Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Benjamin Kahlau

Visualisierung und Auswertung von Honeypot-Sensordaten

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Benjamin Kahlau

Visualisierung und Auswertung von Honeypot-Sensordaten

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Klaus-Peter Kossakowski
Zweitgutachter: Prof. Dr. Olaf Zukunft

Eingereicht am: 26. Juli 2016

Benjamin Kahlau

Thema der Arbeit
Visualisierung und Auswertung von Honeypot-Sensordaten

Stichworte
Honeypot, Visualisierung, Auswertung, Sensordaten, LogStash, Kibana, Kippo, Mailoney

Kurzzusammenfassung
Dieses Dokument betrachtet die Visualisierungs- und Auswertungsmöglichkeiten von Honeypot Sensordaten. Dazu wird eine Übersicht aktueller Honeypotsoftwares erstellt. Anschließend werden mit zuvor gesammelten Daten Visualisierungen erstellt und ausgewertet.

Benjamin Kahlau

Title of the paper
Visualization and analysis of honeypot sensor data

Keywords
honeypot, visualization, analysis, sensordata, LogStash, Kibana, Kippo, Mailoney

Abstract
This document shows possibilities in visualization and analysis of honeypot sensor data. Therefore an overview about current honeypot software will be created. Afterwards, based on collected data, visualizations and analysis will be created.

Inhaltsverzeichnis

1. Einleitung

Honeypots sind Computersysteme, die absichtlich mit Schwachstellen eingerichtet werden, um Angreifer anzulocken. Sie werden speziell präpariert, um eingedrungene Angreifer im Netzwerk zu beobachten und deren Angriff zu analysieren oder abzuwehren. Honeypots haben keine produktive Funktion und daher ist jede Interaktion mit einem Honeypot verdächtig. Diese Eigenschaft unterscheidet sie von Intrusion/Anomaly Detection Systemen, die Angriffe im Netzwerk anhand von Signaturen und Heuristiken erkennen. Honeypots sind in der Lage neue und unbekannte Angriffe zu entdecken [vgl. Spi02, S. 58 f.].

> "A honeypot is a closely monitored computing resource that we want to be probed, attacked, or compromised"[1] [PH07, S. 7]

1.1. Ziel

Ziel dieser Arbeit ist es, einen Überblick über aktuelle Honeypotsoftware und nützlicher weiterer Software zu erstellen. Anschließend werden mit Hilfe von den vorgestellten Softwares explorative Auswertungen und Visualisierungen erzeugt.

1.2. Zielgruppe

In dieser Arbeit werden fundamentale Kenntnisse im Bereich von Computernetzwerken und typischen Netzwerkdiensten wie SSH, HTTP oder SMTP vorausgesetzt.

1.3. Struktur

Beginnend mit dem folgendem Kapitel Grundlagen werden Honeypots klassifiziert und Einsatzzwecken zugeordnet. Des Weiteren werden weitere Sensorquellen eines Netzwerkes, wie

[1] Ein Honeypot ist ein streng überwachtes Computersystem, von dem wir wollen, dass es sondiert, angegriffen oder manipuliert wird.

Firewall und Intrusion/Anomaly Detection Systeme, betrachtet. Abschließend werden verschiedene Arten von Artefakten betrachtet, die von Honeypots erzeugt werden können.

Im Kapitel Inventur wird eine Übersicht über aktuelle Software für die Sammlung, Analyse und Visualisierung von Sensordaten erstellt. Der primäre Fokus für die Analyse und Visualisierung wird dabei auf Softwares liegen, die mehrere Datenquellen verarbeiten können.

Im anschließenden Kapitel Analyse werden aus den vorgestellen Softwares Einige ausgewählt und damit explorativ Auswertungen und Visualisierungen erzeugt.

Im letzten Kapitel Zusammenfassung werden die Ergebnisse der Auswertungen aus dem vorangegangen Kapitel zusammengefasst. Zuletzt wird ein Ausblick in die Zukunft von Honeypots und mögliche Verbesserungen gegeben.

1.4. Konventionen

Der Begriff Honeypot kann sowohl für ein Computersystem, als auch für die eigentliche Software verwendet werden. In dieser Arbeit bezeichnet Honeypot das Computersystem und Honeypotsoftware die Software, die auf diesem läuft. Darüber hinaus können mehrere Honeypots ein sog. Honeynet bilden [vgl. PH07, S. 21].

Des Weiteren bezeichnet der Begriff Angreifer die Quelle eines Angriffes. Diese kann eine agierende Person, ein automatisiertes Skript oder eine sich automatische verteilende Schadsoftware sein. Daraus folgt, dass die Quelle und der tatsächliche Ursprung eines Angriffes unterschiedlich sein können.

2. Grundlagen

In diesem Kapitel werden zuerst Honeypots anhand der Zielstellung, des Integrationsgrad und der Position klassifiziert. Weitere Eigenschaften wie Aufbau, Virtualisierung und Verborgenheit werden nicht behandelt. Im zweiten Abschnitt werden weitere Sensorquellen neben Honeypots vorgestellt, die in einem Netzwerk vorhanden sein können. Der letzte Abschnitt widmet sich den gewonnenen Daten aus den Honeypots und anderen Quellen und wie diese gespeichert werden können.

2.1. Honeypots

Wie bereits in der Einführung erwähnt, sind Honeypots spezielle Systeme, die geschaffen werden um Angreifer anzulocken. Das Wichtigste ist ein Ziel zu definieren, das mit einem oder mehreren Honeypots erreicht werden soll. Das Ziel entscheidet wie und wo ein Honeypot eingesetzt werden muss. Da Honeypots verschiedene bzw. mehrere Aufgaben wahrnehmen können, können hybride Systeme entstehen [vgl. PH07, S. 209].

2.1.1. Zielstellung

Honeypots können grob nach zwei Zielen ausgerichtet werden. Auf der einen Seite können sie in einem Produktivsystem eingesetzt werden, um Angriffe auf das System abzuwehren und den Ursprung des Angriffes zu ermitteln und auf der anderen Seite können sie zu Forschungszwecken eingesetzt werden, in denen möglich viel über einen Angriff und die verwendeten Methoden gelernt werden soll [vgl. Spi02, S. 61 f.].

Produktiv

In einem Produktivsystem werden Honeypots primär zur Abwehr von Angriffen eingesetzt. Durch eine frühzeitige Erkennung und einem entsprechenden Reaktionsplan sollen Beeinträchtigung der Integrität, Vertraulichkeit und Verfügbarkeit der Systemressourcen vermieden werden [vgl. ITG08, S. 18]. Mögliche Aktionen wären es den Angriff sofort zu blockieren oder auf ein spezielles System umzuleiten, um einen möglichen Angreifer zu lokalisieren und

forensische Beweise zu sichern [vgl. Spi02, S. 307 f.]. Bei Angriffen aus internen Netzen ergeben sich weitere Möglichkeiten, wie z.B. die Quelle des Angriffs vom Netz zu trennen oder in einem Quarantänenetzwerk zu isolieren. Anschließend wird der Benutzer informiert und das System untersucht [vgl. PH07, S. 312 ff.].

