Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Jannick Asmus

Umsetzung des Threat Hunting Paradigmas zur Identifizierung kompromittierter IoT-Geräte

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Jannick Asmus

Umsetzung des Threat Hunting Paradigmas zur Identifizierung kompromittierter IoT-Geräte

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

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

Betreuender Prüfer: Prof. Dr. Klaus-Peter Kossakowski
Zweitgutachter: Prof. Dr. Ulrike Steffens

Abgegeben am 23.03.2021

Jannick Asmus

Thema der Arbeit

Umsetzung des Threat Hunting Paradigmas zur Identifizierung kompromittierter IoT-Geräte

Stichworte

Threat Hunting, IoT-Geräte, Threat Intelligence, IT-Sicherheit

Kurzzusammenfassung

Diese Arbeit befasst sich mit der Entwicklung eines Verfahrens in der IT-Sicherheit, um vorhandene IoT-Geräte im Netzwerk auf Verhaltensmuster und Anomalien zu untersuchen. Ziel ist es festzustellen, ob sich ein Angreifer mittels möglicher Schwachstellen in vorhandenen IoT-Geräten bereits im Netzwerk befindet und von dort aus mit anderen Geräten interagiert. Zur Umsetzung dieses Verfahrens werden geeignete Lösungen ermittelt, getestet, bewertet und anschließend die bestmögliche Lösung exemplarisch installiert. Mit dieser Lösung wird dann exemplarisch der Ablauf eines Threat-Hunting-Prozesses in einer Testumgebung simuliert.

Jannick Asmus

Title of the paper

Implementation of the Threat Hunting Paradigm to identify compromised iot-devices

Keywords

threat hunting, iot devices, threat intelligence, it security

Abstract
This thesis deals with the development of a procedure in it security to examine existing iot devices in the network for behavioral patterns and anomalies. The goal is to determine whether an attacker is already in the network by means of possible vulnerabilities in existing iot devices and interacts with other devices there. To implement this process, suitable solutions are identified and tested, rated and then the best possible solution is installed as an example. This solution is then used to simulate a threat hunting process in a test environment.

Inhaltsverzeichnis

Abbildungsverzeichnis[1]

[1] Tabellen wurden aufgrund ihrer geringen Anzahl in dieser Arbeit zur einfacheren Übersicht als Bilder hinterlegt und somit auch im Abbildungsverzeichnis mit aufgelistet.

Glossar[2]

A

Augmented Reality · Beschreibt die computergestützte Erweiterung der Realitätswahrnehmung.

B

Big Data · Beschreibt die Verarbeitung von riesigen Datenmengen, bei denen herkömmliche Methoden nicht ausreichen.

Bruteforce · Grobes Ausprobieren aller Möglichkeiten (z.B. um Passwörter zu knacken).

C

CERT · Ist ein Computer-Emergency-Response-Team zur Bekämpfung von konkreten Sicherheitsvorfällen. Sie warnen zudem vor Sicherheitslücken und arbeiten an konkreten Lösungen für Sicherheitsvorfälle oder präventiven Lösungsansätzen für Bedrohungen mit.

Checksums · Steht für Prüfsummen, mit der die Integrität von Daten überprüft werden kann. Bad checksums stehen für veränderte Daten.

[2] Aus Gründen der Leserlichkeit wurde an dieser Stelle das Abkürzungsverzeichnis in das Glossar inkludiert.

Cloud Computing · Ist eine IT-Infrastruktur, die z.B. über das Internet verfügbar ist und Ressourcen wie Speicherplatz, Rechenleistung oder Anwendungen bereitstellt.

CoAP · Das Constrained Application Protocol ist ein spezielles Machine to Machine Web-Transfer-Protokoll für IoT-Geräte.

Command-and-Control-Struktur · Im Zusammenhang aus dem Beispiel aus Punkt 2.1.2 ist damit eine Struktur mit einem zentralen Server gemeint, der Befehle an ein Botnetz sendet und Berichte von diesem empfängt.

CronJob · Zusammengesetztes Wort aus Cron-System und einer festgelegten Aufgabe. Der Cron-Dienst startet automatisch Skripte und Programme zu einer konfigurierten Uhrzeit.

Cryptomining · Errechnen virtueller Geldwährungen zulasten von Computerressourcen.

CTI · Steht für Cyber Threat Intelligence, welche unterstützende Informationen über Bedrohungen für die Informationssicherheit liefert. Sie unterstützt Menschen dabei zukünftige Situationen vorherzusagen, oder Entscheidungen zu treffen.

D

Dark Web · Verborgener Teil des Internets, der z.B. mit spezieller Software wie Tor zu erreichen ist.

DDOS-Attacken · Sind sogenannte Distributed-Denial-of-Service-Angriffe, die einen Dienst lahmlegen oder stark einschränken.

DICOM · Digital Imaging and Communications in Medicine ist ein offener Standard zum Austausch von Informationen im medizinischen Bereich, wie z.B. Bildern.

DNS Amplification Attacks · Sind extrem große Antworten auf eine DNS-Anfrage eines Angreifers bei einem DNS Server. Diese Antworten werden an das System des Opfers gesandt, um dessen Dienste/Verfügbarkeit lahmzulegen.

DNS Tunneling · Beim DNS Tunneling nutzen Angreifer das DNS-Protokoll, um heimlich mit dem Zielrechner zu kommunizieren und z.B. Daten abzugreifen, oder Befehle zu übermitteln.

DNS-Zonentransfer · Übertragung von Zonen im Domain Name System. Dient z.B. zur Datensynchronisation mehrerer DNS Server, um Ausfällen durch Redundanz vorzubeugen.

Docker · ist eine freie Software zur Isolierung von Anwendungen mit Hilfe von Containervirtualisierung.

E

Exploits · Systematische Möglichkeit, um Schwachstellen einer Software auszunutzen. Beispielhaft sind hier aus Abbildung 2.4 ungewollte Netzwerkscans, remote ausgeführte Befehle z.B. über eine Schwachstelle einer Software (Remote Code Execution), oder mittels des Zielbetriebssystems (Command Injection). Weitere Möglichkeiten sind das Ausnutzen von Pufferüberläufen in Programmen (Buffer Overflow), z.B. zur Datenmanipulation, das Ausnutzen von Sicherheitslücken im SQL-Datenbankbereich (SQL Injection) sowie Zero Day Exploits (die bereits separat im Glossar behandelt werden).

F

Firmware · Eingebettete Software eines Gerätes, die grundlegende Funktionen leistet.

Forensischer-Cyber-Threat-Intelligence-Report · Cyber-Threat-Intelligence-Bericht mit methodischem Analyseansatz.

Forward and Reverse Shell · Bei Forward-Shell-Zugriffen für den Remote-Zugriff verbindet sich ein Client mit einem Server über einen bestimmten Port. Bei Reverse-Shell-Zugriffen ist das umgekehrt. Um eingehende Firewall-Portbeschränkungen im Ziel-Netzwerk zu umgehen wird der Server zum Client und der Client zum Server, wobei dieser auf eingehende Verbindungen des Clients wartet. Dies funktioniert, da Firewalls oft nur den eingehenden Verkehr beschränken.

H

Hacktivists · Personen oder Gruppen, die politische und ideologische Ziele mithilfe von Computern und Rechnernetzen erreichen wollen.

Hash-Wert · Bezeichnet die Umwandlung einer Zeichenfolge in einen kürzeren, numerischen Wert mit einer festen Länge. Dient z.B. zur schnelleren Suche in Datenbanken oder zur Sicherstellung der Integrität von Daten.

Heartbleed Bug · Ist ein Programmfehler in älteren Versionen von OpenSSL, durch den private Daten aus verschlüsselten TLS-Verbindungen z.B. zwischen Client und Server ausgelesen werden können.

I

ICMP · Das Internet Control Message Protocol dient dem Austausch von Informations- und Fehlermeldungen mittels IPv4-Protokoll.

Insider-Angriff · Bedrohung aus dem Inneren, also z.B. Mitarbeiter oder interne Personen mit Zugriff auf Vermögenswerte eines Unternehmens.

Intrusion-Detection-System · Sicherheitssystem zur Erkennung von Angriffen, die auf Computer, Server und Netzwerke gerichtet sind.

IoA · Indicators of Attack sind Indizien dafür, dass sich jemand bereits Zugang zum Netzwerk verschafft hat, oder gerade dabei ist.

IoC · Indicators of Compromise sind Hinweise dafür, dass sich jemand Zugang zum Netzwerk verschafft hat und möglicherweise bereits Schaden verursacht hat.

K

Kompromittierung · Unberechtigtes Eindringen in ein Computersystem mit schädlichen Folgen, wie z.B die Verletzung der Integrität von Daten.

Kundentelemetrie · Übertragene Messwertdaten von Kunden für den weiteren Erkenntnisgewinn.

L

Load Balancer · Dienen als Lastverteiler für Anfragen oder umfangreiche Berechnungen auf verschiedene parallel arbeitende Systeme, um z.B. die Performance oder Verfügbarkeit von Servern zu verbessern.

M

M2M Communication · Beschreibt die Kommunikation im zwischenmaschinellen Bereich, also von Maschine zu Maschine.

Malware · Ist im Kontext von Abbildung 2.4 z.B. Schadsoftware wie Ransomware (also Erpressungstrojaner), Backdoortrojaner (verschaffen Zugriff auf sonst gesicherte Computer oder Programme) oder Botnetze, also Gruppen von fremdgesteuerten Rechnern. Ein weiteres Beispiel sind Würmer, also selbst vervielfältigende Schadprogramme.

Managed Switch · Sind selber verwaltbare Switches, die mehr Einstellungsmöglichkeiten für das Netzwerk bieten als Unmanaged Switches.

MITRE ATT&CK · Steht für Adversarial Tactics, Techniques & Common Knowledge und bezeichnet eine offene, kostenlose Wissensdatenbank, die eine systematische Kategorisierung von möglichen Gegenspieler-Verhaltensmustern auf der Grundlage von direkten Beobachtungen erstellt.

MQTT · Message Queuing Telemetry Transport ist ein offenes Netzwerkprotokoll zur Machine to Machine Kommunikation.

O

Open Source · Unter einer speziellen Lizenz veröffentlichter quelloffener Code, der von jedem zugänglich und nach Belieben veränderbar und teilbar ist.

P

P2P C2 · Kommunikation zum Command Server über andere Clients des Botnetzes.

Patch Level · Versionsnummer von Software.

Phishing-Angriff · Abgreifen persönlicher Daten eines Nutzers durch gefälschte Webseiten, E-Mails, Nachrichten etc.

Port Scanner · Software mit der überprüft werden kann, welche Dienste ein mit UDP/TCP arbeitendes System über das Internetprotokoll anbietet.

Predictive Analytics · Verwendung historischer Daten, um zukünftige Ereignisse vorherzusagen.

Protokoll · In diesem Kontext sind Netzwerkprotokolle, also bestimmte Abfolgen und Standards in der Kommunikation zwischen verschiedenen Systemen gemeint.

Prototyp · Vereinfachtes Versuchsmodell eines geplanten Produktes.

R

Ransomware · Auch Erpressungssoftware genannt, sind Schadprogramme, die dem Nutzer den Zugriff auf die eigenen Daten, Programme oder dessen Nutzung einschränken bzw. verhindern.

Registry · Registrierungsdatenbanken sind vor allem unter Windows bekannt. Sie speichern beispielsweise Konfigurationen eines oder mehrerer Geräte in einer Datenbank.

Reverse Hostname · DNS-Abfrage zur Ermittlung des Hostnames durch dessen IP-Adresse.

RFC · Steht für Request for Comments und bildet eine Reihe technischer und organisatorischer Dokumente zum Internet. Ein Großteil der im Internet verwendeten Standards ist in RFCs veröffentlicht.

RFID-Technologie · Radio Frequency Identification ist eine Technologie von Sender-Empfängersystemen für den berührungslosen Datenaustausch, z.B. zur Identifikation und Lokalisation von Objekten und Lebewesen durch Radiowellen.

Rohdaten · Unbearbeitete, unmittelbar gewonnene Primärdaten einer Messung, Beobachtung oder Datenerhebung.

S

SIEM · Security Information and Event Management ist ein System zur Verwaltung von Logfiles und Meldungen vieler verschiedener Geräte.

Smart Robots/Home · Beschreibt die Vernetzung von Gebäudetechnik und deren zentraler Steuerung im Heimbereich.

SOC · Ein Security Operations Center bildet die Zentrale einer IT-Infrastruktur, die sich um dessen Schutz kümmert. Kombiniert spezialisierte Analysten, Prozesse und Tools, um zielgerichtet Bedrohungen für die IT-Infrastruktur zu identifizieren und entsprechend darauf zu reagieren.

Social Engineering · Manipulation, bzw. Beeinflussung von Menschen zur Herausgabe von vertraulichen Informationen.

SSL/TLS Handshake · Bezeichnet den Verbindungsaufbau zwischen Client und Server für eine verschlüsselte Kommunikation.

Standardgateway · Bezeichnung für einen Router in einem IP-Netzwerk, der sich um IP-Pakete kümmert, für die keine anderen Routing-Informationen gefunden wurden.

SYN Flag · Teil des TCP-Protokolls, das gesetzt wird, wenn eine Verbindung zwischen Sender und Empfänger aufgebaut werden soll.

T

TCP-Protokoll · Transmission Control Protocol heißt auf Deutsch Übertragungssteuerungsprotokoll. Dabei handelt es sich um ein verbindungsorientiertes Protokoll, welches Datenverluste verhindern soll.

Team Blue · Verbessert und verteidigt seine IT-Infrastruktur permanent gegen Team Red und echte Angreifer.

Team Red · Simuliert Angriffe auf die IT-Infrastruktur von Team Blue, um dessen Sicherheit und Abwehrmechanismen zu stärken.

Threat Hunting · Befasst sich mit der Suche nach Schwachstellen und abnormalen Aktivitäten im Netzwerk, die Anzeichen für eine Kompromittierung, einen Angriff, oder einen Datendiebstahl sein können.

Threat Intelligence Feed · Kontinuierlich aktualisierter Datenstrom über potenzielle Gefahren für die Sicherheit einer Organisation.

Threat-Intelligence-Plattform · Sichert den Informationsaustausch über Bedrohungsinformationen. Sie unterstützt bei der Analyse und Verarbeitung der Daten sowie der Erstellung von IoC's.
TLS · Transport Layer Security ist ein Protokoll zur verschlüsselten Datenübertragung im Internet.
Tor (Netzwerk) · Ist ein Netzwerk zur Anonymisierung von Verbindungsdaten.
Traceroute · Ist ein Programm, das ermittelt, über welche Router und Internet-Knoten IP-Datenpakete bis zum abgefragten System gelangen.
Trojanisches Pferd · Metapher für einen Angreifer von innen heraus.
TTP · Steht für Tactics, Techniques und Procedures. Gemeint sind in dem Angreifer-Kontext die Skillsets eines Angreifers (Techniken) und wie sie angewandt werden (Taktiken). Zu den Prozeduren gehören die Details, wie ein Angreifer die Techniken einsetzen wird, um sein Ziel zu erreichen.

U

UEBA · User and Entity Behavior Analytics ist ein Sicherheitsprozess zur Erkennung und Reaktion auf Angriffe und Bedrohungen für ein Unternehmen, z.B. anhand von verdächtigem Benutzerverhalten.
User Practice Threats · Benutzerbezogene Bedrohungen sind im Kontext von Abbildung 2.4 leicht zu knackende Standardpasswörter, Phishing von persönlichen Informationen, sowie Cryptojacking, also das Errechnen von Crypto-Währungen z.B. auf dem Rechner eines Benutzers.

V

VLAN · Logische Netzwerktrennung eines physischen Netzwerks.

X

X509-Zertifikat · Digitale Datei zur Authentifizierung und Verifizierung der Identität von einer Webseite oder einem Host.

Z

Zero-Day-Angriff · Ausnutzen einer Sicherheitslücke, die den Entwicklern noch unbekannt ist, bzw. für die es noch keinen Patch gibt.

1 Einleitung

1.1 Motivation

Internet of Things (kurz IoT) -Geräte finden seit Jahren stetigen Zuwachs in unseren Haushalten, aber auch teilweise in Firmennetzwerken. Es handelt sich bei ihnen um Geräte und Komponenten, die an ein Netzwerk angeschlossen sind und dabei Daten erfassen, speichern, verarbeiten und übertragen. Dazu gehören z.B. Netzwerk-Kameras, Smart TVs, IP-Steckdosen, Wearables, Sprachassistenten etc.
IoT-Geräte werden fast überall in Netzwerken benutzt, ohne dass die meisten Nutzer überhaupt wissen, was genau dort eigentlich kommuniziert wird und mit wem überhaupt. Dabei wird oft diese mögliche Sicherheitsschwachstelle im Netzwerk nicht als solche wahrgenommen. Durch diese Arbeit soll sich das Verständnis, als auch die Wahrnehmung bezüglich der Sicherheit und der Funktion von IoT-Geräten zur besseren Vorsicht hin verändern.

1.2 Zielsetzung

Bei IoT-Geräten wird oft vergessen, wie unsicher diese Geräte sein können bzw. welches Schadenspotenzial sie dabei mit sich tragen. Selbst mit einer gut konfigurierten Firewall lässt sich nicht immer sicherstellen, dass sich ein Angreifer nicht schon aufgrund möglicher Schwachstellen vorhandener IoT-Geräte im Netzwerk befindet und dieses beeinflussen kann. Dabei könnte der Angreifer jene Geräte nutzen, um im Netzwerk mit anderen Geräten zu interagieren und diese ebenfalls zu kompromittieren.

Diese Arbeit hat das Ziel, ein Verfahren vorzuschlagen, mit dem Abweichungen vom normalen Verhalten (Anomalien) oder schädliches Verhalten von

IoT-Geräten im lokalen Netzwerk erkannt werden können. Erkannte Ereignisse weisen möglicherweise auf Kompromittierungen eben dieser IoT-Geräte hin, die daraufhin weiter untersucht und gegebenenfalls vom Netz genommen werden sollten. Während die Definition der erkannten Ereignisse sowie die Sicherstellung der Erkennung den Hauptteil der Arbeit bilden, wird die Nachsorge - also die Untersuchung der möglicherweise kompromittierten Systeme sowie ggf. deren Säuberung - für diese Arbeit ausgeklammert.

Für das Verfahren werden softwarebasierte Lösungen gemäß den Anforderungen an die Erkennung und den geplanten Einsatz in lokalen Netzwerken untersucht und miteinander verglichen. Die gewählte Softwarelösung wird anschließend aufgesetzt und konfiguriert, so dass ein einsatzfähiger Prototyp entsteht.

Dieser Prototyp wird in einer ebenfalls im Rahmen der Arbeit zu erstellenden Testumgebung getestet. Die Testumgebung wird hierbei aus einem Standardgateway, einem Server für die Detektion sowie den IoT-Geräten, die überwacht werden sollen, bestehen.

Dabei liegt der Fokus auf der Softwarelösung an sich und nicht auf der dabei verwendeten Hardware. Ob die Lösung später virtuell oder physisch auf einem System läuft, spielt dabei eine untergeordnete Rolle. Die geeignete/n Softwarelösungen werden dabei gemäß den Anforderungen für diese Aufgabe recherchiert, getestet und anschließend miteinander verglichen. Die genauen Anforderungen werden im Abschnitt „Anforderungsanalyse“ noch genauer hinsichtlich Kriterien und deren Priorität verdeutlicht.

1.3 Zielgruppe

Zielgruppe dieser Arbeit sind vorzugsweise IT-Fachkräfte im Sicherheitsbereich, sowie interessierte Fachkundige, die ihre Sichtweise gegenüber IoT-Geräten verändern wollen und möglicherweise das hier erstellte Verfahren in irgendeiner sinnvollen Art und Weise für sich verwenden. Beispielhaft dafür

wäre z.B. die Nutzung des Verfahrens als Grundlage für ein Threat-Hunting-System oder für Schulungen von Fachpersonal.

1.4 Abgrenzung

Es gibt sehr viele verschiedene IoT-Geräte, darunter auch weit komplexere Systeme im Industrie-Bereich als die hier genannten Geräte im Heimnetzwerk-Bereich. Die industriellen Geräte werden nicht Teil der Arbeit sein. Der Fokus dieser Arbeit liegt auf maximal drei der oben genannten Gerätetypen, die größtenteils im privaten Umfeld laufen. Eine mögliche Übertragung der Ergebnisse dieser Arbeit auf industrielle IoT-Geräte schließe ich aber nicht aus.

Eine Verbesserung der Sicherheit des Heim-Netzwerks vor oder nach einem Angriff auf ein IoT-Gerät ist nicht Teil dieser Bachelorarbeit.

1.5 Struktur der Arbeit

Kapitel 1 - Einleitung

In Kapitel eins wird die Arbeit thementechnisch eingeführt, sowie die vorliegende Problemstellung und deren mögliche/n Lösung/en erläutert.

Kapitel 2 - Grundlagen

Kapitel zwei befasst sich mit den technischen Grundlagen rund um IoT-Geräte, Threat Intelligence sowie dem damit verbundenen Threat Hunting. Zudem werden Konzepte und notwendige Begriffe erläutert.

Kapitel 3 - Anforderungsanalyse

In Kapitel drei werden die genauen Anforderungen und Rahmenbedingungen an die Lösung des Problems gestellt. Dabei werden auch die wichtigsten Kriterien herauskristallisiert, welche die Lösung erfüllen sollte, um der Aufgabe dieser Arbeit gerecht zu werden.

Kapitel 4 - Untersuchung möglicher Lösungen

In Kapitel vier werden vier verschiedene Best-Practice-Lösungen, die im Vorfeld ermittelt und testweise installiert wurden miteinander verglichen, in einer Entscheidungsmatrix bewertet und anhand der Kriterien die passendste Lösung ermittelt. Die Kriterien entstammen dabei aus der Anforderungsanalyse in Kapitel 3.

Kapitel 5 - Exemplarischer Einsatz der Ziellösung

In Kapitel fünf werden zunächst die genauen Systemanforderungen für die Testumgebung festgelegt. Anschließend wird der genaue Aufbau der Testumgebung und die Installation der ausgewählten Ziellösung aus Kapitel 4 aufgezeigt. Bevor es an das Threat Hunting im Netzwerk geht, werden zum besseren Verständnis der Software noch einmal die Besonderheiten und Funktionen der gewählten Lösung vorgestellt. Die aus dem Threat Hunting resultierenden Ergebnisse werden zu guter Letzt noch bewertet.

Kapitel 6 - Fazit

Kapitel sechs beginnt mit meiner Zusammenfassung bezüglich der Arbeit und der vorgestellten Lösungen. Zudem erfolgt ein Rückblick über das Projekt, indem auch auf Schwerpunkte und Schwierigkeiten, sowie im Laufe des Projektes angeeignetes Wissen eingegangen wird. Im Fazit wird betrachtet, ob die Aufgabenstellung dieser Arbeit erreicht wurde. Als letzter Punkt wird ein Ausblick über weitere hier nicht abgebildete Möglichkeiten und Wege im Threat Hunting basierend auf den Erkenntnissen aus diesem Projekt gegeben.

