TECHNISCHE UNIVERSITÄT DRESDEN

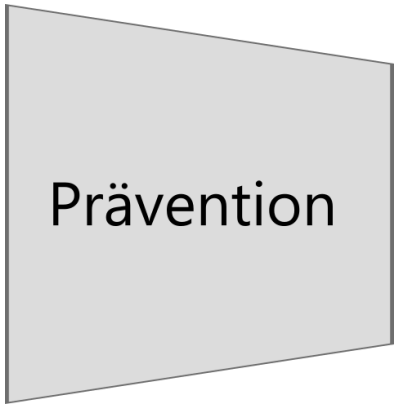
DRESDEN concept

Kontextsensitivität für Netzwerk-Sicherheits-Monitoring

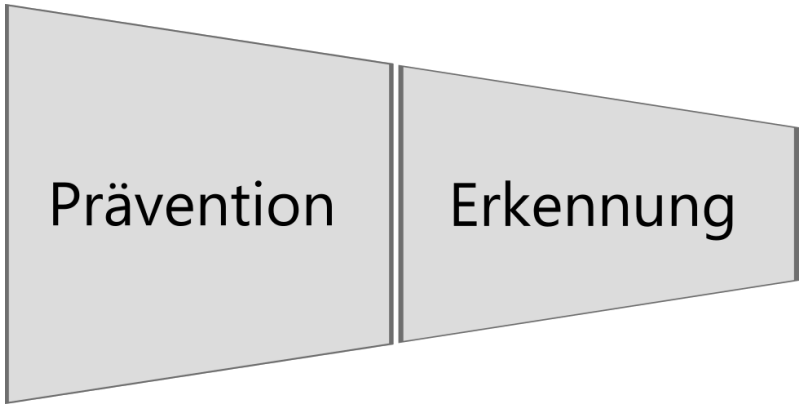
Tobias Köllner, Verteidigung Bachelorarbeit

15.11.2022

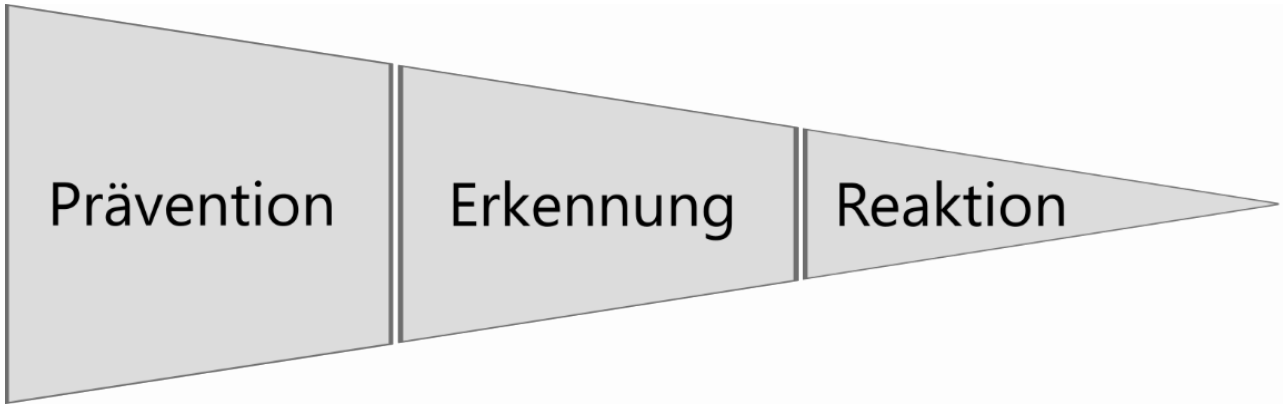
Netzwerksicherheit



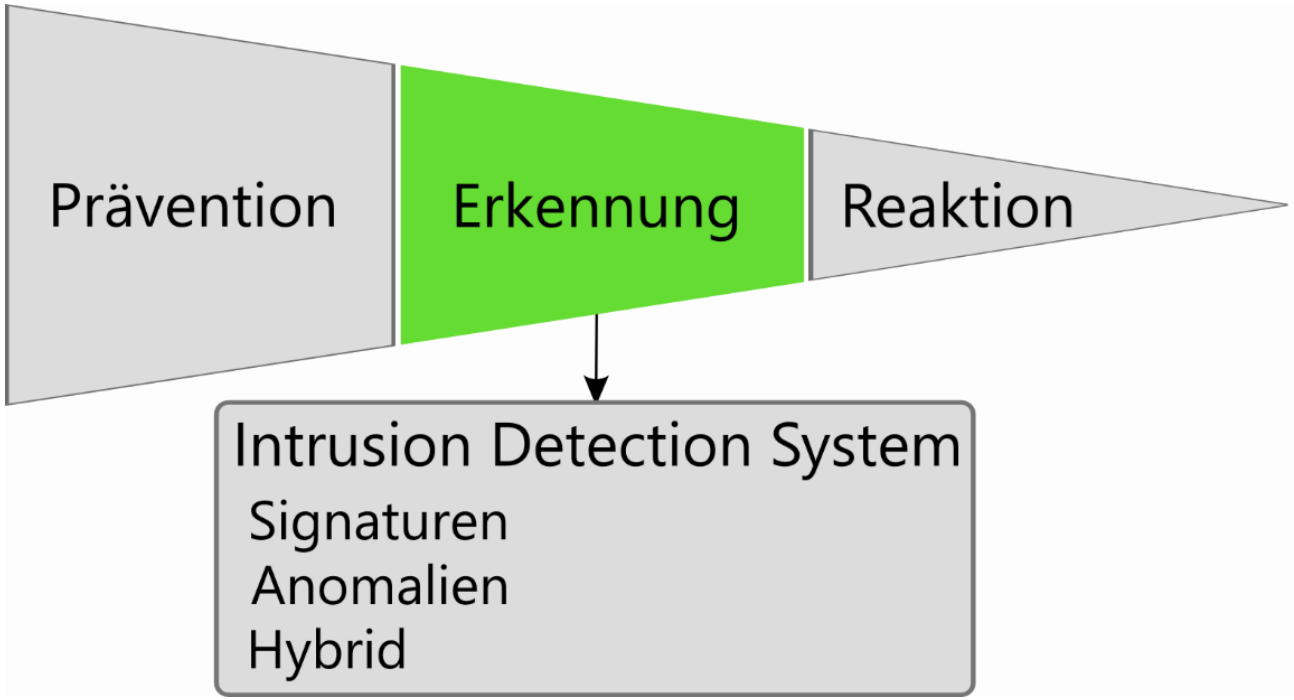
Netzwerksicherheit



Netzwerksicherheit



Netzwerksicherheit



Kontextdefinitionen

Kontextdefinitionen

Schilit et al.