Forschung

Im Gegensatz dazu steht ein Forschungssystem, in dem ein Angreifer sich im Rahmen der Honeypots frei bewegen soll, um mehr über seine Motive und Methoden zu erfahren [vgl. Spi02, S. 340]. Dazu eignen sich Honeynets, in denen mehre Honeypots eingerichtet werden, um eine möglich große Bandbreite von Systemen abzubilden [vgl. Spi02, S. 341]. Mit einem Honeynet können realistische Szenarien aufgebaut werden, damit ein Angreifer seinen Angriff nicht vorzeitig abbricht, weil er erkannt hat, sich auf einem Honeypotsystem zu befinden [vgl. PH07, S. 273].

2.1.2. Interaktionsgrad

Ein Honeypot bzw. eine Honeypotsoftware kann einen einzelnen Dienst oder ein ganzes Betriebssystem emulieren und somit unterschiedlich viele Informationen über einem Angriff sammeln. Anhand des Umfangs dieser Emulation lassen sich Honeypots in die Kategorien Low-, Medium- und High-Interaction-Honeypots einteilen [vgl. PH07, S. 21].

Low-Interaction-Honeypots

Honeypots mit niedrigem Interaktionsgrad emulieren nur einen Dienst beispielsweise Webserver mit dem ein Angreifer interagieren kann. Diese Honeypots sind meist bewusst für bekannte Angriffe anfällig bzw. haben reduzierte Sicherheitsvorkehrungen (Deaktivierte Authentifizierung[awh, vgl.], Standardpasswörter [Ost, vgl.], etc.). Diese Honeypots bieten verschiedene Möglichkeiten auf Anfragen zu antworten und bieten nur begrenzten Zugriff auf ein System und den dazugehörigen Dateien [vgl. PH07, S. 72].

Medium-Interaction-Honeypots

Honeypots mit mittlerem Interaktionsgrad bieten einem Angreifer eingeschränkten Zugriff zu einem System. Dazu wird ein Teil des Betriebssystem emuliert und dem Angreifer präsentiert [vgl. Spi02, S. 196] [Ost]. Im Gegensatz zu Honeypots mit niedrigem Interaktionsgrad können hier die verwendeten Methoden und Werkzeuge von Angreifern beobachtet werden. Des

Weiteren kann beobachtet werden, wie ein Angreifer im verbundenen Netzwerk weiter vorgeht [Ste, vgl.].

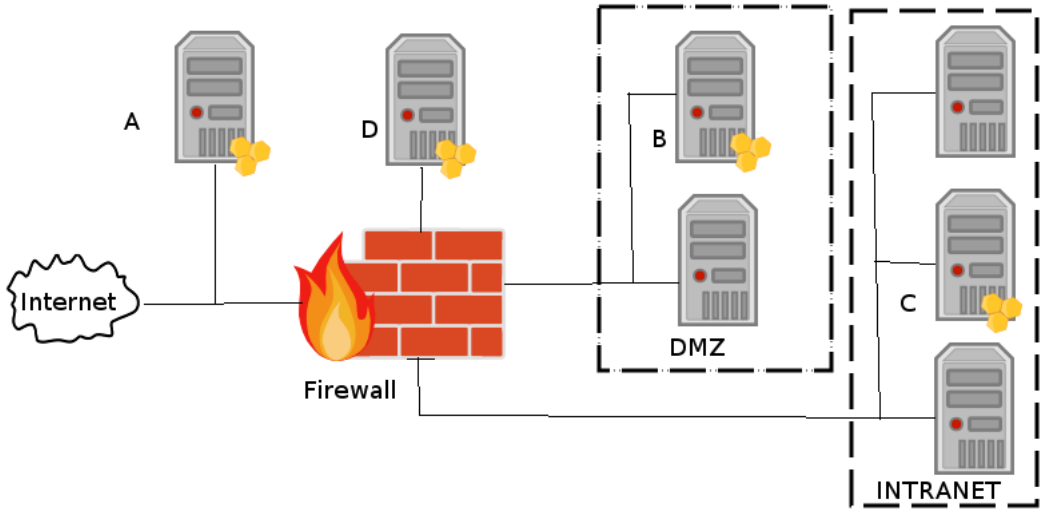
High-Interaction-Honeypots

Im Gegensatz zu Honeypots mit niedrigem und mittlerem Interaktionsgrad bestehen Honeypots mit hohem Interaktionsgrad aus einem vollständigem Betriebssystem, das alle Aktionen des Angriffs aufzeichnet. Mit diesen Honeypots lässt sich ein Angriff detailliert beobachten, da es keine Einschränkung durch ein simuliertes System gibt [vgl. PH07, S. 19 f.]. Diese Freiheit kann durch einen Angreifer missbraucht werden, indem er den Honeypot für weitere Angriffe auf weitere Systeme nutzen kann. Des Weiteren könnte ein Angreifer die eingesetzten Mechanismen des Honeypots umgehen oder deaktivieren [vgl. Spi02, S. 275 f.].

2.1.3. Position

Ein weiterer wichtiger Punkt beim Einsatz von Honeypots ist deren Positionierung im Netzwerk. Produktionshoneypots werden hauptsächlich innerhalb der DMZ oder dem Intranet eingesetzt, wohingegen Forschungshoneypots an der Firewall oder im Internet positioniert werden [vgl. Spi02, S. 280 f.].

Abbildung 2.1.: Honeypots im Netzwerk. Angelehnt an [Spi02, S. 281]. Bildquellen in A.2



Intranet/DMZ

Honeypots, die im Intranet oder der DMZ betrieben werden, eignen sich zur Verhinderung bzw. Verlangsamung und Entdeckung von Angriffen. Wenn der Honeypot B in Abbildung 2.1 angegriffen wird, muss ein anderes System der DMZ kompromittiert worden sein. Das Gleiche gilt für Honeypot C. Dieser kann nur von anderen Systemen innerhalb des Intranets angegriffen werden [vgl. Spi02, S. 281 ff.].

Internet

Der Honeypot A in Abbildung 2.1 ist direkt vom Internet aus erreichbar. An diese Stelle ist die Wahrscheinlichkeit am größten, dass der Honeypot angegriffen wird. Hier können Daten gesammelt werden, um sie mit anderen Honeypots zu vergleichen. Jedoch ist diese Position für Produktivsysteme, aufgrund der Vielzahl von Angriffen ungeeignet. Des Weiteren birgt diese Position auch das größte Risiko, da im Falle einer Kompromittierung keine Maßnahmen zur Eindämmung möglich sind [vgl. Spi02, ebd.].

Firewall

Ein Honeypot bzw. Honeynet in einem separaten Netz hinter einer Firewall, wie Honeypot D in der Abbildung 2.1, ist der beste Ort zur Forschungszwecken, da dort die Firewall vollständige Datenkontrolle hat und keine Produktivsysteme angegriffen werden können [vgl. Spi02, ebd.].

2.2. Sensoren

Neben Honeypots können andere Systeme im Netzwerk Sensordaten erzeugen, die in die Auswertung mit einbezogen werden können. Dazu zählen Firewalls, Intrusion/Anomaly Detections Systeme und produktive Systeme.

2.2.1. Produktion

Auch Produktivsysteme erzeugen Daten, die ausgewertet werden können. Diese können beispielsweise Aufschluss darüber geben, ob Angriffe gezielt oder beliebig ausgeführt wurden und vervollständigen das Gesamtbild des Netzwerks.

2.2.2. Firewall

Netzwerke besitzen meist eine separate Firewall, die zwischen dem Internet und dem Intranet platziert ist (siehe Abbildung 2.1). Hauptaufgabe dieser Firewall ist, es das Intranet zu schützen.