2 Grundlagen

2.1 Internet of Things[3]

Internet of Things bezeichnet Geräte oder "physische Dinge" die mit dem Internet oder einer anderen Art Netzwerk verbunden sind und dabei selbstständig Daten erfassen, speichern und verarbeiten. Sinn und Aufgabe ist es den Menschen in seinen Aufgaben zu ergänzen oder gar zu ersetzen (vgl. Arndt Borgmeier et al. 2017, S. 5).

Ein Beispiel aus der Praxis ist z.B. der Kühlschrank, der erkennt, dass die Milch aufgebraucht ist und daraufhin selbstständig Milch bei einem Onlinehändler nachbestellt (vgl. Patrick-Benjamin Bök et al. 2020, S. 321).

Dabei gibt es noch unendlich andere Anwendungsmöglichkeiten, von automatischen Lampen die ausgehen, wenn jemand das Haus verlässt, oder Heizungen die dann automatisch herunterregeln bis zu Smart Buildings, die automatisch Kaffee & Kekse beim Empfang bestellen wenn die Gäste über einen Smart-Teppich vor einem Meetingraum laufen (vgl. Tanja Ulmen 2019).

Der Begriff geht vermutlich auf Kevin Ashton zurück, der das Wort in seiner Präsentation bei Procter & Gamble im Jahre 1999 zum Thema RFID-Technologie benutzte. Ihm ging es damals darum, dass Maschinen lernen müssen selbstständig Daten zu erfassen, zu verarbeiten und damit wirtschaftliche Verluste zu senken, sowie unnötige Kosten zu vermeiden (vgl. Kevin Ashton 2009).

[3] Zur Vereinheitlichung werden in dieser Arbeit Wörter aus dem Englischen, die mit einem deutschen Wort gekoppelt sind, mit einem Bindestrich verbunden.

Eine genaue Definition und Abgrenzung des Begriffes zu anderer Technik besteht bis heute nicht. Die Bundesregierung hat deshalb im Jahr 2012 ihre eigene Definition von dem Begriff erstellt, die wie folgt lautet: (vgl. Patrick-Benjamin Bök et al. 2020, S. 321).
Das Internet der Dinge ist „die technische Vision, Objekte jeder Art in ein universales digitales Netz zu integrieren“ (Patrick-Benjamin Bök et al. 2020, S. 321)

Damals noch Zukunftsmusik, werden mittlerweile bis zu 75 Milliarden vernetzte IoT-Geräte bis 2025 prognostiziert. Etwa dreimal so viele wie 2019 (vgl. Michael Kroker 2019).

Heutzutage wird der Begriff noch viel breiter genutzt und eine Vielzahl an Technologien fallen unter den Begriff wie in Abbildung 2.1 dargestellt (vgl. Klaus Hauptfleisch 2015, zitiert nach Arndt Borgmeier et al. 2017) .

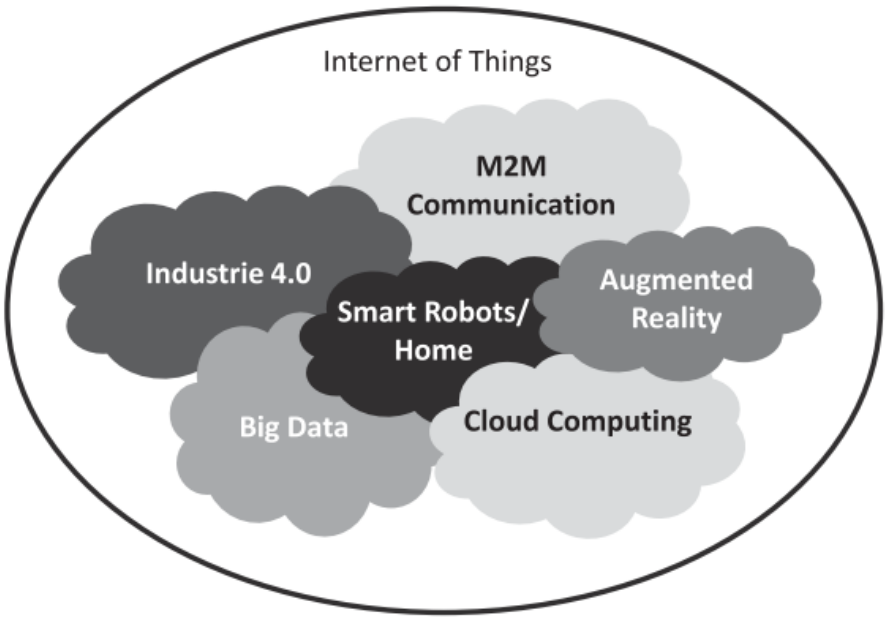


Abbildung 2.1: Aspekte, die dem IoT zugeordnet sind (Klaus Hauptfleisch 2015, zitiert nach (Arndt Borgmeier et al. 2017)

„M2M Communication“ beschreibt, wie bereits aus dem Namen hervorgeht, die Kommunikation im zwischenmaschinellen Bereich, also Maschine zu Maschine.
„Smart Robots/Home“ fällt in den Heimbereich und befasst sich mit der Vernetzung von Gebäudetechnik und deren zentraler Steuerung. Die bei der Vernetzung und Überwachung entstandenen Daten werden wiederrum in der Cloud (Cloud Computing) gesammelt und können von überall abgerufen werden.
„Big Data“ wiederrum verarbeitet riesige Datenmengen und bereitet diese auf. Der letzte Punkt „Augmented Reality“ deckt den Bereich der computergestützten Erweiterung der Realitätswahrnehmung ab (vgl. Arndt Borgmeier et al. 2017, S. 6).

Wie bereits in der Einleitung erwähnt, wird IoT auch in der Industrie genutzt. Die Bundesregierung nutzt hierfür auch den Begriff „Industrie 4.0“ (vgl. BMWi).
Weiterhin wird die IoT-Industrie auch zukünftig vom Ausbau des 5G Netzes profitieren. Zum einen durch die hohe Verbindungssicherheit, als auch durch die erhöhte Anzahl von vielen tausend möglichen Geräten, die in einer Funkzelle kommunizieren können (vgl. Konstantin Matern 2019).

2.1.1 Aufbau, Funktionsweise und Netzwerk

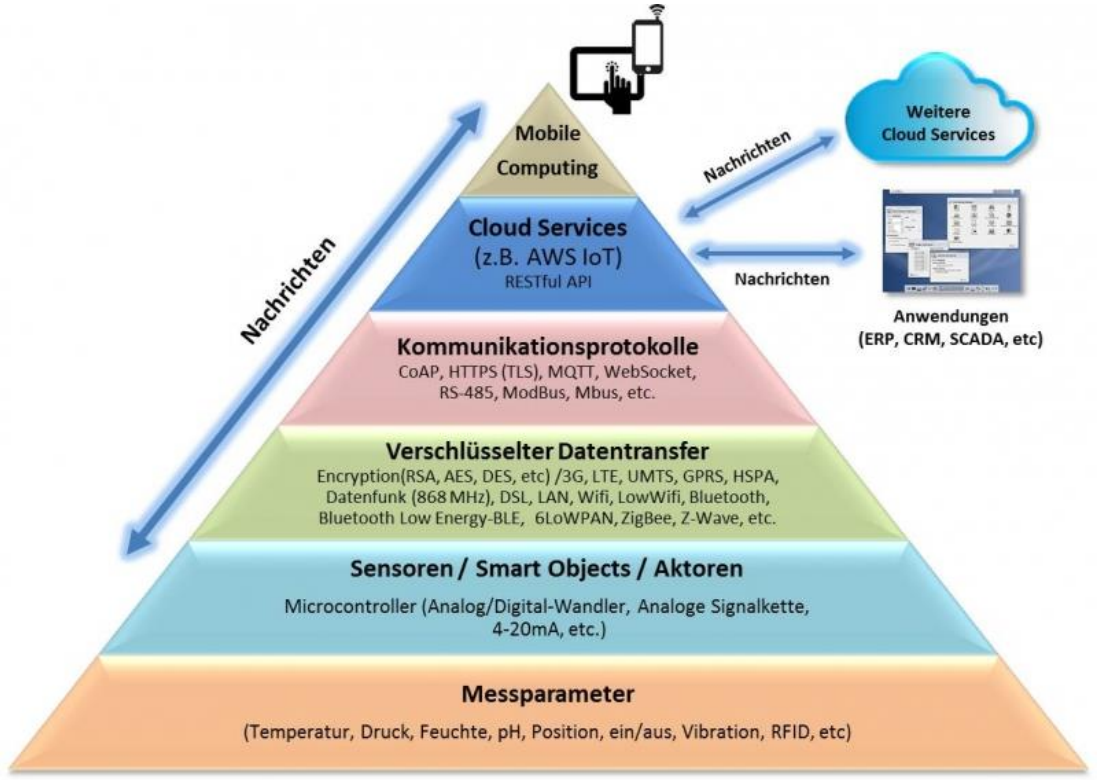


Abbildung 2.2: Aufbau und Kommunikation von IoT-Geräten (Wolfgang Gotscharek 2016)

In Abbildung 2.2 wird die Funktionsweise von IoT-Geräten verdeutlicht. Auf den untersten Ebenen werden Daten mittels Sensoren gemessen und verarbeitet. Dabei werden nicht nur steuernde Aktoren ausgelöst, sondern auch mittels Cloud Services die Daten weiterverarbeitet und aufbereitet, um dem Nutzer z.B. im Rahmen von Predictive Analytics einen Mehrwert zu bieten (vgl. Wolfgang Gotscharek 2016).

Dabei steht genau diese Kommunikation im Netzwerk als auch zu den Cloud Services im Fokus dieser Arbeit.

Da Amazon einen großen Teil der IoT-Branche sowohl im Privaten als auch im Industriellen Segment bedient und eines der im Verlauf dieser Arbeit zu testenden IoT-Geräte von Amazon stammt, wird hier ein Beispiel mit Fokus auf Amazon Webservices (AWS) gegeben. Es stellt einen beispielhaften Ablauf der Kommunikation von IoT-Geräten und AWS Cloud Services dar:

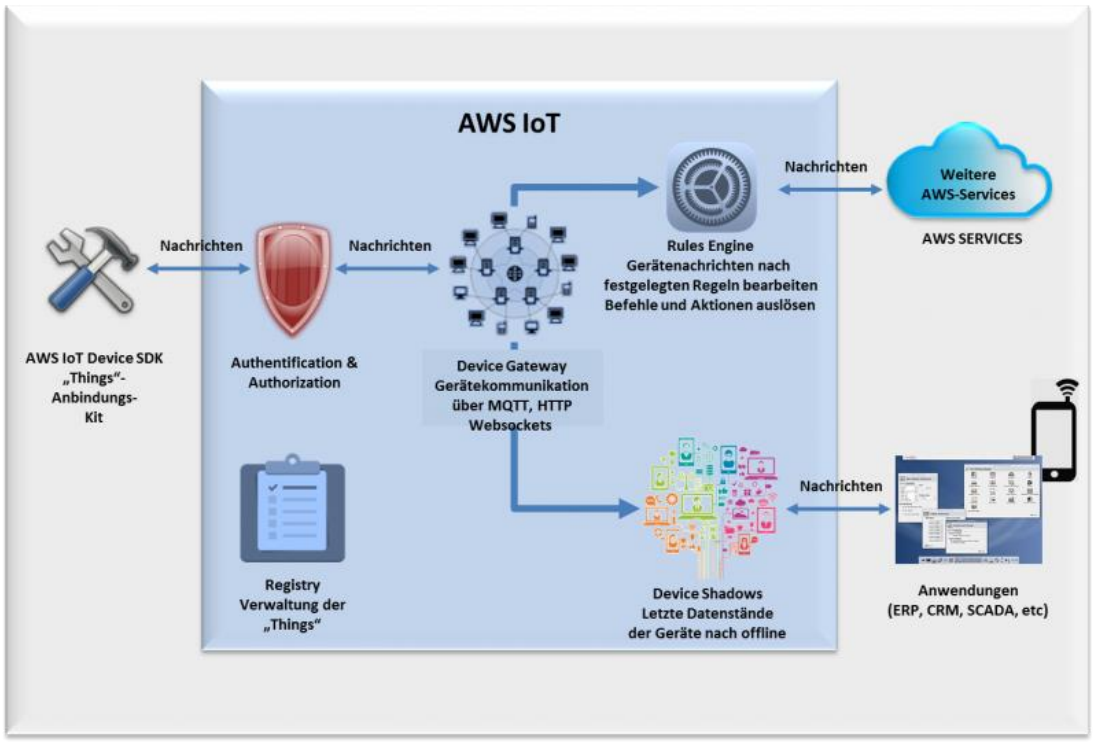


Abbildung 2.3: Amazon AWS IoT (Wolfgang Gotscharek 2016)

Die folgende Beschreibung des obigen Bildes und der untere Fließtext wurden von Gotscharek abgeleitet (vgl. Wolfgang Gotscharek 2016).
AWS IoT ist eine IoT-Plattform, die mit verschiedenen Cloud Services arbeitet. Dazu gehören z.B. AWS Lambda, Amazon API Gateway, Amazon DynamoDB etc.

AWS IoT Device SDK ist ein Software Development Kit mit einer Bibliothek, um Geräte im IoT-Portal zu authentifizieren, zu registrieren und zu verbinden. Zu diesen Kits gehören Microcontroller-Entwicklungsboards, Sensoren,

Aktoren als auch eine Kopie des Software Development Kits. Die Authentifizierung und Autorisierung funktionieren mittels eindeutiger Identität des IoT-Gerätes, z.B. über ein X509-Zertifikat und/oder AWS-Schlüssel. Mittels des Device Gateways mittig im Bild wird zwischen Gerät und AWS IoT kommuniziert. Die Datenübertragung vom IoT-Gerät zum AWS IoT erfolgt dabei verschlüsselt über das TLS Protokoll.
Weiter genutzte Protokolle im privaten IoT-Umfeld sind HTTP, MQTT und CoAP.

In der „Things" Registry werden die eindeutige Kennung und Metadaten wie z.B. Geräteattribute und Funktionen zu dem IoT-Gerät gespeichert. Zu den typischen Metadatenarten gehören Messparameter wie Temperatur, Druck, Position, Hersteller, Firmware-Version, Seriennummer etc.

Device Shadows sind virtuelle Versionen eines IoT-Gerätes, die den Stand der letzten Verbindung gespeichert haben. Um keine Probleme bei Verbindungsabbrüchen zu bekommen, greifen Applikationen wie ERP, CRM etc. über diese Kopien auf die Daten des Gerätes zu. Zu den Daten gehören u.a. Gerätestatus (gemeldeter und gewünschter Status), Metadaten wie z.B. Sensortypen, Client Token (eindeutige ID), eine Dokumentenversion, die jedes Mal hochgezählt wird, sowie Zeitstempel der letzten Nachricht an den AWS.

Die zentrale Komponente im AWS IoT bildet die Rules Engine. Wie Daten erfasst und verarbeitet werden und wie Funktionen ausgeführt werden ist hier festgelegt. Sie bildet also den Motor des Ganzen.

Die AWS Services lassen dabei nicht nur IoT-Geräte von Amazon zu, sondern bieten laut eigener Angabe Unterstützung für Milliarden von Geräten an (vgl. Amazon Web Services, Inc.). Beispielhaft wären z.B. Raspberry Pi-Geräte, IP-Kameras von bestimmten Herstellern, Sensoren, intelligente Haushaltsgeräte etc. (vgl. Amazon Web Services, Inc.). Wie genau die Einbindung dieser Geräte funktioniert, ist im Einzelnen bei Amazon nachzulesen. Vorteile dieser Einbindung sind wie oben genannt, die Verbindung und Verwaltung unzähliger IoT-Geräte mit Cloud-Diensten von Amazon. Das in Abbildung 2.3 gezeigte AWS IoT Device SDK zur Anbindung der Geräte an die AWS Cloud besteht aus

Open-Source-Bibliotheken. Beliebte Alternativen zu dieser Cloud-Struktur sind z.B. Microsoft Azure, Google Cloud, oder die IBM Cloud (vgl. Opslyft 2020). Weiterhin lässt sich mit dem entsprechenden Wissen und der entsprechenden Hardware auch eine eigene Cloud-Struktur errichten. Ob dies immer sinnvoll ist, muss im Einzelfall überprüft werden (vgl. Anand Tamboli 2019).

Die meisten IoT-Geräte nutzen Open Source Linux-Distributionen als Betriebssystem. Vertretene Programmiersprachen für diese Geräte sind im wesentlichen Java, JavaScript, C und Python (vgl. Wolfgang Gotscharek 2016).
Vorteile von Open Source werden in Punkt 3.1 näher erläutert.

2.1.2 Sicherheit

Sicherheitstechnisch gesehen entsprechen IoT-Geräte oft einem trojanischen Pferd im Netzwerk. Sicherheit ist für die Hersteller nur zweitrangig und passt nicht zum Designprozess, sie würde den Entwicklungsprozess bremsen und den Preis erhöhen, weshalb niemand gerne darüber redet (vgl. Steven Feurer und Peter Schmitz 2019).

Aber auch die fortschreitende Datensammelwut der Hersteller und dessen Einfluss auf Konsumentscheidungen stellt ein Problem dar (vgl. Patrick-Benjamin Bök et al. 2020, S. 321–322).

Der nachfolgende Text befasst sich mit der Analyse eines Threat-Intelligence-Teams vom Palo Alto Netzwerk, einem führenden Anbieter von Cybersicherheitslösungen, dessen Analyse von Jürgen Schreier als Dokument zusammengefasst und erweitert wurde (vgl. Jürgen Schreier 2020).

Das Unit 42 Threat-Intelligence-Team vom Palo-Alto Netzwerk analysierte zwischen 2018 und 2019 Sicherheitsvorfälle von über 1,2 Millionen IoT-Geräten mittels deren Sicherheitsprodukt „Zingbox“. Es stellte sich heraus, dass sich die Sicherheitslage bei IoT-Geräten verschlechtert hat, wobei nicht nur moderne Angriffstechniken eine Rolle spielten, sondern auch alte Techniken, die bei vielen IT-Teams schon in Vergessenheit geraten sind.
Erschreckenderweise sind knapp 98 Prozent des gesamten Verkehrs aller IoT-Geräte unverschlüsselt. Ein Beispiel für Angriffe sind Phishing-Attacken, mittels denen sich Angreifer Zutritt zum Netzwerk verschaffen, um anschließend eine Command-and-Control-Struktur z.B. für Botnetze einzurichten. Mittels dieser Struktur wird anschließend der Netzwerkverkehr abgehört und vertrauliche, sowie persönliche Informationen mitgeschnitten. Oft werden diese Daten dann z.B. im Darknet gewinnbringend verkauft. Laut Jürgen Schreier sind 57 Prozent der untersuchten IoT-Geräte anfällig für mittlere und schwere Angriffe, womit IoT-Geräte oft ein „leichter“ zu erreichendes Ziel für Hacker sind. Dies hängt auch mit dem meist niedrigen Patch Level der Geräte zusammen, wobei oft Exploits über bekannte Schwachstellen oder Passwortangriffe wie Bruteforce mit Standard-Passwörtern genutzt werden.

Besonders kritisch wird das Ganze, wenn es sich um Geräte im medizinischen Bereich handelt, bei denen auch die Gesundheit von Patienten auf dem Spiel steht.
Dabei besteht insbesondere in dieser Branche oft eine schlechte Netzwerksicherheitshygiene. 72 Prozent der VLANs im Gesundheitswesen vermischen IoT-Geräte und andere IT-Ressourcen miteinander. Das birgt die Gefahr, dass sich Malware von Computern und anderen Endpunkt-Geräten der Nutzer auf ohnehin schon gefährdete IoT-Geräte im selben Netzwerk verbreiten kann.

Denn IoT-Geräte kommunizieren nicht nur von innen nach außen und vice versa, sondern auch von Ost nach West, also im Netzwerk selbst. Eine Firewall greift in dem Fall also nicht mehr, weshalb es so wichtig ist diesen Netzwerkverkehr zu analysieren und abzusichern, um Malware und andere Angriffe zu erkennen (vgl. Jürgen Hill 2020).

Ein weiteres Problem ist die Entwicklung von Bedrohungen, die neue Techniken wie P2P C2 Kommunikation und wurmartige Funktionen zur Selbstverbreitung im Netzwerk nutzen. Zudem gibt es Geräte, die jahrzehntealte Protokolle wie DICOM nutzen, bei denen Angreifer die Schwachstellen bereits kennen und dadurch in der Lage sind kritische Geschäftsfunktionen zu beeinträchtigen (vgl. Jürgen Schreier 2020).

Aber warum wird so wenig in die Sicherheit dieser Geräte investiert?
Bei der Entwicklung von IoT-Geräten liegt der Schwerpunkt schon oft auf einer kompakten, leichten Bauweise, bei der auch der Prozessor sowie die Speicherkapazität verhältnismäßig schwach ist (vgl. Tanja Ulmen 2019).

Viele Hersteller nehmen Sicherheitslücken in Kauf, um die Entwicklungskosten zu verringern und Produkte schneller auf den Markt zu bringen.
Sicherheit wird für Bequemlichkeit geopfert, was auch daran erkennbar ist, dass die Firmware der Geräte oft von Drittanbietern geliefert wird, ohne Updatemöglichkeit.
Meistens sind die Endanwender auch „sorglos“ unterwegs und Standardpasswörter werden nicht geändert, oder Geräte werden einfach ins Netzwerk

gestellt, ohne sich über die Sicherheit und dessen Folgen Gedanken zu machen (vgl. Bitdefender GmbH 2019).

In Abbildung 2.4 wird die Aufteilung der meistgenutzten IoT-Angriffsarten verdeutlicht, dabei machen Exploits, also Schwachstellenausnutzungen den größten Teil der Angriffe aus.

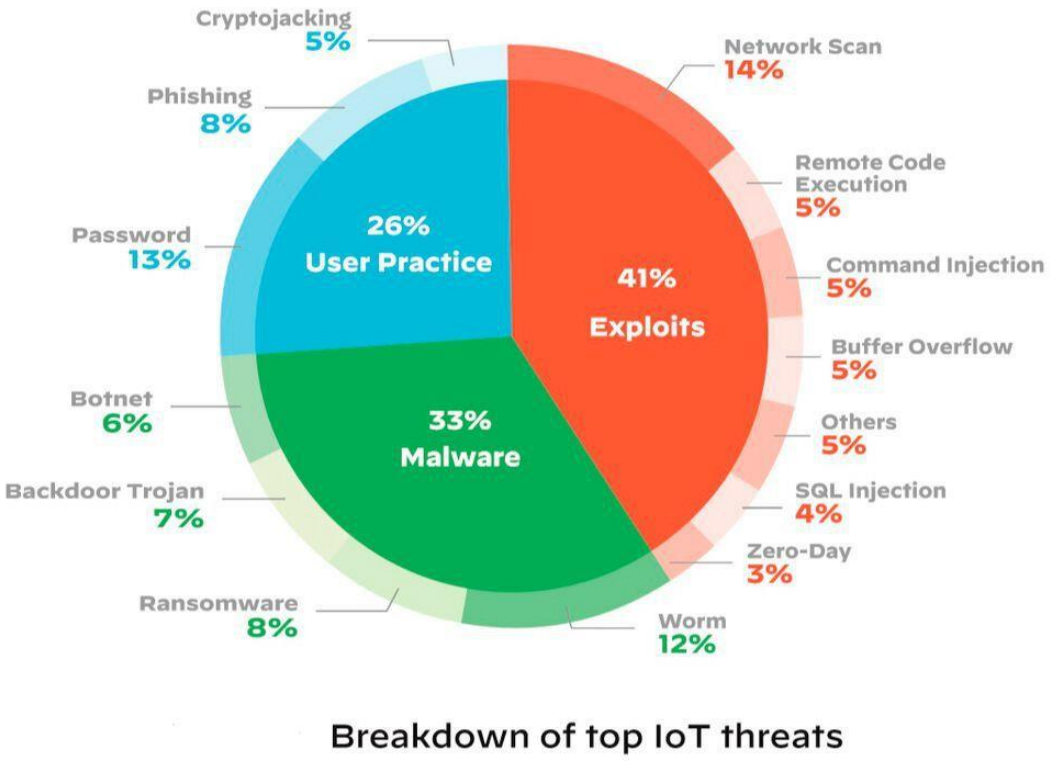


Abbildung 2.4: Aufteilung der IoT-Angriffsarten (Unit42 2020)

2.2 Threat Intelligence

Eine Definition von Threat Intelligence lautet wie folgt:

„Cyber Threat Intelligence (CTI) sind Informationen über feindselige Bedrohungen für die Informationssicherheit, die in einem spezifischen Kontext gesetzt sind und Menschen im Rahmen einer Analyse unterstützen, zukünftige Situationen vorherzusagen oder Entscheidungen zu treffen“ (Dr. Svilen Ivanov 2018).

