



**TECHNISCHE  
UNIVERSITÄT  
DRESDEN**

---

Faculty of Computer Science Institute of Systems Architecture, Chair of Privacy and Data Security

---

Bachelor-Arbeit

# Thesis title

Student name

Geboren am: 25. Dezember 1999 in Dresden

Matrikelnummer: 4803900

Immatrikulationsjahr: 2018

zur Erlangung des akademischen Grades

**Bachelor of Science (B.Sc)**

Betreuer

**Supervisor**

Betreuender Hochschullehrer

**Prof. Dr. Professor**



## **AUFGABENSTELLUNG FÜR DIE BACHELORARBEIT**

Name, Vorname des Studenten: Köllner, Tobias  
Immatrikulationsnummer: 4803900  
Studiengang: Bachelor Informatik  
E-Mail: tobias.koellner@mailbox.tu-dresden.de

Thema: Kontextsensitivität für Netzwerk-Sicherheits-Monitoring

### *Zielstellung:*

Zur Erhöhung der Sicherheit von Netzwerken werden häufig Techniken wie Network security monitoring (NSM) sowie intrusion detection (IDS) verwendet. Insbesondere die Menge an falsch-positiven Meldungen stellt hierbei eine große Herausforderung dar. Um dieses Problem sowie die generelle Interpretierbarkeit der Beobachtungsdaten zu verbessern können zusätzliche Informationen gesammelt werden, die man als Netzwerkcontext bezeichnen kann.

Der erste Teil der Arbeit soll untersuchen, wie solche Kontextinformationen sowohl während des Betriebs von NSM-Systemen als auch während der Signatur/Modell-Erstellung im Rahmen von IDS gesammelt und verwendet werden können. Ziele hierbei sind, solche Möglichkeiten auf theoretischer Ebene wie auch von open-source Implementierungen zu sichten und hierbei einen kritischen Kontextbegriff im Vergleich zu etablierten Kontextbegriffen in der Literatur in Bezug auf Netzwerke zu entwickeln.

Aufbauend auf diesen Resultaten kann eine einfache praktische Auswertung verfügbarer Implementierungen mit explizitem Hinblick auf die kontextsensitiven Fähigkeiten durchgeführt werden. Hierfür ist ein Plan für eine minimale Testumgebung sowie das Design von geeignetem Netzwerkverkehr erforderlich.

### *In der Arbeit sollen schwerpunktmäßig folgende Teilaufgaben bearbeitet werden:*

- Überblick und Analyse existierender NSM-Ansätze in Bezug auf Kontext- und Umgebungsbewusstsein. Hierbei sollen mindestens folgende Forschungsfragen bearbeitet werden:
  - Lassen sich generelle Kategorien von Kontext im Rahmen von NSM formulieren?
  - Welche Kontextinformation kann während der Regelgenerierung im Unterschied zu generell verfügbaren Regeln aus öffentlichen Quellen berücksichtigt werden?
  - Welche Techniken wurden vorgeschlagen, um Kontextinformation während des regulären Betriebs zu sammeln und wie werden diese dann genutzt?
  - Inwiefern gibt es Möglichkeiten der Anpassungen an Änderungen der Netzwerkkumgebung, etwa der Topologie oder der genutzten Protokolle?
  - Wie stark trägt Kontextinformation zur Reduktion von falsch-Positiven bei?
- Es soll eine Evaluation der Fähigkeit von Kontextnutzung von verfügbaren open-source-Implementierungen (etwa Plugins in Zeek oder Netzdefinitionen in Suricata) durchgeführt werden.

Verantwortlicher Hochschullehrer:  
Institut:  
Beginn am: 3.6.2022

Dr. Stefan Köpsell  
Systemarchitektur  
Einzureichen am: 19.8. 2022

3.6.22, T Köllner  
.....  
Datum, Unterschrift der/des Studierenden

.....  
Unterschrift des betreuenden Hochschullehrers

### Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit mit dem Titel *Thesis title* selbstständig und ohne unzulässige Hilfe Dritter verfasst habe. Es wurden keine anderen als die in der Arbeit angegebenen Hilfsmittel und Quellen benutzt. Die wörtlichen und sinngemäß übernommenen Zitate habe ich als solche kenntlich gemacht. Während der Anfertigung dieser Arbeit wurde ich nur von folgenden Personen unterstützt:

Supervisor name

Weitere Personen waren an der geistigen Herstellung der vorliegenden Arbeit nicht beteiligt. Mir ist bekannt, dass die Nichteinhaltung dieser Erklärung zum nachträglichen Entzug des Hochschulabschlusses führen kann.

Student name

## **Zusammenfassung**

The abstract serves as a summary of the work. It should concisely (around half a page) answer the following questions:

- What is the problem tackled in this thesis?
- Why is this problem relevant?
- What contributions does this thesis contain with regards to the problem?
- Which scientific method(s) was/were used and what were their results?

# Inhaltsverzeichnis

Zusammenfassung	4
1 Einführung	7
2 Hintergrund	8
2.1 Definition eines Kontextbegriffs im Rahmen der Zugriffskontrolle . . . . .	8
2.1.1 Historische Entwicklung des Kontextbegriffs . . . . .	8
2.1.2 Zugriffskontrolle . . . . .	9
2.2 Kontextkategorisierung . . . . .	9
2.2.1 Analyse bereits vorgeschlagener Kategorien . . . . .	9
3 Voraussetzungen und verwandte Arbeiten	11
4 Design	12
4.1 Notwendigkeit einer Taxonomie . . . . .	12
4.2 Erstellung einer Kontexttaxonomie . . . . .	12
4.2.1 Konzeptionell . . . . .	12
4.2.2 Betrieblich . . . . .	13
4.2.3 Historie . . . . .	14
4.2.4 Logisch . . . . .	14
4.2.5 Verhaltensbezogen . . . . .	15
4.2.6 Netzwerk . . . . .	16
4.3 Form des Kontextes . . . . .	16
4.3.1 Historie . . . . .	17
4.4 Kontextgewinnung . . . . .	17
4.5 Umwandlung der Taxonomie in IDS-Signaturen . . . . .	17
4.5.1 Entscheidung für ein IDS . . . . .	18
4.5.2 Verbesserung der Schwächen . . . . .	18
4.6 Anspruch an das IDS . . . . .	19
4.6.1 Einordnung der Kriterien . . . . .	20
4.6.2 Auswahl der Kriterien . . . . .	20
5 Implementierung	21
5.1 Versuchsaufbau . . . . .	21
5.1.1 Netzwerk . . . . .	21
5.1.2 Netzwerkverkehr . . . . .	21
5.1.3 Zeek . . . . .	22

5.2	Skripte . . . . .	22
5.2.1	Geografische Koordinaten und Ortszeit . . . . .	22
5.2.2	Laufende Prozesse . . . . .	23
5.2.3	Größe des Netzwerks . . . . .	23
5.2.4	Rechte eines Nutzers . . . . .	23
5.3	Angriffe . . . . .	23
5.3.1	ARP-Spoofing . . . . .	23
5.3.2	NAT-Slipstreaming . . . . .	23
5.3.3	DoS/DDoS . . . . .	23
<b>6</b>	<b>Evaluation</b>	<b>24</b>
6.1	Vergleich der Kontextkategorien . . . . .	24
6.1.1	Verfügbarkeit der Informationen . . . . .	24
6.1.2	Qualität der Informationen . . . . .	26
6.2	Leistungsverbesserung . . . . .	28
<b>7</b>	<b>Konklusion/Schlussfolgerung</b>	<b>29</b>
7.1	Dateilose Prozesse . . . . .	29

