



# Secrets Management in großen Firmenumgebungen

**Bericht Praxis I  
T1000**

des Studiengangs Informationstechnik (B.Sc.)  
an der Dualen Hochschule Baden-Württemberg Stuttgart

von  
**Steffen Walter**

03.09.2018

<b>Bearbeitungszeitraum</b>	9 Wochen
<b>Matrikelnummer, Kurs</b>	1145690, TINF17IN
<b>Ausbildungsunternehmen</b>	science + computing ag, Tübingen
<b>Betreuer des Ausbildungsunternehmens</b>	Dr. Marcus Camen

## **Erklärung**

Ich versichere hiermit, dass ich meine Studienarbeit mit dem Thema:  
“Secrets Management in großen Firmenumgebungen“ selbstständig verfasst  
und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten  
Fassung übereinstimmt.

.....

<i>Ort</i>	<i>Datum</i>	<i>Unterschrift</i>
------------	--------------	---------------------

## **Zusammenfassung**

Das Thema Secrets Management wird zunehmend zu einem zentralen Thema in der elektronischen Datenverarbeitung. Unter dem Begriff ist im Folgenden vor allem der Umgang mit geheimen Informationen gemeint. Geheim sind Informationen dann, sobald es schädlich ist wenn diese Informationen Unbefugten zugänglich sind. Beispiele für derartige Informationen sind Passwörter, geheime Schlüssel und geheime Dokumente. In der Praxisarbeit soll evaluiert werden welche Anforderungen große Unternehmen an ihr Secrets Management stellen und welche Programme dabei häufig zum Einsatz kommen. In einem weiteren Schritt soll festgestellt werden welche Anforderungen durch jene Programme nicht erfüllt werden. Im Anschluss soll eine voll umfängliche Secrets Management Software daraufhin untersucht werden, in wie fern Unzulänglichkeiten von gängiger Software behoben werden können. Es soll auch beleuchtet werden, welche zusätzlichen Anforderungen Cloud Umgebungen mit sich bringen. Hierbei zu beachten ist, welche Vorteile durch alternative Software zu erzielen sind. Für die Evaluierung soll eine virtuelle Umgebung erstellt werden, in welcher die Funktionen getestet werden können.

## **Abstract**

The topic of secrets management is becoming an uprising matter in digital data processing. In this report the term of secrets management is mainly used to describe the handling of confidential information. Information is considered confidential, if it can be used to harm the owner of the secret in any way. Examples for information that meets this criteria would be password, private digital key or simply confidential documents. In the paper shall be evaluated which requirements enterprises have for their secrets management and which software is currently used to try and fulfill those needs. Subsequently the gaps between requirements and the functional range of the used Software shall be emphasized. Furthermore a secrets management software with a modern approach shall be examined to find out whether or not this software is able to fill in the gaps of traditional software. Also the aspect of cloud environment shall be considered as a factor of importance. To achieve this task a virtual environment shall be implemented to test the features of the chosen software.

---

# Inhaltsverzeichnis

Abkürzungsverzeichnis	i
Abbildungsverzeichnis	ii
Tabellenverzeichnis	ii
Glossar	iii
<b>1 Einleitung</b>	<b>1</b>
1.0.1 Direkte Kosten . . . . .	2
1.0.2 Indirekte Kosten . . . . .	2
1.0.3 Versuch einer Quantifizierung . . . . .	3
1.0.4 Anforderung an Sicherheitssoftware . . . . .	3
<b>2 Stand der Technik</b>	<b>5</b>
2.1 Das Passwort . . . . .	5
2.2 Identitäten . . . . .	7
2.3 Digitale Zertifikate . . . . .	8
2.4 Tokenbasierte Authentifizierung und Autorisierung . . . . .	9
2.5 Cloud Computing . . . . .	9
2.6 Vergleichende Aufstellung unterschiedlicher Softwarepro- dukte . . . . .	12
2.6.1 Kerberos . . . . .	12
2.6.2 Pleasant Password Server . . . . .	15
2.6.3 Hashicorp Vault . . . . .	16
2.7 Motivation . . . . .	18
2.8 Projektumgebung . . . . .	19
2.8.1 Unternehmensvorstellung . . . . .	20
2.8.2 Arbeitsumgebung . . . . .	21
2.8.3 Unternehmensspezifische Anforderungen . . . . .	21
<b>3 Evaluierung von Hashicorp Vault</b>	<b>23</b>
3.1 Anforderungen . . . . .	25
3.2 Planung der Testumgebung . . . . .	26
3.3 Aufbau Testumgebung . . . . .	27
3.3.1 Installation Vault . . . . .	27
3.3.2 Installation von OpenLDAP . . . . .	30

---

3.4	Test der Funktionen . . . . .	31
3.4.1	OpenLDAP Anbindung . . . . .	31
3.4.2	PKI Integration . . . . .	34
3.4.3	Weitere Funktionen . . . . .	37
<b>4</b>	<b>Fazit und Ausblick</b>	<b>39</b>
	<b>Literatur</b>	<b>40</b>

## Abkürzungsverzeichnis

<b>ACL</b>	Zugriffskontrollliste (engl. Access Control List)
<b>AD</b>	Active Directory
<b>API</b>	Schnittstelle zur Anwendungsprogrammierung (engl. Application Programming Interface)
<b>AWS</b>	Amazon Web Dienste (engl. Amazon Web Services)
<b>CA</b>	Zertifizierungsstelle (engl. Certificate Authority)
<b>CAE</b>	Rechnergestütztes Entwicklung (engl. Computer Aided Engineering)
<b>HCL</b>	Hashicorp Konfigurations Sprache (engl. Hashicorp Configuration Language)
<b>IT</b>	Informationstechnik
<b>JSON</b>	Java Script Objekt Notation
<b>JWT</b>	Java Script Objekt Notation (JSON) Web Token
<b>LDAP</b>	Leichtgewichtiges Verzeichniszugriffsprotokoll (engl. Lightweight Directory Access Protocol)
<b>MIT</b>	Massachusetts Institute of Technology
<b>OECD</b>	Organisation für wirtschaftliche Zusammenarbeit und Entwicklung (engl. Organisation for Economic Co-operation and Development)
<b>OIDC</b>	OpenID Connect
<b>PKI</b>	Public Key Infrastruktur
<b>RADIUS</b>	Authentifizierungsdienst für sich einwählende Benutzer (engl. Remote Authentication Dial-IN User Service)
<b>SSL</b>	Spezifikation zur Verschlüsselten Datenübertragung (engl. Secure Socket Layer)
<b>TLS</b>	Transportschicht Sicherheit (engl. Transport Layer Security)
<b>URL</b>	einheitlicher Ressourcenzeiger (engl. Uniform Resource Locator)

## Abbildungsverzeichnis

1	Kerberos . . . . .	14
2	Barbican . . . . .	24
3	Netzplan . . . . .	26

## Tabellenverzeichnis

1	Softwarevergleich . . . . .	12
---	-----------------------------	----

## Listings

1	Installation Vault . . . . .	28
2	Initialisierung Vault . . . . .	29
3	Vault Service Datei . . . . .	29
4	Installation Docker . . . . .	30
5	Beispiel HCL . . . . .	32
6	LDAP Authentifizierung . . . . .	32
7	Skript PKI . . . . .	35
8	Entsiegelungsprozess . . . . .	37



## Glossar

**Authentifizierung:** der Identitätsnachweis einer Person, Maschine oder eines Dienstes, gegenüber einer weiteren Instanz

**Autorisierung:** die explizite Freigabe um auf einen geheimen Inhalt zuzugreifen

**Chef:** eine Software zur automatisierten Installation von fertig konfigurierten Betriebssystemabbildern.

**Computercluter:** eine Ansammlung von, meist identischer, Hardware, welche zu einer logischen Einheit zusammengefasst wird. Der Cluster wird meist verwendet um komplexe Berechnungen zu lösen, die auf einzelnen Computern sehr lange dauern würde.

**Dienst:** eine autarke Einheit, welche eine spezifizierte Aufgabe bzw. Funktionalitaet erfuehlt und diese ueber keine klar definierte Schnittstelle zur Verfuegung stellt

**KeePass Passwort Safe:** ein quelloffener Passwortmanager, dazu designed wurde viele Passörter auf sichere Art und Weise zu speichern. Der Zugang zu diesem Passwortsafe wird allein durch ein einziges Masterpasswort gewährt.

**OpenStack:** der Markenname eines quelloffenen Betriebssystems welches entwickelt wurde um eine Cloud Computing (siehe Kapitel 2.5 auf Seite 9) Infrastruktur bereit zu stellen. [3, S. 2]

**Skalierbarkeit:** die Fähigkeit eines Systems zur flexiblen Änderung ihrer Größe. Der Begriff wird meist dann verwendet wenn es um ein Ausweitung des gefragten Systems geht.

## 1 Einleitung

Mit der fortschreitenden Digitalisierung nahezu aller Wirtschaftszweige steigt auch die Relevanz für die Absicherung der daraus resultierenden Informationstechnik (IT)-Infrastrukturen. Zusätzlich zu den eigenen Sicherheitsbelangen des Betreibers einer informationstechnischen Umgebung kommen noch gesetzliche Regelungen wie zum Beispiel das IT-Sicherheitsgesetz zum Tragen. Vor allem im Bezug auf eine zunehmende Verlagerung des IT-Betriebs hin zum Cloud Computing und den damit verbundenen Problemen dezentraler Datenhaltung (im speziellen bei den public und hybrid Modellen, siehe Kapitel 2.5 auf Seite 9), entstehen häufig unübersichtliche Sicherheitskonzepte. Durch die große Varianz der Szenarien, bei unterschiedlichen Sicherheitsniveaus der Daten, sind die verwendeten Konzepte sehr unterschiedlich. [1, S. 7f]

Spionage- und Sabotage-Angriffe werden aus den unterschiedlichsten Motivationen und auch von den unterschiedlichsten Subjekten verübt. So reicht das Spektrum der Angreifer von Kleinkriminellen über Geheimdienste und Terroristen bis hin zur organisierten Kriminalität. Die Aufgabe der IT-Sicherheit besteht darin, die potentiellen Angreifer in ihren Erwägungen zu berücksichtigen und die Werte und Geheimnisse der Unternehmen zu schützen. Ein zentraler Faktor bei der Entwicklung eines Sicherheitskonzepts auf dieser Grundlage ist es, ein stringentes Konzept zur Kontrolle der Autorisierung einer Person oder eines Dienstes beim Zugriff auf die verschiedenen Bereiche in der betreuten Umgebung. Neben der Autorisierung spielt auch die Authentifizierung eine wichtige Rolle, denn es muss zu jedem Zeitpunkt sichergestellt werden, dass die Autorisierung auch der richtigen Person oder Anwendung übertragen wurde. [1, S. 9]

Der Aufwand welcher für IT-Sicherheitskonzepte betrieben wird bemisst sich meistens am Schaden, der zu erwarten ist, sollten geheime Daten in die Hände Dritter gelangen. Da es kaum verlässliche Daten zur Quantität

der Kosten gibt, die durch einen kritischen Sicherheitsvorfall verursacht werden können, ist die daran orientierte Bemessung der Sicherheitsvorkehrungen umstritten. Generell gibt es verschiedene Faktoren die bei der qualitativen Kostenabschätzung berücksichtigt werden müssen, so wird im allgemeinen zwischen direkten und indirekten Kosten unterschieden. [4, S. 12]