Eine ausführlichere Erklärung bietet hingegen Egon Kando, die im folgendem grob von mir zusammengefasst wurde (vgl. Egon Kando 2019). Je mehr Wissen über potenzielle Bedrohungen ein Unternehmen besitzt, desto besser kann es sich rüsten. Dafür gibt es die sogenannte Threat Intelligence.
Übersetzt bedeutet das in etwa „Wissen über Bedrohungen“ und beschreibt die Sammlung von sicherheitsrelevanten Informationsquellen.
Das gebündelte Wissen über verschiedenartige Bedrohungen, Bedrohungsgattungen, Akteure, Exploits, Malware, Schwachstellen, Kennzahlen oder Sicherheitsverstöße bildet die Grundlage für ein effektives Sicherheitsprogramm. Threat Intelligence besteht aus verschiedenen Quellen, Plattformen und Feeds, um Unternehmen bei der Erfassung, Analyse und der Reaktion auf Bedrohungen zu unterstützen.
Threat-Intelligence-Lösungen sortieren und analysieren Rohdaten um sie anschließend zu Intelligence-Feeds zusammenzufassen, mit denen gearbeitet werden kann.

Für Unternehmen wird meistens eine Kombination aus internen und externen Quellen empfohlen. Zu den internen Quellen gehören SOC’s, Fachgemeinschaften, Sicherheitsmeldungen, Blogs oder Dark Web Researches. Offizielle externe Feeds stammen hauptsächlich aus der Kundentelemetrie, dem Dark Web, Open-Source-Datenbanken, der Malware-Verarbeitung sowie der manuellen Ereignisanalyse und der Sicherheitsforschung.

Threat Intelligence Feeds werden meist aus mehreren Quellen zusammengesetzt und konzentrieren sich größtenteils auf einen Schwerpunkt.

Schwerpunkte sind z.B. Botnet-Aktivitäten, Domänen oder bösartige IP-Adressen. Sobald neuartige Bedrohungen entdeckt werden, erhalten die Abonnenten über den Feed Informationen darüber. Schnelligkeit ist hier ein entscheidender Faktor, um sich möglichst frühzeitig gegen neue Bedrohungen wehren zu können. Diese Feeds können auf verschiedene Arten verwendet werden. Eine Möglichkeit wäre den Feed direkt an vorhandene Firewalls zu senden, um neuartige Entdeckungen sofort zu berücksichtigen und zu blockieren. Eine Alternative wäre das manuelle Überprüfen von Informationen durch Analysten, was aber sehr zeitaufwändig ist. Die letzte Alternative besteht aus einem SIEM- oder UEBA-System, bei denen die Bedrohungsdaten mit internen Sicherheitsereignissen verbunden werden und Warnmeldungen erzeugen, wenn ein Sicherheitsfall eintritt. Komplette Threat-Intelligence-Plattformen bieten die Möglichkeit, mehrere Feeds gleichzeitig zu erfassen, zu analysieren und sie miteinander zu vergleichen. Gängige Threat-Intelligence-Plattformen sind z.B. ThreatQuotient, Anomali ThreatStream und Palo Alto Networks' AutoFocus, die jeweils einen etwas anderen Fokus haben.
Eine große Herausforderung bei Threat Intelligence ist es effektiv zu planen und zu priorisieren, um nicht in den großen Datenmengen zu versinken. Die Bedrohungsdaten der Feeds müssen auch immer in einem gewissen Kontext betrachtet werden, um sinnvoll für den Sicherheitsexperten zu sein. Zu guter Letzt setzt Threat Intelligence noch spezielle Prozesse und Fähigkeiten des Security-Teams voraus. Ein nicht zu unterschätzender Faktor, denn es gibt einen globalen Mangel an Sicherheitsexperten.

Moderne SIEM-Plattformen sollen den Sicherheitsexperten durch integrierte Automatisierungs- und Analysefähigkeiten entlasten. Daten werden dabei möglichst genau dann bereitgestellt, wenn sie auch benötigt werden. Die automatisierte Reaktion bei Vorfällen gibt Analysten die Möglichkeit Daten aus vielen Tools zu sammeln, Vorfälle zu identifizieren, sie mit Bedrohungsdaten zu verbinden und nötige Schritte zu unternehmen, um den Schaden bestmöglich einzudämmen. Ein weiterer Schritt, um die Datenüberlastung zu reduzieren liegt in der Verbindung von Analysen zur Identifikation von anomalem Verhalten mit Daten aus der Bedrohungsanalyse.

Wenn Threat Intelligence also von Sicherheitsexperten in Kombination mit SIEM richtig eingesetzt wird, bietet es einen erheblichen Sicherheitsvorteil für fast jeden Aspekt der Cybersicherheitsoperationen eines Unternehmens.

2.2.1 Threat Intelligence Lifecycle

Der Threat Intelligence Lifecycle könnte in Unternehmen wie folgt aussehen nach Socradar.io (vgl. SOCRadar Cyber Intelligence Inc. 2020):

Planung und Steuerung

Zunächst werden die Anforderungen für die Datenerfassung definiert, dabei sollten die richtigen Fragen gestellt werden, um verwertbare Informationen zu erhalten.
Beispielhafte Fragen nach einem Phishing-Angriff wären z.B. wie erfolgreich diese Art von Angriffen ist, wie oft sie erfolgen und ob andere Organisationen dieselben Attacken erleiden. Der Firma sollte eine Liste der eigenen priorisierten Güter und Werte vorliegen, um überhaupt mögliche Ziele zu erkennen.

Sammlung

Nachdem die Anforderungen der Datenerfassung definiert sind, werden Rohdaten über aktuelle oder zukünftige Bedrohungen erfasst. Dazu werden verschiedene Quellen eingesetzt, wie z.B. interne Netzwerklogs, alte Warnmeldungen sowie das Internet, dark web und andere technische Quellen. Weiterhin wird festgelegt, welche Tools und Methoden benötigt werden, um den Anforderungen in Schritt eins gerecht zu werden.

Verarbeitung

Die Daten werden anschließend auf unterschiedliche Weise weiterverarbeitet. Sie werden z.B. mit Metadaten-Tags versehen. Redundante Informationen, falsch-positive, aber auch falsch-negative Daten werden herausgefiltert. Die Daten werden in passende Formate aggregiert etc. Der Verarbeitungsprozess

kann manuell, semi-manuell und automatisch z.B. mittels Threat-Intelligence-Plattform erfolgen.

Analyse und Produktion

In dieser Phase entsteht das Endergebnis, das die Fragen „Warum passierte Event XY“ beantworten sollte. Nachdem die verarbeiteten Daten analysiert wurden, werden sie vom CTI-Team den Entscheidungsträgern der Organisation vorgestellt. Informationen können hier als Entscheidungsgrundlage dienen, ob z.B. potenzielle Bedrohungen weiter untersucht werden sollten, welche Maßnahmen getroffen werden sollten etc. Je nach Skill-Fertigkeiten der Entscheidungsträger muss der Bericht mehr oder weniger technisch ausfallen. Zur Erstellung der Berichte können Frameworks wie MITRE ATT&CK genutzt werden.

Verbreitung und Feedback

Hier geht es daran, die Daten in einem passenden Format an die entsprechenden Gruppen bzw. Security-Teams innerhalb der Organisation zu bringen. Anschließend entsteht bestenfalls ein Feedback zu den geteilten Daten, um zu überprüfen ob alle Anforderungen eingehalten wurden. Danach werden neue Anforderungen definiert und der Lifecycle kann von vorne beginnen.

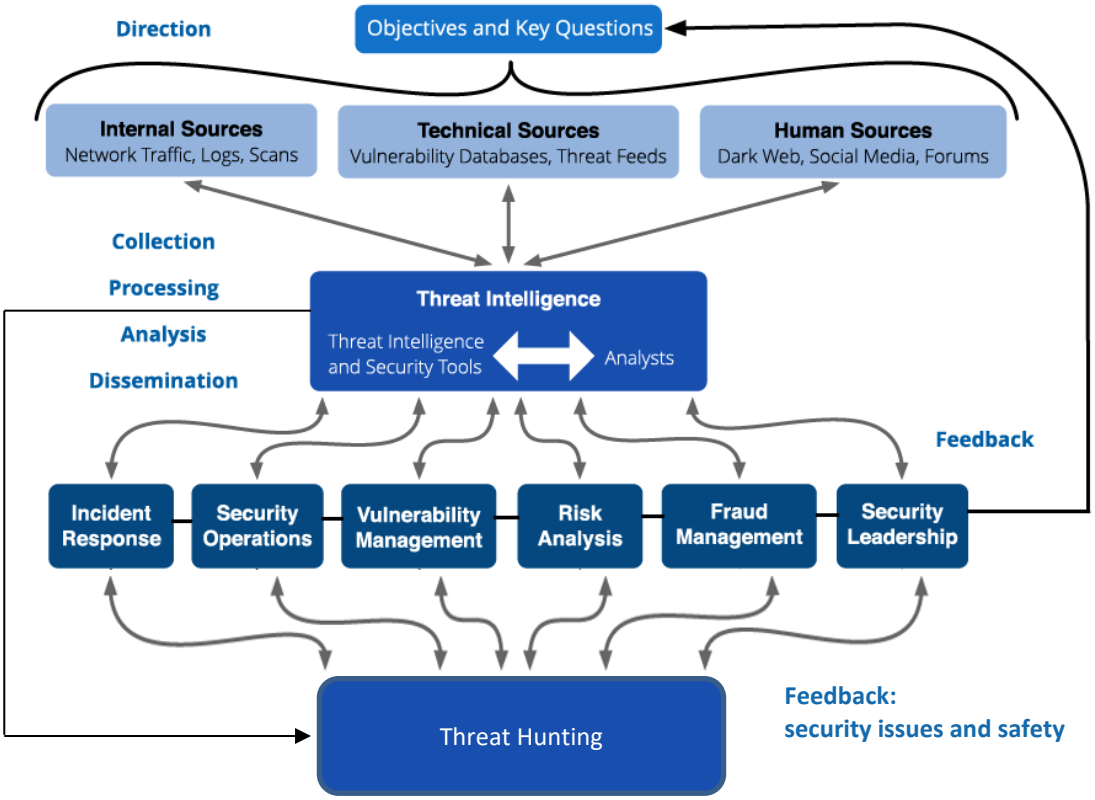


Abbildung 2.5: The Threat Intelligence Lifecycle, mit eigens angepasster Threat-Hunting-Erweiterung (vgl. The Recorded Future Team 2020)

Die obige Grafik wurde im unteren Teil erweitert, um auch den Aspekten des Threat Huntings im Threat Intelligence Lifecycle gerecht zu werden. Der erweiterte Lifecycle zeigt auf, dass Kommunikation und Feedback zwischen den Threat Huntern und den Security-Teams ebenfalls wichtig ist. Threat Hunter können hier ihr Wissen über Sicherheitslücken (z.B. in einer Webseite) und anderen Auffälligkeiten, die z.B. mit Hilfe von Threat Intelligence erkannt wurden an die entsprechenden verantwortlichen Personen und Teams weitergeben (vgl. The Recorded Future Team 2020).

2.2.2 IoC & IoA

Indicators of Compromise (IoC) und Indicators of Attack (IoA) werden oftmals unter dem IoC-Begriff zusammengefasst, obwohl sie sehr unterschiedlich sind.
Indicators of Attack sind eher Indizien dafür, dass sich jemand Zugang zum Netzwerk verschafft hat, oder gerade dabei ist. Es könnte also ein Zeichen für einen bevorstehenden Angriff in einem Netzwerk sein.
Indicators of Compromise hingegen sind ein Zeichen dafür, dass sich jemand bereits Zugang zum Netzwerk verschafft hat und möglicherweise bereits Daten gestohlen oder der Firma anderweitigen Schaden verursacht hat (vgl. Endace Ltd.).

Abbildung 2.6: Comparing an IoA to an IoC (Jessica DeCianno 2014)

IoA's verfolgen also einen proaktiveren Ansatz als IoC's.
Wie sich solche Indikatoren mithilfe von Threat Intelligence automatisiert erstellen lassen, wird später im Praxisteil unter Abschnitt 5.3 verdeutlicht.

2.2.3 Arten der Threat Intelligence

Wie bereits in Punkt 2.2.1 dargestellt, gibt es verschiedene Gruppen mit verschiedenen Anforderungen für das Threat-Intelligence-Endprodukt. Je nach Anforderungen, Quellen und der entsprechenden Zielgruppe kommt ein anderes Threat-Intelligence-Produkt am Ende des Life Cycles zum Vorschein. Basierend darauf gibt es nach Brook die folgenden drei Arten von Threat Intelligence (vgl. Chris Brook 2020).

Strategische Threat Intelligence

Zielgruppe hierbei ist ein nicht-technisches Publikum, Inhalt sind hier langfristige Themen und breite Trends wie z.B. strategische Erkenntnisse über Bedrohungen. Diese können ein umfassendes Bild von Absicht und Fähigkeiten von Cyber-Bedrohungen vermitteln und helfen dabei, wichtige Entscheidungen zu treffen, als auch Warnungen zu generieren.

Taktische Threat Intelligence

Zielgruppe ist hierbei schon ein technisch etwas versierteres Publikum. Themen betreffen eher das daily Business im Sicherheitsbereich, wie das Bearbeiten und Erstellen von IoC's. Inhalte sind hier eher sogenannte TTP's (Taktiken, Techniken und Prozeduren) von Angreifern.

Operative Threat Intelligence

Zielgruppe ist hierbei ein hoch spezialisiertes und technisches Publikum. Themenbereiche sind spezifische Attacken, Kampagnen, bestimmte Malware oder Tools. Die Form der Übertragung kann als forensischer-Cyber-Threat-Intelligence-Report erfolgen.

2.2.4 Praxisumfrage

Der nachfolgend grob zusammengefasste Threat Intelligence Report 2018 welcher von Holger Schulze der Firma AlienVault erstellt wurde (vgl. Holger Schulze 2018) zeigt mittels einer Umfrage einige wichtige Aspekte des aktuellen Standes von Threat Intelligence in Unternehmen auf. Die meisten der Befragten Experten sind Spezialisten, Manager und Consultants aus den Bereichen IT-Security, IT-Operations und Engineering.

Dabei geben 77% aller Befragten an, dass sie Threat Intelligence für die Sicherheit in ihrem Unternehmen für sehr wichtig halten.
An erster Stelle des Nutzens von Threat Intelligence steht das Erkennen von Angriffen und Threats, also Bedrohungen. Dabei schätzen 59% aller Befragten ihre Effektivität im Kampf gegen Threats nur als durchschnittlich oder schlechter ein. Der Hauptnutzen durch Threat Intelligence teilt sich in eine bessere Bedrohungsanalyse, eine schnellere Bedrohungserkennung und Reaktionszeit als auch effizientere Sicherheitsvorgänge auf. Mehr als die Hälfte aller Befragten geben an, mehr als fünf Stunden pro Woche mit dem manuellen Recherchieren von Threat-Intelligence-Feed-Alarmen zu verbringen. Dabei sieht sich jeder fünfte Teilnehmer mit einer False Positive Rate von über 20% bei Threat-Intelligence-Alarmen konfrontiert. Zu den größten Bedrohungen für Firmen zählen Phishing-Angriffe an erster Stelle, gefolgt von Zero-Day-Angriffen und Insider-Angriffen an dritter Stelle.
Die geschätzte Erkennungsrate für Threats liegt in der Mehrheit genau wie die oben genannte Effektivität eher im durchschnittlichen bis schlechten Bereich. Nur 7% aller Befragten schätzen sich weit über dem Durchschnitt ein.
Bei der Frage nach den Prioritäten im Threat Management über die nächsten zwölf Monate sah die Verteilung wie in der Grafik unten aus, dabei gaben nur 24% der Befragten an, ihr Budget in dem Bereich zu erhöhen und 70% es gleich zu lassen.

Abbildung 2.7: Threat Management Priorities (Holger Schulze 2018)

Die genannten Aussagen bezüglich SIEM als meistgenutzte Plattform, als auch das Problem mit dem fehlenden Sicherheitspersonal finden sich ebenfalls in dem Threat Intelligence Report wieder.
Der Report zeigt auf, dass zwar ein Großteil von IT-Spezialisten Threat Intelligence für wichtig und nötig hält, aber selbst nicht wirklich von der eigenen Effektivität und der Erkennungsrate von Bedrohungen überzeugt ist.

2.3 Threat Hunting

Was ist eigentlich Threat Hunting?
Eine Definition dafür lautet wie folgt:

„Threat Hunting, often referred to as Incident Response without the Incident, is an emergent activity that comprises the proactive, iterative, and human-centric identification of cyber threats that are internal to an Information Technology network and have evaded existing security controls“ (Government UK 2019).

Dabei kann Threat Hunting den Unterschied zwischen wenigen tausend Euro Schaden bei einer schnellen Entdeckung von Bedrohungen und mehreren Millionen Euro bei einer vollkommen kompromittierten Umgebung bei später Entdeckung machen (vgl. Brent Murphy und David French 2020, S. 4).

Threat Hunting befasst sich also mit der Suche nach Schwachstellen und abnormalen Aktivitäten im Netzwerk, die Anzeichen für eine Kompromittierung, einen Angriff oder einen Datendiebstahl sein können. Das passiert ohne, dass irgendeine Art Alarm im Netzwerk ausgelöst wurde. Die Suche findet „repetitiv“ statt und ist hauptsächlich vorbeugende Arbeit. Man geht davon aus, dass ohne Threat Hunting im Schnitt 220 Tage zwischen dem Einbruch ins Netzwerk und dessen Entdeckung (meist von externer Seite) vergehen (vgl. Peter H. Gregory 2017, S. 8–9).
Bis zu der Eindämmung eines Sicherheitsvorfalles dauert es dann noch einmal durchschnittlich 73 Tage (vgl. F-Secure Corporation, S. 7).

Threat Hunter gehen davon aus, dass das Netzwerk bereits kompromittiert ist. Sie arbeiten aus dem Blickwinkel und der Denkweise von Angreifern heraus. Dabei ist es ein kontinuierlicher Prozess, verbesserte Erkennungsszenarien zu entwickeln und somit das Netzwerk besser zu sichern. Threat Hunting ersetzt dabei keine anderen Erkennungs- und Reaktionsstrategien oder Sicherheitssysteme (vgl. F-Secure Corporation, S. 6).

Threat Hunting bedeutet nicht nur, dass „Böse“ im System zu finden, sondern vielmehr die Hinweise darauf, dass sich ein Angreifer im Netz befindet oder befunden hat. Die reine Suche nach IoC’s ist hier also zu wenig (vgl. Peter H. Gregory 2017, S. 10).

Für ein effektives Threat Hunting werden Tools benötigt, die Prozesse, geöffnete Dateien, sowie die Kommunikation im Netzwerk besser darstellen, untersuchen und dokumentieren können (vgl. Peter H. Gregory 2017, S. 9).

Dabei arbeiten Mensch und Maschine gemeinsam gegen Bedrohungen und Angriffe. Der Mensch entwirft die Strategien und Skripte, die wiederum von der Maschine automatisiert ausgeführt werden (vgl. Laimingas 2018).

Voraussetzungen für ein effektives Threat Hunting sind menschliche Expertise, detaillierte, vollständige Daten von allen Endgeräten und Netzwerkknoten, sowie eine Threat-Intelligence-Plattform mit den neuesten Erkenntnissen über externe Trends in Bezug auf Bedrohungen. Dabei sollte Threat Hunting bestenfalls 24/7 das ganze Jahr über stattfinden (vgl. CrowdStrike Holdings, Inc. 2019).

Threat Hunting sollte dabei nicht nur proaktiv und methodisch ablaufen, sondern auch bestenfalls im Verborgenen, um den oder die Angreifer nicht aufzuschrecken (vgl. Karen Scarfone 2016, S. 6).

2.3.1 Sicherheitsteams in der IT-Sicherheit

In den folgenden Absätzen wird auf die gängigsten Arten von Sicherheitsteams in der IT-Sicherheit und deren Aufgabenbereich eingegangen.

Team Blue:

Im Bereich Cybersecurity wird oft von "Team Red" und "Team Blue" gesprochen.
Blue Teams werden dabei aus organisationsinternen IT-Sicherheitsexperten zusammengestellt. Der Unterschied zu "normalen" Sicherheitsteams liegt dabei in der ständigen Wachsamkeit und der Verteidigungsbereitschaft dieser Teams. Sie verteidigen die IT-Infrastruktur sowohl gegen echte als auch simulierte Angriffe von Team Red. Blue Teams analysieren laufend die IT-Systeme, finden Schwachstellen und prüfen die Effektivität von getroffenen Sicherheitsmaßnahmen (vgl. Stefan Luber und Peter Schmitz 2020).

Blue Teams arbeiten dabei proaktiv und sorgen für einen vorbeugenden Schutz der Infrastruktur, wozu natürlich auch Threat Hunting gehört. Denn mit reinen defensiven Sicherheitstaktiken alleine verschenkt man menschliches Erfindungspotenzial (vgl. Arran Purewal und Peter Schmitz 2020).

Team Red:

Red Teams bestehen meistens aus externen Fachleuten, die sich aus Sicherheitsexperten mit speziellem Angreifer-Know-how zusammensetzen. Ehemalige oder aktive Hacker sind oft Teil solcher Teams. Penetrations- und Sicherheitstests, sowie Hacker Tools, Malware, Phishing und Social Engineering gehören zum Portfolio dieser Teams. Ziel ist es dabei in die Netzwerke und IT-Systeme von Team Blue einzudringen und an sensible Daten zu gelangen. Diese Teams verursachen dabei keinen echten Schaden, sondern zeigen lediglich potenzielle Risiken und Schwachstellen auf, aus denen wiederrum Sicherheitsmaßnahmen von Team Blue erstellt werden, um ihre Systeme zu stärken (vgl. Stefan Luber und Peter Schmitz 2020).