# 1 Einführung

## 2 Hintergrund

### 2.1 Definition eines Kontextbegriffs im Rahmen der Zugriffskontrolle

Kontext ist ein Begriff unter dem sich die meisten Menschen zwar etwas vorstellen, aber nur schwer erläutern oder gar korrekt und vollständig definieren können [7]. Auch herrscht in verschiedenen Fachbereichen, jede davon mit anderen Sachverhalten und Problemen die die jeweiligen Autoren versuchend zu lösen, eine andere Auffassung darüber welche Ansprüche eine Definition erfüllen muss. Das erschwert eine allumfassende, konkrete Definition zusätzlich [4]. Ich will in diesem Kapitel also nicht versuchen Kontext abschließend zu definieren, da dies in Anbetracht der vielen verschiedenen Definitionen und dem fehlenden Konsens wie Kontext zu definieren ist [17, 3] schlicht nicht möglich ist bzw. den Rahmen dieser Arbeit übersteigen würde. Stattdessen werde ich versuchen mithilfe meiner Ansicht nach relevanter Definitionen für den Bereich der Informatik bzw. der Zugriffskontrolle, die historische Entwicklung des Kontextbegriffs darzustellen. Das soll dem Leser ermöglichen nachzuvollziehen auf welcher Grundlage ich mich für die von mir gewählte Definition entschieden habe.

#### 2.1.1 Historische Entwicklung des Kontextbegriffs

Eine der frühesten Definitionen von Kontext von Schilit et al. [15] bestimmt als 3 Hauptaspekte von Kontext an welchem Ort, mit welchen Personen und in der Nähe welcher Ressourcen man sich befindet.

Des weiteren beinhaltet laut [15] Kontext Attribute wie Beleuchtung, Lautstärke, den Grad der Netzwerkverbindung, Kommunikationskosten, Kommunikationsbandbreite und die soziale Situation.

Nach der Definition von Dey et al. [7] ist "Kontext jede Information die genutzt werden kann um die Situation einer Entität zu charakterisieren. Eine Entität ist eine Person, ein Objekt oder ein Ort mit Relevanz für die Interaktion zwischen Nutzer und Anwendung. Das schließt auch Nutzer und Anwendung selbst mit ein". Sie wird allgemein hin von den meisten anderen Autoren als Quasikonsens akzeptiert [17, 3, 2] oder als Ausgangspunkt für ihre eigene Definition genutzt [19, 10].

Kaltz et al. [18] versteht Kontext als ein Kontextrraum also eine Kombination aus Kontextparametern, Elementen einer domänenspezifischen Ontologie und Dienstleistungsbeschreibungen in Form von  $C = \{U; P; L; T; D; I; S\}$  definiert. Dabei ist  $U$  das Set aus Nutzern und den dazugehörigen Rollen,  $P$  die Prozesse und Aufgaben,  $L$  der Ort,  $T$  der Zeitfaktor,  $D$  beschreibt



das Gerät,  $I$  die verfügbaren Informationen und  $S$  die verfügbaren Dienstleistungen. Ein spezifischer Kontext ist somit ein Punkt in diesem Raum.

Bazire und Brézillon [4] haben 150 Kontextdefinitionen analysiert und sind dabei zu der Erkenntnis gekommen, dass Kontext wie eine Begrenzung fungiert, welche das Verhalten eines Systems, Nutzers oder Computers in einer bestimmten Tätigkeit beeinflussen. Allerdings herrscht ihrer Ansicht nach kein Konsens darüber, ob Kontext extern oder intern, ein Set aus Informationen oder Abläufen, statisch oder dynamisch ist.

Kayes et al. [10] definieren Kontextinformationen in Bezug auf Zugriffskontrollentscheidungen als relevante Informationen über den Zustand einer Entität (Nutzer, Ressource, Ressourcenbesitzer) und deren Umgebung oder die Beziehung zwischen Entitäten.

### **2.1.2 Zugriffskontrolle**

## **2.2 Kontextkategorisierung**

Nachdem ein Überblick über das Kontextverständnis in der gängigen Literatur gegeben und Kontext im Bezug auf Kontextkontrolle definiert wurde, folgt eine Kategorisierung von Kontext.

### **2.2.1 Analyse bereits vorgeschlagener Kategorien**

Auch bei der Kategorisierung von Kontext gibt es richtungsweisende Vorschläge. Abowd et al. [1] haben einen der führenden Mechanismen zur Definition von Kontexttypen vorgeschlagen. Sie identifizierten Ort, Zeit, Identität, und Aktivität als primäre Kontexttypen. Weiterhin wird sekundärer Kontext als Kontext definiert, der durch Nutzung von Primärkontext erschlossen werden kann.

Schilit et al. [15] kategorisieren Kontext basierend auf 3 Fragen, die genutzt werden können, um den Kontext zu bestimmen, in 3 Kategorien:

1. Wo man sich befindet: Enthält alle Informationen, die sich auf einen Ort beziehen, beispielsweise GPS-Koordinaten, Namen von Institutionen oder Gebäuden (ein Café, ein Krankenhaus, eine Universität), spezifische Namen (z.B.: Technische Universität Dresden) spezifischen Adressen (z.B.: APB Nöthnitzer Str. 46) oder Nutzerpräferenzen (z.B. das Lieblingsrestaurant eines Nutzers)
2. Mit wem man sich zusammen aufhält: Information über die Personen, die um einen herum anwesend sind
3. Welche Ressourcen sich in der Nähe befinden: Informationen darüber, welche Ressourcen (Maschinen, technische Geräte, Betriebsmittel) sich im direkten Umfeld eines Nutzers befinden

Henricksen et al. [9] ordnet Kontext basierend auf der betrieblichen Kategorisierungstechnik in 4 verschiedene Kategorien:

1. Messbar: Informationen, die aus direkt messbaren Werten bestehen. Diese ändern sich oft oder gar kontinuierlich.
2. Statisch: Informationen, die sich während der Lebenszeit eines Systems gleich bleiben.
3. Profiliert: Informationen, die sich selten ändern.
4. Abgeleitet: Informationen, die unter Verwendung anderer Daten gewonnen wurden.

Van Bunningen et al. [16] ordnen Kategorisierungsversuche in zwei übergeordnete Gruppen: Betrieblich und Konzeptionell.

