Data Leakage Protection für Gebäude

Abschlussarbeit im Studiengang Bachelor of Science Informatik
an der Fakultät für Mathematik und Informatik
der FernUniversität in Hagen

von

Eva Maria Anhaus
Matrikelnummer 7694687

Prüfer: Prof. Dr. Jörg Keller
Dr. Steffen Wendzel, Fraunhofer FKIE

Danksagung

An dieser Stelle möchte ich mich bei all jenen bedanken, die mich beim Entstehen der vorliegenden Arbeit und in den letzten Jahren unterstützt und begleitet haben:

Ein besonderer Dank gilt den Betreuern Herr Prof. Dr. Jörg Keller von der Fern-Universität in Hagen und Herr Dr. Steffen Wendzel vom Fraunhofer FKIE, die mit viel Engagement die Anfertigung dieser Arbeit verfolgt haben.

Joel J. Bender, der mir bei der Bearbeitung des praktischen Teils mit Rat und Tat zur Seite gestanden hat und seine Implementierung des BFR nähergebracht hat.

Anja, die mit viel Geduld und Verzicht die Zeit des Studiums mit mir durchgestanden und immer unterstützend und mit aufbauenden Worten zur Seite gestanden hat. DANKE!

Meinen Freunden, die mich durch ihre positiven Worte motiviert oder durch negative Worte erst recht motiviert haben.

Mauro B. für seinen kreativen Beitrag.

Abbildungsverzeichnis

Abkürzungsverzeichnis

APCI	Application Protocol Control Information
APDU	Application layer protocol data unit
ASHRAE	American Society of Heating, Refrigerating and Air-Conditioning Engineers
BBMD	BACnet Broadcast Management Device
BDT	Broadcast Distribution Table
BFR	BACnet Firewall Router
BVLC	BACnet Virtual Link Control
BVLL	BACnet Virtual Link Layer
DDC	Direct Digital Control
DLP	Data Leakage Protection
FDT	Foreign Device Table
IEEE	Institute of Electrical and Electronics Engineers
ICI	Interface Control Information
ISO	International Organization for Standardization
MSB	Most Significant Bit
NAT	Network Address Translation
NIC	Network Interface Controller
NPDU	Network layer protocol data unit
PDU	Protocol Data Unit
TSM	Transaction State Machine
TTL	Time To Live
WLAN	Wireless Local Area Network

1 Einleitung

1.1 Ausgangssituation und Problemstellung

Der Bereich Gebäudeautomation hat sich in den letzten Jahren zu einem stark wachsenden Markt entwickelt. Längst wurden die Vorzüge der Gebäudeautomation auch für viele andere als die ‚Ursprungsbereiche' Heizungs-, Kühlungs-, Lüftungs- und Klimaanlagenbau entdeckt. Unter dem Sammelbegriff „Ambient Assisted Living" wird der Automationsbereich zur Unterstützung von Menschen mit körperlichen Gebrechen zusammengefasst, das Schlagwort „Hausautomation" als Unterbereich der Gebäudeautomation ist seit Jahren in aller Munde. Der Fokus der Entwicklung lag – und liegt nach wie vor – auf der Optimierung der Funktionalität. Allerdings gewinnen Sicherheitsaspekte immer mehr an Bedeutung. Angreifer haben erkannt, dass das Ausnutzen von Sicherheitslücken und die Aneignung von sensiblen oder personenbezogenen Daten wirtschaftlich verwertbar sind. Doch auch die Privatsphäre kann durch den Abfluss von Daten aus dem Netzwerk verletzt werden, wenn es sich um personenbezogene, sensible Daten handelt. Erleichtert wird dieser Abfluss dadurch, dass die Automationsnetze immer häufiger über eine Anbindung an öffentliche Netzwerke (Internet) verfügen. Der Preis, der für diese unmittelbare und allverfügbare Zugänglichkeit der Daten bezahlt wird, ist das Risiko, dass diese Zugänglichkeit auch für unerwünschte Zwecke missbraucht wird. Doch nicht nur die geographische, sondern auch die protokollmäßige Öffnung der Gebäudenetzwerke kann den Zugang zu sensiblen Daten erleichtern. Während es bei proprietären Protokollen schwierig sein kann, die Daten zu interpretieren und an Bereiche außerhalb des Automationsnetzes weiterzugeben, so ist dies bei der Verwendung von (standardisierten) Basis-Protokollen aus den IT-Netzwerken ungleich einfacher.

Angriffen von außerhalb kann – vorausgesetzt dies ist der einzige Einwahlpunkt in ein Netzwerk - durch den Einsatz einer konventionellen Firewall durchaus entgegengewirkt werden. Was aber, wenn der „Feind" in den eigenen Reihen sitzt und Informationen wissentlich und gewollt von Mitarbeitern oder anderen Personen, die Zugriff auf die Daten haben, weitergereicht werden?

In der konventionellen IT haben sich auch für die internen Sicherheitsbedürfnisse Firewalls bewährt. Diese bieten Möglichkeiten, den Zugriff auf bestimmte Teilnetz-

werke des Betriebes zentral einzuschränken. Zum jetzigen Zeitpunkt (2014) gibt es in der Gebäudeautomation noch kein Werkzeug, welches im produktiven Einsatz etwas Ähnliches leisten kann.

1.2 Ziel der Arbeit

Diese Arbeit befasst sich mit dem Thema „Data Leakage Protection für Gebäude“. In der Literatur wurde der Sicherheitsaspekt der Gebäudeautomation bisher wenig beleuchtet. Auf den folgenden Seiten wird nun aufzeigt, welche Komponenten in einem Gebäudenetz dem Risiko eines unerwünschten Datenabflusses ausgesetzt sind und wie dieses Risiko minimiert bzw. verhindert werden kann.

1.3 Aufbau und Methodik der Arbeit

Neben der Definition der grundlegenden Begriffe macht die Arbeit einen kurzen Streifzug durch die Geschichte und Entwicklungen der Gebäudeautomation, bevor in Kapitel 3 eine Analyse der sicherheitstechnischen Schwachstellen in der Gebäudeautomation vorgenommen wird. In Kapitel 4 wird anschließend untersucht, wie die eruierten Schwachstellen beseitigt werden können. Kapitel 5 bewertet die Ansätze des Vorkapitels. Im letzten Kapitel wird schließlich ein Fazit aus den Erkenntnissen dieser Arbeit gezogen und ein kurzer Ausblick auf die zukünftigen Entwicklungen gewagt.

2 Grundlagen

2.1 Grundlagen Netzwerktechnik

2.1.1 ISO/OSI Referenzmodell

Jeder, der sich mit der Kommunikation in Rechnernetzen auseinandersetzt, wird früher oder später mit dem ISO/OSI-Referenzmodell konfrontiert. ISO steht hierbei für die International Organization for Standardization, OSI für Open Systems Interconnection. Das ISO/OSI-Referenzmodell wurde Anfang der 80er Jahre unter dem Namen „Information Processing Systems: Open Systems Interconnection – Basic Reference Model“ als ISO/IS 7498 veröffentlicht. Es war ein erster ernsthafter Ansatz, die Kommunikation in verteilten Systemen zu standardisieren. Der ISO-Standard sah keine konkreten Implementierungen von einzelnen Protokollen vor und wurde bis dato auch in der dort definierten Form mit klar abgegrenzten Diensten pro Schicht nicht umgesetzt. Vielmehr wird das ISO/OSI-Modell als Referenz für die Kommunikation in offenen Systemen herangezogen.

Das ISO/OSI-Kommunikationsmodell ist als ein Schichtenmodell organisiert, wobei jeder Schicht bestimmte Aufgaben zugewiesen sind und sie der direkt darüber liegenden Dienste zur Verfügung stellt. Jede Schicht stellt dafür eine klar definierte Schnittstelle zur Verfügung. Das Schichtenmodell ist in Abbildung 1 dargestellt.