### **1.0.1 Direkte Kosten**

Die direkten Kosten die durch Wirtschaftsspionage entstehen können, bemessen sich zu allererste einmal an dem direkten Wert des gestohlenen Eigentums. Dieser Wert lässt sich ermitteln durch den finanziellen Gegenwert, den das Eigentum hat und an den zu erwartenden Gewinneinbußen durch den Verlust (der Exklusivität) des Eigentums. Weiterhin entstehen direkte Kosten durch das Disaster-Recovery, das heißt durch die Schritte die eingeleitet werden müssen um den Status Quo von vor dem Angriff wieder herzustellen. Zu guter Letzt werden noch diejenigen Kosten hinzugezählt, die durch die Prävention einer Wiederholung des Ereignisses entstehen, dazu zählen Prozessänderungen, Sicherheitsunterweisungen und ähnliche direkte Maßnahmen. [4, S. 13]

### **1.0.2 Indirekte Kosten**

Zu den indirekten Kosten werden die Umsatzausfälle durch Image- und Markenschäden gezählt. Außerdem können hohe Umsatzeinbußen durch Plagiate entstehen, welche in Folge des Gestohlenen Eigentums verbreitet werden. Plagiate können deutlich günstiger angeboten werden, da die Kosten für Forschung und Entwicklung nicht in den Preis eingerechnet werden müssen. [4, S. 14]

### **1.0.3 Versuch einer Quantifizierung**

Nach Schätzungen der Organisation für wirtschaftliche Zusammenarbeit und Entwicklung (engl. Organisation for Economic Co-operation and Development, OECD) belaufen sich die Schäden durch Fälschungen und Produktpiraterie weltweit auf 638 Milliarden US-Dollar pro Jahr. Die Schätzungen diesbezüglich gehen aber weit auseinander. Es scheint jedoch sicher zu sein, dass sich die Schäden im dreistelligen Milliardenbereich bewegen. Für die deutsche Wirtschaft liegen die Schätzungen zwischen 20 und 50 Milliarden Euro. [4, S. 16f]

### **1.0.4 Anforderung an Sicherheitssoftware**

Spezifische Anforderungen die an eine Software zum Secrets Management gestellt werden, können wie folgt zusammengefasst werden:

- Auffindbarkeit - Es muss zu jedem Zeitpunkt klar sein, wo sich geheime Informationen im Unternehmen befinden.
- Nachvollziehbarkeit - Es muss möglich sein, Verantwortliche zu nennen.
- Break Glass Szenario - Es muss einen Weg geben, im Fall eines Angriffs den Schaden einzugrenzen.
- Verfügbarkeit - Daten müssen ständig zugänglich sein.
- Integrität - Daten müssen vollständig und korrekt sein.
- Verlässlichkeit - Daten müssen authentisch sein, Kommunikationswege nachvollziehbar.
- Autorisierung - Der Zugriff auf Daten soll nur dann möglich sein, wenn die betroffene Person/Maschine berechtigt ist (minimale Berechtigungsvergabe).

- Benutzerfreundlichkeit - Einfacher Zugriff von Mensch und Maschine.
- Authentifizierung - Sichere Anmeldeverfahren müssen zur Verfügung stehen.
- Credential Management Systeme zur sicheren Erstellung, Speicherung und Änderung sollen zur Verfügung stehen.
- Das Risiko, dass kompromittierte Verschlüsselung zum Verlust oder dem unberechtigten Offenlegen von Informationen führt, muss kontrollierbar sein.

Rechtliche Aspekte werden durch den Vorliegenden Bericht nicht weiter beleuchtet.<sup>1</sup>

---

<sup>1</sup>Die Inhalte aus Kapitel 1.0.4 auf der vorherigen Seite wurden in Anlehnung an interne Dokument eines großen Unternehmens und Gesprächen mit IT-Administratoren herausgearbeitet

## 2 Stand der Technik

Das folgende Kapitel liefert die theoretischen Grundlagen für die praktische Umsetzung der Projektarbeit. Es wird eine Breite Übersicht über den Themenkomplex des Secrets Management gegeben und es wird gezeigt welche Softwareprodukte aktuell in Unternehmen zum Einsatz kommen um Probleme die mit den Themenkomplexen Authentifizierung und Autorisierung in Verbindung stehen zu lösen. Außerdem wird aufgezeigt welche neuen Herausforderungen das Arbeiten unter Verwendung von Cloud Computing mit sich bringt.

### 2.1 Das Passwort

Das Passwort ist die einfachste und am häufigsten verwendete Methode um eine Authentifizierung zwischen einem Mensch und einem Computersystem durchzuführen. Es ist leicht zu implementieren und zu bedienen. Es gibt allerdings eine Reihe von Angriffsszenarien bei denen das klassische Passwort als authentifizierender Faktor versagt.

- Das Passwort kann beim Eintippen durch einen Dritten gesehen oder gefilmt werden.
- Passwörter können über sogenannte Brute-force Angriffe “erraten“ werden. Um diese Angriffe durchzuführen gibt es spezielle Programme, die so lange alle möglichen Kombinationen von Tastatureingaben ausprobieren, bis sie das Passwort herausgefunden haben.
- Wenn Passwörter über ein Netzwerk übertragen werden, können sie durch spezielle Software abgegriffen werden. Außerdem gibt es spezielle Software, welche die Tastatureingaben am Computer protokollieren kann. Solche Software kann dazu verwendet werden Passwörter die an betreffenden Endgeräten eingegeben werden an einen Angreifer zu übermitteln.

- Ein weiteres Angriffsszenario stellt das sogenannte Login Spoofing dar. Hierbei wird ein Anmeldefenster gefälscht, welches möglichst gleich aussieht wie das original. Wenn der Nutzer sein Passwort in das gefälschte Fenster eingibt, wird es gespeichert oder direkt an den Angreifer übermittelt.

Jeder dieser Angriffe reicht aus, um Systeme zu täuschen die zur Authentifizierung allein auf Passwörter setzen. In den meisten Unternehmen ist es aus diesen Gründen nicht erlaubt das gleiche Passwort für unterschiedliche Dienste zu verwenden. Außerdem gibt es versuche durch Passwortrichtlinien die Komplexität der Passwörter zu steigern, sodass zum Beispiel Brute-force Angriffe länger dauern und sich nicht mehr lohnen. Zusätzlich werden Schwellwerte für fehlgeschlagene Anmeldeversuche festgelegt um zu vermeiden, dass Passörter beliebig oft ausprobiert werden können. Diese Maßnahmen verringern zwar die Zahl der erfolgreichen Angriffe, stehen jedoch im Widerspruch zur einfachen Benutzbarkeit. Ein hoher Aufwand für die IT-Abteilungen muss aufgebracht werden um Passwörter zurückzusetzen und die Sicherheitsmechanismen für jeden Dienst zu implementieren. [19, S. 3ff]

Auf Grund der vielen Angriffsmöglichkeiten gibt es Stimmen, die das Ende des Passworts, wie wir es kennen, fordern. Es gibt außerdem viele Empfehlungen wie ein Passwort am besten zu erzeugen ist um eine Steigerung der Sicherheit gegenüber Angriffen zu erreichen. Was bleibt ist die menschliche Komponente beim Umgang mit Passwörtern. Da Menschen sich aus Bequemlichkeit nicht an die Empfehlungen zur Erstellung eines "sicheren" Passworts halten und dies sich auch nur bedingt prüfen lässt, ohne neue Angriffsflächen zu bieten, bleibt das Problem bestehen, dass Passwörter "geknackt" werden und Angreifer Zugriff auf sensible Daten bekommen. Es gibt mittlerweile einen Trend zur sogenannten Zwei-Faktor Authentifizierung wobei diese setzt beim Identitätsnachweis nicht allein auf das Passwort, es wird zusätzlich ein zweiter Nachweis einge-

fordert. Es gibt unterschiedliche Methoden wie solch ein weiterer Faktor aussehen kann, so gibt es zum Beispiel Codes die über die Mobiltelefone der Benutzer zugeschickt werden und die nur einmal und für eine kurze Zeitperiode verwendet werden können. Da es bis jetzt aber noch keine vollkommene Alternative zu Passwörtern gibt, ist es sehr wahrscheinlich, dass das Passwort noch einige Zeit als Teil des Authentifizierungsprozesses bestehen bleibt. [2]

## 2.2 Identitäten

Vorwiegend wird der Begriff der Identität in der Soziologie besprochen, er spielt allerdings auch in der Informatik eine wichtige Rolle. Identitäten sind definierte Zusammensetzungen von Rollen und Eigenschaften eines Objekts. Zu dieser Kombination kommt ein, innerhalb einer Organisationseinheit, eindeutiger Identifikator. Durch ihre Beschreibung kann abgeleitet werden wie die Identität vorzugehen hat und welche Schnittstellen für sie von Relevanz sind. Zudem wird davon ausgegangen, dass eine Identität einer gewissen Persistenz unterliegt, das heißt, dass sich Eigenschaften und Rollen nicht ständig ändern. Genauso wie es möglich ist, dass mehrere Objekte die gleiche Zusammensetzung von Eigenschaften haben, kann es ebenso vorkommen, dass einzelne Objekte mehreren Identitäten zugeordnet werden. Für Menschen die sich selten im Kontext der IT Administration bewegen mag der Gedanke Nahe liegen, dass es sich bei einer Identität um eine natürliche Person handelt. Dieser Schluss ist zwar nicht falsch, greift aber zu kurz, denn nicht ausschließlich natürliche Personen erfüllen die Voraussetzungen die eine Identität ausmachen. Neben natürlichen Personen werden auch IT-Systeme und IT-Anwendungen durch den Begriff der Identität beschrieben. [25, S. 21ff]



## 2.3 Digitale Zertifikate

Neben der in Kapitel 2.1 auf Seite 5 beschriebenen Authentifizierungsmethode via Passwort gibt es weitere Möglichkeiten die Integrität einer Identität gegenüber einem informationstechnischen System nachzuweisen. Eine wichtige Rolle spielen hierbei sogenannte digitale Zertifikate. Diese Zertifikate können dazu verwendet werden den öffentlichen Schlüssel eines asymmetrischen Verschlüsselungssystems, einer Identität zuzuordnen. Der öffentliche Schlüssel wird zu diesem Zweck mit mehreren Merkmalen verknüpft, die gesammelt dazu geeignet sind den Eigentümer des Schlüssels eindeutig zu authentifizieren. Zur Validierung eines Zertifikats wird eine sogenannte Zertifizierungstelle (engl. Certificate Authority, CA) verwendet; diese übernimmt die Aufgabe der Verknüpfung der zu authentifizierenden Identität und dem öffentlichen Schlüssel.

