IT-Sicherheit für IoT-Protokolle

Mohamed Sohil
Mohamed.Sohil@stud.h-da.de
Wissenschaftliches Arbeiten II
Hochschule Darmstadt

28. Mai 2025

Abstract Der Begriff (Internet der Dinge), kurz IoT (Internet of Things), beschreibt die Vernetzung von kleinen, mobilen Geräten bis hin zu smarten Fabriken und Häusern oder autonomen Systemen. Die Verwendung des Internets wird sich in der Zukunft nicht nur auf PC oder Smartphone beschränkt, denn schon heute gibt es Embedded Geräte, die miteinander vernetzt werden. Doch die Vernetzung von IoT-Geräten wird von Sicherheitsrisiken wie Hackerangriffen oder Sicherheitslücken begleitet, die wirtschaftliche Schäden verursachen können. Diese Arbeit befasst sich mit der IoT Protokolle, wie MQTT,HTTP ,CoAP, XMPP, die Beschreibung der Protokolle und Kommunikationsarchitektur. es wird danach die Angriffe auf diese Protokolle behandelt. Es wird auch nach Maßnahmen gegen die Angriffe erforscht werden.

Inhaltsverzeichnis

1 Methodik

zu Beginn dieser Arbeit werden die Forschungsfragen bestimmt , und die Recherchierung der Literaturquellen durchgeführt und gesammelt ,um die folgenden Forschungsfragen beantworten zu können :
was ist Internet der Dinge , und in welchem Anwendungsfelder wird es verwendet ?
Welche sind die IoT-Protokolle?
Welche Schwachstellen haben sie ?
Welche Maßnahmen werden befolgt ,um die IoT-Protokolle zu sichern?
Das Suchverfahren wurde in Suchmaschinen , sowie in Webseiten durchgeführt ,Google, Recherchiere nach Büchern in der Bibliothek , Am anfange der Arbeit wird nach der Definition vom Internet der Dinge , und In Welchem Bereich gesucht , es wird hier im Paper kurz zusammengefasst , und geschrieben ,dann werden IoT-Protokolle ausführlich erläutert , diskutiert werden folgende:
-Beschreibung des Protokolles
-die Kommunikationsarchitektur
-die Kommunikationsform
-die Kommunikatinswege , sowie die Adressierung , -die Kommunikationsart , und format , am Ende wird kurz die Vor –und Nachteile des Protokolles.
die zweite Phase der Arbeit wird nach möglichen Angriffen auf die IoT-Protokolle, welche Gefahren verstecken, wenn die Kommunikation nicht abgesichert wird. bei der letzten Phase der Arbeit wird erforscht , wie die IoT-Protokolle gesichert , was berücksichtigt wird drüberhinaus werden Ergebnisse in worten zusammengefasst , und dargestellt ,das heißt , qualtative Forschung wird durchgeführt.

2 Einführung

Das Internet der Dinge vernetzt verschiedene Objekte Das Internet der Dinge vernetzt verschiedene Objekte über das Internet miteinander. Dadurch können diese Objekte miteinander kommunizieren und verschiedene Aufgaben für die Nutzer erfüllen[1].

solche Objekte, die Geräte oder Maschinen sein können, werden dazu mit Prozessoren und eingebetteten Sensoren ausgestattet[2], diese Geräte können Daten sammeln und austauschen , sowie für weitere aufgaben verwendet werden [3] . Beispiele für die Geräte sind die Mobiltelefone bzw. Smartphones und PCs. Mittlerweile werden aber nicht nur Smartphones oder PCs miteinander verbunden, sondern auch immer mehr Geräte des täglichen Lebens. Gemäß der Aussage der Experten des IEEE (Institute of Electrical and Electronics Engineers) wird sich die Anzahl der vernetzten Gegenstände bis zum Jahr 2020 auf rund 100 Mrd. erhöht. Dazu gehören Maschinen ,die in der Industrie benutzt werden, aber auch alles, was man sich vorstellen kann: vom Kaffemaschine über die allgemeine Haustechnik(Wasch-Spülmachische , Kühlschrank), Fahrzeuge, Sensoren aller Art bis hin zu Warenverpackungen[4]. Das Internet der Dinge wurde im gleichen Zeitraum wie das Internet selbst entstanden. Im Jahr 1990 wurde Das erste vernetzte Haushalt-Gerät probiert. Dies war ein Toaster, der über das Internet (On-line) ein und ausgeschaltet werden konnte[5]. Der Begriff Internet der Dinge ist trotzdem noch nicht so lange bekannt. Seit 2009 beginnt offiziell mehr Geräte als Menschen die mit dem Internet vernetzt zu sein. Dies war eine große Fortschritt und wurde von Cisco sogar als die Geburtsstunde des Internet der Dinge betrachtet[6]. Durch die Allgegenwärtigkeit, ist es selbstverständlich, dass diese Technologie in verschiedensten Feldern zur Anwendung kommt. Dazu gehören: -Smarthome - Mobilität - Gesundheitswesen - Logistik in Unternehmen - Produktion und Fertigungsplanung innerhalb von Unternehmen