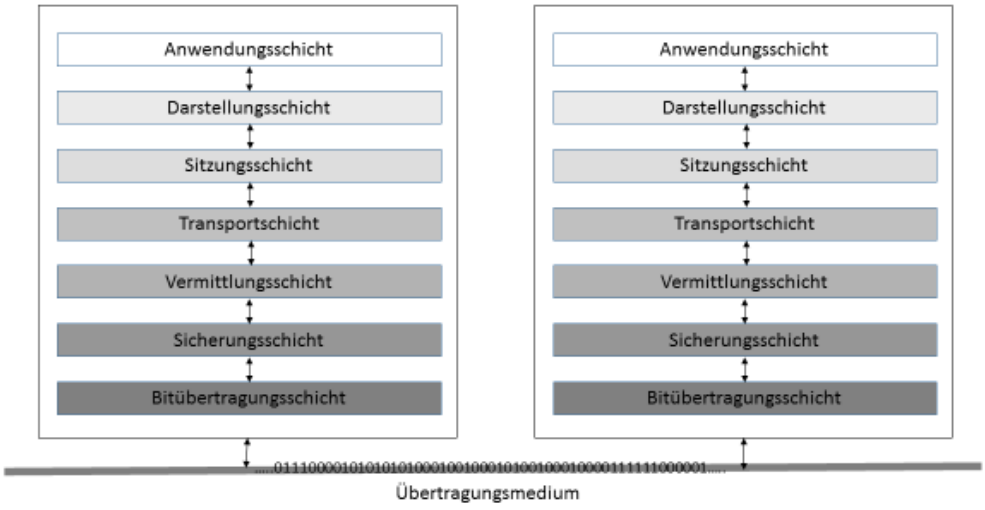


Abbildung 1: ISO/OSI-Referenzmodell

Von den einzelnen Schichten werden dabei folgende Aufgaben übernommen, wobei das Referenzmodell in dieser Ausführung – wie auch im praktischen Einsatz - von oben nach unten durchlaufen wird:

Anwendungsschicht (Schicht 7): In dieser Schicht ist die Anwendungsfunktionalität angesiedelt.

Darstellungsschicht (Schicht 6): Die Darstellungsschicht soll Dienste zur Verfügung stellen, mittels welcher Unterschiede in der Darstellung auf verschiedenen Systemen verborgen werden und Interoperabilität gewährleistet wird. Außerdem werden von dieser Schicht Kompression und Verschlüsselung angeboten.

Sitzungsschicht (Schicht 5, auch Kommunikationsschicht genannt): Die Sitzungsschicht stellt Dienste für einen organisierten Datenaustausch in der Netzwerkkommunikation (z.B. in Form von Dialogkontrolle) zur Verfügung. Die Dienste der Sitzungsschicht erweitern im Grunde jene der Transportschicht und ermöglichen unter anderem auch eine Wiederherstellung der Kommunikation nach einem Verbindungsabbruch.

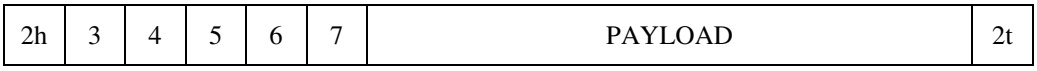
Transportschicht (Schicht 4): Die Transportschicht bietet Dienste für den Verbindungsauf- und -abbau. Als weiterer wichtiger Dienst werden auf dieser Schicht die Datenpakete segmentiert.

Vermittlungsschicht (Schicht 3, auch Netzwerkschicht genannt): Die Hauptaufgaben der Vermittlungsschicht liegen in der Suche des besten Weges der Datenpakete vom Sender zum Empfänger (Routing).

Sicherungsschicht (Schicht 2): Die Sicherungsschicht unterteilt die Datenblöcke. Aus diesen Blöcken wird eine Prüfsumme berechnet und dem Datenblock - dem so genannten „Rahmen“ - als Trailer hinzugefügt. Der Sicherungsschicht obliegen Aufgaben, eventuelle Fehler, welche während der Übertragung auftreten, mittels Prüfsummen auszumachen und gegebenenfalls zu korrigieren.

Bitübertragungsschicht (Schicht 1): Die eigentliche Übertragung der Daten passiert in der Bitübertragungsschicht. Sie definiert, wie die übertragenen Bit auf dem Übertragungsmedium dargestellt werden sollen (z.B. elektromagnetische Wellen in drahtlosen Netzwerken oder Spannungsstärke bei Kupferkabel).

Bei der Übermittlung fügen die verschiedenen Protokolle der Nachricht (bzw. den Übertragungspaketen) jeweils einen Header hinzu, die Sicherungsschicht zudem einen Trailer. Abbildung 2 veranschaulicht, wie ein Datenpaket nach dem Durchlaufen des gesamten ISO/OSI-Protokollstapels aussehen würde, bevor es Bit für Bit seine Reise zum Zielsystem antritt.



7… Header der Anwendungsschicht
6… Header der Darstellungsschicht
5… Header der Sitzungsschicht
4… Header der Transportschicht
3… Header der Vermittlungsschicht
2h... Header der Sicherungssicht
2t... Trailer der Sicherungsschicht

Abbildung 2: Aufbau eines Übertragungspaketes

2.1.2 Übertragungsmedien

Der technologische Fortschritt hat dafür gesorgt, dass die Vernetzung der kommunizierenden Parteien längst nicht mehr ausschließlich mittels Kabel passiert. Eine kabellose Datenübertragung per Funk gehört inzwischen genauso zum Standard wie Lichtwellenleiter, wobei sich die Reichweite und Bandbreite teils erheblich unterscheiden.

Die kabelgebundene Datenübertragung ist wohl das älteste der aktuell verwendeten Übertragungsmedien. In der Zeit, in der sich die Vernetzung von Rechnern immer mehr verbreitete, wurden Koaxialkabel mit Kupferleitern genutzt. Heute kommen vorrangig Twisted-Pair-Kabel (verdrillte Kupferkabel) in sternförmigen Netzwerken (siehe 2.1.3) zum Einsatz. Twisted-Pair-Kabel werden in Kategorien unterteilt, welche je nach Grad der Abschirmung der Kabeladern für unterschiedliche Frequenzbereiche, Übertragungsgeschwindigkeiten und räumliche Distanzen geeignet sind. Die feste Verkabelung bringt viele Vorteile, aber auch einige Nachteile mit sich. Ein Vorteil liegt sicher in der verhältnismäßig kostengünstigen Umsetzung bei relativ großer Bandbreite. Als Nachteil sollte die mangelnde räumliche Flexibilität der verkabelten Geräte erwähnt werden.

Kabellose Datenübertragung mittels Funknetzwerken findet mittlerweile auch in vielen privaten Haushalten statt. Oftmals erfolgt der Internetzugang in Heimnetzwerken mittels WLAN. Viele Betriebe und Bildungseinrichtungen, aber auch die öffentliche Verwaltung stellen eine kabellose Netzwerkanbindung zur Verfügung. Die

Norm IEEE 802.11 spezifiziert die Standards der Nutzung von Funkverbindungen im WLAN. Besonderes Augenmerk ist in kabellosen Netzwerken auf die Sicherheit zu richten, da das Übertragungsmedium – im Gegensatz zu verkabelten Netzwerken – quasi frei zugänglich ist. Vorteile bringt die Funktechnik im Bereich der Mobilität. Aber auch in Bereichen, in denen eine Verkabelung schwierig oder unmöglich ist, kann ein Funknetzwerk das Mittel der Wahl sein.

Glasfasern (auch als Lichtwellenleiter bezeichnet) kommen als Übertragungsmedium insbesondere dort zum Einsatz, wo eine hohe Bandbreite und/oder eine niedrige Störungsanfälligkeit wichtig sind. Die Übertragung erfolgt mittels optischer Signale (Lichtimpulse). Den hohen Anschaffungskosten von Lichtwellenleitern steht eine hohe Lebenserwartung gegenüber.

In der Gebäudeautomation spielt die Übertragungsgeschwindigkeit zumeist eine sekundäre Rolle. Ebenso ist die Quantität der übertragenen Daten überschaubar. Ist bereits eine IT-Infrastruktur vorhanden, welche ohne große Investition auch von der Gebäudeautomation genutzt werden kann, so teilen sich die beiden Bereiche oftmals die Übertragungswege. Werden die Komponenten eines Gebäudenetzwerkes hingegen nachgerüstet oder die Gebäudeautomation als autonomes System geplant, wird aus Gründen der Kosteneffizienz auf langsame Technologien zurückgegriffen.

2.1.3 Netzwerktopologien

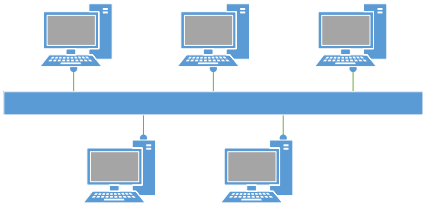
In einem Netzwerk können die Teilnehmer auf diversen Wegen miteinander kommunizieren. Je nach Anordnung der Komponenten im Netzwerk kann zwischen folgenden Topologien unterschieden werden (Abbildung 3 stellt die verschiedenen Topologien dar):

Bustopologie: Bei der Bustopologie sind alle Kommunikationsteilnehmer an einen gemeinsamen Bus angeschlossen. Protokolle müssen in diesem Fall die Behandlung von Kollisionen bei der Busverwendung berücksichtigen und für eine geordnete Kommunikation sorgen. Gibt es in einem Busnetzwerk eine Unterbrechung am Bus, so wird es allen hinter dem Unterbrechungspunkt liegenden Teilnehmer unmöglich, weiterhin mit den anderen Teilnehmern zu kommunizieren.