SOC:

SOC steht für Security Operations Center und bildet dabei eine Art Kommandozentrale für sämtliche Belange, die mit Cyber Security zu tun haben. Hier werden spezialisierte Analysten, Prozesse und Tools kombiniert, um

zielgerichtet Bedrohungen für die IT-Infrastruktur zu identifizieren und entsprechend darauf zu reagieren. SOC's arbeiten dabei rund um die Uhr, um auftretende Schäden schnellstmöglich einzudämmen (vgl. Volker Scholz 2019).
Sie arbeiten also proaktiv und versuchen Schwachstellen in der IT-Infrastruktur frühzeitig zu erkennen und zu beseitigen. Zudem reagieren sie auch reaktiv auf Angriffe wie DOS-Attacken. Das Management des Unternehmens bzw. der Organisation wird dabei regelmäßig mit Reportings über die Sicherheit der IT-Systeme und der Arbeit des SOC's informiert (vgl. Stefan Luber und Peter Schmitz 2017).

CERT:

Ein Computer-Emergency-Response-Team besteht aus IT-Fachleuten und Sicherheitsexperten. Sie wirken gemeinsam an Lösungen für Sicherheitslücken oder konkreten Sicherheitsvorfällen mit. Beispielweise kann das Team bei der Ausbreitung neuartiger Viren, Veröffentlichung neuer Sicherheitsschwachstellen oder gezielten Serverangriffen agieren. CERT-Teams können auch vor Sicherheitslücken warnen, oder präventive Lösungsansätze für Bedrohungen liefern (vgl. Stefan Luber und Peter Schmitz 2018). In Deutschland gibt es von offizieller Seite aus die folgenden zwei Arten von CERT-Teams.
Es gibt den CERT-Bund, der in erster Linie den Bundesbehörden zur Verfügung steht und den Bürger-CERT, der eher interessierte Privatpersonen informiert (vgl. BSI).

Ein SOC ist also in erster Linie darauf fokussiert, Bedrohungen zu erkennen und erste Gegenmaßnahmen einzuleiten, während das CERT sich eher um detaillierte Analysen und die Behebung von Sicherheitsvorfällen kümmert. Man kann auch sagen, dass ein SOC einen breiteren Sicherheitsfokus hat (vgl. Volker Scholz 2019). Je nach Unternehmen kann ein SOC dabei auch den Incident Response Part eines CERT mit übernehmen (vgl. Ed Moyle 2019). Das CERT ist zudem für das Verwundbarkeitsmanagement verantwortlich. SOC's und CERT's ergänzen sich also gegenseitig und sorgen für eine starke Einheit im Bereich Cyber Security.

2.3.2 Arten des Threat Huntings

Nach Crowdstrike gibt es die folgenden drei verschiedenen Arten des Threat Huntings (vgl. CrowdStrike Holdings, Inc. 2019):

Hypothesengetriebene Untersuchung:

Zum einen gibt es die hypothesengestützte Untersuchung, bei der neuartige Bedrohungen aus einer Menge von Daten identifiziert werden und Einblicke in Taktiken, Techniken und Verfahren der Angreifer geben (TTP).
Threat Hunter versuchen dann dieses spezielle Verhalten in Ihrem Netzwerk bzw. Ihrer Umgebung zu finden, um herauszufinden, ob sie bereits infiziert sind.

Untersuchung auf der Grundlage von IoC's oder IoA's:

Die zweite Art ist die Untersuchung auf Grundlage von IoC's und IoA's. Dabei werden mittels Threat Intelligence bestimmte IoC's und IoA's erstellt und überwacht, die als „Trigger" fungieren. Lösen diese aus, können sie auf bestimmte neue Bedrohungen hinweisen.

Fortgeschrittene Analysen und maschinelles Lernen:

Die letzte Art kombiniert maschinelles Lernen mit einer ausführlichen Datenanalyse, um riesige Datenmengen auf Anomalien und Verhaltensmuster zu überprüfen, die auf Bedrohungen hinweisen können. Spezialisten untersuchen diese Verhaltensmuster dann wiederrum, um mögliche Bedrohungen zu identifizieren.

Wie bei allen Ansätzen zu erkennen ist, besteht immer eine Zusammenarbeit zwischen Mensch und Maschine, um das bestmöglichste Ergebnis zu erzielen.

2.3.3 Besonderheiten bei IoT-Geräten

IoT-Geräte sind besonders schwer zu sichern. Im Gegensatz zu beispielsweise Desktop-Geräten oder Servern, die viele verschiedene Dienste anbieten, sind IoT-Geräte meist nur für einen bestimmten Zweck bestimmt. Beispielhafte Einsatzmöglichkeiten sind in Kapitel 2.1 bereits näher erläutert. Eine hohe Effizienz hat eine starke Priorität bei diesen Geräten, weshalb die Betriebssysteme auf die Funktion des Gerätes hin optimiert sind um Strom, Arbeitsspeicher und CPU-Ressourcen einzusparen. Das macht es besonders schwer, bis unmöglich eigene Software auf diesen Geräten zu installieren oder überhaupt an die Systemlogs dieser Geräte zu gelangen. Ein Einblick, was in diesen Geräten überhaupt vorgeht, ist also nur sehr schwer zu bekommen. Somit werden Threat Hunter in diesem Bereich dazu gezwungen von außen über das Netzwerk auf das Gerät zu schauen. Dabei besteht zumindest der Vorteil, dass sich das Netzwerk-Threat-Hunting zwischen Desktops, Servern und IoT-Geräten nicht unterscheidet (vgl. Chris Brenton 2020).
Wie man das Threat Hunting für IoT-Geräte genau angeht, wird in Schritt 5.5 näher erläutert.

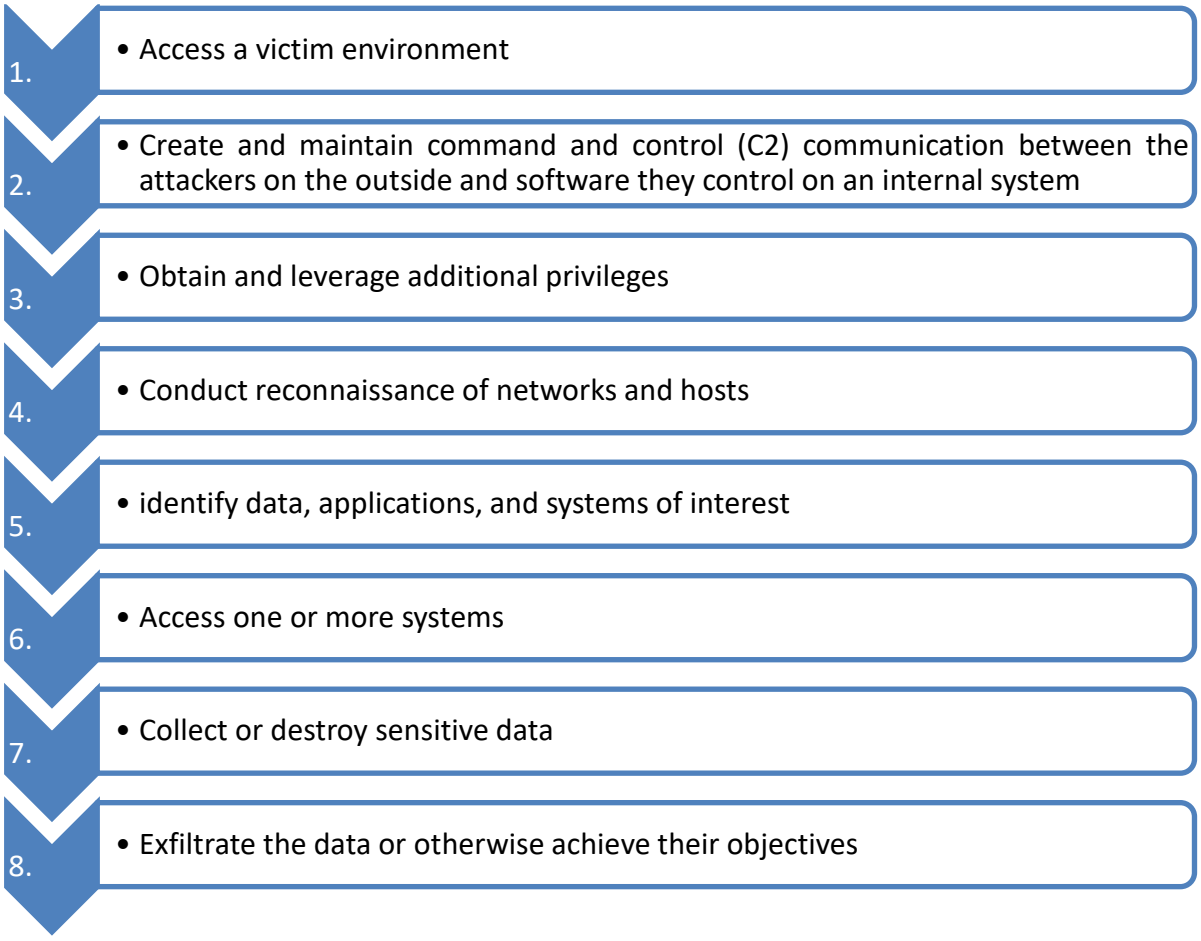
2.3.4 Vorgehensweise im IoT-Bereich

Wie bereits in Punkt 2.3.2 erwähnt, ist das Threat Hunting im IoT-Bereich eher auf das Netzwerk beschränkt. Dazu sollten im Vorfeld folgende Überlegungen über das Umfeld getroffen werden. Wie sieht der normale Netzwerkverkehr aus bzw. wie könnten Abweichungen aussehen. Welche Wissensdatenbanken (z.B. Threat Intelligence) können verwendet werden, bzw. auf welchem Wissen kann aufgebaut werden? Welche Schwachstellen und Angriffspunkte gibt es und wie gefährlich kann Social Engineering für einen werden? Wie würde jemand angreifen, wie geht er vor und was ist technisch überhaupt machbar von seiner Seite aus (vgl. Peter H. Gregory 2017, S. 23–28)?

Für eine Übersicht über mögliche Angriffsweisen und Verhaltensmuster gibt es z.B. MITRE ATT&CK, eine globale, offene, kostenlose Wissensdatenbank, dessen Ziel es ist eine sicherere IT-Welt zu schaffen (vgl. Mitre Corporation).

Gerade bei Firmen gibt es verschiedene Motive anzugreifen, dazu gehören finanzielle, staatlich motivierte oder auch politische, durch sogenannte Hacktivists (vgl. Brent Murphy und David French 2020, S. 12). Hacktivisten verfolgen dabei oft eine eigene Ideologie. Eine der bekanntesten Hacktivisten-Gruppen ist Anonymous, die schon viele Unternehmen und Dienste mittels DDOS-Attacken lahmgelegt haben (vgl. Kevvie Fowler 2016, S. 9–10). Die Tactics, Techniques und Procedures (TTP), sowie die verfolgten Ziele und Zielgruppen sind dabei sehr unterschiedlich (vgl. Brent Murphy und David French 2020, S. 12). Die folgenden Schritte wurden inhaltlich übernommen von (Brent Murphy und David French 2020, S. 13–14) und in dieser Arbeit zum besseren Verständnis als SmartArt aufbereitet. Sie sind eine häufige Vorgehensweise von Angreifern.

Geht man davon aus, dass sich Angreifer wie in der SmartArt unter diesem Text verhalten, gibt es die Möglichkeit nach Spuren dieser Vorgehensweisen zu suchen (vgl. Brent Murphy und David French 2020, S. 14).



Dazu sollte der zuständige Threat Hunter sein Netzwerk kennen und wissen, wie der Normalzustand auszusehen hat, was die ganze Sache schon etwas schwerer gestaltet. Es gibt verschiedene Fragen, die gestellt werden sollten, um den Netzwerkverkehr entsprechend zuzuordnen. Beispielsweise: Welche Tools werden normalerweise im Netzwerk genutzt? Wo finden sich die sensibelsten Daten im Netzwerk und wie wird darauf zugegriffen? Welche Netzwerkverbindungen existieren zwischen dem Unternehmen und Drittanbietern wie Lieferanten oder Dienstleistern (vgl. Brent Murphy und David French 2020, S. 17)? Welche Onlinezeiten sind normal in meiner Umgebung? Durch den Fokus auf IoT-Geräte in dieser Arbeit lassen sich die Fragen deutlich eingrenzen. Nähere Beispiele für die Umsetzung in der Praxis finden sich unter Abschnitt 5.5.

3 Anforderungsanalyse

3.1 Funktionale Anforderungen

Die Lösung sollte bestmöglich dokumentiert sein. Zudem sollte sie die Metadaten des Netzwerkverkehrs der IoT-Geräte vollständig detailliert mitschneiden und in Echtzeit darstellen können. Ein hochwertiger Informationsgehalt spielt also eine wichtige Rolle. Wichtige Informationen sollten bestenfalls in irgendeiner Art herausstechen bzw. aus der Menge an Daten erkennbar gemacht werden können. Zudem sollte die Lösung in der Lage sein historische Daten mit aktuellen Daten zu vergleichen, um Anomalien, Muster und Abweichungen einzelner Geräte zu erkennen. Dabei sollte die Lösung bestenfalls in irgendeiner Form intelligent arbeiten, um Verhaltensauffälligkeiten im Netzwerk frühzeitig und bestmöglich zu erkennen und um dem Threat Hunter Arbeit abzunehmen. Zusätzliche Funktionen und Schnittstellen wie die Einbindung von Threat-Intelligence-Anbietern oder anderen Tools zur besseren Datenverarbeitung spielen auch eine wichtige Rolle. Die Softwarelösung sollte zudem kostenlos und bestenfalls Open Source sein, um höhere Kosten für dieses Projekt zu vermeiden.
Der Vorteil von Open Source, also offenem Quellcode liegt oft in einer großen Entwickler-Community, die das Projekt voranbringt. Dies verringert zudem das Risiko für Hintertüren und Fehler in der Software. Davon ausgehend sollte die Lösung auch von dem Hersteller und/oder der Community-Seite aus hilfetechnisch ausreichend unterstützt werden.

3.2 Nichtfunktionale Anforderungen

Die Lösung sollte in erster Linie einfach bedienbar und übersichtlich sein.
Der Server muss performant genug sein, um die Daten mehrerer Netzwerkgeräte mitzuschneiden, sie aufzubereiten und in Echtzeit darzustellen. Es darf keine Funktionseinbußen bei hoher Aktivität durch langsame Hardware geben. Der Server muss zuverlässig sein und darf bestenfalls keine Ausfälle jeglicher Art haben. Das gleiche gilt natürlich für die IoT-Geräte. Zudem sollte er leicht und schnell wartbar sein, um aktuell zu bleiben und um mögliche Sicherheitsprobleme zu mindern. Die Netzwerk-Daten müssen korrekt und vollständig ohne die Möglichkeit auf ungewollten Fremdzugriff gehalten werden. Die Benutzbarkeit der Software sollte möglichst intuitiv und einfach laufen. Die Lösung sollte als Zusatzpunkt möglichst gut skalierbar sein für größere Lastaufkommen.

3.3 Softwareanforderungen zusammengefasst

In der Abbildung 3.1 hier drunter sind die funktionalen und nichtfunktionalen Anforderungen, die der Lösungssoftware gestellt werden samt Gewichtung und Begründung als Tabelle zusammengefasst. Die Gewichtung wurde hier in Prozenten aufgeteilt, wobei die gewählten wichtigsten Kriterien der Funktionsumfang, die vollständige Daten Echtzeit-Darstellung sowie der Informationsgehalt der Daten sind.

Kriterien	Gewichtung	Begründung der Gewichtung
Bedienbarkeit	10%	Die Lösung sollte für den täglichen Betrieb unbedingt intuitiv bedienbar sein
Dokumentation	10%	Ein wichtiger Punkt, um den Funktionsumfang der Lösung und dessen Bedienung besser zu verstehen
Support & Community	10%	Wichtig für Hilfe und Support, bei dem die Dokumentation nicht mehr ausreicht
Funktionsumfang & Schnittstellen	15%	Ein wichtiger Punkt, um z.B. durch Threat-Intelligence-Einbindung oder anderen zusätzlichen Tools noch "bessere" Erkenntnisse aus den ermittelten Daten zu gewinnen
Open Source	5%	Ein eher kleiner Punkt, dennoch aufgrund der Begründungen in Abschnitt 3.1 ein Bewertungskriterium
Echtzeit-Darstellung	15%	Die Daten sollten in Echtzeit dargestellt und mit historischen Daten verglichen werden können
Intelligente Datenverarbeitung	5%	Ein kleiner, aber hilfreicher Punkt zur besseren Datenverarbeitung
Informationsgehalt	20%	Die Daten müssen vollständig sein, dabei sollten wichtige Informationen und Details bestenfalls erkennbar gemacht werden können.
Skalierbarkeit	5%	Für weitere, größere Einsatzzwecke ein nicht zu vernachlässigender Punkt
Wartbarkeit	5%	Ebenfalls ein Kriterium für Wartungen an Server und Software, sowie Sicherheits- und Funktionsupdates
Total	**100%**	

Abbildung 3.1: Anforderungen an die Softwarelösung

4 Untersuchung möglicher Lösungen

4.1 Vorstellung der Lösungen

Es gibt unzählige einzelne Paket-Lösungen, die einen Bereich des Threat Huntings abdecken. Sei es der Netzwerkscan, um Daten zu generieren, die Datenaufbereitung, Zusatzmodule für weitere Funktionalitäten und Kombinationen mit anderen Lösungen, wie Threat-Intelligence-Plattformen oder die Dashboards an sich. Um den Vergleichsrahmen nicht zu sprengen, wurden deshalb im Vorfeld verschiedene Best-Practice-Kombinationslösungen aus dem Internet recherchiert, von denen die vier vielversprechendsten in diesem Kapitel vorgestellt werden. Durch die Anzahl der verglichenen Lösungen entsteht ein umfänglicher Überblick über die verschiedenen Funktionalitäten und Schwerpunkte von Software in diesem Sicherheitsbereich. Die Kriterien stammen dabei aus der Anforderungsanalyse in Punkt 3.3. Dabei gibt es auch Lösungen, die von Entwickler & Projektteams zusammengestellt worden und einem damit das eigenständige Installieren vereinfachen. Vier dieser Lösungen werden im folgendem genauer miteinander verglichen. Die genannten Lösungen wurden im Vorfeld einmal testweise installiert und dabei anhand der genannten Kriterien überprüft. Sie bestehen größtenteils aus den oben genannten einzelnen Paketen, die jeweils einen bestimmten Zweck erfüllen und zusammen der Aufgabe des Threat Huntings bestmöglich gerecht werden sollen.

Chiron-Elk:

Chiron-Elk ist ein „Intrusion-Detection-System“, das für den Heimbereich konzipiert wurde. Es wurde von einem Entwicklerteam basierend auf dem bekannten ELK-Stack erstellt und wird mit einem auf Machine Learning basierendem Threat Detection Framework namens AKTAION erweitert. ELK steht für Elasticsearch, Logstash und Kibana. Mittels Elasticsearch lässt sich nach Belieben auf bestimmten Daten filtern und suchen. Logstash ist für die Normalisierung und Verarbeitung von Logdateien zuständig. Kibana ist das Dashboard für die Anzeige der Daten. Die dabei erfassten Daten stammen von den Netzwerkanalysetools Zeek (ehemals Bro genannt), P0f sowie NMAP. Zeek ist ein mächtiges Open Source Netzwerkanalyse-Tool, um den Netzwerkverkehr zu überwachen. Es schneidet Netzwerkpakete dabei kompakt auf Metadaten-Ebene mit und stellt diese in verschiedenen Logdateien zur Verfügung. P0f ist ein Netzwerkanalyse-Tool, welches mittels passivem Fingerprinting Systeme identifiziert. NMAP wiederrum ist die Abkürzung für Network Mapper und dient als Port Scanner zur Auswertung von Hosts eines Rechnernetzes. Chiron ist dabei ein Komplettpaket, dass sich selbstständig wartet und ohne Benutzereingaben das Netzwerk selbstständig durchscannt. Auf dem Kibana Dashboard lassen sich alle möglichen Informationen über das Netzwerk wie genutzte IP-Adressen, Ports, Zeiträume, Dienste etc. anzeigen und weiter mittels Query auf Elasticsearch filtern. Die Query-Sprache hierbei heißt Lucene Query Syntax. Die Ergebnisse werden dabei grafisch oder als Text dargestellt.
AKTAION scannt alle vier Stunden die Bro (Zeek) logs nach Ransomware/Phishing-Angriffen durch. Diese Daten werden mit einem gutartigen Trainingsdatensatz verglichen, um Auffälligkeiten zu erkennen. Chiron-Elk nutzt in seiner ausgelieferten Version Ubuntu 16.04 LTS als Distribution.
Chiron-Elk Bezugsinformationen: (GITHUB:Joseph Zadeh et al. 2020).

Splunk + Zeek + Dovehawk + MISP + Corelight Splunk Plugin:

Diese Kombination wird ebenfalls oft beworben. Splunk ist hierbei das Webinterface, das die Daten letztendlich darstellt und auf dem mittels Splunk Search Processing Language (SPL) Daten gefiltert werden. Dashboards zur besseren Datenübersicht lassen sich hier ebenfalls erstellen. Die Netzwerkmitschnitte stammen hier wiederrum von dem Programm Zeek, das seine Mitschnitte in verschiedene Logdateien aufteilt. Mittels Dovehawk-Modul lassen sich Signaturen und IoC's über MISP (Malware Information Sharing Platform), einer Threat-Intelligence-Plattform von bekannten Threat-Intelligence-Anbietern, live mit dem Zeek-Verkehr abgleichen. Wird ein eigener oder automatisch erstellter IoC dabei ausgelöst, wird dies zu MISP zurückgemeldet und der „Sighting" Zähler in dem entsprechenden Attribut wird um eins hochgezählt. Sightings sind die Sichtungen eines gewissen Attributes im Netzwerk. Dadurch wird der Netzwerkverkehr zusätzlich mit den heruntergeladenen Signaturen überprüft, um potenziell gefährliche Bewegungen im Netzwerkverkehr festzustellen. Weiterhin lassen sich mittels der MISP Software auch eigene IoC's erstellen, um bei bestimmten ungewöhnlichen Aktivitäten direkt eine Warnmeldung zu bekommen. Das Corelight Plugin sorgt für vorgefertigte Dashboards, um den Einstieg in Splunk zu vereinfachen.
Für diese Programmkombination bietet sich Ubuntu oder Debian als Distribution an. Splunk Bezugsinformationen: (Splunk, Inc. Version: 8.1.0), Zeek: Bezugsinformationen: (Vern Paxson Version 3.2.2), Dovehawk Bezugsinformationen: (GITHUB:tylabs Version 1.02.001), MISP Bezugsinformationen: (DOCKER:harvarditsecurity 2020), Corelight App for Splunk Bezugsinformationen: (Corelight Inc. Version: 2.1.0).