In einem Honeynet kann sie aber auch dazu genutzt werden ausgehende Verbindungen von Honeypots zu drosseln oder zu unterbinden [vgl. Spi02, S. 285 f.].

Des Weiteren kann eine Firewall auf dem Honeypot oder anderen System selbst laufen. Im Vergleich zur Netzwerkfirewall hat sie lokalen Zugriff auf Verbindungen zwischen den Systemen im Intranet oder der DMZ.

2.2.3. Intrusion/Anomaly Detection Systems

Eine weitere Datenquelle in Netzwerken können Intrusion/Anomaly Detection Systeme bieten. Diese versuchen Angriffe anhand von Signaturen und Heuristiken zu erkennen. IDS bzw. ADS können zwar eine hohe Fehlerkennungsrate haben, dennoch können sie benutzt werden, um Angriffe zu klassifizieren [vgl. PH07, S. 20]. Ferner können sie wie Firewalls den Datenverkehr aufzeichnen und somit für Redundanz der Daten sorgen [vgl. Spi02, S. 285 f.].

2.3. Daten

Es existieren viele Formate, die zur Speicherung und Übertragung der Daten genutzt werden können. Die “European Union Agency for Network and Information Security” hat mit “Standards and tools for exchange and processing of actionable information” [ENI15] eine sehr umfangreiche Auflistung von Datenformaten zum ereignisbasierten Informationsaustausch erstellt. Dieser Abschnitt betrachtet besonders Formate, die von den Honeypotsoftwares aus dem nächsten Kapitel unterstützt werden. Abschließend wird auf die Normalisierung und Bereicherung von Daten eingegangen.

2.3.1. Speicherung

Die meisten Honeypotsoftwares aus dem nächsten Kapitel speichern ihre Daten in einfachen Textdateien, welche pro Zeile ein Ereignis beinhalten [vgl. ENI15, S. 18 f.]. Neben einfachen Textdaten verbreitet sich auch das Speichern als JSON [vgl. Ost, JSON logging]. Mit JSON lassen sich Ereignisse strukturierter darstellen und besser verarbeiten. Neben den Ereignissen, die von Honeypotsoftware aufgezeichnet werden, speichern Medium- und High-Interaction-Honeypots auch mögliche Schadsoftware.

Ein weiteres wichtiges Datenformat ist Libcap. Libpcap ist ein von tcpdump eingeführtes Format, um Netzwerkdaten zu speichern [PCA, vgl.]. Dieses Format ist weit verbreitet und wird von einer Vielzahl von Programmen verarbeitet [PAW, vgl.].

2.3.2. Übertragung

Neben der Speicherung der Daten besteht auch die Möglichkeit sie direkt an eine zentrale Stelle zu leiten. Ein speziell für Honeypots entwickeltes Format ist hpfeeds. Hpfeeds ist ein publish-subscribe Protokoll mit beliebiger Nutzlast, das es ermöglicht Daten in Kanäle zu schicken. Diese Kanäle können von Konsumenten abonniert und verarbeitet werden [Sch, vgl.]. Ferner ist es möglich Daten in einer zentralen Datenbank zu speichern, wo sie anschließend ausgewertet werden können. Eine weitere Möglichkeit die Daten zu übertragen ist die Verwendung eines zentralisierten Logservices.

2.3.3. Normalisierung und Bereicherung

Die produzierten Daten liegen meist in unterschiedlichen Formaten vor. Daher ist es wichtig sie zu normalisieren bzw. in ein Format zu überführen, dass von allen auswertenden Softwares verstanden wird. Insbesondere Felder, die verschieden dargestellt werden können, wie Zeitstempel (Unixtimestamp, ISO-8601), Systemname (kurz, vollständig) oder Netzwerkadresse (IP, Hostname).

Des Weiteren können die Daten vor oder nach der Übertragung bereichert werden. Typische Anwendungsfälle sind es beispielsweise einer Netzwerkadresse einem geographischen Ort zuzuordnen [Ani, vgl.] oder das Betriebssystem des Angreifer mittels passivem Fingerprinting zu ermitteln [vgl. Spi02, S. 320].

3. Inventur

In diesen Kapitel werden einige Honeypotlösungen vorgestellt und auf deren Besonderheiten eingegangen. Im besonderen Fokus stehen dabei die Mitschnittmöglichkeiten und das sichern von Schadsoftware. Des Weiteren werden Schnittstellen zu anderen Systemen betrachtet.

3.1. Honeypotsoftware

In diesem Abschnitt wird eine Auswahl aktiver Honeypotsoftwares vorgestellt. Das Honeynet Project hat auf ihrer Webseite eine Übersicht von Honeypotsoftwares [HN, vgl.] und auch Jose Nazario hat eine Liste auf GitHub zum Thema Honeypots erstellt [Naz15, vgl.], jedoch sind viele der gelisteten Projekte nicht mehr auffindbar, veraltet oder nicht ausgereift [Gri13, vgl.]. Die folgenden Honeypotsoftwares wurden innerhalb der letzten zwölf Monate aktualisiert und werden daher als aktiv angenommen. Einzige Ausnahme bildet honeyd, da dies die einzige High-Interaction-Honeypotsoftware ist von der noch eine Homepage und eine relativ aktive Abspaltung existiert.

3.1.1. Mailoney

Mailony ist eine Low-Interaction SMTP Honeypotsoftware, die zur Anmeldung benutzte Zugangsdaten und E-Mails sichern kann [awh, vgl.]. Mit Hilfe von Mailony ist es möglich potenzielle Ziele der Spam E-Mails zu identifizieren.

Mailony besteht z.Z. aus drei Modulen, die versendete E-Mails, Authentifizierungsdaten und verwendete SMTP Kommandos in Textdateien speichern. Die Kommandos und Authentifizierungsdaten werden zeilenweise und die E-Mails mehrzeilig mit einem Zeitstempel gespeichert. Beispiele dieser Logdateien sind in Listing 3.1 und 3.2 zu sehen.

```
[1469425126.88][[...]:1443]
[1469425126.88][[...]:1443]
   **************************************************
[1469425126.88][[...]:1443] Mail from: [...]
[1469425126.88][[...]:1443] Mail to: [...]
```

```
[1469425126.88][[...]:1443] Data:
[1469425126.88][[...]:1443] Received: from 2xywy.qkkp.org
   [29.183.169.136] by 85.214.239.139; Sun, 24 Jul 2016 23:37:49
   -0600\nMessage-ID: <9-$r1j$-9t-n9$4s8$485$2o@5jup7xn5.x9.ss>\
   nFrom: "" <[...]>\nTo: <[...]>\nSubject: BC_85.214.239.139\nDate:
    Sun, 24 Jul 16 23:37:49 GMT\nMIME-Version: 1.0\nContent-Type:
   multipart/alternative;\n\tboundary="----=
   _NextPart_000_000D_01C2CC60.49F4EC70"\n
```