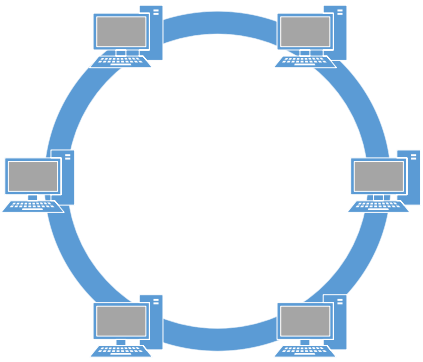
Ringtopologie: Ist das Netzwerk als Ringtopologie organisiert, so sind die Teilnehmer in Form eines Rings angeordnet. Im Vergleich zur Bustopologie bietet dies den

Vorteil, dass eine Unterbrechung an einem Punkt nicht das gesamte Netzwerk lahmlegt. Eine Kommunikation ist in diesem Fall weiterhin möglich.

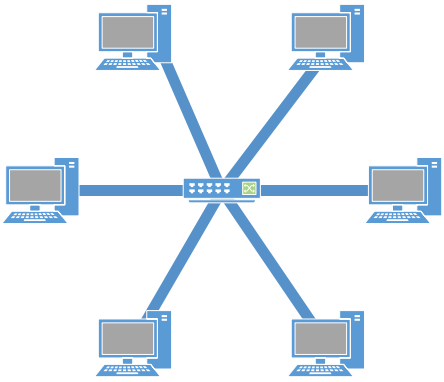
Sterntopologie: Bei einer Sterntopologie sind die Teilnehmer sternförmig angeordnet. Der Ausfall des zentralen Punktes führt zu einer vollständigen Kommunikationsunterbrechung.



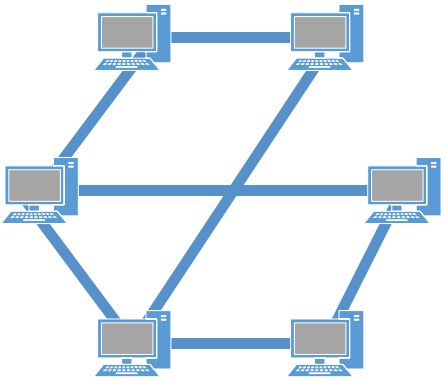
Bustopologie



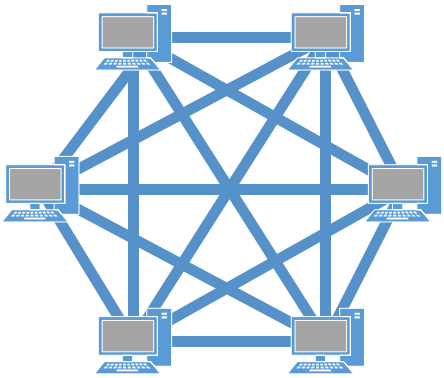
Ringtopologie



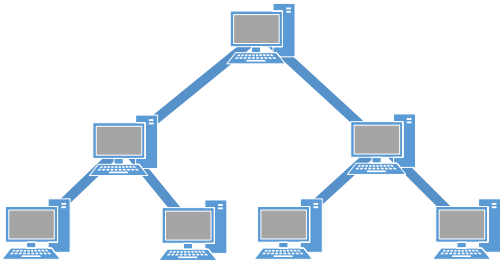
Sterntopologie



Maschentopologie, lose Vermaschung



Maschentopologie, Vollvermaschung



Baumtopologie

Abbildung 3: Netzwerktopologien

Maschentopologie: Bei der Maschentopologie handelt es sich um eine Punkt-zu-Punkt-Verbindung. Hat jeder Teilnehmer im Netzwerk zu jedem anderen eine direkte Verbindung, so spricht man von einer vollständigen Vermaschung. Sind nur einzelne Teilnehmer untereinander vernetzt, so handelt es sich um eine sogenannte lose Vermaschung. Der Ausfall eines Teilnehmers oder eines Übertragungsweges wirkt sich kaum bzw. überhaupt nicht auf die Erreichbarkeit der anderen Teilnehmer aus.

Baumtopologie: Bei der Baumtopologie sind die Teilnehmer baumartig miteinander verbunden. Fällt einer der Vaterknoten aus, so sind seine gesamten Söhne nicht mehr von den darüber liegenden Ebenen erreichbar.

Ein Netzwerk muss dabei nicht streng homogen nach einer Topologie aufbaut sein, auch Mischformen sind möglich.

In der Gebäudeautomation findet die Kommunikation auf der unteren Ebene zumeist über einen Bus statt, an welchen sämtliche Sensoren und Aktuatoren angeschlossen sind. Auf der Automations- und Managementebene (siehe hierzu auch Abschnitt 2.2.4) sind auch Netzwerke anzutreffen, welche als Sterntopologie organisiert sind.

2.2 Grundlagen der Gebäudeautomation

2.2.1 Entwicklung

Die Gebäudeautomation ist ein Bereich, der insbesondere in den letzten Jahren von einem konstant starken Wachstum geprägt ist. Die Wiege dieses Bereichs findet sich in den Segmenten Heizungs-, Lüftungs- und Klimaanlagenbau, in welchen elektronische Instrumente zur Steuerung, Regelung und Überwachung der einzelnen Komponenten schon seit vielen Jahren zum Einsatz kommen. Mehrere Hersteller haben die Vorzüge einer Automation bereits in frühen Jahren erkannt und die Entwicklung vorangetrieben. Dies resultierte darin, dass zwar mehrere Hersteller ihre Lösungen anboten, es allerdings (noch) keinen allgemeingültigen Standard gab, an welchem sich alle gemeinsam orientierten. Ein wahrer Wildwuchs von proprietären Lösungen und Protokollen war das Ergebnis, eine Koexistenz von Komponenten verschiedener Hersteller mit einer gemeinsamen Steuerung in einem einzigen System war nicht denkbar. In der Folge war der Endkunde damit an einen Hersteller gebunden.

Im Jahr 1987 traf im Zuge des Jahrestreffens der ASHRAE (American Society of Heating, Refrigerating and Air-Conditioning Engineers) zum ersten Mal das Projekt-

Komitee 135P zusammen. Erst acht Jahre später, im Jahr 1995, wurde der vom Komitee erstellte Standard als ANSI/ASHRAE Standard 135 veröffentlicht. Es enthielt eine Reihe von Regeln, welche Format, Inhalt und Bedeutung der in einem Gebäudenetzwerk ausgetauschten Daten vereinheitlichen sollte: die BACnet-Protokollschichten waren geboren. Es war ein erster Ansatz, die Herstellerabhängigkeit zu beseitigen. Im Jahr 2003 erlangte der Standard schließlich internationalen Status und wurde von der International Standards Organization unter der ISO 16484-5 veröffentlicht.

Der Weg zur Erweiterung der ursprünglichen Automationsbereiche auf andere Bereiche war geebnet. Längst hat die Gebäudeautomation den Weg in die privaten Haushalte gefunden und wird unter dem Begriff „Hausautomation“ zusammengefasst. „Ambient Assisted Living“ ist mittlerweile zur Sammelbezeichnung für Lösungen geworden, welche betreuungsbedürftigen Menschen einen möglichst unabhängigen und selbstbestimmten Alltag ermöglichen, aber auch der einfachen Unterstützung älterer Personen dienen sollen. Der zentrale Aspekt der Funktionalität wird um den immer weiter in den Mittelpunkt rückenden Faktor Komfort ergänzt.

BACnet ist nicht der einzige Kommunikationsstandard in der Gebäudeautomation, aber einer der – insbesondere in den USA und Australien - meistverwendeten.

2.2.2 Begriffsbestimmung und Abgrenzung

Merz et al. grenzen in [1] den Begriff Gebäudeautomation von jenem der Gebäudeüberwachung ab. Sie beziehen sich dabei auf die in der Norm VDI 3814 Blatt 1-5 enthaltene Definition, welcher jener der DIN EN ISO 16484-2 entspricht. Die Gebäudeautomation ist dort bestimmt als die „Bezeichnung der Einrichtungen, Software und Dienstleistungen für automatische Steuerung und Regelung, Überwachung und Optimierung sowie für Bedienung und Management zum energieeffizienten, wirtschaftlichen und sicheren Betrieb der Technischen Gebäudeausrüstung.“ [2].