Bazire et al.

Kaltz et al.

Dey

Aguilar et al.

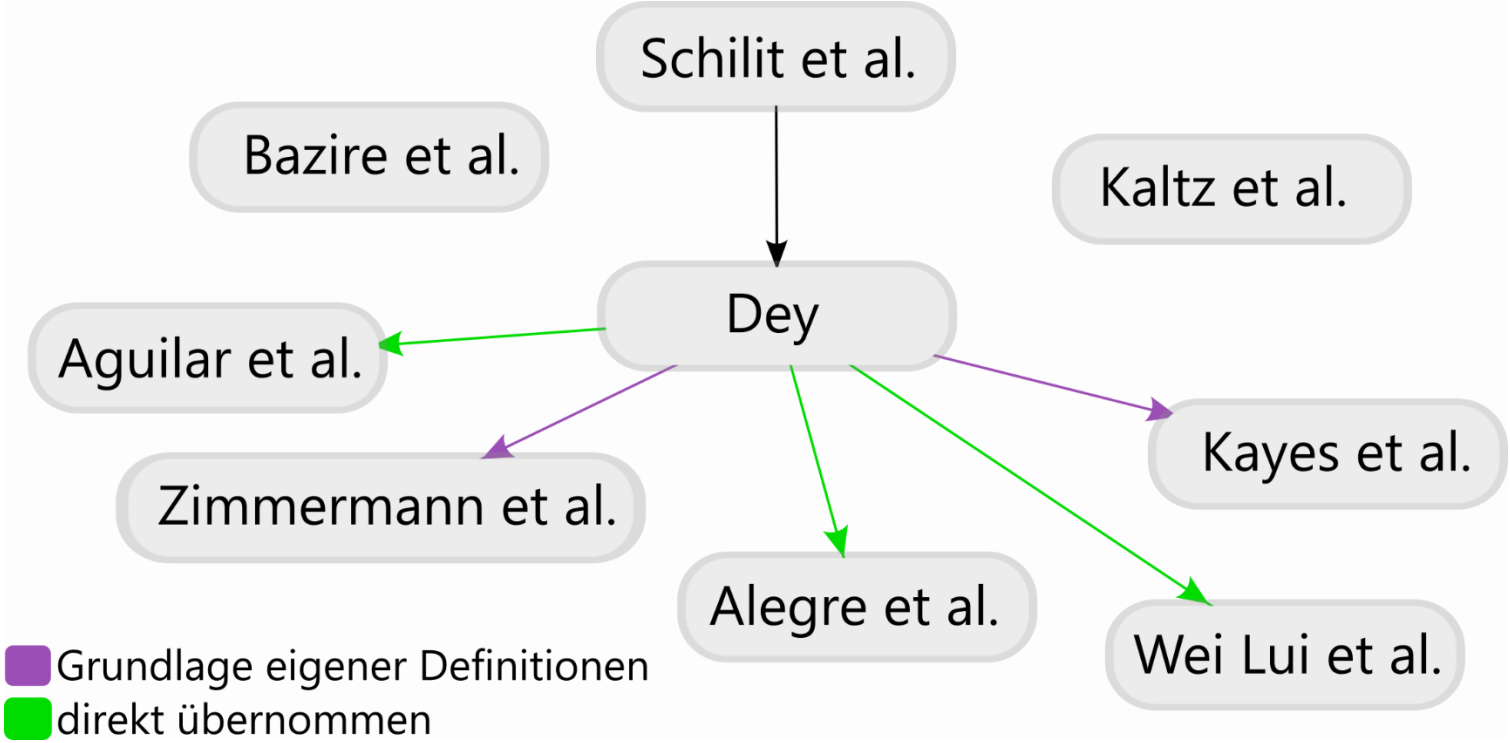
Kayes et al.

Zimmermann et al.

Alegre et al.

Wei Lui et al.