1. Betriebliche Kategorisierung: Einordnung anhand dessen wie der Kontext akquiriert, modelliert und behandelt wird.
2. Konzeptionelle Kategorisierung: Einordnung anhand der Bedeutung des Kontextes und der konzeptionellen Beziehungen

Chong[5] schlägt Historie als Kontextkategorie vor. Dabei werden der zeitliche Verlauf, der Werte die eine bestimmte Messgröße in der Vergangenheit angenommen hat, als Kontext definiert. Das erlaubt die Festlegung eines Standardzustandes dieser Werte. Damit Unter Umständen eine Vorhersage darüber welche Werte die Messgrößen zukünftig annehmen werden

# 3 Voraussetzungen und verwandte Arbeiten

**Kontextsensitivität** Ein System ist kontextsensitiv wenn es Kontext verwendet um dem Nutzer mit für ihn relevanten Informationen und/oder Dienstleistungen zu versorgen. Die Relevanz hängt dabei von der Aufgabe des Nutzers ab [7].

**Situation** Eine Situation

**Umgebung** Umgebung ist ein Synonym für Kontext [1].

## Netzwerk-Sicherheits-Monitoring

**Intrusion Detection** Angriffserkennung ist "der Prozess der Überwachung von Ereignissen in einem Computersystem oder Netzwerk, und die Analyse dieser Ereignisse auf Anzeichen eines möglichen Zwischenfalls. Gemeint sind damit Verstöße oder unmittelbare Bedrohungen von Sicherheitsrichtlinien, Akzeptanzrichtlinien oder Standardsicherheitspraktiken"[14].

**Intrusion Detection System** Auf Grundlage der Definition von Intrusion Detection, ist ein IDS somit Software die den Prozess, einen Eingriff zu erkennen, automatisiert [14].

## 4 Design

Zuerst erläutere ich die meiner Meinung nach bestehende Notwendigkeit für eine Taxonomie. Dann definiere ich die Kontextkategorien meiner Taxonomie. Danach definiere ich in welcher Form die Kontextinformationen in den einzelnen Kategorien vorliegen müssen und wie man Informationen aus unterschiedlichen Quellen in einer für IDS-Signaturen verwendbaren Form gewinnt. Zusätzlich gehe ich noch darauf ein welchen Anforderungen ich an ein IDS stelle und welche Schwächen ich in dem von mir gewählten IDS mithilfe von Kontextsensitivität lösen will. Abschließend bewerte ich wie gut sich die benötigten Kontextinformationen gewinnen lassen.

### 4.1 Notwendigkeit einer Taxonomie

Die Evaluation verschiedener Kategorisierungsschemata zeigt das keine Kategorisierung allen Ansprüchen gerecht werden kann[13].

### 4.2 Erstellung einer Kontexttaxonomie

Im folgenden möchte ich meinen Versuch einer Kombination der bereits vorgeschlagenen Kategorisierungsschemata erläutern. Dazu lege ich zuerst die Definition der einzelnen Kategorien im Bezug auf kontextsensitive Zugriffskontrolle fest. Die Kategorien müssen dafür, abhängig davon mit welchem Fokus sie der jeweilige Autor konstruiert hat, mehr oder weniger stark angepasst werden.

#### 4.2.1 Konzeptionell

Einordnung anhand der Bedeutung des Kontextes und der begrifflichen Beziehungen.

##### **Primär**

Kontextinformationen die gesammelt werden können ohne bereits vorhandene Daten zu verwenden oder zu kombinieren [1].

## Sekundär

Kontext der durch das verarbeiten von primärem Kontext erschlossen werden kann. Dies kann durch die Kombination einzelner Datenpunkte einer oder mehrerer Kategorien oder durch Abfragen weiterer Informationen mithilfe der primären Informationen geschehen [1].

### 4.2.2 Betrieblich

Wie schon in der Analyse bereits vorgeschlagener Kategorien bedeutet die betriebliche Kategorisierung: "Einordnung anhand dessen wie der Kontext akquiriert, modelliert und behandelt wird".

**Zeit** Zu welcher Zeit ein Zugriffsanfrage erfolgt.

**Ort** Von welchem Ort eine Zugriffsanfrage stammt.

**Identität** Wer Zugriff auf eine Ressource erfragt.

**Aktivität** Was in einer Situation passiert bzw. welche Aktion eine Entität ausführt.

**Grund** Placeholder

Konzeptionell	
Betrieblich	Primär
	Sekundär
	Zeit
	Bestimmung der Uhrzeit mit NTP
	Vorhersage der Auslastung basierend auf aktuellem Netzwerkverkehr
Betrieblich	Ort
	Bestimmung der Herkunft eines Pakets durch die IP-Adresse
	Verwendung der Signalstärken von Hotspots oder Metadaten versendeter Dateien
	Identität
Betrieblich	Identifikation durch die MAC-Adresse einer Entität
	Beziehungen zwischen Entitäten
	Aktivität
Betrieblich	Rückschlüsse aufgrund der verwendeten Ports
	Erkennung der vom Nutzer ausgeführten Tätigkeit anhand der laufenden Prozesse
Betrieblich	Grund
	Mithilfe anderer Kategorien ableiten weshalb eine Aktion ausgeführt wurde

### 4.2.3 Historie

Die Historie einer Entität wird in dieser Taxonomie zweigeteilt. Sie besteht aus:

1. Einem aktiven Teil also ihrem Verhalten, beispielsweise früheren Verbindungen bzw. Verbindungsanfragen
2. Einem passiven Teil also dem Zustand der für sie charakteristischen Attribute, beispielsweise einem Nutzernamen oder die Versionsnummer eines bestimmten Programms

Die Existenz einer Historie trifft die implizite Annahme das eine Entität eindeutig und über einen längeren Zeitraum im Netzwerkverkehr identifizierbar ist.

#### Dynamik

Wie beschrieben können sich die Werte, aus denen sich die Historie einer Entität zusammensetzt, je nach dem welchem Teil sie zugeordnet werden, verschieden oft ändern. Statische Messgrößen ändern dabei ihre Werte nie oder nur sehr selten. Dynamische Messgrößen hingegen sehr oft. In diesem Fall wird eine Unterteilung jährlich, monatlich, wöchentlich, täglich, stündlich, minütlich und sekundlich vorgenommen.

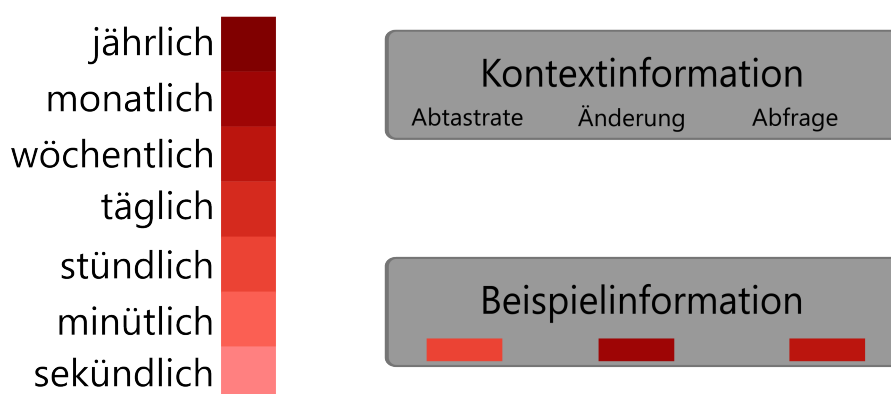
#### Raten

Die Historie ist weiterhin in 3 verschiedene Bereiche unterteilt. Abhängig davon wo die Änderung auftritt und ob das IDS oder eine Entität die Aktualisierung des Wertes auslöst.