Merz et al. ordnen die Komponenten, die in der Gebäudeüberwachung anzutreffen sind, auf drei Hierarchie-Ebenen an, während die Hierarchie in der Gebäudeautomation auf fünf Ebenen organisiert ist. Abbildung 4 stellt die zwei Bereiche gegenüber und veranschaulicht, dass eine Komponente auch mehr als eine einzige Ebene abdecken kann.

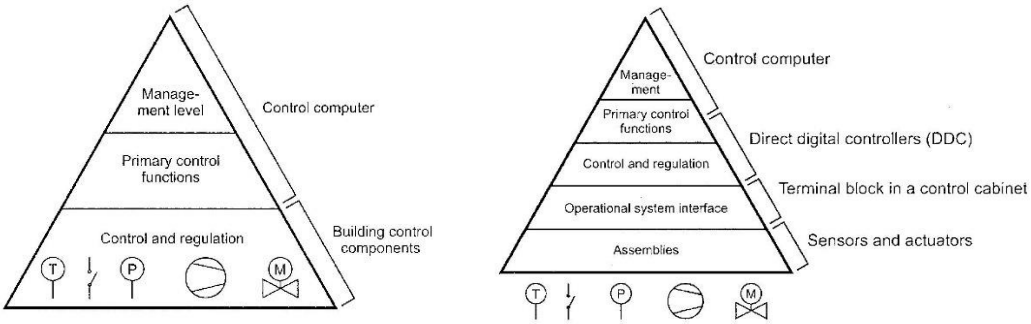


Abbildung 4: Gegenüberstellung: Gebäudeüberwachung - Gebäudeautomation [1]

Die Gebäudeüberwachung ist als ein Teilbereich der Gebäudeautomation zu verstehen, welcher auf räumlich abgegrenzte Bereiche beschränkt ist und das Hauptaugenmerk auf elektrische Installationen richtet.

Im Unterschied zur Gebäudeautomation kommt die Gebäudeüberwachung ohne Digitale Automationscontroller (Direct Digital Control, kurz DDC) aus. Im Bereich der Gebäudeüberwachung werden die Funktionen, welche zur Steuerung der Aktoren benötigt werden, direkt in oder nahe an den Endgeräten untergebracht. In der Gebäudeautomation werden diese Aufgaben zentralisiert von DDCs übernommen, welche aufgrund der Messungen der Sensoren oder der von Kontrolleinheiten erhaltenen Anweisungen reagieren.

2.2.3 Infrastruktur, Komponenten

Ein Gebäudeautomations-System besteht aus einer Vielzahl von Komponenten, welche nach der DIN EN ISO 16484-2 Hardware, Software und Dienstleistungen (Engineering) umfassen [2].

Unter den Begriff „Hardware“ fallen:

- Sensoren, die der Erfassung der Werte dienen
- Aktoren, welche Aktionen ausführen, die eine Änderung des Ist-Zustandes herbeiführen
- Medien zur Energieversorgung
- Übertragungsmedien
- Übertragungseinrichtungen
- Rechen- und Steuereinrichtungen
- Gebäudeleittechnik

Die Software umfasst alle Programme, welche zur Überwachung, Steuerung und Regulierung notwendig sind.

2.2.4 Kommunikation

Die Kommunikation findet in der Gebäudeautomation logisch auf drei Ebenen statt. Es handelt sich um die Feldebene, die Automationsebene und die Managementebene. Abbildung 5 stellt diese Ebenen und die auf der jeweiligen Ebene angesiedelten Protokolle dar.

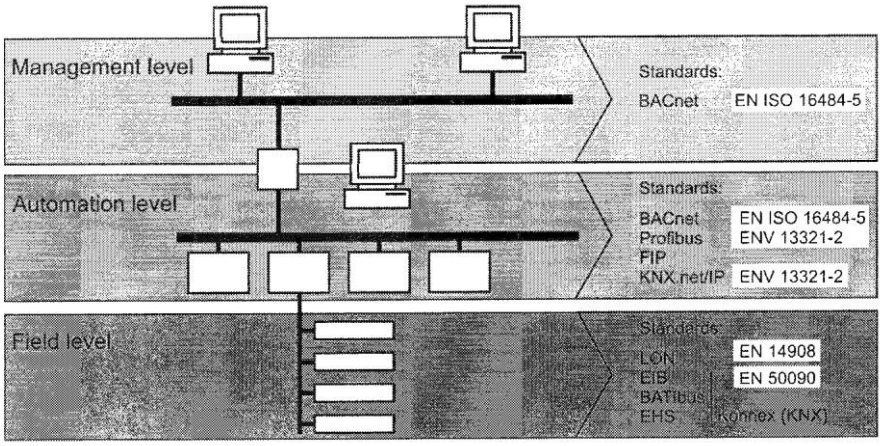


Abbildung 5: Ebenen und Protokolle in der Gebäudeautomation [1]

Auf der untersten Ebene, der Feld-Ebene, findet die Kommunikation zwischen Sensoren und Aktoren, sowie die Kommunikation in Richtung Automationsebene statt. Dazu stellt die Feld-Ebene eindeutig und vollständig definierte Schnittstellen zur Verfügung. Die Aktoren und Sensoren sind auf der Feld-Ebene über einen Bus verbunden (siehe Bustopologie, Abschnitt 2.1.3), es kommt aber auch kabellose Übertragung über Funk zum Einsatz.

Auf der Automationsebene sind die DDC zumeist über Bussysteme verbunden. Diese leiten Informationen zur Verarbeitung nach oben an die Managementebene und nach unten an die Aktoren der Feldebene weiter.

Die Managementebene überwacht sämtliche Abläufe im Gebäudenetzwerk. Auf dieser Ebene werden die Daten, welche von der Automatisierungsebene übermittelt wurden, ausgewertet, aufbereitet und visualisiert. Aufgrund der Auswertung wird entschieden, wie auf die empfangenen Daten reagiert werden soll und welche Befehle an die DDC zu erteilen sind.

Zwischen den einzelnen Ebenen findet jeweils eine sowohl horizontale, als auch vertikale Kommunikation statt.

An dieser Stelle soll erwähnt werden, dass für die Kommunikation im Gebäudenetzwerk dieselben Übertragungswege wie in der konventionellen Netzwerktechnik genutzt werden können – und effektiv auch genutzt werden. Gebäudeautomation und Computernetzwerke teilen sich somit die Kommunikationswege. In Kapitel 3 wird beleuchtet, wie sich diese Tatsache auf die Sicherheit auswirkt.

2.3 Grundlagen BACnet

Als der wohl wichtigste Vertreter der Protokolle, welche für den Bereich Gebäudeautomation konzipiert wurden, soll an dieser Stelle BACnet (Building Automation and Control Network) vorgestellt werden. Im Unterschied zu mehreren anderen Vertretern deckt BACnet die Kommunikation auf allen drei Systemebenen, der Feld-, Automations- und Managementebene, ab (siehe hierzu auch 2.2.4).

2.3.1 Dienste

Die Norm ISO 16484-5 [3] definiert auf der Anwendungsebene folgende Klassen von Diensten, welche von BACnet-Geräten angeboten werden können:

- Dienste zur Alarm- und Ereignisverarbeitung (Punkt 13 der Norm)
- Dienste für den Dateizugriff (Punkt 14 der Norm)
- Dienste für den Objektzugriff (Punkt 15 der Norm)
- Dienste für das Remote-Management von Geräten (Punkt 16 der Norm)
- Virtual-Terminal-Dienste (Punkt 17 der Norm)
- Dienste für die Netzwerksicherheit (Punkt 24 der Norm)

Die Kommunikation in BACnet erfolgt unter Nutzung dieser verschiedenen Dienste. Der Datenaustausch wird dabei nach dem Client-Server-Prinzip durchgeführt. Das anfordernde Gerät sendet ein Datenpaket, welches die Anfrage (Request PDU) an das Gerät, welches den Dienst anbietet, enthält. Das Gerät, welches den Dienst zur Verfügung stellt, antwortet mit einem Paket, welches die angeforderten Daten enthält (Response PDU).

2.3.2 Objekte

Bei BACnet handelt es sich auf Anwendungsebene um ein objektorientiertes Protokoll. BACnet modelliert ein Gerät als eine Ansammlung von Objekten. Jede von

einem Gerät angebotene Funktion wird als Objekt definiert. Dadurch kann ein Gerät auch durch mehrere Objekte beschrieben sein. Aktuell sind in der Norm 54 Objekttypen definiert.

2.3.3 Protokollschichten

Die Anforderungen, welche an die Kommunikation in einem Gebäudenetzwerk gestellt werden, weichen in vielen Aspekten von jenen der IT ab. Da die Komponenten in einem Gebäudenetzwerk oftmals über wenig eigene Rechnerleistung verfügen und die Netzwerksegmente größenordnungsmäßig in der Regel relativ überschaubar sind, ist eine komplette Implementierung des ISO/OSI-Stacks nicht notwendig. BACnet kommt mit einem reduzierten Protokollstapel von vier Schichten aus. Abbildung 6 veranschaulicht diese Architektur.

