Zentrum für sichere Informationstechnologie – Austria
Secure Information Technology Center – Austria

A-SIT

Seidlgasse 22 / 9, 1030 Wien
Tel.: (+43 1) 503 19 63–0
Fax: (+43 1) 503 19 63–66

Inffeldgasse 16a, 8010 Graz
Tel.: (+43 316) 873-5514
Fax: (+43 316) 873-5520

http://www.a-sit.at
E-Mail: office@a-sit.at

DVR: 1035461
ZVR: 948166612
UID: ATU60778947

ANLEITUNG FÜR INTEGRIERTES THREAT MODELING

Version 1.0 vom 23.05.2019
Peter Aufner – peter.aufner@iaik.tugraz.at

Abstract/Zusammenfassung: Dieses Dokument bietet einen Ansatz für integriertes Threat Modeling von allen Komponenten eines IoT Geräts. Von der Software Engineering Seite kommend bildet STRIDE hierfür die Basis zur Kategorisierung von Bedrohungen. Für die der Anwendung zugrunde liegenden Schichten des IoT Geräts werden STRIDE Kategorien nach Common Attack Pattern Enumeration and Classification (CAPEC) entwickelt bzw. Ansätze geboten um ein realistisches Bild über die Bedrohungslandschaft zu gewinnen.

Inhaltsverzeichnis

1. Bestandteile eines IoT Geräts

Die einfachste Definition von einem IoT Gerät stellt dieses als aus 3 Schichten bestehend dar:

1. Sensoren und Aktuatoren
2. Netzwerkkommunikation
3. Anwendung

Diese Darstellung ist nützlich um ein Bild für die Softwareentwicklung zu bekommen, jedoch ist sie für ein detailliertes Threat Model zu ungenau und befasst sich außerdem eher mit der Softwaresicht. Trotzdem können diese drei Ebenen genutzt werden, um näher an die physische Hardware zu gelangen. Wir werden sie daher in den folgenden Unterkapiteln erweitern.

1.1. Sensoren und Aktuatoren

Diese Schicht bezeichnet alle physischen Komponenten des Geräts, welche für die Interaktion mit der Umwelt notwendig sind. Um sie für das Threat Model zugänglicher zu machen, wollen wir sie genauer aufteilen.

Sensoren bezeichnen alle Hardwarekomponenten, welche Daten liefern. Dazu zählen sowohl jene Sensoren, die für den Zweck des Geräts notwendig sind, z.B. ein Thermometer für ein Thermostat, als auch jene, die unterstützend wirken, wie ein Batterieladestandsmesser für ein batteriebetriebenes IoT Gerät.

Aktuatoren beinhalten alle Elemente des Geräts, welche mit der Umwelt interagieren. Dazu zählt z.B. der Auslöser für den Türöffner bei einem smarten Schloss. Außerdem können hierzu auch Komponenten gezählt werden, welche indirektere Auswirkungen auf die Umwelt haben können, z.B. der Anschluss an das Stromnetz.

1.2. IT Komponenten

Die Komponenten, welche im vorherigen Unterkapitel besprochen wurden, bilden die Brücke zur analogen Welt. Jetzt geht es um jene Komponenten, die die Basis für den digitalen Teil des IoT Geräts bilden.

Hier sollten alle Komponenten, die aus PCs bekannt sind, so weit vorhanden, aufgezählt werden:

1. CPU (inkl. Architektur)
2. ROM (nicht überschreibbar)
3. Flash Speicher (wiederbeschreibbar)
4. RAM
5. Netzwerkschnittstellen (LAN, WLAN, Bluetooth, Zigbee, …)
6. Display
7. Netzteil
8. Ports (USB, JTAG, UART, Seriell, …)
9. Kartenleser

1.3. Betriebssystem

Falls das IoT Gerät darüber verfügt, ist es notwendig das Image des Betriebssystems genau zu untersuchen. Wichtige Elemente sind:

1. Basis (Linux, BSD, Windows, …)
2. Version des Betriebssystems (‚Kernel-Version' und ‚Release')
3. Versionen mitgelieferter Libraries
4. Vorab installierte Softwarekomponenten bzw. Libraries und ihre Versionsnummern
5. Spezielle Hersteller-Libraries für Sensoren und Aktuatoren
6. Laufende Prozesse bei Start des ‚leeren' Images (SSH, Webserver, …)
7. Für die Nutzung des IoT Geräts zusätzlich installierte Software (Application Server, Interpreter, …)

1.4. Netzwerkkommunikation

Hierbei handelt es sich wieder um eine Schicht des ursprünglichen IoT Modells. Gemeint ist Netzwerkkommunikation auf Softwareebene. Dazu zählt, welche Protokolle genutzt werden, ob diese verschlüsselt sind, mit welchen Services kommuniziert wird, wie oft diese Kommunikation stattfinden soll, ob andere Geräte aktiv Verbindungen zu dem IoT Gerät aufbauen dürfen, oder nur das IoT Gerät zu einem Server.