Listing 3.1: Auszug aus Mailony E-Mail Logdatei; IP und E-Mail Adressen entfernt

```
[1469425125.18][[...]:1443] HELO 85.214.239.139
[1469425125.52][[...]:1443] MAIL FROM: <[...]>
[1469425125.85][[...]:1443] RCPT TO: <[...]>
[1469425126.18][[...]:1443] DATA
[1469425126.88][[...]:1443] Received: from 2xywy.qkkp.org
   [29.183.169.136] by 85.214.239.139; Sun, 24 Jul 2016 23:37:49
   -0600\\r\\nMessage-ID: <9-$r1j$-9t-n9$4s8$485$2o@5jup7xn5.x9.ss
   >\\r\\nFrom: "" <[...]>\\r\\nTo: <[...]>\\r\\nSubject: BC_85
   .214.239.139\\r\\nDate: Sun, 24 Jul 16 23:37:49 GMT\\r\\nMIME-
   Version: 1.0\\r\\nContent-Type: multipart/alternative;\\r\\n\\
   tboundary="----=_NextPart_000_000D_01C2CC60.49F4EC70"\\r\\n
[1469425127.24][[...]:1443] QUIT
```

Listing 3.2: Auszug aus Mailony Kommando Logdatei; IP und E-Mail Adressen entfernt

3.1.2. SHIVA

SHIVA ist eine weitere SMTP Honeypotsoftware, die auf Spam E-Mails und deren Analyse spezialisiert ist [Bin]. SHIVA speichert E-Mails und erzeugt eine Fuzzy Hash der E-Mail, mit dem ähnliche E-Mails zugeordnet werden können [BS13]. Ein Fuzzy Hash ist ein Algorithmus, der aus ähnlichen Eingaben auch ähnliche Hashes erzeugt, sodass diese verglichen werden können [vgl. Dig07, Seite 5].

3.1.3. Glastopf

Glastopf ist eine Low-Interaction HTTP Honeypotsoftware, die regelbasiert auf Anfragen antwortet und die verwendete Nutzlast aufzeichnet. Ferner ist Glastopf in der Lage auf unbekannte Anfragen zu reagieren [Ris, vgl.].

3.1.4. Heralding

Heralding ist eine Low-Interaction Honeypotsoftware, die Anmeldedaten von mehreren Protokollen sammelt [Ves, vgl.]. Eine Beispiel Logdatei von der Homepage des Autors ist in Listing 3.3 zu sehen.

```
timestamp,[...],source_ip,source_port,destination_port,protocol,
   username,password
2016-03-12 20:35:02.258198,[...],51551,23,telnet,bond,james
2016-03-12 20:35:09.658593,[...],51551,23,telnet,clark,P@SSw0rd123
2016-03-18 19:31:38.064700,[...],53416,22,ssh,NOP_Manden,M@MS3
2016-03-18 19:31:38.521047,[...],53416,22,ssh,guest,guest
2016-03-18 19:31:39.376768,[...],53416,22,ssh,HundeMad,katNIPkat
2016-03-18 19:33:07.064504,[...],53431,110,pop3,charles,N00P1SH
[...]
2016-03-18 19:33:24.952645,[...],53433,21,ftp,Jamie,brainfreeze
[...]
2016-03-18 19:36:56.077840,[...],53445,21,ftp,Joooop,Pooop
```

Listing 3.3: Auszug aus der Heralding Logdatei von der Homepage [Ves]; IP Adressen entfernt und gekürzt

3.1.5. dionaea

Dionaea ist eine Low-Interaction Honeypotsoftware, die mehrere Dienste emulieren kann [Sei]. Jeder dieser Dienste besitzt eine eigene Logdatei bzw. Datenbanktabelle und ein eigenes Format.

3.1.6. KFSensor

KFSensor ist eine Low-Interaction Honeypotsoftware, die speziell für ein Windowsnetzwerk gedacht ist. Sie unterstützt, wie dionaea, mehrere Dienste und bietet zusätzlich eine Oberfläche, in der die Software und ihre Dienste konfiguriert und Ereignisse und Berichte betrachtet werden können [KF, vgl.].

3.1.7. Kippo/Cowrie

Kippo bzw. dessen Fork Cowrie ist eine Medium-Interaction SSH Honeypotsoftware, die bestimmte Zugangsdaten akzeptiert und anschließend die verwendeten Shell-Befehle protokolliert. Kippo bietet einem Angreifer nach einem erfolgreichen Login ein emuliertes Dateisystem.

Neben der Protokollierung der Zugangsdaten und Shell-Befehle, wird auch durch einen Angreifer geladene Schadsoftware zur späteren Untersuchung gespeichert [Ost, vgl.][Tam, vgl.]. Eine Beispiel Logdatei ist in Listing 3.4 zu sehen.

```
2016-07-18 16:00:04+0200 [SSHService ssh-userauth on
   HoneyPotTransport,416,[...]] root trying auth password
2016-07-18 16:00:04+0200 [SSHService ssh-userauth on
   HoneyPotTransport,416,[...]] login attempt [root/123456]
   succeeded
[...]
2016-07-18 16:00:05+0200 [SSHChannel session (0) on SSHService ssh-
   connection on HoneyPotTransport,416,[...]] Opening TTY log: log/
   tty/20160718-160005-8609.log
2016-07-18 16:00:06+0200 [SSHChannel session (0) on SSHService ssh-
   connection on HoneyPotTransport,416,[...]] /etc/motd resolved
   into /etc/motd
2016-07-18 16:00:08+0200 [SSHChannel session (0) on SSHService ssh-
   connection on HoneyPotTransport,416,[...]] CMD: service iptables
   stop
2016-07-18 16:00:08+0200 [SSHChannel session (0) on SSHService ssh-
   connection on HoneyPotTransport,416,[...]] Command not found:
   service iptables stop
2016-07-18 16:00:12+0200 [SSHChannel session (0) on SSHService ssh-
   connection on HoneyPotTransport,416,[...]] CMD: wget http
   ://[...]:42211/Linux2.6
2016-07-18 16:00:12+0200 [SSHChannel session (0) on SSHService ssh-
   connection on HoneyPotTransport,416,[...]] Command found: wget
   http://[...]:42211/Linux2.6
```

Listing 3.4: Auszug aus Kippo Logdatei; IP Adressen entfernt

3.1.8. Honeyd

Honeyd ist ein High-Interaction Honeypot Framework, mit dem virtuelle Honeypots und Dienste erstellt werden können. Die virtuellen Honeypots werden über eine Konfigurationsdatei erstellt, in der Eigenschaften wie Betriebssystem, Laufzeit und angebotene Dienste festgelegt werden können [vgl. PH07, S. 105 ff.] [HD, vgl.].

Honeyd zeichnet den Netzwerkverkehr (bei verbindungsorientierten Protokollen nur Beginn und Ende) und die durch die Dienste erzeugten Daten auf. Letztere können sich von Dienst zu Dienst unterscheiden [vgl. PH07, S. 133].

Tabelle 3.1.: Honeypotsoftware

Name	Integration	Artefakte	Formate	Protokolle
Mailoney	Niedrig	Anmeldedaten, E-Mails	Text	SMTP
SHIVA	Niedrig	E-Mails, Anhänge	SQL, HPFeeds	SMTP
Glastopf	Niedrig	verschiedene Webserver Angriffe	Text, HPFeeds, SQL, syslog, E-Mail, Taxii, Logstash	HTTP
Heralding	Niedrig	Anmeldedaten	Text	FTP, HTTP, POP, SMTP, SSH, TELNET
dionaea	Niedrig	Schadsoftware, Dienstlogdateien	Text, JSON, SQL	SMB, HTTP, FTP, MSSQL, MySQL, UPNP, memcache und weitere
KFSensor	Niedrig	Dienstlogdateien	SQL	HTTP, SMTP, SOCKS, NetBios, SMB, CIFS, MSSQL und weitere
Kippo/Cowrie	Mittel	Anmeldedaten, Konsolen-Befehle, Schadsoftware	Text, (Cowrie: JSON, HPFeeds)	SSH
Honeyd	Hoch	Netzwerkverkehr und Dienstlogdateien	Text	SSH, MySQL, HTTP, IMAP, TELNET und weitere

3.1.9. Zusammenfassung

Tabelle 3.1 zeigt die Übersicht der vorgestellten Honeypotsoftwares. Aus der Übersicht wird ersichtlich, dass fast alle Honeypotsoftwares ihre Daten in Textdateien ablegen. Des Weiteren sind Datenbanken und HPFeed häufig integriert.