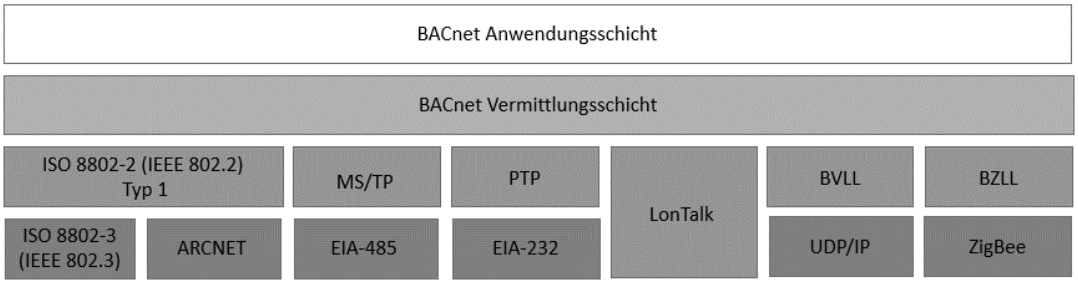


Abbildung 6: BACnet-Protokollschichten

Die Vielzahl der auf den untersten beiden Schichten verwendeten Protokolle ermöglicht den flexiblen Einsatz verschiedener Übertragungsmedien.

Im weiteren Verlauf dieser Arbeit wird BACnet als Referenzprotokoll für die im Bereich Gebäudeautomation verwendeten Protokolle herangezogen.

2.3.4 Konformität

Die Norm ISO EN 16484-6 spezifiziert die Konformitätsprüfung für BACnet-Geräte. Hierin ist definiert, welche Testprozeduren von einem Gerät durchlaufen werden müssen, um mit dem Standard konform zu sein. Selbstverständlich müssen auch die in dieser Arbeit erarbeiteten Lösungsansätze für DLP im Gebäudenetzwerk diese Voraussetzungen erfüllen.

2.4 Grundlagen IT-Sicherheit

Es vergeht kaum eine Woche, ohne dass in den Medien Schlagzeilen über neue Sicherheitsbedrohungen, Cyber-Angriffe oder Datendiebstahl veröffentlich werden.

Sicherheit ist in der IT wichtiger denn je zuvor geworden. Ein bestimmtes Maß an Sicherheitsstandards muss inzwischen auch von kleinen Unternehmen eingehalten werden, wenn es bei einer eventuellen Kreditanfrage keine Nachteile befürchten möchte. An dieser Stelle soll nur eine kurze Einführung in das Thema IT-Sicherheit gegeben werden.

2.4.1 Begriffsbestimmung

Ein System wird allgemeinhin als sicher bezeichnet, wenn die drei „klassischen" Schutzziele Integrität, Vertraulichkeit und Verfügbarkeit garantiert werden können. In der Literatur werden als weitere Schutzziele Authentizität, Funktionalität, Betriebssicherheit, Verlässlichkeit oder Stabilität angeführt.

Integrität bedeutet, dass die Daten richtig und vollständig, also ohne unautorisierte Veränderung vom Absender zum Empfänger übermittelt werden.

Vertraulichkeit bedeutet im Zusammenhang mit IT-Sicherheit, dass Informationen den Parteien nur dann zugänglich sind, wenn sie dazu befugt sind. Data Leakage gehört damit zu einer klassischen Verletzung des Schutzziels Vertraulichkeit.

Das Bundesamt für Sicherheit in der Informationstechnik (BSI) bezeichnet ein System als verfügbar, wenn dies von den Anwendern stets wie vorgesehen genutzt werden kann.

2.4.2 Arten der Sicherheitsbedrohungen

Die im Abschnitt 2.4.1 genannten Schutzziele können auf mehrere Arten gefährdet werden [4]:

- Abfangen: Verschafft sich ein unbefugter Dritter Zugriff zu Daten, so handelt es sich um ein Abfangen. Dies kann in Form von Abhören der Kommunikation oder auch in Form von Zugriff auf das Dateisystem passieren.
- Stören: Werden die Übertragungswege oder die Daten selbst dahingehend manipuliert, dass sie dadurch nicht mehr oder nur mehr teilweise brauchbar sind, so spricht man von einer Störung.
- Verändern: Finden nicht autorisierte Änderungen statt, welche dafür sorgen, dass die ursprünglichen Absichten der Datenübertragung nicht mehr zielführend weiterverfolgt werden können, so handelt es sich um eine Veränderung. Das ursprünglich angestrebte Ziel wird durch ein nicht erwünschtes ersetzt.

- Einbringen: Ein Angreifer kann zusätzliche Nachrichten oder Daten einbringen, welche ihm einen Vorteil verschaffen oder einen bestimmten Zweck verfolgen.

2.4.3 Allgemeine Schutzmaßnahmen

Das Wissen allein, von welchen Sicherheitsbedrohungen Gefahr ausgeht, bietet noch keinen Schutz davor. Tanenbaum und van Steen führen in [5] als wichtige Sicherheitsmechanismen Verschlüsselung, Authentifizierung, Autorisierung und Kontrolle an.

Bei der Verschlüsselung werden die Daten, welche geschützt werden sollen, mittels eines Verfahrens derart umgewandelt, dass nur derjenige den Klartext wiederherstellen kann, der im Besitz eines Entschlüsselungsschlüssels ist. Hierbei kann zwischen symmetrischen und asymmetrischen Verfahren unterschieden werden.

Authentifizierung bedeutet, dass die Identität des Benutzers überprüft werden kann und er beweisen kann, tatsächlich jener zu sein, der er vorgibt zu sein.

Bei der Autorisierung wird hingegen überprüft, ob der Benutzer effektiv auf die Ressourcen zugreifen darf, auf welche er zugreifen möchte.

Kontrolle kann Sicherheitsbedrohungen zwar nicht abwenden, allerdings ermöglicht sie die Nachvollziehbarkeit von potentiellen oder effektiven Angriffen und erlaubt es dadurch, Schutzmaßnahmen zu ergreifen.

2.5 Grundlagen Data Leakage

Der Begriff „Data Leakage“ wird oft im Zusammenhang mit Sicherheits-Zwischenfällen in der IT verwendet. Doch wie ist Data Leakage definiert und was fällt alles unter das Thema Data Leakage? Wo findet Data Leakage statt? Und wie kann man sich vor Data Leakage schützen?

2.5.1 Begriffsbestimmung

Shabtai, Elovici und Rokach definieren in [6] den Begriff „Data Leakage als „versehentliche oder ungewollte Verteilung privater oder sensibler Daten an einen nicht berechtigten Dritten“. Diese Definition des Begriffs wird auch von den führenden Herstellern von Data Leakage Protection Lösungen geteilt und gleichartig in einschlägiger Literatur verwendet. Ebenso wird diese Definition für die gegenständliche

Arbeit verwendet. Das „ungewollt“ ist hierbei als Synonym zu „unerwünscht aus der Sicht des Unternehmens“ zu verstehen.

2.5.2 Position gefährdeter Daten

Produkte, welche einen Schutz vor Data Leakage gewährleisten sollen, unterscheiden zwischen folgenden „Aufenthaltsorten“ von Daten:

- Auf Datenträgern gespeicherte Daten („Data-At-Rest“)
- Daten, welche sich in Bearbeitung befinden („Data-In-Use“)
- Daten auf dem Übertragungsweg („Data-In-Motion“)

Je nach Position der Daten wird ein bestimmter Ansatz verfolgt, einen unkontrollierten Abfluss von Daten gezielt zu vermeiden.

2.5.3 Überwachungspunkte

Da die Daten, welche schützenswürdig sind, sich nicht statisch an einem Ort befinden, müssen auch verschiedene Punkte im Netzwerk auf Data Leakage überwacht werden. Daten können sowohl an den Endpunkten, als auch auf dem Übertragungsweg (Netzwerk) abfließen [6]. Anbieter von Lösungen zur Vermeidung von Data Leakage führen aus diesem Grund eine Analyse der Daten an den Endpunkten (Clients, Server, Gateways) und im Netzwerk (vordefinierte Kontrollpunkte) durch.