3 IoT Protokolle der Anwendungsschicht

Anwendungsprotokolle sind Protokolle ,die in den schichten 5 bis 7 gemäß des OSI-Referenzmodells enthalten ,und dienen zur Kommunikation zwischen Anwendungen (Anwendungsprogramm) und wie die Informationen korrekt über Transportschicht vermittelt werden sollen[7]die Transportschicht ermöglicht unterschiedliche Kommunikationsformen die verbindungsorientiere und verbindungslose Kommunikation, die später erläutert werden[37].
Im Internet gibt es viele unterschiedliche Anwendungen, die von Protokollen unterstützt werden. Hier gehören Protokollen wie (SMTP) oder das Simple Mail Transfer Protocol (Einfaches E-Mail-Transportprotokoll) welches die Mails von einem Sender zum Empfänger überträgt, das File Transfer Protocol (FTP) welches für Datenübertragungen innerhalb eines Netzwerks sorgt, mit DNS (Abkürzung von Domain Name Service) können vernetzten Computern sich gegenseitig lokalisiert werden und das DHCP (Abkürzung von Dynamic Host Configuration Protocol), welches dafür sorgt, dass Rechner in einem Netzwerk eine IP-Adresse (logische Adresse) zugewiesen können[8].
HTTP, CoAP und XMPP sind ebenfalls Protokolle die Anwendungsschicht gehören. Das Besondere an diesen Protokollen ist, dass sie innerhalb von IoT Netzwerken zur Verwendung kommen

3.1 MQTT

3.1.1 Beschreibung

MQTT : ist abkürzung von Message Queuing Telemetry Transport Bei MQTT handelt es sich um ein Publish-Subscribe-Verfahren. Und für die M2M (Machine to Machine) Kommunikation entwickelt ,die Teilnehmer im Netzwerk können dabei nicht nur Daten anderer Teilnehmer abfragen, sondern können sich automatisch darüber informieren lassen, wenn sich Werte ändern[9].
MQTT ist ein Protokoll, das über einen minimalen Overhead verfügt,und ermöglicht es, eine einfach Kommunikation zu implementieren[10]. Drüberhinaus ist es bekannt , dass MQTT-Protokoll ein leichtgewichtiges, effizientes, sicheres, ereignis- und nachrichtenorientiertes Anwendungsprotokoll ist[39].

3.1.2 Kommunikationsarchitektur

MQTT beruht auf der Publish/Subscribe-Architektur, bei dem die Nachrichten von einem Nachrichtenvermittler (Broker) empfangen oder weitergeleitet werden.die Kommunikation erfolgt nicht direkt zwischen dem Nachrichtensender und -empfänger[39]. Die Clients müssen nichts voneinander wissen,und Das Senden und Empfangen erfolgt ausschließlich über den Broker[11].

3.1.3 Kommunikationsform

weil das MQTT anders als ein Request/Respone-Modell basiert wird , ist die Kommunikationsform nicht einfach festzulegen. Die Kommunikation erfolgt zwischen dem Subscribers und dem Broker „asynchron“ . Der Subscriber abonniert ein Topic und erhält eine Nachricht/ oder Antwort, sobald etwas an diesen Topic publiziert wurde.die Kommunikation durch den Publisher erfolgt hingegen grundsätzlich weder synchron noch asynchron[39]. Er schickt eine Nachricht an den Broker. Wenn die Möglichkeiten des QoS-Levels (Quality of Service) berücksichtigt werden, bei dem Nachrichten auch vom Empfänger bestätigt werden, so findet eine synchrone Kommunikationskomponente zwischen Sender und Empfänger (Publisher und Broker/Broker und Subscriber) statt [12].

3.1.4 Kommunikationswege und Adressierung

die Adressierung der Nachrichten erfolgt über Topics. Beispiel : Der Temperatursensor misst 21 Grad , und veröffentlicht, was er liest als Publisher unter dem Topic „temperature“,der Laptop bzw. mobile device haben vorher dem Broker mitgeteilt , dass sie Nachrichten mit dem Topic „Temperature“ abonnieren will, der Broker empfängt die Nachricht „21 C“,unter dem Topic „temperature“,prüft,welche Clients diese abonniert haben, und sendet diese an entsprechenden Clients wie Laptop oder mobile Device. Die Adressierung auf der Transportschicht übernimmt der Broker. Die Clients müssen die IP-Adresse oder den Hostnamen des Brokers beim Aufbau der Verbindung kennen[39].

3.1.5 Kommunikationsrichtung

bei dem MQTT handelt sich um unidirektionale Kommunikation von der Publischer-Seite (Über den Broker) zur Subsriber-Seite , das liegt daran , dass die Publisher und Subscriber nichts von iherer gegenseitigen Existenz wissen , außerdem findet die Verbindung über den Broker statt ,aber die Kommunikation zwischen dem Client und dem Broker erfolgt hingegen bidirektional,wodurch der Empfänger die Einreichung der Nachricht bestätigt[11].

3.1.6 Nachrichtenformate

MQTT ermöglicht dem Entwickler, in welchem Format die Nachricht übertragen werden soll. Das MQTT- Protokoll ermöglicht sowohl die Übertragung in Form von binären Daten als auch reinen Plain Texte und mit JSON oder XML strukturierten Texte[14]. .

3.1.7 Die Vor- und Nachteile

Die Vor- und Nachteile, die im Laufe der Recherchierung ersichtlich wurden, werden auch in anderen Berichten genannt[15] und sind hier aufgelistet.
Die Vor- und Nachteile der Verwendung vom MQTT-Protokoll werden erforscht , und zusammengefasst und hier aufgelistet:
Vorteile:
• MQTT-Protokoll wird mit geringer Overhead aufgebaut
• es ist skalierbar, und umfasst bis zu mehreren hunderttausenden Clients pro Server
• unterstützt Protokollfeatures, die für IoT-Anwendungsfälle entwickelt wurden
• das Protokoll ist einfach zu implementieren ,deswegen wird für Vernetzung der Geräte mit geringen Reessourcen konziptiert
• Clientimplementierungen sind für alle gängigen Programmiersprachen verfügbar
falls die Verbrindung abgebrochen ist,können sie wieder aufgebaut,und fortgesetzt werden.
Nachteile:
die Benutzung von Request/Response-Architekturen ist mit MQTT kann nur mit Aufwand umgesetzt bzw. mitgerechnet werden [39].