1.5. Anwendung

Im Modell ist hiermit die Anwendung gemeint, die die Value-Proposition des IoT Geräts ist, also das, was das Gerät ‚smart' macht.

Die Anwendung besteht somit üblicherweise aus 3 Komponenten:

1. Software am IoT Gerät
2. Cloud Service
3. Front-End Anwendung (Smartphone, Web, Sprachassistent, …)

Für das Threat Model ist aktuell die Software am IoT Gerät relevant. Diese sollte bereits durch ein bestehendes Threat Model für die Anwendung abgedeckt sein. In Zusammenhang mit der Software am IoT Gerät ist das Cloud Service noch insofern wichtig, als dass es gilt festzustellen, inwieweit Zwischenfälle in der Cloud zu Gefahren für das Gerät bzw. die AnwenderInnen führen kann. Gemeint ist, ob beispielsweise durch Kompromittierung der Cloud ein Mikrofon im IoT Gerät für Spionage-zwecke missbraucht werden könnte.

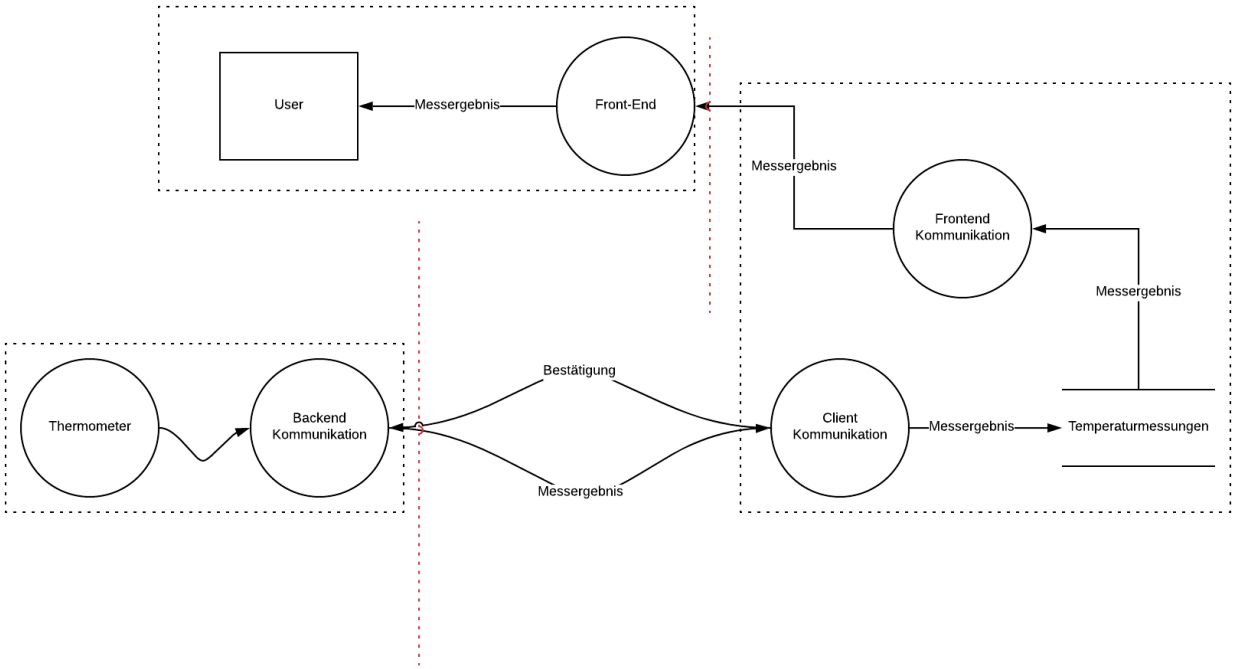
2. Grundlagen STRIDE

Für den Rest des Dokuments wird angenommen, dass für die Anwendung beim Softwareent-wicklungsprozess bereits ein Threat Model in STRIDE erstellt wurde. Sollte dies nicht der Fall sein, folgt eine Einführung in STRIDE und die Anwendung.