**Änderungsrate** Gibt an wie oft eine Entität eine Werteänderungen mitteilt.

**Abtastrate** Wie oft Werte einer Entität vom IDS abgefragt bzw aktualisiert werden.

**Abfragerate** Wie oft Werte im Netzwerk von einer Entität, die nicht das IDS ist, abgefragt werden.



### 4.2.4 Logisch

#### Protokoll

Kategorisierung anhand des verwendeten Kommunikationsprotokolls. Orientiert sich am ISO/OSI-Referenzmodell [6]. Die Zuordnung zu einer bestimmten Schicht und damit Katego-

rie erfolgt anhand der für die einzelnen Schichten üblichen Protokolle. Ermöglicht die Identifikation von Entitäten anhand der von ihnen genutzten Protokolle. So kann beispielsweise ein Switch oder Router von einem Nutzer unterschieden werden.

**Anwendung** Beinhaltet allen Netzwerkverkehr der sich der Sitzungsschicht, Darstellungsschicht oder Anwendungsschicht zuordnen lässt

**Transport** Pakete die sich der Transportschicht zuordnen lassen.

**Vermittlung** Netzwerkverkehr der zur Vermittlungsschicht gehört.

### **Attribute der Entitäten**

Kategorisierung von Kontextinformationen abhängig davon welche Art von Entität sie betreffen. Diese Unterscheidung setzt genauso wie die Historie eindeutig identifizierbare Entitäten voraus.

**Gerät** Informationen die sich auf ein spezifisches Gerät beziehen

1. Ein Gerät ist beispielsweise einen Router oder das Endgerät eines Nutzers.
2. Informationen können beispielsweise die Liste an installierter Software, die Menge laufender Prozesse oder die Auslastung der Hardwarekomponenten sein.

**Nutzer** Informationen die sich auf einen Nutzer beziehen.

1. Ein Nutzer im Sinne eines Computersystems und keine physische Person. So kann eine physische Person durchaus mehrere verschiedene logische Nutzerkonten haben.
2. Informationen können zum Beispiel die Zugriffsrechte eines Nutzers sein.

**Anwendung** Informationen die sich auf eine Anwendung beziehen.

1. Anwendung umfasst jegliche Softwareprozesse die für sich oder in Kombination einen bestimmten Zweck erfüllen.
2. Eine Anwendung erzeugt für sie charakteristischen Netzwerkverkehr, versendet oder empfängt also bestimmte Informationen.

### **4.2.5 Verhaltensbezogen**

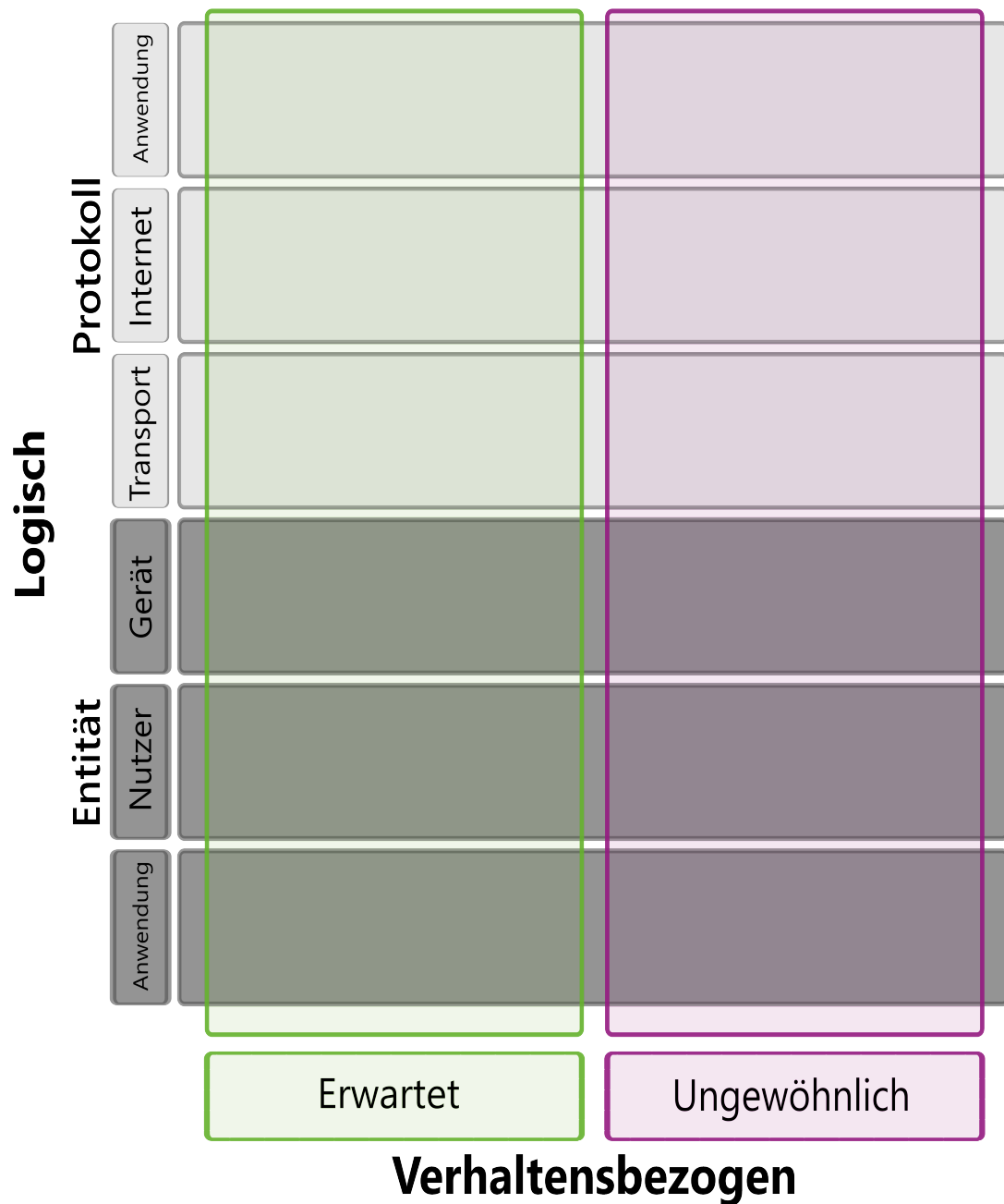
Einordnung der Kontextinformationen abhängig davon sie der für den Anwendungsfall definierten Normen entsprechen.

#### **Erwartetes Verhalten**

Werte der Kategorien oder Verhalten von Entitäten passend zu den im Netzwerk vorherrschenden Bedingungen.

#### **Ungewöhnliches Verhalten**

Werte der Kategorien oder Verhalten von Entitäten die in Kombination mit den im Netzwerk vorherrschenden Bedingungen entweder auffällig oder irrational sind.



#### 4.2.6 Netzwerk

Kontext der nicht Entitäten im Netzwerk betrifft sondern das Netzwerk selbst.