Grundlage für eine erfolgreiche Authentifizierung mit digitalen Zertifikaten ist das Vertrauen gegenüber der CA. Jedes Zertifikat enthält eine Seriennummer die von der CA nur einmal vergeben wird, damit kann der Halter des Schlüssels eindeutig identifizieren werden. Die bekannteste Implementierung einer derartigen Public Key Infrastruktur (PKI) liegt der Standard des X.509 Zertifikat zugrunde, welcher zum Beispiel bei der Absicherung von Netzwerkkommunikation mit Transportschicht Sicherheit (engl. Transport Layer Security, TLS) (häufig als Spezifikation zur Verschlüsselten Datenübertragung (engl. Secure Socket Layer, SSL) bezeichnet) eingesetzt wird. Neben der Seriennummer enthalten digitale Zertifikate Informationen wie Versionsnummer, die Identität, die Gültigkeitsdauer, die digitale Signatur der validierenden CA, Informationen zum verwendeten Verschlüsselungsalgorithmus und zum Gültigkeitsbereich sowie des vorgesehenen Anwendungsbereichs des kryptographischen Schlüssels. [25, S. 144]

Das Zertifikat an sich enthält also ausschließlich öffentliche Informatio-

nen und kann somit nicht allein zur Authentisierung verwendet werden. Allein der Besitz des privaten Gegenstücks zum öffentlichen Schlüssel, der im Zertifikat seine Zuordnung erhält, eignet sich zum eindeutigen Identitätsnachweis. Der kritische Punkt ist der private Schlüssel, welcher unbedingt vor dem Zugriff dritter geschützt werden muss, denn andernfalls kann ein unrechtmäßiger Besitzer das Zertifikat verwenden um sich für den tatsächlichen Halter des Zertifikats auszugeben und damit beispielsweise an geheime Informationen gelangen. Typischerweise sind derartige Zertifikate ein bis zwei Jahre gültig. [25, S. 145f]

## **2.4 Tokenbasierte Authentifizierung und Autorisierung**

Ein Token wird verwendet um einen Zugangsschlüssel zu erzeugen welcher sich dazu eignet sich gegenüber einem geschützten System zu authentifizieren. Beim Token handelt es sich um ein Stück Hardware das nicht selten über ein LCD-Display verfügt um dort den generierten Zugangsschlüssel auszugeben. Umgangssprachlich wird auch der erzeugte Zugangsschlüssel in der IT oft als Token bezeichnet zur Unterscheidung wie im Folgenden von Hardware- bzw. Software-Token gesprochen. Wenn nur von Token die Rede ist sind in dieser Arbeit Software-Token gemeint. [25, S.141ff]

## **2.5 Cloud Computing**

Der Begriff des Cloud Computing beschreibt den bedarfsorientierten Zugriff auf Internetdienste und andere IT Ressourcen die durch den Provider Schlüsselfertig zur Verfügung gestellt werden und dynamisch an den Bedarf des Kunden angepasst werden können. Bei der Bereitstellung von Cloud Services wird zwischen folgenden Charakteristika unterschieden [6, S. 8]:

**Bedarfsgerechter Zugriff:** Der Bedarf des Kunden kann in Echtzeit an die aktuellen Bedürfnisse angepasst werden und findet insbesondere ohne menschliche Interaktion auf Seite des Providers statt. Parameter die unter diese Art von Anpassungen fallen sind Rechenleistung, Speichergröße und Bandbreitenkapazität. [6, S. 8]

**Netzwerkanbindung:** Viele Cloud Provider ermöglichen den Zugriff auf ihre Dienste von sehr unterschiedlichen Endgeräten. Eine Steuerung über Smartphone oder Tablet ist dabei keine Seltenheit. Die Netzwerkanbindung ist durch eine Breitbandverbindung realisiert und verfügt über definierte Schnittstellen. [6, S. 8]

**Ressourcenbündelung:** Durch gesetzliche Vorschriften und eigenen Sicherheitsbeschränkungen sind einige Nutzer verpflichtet bestimmte Daten auf Servern in festgelegten geografischen Regionen zu lagern. Dieses Vorgehen widerspricht dem Prinzip der Ressourcenbündelung, die sieht nämlich vor, dass die Bereitstellung der gewünschten Ressource, durch dasjenige Rechenzentrum geschieht, welches diesem Zeitpunkt am wenigsten Auslastung zu verzeichnen hat. Einige Cloud Provider bieten daher die Möglichkeit der regionalen Eingrenzung an. [6, S. 8]

**Skalierbarkeit:** Wenn die vorangegangenen Punkte kombiniert werden, ergibt sich daraus die Möglichkeit flexibel und bedarfsgerecht Ressourcen zu erhöhen und wieder freizugeben. In vielen Fällen erfolgt dieser Prozess vollautomatisiert und nimmt dem Nutzer in diesem Bereich jegliche nicht-monetäre Ressourcenplanung ab. [6, S. 9]

**Verbrauchsabhängige Bezahlung:** Die Bezahlung von Cloud Diensten orientiert sich nicht selten an den tatsächlich verbrauchten Ressourcen in Relation zur zeitlichen Belegung. So können relativ einfach Kalkulationen angestellt werden, aus denen sich ergibt ob sich die Nutzung von Cloud Diensten, für einen definierten Einsatzzweck, lohnt. [6, S. 9]

**Dynamische Zertifizierung:** das in Kapitel 2.3 auf Seite 8 beschriebene Verfahren der Nutzung von digitalen Zertifikaten in seiner klassischen Form, stößt beim Betrieb von Cloud Anwendungen an seine Grenzen und wird daher durch neue Konzepte wie zum Beispiel die dynamische Zertifizierung ersetzt. Klassische Methoden zur Zertifizierung liegt die Annahme zu Grunde, dass Identitäten über längere Zeit eine konstante Zusammensetzung von Rollen und Eigenschaften haben. Falls nicht manuell getriggert, wird nach der definierten Gültigkeitsperiode eines Zertifikats geprüft ob das Objekt, weiterhin die Voraussetzungen für den Zertifizierungsprozess erfüllt, um bei erfolgreicher Prüfung eine Rezertifizierung einzuleiten. [5]

Durch die Schnellebigkeit von Cloud-Diensten kommt es häufig vor, dass sich die Eigenschaften eines Dienstes im laufenden Cloudbetrieb ändern. Durch den Aufbau von Cloudumgebungen sind diese Änderungen für den Endnutzer allerdings nicht erkennbar und ein Zertifikat, welches nach den klassischen Prozessen erzeugt wurde lässt darauf schließen, dass der Dienst für die Gültigkeitsdauer des Zertifikats weiterhin die Ursprünglichen Eigenschaften aufweist. Dynamische Zertifizierung versucht dieses Problem zu lösen indem eine ständige Kontrolle der Zertifikatsanforderungen von Cloud-Diensten zur Laufzeit durchgeführt wird, um Änderungen in den Eigenschaften direkt protokollieren zu können. Um ein kontinuierliches Audit der Dienste realisieren zu können, werden verschiedene Ansätze verfolgt. Neben spezieller Test- und Monitoringsoftware, die eine Anwendung zur Laufzeit überprüft, können auch sogenannte Trusted Platform Module verwendet werden. Diese Module erzeugen dann eine Prüfsumme über ein gewünschte Ziel (zum Beispiel die zu kontrollierende Cloud-Anwendung) und vergleichen sie dann mit dem zuvor gespeicherten historischen Wert der Prüfsumme. Um solche Auditierungsvorgänge in bestehende Zertifizierungsmodelle zu integrieren, werden häufig zusätzlich ausgebildete Auditoren beschäftigt, die An-

derungen in Dokumenten und bei den beteiligten Identitäten feststellen sollen. [6, S. 114ff]

## 2.6 Vergleichende Aufstellung unterschiedlicher Softwareprodukte

Um die in Kapitel 1 auf Seite 1 beschriebenen Themenkomplexe der Autorisierung und Authentifizierung in Firmenumgebungen umzusetzen gibt es verschiedene Ansätze. Im folgenden sollen einige gängige Softwareprodukte, die die genannten Aufgaben erfüllen sollen vorgestellt und auf ihre Tauglichkeit zur voll umfänglichen Erfüllung der in Kapitel 1.0.4 auf Seite 3 beschriebenen Anforderung untersucht. Die Programme wer-

<b>Funktion</b> Näheres: Kapitel 1.0.4 auf Seite 3	<b>Kerberos</b>	<b>Pleasant Password Server</b> (Multi-User KeePass)	<b>Hashicorp Vault</b>
Auffindbarkeit	✓	✓	✓
Nachvollziehbarkeit	×	✓	✓
Break Glass Szenario	×	×	✓
Verfügbarkeit	✓	✓	✓
Integrität	✓	✓	✓
Verlässlichkeit	✓	?	✓
Autorisierung	×	✓	✓
Authentifizierung	✓	✓	✓
Credential Management	✓	✓	✓
Kontrollierbarkeit	×	✓	✓

Tabelle 1: Vergleich verschiedener Softwareprodukte [9, S. 4f, 13, 57 ] [15] [12]

den in unterschiedlichen Versionen angeboten. Einige der beschriebenen Funktionen stehen möglicherweise in der kostenfreien Version nicht zur Verfügung.

### 2.6.1 Kerberos

Der Name Kerberos kommt ursprünglich aus der griechischen Mythologie, wo er für den dreiköpfigen Hund verwendet wurde, der den Eingang

zur Unterwelt bewacht. Das Projekt wurde in den achtziger Jahren am Massachusetts Institute of Technology (MIT) als Teil des sogenannten Athena-Projekts ins Leben gerufen. Die großen Stärken von Kerberos liegen in der Authentifizierung und der verschlüsselten Nachrichtenübermittlung. Schon zu Beginn des Projekts lag, neben den Sicherheitsaspekten, der Fokus auf der Skalierbarkeit des Systems. Verschlüsselt wird mit symmetrischer Verschlüsselung und einem vorher vereinbarten geheimen Schlüssel. Kerberos unterstützt auch den Einsatz von Passwörtern, ist aber ursprünglich nicht dafür entwickelt worden. Mit diesen Grundfunktionen kann eine relative sichere und authentifizierte Kommunikation zwischen Nutzer und Dienst in einem umfangreichen Netzwerk gewährleistet werden.

Durch die Tatsache, dass die Software unter einer Open-Source-Lizenz veröffentlicht wurde, können Sicherheitslücken schnell aufgespürt und behoben werden. Außerdem ist es dadurch möglich, dass unterschiedliche Unternehmen an dem Code mitarbeiten und diesen ständig verbessern. Die Weiterentwicklung wird unter anderem durch Microsoft vorangetrieben. Kerberos kommt heutzutage selten als Einzelsystem vor, es wird entweder mit anderen Softwarekomponenten kombiniert, oder kommt als Bestandteil einer umfangreicheren Software wie zum Beispiel Microsoft Active Directory (AD) vor. [25, S. 137f]

Die Authentifizierung mit Kerberos wird über sogenannte Tickets organisiert. Der Client fordert ein Ticket beim Authentifizierungsserver an. Der Authentifizierungsserver verschlüsselt anschließend das Ticket mit dem Schlüssel des Clients und dem Schlüssel des gewünschten Dienstes. Mit diesem Ticket kann der Client sich nun mit dem gewünschten Dienst verbinden und es ist gewährleistet, dass sowohl der Dienst, als auch der Client diejenigen sind für die sie sich ausgeben. Der Zugrundeliegende Ablauf wird in Abbildung 1 auf der nächsten Seite nochmal genauer und anschaulicher gezeigt. Bei der gegenseitigen Authentifizierung ist es unerheblich auf welchem Betriebssystem Client und Server basieren. Weil

dies der Fall ist kann bei Kerberos von einer plattformübergreifenden Software gesprochen werden. Kritisch im Zusammenhang mit Kerberos