STRIDE ist eine Abkürzung und steht für:

- Spoofing
- Tampering
- Repudiation
- Information Disclosure
- Denial of Service
- Elevation of Privilege

Ziel des Frameworks ist es allen Bestandteilen eines Softwareengineeringprojekts die passenden Schwachstellenklassifikationen zuzuordnen. Dazu werden Datenfluss Diagramme (DFD) genutzt. Ein Beispiel für ein Datenfluss Diagramms einer einfachen Thermostatanwendung sieht wie folgt aus:



Je nachdem, wie sorgfältig bei deren Erstellung vorgegangen wurde, müssen vorhandene komplexe Prozess-Elemente nur noch in Primitive aufgelöst werden (dies trifft in der Abbildung oben auf alle Kreise zu, da es sich nur um eine high-level Übersicht handelt) und um Vertrauensgrenzen erweitert werden. Anschließend werden alle Elemente, dazu zählen auch Datenströme selbst, durchnummeriert und nach Vorgaben des Frameworks beurteilt. Die korrekte Beurteilung kann der Tabelle unten entnommen werden.

DFD Element	S	T	R	I	D	E
Externe Entität	X		X			
Datenfluss		X		X	X	
Datenspeicher		X	X	X	X	
Prozess	X	X	X	X	X	X

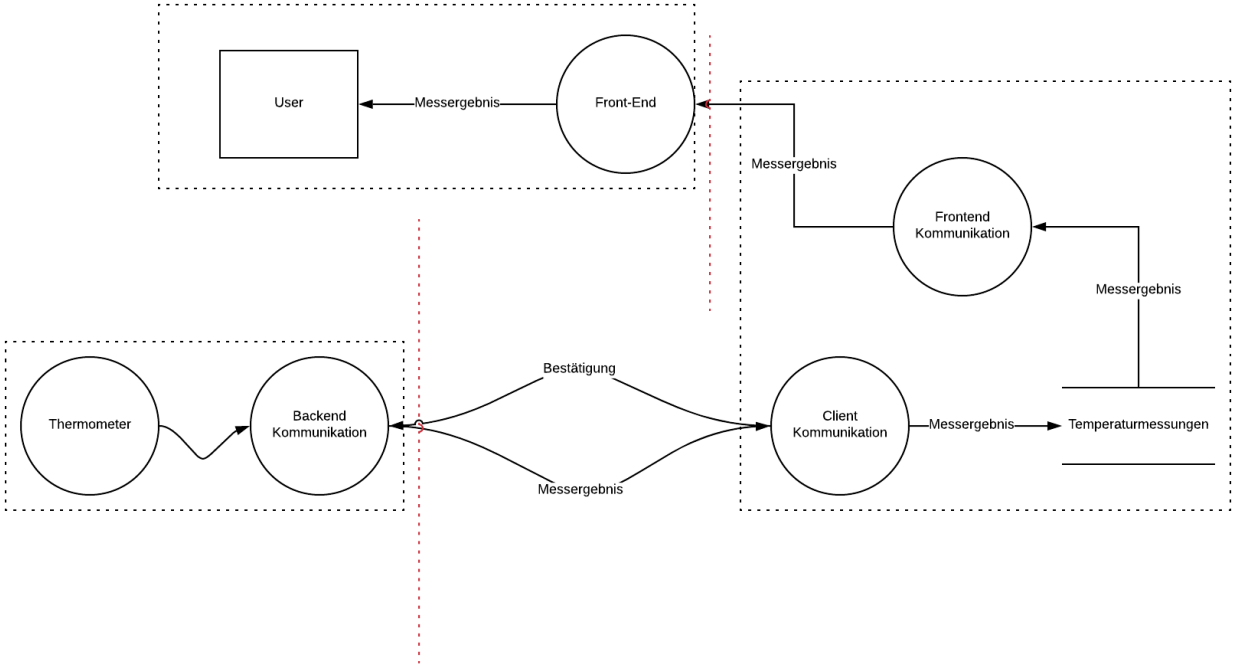
Die anschließende Bewertung der Risiken ist in diesem Dokument nicht vorgesehen, da dies auch individuell vom zu beurteilenden Gerät abhängt.