#### 4.3 Form des Kontextes

Nachdem die einzelnen Kategorien definiert wurden muss festgelegt werden in welcher Form die Kontextinformationen vorliegen sollen bzw. gebracht werden müssen. Dies ist hauptsächlich davon abhängig wie die Informationen in das IDS eingelesen werden.



### 4.3.1 Historie

Die Historie einer Entität muss genug Informationen enthalten um damit neue Entscheidungen in der Gegenwart oder Zukunft zu treffen und sie eindeutig zu identifizieren. Die Identifikation erfolgt anhand der Headerinformationen der einzelnen Kommunikationsschichten. In der Historie gespeichert werden in entweder die gesamte Payload eines Pakets oder zumindest ein ausreichend aussagekräftiges Subset. Die Regelmäßigkeit mit der die Historie aktualisiert bzw. erweitert wird ist vom jeweiligen Gegebenheiten und Ansprüchen abhängig.

Identifikationsmerkmale der Anfrage	Zugriffsentscheidung
MAC-Adresse	MAC-Adresse des Ziels
IP-Adresse	IP-Adresse des Ziels, gesetzte Flags, TTL, Protokoll
Port	Port des Ziels, protokollspezifische Informationen

## 4.4 Kontextgewinnung

. Dabei sollte man bedenken an welcher Stelle und wie oft man Informationen abrufen oder automatisch aktualisiert. [13]

Es folgt eine Übersicht einiger Beispiele für Möglichkeiten Kontextinformationen zu gewinnen. - osquery - network (open ports) | Nessus - devices in network | Configuration Management Database (CMDB), nmap - cve reference - yes/no - protocol header

## 4.5 Umwandlung der Taxonomie in IDS-Signaturen

Um die im Abschnitt Kontexttaxonomie festgelegten Kategorien in für ein IDS nutzbare Form zu bringen gilt es gewisse Dinge zu beachten:

Die Kontextsensitivität eines Computers unterscheidet sich drastisch von der eines Menschen. Rechensysteme sind sehr gut darin Daten zu erfassen und zu sammeln, aber Menschen sind immer-noch nötig um verschiedene Kontexte zu erkennen und zu entscheiden welches Handeln in einer bestimmten Situation angemessen ist [7].

Der limitierende Faktor des Potenzials eines kontextsensitiven Systems ist das Maß an Kontext das ein Entwickler vorhersehen und kodieren kann.

Es ist aber weder beim Design noch später bei der Implementierung unmöglich alle Zusammenhänge vorherzusehen.

Dementsprechend schwer wird es ein in sich geschlossenes und allumfassendes Regelset festzulegen [13].

Nach Greenberg et al. [8] gibt 3 non-triviale Hauptaspekte die man beim Entwerfen eines kontextsensitiven Systems beachten sollte:

1. Spezifizieren aller möglichen Kontextzustände
2. Wissen welche Informationen einen konkreten Kontextzustand akkurat festlegen.
3. Welche Aktion im jeweiligen Zustand ausgeführt werden sollen.

### 4.5.1 Entscheidung für ein IDS

IDSs können anhand der überwachten Plattform, der verwendeten Erkennungsmethode und der Struktur in der sie eingesetzt werden kategorisiert werden [12].

Eigenschaft	IDS Typ	Beschreibung
Plattform	Host	Überwacht Aktivitäten auf dem System auf dem es eingesetzt wird um lokale Angriffe zu erkennen.
	Netzwerk	Überwacht Aktivitäten im Netzwerk um Angriffe, die über eine Netzwerkverbindung ausgeführt werden zu erkennen.
	Hybrid	Kombiniert host- und netzwerk-basierte Intrusion Detection Systeme.
Angriffserkennung	Signatur	Überwacht System- und/oder Netzwerkaktivitäten anhand einer Reihe von Signaturen bekannter Angriffe auswertet. Daher ist es nicht in der Lage, Zero-Day-Angriffe zu erkennen, d. h. Angriffe, die Schwachstellen ausnutzen, die vor der Ausführung der Angriffe nicht öffentlich bekannt sind.
	Anomalie	Verwendet ein Basisprofil regulärer Netz- und/oder Systemaktivitäten als Referenz, um zwischen regulären und auffälligen Aktivitäten zu unterscheiden, wobei letztere als Angriffe behandelt werden. Wird typischerweise durch die Überwachung regulärer Aktivitäten trainiert, um ein Basisaktivitätsprofil zu erstellen.
	Hybrid	Verwendet sowohl signatur-basierte als auch anomalie-basierte Angriffserkennungsmethoden.
Struktur	Zentral	Kann nur an einem einzigen Standort eingesetzt werden.
	Verteilt	Besteht aus mehreren Teilsystemen für die an verschiedenen Standorten eingesetzt werden können und miteinander kommunizieren, um für die Erkennung von Angriffen relevante Daten, z. B. Angriffswarnungen, auszutauschen. Kann koordinierte Angriffe auf mehrere Standorte in einer bestimmten zeitlichen Abfolge erkennen.

### 4.5.2 Verbesserung der Schwächen

Der Hauptnachteil eines signatur-basierten IDS ausschließlich Angriffe zu erkennen die vordefinierten Verhaltensmustern entsprechen. Der Hauptnachteil eines anomalie-basierten IDS ist die Notwendigkeit die Baseline, mitunter sehr oft zu aktualisieren. Um diese Nachteile auszugleichen bietet es sich an die bereits vorhandenen, gesammelten und aufbereiteten Kontextinformationen zu verwenden. Dies ermöglicht die schon vorhanden Signaturen so zu verbessern das sie eine größere Menge von Ereignissen abdecken oder bereits definierte Ereignisse genauer zu spezifizieren und so weniger (falsche) Meldungen zu generieren ohne dabei das Verhalten des Netzwerkverkehrs dauerhaft neu bewerten zu müssen.

## 4.6 Anspruch an das IDS

Um die Performance des gewählten IDS verbessern zu können, muss man festlegen welche Kriterien die Leistung beeinflussen bzw. bestimmen. Mell et al.[11] nennen in ihrer Übersicht verschiedene Charakteristiken zur quantitativen Bestimmung der Erkennungsgenauigkeit eines IDS und erläutern zusätzlich die Wechselwirkungen zwischen einzelnen Kriterien, die beim Vergleich verschiedener IDS-Lösungen beachtet werden sollten.

### Abdeckung

Gibt an welche Typen von Angriffen ein IDS unter idealen Bedingungen erkennen kann.

