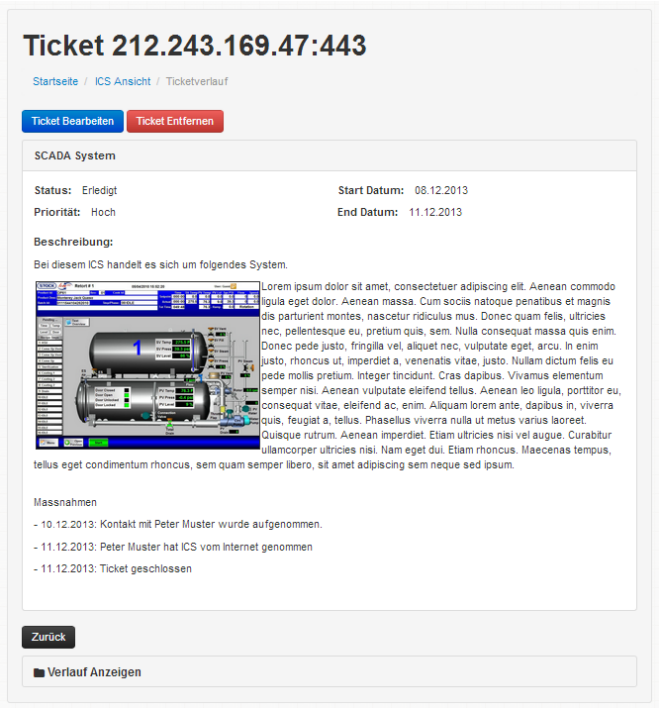


Abbildung 18 Trouble Ticket System

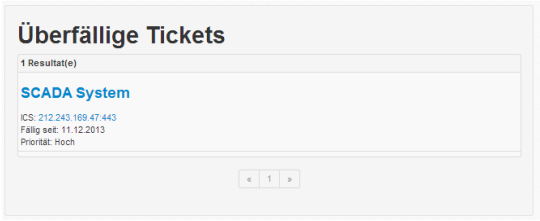


Abbildung 19 Überfällige Tickets

6.2 Optimierungen

Mit unserer Webapplikation wurde eine gute Grundlage geschaffen, auf der aufgebaut werden kann. Aus Zeitgründen können wir das Thema nicht weiter vertiefen. Das Projekt kann durch weitere Funktionen weiterentwickelt und optimiert werden. Wir konzentrierten uns vor allem auf die Funktionen der Priorität 1 und konnten diese auch grösstenteils realisieren. Folgende Funktionen, die mit Priorität 2 gewichtet worden sind, können bei einem weiteren Release erweitert werden.

Google als Suchmaschine
Die Web Applikation funktioniert mit Shodan. Aus zeitlichen Gründen konnte eine zweite Suchmaschine wie Google nicht mit eingebunden werden.

Preview Ansicht Filter
Bevor neue Filter in der Filterverwaltung hinzugefügt werden, wäre für die Benutzer eine Preview Ansicht hilfreich, um die Suchergebnisse vorher zu überprüfen.

ICS Gruppieren
Verschiedene ICS die ähnlich sind, sollten Gruppiert werden können.

Bevölkerungspopulation Karte
Um einen Überblick zu erhalten, welche ICS Systeme die grössten Auswirkungen auf die Bevölkerung haben, kann eine Populations-Visualisierung in die bestehende Google Karte integriert werden.

Whitelist/Blacklist führen

Mit zunehmenden Daten wächst auch das Bedürfnis eine Blacklist einzuführen, der die Filter aus der Filterliste eingrenzt. Das Update Script holt sich die Filter und negiert die Begriffe aus der Blacklist in den Filter.

Verfügbarkeit der ICS überprüfen
Die bestehenden ICS in der Web Applikation sollen überprüft werden, ob diese in den jeweiligen Suchmaschinen noch verfügbar sind. Veraltete Daten sollen archiviert werden.

7. Problemlösungen

Bei der Umsetzung der Web Applikation tauchten unterschiedliche Probleme auf, die es zu lösen galt. In den folgenden Kapiteln gehen wir auf die Probleme ein und beschreiben, wie wir diese gelöst haben.

Um detailliertere Beschreibungen oder Vorgehensweisen nachzuvollziehen, verweisen wir hier gern auf unser Technologie- & Filteranalyse Dokument sowie auf unsere Software Engineering Dokumente.

7.1 Technologie

Das Webprojekt soll mit Open Source Technologien realisiert werden. Welche Technologie dabei eingesetzt wird, ist uns Entwicklern überlassen. Damit wir für das Web Projekt die richtigen Technologien wählen können, müssen zuerst die Anforderungen analysiert werden.

- Open Source (Wo immer möglich)
- Betriebssystem unabhängig
- Browser unabhängig
- Installation Dokumentation
- E-Mail Support für Statistik Report
- Daten werden in einer Datenbank verwaltet

Im Verlauf der Studienarbeit war uns sehr schnell klar, welche Technologien wir im Rahmen dieser Arbeit verwenden wollen. Im Folgenden eine Auflistung davon:

- **Server -** Uns wurde ein virtueller Server mit Betriebssystem Ubuntu 12.04.3 zur Verfügung gestellt, welches aufgrund des mitgelieferten Redmine und Jenkins bereits mit Apache2 und MySQL vorinstalliert ist.
- **Sendmail -** Ist eine freie E-Mail-Software für die Unix Community. Sendmail wurde auf dem virtuellen Server installiert und wird insbesondere bei der Emailbenachrichtigung von neu registrierten Benutzern und wann das letzte Update Script durchgeführt wurde, verwendet.
- **MySQL -** Für die Datenspeicherung wird MySQL verwendet. Die Datenbank liegt auf den gleichen Server wie die Web Applikation.
- **Zend Framework 2 -** Für die Realisierung haben wir uns für das Zend Framework 2 entschieden. ZF2 ist ein Backend-Framework geschrieben in PHP, welches Konzepte wie MVC oder Frontcontroller fest integriert hat.
- **Twitter Bootstrap -** Für das Frontend verwenden wir Twitter Bootstrap. Dies erleichtert uns die Gestaltung. Ausschlaggebend für den Einsatz von Bootstrap waren aber die Argumente, dass Bootstrap *responsives Webdesign* unterstützt und das es auf sehr vielen Browsern gleich aussieht.
- **JavaScript -** Für die Visualisierung der ICS auf einer Google Maps Karte und für Statistiken wurde JavaScript verwendet. JavaScript ist eine clientseitige Skriptsprache, welche für die Erzeugung von dynamischen HTML verwendet wird.

7.2 Suchfilter

Mit dem Projektstart fing auch gleich die Suche nach ungeschützten ICS/SCADA Systemen an. Der Fokus liegt auf den Suchmaschinen Shodan und Google. Es ist denkbar, dass in Zukunft weitere Suchmaschinen hinzukommen. Bei Shodan konnte uns unser Betreuer unlimitierte Accounts beschaffen. Mit diesen galt es, ICS/SCADA Systeme zu suchen und zu dokumentieren. Wichtig ist es dabei die Suchmöglichkeiten intelligent einzusetzen, um die Suchresultate soweit wie möglich einzudämmen. Unsere erstellte Filterliste sollte später in unsere Applikation integriert werden.

In einem ersten Schritt befassten wir uns mit den Filtermöglichkeiten der jeweiligen Suchmaschinen. Dies war aber nicht immer einfach, weil Shodan vordefinierte Suchfilter wie *ICS* oder *SCADA* hatte, die jedoch ungenaue Suchresultate lieferten. In einem weiteren Schritt suchten wir nach möglichen Schweizer Industriefirmen, wie Wasserversorgungen oder Elektrizitätswerken. Wir nutzten vor allem öffentliche Quellen wie *search.ch* oder *suche.ch*. Diese Unternehmen integrierten wir in unsere Suchfilter. Anschliessend bauten wir typische Ports für Steueranlagen in die Filter ein, um die Suchergebnisse weiter einzudämmen. Diese wären beispielsweise *22 SSH*, *23 Telnet*, *80 HTTP*, *443 HTTPS*, *161 SNMP*.

Wir realisierten, dass unsere Suchergebnisse auch Router und Webserver lieferten, die nicht wirklich erwünscht sind. Aus diesem Grund entschieden wir uns, eine Liste von Kontrollsystemen mit Herstellern und Produkten spezifisch zu führen.

Im Laufe der Studienarbeit bemerkten wir, dass auch Shodan immer mehr Suchfilter zusammenträgt, die spezifisch nach Hersteller und Produkte sucht. Auch der Begriff *ICS* erlangte eine neue Bedeutung.

Für die gefundenen Suchfilter wird eine Liste geführt. Die Filterliste beinhaltet zum Teil eigene Suchfilter, Filter die Shodan veröffentlicht hat und gefundenen Filter auf der Internetseite ScadaStrangeLove.blogspot.ch. Mittlerweile haben wir 140 Shodan und Rund 13 Google Suchfilter.

Am 12.12.2013 fanden wir mit unserer Filterliste bis zu 1300 potentiell Angreifbare ICS Systeme.

7.3 Datenbeschaffung

Nachdem wir unsere Filterliste zusammengestellt haben, implementierten wir ein kleines Programm, das täglich die abgefragten Daten aus den Suchmaschinen holt. Die gelieferten Suchresultate werden in die Datenbank abgespeichert, bei bestehenden Daten werden sie aktualisiert und bei entfernten Daten archiviert. Die Suchfilter werden ebenfalls in die Datenbank abgelegt, damit die Benutzer später die Liste beliebig bearbeiten oder ergänzen können. Zudem soll nachvollziehbar sein, welche ICS mit welchen Suchfiltern gefunden wurden.

Damit wir unsere ICS eindeutig ansprechen können, verwenden wir IP- und Port-Kombinationen. Wir gehen davon aus, dass die gleiche IP Adresse mehrere ICS auf verschiedensten Ports haben können. Aber ein System unter eindeutiger IP- und Port-Kombination hat immer nur ein System in Betrieb.

7.4 Archivierung

Damit wir Datenverluste vermeiden können, entschieden wir uns prinzipiell keine bestehenden Daten aus der Datenbank zu löschen. Bestehende Daten werden nur an den nötigsten Stellen aktualisiert. Mit einem Archive Tag, werden alte Einträge archiviert und neue hinzugefügt. Durch die Archivierung ist es möglich, eine History anzubieten.

7.5 Benutzeränderungen

Das Rückgrat der Webapplikation ist die MySQL Datenbank. In der Datenbank werden alle gesammelten Daten abgespeichert und verwaltet. Die Daten werden von zwei Faktoren beeinflusst. Einerseits werden sie von einer Suchmaschine wie Shodan oder Google hinzugefügt. Andererseits werden die Daten auch von Benutzern bearbeitet. Die Problematik an diesem Scenario ist, wie mit Aktualisierungen und mit Benutzeränderungen umgegangen wird.

Folgende Probleme tauchten auf:

Benutzeränderungen
Benutzer sollen bestehende ICS Daten bearbeiten können. Diese Benutzeränderungen sollen jederzeit nachvollziehbar sein. Es soll ersichtlich sein wer, wann und wo Änderungen gemacht wurden.

Aktualisierungen und Ausgabe
Die Ausgabe von ICS Daten erwies sich nicht als trivial. Aktualisierungen werden durch Benutzer, aber auch durch Suchmaschinen gemacht. Um aber zu garantieren, dass die Aktualisierungen einer Suchmaschine, nicht die der Benutzer überschreiben, müssen die ICS Daten und die Benutzeränderungen individuell behandelt werden. Ein weniger guter Ansatz wäre nach einer Benutzeränderung, keine Informationen von einer Suchmaschine mehr zu akzeptieren. Mit diesem Vorgehen, verlieren wir aber wichtige Informationen, welche wir den Benutzern der Applikation zur Verfügung stellen wollen. Die realisierte Lösung behandelt ICS von Suchmaschinen und Benutzeränderungen parallel und somit unabhängig voneinander. Die Benutzeränderungen speichern nur die getätigten Änderungen. Bei der Ausgabe wird das ICS der Suchmaschine geladen und mit den Benutzeränderungen überschrieben.

Dieses Vorgehen lieferte uns folgende Vorteile:

- Aktualisierungen von Suchmaschinen können jederzeit erhalten und verarbeitet werden.
- Benutzeränderungen können jederzeit erhalten und verarbeitet werden.
- Benutzeränderungen sind jederzeit nachvollziehbar.
- ICS aus einer Suchmaschine sind jederzeit nachvollziehbar.
- Eine History ist mit wenig Aufwand möglich.
- Bei der Ausgabe sind die letzten Änderungen ersichtlich.
- An leeren Stellen werden die Standartwerte von einer Suchmaschine wie Shodan oder Google gesetzt.

8. Schlussfolgerungen

Entstanden ist eine moderne und einfach zu bedienende Web Applikation. Damit aber ICS ThreatMap einen Mehrwert bringt, müssen die Suchfilter von den Benutzern aktuell gehalten werden. Das bedeutet, dass die Recherchen nach Suchfiltern von den Benutzern weitergeführt werden müssen. Eine gute Filterliste liefert auch die kritischen ICS Kontrollsysteme, die es zu schützen gilt. Wie die Ergebnisse zeigen, unterstützt uns auch Shodan mit Suchfilter, die laufend in unsere Filterliste aufgenommen werden müssen. Aber auch Quellen wie scadastrangelove.org beschäftigen sich mit kritischen Anlagen.

In Zukunft werden immer neue Produkte für ICS/SCADA Kontrollsysteme auf den Markt kommen. Die Produkte weisen Muster im Web Interface oder im Protokoll-Header auf, nach denen gezielt gesucht werden kann. Diese Produkte müssen laufend ausfindig gemacht und mit Suchfiltern gesucht werden.

Die Applikation bietet eine Basis, um gefundene ICS Systeme abzuarbeiten. Dabei sollen betroffene Firmen kontaktiert und die getätigten Massnahmen protokolliert werden. Mit unserem Trouble Ticket System können die Massnahmen laufend überprüft werden. Damit sich die Sicherheit in der Schweiz längerfristig verbessern kann, muss auch ein Team dahinterstecken, das bereit ist, die kritischen ICS abzuarbeiten. Da viele Kontrollsysteme hinter einem Provider stecken, sind die gegebenen Informationen nicht immer zu 100% zuverlässig. Auch die Geolocation-Daten sind mit Vorsicht zu geniessen.

Für die Realisierung war der Einsatz eines PHP Frameworks eine gute Wahl. Zend Framework 2 basiert auf dem Konzept von MVC und es kann objektorientiert programmiert werden. Dieses Framework würden wir in weitere Projekte wieder einsetzen. Viele Funktionen wie eine erweiterte Benutzerverwaltung mussten ausprogrammiert werden, auch wenn diese wenig mit der gegebenen Problemstellung zu tun hatte. In einem nächsten Projekt würden solche Funktionen schneller realisiert werden, da bereits eine Vorlage besteht.

Wie bereits erwähnt, ist eine erste Applikation für die Bekämpfung von kritischen Anlagen vorhanden. Der Betrieb von ICS ThreatMap wird nun zeigen, ob die Sicherheit der Kontrollsysteme in der Schweiz verbessert werden kann.

9. Abbildungsverzeichnis

ICS ThreadMap - v1.0

Glossar
(GLO)

Benjamin Kehl
Dominique Sorg

1. Glossar

Begriff	Beschreibung
CronJob	Der CronJob bzw. der CronDaemon startet automatisch Skripte oder Programme zu den gegebenen Zeiten.
CRUD	Steht für Create Read Update und Delete.
CSS	Steht für Cascading Style Sheets und ist eine deklarative Sprache für Stilvorlagen von strukturierten Dokumenten (z.B. in HTML oder XML).
DB	Abkürzung für Database.
DCS	Steht für Distributed Control System und wird für in jegliche Arten von Industrieprozessen wie Öl, Gas, Wasser, Chemie usw. verwendet.
DRY	Abkürzung für Don't Repeat Yourself. DRY ist ein Prinzip und besagt, dass Redundanz in Codes vermeidet werden sollte (keine Duplizierungen).
ERP	Enterprise-Resource-Planning (ERP) bzw. Unternehmensressourcenplanung bezeichnet die unternehmerische Aufgabe, die in einem Unternehmen vorhandenen Ressourcen (z.B. Kapital, Betriebsmittel oder Personal) möglichst effizient für den betrieblichen Ablauf einzusetzen und somit die Steuerung von Geschäftsprozessen zu optimieren.
HMI	Unter Human Machine Interface (dt. Benutzerschnittstelle) versteht man Produkte, die neben der Bedienung der Maschine auch das Beobachten oder Eingreifen in den Prozess erlauben.
HTML	Steht für Hyper Text Markup Language. Sie dient zur Strukturierung von Inhalten in Webseiten.
ICS-CERT	*Industrial Control Systems Cyber Emergeny Response Team* ist eine internationale Organisation, bestehend aus IT-Sicherheitsfachleuten, die sich mit der Lösung von konkreten IT-Sicherheitsvorfällen befassen und als Koordinatoren mitwirken.
ISO 9126	ISO steht für International Organization for Standardization und ist verantwortlich für Standardisierungen in verschiedensten Bereichen. Das ISO 9126 ist eine empfohlene Richtlinie von ISO, die sich speziell auf Qualitätsmerkmale im Bereich Software richtet.
JavaScript	JavaScript ist eine Skriptsprache auf der Client-Ebene und dient der Entwicklung von dynamischen HTML in Webbrowsern.
JQPLOT	JQPLOT ist ein auf javascript-basiertes Plugin, das Grafiken und Charts auswerten und darstellen kann.
JQuery	JQuery ist eine freie JavaScript-Bibliothek.
KISS	Steht für Keep it Simple und ist ein Prinzip in der Programmierung, dass besagt, man solle seine Lösung so einfach wie möglich umsetzen.

MELANI	Die Melde- und Analysestelle Informationssicherung befasst sich ebenfalls mit IT Security, jedoch sind ihre Tätigkeiten auf die Schweiz gerichtet.
Member	Akteur mit Leserecht.
MVC	Steht für Model View Controller und ist ein Pattern, welches für die Strukturierung von Software-Entwicklung weit verbreitet ist. Das MVC unterteilt die Software in die drei Einheiten Model, View und in Controller.
MySql	MySql ist eine populäre Open-Source Datenbank.
Operator	Akteur mit Lese- und Schreibrecht.
PHP	Steht für Hypertext Preprocessor und ist eine Skriptsprache zur Erstellung von dynamischen Webseiten.
PLC	Steht für Programmable Logic Controller (dt. speicherprogrammierbare Steuerung) und ist ein Gerät, das zur Regelung oder Steuerung einer Maschine eingesetzt wird.
Responsive Webdesign	Unter dem Begriff Responsive Webdesign versteht man einen gestalterischen und technischen Vorgang für die Erstellung von Webseiten. Dabei reagiert es auf Eigenschaften des jeweils benutzten Endgeräts. Der Aufbau der Webseite basiert dann auf den Anforderungen des jeweiligen Gerätes wie Mobile, Tablet, PC, etc. Dies betrifft vor allem Elemente wie die Navigation, Aufteilung von Seitenspalten und Text, wie Anreihungen von Bildern. Responsive Webdesign kann mit den Webstandards HTML5, CSS3 und JavaScript realisiert werden. Ausserdem gibt es auch diverse Frameworks wie beispielsweise Bootstrap von Twitter.
RUP	Abkürzung für Rational Unified Process. RUP ist ein Vorgehensmodell in der Softwareentwicklung.
SCADA	Ist ein Computer System und steht für Supervisory Control and Data Acquisition (SCADA). In SCADA-Systeme können technische Prozesse überwacht und gesteuert werden. SCADA ist eines der grössten Teilbereiche von industriellen Kontrollsystemen (ICS).
Stuxnet	Stuxnet ist ein Schadprogramm das speziell in bestimmten Systeme wie SCADA entwickelt wurde.
TableGateway	Ist ein Design Pattern, welches ein Objekt als eine Tabelle mit ihren Attributen in einer Datenbank repräsentiert.
TTS	Abkürzung für Trouble Ticket System.
UC	Abkürzung für Use Case.

UPS	Eine unterbrechungsfreie Stromversorgung (USV), englisch Uninterruptible Power Supply (UPS), wird eingesetzt, um bei Störungen im Stromnetz die Versorgung kritischer elektrischer Lasten sicherzustellen. Zu unterscheiden hierzu ist die allgemeine Ersatzstromversorgung (AEV, auch als "Netzersatzanlage" bezeichnet), da diese bei der Umschaltung eine kurze Unterbrechung der Stromversorgung aufweist.
Whois	Ist ein Protokoll, mit dem Informationen zu Internet-Domains, IP-Adressen und deren Eigentümern abgefragt werden können.
Zend Framework 2	Zend Framework 2 ist ein open-source-Framework für PHP.
ZF2	Abkürzung für Zend Framework 2.
ICS	Steht für *Industrial Control Systems* und ist ein allgemeiner Begriff, der mehrere Typen von Kontrollsystemen in Industriegebieten beschreibt.

Literaturverzeichnis
(LIV)

BABS, B. f. (2013). *Schweizer Eidgenossenschaft.* Abgerufen am 12. 10 2013 von http://www.bevoelkerungsschutz.admin.ch/internet/bs/de/home/themen/ski.html

Berg, S. J. (kein Datum). *Syssec.* Von http://www.syssec.rub.de/media/emma/arbeiten/2012/11/16/2011-12-05-MA-Berg.pdf abgerufen

Browsershots. (2011). Abgerufen am 12. 12 2013 von http://browsershots.org/

Browserstack. (2013). Abgerufen am 12. 12 2013 von Browserstack: http://www.browserstack.com/

Eggert, R. (2013). *Zend Framework 2 - Das Praxisbuch.* (S. Mattescheck, Hrsg.) Galileo Press Computing.

Einführung in JSON. (2013). Abgerufen am 16. 09 2013 von JSON: http://www.json.org/

Google Address Finder. (2013). Abgerufen am 12. 10 2013 von Google: http://ctrlq.org/maps/address/?address=

Google Filter. (2013). Abgerufen am 5. 11 2013 von Google: https://support.google.com/websearch/answer/136861?hl=en

Google Filter. (2013). Abgerufen am 08. 10 2013 von Google: http://www.googleguide.com/advanced_operators_reference.html

Google Mapgs API. (2013). Abgerufen am 16. 09 2013 von Google: https://developers.google.com/maps/

Haederli, F. I. (01. 12 2013). *SonntagsZeitung*. Von SonntagsZeitung: Von http://www.sonntagszeitung.ch/fokus/artikel-detailseite/?newsid=268454 abgerufen

Integraxor. (17. 12 2013). *Demo HMI Integraxor*. Abgerufen am 17. 12 2013 von http://www.integraxor.com/demo.htm

Mixabest. (7. March 2008). File:MITSIBISHI PLC Panel.jpg. *PLC*. doi:This file is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported license.

Mixabest. (18. 12 2012). File:MITSIBISHI PLC 04.png. *PLC*. doi:This file is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported license.

MySQL Download. (2013). Abgerufen am 16. 09 2013 von MySQL: https://dev.mysql.com/downloads/mysql

PHP Manual. (2013). Abgerufen am 16. 09 2013 von PHP: http://php.net

Pixabay. (kein Datum). Autos Technik Vw Parkhaus Lagerhalle Fahrzeuge. doi:Public Domain CC0 https://creativecommons.org/publicdomain/zero/1.0/deed.de

Pixabay. (kein Datum). Kernkraftwerk Zentrale Dampf Energie Non. doi:Lizenz Public Domain CC0 https://creativecommons.org/publicdomain/zero/1.0/deed.de

Pixabay. (kein Datum). Segovia Dam Ponton Hochwasserentlastung Bauarbeiten. doi:Public Domain CC0 https://creativecommons.org/publicdomain/zero/1.0/deed.de

Pixabay. (kein Datum). Solarzellen Energie Strom Umweltfreundlich. doi:Public Domain CC0 https://creativecommons.org/publicdomain/zero/1.0/deed.de

Speed Test Messung. (2013). Abgerufen am 18. 12 2013 von Pingdom Tools: http://tools.pingdom.com

Suche.ch. (2013). Abgerufen am 20. 10 2013 von Das Schweizer Internet Portal: www.suche.ch

Wikipedia SCADA. (2013). Von Wikipedia: https://de.wikipedia.org/wiki/Supervisory_Control_and_Data_Acquisition abgerufen

ICS ThreadMap - v1.0

Technologie- & Filteranalyse (TFA)

Benjamin Kehl
Dominique Sorg

Änderungsgeschichte

Datum	Version	Änderungen	Autor
25.11.2013	0.1	Erstellung, ZF2 & Bootstrap, Filter (Google/Shodan), Fachsymposium, Filterliste	Dominique Sorg
10.12.2013	0.2	Überarbeitung, Einleitung Was ist ICS/SCADA, Wo sind die Gefahren, Wie funktionieren die Kontrollsysteme	Dominique Sorg
19.12.2013	0.3	Anpassungen Formatierungen	Benjamin Kehl

Inhalt

1. Einführung

1.1 Zweck

Dieses Dokument beschreibt die Analyse verschiedenster Technologien und die Zusammensetzung von Suchfilter um gezielt nach ICS zu suchen.

1.2 Gültigkeitsbereich

Die Technologie Analyse ist über die gesamte Projektdauer gültig. Damit das Dokument immer auf dem neusten Stand ist, wird es laufend angepasst.

1.3 Übersicht

Dieses Dokument liefert eine Analyse über eingesetzte Technologien sowie externen Ressourcen. Dabei werden Begründungen geliefert weshalb gewisse Technologien oder Dienste für unsere Studienarbeit eingesetzt werden.

2. Datenbeschaffung durch Filter

Die Datenbeschaffung, wie sie detaillierter in der Software Architektur beschrieben ist, wird über externen Quellen wie Shodan oder Google abgefragt. Durch recherchierten und selbst erstellten Filtern kann spezifisch nach den gewünschten ICS/SCADA Systeme in der Schweiz gesucht werden. Die Filter werden durch mehrere Schlüsselbegriffe, die jeweils die ausgewählte Suchmaschine unterstützt, zusammengesetzt und ausgeführt. In den folgenden Seiten beschreiben wir die beiden Suchmaschinen Google & Shodan und zeigen die Möglichkeiten auf wie nach spezifischen ICS gesucht werden kann.

2.1 Google Filter

2.1.1 Grundlage

Google ist der deutliche Marktführer der Suchmaschinen im Internet. Jeder der heutzutage etwas im Internet sucht benutzt Google. Doch Google bietet weit mehr als nur eine einfache Suche. Durch gezielt gesetzte Suchbegriffe kann die Suchresultate extrem eingedämmt werden. Diese extrem spezifischen Suchmöglichkeiten können wir für unser Web Applikation einsetzen um gezielt nach ICS zu suche die öffentlich im Internet erreichbar sind.

Die folgende Tabelle zeigt eine kurze Referenz* von Suchfilter bei Google:

Filter	Beschreibung
allinanchor:	Return only pages in which the anchor texts **on links** to the pages contain the words.
allintext / intext:	Return only pages in which the **words appearing in the text** of the page.
allintitle / intitle:	Return only documents that **contain the words in the title**.
allinurl / inurl	Return only results to documents that **containing that word in the URL.**
site:	Search within a specific site.
filetype:	Search by file type.
relate:	Find related pages.
1..100	Search numbers in a **range (1 to 100)**
hsr 8640 map	**Search by location (8640)** – only ZIP, Search locations by zip and area codes

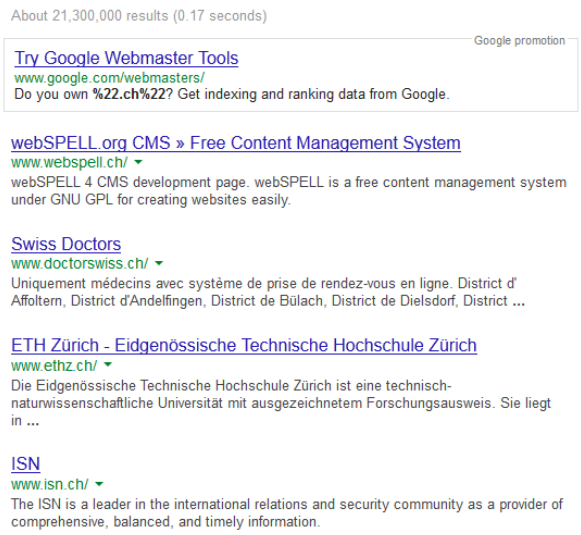
*Im Quellenverzeichnis befindet sich ein paar Links zu weiteren Google Filtern.

Wichtig ist dabei zu beachten, dass Google eine Inhalt und URI/URL basierte Suchmaschine ist, die in verschiedensten Web-Domänen sucht.

2.1.2 Beispiele

Mit dem folgenden Suchfilter kann über Google nach Domänspezifischen Seiten gesucht werden. Dieser setzt sich mit dem Schlüsselwort ‚site' und dem country code des jeweiligen Land zusammen.

site:".ch"

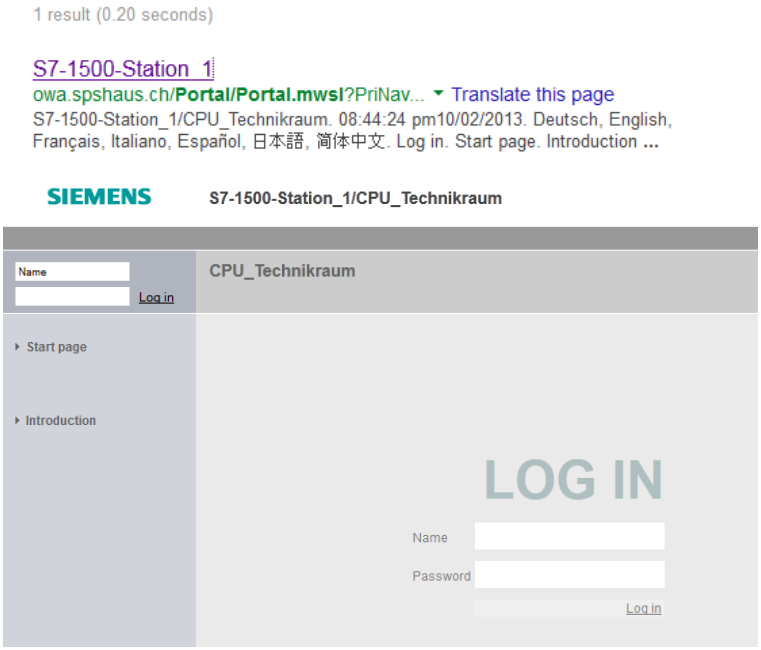


Ein weiteres Beispiel illustriert die Suche nach einem Siemens ICS Produkt welches über Google auffindbar ist. Dabei wird angenommen dass die ICS Produkte eines Herstellers Muster aufweisen, nach denen gesucht werden kann. Diese Muster können durch das Logo des Herstellers, die allgemeine Applikation Struktur oder durch den Seitenaufbau des Web Interface auftreten, die vom Kunden nicht verändert werden kann. Mit dem Pattern „/Portal0000.htm“ wird eine solche Suche nach Siemens ICS Web Interfaces gestartet die in der URL die gleiche Endung wie das Pattern aufweisen.

inurl:/Portal0000.htm

Noch ein Beispiel für ein ICS Produkt, welches sich in der Schweiz befindet.

inurl:Portal/Portal.mwsl site:".ch"

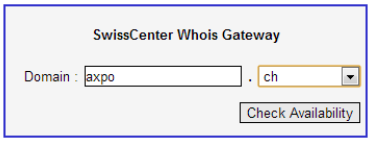


2.1.3 Google und ICS ThreatMap

Über Google lässt sich durchaus ICS finden, welche nicht in anderen Diensten zu finden sind (siehe nächstes Kapitel Shodan). Da Google nach Inhalte, URI/URL oder Dokumenten in Web Domänen sucht, ist dieser eine wichtige Ressource für das Auffinden von öffentlich zugänglichen ICS/Scada Systeme für unsere Applikation.

Die Daten aus Google müssen jedoch selbst gesucht und zusammengetragen werden. Ausserdem liefert Google keine GEO Daten, die in unsere Threat Map eingebunden werden können. Dazu müssen wir auf externe Dienste wie bspw. Whois.com zugreifen um die Kontaktdaten eines Servers abzufragen. Die Adresse kann anschliessend mit dem Google Address Finder in Koordinaten umgewandelt werden.

Whois Daten sind jedoch nicht immer zuverlässig. Da Systeme hinter einem Provider die Kontaktadresse des Providers liefert, statt dieser des Besitzers des ICS.



Le domaine **axpo.ch** n'est pas disponible

whois: This information is subject to an Acceptable Use Policy.
See http://www.nic.ch/terms/aup.html

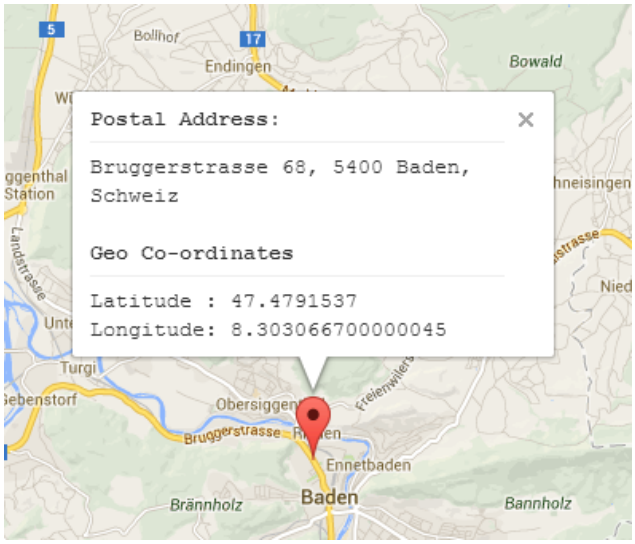
Domain name:
axpo.ch

Holder of domain name:
Axpo Informatik AG
SÃ¤uberli Daniel
E-Mail: webmaster@axpo.ch
Bruggerstrasse 68
CH-5400 Baden
Switzerland
Contractual Language: German

Technical contact:
Axpo Informatik AG
SÃ¤uberli Daniel
E-Mail: webmaster@axpo.ch
Bruggerstrasse 68
CH-5400 Baden
Switzerland

DNSSEC:N

Name servers:
ns1.axpo.com
ns2.axpo.com



2.2 Shodan Filter

2.2.1 Grundlage

Shodan ist eine kostenpflichtige Alternative zu Google. Im Gegensatz zu Google, sucht Shodan aber nicht nach Inhalte und URI/URL sondern nach HTTP Headers auf verschiedensten Ports. Die Daten werden bei Shodan Indexiert und werden von Beginn weg mit Kontaktdaten und Geodaten angereichert. Dabei sind die Daten auch bei Shodan mit Vorsicht zu geniessen, da diese ebenfalls falsche Informationen liefern kann. Auch hier würde das Beispiel passen, wenn ein ICS Betreiber hinter einem Provider sitzt, werden nicht die Kontaktdaten des Betreibers abgefragt sondern die des Providers.

Generell können die Suchfilter bei Shodan folgendermassen angewendet werden:

- Much like Google and other search engines, SHODAN also lets you use boolean operators ('+', '-' and '|') to include/ exclude certain terms. By default, every search term has a '+' operator assigned to it.

- In addition to boolean operators, there are special filters to narrow down the search results

- **General**
  All filters have the format 'filter:value' and can be added anywhere in the search query.
  Notice that there's no space before or after the ':'.

Shodan bietet folgende Filter für eine spezifischere Suche.

city	- Apache servers located in Zürich: apache city:"Zürich" - Nginx servers located in San Diego, USA: nginx city:"San Diego" country:US

country	- Apache servers located in Switzerland: apache country:CH - Nginx servers located in Germany: nginx country:DE
geo	- Apache servers near 42.9693,-74.1224: apache geo:42.9693,-74.1224 - Devices within a 50km radius of San Diego (32.8,-117): geo:32.8,-117,50
hostname	- GWS with 'google' in the hostname: "Server: gws" hostname:google - Nginx with '.de' in the hostname: nginx hostname:.de
net	- All data for IP 216.219.143.14: net:216.219.143.14 - All data in the subnet 216.219.143.*: net:216.219.143.0/24 - All data in the subnet 216.219.*: net:216.219.0.0/16 - Apache servers in the subnet 216.*: apache net:216.0.0.0/8
os	- Microsoft-IIS running on Windows 2003: microsoft-iis os:"windows 2003" - JBoss running on Linux: JBoss os:linux
port	- Look only at the FTP banners for ProFTPd: proftpd port:21
before/ after	- Nginx server banners found before January 18 2010: nginx before:18/01/2010 - Apache servers in Switzerland found between March 22 2010 and June 4 2010: - apache country:CH after:22/03/2010 before:4/6/2010

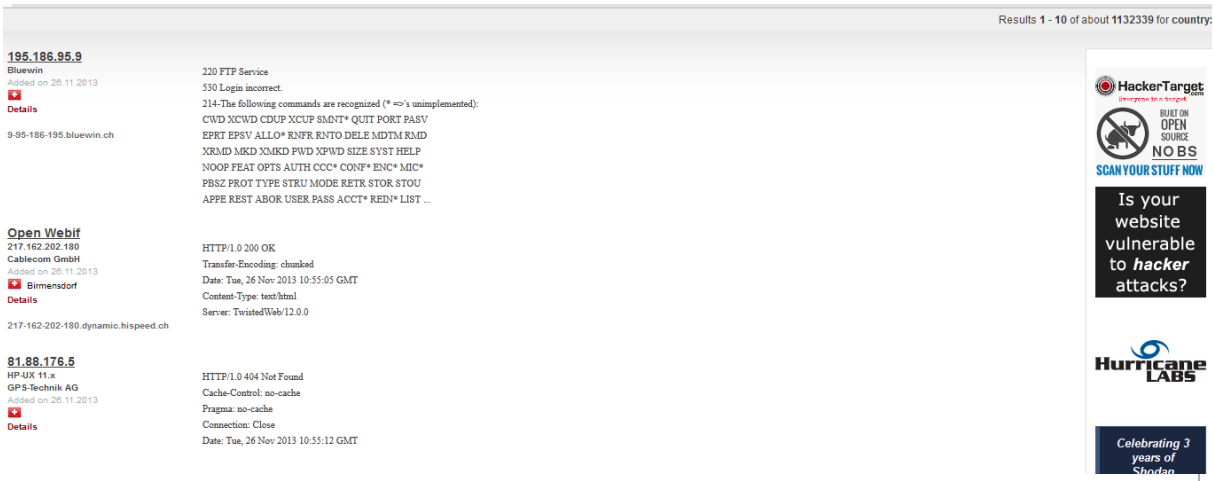
2.2.2 Beispiele

Wie bereits im Projektplan in der Rubrik Projekt Übersicht beschrieben wurde, kann in Shodan nach den Begriffen ICS und deren Kontrollsystem-Typen gesucht werden. Die folgende Tabelle zeigt den Stand vom 26.11.2013. Mittlerweile hat Shodan seine Suchfilter ICS, BMS, ERP, HMI, PDU, PLC, PLCND, SCADA, UPS angepasst und liefert genauere und bessere Suchresultate.

ICS Suchbegriffe	Filter	Anzahl Resultate
ics	ics country:ch	8
bms	bms country:ch	4
erp	erp country:ch	14
hmi	hmi country:ch	1
pdu	pdu country:ch	48
plc	plc country:ch	66
plcnd	plcnd country:ch	0
scada	scada country:ch	0
ups	ups country:ch	3

Um in Shodan nach Einträgen aus der Schweiz zu suchen kann folgender Befehl helfen.

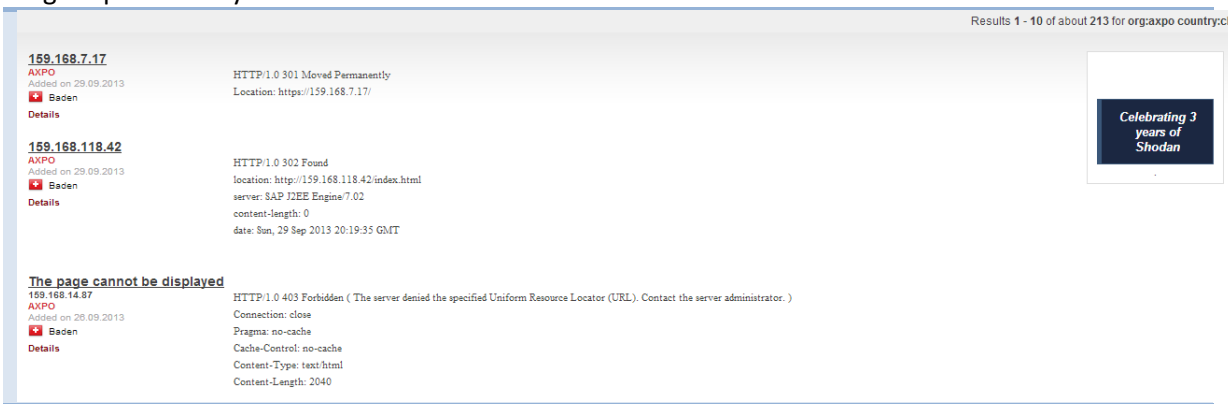
country:ch



1'132'339 Resultate

Um in der Schweiz nach einer Organisation suchen zu können hilft folgender Befehl.

org:"axpo" country:ch



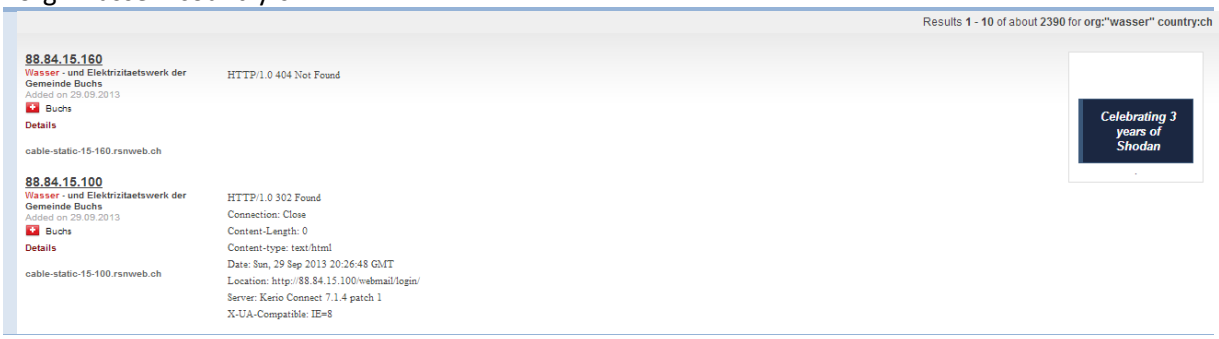
213 Resultate

Um nach weiteren Schweizer Elektrizitätswerke, Wasserkraftwerke, Wasserversorgung, Stromversorgungen, etc. in Shodan suchen zu können, hilft www.suche.ch weiter:

- suche.ch ist eine der meistbesuchten Suchmaschinen der Schweiz. Ideal für die Suche nach Firmen, Branchen, Produkten, Marken und Dienstleistungen.
- suche.ch ist eine Dienstleistung der CREA SWISS AG.

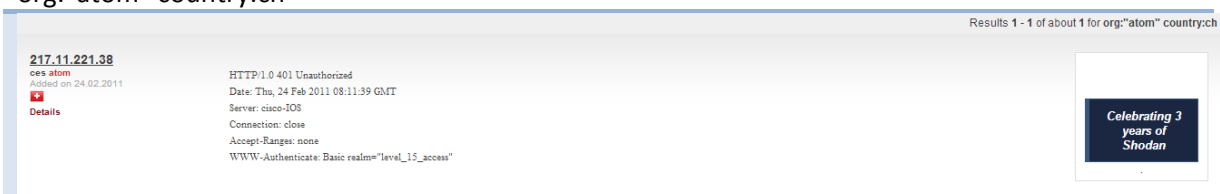
Es kann auch nach Begriffen im Firmennamen in Shodan gesucht werden:

org:"wasser" country:ch



2'390 Resultate

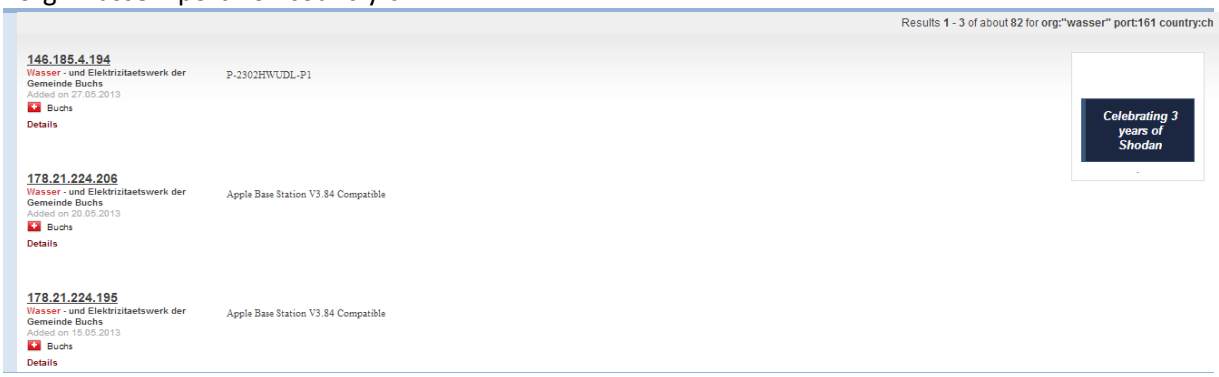
org:"atom" country:ch



1 Resultat

Dieses Beispiel zeigt Resultate die im Firmenname das Wort „Wasser" enthalten und den Dienst SNMP laufen haben.