3. Erweiterung für IoT Geräte

In diesem Kapitel ist das Ziel Richtung eines ganzheitlichen Threat Model zu gelangen. Wir fangen hierfür mit der Software-Seite an. Im Anschluss werden wir die Common Attack Pattern Enumeration and Classification (CAPEC) [1] nutzen, um weitere Angriffsmöglichkeiten in das bestehende STRIDE Modell einzugliedern.

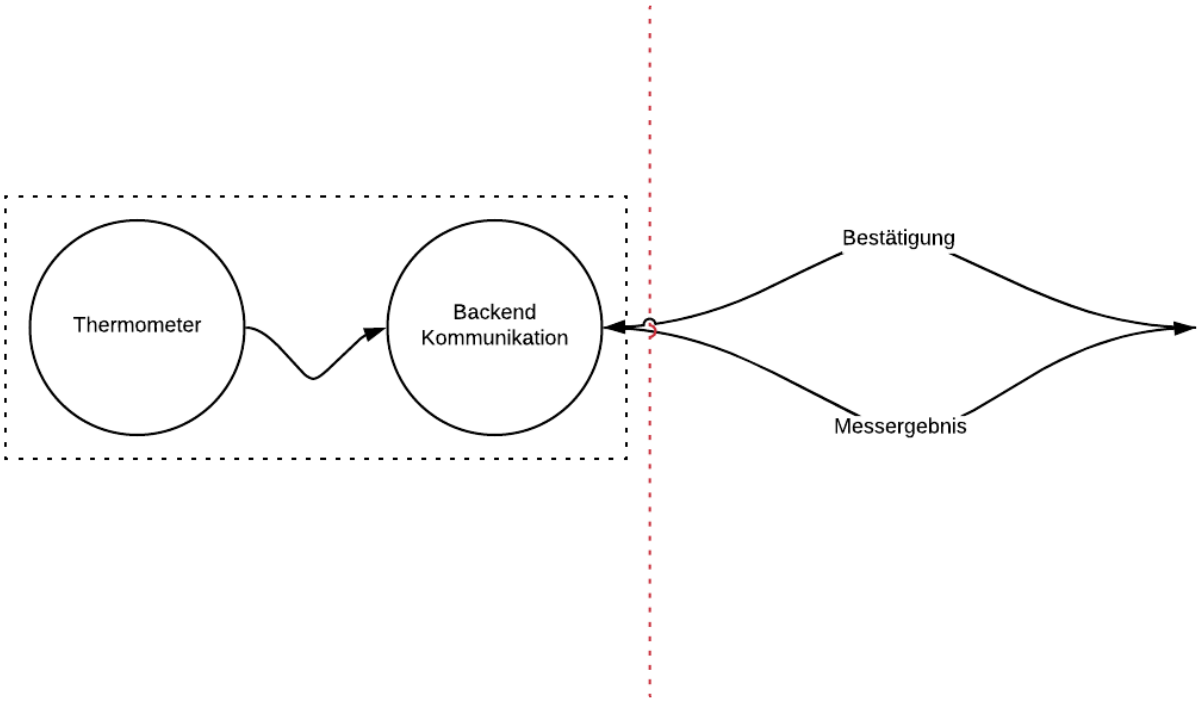
3.1. Anwendung

Als Beispiel wollen wir ein einfaches Datenfluss Diagramm verwenden. Es handelt sich dabei darum, wie der Datenaustausch zwischen einem Smart Thermometer und einer Front-End Smartphone App aussehen kann. Das Hardware Thermometer liefert die Temperaturmessergebnisse an eine Backendkommunikationskomponente. Diese übermittelt die Daten an einen Server, der sie in einer Datenbank speichert um sie bei Bedarf einer Smartphone App zu Verfügung zu stellen.

