Sicherheitsanalyse eines Firmware-Updates Prozesses von IoT-Geräten in einem Information Centric Networkk

Svenja Dittmers

01.09.2022
Advanced Internet and IoT Technologies, SoSe 21, Master Informatik, HAW Hamburg, Prof. Dr. Thomas Schmidt

Das Internet der Dinge findet einen immer breiteren Anwendungsbereich, auch in sicherheitskritischen Umgebungen. Um die Geräte vor Schwachstellen zu schützen und auf dem neusten Stand zu halten sind Firmware-Updates unerlässlich. Die Verbreitung dieser findet oft nicht mehr über herkömmliche Netzwerkprotokolle statt, da diese oft nicht den Ansprüchen eines low-power Netzwerkes und der IoT-Geräte gerecht werden. Ein Ansatz der sich als besonders geeignet erwiesen hat ist das Name Data Networking. Doch auch, wenn dieses eingebaute Sicherheitsmechanismen mit sich bringt, ist es nicht vor Angriffen geschützt. In dieser Arbeit soll eine Sicherheitsanalyse des Firmware-Update Prozesses mit NDN durchgeführt werden, um aufzuzeigen, welche Angriffsvektoren es gibt und welche Komponente es zu schützen gilt.

Inhaltsverzeichnis

1 Einleitung

Das Internet der Dinge, oder Internet of Things (IoT) findet immer häufiger Anwendung, nicht nur im heimischen Bereich, sondern auch in der Industrie, sowie in Bereichen, wo Sicherheit eine große Rolle spielt. Der wachsende Anwendungsbereich hat unter anderem mit der besseren Verfügbarkeit von Hardware, Technologien die geringe Leistungen unterstützen und Echtzeit-Systemen, zu tun. Trotz unterschiedlicher Hersteller, Hardware und Frameworks unterscheiden sich die Geräte konzeptuell nicht groß voneinander. Jedoch weichen IoT-Geräte von Standard Anwendungen ab, da sie durch ihre Vielzahl an Komponenten Komplexität mit sich bringen [1].
Um die Geräte gegen Schwachstellen abzusichern, ist es wichtig sie auf dem aktuellsten Stand zu halten. Der Prozess der Firmware Aktualisierung bei IoT-Geräten ist nicht immer trivial. Geräte können an schlecht zu erreichenden Gegenden zum Einsatz kommen. Eine manuelle Aktualisierung ist daher oft nicht möglich. Um so wichtiger ist es einen Update-Prozess zu gewährleisten, der auf mögliche Schwachstellen hin untersucht wurde.
In dieser Arbeit soll eine Sicherheitsanalyse für den Prozess eines Firmware-Updates für IoT-Geräte in einem Information Centric Network (ICN) durchgeführt werden. Da erste Kapitel befasst sich mit dem Hintergrund des Prozesses selbst. Es wird auf den allgemeinen Update-Prozess der Firmware auf IoT-Geräten eingegangen und im Anschluss die Besonderheiten von Named-Data Networking (NDN) in diesem Zusammenhang erklärt. Darauf aufbauend wird ein Update-Prozess beschrieben, worauf die Sicherheitsanalyse aufbaut.

2 Hintergrund

Ein gut funktionierender und abgesicherter Update-Prozess behebt nicht nur Schwachstellen, er unterstützt auch Konfigurationsupdates und implementiert neue Funktionalitäten. Er ist ein wichtiger Bestandteil im Lifecycle eines IoT-Gerätes [1]. Insbesondere, wenn diese eine lange Lebensdauer haben, oder in unzugänglichen Gebieten eingesetzt werden, wo es nicht möglich ist die Geräte manuell auf dem neusten Stand zu halten. Ein automatischer und gut abgesicherter Update-Prozess ist daher unumgänglich. Und um diesen Prozess entsprechend abzusichern, ist es um so wichtiger eine Sicherheitsanalyse durchzuführen um jegliche Schwachstellen im Prozess selbst auszuschließen und mit entsprechenden Gegenmaßnahmen entgegen zu wirken. Um eine zielführende Sicherheitsanalyse für den Prozess des Firmware-Updates eines IoT-Gerätes durchführen zu können, gilt es daher den Prozess und die Architektur dahinter zu verstehen und die Besonderheit im Zusammenhang mit NDN zu erläutern.

