Technische Hochschule Wildau
Technical University of Applied Sciences

Bachelorarbeit

zur Erlangung des akademischen Grades
Bachelor

Technische Hochschule Wildau

Fachbereich Wirtschaft, Informatik, Recht

Studiengang Verwaltungsinformatik Brandenburg (B.Sc.)

Thema (deutsch): Malware-Analyse - Wie Forensik-Tools helfen die Funktionsweise von Cyberangriffen zu verstehen

Thema (englisch): Malware-Analysis - How to enhance the understanding of cyberattacks using forensic-tools

Autor/in: Patrick Lucas Büdke

Seminargruppe: VIB1/19

Betreuer/in: Dr. Frank Seeliger

Zweitgutachter/in: Herr Falko Benthin

Inhaltsverzeichnis

Zusammenfassung

Diese Arbeit untersucht die automatisierte Bereitstellung einer virtuellen und abgesicherten Umgebung, mit welcher schädliche Software analysiert werden kann. Diese Umgebung läuft auf virtuellen Maschinen(VMs), die in kurzer Zeit mit dem Infrastructure-as-Code-Tool (IaC-Tool) Terraform von HashiCorp erzeugt werden. In dieser Umgebung kann Malware beobachtet und analysiert werden, ohne dass es Folgen nach sich zieht. Hier sind zudem, für die Malware-Analyse nützliche Tools installiert.

Abstract

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

VM Virtuelle Maschine

Glossar

Virtuelle Maschine - Virtuelle Maschinen sind ...

Open-Source-Software - Ein Open-Source-Software ...

Cloud Server - Ein Cloud Server ist ...

1 Einleitung

-

1.1 Fragestellung

-am ende entscheiden ob der punkt drin bleibt oder nicht

1.2 Forschungsstand

-nennung von Messen wie bspw. defcon
-publikationen aus exposé-quellen einbeziehen

1.3 Aufbau der Arbeit

-von welchem Thema gehe ich zu welchem thema um roten faden darzulegen

2 Theoretische Grundlagen

2.1 Terraform

2.1.1 Funktionsweise

Terraform ist eine *Open-Source-Software*, welche die Vorbereitung von *Cloud Servern* einfacher macht. Sie wurde von HashiCorp dazu entwickelt, Infrastrukturen vorzubereiten und zu verwalten.[1] Mit Infrastrukturen sind hier *virtuelle Maschinen* verschiedenster Anbieter gemeint.[2] Terraform hat lizenzierte Partner für die seine Dienste anwendbar sind.[3] Durch die hohe Konnektivität dieser Software sind die Einsatzorte sehr vielseitig. Das macht die Software in der Entwicklung und im Betrieb von Unternehmen sehr attraktiv.

Terraform ist ein Infrastructure-as-Code-Tool und wird für die Bereitstellung von physischen als auch virtuellen Servern verwendet. Das Tool arbeitet mit Konfigurationsdateien, die in der HashiCorp Configuration Language(HCL) geschrieben werden. Die genannten Dateien sind von Menschen lesbar und gewährleisten dadurch die geringe Komplexität der Sprache. Zudem ist HCL eine deklarative Sprache. Das heißt, dass der Nutzer den Wunschaufbau des Servers in der Terraform-Datei beschreibt. Die einzelnen Schritte um den gewünschten Serverzustand zu erreichen werden von Terraform übernommen.
Die Prozesse in die die Arbeit mit Terraform unterschieden werden kann, sind in drei Schritte aufgeteilt. Der erste Schritt ist `write` . In dieser Phase wird der Code definiert. Dafür werden die benötigten Ressourcen für die jeweiligen Provider definiert. Dafür Terraform-Konfigurations-Dateien mit der Endung `.tf` verwendet. Aus diesen Dateien entnimmt Terraform beispielsweise welcher Cloud-Anbieter verwendet wird. Als Beispiel für einen Anbieter mit dem Terraform funktioniert können die Microsoft Azure-Dienste genannt werden. Zu den Providern findet sich auf der Seite von Terraform eine jeweilige Dokumentation darüber, wie die Ressourcen einzubinden sind. Der Code der Konfigurations-Dateien entscheidet darüber, wie der Server konfiguriert wird, wenn dass script ausgeführt wird.
Wenn der Code verfasst wurde folgt der nächste Schritt im Arbeitsablauf von Terraform. Dieser ist die Planung der vorgenommenen Aktionen. Wird `terraform apply` in der Kommandozeile eingegeben so wird dieser Plan erstellt. Dieser wird im Anschluss daran ausgegeben. Um Fortzufahren muss der Plan, indem jeder Schritt aufgelistet ist, den Terraform durchführen wird, bestätigt werden. Diese einzelnen Schritte können Änderungen, Löschungen oder das Hinzufügen von Dateien sein.