3.2 HTTP

3.2.1 Beschreibung

HTTP oder (Hyper Text Transfer Protokoll) ist ein Request/Response Protokoll und mit dem Internet entstanden , und landet in der siebten Schicht im ISO/OSI-Schichtenmodell , und gilt als einfaches Protokoll für die Übertragung von Nachrichten bzw. Daten im MIME Stil. http ist nicht nur für die Übertragung von HTML-Seiten geeignet, sondern kann fast überall genutzt werden, wo ein grundliegender Zugriff auf Hypermedia benötigt wird wie FTP [16]. erwäheneswert ist http ein zustandsloses Protokoll , nach der Antwort

der Anfrage wird die Verbindung wieder abgebaut. http-Protokoll kümmert sich darum , dass Anfragen/Request an den Server übertragen wurden und der Server diese Anfrage empfängt , und erfolgreicht bearbeitet , und sendet dazu das angemessene Response. Dazu hat der Client eine TCP/IP Verbindung mit dem Host hergestellt, wobei er den Domain-Namen oder die IP-Adresse und zusätzlich den Port angeben musste. Der Standardport für HTTP ist 80. Dieser wird verwendet, solange kein anderer angegeben ist. Die Anfrage und die Antwort bestanden aus ASCII-Zeichen. Die Anfrage an den Server enthielt die GET-Methode ein Leerzeichen und die Adresse des angefragten Dokuments. Wie GET /informationtext.html, (und falls das angefragete Dokument verfügbar ist , sendet es der Server an den Client http-Status OK. Der Server benutzte für das response anmessene Sprache für HTML. HTTP wie oben erwähnt ist [16]. Kurzbeschreibung: Weit verbreitetes und vielseitig unterstütztes Protokoll[39].

3.2.2 Kommunikationsarchitektur

HTTP-Protokoll bassiert auf der klassischen Request/Response-Architektur. Der Server erhält eine oder mehrere Anfragen (Request) vom Client , die Anfrage wird vom Server bearbeitet , und die Server sendet Antwort(Response) an den Client, dass er die Anfrage bearbeitet hat , diese Anfrage und Antwort bestehen aus ASCII-Code [17].

3.2.3 Kommunikationsform

Bei der Kommunikation zwischen dem Client und Server erfolgt über HTTP synchron .

3.2.4 Kommunikationswege und Adressierung

es sind nur Unicast-Nachrichten möglich , weil es nur Verbundung zwischen Client und Server gibt , das bedeutet eine oder meherere Nachrichten dürfen nicht versandt werden[17]. Bei der Adressierung gibt es den sogenannten URI oder (Unified Ressource Identifier .Die Adressierung erfolgt über einen URI (Unified Ressource Identifier) genauer über einen URL (Unified Ressource Locator). hier erfolgt die Adressierung über Hostnamen (lesbar für die Menschen) oder über die IP-Adresse [17]

3.2.5 Kommunikationsrichtung

da Kommunikation ist nur zwischen Client und Server (es basiert auf Request/Response-Modell) erfolgt , ist die Verbundung bidirektional[17].

3.2.6 Nachrichtenarten

HTTP bietet verschieden Nachrichtenarten , und stellt daher Request-methoden zur Verfügung Sie werden hier aufgelistet und kurz beschrieben :
GET
Mit GET Request-Method wird vom Server über Resourcen angefordert,daher bleibt Body leer.Der Server wird dann Antworten[16]. GET-Methode ist sicher und idempotent , sicher bedeutet , dass GET den Zustand auf der Server-Seite niemals verändert werden, und GET Methode wird als idempotent betrachtet ,wenn der status von einem Request ist das gleiche , wenn der Server das gleiche Request mehrmals bekommt[38].
HEAD
Diese Request-Methode ähnelt der GET-Methoder beim Aufbau ,doch bei der Antwort wird nur der Header gesendet. Diese Methode wird meistens verwendet , um zu prüfen , ob die URLs verfügbar sind[16].

POST
im Gegensatz zu GET können Bei einer POST-Methode Informationen im Body an den Server gesendet werden.bei POST ist die Länge nicht begrenzt im Vergleich zu GET[16].
PUT
Mit PUT-Methode wird vom Server aufgefordert, Resources bzw. Informationen zu ändern ,oder zu aktualisieren und wieder zu speichern. PUT-Methode ist unsicher, aber als idempotent betrachtet[38]
DELETE
Mit DELETE-Methode wird vom Server aufgefordert , eine angegebene Ressource zu löschen [16].

3.2.7 Nachrichtenformate

Das Headerformat bei der http ist in Textform. die Nachrichten können trotzdem in Binärdaten zwischen dem Netzwerkteilnehmer ausgetauscht werden[16]

3.2.8 Die Vor- und Nachteile

Die Vor- und Nachteile, die im Laufe der Recherchierung ersichtlich wurden, und sind hier aufgelistet.
Vorteile :
• der Protokoll wird sehr einfach aufgebaut
Nachteile
• jede Anfrage verlangt großen Overhead • unterstützt Keine Push-Nachrichten

3.3 CoAP

3.3.1 Beschreibung

CoAP ist die Abkürzung von (Constrained Application Protocol) und wird für die Verwendung in eingeschränkten verwendet, die zumbeispiel mit wenig Energie und auch einem geringen Datenverkehr auskommen müssen. Entwickelt wurde es von der IETF Contrained RESTful environments (CoRE) Working Group.das Hauptziel für das Protokolldesign ist das geringe Overhead. CoAP sollte ein Protokoll entwicklet werden, das nach den Prinzipien der REST-Architektur und ähnlich dem HTTP-Protokoll für M2M-Netzwerke optimiert ist[18]. Kurzbeschreibung: Leichtgewichtiges M2M-Protokoll mit der einfachen Möglichkeit der Überführung in HTML.