ELK + Zeek + Dovehawk + MISP:

Bis auf Splunk ist dies im Prinzip dieselbe Konfiguration wie das zweite vorgestellte Produkt, weshalb die überschneidenden Eigenschaften hier nicht wiederholt werden. Das Kibana Dashboard ist Open Source und die Suche auf Elasticsearch lässt sich mittels der Lucene Query Syntax oder der vereinfachten Kibana Query Language durchführen. Elasticsearch speichert seine Daten in einer NoSQL Datenbank. Es lassen sich hier ebenfalls verschiedene

Visualisierungen wie Liniendiagramme, Flächendiagramme und Tabellen aus den Daten erstellen. ELK kann unzählige verschiedene Arten von Daten und Protokollen verschiedenster Systeme analysieren, verarbeiten und darstellen, was es zu einem sehr mächtigen Tool macht. Im Gegensatz zu Chiron-Elk wird hier zusätzlich MISP für die automatisierte indikatorenbasierte Suche verwendet. Die Daten von Zeek werden hier mittels Filebeat-Modul an Elasticsearch übertragen. Filebeat dient dabei der einfachen Übertragung von Logdaten. Es wurde statt Logstash genutzt, da es für meine Zwecke ausreicht und auch weniger ressourcenlastig ist. ELK Bezugsinformationen: (Elastic NV Version 7.10.0).

IVRE.ROCKS:

IVRE.ROCKS ist ein Netzwerkaufklärungstool mit diversen Einsatzmöglichkeiten. Es nutzt Tools wie Nmap, Massscan, ZGrab2, ZDNS und Zeek, dessen Daten in einer MongoDB Datenbank gespeichert werden. Massscan ist ein sehr schneller Port Scanner, der ähnliche Ergebnisse wie Nmap liefert. ZGrab2 ist ein Netzwerkscanner auf Applikationsebene, der mittels gelesenem Banner Dienste im Netzwerk zuordnen kann. ZDNS wiederrum dient als Dienst für DNS Lookups. Die Netzwerkdaten lassen sich in IVRE.ROCKS ebenfalls nach bestimmten Kriterien wie meistbenutzte Services, Ports oder Anbieter durchsuchen. Mit diesem Tool lässt sich zudem auch eine Heat Map erstellen, um Herkunft und Ziel der verschiedenen Netzwerkpakete anzuzeigen. IVRE.ROCKS lässt sich außerdem als eigenes Shodan, Zoomeye, Censys, Binaryedgeio nutzen. Dabei handelt es sich um verschiedene Computer-Suchmaschinen, auch OSINT Tools genannt, die bestimmte Arten von Computern und Diensten im Internet anzeigen. Jede der genannten Suchmaschinen hat dabei verschiedene Stärken. Hauptsponsor für dieses Projekt ist die französische Atomenergiekommission. Für dieses Programm wird vorzugsweise Kali-Linux oder Arch vorgeschlagen.
IVRE.ROCKS Bezugsinformationen: (GITHUB:Lalet, Pierre and Leroy, Emma and Monjalet, Florent and Mougey, Camille and Ruello, Vincent and Venuti, Viven Version: 0.9.16).

4.2 Entscheidungsmatrix

Kriterien	Gewichtung	ELK + Zeek + Dovehawk +MISP		Chiron-ELK		IVRE.ROCKS		Splunk + Zeek + Dovehawk + MISP + Corelight Splunk Plugin	
		Bewertung	*Total*	*Bewertung*	*Total*	*Bewertung*	*Total*	*Bewertung*	*Total*
Bedienbarkeit	10%	9	0,9	7	0,7	5	0,5	8	0,8
Dokumentation	10%	8	0,8	4	0,4	6	0,6	8	0,8
Support & Community	10%	9	0,9	4	0,4	8	0,8	8	0,8
Funktionsumfang & Schnittstellen	15%	8	1,2	3	0,45	6	0,9	8	1,2
Open Source	5%	10	0,5	10	0,5	10	0,5	6	0,3
Echtzeit-Darstellung	15%	7	1,05	8	1,2	4	0,6	7	1,05
Intelligente Datenverarbeitung	5%	10	0,5	10	0,5	0	0	10	0,5
Informationsgehalt	20%	8	1,6	7	1,4	5	1	7	1,4
Skalierbarkeit	5%	10	0,5	0	0	10	0,5	10	0,5
Wartbarkeit	5%	7	0,35	10	0,5	3	0,15	10	0,5
Total	**100%**		8,3		6,05		5,55		7,85

Bewertung: Punkte 0 (sehr schlecht) -10 (sehr gut)

Abbildung 4.1: Entscheidungsmatrix

Die Gewichtung wurde, wie in Punkt 3.3 erwähnt, in Prozenten aufgeteilt, wobei der Funktionsumfang der Lösung, die vollständige Echtzeit-Darstellung der Daten, sowie dessen Informationsgehalt den größten Nutzen darstellte. Wie bereits in Punkt 4.1 erwähnt, wurden diese Lösungen bereits im Vorfeld installiert und getestet, um den Rahmen nicht zu sprengen. Die Bewertung dessen findet in diesem Kapitel statt.
Die Ausprägung der Skala reichte von null (sehr schlecht) bis maximal zehn Punkten (sehr gut) für eine Kategorie. Je niedriger die Punktzahl, desto schlechter die Bewertung und desto weniger Gesamtpunkte erhält eine Lösung. Die Bewertung wird also mit der Prozentzahl multipliziert und in der Total-Zeile angezeigt. Das endgültige zusammenaddierte Ergebnis steht abschließend in der Total-Zeile unter dem Strich.

Den letzten Platz belegte hierbei IVRE.ROCKS. Trotz vieler angegebener Funktionalitäten waren die Installation und die Bedienung nicht intuitiv. Die Dokumentation war ok, dennoch gab es Probleme, bei denen die Entwickler und die Community aber helfen konnten. Die Einarbeitung dauerte lange und das Endergebnis in Bezug auf Threat Hunting mit den gegebenen Daten war eher befriedigend. Die Daten mussten zudem stets manuell neu importiert werden, um aktuell zu sein. Es gab eine Timeline-Funktion zur Darstellung der Daten der letzten 24h, diese war in meinen Augen aber nicht sehr aussagekräftig. Die einzigen überzeugenden Punkte waren die möglichen Funktionalitäten in Richtung Shodan.io, die Skalierbarkeit, sowie die Open Source Eigenschaft von IVRE.ROCKS. Zu den Wartungs- bzw. Update-Möglichkeiten wurden seitens der Entwickler nicht viele Infos veröffentlicht, es sieht aber danach aus, als müssten Pakete einzeln aktualisiert und auf Kompatibilität hin überprüft werden. Machine Learning oder irgendeine andere Form von intelligenter Datenverarbeitung gab es nicht.

Chiron Elk belegte den vorletzten Platz, trotz überzeugender Beschreibung. Die Dokumentation war nicht sehr hilfreich, so dass vieles ausprobiert werden musste, um ein besseres Verständnis von den Funktionen zu erlangen. Supporttechnisch gab es nicht viele Anlaufstellen für weitere Unterstützung, weshalb es hier auch nur wenig Punkte gibt. Die Bedienung und die gewonnenen Daten wirkten dafür etwas aufschlussreicher für den Threat-Hunting-Zweck

als die von IVRE.ROCKS. Die Daten mussten für eine grafische Liveansicht stets manuell aktualisiert werden. Dennoch gab es auch bei dieser Softwarelösung nicht genügend Funktionalitäten und Einstellungsmöglichkeiten zur Auswahl. Die Lösung ist nur für fünf bis zehn Nutzer angegeben, also nicht für größere Einsatzgebiete geeignet. Aufgaben wie Wartung und Pflege werden laut Entwickler selbstständig durchgeführt. Da die Funktion des Machine Learnings durch das Tool Aktaion gegeben ist und Chiron Elk Open Source ist, ergab sich in diesem Zusammenhang die volle Punktzahl.

Die Entscheidung zwischen Splunk + Zeek + Dovehawk + MISP + Corelight Splunk Plugin und ELK + Zeek + Dovehawk + MISP war hingegen knapper. Die ELK-Kombinationslösung überzeugt durch Open Source und einer größeren Community. Die Splunk-Kombinationslösung erhielt in diesem Bereich hingegen nur sechs Punkte, da Splunk und Corelight nicht Open Source sind. Dokumentationstechnisch sind beide gut aufgestellt. Die Daten lassen sich in ELK subjektiv gesehen etwas besser darstellen und kombinieren. Zusammenhänge lassen sich einfach zusammenstellen, ohne dass die Query Sprache dafür beherrscht werden müsste. Elasticsearch und Splunk besitzen beide eine Machine-Learning-Funktion, um Anomalien und Verhaltensmuster in Logfiles besser zu erkennen, weshalb es hier die volle Punktzahl gab. Beide Lösungen besitzen mindestens eine Schnittstelle, um z.B. Threat-Intelligence-Lösungen wie MISP einzubinden. Zudem gibt es bei beiden Lösungen etliche Funktionen, um die gesammelten Daten weiterzuverarbeiten, wobei bei Splunk mehr über zusätzliche Addons läuft (wie z.B. Corelight). Splunk Logs sind in der kostenlosen Version auf 500 MB Speichervolumen täglich begrenzt. Laut dem Blog von Asaf Yigal (vgl. Asaf Yigal 2017) sei der Einstieg in Splunk leichter, was im Laufe der Untersuchungen nicht festgestellt werden konnte. Eine komplette Echtzeitdarstellung existierte bei beiden Lösungen in dem Sinne nicht. Für eine aktuelle grafische Übersicht war manuelles aktualisieren erforderlich. Wartungstechnisch überzeugte Splunk zumindest laut Entwicklerdokumentation mit einer einfacheren Updateanleitung, sowie weniger Einzelpaketen, um die sich bei Wartungen gekümmert werden muss.

Die Kombinationslösung, die die Anforderungen dieser Ausarbeitung am besten erfüllt, ist hier also ELK + Zeek + Dovehawk + MISP mit einer Gesamtwertung von 8,3. Diese wird im nächsten Kapitel beispielhaft in einer Testumgebung installiert, um anschließend mit dem Threat Hunting fortzufahren.

5 Exemplarischer Einsatz der Ziellösung

5.1 Systemanforderungen

Für die Verfahrensentwicklung wird ein kleines Netzwerk benötigt, indem sich drei IoT-Geräte befinden, ein Standardgateway, um mit dem Internet zu kommunizieren, sowie ein recht leistungsstarker Server, auf dem die Ziellösung installiert wird. Um auch den Netzwerkverkehr zwischen den IoT-Geräten mitschneiden zu können, wird ein Managed Switch installiert und mittels Port Mirroring alle Daten von den Ports, auf denen die IoT-Geräte installiert sind, auf den Port mit dem angeschlossenen Server gespiegelt.

5.2 Aufbau der Testumgebung

Server:

Als Server Host wird ein zweiter Rechner genutzt. Der Rechner hat insgesamt 8GB Arbeitsspeicher, 240GB SSD-Festplattenspeicher und 4 Prozessorkerne mit je 2,9GHz Leistung. Als Betriebssystem wurde aufgrund von Herstellerempfehlung, Tutorials und eigener Erfahrungen Ubuntu 20.04.1 LTS gewählt (Canonical Ltd. Version: 20.04.1 LTS). Die grafische Benutzeroberfläche (GUI) blieb dabei aufgrund genügend vorhandener Hardwareressourcen eingeschaltet. Die Netzwerkkarte im Server ist zur Datenübertragung im Gigabit-Bereich fähig.

Netzwerk und Geräte:

Das Netzwerk ist ein klassisches Class C oder auch /24 Netzwerk mit rechnerisch möglichen 254 Stationen im Netzwerk.
Die folgenden Gerätschaften sind mit Ausnahme vom Echo Dot alle per 1Gbit Duplex-Lan-Kabel miteinander verbunden, um Engpässen im Netzwerk durch Überlastung vorzubeugen.
Der Router/Standardgateway, der auch als DNS Server in den IoT-Geräten hinterlegt ist, wird an Port eins eines Netgear GS308E 8 Port Gigabit Managed Switch angeschlossen.
Der Server ist mit dem Switch auf Port zwei verbunden.
Der Smart TV (LG 32LS575S, 2012) wird auf Port drei vom Switch angeschlossen. Auf Port vier befindet sich ein AVM FRITZ!-WLAN Repeater 450e (450 Mbit/s), der als Access Point für den Amazon Echo Dot (4.Generation) dient. Auf Port fünf ist eine IP-Kamera von Kamtron angeschlossen, die auch über das Internet erreichbar ist. Die Ports eins, drei, vier und fünf werden dabei mittels Port Mirroring auf Port zwei gespiegelt, um den internen Netzwerkverkehr zwischen den Geräten, als auch den externen Verkehr auf dem Server mitschneiden zu können.

Der Smart TV wurde u.a. gewählt, da dieser einer älteren Generation angehört (2012), aber dennoch als IoT-Gerät gilt. Es gibt keine Updates mehr für dieses Gerät und damit ist es zumindest theoretisch sicherheitstechnisch anfälliger für Lücken und Angriffe als z.B. der neue Amazon Echo Dot 4. Im Internet finden sich öfters geleakte, also nicht autorisiert veröffentlichte IP-Kamera-Streams und oft negative Presse über deren Sicherheit, weshalb sich als drittes Gerät dafür entschieden wurde. Insgesamt sind drei IoT-Geräte abgebildet, die vermehrt in Haushalten zu finden sind. Alle Geräte sind updatetechnisch auf dem letzten Stand, der vom Hersteller offiziell zur Verfügung gestellt wurde. In Abbildung 5.1 wird der genaue Netzwerkaufbau noch einmal grafisch verdeutlicht.

Netzwerkplan:

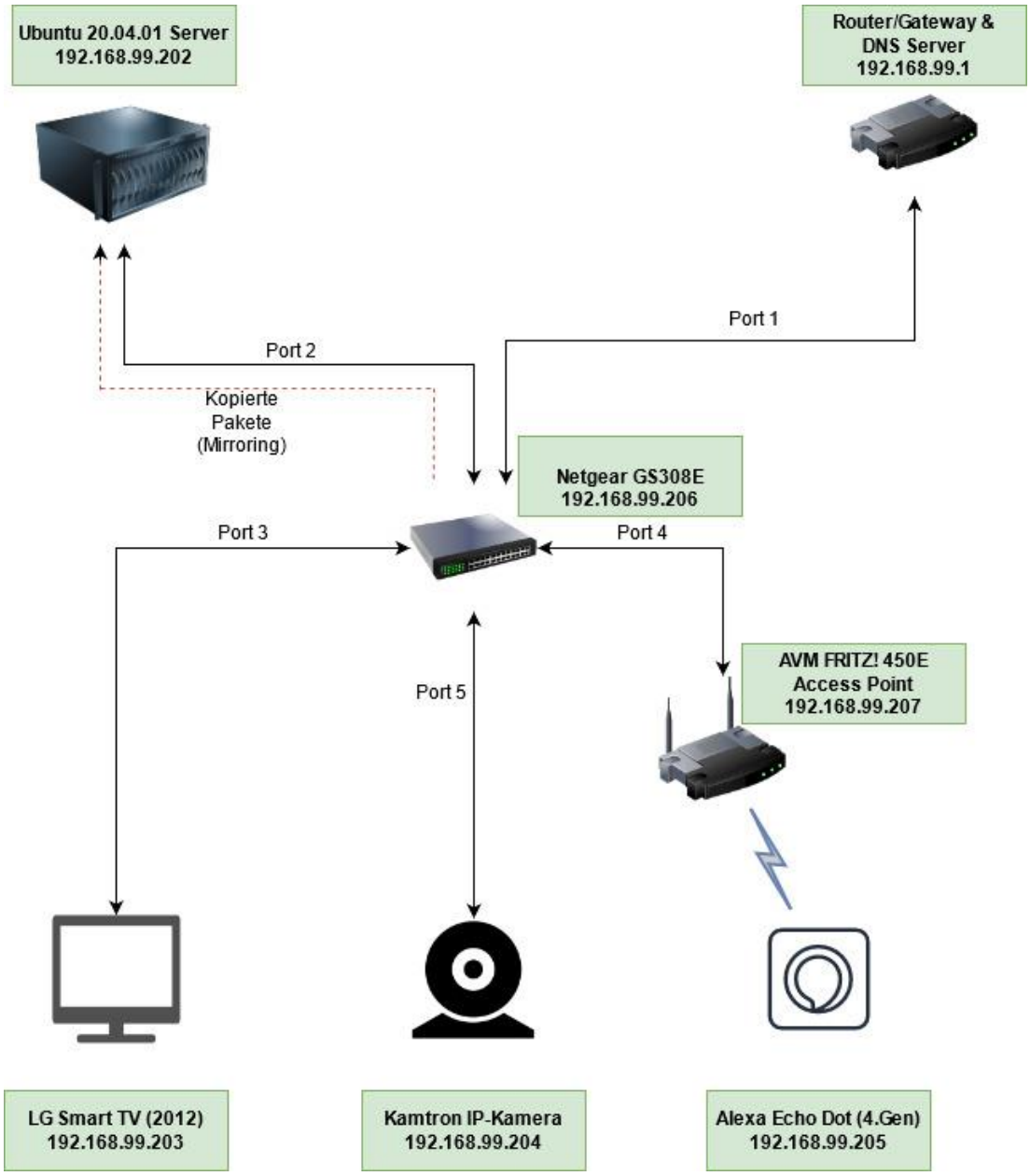


Abbildung 5.1: Netzwerkplan

5.3 Installation der Lösung

Zunächst wurde, wie in Punkt 5.2 erwähnt, Ubuntu 20.04.01 LTS installiert und auf den aktuellsten Stand gebracht. Anschließend wurden Zeek, Elasticsearch, Logstash, Kibana und Filebeat mitsamt zusätzlicher Zusatzmodule und Konfigurationen nach dem Kenntnisstand von Richard K. Medlin installiert und eingerichtet. (Richard K. Medlin 2020, 7-33, 56-90)

Wie bereits in Punkt 4.1 erläutert, schneidet Zeek dabei die Netzwerkpakete der Netzwerkkarte auf Metadaten-Ebene mit und speichert sie in Logdateien. Durch das in Punkt 5.2 erwähnte Port Mirroring werden sämtliche Netzwerkpakete der Geräte mitgeschnitten. Mittels Filebeat werden die Pakete anschließend an Elasticsearch gesandt und auf dem Kibana Dashboard grafisch dargestellt.

Weiterhin wurden über 30 zusätzliche Skripte vom oben genannten Autor Richard K. Medlin importiert und aktiviert. Die genaue Liste samt Erklärung der einzelnen Skripte findet sich im Anhang dieser Arbeit. Diese Skripte reagieren wie IoC's von Threat Intelligence Feeds auf bestimmte Verhaltensweisen im Netzwerk. Dabei gibt es z.B. Skripte zur Erkennung von möglichen SSH-Bruteforce-Angriffen anhand von bestimmten Paketgrößen und anderen Metadaten, die Erkennung von Port und IP Scans im Netzwerk, die Erkennung von Cryptomining etc. Online finden sich noch mehr Skripte, die sich ebenfalls alle je nach Einsatzzweck installieren und konfigurieren lassen. Es lassen sich auch eigene Skripte mit dem entsprechenden Wissen erstellen. Gespeichert werden erkannte Ereignisse dann im Notice.log, welche sich auch in Kibana darstellen lassen. Im Notice.log werden potenziell interessante Ereignisse gespeichert.

Um das Netzwerk zusätzlich mittels IoC's heruntergeladener Signaturen zu prüfen, wurde wie bereits in Punkt 4.1 erwähnt, eine Threat Intelligence Software namens MISP mittels Docker und das zugehörige MISP-Verbindungs-Modul für Zeek namens Dovehawk installiert und konfiguriert. Für den Testaufbau wurden drei verschiedene Threat Intelligence Feeds abonniert, diese beinhalten bereits mehr als 12000 generierte IoC's. Die Feeds waren „CIRCL

OSINT Feed“, „Malwaredomainlist“ und „Phishtank online valid phishing“. Werden diese IoC’s ausgelöst, sieht man das unter anderem in der Konsole, aber auch dank Dovehawk im MISP Webinterface unter dem entsprechenden Event als „Sighting“. Je nach ausgelöstem IoC muss entsprechend reagiert werden. Wie aus der Abbildung unten erkennbar, teilen sich die rund 12000 IoC’s in verschiedene Kategorien wie IP-Adressen, Domains, URL‘s etc auf. Die Indikatoren werden alle 4,5 Stunden mittels CronJob aktualisiert.

```
root@ubuntu:~/db# cd /opt/zeek/bin
root@ubuntu:/opt/zeek/bin# ./zeek -i enp1s0 -C dovehawk
listening on enp1s0

Downloading Signatures 2020/12/16 12:27:59 [1.02.001]
NETSTATS: pkts_dropped=0  pkts_recvd=1  pkts_link=45  bytes_recvd=1506
Local Directory: /opt/zeek/share/zeek/site/dovehawk/./scripts/.
MISP Server: https://localhost/
Downloading Indicators...
Downloading Signatures...
Updating File signatures.sig
    Finished Updating File: signatures.sig
    Signatures file contains: 0 signatures
Processing Indicators...
Number of Indicators 12292
 Intel Indicator Counts:
    Intel::DOMAIN:    1218
    Intel::ADDR:        3849
    Intel::URL:         587
    Intel::SUBNET:    0
    Intel::SOFTWARE:  1
    Intel::EMAIL:     32
    Intel::USER_NAME: 0
    Intel::FILE_HASH: 6097
    Intel::FILE_NAME: 507
Finished Processing Indicators
```

Abbildung 5.2: Initialisierung der Zeek- und Dovehawk-Software

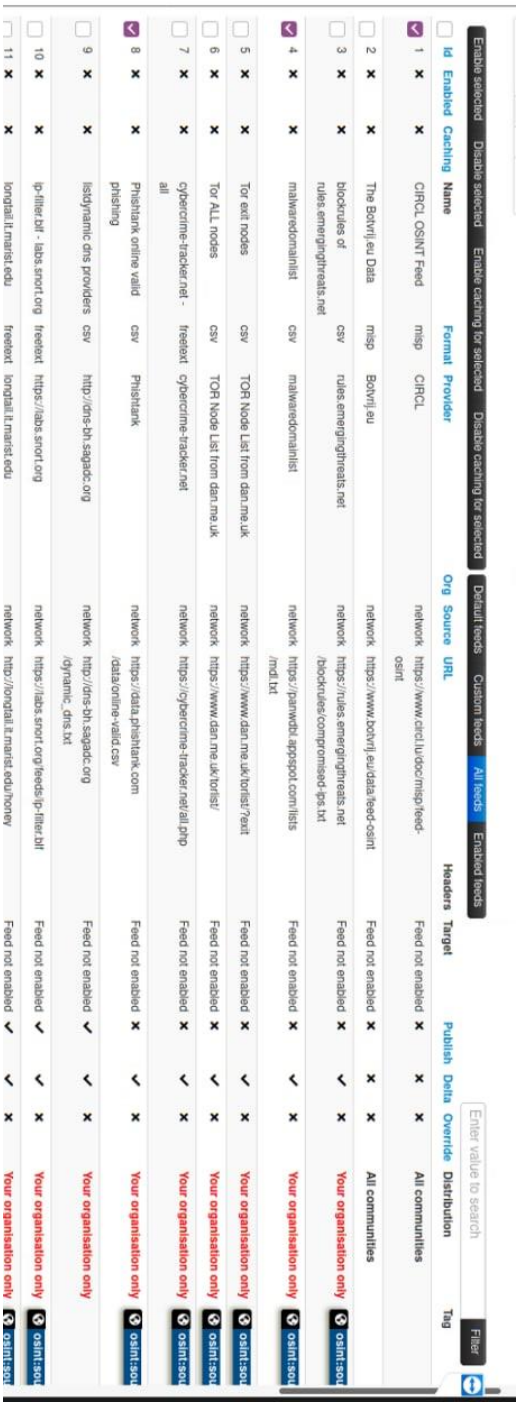


Abbildung 5.3: MISP Threat Intelligence Feeds

Bestimmte Feeds mit gefährlichen IP-Adressen lassen sich z.B. direkt in eine eigene Firewall importieren und blockieren.

