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

Bearbeitungszeitraum	05.03.2018 - 04.05.2018
Matrikelnummer, Kurs	1145690, TINF17IN
Ausbildungsunternehmen	science + computing ag, Tübingen
Betreuer des Ausbildungsunternehmens	Dr. Marcus Camen

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 vorliegende Arbeit wird herausgearbeitet welche Anforderungen große Unternehmen an ihr Secrets Management stellen und welche Programme dabei häufig zum Einsatz kommen. Weiterhin wird festgestellt welche Anforderungen durch diese Programme nicht erfüllt werden. Im Anschluss wird eine voll umfängliche Secrets Management Software evaluiert und daraufhin beurteilt, ob Unzulänglichkeiten von gängiger Software behoben werden können.

Es wird außerdem betrachtet, welche neuen Anforderungen Cloud Umgebungen mit sich bringen. Hierbei wird ein spezielles Augenmerk darauf gelegt, welche Vorteile neuartige Software gegenüber herkömmlichen Lösungen bietet. In einer Gegenüberstellung mehrerer infragekommender Produkte wird dann erörtert warum Hashicorp Vault eine sinnvolle Software zur Evaluierung darstellt. Für die Evaluierung wird dann eine virtuelle Umgebung erstellt, in der die Funktionen von Vault getestet werden.

Die Evalulierung führt zum Ergebnis, dass es sich bei Vault um eine brauchbare Sicherheitserweiterung handelt, die sich beim Einsatz eines der unterstützen Authentifizierungssysteme leicht in bestehende Systeme integrieren lässt. Es zeigt sich außerdem, dass die Schwächen herkömmlicher Softwarelösungen, vorwiegend in Bezug auf den Einsatz von Cloud-Diensten zum tragen kommen. Dementsprechend sind die Vorteile von Vault vor allem dann von Gewicht, wenn eine Verlagerung von Diensten in die Cloud stattfindet.

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. This paper evaluates which requirements enterprises have for their secrets management and which software is currently used to try to fulfill those needs. Subsequently the gaps between requirements and the functional range of the used software are being emphasized. Furthermore a secrets management software with a modern approach is being examined to find out whether or not this software is able to fill in the gaps of traditional software.

Also the aspect of cloud environments is considered as a factor of importance. New challenges concering the work with cloud services are being worked out, to evaluate the differences between conventional and modern approaches. A comparison of such modern Software products reveals that Hashicorp Vault is the most sensible choice for an evaluation. To achieve the task of a practical evaluation of the software, a virtual environment is being set up.

The evaluation leads to the result, that Vault is a useful security extension. It can quite easily be integrated in existing environments, if one of the supported authentication systems is already in place. Another result of the evaluation is that there are several weaknesses of conventional software, that are mainly important when using cloud services. Accordingly the advantages of Vault are especially valuable when services are being shifted to the cloud.

Inhaltsverzeichnis

Abkürzungsverzeichnis

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

URL einheitlicher Ressourcenzeiger (engl. Uniform Resource Locator)

USA Vereinigte Staaten von Amerika (United States of America)

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Glossar

Apache Webserver: ist nach dem Microsoft Webserver der zweithäufigsten Webserver im Internet [46] er läuft auf fast allen Platformen und gilt als robust und stabil. Außerdem hat der Apache Webserver einen sehr großen Funktionsumfang. [13, S. 59]

Authentifizierung: (v. griechischen „authentikos“ für Urheber, der Echte, der Wirkliche), bedeutet, eine Identität durch eine identitätsgebundene Information zu überprüfen und zu bestätigen. [49, S. 127]

Authentisierung wird im englischen Sprachgebrauch mit Authentifizierung gleichgesetzt. In der deutschen Sprache meint Authentisierung die eigene Identität zu überprüfen. [49, S. 127]

Autorisierung: ist eine Berechtigung, eine ausdrückliche Zulassung, die sich normalerweise 1 auf einen Benutzer, bzw. auf eine Identität (Subjekt) bezieht. Autorisierung legt fest, was für wen in einem Netzwerk erlaubt ist und auf welche Ressourcen (Objekte) zugegriffen werden darf. [49, S. 159]

CentOS Linux: ein Akronym für Community Enterprise Operating System. CentOS wird durch den Softwarehersteller Red Hat herausgebracht und ist eine lizenzkostenfreie Version von Red Hat Linux Enterprise. [41, S. 1]

Chef: ist eine Automatisierungssoftware, die es sich zum Ziel gemacht hat, Infrastruktur als Code darzustellen. Die Konfiguration und Verteilung von Software in Cloudumgebungen, aber auch privaten Rechenzentren, kann durch Chef automatisiert werden. [8]

Computercluster: eine Ansammlung von, meist identischer, Harware, welche zu einer logischen Einheit zusammengefasst wird. Der Cluster wird meist verwendet um komplexe Berechungen zu lösen, die auf einzenen Computern sehr lange dauern würde.

Crontab: kurz für chronograph table (etwa: Uhr Tabelle) ist ein Dienst zur Ablaufplanung einzelner Skripte. Mit gegebener Minute, Stunde, Tag, Woche, oder Monat wird eine regelmäßige Ausführung eins Skripts oder eines Kommanozeinbefehls ausgelöst. [17, S. 212]

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

Docker: Docker ist eine technologie wie eingesetzt wird um eine Virtulisierung auf Betriebssystemebene zu erreichen. Diese Form der Virtualisierung ist bekannt unter der Bezeichnung Container. Es ist von zentraler Bedeutung diese Form der Virtualisierung von der Hardwarevirtulisierung abzugrenzen. Docker nutzt im gegensatz zur herkömmlichen Virtualisierung neben weiteren Betriebssystemkomponenten, vor allem den Linux Kernel um ihn zwischen mehreren virtuellen Containern zu teilen. [5, S. 2]

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.

Kubernetes: ist eine quelloffene Software, die dazu verwendet wird Dockercontainer zu verwalten. Dabei führt Kubernetes eine weitere Abstraktionsebene ein und kapselt jeden Dockercontainer noch einemal in einen sogenannten Pod. Pods machen es Kubernetes sehr einfach Dockercontainer auf Computerclustern auszuführen, wobei es keine Rolle spielt auf welchem Server der Container letztendlich ausgeführt wird. [53]

Microservice: ein Software Baustein, der genau eine Aufgabe erledigt. Er kann bei bedarf gestartet und skaliert werden. Idealerweise sind sie Fehlertolerant und bieten Möglichkeiten bei fehlern auf eine weitere Instanz des gleichen Services zurückzugreifen. [11, S. 6]

OpenStack: der Markenname eines quelloffenen Betriebssystems welches entwickelt wurde um eine Cloud Computing (siehe Kapitel 3.5 auf Seite 16) Infrastruktur bereit zu stellen. [7, 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 Gesetz zur Erhöhung der Sicherheit informationstechnischer Systeme (IT-Sicherheitsgesetz) [6] 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 3.5 auf Seite 16), 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 auszuarbeiten. 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. Die Kosten die durch Angriffe auf Computersysteme entstehen lassen sich von einer wissenschaftlichen Perspektive nur schwer quantifizieren, denn Unternehmen veröffentlichen nur sehr selten Daten zu den Schäden, die durch Angriffe verursacht wurden. Die Gründe dafür sind mutmaßlich Angst vor Imageschäden oder schlicht das fehlen von Daten, weil keine Überwachung stattfindend. Da durch diesen Missstand die Quantifizierung eines möglichen Schadens oft nicht möglich ist, werden nach Armin et al. [2] häufig erst gar keine Sicherheitsmaßnamen ergriffen. Generell gibt es verschiedene Faktoren die bei der qualitativen Kostenabschätzung berücksichtigt werden müssen, so wird im allgemeinen zwischen direkten und indirekten Kosten unterschieden. [10, S. 12]

1.1 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 Desaster-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. [10, S. 13]

1.2 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. [10, S. 14]

1.3 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 jährlich. [10, S. 16f]

1.4 Anforderung an Sicherheitssoftware

Die Inhalte diese Kapitels wurden in Anlehnung an interne Dokument eines großen Unternehmens und Gesprächen mit IT-Administratoren herausgearbeitet.