2.1 Firmware Updates eines IoT-Gerätes (SUIT-Modell)

SUIT[2] ist eine von der IETF entworfene Architektur, welches einen standardisierten, zuverlässigen und sicheren Update-Mechanismus für eingeschränkte IoT-Netzwerke darstellen soll. Eine wichtige Komponenten neben dem Firmware-Image selbst, ist die Manifest-Datei, die maschinell verarbeitbare Metadaten enthält, sowie Informationen zu den Firmware-Updates. Geräte benötigen Mechanismen um Firmware Server zu erkennen, sowie Protokolle um die Firmware-Images auszuliefern. Diese unterliegen oft bestimmten Standards.
Updates müssen robust installiert werden um die Gerätefunktionalität während des Updates nicht zu beeinträchtigen. Tests und Wiederherstellungsstrategien helfen, falls die Aktualisierung nicht erfolgreich durchgeführt werden kann. Da Updates ein komplexer Entscheidungsprozess sind, sollte die Bereitstellung von Updates rechtzeitig erfolgen. Ebenso sollte energieeffizient gearbeitet werden, denn die Funk-Kommunikation und das Schreiben des neuen Images auf den Flash-Speicher ist eine energieintensive Aufgabe[3].
Der folgende Abschnitt beschreibt das SUIT-Modell näher, um die wichtigsten Mechanismen aufzuzeigen.

2.1.1 Architektur

IoT-Geräte sind oft über das Internet miteinander verbunden. Dies führt dazu, dass Updates ebenfalls darüber bereit gestellt werden müssen. Folgende Komponenten sind für ein Update auf dem Gerät notwendig:

- Internet-Protokoll-Stack welcher für Flusskontrolle, Fragmentierung und Wiederzusammensetzung, sowie Mechanismen zur Wiederaufnahme bei Unterbrechung oder Beschädigung der Übertragung sorgt.
- Die Fähigkeit das Firmware Image in einen dauerhaften Speicher, wie etwa der Flash-Speicher, zu schreiben.
- Die Möglichkeit die digitale Signatur zu überprüfen.
- Die Fähigkeit das Image zu entpacken, dekomprimieren und zu entschlüsseln.
- Einen Status-Tracker der Mitteilungen über neue Firmware-Versionen, Informationen über Software- und Hardware-Eigenschaften liefert, sowie über den Aktualisierungsprozess der Firmware.

Der Status-Tracker wird dabei in eine Client- und eine Server-Seite unterteilt. Dabei informiert die Server-Seite den Client über eine neue Firmware Version.

- Client-initiierte Aktualisierung (Polling): Es wird proaktiv vom Client nach Aktualisierungen gesucht.

- Server-initiierte Aktualisierung (Push): Es wird durch den Server bestimmt, welche Geräte für eine Aktualisierung in Frage kommen und informiert diese.
- Hybride Aktualisierung: Der Server sendet eine Benachrichtigung über die Verfügbarkeit eines Updates und forder den Client auf, sich das Manifest und das Image zu ziehen.

Geräte unterstützen eine Vielzahl an Technologien (CoAP, MQTT, HTTP, WiFi, BLE, low-power WAN). In bestimmten Fällen kann es auch sinnvoll sein, dass die Images via Broadcast oder Multicast verteilt werden.

2.1.2 Manifest

Das Manifest ist eine Datei die verschiedene Informationen, sowie Metadaten zu Abhängigkeiten und Komponenten, Sequenznummern und Versionsnummern enthält. Auch Befehlssequenzen die Anweisungen zur Installation eines Images liefern sind enthalten.
Neben dem Schutz, durch Authentifizierung und Integritätsschutz des Firmware-Images, dient es auch zum Schutz anderer Konfigurationsdateien. Asymmetrische Kryptografie verhindert das böswillig veränderte Images auf ein Gerät geschrieben werden können.

2.1.3 Aufruf der Firmware

Hat ein IoT-Gerät das Manifest und Firmware-Image erhalten, müssen diese verarbeitet werden. Das beinhaltet meist ein Anhalten der aktuell laufenden Version, sowie die Verifizierung der Firmware, Überprüfung der Authentizität und Integrität und den Aufruf der neuen Firmware selbst. Letzteres ist dabei besonders sicherheitsrelevant, da ein Angreifer Reverse Engineering betreiben kann und versuchen könnte ein manipuliertes Images zu laden. Daher muss die Verifizierung des Images einige Sicherheitsprüfungen durchlaufen. Zum Einen den Firmware-Verifizierer, der das zu bootende Image verifiziert. Zum Anderen bedarf es einer Wiederherstellungsstrategie, die im Fall eines fehlgeschlagenen Prozesses greift.