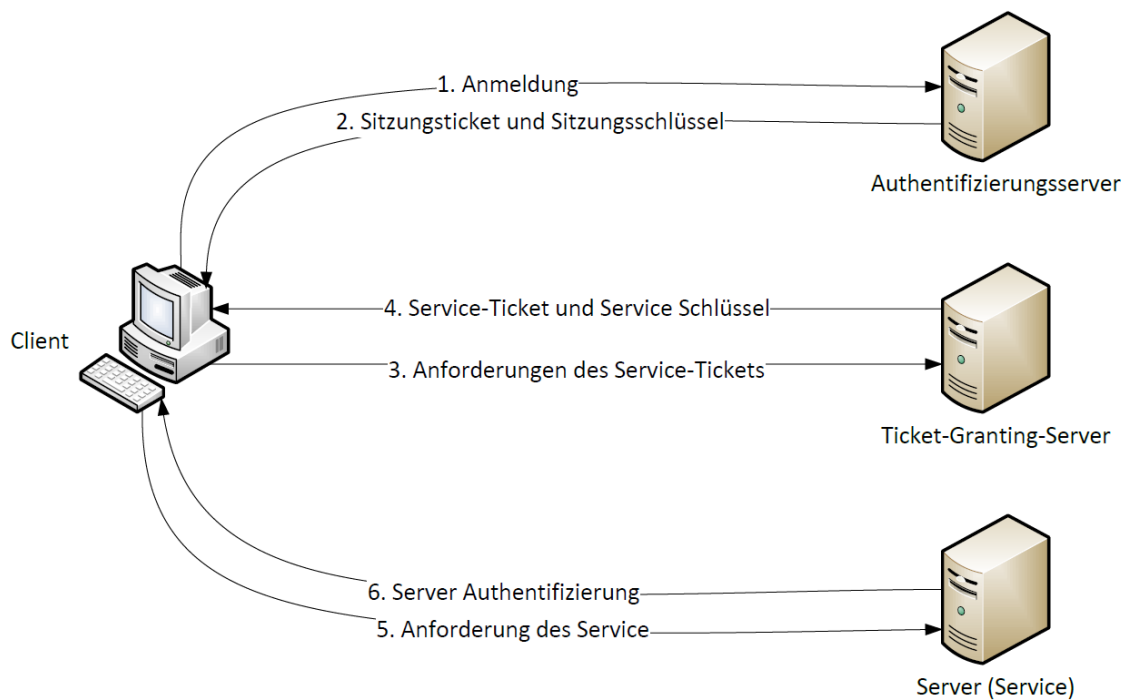


Abbildung 1: Authentifizierung mit Kerberos [25, vgl. S.140]

ist die Fokussierung auf einen Zentralen Server, dieser muss so dimensioniert sein, dass er alle Anfragen verarbeiten kann und, dass durch einen Hardwaredefekt der Betrieb nicht stoppt. Ein weiterer Nachteil ist die fehlende Verschlüsselung des Netzwerkverkehrs. Außerdem werden Sitzungsschlüssel zwischenzeitlich auf dem Client gespeichert und sind so ein leichtes Ziel für Angreifer die Zugriff auf die Clientmaschine haben. Es kann ebenfalls kritisiert werden, dass die Änderung eines Nutzerpassworts mit der Änderung des geheimen Schlüssels einhergeht. Kerberos erfüllt seine Aufgaben sehr zuverlässig, was durch seine Popularität untermauert wird. Mit diversen Erweiterungen lassen sich die erwähnten Unzulänglichkeiten auch umgehen. [25, vgl. S.138f]

### 2.6.2 Pleasant Password Server

Beim Pleasant Password Server handelt es sich nach eigenen Angaben um einen serverbasierten KeePass Passwort Safe. Im Unterschied zu seinem kleinen Bruder ist der Pleasant Password Server allerdings, wie der Name schon vermuten lässt, keine reine Client Software mehr, sondern er fügt sich als Serverdienst in IT-Umgebungen ein. Das Ziel der Anwendung ist es eine sichere Ablagemöglichkeit für Passwörter, Produkt-Keys, Kreditkarteninformationen und Dateien zu bieten. Zur verschlüsselten Netzwerkkommunikation werden TLS-Zertifikate verwendet. Die Client-Software wird als installierbares Programm zur Verfügung gestellt welches auf einigen gängigen Betriebssystemen installiert werden kann. Die Serveranwendung hingegen wird ausschließlich für Microsoft Windows zur Verfügung gestellt. Zusätzlich zur normalen Client-Software wird eine Webanwendung mitgeliefert, die clientseitig plattformunabhängige Nutzung ermöglicht. [15]

Eine Rollen- und Eintragsbasierte Rechtevergabe ist vorgesehen, womit der Zugriff durch ausschließlich autorisierte Identitäten sichergestellt werden soll. Mit seiner Initialen Veröffentlichung im Oktober 2012 handelt es sich beim Pleasant Password Server noch um eine relativ junge Software, die jedoch durch den Hersteller, dem Versionsstand nach zu urteilen, regelmäßig mit Updates versorgt wird. [15]

Wie aus Tabelle 1 auf Seite 12 hervorgeht, erfüllt die Software schon relativ viele der festgelegten Kriterien. Da es sich jedoch nicht um ein quelloffenes Programm handelt, gibt es wenige Informationen über die internen Mechanismen und deren Sicherheit. Aus dem Lizenzmodell geht zudem hervor, dass sich die Software eher an kleine bis mittelständige Unternehmen richtet.



### 2.6.3 Hashicorp Vault

Bei Vault handelt es sich dem Hersteller Hashicorp zufolge um ein vollumfängliches Secrets Management System. Die Software wird in der Community Edition unter einer quelloffenen Lizenz veröffentlicht und verfügt über eine umfangreiche Dokumentation. Folgende Funktionen und Funktionsweisen liefert Vault dem Hersteller zufolge:

#### **Funktionen von Vault:**

Willkürliche Verbindungen von Identifikator und Wert können auf “sichere“ Art und Weise durch Vault abgelegt werden. Dabei werden die Inhalte verschlüsselt bevor sie in einen persistenten Speicher geschrieben werden. Für eine zunehmende Anzahl an Diensten kann Vault dynamische Zugangsdaten generieren. Wenn zum Beispiel ein Dienst Zugriff auf eine Datenbank erhalten will, kann Vault einen (zeitlich Beschränkten) Zugriff gewähren. Dabei werden temporäre Zugangsdaten (oder ein Schlüsselpaar) erstellt, welche durch Vault nach Ablauf der Gültigkeit widerrufen werden. Dateien mit sensiblen Inhalte, können durch Vault verschlüsselt werden, ohne dass sie durch Vault im eigenen Backend gespeichert werden müssen. Entwickler sind damit in der Lage Daten durch Vault verschlüsseln zu lassen, um sie im Anschluss nach Belieben weiterverarbeiten zu können.

Alle Geheimnisse welche durch Vault gespeichert werden haben eine Gültigkeitsdauer. Nach Ablauf der zugeordneten Gültigkeitsdauer werden die betroffenen Geheimnisse durch Vault widerrufen. Clients können durch eine Schnittstellen zu Anwendungsprogrammierung (API) die Gültigkeit eines Geheimnisses verlängern bzw. erneuern. Geheimnisse können auch durch Administratoren widerrufen werden. Dabei bietet Vault die Funktion, dass bei Bedarf ganze Baumstrukturen an Geheimnissen auf einmal ihre Gültigkeit verlieren. Es kann auf unterschiedliche Weisen gefiltert werden, so können zum Beispiel alle Geheimnisse

widerrufen werden auf die ein spezieller Benutzer zugegriffen hat.

### **Die Komponenten von Vault:**

*Storage Backend:* Vault benötigt ein Storage Backend um verschlüsselte Daten abzulegen. Die einzige Anforderung an das Storage Backend ist, dass es möglichst strapazierbar ist. Vault vertraut dem Storage Backend nicht und es wird nicht davon ausgegangen, dass es speziell gegen fremde Zugriffe geschützt ist.

*Barriere:* Zwischen Storage und Vault wird die Kommunikation durch eine Art Kontrollpunkt geprüft. Durch diesen Mechanismus soll sichergestellt werden, dass alle Daten welche von Vault in Richtung Storage Backend übermittelt werden, zwangsläufig verschlüsselt werden. Außerdem ist der Mechanismus dafür zuständig alle Daten die aus dem Storage Backend gelesen werden zu verifizieren und zu entschlüsseln.

*Secrets Speicher:* Der Secrets Speicher ist dafür Zuständig auf Anfrage ein angefordertes Geheimnis preiszugeben. Bei einigen Systemen ist der Prozess statisch organisiert. Das bedeutet, dass bei der gleichen Anfrage die gleiche Rückgabe zu erwarten ist. Andere Systeme arbeiten hier komplexer, sie sind dazu in der Lage dynamische Zugangsdaten zu erzeugen. Diese Möglichkeit schafft eine zusätzliche Sicherheitsebene. Das Feature steht allerdings nicht für jede Anwendung zur Verfügung.

*Client Software-Token:* Ein Client Software-Token wird ausgestellt, um einen Client über die Dauer einer Sitzung gegenüber Vault zu authentisieren.

*Server:* Vault wird als einzelne Binärdatei zur Verfügung gestellt, sie kann sowohl als Client als auch als Server ausgeführt werden. Wenn der Server gestartet wurde, kümmert er sich um die Kommunikation mit Hintergrunddiensten und stellt ein Schnittstelle zur Anwendungsprogrammierung (engl. Application Programming Interface, API) für die Clientinteraktion bereit. Außerdem ist er verantwortlich für die Anwen-

dung der Zugriffskontrollliste (engl. Access Control List, ACL) und den Widerruf abgelaufener Geheimnisse. Neben einigen weiteren Aufgaben erstellt der Server auch ein Log in welchem Interaktionen mit Vault dokumentiert wird.

## 2.7 Motivation

In den vorigen Abschnitten wurden Problematiken und Widersprüche aufgezeigt, die zur Wahl des Themas für die vorliegende Arbeit geführt haben. Der Diskurs um Sicherheits- und Secrets- Management in großen Firmenumgebungen und der Trend hin zur Auslagerung eigener Rechenzentren zu Cloud-Providern, wirft die Frage nach funktionierenden Sicherheitskonzepten auf. Nicht zu Letzt, die zunehmenden erfolgreichen Angriffe, die auf ein unzureichendes Sicherheitskonzept zurückzuführen sind werfen die Frage nach einem umfangreichen Sicherheitssystem auf, welches dazu in der Lage ist neue Bedrohungen und sich verändernde Bedingungen abzudecken.

Eine der wichtigsten Herausforderungen bei Cloudumgebungen stellt nicht allein das Management von Zugangsdaten von Benutzern dar, es wird zunehmend wichtiger auch die Verwaltung von Diensten und deren Zugangsdaten zu beleuchten. Klassischerweise geht es dabei zum Beispiel um Datenbankpasswörter, es kommen aber immer mehr sogenannte Microservices dazu. Bei Microservices handelt es sich um kleine Programme, die meist eine einzige Aufgabe erfüllen und über eine API steuerbar sind. Wenn im Netzwerk viele Microservices aktiv sind, die nicht unwahrscheinlich auch untereinander kommunizieren und sich Authentifizieren müssen, verliert ein Administrator leicht die Übersicht.

Auf Grund der Problematiken die in Kapitel 2.1 auf Seite 5 aufgezeigt wurden ist es also essentiell auch hier für einen möglichst sicheren Ablauf zu sorgen um zu erreichen, dass es für Angreifer sehr schwer wird

gefälschte Dienste einzuschleusen um an geheime Informationen zu kommen, bzw. bestehende Dienst für diesen Zweck zu missbrauchen.<sup>2</sup>

## 2.8 Projektumgebung

Das Thema Sicherheit in der Datenverarbeitung ist seit der Einführung eben jener ein wichtiges Thema. Über die Jahre haben sich die Möglichkeiten zur Absicherung in der IT stetig weiterentwickelt, sodass die Absicherung von Computern, Netzwerken, Servern und ganz allgemein Informationen zu einem wichtigen Zweig der Branche geworden ist. Mit neuen Technologien stellt sich auch die Frage nach den Sicherheitsaspekten.