3.3.2 Kommunikationsarchitektur

CoAP lehnt an http-Protokoll an ,deswegen beruht die CoAP-Architektur auf Anfrage/Request-Modell ,im Unterschied zu http –Protokoll kann die Resource mit GET-Methode abonnieren , und das Abonnement ist nicht zeitlich begrenzt. Wie bei MQTT-Protokoll[18]. Der Client muss dann vor Ablauf der sogenannten „Subscription-lifetime“ erneut eine Anfrage stellen. [19].

3.3.3 Kommunikationsform

da die Kommunikation auf Request/Response-Modell basiert , ist die Verbindung synchron aufgebaut [18].Es besteht die Möglichkeit, dass zeit sich verzögert und mehrfach eine Antwort auf einen Request zu bekommen , eine asynchrone Kommunikationsform hinzu [19].

3.3.4 Kommunikationswege und Adressierung

CoAP wird auf UDP-Protokoll basiert und Durch die Verwendung von UDP kann CoAP Muiticast-Nachrichten an mehrere Empfänger gleichzeitig versenden . Die Adressierung ähnelt der Adressierung von HTTP-Protokoll über CoAP-URI. Als Host kann entweder ein Name oder eine IP-Adresse benutzt werden [18].

3.3.5 Kommunikationsrichtung

bidirektionale Kommunikation ist bei Dem CoAP , da die Endgeräte direkt miteinander kommunizieren

3.3.6 Nachrichtenarten

CoAP verwendet die HTTP Request-Methode
(POST,GET,DELETE,PUT,HEADER)[18].

3.3.7 Nachrichtenformate

CoAP unterstützt das Textform . die Nachrichten können auch in Binärdaten zwischen Server und Client ausgetauscht werden[18]

3.3.8 Die Vor- und Nachteile

Die Vor- und Nachteile, die im Laufe der Recherchierung ersichtlich wurden,und sind hier aufgelistet.
Vorteile :
• das Standard ist offen
• hat geringen Overhead
Nachteile
• Primär für die direkte Kommunikation zwischen zwei Geräten konzipiert.
• das Protokoll ist kein richtiges Publish/Subscribe

3.4 XMPP

3.4.1 Beschreibung

XMPP ist die Abkürzung von(Das Extensible Messaging and Presence Protocol) also das Erweiterbare Nachrichten- und Anwesenheitsprotokoll (früher Jabber) kommt aus dem Bereich des Instant Messaging. Es ermöglicht den Austausch von XML (Extensible Markup Language)-Datenformat. Es ist erweiterbar, was die Einsatzmöglichkeiten ausweitet[20].
Kurzbeschreibung: Vielfältig einsetzbares und erweiterbares Protokoll

3.4.2 Kommunikationsarchitektur

Die Kommunikation bei XMPP erfolgt über sogenannte Streams,das als Container für Datenaustausch verwendet wird . ein Stream wird zwischen zwei Netzwerkpunkten aufgebaut. Die Nachricht wird in einem „XML Stanza“ übertragen. Diese kann entweder eine einfache Nachricht (message), eine Anwesenheits-/Aktivitätsanfrage (presens) oder eine Abfrage (iq = info/query) sein. Über iq kommt die Request/Response-Architektur zum Einsatz[20].
XMPP hat auch eine sogenannte Extension, mit der auch eine Puplish/Subscribe-Architektur umgesetzt werden kann[21].

3.4.3 Kommunikationsform

die Kommunikation kann sowohl synchrone als auch asynchrone sein [20]. Drüberhinaus gibt es Erweiterung , mit der auch das sogenannte Pipelining möglich ist[22].

3.4.4 Kommunikationswege und Adressierung

es handelt sich bei XMPP zunächst um eine Ende-zu-Ende-Kommunikation über Unicast[20]. Mit der Erweiterung können auch Multicast-Nachrichten versendet werden. Unter Voraussetzung, dass der Server diese Erweiterung aufweist. Der Client kann dies über Info-Abfrage an den Server vorab feststellen [**?**]. Analog zu MQTT-Protokoll läuft die Kommunikation über einen zentralen Verwaltungsserver. Außerdem können Nachrichten auch per Broadcast an alle mit dem Server verbundenen Endgeräte gesendet werden[20]. bei der Adressierung werden bei XMPP global einzigartige Adressen verwendet (basierend auf DNS), um die Nachrichten über das Netzwerk zu führen.

3.4.5 Kommunikationsrichtung

bei XMPP handelt sich um bidirektionale Kommunikation . Am anfang wird bei der Verbindungsaufnahme mit Empfänger [20].

3.4.6 Nachrichtenarten

viele Nachrichten können bis zum Beenden der Verbindung gesendet werden. Sobald ein XMPP-Stream Verbindung aufgebaut wurde . es gibt drei verschiedene Nachrichtenarten [20]. <message/> Eine message-Nachricht, bei der ein Endgerät eine Nachricht an ein anderes Endgerät sendet. Die Nachricht enthält ein einziges Attribut („to“)[20]. <presence> , <iq> [20]

3.4.7 Nachrichtenformate

XMPP unterstützt das XML-Daten bzw.XML-Format (Textdaten) zwischen zwei Netzwerkteilnehmern. Aber über Erweiterungen können auch Binärdaten geschickt werden[20] .