2.1.4 Absicherung des Firmware-Updates

Den Aktualisierungsprozess der Firmware gilt es abzusichern, um Angreifer vor der Kontrollübernahme des Gerätes abzuhalten. Dazu werden Ende-zu-Ende Verschlüsselungsmechanismen eingesetzt. Die Überprüfung des Images sowie des Manifests beinhalten folgende Punkte:

1. Authentifizierung: Das Gerät kann den/die Autor(en) des Images und Manifests kryptografisch identifizieren.
2. Integrität: Sichert das Image vor fremder und oder böswilliger Veränderung ab. Dabei wird die Signatur des Manifests geprüft.

3. Vertraulichkeit: Niemand außer dem/den vorgesehenen Geräte dürfen das Firmware-Image entschlüsseln. Dazu werden Zertifikate eingesetzt.

Der Autor eines Firmware-Images kann aus Sicherheitsgründen das Image selber nicht auf dem Firmware-Server bereitstellen. Stattdessen stellt der Autor das Image den Gerätetreibern zur Verfügung.

2.2 Named Data Networking

Die Verwendung von IP und ähnlichen Protokollen erweist sich für das Internet der Dinge als nicht angemessen. Eine vorgeschlagene Internetarchitektur, welche die Herausforderungen die IoT mit sich bringen angehen soll, ist das Named Data Networking (NDN)[4]. Anstatt Empfänger durch IP-Adressen zu identifizieren, ermöglicht NDN es die Daten über Namen auf der Anwendungsschicht anzufordern. Durch die Benennung der Daten wird es ermöglicht, diese direkt auf der Netzwerkschicht zu sichern. Erreicht wird dies darüber, dass der Inhalt jedes Datenpakets überprüfbar und vertraulich gemacht wird [5].
Möchte ein Konsument ein Datenpaket erhalten, sendet er ein "Interest-Paket" welches den Namen oder Namenspräfixe enthält und das NDN Netzwerk nutzt den Namen um die angefragten Daten abzurufen. Diese Namen folgen einer hierarchischen Struktur. Dabei können sie den Umfang der Daten definieren ("/Some-Place"), anwendungsspezifische Semantiken beschreiben (".../Temperatur") oder eindeutige Bezeichner für Instanzen oder Versionen liefern (".../1234567890"). Dies ermöglicht IoT-Anwendungen die NDN basiert sind, mit Paketen zu arbeiten, die Dinge und ihre Daten beschreiben. Wird ein Interest-Paket angefordert, prüft ein Forwarder in seinem Content-Store Daten, welche alle empfangenen Datenpakete hält, die entweder mit dem Interest-Namen oder den Namen als Präfix übereinstimmen. Findet sich keine Übereinstimmung prüft der Forwarder seine Pending Interest Tabelle (PIT). Diese enthält kürzlich empfangene Interest-Pakete und deren Schnittstellen. Die Forwarding Information Base (FIB) enthält alle Namenspräfixe, sowie Interfaces zu denen Interest-Pakete weitergeleitet werden sollen. [6]. NDN

2.3 Sicherheitsanalyse

Die Sicherheitsanalyse beschreibt ein bestimmtes Vorgehensmodell, welches zur Bewertung und zur Optimierung der Sicherheit in einem zu überprüfenden IT-System eingesetzt wird. Sicherheit beschreibt hierbei die Eigenschaft eines Systems, schützenswerte Güter durch besondere Maßnahmen vor als bedeutsam angesehenen Bedrohungen, soweit zu minimieren, dass das verbleibende Risiko akzeptabel bleibt.
Die Aufgabe der IT-Sicherheitsanalyse besteht darin, systematische Identifikation und