Cloud Computing ist aktuell ein sehr wichtiger und schnell wachsendes Faktor in der IT-Branche, im Ausland werden Technologien rund um das Thema Cloud stark gefördert und es gibt eine Vielzahl an Softwareprodukten die sich diesem Feld verpflichtet haben. Für die Nutzung der Vorteile dieser Technologien in Deutschland gibt es einige Faktoren die es zu betrachten gilt. Zum Beispiel gilt es zu kalkulieren, ob eine Verlagerung von Anwendungen oder Daten auf Cloud Infrastruktur wirtschaftlich ist oder nicht. Dabei spielt die vorhandene Infrastruktur eine Zentrale Rolle und es berücksichtigt werden ob ein schrittweiser oder teilweiser Umzug das Mittel der Wahl sein könnte. Vor diesem Hintergrund spielt das Thema Sicherheit eine wichtige Rolle. Daten auf "fremder" Hardware zu speichern, wie es in Cloud Umgebungen gängige Praxis ist, stellt ein Risiko dar. Es muss darauf geachtet werden, dass kritische Daten und Anwendungen nur autorisierten Identitäten zur Verfügung gestellt werden. Weiterhin ist abzuwägen in wie fern sich die aktuelle Software, welche zur Authentifizierung und Autorisierung eingesetzt wird, dazu geeignet ist die genannten Herausforderungen zu bewältigen. In dieser Arbeit sollen Aspekte wie dieser genauer beleuchtet werden.

---

<sup>2</sup>Eigene Darstellung zur Basis der vorangegangenen Abschnitte

### 2.8.1 Unternehmensvorstellung

Die science + computing ag, kurz s+c, ist als Tochtergesellschaft der ATOS SE ein Unternehmen des IT-Dienstleistungs- und Consultingbereichs. Mit seinen knapp 300 Mitarbeitern betätigt sich die s+c hauptsächlich in den Bereichen High Performance Computing, IT-Sicherheit, 3D-Virtualisierung und System-Management. Durch OpenSoftware-Dienstleistungen und diverse Softwareprodukte betätigt sich das Unternehmen zudem im Bereich der Softwareentwicklung. Neben Großkunden der Automobilindustrie besteht der diverse Kundenkreis der s+c aus Mikroelektronikherstellern, Chemie- und Pharmaunternehmen, Maschinen- und Anlagenbauern, Forschungs- und Bildungseinrichtungen sowie Unternehmen aus der Luft- und Raumfahrtbranche.

Zum Hauptsitz in Tübingen kommen noch vier weitere Standorte in Berlin, Düsseldorf, Ingolstadt und München hinzu. Bis zur Übernahme durch den französischen Computerhersteller Bull war die 1989 gegründete science + computing ag ein eigenständiges Unternehmen. [23] Im Jahr 2014 wurde die BULL-Gruppe ihrerseits durch die ATOS SE übernommen wodurch auch die science + computing ag nun zum ATOS-Gesamtkonzern gehört. [13]

ATOS SE ist ein International agierender Konzern mit knapp 100.000 Mitarbeitern, einem jährlichen Umsatz von 14 Milliarden Dollar und Hauptsitz in Bezons (Frankreich). Nach der Forbes Liste der 2000 größten Unternehmen der Welt belegt ATOS SE mit den oben genannten Werten den Platz 858 (Stand Juni 2018). [10] Im Zuge der Übernahme durch ATOS SE wird die Marke s+c seit 2016 nicht mehr weitergeführt. Stattdessen wird nun die Konzernmarke ATOS verwendet.

### 2.8.2 Arbeitsumgebung

Die Durchführung der Projektarbeit findet im Team Rechnergestütztes Entwicklung (engl. Computer Aided Engineering, CAE)<sup>3</sup> der science + computing ag statt. Im Team CAE<sup>3</sup> werden Kunden betreut, die unter Verwendung von High Performance Computing Berechnungen anstellen. Der Fokus des Teams liegt dabei auf Kundenkreisen die sich im Bereich CAE betätigen. Im Team gibt es verschiedene Kompetenzen um die volle Bandbreite der Kundenwünsche abdecken zu können. Zu den Kernkompetenzen gehören hierbei Weiter-/Entwicklung von systemunterstützender Software, Administration von Computerclustern inklusive Überwachung der Systemkomponenten sowie Administration und Bereitstellung von speziellen Speicherinfrastrukturen.

Die Evaluierung welche Teil dieses Praxisberichts ist, soll erste Informationen liefern aus denen sich die Nutzbarkeit von Secrets Management Software zur interne Nutzung ableiten lässt. Der praktische Teil der Evaluierung wird auf einer virtualisierten Umgebung auf Basis von VMware Workstation durchgeführt.

### 2.8.3 Unternehmensspezifische Anforderungen

Bei s+c gibt es laufend Projekte, die sich mit zukunftsweisenden Technologien beschäftigen, um eine Übernahme in das Leistungsportfolio der Firma zu erörtern. Bei den Projekten geht es sowohl um die Verbesserung eigener Geschäftsprozesse und Arbeitsabläufe als auch um die Optimierung der Leistungen gegenüber dem Kunden. Wichtige Faktoren die bei einer derartigen Einschätzung betrachtet werden müssen sind Einsatzmöglichkeiten, Einsatzbereiche, Kosten, Schnittstellen und Skalierbarkeit.

Über die Einsatzmöglichkeiten kann entschieden werden, wenn klar ist, welche Voraussetzungen die Secrets Management Software zum Betrieb

benötigt und ob sich der Einsatz mit rechtlichen und betriebsrätlichen Beschränkungen vereinbaren lässt. Dem Trend zu einheitlichen Arbeitsmitteln folgend ist der Einsatzbereich so weit wie möglich zu wählen, ohne dabei jedoch die Einzelnen Bereiche in ihrer Arbeit zu beeinträchtigen. Nach erfolgreicher Evaluierung kann ein produktives Pilotprojekt in einem definierten Umfeld gestartet werden. Zum Zweck der Evaluierung sollte eine Testversion oder die kostenfreie Version des Secrets Management Systems eingesetzt werden.

Weiterhin sollte eine Kalkulation angestellt werden, welche eine qualifizierte Aussage über die Wirtschaftlichkeit der kostenpflichtigen Version gegenüber der kostenfreien Version macht. Dabei ist zu beachten, inwiefern interne Kompetenzen zum Betrieb der kostenfreien Version verwendet werden können, welche den kommerziellen Support ersetzen könnten. Weiterhin ist ein finanzieller Vergleich vom Status Quo zum geplanten Ziel über einen definierten Zeitraum anzustellen. Finanzielle Erwägungen sind jedoch nicht Teil der vorliegenden Arbeit und werden nur der Vollständigkeit halber erwähnt.

Um qualifizierte Aussagen über den Aufwand treffen zu können ist es wichtig die Schnittstellen zu bestehenden Systemen zu bewerten und auf ihre Funktionalität zu untersuchen. Das Secrets Management Konzept ist derart anzulegen, dass es sich möglichst einfach in eine bestehende Umgebung integrieren lässt. Bei der Evaluierung muss auch darauf geachtet werden inwiefern sich eine Secrets Management Software in großen Umgebungen einsetzen lässt. Dabei ist es von Vorteil wenn die Software über die Möglichkeit verfügt über mehrere Hardwareinstanzen hinaus erweiterbar zu sein. Außerdem ist zu betrachten ob das System auch rechenzentrenübergreifend eingesetzt werden kann. Letzteres ist vor allem bei sehr großen Umgebungen von Relevanz.

### 3 Evaluierung von Hashicorp Vault

Auf Grund von teaminternen Erwägungen wird als Secrets Management Software Hashicorp Vault evaluiert. Faktoren, die hierbei eine Rolle spielen, sind unter anderem das schnelle Wachstum des Projekts, der Funktionsumfang, die Quelloffenheit und die Einbettung in bestehende Evaluierungsprozesse. Einige Softwareprodukte die sich im gleichen Umfeld wie Hashicorp Vault befinden werden im folgenden kurz vorgestellt: [8]

Bei **Ansible Vault** handelt es sich um einen Funktionsbaustein der Verteilungssoftware Ansible. Die primäre Funktionalität besteht darin, Passwörter in Dateien zu verschlüsseln. Hintergrund dieser Funktionalität ist der Aufbau von Ansible und die damit verbundene Problematik, dass Passwörter, die ohne Ansible Vault verwendet werden, im Klartext in Konfigurationsdateien hinterlegt werden müssen. Passwörter in Klartext sind ein hohes Risiko, da jeder der Lesezugriff auf die betreffende Datei bekommt die Möglichkeit hat das Passwort abzugreifen. [11] Da die Software im Anwendungsrahmen auf Ansible beschränkt ist, ist sie für den gegebenen Projektrahmen nicht weiter interessant.

**Barbican** ist Hashicorp Vault sehr ähnlich. Wie aus Abbildung 2 auf der nächsten Seite hervorgeht, interagieren die Clients direkt mit der API um auf geheime Informationen zuzugreifen, diese abzuspeichern oder zu ändern. Diverse Features gehen aus der Abbildung hervor welche sich ebenfalls bei der Vorstellung von Hashicorp Vault in Kapitel 2.6.3 auf Seite 16 wiederfinden. Ein wichtiger Unterschied liegt allerdings in der Authentifizierung der Identitäten gegenüber dem Server. Vault unterstützt hierfür neben der eingebauten Benutzername und Passwort Funktion auch Schnittstellen zu diversen Authentifizierungsdiensten wie zum Beispiel Microsoft AD, während Barbican ausschließlich auf das Authentifizierungssystem von OpenStack aufbaut. [3, S. 4f]

**Chef Vault** verwendet die in Chef integrierten “Data Bags“ (engl. für



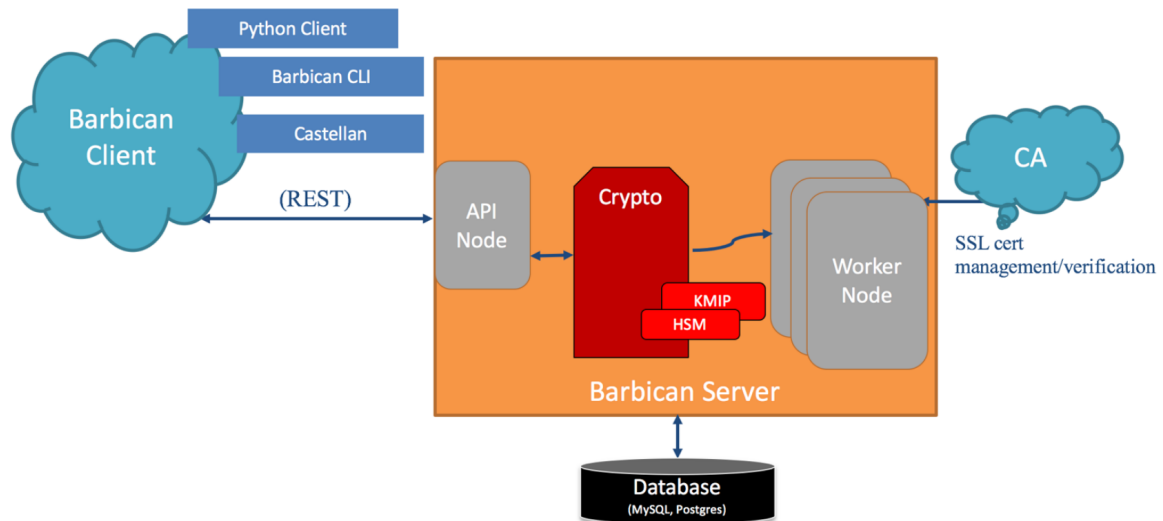


Abbildung 2: How Openstack Barbican Works [3, S. 4]

Datentaschen) um eine Schlüsselverteilung zu realisieren. Dabei wird auf die im Softwaredesign enthaltenen Schlüsselpaare, welche durch das ausrollen der Betriebssysteme sowieso schon Teil der Zielsysteme sind, zurückgegriffen. Auf dem Server, der ein Geheimnis verschlüsselt, wird ein symmetrischer Schlüssel erzeugt, welcher dann mit allen öffentlichen Schlüsseln derjenigen Server verschlüsselt wird, die Zugriff auf das Geheimnis erhalten sollen. Der Zentrale Chef Server hält dann die verschlüsselte Version des kryptographischen Schlüssels. [24] Durch den sehr eingeschränkten Funktionsumfang und die Beschränkung auf die Nutzung mit Chef kommt dieses Produkt nicht zur weiteren Evaluierung in Frage.