2.5.4 Identifizierung gefährdeter Daten

Um festlegen zu können, wann es sich effektiv um Data Leakage handelt, muss zuerst festgelegt werden, bei welchen Daten es sich konkret um „private oder sensible Daten“ handelt. Gelangen die auf dem Server gespeicherten Ergebnisse der letzten Kegelpartie eines Mitarbeiters in die Hände einer nicht berechtigten Person, so ist diesem Vorfall sicher eine andere Gewichtung beizumessen als dem Abhandenkommen von Skizzen der nächsten Produktentwicklung.

Steht fest, welche Daten als kritisch oder sensibel eingestuft sind, so kann die Identifizierung durch das System auf zwei Arten erfolgen [6] [7]. Wird der Kontext untersucht, in welchem Operationen auf Daten stattfinden, so bezeichnet man dies als kontextbasierte Analyse. Wird hingegen der Inhalt einer Datei überprüft, so spricht man von inhaltsbasierter Analyse.

Die kontextbasierte Analyse nutzt die Ansätze, wie sie in Firewalls, Spamfiltern oder IDS zum Einsatz kommen. Es wird untersucht, ob eine gegebene Operation in einem

bestimmten Kontext zulässig bzw. verdächtig ist. Hierbei werden die Daten auf die Übereinstimmung mit bestimmten Kriterien hin geprüft. Ein Beispiel hierfür wäre die die Kontrolle, ob ein bestimmter Benutzer befugt ist, sich am Rechner des Abteilungsleiters anzumelden.

Bei der inhaltsbasierten Inspektion können verschiedene Techniken angewendet werden:

- Lexikalische Analyse des Inhalts, Wörterbuchabgleich
- Statistische Analyse, Häufigkeitsanalyse
- Verwendung regulärer Ausdrücke, um nach Daten zu suchen, die ein bestimmtes Format aufweisen (z.B. Bankkoordinaten)
- Übereinstimmung einer Datei mit einer als sicherheitskritisch bekannten
- Teilweise/vollständige Übereinstimmung des Inhalts mit einem als sicherheitskritisch bekannten
- Vergleich der Fingerprints mit jenen sicherheitskritischer Daten/Dateien

Bei der inhaltsbasierten Analyse ist es wichtig, dass auch die Tatsache berücksichtigt wird, dass Daten in „inhaltsfremden“ Dateien untergebracht sein können (z.B. Einfügen einer Excel-Tabelle in eine AutoCAD-Zeichnung). Ebenso sollte die Möglichkeit einer Entschlüsselung verschlüsselter Daten gegeben sein.

2.5.5 Schutz vor Data Leakage

Nachdem in den vorangegangenen Abschnitten aufgezeigt wurde, wo Daten abfließen können und wie sensible Daten ausgemacht werden können, soll nun kurz erläutert werden, welche Ansätze zum Verhindern von Data Leakage verfolgt werden können [6]:

- Ansätze, die ihren Fokus auf das Erkennen sensibler Daten richten: Bei dieser Methode werden vor allem jene Techniken eingesetzt, welche den in Abschnitt 2.5.4 vorgestellten entsprechen. Werden sensible Daten identifiziert, so kann entschieden werden, welche Schritte ergriffen werden. Ein möglicher Schritt könnte das Verweigern der Weiterleitung der Pakete sein.

  Einen weiteren Ansatz zum Erkennen von sensiblen Daten stellt das so genannte „Content tagging“ dar. Hierbei werden Dateien, welche sensible Daten enthalten, mit einer speziellen Markierung (dem Tag) versehen.

- Präventive Ansätze: Soll die Möglichkeit, sensible Daten in Umlauf zu bringen, bereits vorab so weit wie möglich eingeschränkt werden, spricht man von präventiven Ansätzen. Diese umfassen vor allem restriktive Zugriffsbeschränkungen, das Einschränken der Funktionalität (z.B. Copy & Paste) und die obligatorische Verschlüsselung sensibler Daten. Ein nicht zu vernachlässigender Faktor ist die Bewusstseinsbildung der Mitarbeiter. Je besser diese darüber informiert sind, bei welchen Daten es sich um kritische Daten handelt und wie mit diesen umgegangen werden soll, desto umsichtiger wird dieser Umgang sein.

2.5.6 Data Leakage in Zahlen

Ein unerwünschter Abfluss sensibler Daten kann für ein Unternehmen nicht nur einen immensen wirtschaftlichen Schaden zur Folge haben, sondern auch einen bedeutenden Vertrauensverlust nach sich ziehen. Gerade Unternehmen, welche vom Verbrauchervertrauen abhängig sind, kämpfen mit den Konsequenzen eines Sicherheitsvorfalls.

Der Anbieter von Lösungen im Bereich Informationssicherheit und Schutz geschäftlicher Informationen „InfoWatch“ veröffentliche für das Jahr 2013 folgende Zahlen [8]:

- 2013 wurden um 22% mehr Fälle von Data Leakage registriert als im Jahr 2012
- 561 Mio. Datenbank-Einträge (u.a. Finanz- und Personaldaten) wurden enthüllt
- In 85% der Fälle handelt es sich bei den abgeflossenen Daten um Personendaten
- Der in Massenmedien veröffentlichte Verlust (einschließlich Ausgaben zur Beseitigung der Folgen von Datenlecks, Gerichtsverhandlungen, Entschädigungszahlungen), den Unternehmen durch Datenlecks 2013 erlitten haben, beträgt 7,79 Mrd. Dollar

Ebenso veröffentliche „InfoWatch“ folgende Statistiken, welche Data Leakage nach Verursachern und Quellen kategorisieren [8]:

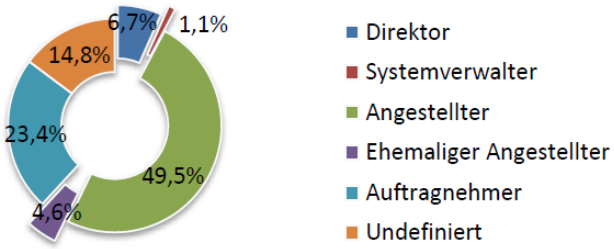


Abbildung 7: Verursacher von Data Leakage, 2013 [8]

Wie aus Abbildung 7 ersichtlich ist, ist es in der Regel nicht der mit Privilegien ausgestattete Systemadministrator, welcher für Data Leakage verantwortlich ist. In beinahe jedem zweiten Fall werden Daten von (oftmals unzufriedenen) Mitarbeitern ohne Ermächtigung an Dritte weitergegeben.

Bemerkenswert sind auch die Abbildung 8 zu entnehmenden Daten. Dort ist eine auffällige Verschiebung der Quelle des Datenabflusses zu beobachten: Während der Datenabfluss aufgrund des physischen Abhandenkommens des Geräts im Zeitraum 2012-2013 um mehr als ein Viertel zurückgegangen ist und die abgeflossen Daten 2013 gar nur mehr in 1,5 % der Fälle Mobilgeräten entstammen, ist eine starke Zunahme von Data Leakage im Bereich Netzwerk zu beobachten. Hier haben sich die Zwischenfälle im Vergleichszeitraum mehr als verdoppelt. Dies verdeutlicht, dass gerade dieser Bereich besondere Aufmerksamkeit verdient.

Abbildung 8: Quelle von Data Leakage, 2012-2013 [8]

3 Analyse

Nachdem in Kapitel 2 die zentralen Begriffe „Data Leakage“ und „Gebäudeautomation“ eingeführt wurden, wird in diesem Kapitel analysiert, mit welchen Arten von Data Leakage in der Gebäudeautomation gerechnet werden muss und an welcher Stelle des Gebäudenetzes die Daten abfließen können.

3.1 Ausgangssituation in der Gebäudeautomation

Im Bereich der Gebäudeautomation war das Hauptaugenmerk stets auf Funktionalität und Optimierung gerichtet. Sicherheitsaspekte nahmen höchstens den Status von Randbetrachtungen ein. Für lange Zeit hat es auch schlicht und einfach keine konkrete Veranlassung gegeben, sich mit dem Thema Sicherheit auseinanderzusetzen. Der Wildwuchs von proprietären Protokollen in der Gebäudeautomation war Fluch und Segen zugleich. Eine Interoperabilität der Geräte verschiedener Hersteller war unter den gegebenen Umständen nicht möglich. Allerdings war es aufgrund der Tatsache, dass die Details der Kommunikation im Gebäudenetzwerk nur dem Hersteller selbst bekannt waren und die Daten für Außenstehende unverständlich waren, auch kaum möglich, versehentlich oder absichtlich erworbene Daten aus dem Gebäudenetzwerk zu nutzen.