Bewertungen von Bedrohungen vorzunehmen, sowie anschließend Sicherheitsanforderungen zu bestimmen und Sicherheitsmaßnahmen ableiten, welche in das IT-System integriert werden. Sie lässt sich in einen mehrstufigen Prozess unterteilen. (i) Den momentane Zustand eines IT-Systems hinsichtlich seiner Sicherheit analysieren. Das bedeutet alle relevanten Komponente identifizieren, die zu schützenden Güter erfassen und das System auf Schwachstellen analysieren. (ii) Schwachstellen, welche zu potenziellen Bedrohungen führen, identifizieren und Gegenüberstellung mit Sicherheitsanforderungen. Das beinhaltet die Identifikation von Bedrohungen, die zu schützenden Güter und deren Schwachstellen mit den zuvor identifizierten Bedrohungen verknüpfen und das Risiko eben jener zu analysieren, sowie Sicherheitsanforderungen zu analysieren. (iii) Sicherheitsmaßnahmen entwickeln. Die bereits bestehenden Sicherheitsmaßnahmen gilt es zu identifizieren und zu analysieren und anschließend Pläne zur Optimierung der Sicherheit konzipieren. (iv) Maßnahmen integrieren. Anschließend die konzipierten Sicherheitsmaßnahmen implementieren [7].

3 Firmware Update Prozess

Der zu analysierende Firmware Update Prozess [8] stellt ein sicheres und zuverlässiges System für Firmwarerollouts da, welches die Bereitstellung von Software-Updates an zahlreiche IoT-Geräte unter der Verwendung von Named-data Networking (NDN) durchführt. NDN zeigte in Studien eine gewisse Überlegenheit des Datentransfers in eingeschränkten und low-power Netzwerken, gegenüber CoAP und "message queuing telemetry transport for sensor networks" (MQTT-SN). Bei dem Update Prozess wird das oben beschriebene SUIT Modell [2] als Vorlage verwendet. Figure 1 beschreibt den Aufbau, der aus drei Komponenten besteht: Veröffentlichung von Firmware-Images und Manifest-Dateien. Verwaltung und Speicherung der Software-Updates, sowie der Zugriff auf die IoT-Einsatzorte. Und als letztes die rechtzeitige Benachrichtigung über Updates und Auslieferung dieser an die IoT-Geräte.

3.1 Vorbereitung und Veröffentlichung

Auf Grund der Tatsache, dass Geräte von verschiedenen Herstellern stammen können, benötigt es Interoperabilität um ein energieeffizientes System zu erstellen. Interaktionen zwischen Anbieter und IoT-Geräte werden dabei durch ein systematisches Namespace-Management geregelt. Dabei besitzt jeder Einsatzort eines IoT-Gerätes einen einzigartigen globalen Namen. Dieser setzt sich aus dem Namen des Einsatzortes, dem Anbieter, einer Geräteklasse sowie eines Zeitstempels zusammen z.B. /SomePlace-3/IoTFirma/Geräteklasse/123456789. Zum Schutz der Integrität des Images wird vom Hersteller ein Hashwert erzeugt.
Auf Grund kleiner MTU-Größen ist die Fragmentierung des Firmware-Images auf Anbieterseite und die Wiederzusammensetzung auf Seite des IoT-Gerätes erforderlich. Dabei muss die Wiederzusammensetzung der Datenblöcke so simpel wie möglich sein.

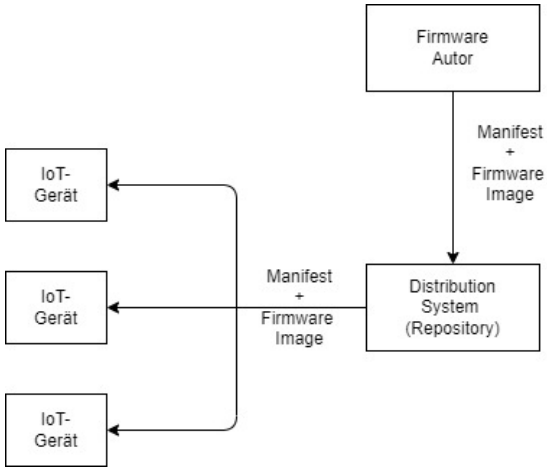

Figure 1: Aufbau des Firmware Update Prozess

Daher wird der Ansatz einer linearen Aufteilung verfolgt. Dabei können die Größen der Blöcke variieren. Jeder Datenblock erhält eine ID.
Die Manifest-Datei wird nach dem SUIT Modell erstellt und enthält die Binarygröße, Hashwert, sowie Parameter für den Datenblock-Algorithmus. Für die Authentizität wird diese vom Hersteller signiert. Die Manifest-Datei und das Firmware-Image werden dann an ein Firmware-Repository ausgeliefert. Dieses verwaltet die versionierten Dateien der Hersteller.