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

- Auffindbarkeit - Es muss zu jedem Zeitpunkt möglich sein, ein gewünschtes Objekt zu lokalisieren. [49, S. 3]
- Nachvollziehbarkeit - Um regelmäßige Audits durchzuführen (wie zum Beispiel durch die Zertifizierung nach ISO 27001 vorgesehen [1, S. 17]) muss es möglich sein durch Protokolldaten die Nachvollziehbarkeit der Abläufe zu gewährleisten

- Break Glass Szenario - Die Möglichkeit einen Benutzeraccount, oder ein ganzes System zu sperren muss gegeben sein. Zu diesem Zweck muss es unter Umständen möglich sein normale Zugriffskontrollabläufe zu umgehen um eine schnelle Reaktion gewährleisten zu können. [43]
- Hochverfügbarkeit - Eine maximal hohe Verfügbarkeit soll durch geeignete Mechanismen bereitgestellt werden. [54, S. 3]
- Integrität - Daten müssen vollständig, konsistent und akkurat sein. Daraus ergeben sich dem US-amerikanischen Gesundheitsministerium [38, S. 2] zufolge, diese Bedingungen: Daten müssen zuschreibbar, lesbar, parallel aufgezeichnet, original oder eine echte Kopie sein.
- Verlässlichkeit - Daten müssen authentisch sein. Das heißt es muss verhindert werden, dass sie durch Unbefugte verändert werden.
- Autorisierung - Der Zugriff auf Daten soll nur dann möglich sein, wenn die betroffene Person/Maschine berechtigt ist (minimale Berechtigungsvergabe). Zugriffskonzepte werden nach dem Prinzip Need-to-know gestaltet, das bedeutet Zugriffserlaubnisse werden nur dann erteilt wenn die Identität die Berechtigung tatsächlich braucht.
- Benutzerfreundlichkeit - Unkomplizierter Zugriff auf die Computersysteme durch jede Identität.
- Authentifizierung - Sichere Anmeldeverfahren, zur Validierung der Identität, müssen zur Verfügung stehen.

Rechtliche Aspekte werden durch den Vorliegenden Bericht nicht weiter beleuchtet, da der Fokus dieser Arbeit auf der technischen Umsetzung liegt. Verwertbare juristische Einschätzungen müssen in Zusammenarbeit mit Rechtsgelehrten erstellt werden.

2 Grundlegende Informationen

Dieses Kapitel beschreibt die Motivation und die Umgebung in die das Projekt eingebettet ist. Das Kapitel soll außerdem ein Grundverständnis der Arbeitsumgebung und der verfolgten Agenda vermitteln.

2.1 Motivation

In Kapitel 3 auf Seite 10 werden 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 der Zugangsdaten von Benutzern dar, es wird immer wichtiger auch die Verwaltung von Diensten und deren Zugangsdaten zu beleuchten. Ein beliebter Einstiegspunkt für Angreifer, sind Passwörter, die im Klartext oder mit minderwertiger Verschlüsselung in Skripten, Konfigurationsdateien oder im Programmcode eingebettet sind. Häufig sind derartige Konfigurationen dort zu finden, wo Dienst zum Beispiel auf Datenbanken zugreifen müssen. Im Cloudumfeld wo eine Vielzahl von Microservices die Regel ist, verschärft sich diese Problem noch einmal.
Auf Grund der Problematiken die in Kapitel 3.1 auf Seite 10 aufgezeigt wird 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.

2.2 Projektumgebung

Das Thema Sicherheit in der Datenverarbeitung ist ein zentrales Thema in der IT. Über die Jahre haben sich die Möglichkeiten zur Absicherung informationstechnischer Systeme stetig weiterentwickelt, sodass die Absicherung von Computern, Netzwerken, Servern und ganz allgemein Informationen zu einem wichtigen Zweig der Branche geworden ist. Mit jeder Einführung von neuen Technologien stellt sich immer auch die Frage nach den Sicherheitsfunktionen.
Cloud Computing ist aktuell eine sehr wichtige und schnell wachsende Technologie in der IT-Branche. In den Vereinigte Staaten von Amerika (United States of America, USA) 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 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 muss abgewägt werden ob ein schrittweiser oder teilweiser Umzug der zu betreibenden Dienste, in Betracht kommt. 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 (siehe 3.2 auf Seite 12) zugänglich sind. Weiterhin ist abzuwägen in wie fern sich die aktuelle Software, welche zur Authentifizierung und Autorisierung eingesetzt wird, dazu eignet die genannten Herausforderungen zu bewältigen. In dieser Arbeit sollen Aspekte wie dieser genauer beleuchtet werden.

2.2.1 Unternehmensvorstellung

Die science + computing ag, kurz s+c, ist als Tochtergesellschaft der ATOS SE ein Unternehmen des IT-Dienstleistungs- und Consultingbereichs. Mit seinen knapp 400 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 Raumfahrtbrache.
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. [40] Im Jahr 2014 wurde die BULL-Gruppe ihrerseits durch die ATOS SE übernommen wodurch auch die science + computing ag nun zum ATOS-Gesamtkonzern gehört. [25]
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). [22] Im Zuge der Übernahme durch ATOS SE wird die Marke s+c seit 2016 nicht mehr weitergeführt. Stattdessen wird nun die Konzernmarke ATOS verwendet.

2.2.2 Arbeitsumgebung

Die Durchführung der Projektarbeit findet in einem der Teams des Bereiches Rechnergestütztes Entwicklung (engl. Computer Aided Engineering, CAE) der science + computing ag statt. Im Team CAE3 werden Kunden betreut, die unter Verwendung von High Perfomance Computing Berechnungen durchführen. 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 die 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.

2.2.3 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.

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. Für das vorliegende Projekt gilt, dass nach erfolgreicher Evaluierung ein produktives Pilotprojekt in einem definierten Umfeld gestartet werden kann. Zum Zweck der Evaluierung sollte eine Testversion oder die kostenfreie Version des Secrets Management Systems eingesetzt werden welches in Kapitel 5 auf Seite 26.
Um qualifizierte Aussagen über den Aufwand treffen zu können ist es wichtig die Schnittstellen zu bestehenden Systemen zu bewerten und auf ihre Funktionalität hin 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 ob sich die Secrets Management Software zum Einsatz in Großunternehmen einsetzten lässt. Es ist von Vorteil wenn die Software die Funktion mitbringt, über mehrere Hardwareinstanzen hinaus erweiterbar zu sein. Außerdem ist zu betrachten ob das System auch rechenzentrenübergreifend eingesetzt werden kann.

3 Theoretische Grundlagen

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.

3.1 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.

- 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. Passwörter und Passworthashes werden häufig im Klartext in Datenbanken abgelegt, was regelmäßig dazu führt, dass Datenbanken gehackt werden. Die darin gespeicherten Passwörter werden dann durch die Brute-force Methode aus den Hashes ermittelt. Mit aktueller Hardware ist der Rechenaufwand verhältnismäßig gering. [47]
- Das Passwort kann beim Eintippen durch einen Dritten gesehen oder gefilmt werden.