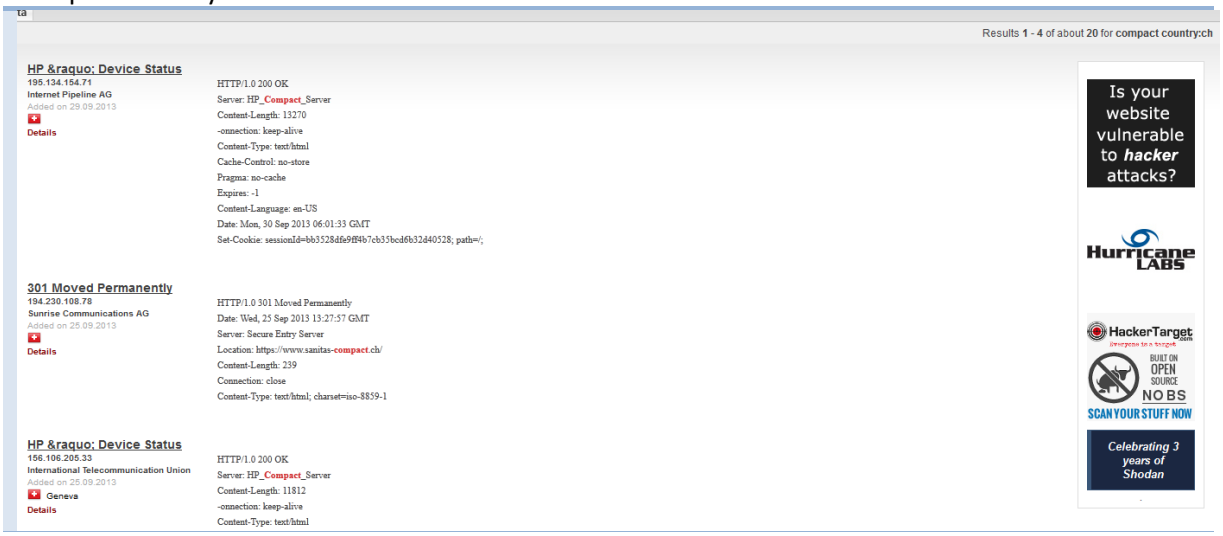
org:"wasser" port:161 country:ch



82 Resultate

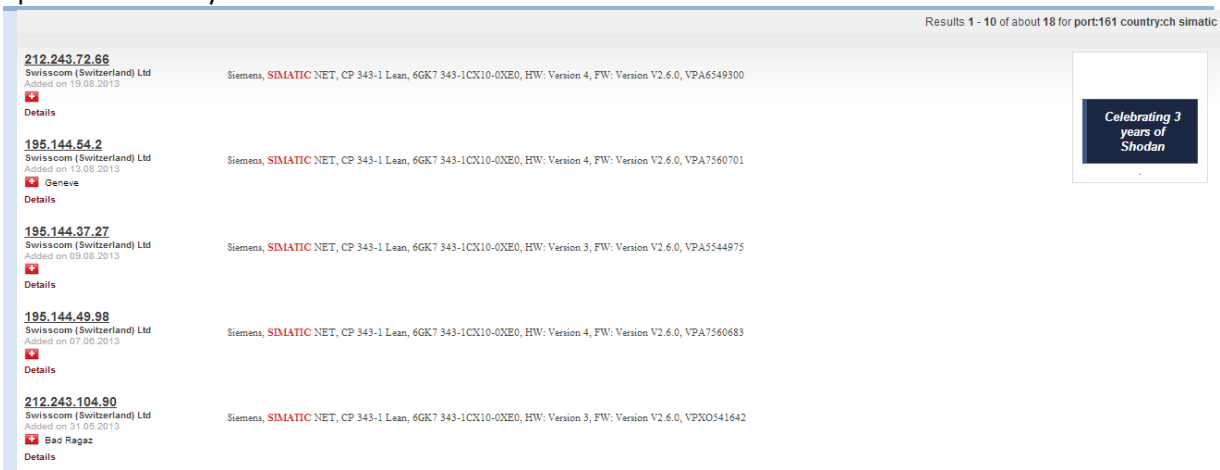
Diese Suche illustriert die Suche nach einem spezifischen Produkt

compact country:ch



20 Resultate

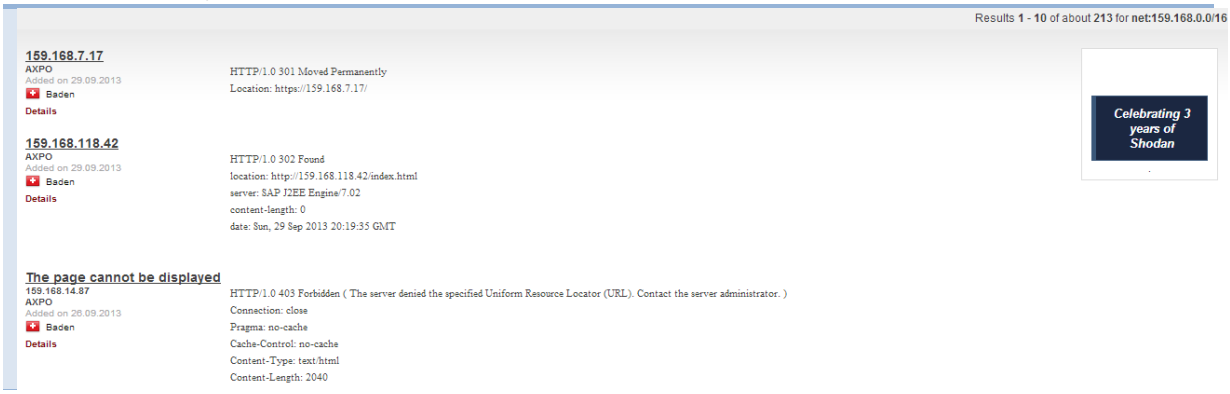
port:161 country:ch simatic



18 Resultate

Es ist auch möglich einen gesamten IP Range abzufragen. In diesem Beispiel wird dies am Beispiel von Axpo demonstriert.

net:159.168.0.0/16



213 Resultate

2.2.3 Shodan und ICS ThreatMap

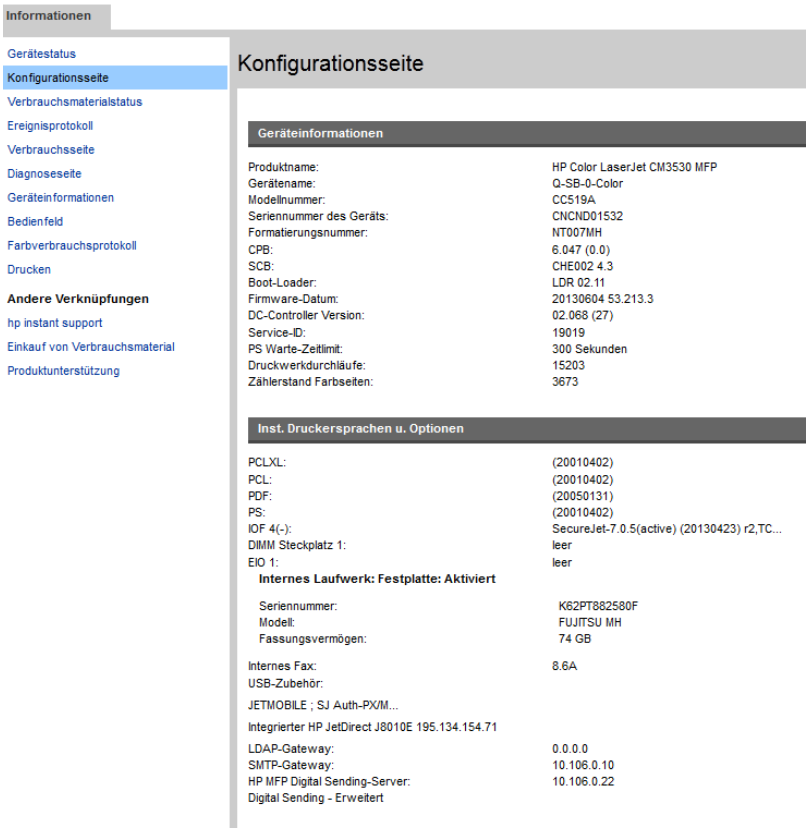
Shodan liefert genauere Einträge als Google es tut, da dieser die Information über HTTP Request holt. Des Weiteren kann mit wenig Aufwand einfache und spezifische Suchfilter zusammengesetzt werden. Durch Shodan sind viele ICS auffindbar mit einem Login Schutz, teils auch nur mit Default Login Daten ausgestattet, die für einen potenziellen Hacker ausgenutzt werden kann. Hinter diesen Login Bereichen stehen ganze Firmen Infrastrukturen (ICS/SCADA).

Aus Online News wird laut diversen Studien ersichtlich, dass auch ein Geschütze Applikation nicht vor Angriffen bewahrt wird. Dabei können Angriff Szenarien folgendermassen aussehen.

- → Angriffs-Scenario: Hacker mietet mit gefälschten Daten Server / Workstation bsp. in Kongo, China, usw. und macht Dictionary/Brute Force Attacken auf das Login Bereich (evt. über Tor)
- → Angriffs-Scenario: Hacker mietet ein Botnet-Dienst und startet Dictionary/Brute Force Attacke auf Login
- → Angriffs-Scenario: Hacker sucht nach Vulnerabilitäten eines Gerätes z.B. in National Vulnerability Database, CVE, Security Focus, Metasploit und führt gezielte Angriffe aus.

Ausserdem liefert Shodan auch diverse andere ungeschützte System wie bspw. Drucker, Router von denen aus weitere Informationen für das Auffinden von ICS Systeme genutzt werden können. Wir beschränken uns aber in dieser Arbeit auf ICS Systeme!

Beispiel eines Druckers liefert interne Konfigurations-Informationen



3. Gesammelte Filterliste

Die Web Applikation ICS ThreatMap lebt von Ihrer dynamischen Zunahme von Suchfilter aus Shodan oder Google. Diese Filter werden laufend von Nutzer wie Bund, MELANI und Sicherheitsfirmen hinzugefügt und verwaltet, da sich ständig neue Hersteller mit neuen Produkte auf dem Markt etablieren wollen. Die Produkte müssen untersucht und Suchfilter für Shodan, Google und Co. erstellt werden. Für die Entwicklung sammelten wir erste mögliche Filter. Statische Funde werden nicht als Filter erfasst sondern direkt als ICS in der Applikation eingetragen.

Wir trugen eine Tabelle von ICS Hersteller und deren Produkte zusammen. Diese kann dann weitergeführt werden. Unteranderem wurde die Tabelle auch Dank von http://scadastrangelove.blogspot.ch/ oder http://www.shodanhq.com/ zusammengetragen.

Vendor	Product
ABB	RTU500
ADCON	A850 Telemetry Gateway
	addUPI-OPC Server
AKCP	
Alcea	
Allen-Bradley	
Allied Telesis	
APC	

Barik	
Beck IPC	IPC@CHIP
BroadWeb	
Caterpillar	
Cimetrics	Eplus - B/IP to B/WS Gateway
CIMON	
Control4	
CODESYS	WebVisu
Clorius Controls	
Datawatt	
Delta Controls	enteliTOUCH
Digi	
Ecessa	
Echelon	i.LON 600
Ericsson	
Emerson	
EIG	
Electro Industries GaugeTech	
Elster EnergyICT	RTU
	eiPortal
EnergyICT	
Falcon	
Force10	
Funkwerk	
Fujitsu	ServerView
GE	
Genohm	
General Electric	Cimplicity
	Proficy
Hirschmann	
Honeywell	
HMS	EtherNet/IP / Modbus-TCP Interface
Liebherr	
LOGPAC	
Itron	
Koyo	
KMC	
Komatsu	
Lennox	
Leica	
Lancom	
Lantronix	
Moxa	
LonWorks	
LG	
Mitsubitshi	
Motorola	
Moxa	ioLogik
Niagara	
National Instruments	

NRG Systems	WindCube
Novatech	
OMRON	
openSCADA	
Ourman	
Phoenix Contact	
Phillips	
mGuard	
Schneider Electric	CitectSCADA
	Modicon
	PowerLogic ECC
	PowerLogic EGX
	PowerLogic ION
	PowerLogic PM
	Tac XENTA 913
Schleifenbauer	SPbus gateway
Siemens	Scalance S
	Scalance W
	Scalance X
	Simatic HMI
	Simatic NET
	Simatic S7
Rabbit	
Reliance	Reliance 4 SCADA/HMI system
RUGGEDCOM	
Rockwell Automation	Micrologix
RTS Services	
Powertech	
SAP	NetWeaver Application Server
STULZ	
SoftPLC	
Somfy	
SpiderControl	
THUS	
Telemecanique	
Trend	IQ3xcite
Tridium	
VxWorks	
Wago	
WindWeb	
Wind River	
Wonderware	

3.1 Filterliste Google

Vendor	Product	Google Dork
Siemens	S7-3** , PCS7	inurl:/Portal0000.htm
	Siemens S7	inurl:"Portal/Portal.mwsl"
	Simatic HMI	intitle:"Miniweb Start Page" | "/CSS/Miniweb.css"

Allan-Bradley Rockwell Automation	CompactLogix	intitle:"Rockwell Automation" "Device Name" "Uptime"
	SLC5	inurl:dtm.html intitle:1747-L552 inurl:dtm.html intitle:1747-L551
	MicroLogix	inurl:home.htm intitle:1766
Schneider Electric	Modicon	intitle:"Quantum CPU Web Server"
	Quantum/Premium/Micro	intitle:"Premium CPU Web Server"
	CitectSCADA	intitle:"Citect Web" inurl:scada
	ClearSCADA	intitle:"ClearSCADA Home"
General Electric	Cimplicity	intitle:"CIMPLICITY WebView" inurl:main.html
	Proficy	inurl:ProficyPortal/default.asp

3.2 Filterliste Shodan

1.1-rr-std-b12 port:80
A850 Telemetry Gateway
ABB Webmodule
addUPI Server
AKCP
AKCP Embedded Web Server
Allen-Bradley
BroadWeb
Cimetrics Eplus Web Server
CIMPLICITY-HttpSvr
CitectSCADA
ClearSCADA
Console terminal type
EIG Embedded Web Server
eiPortal
Embedded Web Server Version 200 OK
EnergyICT
EnergyICT RTU
enteliTOUCH
GoAhead-Webs InitialPage.asp
helmholz
HMI, XP277
HMS AnyBus-S WebServer
honeywell BNA
honeywell Excel
i.LON
ics
ioLogik Web Server
IPC@CHIP
IPC@CHIP title:Start
iq3
ISC SCADA Service HTTPserv:00001
Jetty 3.1.8 (Windows 2000 5.0 x86)
Jetty 3.1.8 (Windows 2000 5.0 x86) '200 OK'
Location: Default.html -apache -nginx -microsoft -etag -goahead -vxworks -jetty -GoAhead 302 -Cookie
MC Works64
Micro-AT
Micrologix
Modbus Bridge

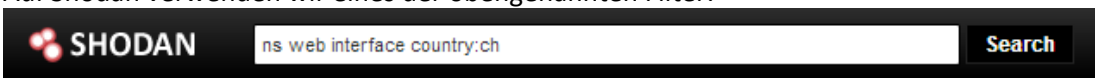
ModbusGW
Model name : 5232-N port:23
modicon
Modicon M340
Modicon M340 CPU
Moxa
MoxaHttp
NET ARM Web Server/1.00
niagara 200
Niagara Web Server
niagara_audit
niagara_audit -login
NovaTech HTTPD
ns web interface
openerp
openerp server:CherryPy
PLC
port:161 simatic
port:161 SLC5
port:21 'CJ2M-EIP21'
port:23 'Meter ION'
port:23 vxworks -login
Portal0000
Portal0000.htm
Power Measurement Ltd
Power Measurement Ltd ION8650
powered by SpiderControl TM
Powerlink
PowerLogic ECC
PowerLogic EGX
PowerLogic PM800
ProficyPortal
Reliance 4 Control Server
rockwell 1756
Rockwell Automation
RTS SCADA Server
RuggedCom
S7-200
S7-300
samsung Data Management Server
SAP NetWeaver Application Server
scada
scada port:80
Scalance S
Scalance W
Scalance X
Schleifenbauer
Schleifenbauer SPbus gateway
schneider
Schneider Electric
Schneider Electric ECC21
Schneider Electric EGX100MG
Schneider Electric PM820SD
Schneider Electric PM870SD

SCHNEIDER TSX ETG3021
Schneider-WEB
Series C Revision
Server: eCos Embedded Web Server title:'Danfoss Solar Inverters'
Server: VTS -IIS -Apache -nginx 401 -500 -Boa -Sitewatch -Apple -httpd -cpsrvd -Ubicom -DCS-6620
Server: VTS' -IIS -Apache -nginx 401 -500 -Boa -Sitewatch -Apple -httpd -cpsrvd -Ubicom -DCS-6620
server:iq3
serverview
Simatic -S7 HMI
Simatic -S7 -HMI
Simatic+S7
slc 505
SLC5
SoftPLC
SpiderControl
Stulz GmbH Klimatechnik
Tac XENTA 913
TAC/Xenta
TELEMECANIQUE BMX
THUS plc
title:adcon
title:alarm
title:EagleSDV
title:eSolar
title:ics
title:logic
title:phasefale
title:phasefale Z-World Rabbit
title:PowerLogic
title:scada
title:'Schneider Electric'
title:Somfy
TLP 700TV
VxWorks
VxWorks port:21 'logged in'
WAGO
webSCADA-Modbus
Webvisu
Welcome to the Windows CE Telnet Service on HMI_Panel
wince Content-Length: 12581
Wind River
WindRiver-WebServer
WindWeb
Z-World Rabbit 200 OK
Z-World Rabbit '200 OK' html:index.zht
FactoryCast

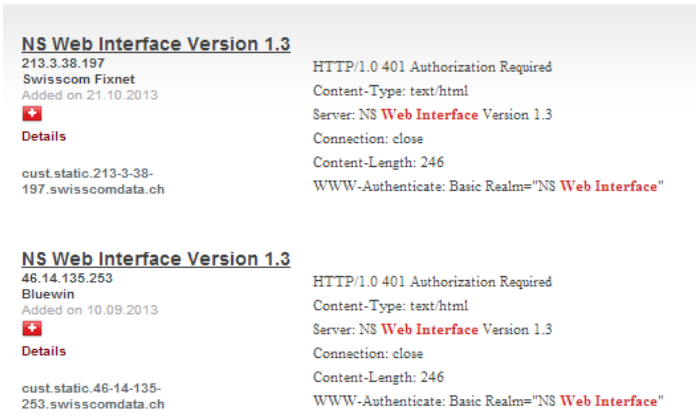
4. Vorgehensweise zum Auffinden von ICS Anlagen

In diesem Kapitel wird Beschrieben wie ein Angreifen Shodan nutzen könnte um ungeschützte ICS Anlagen in der Schweiz zu suchen.

1) Auf Shodan verwenden wir eines der obengenannten Filter:



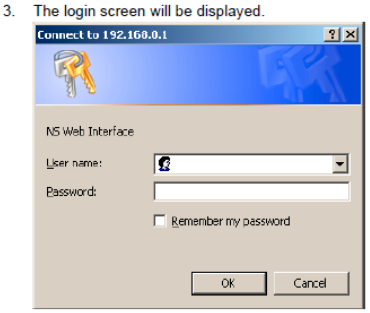
2) Shodan liefert uns am 07.12.2013 zwei Resultate:



3) Wie in der obigen Grafik ersichtlich ist, werden in den HTTP Headerdaten diverse Informationen mitgeliefert. Unter anderem Interessiert uns hier „WWW-Authenticate: Basic Realm="NS Web Interface"“.

4) Eine Suche nach den Standart Credential bei Google „NS Web Interface Version 1.3“ verweist uns direkt zum Handbuch (https://www.google.ch/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&ved=0CCsQFjAA&url=http%3A%2F%2Fforums.mrplc.com%2Findex.php%3Fapp%3Dcore%26module%3Dattach%26section%3Dattach%26attach_id%3D12467&ei=DtGiUtS2C8Oo0QWQx4GgCQ&usg=AFQjCNGKqJPnnAKLEyTRa7pqc_yieNFysw&sig2=PtspOs_ukMz4APAGWNxbzg&bvm=bv.57752919,d.d2k)

5) Im Handbuch wird detailliert beschrieben wie die Software eingesetzt wird. Unter anderem auch das Login. Dies ist aber nicht immer notwendig, da auch viele Anlagen am Internet angebunden sind, die keine Authentisierung verlangen. Ein junges Beispiel ist die ICS Anlage des Fussballstadiums in Basel, die während einem Jahr ohne Authentisierung von einem Hacker bedient werden konnte. Der Artikel kann unter folgenden Link gefunden werden. http://www.sonntagszeitung.ch/fokus/artikel-detailseite/?newsid=268454



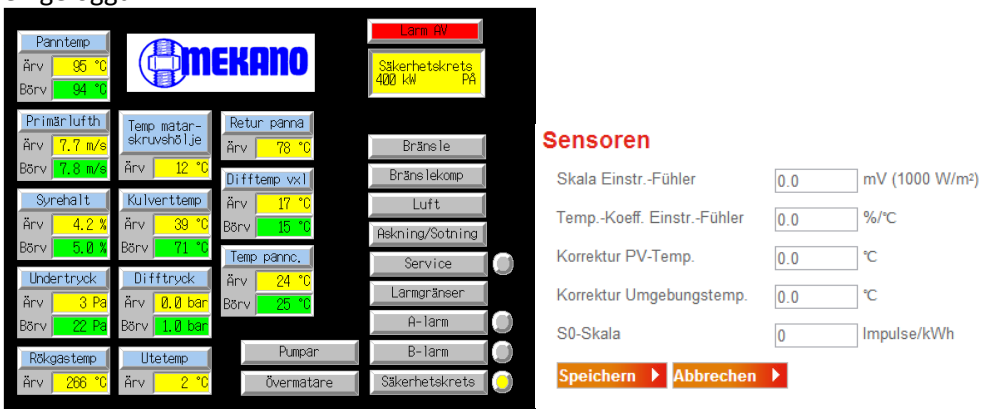
4. Enter the user name and password.

The factory settings for the user name and password are as follows.

User name	default
Password	default

6) Bis zu diesem Punkt konnten viel Informationen bereits beschaffen werden um auf eine ICS Anlage zuzugreifen ohne dabei direkt auf die Seite des Betreibers zuzugreifen.

7) Das Ergebnis bei einem Zugriff könnte folgendermassen aussehen. Aus Schutz wurden hier keine Systeme der obigen Suche verwendet. Es wurde sich auch nicht in die obigen Systeme eingeloggt.



5. Zend Framework 2 & Bootstrap

Bevor wir mit der Entwicklung der Web Applikation begannen, überlegten wir uns, welche Technologien wir für dieses Projekt einsetzen könnten. Dabei spielt es für unseren Betreuer selber keine grosse Rolle ausser das die Eingesetzten Technologien möglichst Open Source sein müssen. Schlussendlich entschieden wir uns für den Einsatz von Zend Framework 2 und für das Bootstrap Framework. Durch den Einsatz dieser Frameworks können wir uns auf die Problemstellung konzentrieren und müssen uns nicht mit Grundprobleme herumschlagen. Da dies unser erstes Projekt mit Zend Framework 2 ist, erwarten wir Mehraufwand für die Einarbeitung und für die Erstellung einzelner Komponente.

Die Anforderungen unserem Betreuer werden mit den folgenden Technologien gedeckt.

Technologie Anforderungen an ICS ThreatMap

- Open Source (Wo immer möglich)
- Betriebssystem unabhängig
- Browser unabhängig
- Installation Anweisung
- E-Mail Support für Statistik Report
- Datenbank

PHP (Hypertext Preprocessor)

- PHP ist eine Skriptsprache für Webentwicklung → schnelle Lernkurve
- Schnelle und einfache Installation
- Objektorientierung möglich
- Einfache Datenbankanbindung
- Geringe Serverbelastung (Vermeidung von vielen Systemcalls und neue Prozesse)
- Ist Browser unabhängig
- Betriebssystem unabhängig (Windows, Linux, …)
- PHP ist Open Source
- Sehr verbreitet: z.B. CMS (Yoomla!, TYPO3, Drupal,…), Blogs (Wordpress, Serendipity), Datenbankverwaltung (phpMyAdmin, PhpPgAdmin)
  → Grosse Community

Was ist ZF2 (Zend Framework 2)

- Open Source Framework
- Konzentriert sich auf Entwicklung von Web Applikationen
- ZF2 garantiert 100% objektorientierten Code
- Orientiert sich an die neuen Funktionen von > PHP 5.3
- Unterstützt Namespaces, Lambda Funktionen, Closures, Late Static Binding
- Framework weist eine grosse Community mit über 15 Mio. Downloads auf
- Einheitliche Komponenten Struktur
- ZF2 folgt dem SOLID object oriented design Prinzip
- Flexible Benutzung der gewünschten Komponenten «“use-at-will” design»
- Pyrus und Composer als Installation und Abhängigkeit Tracking werden unterstützt
- PHPUnit für UnitTest in PHP
- Robuste, Hoch Performante MVC Implementierung
- Datenbank Abstraktion, Formulare, HTML5 Rendering, Validierung, Filterung
- ZF2 Partner unteranderem Google, Microsoft, Strikelron → Anbindung zu Web Dienste

Warum ZF2 (Zend Framework 2)

- Warum das Rad neu erfinden wenn ein Framework vorhanden ist, welches auch erfahrenen Programmierern einsetzen
- Fokus auf das Problem bzw. auf das Wesentliche der Applikation setzen
  (Keine Zeitvergeudung durch Nebensächliche Implementationen bsp. Login (Zeit ist Geld))
- Grosse Auswahl an Komponenten
- Durchdachte Programmierstruktur
- Stabil und mit Unit Tests abgesichert
- In allen Bereichen Erweiterbar (Alles lässt sich anpassen)
- Fortlaufende Weiterentwicklung (hat Zukunft, es gibt Literatur Bücher, Video Trainings)
- Umfangreiche Dokumentation (API, User Guide)
- Zwingt Entwickler nach dem MVC Prinzip zu entwickeln
- Grosse Community → Support → Es gibt schneller eine Lösungen zu Bugs, Problemstellungen
- Grosse Firma Zend steckt dahinter
- Komponente sind Lose gekoppelt → Applikation ist daher nicht unnötig überladen mit Komponenten → Komponente können von Entwickler beliebig hinzugefügt/entfernt werden
- Fehlende Komponente können selber programmiert werden → **Scalability**
- Schwachstellen und Bugs werden schneller entdeckt als von Hacker, da grosse Community
  - **Security by Obscurity** hat noch nie wirklich funktioniert, wenn man auf eigene Entwicklung setzt, geschweige von Bugfixes → Hacker sind dann meist auch schneller
- Bootstrap ist automatisch schon integriert

Grenzen von ZF2 (Zend Framework 2)

- Es braucht Zeit für die Einarbeitung in ein fremdes Framework und studieren von dessen Dokumentationen
- Zend nimmt gewisse Arbeit ab, jedoch muss für die Verwendung der eigenen Bedürfnisse, Problemstellungen und die Integration/Anpassungen der gewünschten Komponente Arbeit einkalkuliert werden

6. Google Maps

Für die Dartstellung der ICS entschieden wir uns für Google Maps aus folgenden Gründen

- Visualisiert Geodaten
- Hat eigene API (Application Programming Interface) für die Einbindung der Karte auf die eigene Webseite
- Kostenlose Verfügbarkeit (Open Source)
- Koordinaten als Dezimaldarstellung (Rohgeometrie ist schon vordefiniert)
- Zoomstufen
- Daten können durch Overlays zusammengefasst werden
- Daten können mit eigenen oder auf dem Internet abrufbaren Informationen dargestellt werden
- HTML & JavaScript (weit verbreitet)

7. jqPlot

Für die visualisierung von Statistiken wird jqPlot eingesetzt. Für jqPlot sprechen folgende Argumente

- zeichnet Grafik als Canvas (HTML5 Element)
- einfache Nutzung (JavaScript, JQuery) - OpenSource
- Grosse Auswahl von Funktionen und Grafiken (Zoom, Mouseover, labels, charts)
- Kann Grafik an das Browserfenster automatisch anpassen was zum restlichen Responsiven Webdesign passt
- Daten müssen nicht übergeben werden, wie dies bei Google Chart der Fall wäre da Library lokal vorliegt.
- Supported in IE, Chrome, Firefox, Opera, Safari, …

ICS ThreadMap - v1.0

Projektplan
(PRP)

Dominique Sorg
Benjamin Kehl

Änderungsgeschichte

Datum	Version	Änderung	Autor
16.09.2013	0.1	Erstellung Dokument	Benjamin Kehl
20.09.2013	0.2	Technische Risiken	Dominique Sorg
20.09.2013	0.3	Grobplanung Meilensteine	D. Sorg, B. Kehl
23.09.2013	0.4	Kapitel Einführung, Projektübersicht und Projektorganisation hinzugefügt	Benjamin Kehl
23.09.2013	0.5	Kapitel Qualitätsmanagement hinzugefügt	Dominique Sorg
23.09.2013	0.6	Kapitel Projektmanagement und Arbeitspakete hinzugefügt	Benjamin Kehl
04.10.2013	0.7	Überarbeitung Dokument	Dominique Sorg
05.10.2013	0.8	Überarbeitung Referenzen	Benjamin Kehl
06.10.2013	0.9	Änderung Verwendungstools im Kapitel Infrastruktur	Benjamin Kehl
11.10.2013	1.0	Anpassung Projektplan	Dominique Sorg
26.11.2013	1.1	Überarbeitung Projektplan	Dominique Sorg

Inhalt

1. Einführung

1.1 Zweck

Dieses Dokument beschreibt den Projektplan für die Semesterarbeit ICS Threat Map.

1.2 Gültigkeitsbereich

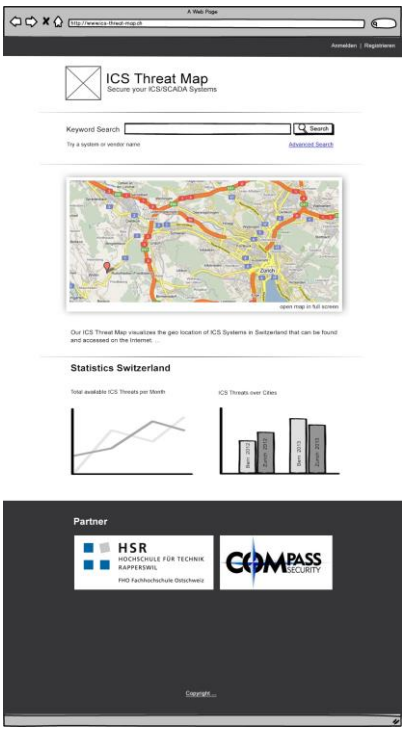
Der Projektplan gilt als Grundlage des Projektes und ist daher über die gesamte Projektdauer gültig. Damit der Projektplan immer auf dem neusten Stand ist, wird er laufend angepasst.

1.3 Referenzen

Folgende Informationen wurden auf ein separates Dokument in Redmine ausgelagert. Diese befinden sich auf der Homepage http://sinv-56043.edu.hsr.ch/redmine im Projekt *ICS Threat Map* und im Bereich *Dokumente*.

Information	Pfad
Quellenverzeichnis	Quellenverzeichnis.pdf
Glossar	Glossar.pdf
Meilensteine	Meilenstein.xlsx
Risikoanalyse	TechnischeRisiken.pdf

2. Projekt Übersicht



Das Internet bietet heutzutage viele Möglichkeiten und Vorteile, um beispielsweise im geschäftlichen Bereich schnell an Informationen zu gelangen, um zu kommunizieren oder über das Internet Arbeitsprozesse abzuwickeln.

Diesen Vorteil lassen auch die Schweizer Industrie sich nicht entgehen, um ihre Arbeitsprozesse zu verbessern und um flexibler von Zuhause aus ihre ICS Systeme zu verwalten (Remote Access).

ICS ist dabei ein genereller Überbegriff von verschiedenen Kontrollsystem-Typen, wie bzw. SCADA, UPS, BMS, ERP, HMI, PDU, PLC, PLCND und viele weitere Systeme, die Anwendung in Industriellen Umgebungen finden. Die ICS Systeme werden typischerweise von Elektrizitätswerken, Wasserversorgungen, Öl-/Gasplattformen, usw. eingesetzt.

Trotz den Vorteilen des Internets sollten wir auch die Schattenseite nicht ausser Acht lassen, die einfach von Betrügern und Kriminellen ausgenutzt werden können. Dabei stellen die angeschlossenen ICS Systeme am Internet eine ernstzunehmende Bedrohung für die Schweiz dar. Nach ersten Abklärungen und Analysen mit Suchmaschinen wie Shodan und Google, oder durch diverses Einlesen in Artikeln, konnte festgestellt werden, dass viele der ICS Systeme ohne Authentisierung am Internet angeschlossen sind.

Aus diesem Grund soll in dieser Arbeit eine Web-Applikation entwickelt werden, dass die aktuelle und historische Gefahr im Bereich ICS der Schweiz visualisiert werden kann. Genauer soll auf einer Landkarte die ICS und die damit verbundenen Risiken visualisiert werden.

Neben der Darstellung sollen auch ICS nach verschiedensten Filtern wie der Schweregrad der Verletzlichkeit, Standort, verwendete Protokolle, usw. gesucht werden können.

Falls ein ICS System ein sehr hohes Bedrohungsausmass aufweist, so muss dieses System mittels einem Trouble Ticket System beobachtet und bearbeitet werden können.

2.1 Zweck und Ziel

Das Ziel der Web-Applikation besteht darin, dass der Nutzer der Web-Applikation (z.B. ICS-CERT, MELANI, Bund, Security Firma) in der Lage sein soll, den Betreibern der ICS die Gefahr aufzuzeigen und zu dokumentieren, welche Aktionen er getroffen hat, so dass die Bedrohungen für die Schweiz minimiert werden können.

2.2 Lieferumfang

ICS Threat Map soll folgende Funktionen zur Verfügung stellen:

- Suchen von ungeschützten ICS in der Schweiz und aufzeigen von neuen ICS
- ICS in einer Datenbank speichern und mit zusätzlichen Informationen anreichern
- Identifizieren des Anlagen-Standortes und des Anlagen-Betreibers
- Klassifizieren der Anlage (Typ der Anlage, Bedrohungsausmass, Vernetzung)
- Integration eines Trouble-Ticket-Systems, mit welchem die Aktionen bezüglich eines ICS dokumentiert werden
- Eine Landkarte, auf welcher die ICS und die damit verbundenen Risiken visualisiert werden

- Eine Emailbenachrichtigung bei der Registrierung und Freischaltung von Benutzern, sowie beim Ablauf des Update Script

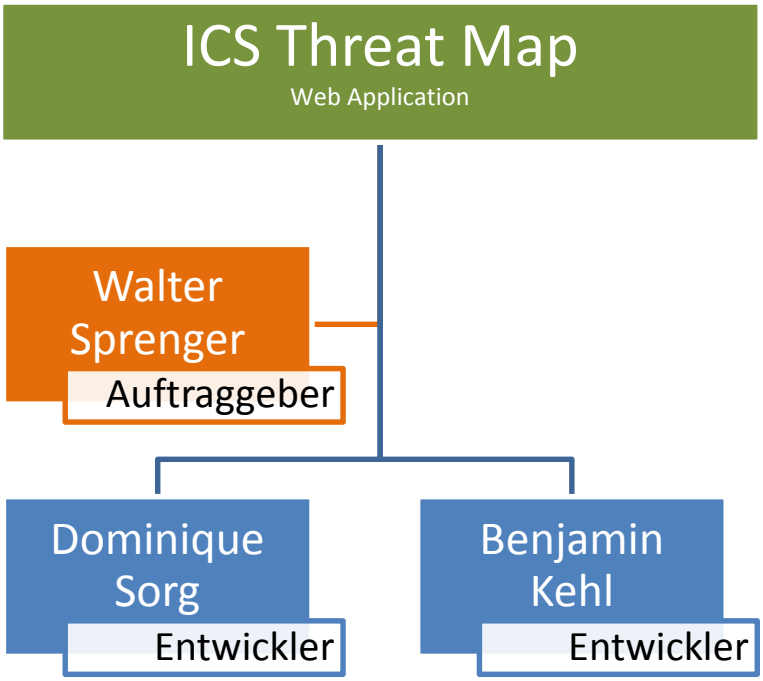
2.3 Annahmen und Einschränkungen

Um Zugang zu der Webapplikation zu erhalten, ist ein Login-Prozess zwingend notwendig. Die Registration eines Nutzers muss von einem Administrator bestätigt werden.

3. Projektorganisation

Die Semesterarbeit wird von Dominique Sorg und Benjamin Kehl durchgeführt. Jeder dieser Mitglieder hat seinen eigenen Verantwortlichkeits- und Aufgabenbereich. Die Arbeitspakete werden je nach Aufgabenbereich und Verantwortlichkeit zugeteilt. Dabei können auch nicht zugewiesene Arbeitspakete bearbeitet werden. Walter Sprenger aus Compass Security AG, ist über dem gesamten SA unser Projektbetreuer.

3.1 Organisationsstruktur



3.2 Team

Benjamin Kehl	
Mailadresse	bkehl@hsr.ch
Motivation	Neue Herausforderungen motivieren mich sehr und dienen zugleich als eine persönliche Weiterbildung (Umgang mit MySQL Datenbank, PHP, JavaScript, ...). Ich freue mich bereits auf die Implementation und bin davon überzeugt meinen Teil zu einer gutfunktionierenden und nützlichen Webapplikation zu leisten.
Aufgabenbereich	- MySQL, Zend Framework 2
Verantwortlichkeiten	- ICS in der DB verwalten, Suchen und Filtern nach ICS, Klassifizierung von ICS, Login & Benutzerverwaltung

Dominique Sorg	
Mailadresse	dsorg@hsr.ch
Motivation	Die Entwicklung an einem komplexen Webprojekt stellt mich vor neuen Herausforderungen und ermöglicht meiner Kreativität freien Lauf zu lassen. Des Weiteren lerne ich neue Technologien im Webbereich kennen, bsp. (Google Maps, Google Charts, Bootstrap 3, PHP Web Dienst, ZF2, ...). Da ich auch ein grosses Interesse an IT Security habe, motiviert es mich zusätzlich ein Projekt in diesem Bereich abwickeln zu können.
Aufgabenbereich	- Zend Framework 2, Javascript, JSON
Verantwortlichkeiten	- ICS aus Shodan beschaffen, Design der Webapplikation (Frontend), Standortanzeige mittels Map, Statistik anzeige mit Charts, Trouble Ticket System

3.3 Externe Schnittstellen

Während unserem Projekt sind ausser Walter Sprenger keine weiteren externen Personen involviert. Da unser Betreuer ein Mitgründer der Compass Security AG ist, können Usability Tests mit Mitarbeiter durchgeführt werden, von denen wir uns Feedback erhoffen. Jedoch wäre es durchaus vorstellbar auch unseren ehemaligen Betreuer von SE2-Projekt Daniel Keller bezüglich datenbanktechnischer Fragen zu kontaktieren.

4. Management Abläufe

Die zeitliche Planung wurde zuerst grob in Meilensteine eingeteilt. Diese Meilensteine wurden danach in Redmine erfasst und anschliessend mit Arbeitspaketen (Detail-Planung) ergänzt.

4.1 Kostenvoranschlag

Die Studienarbeit „ICS Threat Map“ beginnt mit dem Kickoff-Meeting am Montag den 16.09.2013. Anschliessend läuft die SA 14 Wochen bis zur definitiven Abgabe am Freitag den 20.12.2013.

Pro Teammitglied muss mindestens 240 Arbeitsstunden im Sinne der SA investiert werden. Daraus leitet sich eine totale Anzahl an Arbeitsstunden von 480 Stunden ab. Wir gehen von einer maximalen Obergrenze an Arbeitsstunden von 30% aus, das heisst zusätzliche 140 Stunden.

Sollte der Projektumfang bis Abgabe schneller als geplant fertiggestellt werden, so können optionale Funktionen hinzugenommen werden.

4.2 Zeitliche Planung

Unsere Studienarbeit besteht aus 8 Meilensteinen, die zwingend einzuhalten sind. Zusätzlich werden wir wöchentlich mit unserem Betreuer ein Meeting führen. Die Reviews werden immer nach Ende des letzten Meilensteins geführt.

Die Meilensteine werden in Redmine als Versionen abgebildet und die groben Arbeitspakete als Tickets angegeben. Redmine bietet zusätzlich ein integriertes Zeitplan-Diagramm (Gantt-Diagramm). Dies liefert uns immer einen direkten IST-SOLL Vergleich zu unserer Projektmappe, ob wir den Zeitplan einhalten oder ob wir vom Zeitplan abweichen. Die Arbeitspakete werden in Kapitel 6 detaillierter beschrieben.

4.2.1 Phasen / Iterationen

Unsere SA wird nach RUP in Inception, Elaboration, Construction und Transition unterteilt. Jede Phase enthält eine oder mehrere Iterationen mit einer Grössenordnung von etwa 2 Wochen. Dazu beachten Sie unsere Zeitplanung in Redmine.

Die SA beginnt mit der Inception Phase und endet dann nach dem Durchlauf aller RUP Phasen mit der Transition Phase. Die Iterationen sind ebenfalls in Redmine als Meilensteine definiert.

4.2.1.1 Inception

Die Inception Phase dient als Konzeptionsphase in der die Vision, die Ziele und die Risiken des Projektes wie auch die wichtigsten Funktionalitäten beschrieben werden. Vom Projektbeginn begann die Inception Phase zu Beginn des Kickoff-Meetings am 16.09.2013 und endet eine Woche darauf am 23.09.2013.

4.2.1.2 Elaboration

Anschliessend läuft die Elaboration Phase 4 Wochen, somit mit 2 Iterationen. In der ersten Iteration werden 80% der Anwendungsfälle mittels Artefakten beschrieben und der Projektplan endgültig beim Betreuer vorgelegt. In der zweiten Iteration werden mehrere Architekturprototypen erstellt, um mögliche Risiken bereits zu bearbeiten.

4.2.1.3 Construction

Nachdem die Architektur ausgearbeitet worden ist, beginnen wir in der Construction Phase mit der Entwicklung und dem Testen. Die Construction Phase enthält 4 Iterationen mit insgesamt 7 Wochen. Nach Ende jeder Iteration wird ein Architektursprototyp implementiert und ausgebaut.

4.2.1.4 Transition

In der Transition Phase bleiben 2 Wochen übrig. Hier können parallel letzte Anpassungen und Fehlerbehebungen durchgeführt werden sowie für die Abgabe des A0-Posters und der Dokumentation vorbereitet werden.

4.2.2 Meilensteine

Das Projekt wurde in 9 Meilensteine definiert. Beachten Sie unser Meilenstein Dokument oder Redmine für die detailierte Plannung.

4.2.2.1 M1 Projektbeginn

Mit dem Kick Off Meeting bei der Compass AG werden die ersten Vorabklärungen gemacht und die Aufgabenstellung besprochen.

4.2.2.2 M2 Projektplan

Die technischen Risiken werden im Dokument Technische Risiken analysiert und abgearbeitet. Die Meilensteine werden definiert und eine Detailplanung in Redmine mit Arbeitspakete ausgearbeitet. Mit dem Projektplan und den Wireframes wird ein Lösungsvorschlag ausgearbeitet.

4.2.2.3 M3 Anforderung & Analyse, Design, Prototyp

Die Anforderungsspezifikation werden abgearbeitet und ein Datenbank Modell erstellt. Es werden die Use Cases, die nicht funktionalen Anforderungen, ein erstes Design und Prototypen erstellt.
Es wird ein Prototyp mit Google Map, für die Datenbeschaffung oder für die Statistikanzeige erstellt.

4.2.2.4 M4 Implementation: ICS in Datenbank anreichern

Dieser Meilenstein beschäftigt sich mit der Datenbeschaffung von kritischen ICS aus externen Quellen wie Shodan. Diese werden in unserer ausgearbeiteten Datenbank angereichert.

4.2.2.5 M5 Implementation: Frontend Design

Im Frontend Design, fallen die Komponenten wie Suchfunktion, Kartenansicht, Statistik Charts und das visuelle Design der Web Applikation. In diesem Meilenstein werden die erstellt Prototypen ausgebaut und in die bestehende Applikation miteingebaut.

4.2.2.6 M6 Implementation: Benutzerverwaltung / TT-System

Ein grosser Bereich der Applikation ist die Benutzerverwaltung und das Trouble Ticket System. Zudem wird das Hinzufügen und Bearbeiten von ICS Daten realisiert. Darunter fällt auch die Klassifizierung von ICS.

4.2.2.7 M7 Refactoring & Testing

Ein eigener Meilenstein befasst sich mit Refactoring und Testen. Die bestehende Lösung wird überarbeitet und ausgearbeitet. Mit diesem Meilenstein wollen wir eine bessere Usability und eine zuverlässige Applikation gewährleisten. Dieser Meilenstein dient auch als Reserve, um die nicht komplett fertiggestellten Funktionen zu vervollständigen.

4.2.2.8 M8 Abgabe Kurzfassung & A0-Poster

Ein vordefinierter Meilenstein der HSR ist die Abgabe einer Kurzfassung und einem A0 Poster an unserem Betreuer.

4.2.2.9 M9 Definitive Abgabe

Fertigstellung des Projekts und letzte Anpassungen gemacht. Abgabe des Projekts an die HSR.

4.3 Besprechungen

Da beide Teammitglieder denselben Stundenplan bzw. die gleichen Module besuchen, werden die Besprechungen sehr oft in den Zwischenzeiten durchgeführt.
Die Besprechungen mit Herrn Sprenger soll weitgehend wöchentlich am Montag durchgeführt werden. Einzige Ausnahme zwischen 30.09.2013 und 05.10.2013 wegen Abwesenheit von Herrn Sprenger.

5. Risikomanagement

Das Risikomanagement wird in einem separatem Dokument *TechnischeRisiken.xlsx* ausführlicher beschrieben und aufgelistet. Dabei wurden insbesondere die technischen Risiken für unsere Studienarbeit analysiert und erläutert.

5.1 Risiken

Die Risiken werden im technischen Risiken Dokument aufgelistet und bewertet.

5.2 Umgang mit Risiken

Mit Risiken und Problemen in einem Projekt muss immer gerechnet werden. Aus diesem Grund ist es wichtig genügend Zeitreserven einzuplanen, um die kritischsten Risiken zu beseitigen und die Kernarchitektur zu bestimmen. Die Einarbeitung in diesen Themenbereichen ist schon sehr früh im vollen Gange.

Die Risiken müssen bis zum Abschluss des Projektes im Auge behalten werden. Falls Probleme auftauchen, dienen Sitzungen für die Besprechung und anschliessend deren Lösungssuche. Zusätzlich appellieren wir an unsere Eigeninitiative, dass bei Problemen sofort die anderen Teammitglieder informiert werden, damit wir als Team die Risiken und Probleme in Griff bekommen. Zudem müssen laufend neue Risiken protokolliert werden.

Zusätzlich werden in der Elaboration-Phase Technologieprototypen entwickelt, um allfällige Risiken, die aufgrund unbekannter Technologien auftauchen könnten, zu minimieren.

6. Arbeitspakete

Alle Arbeitspakete werden in Redmine definiert und abgearbeitet. Hier werden die Arbeitspakete pro Meilenstein als Screenshot gezeigt.