3.2 Update Prozess

Das Rollout-Konzept ist so designt, dass immer die vollständige Binärdatei ausgeliefert wird. Die Versionsvermittlung beruht auf dem Pull-Ansatz, bei dem das IoT-Gerät in regelmäßigen Abständen eine Manifest-Datei eines bestimmten Zeitfenster's anfordert. Der dabei angeforderte Zeitrahmen muss dabei größer sein, als der der lokal laufenden Firmware. Das Firmware-Repository liefert die Datei, sofern das angeforderte Update erhältlich ist. Sollte ein Gerät länger nicht verbunden sein und mehrere Versionen hinterher sein, muss nicht versucht werden veraltete Versionen abzurufen. Um die Anzahl der einzelnen Anfragen zu reduzieren wird ein Versionserkennungsprozess verwendet. Sobald ein Rand-Knoten ein Versionsupgrade feststellt, bereitet es den Abruf des Binaryimages vor. Dabei wird zunächst die Signatur der Manifest-Datei validiert. Bei einer fehlgeschlagenen Prüfung bricht der Prozess ab. Bei einer gültigen Signatur werden alle im Manifest angegebenen Firmware-Teile ausgelöst. Dabei wird jeder Block des Images mit seiner ID adressiert. Diese beginnt bei 0 und steigt bis zur im Manifest festgelegten maximalen Nummer. Um Speicher- und Netzressourcen zu sparen wird automatic repeat-request (ARQ) zur Fehlerkontrolle genutzt. Für den Prozess werden zwei Strategien zum Abrufen von Blöcken definiert. Die erste Methode erlaubt eine gleichzeitige Aktualisierung der Firmware von Knoten die sich auf dem selben

Pfad befinden. Die zweite Methode erlaubt eine geordnete Aktualisierung, die sich kaskadenartig im Netz ausbreitet.
Nach erfolgreichem Abruf befinden sich alle Datenblöcke, und damit das vollständige Image, im Chunk-Buffer. Ein Knoten erstellt einen Hashwert über den Buffer und vergleicht ihn mit dem zuvor empfangenen Firmware Hashwert. Bei positiver Prüfung wird das Image in den Flash-Bereich kopiert und der Bootloader wird benachrichtigt die neue Firmware aufzurufen. Nach abgeschlossener Aktualisierung kann ein Gerät das neuste Manifest und Datenblöcke an unterliegende Geräte weiterleiten. Der Vorteil besteht darin, dass die Blöcke direkt aus dem read-only Flash bedient werden und damit keinen Hauptspeicher verbrauchen. Die Firmware kann daher ohne Firmware-Server, bei z.B. einem Verbindungsabbruch, weiter in das IoT-Netzwerk gegeben werden.
Eine kryptografische Überprüfung jedes einzelnen Datenblock's ist nicht möglich. Die Integrität und Authentizität des Images wird erst nach Empfang aller Blöcke geprüft. Daher wird eine Verifizierung bei Empfang mit einem verschlüsselten Hash Message Authentication Code (HMAC[5]) durchgeführt. Der Block wird erst nach korrekter Verifizierung in den Chunk-Buffer aufgenommen.

4 Methodik

Eine Sicherheitsanalyse kann auf verschiedene Weisen durchgeführt werden. Eine weit verbreitete Methode ist die, der Bedrohungsanalyse. Dabei werden Schwachstellen identifiziert die ein Angreifer ausnutzen kann. Im folgenden wird auf die Durchführung einer Bedrohungsanalyse eingegangen und beschrieben wie dies für den Prozess des Firmware Updates von IoT Geräten in einem multi-hop Netzwerk umgesetzt werden kann.

4.1 Threat Modeling

Die Bedrohungsanalyse wird mit Hilfe des STRIDE Ansatzes[9] durchgeführt. STRIDE steht dabei für:

- **S**poofing: Illegale Zugriff auf Authentifizierungsdaten eines Benutzers und deren anschließende Verwendung.
- **T**ampering: Böswillige Manipulation von Daten
- **R**epudiation: Ablehnung einer durchgeführten Aktion von einem Benutzer.
- **I**nformation Disclosure: Offenlegen von Informationen für Benutzer, die normalerweise keinen Zugang zu diesen haben.
- **D**enial of Service: Verweigerung eines Dienstes. Entweder weil er nicht verfügbar ist oder nicht zu erreichen ist

- **E**levation of Privilege: Aneignung von privilegiertem Zugang eines nicht privilegierten Nutzer.