Der BACnet-Standard hat durch die Definition der BACnet-Protokollsuite den Weg zu einer offenen Kommunikation geebnet. Der Erweiterung auf neue Anwendungsgebiete stand damit nichts mehr entgegen. Genauso mannigfaltig wie die Einsatzmöglichkeiten ist nun auch die Art der übertragenen Daten. Werden die Vorzüge der Gebäudeautomation beispielsweise im Bereich der altersgerechten Assistenzsysteme für ein selbstbestimmtes Leben (Ambient Assisted Living) genutzt, so werden dabei durchaus auch Daten übertragen, welche nicht für die Augen der Öffentlichkeit bestimmt sind und als sensibel eingestuft werden sollten.

Die Normungsgremien des BACnet-Standards erkannten schon recht früh, dass das Thema Sicherheit auch in Gebäudenetzwerken berücksichtigt werden muss. Der Standard ANSI ASHRAE 135 wurde in der Version 135-2008 um ein Addendum g ergänzt, welches Sicherheitsfeatures für BACnet einführt. Diese Features sind in die DIN EN ISO 16484-5 als Kapitel 24 bzw. als Abschnitte in die Kapitel 6, 12, 18, Annex K und Annex R eingeflossen. In der Version DIN EN ISO 16484-5:2012 [3]

unterstützt BACnet als Verschlüsselungsalgorithmus AES und ersetzt damit den von vorhergehenden Versionen vorgesehen, aber inzwischen als unsicher geltenden DES-Standard. Zudem bietet BACnet MD5 als Hashverfahren an.

Die Norm gibt eine detaillierte Auskunft darüber, welche Verschlüsselungs- bzw. Signaturverfahren verwendet werden können. Über die Methode, mittels welcher die Schlüssel sicher vom in der Norm erwähnten Keyserver auf die Geräte verteilt werden können, wird jedoch keine Aussage getroffen, wie auch Granzer et al. in [9] feststellen.

Zusätzlich zur Verschlüsselung wurde im Addendum g des Standard ANSI ASHRAE 135-2008 erstmals ein Sicherheitsfeature angeführt, welches als „Data Hiding“ bezeichnet wird. Dieser Ansatz findet sich in der DIN EN ISO 16484-5:2012 [2] unter Punkt 24.16.7. „Data Hiding“ bedeutet in diesem Zusammenhang, dass in einer Kategorie von sicheren Geräten (im Standard als „secure devices“ bezeichnet), welche diesen Ansatz unterstützen, bestimmte Daten oder Dienste nicht frei zugänglich sind, sondern der Zugriff darauf eingeschränkt werden kann.

Im Abschnitt 24.14 der DIN EN ISO 16484-5:2012 wird außerdem die Benutzerauthentifizierung für den BACnet-Standard eingeführt.

3.2 Konkrete Sicherheitsbedrohungen in der Gebäudeautomation

Shabtai, Elovici und Rokach identifizieren in [6] drei „Aggregatzustände“ von Daten, welche in einem verteilten System abfließen können (siehe auch 2.5.2). Es wird nun untersucht, inwieweit dies auf die Gebäudeautomation übertragen werden kann.

3.2.1 Auf Datenträgern gespeicherte Daten („Data-At-Rest“)

In einem Gebäudenetzwerk werden Daten an mehreren Stellen gespeichert.

Ein Punkt, welcher hier zu betrachten ist, ist die Feldebene. Hier werden aktuelle Daten in Objekten als Werte von BACnet-Attributen abgelegt. Diese Daten können bei Bedarf von verschiedenen Entitäten ausgelesen oder verändert werden.

Eine ungleich umfangreichere Datenansammlung findet sich hingegen auf der Managementebene. Managementstationen bieten Zugriff auf den gesamten Datenbestand im Gebäudenetzwerk, da an dieser Stelle die zentrale Überwachung, Steuerung und Visualisierung des Netzwerkes stattfindet. Während auf der Feldebene nur

Informationen über den lokalen Zustand eines Gerätes zur Verfügung stehen und damit begrenzt verwertbar sind, kann ein versehentlicher oder unerwünschter Datenabfluss auf der Managementebene potentiell in großem Umfang erfolgen.

Daten werden auf der Managementebene häufig in Datenbanken abgelegt. Dort werden auch die historisch erfassten Daten verfügbar gehalten. Diesen Datenbanken sollte bei der Implementierung eines Schutzmechanismus gegen unerwünschten Datenabfluss Beachtung geschenkt werden.

Nicht vergessen werden sollten bei der Betrachtung potentieller Datenabflusspunkte zudem auf sämtliche Datenträger, welche für Backupzwecke verwendet werden. Backups werden in der Regel systemweit zentral durchgeführt. Ein Zugang zu diesen Backups impliziert das Risiko, dass ein großer Datenbestand abfließen kann.

Die Art, wie gespeicherte Daten in einem Gebäudenetzwerk behandelt werden, unterscheidet sich nicht wesentlich von der aus der IT bekannten. Zumal Managementcomputer zumeist mit einem konventionellen Betriebssystem ausgestattet sind, eignen sich COTS-DLP-Lösungen (Commercial off-the-shelf, also serienfertige Produkte) durchaus auch für die in einem Gebäudenetzwerk gespeicherten Daten.

3.2.2 *Daten in Bearbeitung („Data-In-Use“)*

Unter „Daten in Bearbeitung“ werden all jene Informationen zusammengefasst, welche gerade für die Interaktion mit dem Benutzer in Verwendung sind [6]. In der Gebäudeautomation finden sich Daten in Bearbeitung primär auf den Managementstationen. Dem Benutzer, der mit den notwendigen Berechtigungen ausgestattet ist, die Managementstation zu bedienen, stehen ohne Schutzmechanismen mehrere Möglichkeiten zur Verfügung, Daten abfließen zu lassen. Es sind dies unter anderem:

- Kopieren von Dateien auf externe Datenträger
- Übertragen sensibler Informationen auf Ausgabegeräte
- Steganographie (Verstecken von sensiblen Informationen in nicht verdächtigen Dateien), welche ausschließlich der Verschleierung des Datenabflusses dient
- Instant Messaging
- Schreiben von Mails
- Copy&Paste-Vorgänge
- Screenshots

Die Situation in der Gebäudeautomation unterscheidet sich für Daten in Bearbeitung kaum von der Realität in IT-Netzwerken. Um Data Leakage für Daten in Bearbeitung vorzubeugen ist der Einsatz von konventionellen DLP-COTS-Lösungen auf den Managementstationen sinnvoll und zielführend.

3.2.3 *Daten in Übertragung („Data-*In*-Motion“)*

Nachdem in den vorangehenden Abschnitten 3.2.1 und 3.2.2 festgestellt wurde, dass für auf Datenträgern gespeicherte Daten und Daten in Bearbeitung die Risiken eines Datenabflusses auch ohne maßgeschneiderte Lösungen eingedämmt bzw. abgewendet werden können, soll in diesem Abschnitt analysiert werden, wie es um Daten in Bewegung in einem Gebäudenetzwerk bestellt ist.

Die Kommunikation in einem Gebäudenetzwerk erfolgt teilweise (insbesondere auf Managementebene unter Ethernet-Netzwerken) über die Infrastruktur der Rechner und Geräte des konventionellen IT-Netzwerkes. Dies hat zur Folge, dass Daten – sei es informative Ereignisse, als auch sensible Informationen – von beiden Seiten über diese Kommunikationswege übertragen werden können. Während für den Bereich IT ausgeklügelte Sicherheitssysteme, die den Datenfluss auf seine Zulässigkeit prüfen, sowohl in Form von Hard- als auch Software angeboten werden, steckt die Entwicklung äquivalenter Systeme für die Gebäudeautomation noch in den Kinderschuhen. Bisher gibt es praktisch keine zuverlässige und ausgereifte Lösung, die diesen Bereich abdeckt. Aus diesem Grund ist den Daten in Übertragung in der Gebäudeautomation besondere Bedeutung zuzumessen.

