



Aufgabenstellung Bachelorarbeit

„Orchestrierung von Security-Updates für Linux-Serversysteme“

FS 2016

1. Auftraggeber und Betreuer

- *Auftraggeber:* Nine Internet Solutions AG, Albisriederstrasse 243a, 8047 Zürich
- *Ansprechpartner Auftraggeber:* Hr. Samuel Sieg, samuel.sieg@nine.ch, +41 44 637 40 18
- *Betreuer:* Prof. Dr. Farhad Mehta

2. Studierende

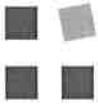
Diese Arbeit wird als Bachelorarbeit an der Abteilung Informatik durchgeführt von

- Hr. Philipp Christen
- Hr. Ueli Bosshard

3. Ausgangslage

Das Unternehmen nine.ch betreibt über 1500 Linux-Serversysteme mit unterschiedlichen Funktionen sowie Hardware- und Software-Ausstattungen. Die Installation der Software-Komponenten geschieht über herkömmliche Debian-Pakete. Für diese Pakete werden regelmässig Sicherheitsupdates veröffentlicht, welche zeitnah auf sämtlichen betroffenen Serversystemen installiert werden müssten.

Aktuell wird dies mit einer Sammlung von Bash-Scripts erreicht, welche manuell auf einzelnen Systemen aktiviert werden. Automatisches Auflösen von Abhängigkeiten oder zuverlässige Feedback-Mechanismen existieren nicht. Jede Woche verursacht dies mehrere Stunden Aufwand für einen Mitarbeiter. Zudem beträgt die Zeitspanne zwischen Veröffentlichung des Updates und der Installation auf einem System im schlimmsten Fall mehrere Tage.



4. Beschreibung der Aufgabe

Im Rahmen der Bachelorarbeit wird eine Software-Lösung entwickelt, welche das Einspielen der Updates auf diesen Serversystemen vereinfacht. Die Software bietet dem Systemadministrator die Möglichkeit, in Auftrag zu geben, auf welchen Systemen welche Updates durchgeführt werden sollen. Sie stellt zudem sicher, dass gegebene Regeln und Abhängigkeiten nicht verletzt werden. Solche Regeln können beispielsweise Abhängigkeiten von Systemen sein, etwa dass High-Availability-Cluster nicht alle gleichzeitig das gleiche Update einspielen und so gewisse Services nicht verfügbar sind oder dass bestimmte Updates zuerst auf Test-Umgebungen statt auf den produktiven Systemen installiert werden sollen.

Die Software zeigt darüber hinaus den aktuellen Zustand der Systeme an, d.h. sie zeigt, welche Updates wo ausstehend sind und welche Updates erfolgreich beziehungsweise fehlerhaft ausgeführt wurden.

Dabei müssen folgende Punkte beachtet werden:

1. Die Applikation kann über den Web-Browser bedient werden.
2. Die Anwendung soll es ermöglichen, anstehende Updates auf allen verbundenen Systemen einzusehen.
3. Der verfügbare Verlauf über installierte Updates soll ebenfalls ersichtlich sein.
4. Auf den Linux-Systemen, welche verwaltet werden soll, läuft ein Agent, welcher mit dem zentralen Managementsystem kommuniziert.
5. Die Kommunikation zwischen den Agents und dem Managementsystem soll verschlüsselt ablaufen.
6. Fehlermeldungen und benötigte Interaktionen für ein Update (z.B. Neustart eines Services oder des Systems) sollen dem Benutzer angezeigt und es sollen dafür Vorschläge angeboten werden.
7. Es wird ein detailliertes Logging darüber geführt, wer wann auf welchem System welche Updates freigegeben hat, bei Bedarf auch mit Kommentar/Grund.

Folgende nicht-funktionale Anforderungen sind besonders wichtig:

8. Zuverlässigkeit: Die Software hat jederzeit einen korrekten und konsistenten Datenstand. Die verschiedenen Szenarien sind mit Software-Tests geprüft.
9. Bedienbarkeit: Die Software ist für den Benutzer einfach zu bedienen und verständlich. Die Vorgänge sind nachvollziehbar.

Es gelten folgende Vorgaben und Rahmenbedingungen:

10. Die Software soll Open Source sein; es wird die MIT-Lizenz verwendet.
11. Es wird ausschliesslich Ubuntu (Versionen 12.04 und 14.04) als Server-Betriebssystem unterstützt.
12. Als Programmiersprache muss, wenn möglich, Ruby verwendet werden. Für die systemnahe Komponente (Agent) kann alternativ auch Python oder Go eingesetzt werden.
13. Als Datenbankserver wird PostgreSQL verwendet.