### Wahrscheinlichkeit falscher Alarme

Gibt die Wahrscheinlichkeit das durch ein IDS ausgelöste Alarme durch gutartigen bzw. nicht-schädlichen Netzwerkverkehr verursacht wurden an.

$$\text{Rate an falsch – Positiven Meldungen} = \frac{\text{Anzahl falscher Alarme}}{\text{Anzahl aller Alarme}}$$

### Wahrscheinlichkeit einer Erkennung

Gibt die Rate der durch das IDS korrekt erkannten Angriffe an.

$$\text{Erkennungswahrscheinlichkeit} = \frac{\text{Anzahl korrekt erkannter Angriffe}}{\text{Anzahl aller Angriffe}}$$

### Resistenz

Ein signatur-basiertes IDS bzw. der menschliche Administrator hinter dem System weisen Probleme auf die nicht direkt beim Umgang mit verarbeitetem Netzwerkverkehr, sondern schon bei der bewussten, unbewussten, oder erzwungenen Entscheidung welcher Netzwerkverkehr überhaupt in Frage kommt, entstehen:

1. Ein zu große Menge an zu verarbeitendem Netzwerkverkehr die die Verarbeitungskapazität eines IDS übersteigt kann dazu führen das Netzwerkpakete verworfen und Angriffe nicht erkannt werden
2. Pakete die zwar nicht bösartig sind aber so konstruiert das sie möglichst viele IDS Signaturen auslösen, überfordern den Administrator oder stören eventuell sogar die Verarbeitung von Paketen generell.
3. Ein Angreifer könnte eine Vielzahl "harmloserer" aber trotzdem noch als schädlich zu deklarierende Pakete senden um einen größeren Angriff im Netzwerkverkehr zu verschleiern.
4. Pakete die möglicherweise vorhandene Fehler im IDS selbst ausnutzen.

### Korrelation zwischen Einzelereignissen

Demonstriert wie gut ein IDS eine Korrelation zwischen einzelnen Ereignissen, möglicherweise verschiedenen Ursprungs herzustellen. Die Ereignisse können dabei aus Routern, Firewalls, Anwendungen, dem IDS selbst oder einer großen Bandbreite anderer Quellen stammen.

## **Unbekannte Angriffe vorhersehen**

Gibt an wie gut ein IDS einen Angriff erkennt der so noch nicht aufgetreten ist. Signatur-basierte IDS sind allgemein, mit wenigen Ausnahmen, nicht in der Lage solch einen Angriff zu erkennen. Normalerweise erhöht die Fähigkeit eines Systems einen noch unbekannten Angriff zu erkennen, im Vergleich zu Systemen die dies nicht versuchen, zusätzlich die Rate an falsch-positiven Meldungen.

## **Identifizieren von Angriffen**

Wie gut ein IDS einem Angriff den es erkennt, einen Namen oder eine Kategorie, beispielsweise ein CVE-Nummer, zuordnen kann.

## **Beurteilung eines Angriffs**

Indikator dafür ob ein IDS den Erfolg und die Auswirkungen eines Angriffes korrekt beurteilen kann. In aktuellen Netzwerkkumgebungen schlagen viele Angriffs(-versuche) fehl. Die meisten IDS unterscheiden allerdings nicht zwischen erfolgreichen und fehlgeschlagenen Angriffen. Für den selben Angriff können manche IDS die Anzeichen dafür ob ein Angriff erfolgreich war erkennen, andere lediglich das ein Angriff stattgefunden hat, allerdings ohne feststellen zu können ob er erfolgreich war. Die Fähigkeit, den Grad zu dem ein Angriff auf das überwachte System erfolgreich war zu beurteilen ist essenziell. Eine Vorfilterung der Meldungen durch das IDS vereinfacht die Arbeit des Netzwerkadministrators bzw. Analysten stark, da so eine Analyse des Angriffsszenarios und der Korrelation einzelner Angriffe vereinfacht wird. Diese Fähigkeit bei einem gegebenen IDS messen zu können setzt das Wissen, darüber welche Angriffe erfolgreich sind und welche nicht, voraus.

### **4.6.1 Einordnung der Kriterien**

Die Wahrscheinlichkeit einen Angriff zu erkennen variiert mit der Rate an falsch-positiven Meldungen. Bei der Konfiguration eines IDS kann entweder die Falsch-positiv-Rate oder die Erkennungsrate optimiert werden.

### **4.6.2 Auswahl der Kriterien**

[12] In dieser Arbeit beschäftige ich mich ausschließlich mit sicherheitsrelevanten Leistungsmetriken. Dabei will ich hauptsächlich den Einfluss von Kontextsensitivität auf die Abdeckung, die Erkennungswahrscheinlichkeit, die Wahrscheinlichkeit falscher Alarme und die Interpretierbarkeit von Meldungen durch einen menschlichen Nutzer betrachten. Zusätzlich will ich beleuchten inwiefern Kontext die Korrelation zwischen Einzelereignissen und das Identifizieren von Angriffen ermöglicht bzw. verbessert.

# 5 Implementierung

## 5.1 Versuchsaufbau

Der Ablauf für alle Anwendungsfälle ist grundsätzlich sehr ähnlich:

1. Erzeugung des Netzwerkverkehrs
2. Mitschnitt des Netzwerkverkehrs
3. Analyse mittels Zeek-Skripts
4. Generierung von Logs

### 5.1.1 Netzwerk

Das Netzwerk besteht aus einem gleichbleibenden Endgerät in Form eines PC der als Ziel bzw. Endpunkt für den simulierten Netzwerkverkehr dient und verschiedenen anderen Teilnehmern bzw Angreifern. Die genauen Informationen der einzelnen Teilnehmer sind im jeweiligen Szenario genauer spezifiziert.

### 5.1.2 Netzwerkverkehr

#### Erzeugung

Der Netzwerkverkehr wurde mittels Scapy erzeugt. Das ermöglicht den für die verschiedenen Szenarien benötigten Netzwerkverkehr zu erzeugen und die einzelnen Schichten eines Packets zur Verdeutlichung des jeweiligen Anwendungsfalls anzupassen. Für einzelne Skripte ist ausschnittsweise der erzeugte Netzwerkverkehr oder das dazugehörige Skript eingebunden. Eine Übersicht über alle dafür verwendeten Skripte findet sich im Anhang.

#### Mitschneiden

Der Netzwerkverkehr wurde mittels WireShark mitgeschnitten.

### 5.1.3 Zeek

## 5.2 Skripte

### 5.2.1 Geografische Koordinaten und Ortszeit

Das Volumen von durch Menschen verursachten Netzwerkverkehrs ist meist tageszeitabhängig. So wird in der Regel nachts weniger kommuniziert als am Tag. Deshalb ist es sinnvoll den Grenzwert für erzeugten Verkehrsaufkommen ab dem ein Sender als potenziell böswillig eingestuft wird je nach Uhrzeit anzupassen.