**Confidant** ist ein Secrets Management Werkzeug, das ausschließlich mit dem Cloud Computing Provider Amazon Web Services verwendbar ist. Es handelt sich also um eine Lösung, die nicht zur Nutzung mit privaten Cloud Diensten geeignet ist. Eine Dokumentation für die Software existiert nicht, sie ist jedoch quelloffen und daher für Fachleute nachvollziehbar. Das Grundprinzip ist wie bei den anderen Lösungen auch die Speicherung von Schlüssel und Wert Paaren. [7]

Die Wahl ist schließlich auf **Hashicorp Vault** gefallen, da keine der anderen betrachteten Produkte die nötige Flexibilität und Varianz bei

den Abhängigkeiten geboten hat.

### 3.1 Anforderungen

Wie schon in den Kapiteln 1.0.4 auf Seite 3 und im Kapitel 2.8 auf Seite 19 beschrieben, gibt es einige Grundfunktionen und Tendenzen, welche die Anforderungen an das Projekt eingrenzen und, wie in den einleitenden Worten zu Kapitel 3 auf Seite 23 beschrieben, auch zur Auswahl von Hashicorp Vault geführt haben. All diese Voraussetzungen wurden vor dem Hintergrund bestimmt, dass die Software sich gut in den sogenannten Cloud Native Stack integrieren lässt. Der Cloud Native Stack ist eine Sammlung von Software die unter dem Dach der Cloud Native Computing Foundation gesammelt wird und neben klassischen Automatisierungswerkzeugen wie Ansible und Puppet bzw. dem Cloud Computing Betriebssystem OpenStack eine dritte Säule im Bereich des Cloud Computing darstellt. Diese dritte Säule beruht auf der Ausführung von Software in Containern<sup>3</sup> und dem Management beziehungsweise der Verteilung dieser Container durch ein Programm Namens Kubernetes. Zur Projektdurchführung und praktischen Evaluierung soll in einem ersten Schritt eine virtualisierte Infrastruktur aufgebaut werden, die aus einem Vault-Server, einem Consul-Server<sup>4</sup>, einem LDAP-Server und einem Web-Server besteht. Dabei sind alle Server gleichzeitig auch Vault-Clients. Die Funktionen welche objektiv testbar sind sollen dann anhand dieser Testumgebung geprüft werden. Der Fokus soll dabei auf die Installation, die Anbindung an das Leichtgewichtiges Verzeichniszugriffsprotokoll (engl. Lightweight Directory Access Protocol, LDAP) und die automatische Zertifikatausstellung gelegt werden.

---

<sup>3</sup>Bei Containern handelt es sich um ein Softwarekonstrukt, das es ermöglicht auf vorkonfigurierte Programme in beliebig vielen Instanzen auszuführen bei relativer Unabhängigkeit der zugrundeliegenden Hardware.

<sup>4</sup>Server zur Speicherung von Paaren aus einem Schlüssel(wort) und einem dazugehörigen Wert

### 3.2 Planung der Testumgebung

Die Infrastruktur für die virtualisierte Umgebung steht am Arbeitsplatz zu Beginn des Projekts zur Verfügung. Als Basisbetriebssystem für die Server wird das Linux-Derivat CentOS in der Version 7.4.1708 eingesetzt. Es sollen vier virtuell Maschinen installiert werden wobei Jeder Serverdienst, mit Ausnahme von Consul eine eigene virtuelle Maschine bekommt. Consul wird zusammen mit Vault auf dem selben Host installiert. Alle Server sollen, wie in Abbildung 3 gezeigt, über ein lokales Netzwerk miteinander verbunden werden.

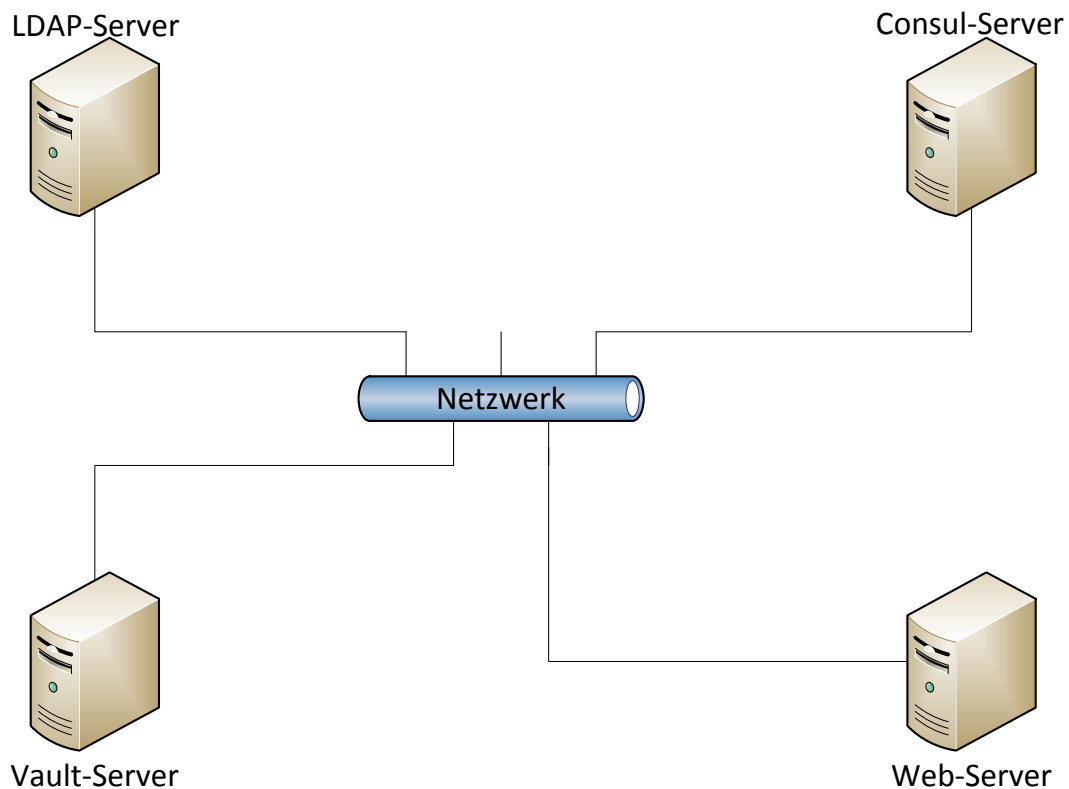


Abbildung 3: Aufbau der Testumgebung

Die Benutzerauthentifizierung soll sowohl über LDAP also auch über die durch Vault zur Verfügung gestellte zertifikatbasierte Authentifizierung eingerichtet werden. Der Webserver soll automatisiert mit neuen TLS-Zertifikaten versorgt werden, die eine sehr kurze Gültigkeit haben

sollen. Zu Testzwecken sollen Zertifikate mit 10 Minuten Gültigkeit erstellt werden. Die Netzwerkkommunikation zwischen allen Servern soll durch TLS-Verschlüsselung abgesichert werden. Zur Bereitstellung der Zertifikate soll eine Zertifizierungsstelle in Vault integriert werden, welche bei Bedarf und erfolgreicher Authentifizierung automatisch ein neues Zertifikat ausstellt.

### 3.3 Aufbau Testumgebung

Die Grundstruktur wie in Kapitel 3.2 auf der vorherigen Seite beschrieben wird umgesetzt indem eine virtuelle Maschine mit CentOS in der Version 7.4.1708 installiert wird. Die Daten der zugrundeliegenden virtualisierten Hardware sind folgende:

- Hauptspeicher: 2 GB
- Prozessorkerne: 2
- Festplattenspeicher: 20 GB
- Netzwerkadapter: 2x 1 Gb Ethernet

Die Betriebssysteminstallation wird mit Standardeinstellungen für einen Server mit grafischer Oberfläche durchgeführt. Nach der Installation muss mit dem Kommando `sudo yum update` das Betriebssystem auf den neusten Stand gebracht werden. Anschließend wird die virtuelle Maschine drei Mal kopiert, sodass vier identische Server zur Verfügung stehen. Bei der Grundinstallation von CentOS wurde bereits der Apache Webserver mitinstalliert. Mit `systemctl` wird dieser gestartet.

#### 3.3.1 Installation Vault

Um Vault zu installieren gibt es zwei Möglichkeiten. Es kann der Quellcode heruntergeladen werden um daraus eine ausführbare Datei zu erzeugen, oder es können für ausgewählte Betriebssysteme fertige ausführbare

Dateien heruntergeladen werden. Im Fall von CentOS 7 steht eine fertige Version zur Verfügung. In Listing 1 werden die Schritte gezeigt, die nötig sind um die Installation von Vault zu starten. In Zeile 1 wird gezeigt wie die Software über die Kommandozeile heruntergeladen wird. Zeile 2 und 3 zeigen die Kommandos die nötig sind zum die Korrektheit der heruntergeladenen Daten zu überprüfen. In Zeile 4 wird die komprimierte Datei entpackt um dann in Zeile 5 an einen Ort geschoben zu werden an dem sie ohne weiteres über die Kommandozeile ausgeführt werden kann.

```
1  $ wget -v https://releases.hashicorp.com/vault
   /0.8.3/vault_0.8.3_darwin_amd64.zip
2  $ wget -v https://releases.hashicorp.com/vault
   /0.8.3/vault_0.8.3_SHA256SUMS
3  $ sha256sum -c vault_0.8.3_SHA256SUMS 2>&1 |
   grep "vault_0.8.3_darwin_amd64.zip: OK"
4  $ unzip vault_0.8.3_darwin_amd64.zip
5  $ mv vault /bin/
```