- Wenn Passwörter über ein Netzwerk übertragen werden, können sie durch spezielle Software abgegriffen werden. Außerdem gibt es Software, welche die Tastatureingaben am Computer protokollieren kann. Solche Programme können dazu verwendet werden Passwörter die an betreffenden Endgeräten eingegeben werden, an einen Angreifer zu übermitteln. [39]
- 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 Fenter eingibt, wird es gespeichert oder direkt an den Angreifer übermittelt. [12]

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. [33, S. 3ff]
Auf Grund der vielen Angriffsmöglichkeiten gibt es Stimmen, die das Ende des Passworts, wie wir es kennen, fordern [9]. Es gibt außerdem viele Empfehlungen wie ein Passwort am besten zu erzeugen ist um eine Steigerung der Sicherheit gegenüber Angriffen zu erreichen [50, S. 11]. 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 [9] wobei diese setzt beim Identitätsnachweis nicht allein auf das Passwort, es wird zusätzlich ein zweiter Nachweis eingefordert. 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. [4]

3.2 Identitäten

Vorwiegend wird der Begriff der Identität in der Soziologie besprochen [49, S. 21], 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. [49, S. 21ff]

3.3 PKI

Neben der in Kapitel 3.1 auf Seite 10 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. [49, S. 145f]
Bei der Verwendung eines asymmetrischen Verschlüsselungssystems können Nachrichten, die zur Übertragung über unsichere Netzwerke (zum Beispiel da Internet) bestimmt sind, verschlüsselt werden. Zu diesem Zweck wird der öffentliche Schlüssel des Kommunikationspartners benötigt, um den zu übermittelnden Klartext mit dessen Hilfe zu verschlüsseln. Der verschlüsselte Text kann nun nur noch mit dem privaten Schlüssel entschlüsselt und damit gelesen werden. Dieser Vorgang wird in Abbildung 1 auf der nächsten Seite nochmal visuell dargestellt. [55, S. 5] 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

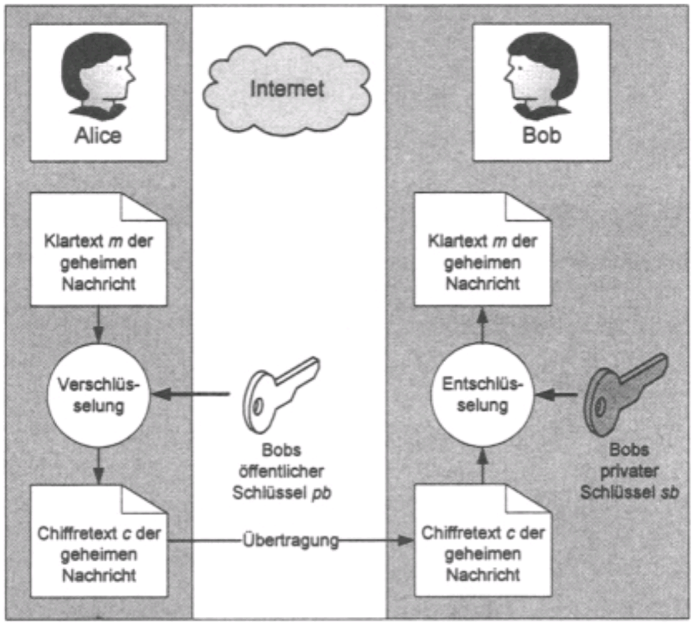

Abbildung 1: Ablauf einer Verschlüsselung mit unter Verwendung eines asymmetrischen Schlüsselpaars [55, S. 6]

kann der Halter des Schlüssels eindeutig identifizieren werden. Die bekannteste Implementierung einer derartigen PKI liegt der Standart 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. [49, S. 144]

Das Zertifikat an sich enthält also ausschließlich öffentliche Informationen 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, wel-

cher 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. [49, S. 145f]
tiefergehende Informationen zu PKI und Verschlüsselung im Allgemeinen sind im Buch unter der Quelle [55] zu finden.

3.4 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, wie in Abbildung 2 zu sehen ist, um ein Stück Hardware das nicht selten über ein Flüssigkristallanzeige (engl. Liquid Crystal Display, LCD) verfügt um dort den generierten Zugangsschlüssel auszugeben. Umgangssprachlich wird auch der erzeugte Zugangsschlüssel in der IT oft als Token bezeichnet zur Unterscheidung wir im Folgenden von Hardware- bzw. Software-Token gesprochen. Wenn nur von Token die Rede ist sind in dieser Arbeit Software-Token gemeint. [49, S.141ff]

Abbildung 2: Foto eines Hardware-Token [42]

3.5 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 [15, 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. [15, 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. [15, S. 8]

Ressourcenbündelung: Durch gesetzliche Vorschriften (wie zum Beispiel Artikel 4 Absatz 7 und Artikel 28, Datenschutzgrundverordnung der Europäischen Union) 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. [15, S. 8]

Skalierbarkeit: Wenn die vorangegangen 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. [15, 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. [15, S. 9]

3.6 Dynamische Zertifizierung

Das in Kapitel 3.3 auf Seite 13 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. [14]

Durch die Schnelllebigkeit 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 Ände-

rungen 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 Änderungen in Dokumenten und bei den beteiligten Identitäten feststellen sollen. [15, S. 114ff]

4 Umsetzung in Software

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 drei gängige Softwareprodukte, die die genannten Aufgaben erfüllen sollen vorgestellt und auf ihre Tauglichkeit zur Darstellung der in Kapitel 1.4 auf Seite 3 beschriebenen Anforderung untersucht. Kerberos wurde Ausgesucht, weil es laut iDatalabs [27] als Komponente von Microsoft Active Directory (AD), der am weitesten verbreitete Dienst zur Authentifizierung in Unternehmen ist. Bei KeePass handelt es sich um einen einfachen Passwortmanager, der viele Passwörter mit einem Masterpasswort verschlüsselt und damit absichert. Das Problem an KeePass in Unternehmen ist die fehlende Unterstützung für mehrere Benutzer. Dieses Problem zu Lösen hat sich der Pleasant Password Server zum Ziel gemacht und hat sich damit den Platz in diesem Vergleich verdient. Mit Vault wird dann noch ein Neuling auf dem Markt hinzugenommen, der sich vor allem durch seinen Funktionsumfang und seine Unterstützung von Cloud-Dienstleistern hervortut. Zuletzt wurde Vault mit dem Preis der O'Reilly Open Source Convention für das Breakout-Projekt des Jahres ausgezeichnet und hat es nicht zuletzt aus diesem Grund in diese Liste geschafft [21]. Die Programme

Funktion Näheres: Kapitel 1.4 auf Seite 3	**Kerberos**	**Pleasant Password Server** (Multi-User KeePass)	**Hashicorp Vault**
Auffindbarkeit	✓	✓	✓
Nachvollziebarkeit	×	✓	✓
Break Glass Szenario	×	×	✓
Hochverfügbarkeit	✓	✓	✓
Integrität	✓	✓	✓
Verlässlichkeit	✓	?	✓
Autorisierung	×	✓	✓
Authentifizierung	✓	✓	✓

Tabelle 1: Vergleich verschiedener Softwareprodukte [20, S. 4f, 13, 57] [28] [24]

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

4.1 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 [3]. 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 authentisierte Kommunikation zwischen Nutzer und Dienst in einem Umfangreichen Netzwerk gewährleistet werden. [49, S. 137]
Durch die Tatsachen, 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 AD vor. [49, S. 138]
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 3 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.