Projektbeginn

#	Tracker	Status	Priorität	Thema	Zugewiesen an	Aktualisiert	Kategorie	Zielversion
37	Meeting	Erledigt	Normal	Sitzung 16.09.2013		23.09.2013 09:29		Projektbeginn
36	Feature	Erledigt	Hoch	Vorbereitung		23.09.2013 09:24		Projektbeginn

Projektplan

#	Tracker	Status	Priorität	Thema	Zugewiesen an	Aktualisiert	Kategorie	Zielversion
103	Report	Abgewiesen	Normal	Wochen Report KW 40		07.10.2013 14:36		Projektplan
102	Report	Erledigt	Normal	Wochen Report KW 39		07.10.2013 12:51		Projektplan
79	Feature	Erledigt	Normal	Suchfilter für ICS zusammentragen		05.10.2013 10:52		Projektplan
70	Feature	Erledigt	Normal	Einarbeitung PHP Zend Framework 2	Benjamin Kehl	07.10.2013 09:49		Projektplan
67	Feature	Erledigt	Normal	Einarbeitung Google Search		03.10.2013 10:25		Projektplan
64	Feature	Erledigt	Normal	Einarbeitung Bootstrap		04.10.2013 14:52		Projektplan
51	Meeting	Erledigt	Normal	Sitzung 7.10.2013		07.10.2013 18:58		Projektplan
50	Meeting	Erledigt	Normal	Sitzung 26.09.2013		30.09.2013 08:52		Projektplan
48	Feature	Erledigt	Hoch	[Projektplan] Dokument abschliessen		04.10.2013 14:54		Projektplan
47	Feature	Erledigt	Normal	Einarbeitung MySQL	Benjamin Kehl	07.10.2013 12:38		Projektplan
46	Feature	Erledigt	Normal	[Projektplan] Infrastruktur und Qualitätsmassnahmen		03.10.2013 15:19		Projektplan
45	Feature	Erledigt	Normal	[Projektplan] Risikomanagement		25.09.2013 09:34		Projektplan
43	Feature	Erledigt	Normal	[Projektplan] Projektübersicht und Projektorganisation	Benjamin Kehl	25.09.2013 17:36		Projektplan
42	Feature	Erledigt	Normal	Git einrichten		03.10.2013 10:25		Projektplan
41	Report	Erledigt	Normal	Wochen Report KW 38		23.09.2013 09:51		Projektplan
40	Feature	Erledigt	Normal	[Projektplan] Detailplanung Arbeitspakete		03.10.2013 13:56		Projektplan
39	Feature	Erledigt	Normal	Virtuellen Server einrichten		25.09.2013 09:32		Projektplan
38	Feature	Erledigt	Normal	Entwicklungsumgebung Lokal einrichten		23.09.2013 09:38		Projektplan
35	Feature	Erledigt	Normal	[Projektplan] Wireframe		25.09.2013 09:33		Projektplan
34	Feature	Abgewiesen	Niedrig	Einarbeitung: Google Charts API		03.10.2013 15:20		Projektplan
33	Feature	Erledigt	Normal	Einarbeitung Google Map API		25.09.2013 17:59		Projektplan
31	Feature	Erledigt	Hoch	Projektplan erstellen		25.09.2013 09:06		Projektplan
30	Feature	Erledigt	Dringend	[Projektplan] Managementabläufe		25.09.2013 09:28		Projektplan
29	Feature	Erledigt	Normal	Einarbeitung PHP		03.10.2013 16:27		Projektplan
28	Feature	Erledigt	Hoch	Einarbeitung Shodan API		25.09.2013 09:33		Projektplan

Analyse & Design

#	Tracker	Status	Priorität	Thema	Zugewiesen an	Aktualisiert	Kategorie	Zielversion
110	Report	Erledigt	Normal	Präsentation Prototypen		25.10.2013 22:15		Analyse & Design
109	Feature	Erledigt	Normal	[Prototyp] Anreicherung von ICS Daten in die Datenbank		21.10.2013 09:37		Analyse & Design
108	Feature	Erledigt	Normal	[Prototyp] Zend Framework 2		19.10.2013 18:58		Analyse & Design
105	Report	Erledigt	Normal	Wochen Report KW 42		21.10.2013 22:09		Analyse & Design
104	Report	Erledigt	Normal	Wochen Report KW 41		18.10.2013 14:01		Analyse & Design
100	Feature	Erledigt	Normal	Abarbeitung/ReAnalyse Technische Risiken		19.10.2013 18:55		Analyse & Design
99	Feature	Erledigt	Normal	[Prototyp] Filter Google		19.10.2013 18:58		Analyse & Design
98	Feature	Erledigt	Normal	[Prototyp] Filter Shodan		18.10.2013 13:57		Analyse & Design
97	Feature	Erledigt	Normal	[Prototyp] Map		18.10.2013 13:56		Analyse & Design
90	Feature	Erledigt	Normal	Projektstruktur		18.10.2013 13:55		Analyse & Design
73	Feature	Abgewiesen	Normal	Dokumentation: Bericht		21.10.2013 09:35		Analyse & Design
71	Feature	Erledigt	Normal	Dokumentation: Bericht		14.10.2013 09:36		Analyse & Design
66	Feature	Erledigt	Normal	[Use Case] Brief/Detailed		21.10.2013 10:04		Analyse & Design
65	Feature	Erledigt	Normal	[Prototyp] ICS Datenbank	Benjamin Kehl	19.10.2013 18:57		Analyse & Design
63	Feature	Erledigt	Normal	[Prototyp] Benutzerverwaltung	Benjamin Kehl	23.10.2013 19:42		Analyse & Design
54	Feature	Erledigt	Hoch	Datenbankmodelle erstellen	Benjamin Kehl	11.10.2013 11:55		Analyse & Design
53	Meeting	Abgewiesen	Normal	Sitzung 21.10.2013		24.10.2013 11:41		Analyse & Design
52	Meeting	Erledigt	Normal	Sitzung 14.10.2013		18.10.2013 14:00		Analyse & Design
32	Feature	Erledigt	Normal	Requirments Analysis: Use Case Diagram		03.10.2013 16:11		Analyse & Design

ICS in Datenbank

#	Tracker	Status	Priorität	Thema	Zugewiesen an	Aktualisiert	Kategorie	Zielversion
120	Feature	Erledigt	Normal	[Suchfunktion] Rechtevergabe	Benjamin Kehl	01.11.2013 19:53		Implementation: ICS in Datenbank
119	Fehler	Erledigt	Hoch	[Suchfunktion] Modul für das Grundgerüst erstellen	Benjamin Kehl	01.11.2013 19:47		Implementation: ICS in Datenbank
118	Feature	Erledigt	Normal	[Suchfunktion] Refactoring		07.11.2013 14:33		Implementation: ICS in Datenbank
117	Feature	Erledigt	Normal	[Suchfunktion] Paginator implementieren	Benjamin Kehl	01.11.2013 19:44		Implementation: ICS in Datenbank
116	Feature	Erledigt	Normal	[Suchfunktion] Resultate anzeigen		07.11.2013 14:32		Implementation: ICS in Datenbank
114	Feature	Erledigt	Normal	[Suchfunktion] Suchmaske implementieren + Filter Logik		07.11.2013 14:31		Implementation: ICS in Datenbank
107	Report	Erledigt	Normal	Wochen Report KW 44		07.11.2013 14:33		Implementation: ICS in Datenbank
106	Report	Erledigt	Normal	Wochen Report KW 43		27.10.2013 21:31		Implementation: ICS in Datenbank
91	Feature	Erledigt	Normal	Implementation Suchmaske		07.11.2013 14:31		Implementation: ICS in Datenbank
89	Feature	Erledigt	Normal	Implementation Mail mit Statistik		07.11.2013 14:30		Implementation: ICS in Datenbank
87	Feature	Erledigt	Normal	Implementation Datenbeschaffung		01.11.2013 17:08		Implementation: ICS in Datenbank
86	Feature	Erledigt	Normal	Implementation API DB		11.11.2013 15:32		Implementation: ICS in Datenbank
75	Feature	Abgewiesen	Normal	Dokumentation: Bericht		07.11.2013 14:35		Implementation: ICS in Datenbank
74	Feature	Abgewiesen	Normal	Dokumentation: Bericht		27.10.2013 21:07		Implementation: ICS in Datenbank
69	Feature	Erledigt	Normal	Datenbank Aufsetzung		27.10.2013 21:07		Implementation: ICS in Datenbank
68	Feature	Erledigt	Normal	Installations-Script		07.11.2013 14:30		Implementation: ICS in Datenbank
56	Meeting	Erledigt	Normal	Sitzung 4.11.2013		07.11.2013 14:32		Implementation: ICS in Datenbank
55	Meeting	Erledigt	Normal	Sitzung 28.10.2013		01.11.2013 17:11		Implementation: ICS in Datenbank
49	Feature	Erledigt	Normal	JSON analysieren zum parsen		21.10.2013 20:55		Implementation: ICS in Datenbank

Frontend Design

#	Tracker	Status	Priorität	Thema	Zugewiesen an	Aktualisiert	Kategorie	Zielversion
128	Feature	Erledigt	Normal	[Konfigurationsbereich] Paginator	Benjamin Kehl	20.11.2013 13:37		Implementation: Frontend Design
127	Feature	Erledigt	Normal	[Konfigurationsbereich] Refactoring	Benjamin Kehl	20.11.2013 13:37		Implementation: Frontend Design
125	Feature	Erledigt	Normal	[Konfigurationsbereich] Suchfilter CRUD	Benjamin Kehl	20.11.2013 13:19		Implementation: Frontend Design
124	Feature	Erledigt	Normal	[Konfigurationsbereich] Routing bestimmen	Benjamin Kehl	12.11.2013 18:30		Implementation: Frontend Design
123	Feature	Erledigt	Dringend	[Konfigurationsbereich] Modul erstellen	Benjamin Kehl	12.11.2013 18:37		Implementation: Frontend Design
122	Feature	Erledigt	Normal	Webseite auf Webserver aufschalten		21.11.2013 16:41		Implementation: Frontend Design
121	Feature	Erledigt	Normal	Implementation Statistik anzeige Hauptseite		21.11.2013 16:46		Implementation: Frontend Design
92	Feature	Erledigt	Normal	Implementation Frontend Design (Suchseite, Datenanzeige)		21.11.2013 16:45		Implementation: Frontend Design
88	Feature	Erledigt	Normal	Implementation Map (Datenanzeige)		11.11.2013 15:32		Implementation: Frontend Design
77	Feature	Erledigt	Normal	Dokumentation: Bericht		21.11.2013 16:48		Implementation: Frontend Design
76	Feature	Erledigt	Normal	Dokumentation: Bericht		08.11.2013 17:24		Implementation: Frontend Design
72	Feature	Erledigt	Normal	Implementation Frontend Design (Hauptseite)		20.11.2013 13:16		Implementation: Frontend Design
58	Meeting	Erledigt	Normal	Sitzung 18.11.2013		21.11.2013 16:48		Implementation: Frontend Design
57	Meeting	Abgewiesen	Normal	Sitzung 11.11.2013		11.11.2013 15:35		Implementation: Frontend Design

Login, Verwaltung, TTS

#	Tracker	Status	Priorität	Thema	Zugewiesen an	Aktualisiert	Kategorie	Zielversion
140	Feature	Erledigt	Normal	Umschaltung Apache auf https	Benjamin Kehl	02.12.2013 11:26		Implementation: Login, Verwaltung, TTS
139	Feature	Erledigt	Normal	[TicketSystem] Ticket löschen		28.11.2013 19:24		Implementation: Login, Verwaltung, TTS
138	Feature	Erledigt	Normal	[TicketSystem] Ticket bearbeiten		28.11.2013 19:24		Implementation: Login, Verwaltung, TTS
137	Feature	Erledigt	Normal	[TicketSystem] Ticket hinzufügen		28.11.2013 19:23		Implementation: Login, Verwaltung, TTS
136	Feature	Erledigt	Normal	[TicketSystem] Modul erstellen	Dominique Sorg	18.12.2013 17:05		Implementation: Login, Verwaltung, TTS
134	Feature	Erledigt	Normal	[User] Rollen erweitern	Benjamin Kehl	25.11.2013 09:45		Implementation: Login, Verwaltung, TTS
133	Feature	Erledigt	Normal	CronJob UpdateScript einrichten	Dominique Sorg	21.11.2013 16:50		Implementation: Login, Verwaltung, TTS
132	Feature	Erledigt	Normal	[User] Activation	Benjamin Kehl	30.11.2013 09:04		Implementation: Login, Verwaltung, TTS
131	Feature	Erledigt	Normal	[Konfigurationsbereich] ICS bearbeiten	Benjamin Kehl	30.11.2013 09:03		Implementation: Login, Verwaltung, TTS
126	Feature	Erledigt	Normal	[Konfigurationsbereich] ICS hinzufügen	Benjamin Kehl	30.11.2013 09:02		Implementation: Login, Verwaltung, TTS
96	Feature	Abgewiesen	Normal	Vorbereitung Usability Tests		04.12.2013 14:06		Implementation: Login, Verwaltung, TTS
95	Feature	Erledigt	Normal	Überarbeitung Installations Routine		25.11.2013 09:45		Implementation: Login, Verwaltung, TTS
93	Feature	Erledigt	Normal	Implementation Login		25.11.2013 09:44		Implementation: Login, Verwaltung, TTS
80	Feature	Erledigt	Normal	Dokumentation: Bericht		30.11.2013 09:00		Implementation: Login, Verwaltung, TTS
78	Feature	Erledigt	Normal	Dokumentation: Bericht		25.11.2013 09:44		Implementation: Login, Verwaltung, TTS
60	Meeting	Abgewiesen	Normal	Sitzung 2.12.2013		04.12.2013 14:05		Implementation: Login, Verwaltung, TTS
59	Meeting	Erledigt	Normal	Sitzung 25.11.2013		28.11.2013 19:25		Implementation: Login, Verwaltung, TTS

Überprüfung Poster & Abstract

#	Tracker	Status	Priorität	Thema	Zugewiesen an	Aktualisiert	Kategorie	Zielversion
83	Feature	Erledigt	Normal	Poster erstellen		17.12.2013 12:33		Abgabe Poster
82	Feature	Neu	Normal	Dokumentation: Bericht		03.10.2013 14:08		Abgabe Poster
62	Meeting	Abgewiesen	Normal	Sitzung 16.12.2013		17.12.2013 12:33		Abgabe Poster

7. Infrastruktur

Eine Anforderung unseres Projektes ist es, wenn möglich Open-Source Tools zu verwenden. Dabei haben wir folgende Tools im Einsatz:

Software / Systeme	Beschreibung / Einsatzbereich
Betriebssystem	In unserem Team kommen Windows 7 / 8 und Linux zum Einsatz
Eclipse for Zend (PDT)	Zur Webentwicklung hat sich PDT als sehr praktisch erwiesen.
ZendFramework 2	Ein komponentenbasiertes MVC Framework für PHP.
HTML, CSS	Das Design und Layout der Webapplikation wird mit HTML und CSS realisiert.
Twitter Bootstrap	Framework für Responsive Web Design mit CSS und HTML.
PHP, Javascript	PHP wird als serverseitige und Javascript als clientseitige Programmiersprache verwendet
PHPUnit	PHPUnit ist ein Testing-Framework für PHP
MySQL	Für die Speicherung der Daten soll MySQL verwendet werden.
Git	Git soll für die Versionsverwaltung von Dateien und Quellcode verwendet werden. Link zum Remote-Repo: https://git.hsr.ch/git/ICSThreatMap
Redmine	Realbasiertes Projektmanagementtool. Wird für das Projektmanagement, das Wiki und den Stundenrapport verwendet. Link: http://sinv-56043.edu.hsr.ch/redmine
Ms Office	Für die Dokumentation kommt Office zum Einsatz
Dropbox	Innerhalb vom Team wird zusätzlich Dropbox für den Austausch von Dokumenten genutzt.

8. Qualitätsmassnahmen

Nach jedem Abschluss eines Arbeitspakets, muss das andere Teammitglied die Arbeit überprüfen. Durch die ständige Zusammenarbeit an der HSR werden abgeschlossene Arbeiten angeschaut und besprochen. Durch die wöchentlichen Meetings mit unserem Betreuer erhalten wir Feedbacks zu der geleisteten Arbeit.

8.1 Dokumentation

Die Dokumentation wird zwischen den Teammitgliedern auf Dropbox geteilt. Zusätzlich befindet sich auf jedem privaten Rechner eine Kopie als Backup. Auf Redmine werden die Dokumente als *.pdf hochgeladen.

8.2 Projektmanagement

Als Projektmanagement Tool setzen wir Redmine ein. Dabei verwenden wir folgende Logins:

- dsorg
- bkehl
- wsprenger

8.3 Entwicklung

Der Source Code befindet sich auf dem GIT Remote-Repository der HSR https://git.hsr.ch/git/ICSThreatMap und lokal auf unseren Workspaces.

8.3.1 Vorgehen

Sobald das Projekt, um eine weitere Funktionalität erweitert wird, wo nötig getestet, ob die Erweiterung auch den Erwartungen entspricht.

8.3.2 Unit Testing

Für den Server Teil kommt PHP zum Einsatz. Für PHP gibt es ein Unit Test Framework namens PHPUnit.

8.3.3 Code Reviews

Mindestens einmal pro Iteration findet ein Codereview durch das andere Teammitglied statt. Damit behalten beide den Überblick über das gesamte Projekt und der Code wurde von mindestens einer anderen Instanz überprüft.

8.3.4 Code Style Guidelines

Entwickelt wird mit Eclipse für Zend. Wir verwenden den Standard Code Style Guideline der Entwicklungsumgebung.

8.4 Testen

8.4.1 Integrationstest

Unser Projekt hat mehrere Schnittstellen (Client zu Server, Server zu Shodan, Client zu Google API), welche in der Construction Phase nach jeder Iteration getestet werden.

8.4.2 Systemtest

Die Web Applikation wird in der Construction Phase nach jeder Iteration einen Systemtest unterlaufen, damit sichergestellt wird, ob alle Requirements implementiert sind und funktionieren.

8.4.3 Usability

Mit Usability Tests werden Meinungen eingeholt, um die Web Applikation zu verbessern.

Risikomanagement

Projekt: ICS Threat Map
Erstellt am: 18.09.2013
Autor: Dominique Sorg, Benjamin Kehl
Gewichteter Schaden: 43.4

Nr	Titel	Beschreibung	max. Schaden [h]	Eintrittswahrscheinlichkeit	Gewichteter Schaden	Vorbeugung	Verhalten beim Eintreten
R1	Datenbeschaffung	Die Datenbeschaffung bei Shodan in unsere Datenbank dauert zu lange oder wird mit nicht effizienten Filtern abgefragt. Ebenfalls kann auf der Clientseite die Datenbeschaffung z.B. für die Map zu lange dauern, da 1'000'000 Daten deutlich länger zum Laden benötigen. Dasselbe ist, wenn der Zugriff mit dem Smartphone über das Mobilnetz erfolgt.	15	20%	3	Einarbeitung mit Prototyp, API von Shodan benutzen, Es braucht einen kostenpflichtigen Shodan Account um Vollzugriff auf die API und Querymöglichkeiten zu erlangen. -> Wir erhielten mit Absrpache mit Walter Sprenger und dem Shodan Gründer einen unlimitierten Shodan Account	Kontakt zu Walter Sprenger und dem Shodan Gründer besteht. Somit können diese zur Hilfe angefragt werden. Sonst Algorithmen, Filter, Queries neu testen.
R2	Shodan Webserver Unerreichbar	Wir haben festgestellt, dass der Shodan Webserver, ab und zu morgens, unerreichbar ist.	2	40%	0.8		
R3	Google Map API	Daten auf einer Map anzeigen zu lassen schlägt fehl oder braucht zu lange Zeit.	16	40%	6.4	Einen Prototyp einer Google Map mit Beispieldaten erstellen. Dabei soll die Beispieldaten über JSON verschickt, von der Map interpretiert und dargestellt werden.	Ursache finden und wenn Möglich beheben.
R4	Datenbank Aktualisierungen	Aktualisierungen werden falsch in die Datenbank abgespeichert. Bei dieser grosse Menge an Daten kann nicht überprüft werden, ob alle Daten richtig vorhanden sind.	30	40%	12	Einarbeitung in MySQL, JSON, PHP, SQL. Evt wäre ein Prototyp mit wenig Daten zu beginn sinnvoll um unsere Funktionen zu testen. Saubere Planung und Strukturierung der Datenbank.	Mit wenigen Daten Testen z.B. nur 10 Einträge.
R5	Loginbereich	Loginbereich benötigt zu viel Zeit, da es nicht eine Kernfunktion ist sondern eifach für das Trouble Ticket System benötigt wird.	16	50%	8	Prototyp und Konzentration auf den wesentlichen Einsatzort	
R6	Sicherheit	Datenbank kann nach der Implementierung der Suchfunktion usw. auf SQL Injection anfällig sein.	16	70%	11.2	Einsatz von Plugins die SQL Injection während der Implementation überprüft. An die APIs halten.	Nachschlagen in den jeweiligen APIs und Sicherheitslöcher schliessen.
R7	Backup	Falls der Server aus verschiedensten Gründen nicht mehr verfügbar ist oder Daten verloren gehen.	12	10%	1.2	Einsetzen von Snapshots für unseren Virtual Server.	letzte funktionierende Version zurückspielen.
R8	Git	Wenn Git nicht mehr verfügbar ist, können keine Commits und Pushes gemacht werden. Im Extremfall ist dia aktuellste Version auf dem Repository nicht mehr vorhanden.	8	10%	0.8	Regelmässig lokal sichern. Den aktuellsten Stand so häufig wie möglich mit Push beziehen.	
Summe			**115**		**43.4**		

ICS ThreadMap - v1.0

Anforderungsspezifikation (SAS)

Benjamin Kehl
Dominique Sorg

Änderungsgeschichte

Datum	Version	Änderungen	Autor
11.10.13	0.1	Erstellung Dokument	Benjamin Kehl
11.10.13	0.2	Kapitel Use Cases hinzugefügt	Benjamin Kehl
30.10.13	0.3	Aktualisierung UC Diagramm und zusätzlicher Use Case	Benjamin Kehl
16.11.13	0.4	Beschreibung Qualitätsmerkmale, Anpassung Anforderungen	Dominique Sorg
23.11.13	0.5	Ausbau Use Cases, UC Fully Dressed	Benjamin Kehl, Dominique Sorg
26.11.13	0.6	Priorisierung	Benjamin Kehl
12.12.13	0.7	Anpassungen, Korrekturen	Benjamin Kehl
19.12.13	0.8	Überarbeitung und Korrekturen	Dominique Sorg

Inhalt

1. Einführung

1.1 Zweck

Dieses Dokument beschreibt die Anforderungen für die Semesterarbeit ICS Threat Map.

1.2 Gültigkeitsbereich

Die Anforderungsspezifikation gilt als Grundlage des Projektes und ist daher über die gesamte Projektdauer gültig. Damit das Dokument immer auf dem neusten Stand ist, wird er laufend angepasst.

1.3 Übersicht

Dieses Dokument gibt eine Übersicht über die Anforderungen für unsere Studienarbeit *ICS Threat Map*. Im ersten Kapitel werden die Anforderungen allgemein beschrieben, damit die Aufgabenstellung der Semesterarbeit besser ersichtlich ist. Danach werden die Anforderungen als Use Cases erfasst und als grobe Übersicht in einem Use Case Diagramm aufgezeigt. Die wichtigsten Use Cases werden zusätzlich als UC Fully Dressed detaillierter beschrieben. Zum Schluss werden weitere Anforderungen in Form von nichtfunktionalen Anforderungen aufgezählt und beschrieben.

2. Allgemeine Beschreibung

2.1 Produkt Perspektive

ICS ThreatMap ist eine öffentlich zugängliche Web Applikation, für die primär Mitglieder aus ICS-CERT, MELANI, Bund oder Security Firmen Zugang erhalten. ICS ThreatMap hilft den Benutzern die Gefahren und Bedrohungen der Industrial Control Systems (kurz ICS, dt.: Kontrollsystem) gegenüber den ICS Betreiber aufzuzeigen. Das primäre Ziel ist es, die Bedrohungen in der Schweiz durch schlecht konfigurierte ICS zu minimieren.

2.2 Produkt Funktion

Der Benutzer erhält durch die Web Applikation eine visualisierte Darstellung der Bedrohungen in der Schweiz auf einer Landkarte angezeigt. Mit einer Suchfunktion soll er spezifischer nach Systemen suchen können. Eine Statistik zeigt zudem, wie viele mangelhaft geschützte ICS täglich gefunden und welche in unserer Datenbank hinzugefügt wurden. Damit kann ein steigender oder fallender Trend während dem Einsatz festgestellt und darauf reagiert werden. Mitglieder sollen in der Lage sein, betroffene Systeme anzuschauen, zu klassifizieren und mit Hilfe eines Trouble Ticket System die Bedrohungen abzuarbeiten und zu kommentieren. Des Weiteren kann der Administrator, Mitgliedern den Zugang zur Applikation, nach der Registration gewähren oder verweigern.

2.3 Benutzer Charakteristik

Wie bereits erwähnt, ist die Web Applikation der Öffentlichkeit nur beschränkt zugänglich. Dabei können anonyme Besucher die Bedrohungen auf der Landkarte ansehen, jedoch erhalten sie keine detailliere Informationen, wie sie beispielsweise von der Suchfunktion bereitgestellt wird.

Registrierte und bestätigte Benutzer, können sich detailliertere Informationen über die ICS anschauen und bearbeiten. Zudem sollen sie in der Lage sein, weitere Filter anzugeben, um noch mehr gefährdete Systeme in die Applikation zu erfassen.

Der Administrator verwaltet die Web Applikation und hat das Recht neu registrierte Mitglieder zu aktivieren und weitere Rechte zu vergeben.

2.4 Einschränkungen

ICS ThreatMap ist nur für autorisierte Benutzer zugänglich.

2.5 Abhängigkeiten

Für die Implementierung wird Zend Framework 2 eingesetzt. Der Server, auf dem ICS Threat Map aufgesetzt wird, muss daher ZF2 und PHP nutzen.

3. Use Cases

3.1 Überblick

In unserem Projekt müssen folgende funktionale Anforderungen umgesetzt werden:

Use Case ID	Use Case Titel	Priorität
01	Registration	1
02	An-/Abmeldung	1
03	Benutzer CRUD	1
04	Benutzer aktivieren	1
05	ICS suchen und ansehen	1
06	ICS Daten auf Map ansehen	1
07	ICS CRUD	1
08	Filter CRUD	1
09	Trouble Ticket System CRUD	1
10	ICS klassifizieren	1
11	White-/Blacklist führen	2
12	Filterauflistung eines ICS	2
13	Preview Ansicht bei Filtersetzung	2
14	Datenbeschaffung Updatescript	1
15	Verfügbarkeit der ICS in Engines überprüfen	2
16	Emailbenachrichtigung	1

3.2 Priorisierung

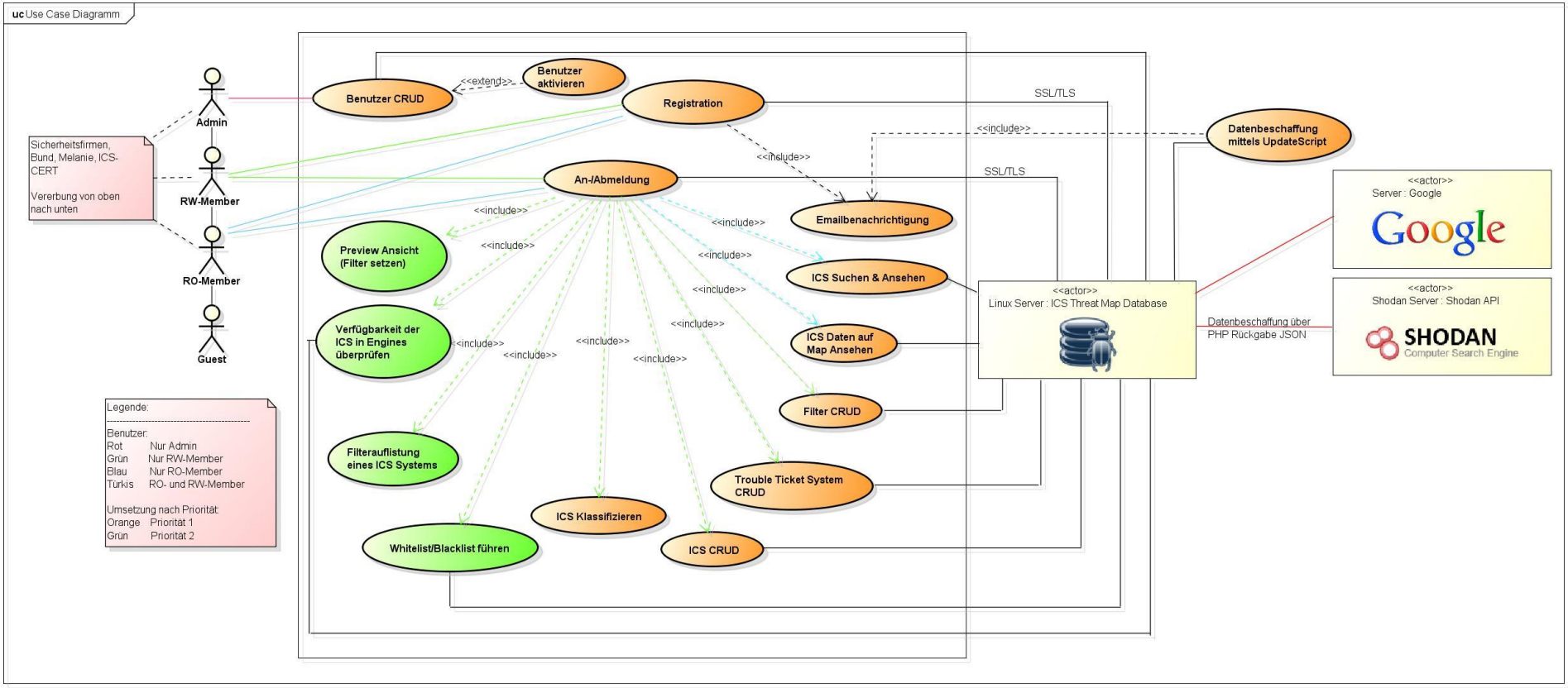
Wie in der Tabelle oben ersichtlich ist, handelt es sich um 16 Use Cases. Da nicht alle Use Cases gleich wichtig sind, wurden sie in Priorität 1 oder 2 eingestuft. Wobei Priorität 1 für *wichtig* und Priorität 2 für *normal* steht.

3.3 Aktoren & Stakeholder

Der User (oder Benutzer) ist der Hauptaktor auf der Webapplikation. Als Mitglieder aus den bereits aufgezählten Organisationen, haben sie die Möglichkeit, die Industriesysteme in der Schweiz auf einer Karte zu visualisieren und nach bestimmten Systemen zu suchen. Auch neue ICS oder Filter manuell zu erstellen, gehören in den Aufgabenbereich des Benutzers. Zusätzlich zu der Verwaltung und Pflege der Systeme, sollen sie mittels Troubleshoot Tickets einzelne Systeme nach ihren Aktionen dokumentieren können. Da nicht jeder Benutzer, für die aufgezählten Aufgabenbereiche eine Berechtigung hat, werden unter Benutzer vier weitere Aktoren unterschieden.

Aktoren	Beschreibung
Guest	Der Aktor Guest ist der Besucher der Webapplikation und hat nur die Möglichkeit auf der Hauptseite zu interagieren. Er kann die ICS in der Karte ansehen, jedoch ohne Details.
Member	Nach einer erfolgreichen Registration sind die Nutzer als Member aktiv.
Operator	Falls gewisse User zusätzliche Rechte benötigen, dann können sie auch als Operator hochgestuft werden.
Admin	Der Admin hat am meisten Rechte und ist der Einzige, der bestimmte Operationen ausführen darf. Z.B. Rechtevergabe oder Löschen von Daten.
ICS Threat Map Database	Die ICS Threat Map Database ist nicht nur zuständig für das Speichern von Daten, sondern soll auch täglich nach neuen Industriesysteme über Suchmaschinen wie Shodan oder Google suchen und die Ergebnisse in der Datenbank anreichern.

3.4 Use Case Diagramm



3.5 Beschreibungen (Brief)

UC 01	Registration
Jeder User muss sich zuerst registrieren bevor er Zugriff auf die Webapplikation bekommt. Ansonsten haben sie keine Berechtigung für das Einsehen der Daten. Um jedoch eine Registration tätigen zu können, müssen sie in einer Gruppe z.B. zu einer bereits registrierten Sicherheitsfirma dazugehören oder allgemein für die Administratoren bekannt sein.	
UC 02	**An-/Abmeldung**
Nicht angemeldete User dürfen keinen Zugriff auf die Inhalte der Webapplikation haben. Die Inhalte sind geschützt und dürfen nur mit einer erfolgreichen Anmeldung darauf zugreifen.	
UC 03	**Benutzer CRUD**
Alle registrierten Benutzer müssen verwaltet werden können. Das heisst der Benutzer kann vom Administrator gelöscht, bearbeitet oder aktiviert/deaktiviert werden.	
UC 04	**Benutzer aktivieren**
Jedes Neumitglied wird nach der Registration standardmässig deaktiviert, damit nicht jemand seine Identität vortäuschen kann und gleich direkten Zugriff auf die Daten bekommt. Deshalb muss der Administrator die Neumitglieder überprüfen und aktivieren.	
UC 05	**ICS suchen und ansehen**
Ein angemeldeter User kann nach ICS suchen und für spezifischere Suche Filter anwenden (z.B. nach Geografie, Typ des Systems, Klassifizierung, usw.).	
UC 06	**ICS Daten auf Map ansehen**
Jedes System muss auf einer Google Map dargestellt werden. Zusätzlich sollen die wichtigsten Daten wie IP, Organisation und Typ angezeigt werden.	
UC 07	**ICS CRUD**
Gewisse ICS werden über Suchfilter nicht gefunden. Daher ist es nötig manuell ICS verwalten zu können. Die Löschfunktion ist nicht direkt das Löschen selbst, sondern lediglich das Deaktivieren des ICS.	
UC 08	**Filter CRUD**
Es müssen Filter verwaltet werden um z.B. neue Systeme, die aufgrund eines alten Filters nicht auffindbar sind, mit einem zusätzlichen Filter hinzugefügt werden.	
UC 09	**Trouble Ticket System CRUD**
Für die Dokumentation und geplante Aktionen eines Industriesystems benötigt es ein Trouble Ticket System.	
UC 10	**ICS klassifizieren**
Jedes ICS muss nach Bedrohungsausmass, Kategorie, Typ usw. identifiziert und klassifiziert werden können.	
UC 11	**White-/Blacklist führen**
Es gibt immer wieder Resultate, die nicht unbedingt zu einem Industriesystem zugeordnet werden können, wie bspw. Mailserver oder Apache. Solche Informationen müssen in einer Blacklist aufgeführt werden, damit diese nicht mehr als möglich neue Industriesysteme vorgeschlagen werden.	
UC 12	**Filterauflistung eines ICS**
Jedes ICS kann über mehrere Filter gefunden werden.	
UC 13	**Preview Ansicht bei Filtersetzung**
Für eine neue Filtersetzung kann mittels einer Preview-Ansicht ermittelt werden, ob der Filter die zu erwarteten Resultate zurückliefert.	
UC 14	**Datenbeschaffung mittels UpdateScript**
Die Datenbeschaffung muss täglich mittels UpdateScript erfolgen, damit stets neue ICS in der Schweiz gesammelt werden können um diese dann auszuwerten. Das UpdateScript wird einmal früh morgens oder spät abends auf dem Webserver ausgeführt.	
UC 15	**Verfügbarkeit der ICS in Engines überprüfen**
Es könnte vorkommen, dass gewisse ICS in Shodan oder Google nicht mehr auffindbar sind, da sie entweder vom Netz genommen wurden oder die Betreiber etwas gegen ihre Verwundbarkeit unternommen haben. Diese Änderungen müssen ebenfalls erkannt werden und müssen in der ICS Threat Map Datenbank als deaktiviert / gelöscht markiert werden.	
UC 16	**Emailbenachrichtigung**

Für gewisse Use Cases muss eine Emailbenachrichtigung möglich sein. Beispielsweise, wenn ein Benutzer sich registriert hat oder wenn das UpdateScript erfolgreich durchgeführt wurde.

3.6 Fully Dressed

UC 02	An- / Abmelden	
Umfang:	ICS ThreatMap Web Applikation	
Ebene:	Security, Anwendung	
Primärer Aktor:	User (member, operator und admin)	
Vorbedingungen:	- Der Benutzer hat bereits ein Account	
Nachbedingung:	- Session wird erstellt / gelöscht - Die Inhalte werden anhand der Benutzerrolle sichtbar / unsichtbar	
Hauptszenario: (User-/Systemverantwortlichkeiten)	Aktoren, Aktionen: **Anmelden** 1. Der Benutzer meldet sich mit Emailadresse und Passwort an. **Abmelden** 1. Der Benutzer meldet sich ab	Systemverantwortlichkeiten: **Anmelden** 2. Die Angaben werden anhand des Formats überprüft 3. Die Emailadresse wird in der Datenbank gesucht 4. Das Passwort wird verschlüsselt und mit dem verschlüsseltem Passwort in der DB verglichen 5. Session wird erstellt und der Benutzer wird zu der Userpage weitergeleitet **Abmelden** 2. Der Benutzer wird identifiziert 3. Session wird zerstört
	Anmelden 2a. Das Emailformat stimmt nicht (@ vergessen, Domain falsch oder Punkt vergessen) → Eine Meldung wird angezeigt, dass das Format falsch sei 3b. Die Emailadresse konnte in der Datenbank nicht gefunden werden → Eine Meldung wird angezeigt, dass die Emailadresse nicht gefunden wurde 4c. Das Passwort stimmt nicht mit dem Passwort in der Datenbank überein → Eine Meldung wird angezeigt, dass das Passwort nicht korrekt ist **Abmelden** 2a. Der Benutzer konnte nicht identifiziert werden → Die Session des Benutzers ist abgelaufen ist somit bereits abgemeldet. → Er wird zur Home Page weitergeleitet	
Häufigkeit des Auftretens:	An- / Abmeldungsprozedur	

UC 03	Benutzer CRUD	
Umfang:	ICS ThreatMap Web Applikation	
Ebene:	Security, Anwendung	
Primärer Aktor:	Admin	
Vorbedingungen:	- Es sind User auf ICS ThreatMap registriert - Berechtigungen funktionieren einwandfrei, d.h. der Admin kann auch User bearbeiten	
Nachbedingung:	- Der zu bearbeitende User wurde in der Datenbank aktualisiert - Die übernommenen Änderungen müssen wirken. Z.B. Rechte herabstufen	
Hauptszenario: (User-/Systemverantwortlichkeiten)	Aktoren, Aktionen: 1. Der Admin möchte ein neuregistrierten User bearbeiten oder löschen 2. Admin navigiert im Menü zur *Konfiguration* und dann *Benutzer verwalten* 3. Admin sucht nach dem zu bearbeitenden User und wählt *Bearbeiten*	Systemverantwortlichkeiten: 4. Die Änderungen werden mittels vordefiniertem SQL Statements gespeichert. 5. Wenn erfolgreich, wird der Admin wieder zu der Benutzerliste weitergeleitet
	5a. Die Änderungen konnten nicht gespeichert werden, da die Emailadresse bereits vergeben ist und nur einmal vorkommen darf → Eine andere Emailadresse, welches noch nicht vorhanden ist, muss angegeben werden 5b. Der User (die Emailadresse) konnte in der Datenbank nicht gefunden werden → Die Seite der Benutzerverwaltung neu laden und nochmals versuchen 5c. Der Datenbankzugriff ist fehlgeschlagen → Sich an den Verantwortlichen wenden	
Häufigkeit des Auftretens:	Nach mindestens jeder Neuregistration	

UC 07	ICS CRUD	
Umfang:	ICS ThreatMap Web Applikation	
Ebene:	Anwendung	
Primärer Aktor:	Admin, Operator	
Vorbedingungen:	- Es sind Informationen für das Hinzufügen / Bearbeiten des ICS vorhanden - Grund für das Löschen eines ICS muss vorhanden sein	
Nachbedingung:	- Das ICS wurde erfolgreich in der Datenbank hinzugefügt - Bei einer Änderung wird ebenfalls ein neuer DB Eintrag erstellt - Nach dem Löschprozess darf das ICS nicht mehr verfügbar sein	
Hauptszenario: (User-/Systemverantwortlichkeiten)	Aktoren, Aktionen: 1. Der User fügt ein neues ICS hinzu, bearbeitet es oder möchte es löschen	Systemverantwortlichkeiten: 2. Beim Hinzufügen und Bearbeiten wird ein neuer Eintrag mit den angegebenen Daten erstellt und bei der Löschung muss das Flag *activated* von *true* auf *false* gesetzt werden
	2a. Es konnte nicht hinzugefügt, bearbeitet oder gelöscht werden, da benötigte Angaben nicht korrekt oder nicht vorhanden sind. → Eine Fehlermeldung wird den User aufmerksam machen → User muss die Angaben überprüfen und ergänzen 2b. Es konnte nicht hinzugefügt werden, da das ICS bereits existiert → Es wird eine Fehlermeldung mit dem bereits vorhandenem ICS angezeigt 5c. Es konnte nicht bearbeitet oder gelöscht werden, da das ICS gar nicht existiert → Die Seite muss neu geladen werden	
Häufigkeit des Auftretens:	Verwaltung der ICS	

UC 14	Datenbeschaffung mit UpdateScript	
Umfang:	Datenbank	
Ebene:	Datenbank	
Primärer Aktor:	Web-Server	
Vorbedingungen:	- MySql Dienst ist aktiviert und läuft einwandfrei - Datenbank und Schema sind vorhanden - Rechte für das Script ausführen sind gesetzt - Die API Page von Shodanhq.com muss verfügbar sein - - Es muss ein gültiger API-Key mit genügend Query-Punkten vorhanden sein	
Nachbedingung:	- Bereits vorhandene Daten müssen mit demselben Datum ignoriert werden. Falls der Eintrag von Shodan ein neueres Datum hat, muss dieser Eintrag als neuer Eintrag in die ICS Threat Map Datenbank gespeichert werden.	
Hauptszenario: (User-/Systemverantwortlichkeiten)	Aktoren, Aktionen: 1. Das Script wird auf dem Server mittels CronJob früh morgens oder spät abends ausgeführt	Systemverantwortlichkeiten: 2. Das Script führt jeden Filter aus und fügt neue Systeme in die Datenbank hinzu. Bereits vorhandene Systeme werden nicht mehr beachtet, wenn beide das gleiche Datum haben 3. Die neuen Systeme werden zusätzlich mit Whois-Informationen angereichert 4. Das Script wurde erfolgreich ausgeführt und beendet
	1a. Das Skript wurde nicht ausgeführt, da der CronJob keine Ausführrechte hat → Ausführrechte müssen gesetzt werden 2a. Es konnte keine Request mit diesem Filter durchgeführt werden, weil Shodanhq.com nicht erreichbar ist 1. Das Script wird nach 30 Sekunden Timeout abgebrochen, Email mit Durchführstatus wird versandt 2. später nochmals versuchen 3a. Die API-Key ist ungültig oder nicht genügend Query-Punkten sind vorhanden →Shodan Account überprüfen	
Häufigkeit des Auftretens:	einmal am Tag	

4. Weitere Anforderungen

4.1 Qualitätsmerkmale

Die ICS ThreatMap Applikation stützt sich auf die Qualitätsmerkmale gemäss ISO 9126. Nachfolgend werden alle Punkte erläutert und entschieden, in wie Fern die Merkmale für unsere Applikation wichtig sind.

4.1.1 Zuverlässigkeit

Die Datenbeschaffung, welche durch ein eigenes Script abgewickelt wird, muss so weit wie möglich zuverlässig laufen können, damit alle ICS automatisch laufend erfasst werden. Die Daten selber werden von den Mitgliedern oder von externen Engines übernommen. Damit die Vertraulichkeit gewährleistet werden kann, muss die Applikation einen Login-Schutz haben, damit nicht-autorisierte User keinen Einblick in die gesammelten Daten erhalten. Eine History zeigt die zuletzt getätigten Änderungen einer Engine oder des Benutzers an. Durch die History sind Änderungen für den Benutzer in der Applikation nachvollziehbar.

4.1.2 Benutzbarkeit

Die Applikation soll einfach und schnell erlernbar sein. Die jeweiligen Bereiche sollen beschrieben werden und unser Betreuer/Partner/Kunde hat beim Design ein Mitspracherecht. Die Applikation hält sich an Web Standards und ist mit Maus und Tastatur bedienbar.

4.1.3 Effizienz

Die Effizienz der Web Applikation wird anhand von diversen Studien gemessen. Dabei ist der maximale Richtwert 7 Sekunden. Antwortzeiten einer Web Applikation, die über 3 bis 4 Sekunden gehen, werden schon als störend empfunden. Bei Antwortzeiten über 8 bis 10 Sekunden versagt nicht nur die Geduld der Besucher, sondern unterbricht auch deren roten Faden, welche Tätigkeit er gerade nachgehen wollte.

Damit wir anständige Antwortzeiten erreichen können, werden während der Implementierung Refactorings vorgenommen. Grosse Datenmengen werden nicht auf einem Mal geladen. Wir setzen uns daher selber den Richtwert von 3 Sekunden, die eine Seite beim Laden maximal brauchen darf.

Gelassener sehen es wir bei der Datenbeschaffung. Die Datenbeschaffung wird mit Hilfe eines Scripts gemacht, welches täglich die ICS Daten holt. Da wir davon ausgehen, dass zukünftig vor allem Schweizer Nutzer die Web Applikation nutzen werden, wird das Script über Nacht ausgeführt. Ob das Script paar Minuten länger oder weniger lang läuft, ist dabei nicht entscheidend. Als eine einzige Bedingung sehen wir, dass das Script auch bei grossen Datenmengen in einer akzeptablen Zeit (nicht die ganze Nacht) ablaufen kann.

4.1.4 Wartbarkeit

Die Wartbarkeit der ICS ThreatMap Applikation ist ein wichtiger Bestandteil, da die Endversion auch weitergeführt wird. Aus diesem Grund sind auch die Modifizierbarkeit und die Skalierbarkeit wichtig. Dies betrifft einerseits die Programmierung der Applikation selber, jedoch auch die dahinterliegende Datenbank. Ausserdem sollte die Architektur der Applikation so konzipiert sein, dass unerwartete Wirkungen von Änderungen keinen Einfluss auf andere Teilkomponente haben. Dadurch sollen Unstabilität vermieden werden und Fehler in ihrem Problembereich isoliert sein.

4.1.5 Übertragbarkeit

ICS ThreatMap soll betriebssystemunabhängig betrieben werden können. Dabei ist die Übertragung der Applikation auf andere Web Server durchaus möglich. Ein Installationshandbuch soll dem Administrator beim Einrichten der Applikation behilflich sein.

4.2 Schnittstellen

4.2.1 Benutzerschnittstelle

Die Schnittstelle zwischen dem Benutzer und der Daten ist das grafische Web Frontend, welches auf HTML, PHP, JavaScript, JQuery, etc. basiert. Über das Frontend kann der Benutzer seine Anpassungen und Aktionen über Formulare, über die Karte oder über die Suchfunktion ansehen und verwalten.

4.2.2 Hardwareschnittstelle

Die MySQL und die Web Applikation, sowie das Datenbeschaffungsskript, werden auf demselben Web Server in Betrieb genommen.

4.2.3 Softwareschnittstelle

Es bestehen externe Schnittstellen zu den jeweiligen Engines wie Shodan, Google, etc., von denen die erforderlichen Daten geholt werden. Ausserdem wird die Zend Framework 2 Library (MVC Framework), Google Map und JqPlot Library (Chart Library) benötigt.

ICS ThreadMap - v1.0

Software Architektur Dokument (SAD)

Dominique Sorg
Benjamin Kehl

Änderungsgeschichte

Datum	Version	Änderung	Autor
03.11.2013	0.1	Erstellung & Einleitung	Dominique Sorg
08.11.2013	0.2	Kapitel Datenspeicherung	Benjamin Kehl
22.11.2013	0.3	Überarbeitung Systemübersicht	Dominique Sorg
22.11.2013	0.3	Datenbeschaffung	Dominique Sorg
22.11.2013	0.3	Benutzermanagement	Dominique Sorg, Benjamin Kehl
25.11.2013	0.4	Suchfunktion, Detailansicht	Dominique Sorg
26.11.2013	0.5	Klassen Diagramm der Module	Dominique Sorg
26.11.2013	0.6	Projektstruktur und Modulaufbau	Benjamin Kehl
18.12.2013	0.7	Überarbeitung	Dominique Sorg
19.12.2013	0.8	Korrekturen Dokument	Benjamin Kehl

Inhalt

1. Einführung

1.1 Zweck

Dieses Dokument ist ein Teil des Designs und beschreibt die Webarchitektur für das ICS ThreatMap Projekt. Hier werden die wichtigsten Entscheidungen beschrieben.

1.2 Gültigkeitsbereich

Dieses Dokument gilt während des gesamten Projekts und bildet dessen Grundlage.

2. Systemübersicht

ICS ThreatMap wird als Webprojekt realisiert, welches mit PHP auf Zend Framework 2 setzt. Dabei greift der Client über seinen Webbrowser auf die Webapplikation zu. Die Daten werden dynamisch über eine MySQL-Datenbank im Hintergrund abgefragt. ICS ThreatMap kann über diverse Browser wie Chrome, Firefox, Internet Explorer sowie auf diverse Endgeräten wie Smartphone, Computer oder Tablet aufgerufen werden. Wie auf der folgenden Abbildung ersichtlich ist, besteht die Architektur aus einem Client-Server System.