Kontextdefinitionen



Kategorisierung von Kontext

Kategorisierung von Kontext

Sozial | Computing | Netzwerk

Zeit | Ort | Identität | Aktivität | Grund

Identität | Ort | Ressourcen

Statisch | Dynamisch

Operational | Conceptual

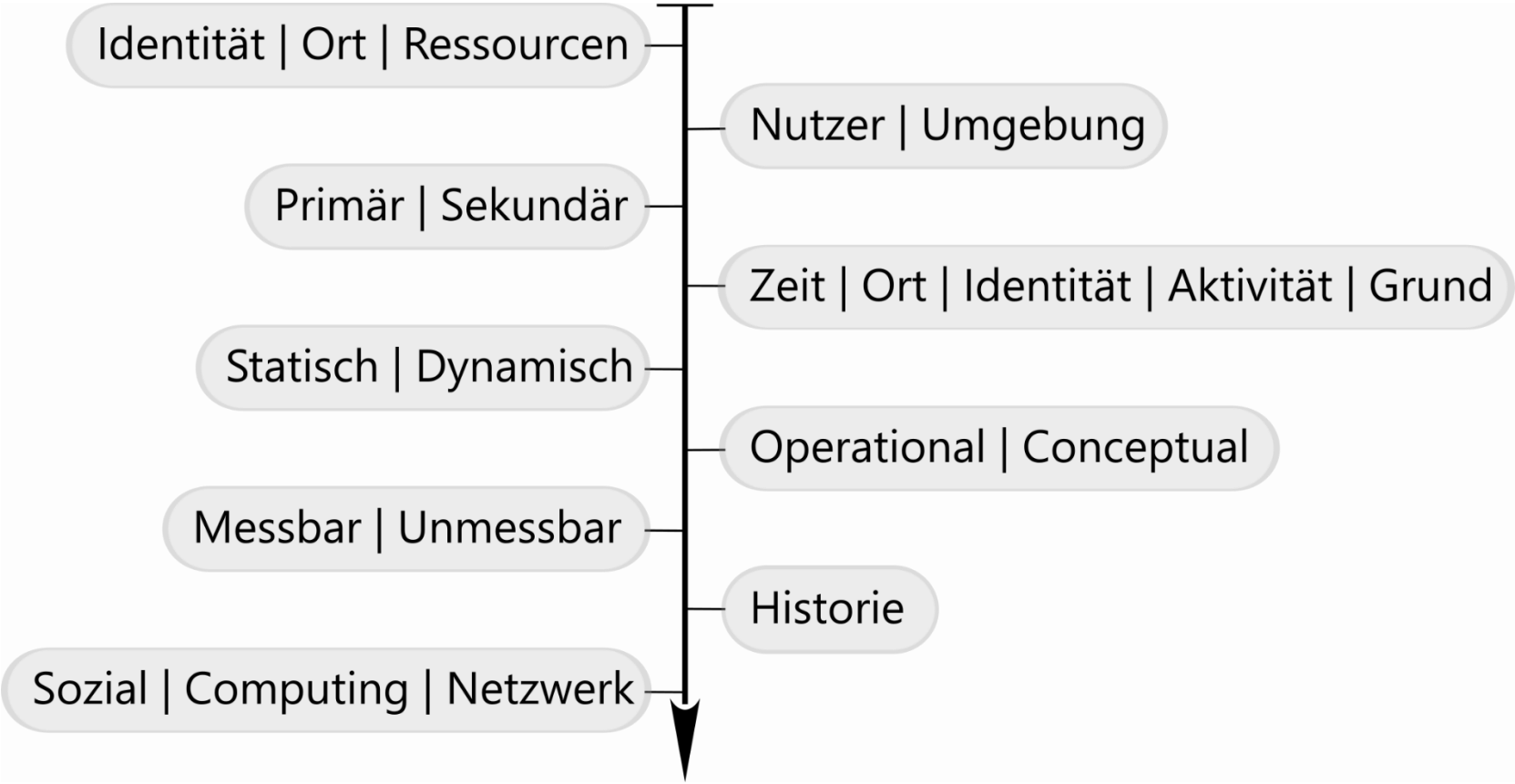
Historie

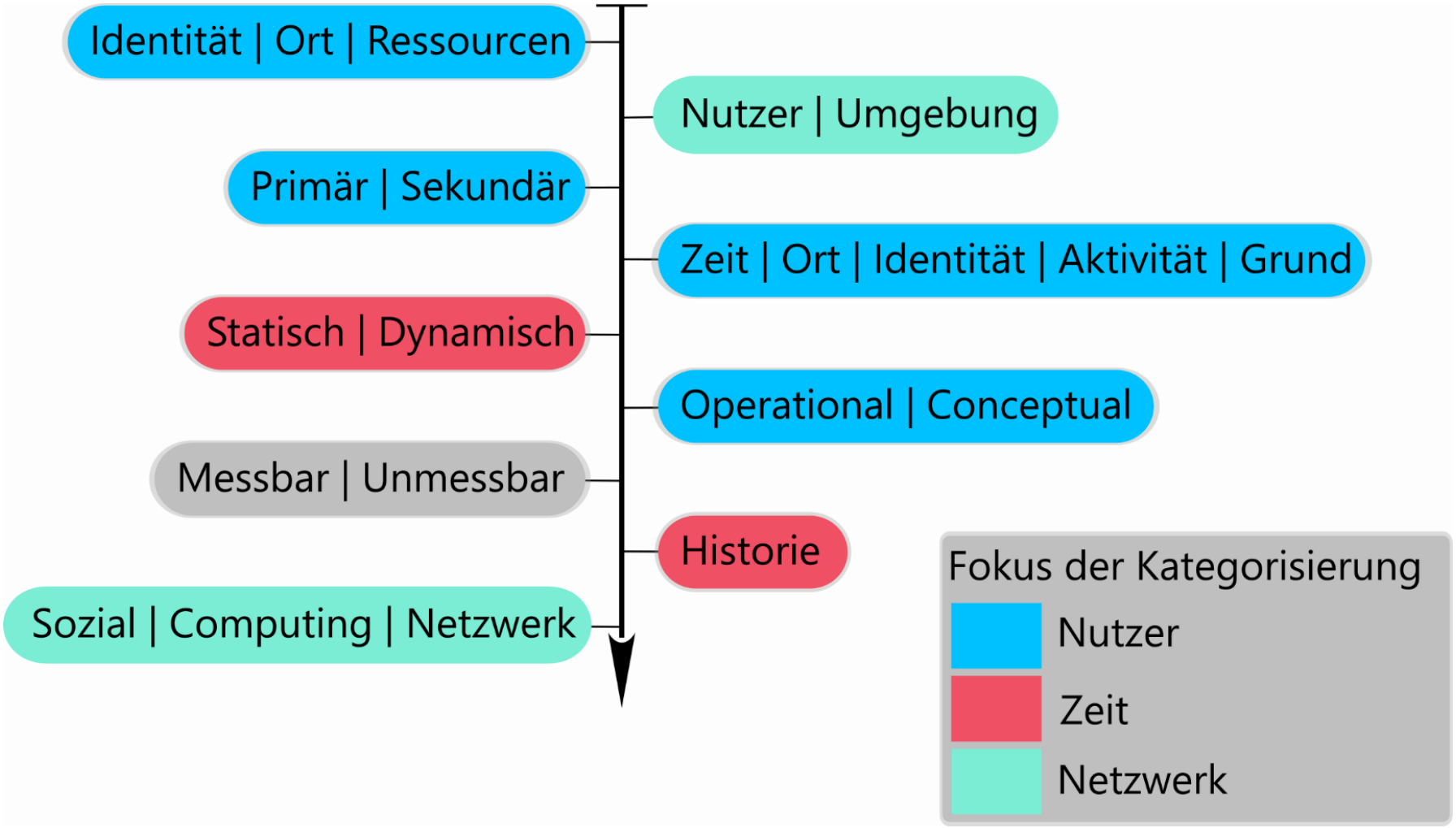
Nutzer | Umgebung

Primär | Sekundär