3.4.8 Die Vor- und Nachteile

Die Vor- und Nachteile, die im Laufe der Recherchierung ersichtlich wurden, und sind hier aufgelistet. Vorteile
• Implementierungen für Programmiersprachen
• Sehr viele Features durch Erweiterungen
Nachteile
• Protokolloverhead ist hoch
• Optimiert für Instant Messenger-Anwendungsfälle

3.5 DDS

3.5.1 Beschreibung

das DDS Protokoll ist (for Data Distribution System)und wurde für den Bereich Luftfahrt und Verteidigung konzipiert. Es ist seit vielen Jahren und bei dezentralen Echtzeit-Systemen im Einsatz[24].

3.5.2 Kommunikationsarchitektur

Bei DDS handelt es sich um ein datenzentriertes Publish/Subscribe-Modell, welches im Unterschied zu dem MQTT-Protokoll ohne Message-Broker eingesetzt wird[24].

3.5.3 Kommunikationsform

bei DDS handelt es sich um eine asynchrone Kommunikationsform beim Datenaustausch[24]

3.5.4 Kommunikationswege und Adressierung

über DDS werden Nachrichten vom Publisher an einen Topic in das Netztwerk geschickt ,die Subscriber können in diesem Fall auf die Daten zugreifen , die Adressierung bei DDS findet nicht dirket von oder an einen Gerrät , und die Verbindung ist eher als Multicast zu sehen [25].

3.5.5 Kommunikationsrichtung

es handlt sich um bidirektionale Kommunikation zwischen den Netzteilnehmern[24].

3.5.6 Nachrichtenformate

Diese Struktur in dem die Nachrichtformat dargestellt wird ,entspricht einem Objekt und kann verschiedene Eigenschaften primitiver Datentypen enthalten[26].

4 Angriffe auf IoT Protokolle

In diesem Kapitel werden Beispiele für Angriffe gegen IoT-Protokolle erläutert. Denn durch das Ausnutzen der Schwachstellen können Angreifer eine Vielzahl von Angriffen durchführen, die eine Vielzahl von IoT-Systemen lahmlegen können.
Der Einsatz ungesicherter oder schlecht gesicherter Kommunikationen lösen Gefahren aus, auch Netztwerkteilnehmer , die Schwachstellen aufweisen. Sind Schwachlücken vorhanden, besteht die Möglichkeit,Hackers das System angreifen zu können. In der Folge werden über Angriffe auf IoT-Protokolle sowie IoT-Geräte erläutert . Das erste Beispiel für Angriffe ist DDoS-Angriff : Die Abkürzung für DDoS steht für Distributed Denial of Service , DDoS-Angriff besteht meistens aus einer Unmenge Anfragen ,die aus großem ferngesteuerten Botnetz stammen .die Gefahr besteht drin , wenn das Botnetz größer ist. Die Hersteller der IoT-Devices vernachlässigt häufig die Absicherung ihrer Produkte , und sorgen sich nur um die Funktionalität ihrer Geräte . bei vielen Produkten wird entweder gar keine oder lediglich begrenzte Sicherheitsmechanismen befolgt. Beides ermöglicht einen Angreifer das System attackieren zu können [29].

4.0.1 Angriffsszenarien und Schwachstellen

4.0.2 MQTT-Protokoll

die Kommunikation über MQTT ist unsicher. Die Kommunikation zwischen dem Publischer und Subscriber erfolgt ohne Verschlüssslung , und Authentifizierung.Ein Angreifer kann nicht nur mithören , sondern auch die Daten abändern,löschen , und weitersenden. Das Problem dabei liegt nicht dran , dass das MQTT-Protokoll unsicher ist, sondern die MQTT-Broker nicht ordentlich konfiguriert ist und immer ins Internet öffentlich gestellt wird. Das bedeutet, dass jeder Angreifer ohne Benutzername und Passwort auf den Broker einfach zugreifen, den Angreifer sind gelungen , Nachrichten zu schicken oder vom Broker

zu empfangen . Die Betreiber vernachlässigt also auf Authentifizierung . Im Browser der Internet der Dinge (Shodan) lässt sich die Broker finden ,wo einfach Publisher finden können , die sich in kritischen Infrastrukturen befinden und unsicher über das Netz kommunizieren. Um sich mit einem MQTT-Broker kommunikatzieren zu können, reicht nur ein Client, der per Hashtag alle Topics abonniert. Das Problem liegt nicht dran , dass sich die gesendeten Nachrichten lesen lassen, sondern, dass Broker viele Nachrichten von Publishern empfangen und an andere Subscriber weiterleiten. Auf diese Weisen werden Subscriber durch falsche Nachrichten manipuliert können . Angreifer schickt also als Publisher eine Nachricht mit Topic an den Broker , der ungeprüft an den Subscriber weiterleitet [28].

4.0.3 HTTP-Protokoll

Manche Daten ,die auf einem Webserver zu bearbeiten sind, sind nicht für jeden Menschen bzw. für jeden Anwender bestimmt und sollen nur für begrenzte Zahl von Personen zugänglich sein. Dazu gibt es die sogenannte http-Authentifizierung , in dem der Nutzer einen Benutzernamen und ein Passwort für die Authentifizierung verwendet.um sich einloggen zu können. Ohne den Benutzernamen und Passwort kann jeder auf Informationen bzw. Daten zugreifen , und manipulieren (Daten abändern ,oder löschen) und an andere Benutzer weiterleiten[29].