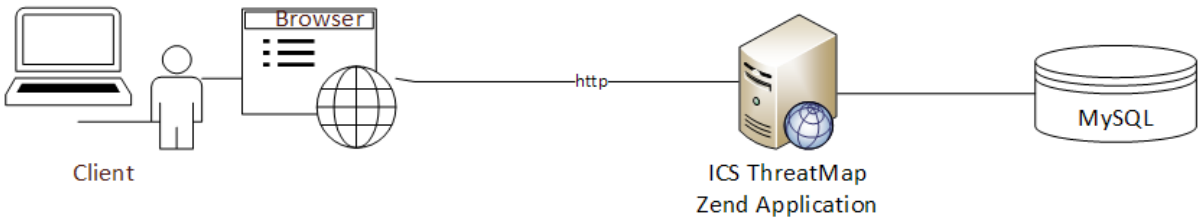


Abbildung 1 Übersicht ICS ThreatMap

Die Applikation ist in drei Hauptschichten nämlich in *Client Side*, *Server Side* und in *Engines* unterteilt. Diese werden in einzelnen Systembausteinen in den folgenden Abschnitten genauer beschrieben.

2.1 Server

Die von der HSR zur Verfügung gestellte virtuelle Maschine wurde mit Apache und mit MySQL ausgeliefert. Bereits vorinstalliert sind Redmine und Jenkins, wobei Redmine bereits im Rahmen dieser Arbeit eingesetzt wird. Für die produktive Applikation wurde eine Domäne namens *icsmap.ch* registriert, welche HTTP-Requests entgegennimmt und diese auf HTTPS umleitet.

2.1.1 Zend Framework 2

ICS ThreatMap ist praktisch komplett in PHP geschrieben und setzt Zend Framework 2 ein, welches ein objektorientiertes MVC-Framework anbietet. Dank des Apache-Moduls *mod_rewrite* kann über ZF2 die Seitennavigation bzw. die Routings selber verwaltet werden. Ein weiterer Vorteil von ZF2 ist die Unterteilung in Komponenten. So können riesige Funktionalitäten in einzelne Komponenten (Module) unterteilt werden, wie bspw. die Benutzerverwaltung, Suchfunktion oder das Datenbeschaffungsskript (auch UpdateScript). Die Kommunikation zwischen den Modulen ist trotzdem noch möglich und vermeidet somit duplizierten Code.

2.1.2 CronJob

Ein essentieller Teil der Applikation ist die Datenbeschaffung mit einem Skript. Das Update-Script wird über ein CronJob täglich um 04:00 Uhr früh morgens aufgerufen und sucht nach neuen ICS aus den verschiedenen Engines.

2.1.3 Datenbank

Die Datenbank ist für die Speicherung und Archivierung der ICS Daten zuständig. Dabei wird die Open-Source Plattform *MySQL* verwendet. Zend Framework 2 kann in verschiedenen Arten als Schnittstelle zwischen Datenbank und Businesslogik agieren. Dabei wurden hauptsächlich zwei Arten genutzt: Für eine Tabelle wurde das TableGateway-Prinzip verwendet und bei mehreren Tabellen ein Adapter, damit gezielte SQL-Abfragen generiert werden konnten.

2.2 Client

Wie bereits erwähnt greift der Client über seinen Browser auf die ICS Threat Map Applikation zu. Damit dies überhaupt möglich ist, benötigt dieser HTML 5, CSS 3, JQuery und JavaScript für die Darstellung der Google Map und der jqPlot Statistiken.

2.3 Engines

Die Daten werden über das Update-Script direkt aus den jeweiligen Engines wie Shodan oder Google geholt. Da wir aus der ICS Analyse das Fazit gezogen haben, dass die Zuverlässigkeit der Kontaktdaten aus Shodan oder Google nicht immer gewährleistet oder teils gar nicht vorhanden war, entschieden wir uns, selber Whois Abfragen aus den gegebenen Daten zu tätigen. Die Abfragen dienen dem Nutzer als Möglichkeit für eine erste Kontaktaufnahme mit betroffenen ICS Betreibern. Betreibern, die hinter einem Provider sitzen, können keine Kontaktadresse angeboten werden. Die Kontaktadresse kann aber von einem ICS ThreatMap Nutzer manuell hinzugefügt werden. Damit Whois Abfragen überhaupt möglich sind, muss Port 43 geöffnet sein.

3. Logische Architektur

Die logische Architektur besteht aus vier Schichten: *Presentation Layer*, *Business Infrastructure Layer*, *Data Access Layer* und *Data Layer*.

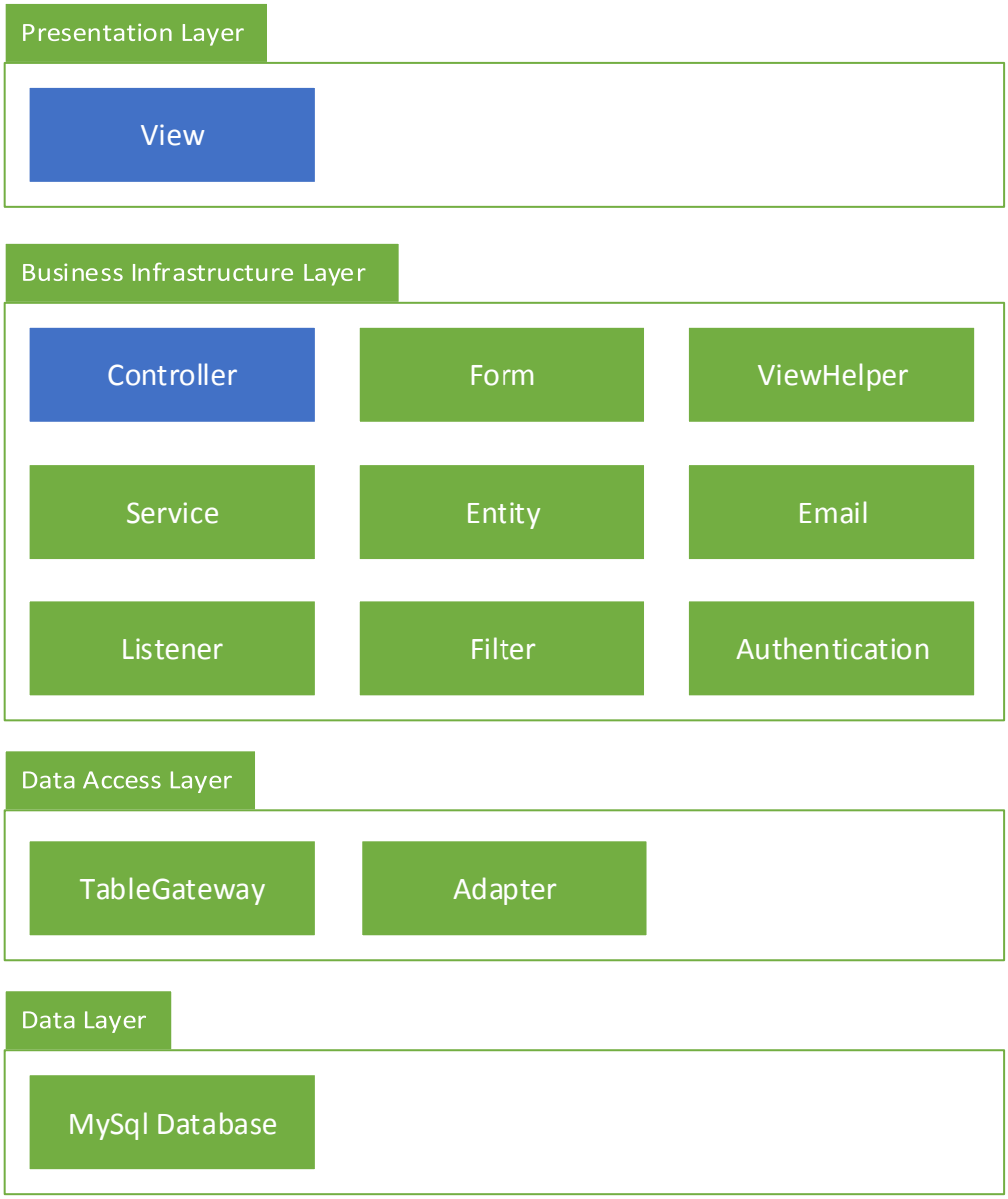


Abbildung 2 Logische Architektur in Schichten

Die blau gekennzeichneten Elemente (siehe Abbildung 2) entsprechen dem MVC-Prinzip, obwohl das Model gar nicht vorhanden ist. Das fehlende Model ist im Sinne nichts anderes als die restlichen Elemente im *Business Infrastructure Layer*, *Data Access Layer* und das *Data Layer*, da diese nahezu von Grund auf entwickelt werden müssen.

3.1 Presentation Layer

Im *Presentation Layer* ist das gesamte Frontend der Webapplikation abgelegt. Dazu gehören das Layout, Design und sonstige öffentlich zugängliche Ressourcen wie JavaScript und CSS.

3.2 Business Infrastructure Layer

Der Business Infrastructure Layer enthält die gesamte Logik, wie die Daten zu verstehen, interpretieren und zu repräsentieren sind.

- Controller
  Die Hauptaufgabe der Controller ist es, zu kontrollieren und zu delegieren. Der Controller empfängt in einer typischen Anfrage in einer MVC-Web-Architektur Benutzereingaben. Diese Benutzereingaben werden in Aktionen eines oder mehrerer Models übersetzt und liefern eine Ausgabe zurück, die von der View auf Basis der Ergebnisse der Model-Aktionen dargestellt werden.

- Service
  Der Service dient als Schnittstelle zwischen dem Controller und den verschiedenen Model-Elementen wie Form, Entity, Email, Authentication, usw.

- ViewHelper
  Im ViewHelper werden die vom Controller übergebenen Daten dargestellt.

- Entity
  Die Entities stellen in einer Datenbank eine Tabelle dar. Der TableGateway nutzt die Entities um Daten auf die gegebene Tabellen zu mappen.

- Listener
  Im Listener sind alle Objekte abgelegt, die bei jedem Seitenaufruf eine Aktion treffen müssen.

3.3 Data Access Layer

Das Data Access Layer dient als Schnittstelle zwischen der Datenbank und der Business Logik (Business Infrastructure Layer).

- TableGateway
  Der TableGateway ist ein Design-Pattern, in dem ein Objekt den direkten Zugang zu einer Tabelle in der Datenbank ermöglicht. Das ZF2 stellt unter *Zend\Db\TableGateway* ein TableGatway zur Verfügung.

- Adapter
  Um spezifischere Abfragen zu ermöglichen, die über mehrere Tabellen hinweg gehen, wurde ein einfacher Adapter verwendet. Unter *Zend\Db\Adapter* kann auf einen solchen Adapter zugegriffen werden.

3.4 Data Layer

Der Data Layer enthält alle Daten und repräsentiert die Datenbank MySql.

3.5 Model View Controller

Zend Framework 2 verwendet das Model-View-Controller-Prinzip (MVC), um auf stark strukturierte Weise die Zuständigkeitsbereiche einer Anwendung zu trennen. Dies hat insbesondere in grossen Web-Projekten den Vorteil, dass der Code nicht unter seinem eigenen Gewicht zusammenbricht und zum berüchtigten „Spaghetti-Code“ wird, wie es bei vielen desorganisierten PHP-Anwendungen der Fall ist. Mit Hilfe des MVC befolgt der Code Prinzipien wie „Don’t Repeat Yourself“ (DRY) und „Keep It Simple Stupid“ (KISS), welche von chaotischer Codeduplizierung abraten und Wiederverwendbarkeit und Wartbarkeit fördern.

4. Projektstruktur

Die Projektstruktur besteht aus mehreren Verzeichnissen. Diese Verzeichnisse stammen von Zend Framework 2 und dienen zur logischen Gliederung von bestimmten Aufgaben.

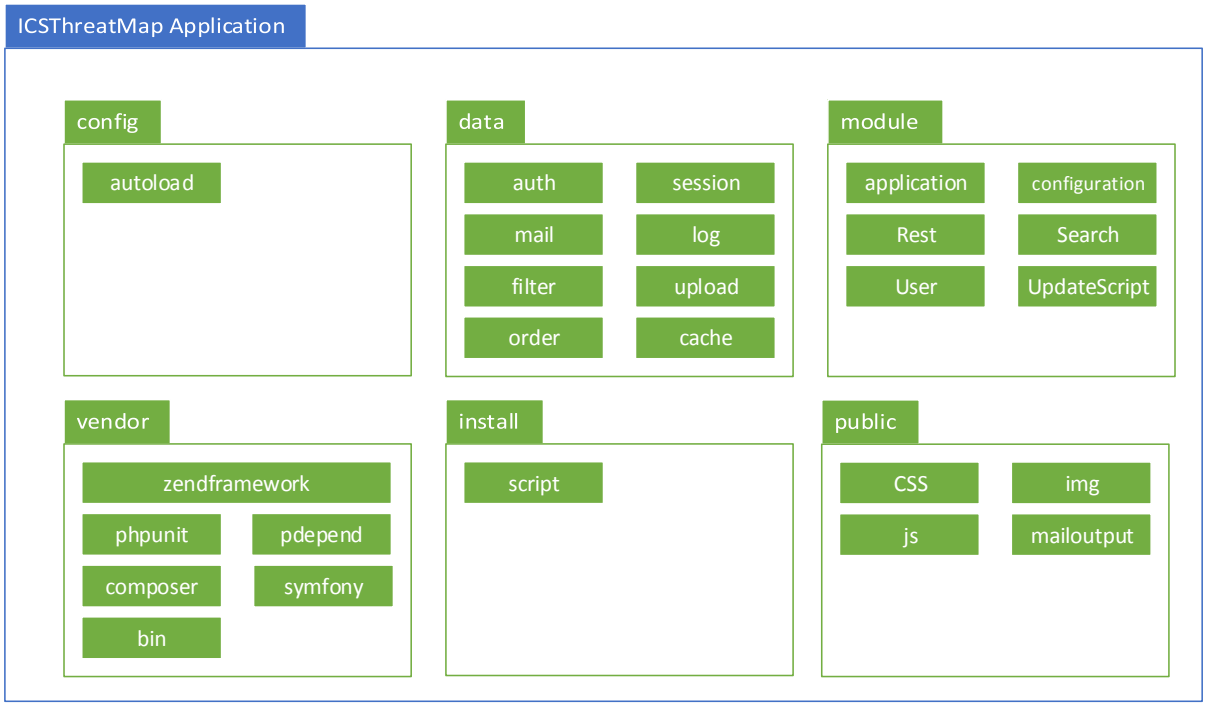


Abbildung 3 Projektstruktur ICS Threat Map

- /config
  Dieses Verzeichnis enthält die Konfigurationsdaten der Anwendung. Diese Konfigurationsdaten wie bspw. Datenbankanbindung werden automatisch geladen und können aus den Modulen überschrieben werden.
- /data
  Im Datenverzeichnis werden alle vergänglichen Daten gespeichert, wie Logs, Cache, Sessions-Daten usw.
- /module
  Dieses Verzeichnis ist der Ablageort für alle Anwendungsmodule wie die Benutzerverwaltung (User), Datenbeschaffung (Update-Script), Webseite (Application) oder die Suchfunktion (Search).
- /vendor
  Im Vendor-Verzeichnis werden jegliche Fremdmodule abgelegt. Das Fremdmodul „Composer“ wird für die Installation von Abhängigkeiten genutzt, wie z.B. das neueste Zend Framework 2 aus Github zu installieren. Das Bin-Verzeichnis dient zur Ausführung von PHP-Scripts aus der Kommandozeile. Das wichtigste Fremdmodul jedoch ist die Zend Framework Library.
- /install
  Das Installationverzeichnis beinhaltet die Installationsdaten für die ICS Threat Map Anwendung.
- /public
  Das Public-Verzeichnis enthält alle öffentlich abrufbaren Dateien. Diese sind Grafiken wie CSS- und JavaScript-Dateien oder die Front-Controller-Datei „index.php“, welches die Konfigurationen und Module der Webanwendung lädt.

5. Modulaufbau

Das Modulverzeichnis ist ebenfalls stark in verschiedene Komponenten und Aufgaben gegliedert. Ein möglicher Aufbau ist in Abbildung 4 Modulaufbau zu sehen.

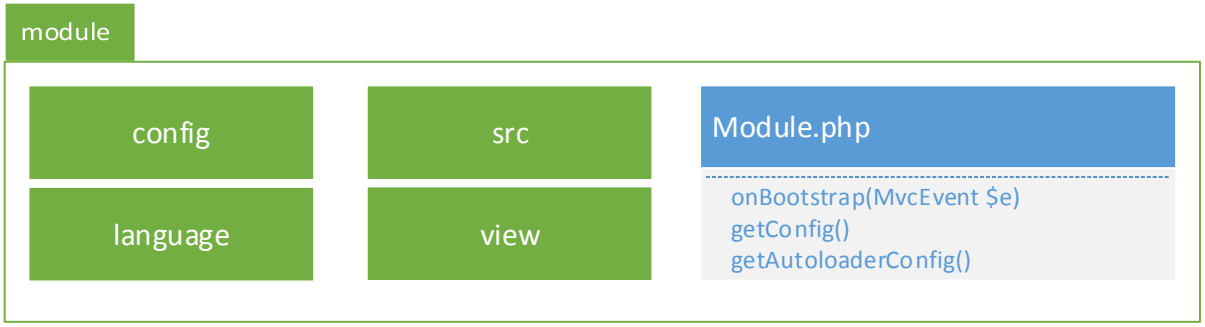


Abbildung 4 Modulaufbau

- /config
  Auch die Module besitzen ein Konfigurationsverzeichnis. Dieses Verzeichnis besteht meist aus einer PHP-Datei, welche das Routing des Moduls als Array zurückgibt.
- /src
  Im Source-Verzeichnis werden alle Klassen wie z.B. Controller, View, View-Helper, Services, usw. abgelegt (siehe 4. Projektstruktur).
- /language
  Für die Internationalisierung können auch Sprachdateien abgelegt werden. Diese wurden im Application-Modul (siehe Abbildung 3 Projektstruktur ICS Threat Map) abgelegt und standardmässig auf Deutsch eingestellt.
- /view
  Das View-Verzeichnis des Moduls wird für alle View-Scripts und Layout-Scripts verwendet.
- Module.php
  Die Datei *Module.php* ist der Kern des Moduls. Sie dient zum Laden des ganzen Moduls und muss somit zwingend enthalten sein. In der Regel beinhaltet die Datei drei Methoden: die *onBootstrap*-Methode wird nach jedem Seitenaufruf aufgerufen und hat den Vorteil, dass in diesem Modul Klassen geladen werden, die zwingend benötigt werden, z.B. die Benutzerrechte überprüfen. Die Methode *getConfig()* ist für das Laden der Konfigurationsdaten zuständig. Diese Daten werden in der Regel im Verzeichnis *config* abgelegt.
  In der Methode *getAutoloaderConfig()* wird das Autoloading für das Modul konfiguriert. Dabei wird der Standard-Autoloader verwendet, der die zu findenden Klassen im */src/Application* des Moduls suchen soll.

6. Factory-Pattern

In den meisten Modulen wird das Konzept der Factories verwendet. Das Factory-Objekt holt sich dabei über den Service Manager von ZF2 die benötigten Objekte, erstellt ein neues spezifisches Objekt und übergibt die geholten Objekte an das spezifische Objekt. Dieses Prinzip wird in jedem Modul angewandt. Im folgenden Beispiel soll das Prinzip im User-Modul veranschaulicht werden.

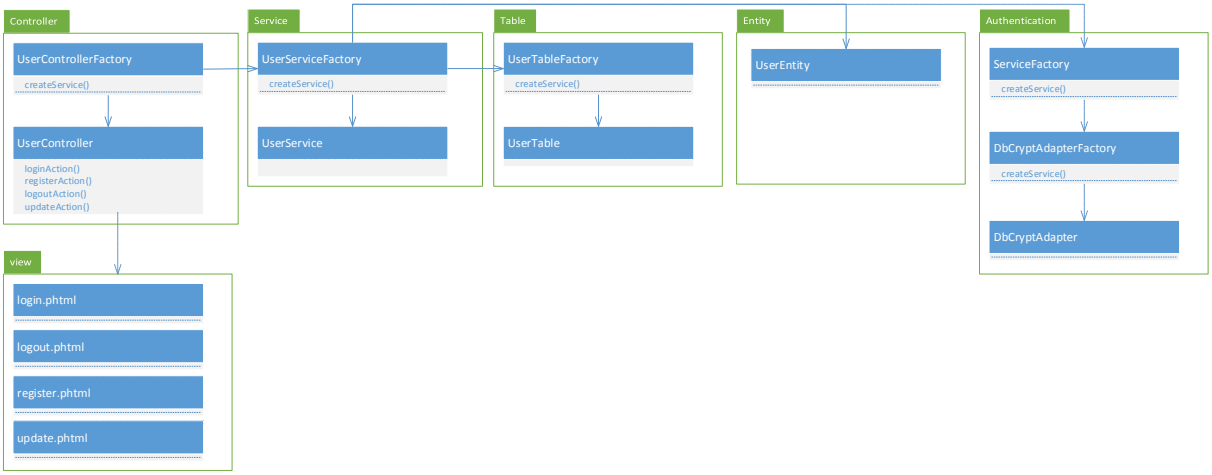


Abbildung 5 Factory Prinzip

In Abbildung 5 holt der *UserControllerFactory* eine Instanz vom *UserServiceFactory*. Der UserServiceFactory holt sich wiederum eine Instanz von einem *UserEntity*, *UserTableFactory* und von einem *ServiceFactory*. Diese Instanzen gibt der UserServiceFactory an dem *UserService* als Parameter weiter. Somit besitzt der UserService, der als *ModelService* fungiert, alle Instanzen und kann somit auf alle Operationen zugreifen. Der UserServiceFactory gibt das UserService-Objekt an dem UserControllerFactory zurück, dieser erstellt ein neues Objekt von einem UserController und übergibt die Instanz vom UserService an ihm weiter. Somit hat der UserController Zugriff zum UserService und kann so auf alle Operationen zugreifen und die Daten an die Views weitergeben.

7. Datenbeschaffung

Die Datenbeschaffung erfolgt durch ein Skript. Dieses Skript holt alle Suchfilter aus der MySQL Datenbank. Die Filter befinden sich auf Grund des flexiblen Hinzufügens und Entfernens weiterer Suchfilter von Clients in der Datenbank. Mit Hilfe der DB werden die erhaltenen Filter überprüft, ob diese von einem Client neu hinzugefügt worden sind oder ob diese schon länger vorhanden sind. Bei einer ersten Ausführung eines Filters wird eine komplette Datenbeschaffung bei der jeweiligen Engine angefragt und verarbeitet. Filter, die schon mindestens einmal ausgeführt worden sind, müssen nur noch Updates empfangen. Dies ist bspw. bei Shodan möglich, indem man das Datum mitschickt. Somit werden Queries gespart und Leerläufe vermieden (Alle Änderungen können mit unserer DB abgeglichen werden).

Falls sich aber schon ein älteres ICS System sich schon in unserer Datenbank befindet, wird der alte Eintrag als Archiv gesetzt und die bestehenden Referenzen zum neuen ICS hinzugefügt sowie in die DB eingetragen.

Die Ordnerstruktur der Datenbeschaffung ist unterteilt in Controller, Model, Service, Mail, Statistik und Data.

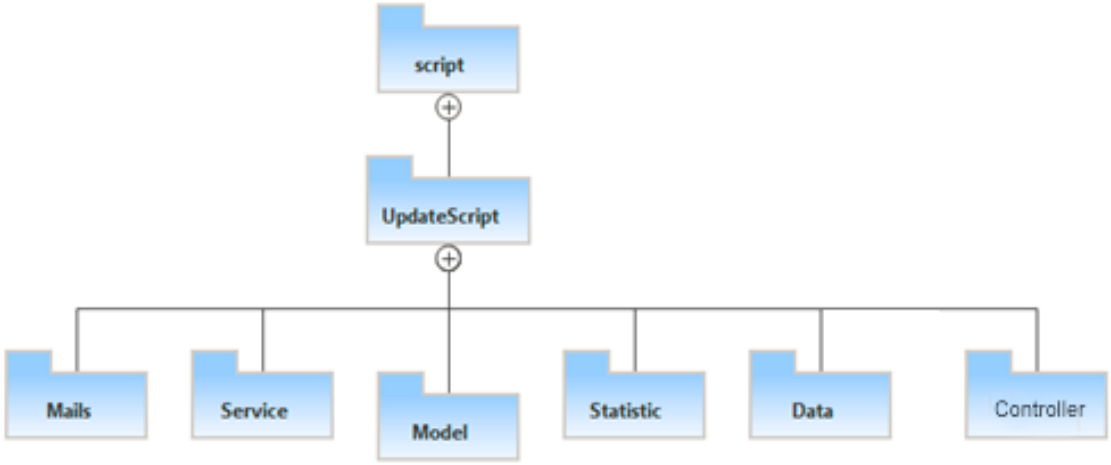


Abbildung 6 Ordnerstruktur UpdateScript

Folgende Grafik visualisiert den Ablauf als Sequenzdiagramm.

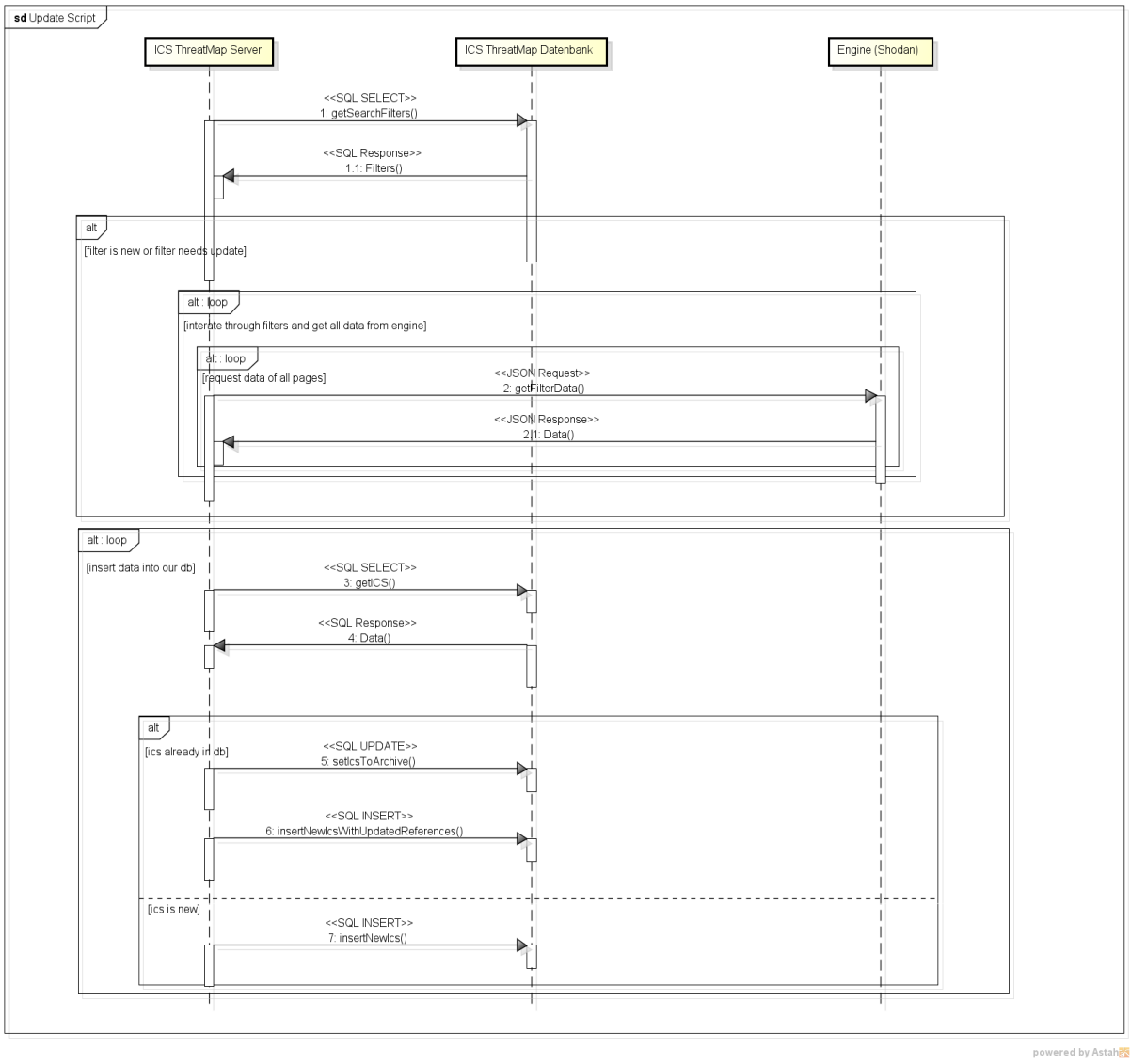


Abbildung 7 Sequenzdiagramm Datenbeschaffung

8. Benutzermanagment

Ein gutes Benutzermanagement spielt für die ICS ThreatMap Webapplikation eine sehr wichtige Rolle, da auf der Webseite sensible Daten vorhanden sind, die nicht für alle zugänglich sein dürfen. Deshalb werden in diesem Kapitel das Konzept der Benutzerrollen und deren Rechte, sowie der Zugriff näher beschrieben und aufgezeigt.

Das Modul der Benutzerverwaltung wurde folgendermassen gegliedert.

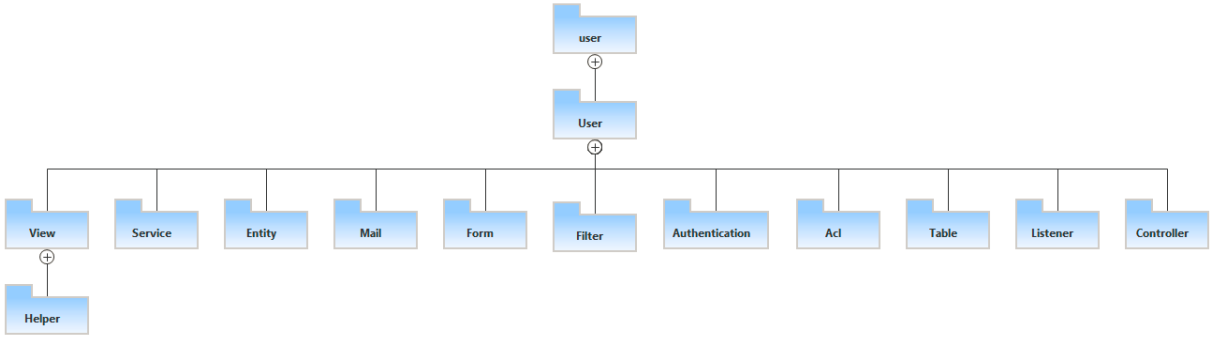


Abbildung 8 Modulstruktur User

8.1 Idee

Der Zugriff auf die ICS ThreatMap erfordert keine Registrierung, jedoch werden die Daten sehr beschränkt angezeigt. Das heisst, der Besucher kann die ICS auf der Karte ansehen, jedoch werden ihm keine Details für die jeweiligen ICS angezeigt. Ausserdem sind ihm nur gewisse Seiten wie das *Impressum*, *Über uns* oder *Kontakt* zugänglich. Den erweiterten Zugriff erhält der Besucher durch eine Registrierung. Nach einer erfolgreichen Registrierung wird der Besucher aufgefordert zu warten, bis der Administrator seinen Account aktiviert hat. Sobald der Administrator den neuen Account aktiviert hat, kann der Besucher sich anmelden.

8.2 Benutzerrollen / -rechte

Es werden insgesamt vier Benutzerrollen unterschieden: *Gast*, *Mitglied*, *Operator* und *Administrator*.

Ressourcen \ Rollen	Gast	Mitglied	Operator	Administrator
Starseite Chart und Login	X	X	X	X
Impressum, About Us, Contact Pages	X	X	X	X
Startseite Map	X*	X	X	X
Startseite Suche		X	X	X
ICS Inhalte ansehen		X	X	X
Nach ICS suchen		X	X	X
ICS verwalten			X	X
Filter ansehen und verwalten			X	X
Tickets erstellen		X	X	X
Tickets verwalten			X	X
Neue Inhalte hinzufügen (Kategorien, Typ, ISP, etc.)			X	X
Benutzer verwalten				X

* Restricted Permissions: Die Daten werden nur sehr beschränkt für den Gast angezeigt, d.h. nur die Standorte und Bedrohungsausmass sind für den Gast sichtbar.

Wie in der obigen Tabelle ersichtlich ist, hat der *Gast* keine Berechtigungen auf die Daten, sondern wie bereits erwähnt, kann er nur die ungefähren Standorte der ICS auf der Map ansehen. Der Gast ist sozusagen ein Benutzer, der sich noch nicht angemeldet hat.

Aus Sicherheitsgründen wird nach jeder Registration den User als Default die Rolle *Mitglied* zugeteilt. Somit hat das Mitglied keine Berechtigung bestehende Ressourcen zu ändern oder hinzuzufügen.

Die O*peratoren* sind die aktiven Mitglieder auf der Webapplikation und haben auf fast alle Ressourcen Zugriff.

Auch der *Administrator* gehört zu den aktiven Mitgliedern. Dessen Aufgabe besteht insbesondere aus der Überprüfung und Festlegung der Rechte von neuen Mitgliedern.

8.2.1 Überprüfung der Benutzerrollen

Die Überprüfung der Benutzerrollen wird über ein View Helper erreicht. Dabei kann festgelegt werden, ob es sich um ein *Admin* handelt, dann sollen alle Daten angezeigt werden, andernfalls nur beschränkt. Das Sequenzdiagramm (in Abbildung 9 Sequenzdiagramm Application View Scope) zeigt den Zugriff auf Inhalte, Kartendaten und auf Statistikdaten. Dabei erhalten Gäste reduzierten Zugriff auf die Inhalte und Kartendaten, jedoch vollen Zugriff auf die Statistikdaten.

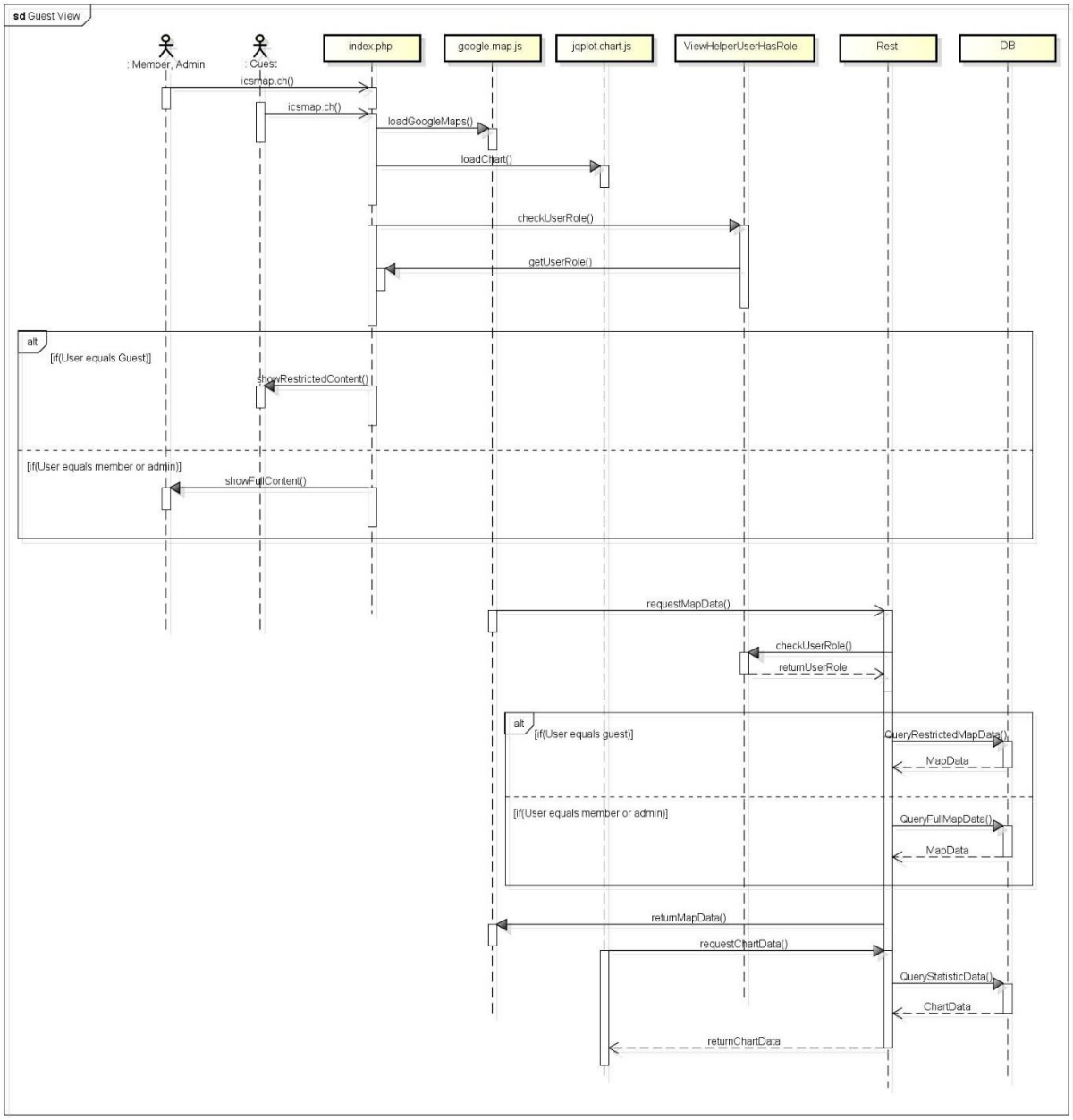


Abbildung 9 Sequenzdiagramm Application View Scope

Ein Gast und ein Admin laden icsmap.ch. Dabei werden nebst der Seite auch die GoogleMaps und die Charts von den jeweiligen Javascript Libraries geladen. Der View Helper *UserHasRole* liefert einen String zurück, um welche Rolle es sich handelt. Falls es sich um einen Gast handelt, dann soll die Methode *showRestrictedContent()* aufgerufen werden und sonst *showFullContent()*.

8.3 Listener

Es kann vorkommen, dass ein Benutzer auf eine Seite zugreifen möchte, für die er keine Rechte hat. In einem solchen Fall wird dem Benutzer erklärt, dass er auf die angeforderte Seite nicht zugreifen darf. Gäste werden aufgefordert, sich einzuloggen. Diese Aufgabe übernimmt der *UserListener.php*. Bei jedem Rendering einer Seite überprüft der UserListener, ob der Benutzer die erforderlichen Rechte hat. Falls nicht, wird der Benutzer auf die Login Page umgeleitet. Im folgenden Sequenzdiagramm soll dies genauer erläutern.

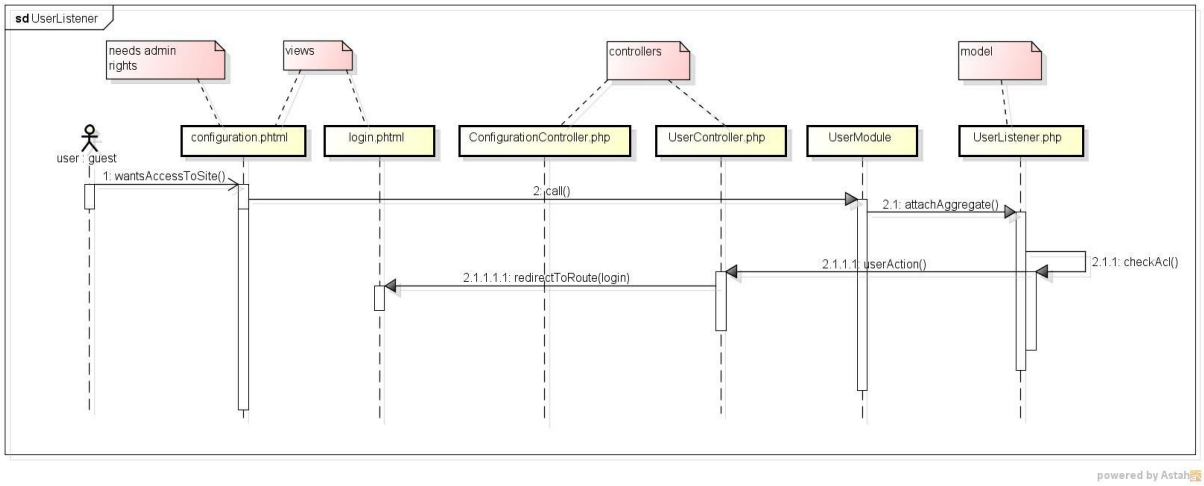


Abbildung 10 User Listener Guest

Ein Gast möchte auf die Konfigurationsseite zugreifen. Diese Seite ist jedoch nur für Admins gedacht. Der Ablauf funktioniert folgendermassen: Das Laden wird vom Render-Event gesteuert. Das UserModule registriert mittels *attachAggregate()* den UserListener beim Render-Event. Dies hat nun zur Folge, dass bei jedem Seitenaufruf der UserListener aufgerufen wird, welcher die Methode *checkAcl()* besitzt. Diese Methode prüft die Berechtigungen der jeweiligen User. Da es sich bei diesem User um einen Gast handelt, leitet der UserListener auf die Aktionsmethode *userAction()* um, welches die *login.phtml* Seite darstellt.

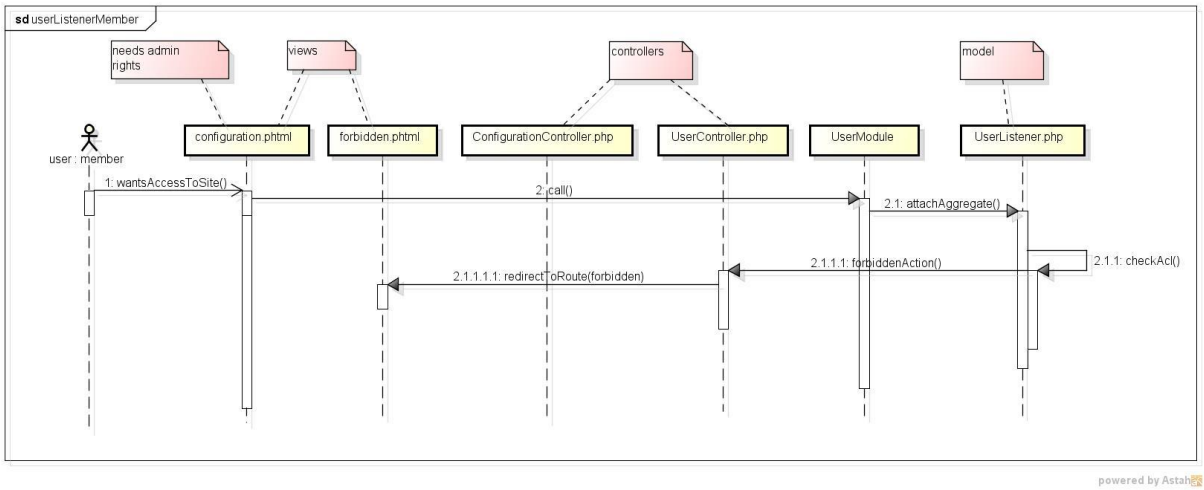


Abbildung 11 User Listener Member

In *Abbildung 11 User Listener Member* wird die Situation als *Member* mit beschränkten Rechten gezeigt. Da ein Member ebenfalls keinen Zugriff auf die Ressource Konfiguration hat, wird er ebenfalls auf eine andere Aktionsmethode umgeleitet. Und zwar auf *forbiddenAction()*, dessen ViewModel *forbidden.phtml* ist.

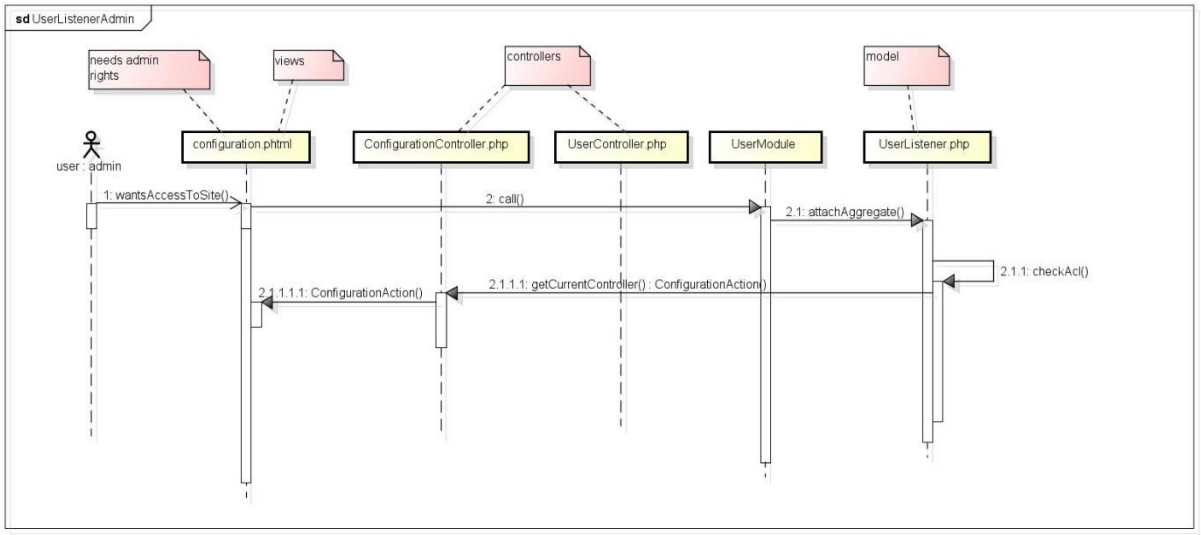


Abbildung 12 User Listener Admin

Zu guter Letzt ein Beispiel mit dem Admin. Da der Admin für diese Seite berechtigt ist, besteht er die Überprüfung von *checkAcl()*. Somit wird der aktuell für die Seite benötigte Controller geholt und aufgerufen. In diesem Fall ist es der *ConfigurationAction()*, welcher als Nächstes das ViewModel für *configuration.phtml* erstellt.

8.4 Identifizieren eines Benutzers

Zend Framework 2 stellt mit *Zend\Session* Container einen einfachen Weg zum Speichern von Daten in Sessions bereit. Dadurch ist es möglich jeden angemeldeten Benutzer einfach und schnell zu identifizieren und spezifische Aktionen zu treffen. Das Sessionhandling wird sehr häufig während der Registration-, Login- und Logout-Prozedur verwendet, um gleich zu Beginn den Benutzer zu identifizieren. Dies hat den Vorteil, dass bereits für angemeldeten Benutzern, direkt gewisse Aktionen, wie sich anmelden oder registrieren unterbunden werden oder nichtangemeldete Benutzer (auch als Gast bezeichnet) auf die Login-Seite umgeleitet werden.