---

```
def send_packet(ip_address):
    source_server = ip_address
    target_server = "172.17.144.87"

    # Send 10 packets
    for x in range(0, 15):
        port = random.randint(40000, 42000)
        layer_2 = Ether()
        layer_3 = IP(src=source_server, dst=target_server)
        layer_4 = TCP(sport=port, dport=port)
        tcp_pkt = layer_2 / layer_3 / layer_4
        sendp(tcp_pkt)
```

---

Abbildung 5.1: Erzeugen von Netzwerkverkehr

---

```
function geolocation(c: connection):double{
    local origin_country = lookup_location(c$Id$orig_h)$country_code;
    local origin_longitude = lookup_location(c$Id$orig_h)$longitude;
    return origin_longitude;
}

function time_at_geolocation(longitude: double): int{
    # 1 degree of longitude is 4 minutes of time
    local time_difference = (longitude - home_longitude)*240;
    local double_time_difference = time_to_double(network_time()) +
        time_difference;
    local epoch_time_difference = double_to_time(double_time_difference);
    local time_at_origin = strftime("%H", epoch_time_difference);
    local useable_time_at_origin = to_int(time_at_origin);
    return useable_time_at_origin;
}

function set_threshold(c_time: int): double{
    if(c_time < opening_time || c_time > closing_time )
        threshold = threshold*0.5;
    else
        threshold = threshold*1.1;
    return threshold;
}
```

---

Abbildung 5.2: Setzen des Grenzwertes von Verbindungsversuchen

5.2.2 Laufende Prozesse

5.2.3 Größe des Netzwerks

5.2.4 Rechte eines Nutzers

## 5.3 Angriffe

5.3.1 ARP-Spoofing

5.3.2 NAT-Slipstreaming

5.3.3 DoS/DDoS

5.3.4

# 6 Evaluation

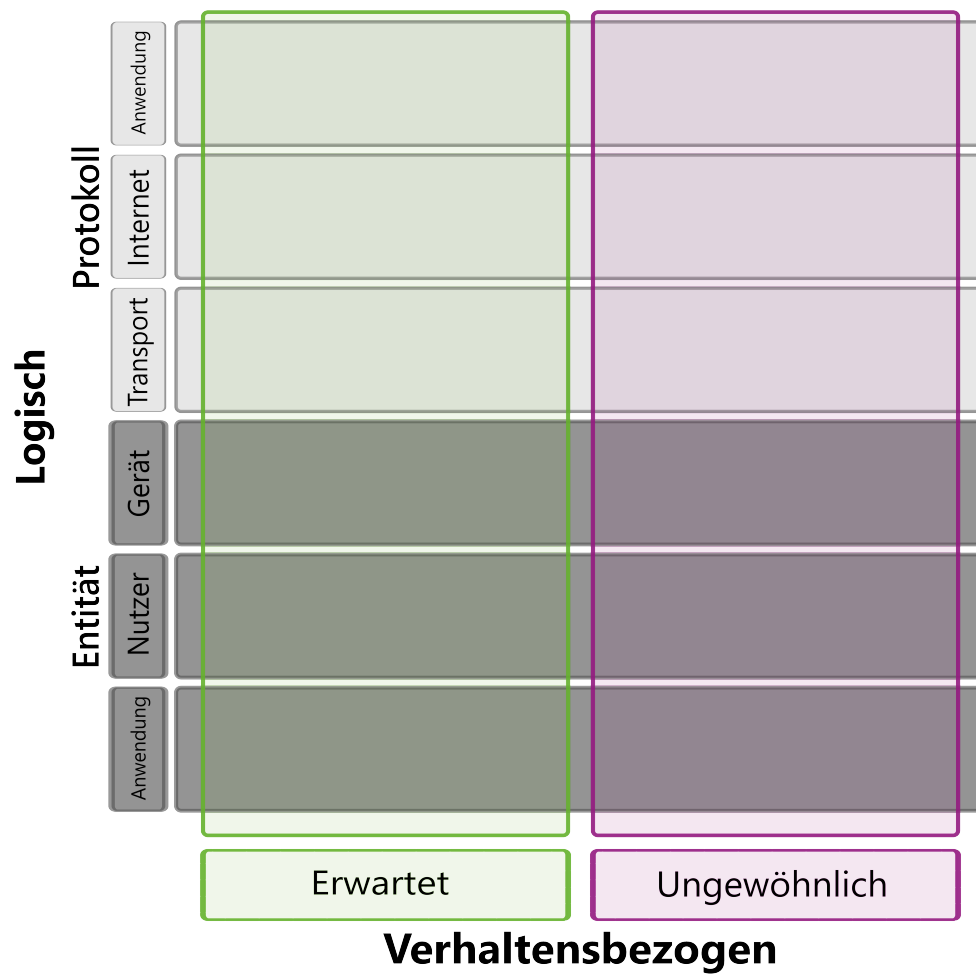
## 6.1 Vergleich der Kontextkategorien

### 6.1.1 Verfügbarkeit der Informationen

Die Informationen die ich in den definierten Kategorien als gegeben vorausgesetzt habe sind in der Praxis unterschiedlich schwer zu akquirieren. Offensichtlich ist es um ein Vielfaches einfacher korrekt die aktuelle Uhrzeit zu ermitteln als den Grund dafür das ein Nutzer eine Aktion ausführt zu bestimmen. Die nachfolgende Grafik dient dazu den meiner Einschätzung nach benötigten Aufwand zur Beschaffung einer Information zu veranschaulichen.

		Konzeptionell	
		Primär	Sekundär
Betrieblich	Zeit	Bestimmung der Uhrzeit mit NTP	Vorhersage der Auslastung basierend auf aktuellem Netzwerkverkehr
	Ort	Bestimmung der Herkunft eines Pakets durch die IP-Adresse	Verwendung der Signalstärken von Hotspots oder Metadaten versendeter Dateien
	Identität	Identifikation durch die MAC-Adresse einer Entität	Beziehungen zwischen Entitäten
	Aktivität	Rückschlüsse aufgrund der verwendeten Ports	Erkennung der vom Nutzer ausgeführten Tätigkeit anhand der laufenden Prozesse
	Grund		Mithilfe anderer Kategorien ableiten weshalb eine Aktion ausgeführt wurde