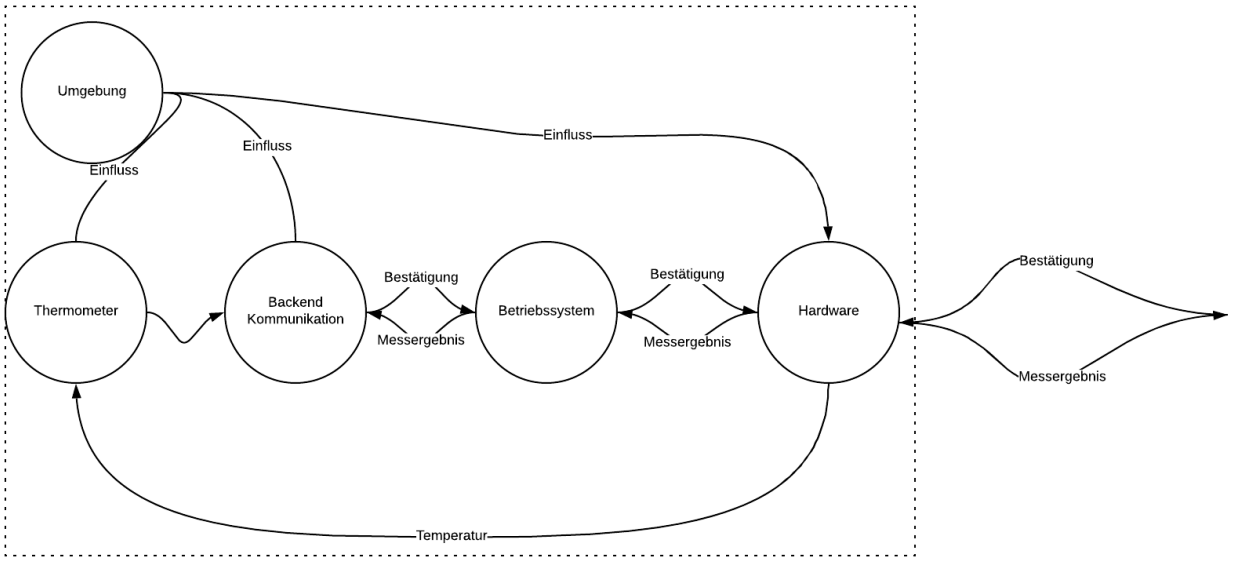


Die erste Version des Diagramms ist ausreichend um aus Software Engineering Sicht einen Überblick über die Datenflüsse zu erlangen. Die Trust Boundaries, welche als rot-strichlierte Linien eingezeichnet sind, geben Aufschluss darüber, wann die Daten über ein unsicheres Netzwerk übertragen werden. Im Folgenden wollen wir uns ausschließlich auf das IoT Gerät links unten im Diagramm befassen.

Das IoT Gerät besteht aus Software-Sicht aus zwei großen Prozessen, der Software um Messungen mit dem Thermometer zu machen und dem Softwareteil, der die Messergebnisse an den Server weiterleitet.



Im üblichen Softwareengineering Kontext ist diese Darstellung ausreichend um ein Threat Model, beispielsweise mit STRIDE, anzufertigen, wenn die beiden Prozesse noch entsprechend aufgelöst werden. Im IoT Kontext sollte trotzdem ständig die Umwelt beachtet werden. Vor allem, wenn man weiter mit STRIDE arbeiten möchte, lohnt es sich, sie als eigenständigen komplexen Prozess zu betrachten, der noch weiter aufgelöst werden muss. Ebenso ist es bei Kommunikation nach außen besonders wichtig auf das Betriebssystem und sogar die Hardware zu achten. Auch diese beiden Elemente können als Prozesse dargestellt werden.



Zu diesem Zeitpunkt verfügen wir über ein Threat Model, das die Anwendung und grobe Züge der Umwelt abdeckt. Nun geht es darum die darunterliegenden Schichten mit einer entsprechenden Kategorisierung zu versorgen.

3.2. Netzwerkkommunikation

Wir befassen uns als nächstes mit der Netzwerkkommunikation. Dabei geht es um die zugrundeliegenden Mechanismen, wie TCP/IP und nicht um den Inhalt der Datenpakete, welche von der Anwendung versandt werden.

Wir untersuchen die Angriffe aus CAPEC Kategorie ‚Communications' (512). Dabei bewegen wir uns jeweils eine Hierarchieebene unter der Kategorie. Darunterliegende Angriffe werden als Beispiele verstanden:

Angriffsmuster	S	T	R	I	D	E
Man in the Middle		X				
Sniffing Attacks				X		
Eavesdropping				X		
Content Spoofing	X				X	
Identity Spoofing		X	X			X
Resource Location Spoofing	X	X		X	X	X
Footprinting				X		
Infrastructure Manipulation	X	X		X	X	X
Protocol Analysis				X		
Communication Channel Manipulation				X		

Protocol Manipulation		X		X	X	X
Traffic Injection		X			X	X
Obstruction					X	

Es bleibt dem/r Modellierer/in überlassen zu identifizieren, welche der genannten Angriffsmuster im konkreten Fall relevant sind.