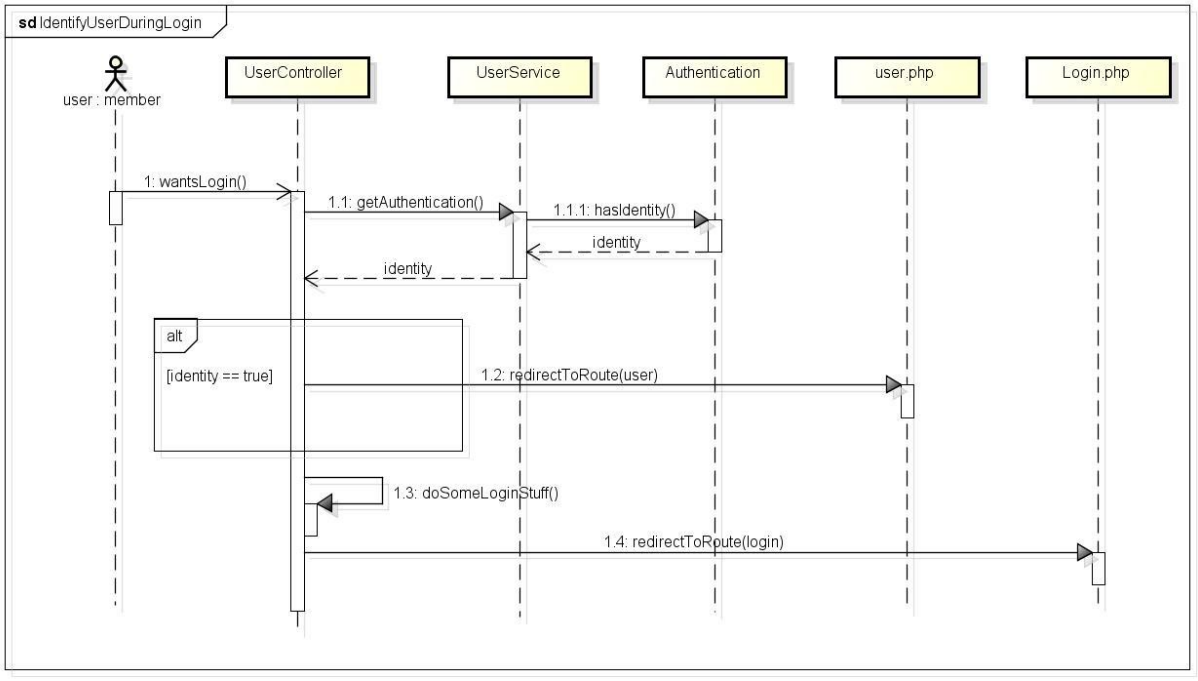


Abbildung 13 Sequenzdiagramm Benutzer identifizieren während einer Anmeldung

Das Sequenzdiagramm in Abbildung 13 zeigt einen angemeldeten Benutzer, der sich nochmals anzumelden versucht. Der User-Controller fragt über den User-Service die Identität des Benutzers von der aktuellen Session ab. Dieser liefert einen Boolean-Wert zurück. Falls der Benutzer identifizierbar ist, wird er auf die Willkommensseite umgeleitet, ansonsten auf die Login-Seite, da es sich um einen nichtangemeldeten Benutzer handelt.

8.5 Benutzeraktionen

Dieses Verfahren mit der Identifizierung eines Benutzers, wurde bei einer Registration vorgestellt und wird auch im Bereich *Login* und *Logout* angewendet. In den Sequenzdiagrammen werden diese für die bessere Lesbarkeit und aus Platzgründen nicht mehr angezeigt.

8.5.1 Register

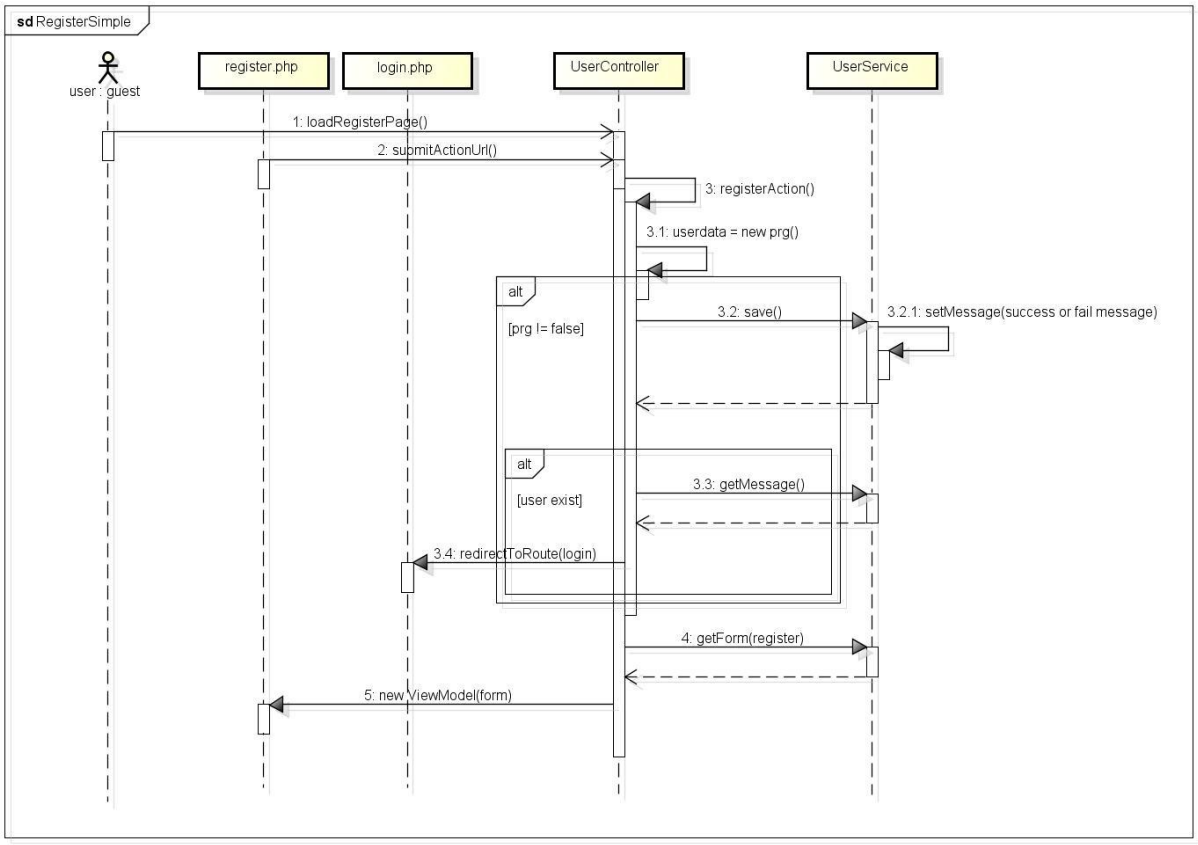


Abbildung 14 Sequenzdiagramm User Registration

Wenn der Benutzer die Registrationsseite aufruft, lädt der User Controller die *registerAction()*-Methode (die *submitActionUrl()*-Methode kann momentan noch vernachlässigt werden). In *registerAction()* werden die Formularelemente geholt und ein *PostRedirectGet-Plugin* (auch PRG-Plugin genannt) wird instanziiert. Da durch das Neuladen einer Seite eine POST-Anfrage erneut versandt werden kann, soll das Plugin dieses Problem unterbinden, indem er die POST-Daten in einer Session speichert, die Anfrage per GET mit HTTP-Status 303 umleitet und die Daten aus der Session wieder ausliest.

Nachdem das Plugin instanziiert wurde, weist es den Wert *false* auf, da beim Öffnen der Seite noch keine Daten übergeben wurden. Deshalb wird die Bedingung, ob das Plugin nicht falsch sei, übersprungen. Der *User Service* holt das passende Formular für die Registration mittels *getForm(register)* und übergibt es an das View-Model. Ab diesem Zeitpunkt sieht der Benutzer das Formular und kann seine Daten eingeben.

Wenn der Benutzer seine Daten eingegeben und bestätigt hat, ruft das Action-Attribut des Formulars, die Seite erneut auf und übergibt an das PRG-Plugin die Formulardaten des Benutzers (siehe *submitActionUrl()*-Methode). Nun hat das Plugin nicht mehr den Wert *false* und geht in die Bedingung hinein. In dieser Bedingung wird über den User Service, die Methode *save()* aufgerufen, welche zuerst einige syntaktische Überprüfungen macht, ob der Benutzer die Daten richtig eingegeben hat. Bei Erfolg erstellt er einen neuen Benutzer und speichert ihn in die Datenbank ab. Ausserdem bekommt der User Controller den Benutzer als Rückgabewert und kann anschliessend den Benutzer auf die Login-Seite mit einer Erfolgs-Meldung umleiten. Falls die Prüfung fehlschlägt, wird der Wert *false* an den User Controller zurückgeliefert und die Formularelemente werden nochmals geladen.

8.5.2 Login

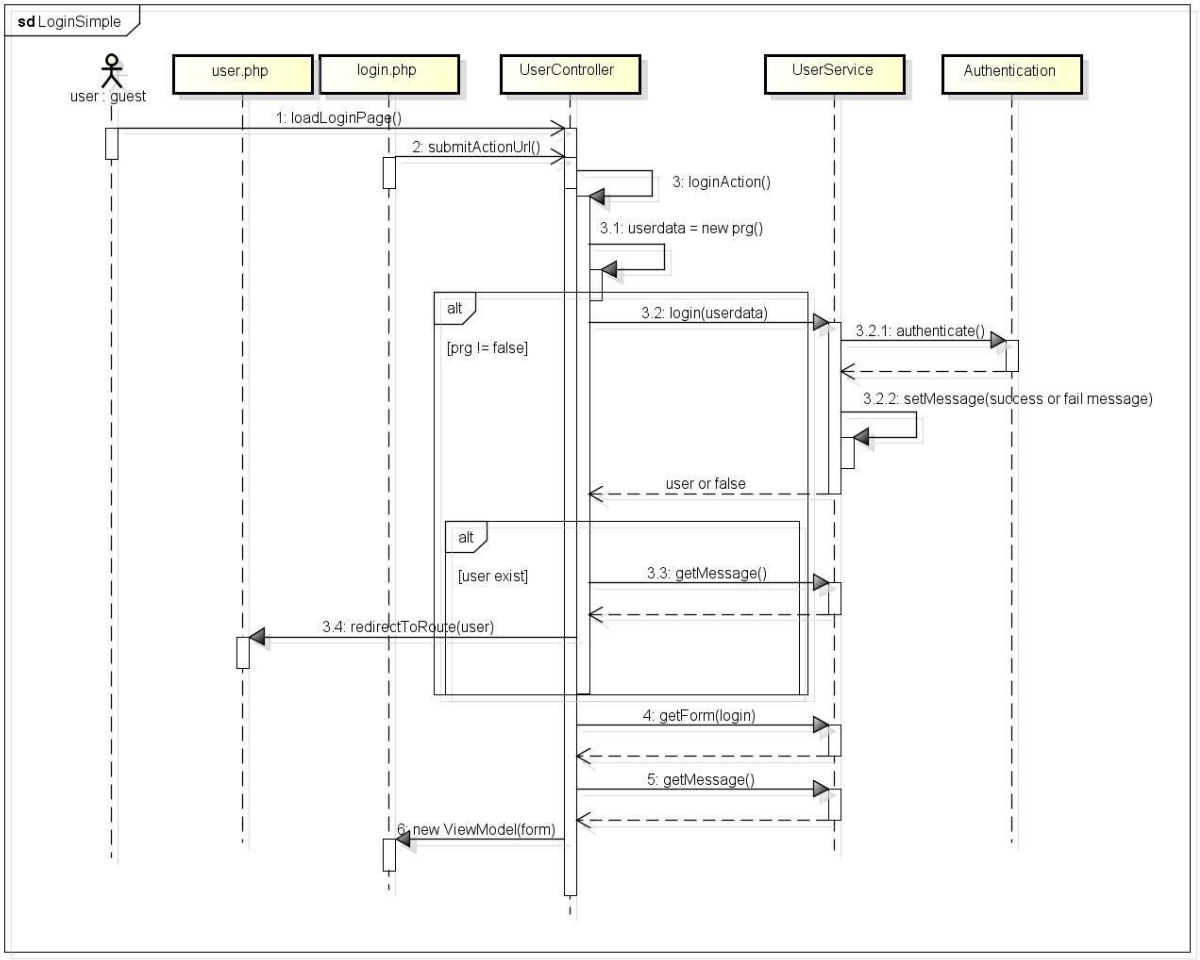


Abbildung 15 Sequenzdiagramm User Login

Das Prinzip bei der Anmeldung ist dieselbe wie bei der Registration. Der User Controller lädt zuerst das PRG-Plugin und holt sich das richtige Formular mittels *getForm(login).* Danach übergibt er ein neues View Model mit dem Formular zurück. Ab diesem Zeitpunkt sieht der Benutzer das Formular und kann seine Credentials eingeben.

Sobald der Benutzer seine Credentials bestätigt, werden die Daten über das PRG-Plugin an die Login-Methode übergeben. Diese wiederum, überprüft anhand der Filterkriterien, ob z.B. die E-Mail-Adresse syntaktisch richtig eingegeben wurde. Danach wird nach der Gültigkeit der Credentials mit Hilfe der Methode *authenticate()* überprüft und bei Erfolg bekommt der User Controller über die Login-Methode die Identität zurückgeliefert, damit er den Benutzer auf seine Willkommensseite

umleiten kann. Falls die Authentifizierung oder Überprüfung der Formulardaten fehlschlagen, wird dem Benutzer eine Nachricht angezeigt, dass er seine Angaben nochmals überprüfen solle.

8.5.3 Logout

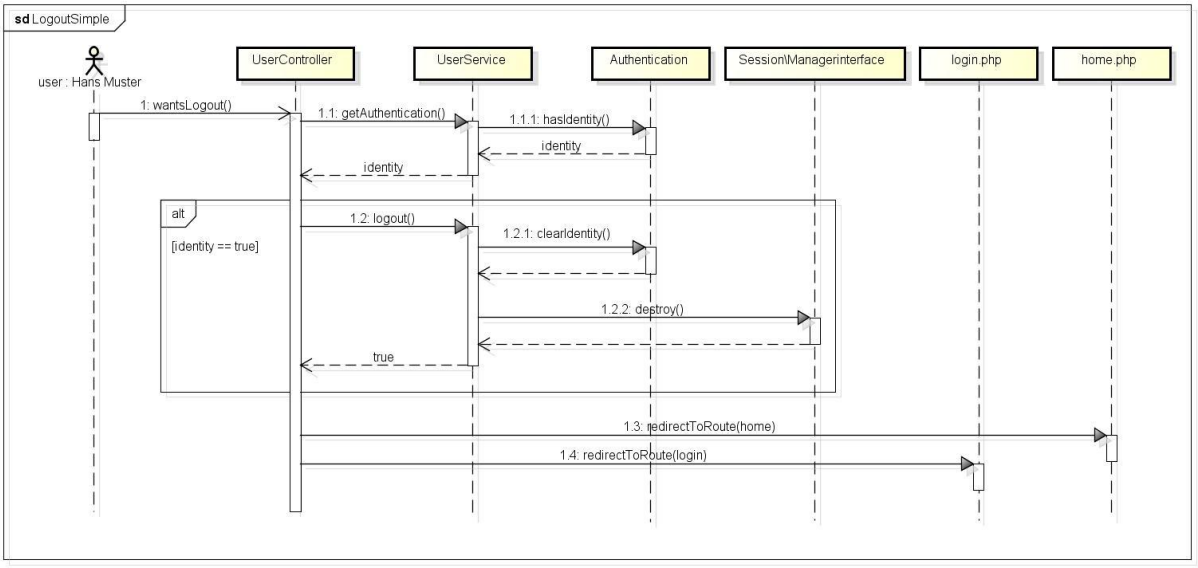


Abbildung 16 Sequenzdiagramm User Logout

Während dem Logout-Prozess, überprüft der User Controller um welchen Benutzer es sich handelt. Falls der Benutzer mit *hasIdentity()* identifizierbar ist, wird in der *logout()*-Methode die bestehende Session mittels *destroy()* gelöscht und dann zur Startseite umgeleitet. Falls der Benutzer nicht identifizierbar ist, handelt es sich um einen nicht eingeloggten Benutzer mit der Rolle Gast. Dieser Benutzer wird folglich direkt zu der Login-Seite umgeleitet.

9. Suchfunktion

Die Suchfunktion liefert spezifischere ICS Abfragen. Dabei können über das Suchformular ein Filter gesetzt werden, um die Suche so genau wie möglich einzuschränken.

Die Suchfunktion wurde folgendermassen gegliedert.

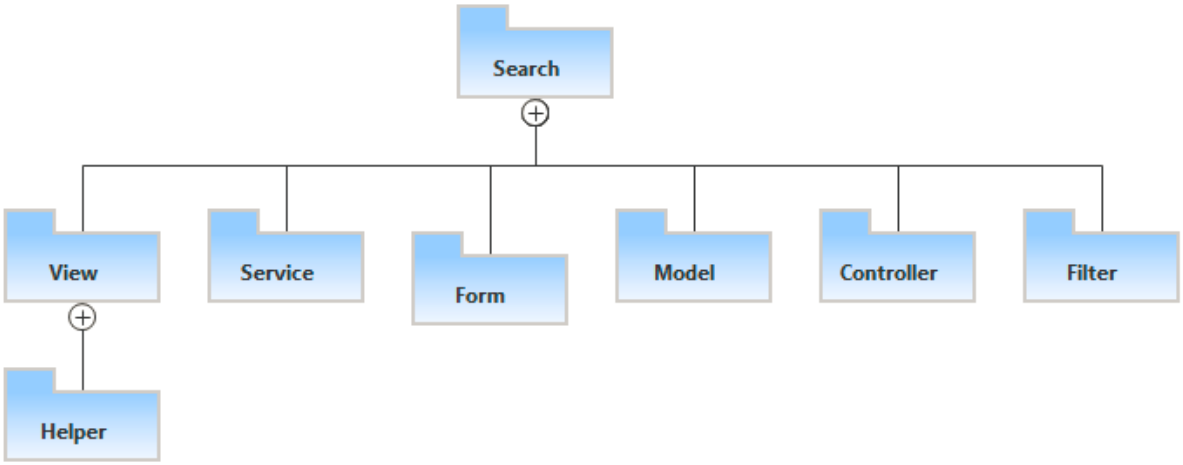


Abbildung 17 Modulstruktur Search

Im folgenden Sequenzdiagramm wird aufgezeigt, wie das Formular geladen wird und wie eine Suche abläuft.

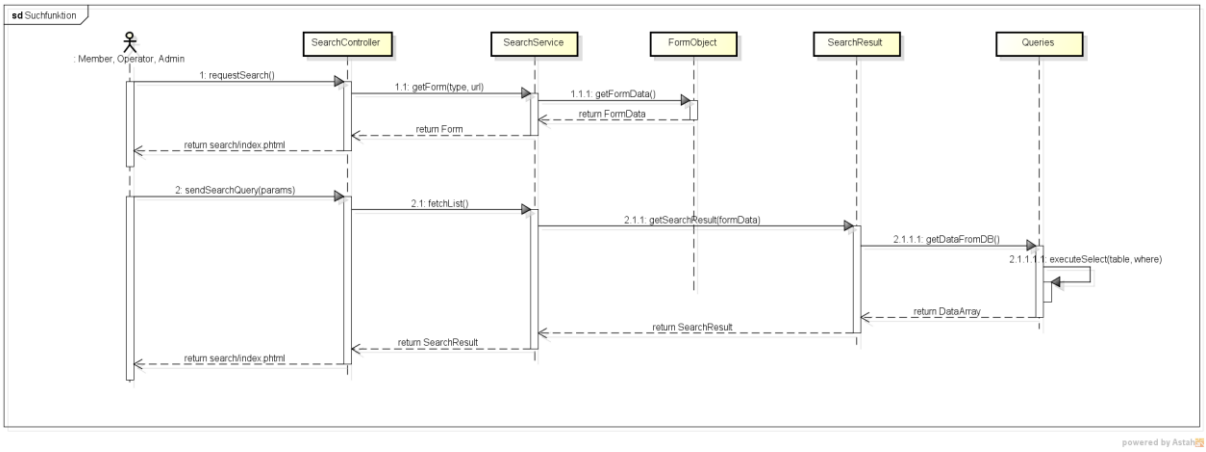


Abbildung 18 Sequenzdiagramm Searchresults

Beim Aufruf der Suchseite wird überprüft, ob über die URL Suchparameter übergeben werden. Wenn keine Suchparameter vorhanden sind, wird die Suchmaske, aus Performance Gründen, ohne Suchresultate geladen. Dadurch wird die Seite schneller geladen und der Nutzer kann nach seinen Wünschen eine spezifischere Suche starten. Anschliessend wird durch die Suchmaske die URL mit den angegebenen Parametern erstellt und die Resultate über einen POST-Request aus der Datenbank abgerufen. Die Idee dahinter ist, dass der URL Link zwischen den Nutzern geteilt wird und die gleichen Abfragen jederzeit wiederholt werden können. Durch die URL wird dann allerdings ein GET-Request abgeschickt.

10. ICS Detailansicht

In der Detailansicht werden alle Informationen und die History über ein ICS dargestellt. Wie in der Suchfunktion wird in der Detailansicht eine ähnliche Abfrage getätigt, um die Daten zu beschaffen.

Dabei wird das ICS eindeutig durch seine IP und Port Kennung abgefragt. Auch in der Detailview wird die ID als URL Parameter übermittelt.

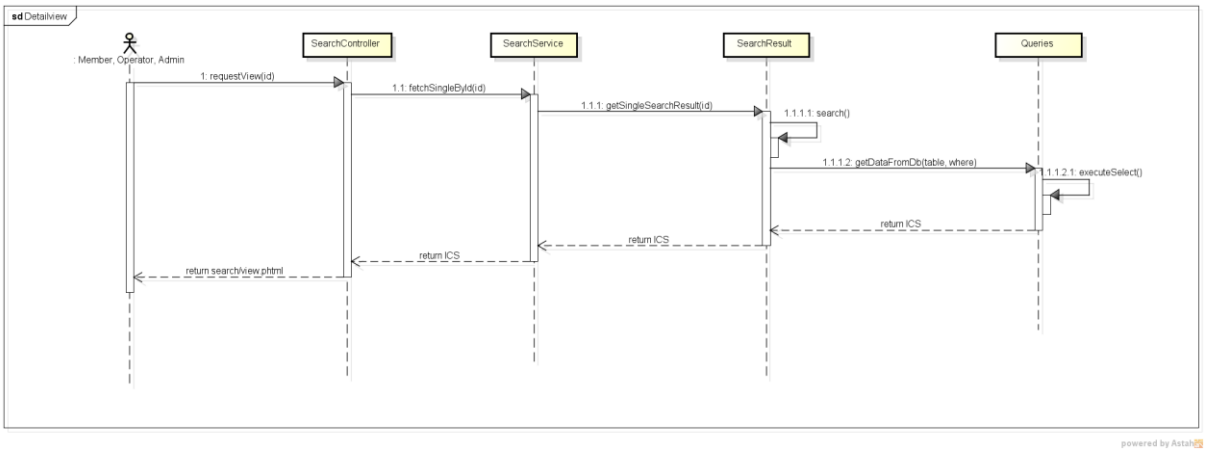


Abbildung 19 Sequenzdiagramm ICS Detail View

Ein wichtiger Bereich der Detailansicht ist das Bearbeiten von bestehenden ICS Einträgen, unabhängig von ihrer Herkunft. Viele der ICS Einträgen kommen von Suchmaschinen wie Shodan. Damit aber Benutzeränderungen nachvollziehbar bleiben und die ICS Rohdaten von Shodan beibehalten werden, können diese nicht einfach überschrieben werden. Dadurch würden diverse Probleme auftauchen, wie bspw. mit Aktualisierungen von Shodan nach Benutzeränderungen umgegangen wird. Aus diesem Grund müssen die Rohdaten aus einer Engine wie Shodan mit den Benutzeränderungen getrennt werden. Dies ermöglicht uns auch eine History von vergangenen Änderungen zu führen. Aus diesen Gründen, entschieden wir uns, unsere ICS zu archivieren und zwischen Rohdaten und Benutzeränderungen zu unterscheiden. Die folgende Grafik visualisiert einen Datenbankeintrag der zweimal aktualisiert wurde. Bei der Detailansicht eines ICS werden auf den bestehenden Haupteintrag (rot) die Benutzeränderungen(blau) mit höherer Priorität gesetzt. Mehr über das Datenbankdesign kann im Kapitel 13. Datenspeicherung nachgelesen werden.

icsId	keyword	title	insertedInDb	updatedInEngine	icstypeId	icsgroupId	operatingsystemId	organisationId	categoryId	locationId	ispId
59	87.245.105.254:80	Start	2013-12-04 21:27:41	2013-11-24 23:59:59	NULL	NULL	NULL	27	NULL	35	26
488	87.245.105.254:80	NULL	2013-12-05 00:47:47	NULL	1	NULL	NULL	NULL	4	NULL	NULL
499	87.245.105.254:80	NULL	2013-12-05 12:50:45	NULL	1	NULL	NULL	NULL	4	NULL	NULL

Abbildung 20 Tabelle ICS

threatseverityId	ipv4Id	ipv6Id	port	productId	contentId	userchangeId	contactId	archive
NULL	52	NULL	80	NULL	58	NULL	23	0
3	52	NULL	NULL	NULL	NULL	1	65	1
3	52	NULL	NULL	1	NULL	2	65	0

Abbildung 21 Tabelle Protocol

Diese Art und Weise liefert uns folgende Vorteile:

- Der Haupteintrag, der meistens von einem Drittanbieter wie Shodan oder Google stammt, kann laufend aktualisiert werden. Dadurch hat man immer die neuesten Daten der Engine.
- Benutzeraktualisierungen können jederzeit rückgängig gemacht werden.
- Es kann eine detaillierte History geführt werden.
- Die Daten werden nicht gelöscht.

Damit die Ansicht nach mehreren Änderungen nicht unendlich viele Einträge laden muss, wird der neueste Eintrag immer mit der letzten Aktualisierung zusammengefasst. Dies hat den Vorteil, dass immer nur zwei Einträge aus der Datenbank abgefragt und zusammengeführt werden müssen.
Das folgende Sequenzdiagramm beschreibt den genauen Ablauf, wie die ICS in der Klasse *SearchResult* und *Queries* zusammengeführt werden.

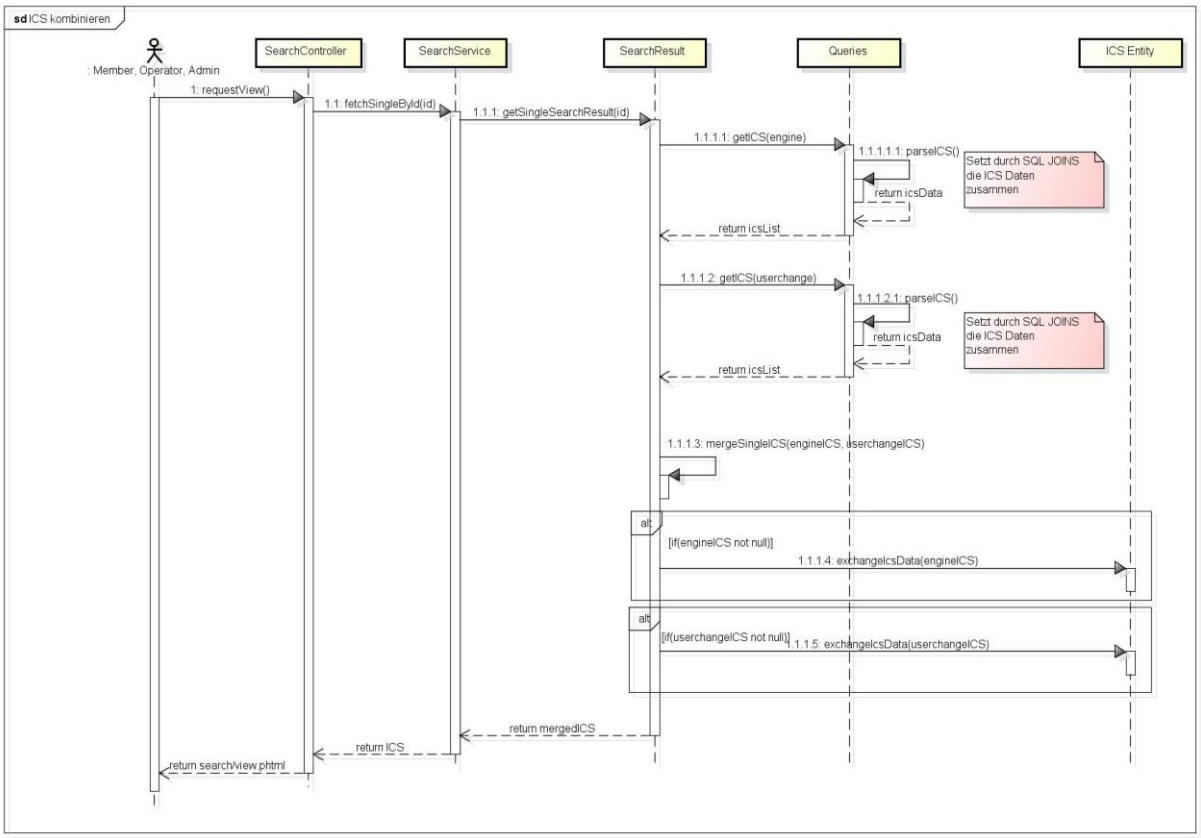


Abbildung 22 Sequenzdiagramm Zusammenführen eines ICS

Der Benutzer möchte ein spezifisches ICS ansehen und lädt die View. Die View enthält in der URL eine ID des ICS, welche der Search Controller holen kann. Anhand dieser Id kann er die Methode *fetchSingleById(id)* aufrufen. Diese Methode übergibt die Id an den SearchResult, das dann innerhalb der Methode *getSingleSearchResult(id)* die Änderungen zwischen Engine und UserChanges zusammenführt.

11. Trouble Ticket

Eines der Anforderungen ist es ein Trouble Ticket System anzubieten, um die ICS Einträge abzuarbeiten. Das Trouble Ticket ist jeweils an einen ICS gebunden. Die Benutzer können im Trouble Ticket ihre Fortschritte protokollieren. Eine History zeigt die letzten Aktivitäten am Ticket auf.

Das Trouble Ticket System wurde folgendermassen gegliedert.

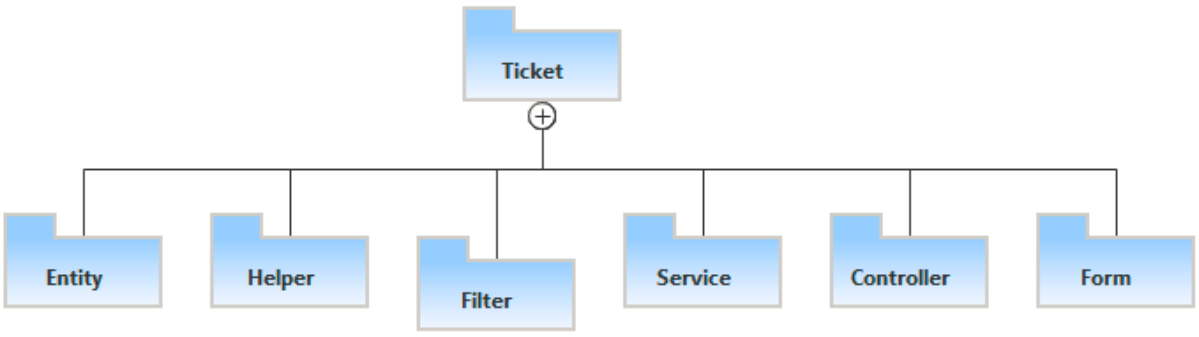


Abbildung 23 Modulstruktur Ticket

Das folgende Sequenzdiagramm visualisiert die Anzeige des Trouble Tickets.

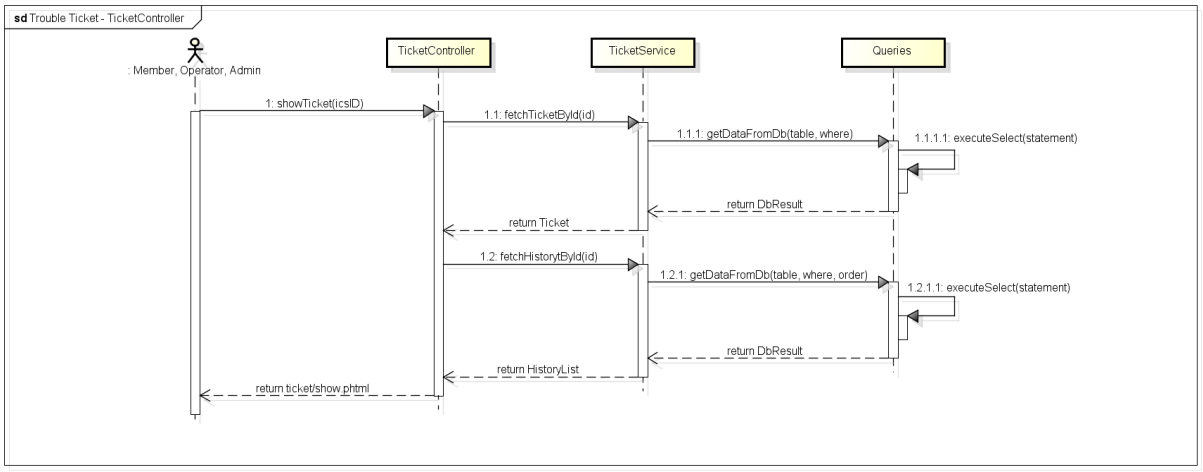


Abbildung 24 Sequenzdiagramm Ticketverlauf

Die Verwaltung eines Tickets wird über den *AdminController* gemacht. Dabei unterscheidet der AdminController zwischen Erstellen, Bearbeiten oder Löschen. Das folgende Sequenzdiagramm beschreibt das Hinzufügen eines Tickets.

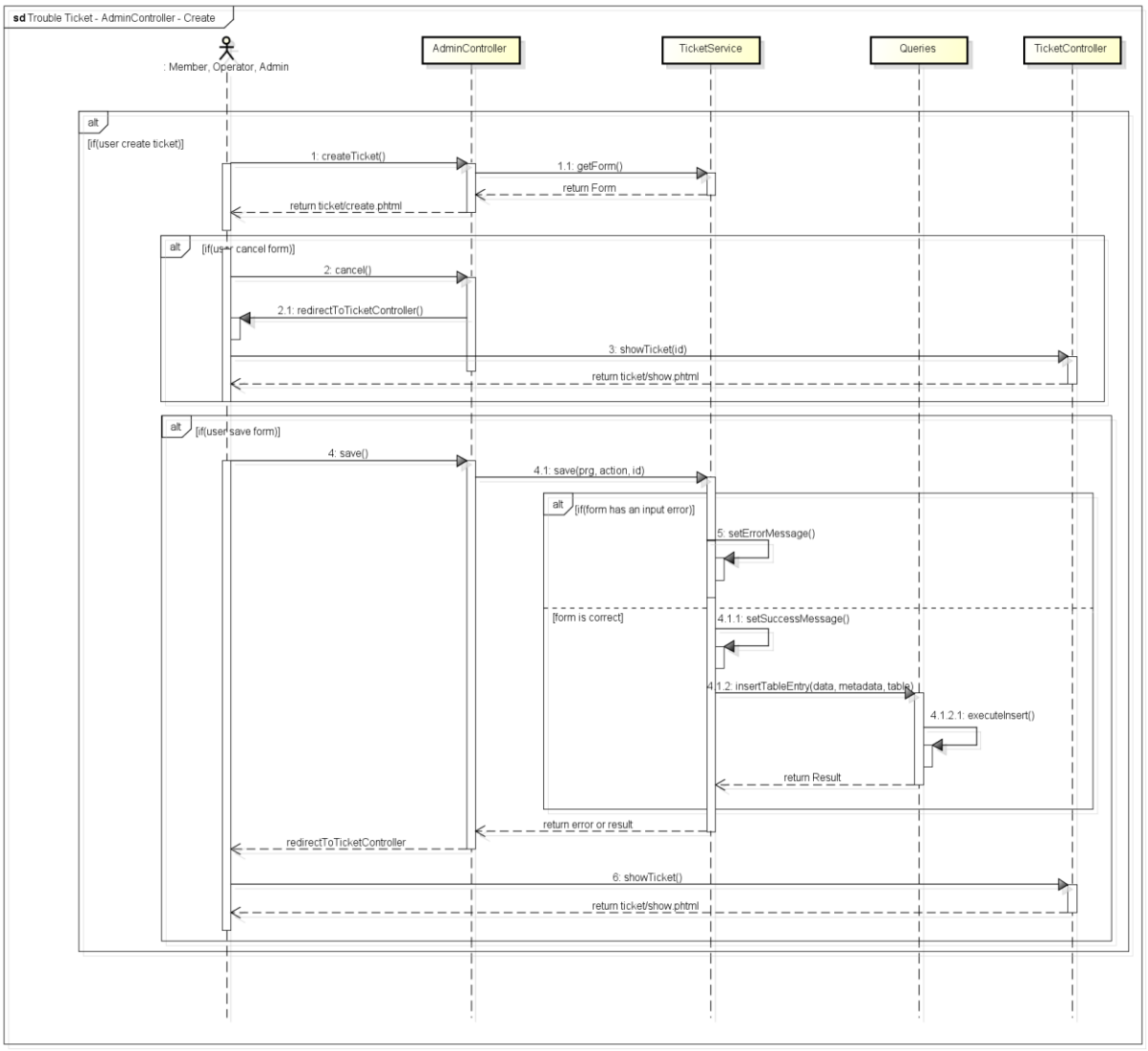


Abbildung 25 Sequenzdiagramm AdminController CRUD

12. Rest

Das Rest Modul bildet die Schnittstelle zwischen den clientseitigen Funktionen, wie GoogleMaps oder jqPlot und der serverseitigen Applikation Logik.

Das Rest Modul ist folgendermassen gegliedert.

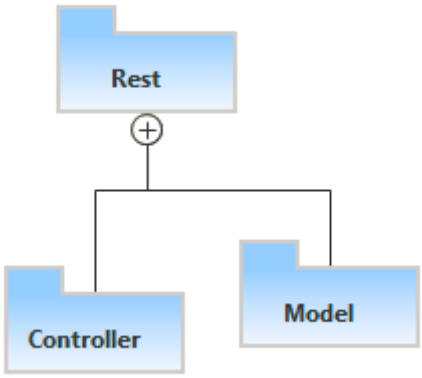


Abbildung 26 Modulstruktur Rest

Für Google Map und jqPlot wurde JavaScript und JQuery verwendet. Mit Ajax werden die Daten beim Rest Module angefragt. Das Rest Module überprüft die Berechtigungen des Besuchers und schickt diesem Benutzerspezifische Resultate.

Die folgende Grafik zeigt einen AJAX Aufruf. Dieser übermittelt die Koordinaten des Kartenausschnitts. Damit werden nicht alle Daten auf einen Schlag geladen.

```
//get window bounderies
var bounds = map.getBounds();
var ne = bounds.getNorthEast(); // LatLng of the north-east corner
var sw = bounds.getSouthWest(); // LatLng of the south-west corder


$.ajax({
    url: '/rest/' + ne.lat() + "/" + ne.lng() + "/" + sw.lat() + "/" + sw.lng(),
    dataType: 'json',
    async: true,
    success: function(result){

        parseData(result);

    }
});
```

Abbildung 27 AJAX GOOGLE MAP

13. Datenspeicherung

Für die Speicherung und Verwaltung der ICS Daten benötigt die icsmap.ch eine Datenbank. Dabei wird die Open Source Datenbank MySql in der Version 5.5 verwendet.

13.1 Datenbankmodell

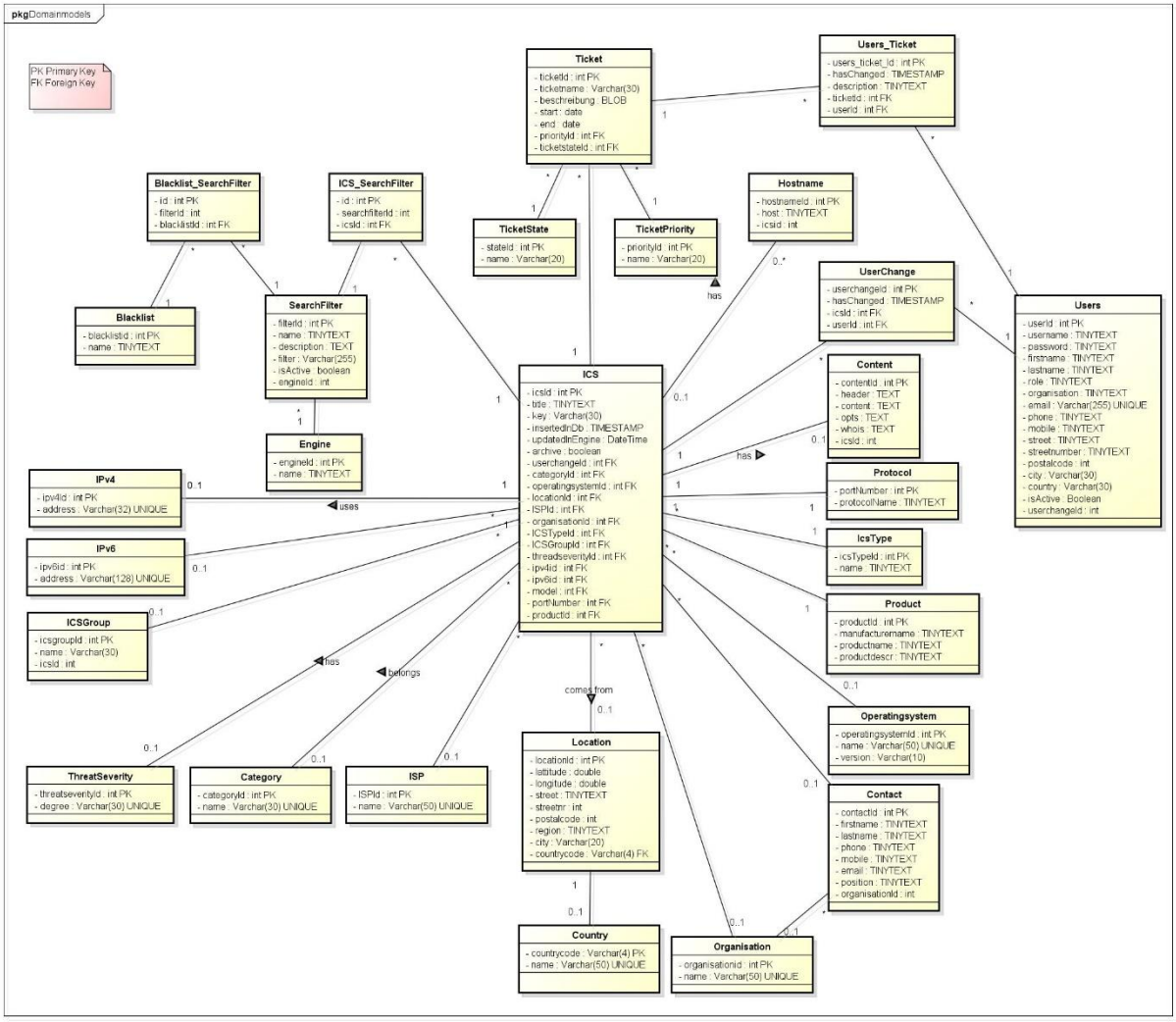


Abbildung 28 Datenbankmodell als Ganzes

Um das Datenbankmodell etwas einfacher zu veranschaulichen, lässt es sich in vier verschiedene Bereiche unterteilen:

- ICS Daten
- SearchFilter
- Ticket
- Usermanagement

13.1.1 Bereich ICS

In der folgenden Grafik wird das Datenbankmodell der ICS Daten aufgezeigt. Die ICS Daten machen im Datenbankmodell etwa 60% der Daten aus.

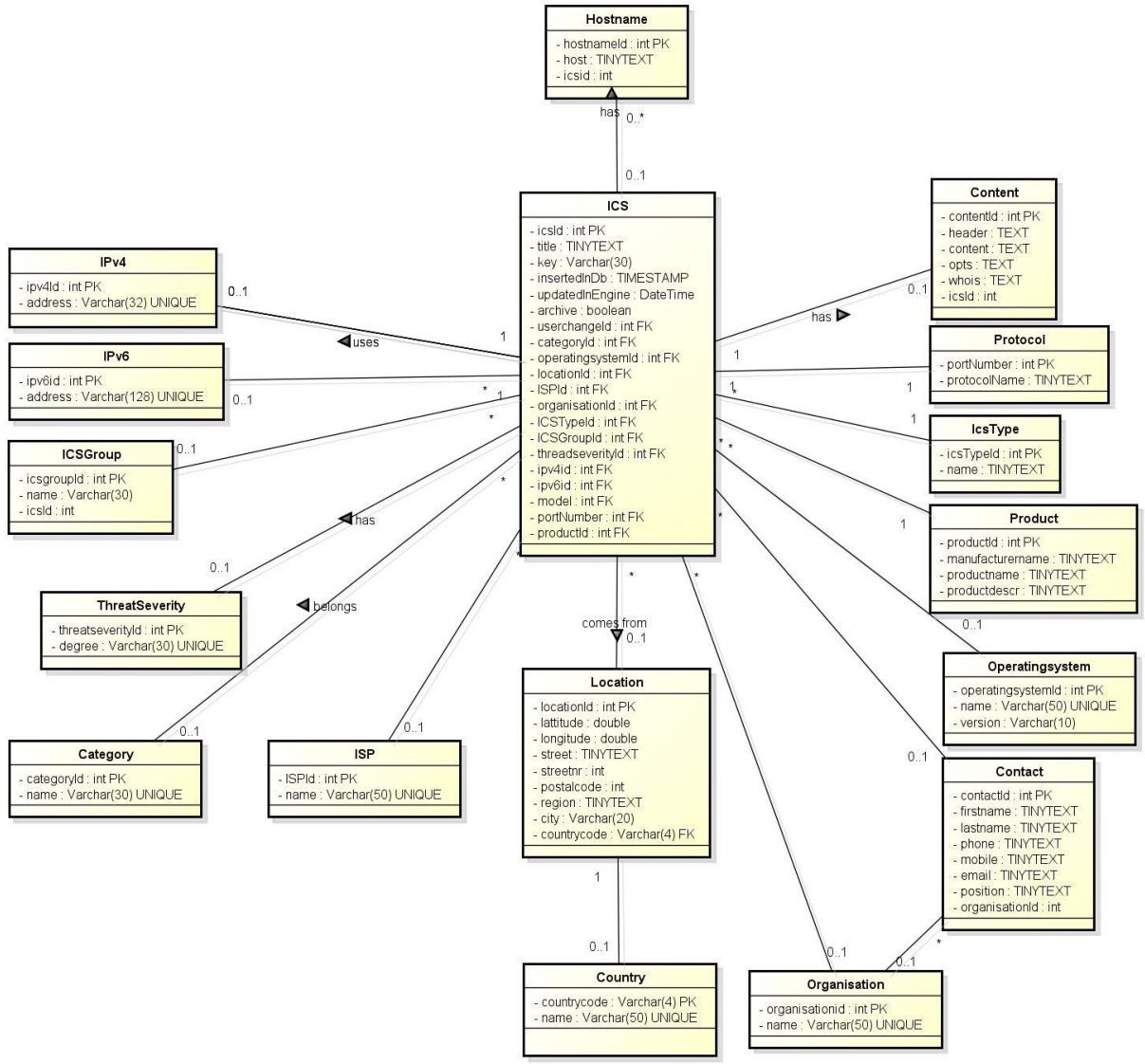


Abbildung 29 Bereich ICS

Das Update-Script sucht täglich nach neuen ICS und fügt diese in die DB ein. Falls ein ICS noch nicht in der DB existiert, wird ein neuer Eintrag erstellt. Ansonsten wird anhand des Attributs *updatedInEngine* mit dem Attribut von Shodan überprüft, ob Shodan ein jüngeres Datum besitzt. Falls ja, wird der vorhandene ICS als neuer Eintrag erfasst. Leider sind die Daten eines ICS von Shodan oder Google sehr unvollständig oder falsch und müssen daher mit zusätzlichen Informationen angereichert werden. Häufige Änderungen der Daten des Users, könnte der *Titel*, *genauer Standort*, *Organisation* oder *Kontakt* sein. Alle anderen Daten, müssen zusätzlich auf der Webapplikation erfasst werden.