4.0.4 XMPP-Protokoll

Beim XMPP-Protokoll besteht die Gefahr drin , wenn kein Verschlüsselungsalogrithmen verwendet werden , denn die Integrität und Vertraulichkeit bedroht werden. Da die Informationen in XML Format übertragen werden , können sie einfach gelesen oder manipuliert werden. Um die Informationen bzw. Daten den Befugten zugänglich sind , und nicht manipuliert (vollständig , und korrekt) geschickt werden[36]

4.0.5 CoAP-Protokoll

CoAP ist wie im vorigen Kapital definiert , und gilt für Machine-to-Machine-(M2M)-Management-Protokoll, das auf IoT-Geräten eingesetzt wird.CoAP-Protokoll wird nicht nur für TCP als auch für UDP entwicklet, und benötigt deswegen keine Authentifizierung, um mit einer Rückmeldung auf eine kleine Request/Anfrage zu antworten

4.0.6 andere Angriffe

noch andere Schwachstellen , die Hackers erleichern , IoT-Geräte oder Netzwerke anzugreifen, und Informationen sowie Daten zu manipulieren hier werden in Kategorien unterteilt:
unsicheres Web-Interface : unsicheres Web-Interface erfolgt , wenn das Kennwort schwach ist oder nach Web-Interface Vulnerabilitäten wie (XSS,SQLi und CSRF) XSS : Cross-site scripting ,und hier besteht die Gefahr ,wenn die Webanwendung die Benutzerdaten ohne überprüfung an den Webserver weitersendet. SQL injection : der Angreifer fügt eine Anfrage über ein Web-Formular per Structured Query Language (SQL) (mit Hilfe von SELECT-Abfrage) hinzu, um auf Daten zugreifen können oder Daten zu manipulieren. CSRF : ist ein Angriff, bei dem ein Anwender eine Aktion innerhalb einer Anwendung ausführt ,ohne zu wissen .das kann ein großes Problem verursachen, denn diese Aktion , die der Anwender ausgeführt hat , kann eine Lücke zu dem Zugriff auf seine eigenen Daten verursacht, oder kann eigene persönliche Daten zusenden
Ungenügende Authentifizierung /Autorisierung : wenn der Anwender auf seinem Konto mit Kennwort, oder jemand anderer , der bevollmächtigt ist , einloggen kann , dann der Server authentifiziert den Anwender (wird geprüft , dass Er der richtige Anwender ist , der den recht auf den Zugriff hat[30] .

5 Absicherung der IoT-Protokolle

In diesem Kapitel werden behandelt, wie die einzelnen IoT-Protokolle abgesichert werden Bevor das Thema IT-Sicherheit der IoT-Protokolle erläutert wird , wird hier kurz über wichtiges Thema Schutzziele diskutiert .oder als CIA-Triade kurz zusammengefasst[31].
C :steht für (confidentiality), Vertraulichkeit :die Information ist nur zugänglich für die Befugten.
I :steht für Integrität (integrity),die Information soll vollständig und korrekt für den Befugten zugänglich sein.
A :steht für Availability (Verfügbarkeit) : die Systeme sollen jederzeit betriebsbereit sein und für die Verarbeitung der Daten auch korrekt ablaufen[32]. neben CIA (Vertraulichkeit,Integrität,Verfügbarkeit) gibt es auch die Authentizität , und Authentifizierung.
Bei Authentizität wird die Quelle der Daten bzw. Information , sogar den Kommunikatinspartner geprüft[31]. bei der Authentifizierung wird nach der Identität eines Benutzers im System geprüft,und verifiziert

5.0.1 MQTT-Sicherheit

Bei dem MQTT-Protokoll muss für die Kommunikation zwischen einem Broker eine Authentifizierung unterstützen werden , indem einen Benutzername und Passwort festgelegt wird. Diese erfolgt beim CONNECT-Aufruf. Zudem ermöglischt das TCP-Protokoll die Verschlüsselung der Daten (Schutz vor dem unbefugten Zugriff und vor Manipulation oder Fälschung) über Transport Layer Security (TLS) [33]. außerdem ermöglicht TLS die Prüfung der Identität der Sender sowie der Empfänger im System.

5.0.2 HTTP-Sicherheit

http unterstützt die Basisauthentifizierung über Benutzername und Passwort sowie die Digest Access Authentication. die Daten können bei der authentifizierung direkt bei der Anfrage an den Host hinzugefügt werden: http://username:password@example.com/ Dabei werden beide aber im Klartext übertragen. Um dies zu ermöglichen ,gibt es die Digest Access Authentication. Nachdem den Server angefragt wird, wird eine zufällig Zeichenfolge erstellt wird , die an den Client gesendet werden. Die an gesendeten Zeichenfolge wird zusammen mit Benutzername und Passwort einen HashWert berechnet , der an den Server als Authentifizierung zurückgesendet wird. Da HTTP auch über TCP (Transmission Control Protocol) übertagen wird, kann zusätzlich eine Verschlüsselung über TLS erfolgen[16].