Dieser Ansatz deckt systematisch Bedrohungen für jede Systemkomponente auf und Berücksichtigt dabei die erforderlichen Sicherheitseigenschaften wie, Authentifizierung, Autorisierung, Vertraulichkeit, Integrität und Nichtabstreitbarkeit. Des weiteren bietet STRIDE einen klaren Überblick über die Konsequenzen der Schwachstellen jeder Komponente und deren Auswirkungen auf das gesamte System.
Der Prozess der Firmware-Aktualisierung muss zum einen vor Rollback-Angriffen geschützt werden, sodass ein Gerät keine veraltete Firmware mit bekannten Schwachstellen installieren kann. Außerdem muss der Prozess vor Angriffen durch Nachahmung, wie etwa Spoofing, Man-in-the-Middle oder Masquerade, geschützt werden. Der Angreifer kann dadurch den Datenverkehr umleiten oder bösartige Daten in einen unsicheren Kanal leiten.

5 Sicherheitsanalyse

Im folgenden Abschnitt wird eine Sicherheitsanalyse von dem Prozess des Firmware Updates von IoT Geräten in einem NDN-Netzwerk durchgeführt. Dabei wird das im vorherigen Kapitel beschriebene STRIDE-Modell zur Bedrohungsanalyse verwendet. Wobei jede Bedrohung entsprechend klassifiziert wird und die mögliche Angriffsfläche genannt wird. Aufbauend auf der Bedrohungsanalyse werden Sicherheitsanforderungen formuliert.
Die Sicherheitsanalyse beschränkt sich dabei nur auf den Prozess des Firmware-Updates, sowie auf die Kommunikation innerhalb des NDN-Netzwerkes.

5.1 Sicherheitsanalyse Firmware Update

5.1.1 Firmware und Manifest

Veraltete Firmware

Klassifizierung Elevation of Privilege

Beschreibung Ein veraltetes, aber valides Firmware Image, sowie Manifest wird an das Gerät geliefert. Enthält die veraltete Firmware bekannte Sicherheitslücken, so kann der Angreifer diese ausnutzen und unter Umständen Kontrolle über das Gerät erhalten.

Angriffsfläche Firmware Repository, Edge-Knoten, das IoT-Gerät selbst.

Nicht übereinstimmendes Firmware-Image

Klassifizierung Denial of Service

Beschreibung Ein gültiges, signiertes Firmware-Image wird an einen falschen Gerätetypen gesendet. Die Firmware wird von dem Gerät positiv verifiziert, da sie von einem Akteur mit der entsprechenden Berechtigung signiert wurde. Unterscheiden sich bei beiden Gerätetypen stark voneinander kann das zur Funktionsfähigkeit des Gerätes führen, oder weitere Schwachstellen offen legen.

Angriffsfläche Edge-Knoten, das IoT-Gerät selbst.

Nicht authentifiziertes Firmware-Images

Klassifizierung Elevation of Privilege

Beschreibung Ein Angreifer kann seine eigene Firmware auf dem Gerät installieren, in dem er die Nutzdaten oder Metadaten manipuliert, kann er vollständige Kontrolle über das Gerät erlangen.

Angriffsfläche Firmware Repository, Edge-Knoten, das IoT-Gerät selbst.

Modifikation des Manifests oder des Images vor der Signierung

Klassifizierung Elevation of Privilege

Beschreibung Gelingt es dem Angreifer vor der Signatur das Manifest oder das Image zu verändern, kann er alle Aktionen als Manifest Autor ausführen. Zum Beispiel durch bösartige Software auf dem Computer eines Entwicklers oder beim Signaturservice.

Angriffsfläche Firmware Repository, Manifest

Exposition vertraulicher Manifest-Daten

Klassifizierung Information Disclosure

Beschreibung Ein Angreifer kann sensible Informationen aus dem Manifest entnehmen.

Angriffsfläche Firmware Repository, Manifest

Kritische Manifest-Elemente überschreiben

Klassifizierung Elevation of Privilege

Beschreibung Ein autorisierter Akteur verwendet einen Überschreibungsmechanismus um ein Element im signierten Manifest zu verändern. Wenn der Akteur den Digest und den URI der Payload überschreibt, kann die gesamte Nutzdatenmenge durch eine seiner Wahl ersetzt werden.