13.1.2 Bereich Suchfilter

Im Bereich Suchfilter werden alle Suchfilter und deren Informationen für die Suche von ICS gesammelt. Da die ICS über verschiedene Suchmaschinen (Engine) auffindbar sind, müssen die verwendeten Filter unbedingt erfasst werden.

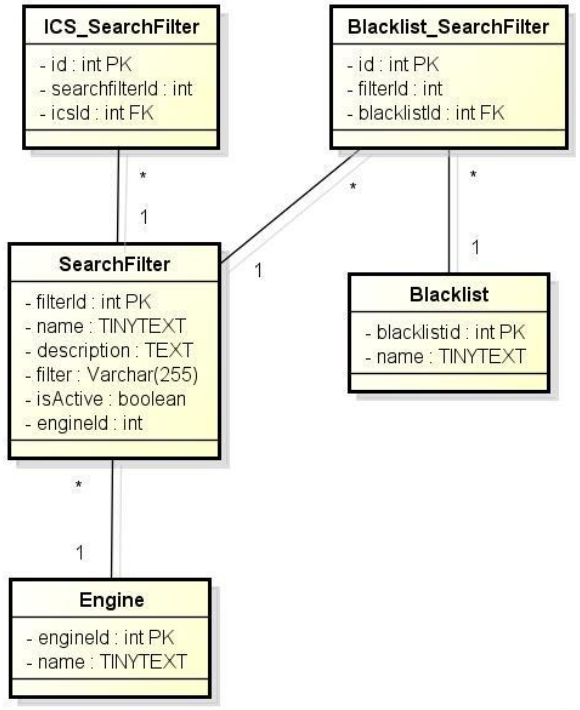


Abbildung 30 Bereich Suchfilter

Für die Suchfilter sind im Wesentlichen drei Tabellen wichtig. Dies sind *Engine*, *SearchFilter* und *ICS_SearchFilter*. In der Engine werden die verschiedenen Suchmaschinen abgelegt wie z.B. Shodan, Google, Nmap, usw. Der SearchFilter enthält die wesentlichen Daten wie den Namen, die Beschreibung, den Filterwert und woher der Filter stammt. Der ICS_Searchfilter ist nur eine Assoziativtabelle zwischen der Tabelle ICS und SearchFilter. Wenn die Zeit ausreicht, kann zusätzlich zu den Suchfiltern, eine Blacklist verwaltet werden, um ungewünschte Filter auszusortieren.

13.1.3 Bereich Ticket

Um nach verwundbaren ICS vorgehen zu können und die Betreiber auf deren ICS aufmerksam zu machen, soll ein Ticket System für die Dokumentation eingerichtet werden.

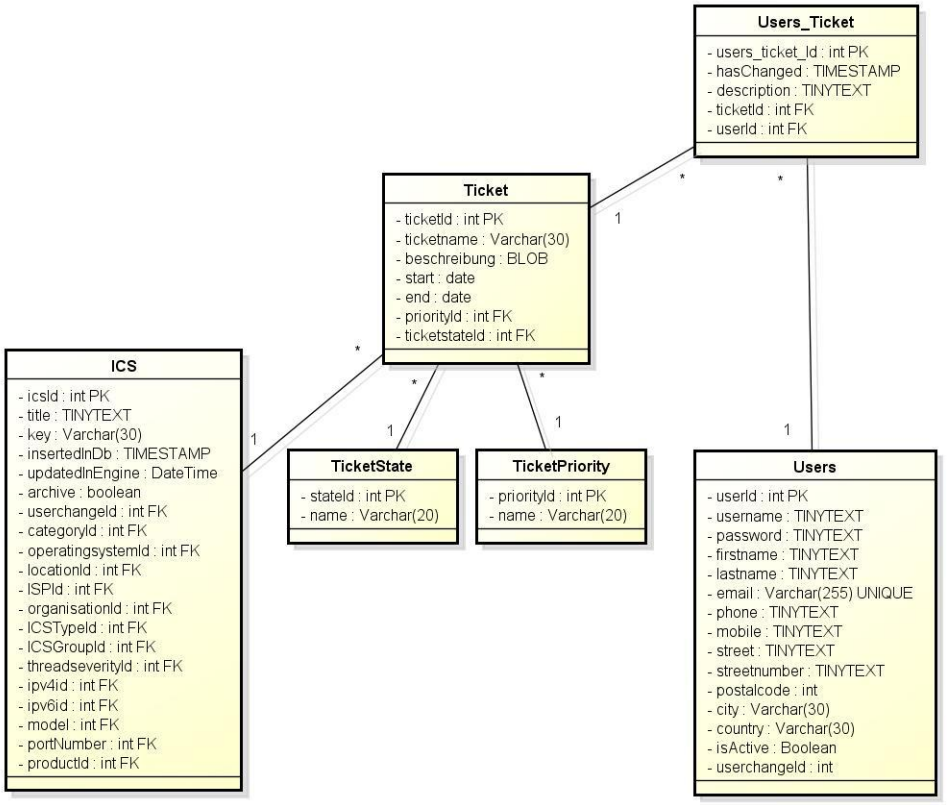


Abbildung 31 Bereich Ticket

Für ein Ticket System muss ein neues Ticket mit seinem Titel, seiner Beschreibung, Priorität, seinem Start- und Enddatum, sowie dem Ticket Status und der Priorität ausgefüllt werden. Diese Informationen werden in der Tabelle *Ticket* gespeichert. Zusätzlich müssen die Änderungen eines Tickets und seiner History aufgezeigt werden. Diese Aufgabe übernimmt die Tabelle *Users_Ticket*, welche die Referenz des Benutzers sowie des Tickets enthält.

13.1.4 Bereich User

Der User Bereich ist das Benutzermanagement. Hier werden alle Daten der registrierten Benutzer, die Zugriff auf die Webapplikation haben dürfen, gespeichert.

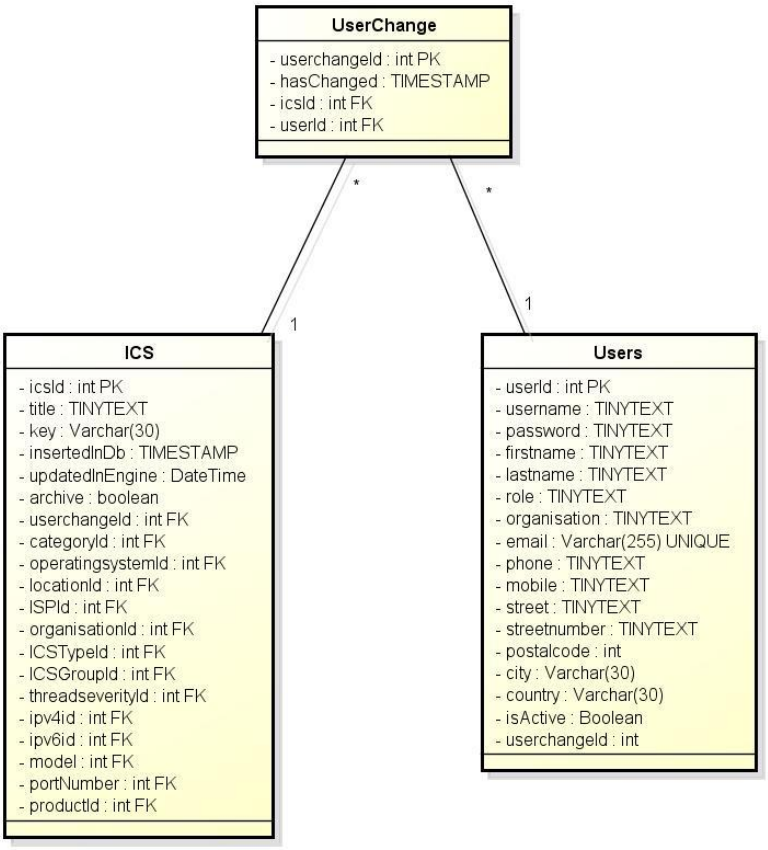


Abbildung 32 Bereich User

Mit dem Attribut *role* wird die Rolle eines Benutzers festgelegt. Die jeweiligen Rollen eines Benutzers, wurden im Kapitel 8.2 Benutzerrollen / -rechte genauer beschrieben. Ein weiteres wichtiges Attribut ist *isActive*. Normalerweise kann sich jeder Benutzer auf einer Webseite registrieren. Meistens wird bei der Bestätigung des Accounts, das Opt-In Verfahren genutzt; Entweder wird eine Mail mit einem Bestätigungs-Link an das Neumitglied verschickt und muss dann bestätigen oder es wird ein Code an den Benutzer versendet, welcher dann auf der Webseite bestätigen muss.

Wir gehen nicht nach dieser Vorgehensweise, sondern jeder neuregistrierter Benutzer muss warten, bis der Admin ihn freigeschaltet hat. Deshalb wird nach jeder Registration ein neuer User in der DB hinzugefügt, jedoch mit dem Attribut *isActive* und dem Wert *false*.

13.2 Installationsskript

Für die einfache und schnelle Erstellung der Daten wurde ein Installationsskript erstellt, welches ein Datenbank-User mit Permissions erstellt, sowie die Datenbank, ihre Tabellen und einige Vorgabedaten.

Das Installationsskript wird stets ausgebaut, so dass die ICS ThreatMap Applikation mit einem Installationsskript installiert werden kann.

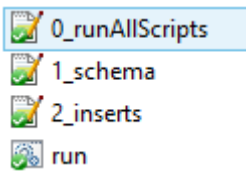


Abbildung 33 Installationsskript

13.3 Historie und Datenbankänderung

Die Daten müssen in der Datenbank so angelegt werden, damit Benutzeränderungen nicht durch die täglichen Updates überschrieben werden und umgekehrt. Das heisst, die Webapplikation muss die

Änderungen in einem gewissen Schema durchlaufen, um ungewollte Überschreibungen zu vermeiden. Zusätzlich muss eine History der Daten möglich sein, damit ein Benutzer beispielsweise nach ICS anhand eines Datums gesucht werden kann.

Um die Vorgehensweise genauer zu erläutern, werden im Folgenden die Tabelle ICS und ihre wichtigen Attribute kurz beschrieben.

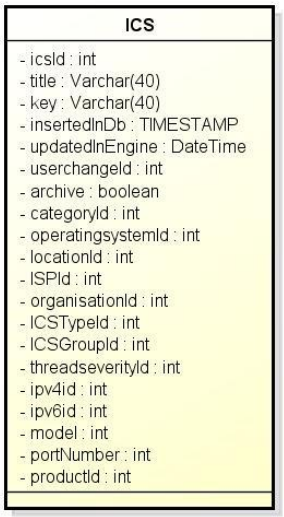


Abbildung 34 ICS Tabelle

- Key
  Das Attribut *Key* ist das Keyword oder auch der Fingerprint eines ICS. Mit diesem Attribut kann eindeutig nach einem ICS unterschieden werden. Für die Unterscheidung wird die Kombination „IP:Port“ verwendet.

- insertedInDb
  Das *InsertedInDb*-Feld hat den Typ *Timestamp* und erzeugt bei jedem Insert einen *DateTime*-Wert.

- updatedInEngine
  Beim Attribut *updatedInEngine* wird der Type *DatumTime* gesetzt. Dieses Attribut ist insbesondere wichtig bei der Abfrage von Shodan Entries, da Shodan ebenfalls ein *updated* Feld für jedes Gerät zurückgibt.

- Archive
  Mit dem Attribut *archive* werden die ICS nach deren aktuellen Status abgefragt. Wenn Archive *true* ist, dann handelt es sich um ein veraltetes ICS, andernfalls um ein aktuelles ICS.

- Engine
  Der Begriff *Engine* ist nicht wie die anderen ein Attribut, sondern eine weitere Tabelle, die mit der ICS Tabelle über eine Zwischentabelle verknüpft ist. Mit dieser Tabelle wird identifiziert, wo das ICS gefunden wurde und enthält einen Wert *User*, mit dem geprüft wird, ob das ICS von einem Benutzer geändert wurde. Dies ist notwendig, damit zwischen Benutzeränderungen und Änderungen vom Update-Script unterschieden werden kann.

13.3.1 Ablauf History

Für jedes Update eines Benutzers oder vom Update-Script wird ein neuer Eintrag in der Tabelle *ICS* erstellt. Dabei wird automatisch im Attribut *insertedInDb* ein Timestamp gesetzt. Falls es sich um ein Update vom Update-Script handelt, dann wird im *updatedInEngine* der neue Wert, welcher die Searchengine zurückliefert, übernommen. Ansonsten bleiben die Werte von *Key* und *updatedInEngine* dieselben. Da nun das Problem mit mehrfachen Einträgen eines ICS auftaucht, werden die veralteten ICS mit Hilfe des Attributs *archive* auf *true* gesetzt. Somit kann eine Abfrage erstellt werden, bei welcher alle ICS mit dem Keyword A sucht, aber nur die Aktuellsten mit dem Wert *true* anzeigt.

13.3.2 Ablauf Datenbankänderung

Die Updates zwischen Benutzer und Update-Scripts werden wie folgt durchgeführt.

		ICS							UserChange			User	
Ablauf	**Operation**	**ICSId**	**keyword**	**InsertedInDb**	**UpdatedInEngine**	**Title**	**Archive**	**Engine**	**UcId**	**HasChanged**		**userId**	**name**
1	UpdateScript	1	48.42.94.14:80	03.11.13	01.11.13	A	false	Shodan					
2	UpdateScript	1	48.42.94.14:80	03.11.13	01.11.13	A	true	Shodan					
		2	48.42.94.14:80	04.11.13	**03.11.13**	B	false	Shodan					
3	User A performs update	1	48.42.94.14:80	03.11.13	01.11.13	A	true	Shodan					
		2	48.42.94.14:80	04.11.13	03.11.13	B	**false**	Shodan					
		3	48.42.94.14:80	05.11.13	03.11.13	C	**false**	**User**	1	05.11.13		1	A
4	User B performs update	1	48.42.94.14:80	03.11.13	01.11.13	A	true	Shodan					
		2	48.42.94.14:80	04.11.13	03.11.13	B	false	Shodan					
		3	48.42.94.14:80	05.11.13	03.11.13	C	**true**	User	1	05.11.13		1	A
		4	48.42.94.14:80	06.11.13	03.11.13	D	**false**	User	2	06.11.13		2	B
5	User A performs update	1	48.42.94.14:80	03.11.13	01.11.13	A	true	Shodan					
		2	48.42.94.14:80	04.11.13	03.11.13	B	false	Shodan					
		3	48.42.94.14:80	05.11.13	03.11.13	C	true	User	1	05.11.13		1	A
		4	48.42.94.14:80	06.11.13	03.11.13	D	**true**	User	2	06.11.13		2	B
		5	48.42.94.14:80	08.11.13	03.11.13	E	**false**	User	2	08.11.13		1	A
6	UpdateScript performs update	1	48.42.94.14:80	03.11.13	01.11.13	A	true	Shodan					
		2	48.42.94.14:80	04.11.13	03.11.13	B	**true**	Shodan					
		3	48.42.94.14:80	05.11.13	03.11.13	C	true	User	1	05.11.13		1	B
		4	48.42.94.14:80	06.11.13	03.11.13	D	true	User	2	06.11.13		2	A
		5	48.42.94.14:80	08.11.13	03.11.13	E	**false**	User	2	08.11.13		1	B
		6	48.42.94.14:80	11.11.13	**09.11.13**	F	**false**	Shodan					

Abbildung 35 Szenario Datenbankänderungen

Zu Beginn werden die Daten über das Update-Script in die Datenbank eingefügt (Ablauf 1). In der folgenden Abbildung wird ein ICS Eintrag mit dem keyword *48.42.96.14:80* hinzugefügt.

Am nächsten Tag (Ablauf 2) wird das Update-Script wieder ausgeführt und stellt anhand des *updatedInEngine* Attributs fest, dass es ein neues Update für diesen Eintrag gibt. Also wird ein neuer Eintrag mit dem Archive-Wert *false* erstellt und der bestehende Wert neu auf *true* gesetzt.

Im Ablauf 3 wird am gleichen Tag eine Benutzeränderung durchgeführt. Das heisst, der Benutzer ändert den Titel des ICS und speichert ihn ab. In der Datenbank wird wieder ein neuer Eintrag erstellt und setzt Archive auf *false*. Zusätzlich wird im Engine auf „User“ gewechselt und in der Tabelle UserChange wird das Ereignis festgehalten. Im Fettgedruckten werden die wichtigen Änderungen angezeigt. Dabei dürfen pro ICS immer nur maximal zwei Einträge im Archive auf *false* gesetzt werden. Einmal der Haupteintrag aus einer Engine und andererseits die Benutzeränderung. Somit kann die Abfrage so ausgewertet werden, dass eine Benutzeränderung (Enginewert *User*) höher priorisiert wird als Änderungen vom Update-Script bzw. von der Engine.

Im Ablauf 4 wird wiederum eine Benutzeränderung durchgeführt. Dabei wird der neue Eintrag auf *false* gesetzt und der ältere Eintrag auf *true*. Die Shodan-Einträge bleiben unverändert.

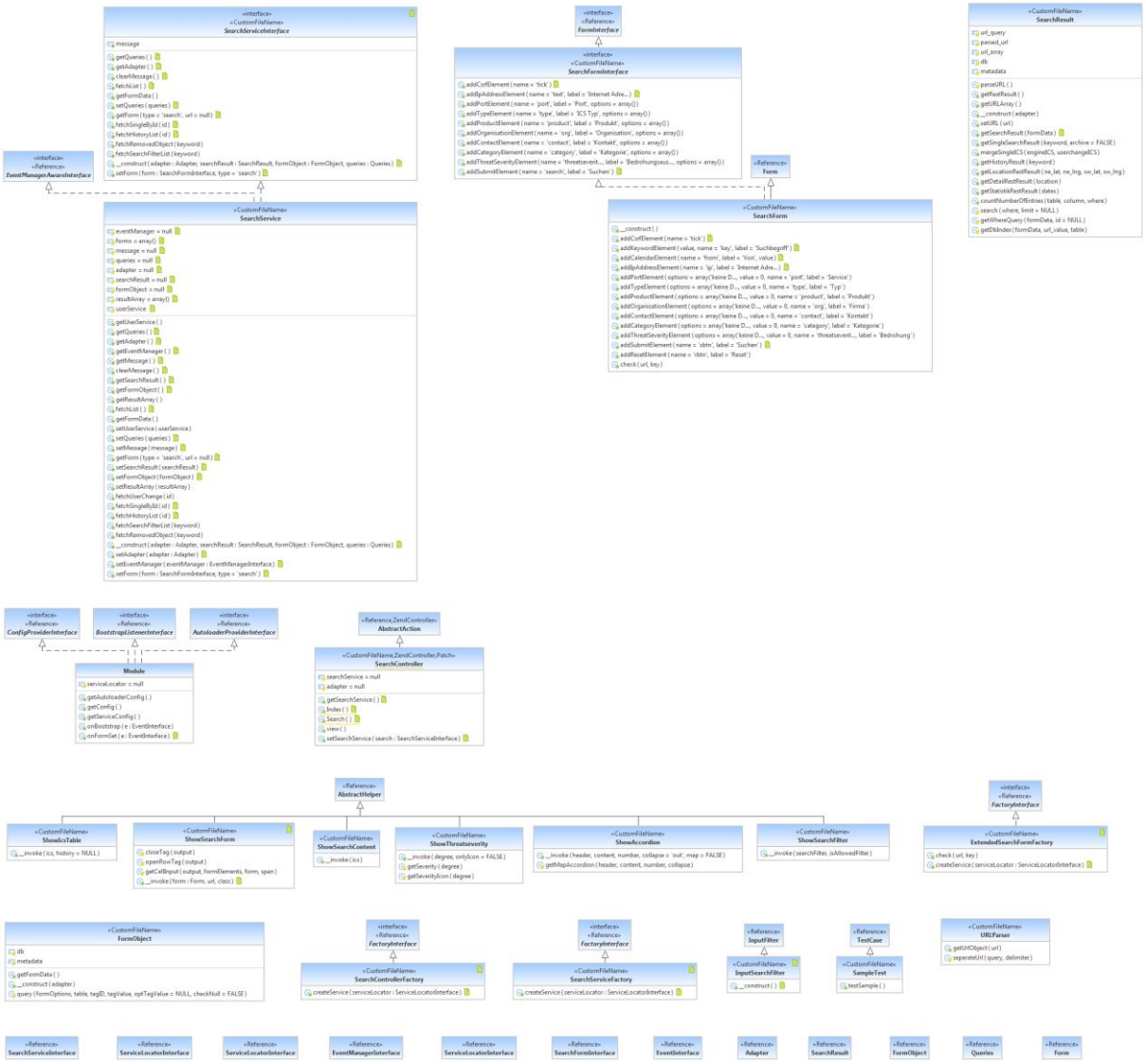
Das Prinzip im Ablauf 5 ändert sich hier nicht mehr.

Im Ablauf 6 findet das Update-Script ein neues Update. Der neue Eintrag enthält das aktualisierte *updatedInEngine* Datum und setzt es auf *false*. Zusätzlich muss der ältere Eintrag vom letzten Update-Script archiviert werden.

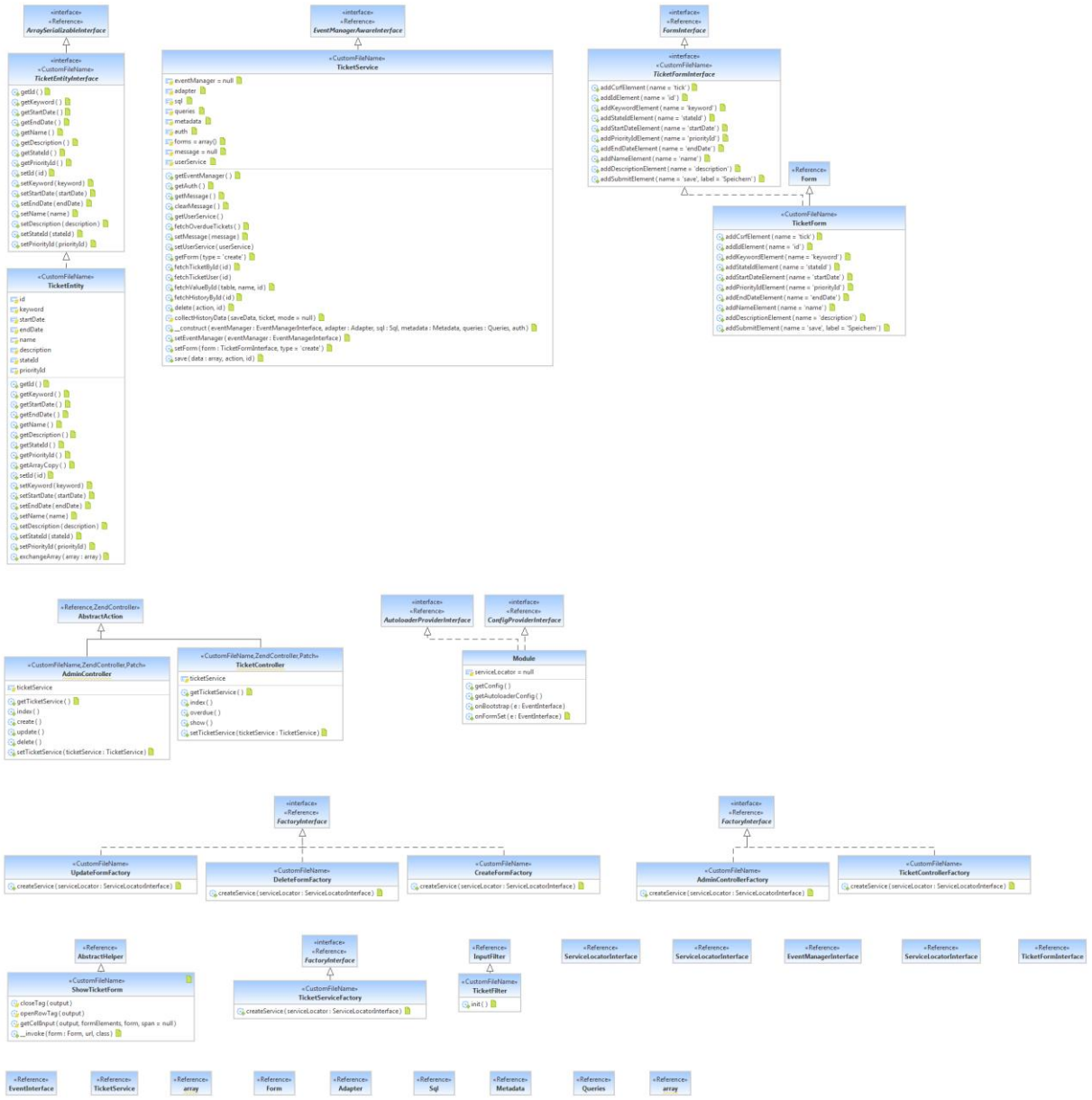
14. Anhang

14.1 Klassendiagramm Benutzerverwaltung

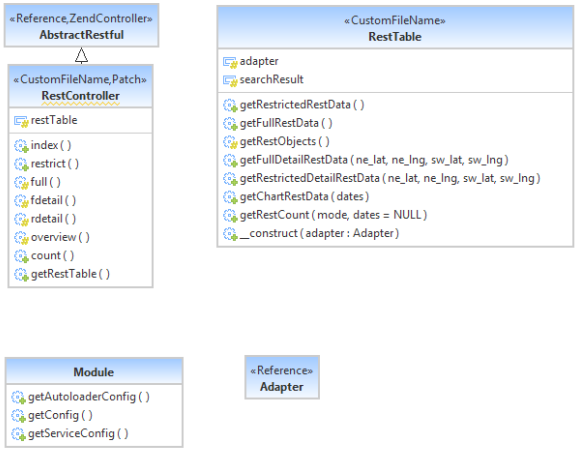
14.2 Klassendiagramm Suche & Anzeige



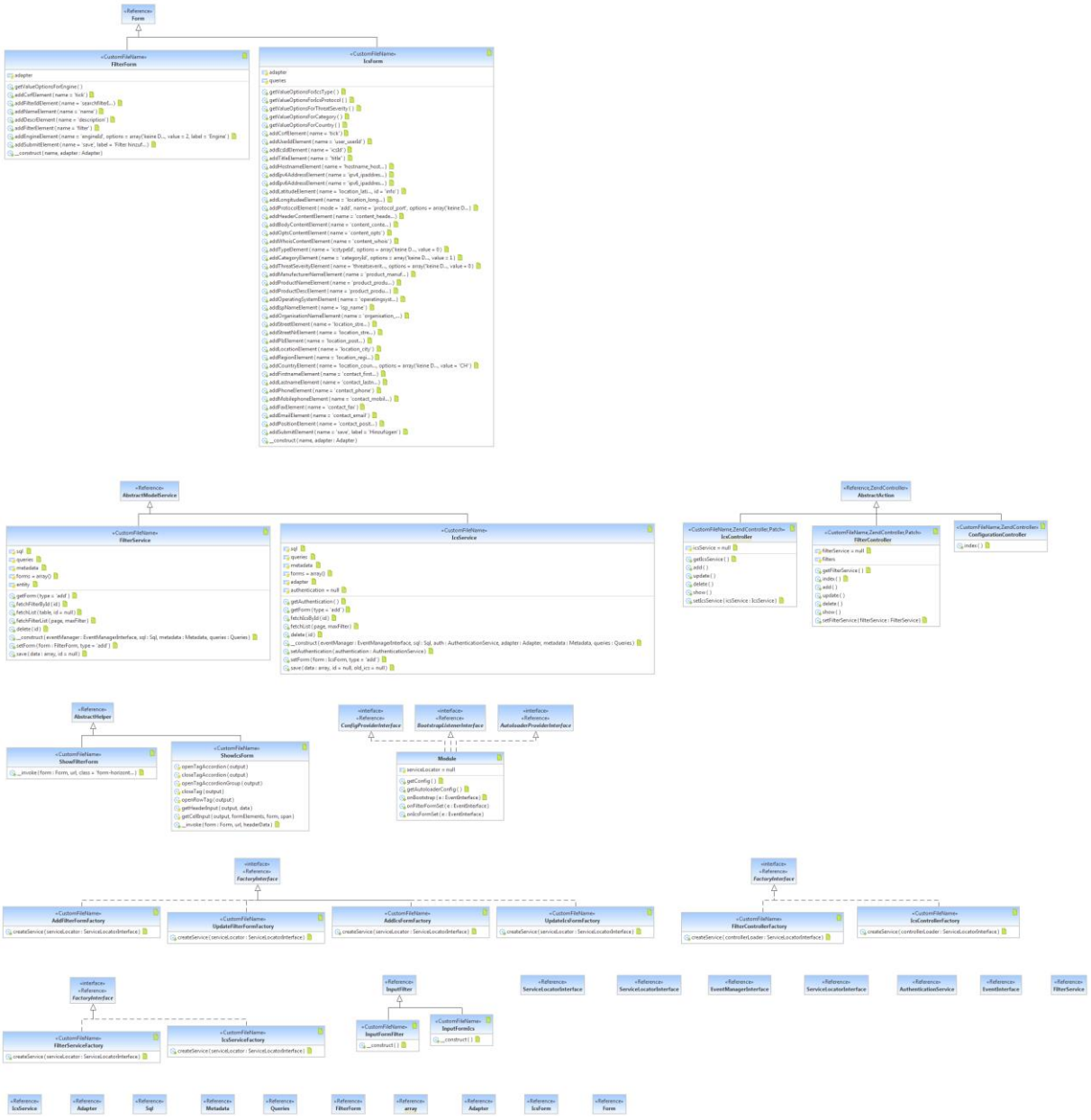
14.3 Klassendiagramm Ticket



14.4 Klassendiagramm Rest



14.5 Klassendiagramm Konfiguration



15. Abbildungsverzeichnis

ICS ThreadMap - v1.0

User Interface
(UI)

Benjamin Kehl
Dominique Sorg

Änderungsgeschichte

Datum	Version	Änderungen	Autor
10.12.13	0.1	Erstellung Dokument	Dominique Sorg

Inhalt

1. Einführung

1.1 Zweck

Dieses Dokument beschreibt die User Interface Wireframes für das ICS ThreatMap Porjekt.

1.2 Gültigkeitsbereich

Das Dokument dient als Grundlage für das Frontend und ist über die gesamte Projektdauer gültig.

1.3 Übersicht

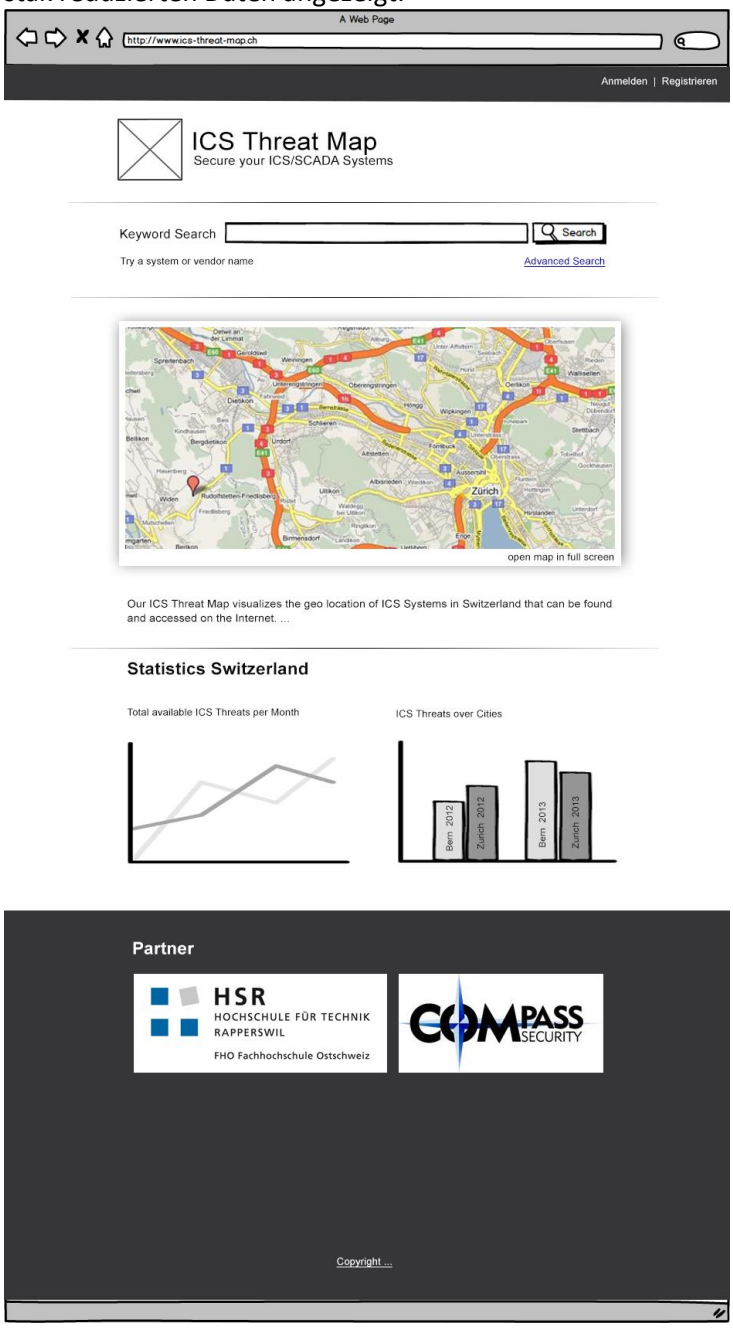
Dieses Dokument gibt eine Übersicht über die erstellten Wireframes von ICS ThreatMap und beschreibt diese.

2. Wireframes

Damit wir unseren Auftragsgeber besser verstehen, wurden Wireframes erstellt. Diese werden hier aufgezeigt und kurz erläutert.

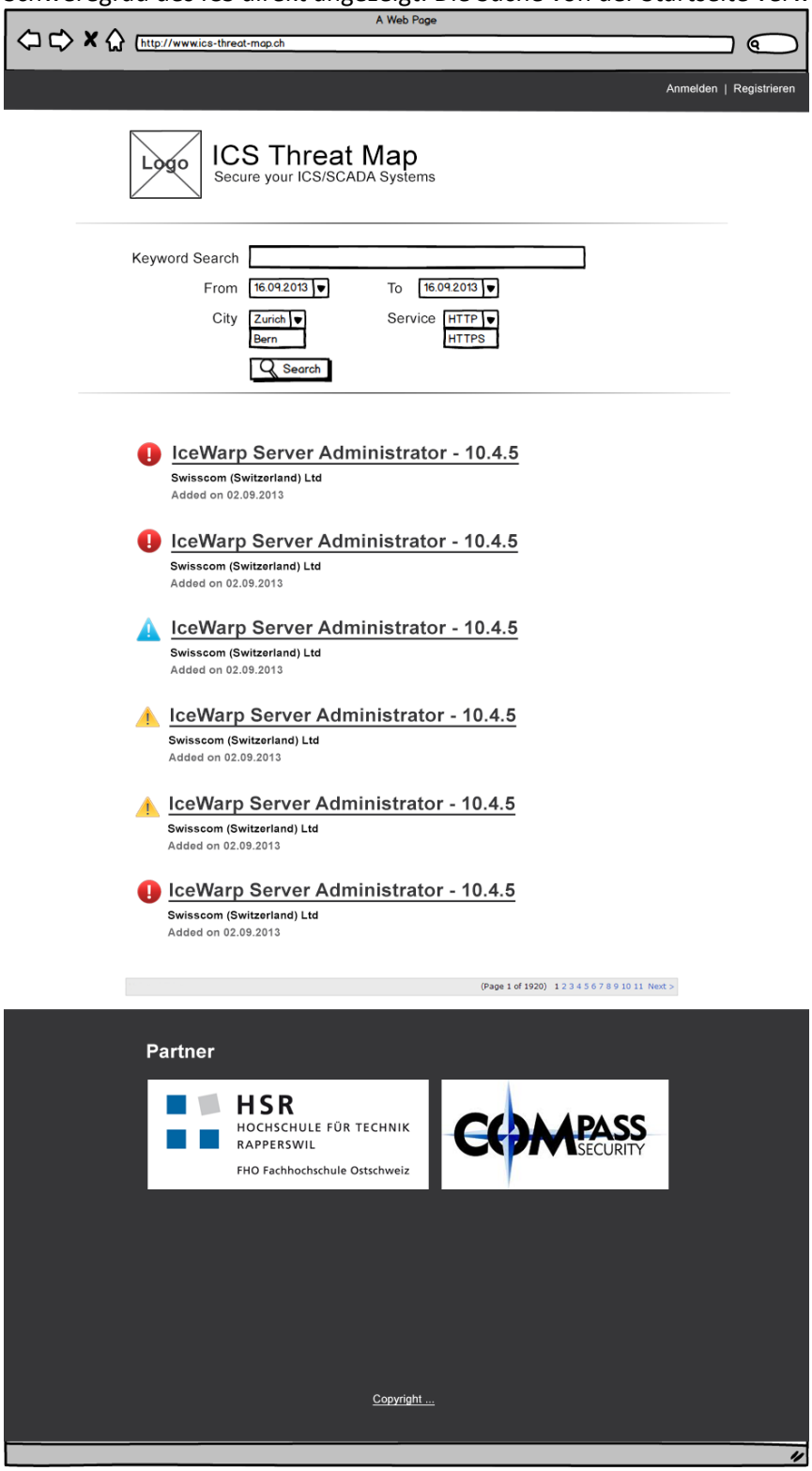
2.1 Startseite

Die Startseite ist der Ausgangspunkt der Web Applikation. Auf der Startseite soll die Karte integriert werden. Eine Statistik zeigt den Verlauf von gefunden ICS an. Eine einfache Suchfunktion kann direkt von der Startseite aus benutzt werden. Angemeldete Benutzer sollen in der Lage sein, alle Funktionen der Applikation zu verwenden. Nicht angemeldete Benutzer erhalten einen Einblick mit stak reduzierten Daten angezeigt.



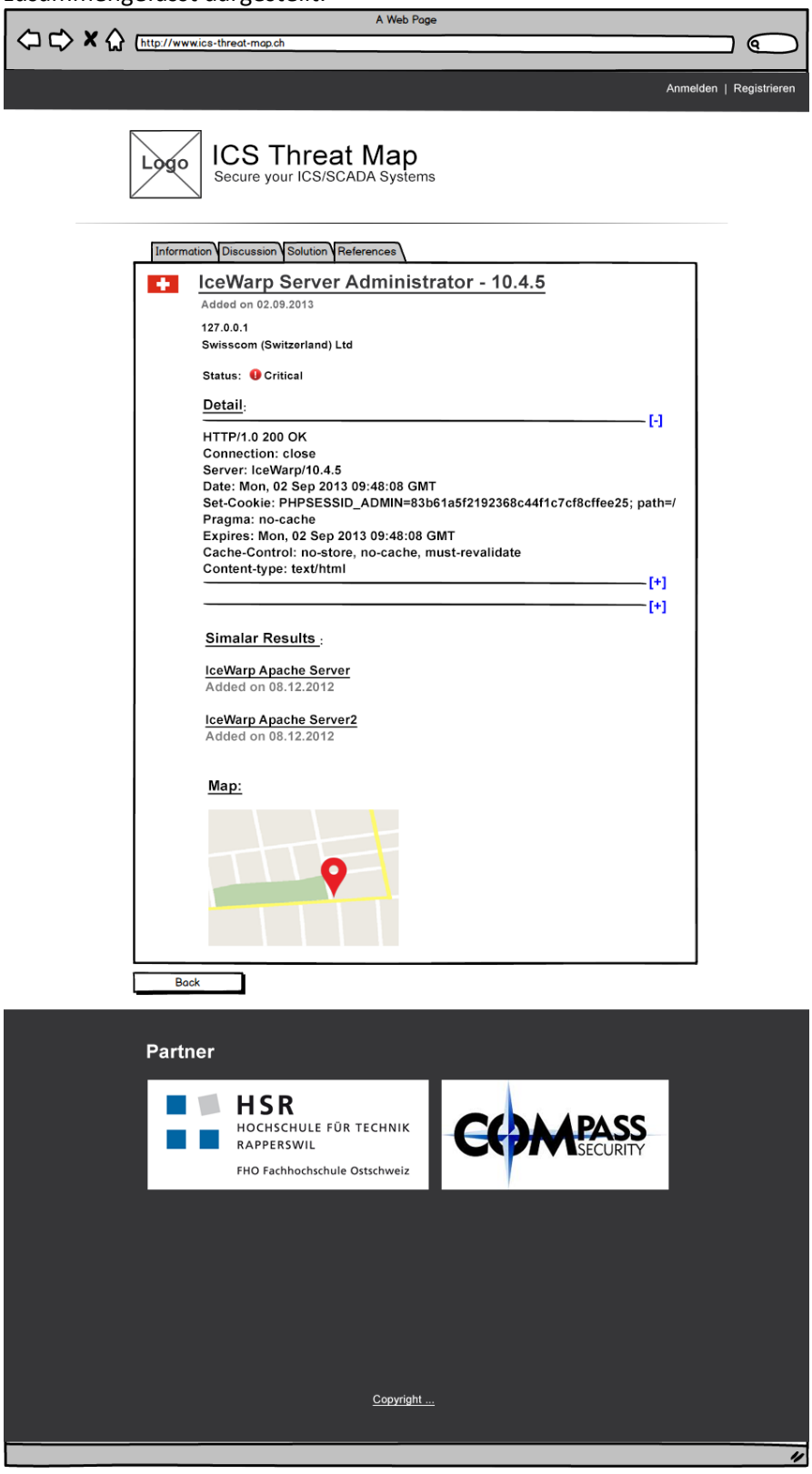
2.2 Suchfunktion

Die Suchfunktion zeigt eine Auflistung von gefundenen ICS Systeme. In der Suche wird der Schweregrad des ICS direkt angezeigt. Die Suche von der Startseite verweist auf die erweiterte Suche.



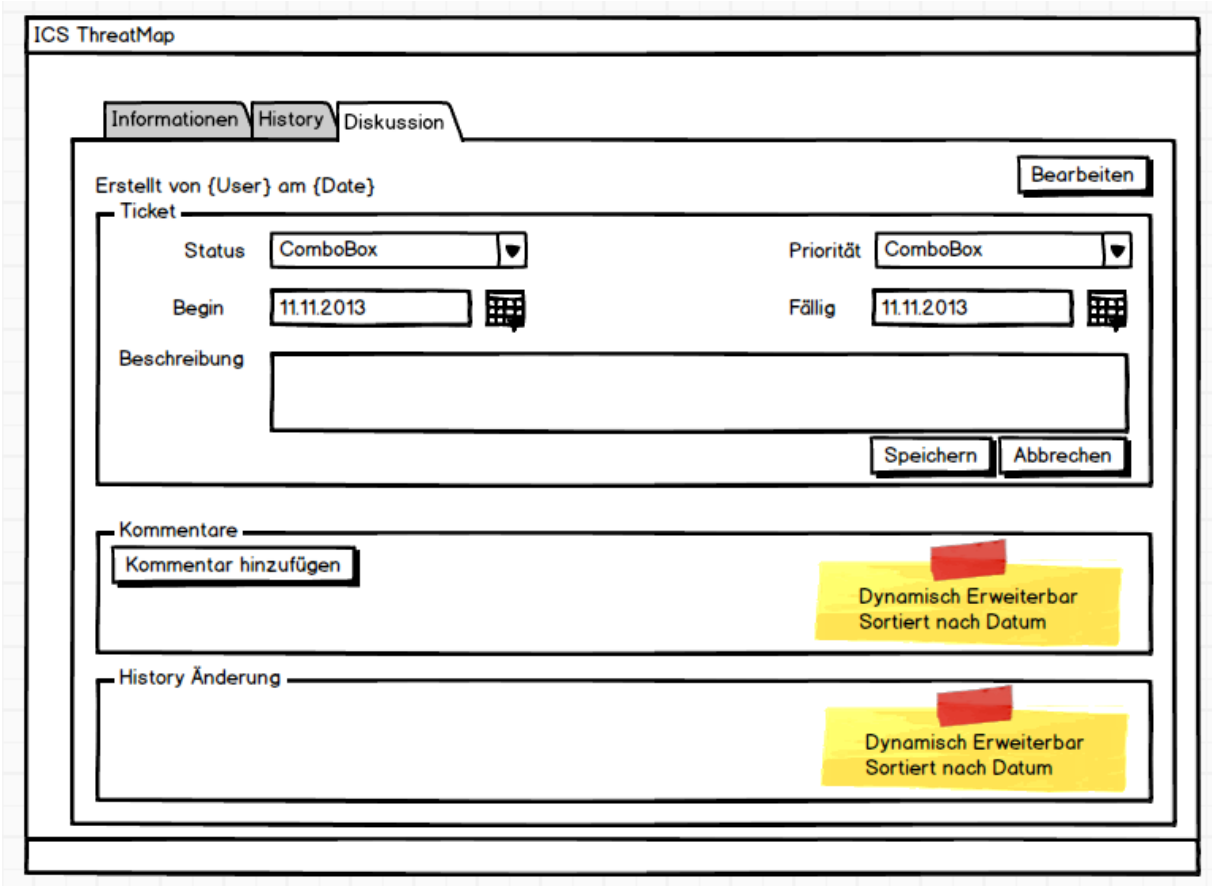
2.3 ICS Detailansicht

Gefundene ICS können in der Detailansicht angezeigt werden. Hier werden alle Daten zu einem ICS zusammengefasst dargestellt.



2.4 Trouble Ticket System

Zu jedem ICS, gehört ein Ticket. Dieses Ticket kann erstellt, bearbeitet und wieder entfernt werden. Eine History zeigt die letzten Änderungen sortiert nach Datum auf.



ICS ThreadMap - v1.0

Prototypen
(SPR)

Dominique Sorg
Benjamin Kehl

Änderungsgeschichte

Datum	Version	Änderung	Autor
03.11.2013	0.1	Erstellung Dokument	Dominique Sorg
04.11.2013	0.2	Überarbeitung und Ergänzung	Benjamin Kehl

Inhalt

1. Zweck

Dieses Dokument beschreibt die Prototypen der ICS ThreatMap Web Applikation. Anhand der Prototypen werden die technischen Risiken abgearbeitet und neu bewertet.

2. Datenbeschaffung Shodan

Die Machbarkeit der Datenbeschaffung bei Shodan wird mit einem PHP Prototyp gezeigt. Die Suchanfragen werden als Queries zu Shodan verschickt und die erhaltenen Suchresultate als JSON ausgegeben. Das Ziel dieses Prototyps ist es die gleichen Daten wie über die Suchfunktion auf der Shodan Webseite abzurufen.