5.0.3 CoAP-Sicherheit

Analog zu http-Protokoll erfolgt bei CoAP auch die Möglichkeit zur Authentifizierung über Benutzername und Passwort. Außerdem kann zur Verschlüsselung der Daten DTLS (Datagramm Transport Layer Security) verwendet werden. DTLS sorgt für den Schutz die privatsphäre in der elektronischen Kommunikationen, die Funktionsweise von DTLS entspricht sehr der von TLS. Außerdem stehen Vier Verschlüsselungsmodi zur Verfügung bei dem CoAP[18].
NoSec
keine Verschlüsselung wird bei diesem Modus benötigt. Die Nachricht wird als coap betrachtet und über UDP verschickt[18].
PreSharedKey
Bei diesem Modus wird das DTLS aktiviert und und aufgelistet , welcher Schlüssel mit welchen Endgeräten benutzt werden kann[39]. Im Extremfall verfügt jedes Endgerät über eigenen Schlüssel. Manchmal ist ein Schlüssel, der für mehr als zwei Endgeräte verwendet wird, nur noch dafür verwendbar, um die Zugehörigkeit des Geräts zu einer Gruppe zu

bestätigen, nicht mehr als einzelner Knotenpunkt[18].
RawPublicKey
Bei diesem Modus wird auch DTLS aktiviert . zum Verschlüsseln wird ein asymmetrisches Verfahren eingesetzt , bei dem wird nicht nut einen einzigen Schlüssel, sondern mit einem Schlüsselpaar benutzt[39],die aus einem öffentlichen und einem privaten Schlüssel besteht . Kein Zertifikat wird für das Schlüsselpaar benötigt[18].
Certificate
Bei diesem Modus wird auch das DTLS aktiviert . zum Verschlüsseln wird ein asymmetrisches Verfahren eingesetzt , bei dem wird nicht nur einen einzigen Schlüssel, sondern mit einem Schlüsselpaar benutzt,die aus einem öffentlichen und einem privaten Schlüssel besteht [39]. Der Unterschied Zwischen diesem Verfahren und RawPublicKey-Moduswird das Schlüsselpaar jedoch von einem „Trust Root“ zertifiziert[18].

5.0.4 XMPP-Sicherheit

XMPP unterstützt eine einfache Authentifizierung mittels SASL (Simple Authentication and Security Layer) ,bei dem es sich um ein Framework handelt, dass es Authentifizierung verschiedener Dienste im Netzwerk unterstützt wird. ,außerdem wird es verwendet , um dem Server die Identität nachzuweisen [34] . Nebenbei kann auch die Verbindung über TSL verschlüsselt ,in diesem Fall bleibt die Kommunikationen zwischen dem Client und Server, dann vom Server zu Server, dann vom Server zum Client verschlüsselt[20].

5.0.5 DDS-Sicherheit

Bei dem DDS-Protokoll werden viele Plugins wie (RTI Connext DDS Secure)eingesetzt ,welches viele Sicherheitsmaßnahmen befolgt , drunter sind Authentifizierung, Verschlüsselung und Zugriffskontrolle[35].

6 Zusammenfassung

dieses Paper befasst sich mit der IoT-Protokolle , es wird fukosiert , welche die Protokolle im Internet der Dinge verwendet werden, außerdem werden sie (Kommunikationsform,Richtung ,Datenformat,die Adressierung ,sowie Nach-und Vorteile der Protokolle) beschrieben.danach wird die die Absicherung der IoT-Protokolle diskutiert ,es wird beobachtet , dass die Entwickler sich nur um Funktionalität der IoT-Protokolle, sowie IoT-Geräte kümmern, ohne die Sicherheit zu beachten.das verursacht Probleme , die Wirtschaft der Länder gefährdet .deswegen war es sinnvoll den tiefen Blick auf die Absicherung der IoT-Protokolle zu werfen , um schwachlücken zu vermeiden , und gegen Angriffe der Hacker zu schützen.

7 Referenzen

Literatur

[1] D. M. Siepermann. (o.J.) Internet der dinge. https://wirtschaftslexikon.gabler.de/definition/internet-der-dinge-53187, abgerufen (10.01.2020)

[2] o.V. (o.J.) Internet der dinge. https://www.gruenderszene.de/lexikon/begriffe/internet-of-things?interstitial, abgerufen (03.01.2020).

[3] Sicari S, Rizzardi A, Grieco LA, Coen-Porisini A (2014) Security, privacy and trust in Internet of Things: the road ahead. Comput Netw 76:146–164

[4] T. H. Volker P. Andelfinger, Internet der Dinge - Technik, Trends und Geschäftsmodelle. Springer, 2015.

[5] "MQTT 5 als Anwendungsprotokoll im Internet der Dinge,Bachelorthesis von Jonas Zimmermann WS 2018/2019 ,Hochschule Offenburg

[6] D. Evans. (2011, apr) The internet of things how the next evolution of the internet is changing everything. https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf, abgerufen(10.01.2020)

[7] I.A.A.C.E.I.T.K.U.M.U.Z.O.Vogel, Software-Architektur(2009), 2nd ed. Heidelberg: Spektrum Akademischer Verlag.

[8] (2016, Oct.) JSON.org. [Online]. http://www.json.org/json-de.html abgerufen (10.01.2020)

[9] W. Huber, Industrie 4.0 in der Automobilproduktion. Wiesbaden: Springer Fachmedien, 2016.