Angriffsfläche Manifest

5.1.2 NDN Architektur

Ein Merkmal von NDN ist, dass Sicherheit und Datenschutz durch Mechanismen von Signaturen und Kryptografie direkt in die Architektur integriert sind und Daten somit vor dem Abfangen von Angreifern schützen. Ein weiteres wichtiges Merkmal von NDN ist das Caching. Jedoch ist diese Architektur anfällig für Angriffe wie der Missbrauch des Content Caching Prozesses, Störung der Erreichbarkeit von Daten, oder Privatsphäre von Daten manipulieren [10].

Cache Privacy Attack

Klassifizierung Information Disclosure

Beschreibung Ein Angreifer versucht den im Cache vorhandenen Content zu ermitteln und welcher User den Content angefordert hat.

Angriffsfläche Cache, Content-Store, Edge Knoten

Content Poisoning Attack

Klassifizierung Denial of Service

Beschreibung Ein Angreifer, der als bösartiger Router oder Produzent auftritt, verbreitet beschädigte oder bösartige Daten an seine Nachbarn. Das führt dazu, dass der Platz im Cache der Nachbar-Routern für den bösartigen Content reserviert wird. Wird dieser Content angefragt, läuft er auch durch andere Router und belegt den Cache. Die Router können auf Grund begrenzter Ressourcen nicht die Gültigkeit der böswilligen Daten prüfen. Dadurch wird eine Verzögerung beim Abruf von legitimen Content erzeugt.

Angriffsfläche Edge Knoten

Interest Flooding Attack

Klassifizierung Denial of Service

Beschreibung Ein Angreifer versucht eine große Menge an Interessen an die Zielknoten zu senden und somit die Ressourcen der Router zu erschöpfen und die Einträge im PIT vom Angreifer reserviert werden. So kann ein legitimer Nutzer keine Einträge mehr reservieren.

Angriffsfläche Edge Knoten

Cache Pollution Attack

Klassifizierung Denial of Service

Beschreibung Ein Angreifer versucht den Cache im Content-Store zu leeren und hindert damit andere Nutzer daran, ihren gewünschten Content aus dem Cache zu erhalten. Es wird böswilliger Content angefordert um Platz im Cache zu reservieren. Dieser wird immer wieder angefordert um den gesamten Platz im Cache zu verbrauchen.

Angriffsfläche Cache, Edge Knoten

5.2 Sicherheitsanforderungen

Den Update-Prozess abzusichern ist genau so wichtig, wie der Firmware-Update Prozess selbst. Mit einem Update wird immer die Ausführung von Code via remote Zugriff erlaubt, welche bei Schwachstellen einem Angreifer viele Möglichkeiten bietet.
Authentifizierung ermöglicht es dem Gerät kryptografisch Autoren der Firmware-Images und Manifest Dateien zu identifizieren. Daher muss ein Gerät sicherstellen können, dass die angeforderte Aktion auch wirklich von einem Akteur mit Berechtigung ausgeführt wird. Des Weiteren gilt es den Schutz der Integrität aufrecht zu erhalten um sicherzustellen, dass kein Dritter das Manifest oder das Image verändern kann. Die IoT-Geräte müssen in Besitzt eines Trust-Anchors sein um Signaturen überprüfen zu können. Der Schutz der Vertraulichkeit muss ebenso erfolgen. Niemand außer dem Firmware Nutzer und andere autorisierte Parteien sollten in der Lage sein ein Image zu entschlüsseln. Dazu muss der Autor des Images im Besitz eines Zertifikates oder Schlüssels eines Gerätes sein. [3]
Einige weitere Sicherheitsmechanismen erfolgen durch das SUIT-Modell. Dazu gehören Hersteller und Geräte-Typ Identifikatoren, welche dafür sorgen, dass Geräte nur die vorgesehene Firmware installieren. Sie müssen wissen, dass ein Update für ihr Modell, ihre Hardware - und Softwareversion gilt. Ein weiterer Mechanismus ist die Verfallszeit, welche angibt wann ein Manifest ausläuft. Ebenso wie ein authentifizierter Speicherort, welcher den Ort auf dem Gerät angibt, an dem die Nutzlast gespeichert werden soll. Dieser muss authentifiziert werden. Statusmeldungen eines Gerätes müssen über einen authentifizierten und vertraulichen Kanal erfolgen um Änderungen oder Verfälschung von Meldungen zu verhindern.
Ein weiterer Punkt ist die geschützte Speicherung von Signierschlüsseln. Die kryptografischen Schlüssel für die Unterzeichnung von Manifesten sollten regelmäßig ersetzt werden und so gespeichert werden, dass sie für vernetzte Geräte unzugänglich sind.
Neben den oben genannten Sicherheitsmechanismen gibt es noch einige weitere welche es zu beachten gibt. Diese finden sich alle im SUIT-Modell. [2]