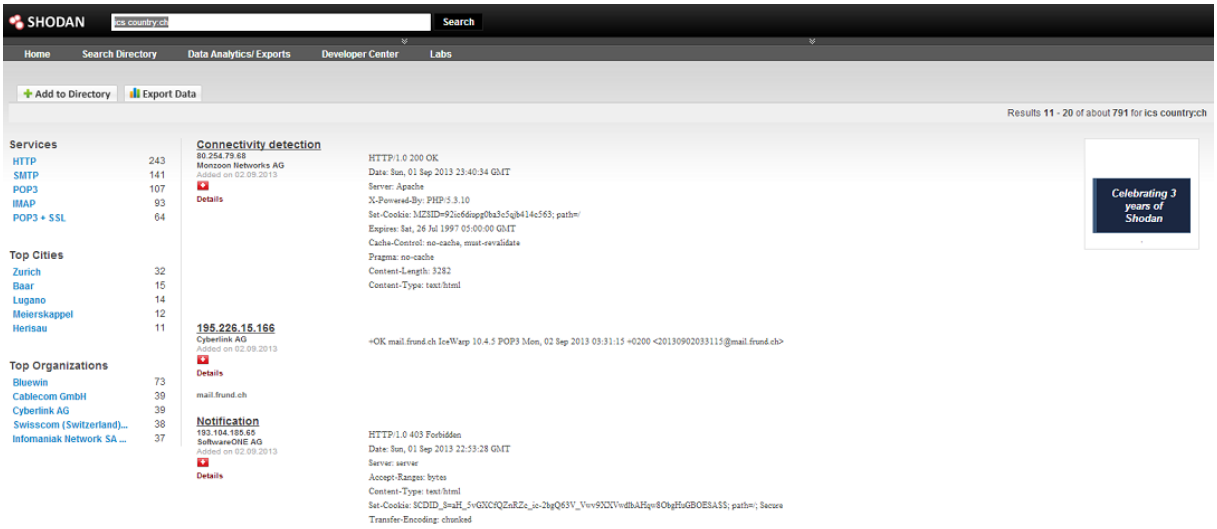


Abbildung 1 Shodan Suche nach ICS

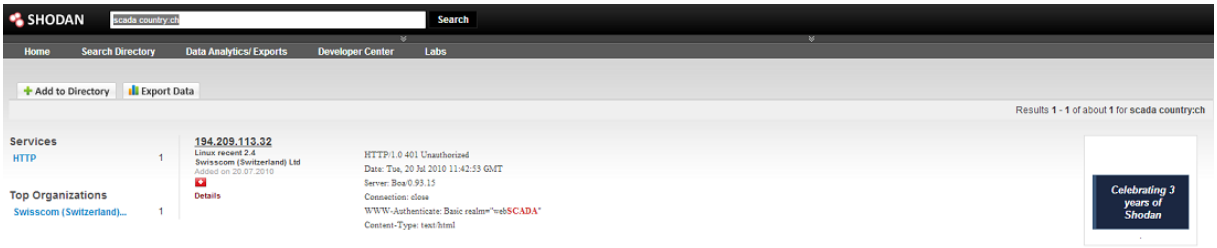


Abbildung 2 Shodan Suche nach SCADA

Das folgende Script (nächste Seite) ist nicht ganz vollständig, aber die Grundfunktionalität funktioniert und es gibt uns zum jetzigen Standpunkt einen guten Einblick auf die Problemstellung. Der PHP Code ist in zwei Dateien unterteilt, das *Main* und die *Shodan* Klasse. Um es zu benutzen kann eine Suchanfrage (Suchanfrage identisch zur Suchfunktion von Shodan) in der Variable *$query* angegeben werden. Zusätzlich muss noch die Art der Query bestimmt werden, die dann ein Resultat als JSON bzw. als Array Object zurückgibt.

Das main.php ist der Startpunkt des Scripts.

Dateiname: `main.php`

```
<?php

include 'shodan.class.php';
```

```
/*
 * 1) enter the desired query below, filter reference at:
 *          - https://developers.shodan.io/shodan-rest.html
 *          - http://www.shodanhq.com/help/filters
 */
```

```
$query = 'ics country:CH';

$api = new ShodanWeb();


/*
 * 2) choose the type of query:
 *      - count = Returns the number of devices that a search query found.
 *      - host  = Lookup all available information for a specific IP address.
 *      - search = Search Shodan for devices using a search query.
 */

$result = $api->count($query);

if(!empty($result)) {
    print_r($result);
} else {
    die("Shodan has no result. Please change your search parameters!");
    exit(1); // A response code other than 0 is a failure
}
?>
```

Die Shodan Klasse tätigt die Query bei Shodan und liefert das Suchergebnis zurück.
Dateiname: `shodan.class.php`

```
<?php

    #
    # Semesterarbeit:   ICS Threat Map V1
    # Institute:        Hochschule für Technik Rapperswil & Compass AG
    # author:           Sorg Dominique / Kehl Benjamin
    # betreuer:         Sprenger Walter
    # version:          beta 0.1
    # date:             20/09/2013
    #

    class ShodanWeb
    {

        private $base_url;
        private $api_key;

        function __construct()
        {
            //default configuration shodan
            $this->base_url = 'http://www.shodanhq.com/api/';
            $this->api_key = 'HApwMdSAHdEt3Mas65fdHl8WQhRBApIKeYF@IL';

            header('Content-type: application/json');
            ini_set('max_execution_time', 300); //300 seconds = 5 minutes
        }

        function __destruct()
        {

        }

        /*
        * Search Shodan for devices using a search query.
        * # Optional Parameter:
        * 1) city -- Use the 'city' filter to find devices located in the given city.
        *            Example: (city:"San Diego", city:"Zurich", ...)
        * 2) country -- The 'country' filter is used to narrow results down by country.
        *               Example: (country:CH, country:DE, ...)
        * 3) geo  -- The 'geo' filter allows you to find devices that are within a
        *            certain radius of the given latitude,longitude,radius.
        *            Example: (geo:32.8,-117,50)
        * 4) hostname -- The 'hostname' filter lets you search for hosts that contain
```

```
*               the value in their hostname.
*               Example: (hostname:"Server: gws", hostname:google, hostname:
*                        nginx, hostname:.de)
* 5) net -- The 'net' filter provides a mechanism for limiting the search
*           results to a specific IP or subnet.
*           Example: (net:216.219.143.14, net:216.219.143.0/24,
*                     net:216.219.0.0/16, apache net:216.0.0.0/8)
* 6) os  -- The 'os' filter is used to search for specific operating systems.
*           Example: (os:"windows 2003", os:linux)
* 7) port -- The 'port' filter is used to narrow the search to specific
*            services. Possible values are: 21, 22, 23 and 80.
*            Example: (port:21, ...)
* 8) before/after -- The before / after filters let you search only for data
*                    that was collected before or after the given date
*                    Example: (before:18/01/2010, apache country:CH
*                              after:22/03/2010 before:4/6/2010, ...)
*
*/
 public function search($query)
 {

    $request = "{$this->base_url}
                search?q={$query}&key={$this->api_key}";
    $response = file_get_contents($request);
    $data = $this->json_decode($response);

    if(empty($data)) {
        return("#| Error, No results for this query.<br />\r\n");
    } else {
        return($data);
    }

 }

/*
 * Returns the number of devices that a search query found.
 */
public function count($query)
{

    $request = "{$this->base_url}
                 count?q={$query}&key={$this->api_key}";
    $response = file_get_contents($request);
    $data = $this->json_decode($response);

    if(empty($data)) {
        return("#| Error, No results for this query.<br />\r\n");
    } else {
        return($data);
    }

}

/*
 * Lookup all available information for a specific IP address
 */
public function host($query)
{

    $request = "{$this->base_url}
                host?ip={$query}&key={$this->api_key}";
    $response = file_get_contents($request);
    $data = $this->json_decode($response);

    if(empty($data)) {
        return("#| Error, No information available for that IP.
               <br />\r\n");
    } else {
        return($data);
    }
```

```
    }

    /*
     * Determine the software based on the banner
     */
     public function fingerprint($query)
     {

        $request = "{$this->base_url}
                    fingerprint?banner={$query}&key={$this->api_key}";
        $response = file_get_contents($request);
        $data = $this->json_decode($response);

        if(empty($data)) {
            return("#| Error, No results for this banner.<br />\r\n");
        } else {
            return($data);
        }

     }

     /*
      * Retrieve information about the current API plan.
      */
     public function info($query)
     {

     }

     /*
      * Search exploit on metasploit, cve, osvdb, exploitdb
      *     01: cve     -- CVE identifier (ex. 2010-0432)
      *     02: osvdb   -- OSVDB identifier (ex. 11666)
      *     03: msb     -- Microsoft Security Bulletin ID (ex. MS05-030)
      *     04: bid     -- Bugtraq identifier (ex. 13951)
      */
     function search_exploits($query)
     {

     }

     /*
      * Search exploit at exploit-db.
      *     01: author    -- Name of the exploit submitter
      *     02: platform  -- Target platform (e.g. windows, linux, etc.)
      *     03: port      -- Service port number
      *     04: type      -- Any, dos, local, papers, remote, shellcode
      */
     function exploit_db_search($query)
     {

     }

     /*
      * Download the exploit code from the Exploit-DB archive.
      */
     function exploit_db_download($id)
     {

     }

     /*
      * Search for a Metasploit module.
      */
     function msf_search($query)
     {
```

```
        }

        /*
         * Download a metasploit module given the fullname of the module (ex.
auxiliary/admin/backupexec/dump).
         */
        function msf_download($query)
        {

        }

       /*
        * Takes a JSON encoded string and converts it into a PHP object
variable.
         */
        public function json_decode($json)
        {
            return json_decode($json);
        }

    }

?>
```

Mit diesem Prototyp konnten die Daten geholt und verarbeitet werden. In einem weiteren Schritt werden die Daten analysiert und verarbeitet. Die folgende Grafik visualisiert, wie die Daten von Shodan geliefert werden. Dabei wird die erhaltene JSON Datei analysiert.

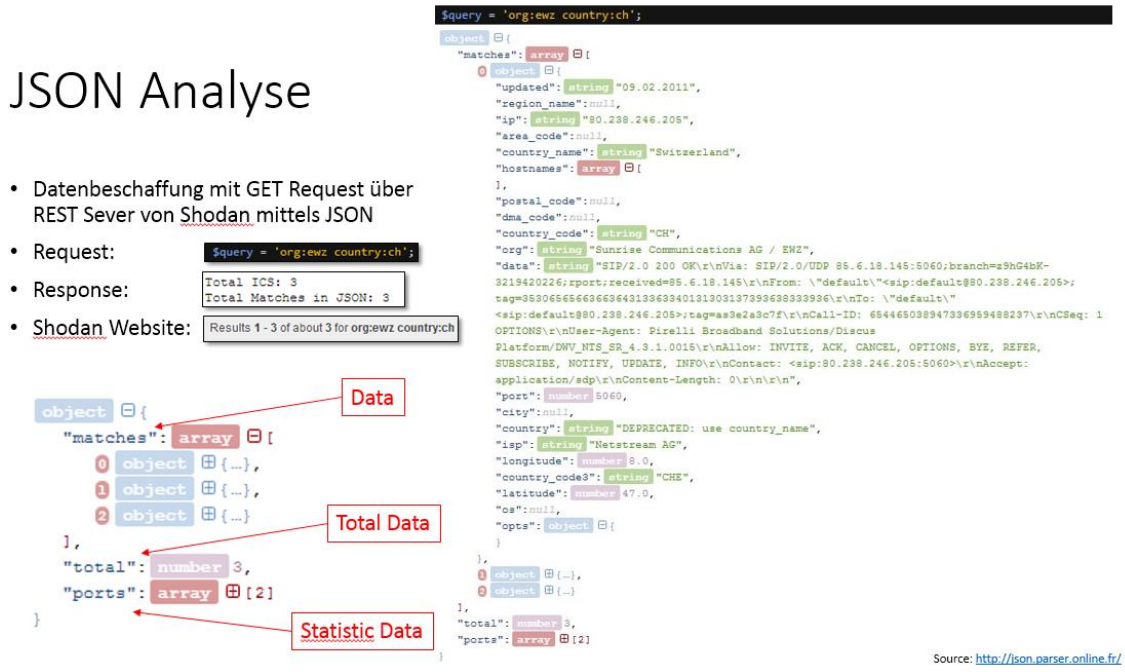


Abbildung 3 JSON Analyse

Jedoch entstanden Probleme bei der Datenbeschaffung die in der folgenden Grafik detaillierter aufgezeigt werden.

Downsides Shodan

- Bei grossen Datenmenge werden die Daten über 'Pages' limitiert übermittelt (Default p=1)

```
public function search($query, $page)
{
    $uri = "{$this->base_url}search?q={$query}&p={$page}&key={$this->api_key}";
    return $this->request($uri, $query);
}
```

- => Pro Page wird 1 Query abgezogen und varriert zwischen 50-100 Einträge pro Page.
- => Für 213 Results sind dies bereits 3 Queries. Bsp: `$query = 'org:axpo country:ch';`
- => Shodan liefert einen Total Wert, jedoch stimmt dieser nicht mit den Anzahl Daten überein (gleiches Problem auch auf der Shodan Webseite:

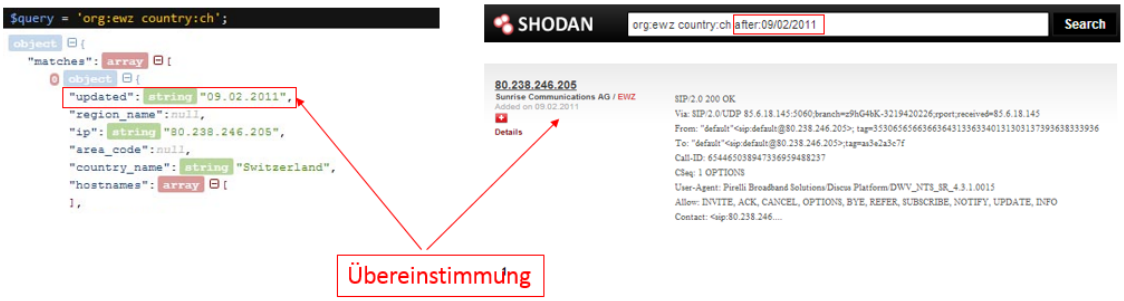
- => Bei Daten-Updates werden ca. 90% der Daten identisch sein, jedoch werden wieder 3 Queries verbucht

Abbildung 4 Probleme mit Shodan

Daraus erarbeiteten wir uns Lösungsvorschläge.

Lösungsvorschlag Query Problem

- Datenbeschaffung unterscheiden zwischen «volle Datenbeschaffung» oder nur die Änderungen abgleichen



- Es werden nur Daten abgefragt, die sicher in der DB abgeglichen werden müssen

Abbildung 5 Lösungsvorschlag Shodan

Damit wir das Update Script in Zend Framework 2 verwenden konnten, entstand ein Prototyp als Zend Framework 2 Modul. Diese Modul nannten wir *UpdateScript*.

Shodan Prototyp

- Datenbeschaffung wird als unabhängiges Modul in ZF2 implementiert
- Modul ist über Command Line, Shell, Bash, etc. Skript abrufbar
- Skript kann als Scheduled Job im OS gesetzt werden
- bsp: Windows Batch Skript

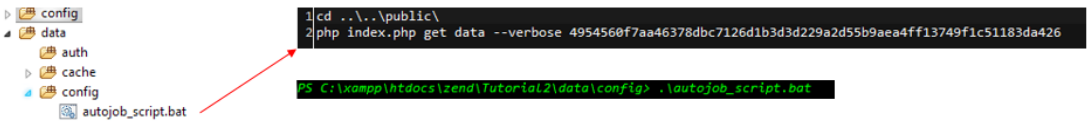


Abbildung 6 ZF2 Datenbeschaffung Shodan

Die Daten kommen in einer lesbaren Form daher. In einem weiteren Schritt können die Daten in eine MySQL-Datenbank abgespeichert werden.

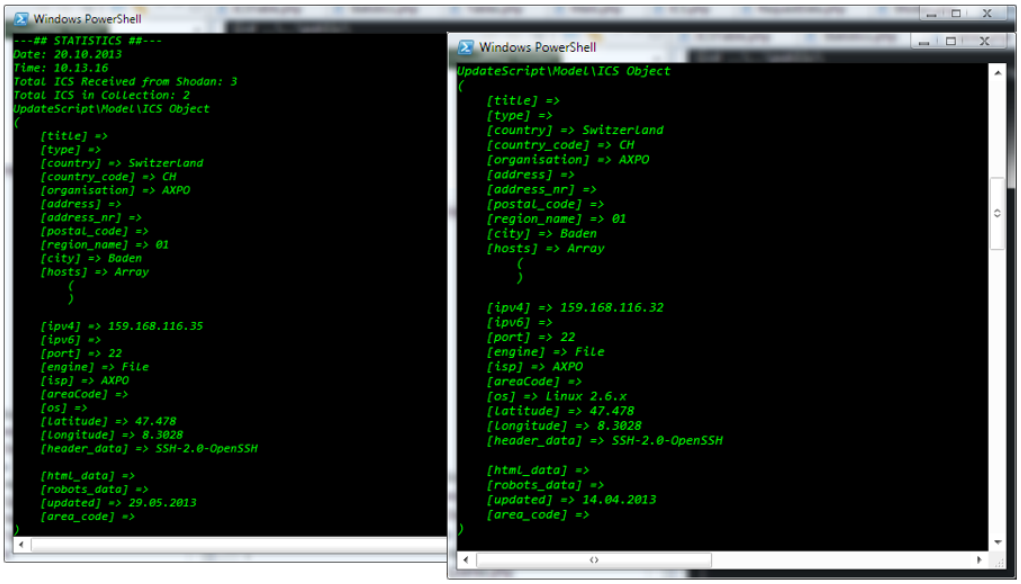


Abbildung 7 Datenausgabe eines ICS aus Shodan

Es kann durchaus vorkommen, dass der Shodan Dienst unerreichbar ist. Dies passierte uns manchmal bei der Entwicklung. Dabei erscheint folgender Fehler.

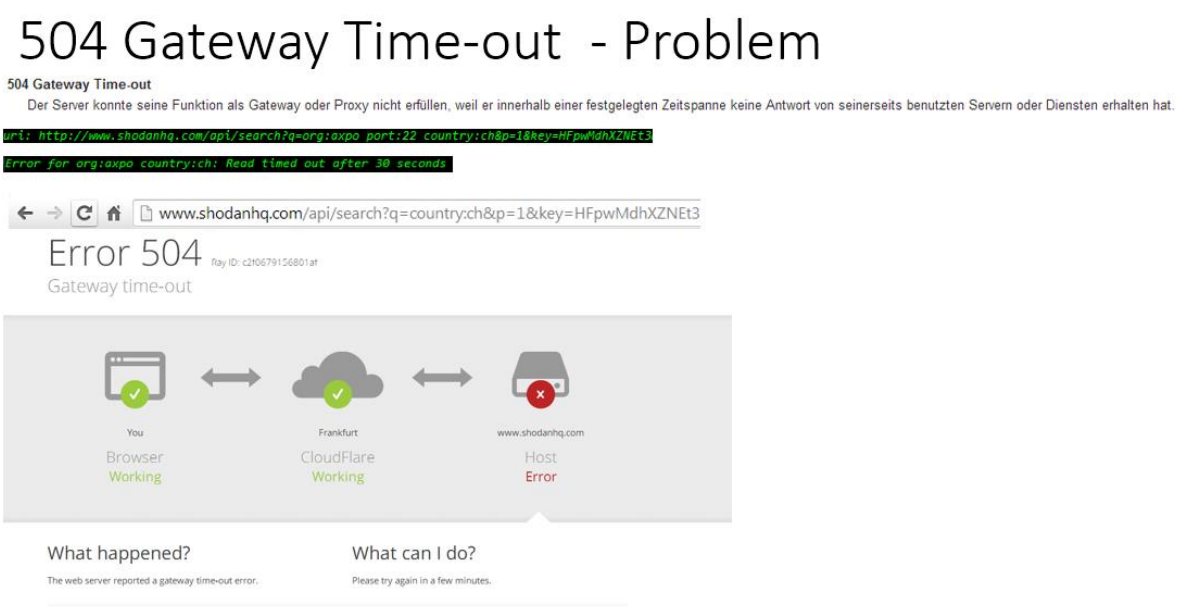


Abbildung 8 Unerreichbarkeit Shodan

Der Shodan-Dienst ist von Zeit zu Zeit nicht erreichbar. Dies hätte Auswirkungen auf unsere Anfragen, da *Timed-Out-Exceptions* auftreten würden. Dies hätte zur Folge, dass das UpdateScript keine neuen Daten an diesem Tag holen konnte.

Damit wir dem Problem vorbeugen können, werden neue Suchfilter in der Datenbank erst nach einer erfolgreichen Abfrage als „nicht mehr neu“ markiert. Um bestehende Daten zu aktualisieren, werden die Ergebnisse des Vortags miteinbezogen. Dadurch können uns keine Suchresultate entgehen. Zudem konnte auch das Problem der Zeitverschiebung zwischen dem Shodan Server und unserem Server gelöst werden. Da wir die Suchresultate vom Vortag miteinbeziehen, kann sichergestellt werden, dass alle Einträge von Shodan in unsere Applikation erfasst werden.

3. Google Map

Die Einarbeitung in die Google Map API erfolgte mit JavaScript und JQuery. Dabei wurde als Prototyp folgende Map erstellt, der einen *Marker* wie auch einen *InfoWindow* beinhaltet. Damit würde dieser Prototyp die Vorlage bilden, um später dynamisch die Marker nach den Koordinaten der Kontrollsysteme von Shodan zu setzen.

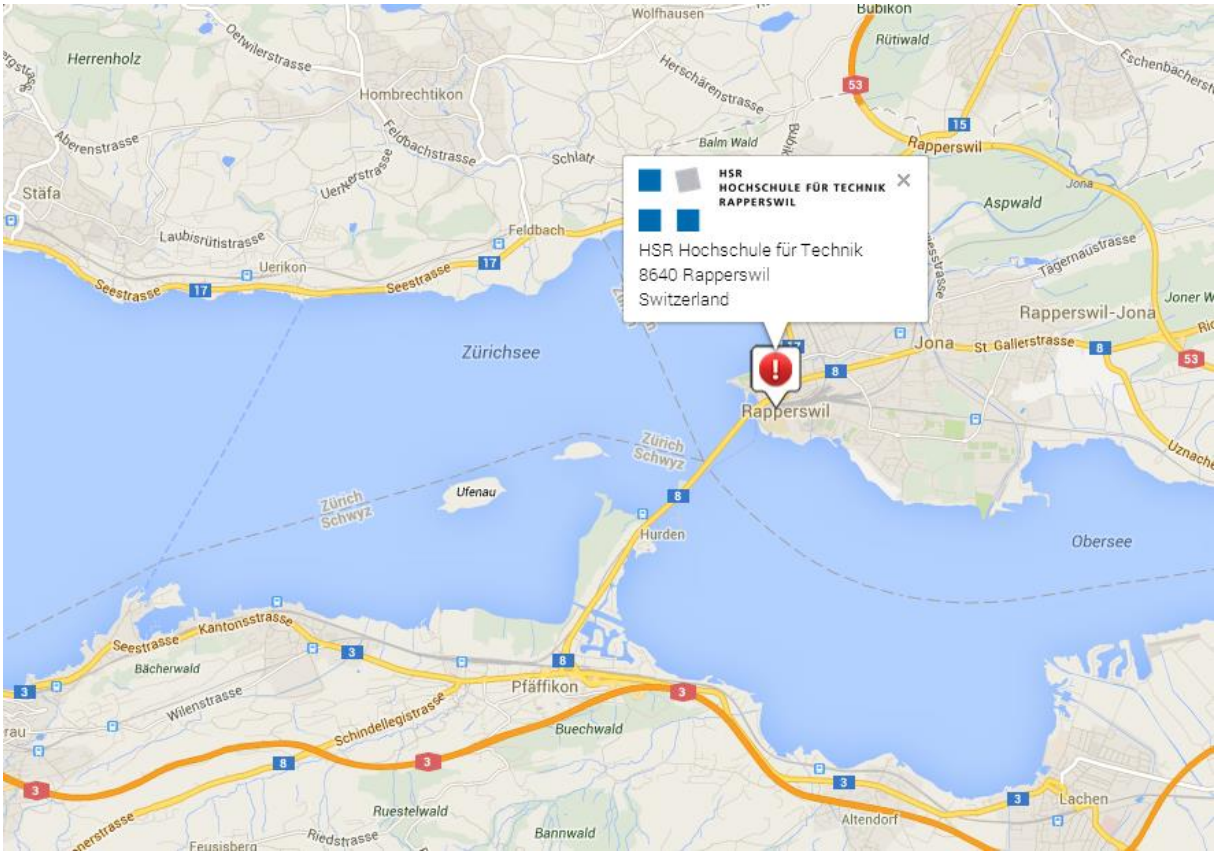


Abbildung 9 Google Map Prototyp

Der Prototyp ist unterteilt in *index.html*, welches die externe JavaScript *google.map.js* lädt. Die Map selbst wird im *index.html* an der Position *id="map_canvas"* geladen.

Dateiname: *marker.png*

Dateiname: *index.html*

```
<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="utf-8">



        <link rel="stylesheet" type="text/css" media="screen, projection,
print" href="stylesheet.css" />

        <meta name="viewport" content="width=device-width, initial-scale=1.0,
user-scalable=no" />
```

```
<script type="text/javascript"
src="http://maps.googleapis.com/maps/api/js?v=3.exp&sensor=false"></script>
<script type="text/javascript" src="google.map.js" charset="utf-8"></script>

        <script type="text/javascript">
            function loadData()
            {
                requestData();
            }
            window.onload = loadData;
        </script>

    </head>

    <body>
        <div id="map_canvas" >loading..</div>
    </body>

</html>
```

Dateiname: *google.map.js*

```
/**
 * Semesterarbeit:       ICS Threat Map V1
 * Institute:            Hochschule fuer Technik Rapperswil & Compass AG
 * author:               Sorg Dominique / Kehl Benjamin
 * betreuer:             Sprenger Walter
 * version:              beta 0.1
 * date:                 20/09/2013
 */

// default latitude/longitude coordinates
var latitude = 47.227109;
var longitude = 8.815101;

// map object
var map;

// Enable the visual refresh
google.maps.visualRefresh = true;

// load default properties and map
function initialize()
{

    var mapProperties = {
        center: new google.maps.LatLng(this.latitude, this.longitude),
        zoom: 8,
        mapTypeId: google.maps.MapTypeId.ROADMAP //Map Types: ROADMAP,
SATELLITE, HYBRID, TERRAIN
      };

      map = new google.maps.Map(document.getElementById("map_canvas"),
mapProperties);

}

function requestData()
{
    /* TODO: request data */

    parseData();
}

function parseData()
{
```

```
    /* TODO: parse data */
    var lat = 47.223378;
    var lon = 8.817312;
    var title = 'HSR Rapperswil';
    var info =  '<div>' +
            '<img src="http://security.hsr.ch/msevote/images/hsr_logo.png"
height="100%" width="180px" /><br />' +
            'HSR Hochschule für Technik\n<br />' +
            '8640 Rapperswil\n<br />' +
            'Switzerland\n<br />' +
            "</div>";

    // load markers with parsed data
    setTimeout(function() {
        coordinates(lat, lon, title, info);
    }, 200);
}

// place markers with a infoWindow on the map
function coordinates(latitude, longitude, title, info)
{
    var image = {
          url: 'marker.png',
          //size: new google.maps.Size(42, 42),
          //scaledSize: new google.maps.Size(40, 40)
    };

    var marker = new google.maps.Marker({
        map: map,
        title: title,
        position: new google.maps.LatLng(latitude, longitude),
        animation: google.maps.Animation.DROP,
        icon: image
    });

    var infoWindow = new google.maps.InfoWindow({
        content: info
    });

    google.maps.event.addListener(marker, 'click', function() {
        infoWindow.open(marker.get('map'), marker);
        marker.setAnimation(google.maps.Animation.BOUNCE);
    });

    google.maps.event.addListener(infoWindow, 'closeclick', function() {
        marker.setAnimation(null);
    });

}

// ensures that the map is placed on the page after the page is fully loaded
google.maps.event.addDomListener(window, 'load', initialize);
```

4. Bootstrap

Die Verwendung von einem UI-Framework wie *Bootstrap* wird mit diesem Prototyp sichergestellt. Damit lassen sich Responsive Webanwendungen realisieren. Mit Bootstrap lässt sich ICS ThreatMap später auch über das Handy oder Tablet bedienen.

Abbildung 10 Bootstrap

5. Zend Framework 2

Mit einem Zend Framework Prototyp entstand folgende Anwendung. Sie diente für die Bestimmung der Layout-Templates und als Basis zu ICS ThreatMap.

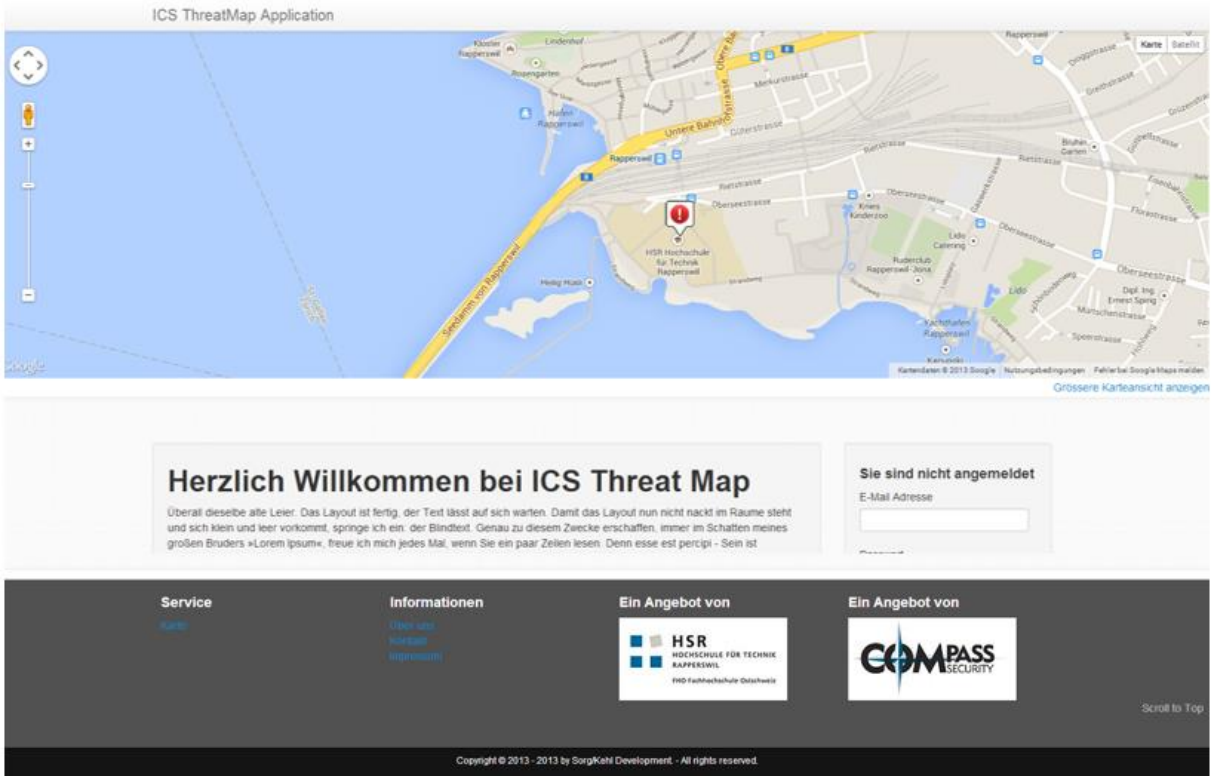


Abbildung 11 Zend Framework Prototyp

Der ZF2-Prototyp konnte bereits die Templates wie *Header*, *Content*, *Sidebar* und *Footer* in richtiger Position darstellen. Dank integriertem Bootstrap ist der Prototyp auch „responsive“ fähig. Die GoogleMap Karte wurde im Content-Bereich integriert. Zusätzlich kann die Karte auch im Vollbildmodus geöffnet werden. Pages wie *Über uns*, *Kontakt* oder *Impressum* wurden ebenfalls implementiert.

6. Benutzerverwaltung

Die Benutzerverwaltung ermöglicht das Registrieren, das Anmelden und das Ausloggen von einem Benutzer. Die weiteren Funktionen wie Benutzer verwalten und erweiterte Rollen werden erst in der Implementierungsphase implementiert.

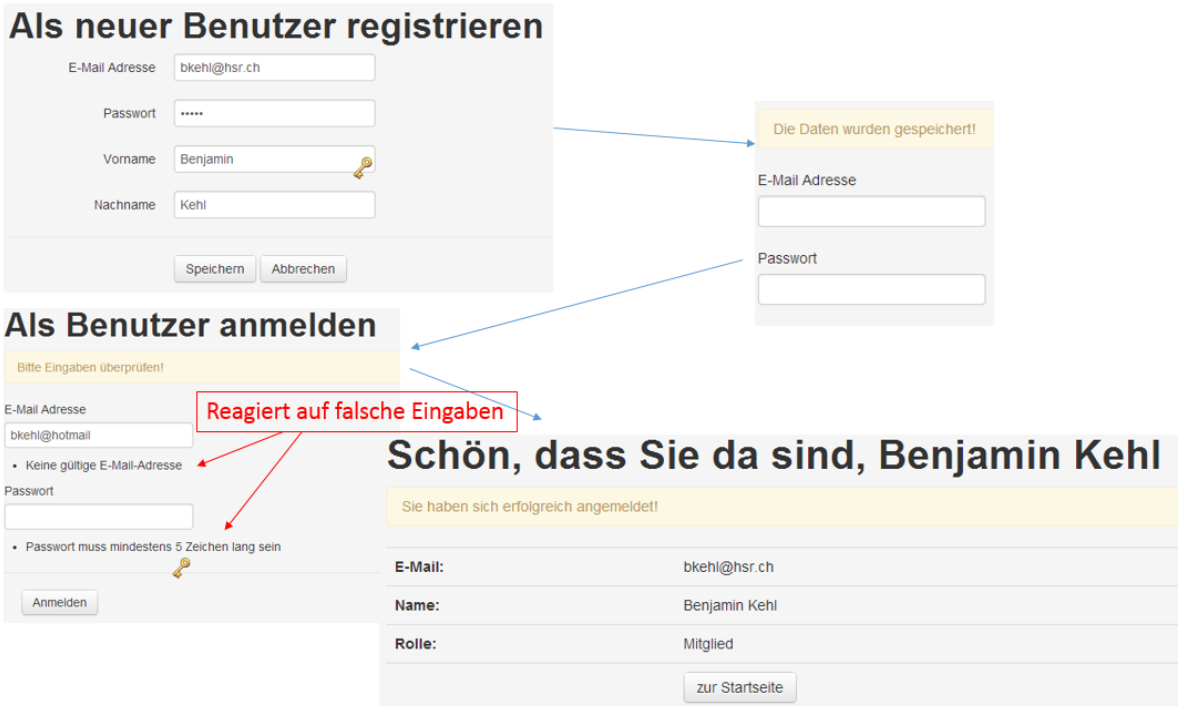


Abbildung 12 Benutzerverwaltung

Im folgendem werden ein paar wichtige Codeschnipsel aufgezeigt:

```php
public function login(array $data)
    {
        // get form and set data
        $form = $this->getForm('login');
        $form->setData($data);

        // check for invalid data
        if (!$form->isValid()) {
            $this->setMessage('Bitte Eingaben überprüfen!');
            return false;
        }

        // get valid user entity object
        $user = $form->getData();

        // get authentication
        $authentication = $this->getAuthentication();
        $authentication->getAdapter()->setIdentity($user['email']);
        $authentication->getAdapter()->setCredential($user['password']);

        // authenticate
        $result = $authentication->authenticate();

        // get messages
        $messages = $result->getMessages();
```

```
        // set first message
        $this->setMessage($messages[0]);

        // check result
        if (!$result->isValid()) {
            return false;
        }

        return $result->getIdentity();
    }
```

```
public function logout()
    {
        // get authentication
        $authentication = $this->getAuthentication();

        // clear identity
        $authentication->clearIdentity();

        // get session namespace
        $authNamespace = new Container(Session::NAMESPACE_DEFAULT);

        // clear session
        $authNamespace->getManager()->destroy();

        // set message
        $this->setMessage('Sie wurden abgemeldet!');

        return true;
    }
```

7. Re-analyse Technische Risiken

	Risiko	Beurteilung
R1	Datenbeschaffung	Mit dem Prototyp ist die Datenbeschaffung bei Shodan möglich. Daten können in die Datenbank gespeichert werden
R2	Shodan Webserver Unerreichbar	Bei Ausfällen von Shodan werden die Suchergebnisse ein Tag später geholt.
R3	Google Map API	Die Google Map Karte kann mit Markers angezeigt werden. Um die Ladezeit zu kürzen wird nur der sichtbare Bereich angezeigt.
R4	Datenbankaktualisierungen	Muss laufend kontrolliert werden.
R5	Loginbereich	Da sich die Anforderungen im Login Bereich geändert haben, erwarten wir ein Mehraufwand.
R6	Sicherheit	SQL Injection werden mit *prepared statements* eingedämmt.
R7	Backup	Es werden Snapshots vom Server gemacht
R8	GIT	Das Projekt ist lokal und auf GIT vorhanden. Damit können wir weiterarbeiten auch wenn GIT mal nicht zur Verfügung steht.

ICS ThreadMap - v1.0

Software Qualitätsmanagement (SQM)

Dominique Sorg
Benjamin Kehl

Änderungsgeschichte

Datum	Version	Änderung	Autor
16.11.2013	0.1	Kompatibilität Spezifikation	Dominique Sorg
14.12.2013	0.2	System Spezifikation	Benjamin Kehl
18.12.2013	0.3	Usability Test	Dominique Sorg

Inhalt

1. Systemtestspezifikation

1.1 Angaben zur Durchführung

Jedes implementierte Feature muss auf die Funktionalität getestet werden. Um Nebeneffekte zu vermeiden, wurden alle Features kurz nach der Entwicklung in verschiedenen Szenarien getestet.

Nachfolgend werden alle Ergebnisse der Systemtests dokumentiert.

1.2 Protokoll

1.2.1 Überblick aller Tests

Use Case Titel	Priorität	Implementiert	Fehler/Unschönheiten	Status
Registration	1	Ja		✓
An-/Abmeldung	1	Ja		✓
Benutzer CRUD	1	Ja		✓
Benutzer aktivieren	1	Ja		✓
ICS suchen und ansehen	1	Ja		✓
ICS Daten auf Map ansehen	1	Ja		✓
ICS CRUD	1	Ja		✓
Filter CRUD	1	Ja		✓
Trouble Ticket System CRUD	1	Ja		✓
ICS klassifizieren	1	Ja	Gruppieren von ICS nicht implementiert	(✓)
White-/Blacklist führen	2	Nein	Nicht implementiert	✗
Filterauflistung eines ICS	2	Ja		✓
Preview Ansicht bei Filtersetzung	2	Nein	Nicht implementiert	✗
Datenbeschaffung Updatescript	1	Ja		✓
Verfügbarkeit der ICS in Engines überprüfen	2	Nein	Nicht implementiert	✗
Emailbenachrichtigung	1	Ja		✓

1.2.2 Implementierte Use Cases

1.2.2.1 Registration

Subtasks	Fehler/Unschönheiten/Verbesserungen	Status
Formular und Validierungen	Schönere Positionierung der Formularelemente	✓
Neuer Benutzer wird erstellt	Organisation des Benutzers sowie die Rolle sind in der Tabelle des Benutzers. Man könnte diese von der Benutzertabelle trennen.	✓
Account deaktivieren		✓

1.2.2.2 An-/Abmeldung

Subtasks Login	Fehler/Unschönheiten/Verbesserungen	Status
Formular und Validierungen	Im Sidebar: Responsive Layout ist hier nicht ganz unterstützt, da es sich um eine Tabelle handelt	✓
Passwort-Hash Vergleich		✓
Session erstellen		✓

Subtasks Logout	Fehler/Unschönheiten/Verbesserungen	Status
Benutzer identifizieren		✓
Session löschen		✓

1.2.2.3 Benutzer CRUD

Subtasks Update	Fehler/Unschönheiten/Verbesserungen	Status
Formular und Validierungen		✓
Daten aus der DB in Formular einbinden		✓
Rollen setzen		✓
Passwort zurücksetzen		✓
Account aktivieren		✓
Speicherung DB		✓

Subtasks Delete	Fehler/Unschönheiten/Verbesserungen	Status
Keine		

Subtasks View	Fehler/Unschönheiten/Verbesserungen	Status
Benutzerliste		✓
Benutzerstatus (Aktiv/Inaktiv)		✓
Paginator		✓

1.2.2.4 ICS suchen und ansehen

Subtasks Suche	Fehler/Unschönheiten/Verbesserungen	Status
Suche nach Titel, IP, Port		✓
Suche nach Bedrohung		✓
Suche nach Kategorie		✓
Suche von und bis	Suche nach Benutzeränderungen wäre hier noch wünschenswert gewesen. Momentan nur wann das ICS in die DB eingefügt wurde.	✓
Suche nach Firma		✓
Suche nach Service		✓
Suche nach Produkt		✓
Suche über die URL		✓
Paginator		✓

Subtasks View	Fehler/Unschönheiten/Verbesserungen	Status
Anzeige ICS mit UserChanges		✓
Anzeige Screenshot		✓
Anzeige Suchfilter		✓
Anzeige History		✓

1.2.2.5 ICS Daten auf Map anzeigen

Subtasks	Fehler/Unschönheiten/Verbesserungen	Status
Beschränkte Ansicht		✓
Volle Ansicht		✓
Anzeige ICS Cluster/Icons		✓
Anzeige ICS Liste		✓
Anzeige in Vollansicht		✓

1.2.2.6 ICS CRUD

Subtasks Add	Fehler/Unschönheiten/Verbesserungen	Status
Formular und Validierungen		✓
Daten aus der DB in Formular einbinden		✓
Benutzeränderung setzen		✓
Speicherung DB		✓

Subtasks Edit	Fehler/Unschönheiten/Verbesserungen	Status
Formular und Validierungen		✓
Daten aus der DB in Formular einbinden		✓
Benutzeränderung setzen		✓
Speicherung DB		✓

Subtasks Delete	Fehler/Unschönheiten/Verbesserungen	Status
Warnungtext anzeigen		✓
Alle ICS Changes werden auf Archiv gesetzt		✓

1.2.2.7 Filter CRUD

Subtasks Add	Fehler/Unschönheiten/Verbesserungen	Status
Formular und Validierungen	Schönere Positionierung der Formularelemente	✓
Daten aus der DB in Formular einbinden		✓
Speicherung DB		✓

Subtasks Edit	Fehler/Unschönheiten/Verbesserungen	Status
Formular und Validierungen	Schönere Positionierung der Formularelemente	✓
Daten aus der DB in Formular einbinden		✓
Speicherung DB		✓

Subtasks Delete	Fehler/Unschönheiten/Verbesserungen	Status
Warnungstext anzeigen		✓
Alle ICS Changes werden auf Archiv gesetzt		✓

Subtasks View	Fehler/Unschönheiten/Verbesserungen	Status
Filter anzeigen		✓
Paginator		✓

1.2.2.8 Trouble Ticket System CRUD

Subtasks Add	Fehler/Unschönheiten/Verbesserungen	Status
Formular und Validierungen	Fehlermeldungen werden teils in Englisch angezeigt	✓
Daten aus der DB in Formular einbinden		✓
Ticketzuordnung mit ICS Keyword		✓
Speicherung DB		✓

Subtasks Edit	Fehler/Unschönheiten/Verbesserungen	Status
Formular und Validierungen	Fehlermeldungen werden teils in Englisch angezeigt	✓
Daten aus der DB in Formular einbinden		✓
Speicherung DB		✓

Subtasks View	Fehler/Unschönheiten/Verbesserungen	Status
Anzeige aller überfälligen Tickets auf der Startseite		✓
Paginator		✓

Subtasks Delete	Fehler/Unschönheiten/Verbesserungen	Status
Warnungstext anzeigen		✓
Ticket wird archiviert		✓
Ticket wird auf der Startseite nicht mehr angezeigt		✓

1.2.2.9 ICS Klassifizieren

Subtasks	Fehler/Unschönheiten/Verbesserungen	Status
Mit Icons klassifizierbar		✓
Setzen des Typs, Hersteller und Produkt		✓
Ähnliche ICS gruppieren		×

1.2.2.10 Filterauflistung eines ICS

Subtasks	Fehler/Unschönheiten/Verbesserungen	Status
Filter werden aufgelistet		✓
Filter sind direkt bearbeitbar		✓

1.2.2.11 Datenbeschaffung UpdateScript

Subtasks	Fehler/Unschönheiten/Verbesserungen	Status
UpdateScript auf Win/Linux ausführbar		✓

1.2.2.12 Emailbenachrichtigung

Subtasks	Fehler/Unschönheiten/Verbesserungen	Status
Emailbenachrichtigung nach Registration an alle Admins		✓
Emailbenachrichtigung eines Benutzers nach dem er aktiviert wurde		✓

Emailbenachrichtigung mit Statistik nach Update Script		✓

1.2.3 Nicht implementierte Use Cases

Folgende Use Cases konnten nicht mehr implementiert werden:

Use Case	Priorität	Bemerkung/Grund
ICS klassifizieren (Gruppieren von ähnlichen ICS)	1	Wurde als zu komplex klassifiziert, da das Gruppieren von ICS folgende Merkmale benötigte: • Geolocation der ICS Daten • Zusätzliche Informationen eines Benutzers benötigt (Z.B. Organisation, Klassifikation) • Eine Gruppierung komplett neu auf der Karte dargestellt werden musste
Preview Ansicht bei Filtersetzung	2	Eine Preview-Ansicht ist ebenfalls nicht ganz einfach, da eine Suche über die REST-Funktion von Shodan durchgeführt und die jeweiligen Resultate als JSON auf der Filterseite als Liste geparst werden musste.
Black- / Whitelist	2	Wurde nicht unbedingt als notwendig erachtet, da durch das spezifische und richtige Setzen von Suchfiltern bereits gute Resultate zurückgeliefert wurden.
Verfügbarkeit der ICS in Engines überprüfen	2	Konnte zeitlich nicht mehr implementiert werden.

1.3 Verbesserungsmöglichkeiten

Alles in allem hat sich ICS ThreatMap zu einer ordentlichen Webapplikation gemausert und ist im jetzigen Stand bereits ein gutes Werkzeug für die Suche und Verwaltung von kritischen Infrastrukturen. Jedoch gäbe es noch einige sinnvolle Funktionen, die wir leider nicht mehr implementieren konnten (Siehe 1.2.3 Nicht implementierte Use Cases).

Im Folgenden eine Liste der Funktionen:

- ICS gruppieren (Use Case *ICS klassifizieren Prio 1*)
- Eigene Kategorien über die Webapplikation hinzufügen
- Suche nach ICS, die von einem Benutzer geändert wurden
- Preview Ansicht bei Filtersetzung (Use Case *Preview Ansicht bei Filtersetzung Prio 2*)
- Verfügbarkeit der Engine überprüfen (Use Case *Verfügbarkeit der Engine überprüfen Prio 2*)
- Black- und Whitelist führen (Use Case *Black- und Whitelist Prio 2*)
- Datenbeschaffung Google und weitere

1.3.1 Mögliche Detailverbesserungen

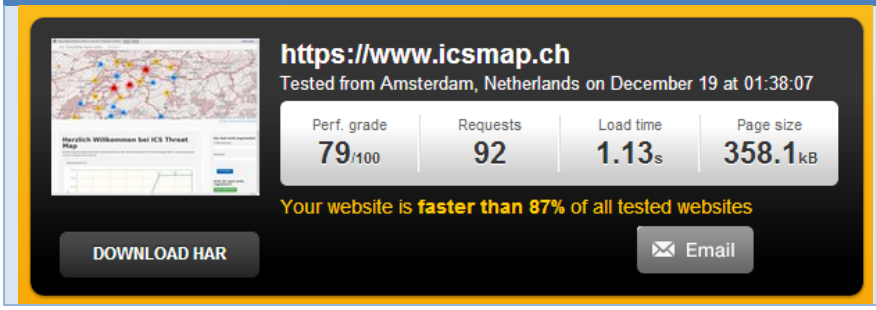
Mögliche Detailverbesserungen wurden bereits im Kapitel 1.2.2 Implementierte Use Cases erwähnt.