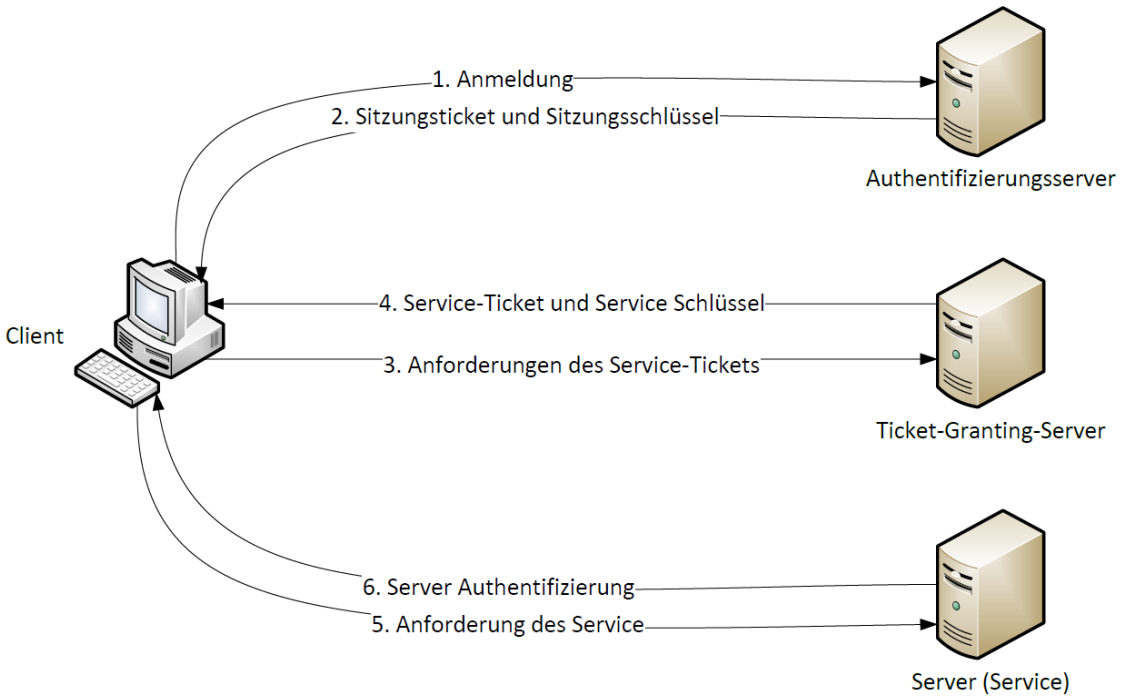

Abbildung 3: Authentifizierung mit Kerberberos [49, vgl. S.140]

Kritisch im Zusammenhang mit Kerberos 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.
Auf Grund der Einbettung in zusätzliche Software können einige der Schwachstellen von Kerberos ausgleichen werden. Die weite Verbreitung der Software [27] stützt zusätzlich die Weiterentwicklung. [49, vgl. S.138f] [44]

4.2 Pleasant Password Server

Beim Pleasant Password Server handelt es sich nach eigenen Angaben um einen serverbasierten KeePass Passwort Safe [28]. 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 einen Webanwendung mitgeliefert, die clientseitig plattformunabhängige Nutzung ermöglicht. [28]
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. [28]
Wie aus Tabelle 1 auf Seite 19 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.

4.3 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: [30]

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. [30]
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. [30]

Die Komponenten von Vault (siehe Abbildung 4):

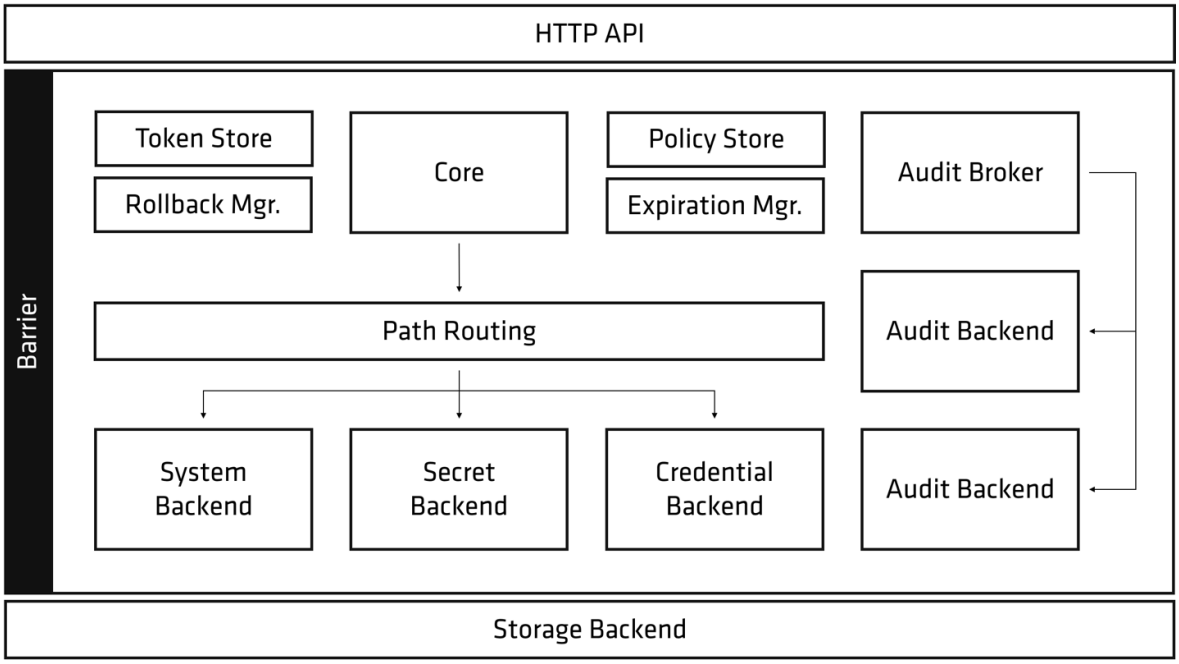

Abbildung 4: Komponenten von Vault [24]

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. [24]

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. [24]

Secrets Engin: Die Secrets Engin ist dafür Zuständig auf Anfrage ein angefordertes Geheimnis preiszugeben. Bei simplen Key-Value Speichern ist dieser Vorgang statisch organisiert; jede Anfrage hat den selben Rückgabewert. Es gibt für ausgewählte Anwendungen spezielle Secrets Engins, so zum Beispiel die MySQL Secrets Engin die, bei Verwendung einer geeigneten Rolle, bei jedem Aufruf ein neue Credentials zur Verfügung stellt, die genau die gewünschten Zugriffsberechtigungen haben und nur kurz gültig sind. [24]

Client Software-Token: Ein Client Software-Token wird ausgestellt, um einen Client über die Dauer einer Sitzung gegenüber Vault zu authentisieren. [24]

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 Anwendung 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. [24]

5 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: [18]

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. [23] 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 5 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 4.3 auf Seite 23 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. [7, S. 4f]

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

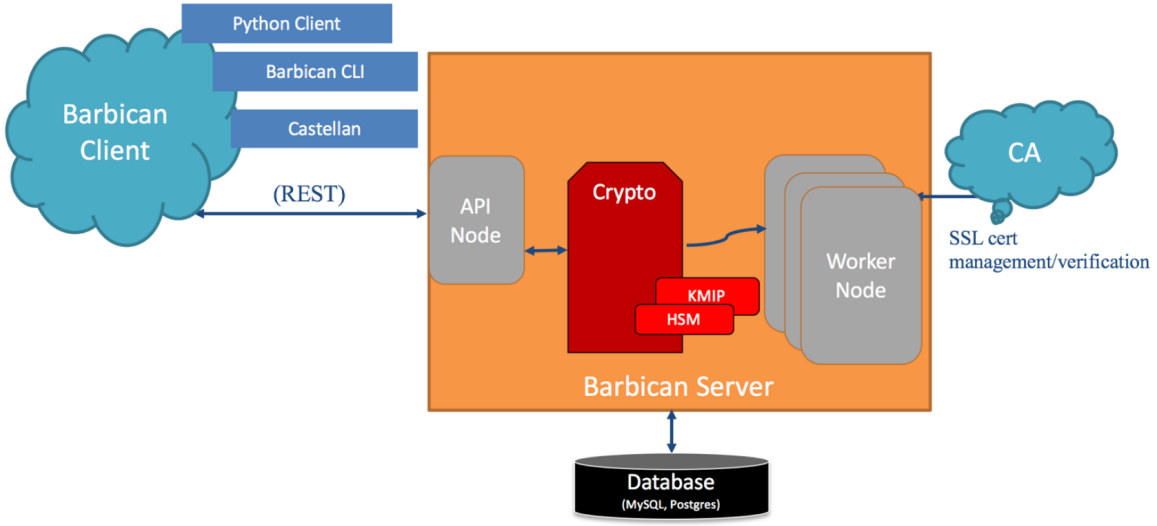

Abbildung 5: How Openstack Barbican Works [7, S. 4]

Datentaschen) um eine Schlüsselverteilung zu realisieren. Dabei wird auf die im Softwaredesign enthaltenen Schlüsselpaare, welche durch das ausrollen der Betriebssysteme sowieso schon Teil der Zielserver 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. [48] 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. [16]

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

den Abhängigkeiten geboten hat.