[10] International Business Machines. (2016, Oct.) MQTT 3.1 Protocol Specifications. [Online]. http://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html#qos-flows abgerufen (05.01.2020)

[11] HiveMQ. (2016, Nov.) HiveMQ – Publish Subscribe. [Online]. http://www.hivemq.com/blog/mqtt-essentials-part2-publish-subscribe,abgerufen(05.01.2020)

[12] HiveMQ. (2016, Nov.) HiveMQ – QoS. [Online]. http://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels ,abgerufen(05.01.2020)

[13] HiveMQ. (2016, Nov.) HiveMQ – Topics. [Online]. http://www.hivemq.com/blog/mqtt-essentials-part-5-mqtt-topics-best-practices ,abgerufen(05.01.2020)

[14] HiveMQ. (2016, Nov.) HiveMQ – Messages. [Online].http://www.hivemq.com/blog/mqtt-essentials-part-4-mqtt-publish-subscribe-unsubscribe ,abgerufen(05.01.2020)

[15] D. Obermaier. (2015, Nov.) Informatik aktuell. [Online]. https://www.informatik-aktuell.de/betrieb/netzwerke/iot-protokolldschungel-ein-wegweiser.html .abgerufen(05.01.2020)

[16] J. G. J. M. H. F. L. M. P. L. T. B.-L. R. Fielding. (1999, Jun.) Hypertext Transfer Protocol – HTTP/1.1. [Online]. https://tools.ietf.org/html/rfc2616 ,abgerufen(06.01.2020)

[17] D. Abts, Masterkurs – Client/Server-Programmierung, 4th ed. Wiesbaden: Springer Fachmedien, 2015

[18] K. H. C. B. Z. Shelby. (2014, Jun.) The Constrained Application Protocol (CoAP). [Online]. https://tools.ietf.org/html/rfc7252 ,abgerufen(05.01.2020)

[19] K. H. E. Z. Shelby. (2010, Oct.) Observing Resources in CoAP. [Online]. https://tools.ietf.org/html/draft-ietf-core-observe-00 ,abgerufen(05.01.2020)

[20] P. Saint-Andre. (2011, Mar.) Extensible Messaging and Presence Protocol (XMPP): Core. [Online]. http://xmpp.org/rfcs/rfc6120.html#streams ,abgerufen(06.01.2020)

[21] P. M. R. M. Peter Saint-Andre. (2016, Oct.) XMPP.org – PubSub Extension. [Online]. http://www.xmpp.org/extensions/xep-0060.html ,abgerufen (06.01.2020)

[22] P. Saint-Andre. (2013, Mar.) XMPP.org – XEP0305 Extension Pipelinging. [Online]. http://xmpp.org/extensions/xep-0305.html ,abgerufen (06.01.2020)

[23] J. H. Peter Saint-Andre. (2015, Sep.) XMPP.org – Extension XEP-0033 Extended Stanza Addressing. [Online]. https://xmpp.org/extensions/xep-0033.html,abgerufen (06.01.2020)

[24] A. Forster. (2016, Jul.) Elektroniknet.de. [Online]. http://www.elektroniknet.de/elektronik/embedded/iot-daten-in-echtzeit-132608-Seite-2.html ,abgerufen (07.01.2020)

[25] S. Roth. (2016, Aug.) Informatik Aktuell. [Online]. https://www.informatik-aktuell.de/betrieb/netzwerke/zuverlaessige-datenkommunikation-im-industrial-internet-of-things-mit-dds.html ,abgerufen(07.01.2020)

[26] B.F.R.W.Gerardo Pardo-Castellote. (2005, Aug.) OMG.org-An Introduction to DDS and Data-Centric Communications. [Online].http://www.omg.org/news/whitepapers/Intro_To_DDS.pdf ,abgerufen (07.01.2020).

[27] Schmitz, Peter.(10.12.18) Massive Schwachstellen in IoT-Protokollen [Online] https://www.security-insider.de/massive-schwachstellen-in-iot-protokollen-a-782187/ ,abgerufen (08.01.2020)

[28] Schnabel,Patrik .(MQTT - Message Queue Telemetry Transport),(01.2020) ,[Online] https://www.elektronik-kompendium.de/sites/net/2204051.htm ,abgerufen (10.01.2020)

[29] Schnabel,Patrik,(HTTP - Hypertext Transfer Protocol) ,(01.2020),[Online] https://www.elektronik-kompendium.de/sites/net/0902231.htm ,abgerufen(10.01.2020)

[30] The OWASP IoT-Security Team, 2018 ,[Online],https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project,abgerufen (10.01.2020)

[31] Bedner, M., Ackermann, T. (2010): Schutzziele der IT-Sicherheit. Datenschutz und Datensicherheit-DuD 34(5), 323–328 . SpringerLink.

[32] Eckert, C. (2014): IT-Sicherheit: Konzepte - Verfahren - Protokolle, 9. Aufl. (De Gruyter Studium), München.

[33] HiveMQ. (2016, Nov.) HiveMQ – Security. [Online]. http://www.hivemq.com/blog/mqtt-security-fundamentals-authentication-username-password ,abgerufen(10.01.2020)

[34] E.,.K.Z.E.A. Melnikov. (2006, Jun.) Simple Authentication and Security Layer (SASL). [Online]. https://tools.ietf.org/html/rfc4422 ,abgerufen(10.01.2020)

[35] RTI. (2016, Nov.) RTI – RTI Connext DDS Secure. [Online]. https://www.rti.com/products/secure.html ,abgerufen(10.01.2020)

[36] Graef, M. (2010) ,Instant Messaging Eine Analyse bezüglich Datensicherheit Datenschutz am Beispiel von ICQ und XMPP,HTWK Leipzig.

[37] Christian Baun (3.2019)Computernetze kompakt ,5.Auflage ,Verlag: Springer Vieweg

[38] Ralf Wirdemann,RESTful Go APIs Design und Implementierung leichtgewichtiger Hypermedia Services,03/2019 ,Verlag: Hanser Fachbuchverlag

[39] Johannes Kässinger,Master Thesis :Anwendungsschicht-Protokolle für das Internet der Dinge ,Wintersemester 2016/17 an der Hochschule Offenburg