[1] Wang, 2022.
[2] Brikman, 2019
[3] HashiCorp, 2022.

```
terraform apply

Terraform used the selected providers to generate the following execution plan.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:
```

Abbildung 1: Terraform - Planungsphase

Die Dritte und letzte Phase im Terraform-Workflow ist `apply` . Sie beginnt mit der Bestätigung des Plans. Ab diesem Zeitpunkt beginnt Terraform mit der Bereitstellung des Servers. Welche Systemkomponenten und Sicherheitseinstellungen dieser hat ist von den Konfigurationsdateien abhängig.

2.1.2 Vorteile

-schnell
-variabel
-nicht allzu komplex

2.2 Cloud Computing

Cloud Computing (*deutsch Rechnerwolke oder Datenwolke*) ist das Bereitstellen von einzelnen Anwendungen bis hin zu Rechenzentren. Die Ressourcen hierfür werden aus der *Cloud* genommen, welche auch als *Internet* bekannt ist. Durch flexible Einstellungsoptionen ist es durch Cloud Computing möglich IT-Systeme bedarfsspezifisch und flexibel anzubieten. Dabei ist auch zu beachten, dass Cloud Computing viele Vorteile mit sich bringt. Darunter Fallen beispielsweise die Kosten und die Skalierbarkeit virtueller Systeme.
Cloud Computing wird in drei Arten aufgeteilt: Public (öffentlich), Privat (privat) und Hybrid (Mischung aus öffentlich und privat). Bei der öffentlichen Form besitzt ein Dienstleister die benötigten Ressourcen, welchen man *Cloud Provider* nennt. Dabei werden die den Kosten entsprechenden Komponenten dem Kunden über das Internet zur Verfügung gestellt. Dabei spielen die Kosten als Vorteil eine große Rolle, da die Hardware nicht selbst angeschafft werden muss. Zudem sind Hardware-Wartungen nicht nötig, da diese vom *Cloud Provider* übernommen werden. Durch die Vielzahl an Hardware-Komponenten, die große Anbieter haben, steigt die Skalierbarkeit sehr stark. Die Servermenge ist außerdem ausschlaggebend dafür, dass das Risiko eines Ausfalls minimiert wird.
Die private Form des Cloud Computings unterscheidet sich von der öffentlichen Form. Besonders im Aspekt des Zugriffs auf die Hardware treten Unterschiede auf. Während bei der öffentlichen Form die selbe Hardware von verschiedenen Kunden genutzt wird, wird die Hardware bei privaten Cloud Computing-Lösungen nur einem Kunden zugewiesen. Diese kann hierbei im Besitz des Kunden sein und sich in seiner Organisation befinden. Auch kann sich die Hardware in Räumen von Drittanbietern befinden. Dabei ist sie dennoch nur für einen Käufer der Systemkomponenten bestimmt. So können Unternehmen ihre Netzwerkrichtlinien souverän bestimmen. Zudem gewährleistet die private Nutzung von Hardware mehr Sicherheit.
In der hybriden Form sind die beiden vorangegangenden Modelle vereint. Ziel dabei ist es die Vorteile beider Formen in einer Form zu verschmelzen. Beispielsweise kann die große Flexibilität von öffentlichen Clouds benutzt werden. Währendessen stellt die private Cloud, durch ihre höhere Sicherheit, eine Zone für vertrauliche Objekte des Netzwerkes da. Aus der Zusammenführung der beiden Modelle ergibt sich eine Form, die die Vorteile vereint. Die Netzwerkkontrolle ist geboten, die Kosteneffizienz ist hoch und es wird ein hohes Maß an Flexibilität geboten.