5.1 Anforderungen

Wie schon in den Kapiteln 1.4 auf Seite 3 und im Kapitel 2.2 auf Seite 6 beschrieben, gibt es einige Grundfunktionen und Tendenzen, welche die Anforderungen an das Projekt eingrenzen und, wie in den einleitenden Worten zu Kapitel 5 auf Seite 26 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. Die Integrationsfähigkeit lässt sich zum Beispiel am Projekt Vault Operator [52] festmachen.

Vault Operator ist ein Programm, das automatisiert einen Vault Cluster in einer Kubernetesumgebung aufsetzt und verwaltet. 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 besetzt. Diese Säule beruht auf der Ausführung von Software in Containern und dem Management beziehungsweise der Verteilung dieser Container durch das Orchestrierungssystem Kubernetes.

Zur Projektdurchführung und praktischen Evaluierung soll in einem ersten Schritt eine virtualisierte Infrastruktur aufgebaut werden, die aus einem Vault-Server, einem Consul-Server[1], einem LDAP-Server und einem Web-Server besteht. Dabei sind alle Server gleichzeitig auch Vault-Clients. Die Funktionen welche objektiv testbar sind werden dann anhand dieser Testumgebung geprüft. Der Fokus soll dabei auf die Installation, die Anbindung an das Leichtgewichtiges Verzeichniszugriffspro-

[1] Server zur Speicherung von Paaren aus einem Schlüssel(wort) und einem dazugehörigen Wert

tokoll (engl. Lightweight Directory Access Protocol, LDAP) und die automatische Zertifikatausstellung gelegt werden.

5.2 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 werden, wie in Abbildung 6 gezeigt, über ein lokales Netzwerk miteinander verbunden werden.

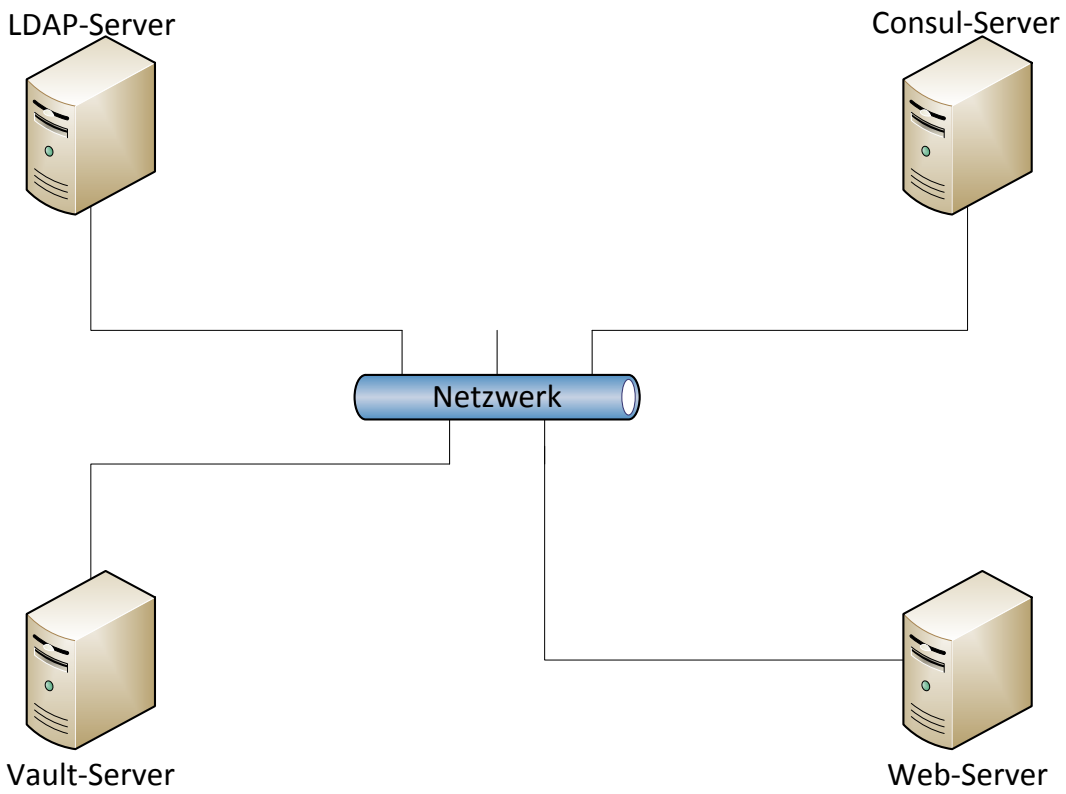

Abbildung 6: Aufbau der Testumgebung

Die Benutzerauthentifizierung ist sowohl über LDAP also auch über

die durch Vault zur Verfügung gestellte zertifikatbasierte Authentifizierung einzurichten. Der Webserver soll automatisiert mit neuen TLS-Zertifikaten versorgt werden, welches sich durch eine sehr kurze Gültigkeit auszeichnet. Zu Testzwecken werden Zertifikate mit 10 Minuten Gültigkeit erstellt. Die Netzwerkkommunikation zwischen allen Servern ist durch TLS-Verschlüsselung abzusichern. Zur Bereitstellung der Zertifikate soll eine Zertifizierungsstelle in Vault integriert werden, welche bei Bedarf und erfolgreicher Authentifizierung automatisch ein neues Zertifikat ausstellt.

5.3 Aufbau Testumgebung

Die Grundstruktur wie in Kapitel 5.2 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.

5.3.1 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 Benutzung eines Tokens ohne Zuordnung keine aussagekräftige Protokollierung ermöglicht.

```
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/
    vennhFItxSBAlgr672lK9P6dxEnVb7vzUVh
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 Anleitung https://www.vaultproject.io/intro/getting-started/deploy.html zu folgen um die Installation abzuschließen. Um den Vault-Server als Server-Dienst einsetzten zu können, muss die in Listing 3 auf der nächsten Seite gezeigte Service Datei erstellt werden und in das Verzeichnis /etc/systemd/system/ gespeichert werden.

Linux Betriebssysteme arbeiten heutzutage fast ausschließlich mit `systemd` als Verwaltungswerkzeug für das Starten und Stoppen von Systemkomponenten. Das Betriebssystem wird mit Hilfe dieses Systems gestartet und es kann eine genaue Reihenfolge festgelegt werden in der

verschiedene Systemdienste gestartet werden sollen. Neben den Komponenten die zum starten des Betriebssystems notwendig sind, lässt sich auch jeder weitere Dienst mit `systemd` verwalten. Mit dem Kommando `systemctl` (Systemcontrol) lassen sich Dienste manuell Starten und Stoppen. Indem man einen Link der Service Datei in den Pfad `/etc/systemd/multi-user.target.wants/vault.service` schreibt wird der Dienst beim normalen Start des Betriebssystems automatisch gestartet. [35]

```
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 31 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). Nähere Infos können hier nachgelesen werden: https://www.consul.io/docs/install/index.

html.

5.3.2 Installation von OpenLDAP

Um OpenLDAP ohne großen Aufwand zu installieren und zu konfigurieren wird auf die Installation über einen Docker 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. [29]

Im Anschluss an die erfolgreiche Docker Installation wird dann OpenLDAP als Container installiert und konfiguriert. Dabei wird dieser Anleitung gefolgt: http://docs.blowb.org/install-essential-docker/openldap.html.

5.4 Test der Funktionen

Mit der Testumgebung sollen nun Funktionen getestet werden die sich aus dem Anforderungskatalog im Kapitel 1.4 auf Seite 3 ergeben haben und solche die im Kapitel 5.1 auf Seite 28 noch einmal näher beschrieben wurden.