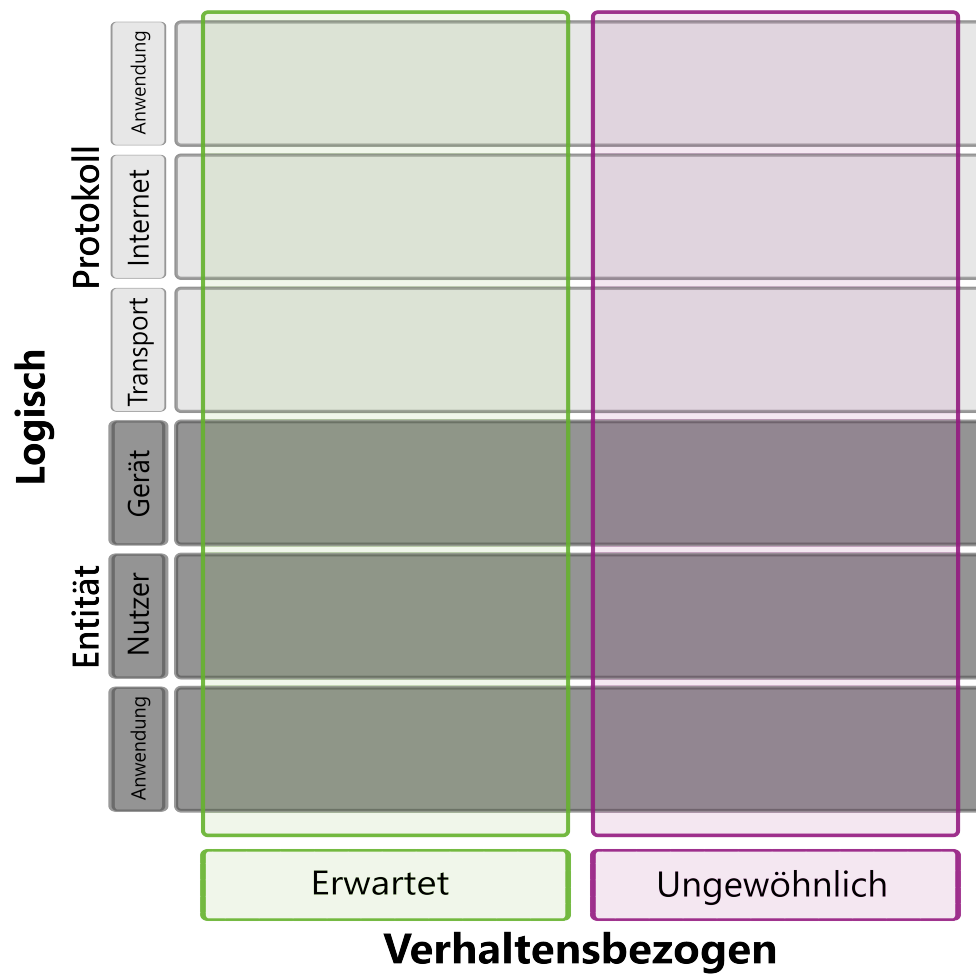




## 6.1.2 Qualität der Informationen

Auch die Qualität der Informationen schwankt in der Praxis. Genauigkeit und Korrektheit der Informationen sind teils unterschiedlich. Auch die Vertrauenswürdigkeit und Aktualität der Akquirierungspunkte ist verschieden.

		Konzeptionell	
		Primär	Sekundär
Betrieblich	Zeit	Bestimmung der Uhrzeit mit NTP	Vorhersage der Auslastung basierend auf aktuellem Netzwerkverkehr
	Ort	Bestimmung der Herkunft eines Pakets durch die IP-Adresse	Verwendung der Signalstärken von Hotspots oder Metadaten versendeter Dateien
	Identität	Identifikation durch die MAC-Adresse einer Entität	Beziehungen zwischen Entitäten
	Aktivität	Rückschlüsse aufgrund der verwendeten Ports	Erkennung der vom Nutzer ausgeführten Tätigkeit anhand der laufenden Prozesse
	Grund		Mithilfe anderer Kategorien ableiten weshalb eine Aktion ausgeführt wurde



## 6.2 Leistungsverbesserung

# 7 Konklusion/Schlussfolgerung

## 7.1 Dateilose Prozesse

Die vorgestellte Implementation bezieht im besten Fall lediglich Prozesse die in ihrem Lebenszyklus auf die Festplatte schreiben mit ein und versucht dies zu Netzwerkverkehr zuzuordnen. Prozesse die nur im Arbeitsspeicher existieren werden nicht betrachtet sind aber ein zunehmendes Problem.

# Literatur

- [1] Gregory D. Abowd u. a. "Towards a Better Understanding of Context and Context-Awareness". In: *Handheld and Ubiquitous Computing*. Hrsg. von Hans-W. Gellersen. Bearb. von Gerhard Goos, Juris Hartmanis und Jan van Leeuwen. Bd. 1707. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, S. 304–307. ISBN: 978-3-540-66550-2 978-3-540-48157-7. DOI: 10.1007/3-540-48157-5\_29.
- [2] Jose Aguilar, Marxjhony Jerez und Tania Rodríguez. "CAMEOnto: Context awareness meta ontology modeling". en. In: *Applied Computing and Informatics* 14.2 (Juli 2018), S. 202–213. ISSN: 22108327. DOI: 10.1016/j.aci.2017.08.001.
- [3] Unai Alegre, Juan Carlos Augusto und Tony Clark. "Engineering context-aware systems and applications: A survey". en. In: *Journal of Systems and Software* 117 (Juli 2016), S. 55–83. ISSN: 01641212. DOI: 10.1016/j.jss.2016.02.010.
- [4] Mary Bazire und Patrick Brézillon. *Understanding Context Before Using It*. en. Hrsg. von David Hutchison u. a. Bd. 3554. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, S. 29–40. ISBN: 978-3-540-26924-3 978-3-540-31890-3. DOI: 10.1007/11508373\_3.
- [5] Suan Khai Chong u. a. "Context-Aware Sensors and Data Muling". In: (2007), S. 15.
- [6] John D Day und Hubert Zimmermann. "The OSI reference model". In: *Proceedings of the IEEE* 71.12 (1983), S. 1334–1340.
- [7] Anind K Dey. "Understanding and Using Context". In: (2001), S. 4.
- [8] Saul Greenberg. "Context as a dynamic construct". In: *Human-Computer Interaction* 16.2-4 (2001), S. 257–268.
- [9] Karen Henriksen. "A framework for context-aware pervasive computing applications". In: (2003).
- [10] A S M Kayes, Jun Han und Alan Colman. "ICAF: A Context-Aware Framework for Access Control". en. In: (2012), S. 8.
- [11] Peter Mell u. a. "An overview of issues in testing intrusion detection systems". In: (2003).
- [12] Aleksandar Milenkoski u. a. "Evaluating Computer Intrusion Detection Systems: A Survey of Common Practices". en. In: *ACM Computing Surveys* 48.1 (Sep. 2015), S. 1–41. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/2808691. URL: <https://dl.acm.org/doi/10.1145/2808691> (besucht am 27.05.2022).

- [13] Charith Perera u. a. "Context Aware Computing for The Internet of Things: A Survey". In: *IEEE Communications Surveys & Tutorials* 16.1 (2014), S. 414–454. ISSN: 1553-877X. DOI: 10.1109/SURV.2013.042313.00197.
- [14] Karen Scarfone, Peter Mell u. a. "Guide to intrusion detection and prevention systems (idps)". In: *NIST special publication* 800.2007 (2007), S. 94.
- [15] Bill N Schilit, Norman Adams und Roy Want. "Context-Aware Computing Applications". In: (1994), S. 7.
- [16] Arthur H Van Bunningen, Ling Feng und Peter MG Apers. "Context for ubiquitous data management". In: *International Workshop on Ubiquitous Data Management*. IEEE. 2005, S. 17–24.
- [17] Wei Liu, Xue Li und Daoli Huang. "A survey on context awareness". In: *2011 International Conference on Computer Science and Service System (CSSS)*. 2011 International Conference on Computer Science and Service System (CSSS). Nanjing, China: IEEE, Juni 2011, S. 144–147. ISBN: 978-1-4244-9762-1. DOI: 10.1109/CSSS.2011.5972040.
- [18] J. Wolfgang Kaltz, Jürgen Ziegler und Steffen Lohmann. "Context-aware Web Engineering: Modeling and Applications". In: *Revue d'intelligence artificielle* 19.3 (1. Juni 2005), S. 439–458. ISSN: 0992499X. DOI: 10.3166/ria.19.439–458.
- [19] Andreas Zimmermann, Andreas Lorenz und Reinhard Oppermann. "An Operational Definition of Context". In: *Modeling and Using Context*. Hrsg. von Boicho Kokinov u. a. Bd. 4635. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, S. 558–571. ISBN: 978-3-540-74254-8. DOI: 10.1007/978-3-540-74255-5\_42.