Messbar | Unmessbar

Kategorisierung von Kontext



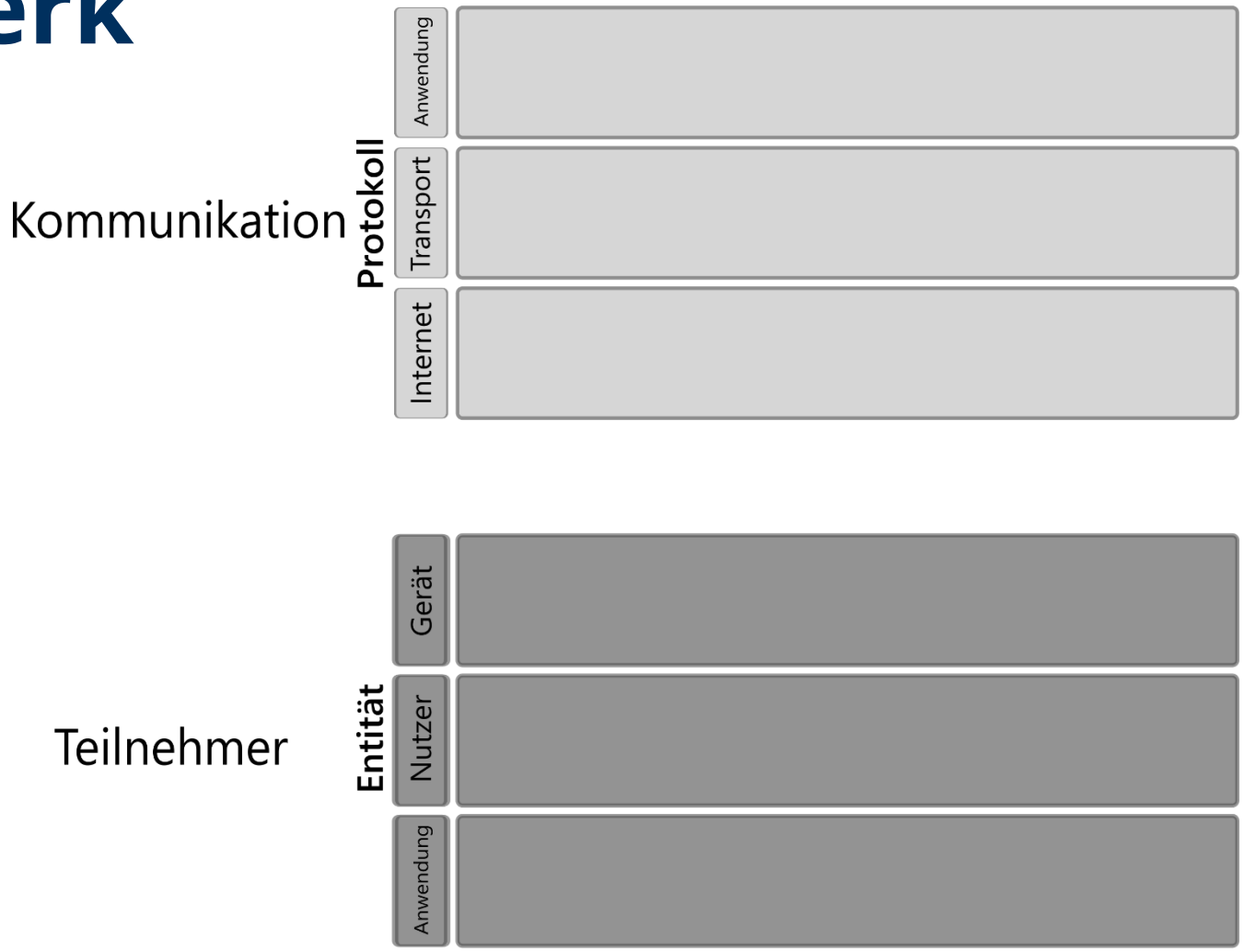

Identität | Ort | Ressourcen
Nutzer | Umgebung
Primär | Sekundär
Zeit | Ort | Identität | Aktivität | Grund
Statisch | Dynamisch
Operational | Conceptual
Messbar | Unmessbar
Historie
Sozial | Computing | Netzwerk
Fokus der Kategorisierung
Nutzer
Zeit
Netzwerk