3.2. Analyse

In diesem Abschnitt wird Software betrachtet, die die von Honeypots erstellten Daten analysieren bzw. anreichen können.

3.2.1. p0f

P0f ist eine Software zum passiven Fingerprinting, womit der Ursprung von Netzwerkverbindungen in Form von Betriebssystem, Client und Verbindung erkannt werden können. P0f kann dabei permanent an eine Netzwerkschnittstelle laufen oder eine mit libpcap aufgezeichnete Sitzung analysieren [Zal].

3.2.2. Cuckoo

Cuckoo ist eine Sandbox-Software zur Analyse von Schadsoftware [CUC, vgl.]. Cuckoo kann während der Analyse mehrere Module durchlaufen, um Informationen zu gewinnen. Viele dieser Module binden Drittsoftwares in Cuckoo ein. Die folgende Liste ist ein Ausschnitt der verfügbaren Module.

- dropped: Liste von erstellten Dateien
- network: Übersicht der entstandenen Netzwerkverbindungen
- screenshots: Bildschirmfotos während der Ausführung
- snort: Intrusion Detection System[1]
- surica: Intrusion Detection System[2]
- targetinfo: Allgemeine Informationen zur Datei
- virustotal: Online Virus und Malwarescanner[3]
- volatility: Speicheranalyse[4]

3.2.3. Logstash

Logstash ist eine Software zur "Datensammlung, Anreicherung und Übertragung" [LS]. Logstash kann eine Vielzahl von Datenformaten lesen und schreiben und zusätzlich mit Filter manipulieren. Besonders interessant sind die Filter date, dns, geoip und grok.

- Der Date Filter kann Datumsangaben in ein einheitliches Format umwandeln.

[1] https://snort.org/
[2] https://suricata-ids.org/
[3] https://www.virustotal.com/
[4] http://www.volatilityfoundation.org/

Tabelle 3.2.: Analyse

Name	Eingabe	Ausgabe
p0f	libpcap	Text
Cuckoo	Schadsoftware	ElasticSearch, JSON, Moloch, MongoDB, HTML
Logstash	Ca. 50 Formate[6]	Ca. 50 Formate[7]

- Der DNS Filter kann einen Hostnamen (A/CNAME) oder eine Netzwerkadresse (PTR) auflösen.
- Der GeoIP Filter kann mit Hilfe einer MaxMind GeoIP Datenbank [5] zu einer Netzwerkadresse eine Standort in Form eines GeoJSON ermitteln.
- Der Grok Filter kann Zeichenketten mit Hilfe von regulären Ausdrucken auswerten und eignet sich somit zum Analysieren der vielen Honeypot Logdateien.

Des Weiteren kann das sog. Codec Multiline mehrzeilige Texte zu einem Ereignis zusammenführen. Dies kann genutzt werden, um beispielsweise Mailony's E-Mail Logdatei zu analysieren.

3.2.4. Zusammenfassung

Tabelle 3.2 zeigt die Übersicht der vorgestellten Softwares zur Analyse von Honeypotdaten. Besonders Logstash ist eine nützliche Software, die dazu verwendet werden kann, die Vielzahl von Logdateien strukturiert zu verarbeiten.

3.3. Visualisierung

In dem letzten Abschnitt von Kapitel 3 werden Softwares zum Visualisieren von Honeypotsensordaten vorgestellt. Viele Honeypots haben ihre eigenene Auswertungs- oder Visualisierungsoftwares wie z.B. Kippo Graph [Kon] oder Glastopf Analytics [Vav]. Um ein Gesamtbild aller Honeypots und weitere Sensordaten zu erhalten, werden im Folgenden Softwares vorgestellt, die auf die allgemeine Visualisierung von ereignisbasierten Daten spezialisiert ist.

3.3.1. Grafana

Grafana ist eine Visualisierungssoftware für ereignisbasierte Daten [GF]. Die Software kann Daten von mehreren Quellen laden, unter Anderem aus ElasticSearch und Amazons CloudWatch. Die Software erlaubt es über eine Weboberfläche Dashboards mit Diagrammen zu

[5] http://dev.maxmind.com/geoip/legacy/geolite/

Tabelle 3.3.: Visualisierung

Name	Datenquellen	Darstellungsarten	Besonderheiten
Grafana	Graphite, Elasticsearch, Cloudwatch, Prometheus, InfluxDB, Weitere durch Plugins	Kuchen-, Balken-, Linen- und Punktediagramm, Tabelle, Weltkarte, (Countdown-) Uhr, Einzelkennzahl, Text	
Kibana	ElasticSearch	Kuchen-, Flächen-, Linien und Flächendiagramm, Tabelle, Weltkarte, Einzelkennzahl, Text	Explorativ

erstellen. Jedes Diagramm kann mithilfe datenquellspezifischen Abfragen mit Daten gefüllt werden.

3.3.2. Kibana

Kibana ist eine weitere Visualisierungssoftware [KB]. Im Gegensatz zu Grafana kann Kibana nur Daten aus ElasticSearch verarbeiten, jedoch können mithilfe von dem im vorherigen Abschnitt vorgestellten LogStash eine Vielzahl von Quellen in ElasticSearch imporiert werden. Des Weiteren können die von Kibana verarbeiteten Daten in Echtzeit durchsucht werden. ElasticSearch, LogStash und Kibana werden alle drei von Elastic entwickelt und bilden zusammen den ELK-Stack.

3.3.3. Zusammenfassung

Tabelle 3.3 vergleicht Grafana und Kibana miteinander. Grafana beherrscht von Hause aus mehr Datenquellen als Kibana. Ansonsten unterscheiden sie sich nur geringfügig, was die Darstellungsarten der Daten betrifft. Alleinstellungsmerkmal von Kibana ist die Möglichkeit die Daten explorativ zu erkunden.

4. Auswertung

In diesem Kapitel werden mithilfe von Elasticsearch, Logstash und Kibana (ELK-Stack) Auswertungen und Visualisierungen erzeugt. Der ELK-Stack wurde gewählt, da LogStash ein hervorragende Software zum Normalisieren und Anreichern von verschiedenen Formaten ist und es bereits ein Einstiegsartikel von elastic zu diesem Thema gibt [Bon14]. Aus diesem Artikel wurden hauptsächlich die Grok Muster übernommen und angepasst.

Die Daten wurden von zwei ans Internet angeschlossenen Systemen erzeugt (Honeypot A in Abbildung 2.1). Für eine einfache und nachvollziehbare Einrichtung der Systeme wurde Ansible[1] verwendet. Auf der CD befinden sich die Instruktionen wie die Honeypots erstellt wurden. In Tabelle 4.1 wird eine Übersicht des Aufbaus gezeigt.