Nachdem alle benötigten Dienste gestartet wurden, findet man die mitgeschnittenen Netzwerkdaten auf dem Kibana Dashboard wieder. Die Dateneigenschaften sind dabei extrem vielfältig. Es handelt sich um Hostnamen, Betriebssysteme, Quell- und Ziel-Ports, IP-Adressen, Paket und Zertifikatsinformationen etc.
Hier lassen sich nun verschiedene Dashboards konfigurieren, Queries auf der Suche ausführen und verschiedene Daten nach ihren Eigenschaften zusammenstellen, um sich eine bessere Übersicht in einem spezifischen Bereich zu verschaffen.
Leider ließen sich nicht alle benötigten Programme mittels Docker funktionell, sicher und aktuell umsetzen, weshalb dies nur teilweise genutzt werden konnte.

5.4 Testen der Lösung

Im folgendem findet sich ein Ausschnitt vom Kibana Dashboard mit den genannten Attributen und zwei Beispielpaketen, die aus meinem Netzwerk ins Netzwerk der HAW gesendet wurden. Auf der linken Seite finden sich die „Available Fields“, um z.B. nur bestimmte Felder anzuzeigen und entsprechend nach ihnen zu filtern. Hierbei lassen sich auch die häufigsten Werte als Dashboard visualisieren und weiter beliebig anpassen.

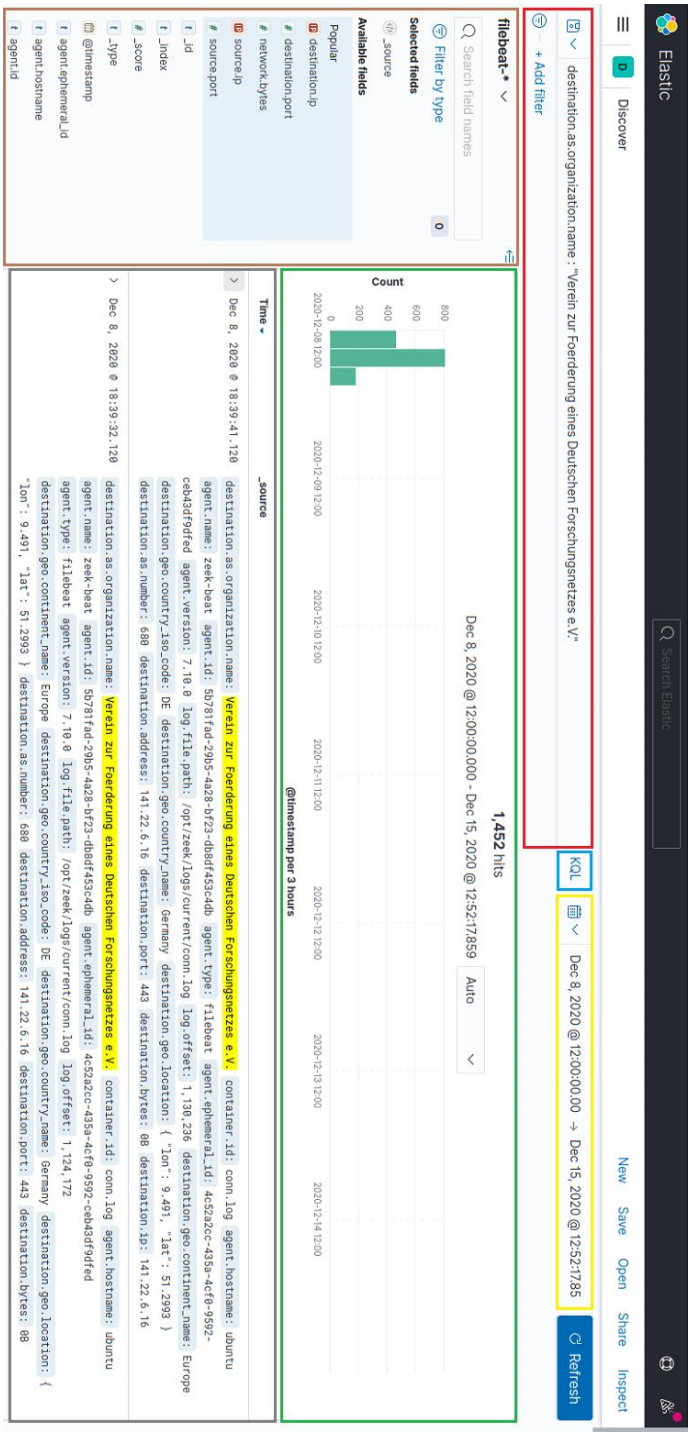


Abbildung 5.4: Kibana-Dashboard-Übersicht

In der Abbildung im oberen Teil befindet sich die Suche, bei der nach bestimmten Attributwerten gesucht werden kann (rot umrandet), wie z.B. IP-Adressen, Hostnamen, Ports etc. Im Grau umrandeten Bereich finden sich die übertragenen Datenpakete. In dem Fall oben sind zwei Beispielpakete, die in das Netz der HAW „Verein zur Förderung eines Deutschen Forschungsnetzes e.V“ gesendet wurden zu sehen. Es werden stets die ganzen Paketinformationen angezeigt, wenn sie mittels > Zeichen ausgeklappt werden (grau umrandeter Bereich) und wenn links nichts weggefiltert wird (braun umrandet).
Die Query-Syntax lässt sich im „KQL“ Feld (blau umrandet) von Kibana Query Language zu Lucene umstellen. Weitere Infos: (Elastic NV). Die Abfragesprache von Kibana basiert seit jeher auf der Lucene-Abfragesyntax. KQL ist als Standard voreingestellt und gefällt auch besser in der Handhabung. Weitere Infos: (Elastic NV). Zusätzlich lassen sich erweiterte Filter mittels Painless Scripting Language erstellen. Weitere Infos: (Elastic NV). Rechts neben dem KQL-Knopf lässt sich der Zeitraum für die angezeigten Datenpakete filtern (gelb umrandet). Oben mittig werden noch die Anzahl der Hits, also der Datenpakete angezeigt und darunter grafisch der Zeitraum, der die Anzahl der übertragenen Pakete zu einer bestimmten Uhrzeit darstellt (grün umrandet).

Weiterhin lassen sich auch komplett eigene Dashboards mit gewünschten Werten unter „Dashboards“ zusammenstellen, oder es wird das voreingestellte von Zeek benutzt, wie in der Abbildung 5.5 unten dargestellt.
Hier zu sehen sind die Aufteilung der meistgenutzten Protokolle, Aufteilung von Traffic auf intern/extern sowie eine Karte mit der Anzeige aller Länder und der Orte, zu denen die Pakete gesendet wurden. Dickere Punkte in rot und dunkelrot zeigen dabei besonders häufige Ziele auf. Weiter unten außerhalb des Bildes finden sich die meistgenutzten Domains, für die DNS-Abfragen gestellt wurden, die meistgenutzten URL Domains und die Anzahl an Sessions (also konstanten Verbindungen) über die Zeit. Zusammengefasst: Die gewählte Lösung überzeugte auch in der Praxis mit ihren Funktionalitäten und ihrer relativ einfachen Bedienung. Auch ohne zusätzliche Skript-Kenntnisse lassen sich schon recht viele Filter erstellen, die einen tieferen Einblick ins Netzwerk gewähren. Die Machine-Learning-Funktion zur automatischen Erkennung von Anomalien konnte leider nicht genutzt werden, da der

Prozessor des gewählten Servers die hardwaretechnischen Anforderungen dafür nicht leisten konnte. Sicherheitstechnisch lassen sich die verschiedenen Zugänge zu dem Server, Kibana, MISP etc. mittels komplexer Passwörter sichern. Zudem ist der Server nicht direkt aus dem Internet erreichbar, was die Sicherheit dessen natürlich signifikant erhöht. Nichtdestotrotz kann es in der Praxis nicht schaden seine Daten zusätzlich zu verschlüsseln und den Server weiter auf ungewollte Zugriffe und Verhaltensweisen zu überwachen und diesen auch softwaretechnisch stets auf dem aktuellsten Stand zu halten.

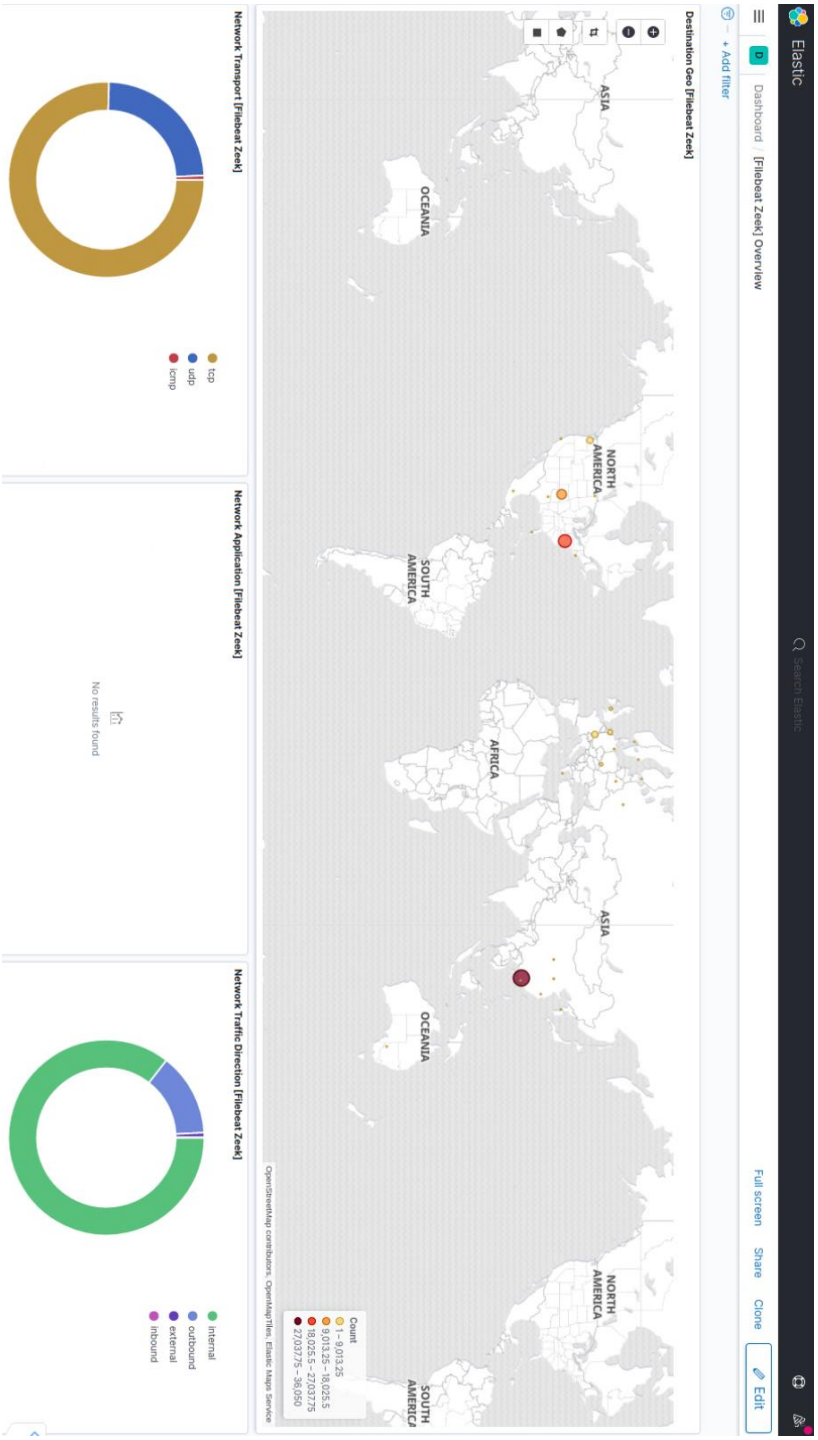


Abbildung 5.5: Vorkonfiguriertes Zeek Dashboard in Kibana

In Abbildung 5.6 wird anhand einer Skizze beispielhaft gemacht, wie die verschiedenen Softwarepakete der gewählten Ziellösung miteinander agieren. Dovehawk stellt dabei das Verbindungsstück zwischen Zeek und MISP da. Es lädt alle viereinhalb Stunden neue MISP-Indikatoren von MISP herunter und importiert diese in Zeek. Wird ein Indikator in Zeek ausgelöst, wird dies über Dovehawk an MISP zurück berichtet. Zeek-Daten werden mittels des ELK-Stack aufbereitet und dargestellt, MISP-Daten, also z.B. Sightings und die Verwaltung von Threat Intelligence Feeds und weiteren Indikatoren mittels des MISP Webinterface.

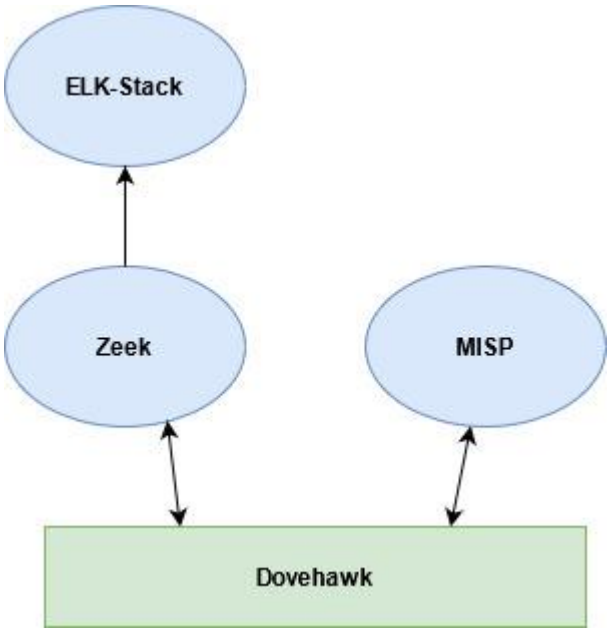


Abbildung 5.6: Skizze zur Kommunikation der gewählten Ziellösung

5.5 Threat Hunting in der Praxis

Wie in Punkt 5.2 erwähnt, besteht das Netzwerk aus drei IoT-Geräten, einem Server und dem Router. Um genügend Testdaten zu erzeugen waren alle Geräte eine Woche lang vom 16.12.20-23.12.20 im Dauerbetrieb und wurden ab und an kurz verwendet, um einen „normalen“ Umgang zu simulieren, wie dieser auch in anderen Haushalten stattfinden würde. Die Geräte waren alle sieben Wochentage im Betrieb, um sicherzugehen, dass bestimmte Netzwerkkontakte oder Updates, z.B. zum Hersteller nicht auf einen bestimmten Wochentag fallen, der dadurch nicht mitgeschnitten werden würde.
Aus der Vorgehensweise von Threat Hunting unter Punkt 2.3.4 lässt sich jetzt nach verschiedenen Anomalien im Netzwerk suchen.
Im folgendem sind 16 Beispielindikatoren im Netzwerk aufgelistet, von denen je IoT-Gerät vier bis fünf verschiedene Indikatoren mittels der gewählten Lösung überprüft werden, um den zeitlichen und inhaltlichen Rahmen nicht zu sprengen. Indikator 14 & 15 werden mithilfe der Zeek-Skripte aus Punkt 5.3 überprüft. Indikator 16 ist die Überprüfung des Netzwerkes mittels der genannten Threat-Intelligence-Anbieter über MISP, ebenfalls in Punkt 5.3 erklärt. Die Indikatoren entstammen verschiedenen Quellen wie (vgl. Eric Ooi), (vgl. Chris Brenton 2020) und (vgl. Derek King 2018). Dies ist nur die Spitze des Eisberges, um den Rahmen nicht zu sprengen, bleibt es bei der Anzahl. Damit die gefilterten Daten nicht zu unübersichtlich werden, teile ich die Query-Abfragen auf die einzelnen genutzten IoT-Geräte auf. Um mögliche User-Port-Zuweisungen (Port 1024-49151) herauszufinden, wurde die zuständige IANA befragt (vgl. Internet Engineering Task Force 2011). Diese ist jedoch nicht verpflichtend, so ist es gut möglich, dass ein anderer Dienst auf einem Port von jemandem genutzt wird als der, der bei der IANA hinterlegt wurde. Weiter genutzte Dienste zur weiteren Informationsverfolgung waren (Pagedesign GmbH), (Shodan) und die jeweilige Whois-Abfrage aus den (APNIC).

Genutzte Indikatoren:

1. Ungewöhnlich hohe Datenmengen im Down- und Upload in einem kürzeren Zeitraum <24h, die von der Norm abweichen, z.B. von einem bestimmten Gerät.
2. Große DNS-Abfragen > 75 Zeichen (mögliches DNS Tunneling oder DNS-Datenextrahierung).
3. DNS-Antworten mit NXDOMAINS (ins Nichts laufende Anfragen).
4. Ungewöhnliche Aktivitätszeiträume (spät in der Nacht, außerhalb von Arbeitszeiten).
5. Nutzen ungewöhnlicher Protokolle und Dienste durch IoT-Geräte. (z.B. SSH, SMB, RDP).
6. Überprüfen von Verbindungen über ungewöhnliche Ports >1024.
7. Überprüfen der Zielländer von Verbindungen (warum sollten meine Geräte nach Russland, Iran, etc. kommunizieren?).
8. Konstant aufrechterhaltende Verbindungen zwischen zwei Hosts.
9. Beaconing-Aktivitäten (z.B. kompromittiertes Gerät kontaktiert permanent C&C Server für neue Anweisungen, Paket z.B. immer gleich groß).
10. Große DNS-Antworten > 475 Zeichen.
11. Anzeigen der Top 10 Netzwerkverbindungen.
12. Top 10 Verbindungen anhand der meistgesendeten Daten.
13. Top 10 Verbindungen anhand der meistempfangenen Daten.

Notice.log (Prüfung anhand von Zeek-Skripten)

14. SSH Bruteforcing im Netzwerk.
15. Suchen nach abgelaufenen Zertifikaten.

MISP (Prüfung anhand von Threat Intelligence)

16. Prüfen von IoC's.

Wie bereits oben in Punkt 5.5 erwähnt, wird in den folgenden Fließtexten auf die Indikatoren einzeln eingegangen und jeweils vor den Satz die Indikatornummer geschrieben, auf die sich die Antwort bezieht. Das heißt, beim Smart TV werden die Indikatoren 1-5 geprüft, bei der IP-Kamera die Indikatoren 6-9 und beim Echo Dot die Indikatoren 10-13.

Smart TV (LG 32LS575S, 2012, 192.168.99.203):

1) Ungewöhnlich hohe Datenmengen im Down- und Upload in einem kürzeren Zeitraum <24h die von der Norm abweichen

Erst wurde einzeln nach Quelladresse und Zieladresse vom TV gefiltert, danach beides zusammen um Down/Uplink zu unterscheiden und den jeweiligen Traffic anzuzeigen.

Filter: *source.ip: 192.168.99.203*
Filter: *destination.ip: 192.168.99.203*
Filter: *source.ip: 192.168.99.203 or destination.ip: 192.168.99.203*

Der Fernseher war vor dem Testaufbau eine längere Zeit nicht eingeschaltet. Es gab bei der Ersteinschaltung an Tag eins und zwei des Experimentstarts deshalb einen erhöhten Netzwerkverkehrsfluss, ansonsten blieb der Traffic relativ konstant. Vermutlich hat der TV die ersten zwei Tage Kontakt zum Hersteller und anderer Software-Hersteller bezüglich Updates betrieben. Der TV versuchte keine Verbindungen zu den anderen Geräten im oben genannten Netzwerk unter Punkt 5.2 herzustellen, außer zum Router und natürlich ins Internet.

2) Große DNS-Abfragen > 75 Zeichen

Beim DNS Tunneling werden heimlich Daten zwischen anfragendem Client und kompromittiertem DNS Server ausgetauscht. Dabei können Daten vom Client exfiltriert (Data Exfiltration) oder einfache Befehle an den Client gesendet werden (Command and Control). Das funktioniert z.B. auch mit FTP, Netcat, SSH etc. ist also sehr wirkungsvoll (vgl. Klaus Nemelka 2020). Netcat

eignet sich zum Monitoren von Netzwerkverbindungen oder zur Übertragung von Daten jeglicher Art über das Netzwerk. DNS-Verkehr wird seltener wirkungsvoll überwacht oder gesperrt, weshalb diese Methode beliebt ist. Auffällig wird dies bei großen DNS Queries, denn dort ist die Wahrscheinlichkeit höher, dass auch andere Daten als DNS-Anfragen übertragen werden. Um über die Länge eines Strings nach großen DNS-Anfragen zu filtern wurde ein extra Skript mittels Painless Scripting Language erstellt.

Filter:

```
{
  "script": {
    "script": {
      "lang": "painless",
      "source": "if (doc['zeek.dns.query'].size() == 0) { return false;} else {return
doc['zeek.dns.query'].value.length() > 75;}"
    }
  }
}
```

source.ip: 192.168.99.203 or destination.ip: 192.168.99.203

Hierbei wurde anhand von Erfahrungswerten von einem IT-Security-Spezialisten (vgl. Eric Ooi) eine Mindestlänge von 75 Zeichen gewählt, dabei gab es aber keine Treffer, was schonmal positiv ist. Andernfalls müsste weiter in der Tiefe geforscht werden, um festzustellen, ob es sich wirklich um eine Kompromittierung handelt.

3) DNS-Antworten mit NXDOMAINS

NXDOMAINS, also Anfragen, die ins Nichts laufen, da kein „Server" für die Anfrage gefunden wurde, können auch ein Hinweis auf eine Datenexfiltration mittels Prinzips wie bei Punkt 2) oder nicht mehr existente verdächtige Domains sein. Werden also extrem viele „NXDOMAIN" Antworten von einem DNS Server an einen Client gesendet, kann dies ein Hinweis auf einen kompromittierten Client sein (vgl. Rainer Singer 2018).

Filter:
destination.ip: 192.168.99.203 and zeek.dns.rcode_name: "NXDOMAIN"

Hier gab es keine Anfragen, die im Nichts verlaufen sind.

4) Ungewöhnliche Aktivitätszeiträume

Unabhängig von den Indikatoren fiel direkt auf, dass der TV meistens gegen Mittag (12:00 Uhr) anfing Datenpakete zu versenden und zu erhalten. Leider musste dieser auch täglich neu eingeschaltet werden (gegen 10-11 Uhr morgens) trotz deaktiviertem Energiesparmodus und deaktivierter Zeitabschaltung.
Die Antwort auf die Frage, ob der Fernseher zu ungewöhnlichen Zeiten Netzwerkverkehr verursacht, konnte aus dem letzten Filter von 1) (mit or) übernommen werden, wobei sich klar herausstellte, dass nachts kein Traffic stattfand, weder im Up- noch im Downlink vom TV. Die grünen Balken zeigen den Netzwerkverkehr zu den entsprechenden Zeiten an.