Taxonomie

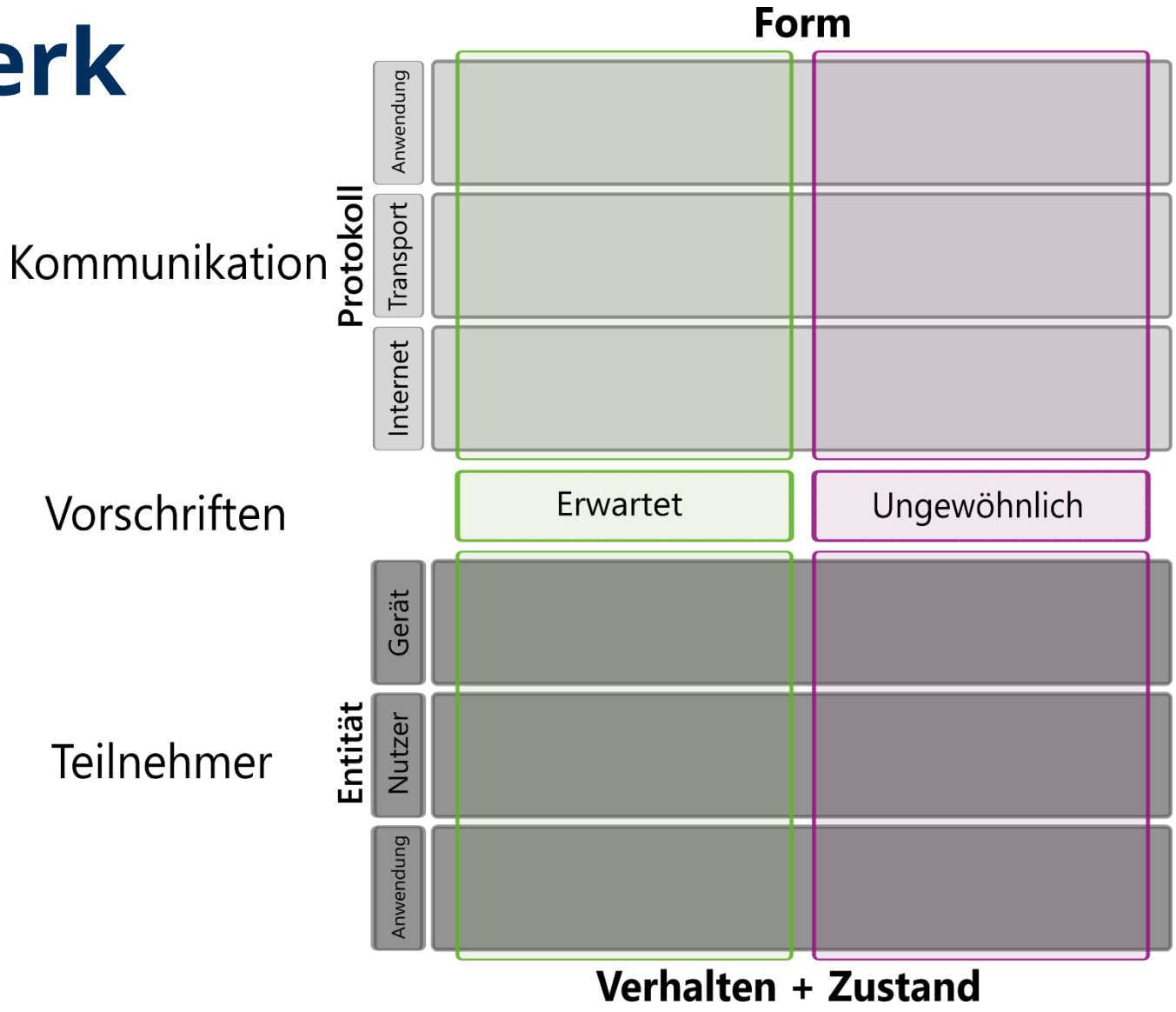
Netzwerk

Teilnehmer

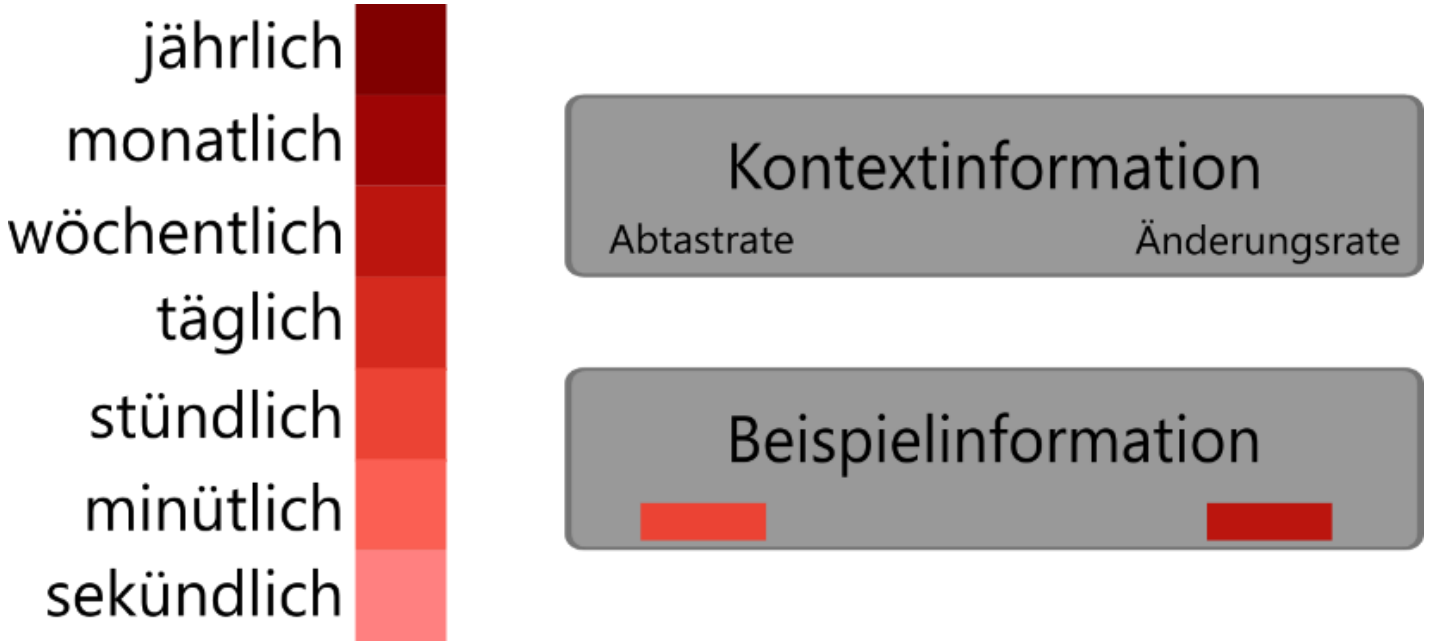
Netzwerk

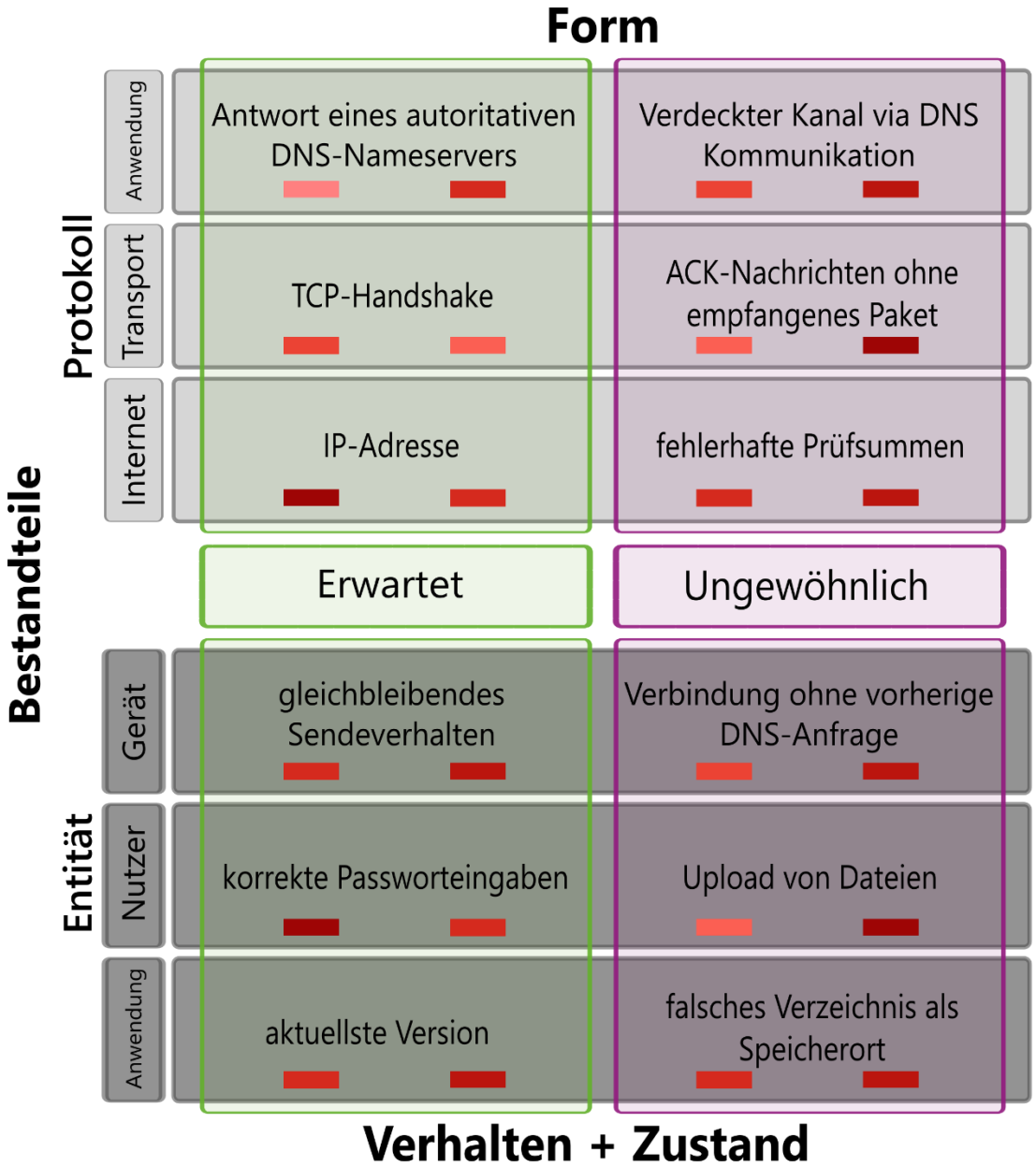

Kommunikation
Protokoll
Anwendung
Transport
Internet
Teilnehmer
Entität
Gerät
Nutzer
Anwendung

Netzwerk


Form
Protokoll
Anwendung
Transport
Internet
Kommunikation
Vorschriften
Erwartet
Ungewöhnlich
Entität
Gerät
Nutzer
Anwendung
Teilnehmer
Verhalten + Zustand

Zeit


jährlich
monatlich
wöchentlich
täglich
stündlich
minütlich
sekündlich
Kontextinformation
Abtastrate
Änderungsrate
Beispielinformation


Form
Bestandteile
Protokoll
Anwendung
Transport
Internet
Antwort eines autoritativen DNS-Nameservers
Verdeckter Kanal via DNS Kommunikation
TCP-Handshake
ACK-Nachrichten ohne empfangenes Paket
IP-Adresse
fehlerhafte Prüfsummen
Erwartet
Ungewöhnlich
Entität
Gerät
Nutzer
Anwendung
gleichbleibendes Sendeverhalten
Verbindung ohne vorherige DNS-Anfrage
korrekte Passworteingaben
Upload von Dateien
aktuellste Version
falsches Verzeichnis als Speicherort
Verhalten + Zustand

Nutzer

Betrieblich	Konzeptionell: Primär	Konzeptionell: Sekundär
Zeit	NTP-Zeitstempel	Vorhersage der Auslastung durch Verkehrsaufkommen
Ort	IP-Adresse	Signalstärken von Hotspots oder Metadaten
Identität	MAC-Adresse einer Entität	Beziehungen zwischen Entitäten
Aktivität	Port eines Prozesses	Identifikation einer Tätigkeit anhand laufender Prozesse
Grund		Ableiten weshalb eine Aktion ausgeführt wurde

Informationsquellen

- Konfigurationsmanagementdatenbank (KMDB)
- Scanner für Netzwerkschwachstellen
- Crawler
- Endpunktagent