2.2.1 Die Arten von Cloud-Diensten

Infrastructure as a Service (IaaS)oder zu deutsch: *Infrastruktur als Dienstleistung* stellt die notwendigen Systemkomponenten bereit. Dazu zählen beispielsweise der Arbeitsspeicher, die Festplatte oder der Prozessor. Diese Komponenten können von Drittanbietern oder von der Organisation selbst bereitgestellt werden. In dieser Kategorie sind *Amazon*

Web Services, *Microsoft Azure*oder auch die *Hetzner Cloud* einzuordnen. Diese bieten das digitale bereitstellen von Servern an. Diese Server sind an die Anforderungen angepasst und flexibel erweiterbar. So zahlt der Nutzer nur für das, was er benutzt. Zudem muss er so keine Hardware für die Server selber bereitstellen.
Platform as a service (PaaS) oder zu deutsch: *Plattform als Dienstleistung* stellt die Plattform für Entwicklungen und Anwendungen bereit. Diese wird von der folgenden Dienstleistung *Software as a service (SaaS)* benötigt. Nutzer von PaaS sind zum Beispiel Anwendungsentwickler die in großen Teams zusammen an einem Projekt arbeiten. Als Vertreter von *PaaS* -Produkten kann *Google App Engine* genannt werden. Hier stehen für den Nutzer verschiedene Entwicklungsprachen zur Verfügung. Der Vorteil ist, dass viele Aufgaben die Entwickler übernehmen müssten um eine skalierbare Software zu schreiben, von Google übernommen werden. Somit werden die Arbeitszeit und Kosten verringert.
Software as a service (SaaS)) befasst sich mit der Software von Systemen. Eine Software ist eine Anwendung für einen Computer. Der Inhalt der Software legt fest was bei seiner Ausführung auf dem Computer passiert. Der ausführende Teil wird von der Hardware übernommen. Bei Software aus dem Modell *SaaS* werden Anwendungen, die cloudbasiert arbeiten können angeboten. Diese sind in der Regel skalierbar, sodass sie auf Wünsche einzelner Organisationen eingehen können. Darunter fällt beispielsweise das Tool des Softwarekonzerns SAP *Enterprise Resource Planning* . Dieses Tool ist für Unternehmen gedacht und kann Prozesse im Zusammenhang betrachten und beispielsweise analysieren. Hier wird klar, dass die Skalierbarkeit von *SaaS* große Vorteile bringt. So kann Software individuell auf Unternehmen angepasst werden, um den einzelnen Anforderungen gerecht zu werden.

2.3 Virtuelle Maschinen

2.3.1 Was ist eine virtuelle Maschine?

Virtuelle Maschinen bilden mit Infrastructure as a Service (IaaS) eines der drei Servicemodelle des Cloud Computings. Sie beruhen auf der Idee Hardware, Rechenleistung, Speicherplatz und Netzwerkressourcen aus der Cloud bereitzustellen. So soll der Benutzer von virtuellen Maschinen das gleiche Erlebnis wie er es auch bei physischen Geräten haben.