Filter:
source.ip: 192.168.99.203 or destination.ip: 192.168.99.203

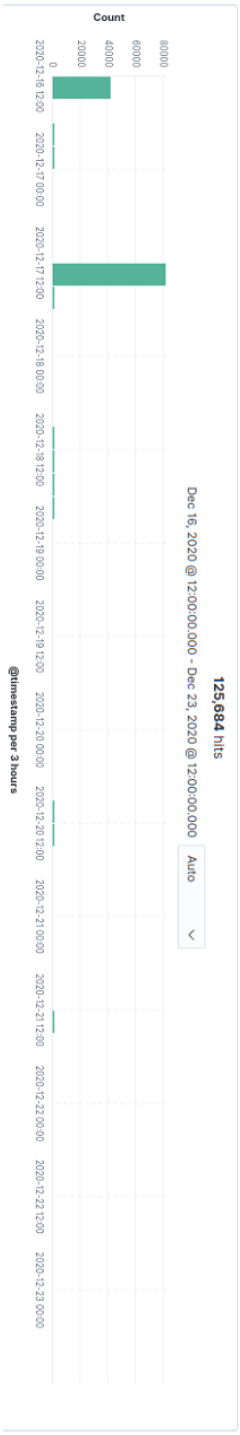


Abbildung 5.7: Erzeugter TV-Netzwerk-Traffic über eine Woche

5) Nutzen ungewöhnlicher Protokolle und Dienste

Normalerweise werden Protokolle wie SSH, SMB oder RDP nicht von IoT-Geräten genutzt, weshalb hier auf jene, in dem Fall ungewöhnliche Protokolle, überprüft wird.

Filter:
network.protocol : "ssh" or network.protocol : "smb" or network.protocol : "rdp"
select source.ip 192.168.99.203

select bedeutet, dass das jeweilige Feld auf der linken Seite ausgewählt wird (siehe Abbildung 5.4, brauner Bereich). Visualize bedeutet, dass das gewählte Feld z.B. zur weiteren Überprüfung grafisch dargestellt wurde in einem Dashboard.

Die Suche ergab dabei keine Ergebnisse. Der Smart TV nutzte nur die drei folgenden Protokolle: „HTTP, DNS und SSL“, sie wurden wie folgt gefiltert:

Filter:
source.ip: 192.168.99.203
select network.protocol

IP-Kamera (Kamtron, 192.168.99.204):

6) Überprüfen von Verbindungen über ungewöhnliche Ports >1024

Filter:
source.ip: 192.168.99.204 and destination.port > 1024

Die IP-Kamera nahm dabei permanent Kontakt zu einer IP-Adresse aus dem asiatischen Raum über Port 7024 mittels TCP-Protokoll auf. Laut IANA ist dieser Port für vmsvc oder auch vormetric service für tcp/udp registriert, was natürlich nicht heißen muss, dass es sich dabei um diesen Service handelt. Leider fanden sich auch nicht viel mehr Informationen über diesen Service. Bei der

Größe der Datenpakete könnte es sich wohl eher um Metadaten zur Nutzung als um Nutzungsdaten an sich handeln. Weiteres dazu wird in Punkt 9) überprüft. Entweder wurden die Netzwerkpakete als einzelne Pakete mit einer Größe von 52 Bytes (75%) oder dreimal hintereinander mit einer Gesamtgröße von 156Bytes (25%) versandt. Weiterhin tauchten Verbindungen zum Apple-Netzwerk und zweimal ins Telekom-Netzwerk auf, als mittels Smartphone eine beabsichtigte Verbindung aus dem Telekom-Netzwerk mit dem Livestream der IP-Kamera hergestellt wurde.

7) Überprüfen der Zielländer von Verbindungen

Filter:
source.ip: 192.168.99.204
select destination.geo.city_name

99,6% des Traffics gingen nach Jiangmen (China) und 0,4% über Frankfurt am Main (Deutschland), vermutlich über den Internet-Knoten DE-CIX. Da die IP-Kamera auch in China hergestellt wurde, liegt dies zumindest nahe, dennoch sollten in Punkt 6) genannte Maßnahmen getroffen werden.

8) Konstant aufrechterhaltende Verbindungen zwischen zwei Hosts

Zur genaueren Identifizierung der Pakete und der Nutzdaten wurden die Verbindungen von der IP-Kamera einmal zusätzlich mittels Wireshark für eine Stunde mitgeschnitten (Gerald Combs Version 3.2.3). Hierbei wurde zusätzlich zu den Erkenntnissen aus Punkt 6) noch ein permanenter Kontakt mittels ICMP-Protokolls festgestellt. Dieser Ping wurde so lange durchgehend ausgeführt, wie das Gerät eingeschaltet und mit dem Netzwerk verbunden war. Vermutlich wird so geprüft, ob die chinesische IP-Adresse erreichbar ist. Auf diesen Ping gab es aber kein Lebenszeichen als Antwort. Eine Verbindung wurde hier aber nicht aufrechterhalten in dieser Zeit.

Wireshark Filter:
Statistiken -> Verbindungen -> (IPV4)

9) Beaconing-Aktivitäten

Filter:
source.ip: 192.168.99.204
select @timestamp
select destination.ip
select network.bytes

Hierbei wird geprüft, ob ein Host permanent Kontakt zu einer oder mehreren IP-Adressen aufbaut. Wie bereits in Punkt 6) und 8) festgestellt, ist dies hier der Fall. Derselbe zeitliche Abstand und dieselbe Paketgröße können ein Hinweis auf eine Abfrage vom Client sein, der auf Befehle, bzw. Eingaben von seinem C&C Server wartet (vgl. Chris Brenton 2020).
In diesem Fall handelt es sich nur um eine Ziel-IP-Adresse, die in unregelmäßigen Abständen, aber dabei mindestens einmal pro Minute kontaktiert wurde. Wie bereits in Punkt 6) erwähnt, handelt es sich hierbei sehr wahrscheinlich nicht um Nutzungs- bzw. Streaming-Daten, da am Tag nur maximal 3,1MB übertragen wurden.
Um mehr über diese Datenübertragungen zu erfahren, wurden auch hier noch einmal mittels Wireshark die Verbindungen überprüft, um den Hintergrund der ständigen Verbindungsaufnahme nachvollziehen zu können. Die IP-Kamera prüfte, wie bereits aus Punkt 8) bekannt ist, permanent mittels ICMP, ob sein Gegenüber erreichbar ist (was nicht der Fall war). Anschließend wurde im Wechsel mit ICMP dauerhaft versucht, eine Verbindung mittels TCP zu der asiatischen IP-Adresse auf Port 7024 (SYN Flag) herzustellen. Diese Versuche schlugen ebenfalls fehl, weshalb es so aussieht, als wenn die hinterlegte IP-Adresse in der Software der Kamtron-Kamera bereits veraltet wäre. Welche Daten letztendlich bei einer erfolgreichen Verbindung ausgetauscht worden wären, konnte so leider nicht geprüft werden. Dieser Zustand der Verbindungsaufnahme zu einer nicht (mehr) verfügbaren IP-Adresse wirkt eher wie eine Anomalie im Netzwerk als eine konkrete Kompromittierung. Dennoch lässt sich dies nicht 100% ausschließen, weshalb es nicht schaden kann, diese permanenten Verbindungen mittels Firewall-Sperrung o.ä. zu unterbinden.

Filter für maximal übertragene Daten am Tag:
*destination.ip: IP-Adresse aus Asien (*Hierbei wurde mittels select von source.ip bereits gezeigt, dass kein anderer Host außer der IP-Kamera Kontakt hatte mit der entsprechenden IP-Adresse)
Visualize X-axis: @timestamp per day
Visualize Y-Axis: Sum of network.bytes

Wireshark Filter:
Statistiken -> Verbindungen -> (TCP)

Amazon Echo Dot (4. Generation, 192.168.99.205):

10) Große DNS-Antworten > 475 Zeichen

Große DNS-Antworten können ein Hinweis auf mögliche DNS Amplification Attacks sein. Kurz gesagt sind dies extrem große Antworten auf Anfragen eines Angreifers bei einem DNS Server, bei der die Antworten an das System des Opfers gesandt werden, um dessen Dienste/Verfügbarkeit lahmzulegen (vgl. Margaret Rouse 2020).

Filter:
destination.ip: 192.168.99.205

```
 {
  "script": {
    "script": {
      "lang": "painless",
      "source": "if (doc['zeek.dns.query'].size() == 0) { return false;} else {return
doc['zeek.dns.query'].value.length() > 475;}"
    }
  }
}
```

Die Suche ergab dabei keine Ergebnisse.

11) Anzeigen der Top 10 Netzwerkverbindungen

Filter:
source.ip: 192.168.99.204
select destination.ip

Die meistkontaktierte Adresse war in erster Linie der Netzwerkrouter, der auch als DNS Server in den Geräten hinterlegt ist. An zweiter Stelle folgten IP-Adressen von Amazon in den USA, was bei Alexa nicht ungewöhnlich sein muss. An dritter Stelle tauchte eine Multicast-DNS-Adresse auf. Multicast DNS wird genutzt, um in kleineren Netzwerken bei der Namensauflösung zu helfen. Dabei wird im Gegensatz zum „normalen" DNS nicht ein DNS Server befragt, sondern alle Teilnehmer im Netzwerk direkt angesprochen. Der Sender fragt dabei nach, zu welchem Teilnehmer im Netzwerk der angefragte Hostname passt. Für diese Art von DNS kommen nur Hostnamen mit der .local Endung in Frage, weshalb Multicast DNS auf lokale Netzwerke beschränkt ist (vgl. 1&1 IONOS SE 2020). Die letzte aus dem Rahmen fallende IP-Adresse gehörte zur "Amazon CloudFront", einem Netzwerk zur Bereitstellung von Inhalten des AWS. Weitere Informationen bezüglich Cloud und IoT-Geräte finden sich unter Punkt 2.1.1.

12) Top 10 Verbindungen anhand der meistgesendeten Daten

Filter:
source.ip: 192.168.99.205
Visualize x-axis: Top ten Values of destination.ip
Visualize y-axis: Sum of network.bytes

Die zehn meistgenutzten Ziel-IP-Adressen Für den Zeitraum vom 16.12.2020 bis zum 23.12.2020 gehörten allesamt zur "Amazon CloudFront". Der Traffic war dabei pro Adresse von 6-106Mb groß, insgesamt wurden knapp 600Mb in dieser Zeit an diese IP-Adressen übertragen. Es sieht also so aus, als wenn die anfallenden Daten des Echo Dots nach gewissen Kriterien aufgeteilt an verschiedene Server gesandt werden. Möglicherweise handelt es sich aber bloß um Load Balancer, um die Lasten des Traffics zu verteilen.

13) Top 10 Verbindungen anhand der meistempfangenden Daten

Filter:
destination.ip: 192.168.99.205
Visualize x-axis: Top ten Values of source.ip
Visualize y-axis: Sum of network.bytes

Hierbei wurde nur geringer Traffic im Kilobyte Bereich von zwei verschiedenen IP-Adressen erzeugt. An erster Stelle der Quell-IP-Adressen war dabei der Smart TV, gefolgt von dem Router. Der Smart TV benutzte viele verschiedene Quellports und versuchte stets Port 50000 oder 50001 vom Echo Dot per UDP zu erreichen (leider ließen sich dazu nicht mehr Informationen finden). Der Router benutzte ausschließlich ICMP, also höchstwahrscheinlich einfache Pings, um zu prüfen, ob das Gerät noch aktiv ist. Dies passierte 1-3x pro Tag.

Notice.log (Prüfung anhand von Zeek-Skripten):

Die bislang geprüften Indikatoren wurden nur für je ein Gerät durchgeführt, um den Rahmen nicht zu sprengen. Folgende Indikatoren lassen sich aber leicht für das ganze Netzwerk prüfen anhand der vorgefertigten Zeek-Skripte (siehe Punkt 5.3), die unter anderem nur in bestimmten oben genannten Situationen ein Ereignis in das Notice.log schreiben.

14) SSH Bruteforcing im Netzwerk

Mit dem genutzten Filter von 5) konnte bereits beim SmartTV festgestellt werden, dass kein SSH Traffic für dieses Gerät im Netzwerk entstanden ist. Im Notice.log waren auch keine anderen fehlgeschlagenen SSH-Authentifizierungsversuche zu finden, weshalb es in diesem Netzwerk keine SSH-Bruteforcing-Attacken gab.

Filter:
log.file.path :"/opt/zeek/logs/current/notice.log"

15) Suchen nach abgelaufenen Zertifikaten

Mittels Notice.log lassen sich ebenfalls Verbindungen mit abgelaufenen, sowie selbstsignierten Zertifikaten anzeigen.
Selbstsignierte Zertifikate sind dabei nicht von einer vertrauenswürdigen Quelle ausgegeben und können auf Betrüger oder schlecht konfigurierte Seiten schließen. Im SSL.log findet man dabei alle SSL/TLS Handshakes.

Filter:
log.file.path :"/opt/zeek/logs/current/notice.log"
select source.ip

Mittels dieses Filters ließ sich herausfinden, dass die einzige IP-Adresse mit abgelaufenen oder selbstsignierten Zertifikaten in der Verbindung der Fernseher war. Dieser versuchte 17x IP-Adressen mit oben genannten Merkmalen zu erreichen. 94% der Adressen gehörten dabei laut dem destination.as.organization.name Attribut zu Amazon und 6% zu Netflix. Mittels weitergehender Überprüfung durch Shodan.io konnte dies auch bestätigt werden. Das Zertifikatsproblem betrifft vermutlich nur deshalb den Smart TV, da dieser schon älter ist und keine Updates mehr bekommt. Hinterlegte IP-Adressen könnten also bereits nicht mehr aktuell sein.

MISP (Prüfung anhand von Threat Intelligence)

16) Prüfen von IoC's

Wie bereits in Punkt 5.3 erwähnt, wurden drei verschiedene TI (Threat Intelligence) Anbieter abonniert. Anhand dessen wird hier stichprobenartig gezeigt, wie die damit erstellten Indikatoren auf „Sightings" geprüft werden können.
Es gibt viele verschiedene Konfigurationen, Einstellungsmöglichkeiten, oder Arten mit MISP oder dessen API zu arbeiten, dieser Weg ist nur einer und möglicherweise nicht der simpelste.
In Abbildung 5.3 sind die abonnierten Feeds noch einmal zu sehen. Entweder wird auf den entsprechenden Tag bei einem Feed geklickt, wodurch sich zu dem oder den erstellten Events gelangen lässt, oder es werden alle Events

angezeigt mittels „Event Actions“ -> „List Events“. Durch Filtern des Erstelldatums, an dem die entsprechenden Feeds im System aktiviert wurden, finden sich die entsprechenden Events. Durch das Klicken auf ein Event, zeigt das System die Übersicht und im unteren Teil die benutzten Attribute an.

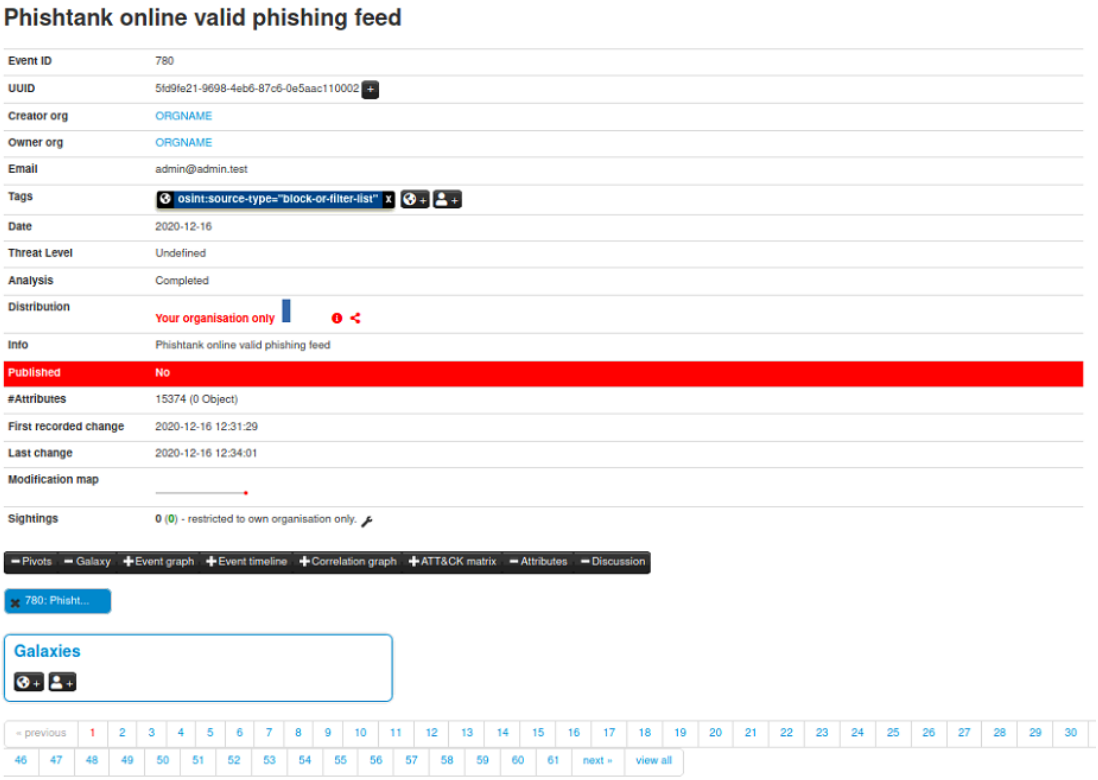


Abbildung 5.8: MISP Event „Phishtank“

An dieser Stelle findet sich die Event-Übersicht ohne die 15374 Attribute im Einzelnen. Es wurden in der gescannten Zeit keine „Sightings“ ermittelt, was natürlich positiv zu beurteilen ist. Weiterhin lassen sich die Sightings auch als JSON-Dokument oder als Rohdaten mittels „Server-IP/Sightings“ als JSON-Dokument oder als Rohdaten im gewählten Internetbrowser anzeigen.
Ein Sighting heißt nicht unbedingt, dass es sich hierbei auch um etwas „Bösartiges“ handelt. Ob z.B. eine URL „böse“ ist, hängt immer noch vom Kontext des Benutzers bzw. der Arbeitsumgebung ab. Es gibt sogenannte MISP

Warninglists die bekannte „False Positives“, also fälschliche „Sightings“ beinhalten (vgl. Computer Incident Response Center Luxembourg 2021).

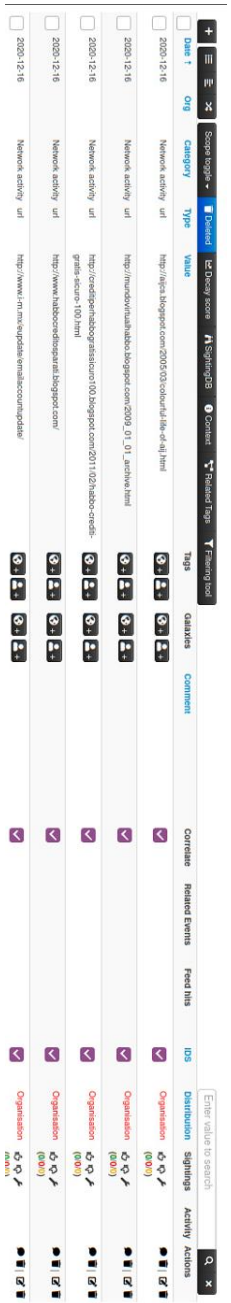


Abbildung 5.9: Fünf Attribute aus dem Phishtank Event in MISP

In Abbildung 5.9 hierdrüber findet sich eine Übersicht von fünf Beispielattributen aus dem oben genannten Feed, die den Netzwerkverkehr auf bestimmte aufgerufene URL's überprüfen.

5.6 Ergebnisbewertung

Ziel dieses Testaufbaus und des „Hunten" an sich war es, kompromittierte Geräte im Netzwerk zu identifizieren, schädliche Verhaltensweisen aufzudecken, sowie Anomalien zu erkennen.
Wie bereits vermutet, finden sich mittels stichprobenartiger Abfragen in dem kleinen Testnetzwerk keine stichhaltigen Beweise für eine Kompromittierung. Die Ergebnisse sind dennoch aufschlussreich, da sich ein spannender Einblick in den Netzwerkverkehr der genutzten IoT-Geräte bietet, wodurch auch ein besseres Gefühl für auffällige Datenpakete, Verhaltensweisen und Anomalien, die es hier ja z.B. bei der IP-Kamera gegeben hat, entsteht. Es wurden zudem einige interessante Indikatoren aufgezeigt, die sich auch in anderen Netzwerken gut überprüfen lassen. Nichtdestotrotz ist dies nur die Spitze des Eisberges und gerade als Laie im Sicherheitsbereich gibt es noch viele Dinge, die unbekannt sind. Es gibt noch weitaus mehr Tools, Indikatoren und andere Kniffe, mit denen noch besser gefiltert, sowie bessere Ergebnisse geliefert werden können. Das oben genutzte Threat-Hunting-Verfahren lässt sich am ehesten der zweiten Kategorie von Punkt 2.3.1 zuordnen und entspricht damit einer Kombination aus indikatorbasiertem Threat Hunting und der teilweise manuellen Suche nach Anomalien und Verhaltensmustern.

6 Fazit

6.1 Zusammenfassung

Schwerpunkt dieser Arbeit war es ein Verfahren in der IT-Sicherheit zu entwickeln, das vorhandene IoT-Geräte im Netzwerk auf Anomalien, Verhaltensmuster und schädliches Verhalten hin untersucht. Dabei war es das Ziel herauszufinden, ob es ein Angreifer bereits ins Netzwerk geschafft hat, mittels Ausnutzung von möglichen Schwachstellen in den vorhandenen IoT-Geräten.

Dafür wurden in der Vorgehensweise dieser Ausarbeitung zuerst Grundlagen zum aktuellen Stand der Forschung analysiert und in Kapitel zwei in die drei Kategorien Internet of Things, Threat Intelligence und Threat Hunting unterteilt. Die Recherche machte dabei einen Hauptteil dieser Arbeit aus, was unter anderem daran liegen könnte, dass der Themenbereich Threat Hunting noch nicht so publik und „alt“ ist, wie z.B. das Thema Firewalls. Infolgedessen musste genau recherchiert und verschiedene Quellen gegeneinander abgewogen werden.
Aufbauend auf den damit geschaffenen Grundlagen wurde in Kapitel drei eine Anforderungsanalyse für die nötige Software zur Umsetzung des Verfahrens ermittelt. Dabei kristallisierten sich die wichtigsten Kriterien teilweise erst beim Testen der verschiedenen Lösungen heraus. Manche Lösungen erschienen im Vorfeld tauglicher als sie es dann im Test wirklich waren. Dies lag zum Teil an undurchsichtigen Dokumentationen, als auch an schlechter Bedienbarkeit. Manche Lösungen mussten vor dem Testen erst auf die eigene Testumgebung angepasst werden, bevor sich dann herausstellte, dass diese doch ungeeignet sind. Dadurch wurden in diesem zeitaufwendigen Prozess manche Softwarelösungen schnell wieder verworfen. Dies war der zweite große Schwerpunkt in dieser Arbeit.

Die vier überzeugendsten Lösungen aus den Vorabtests wurden daraufhin in Kapitel 4.1 vorgestellt, ausgiebig getestet und in der Entscheidungsmatrix von Kapitel 4.2 miteinander verglichen und bewertet.