Für eine einfachere Auswertung wurde der ELK Stack auf einem separaten lokalem System installiert. Die Daten wurden erst einige Tage gesammelt und anschließend einmal von Logstash in Elasticsearch importiert. Dabei wurden den Ursprung der Angriffe mithilfe des GeoIP Filters erweitert und die Zeitstempel normalisiert.

4.1. Kippo

Die Abbildungen 4.1, 4.2 und 4.3 zeigen Daten, die aus der Kippo-Logdatei stammen.

Abbildung 4.1 zeigt die von den Angreifern verwendeten Authentifizierungsmethoden. Die passwortbasierte Authentifizierung nimmt eine deutliche Mehrheit gegenüber den anderen Methoden ein. Die Public-Key Authentifizierung fehlt gänzlich.

[1] https://www.ansible.com/

Tabelle 4.1.: Aufbau

System	Betriebssystem	Kippo	Mailony	iptables	ELK Stack
Roanapur	Ubuntu 14.04	Installiert		Installiert	
Starlord	Debian 8	Installiert	Installiert	Installiert	
Bebop	Debian 8				Installiert

Abbildung 4.2 zeigt die von den Angreifern verwendeten Benutzernamen. Die Benutzername root und admin wurden in über 95% der Verbindungen benutzt.

Abbildung 4.3 zeigt abschließend die verwendeten Passwörter. Überraschenderweise ist das Passwort !@ das am Häufigsten verwendete, gefolgt von aufsteigenden Zahlenreihen und Standardpasswörtern.

Abbildung 4.1.: Authentifizierungsmethoden aus Kippo

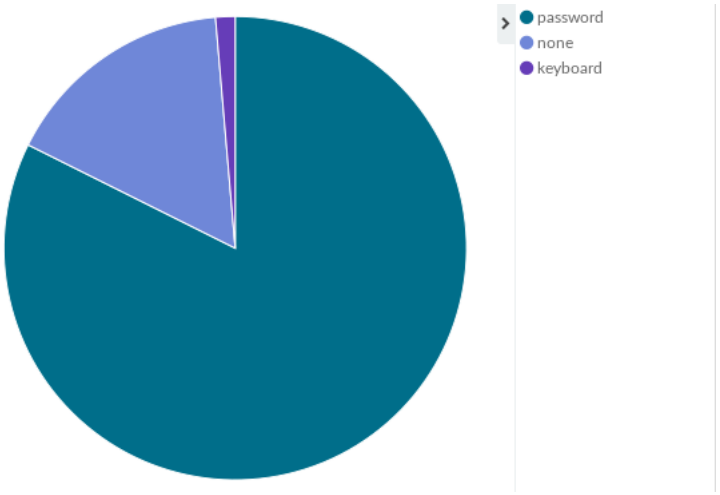


Abbildung 4.2.: Benutzernamen aus Kippo

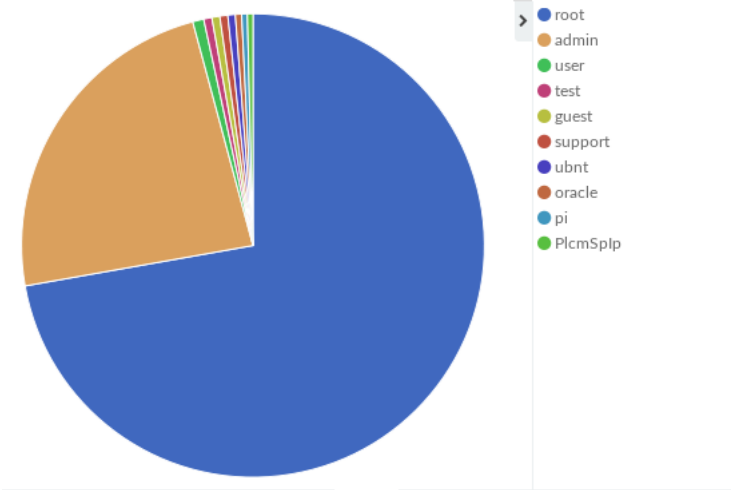
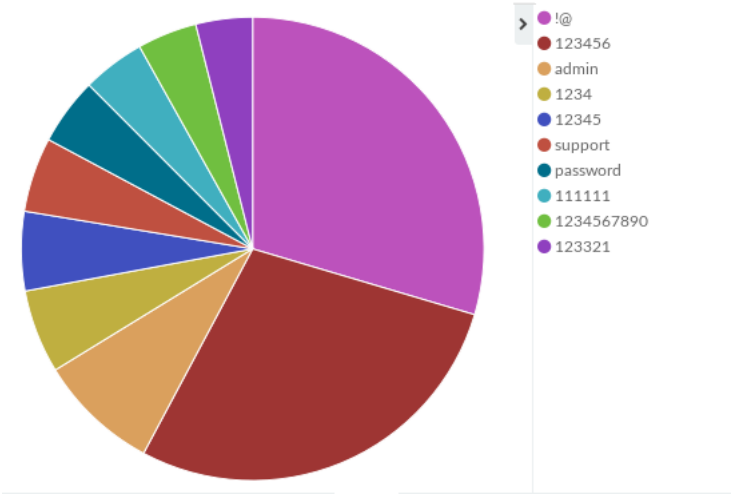


Abbildung 4.3.: Passwörter aus Kippo



4.2. Mailoney

Abbildung 4.4 zeigt E-Mail Adressen an die Spam versandt werden sollte. Die Domain 163.com gehört einer chinesischen Suchmaschine[2].

Abbildung 4.4.: E-Mailziele von Mailoney

E-Mail	Anzahl
xiaonanzi11163@vip.163.com	17
z13699753428@vip.163.com	16
uakh.fpxe@msa.hinet.net	15
xiaonanzi11162@163.com	15
eax_64@yahoo.com	2

4.3. Netzwerk

4.3.1. Ports

Die folgenden Abbildungen zeigen in der Summe alle an den Honeypots aufgerufenen Ports.

Abbildung 4.5 im Speziellen zeigt die Verteilung auf die in RFC 6335 definierten Portbereiche System, Benutzer und Dynamisch [vgl. CIE+11, 6. Port Number Ranges]. Auffällig ist, dass der

[2] https://de.wikipedia.org/wiki/NetEase#163.com

Bereich mit den dynamischen Ports häufiger angefragt wurde, als der Bereich der Benutzer Ports, obwohl dieser viel kleiner ist.

Abbildung 4.6 zeigt die zehn häufigst aufgerufenen Ports. Die Protokolle sind in absteigender Reihenfolge TELNET, NETIS Router, DNS, MS-SQL-S, SIP, MS-V-WORLDS, Unbekannt, HTTP-ALT (z.B. JBoss, Tomcat), HTTP, NDL-AAS [TLM+16, vgl.]. Der Port 53413 wird von Netis Systems Routern verwendet und ist eine Backdoor mit einem Standardpasswort [Kov14, vgl.].