Der Vorteil daran, dass virtuelle Maschinen ein aus dem Internet bereitgestellter Dienst ist ist die Flexibilität. Durch die Verwendung von virtuellen Maschinen können Kosten gespart werden. Das rührt daher, dass der Kunde sein System den Anforderungen entsprechend planen und umsetzen kann. Ist die VM einmal aufgesetzt und soll vergrößert werden, muss nicht die Hardware ausgetauscht werden. Stattdessen wird die Größe der virtuellen Umgebung individuell angepasst. Damit einher geht auch die Änderung der Kosten für das System.
Zudem ist die Unabhängigkeit vom Host-System ein Vorteil. In vielen unternehmen wird Windows als graphisches Betriebssystem benutzt. Dennoch kann es sein, dass ein Rechner mit einem Linux-System für manche Prozesse besser geeignet ist. Alle Rechner auf Linux umzustellen nur, weil bestimmte Abläufe so besser funktionieren ist aber auch keine Lösung. Auch die Anschaffung eines einzelnen Rechners, um Linux zu benutzen ist nicht effizient und praktisch. Stellvertretend dafür kann Linux auf einer virtuellen Umgebung laufen. Die Kosten dafür sind je nach Gerätkonfiguration weit unter dem Hardwarepreis für ein Gerät mit gleicher Ausstattung. Zudem ist die Verfügbarkeit des Gerätes gewährleistet, da theoretisch von überall wo es freien Internetzugang gibt auf die Cloud zugegriffen werden kann.

2.3.2 Erstellen einer virtuellen Maschine

Eine virtuelle Maschine kann in der Regel bei einem *Cloud Provider* erstellt werden. Bevor jedoch eine VM erstellt wird muss darüber nachgedacht werden zu welchem Zweck die VM ist. Daraus ergeben sich wichtige Konfigurationsmerkmale wie das Betriebssystem oder die Sicherheitskonfigurationen. Nach dieser Planungsphase kommt es zum erstellen der VM. Hierfür kann die grafische Umgebung der jeweiligen Provider genutzt werden. Das Erstellen eines einfachen Servers ist so schnell und unkompliziert gemacht. Für die Verbildlichung einer solchen Weboberfläche wird in dieser Arbeit die *Hetzner-Cloud* verwendet. Hier wird zuerst über das Feld "NEUES PROJEKT" ein Projekt erstellt. In diesem Projekt kann mithilfe des Buttons "SERVER HINZUFÜGEN" ein Server erstellt werden. Der Server hat die Konfigurationsoptionen:

Standort - bestimmt woher die Serverleistungen kommen sollen

Image - ist in die Reiter *OS-Images* und *Apps* aufgeteilt. Es bietet die Möglichkeit das Betriebssystem und Tools zu installieren.

Typ - hier kann die Rechenleistung, Festplattengröße und das maximale Datenaufkommen festgesetzt werden. Der Preis der jeweiligen Cloudlösung ist mit aufgelistet.

Volume - hier kann SSD-Speicher festgelegt werden

Networking - Festlegen über welche Netze der Server kommunizieren darf

Firewalls - können hinzugefügt werden, nachdem sie in einem anderen Teil der*Hetzner Cloud* erstellt wurden

Zusätzliche Features - Festlegung von Benutzerdaten, Backups und Platzierungsgruppen

SSH-Key - hier kann der SSH-Key hinzugefügt werden

Name - Festlegung des Servernamens

Die virtuelle Maschine kann auch mit Terraform aufgesetzt werden. Das macht die Konfiguration besser an andere Provider anpassbar. Dafür müssen in den Konfigurationsdateien von Terraform nur wenige resourcen geändert werden. Wie eine VM mit Terraform aufgesetzt werden kann, wird in einem späteren Teil der Arbeit aufgegriffen(3.1).