Implementierung

1. Erzeugung
2. Mitschnitt
3. Analyse
   a) Einlesen von Netzwerkverkehr
   b) Einbindung des zusätzlichen Kontextes
   c) Logging

```python
def traffic(ip_address):
    source_server = ip_address
    target_server = "172.17.144.87"

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


if __name__ == '__main__':
    # different locations for testing
    cn_src = "61.135.0.1"
    us_src = "69.162.81.55"
    de_src = "102.128.165.43"
    ru_src = "103.136.43.65"
    src_list = [cn_src, us_src, de_src, ru_src]
    for i in src_list:
        traffic(i)
```

```python
def traffic(ip_address):
    source_server = ip_address
    target_server = "172.17.144.87"

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

2022-11-08 23:06:56.018707	192.168.178.73	35.163.41.215	TLSv1.2	844	Application Data
2022-11-08 23:06:56.110552	192.168.178.73	172.217.16.195	TCP	55	[TCP Keep-Alive] 52187 → 80 [ACK] Seq=872 Ack=1403 Win=261120 Len=1
2022-11-08 23:06:56.124303	172.217.16.195	192.168.178.73	TCP	66	[TCP Keep-Alive ACK] 80 → 52187 [ACK] Seq=1403 Ack=873 Win=67840 Len=0 SLE=872 SRE=873
2022-11-08 23:06:56.207456	35.163.41.215	192.168.178.73	TCP	60	443 → 58827 [ACK] Seq=3771 Ack=75948 Win=1966 Len=0
2022-11-08 23:06:56.207674	35.163.41.215	192.168.178.73	TLSv1.2	96	Application Data
2022-11-08 23:06:56.208017	35.163.41.215	192.168.178.73	TLSv1.2	191	Application Data
2022-11-08 23:06:56.208025	192.168.178.73	35.163.41.215	TCP	54	58827 → 443 [ACK] Seq=76738 Ack=3950 Win=1021 Len=0
2022-11-08 23:06:56.310007	85.124.84.253	192.168.178.73	TLSv1.3	93	Application Data
2022-11-08 23:06:56.310133	192.168.178.73	85.124.84.253	TLSv1.3	93	Application Data
2022-11-08 23:06:56.310202	192.168.178.73	85.124.84.253	TLSv1.3	78	Application Data
2022-11-08 23:06:56.310210	192.168.178.73	85.124.84.253	TCP	54	52240 → 443 [FIN, ACK] Seq=5440 Ack=226528 Win=262656 Len=0
2022-11-08 23:06:56.310581	85.124.84.253	192.168.178.73	TLSv1.3	78	Application Data
2022-11-08 23:06:56.310593	192.168.178.73	85.124.84.253	TCP	54	52240 → 443 [RST, ACK] Seq=5441 Ack=226552 Win=0 Len=0
2022-11-08 23:06:56.310691	85.124.84.253	192.168.178.73	TCP	60	443 → 52240 [FIN, ACK] Seq=226552 Ack=5377 Win=64512 Len=0
2022-11-08 23:06:56.335290	85.124.84.253	192.168.178.73	TCP	60	443 → 52240 [RST, ACK] Seq=226553 Ack=5416 Win=64512 Len=0
2022-11-08 23:06:56.335657	85.124.84.253	192.168.178.73	TCP	60	443 → 52240 [RST] Seq=226528 Win=0 Len=0
2022-11-08 23:06:56.335855	85.124.84.253	192.168.178.73	TCP	60	443 → 52240 [RST] Seq=226528 Win=0 Len=0
2022-11-08 23:06:56.453209	2606:50c0:8001::154	2003:c2:7f20:f900:1…	TCP	149	[TCP Retransmission] 443 → 60292 [FIN, PSH, ACK] Seq=1 Ack=1 Win=272 Len=75
2022-11-08 23:06:56.474929	192.168.178.73	172.217.16.195	TCP	55	[TCP Keep-Alive] 52120 → 80 [ACK] Seq=1804 Ack=2857 Win=262400 Len=1
2022-11-08 23:06:56.489559	172.217.16.195	192.168.178.73	TCP	66	[TCP Keep-Alive ACK] 80 → 52120 [ACK] Seq=2857 Ack=1805 Win=69888 Len=0 SLE=1804 SRE=1805
2022-11-08 23:06:56.997398	192.168.178.73	172.217.16.195	TCP	55	[TCP Keep-Alive] 52185 → 80 [ACK] Seq=3055 Ack=4911 Win=261888 Len=1
2022-11-08 23:06:57.012368	172.217.16.195	192.168.178.73	TCP	66	[TCP Keep-Alive ACK] 80 → 52185 [ACK] Seq=4911 Ack=3056 Win=73216 Len=0 SLE=3055 SRE=3056
2022-11-08 23:06:57.091490	192.168.178.73	93.184.220.29	TCP	55	[TCP Keep-Alive] 52131 → 80 [ACK] Seq=1302 Ack=2047 Win=262656 Len=1
2022-11-08 23:06:57.105596	93.184.220.29	192.168.178.73	TCP	66	[TCP Keep-Alive ACK] 80 → 52131 [ACK] Seq=2047 Ack=1303 Win=69120 Len=0 SLE=1302 SRE=1303
2022-11-08 23:06:57.524994	2c:91:ab:ac:d6:cc	00:b0:52:00:00:01	HomePl…	60	Qualcomm Atheros, OP_ATTR.REQ (Get Device Attributes Request)
2022-11-08 23:06:57.525546	2c:91:ab:ac:d6:cc	ff:ff:ff:ff:ff:ff	HomePl…	60	Qualcomm Atheros, GET_SW.REQ (Get Device/SW Version Request)
2022-11-08 23:06:57.526043	2c:91:ab:ac:d6:cc	ff:ff:ff:ff:ff:ff	0x8912	60	Ethernet II
2022-11-08 23:06:57.615133	192.168.178.73	239.255.255.250	UDP	698	49471 → 3702 Len=656
2022-11-08 23:06:57.646125	fe80::7481:393e:f399:abf6	ff02::c	UDP	718	49472 → 3702 Len=656
2022-11-08 23:06:57.661088	192.168.178.73	172.217.16.195	TCP	55	[TCP Keep-Alive] 52186 → 80 [ACK] Seq=2620 Ack=4211 Win=262400 Len=1

```python
def traffic(ip_address):
    source_server = ip_address
    target_server = "172.17.144.87"

    # Send 10 attack packets
```

```
2022-11-08 23:06:56.018707 192.168.178.73    35.163.41.215    TLSv1.2   844 Application Data
2022-11-08 23:06:56.110552 192.168.178.73    172.217.16.195   TCP        55 [TCP Keep-Alive] 52187 → 80 [ACK] Seq=872 Ack=1403 Win=261120 Len=1
```

```
#Gets invoked when the result of a query "arrives". Adds result to corresponding table
event query_result(ctx: ZeekAgent::Context, data: Columns){
    local ip_address = to_addr(data$line_content[0]);
    local host_name = data$line_content[1];
    resolved_addresses[ip_address] = host_name + " from local hosts";
}

# Query the local host file via Zeek Agent
# 172.17.144.22 got added to hosts file manually for demonstration
function query_hosts_file(){
    local str_stmt_hosts = "SELECT columns FROM files_columns(\"/etc/hosts\",\"$1:text,$2:text\")";
    local query_event = query_result;
    local _schedule =  30 secs;
    local test_query_join = ZeekAgent::query([$sql_stmt=str_stmt_hosts,$event_=query_event,$schedule_=_schedule]);
}