Abbildung 4.5.: Ports nach Bereiche

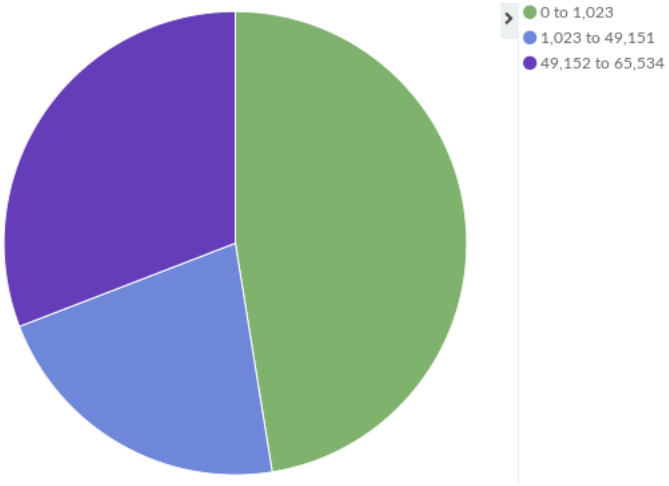
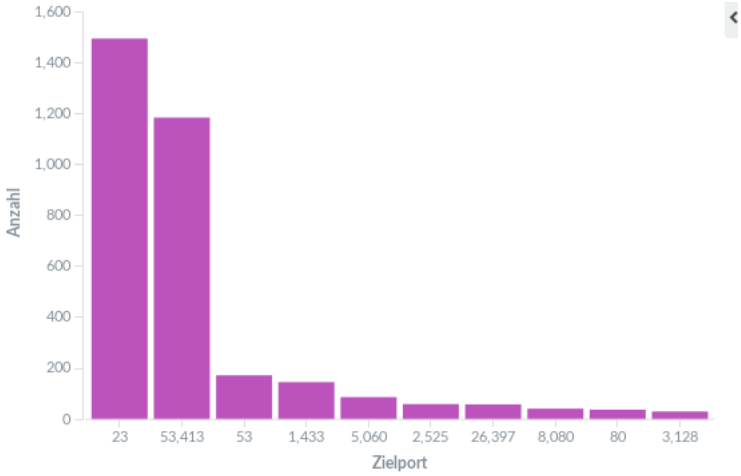


Abbildung 4.6.: Top 10 Ports



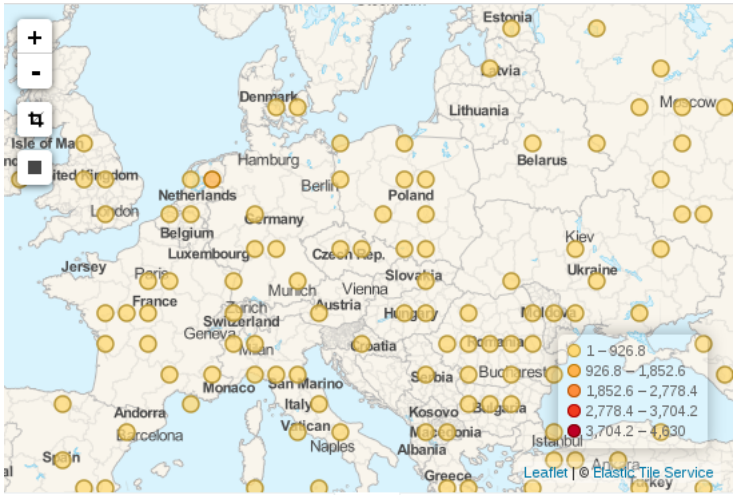
4.3.2. Ursprung

Die folgenden Abbildungen nutzen die Daten, die durch den GeoIp Filter den Quelladressen hinzugefügt wurden.

Abbildung 4.7 zeigt einen Ausschnitt einer Weltkarte, auf der die Angreifer dargestellt werden. Wie aus der Legende zu entnehmen, sind dunklere Punkte Orte aus denen mehrere Angriffe vermutet werden, beispielsweise die Niederlande.

Abbildung 4.8 zeigt das Verhältnis der Angriffe auf die Top 10 der Länder verteilt. Die Mehrheit der Angriffe haben ihren Ursprung in China.

Abbildung 4.7.: Europakarte



4.4. Zusammenfassung

Die letzte Abbildung 4.9 zeigt die Anzahl der vom ELK Stash importierten Ereignisse. Besonders zum Wochenende ist die Anzahl der Ereignisse rapide gestiegen.

Abbildung 4.8.: Top 10 der Länder

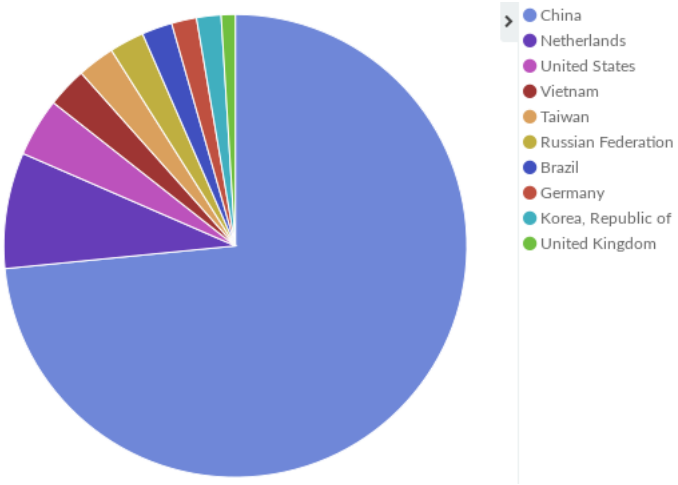
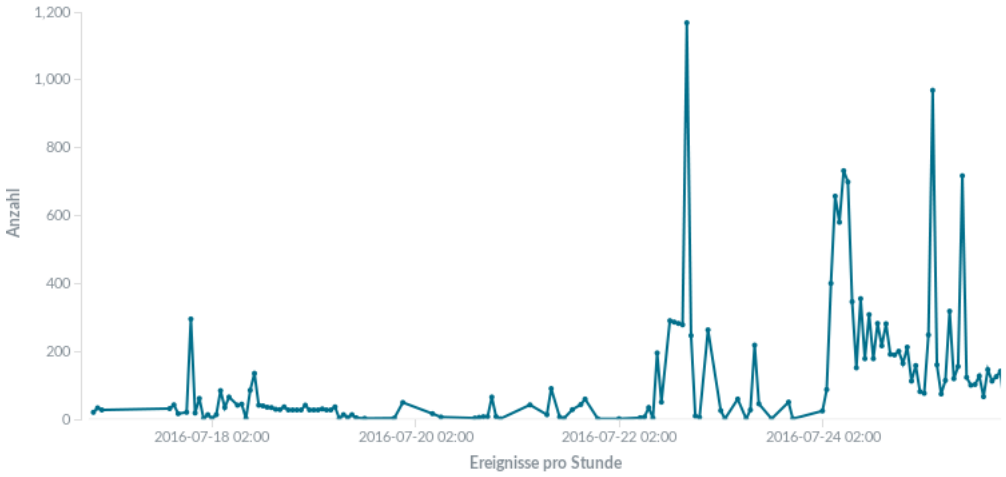


Abbildung 4.9.: Zusammenfassung: Ereignisse pro Stunde



5. Fazit

Ziel dieser Arbeit war es, einen Überblick über aktuelle Honeypotsoftwares zu erstellen und anschließend Auswertungen und Visualisierungen zu erzeugen.

5.1. Zusammenfassung

In den Grundlagen wurde das Wichtigste über Honeypots vermittelt. Insbesondere deren Position in einer Netzwerkinfrastruktur wurde gründlich betrachtet.