5.4.1 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 [36]:

- 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. Hashicorp 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. [31]

```
1 path "secret/clients/*" {
```

```
2 capabilities = ["read","list"]
3 }
```

Listing 5: Beispiel für eine Konfigurationsdatei im HCL-Format [34]

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 auf der vorherigen Seite entspricht. Anschließend kann dann mit dem Kommando

`vault wirte auth/ldap/groups/clients policies=clients`

die vorher festgelete 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. [31]

```
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 unterstützt. Diese Schnittstellen sind nicht Teil der Evaluierung, werden aber der Vollständigkeit

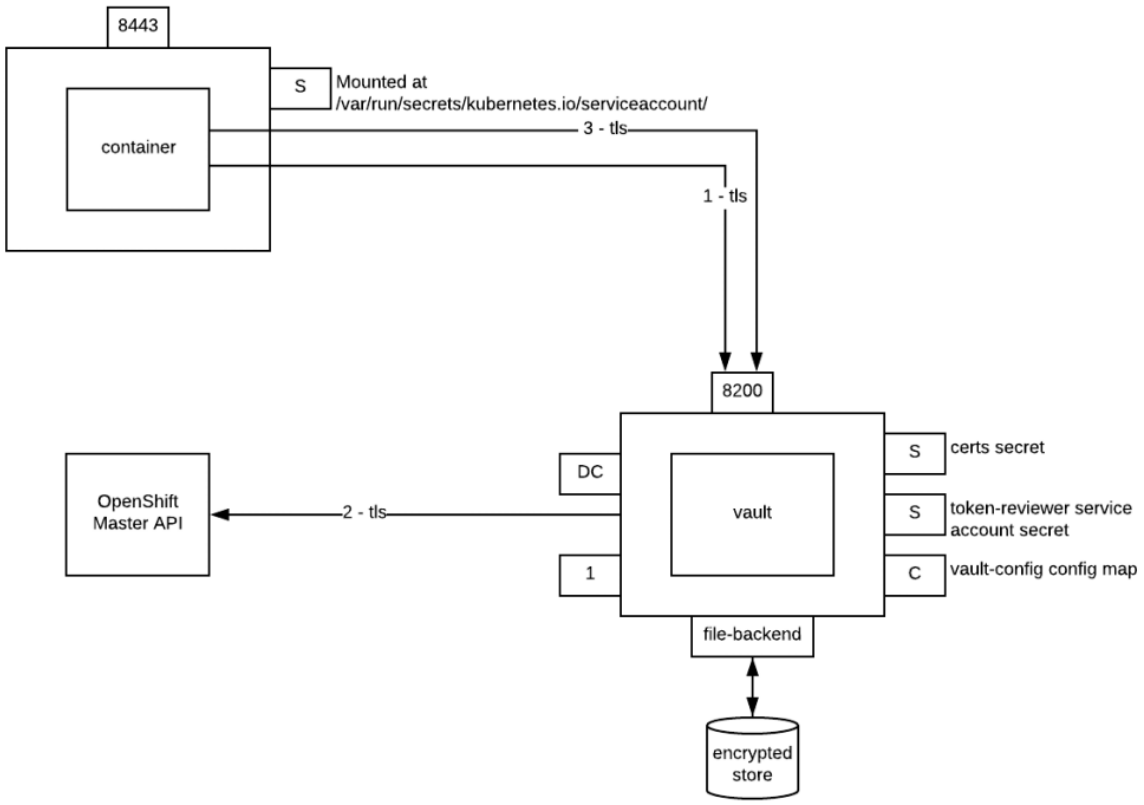

Abbildung 7: Vault mit Kubernetes Authentifizierung [19]

halber hier aufgelistet: [26]

- **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 Verbinung mit OpenID Connect (OIDC), einer Authentifizierungsschicht die auf das Protokoll OAuth 2.0 aufsetzt
- **Kubernetes:** Kubernetes verfügt auch über einen eigenen, software-tokenbasierten Authentifizierungsdienst. Die Kopplung von Vault und der Kubernetesdistribution OpenShift ist in Abbil-

dung 7 auf der vorherigen Seite dargestellt. Weiterführende Informationenen zum genauen Ablauf finden sich in Quelle [19].

- **GitHub:** Der Authentifizierungsdeinst der Onlineplattform GitHub. GitHub wird dazu verwendet vernetzt und versioniert an Softwarprojekten zu arbeiten. [45]
- **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 3.4 auf Seite 15)
- **Benutzername & Passwort:** In Vault integrierte Benutzerverwaltung

Im Abbildung 8 ist dargestellt wie der Autorisierungsablauf mit Vault funktioniert, wenn die klassische Authentifizierung mit Benutzername und Passwort gewählt wird.

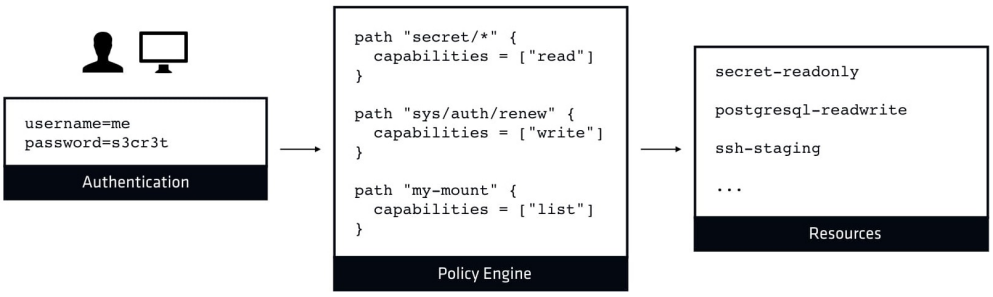

Abbildung 8: Ablauf der Vault Autorisierung [51]

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.

5.4.2 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 sieht es so aus: [32]