#check if destination ip-address got:
    # resolved
    # is a DNS-Server
    # is neither resolved before connection nor a DNS-Server
# and write to log
event check_resolve_table(c : connection){
    local destination_ip = c$id$resp_h;
    if(destination_ip !in resolved_addresses && destination_ip !in dns_server){
        local rec: DNS_Check::Info = [$ts=current_time(), $id=c$id, $notice="Connection without Resolve!"];
        # Store a copy of the data in the connection record so other
        # event handlers can access it.
        c$dns_check = rec;
        Log::write(DNS_Check::LOG, rec);
    }
}

# Generated when a connection's internal state is about to be removed from memory.
# Zeek generates this event reliably once for every connection when it is about to delete the internal state.
event connection_state_remove(c: connection){
    query_hosts_file();
    local destination_ip = c$id$resp_h;
    local conn_service = c$service;
    # Add DNS-Server to set
    if("DNS" in conn_service){
        add dns_server[destination_ip];
    }
```

Beachtenswertes für die Praxis

Technisch

- Abfragen von Informationen auf (leistungsschwachen) Endgeräten
- Zusätzlicher Netzwerkverkehr

Konzeptionell

- Verlässlichkeit der verwendeten Informationen
- Verwaltung der akquirierten Informationen und daraus gezogener Schlüsse

Limitierungen

- Hohe Komplexität durch “Vermenschlichen” eines automatisierten Prozesses

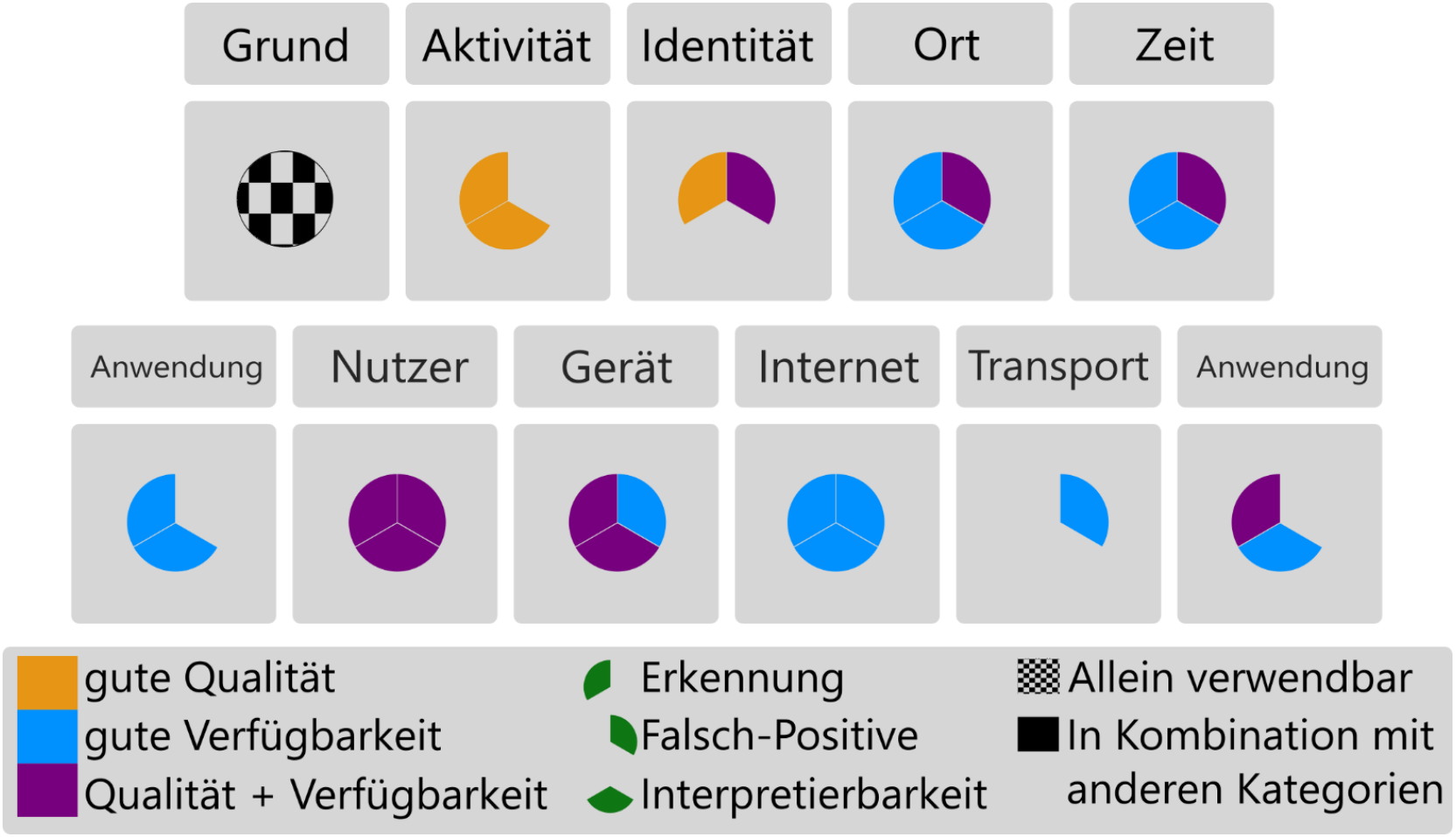
→ Erfordert umfassende Kenntnisse über alle möglichen Angriffsszenarien

→ höhere Einstiegshürden

→ System nur so gut wie der Skripteschreiber

- Erkennung aller Angriffe quasi unmöglich (bspw. Dateilose Schadsoftware)

Evaluation


Grund
Aktivität
Identität
Ort
Zeit
Anwendung
Nutzer
Gerät
Internet
Transport
Anwendung
gute Qualität
gute Verfügbarkeit
Qualität + Verfügbarkeit
Erkennung
Falsch-Positive
Interpretierbarkeit
Allein verwendbar
In Kombination mit anderen Kategorien

Konklusion

- Informationen netzwerkzentrierter Kategorien einfacher sammeln
- Nutzerzentrierte Kategorien einfacher in Skripte einzubinden
- Netzwerkinformationen können Nutzerinformationen ersetzen

→ allerdings zusätzliche Abstraktionsebene

- Kategorien entfalten bis auf wenige Ausnahmen erst in Kombination mit anderen ihr volles Potenzial
- Je nützlicher die Informationen für gewähltes Ziel desto schwerer zu akquirieren