In Kapitel fünf galt es zunächst die Systemanforderungen für die Testumgebung festzustellen und diese Umgebung anschließend zu erstellen.
Die durch Kapitel 4.2 ermittelte Kombination der Zielsoftware aus ELK + Zeek + Dovehawk + MISP wurde dann in Kapitel 5.3 exemplarisch installiert und eingerichtet, wobei in Punkt 5.4 zum besseren Verständnis hin ein Einblick in die Funktionen der Ziellösung geboten wurde.

In Kapitel 5.5 entstand der letzte große Schwerpunkt dieser Arbeit. Der Praxisteil des Threat Huntens an sich, bei dem es galt passende Indikatoren und Ansätze zur Überprüfung für das Netzwerk zu finden und daraus sinnvolle und interessante Ergebnisse zu ermitteln. Dokumentationstechnisch findet sich zwar vieles zur reinen Installation und Konfiguration von Softwarelösungen, nicht aber wie explizit mit ihnen nach Verhaltensweisen und Mustern gesucht werden kann. Dieses Wissen wurde zum Teil selbst generiert, als auch aus verschiedenen Quellen zusammengesucht und auf die eigene Umgebung angepasst.
Wie bereits in Punkt 5.6 erwähnt, gab es zwar keine bestätigte Kompromittierung im Netzwerk, dafür aber einige spannende Erkenntnisse zu den Verhaltensweisen der IoT-Geräte, über die man normalerweise nicht stolpern würde.

6.2 Fazit

Durch das hier erstellte Verfahren bietet sich ein guter Einstieg, um IoT-Geräte im Netzwerk systematisch auf Anomalien oder schädliches Verhalten zu überprüfen. Ebenfalls aufgezeigt wurden Schwerpunkte als auch Schwierigkeiten des Threat Huntings.
Für das Verfahren wurden verschiedene Softwarelösungen miteinander verglichen, wobei durch eigens erstellte Kriterien eine Ziellösung gefunden werden konnte.

Mittels der erarbeiteten Ziellösung der Aufgabenstellung wurden Funktionen und Arbeitsweise dieser in einer Testumgebung vorgestellt und beispielhaft demonstriert, um der Zielaufgabenstellung gerecht zu werden.
Dabei wurden im Netzwerk der Testumgebung Anomalien bei IoT-Geräten festgestellt, wie aus Kapitel 5.5 oder der Ergebnisbewertung in Abschnitt 5.6 hervorgeht. Es wurde zudem gezeigt, wie normaler Netzwerkverkehr von einer Anomalie unterschieden wird und wie erkannte Anomalien anschließend z.B. mit weiterer Software untersucht werden können.
Mittels der Grundlagen zur Vorgehensweise im IoT-Bereich in Kapitel 2.3.3-2.3.4 sowie den gewählten Indikatoren in Kapitel 5.5 bietet sich zudem ein erster guter Einstieg für die Frage „Wonach suche ich und was möchte ich eigentlich mittels Threat Hunting und ggf. Threat Intelligence erreichen?“ an.
Das Kapitel „Threat Hunting in der Praxis“ dieser Arbeit benötigt noch weitere Einarbeitungszeit, um die Möglichkeiten von MISP, Zeek, ELK und dem Threat Hunting an sich vollständig auszuschöpfen. Eine vollständige Auswertung des Netzwerkes würde den Rahmen dieser Arbeit aber sprengen.

6.3 Ausblick

Aufbauend auf der hier geschaffenen Grundlage sowie der Testumgebung und den daraus resultierenden Erkenntnissen lässt sich das Projekt noch ausgiebig weiterführen. Eigens erstellte Zeek-Skripte, weitere Threat Intelligence Feeds, oder gar eigene IoC’s, sowie das Verwenden von Logstash zur erweiterten Datenaggregation sind nur ein Beispiel dafür. Weiterhin lassen sich die Daten noch zusätzlich automatisiert mittels Machine Learning auf Anomalien überprüfen und die Indikatoren für das Threat Hunting in ELK beliebig erweitern. Zudem gibt es bereits Lösungen, bei denen die MISP-Daten in ELK weiterverarbeitet und angezeigt werden können.
Ausgehend vom Machine Learning, automatisierten Benachrichtigungen bei ausgelösten Indikatoren, sowie neuen Ereignissen im Notice Logfile lassen sich Teile des Hunting-Prozesses vereinfachen. Dennoch bleibt Threat Hunting eine Interaktion zwischen Mensch und Maschine, die sich nicht vollständig automatisieren lässt.

Literaturverzeichnis

Literaturverzeichnis

1 1&1 IONOS SE (Hg.) (2020): Multicast DNS: Alternative Namensauflösung im kleinen Stil. Online verfügbar unter https://www.ionos.de/digitalguide/server/knowhow/multicast-dns/, zuletzt geprüft am 12.01.2021.

2 Amazon Web Services, Inc. (Hg.): AWS IoT Core – Häufig gestellte Fragen. Online verfügbar unter https://aws.amazon.com/de/iot-core/faqs/, zuletzt geprüft am 26.02.2021.

3 Amazon Web Services, Inc. (Hg.): Funktionsweise von AWS IoT. Online verfügbar unter https://docs.aws.amazon.com/de_de/iot/latest/developerguide/aws-iot-how-it-works.html, zuletzt geprüft am 26.02.2021.

4 Anand Tamboli (2019): Why Should You Build Your Own IoT Platform? Hg. v. Medium. Online verfügbar unter https://medium.com/tomorrow-plus-plus/why-should-you-build-your-own-iot-platform-dff51578c0c, zuletzt geprüft am 26.02.2021.

5 APNIC (Hg.): World Whois Databases. APNIC. Online verfügbar unter https://www.apnic.net/about-apnic/whois_search/about/what-is-in-whois/which-whois/, zuletzt geprüft am 19.01.2021.

6 Arndt Borgmeier; Alexander Grohmann; Stefan F. Gross (2017): Smart Services und Internet der Dinge: Geschäftsmodelle, Umsetzung und Best Practices: Carl Hanser Verlag GmbH & Co. KG.

7 Arran Purewal; Peter Schmitz (2020): Menschliche Verteidigung gegen Cyberangriffe. Hg. v. Vogel IT-Medien GmbH. Online verfügbar unter https://www.security-insider.de/menschliche-verteidigung-gegen-cyberangriffe-a-949001/, zuletzt geprüft am 03.12.2020.

8 Asaf Yigal (2017): Splunk and the ELK Stack: A Side-by-Side Comparison. Hg. v. MediaOps, Inc. Online verfügbar unter https://devops.com/splunk-elk-stack-side-side-comparison/, zuletzt geprüft am 16.03.2021.

9 Bitdefender GmbH (Hg.) (2019): Was macht das Internet der Dinge so unsicher? Online verfügbar unter https://www.bitdefender.de/box/blog/iot-neuigkeiten/sie-fragen-wir-antworten-5-was-macht-das-internet-der-dinge-so-unsicher/, zuletzt geprüft am 09.12.2020.

10 BMWi (Hg.): Was ist Industrie 4.0. Online verfügbar unter https://www.plattform-i40.de/PI40/Navigation/DE/Industrie40/WasIndustrie40/was-ist-industrie-40.html, zuletzt geprüft am 09.12.2020.

11 Brent Murphy; David French (2020): The Elastic Guide to Threat Hunting. Hg. v. Elastic NV. Online verfügbar unter https://www.elastic.co/pdf/elastic-guide-to-threat-hunting, zuletzt geprüft am 05.12.2020.

12 BSI (Hg.): CERT-Bund. Online verfügbar unter https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Cyber-Sicherheitslage/Reaktion/CERT-Bund/cert-bund_node.html, zuletzt geprüft am 05.03.2021.

13 Canonical Ltd. (Hg.) (Version: 20.04.1 LTS): SOFTWARE:UBUNTU. Online verfügbar unter https://ubuntu.com/download/desktop, zuletzt geprüft am 20.12.2020.

14 Chris Brenton (2020): Threat Hunting IoT and IIoT Devices. Hg. v. Active Countermeasures. Online verfügbar unter https://www.activecountermeasures.com/threat-hunting-iot-and-iiot-devices/, zuletzt geprüft am 28.12.2020.

15 Chris Brook (2020): What is Threat Intelligence? Hg. v. Digital Guardian. Online verfügbar unter https://digitalguardian.com/blog/what-threat-intelligence, zuletzt geprüft am 11.12.2020.

16 Computer Incident Response Center Luxembourg (Hg.) (2021): MISP warninglists. Online verfügbar unter https://circl.lu/doc/misp/warninglists/, zuletzt aktualisiert am 2021, zuletzt geprüft am 17.03.2021.

17 Corelight Inc. (Version: 2.1.0): Corelight App For Splunk. Online verfügbar unter https://splunkbase.splunk.com/app/3884/, zuletzt geprüft am 05.03.2021.

18 CrowdStrike Holdings, Inc. (Hg.) (2019): What is Proactive Threat Hunting? Online verfügbar unter https://www.crowdstrike.com/epp-101/threat-hunting/?utm_campaign=dsa&utm_content=ceu&utm_medium=sem&utm_source=goog&utm_term=&_bt=458397644071&_bk=&_bm=b&_bn=g&_bg=111187839310&gclid=EAIaIQob-ChMIp6XM3sby6wIVyud3Ch3pzg-GEAAYAiAAEgKSRfD_BwE, zuletzt geprüft am 03.12.2020.

19 Derek King (2018): Hunting Your DNS Dragons | Splunk. Hg. v. Splunk, Inc. Online verfügbar unter https://www.splunk.com/en_us/blog/security/hunting-your-dns-dragons.html, zuletzt geprüft am 28.12.2020.

20 DOCKER:harvarditsecurity (2020): misp-docker. Hg. v. Docker, Inc. Online verfügbar unter https://hub.docker.com/r/harvarditsecurity/misp, zuletzt geprüft am 16.12.2020.

21 Dr. Svilen Ivanov (2018): Was ist Cyber Threat Intelligence und wofür kann man es nutzen? Unter Mitarbeit von Dr. Christian Gorecki, Dr. Alexander Hullmann. Hg. v. ISACA. Online verfügbar unter https://www.isaca.de/sites/default/files/isaca_fokus_bonn_cti_ivanov_2018-06-28.pdf, zuletzt geprüft am 08.12.2020.

22 Ed Moyle (2019): CERT vs. CSIRT vs. SOC: What's the difference? Hg. v. TechTarget. Online verfügbar unter https://searchsecurity.techtarget.com/tip/CERT-vs-CSIRT-vs-SOC-Whats-the-difference, zuletzt geprüft am 01.03.2021.

23 Egon Kando (2019): Threat Intelligence – die Grundlage für Cybersicherheit. Hg. v. IT Verlag für Informationstechnik GmbH. Online verfügbar unter https://www.it-daily.net/it-sicherheit/cloud-security/21441-threat-intelligence-die-grundlage-fuer-cybersicherheit, zuletzt geprüft am 08.12.2020.

24 Elastic NV (Hg.): Kibana Query Lanaguage. Online verfügbar unter https://www.elastic.co/guide/en/kibana/master/kuery-query.html, zuletzt geprüft am 04.02.2021.

25 Elastic NV (Hg.): Lucene Query Syntax. Online verfügbar unter https://www.elastic.co/guide/en/kibana/current/lucene-query.html, zuletzt geprüft am 04.02.2021.

26 Elastic NV (Hg.): Painless scripting language. Online verfügbar unter https://www.elastic.co/guide/en/elasticsearch/reference/master/modules-scripting-painless.html, zuletzt geprüft am 07.01.2021.

27 Elastic NV (Hg.) (Version 7.10.0): SOFTWARE:ELK-Stack. Online verfügbar unter https://www.elastic.co/de/, zuletzt geprüft am 15.12.2020.

28 Endace Ltd. (Hg.): Indicators of Attack (IOA) Indicators of Compromise (IOC). Online verfügbar unter https://www.endace.com/solutions/cyber-security/threat-hunting/ioas-and-iocs, zuletzt geprüft am 04.12.2020.

29 Eric Ooi: Zeekurity Zen – Part IV: Threat Hunting With Zeek. Hg. v. LLC. OOI Ventures. Online verfügbar unter https://www.ericooi.com/zeekurity-zen-part-iv-threat-hunting-with-zeek/, zuletzt geprüft am 04.12.2020.

30 F-Secure Corporation (Hg.): Threat Hunting im Klartext. Online verfügbar unter https://www.f-secure.com/content/dam/press/de/media-library/reports/F-Secure-Threat-Hunting-Whitepaper%20(German).pdf, zuletzt geprüft am 07.12.2020.

31 Gerald Combs (Version 3.2.3): SOFTWARE:Wireshark. Hg. v. Gerald Combs. Online verfügbar unter https://www.wireshark.org/, zuletzt geprüft am 01.02.2021.

32 GITHUB:Joseph Zadeh; Rod Soto; mykelxknight (2020): Chiron-ELK. Hg. v. GitHub, Inc. Online verfügbar unter https://github.com/jzadeh/chiron-elk, zuletzt geprüft am 04.03.2021.

33 GITHUB:Lalet, Pierre and Leroy, Emma and Monjalet, Florent and Mougey, Camille and Ruello, Vincent and Venuti, Viven (Version: 0.9.16): IVRE.ROCKS. Hg. v. GitHub, Inc. Online verfügbar unter https://github.com/cea-sec/ivre, zuletzt geprüft am 04.03.2021.

34 GITHUB:tylabs (Version 1.02.001): Dovehawk. Hg. v. GitHub, Inc. Online verfügbar unter https://github.com/tylabs/dovehawk, zuletzt geprüft am 16.12.2020.

35 Government UK (Hg.) (2019): Detecting the Unknown: A Guide to Threat Hunting. Online verfügbar unter https://hodigital.blog.gov.uk/wp-content/uploads/sites/161/2020/03/Detecting-the-Unknown-A-Guide-to-Threat-Hunting-v2.0.pdf, zuletzt geprüft am 01.12.2020.

36 Holger Schulze (2018): Threat Intelligence Report. Hg. v. AT&T Cybersecurity. Online verfügbar unter https://cybersecurity.att.com/resource-center/analyst-reports/threat-intelligence-report, zuletzt geprüft am 09.12.2020.

37 Internet Engineering Task Force (2011): Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry. Hg. v. Internet Engineering Task Force community. Online verfügbar unter https://tools.ietf.org/rfc/rfc6335.txt, zuletzt geprüft am 28.12.2020.

38 Jessica DeCianno (2014): IOC Security: Indicators of Attack vs. Indicators of Compromise. Hg. v. CrowdStrike Holdings, Inc. Online verfügbar unter https://www.crowdstrike.com/blog/indicators-attack-vs-indicators-compromise/#:~:text=What%20is%20an%20Indicator%20of%20Attack%20(IOA)%3F,exploit%20used%20in%20an%20attack., zuletzt geprüft am 10.12.2020.

39 Jürgen Hill (2020): So geht Sicherheit im Internet of Things. Hg. v. IDG Business Media GmbH. Online verfügbar unter https://www.cio.de/a/so-geht-sicherheit-im-internet-of-things,3548403,2, zuletzt geprüft am 09.12.2020.

40 Jürgen Schreier (2020): IoT oder das Internet der unsicheren Dinge. Hg. v. Vogel Communications Group GmbH & Co. KG. Online verfügbar unter https://www.industry-of-things.de/iot-oder-das-internet-der-unsicheren-dinge-a-915722/, zuletzt geprüft am 11.12.2020.

41 Karen Scarfone (2016): The-Hunters-Handbook.pdf. Hg. v. CyberEdge Group, LLC. Online verfügbar unter https://cyber-edge.com/wp-content/uploads/2016/08/The-Hunters-Handbook.pdf, zuletzt geprüft am 03.12.2020.

42 Kevin Ashton (2009): That Internet of Things. Hg. v. RFID Journal LLC. Online verfügbar unter http://www.itrco.jp/libraries/RFIDjournal-That%20Internet%20of%20Things%20Thing.pdf, zuletzt geprüft am 11.12.2020.

43 Kevvie Fowler (2016): Data Breach Preparation and Response. Hg. v. Elsevier B.V. Online verfügbar unter https://www.sciencedirect.com/book/9780128034514/data-breach-preparation-and-response, zuletzt geprüft am 09.12.2020.

44 Klaus Nemelka (2020): Was ist eigentlich DNS-Tunneling? Hg. v. Computerweekly. Online verfügbar unter https://www.computerweekly.com/de/meinung/Was-ist-eigentlich-DNS-Tunneling, zuletzt aktualisiert am 2020, zuletzt geprüft am 08.01.2021.

45 Konstantin Matern (2019): IoT-Trends: Welche Möglichkeiten und Potenziale bietet 5G dem Internet der Dinge? Hg. v. DIGITALE WELT Academy. Online verfügbar unter https://digitaleweltmagazin.de/2019/07/30/iot-trends-welche-moeglichkeiten-und-potenziale-bietet-5g-dem-internet-der-dinge/, zuletzt geprüft am 09.12.2020.

46 Laimingas (2018): was-ist-threat-hunting. Hg. v. Vogel IT-Medien GmbH. Online verfügbar unter https://www.it-business.de/was-ist-threat-hunting-a-808131/, zuletzt geprüft am 02.12.2020.

47 Margaret Rouse (2020): DNS Amplification Attack (DNS-Verstärkerangriff). Hg. v. Computerweekly. Online verfügbar unter https://www.computerweekly.com/de/definition/DNS-Amplification-Attack-DNS-Verstaerkerangriff, zuletzt aktualisiert am 2020, zuletzt geprüft am 12.01.2021.

48 Michael Kroker (2019): internet-of-things-knapp-27-milliarden-vernetzte-geraete-oder-3-iot-gadgets-je-mensch/. Hg. v. HANDELSBLATT MEDIA GROUP GMBH & CO. KG. Online verfügbar unter https://blog.wiwo.de/look-at-it/2019/09/09/internet-of-things-knapp-27-milliarden-vernetzte-geraete-oder-3-iot-gadgets-je-mensch/, zuletzt geprüft am 09.12.2020.

49 Mitre Corporation (Hg.): MITRE ATT&CK. Online verfügbar unter https://attack.mitre.org/, zuletzt geprüft am 03.12.2020.

50 Opslyft (2020): 5 Best AWS Alternatives in 2020. Hg. v. Medium. Online verfügbar unter https://medium.com/guardians-of-cloud/5-best-aws-alternatives-in-2020-33e236707474, zuletzt geprüft am 26.02.2021.

51 Pagedesign GmbH (Hg.): Utrace. IP-Adressen und Domainnamen lokalisieren. Online verfügbar unter www.utrace.de, zuletzt geprüft am 12.01.2021.

52 Patrick-Benjamin Bök; Andreas Noack; Marcel Müller; Daniel Behnke (2020): Computernetze und Internet of Things: Springer Vieweg.

53 Peter H. Gregory (2017): Threat Hunting For Dummies: John Wiley & Sons Inc.

54 Rainer Singer (2018): Ungebetene Gäste gelangen per DNS-Anfragen ins Netzwerk. Hg. v. Computer Weekly. Online verfügbar unter https://www.computerweekly.com/de/meinung/Ungebetene-Gaeste-gelangen-per-DNS-Anfragen-ins-Netzwerk, zuletzt aktualisiert am 2018, zuletzt geprüft am 11.01.2021.

55 Richard K. Medlin (2020): A network defender's guide to threat detection: Independently published.

56 Shodan (Hg.): Shodan. Suchmaschine für das Internet der Dinge. Online verfügbar unter https://www.shodan.io, zuletzt geprüft am 12.01.2021.

57 SOCRadar Cyber Intelligence Inc. (Hg.) (2020): 5 Stages of The Threat Intelligence Lifecycle. Online verfügbar unter https://socradar.io/5-stages-of-the-threat-intelligence-lifecycle/, zuletzt geprüft am 08.12.2020.

58 Splunk, Inc. (Hg.) (Version: 8.1.0): SOFTWARE:Splunk. Online verfügbar unter https://www.splunk.com/de_de, zuletzt geprüft am 05.03.2021.

59 Stefan Luber; Peter Schmitz (2017): Was ist ein Security Operations Center (SOC)? Hg. v. Vogel IT-Medien GmbH. Online verfügbar unter https://www.security-insider.de/was-ist-ein-security-operations-center-soc-a-617980/, zuletzt geprüft am 01.03.2021.

60 Stefan Luber; Peter Schmitz (2018): Was ist ein CERT? Hg. v. Vogel IT-Medien GmbH. Online verfügbar unter https://www.security-insider.de/was-ist-ein-cert-a-702654/, zuletzt geprüft am 01.03.2021.

61 Stefan Luber; Peter Schmitz (2020): Was ist ein Blue Team? Hg. v. Vogel IT-Medien GmbH. Online verfügbar unter https://www.security-insider.de/was-ist-ein-blue-team-a-911741/, zuletzt geprüft am 22.12.2020.

62 Steven Feurer; Peter Schmitz (2019): Warum werden IoT-Geräte nicht einfach sicher? Hg. v. Vogel IT-Medien GmbH. Online verfügbar unter https://www.security-insider.de/warum-werden-iot-geraete-nicht-einfach-sicher-a-843501/, zuletzt geprüft am 09.12.2020.

63 Tanja Ulmen (2019): IoT-Sicherheit – (Un-)möglich? Hg. v. ComConsult GmbH. Online verfügbar unter https://www.comconsult.com/iot-sicherheit-moeglich/, zuletzt geprüft am 11.12.2020.

64 The Recorded Future Team (2020): What the 6 Phases of the Threat Intelligence Lifecycle Mean for Your Team. Hg. v. Recorded Future. Online verfügbar unter https://www.recordedfuture.com/threat-intelligence-lifecycle-phases/, zuletzt geprüft am 16.12.2020.

65 Unit42 (Hg.) (2020): 2020 Unit 42 IoT Threat Report. Online verfügbar unter https://unit42.paloaltonetworks.com/iot-threat-report-2020/, zuletzt geprüft am 09.12.2020.

66 Vern Paxson (Version 3.2.2): SOFTWARE:Zeek. Hg. v. Vern Paxson. Online verfügbar unter https://zeek.org/, zuletzt geprüft am 15.12.2020.

67 Volker Scholz (2019): Security Operations Center (SOC) ermöglicht ganzheitlichen Schutz. Hg. v. VINCI Energies Deutschland ICT GmbH. Online verfügbar unter https://www.axians.de/de/blog/2019/02/28/security-operations-center-soc-ermoeglicht-ganzheitlichen-schutz/, zuletzt geprüft am 01.03.2021.

68 Wolfgang Gotscharek (2016): Internet of Things IoT – Chancen, Risiken, Technik, Vorgehen – Mit Anwendungsbeispielen aus der Praxis. Hg. v. Gotscharek & Company GmbH. Online verfügbar unter https://www.gotscharek-company.com/blog/internet-of-things-iot-%E2%80%93-chancen,-risiken,-technik,-vorgehen-%E2%80%93-mit-anwendungsbeispielen-aus-der-praxis, zuletzt geprüft am 07.12.2020.

Anhang

Auf der beigefügten CD befinden sich folgende Dateien:

1. Die Bachelorarbeit als PDF-Dokument
2. Die Übersicht der genutzten Zeek-Skripte als PDF-Dokument

Selbstständigkeitserklärung

Versicherung über Selbstständigkeit

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

Hamburg, den _____ _____