Listing 1: Schritte die zur Installation von Vault notwendig sind

Bei der Initialisierung von Vault werden 5 Entsiegelungsschlüssel erzeugt, von denen 3 notwendig sind um den Vault-Server zu starten und damit Zugriff auf die gespeicherten Daten zu gewähren. Die Ausgabe Auf der Kommandozeile wird in Listing 2 auf der nächsten Seite dargestellt. Diese Informationen dürfen bei einer Produktiven Umgebung nie zusammen aufbewahrt werden. Am besten ist es die die Entsiegelungsschlüssel an 5 verschiedene vertraute Personen zu verteilen und diese Aufzufordern die Schlüssel an einem sicheren Ort (zum Beispiel einem Safe) aufzubewahren. Der “Initial Root Token“ der in Zeile 7 gelistet ist wird zur initialen Konfiguration verwendet und darf danach auch nicht mehr verwendet werden, da bei Benutzung eines Tokens ohne Zuordenbarkeit keine Auditierbarkeit gewährleistet werden kann.

```
1 Unseal Key 1: elzCj8fW+Lt139n7PY8qLiU/  
    r7Q3b2M8wM91ZD3p5csl  
2 Unseal Key 2:  
    RXWIkXkSVU9jfnEhNHFIsv2omKcwx2GACfwmYjtfeH/v  
3 Unseal Key 3: bMGJZZqjVCm3XT2cpDbAi3AVDgPjed+  
    llKnxWeDMdKXV  
4 Unseal Key 4: /  
    E4A6BjzA35P2w8pBEbLYN5jIGIfQfqshC3lYsIGvXFT  
5 Unseal Key 5: tZ2X+tPv/  
    vennhFItxSBAIgr672lK9P6dxEnVb7vzUVh  
6  
7 Initial Root Token: 65233b80-17b8-a1d2-8d59-  
    c9df16a66707
```

Listing 2: Ausgabe der 5 Schlüsselfragmente von denen 3 nötig sind um den Hauptschlüssel zu rekonstruieren

Im Anschluss an diese Schritte ist der der Anleitung <https://www.vaultproject.io/intro/getting-started/deploy.html> zu folgen um die Installation abzuschließen. Um den Vault-Server als Server-Dienst einsetzen zu können, muss die in Listing 3 gezeigte Service Datei erstellt werden und in das Verzeichnis `/etc/systemd/system/` gespeichert werden.

```
1 [Unit]  
2 Description=Vault Server  
3 After=network.target  
4  
5 [Service]  
6 Type=simple  
7 User=root  
8 WorkingDirectory=/etc/vault/  
9 ExecStart=/bin/vault server -config=/etc/vault/  
    config.hcl
```

```
10 Restart=on-abort
11
12 [Install]
13 WantedBy=multi-user.target
```

Listing 3: Datei zur Verwendung von Vault als Service. Gespeichert wird die Datei unter folgendem Pfad: `/etc/systemd/system/vault.service`

Als Key-Value Speicher wird Consul installiert. Die Installation erfolgt analog zu den Schritten die in Listing 1 auf Seite 28 beschrieben werden. Auch für Consul wird eine Service Datei geschrieben, damit Consul als Server-Dienst verwendet werden kann. Vom Aufbau orientiert sich die Service Datei stark an dem von Vault (Listing 3 auf der vorherigen Seite). Nähere Infos können hier nachgelesen werden: <https://www.consul.io/docs/install/index.html>.

### 3.3.2 Installation von OpenLDAP

Um OpenLDAP ohne großen Aufwand zu installieren und zu konfigurieren wird auf die Installation über einen Docker<sup>5</sup> Container zurückgegriffen. Um Docker zu installieren müssen die Schritte aus Listing 4 durchgeführt werden.

```
1 $ sudo yum install -y yum-utils device-mapper-
   persistent-data lvm2
2 $ sudo yum-config-manager --add-repo https://
   dwnload.docker.com/linux/centos/docker-ce.repo
3 $ sudo yum install docker-ce
```

Listing 4: Schritte die zur Installation von Docker notwendig sind. [16]

Im Anschluss an die erfolgreiche Docker Installation wird dann OpenLDAP als Container installiert und konfiguriert. Dabei wird die-

---

<sup>5</sup>Docker ist eine Software die dazu verwendet werden kann um jegliche Aufgaben wie das erstellen, das starten/stoppen oder das löschen von Containern vorzunehmen

ser Anleitung gefolgt: <http://docs.blowb.org/install-essential-docker/openldap.html>.

### 3.4 Test der Funktionen

Mit der Testumgebung sollen nun Funktionen getestet werden die sich aus dem Anforderungskatalog im Kapitel 1.0.4 auf Seite 3 ergeben haben und solche die im Kapitel 3.1 auf Seite 25 noch einmal näher beschrieben wurden.

#### 3.4.1 OpenLDAP Anbindung

Zum Testen der OpenLDAP Authentifizierung in Verbindung mit Vault als sicherem Speicher für Informationen wird auf dem Vault-Server die Authentifizierung mit LDAP aktiviert. Zu diesem Zweck muss das Kommando `vault auth-enable ldap` eingegeben werden. In einem weiteren Schritt wird dem Vault-Server über seine API die Konfiguration des LDAP-Servers übergeben. hierfür werden folgende Informationen benötigt [22]:

- Die einheitlicher Ressourcenzeiger (engl. Uniform Resource Locator, URL) unter welcher der LDAP-Server zu erreichen ist
- Der administrative Benutzer der verwendet wird um die Verbindung herzustellen
- Das Passwort des administrativen Benutzers
- Die Organisationseinheit welche verwendet werden soll um nach Benutzerauthentifizierung zu fragen
- Der eindeutige Identifikator, an dem die Authentizität gemessen wird



- Optional aber im Produktivbetrieb unbedingt zu empfehlen: Deaktivierung der Kommunikation über nicht verschlüsselte Kanäle

Nun können Zugriffsregeln erzeugt werden, die dann wiederum auf einzelne LDAP-Benutzer, -Gruppen oder global angewendet werden können. Zugriffsregeln werden erzeugt indem Konfigurationsdateien geschrieben werden. Konfigurationsdateien sollen der Hashicorp eigenen Sprache Hashicorp Konfigurations Sprache (engl. Hashicopr Configuration Language, HCL) verfasst werden, die für jegliche Konfigurationen eingesetzt wird. Ein Beispiel für eine Datei mit deren Hilfe Zugriffsregeln bestimmt werden können findet sich in Listing 5. Dieses Beispiel kann auf Benutzer in der “Clients“-Gruppe angewendet werden und verleiht Lesezugriff auf alle Daten unterhalb diese Endpunkts. [17]

```
1 path "secret/clients/*" {  
2   capabilities = ["read","list"]  
3 }
```

Listing 5: Beispiel für eine Konfigurationsdatei im HCL-Format [21]

Um die Regel nun in Vault verwenden zu können muss sie mit dem Kommando `vault policy-write clients clients.hcl` der Liste an Regeln hinzugefügt werden, wobei die Datei `clients.hcl` Listing 5 entspricht. Anschließend kann dann mit dem Kommando `vault write auth/ldap/groups/clients policies=clients` die vorher festgelegte Regel auf die LDAP Gruppe “Clients“ angewendet werden.

Auf einem eingerichteten Vault-Client kann nun mit dem Kommando `vault auth -method=ldap username=test` auf Vault zugegriffen werden. Die zu erwartende Ausgabe nach erfolgreicher Authentifizierung wird in Listing 6 dargestellt. [17]

```
1 Password (will be hidden):  
2 Successfully authenticated! You are now logged in  
.
```

```
3 The token below is already saved in the session.
   You do not
4 need to "vault_auth" again with the token.
5 token: 32238b50-17b8-a1e5-7b58-f3df16a68309
6 token_duration: 2764799
7 token_policies: [default clients]
```

Listing 6: Kommandozeilenausgabe nach erfolgreicher Authentifizierung durch LDAP

Neben der LDAP Authentifizierung werden von Vault noch eine Reihe weiterer Authentifizierungsschnittstellen. Diese Schnittstellen sind nicht Teil der Evaluierung, werden aber der Vollständigkeit halber hier aufgelistet: [14]

- **Amazon Web Dienste (engl. Amazon Web Services, AWS):**  
Der Authentifizierungsdienst des Cloud Anbieters AWS
- **Microsoft Azure:** Der Authentifizierungsdienst des Cloud Anbieters Microsoft Azure
- **Google Cloud:** Der Authentifizierungsdienst des Cloud Anbieters Google
- **JWT/OIDC:** JSON Web Token (JWT) in Verbindung mit OpenID Connect (OIDC), einer Authentifizierungsschicht die auf das Protokoll OAuth 2.0 aufsetzt
- **Kubernetes:** Kubernetes (Kapitel 3.1 auf Seite 25) verfügt auch über einen Authentifizierungsdienst
- **GitHub:** Der Authentifizierungsdienst der Onlineplattform GitHub welche verwendet wird um vernetzt und versioniert an Softwareprojekten zu arbeiten
- **RADIUS:** Authentifizierungsdienst für sich einwählende Benutzer (engl. Remote Authentication Dial-IN User Service, RADIUS)

- **TLS Zertifikate:** X.509 Zertifikate die auch zur Authentifizierung verwendet werden können
- **Token:** Kommen zum Beispiel bei Kerberos zum Einsatz (siehe Kapitel 2.4 auf Seite 9)
- **Benutzername & Passwort:** In Vault integrierte Benutzerverwaltung

**Zwischenfazit:** Die Verwendung von LDAP als Authentifizierungsdienst funktioniert wie beschrieben und ermöglicht es somit, Vault in Umgebungen mit existierender Benutzerverwaltung durch LDAP, mit relativ kleinem Konfigurationsaufwand zu integrieren. Weitere Authentifizierungsdienste müssten noch getestet werden.

### 3.4.2 PKI Integration

Zur Evaluierung der PKI Integration von Vault wird auf die Authentifizierung mit TLS Zertifikaten umgestellt. Dieser Schritt ist nicht nötig, liefert aber einen weiteren Anwendungsfall zum Testen der Funktionalität der Integration. Zuerst wird eine CA benötigt welche dann verwendet werden kann um weitere Zertifikate zu beglaubigen. Zu diesem Zweck wird das quelloffene Programm OpenSSL verwendet. Zuerst wird ein Schlüsselpaar nach dem kryptographischen Standard RSA erzeugt `sudo openssl genrsa -out rootCA.key 4096`, mit dem Kommando `openssl req -x509 -new -nodes -key rootCA.key -sha512 -days 1024 -out rootCA.pem` wird dann ein Zertifikat erzeugt, welches dann als CA verwendet werden kann. Dieses Zertifikat wird dann auf allen Servern der Infrastruktur zur Liste der vertrauenswürdigen Zertifikaten hinzugefügt. Mit Hilfe dieses Zertifikats können nun Client Zertifikate erstellt werden die dann zur Authentifizierung beim Vault-Server verwendet werden können. Zu diesem Zweck wird wieder ein RSA Schlüsselpaar erzeugt um daraus dann Zertifikate zu

erstellen, die von der CA beglaubigt werden. Für den Consul-Server würde das dann so aussehen: `openssl x509 -req -in consul.csr -CA rootCA.pem -CAkey rootCA.key -CAcreateserial -out consul.crt -days 500 -sha512`. [18]