Die folgenden Punkte sind optional:



14. Die Agents sollen paketiert und über Puppet auf den Systemen verteilt werden können.
15. Die Applikation soll für die Gruppenbildung der Systeme eigene Vorschläge anbieten, z.B. aufgrund von Systemeigenschaften wie OS-Version, installierte Software, etc.

Die folgenden Punkte befinden sich nicht im Scope des Projekts und sind somit nicht Ziele der Bachelorarbeit:

16. Abhängigkeiten unter den Paketen auflösen - dies geschieht durch den Paketmanager des Betriebssystems.
17. Automatisches Durchführen von Updates aufgrund von Regeln - der Auslöser eines Updatevorganges ist immer ein menschlicher Benutzer.
18. Bereits installierte Updates können über die Applikation nicht entfernt werden, es wird kein Rollback angeboten.
19. Verfügbarkeit des Services - das Hosting findet bei nine.ch statt.

Die Deliverables umfassen zwei Komponenten:

20. Der Agent, welcher auf jedem einzelnen angeschlossenen System läuft und via apt die anstehenden Updates ausliest und ausführt.
21. Das Control-Center, welches auf einem zentralen Server läuft und wo Benutzer Informationen auslesen sowie Updates starten können.

5. Zur Durchführung

Mit dem Betreuer finden wöchentliche Besprechungen statt. Besprechungen mit dem Auftraggeber sind von den Studierenden nach Bedarf zu initialisieren.

Alle Besprechungen sind von den Studierenden mit einer Traktandenliste vorzubereiten, die Besprechung ist durch die Studierenden zu leiten und die Ergebnisse sind in einem Protokoll festzuhalten, das den Betreuern und dem Auftraggeber per E-Mail zugestellt wird.

Für die Durchführung der Arbeit ist ein Projektplan zu erstellen. Dabei ist auf einen kontinuierlichen und sichtbaren Arbeitsfortschritt zu achten. An Meilensteinen gemäss Projektplan sind einzelne Arbeitsresultate in vorläufigen Versionen abzugeben. Über die abgegebenen Arbeitsresultate erhalten die Studierenden ein vorläufiges Feedback. Eine definitive Beurteilung erfolgt auf Grund der am Abgabetermin abgelieferten Dokumentation.

6. Dokumentation

Über diese Arbeit ist eine Dokumentation gemäss den Richtlinien der Abteilung Informatik zu verfassen (siehe <https://www.hsr.ch/Allgemeine-Infos-Diplom-Bach.4418.0.html>). Die zu erstellenden



Dokumente sind im Projektplan festzuhalten. Alle Dokumente sind nachzuführen, d.h. sie sollten den Stand der Arbeit bei der Abgabe in konsistenter Form dokumentieren. Alle Resultate sind vollständig auf CD/DVD in 3 Exemplaren abzugeben. Der Bericht ist ausgedruckt in doppelter Ausführung abzugeben.

7. Termine

Siehe Terminplan auf <https://www.hsr.ch/Termine-Diplom-Bachelor-und.5142.0.html>.

8. Arbeitsumfang

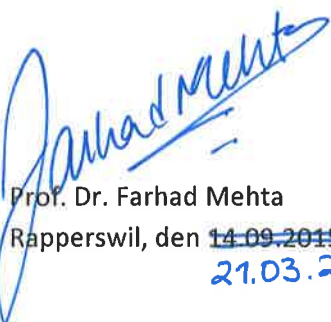
Eine erfolgreiche Bachelorarbeit zählt 12 ECTS-Punkte pro Studierenden. Für 1 ECTS Punkt ist eine Arbeitsleistung von 30 Stunden budgetiert. Die verwendete Arbeitszeit muss erfasst und dokumentiert werden.

9. Beurteilung

Für die Beurteilung ist der HSR-Betreuer verantwortlich. Die Benotung erfolgt gemäss folgender Tabelle.

Gesichtspunkt	Gewicht
1. Organisation, Durchführung	1/6
2. Berichte	1/6
3. Inhalt	3/6
4. Mündliche Prüfung zur Bachelorarbeit	1/6

Im Übrigen gelten die Bestimmungen der Abteilung Informatik für Bachelorarbeiten (siehe <https://www.hsr.ch/Ablaeufe-und-Regelungen-Studie.7479.0.html>).



Prof. Dr. Farhad Mehta

Rapperswil, den ~~14.09.2015~~

21.03.2016