6 Fazit

Durch die wachsenden Einsatzgebiete von IoT-Geräten, auch in sicherheitskritischen Bereichen, ist es wichtig diese gegen Schwachstellen abzusichern. Oft erfolgt diese Absicherung durch ein Firmware-Update. Diese bringen neben der Absicherung von Schwachstellen, oft auch andere neue Konfigurationen mit sich. Dieser Prozess des Firmware-Updates gilt es ebenfalls gegen mögliche Angreifer abzusichern, denn dabei wird etwaiger Code oft remote ausgeführt und bietet eine Angriffsfläche.
Bei dem Firmware-Update von IoT-Geräten spielt das SUIT Modell der IETF eine wichtige Rolle. Dabei wird deutlich, dass der Prozess selbst aus vielen zu sichernden Komponenten besteht. Ebenfalls wurde deutlich, dass NDN sich als sehr nützlich für diesen Zweck erweist und trotz eingebauter Sicherheitsmechanismen dennoch Angriffsflächen bietet. Trotz der erarbeiteten Sicherheitsanforderungen zu den erarbeiteten Bedrohungen muss der Firmware-Update Prozess stets auf neue Bedrohungen überprüft werden.

References

[1] Koen Zandberg et al. "Secure Firmware Updates for Constrained IoT Devices Using Open Standards: A Reality Check". In: *IEEE Access* 7 (2019), pp. 71907–71920. DOI: 10.1109/ACCESS.2019.2919760.

[2] Brendan Moran, Hannes Tschofenig, and Henk Birkholz. *A Manifest Information Model for Firmware Updates in IoT Devices*. Internet-Draft. Work in Progress. URL: https://datatracker.ietf.org/doc/html/draft-ietf-suit-information-model-11.

[3] Brendan Moran et al. *A Firmware Update Architecture for Internet of Things*. RFC 9019. Apr. 2021. DOI: 10.17487/RFC9019. URL: https://www.rfc-editor.org/info/rfc9019.

[4] Van Jacobson et al. "Networking Named Content". In: *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*. CoNEXT '09. Rome, Italy: Association for Computing Machinery, 2009, pp. 1–12. ISBN: 9781605586366. DOI: 10.1145/1658939.1658941. URL: https://doi.org/10.1145/1658939.1658941.

[5] Dr. Hugo Krawczyk, Mihir Bellare, and Ran Canetti. *HMAC: Keyed-Hashing for Message Authentication*. RFC 2104. Feb. 1997. DOI: 10.17487/RFC2104. URL: https://www.rfc-editor.org/info/rfc2104.

[6] Lixia Zhang et al. "Named Data Networking". In: *SIGCOMM Comput. Commun. Rev.* 44.3 (2014), pp. 66–73.

[7] Rüdiger Grimm et al. *Referenzmodell für ein Vorgehen bei der IT-Sicherheitsanalyse*. June 2014. DOI: 10.1007/s00287-014-0807-3.

[8] Cenk Gündogan et al. "Reliable Firmware Updates for the Information-Centric Internet of Things". In: *Proc. of 8th ACM Conference on Information-Centric Networking (ICN)*. Virtual: ACM, Sept. 2021, pp. 59–70. URL: https://doi.org/10.1145/3460417.3482974.

[9] Rafiullah Khan et al. "STRIDE-based threat modeling for cyber-physical systems". In: *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*. 2017, pp. 1–6. DOI: 10.1109/ISGTEurope.2017.8260283.

[10] Abdelhak Hidouri et al. "Cache Pollution Attacks in the NDN Architecture: Impact and Analysis". In: *2021 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. 2021, pp. 1–6. DOI: 10.23919/SoftCOM52868.2021.9559049.