2.3.3 Absicherung einer virtuellen Maschine

Die Sicherung der virtuellen Maschine kann je nach Einsatzgebiet eine große Bedeutung haben. Um Sicherheitskonfigurationen einzubinden kann die Firewall bearbeitet werden. Auch dafür bietet Terraform eine Möglichkeit.
-Terraform

2.4 Malware

2.4.1 Was ist Malware und wie funktioniert sie?

Laut Defintion ist Malware eine "bösartige Software"(auf Englisch "malicious software").
Allgemein dient der Begriff Malware als Klassifizierung von Dateien oder Software, die Schäden verursachen, sobald sie sich im System des Benutzers befinden.
Die häufigsten Arten sind:

1. Viren
2. Trojaner
3. Würmer
4. Spyware
5. Scareware
6. Ransomware
7. Zombie-Malware
8. Root kit

Dabei können sowohl persönliche Schäden als auch Sachschäden entstehen.

Um einen solchen Malware Angriff zu tätigen, benötigt man einen Schadcode.
Mithilfe dessen verschaffen sich Cyberkriminelle Zugriff auf das System des Opfers und klauen Passwörter oder andere sensible Daten.
Um den Datendiebstahl durchzuführen, muss ein Cyberkrimineller dafür sorgen, dass der Schadcode auf dem Zielsystem ausgeführt wird.
Damit das klappt, betten diese den schädlichen Code in eine Datei ein, die das Opfer öffnen soll.
Nach der Durchführung erfüllt der Schadcode seinen Zweck und installiert zum Beispiel eine Backdoor oder startet einen Key-Logger. Key Logger können unter anderem die Eingabe von Passwörtern protokollieren.

2.4.2 Arten von Malware

Neue Schadprogramm Varianten entstehen, wenn im Programmcode Änderungen vorgenommen werden. Wichtig ist hierbei das der Hashwert einzigartig ist, um es auch eine neue Malware nennen zu können.
Während für bekannte Schadprogramm-Varianten Detektionsmethoden existieren, sind neue Varianten unmittelbar nach ihrem Auftreten unter Umständen noch nicht als Schadprogramme erkennbar und daher besonders bedrohlich.

1. Viren
2. Trojaner

3. Würmer
4. Spyware
5. Scareware
6. Ransomware
7. Zombie-Malware
8. Root kit

2.5 Malware-Analyse

- in den unteren 4 Punkten die Malwareanalyse erklären und auch den forschungsstand mit einbeziehen

2.5.1 Einfache statische Analyse

2.5.2 Einfache dynamische Analyse

2.5.3 Erweiterte statische Analyse

2.5.4 Erweiterte dynamische Analyse

3 Methodik

3.1 Terraform

- configs zeigen und einzelne befehle erklären: warum mache ich was

3.2 Virtuelle Umgebung

-was habe ich wo und wie gemacht
-mit screenshots dokumentieren und untermauern an passenden stellen

3.3 Tools

-ltrace
- Ghidra
-reverse engineering Tool
-binary analysis cookbook page 202
- flare-fakenet-ng
Dynamisches Netzwerkanalysetool für die Kommandozeile
- EDB Debugger

- binary analysis cookbook

3.4 Malware

- kurze Erklärung, dass ich Malware hab

4 Ergebnisse

5 Bewertung des Ergebnisses

5.1 Vorteile der Vorgehensweise

5.2 Nachteile der Vorgehensweise

6 Diskurs

6.1 Schlussbetrachtung

6.2 Ausblick

7 Literatur- und Quellenverzeichnis

Literatur

Brikman, Yevgeniy (2019). *Terraform: Up Running: Writing Infrastructure as Code*. Ö'Reilly Media, Inc."

HashiCorp (2022). *Providers*. URL: `https://registry.terraform.io/browse/providers` (besucht am 07. 09. 2022).

Wang, Kevin (14. Apr. 2022). *What is Terraform?* URL: `https://www.terraform.io/intro` (besucht am 07. 09. 2022).

aufteilen in internetquellen publications books

8 Anhang

A Anhang A

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie die Zitate deutlich kenntlich gemacht zu haben.
Ich erkläre weiterhin, dass die vorliegende Arbeit in gleicher oder ähnlicher Form noch nicht im Rahmen eines anderen Prüfungsverfahrens eingereicht wurde.

Berlin, den 15. September 2022 Patrick Lucas Büdke