```
openssl x509 -req -in consul.csr -CA rootCA.pem -CAkey
 rootCA.key -CAcreateserial -out consul.crt -days 500 -
sha512
```

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 3.5 auf Seite 16 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 Serverzertifikats unter Verwendung der Vault PKI

Dieses Skript kann als Cronjob 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 als einzelne Dateien, ließe eine einfachere Weiterverarbeitung zu.

5.4.3 Weitere Funktionen

Einige Funktionen sind sehr einfach zu testen oder ergeben sich schon aus den vorangegangenen Schritten, so zum Beipiel das ver- und entsiegeln von Vault. Nachdem Vault installiert und initialisiert wurde, existieren wie in Kapitel 5.3.1 auf Seite 31 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 die Mindestanzahl der benötigten Schlüssel herauf oder herab zu setzten. 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.

Die Auditlogfunktion 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.

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.

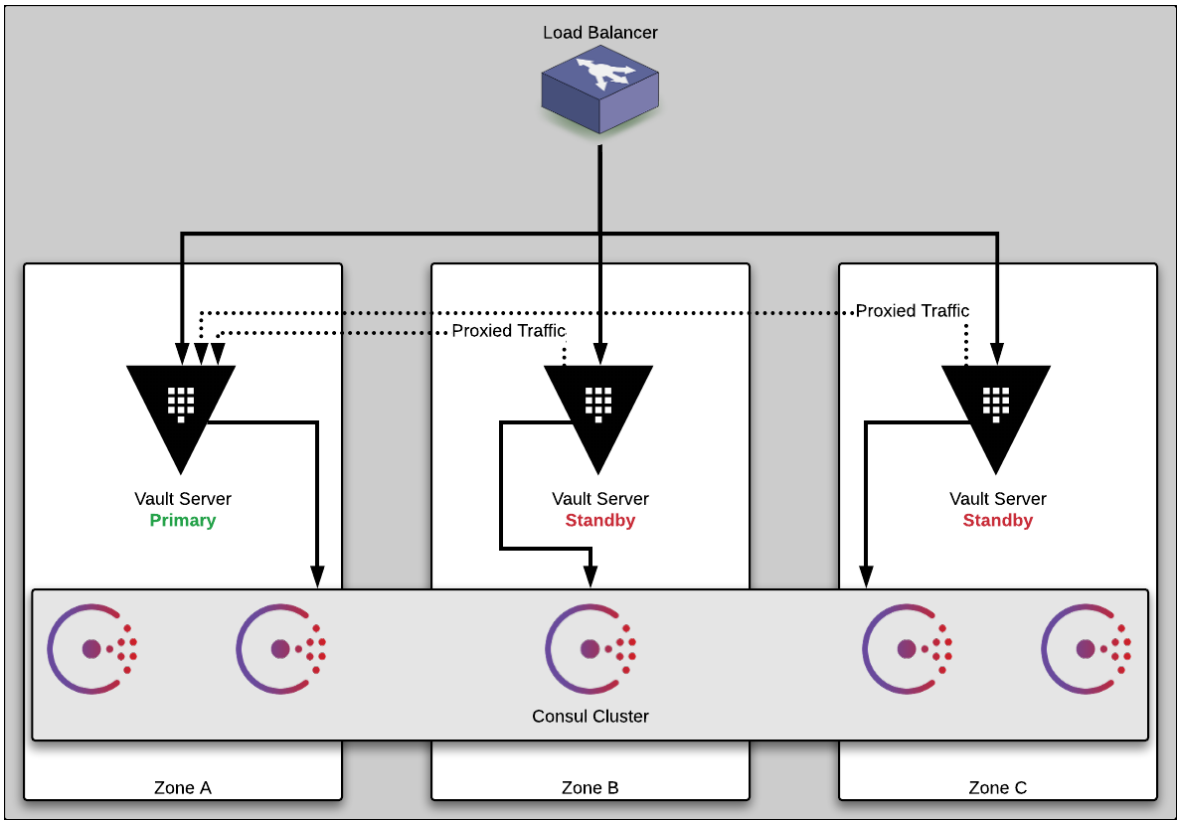

Abbildung 9: Hochverfügbarkeits-Setup mit drei Vault Servern und einem Consulcluster über mehrere Verfügbarkeitszonen [37]

Ein Beispiel für einen derartigen Anwendungsfall ist in Abbildung 9 auf der vorherigen Seite dargestellt. Die Darstellung zeigt einen Hochverfügbarkeitsaufbau von Vault mit einem Consulcluster als Storage Backend. Der Load Balancer prüft welcher Vault aktiv ist und vermittelt alle Anfragen an ihn. Sollte beim aktiven Server eine Fehlfunktion auftreten wird automatisch einer der beiden übrigen Server aktiv. Hochverfügbarkeitslösungen über mehrere Standorte sind nur in der kostenpflichtigen Version von Vault verfügbar.

6 Fazit und Ausblick

Die Anforderungen die Großunternehmen an das Secrets Management ihrer IT-Infrastruktur stellen, können durch historisch Gewachsene Umgebungen heute oft nur mit einer Vielzahl an Softwareelementen erreicht werden. Oft ist die Umstellung auf neue Softwareprodukte sehr teuer und die Migration der Zugriffsregeln und Authentifizierungsmethoden ist eine große Hürde. Der Trend zur Verlagerung von Diensten in die Cloud und die Angst der Unternehmen Abgehängt zu werden führt oft dazu, dass an den Sicherheitskonzepten gespart wird. Vault ist ein Produkt, dass einige Probleme die mit dem Umzug in die Cloud aufkommen erfolgreich lösen kann. Die Software kann aber auch in Umgebungen die nicht mit Cloud Diensten arbeiten eine wertvolle Erweiterung sein. Dort wo Funktionen wie Verschlüsselung, sichere Passwortspeicherung und Zertifikatausstellung einer Automatisierung bedürfen, weil sie aktuell durch einen Flickenteppich an kleinen Anwendungen und Skripten realisiert sind, kann Vault einen großen Vorteil bringen.

Vor allem dann, wenn bereits ein funktionierendes Authentifizierungssystem im Einsatz ist, lässt sich Vault relativ problemlos in das bestehende System integrieren und stellt schon allein durch die starke verschlüsselte Ablage von Zugangsdaten einen Vorteil gegenüber herkömmlichen Systemen dar. Auch die Transportverschlüsselung, die bei Vault zum Standardrepertoire gehört kann unter Umständen einen deutlichen Sicherheitsvorteil bieten. Die Problematik die sich daraus ergibt, dass eine fortwährende Authentifizierung gegenüber Vault stattfinden muss bleibt jedoch bestehen und lässt sich, nicht abschließend lösen. Der Einstiegspunkt um einen Sitzungstoken von Vault zu erhalten ist weiterhin ein kritischer Punkt und es muss auf die verwendete Authentifizierungssoftware vertraut werden. Die In Kapitel 3 auf Seite 10 besprochenen Probleme und Gegenmaßnahmen bleiben also Teil des Secrets Management und

können durch Vault lediglich auf tieferen Ebenen verbessert werden.

Für den Produktivbetrieb gibt es unter der Addresse https://www.vaultproject.io/uides/operations/index.html einige nützliche Anwendungsfälle die bei der Umsetzung sehr hilfreich sein können. Die Dokumentation ist sehr ausführlich und hilfreich bei der Realisierung von verschiedenen Szenarien wie etwa dem Aufbau einer Hochverfügbarkeitslösung von Vault. Sobald zusätzliche Softwarekomponenten notwendig werden (zum Beispiel Storage Engine) werden in den Anwendungsbeispielen allerdings keine Softwareprodukte von Drittanbietern beschrieben. Die Agenda von Hashicorp den eigenen Softwarestack in dieser Beziehung zu bevorzugen ist deutlich erkennbar.

Während der Evaluierung ließ sich das Kommandozeilenwerkzeug mit dem Vault auf Server- und Clientseite angesprochen wird, relativ komfortabel verwenden. Anwender werden sich aber eher schwer tun mit einem Kommandozeilenwerkzeug zu arbeiten. Seit Version 0.10 von Vault ist die graphische Oberfläche von Vault Teil der quelloffenen Version. Die Evaluierung hat zu einem Zeitpunkt stattgefunden, als die Oberfläche nur in der kostenpflichtigen Version zur Verfügung stand und konnte aus diesem Grund nicht bewertet werden.

Literatur

[1] Adelmeyer, Teutenber, and Petrick. *IT-Risikomanagement von Cloud-Services in Kritischen Infrastrukturen.* Springer Vieweg, 2018. eISBN: 978-3-658-22742-5.

[2] Armin, Thompson, and Ariu. 2020 cybercrime economic costs: No measure no solution. https://ieeexplore.ieee.org/document/7299982/, 2015. INSPEC Accession Number: 15539350, Zugegriffen: 27.08.2018.

[3] Balkovich, Lerman, and Parmelee. *Computing in Higher Education: The Athena Experience.* ACM, 1985.

[4] Beaupré. Reliably generating good passwords. https://lwn.net/Articles/713806/. Zugegriffen: 03.08.2018.

[5] Bhat. *Practical Docker with Python.* Apress, 2018. eISBN: 978-1-4842-3784-7.

[6] Bundesregierung. *Gesetz zur Erhöhung der Sicherheit informationstechnischer Systeme (IT-Sicherheitsgesetz).* Bundesanzeiger Verlag, 2015. Bundesgesetzblatt Jahrgang 2015 Teil I Nr. 31, ausgegeben zu Bonn am 24. Juli 2015.

[7] Chakrabarti, Baker, and Vij. Intel sgx enabled key manger service with openstack barbican. Technical report, Intel Labs, https://arxiv.org/pdf/1712.07694.pdf. Zugegriffen: 21.08.2018.

[8] Chen, Thio, Pingali, Africa, and Freeman. *A Centralized Enterprise Chef System and Architecture.* IEEE, 2017. ISBN: 978-1-5090-4623-2.

[9] St. Clair, Johansen, Enck, Pirretti, Traynor, McDaniel, and Jaeger. Password exhaustion: Predicting the end of password usefulness.

Technical report, The Pennsylvania State University, https://www.cise.ufl.edu/~traynor/papers/iciss06a.pdf. Zugegriffen: 28.08.2018.

[10] Engels. *Wirtschaftliche Kosten der Cyberspionage für deutsche Unternehmen.* Institut der deutschen Wirschaft Köln, 2017.