3.3. Betriebssystem

Das Threat Model für das Betriebssystem muss in zwei Teile geteilt werden. Einerseits ist es notwendig, die Gefahren für laufende Daemons, wie ssh, zu modellieren. Bei Open Source Tools ist dies prinzipiell wieder unter Verwendung von Datenfluss-Diagrammen und STRIDE möglich. Andererseits muss das Betriebssystem in seiner Gesamtheit betrachtet werden. Hierbei geht es vor allem um Härtung durch das Setzen entsprechender Rechte für User und Aktualisierungen für Bibliotheken. Es lohnt sich hierfür Härtungsanleitungen für Betriebssysteme zu nutzen, anstatt zu versuchen, ein erschöpfendes Threat Model zu erstellen und sich so der Gefahren für das Betriebssystem bewusst zu werden. Die NIST stellt hierfür Checklisten für viele gängige Programme und Betriebssysteme zu Verfügung. [2]

Generell ist festzuhalten, dass ein umfassendes Threat Model des Betriebssystems und der Dienste die darin laufen mit STRIDE nur dann erstellt werden kann, wenn die Quellen eingesehen werden können. In einem IoT Projekt ist es jedoch unrealistisch, dass das Budget dafür bereit steht, den gesamten Quellcode des Geräts zu untersuchen. Darum empfiehlt sich, die Risiken auf Basis bestehender CVE Einträge und wie schnell Sicherheitspatches bereitstehen, abzuschätzen.

3.4. Hardware und Sensoren/Aktuatoren

Zuletzt bleibt die physische Repräsentation des IoT Geräts. Wir nutzen hierfür wieder die CAPEC und untersuchen relevante Szenarien aus „Hardware“ (515), „Supply Chain“ (437), „Physical Security“ (514):

Angriffsmuster	S	T	R	I	D	E
Content Spoofing	X				X	
Resource Location Spoofing	X	X		X	X	X
Hardware Integrity Attack	X	X	X	X	X	X
Malicious Logic Insertion		X			X	
Contaminate Resource		X		X		X
Fault Injection		X			X	
Modification during Manufacture/Distribution	X	X	X	X	X	X
Theft		X		X	X	
Obstruction					X	

Aus Software Engineering Sicht ist es besonders wichtig zu bedenken, dass die Annahme, ein Gerät wäre in einer sicheren Umgebung, wie einem Büro, und würde dort sorgsam behandelt, für IoT Geräte nicht gilt.

Bezüglich Sensoren und Aktuatoren muss die Frage, was sie an ihrer Funktionalität hindert und sicherheitskritisch ist, sehr individuell gestellt werden. Beispielsweise kann die Manipulation eines Thermometers genutzt werden, um im Kühlkontext Fleisch verderben zu lassen. Da IoT Geräte IT mit vielen anderen Domänen vereinigen, lohnt es sich auf Bedrohungsmodelle der jeweiligen Domänen zu schauen.

4. Abschluss und nächste Schritte

Mit dem integrierten Threat Model sollte die Gefahrenlandschaft für das IoT Gerät um einiges deutlicher und realistischer geworden sein. Mit dem neuen Wissen ist es möglich, die Gefahren für das Gerät umfassend zu bewerten.

Wie die Bewertung stattfindet, bleibt hierbei frei. Microsoft bietet hierfür zu STRIDE passend DREAD an. Allerdings ist es ebenso möglich, lediglich zu entscheiden, ob man ein Risiko beheben, eindämmen oder mit seinen Konsequenzen akzeptieren will. Es ist typisch unmöglich, für alle Szenarien einen Schutz zu bieten und dabei ein Gerät zu einem kompetitiven Preis zu produzieren. Dies gilt vor allem für Angriffe wie Diebstahl, gegen die im physischen Sinn oft wenig zu machen ist. Trotzdem kann dafür gesorgt werden, dass im Fall des Falles die Daten der AnwenderInnen entsprechend geschützt werden.

Referenzen

[1] MITRE, „CAPEC - Common Attach Pattern Enumeration and Classification,“ [Online]. Available: https://capec.mitre.org/data/index.html. [Zugriff am 07 05 2019].

[2] NIST, „NCP - National Checklist Program Repository,“ [Online]. Available: https://nvd.nist.gov/ncp/repository. [Zugriff am 16 05 2019].