Im darauf folgenden 3. Kapitel wurde, unter Berücksichtigung der zum Beginn des Kapitels gemachten Einschränkungen, eine Übersicht aktueller Honeypotsoftwares erstellt. Hinzu kamen weitere Softwares für die Analyse von durch Honeypots erzeugten Daten. Zuletzt wurden zwei Visualisierungssoftwares vorgestellt.

Im abschließenden Kapitel des Hauptteils wurden mit Hilfe von ElasticSearch, LogStash und Kibana verschiedene Diagramme erstellt und ausgewertet. Die dazu verwendeteten Daten wurden zuvor von zwei Honeypots gesammelt.

5.2. Anmerkungen

Während der Auswertung ist aufgefallen, dass die Daten verfälscht worden sind. Der Honeypot Roanapur hat versucht E-Mails zu versenden, was aber aufgrund der Firewalleinstellung fehl schlug. Dies hat ca. 25000 falsche Einträge im Firewalllog erzeugt, sodass im Nachhinein alle ausgehenden Verbindungen von Roanapur aus dem Firewalllog per Hand entfernt wurden. Die Originale befinden sich zusätzlich auf der CD-ROM.

5.3. Ausblick

Die in dieser Arbeit vorgestellten Auswertungen sind nur ein Überblick über die Möglichkeiten, die sich durch ein Netzwerk von Sensoren ergeben. Durch das geordnete Erfassen der Daten ist es möglich diese in Beziehungen zu setzen. Interessante Fragestellungen ergeben sich beispielsweise bei groß angelegten Honeynets. Treten die Angriffe überall auf? Ist eine

Verbreitung zu erkennen? Warum werden z.Z. einige Ports besonders häufig angesprochen. Diese Fragen sind nicht mit der alleinigen Installation von Honeypotsoftwares zu lösen, aber sie helfen enorm. Es existieren mit dem Modern Honey Network Projekt bereits Lösungen, die das Installieren und Verwalten von mehreren Honeypots in eine Software bündeln. Solche Projekte sind ein guter Einstieg in die Thematik.

Kritisch zu betrachten ist hingegen die Tatsache, dass viele Honeypotsoftwares nur als Hobby betrieben werden und somit nicht ihr volles Potenzial entfalten können. Des Weiteren sind viele Honeypotsoftwares einfach verschwunden und teilweise ist nicht mal mehr deren Quellcode zu finden.

A. Anhang

A.1. Inhalt der CD-ROM

1. In der Arbeit verwendete Abbildungen.
2. In der Arbeit verwendete Listings.
3. Die gesammelten Logdateien.
4. Das Ansible Skript zum erzeugen der Umgebung.
5. Diese Arbeit als PDF Datei.

A.2. Honeypots im Netzwerk

- Firewall von cyberscooty https://openclipart.org/detail/171429/firewall
- Server von cyberscooty https://openclipart.org/detail/171426/server
- Internet von witcombem https://openclipart.org/detail/174475/internet
- Honey von nicubunu https://openclipart.org/detail/22169/honey

Literaturverzeichnis

[Ani] Mitchell Anicas. How to map user location with geoip and elk (elasticsearch, logstash, and kibana). https://www.digitalocean.com/community/tutorials/how-to-map-user-location-with-geoip-and-elk-elasticsearch-logstash-and-kibana.

[awh] awhitehatter. Mailoney. https://github.com/awhitehatter/mailoney.

[Bin] Rahul Binjve. Shiva. https://github.com/shiva-spampot/shiva.

[Bon14] Antonio Bonuccelli. Use elk to visualise security data: Iptables and kippossh honeypot. https://www.elastic.co/blog/use-elk-display-security-datasources-iptables-kippo-honeypot, October 2014.

[BS13] Rahul Binjve and Sumit Sharma. Shiva - user manual. https://github.com/shiva-spampot/shiva/raw/1221402402d0408cd6741a3b6d3704ae0664e057/docs/User%20Manual.pdf, September 2013.

[CIE+11] M. Cotton, ICANN, L. Eggert, Nolie, J. Touch, USC/ISI, M. Westerlund, Erricsson, S. Cheshire, and Apple. Service name and port number procedures. https://tools.ietf.org/html/rfc6335, August 2011.

[CUC] Cuckoo sandbox. https://cuckoosandbox.org/.

[Dig07] DigitalNinja. Fuzzy clarity: Using fuzzy hashing techniques to identify malicious code. http://www.shadowserver.org/wiki/uploads/Information/FuzzyHashing.pdf, April 2007.

[ENI15] *Standards and tools for exchange and processing of actionable information.* European Union Agency for Network and Information Security, January 2015.

[GF] Grafana. http://grafana.org/.

[Gri13] Roger Grimes. No honeypot? don't bother calling yourself a security pro. http://www.infoworld.com/article/2614083/, April 2013.

[HD] Honeyd. https://github.com/DataSoft/Honeyd/.

[HN] The honeynet project - projects. http://honeynet.org/project.

[ITG08] *BSI-Standard 100-2: IT-Grundschutz-Vorgehensweise.* 2008.

[KB] Kibana. https://www.elastic.co/products/kibana.

[KF] Kfsensor. http://www.keyfocus.net/kfsensor/.

[Kon] Ioannis Koniaris. Kippo-graph. https://github.com/ikoniaris/kippo-graph.

[Kov14] Eduard Kovacs. Easily exploitable vulnerability found in netis routers. http://www.securityweek.com/easily-exploitable-vulnerability-found-netis-routers, August 2014.

[LS] Logstash. https://www.elastic.co/de/products/logstash.

[Naz15] Jose Nazario. Awesome honeypots. https://github.com/paralax/awesome-honeypots, December 2015.

[Ost] Michel Osterhof. Crowie. http://www.micheloosterhof.com/cowrie.

[PAW] pcap. https://en.wikipedia.org/wiki/Pcap#Programs_that_use_libpcap.2FWinPcap.

[PCA] Tcpdump/libcap. http://www.tcpdump.org/.

[PH07] Niles Provos and Thorsten Holz. *Virtual Honeypots: From Botnet Tracking to Intrusion Detection.* 1 edition, July 2007.

[Ris] Lukas Rist. Glastopf. https://github.com/mushorg/glastopf.

[Sch] Mark Schloesser. hpfeeds. https://github.com/rep/hpfeeds.

[Sei] Phillip Seidel. dionaea. https://www.dinotools.de/en/project/dionaea/index.html.

[Spi02] Lance Spitzner. *Honeypots: Tracking Hackers*. September 2002.

[Ste] Richard Stevens. Honeypots und honeynets. https://www.gi.de/service/informatiklexikon/detailansicht/article/honeypots-und-honeynets.html.

[Tam] Upi Tamminen. Kippo. https://github.com/desaster/kippo/.

[TLM+16] Joe Touch, Eliot Lear, Allison Mankin, Markku Kojo, Kumiko Ono, Martin Stiemerling, Lars Eggert, Alexey Melnikov, Wes Eddy, , Alexander Zimmermann, Allison Mankin, Michael Tuexen, Eddie Kohler, and Yoshifumi Nishida. Service name and transport protocol port number registry. https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.txt, July 2016.

[Vav] Kamil Vavra. Glastopf analytics :: easy honeypot statistics v2.0. https://github.com/vavkamil/Glastopf-Analytics.

[Ves] Johnny Vestergaard. Heralding. https://github.com/johnnykv/heralding.

[Zal] Michal Zalewski. p0f. http://lcamtuf.coredump.cx/p0f3/.

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 26. Juli 2016 Benjamin Kahlau