1.4 Ladezeiten

Die Ladezeiten von ICS ThreatMap wurde mit Pingdom Speed[1] Test gemessen. Dabei wurde ICS ThreatMap aus drei unterschiedlichen Regionen aufgerufen.

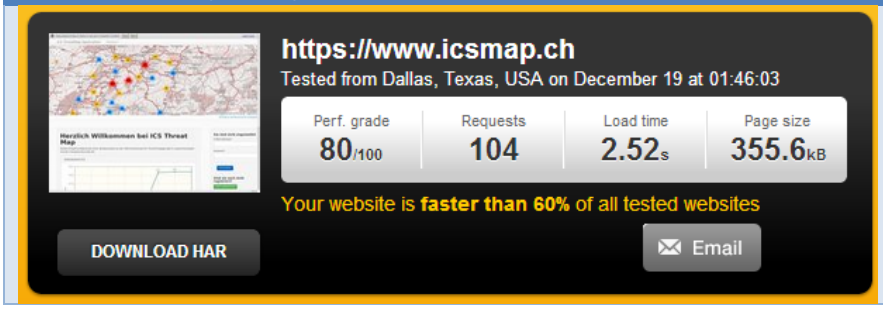
[1] *Speed Test Messung.* (2013). Abgerufen am 18. 12 2013 von Pingdom Tools: http://tools.pingdom.com

Test von: Amsterdam, Netherlands



Test von: New York City, New York, USA

Test von: Dallas, Texas, USA



Eine Anforderung an die Web Applikation ist das Laden von Daten auf einer Google Karte. Die Grundidee ist, das Laden grosser Datenmengen zu vermeiden. Aus diesem Grund werden nur die Daten geladen die der Benutzer auch gerade auf seinen Bildschirmen sehen könnte. Jedoch wurde als Anforderung genannt, dass auch die ganze Schweiz ersichtlich sein soll. Beim Herauszoomen kann es dazu führen das die ganze Datenbank auf der Karte repräsentiert wird. Bei grossen Datenmenge kann dies schon mal etwas dauern. Ausserdem müssen die Rohdaten aus Shodan mit den Benutzeränderungen zusammengeführt werden, um zum Beispiel den korrekten Bedrohungs-Schweregrad anzuzeigen. Aus Performance-Gründen ist dies noch der langsamste Teil der Applikation, welches sich mit der Zunahme von Daten nicht verbessern wird. Standartmässig zoomen wir ein wenig in die Karte rein, damit beim Laden bereits so früh wie möglich Daten dem Benutzer präsentiert werden können. In diesem Bereich können Performance Optimierungen möglich, die wir aus Zeitgründen nicht mehr durchführen und verbessern konnten.

2. Kompatibilitätspezifikation

2.1 Einführung

Es gibt viele unterschiedliche Browser auf dem Markt, wie z.B. Chrome, Firefox, Opera, Internet Explorer, etc. Dabei hat jeder Browserhersteller einige spezielle Tags für die Darstellung der Inhalte im Browser, die von anderen nicht unterstützt werden. Das W3C-Konsortium gibt offizielle Standards vor, um solche Unstimmigkeiten zu vermeiden. Dies hat sich zwar in den letzten Jahren um einiges verbessert, jedoch kann es trotzdem noch zu Browserinkompatibilitäten führen. In diesem Dokument sollen die Ergebnisse protokolliert werden.

2.2 Angaben zur Durchführung

Für die Webapplikation wird ein Browser Kompatibilitätstest von zwei verschiedenen externen Anbietern wie *Browsershots* und *Browserstack*[2] durchgeführt. Diese Anbieter verwenden virtuelle Maschinen mit unterschiedlichsten Betriebssysteme- und Browserversionen. Der Nutzer kann die zu testende URL angeben und nach verschiedenen Betriebssysteme- und Browserversionen filtern. Die angegebenen virtuellen Maschinen erstellen für die zu testende URL einen Screenshot.

Das folgende Bild zeigt, wie wir im Browsershots die verschiedensten Browser mittels verschiedenen Betriebssysteme getestet haben.

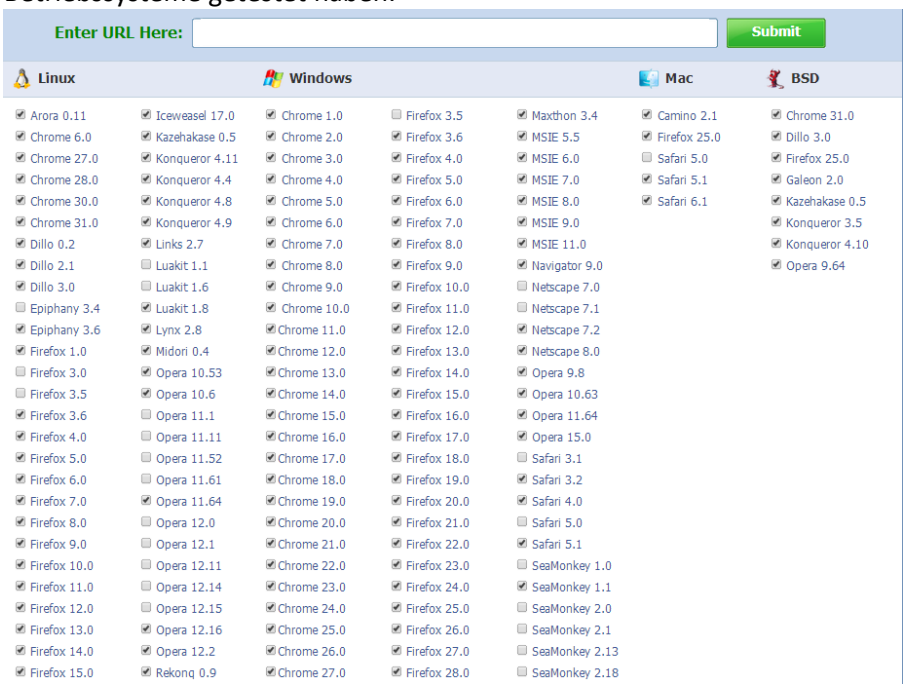


Abbildung 1 Browsershots Compatibility Testing

[2] Siehe Quellenverzeichnis für weitere Informationen

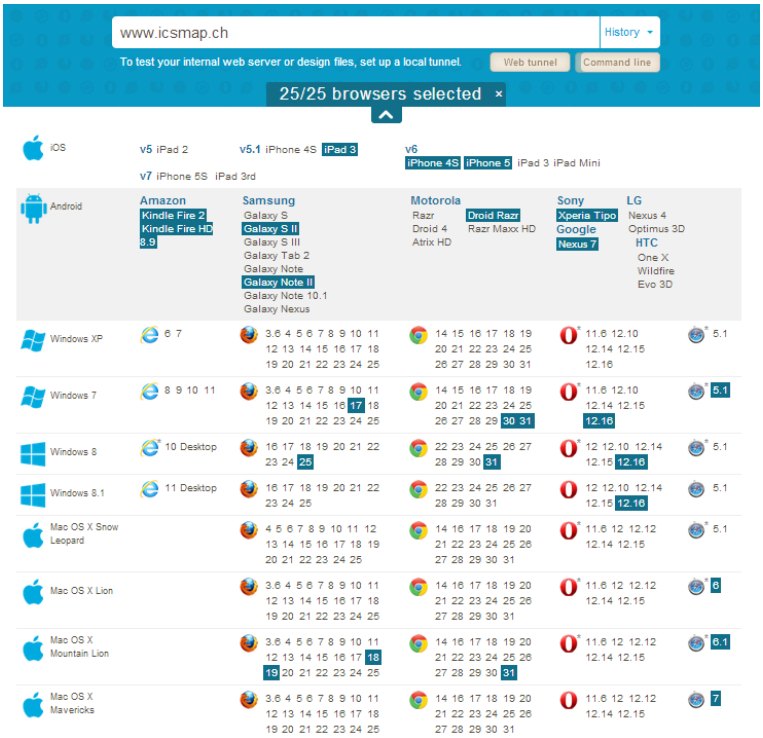


Abbildung 2 Browserstack Compatibility Testing

Nachfolgend werden die Ergebnisse dokumentiert.

2.3 Einschränkung

Wir schränken uns auf die öffentlich kostenlosen Dienste der obengenannten Anbieter ein. Wir testen die Browser Internet Explorer, Firefox, Chrome, Safari und Opera.

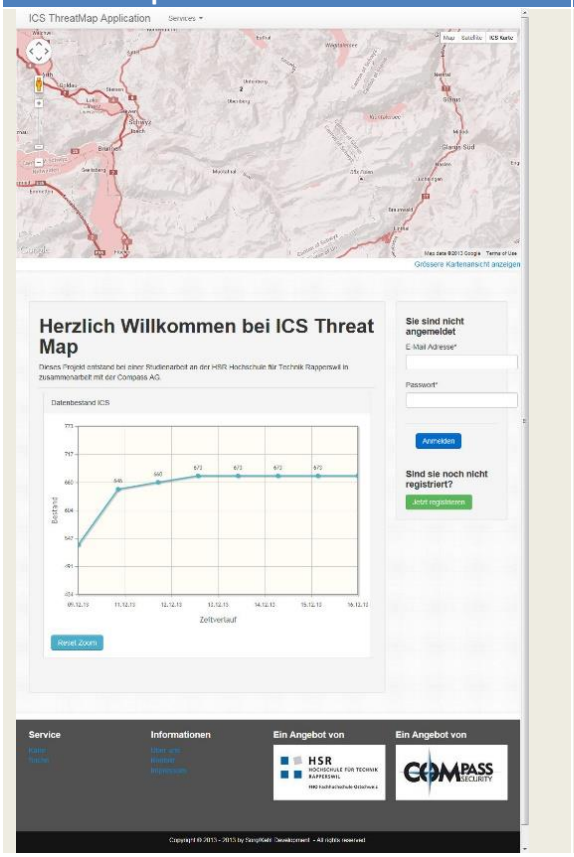
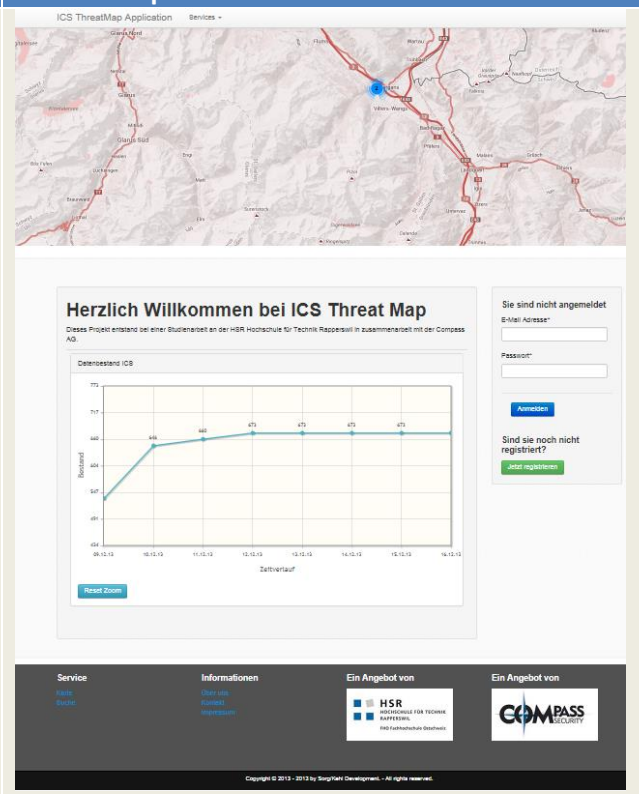
2.4 Protokoll

2.4.1 Überblick aller Tests

Getestet wird vor allem mit modernen Browsern, die heutzutage im Einsatz sind.

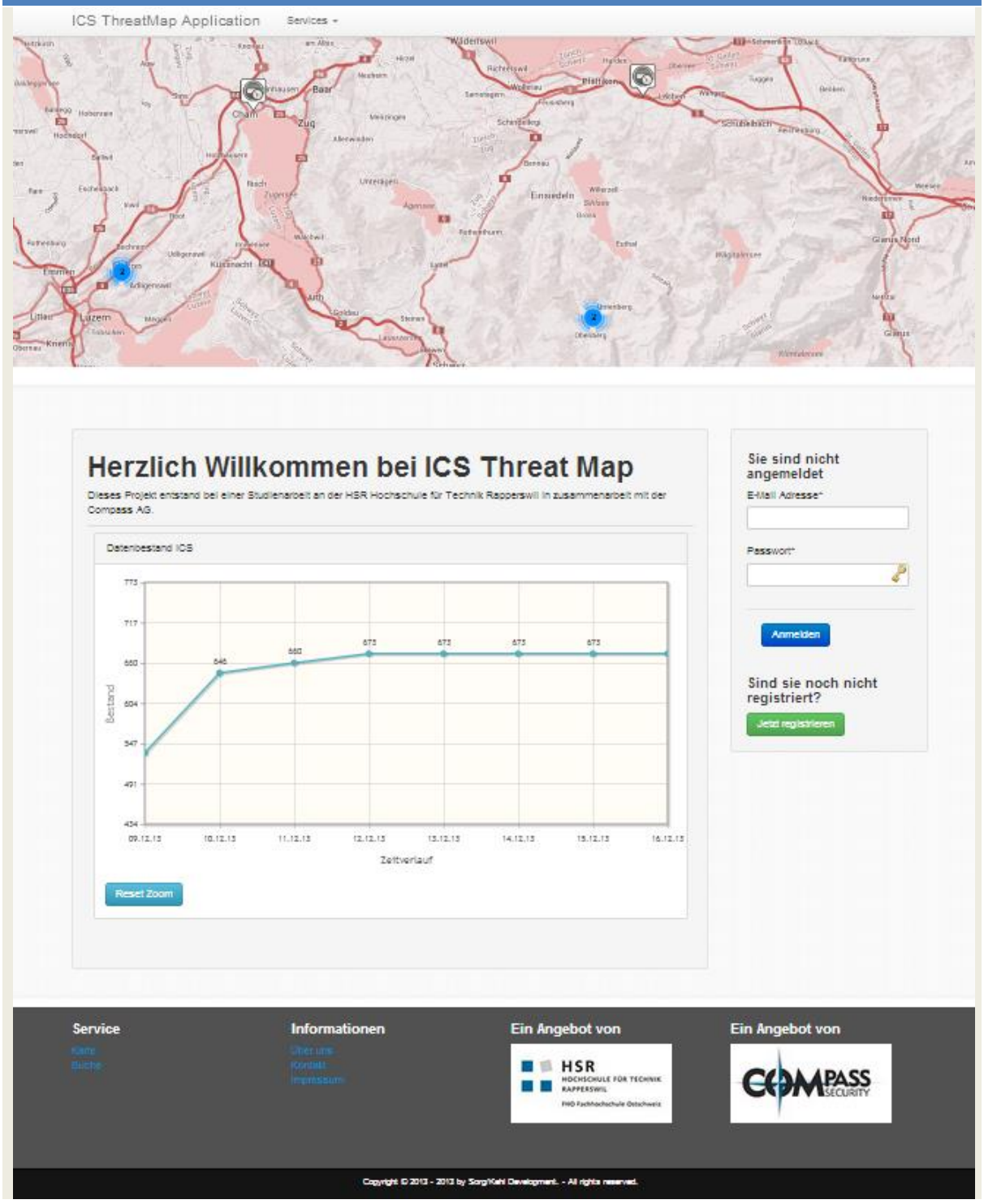
Browser	Fehler/Unschönheiten	Status
Internet Explorer 9		✓
Internet Explorer 10		✓
Firefox		✓
Chrome		✓
Opera		✓
Safari		✓
IPhone Safari		✓
IPad Safari		✓
Android LG Nexus	Screenshot wurde vor dem kompletten Laden der Webseite erstellt	(✓)

2.4.2 Internet Explorer

Internet Explorer 9	Internet Explorer 10
	

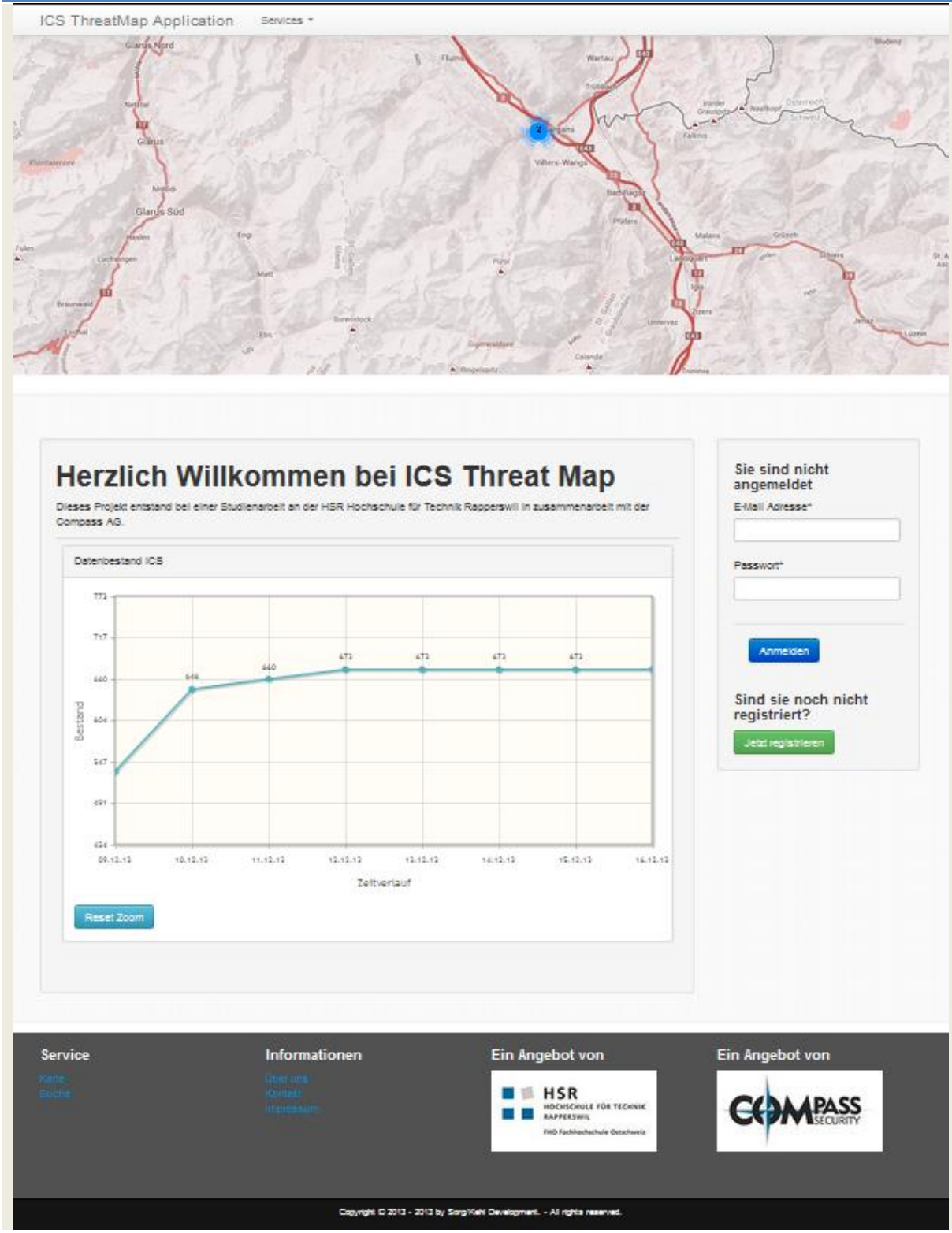
2.4.2.1 Chrome

Chrome V. 31



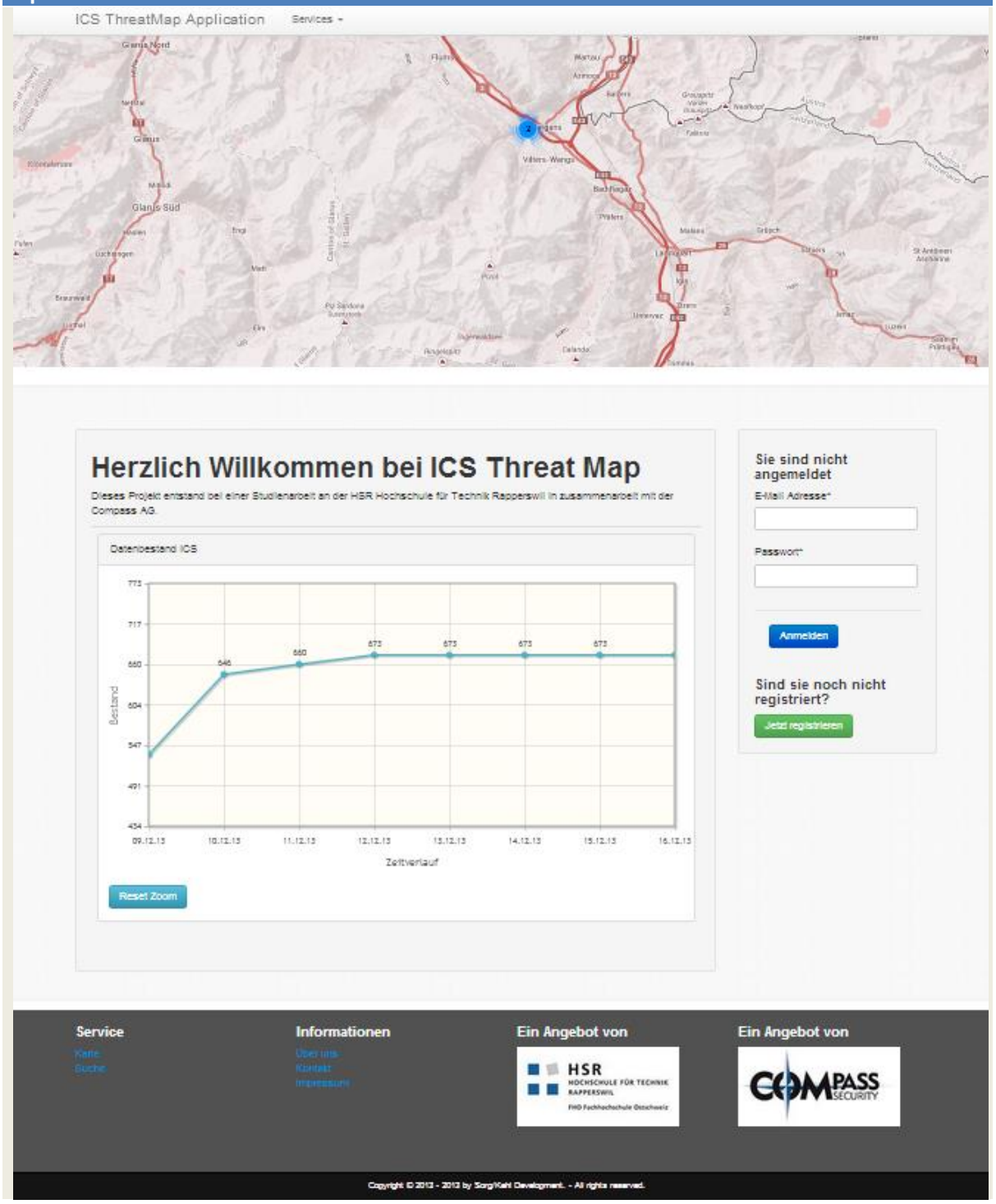
2.4.2.2 Firefox

Firefox 25.0.1



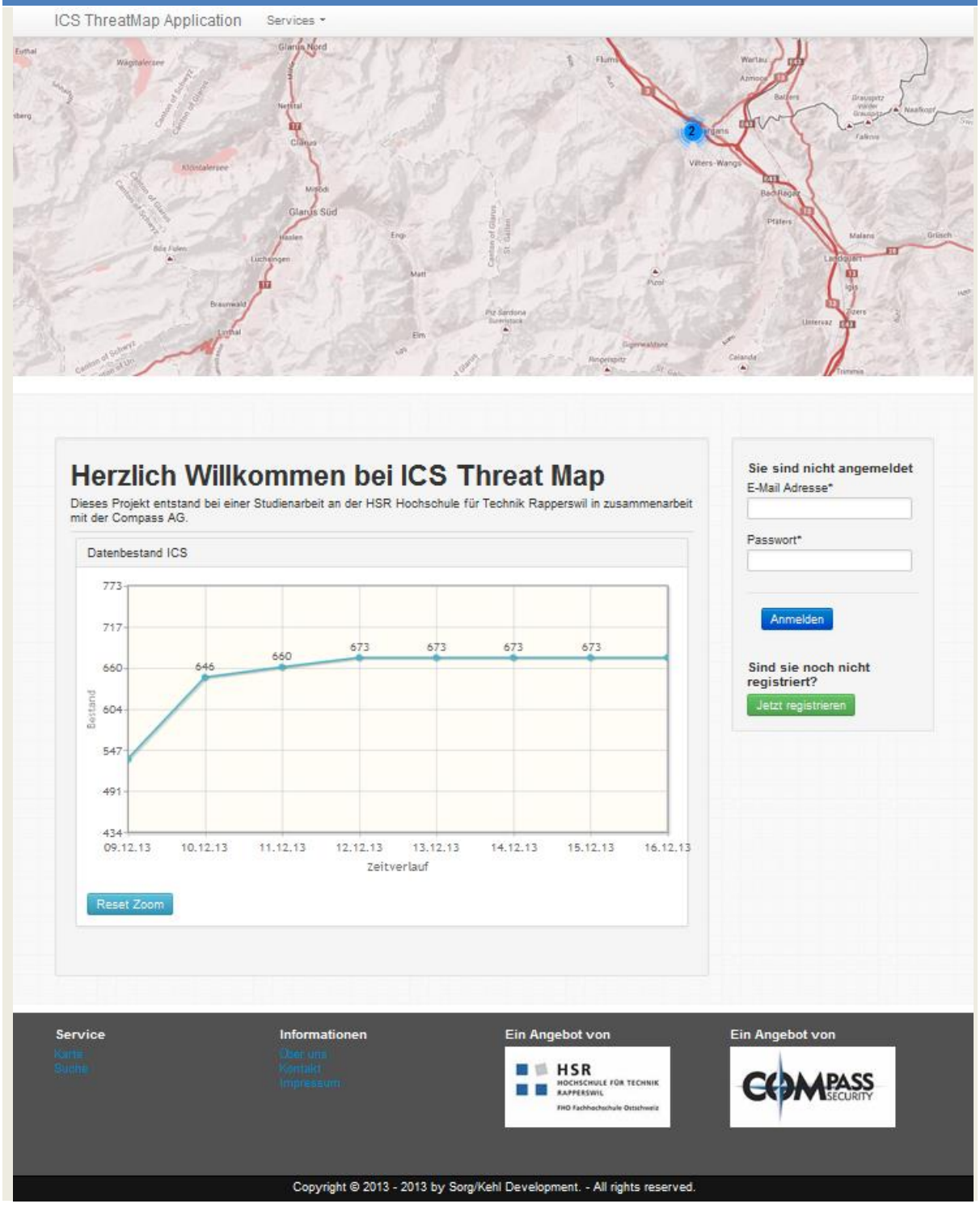
2.4.2.3 Opera

Opera 17.0

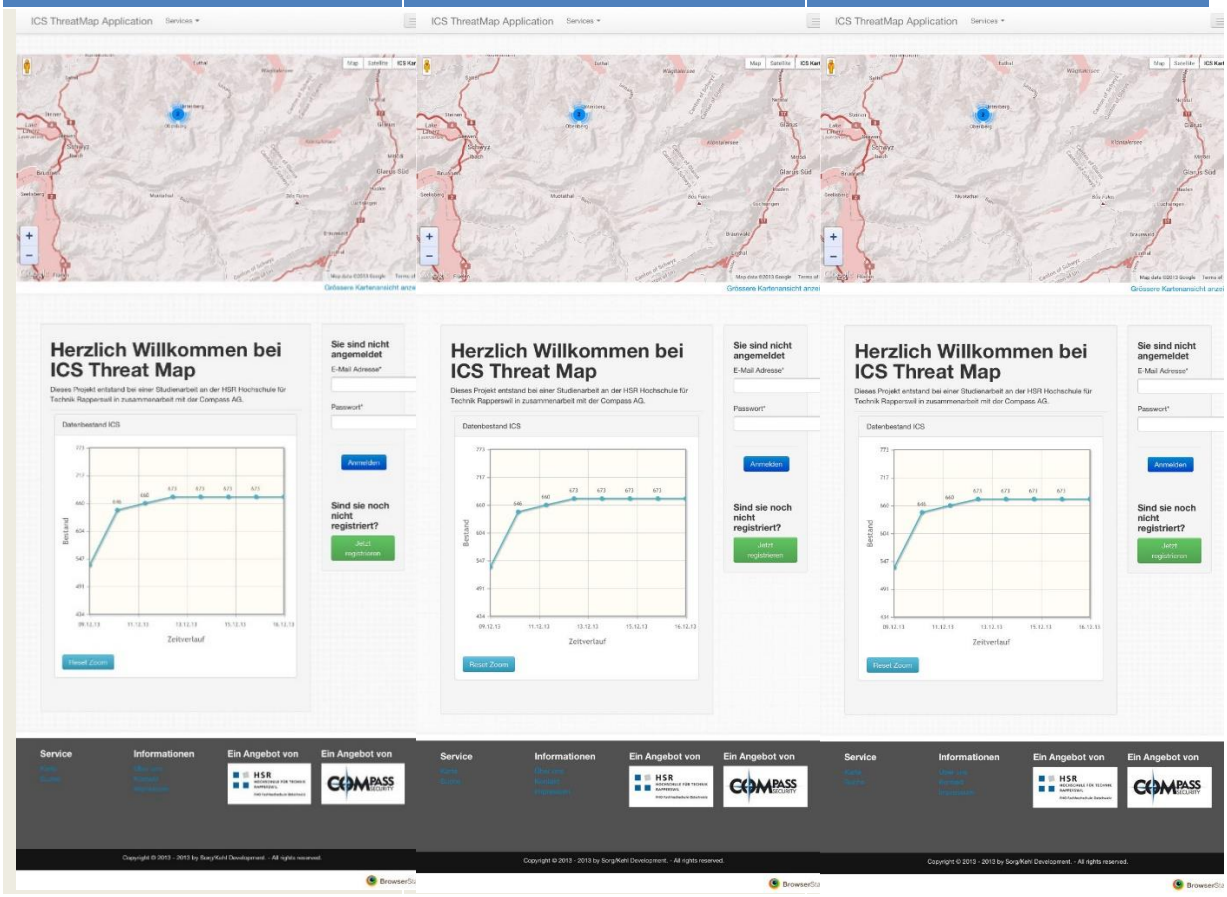


2.4.2.4 Safari

Safari 5.1.7



2.4.3 Tablet

IPad 2	IPad 3	IPad Mini
		

2.4.4 Mobile

IPhone 4S	IPhone 5	IPhone 5S	Android LG Nexus
			

2.5 Auswertung

2.5.1 Probleme

Da ICS ThreatMap v1 momentan ein selbstsigniertes Zertifikat besitzt, welches noch nicht anerkannt wurde, werden als Resultate häufig Screenshots wie im folgenden Bild zurückgeliefert.

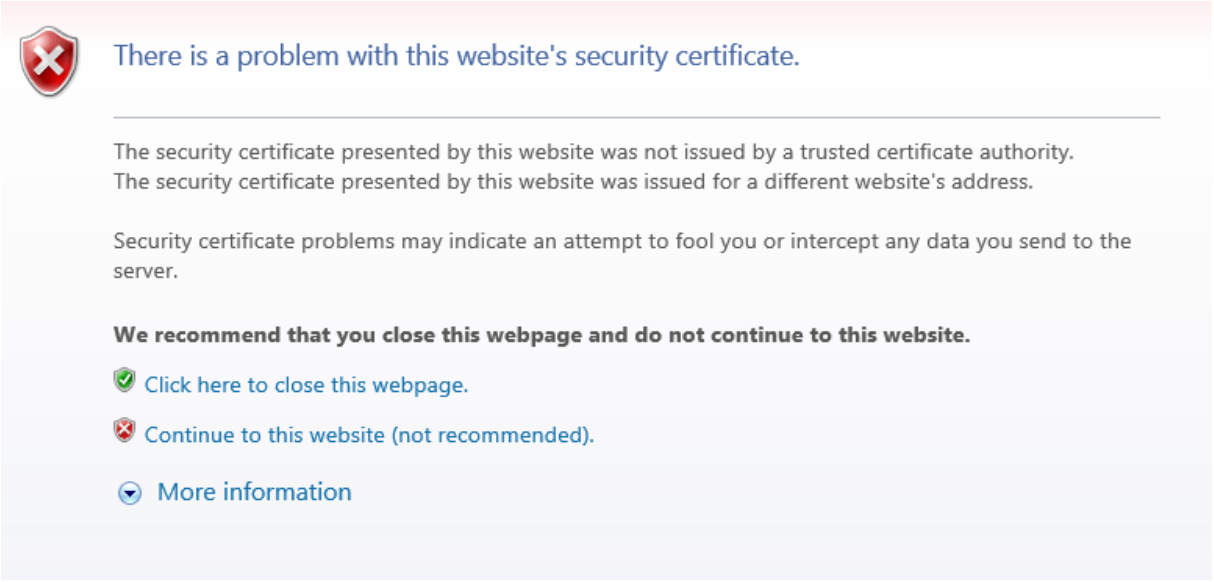


Abbildung 3 Internet Explorer 10: Nicht geprüftes Sicherheitszertifikat

Dennoch konnten einige wichtige Browser nach ihrer Darstellung getestet werden.

2.5.2 Fazit

Dank Bootstrap können die meisten Browser auf Computern die Elemente richtig darstellen. Einzige Ausnahme ist Internet Explorer 9, bei welchem im An-/Abmelde-Widget auf der Seite die Textelemente über den Rahmen fliessen. Der Grund an dieser Fehldarstellung liegt an dem fehlenden Responsive-Verhalten in Tabellen, welches mit Bootstrap in Version 2 noch nicht vorhanden ist.

Ein weiteres Problem ist die Ladezeit der Webseite. Wenn Google Maps länger zum Laden benötigt, als er eigentlich sollte, so kann es passieren, dass noch während dem Laden ein Screenshot erstellt wird.

Auch auf kleineren Displays wie auf Tablets oder sogar auf Smartphones kann ICS ThreatMap dargestellt werden. Leider konnte wegen des Problems, wie im Kapitel 2.5.1 beschrieben, praktisch nur Produkte von Apple getestet werden.

3. Usability Test

3.1 Ziel und Zweck

Dieses Dokument ist ein Teil des Qualitätsmanagements und wertet die getätigten Usability Tests aus. Ziel dieses Dokumentes ist es, anhand der Usability-Szenarien mögliche Stärken und Schwächen in Bezug auf die Benutzerfreundlichkeit der Webapplikation zu finden, indem ausgewählte Testpersonen diese Szenarien durchspielen müssen und ihre Meinungen abgeben. Dabei versuchen sie gewisse Fragen zu beantworten und eine Meinung abzugeben, ob sie Probleme bei der Durchführung hatten.

3.2 Testpersonen

Die Testpersonen konnten von Walter Sprenger frei gewählt werden. Sie müssen jedoch *unbelastet* sein. In anderen Worten dürfen Sie keine ähnliche Applikation (bspw. Konkurrenzproduktes) jemals getestet haben und sie dürfen nicht zu den Entwicklern von ICS ThreatMap gehören.

Es konnten zwei Usability Tests mit unserem Betreuer und von einem Mitarbeiter der Compass Security AG durchgeführt, der die Applikation zum ersten Mal gesehen hat. Die Testprotokolle befinden sich im Anhang.

3.3 Statistiken

Da wir die Tests nur mit zwei Testpersonen durchführen konnten, lohnen sich keine Statistiken. Im nächsten Kapitel werden die Schlussfolgerungen aus den Testergebnissen erläutert.

3.4 Schlussfolgerung aus den Testergebnisse

Aus den Testergebnissen ist ersichtlich, dass die Web Applikation im Bereich der Benutzerfreundlichkeit noch ausbaufähig ist. Auch die Karte könnte optimiert werden, damit die Daten schneller geladen werden können.

Oft sind es aber auch Erweiterungen, die aus Zeitgründen noch nicht realisiert werden konnten. Ein Beispiel wäre hier die Auflistung aller Tickets des Trouble Ticket System. Momentan ist es nur möglich zu einem ICS ein Ticket zu eröffnen.

Für die Testperson, die die Web Applikation zum ersten Mal bediente, stellten wir fest, dass es in einigen Bereichen noch Aufklärungen unsererseits benötigt werden. Diese könnten mit Beschreibungen und Legenden gemacht werden. Des Weiteren sollten Daten, die keine Werte enthalten, gar nicht erst angezeigt werden.

ICS ThreadMap - v1.0

Anhang

Dominique Sorg
Benjamin Kehl

1. ZEITUNGSARTIKEL - FAHRLÄSSIG DURCHLÄSSIG

2. IN BASEL LAG DAS SCHLIESSSYSTEM DES ST.-JAKOB-PARKS WÄHREND EINES JAHRES FÜR HACKER OFFEN

Foto: Freshfocus

2742 Steueranlagen stehen für Hacker offen, darunter selbst Systeme des St. Jakob-Stadions in Basel. Erstmals zeigt ein Test, wie es um die Sicherheit der Schweizer Infrastruktur steht und wie leicht es ist, sie zu manipuliere

Von Florian Imbach und Alexandre Haederli

38 500 Menschen fasst das grösste Stadion der Schweiz, der St.-Jakob-Park in Basel. Tausende besuchen das angeschlossene Shoppingparadies mit Läden wie Manor, C & A oder Kookai. Aber niemand dürfte ahnen, dass man das Schliess- und Kontrollsystem mit einer Lücke, die seit Monaten bekannt ist, übernehmen kann. Erst nach einem Hinweis der SonntagsZeitung hat der Betreiber das System am vergangenen Mittwoch vom Internet getrennt.

Zuvor konnte jeder das «Tür-Management-System M2010» direkt über einen Internetbrowser abrufen und dank einer Passwortlücke kontrollieren. Ein Vandale, ein Einbrecher oder gar ein Attentäter hätte den Personaleingang des Shoppingcenters in der Nacht öffnen und ungehindert unter das Stadion gelangen oder in die Läden einbrechen können. Er konnte Tür- und Sicherheitsalarme deaktivieren und die Fernalarmierung ausschalten. Er konnte aber auch einstellen, wann Zugangstüren offen stehen und wann sie geschlossen sind.

Ans Licht kam das massive Sicherheitsproblem im Rahmen des ersten systematischen Sicherheitschecks von Schweizer Industrieanlagen (siehe Box Seite 14). Durchgeführt hat ihn die italienische Sicherheitsfirma IS Group. Das Ergebnis: Mindestens 2742 Schweizer Anlagen sind gefährdet. Darunter auch Kleinkraftwerke, Kläranlagen und Produktionsbetriebe. Sie sind verwundbar, weil die Betreiber die Steuerung dieser Anlagen direkt ans Internet gehängt haben. Meist aus Bequemlichkeit, um die Anlagen aus der Ferne zu warten.

Laut Experte Francesco Ongaro von IS Group ist das brandgefährlich: «Damit löst der Betreiber das schwierigste Problem für einen Hacker: den direkten Zugriff auf die Anlage via Internet.» Hat der Hacker einmal Kontakt zu einer Steuerungsanlage, findet er meistens eine Lücke, um die Kontrolle über ein Kraftwerk oder eine Wärmepumpe zu übernehmen, denn viele Industriesteuerungen sind heutzutage veraltet und schlecht bis gar nicht geschützt. Beim Sicherheitstest fand die SonntagsZeitung in Zürich zum Beispiel drei Geräte zur Überwachung von Abwasser-

Pumpstationen. Diese Stationen überbrücken Höhenunterschiede im Abwassernetzwerk der Stadt Zürich. Ohne Passwort kamen die Journalisten mit einem Internetbrowser direkt auf die Überwachungsgeräte. Eine direkte Manipulation der Wasserpumpen sei über die Geräte nicht möglich, da die Steuerungen getrennt liefen, sagt die Betreiberin, die Entsorgung und Recycling Zürich (ERZ). Doch für einen Hacker ist dieses Gerät nur der erste Schritt. Ist er erst mal im internen Netzwerk, findet er in der Regel Zugang zu den Maschinen, in diesem Fall den Pumpen.

Auf den Hinweis, dass ihre Steuerung im Internet zugänglich ist, reagierte die ERZ überrascht. Die Geräte seien erst kürzlich ersetzt und offenbar nicht richtig konfiguriert worden. «Das sollte sicher nicht so funktionieren. Wir werden den Prozess beim Aufsetzen solcher Geräte überprüfen, damit das nicht mehr vorkommt.» Das Unternehmen hat die Geräte nach dem Hinweis vom Internet getrennt und neu konfiguriert.

Weniger einsichtig zeigte sich das Waadtländer Kantonsarchiv. In Lucens, unweit von Lausanne, unterhält der Kanton ein Lager für Kunstschätze. Meterdicke Betonwände und ein ausgeklügeltes Lüftungssystem schützen Tausende wertvoller Objekte, von historischen Büchern über Gemälde bis zu Tierpräparaten. Vor vier Jahren hat der Betreiber eine Steuerungsanlage direkt ans Internet gehängt. Wie in Basel kann man das System mit einem längst bekannten Trick übernehmen.

Wer will, kann die Luftfeuchtigkeit und Temperatur im Lager nach Belieben manipulieren. Betroffen seien nur Bücher, keine wertvollen Objekte, sagt das Kantonsarchiv und liess das System nach dem Hinweis erst mal weiterlaufen. Erst nach mehrmaligem Insistieren nahm der Kanton die Anlage vom Netz.

Der Sicherheitstest hat gezeigt, dass in der Schweiz vor allem mittelgrosse Anlagen schlecht geschützt sind. Bei manchen brauche es für den Zugriff nicht einmal einen Hacker, sagt Experte Ongaro: «Auf viele Anlagen kann selbst mein Cousin zugreifen.» Es fehle oft der simpelste Sicherheitsmechanismus. Zum Beispiel sollte ein Gerät nach einer bestimmten Anzahl von Zugriffversuchen den Zugang sperren. Eine Sicherung, die Ongaro fast nie antrifft. Einige Geräte sind gar ohne Passwort erreichbar.

In der Schweiz finden sich zum Beispiel Wärmeverbunde mit einer Leistung von mehreren Hundert Kilowatt, Fotovoltaikanlagen oder Klimasteuerungen, die jeder manipulieren kann. Eine Überlastung der Fernleitung? Das Ausbrennen eines Heizkessels? Oder die Verwandlung eines Grossraumbüros in eine Tropenzone? Mit wenigen Klicks erledigt. Will jemand dem Land Schaden zufügen, hat er damit Tausende Ziele. Experte Ongaro erklärt, wie das funktioniert: «Ein Hacker entwickelt beispielsweise einen Zugriff auf die zehn am häufigsten genutzten Geräte in der Schweiz. Danach kann er Hunderte Anlagen gleichzeitig angreifen.» Eine Attacke auf mehrere kleine Energieproduzenten kann sogar eine Kettenreaktion provozieren, die zu einem grossen Stromausfall führt.

Solche koordinierten Aktionen seien in der Schweiz bisher nicht bekannt geworden, sagt der Vizedirektor der Schweizer Fachstelle Melani, Max Klaus. Doch unsichere Industriesteuerungen sind eine grosse Sorge bei den Internetexperten des Bundes. Bei gewissen Herstellern spiele die Sicherheit eine untergeordnete Rolle, sagt Klaus. Ende Oktober publizierte die

Fachstelle eine Reihe von Massnahmen zum Schutz solcher Systeme. Doch bis heute gibt es keine Mindest-Sicherheitsstandards, die Hersteller oder Betreiber einhalten müssen. Experten wie der ETH-Professor Adrian Perrig schlagen deshalb eine gesetzliche Regelung vor (siehe Interview).

Ein Beispiel für mangelhafte Selbstkontrolle ist der Steuerungshersteller Saia-Burgess. Im Mai berichtete das Fachmagazin «ct» ausführlich über eine Sicherheitslücke bei Industriesteuersystemen des Schweizer Herstellers. Passiert ist bis heute offenbar wenig, denn immer
noch sind unsichere Systeme von Saia-Burgess am Netz. Eines ist
das Schliesssystem des St.-Jakob-Parks.

Saia-Burgess sagt, ihr System sei nicht dafür gedacht, direkt mit dem Internet verbunden zu werden. Ferner habe man die Kunden über die Sicherheitslücke informiert. Diese könnten mit einer Software-Aktualisierung das Passwortproblem beheben. Das hat man in Basel offenbar nicht gemerkt. Wer hier die Verantwortung für die Sicherheitslücke trägt, ist unklar. Der Stadionmanager Basel United verweist auf die Betreiberin ISS. Diese verweist auf die Eigentümerin Wincasa und der Installateur der Anlage auf den Kunden. Die Fragen der SonntagsZeitung beantwortete Wincasa nicht. Sie sagt: «Die Sicherheit ist uns sehr wichtig, so hat dieses Thema höchste Priorität.» Zum Sicherheitssystem selber und zu den einzelnen Massnahmen möchte sie «aus sicher verständlichen Gründen keine Auskunft geben». Wincasa habe das Problem erkannt und kläre, was für Massnahmen vorgenommen werden müssten.

Neugierige können ohne Passwort mithören

In einem ersten Gespräch vor Ort bestätigte die Betreiberin, dass es sich um das «Schliesssystem» handelt und damit die äusseren Zugangstüren gesteuert werden können, mit Zugang zum Einkaufszentrum, zum Stadion und Hochhaus. Eine Darstellung, die sich mit den Recherchen der SonntagsZeitung deckt. Zwei Tage später stellte sich Wincasa auf den Standpunkt, es handle sich um «kein gesamtheitliches Schliesssystem». Es gehe «ausschliesslich um eine Türe in der Aussenhülle des Shoppingcenters». Selbst wenn diese Aussage zutrifft, könnte ein Angreifer mit einem Klick das geschlossene Gebäude betreten.

ETH-Forscherin Myriam Dunn Cavelty sieht den Fall als «schwerwiegenden Sicherheitsverstoss». Dunn Cavelty beschäftigt sich seit Jahren mit Internetsicherheit und Cyberkriminalität. Die unklare Verantwortung sei typisch in solchen Fällen. «Es gibt keine klare Zuständigkeit. Man spielt sich gegenseitig den Ball zu, niemand will verantwortlich sein.»

Der Sicherheitstest zeigt, dass verantwortungsloser Umgang mit der IT in vielen Bereichen der Wirtschaft vorkommt. Eine Beratungsfirma aus Zürich etwa nutzt einen komplett ungeschützten Netzwerkspeicher, der jedem Internetnutzer offensteht. Dort finden sich Protokolle der Verwaltungsratssitzungen, detaillierte Lohnabrechnungen, Buchhaltung und Passwörter aller Mitarbeiter. Würde diese Lücke ausgenutzt, wäre die Firma innert Tagen ruiniert.

Eine renommierte Schweizer Luxusgüterfirma telefonierte, ohne es zu merken, während Monaten mit einem ungeschützten Telefonkonferenzsystem. Konkurrenten oder sonstige Neugierige können per Mausklick ohne Passwort mithören. In einem Fall stiessen die Journalisten auf einen Drucker der internationalen Fernmeldeunion in Genf, eine UNO-Organisation notabene. «Herzlich willkommen» stand da auf der Seite, die jedem Besucher offenbarte, wer gerade welche Dokumente druckte.

recherchedesk@sonntagszeitung.ch

Publiziert am 01.12.2013