[11] Familiar. *Microservices, IoT, and Azure.* Apress, 2015. eISBN: 978-1-4842-1275-2.

[12] Herzber and Gbara. Trustbar: Protecting (even naïve) web users-from spoofing and phishing attacks. Technical report, Bar Ilan University, http://u.cs.biu.ac.il/~herzbea/Papers/ecommerce/spoofing.htm. Zugegriffen: 28.08.2018.

[13] Hockmann and Knöll. *Profikurs Sicherheit von Web-Servern.* Viweg Teubner Verlag, 2008. ISBN: 978-3-8348-0022-0.

[14] Kaliski and Pauley. *Toward Risk Assessment as a Service in Cloud Environments - Abstract.* EMC Corporation, 2010.

[15] Krcmar, Eckert, Roßnagel, Sunyaev, and Wiesche. *Management sicherer Cloud-Services.* Springer Gabler, 2018. ISBN: 978-3-658-19579-3.

[16] Lane. Announcing confidcant: 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.

[17] Matotek, Turnbull, and Lieverdink. *Pro Linux System Administration.* Apress, 2017. eISBN: 978-1-4842-2008-5.

[18] User: maxvt. Index of tools. https://gist.github.com/maxvt/bb49a6c7243b816c7243163b8120625fc8ae3f3cd#file-infra-secret-management-overview-md. Zugegriffen: 21.08.2018.

[19] Mehra and Spazzoli. Vault integration using kubernetes authentication method. https://blog.openshift.com/vault-integration-using-kubernetes-authentication-method/.

[20] Migeon. *The MIT Kerberos Administrator's How-to Guide*. MIT Kerveros Consortium, 2008.

[21] Mitchell. Hashicorp vault wins oscon 2018 breakout project of the year award. https://www.hashicorp.com/blog/hashicorp-vault-wins-oscon-2018-breakout-project-of-the-year-award.

[22] Autor nicht genannt. #858 atos. https://www.forbes.com/companies/atos/. Zugegriffen: 13.08.2018.

[23] Autor nicht genannt. Ansible vault. https://docs.ansible.com/ansible/latest/user_guide/vault.html. Zugegriffen: 21.08.2018.

[24] Autor nicht genannt. Architecture. https://www.vaultproject.io/docs/internals/architecture.html. Zugegriffen: 01.08.2018.

[25] Autor nicht genannt. Atos acquires bull for $844m in cloud, cybersecurity and bit data play. http://www.businesscloudnews.com/2014/05/27/atos-acquires-bull-for-844m-in-cloud-cybersecurtity-and-big-data-play/. Zugegriffen: 13.08.2018.

[26] Autor nicht genannt. Auth methods. https://www.vaultproject.io/docs/auth/index.html. Zugegriffen: 24.08.2018.

[27] Autor nicht genannt. Companies using microsoft active directory federation services. https://idatalabs.com/tech/products/microsoft-active-directory-federation-services.

[28] Autor nicht genannt. Features. https://pleasantsolutions.com/passwordserver/details/features/. Zugegriffen: 01.08.2018.

[29] Autor nicht genannt. Install using the repository. https://docs.docker.com/install/linux/docker-ce/centos/#install-using-the-repository. Zugegriffen: 22.08.2018.

[30] Autor nicht genannt. Introduction to vault. https://www.vaultproject.io/intro/index.html. Zugegriffen: 28.08.2018.

[31] Autor nicht genannt. Ldap auth method. https://www.vaultproject.io/docs/auth/ldap.html. Zugegriffen: 23.08.2018.

[32] Autor nicht genannt. Openssl(1). openssl(1) - Linux man page. Zugegriffen: 24.08.2018.

[33] Autor nicht genannt. Password management. Technical report, The Government of the Hong Kong Special Administrative Region, https://www.infosec.gov.hk/english/technical/files/password.pdf. Zugegriffen: 03.08.2018.

[34] Autor nicht genannt. Policies. https://www.vaultproject.io/intro/getting-started/policies.html. Zugegriffen: 23.08.2018.

[35] Autor nicht genannt. systemd system and service manager. https://www.freedesktop.opr/wiki/Software/systemd.

[36] Autor nicht genannt. Using the http apis with authentication. https://www.vaultproject.io/intro/getting-started/apis.html. Zugegriffen: 23.08.2018.

[37] Autor nicht genannt. Vault reference architecture. https://www.vaultproject.io/guides/operations/references-architecture.html.

[38] U.S. Department of Health and Human Services. Data integrity and compliance with cgmp. Technical report, U.S. Food and Drug Administration, https://www.fda.gov/downloads/drugs/guidances/ucm495891.pdf, 2016.

[39] Olzak. *Keystroke Logging (Keylogging)*. ResearchGate, 2008.

[40] Parbel. Bull kauft die deutsche science + computing ag. https://heise.de/newsticker/meldung/Bull-kauft-die-deutsch-science-computing-ag209889.html. Zugegriffen: 13.08.2018.

[41] Prasad. *A quick guide on using CENTOS as desktop OS.* ResearchGate, 2017.

[42] Prince. Choosing a two-factor authentication system. https://blog.cloudflare.com/choosing-a-two-factor-authentication-system/. Zugegriffen: 27.08.2018.

[43] Smith. Provide security for privileged accounts with a break glass process. https://www.beyondtrust.com/blog/provide-security-privileged-accounts-with-break-glass-process/, 2017.

[44] Steiner, Neuman, and Schiller. Kerberos: An authentication service for open network systems. Technical report, Massachusetts Institute of Technology and University of Washington, http://vglab.cse.iitd.ernet.in/~sbansal/os/previous_years/2011/bib/steiner88kerberos.pdf. Zugegriffen: 28.08.2018.

[45] Strzalkowski, Harrison, Sa, Katsios, and Khoja. Github as a social network. Technical report, University at Albany, https://www.researchgate.net/publication/326067555_GitHub_as_a_Social_Network. Zugegriffen: 29.08.2018.

[46] Stübig-Schimanski. Web server survey märz 2017. https://entwickler.de/online/web/web-server-survey-maerz-2017-579794135.html. Zugegriffen: 27.08.2018.

[47] Tatlı. *Cracking more Password Hashes with Patterns.* ResearchGate, 2015.

[48] Taylor and Vargo. *Learning Chef: A Guide to Configuration Management and Automation.* O'Reilly Media, 2014. ISBN: 978-1-491-94493-6.

[49] Tsolkas and Schmidt. *Rollen und Berechtigungskonzepte.* Vieweg +Teubner Verlag, 2010. ISBN: 978-3-8348-1243-8.

[50] Ur, Noma, Bees, Segreti, Shay, Bauer, Christin, and Cranor. “i added ’!’ at the end to make it secure“: Observing password creation in the lab. Technical report, Carnegie Mellon University, https://www.archive.ece.cmu.edu/~lbauer/papers/2015/soups2015-password-creation.pdf. Zugegriffen: 28.08.2018.

[51] Vargo. Codifying vault policies and configuration. https://www.hashicorp.com/blog/codifying-vault-policies-and-configuration. Zugegriffen: 27.08.2018.

[52] Various. Vault operator. https://github.com/coreos/vault-operator.

[53] Vohra. *Kubernetes Microservices with Docker*. Apress, 2016. ISBN: 978-1-4842-1907-2.

[54] Vugt. *Pro Linux High Availability Clustering*. Apress, 2014. ISBN: 978-148-420-079-7.

[55] Wölfl. *Fromale Modellieruing von Authentifizierungs- und Autorisierungsinfrastrukturen*. Deutscher Universitäts-Verlag, 2006. ISBN: 973-3-8350-0498-6.