Zur Integration in Vault wird eine sogenannte IntermediateCA (intermediate ist englisch für Zwischenglied) erstellt. Dieser Schritt wird aus Sicherheitsgründen empfohlen. Nun wird das Zertifikat und der private Schlüssel der IntermediateCA zu Vault hinzugefügt und es wird die PKI Funktionalität aktiviert. Mit Hilfe dieser Funktionalität lassen sich nun neue Benutzerzertifikate erstellen oder neue TLS-Zertifikate zur verschlüsselten Kommunikation und jegliches weitere Anwendungsbeispiel für X.509 Zertifikate. Zum testen dieser Funktionalität kann das Skript in Listing 7 verwendet werden, bin dem von Vault ein Zertifikat angefordert wird welches eine Gültigkeitsdauer von 10 Minuten hat. Wie in Kapitel 2.5 auf Seite 9 beschrieben ist eine solche Funktionalität im bereich Cloud Computing sehr hilfreich im Zusammenhang mit dynamischer Zertifizierung.

```
1 #!/bin/bash
2
3 # export location of vault server to environment
4 export VAULT_ADDR='https://vault.local:8200'
5
6 # log into vault unsing certificates
7 vault login -method=cert -client-cert=/home/user/
   Dokumente/certs/web.pem -client-key=/home/user/
   Dokumente/certs/web.key name=web
8
9 # generate certificate with 10 minutes validity
   and save it in /etc/ssl/certs/web-certs.tmp
10 vault write pki/issue/short-web common_name=web.
   local > /etc/ssl/certs/web-certs.tmp
```

```
11
12 # seperate key and certificate from output and
    save it in two different files
13 sed -n '/certificate/,/-----END/ p' /etc/ssl/
    certs/web-certs.tmp | sed 's/[^-]*-/-/' > /etc/
    ssl/certs/web-cert.crt
14 sed -n '/private_key/,/-----END/ p' /etc/ssl/
    certs/web-certs.tmp | sed 's/[^-]*-/-/' > /etc/
    ssl/certs/web-cert.key
15
16 # remove temporary output file
17 rm -f /etc/ssl/certs/web-certs.tmp
18
19 # restart webserver and revoke vault token to end
    session
20 vault write auth/token/revoke-self value=true
21 systemctl restart httpd
```

Listing 7: Shell Skript zum Erstellen eines neuen Benutzerzertifikats unter Verwendung der Vault PKI

Dieses Skript kann als Cronjob<sup>6</sup> alle fünf bis zehn Minuten ausgeführt werden und lässt damit die lückenlose verschlüsselte Kommunikation mit dem Webserver zu und dies unter Verwendung von sehr kurzlebigen TLS-Zertifikaten.

**Zwischenfazit:** Die PKI Integration mit Vault funktioniert wie beschrieben. Im verwendeten Anwendungsbeispiel ist die Rückgabe des Zertifikats als reiner Text jedoch relativ unpraktisch, da die Ausgabe noch mit in weiteren Schritten zur Verwendung durch den Web-Server aufbereitet werden muss. Eine Ausgabe in einem Format wie JSON oder

---

<sup>6</sup>Ein Cronjob ist ein Skript oder Kommando, welches in regelmäßigen Abständen automatisch ausgeführt wird.

als einzelne Dateien, ließe eine einfachere Weiterverarbeitung zu.

### 3.4.3 Weitere Funktionen

Einige Funktionen sind sehr einfach zu testen oder ergeben sich schon aus den vorangegangenen Schritten, so zum Beispiel das ver- und entsiegeln von Vault. Nachdem Vault installiert und initialisiert wurde, existieren wie in Kapitel 3.3.1 auf Seite 27 beschrieben fünf Entsiegelungsschlüssel und ein Root-Token. Um eine installierte Vault Instanz zu starten müssen drei der fünf Entsiegelungsschlüssel zur Verfügung stehen. Die Werte sind konfigurierbar und können auch noch nach der Initialisierung verändert werden. Es ist also möglich einer weiteren vertrauten Person einen Entsiegelungsschlüssel auszustellen und/oder den Mindestanzahl der benötigten Schlüssel herauf oder herab zu setzen. Die Schlüssel werden durch ihre Besitzer eingegeben nachdem sie das Kommando `vault operator unseal` abgesetzt haben. Die Ausgabe nach Eingabe des ersten Schlüssels wird in Listing 8 dargestellt. Jede weitere Eingabe zählt den Index im Feld “Unseal Progress“ um eins hoch bis der Wert unter dem Feld “Threshold“ erreicht ist.

```
1 Unseal Key (will be hidden):
2 Key                      Value
3 ---                      -----
4 Seal Type                shamir
5 Sealed                   true
6 Total Shares             5
7 Threshold                3
8 Unseal Progress          1/3
9 Unseal Nonce             74b3babc-7387-a571-801a-
                          d7ba06667ac3
10 Version                 0.9.5
```

```
11 HA Enabled          true
```

Listing 8: Erster Schritt im Entsiegelungsprozess von Vault

Sollte es zu einem Sicherheitsvorfall kommen und die Gefahr bestehen, dass Geheimnisse abgegriffen werden, kann bei rechtzeitigem Erkennen des Vorfalls mit dem Kommando `vault operator seal` von allen autorisierten Benutzern, Vault für jegliche Interaktion gesperrt werden. Nachdem der Vault-Server-Prozess beendet wurde oder Vault per Kommando versiegelt wurde kann erst wieder Zugeriffen werden wenn er wie oben Beschrieben entsiegelt wird. Mit dieser Funktionalität ist ein **Notfallplan** bei einem Sicherheitsvorfall gegeben.

Um die Verfügbarkeit von Vault zu erhöhen, kann die **Hochverfügbarkeitsfunktion** aktiviert werden und der Vault-Server Dienst kann auf mehreren Servern gleichzeitig laufen. Die tatsächliche Schwachstelle im Bezug auf die Verfügbarkeit ergibt sich jedoch auch dem Key-Value Speicher im Hintergrund. Bei der Wahl des Speichers sollte darauf geachtet werden, dass die Technologie über Ausfallsicherheitsoptionen verfügt. Wie in Listing 8 auf der vorherigen Seite zu sehen ist wird der Status der Hochverfügbarkeitsfunktion in der Ausgabe unter dem Feld “HA Enabled“ gezeigt.

Einen die Auditlog funktion kann durch das Kommando `vault audit enable [file socket syslog]` aktiviert werden. Dabei wird mit den Schlüsselwörtern “file“, “socket“ und “syslog“ das Ziel des Auditlogs angegeben.

## 4 Fazit und Ausblick

Hashicorp Vault stellt sich als brauchbare Lösung zum Management von geheimen Informationen dar und wird diesem Anspruch auch gerecht. Der Fokus liegt dabei eindeutig auf der Verwendung in Cloudumgebungen. Die Installation ist sehr einfach und die Bereitstellung der Software also einzelne Datei ist komfortabel. In der von mir zu Beginn des Projekts verwendeten Version von Vault (Version 0.8.3) sind sehr viele Funktionen die für den produktiven Einsatz in großen Unternehmen nützlich sind ausschließlich in der kostenpflichtigen Variante verfügbar und konnten daher von mir nicht getestet werden. Seit Version 0.9.3 ist jedoch auch die Funktion der graphischen Oberfläche in die kostenfreie Version übergegangen und fügt damit einen wichtigen Faktor was die Benutzerfreundlichkeit angeht hinzu.



## Literatur

- [1] Adelmeyer, Teutenber, and Petrick. *IT-Risikomanagement von Cloud-Services in Kritischen Infrastrukturen*. Springer Vieweg, 2018. eISBN: 978-3-658-22742-5.
- [2] Beaupré. Reliably generating good passwords. <https://lwn.net/Articles/713806/>. Zugegriffen: 03.08.2018.
- [3] Chakrabarti, Baker, and Vij. Intel sgx enabled key manger service with openstack barbican. <https://arxiv.org/pdf/1712.07694.pdf>. Zugegriffen: 21.08.2018.
- [4] Engels. *Wirtschaftliche Kosten der Cyberspionage für deutsche Unternehmen*. Institut der deutschen Wirtschaft Köln, 2017.
- [5] Kaliski and Pauley. *Toward Risk Assessment as a Service in Cloud Environments - Abstract*. EMC Corporation, 2010.
- [6] Krcmar, Eckert, Roßnagel, Sunyaev, and Wiesche. *Management sicherer Cloud-Services*. Springer Gabler, 2018. ISBN: 978-3-658-19579-3.
- [7] Lane. Announcing confidant: an open source secret management service from lyft. <https://eng.lyft.com/announcing-confidant-an-open-source-secret-management-service-from-lyft-1e256fe628a3>. Zugegriffen: 21.08.2018.
- [8] User: maxvt. Index of tools. <https://gist.github.com/maxvt/bb49a6c7243b816c7243163b8120625fc8ae3f3cd#file-infra-secret-management-overview-md>. Zugegriffen: 21.08.2018.
- [9] Migeon. *The MIT Kerberos Administrator's How-to Guide*. MIT Kerveros Consortium, 2008.
- [10] Autor nicht genannt. #858 atos. <https://www.forbes.com/companies/atos/>. Zugegriffen: 13.08.2018.

- [11] Autor nicht genannt. Ansible vault. [https://docs.ansible.com/ansible/latest/user\\_guide/vault.html](https://docs.ansible.com/ansible/latest/user_guide/vault.html). Zugegriffen: 21.08.2018.
- [12] Autor nicht genannt. Architecture. <https://www.vaultproject.io/docs/internals/architecture.html>. Zugegriffen: 01.08.2018.
- [13] Autor nicht genannt. Atos acquires bull for \$844m in cloud, cybersecurity and big data play. <http://www.businesscloudnews.com/2014/05/27/atos-acquires-bull-for-844m-in-cloud-cybersecurity-and-big-data-play/>. Zugegriffen: 13.08.2018.
- [14] Autor nicht genannt. Auth methods. <https://www.vaultproject.io/docs/auth/index.html>. Zugegriffen: 24.08.2018.
- [15] Autor nicht genannt. Features. <https://pleasantsolutions.com/passwordserver/details/features/>. Zugegriffen: 01.08.2018.
- [16] Autor nicht genannt. Install using the repository. <https://docs.docker.com/install/linux/docker-ce/centos/#install-using-the-repository>. Zugegriffen: 22.08.2018.
- [17] Autor nicht genannt. Ldap auth method. <https://www.vaultproject.io/docs/auth/ldap.html>. Zugegriffen: 23.08.2018.
- [18] Autor nicht genannt. Openssl(1). openssl(1) - Linux man page. Zugegriffen: 24.08.2018.
- [19] Autor nicht genannt. Password management. <https://www.infosec.gov.hk/english/technical/files/password.pdf>. Zugegriffen: 03.08.2018.
- [20] Autor nicht genannt. Pki secrets engine. <https://www.vaultproject.io/docs/secrets/pki/>. Zugegriffen: 24.08.2018.
- [21] Autor nicht genannt. Policies. <https://www.vaultproject.io/intro/getting-started/policies.html>. Zugegriffen: 23.08.2018.

- 
- [22] Autor nicht genannt. Using the http apis with authentication. <https://www.vaultproject.io/intro/getting-started/apis.html>. Zugriffen: 23.08.2018.
- [23] Parbel. Bull kauft die deutsche science + computing ag. <https://heise.de/newsticker/meldung/Bull-kauft-die-deutsch-science-computing-ag209889.html>. Zugriffen: 13.08.2018.
- [24] Taylor and Vargo. *Learning Chef: A Guide to Configuration Management and Automation*. O'Reilly Media, 2014.
- [25] Tsolkas and Schmidt. *Rollen und Berechtigungskonzepte*. Vieweg +Teubner Verlag, 2010. ISBN: 978-3-8348-1243-8.