Bei der Betrachtung der Daten in Übertragung treffen wir auf eine grundlegend andere Situation als bei Daten in Bearbeitung/gespeicherte Daten. Der große Unterschied der im Gebäudenetzwerk ausgetauschten Daten zu jenen im konventionellen IT-Netzwerk liegt in der Art, wie die Kommunikation stattfindet. Genauer: die Gebäudeautomation bedient sich Protokollen, welche in der IT nicht zum Einsatz kommen (vgl. Abschnitt 2.3.3). Während bei BACnet auf den Schichten 1 und 2 (Bitübertragungs- und Sicherungsschicht) zum Teil Protokolle verwendet werden, welche auch in der IT anzutreffen sind, so sind auf den darüber liegenden Schichten Vermittlungs- und Anwendungsschicht Protokolle definiert, die ausschließlich in der Gebäudeautomation zum verwendet werden. Firewalls, wie sie in jedem kleineren und größeren Unternehmen zum Einsatz kommen, können mit den Paketen auf den oberen Schichten, welche aus dem Gebäudenetzwerk übermittelt werden, schlicht

und einfach nicht umgehen. Dies bedeutet (aus sicherheitstechnischer Sicht) bestenfalls, dass sämtliche Pakete verworfen werden (Whitelist-Strategie) und schlimmstenfalls, dass diese Pakete frei im Netzwerk verkehren können (Blacklist-Strategie). Handelt es sich um eine Blacklist-Strategie, so ist einem Datenabfluss der Weg geebnet. Doch nicht nur im IT-Netzwerk, welches oft für die Anbindung eines Gebäudenetzwerkes genutzt wird, muss für die notwendige Flusskontrolle gesorgt werden. Der Weg der Pakete vom Gebäude- in das IT-Netzwerk muss erst gar nicht bestritten werden, falls die Daten dort nicht benötigt werden (insbesondere Broadcast-Nachrichten). Daten in Bewegung können schon vorher, noch im Gebäudenetzwerk selbst abfließen. Eingeleitet kann dieser Abfluss beim Fehlen von Schutzmaßnahmen im Grunde von all jenen werden, die Zugriff auf das Gebäudenetzwerk haben. Diese Tatsache zeigt auf, dass der Datenfluss in der Gebäudeautomation genauso restriktiv gehandhabt werden muss, wie es in IT-Netzwerken längst gang und gäbe ist.

Um eine sinnvolle Lösung zur Vermeidung eines Datenabflusses anwenden zu können, muss vorab natürlich nicht nur festgelegt werden, welche Stellen des Netzwerkes und welcher Zustand der Daten anfällig für einen möglichen Abfluss sein könnten, sondern in erster Linie, welche Daten überhaupt als sensible Daten eingestuft werden sollen. BACnet-Objekte sind im Grunde eine simple Ansammlung von Objekten und deren Attributen. Nur durch Auslegung kann darüber befunden werden, ob es sich um ein sensibles Objekt handelt oder nicht. Auch dieser Aspekt soll im nächsten Kapitel berücksichtigt werden.

3.3 Zusammenfassende Beurteilung der Sicherheitsbedrohungen

Die Gebäudeautomation ist ebenso wie konventionelle IT-Netzwerke potentiell dem Risiko eines unerwünschten Datenabflusses ausgesetzt. Auch wenn durch die Einführung von Verschlüsselungstechniken und Authentifizierung bereits wichtige sicherheitstechnische Schritte gesetzt wurden, so gibt es immer noch Aspekte, für welche keine Lösungen zur Beseitigung von Schwachstellen angeboten werden.

Aus einem Datenabfluss können sich verschiedene, unliebsame Szenarien ergeben. Fließen Sensorinformationen, welche auf die Anwesenheit von Personen im Gebäude schließen lassen, in kriminelle Hände ab, so können diese für die Planung von nicht gesetzmäßigen Handlungen wie Einbrüche oder Diebstähle genutzt werden. Ist ein

Gebäudekomplex für körperlich eingeschränkte Menschen ausgelegt, welchen durch Ambient Assisted Living wieder ein Stück Lebensqualität geschenkt wird, so kann der Grad der Sensibilität der Daten noch um einiges höher liegen. Betreuungseinrichtungen oder die Pharmaindustrie können durchaus Interesse an den in diesem Netzwerk übertragenen Daten haben und diese wirtschaftlich verwerten.

Zusammenfassend kann festgestellt werden, dass die in den Abschnitten 3.2.1 und 3.2.2 festgestellten Schwachpunkte durch kommerzielle DLP-Lösungen eingedämmt werden können. Diese werden im weiteren Verlauf der Arbeit nicht weiter untersucht.

Im nächsten Kapitel werden Ansätze verfolgt, welche eine Beseitigung der in Abschnitt 3.2.3 festgestellten Schwachstellen zum Ziel haben.

4 Ansätze zur Vermeidung von Data Leakage

Wie in den vorangegangenen Abschnitten geschildert wurde, sind es in Gebäudenetzwerken insbesondere die Daten in Übertragung, welche der Gefahr von Data Leakage relativ ungeschützt ausgesetzt sind. In diesem Kapitel werden nun die BACnet-Anwendungsschicht und die BACnet-Vermittlungsschicht unter die Lupe genommen. Für diese zwei Schichten bzw. Protokolle existieren zum jetzigen Zeitpunkt keine Mittel, um für Daten in Übertragung einen Schutz zu gewährleisten.

Eine sinnvolle Lösung zur Vermeidung von Data Leakage muss jedes der ankommenden Datenpakete darauf prüfen können, ob es in andere Bereiche des Netzwerkes gelangen darf oder ob es verworfen werden muss, da es den implizit oder explizit festgelegten Richtlinien für die Handhabung sensibler Daten nicht entspricht.

Shabtai et al. kategorisieren in [6] Literatur zur Date-Leakage-Erkennung und –Vermeidung unter anderem nach:

Netzwerk-/webbasierter Schutz

Verschlüsselung und Zugriffsüberwachung

Erkennen von böswilligen Angriffen von innen durch Einsatz von Honeypots und Honeytoken

Während der Aufwand für den Einsatz von Honeypots und Honeytoken aufgrund der im Verhältnis zu IT-Netzwerken überschaubaren Größe in Gebäudenetzwerken für nicht angebracht erachtet wird, ist der Einsatz von netzwerkbasierten Schutzmaßnahmen sowie Verschlüsselung und Zugriffsüberwachung auch in Gebäudenetzwerken angebracht. Webbasierter Schutz beschränkt sich im Bereich Gebäudeautomation auf Remote Management Interfaces und werden in dieser Arbeit nicht betrachtet.

BACnet unterstützt seit 2008 Verschlüsselung. Diese ist in Kapitel 24 der DIN EN ISO 16484-5 [3] nebst anderen Techniken für eine sichere Kommunikation definiert.

Zu berücksichtigen ist allerdings, dass die Gefahr, dass bei einer verschlüsselten Kommunikation das Risiko der Entschlüsselung der Daten durch einen Angreifer umso höher ist, je mehr Daten ihm zur Verfügung stehen. Prinzipiell sollte die Kommunikation nur von Punkt zu Punkt (vom Sender zum Empfänger) stattfinden und Datenpakete nicht mehr Übertragungswege als notwendig nutzen.

Der in dieser Arbeit verfolgte Ansatz ist jener einer „Appliance“, welche den Netzwerk-Traffic analysiert, gegebenenfalls filtert oder gar gänzlich unterbindet. Dazu muss in einem ersten Schritt untersucht werden, wie die Pakete auf den einzelnen Schichten aufgebaut sind. Anschließend werden Filtermöglichkeiten vorgeschlagen, anhand welcher die Appliance die Pakete zum gewünschten Ziel passieren oder nicht passieren lässt.

4.1 Anwendungsschicht

Die auf der Anwendungsschicht ausgetauschten Nachrichten enthalten effektiv jene Informationen, welche einen Interessierten neugierig machen könnten. Es ist dies die Schicht, auf der konkrete, von den Sensoren erhobene Daten übertragen werden und Instruktionen an Aktuatoren gesendet werden.

Wie in Abschnitt 2.3.3 erläutert, arbeitet BACnet mit einer reduzierten Anzahl von Protokollschichten. Die Transport-, Sitzungs-, Darstellungs- und Anwendungsschicht werden zu einer einzigen Anwendungsschicht zusammengefasst. Dieser als „BACnet Anwendungsschicht“ bezeichnete Layer übernimmt damit mehrere Aufgaben.

Die Norm DIN EN ISO 16484-5:2012 [3] spezifiziert in Kapitel 22, welchen Bedingungen jedes BACnet-Gerät zumindest erfüllen muss, um dem Standard zu genügen. In den so genannten „PICS“ (Protocol Implementation Conformance Statements) listet der Hersteller eines BACnet-Geräts, welche Funktionen implementiert sind. Ob diese Funktionen effektiv implementiert sind, wird in einem standardisierten Testverfahren festgestellt.

Laut Norm muss ein Gerät auf Anwendungsebene zumindest [3]:

a) genau ein Objekt vom Typ „Device object“ enthalten. Dieses identifiziert das Gerät selbst
b) den Dienst ReadProperty implementiert haben
c) die Dienste „Who-Has“ und „Who-Is“ mit entsprechender Antwort „I-Have“ und „I-Am“ beherrschen (Ausnahme: MS/TP Slave)
d) den Dienst „WriteProperty“ ausführen, wenn das Gerät „WritePropertyMultiple“, „AddListElement“ oder „RemoveListElement“ als Dienste anbietet

e) es dem Dienst „WriteProperty“ erlauben, sämtliche Eigenschaften, welche als veränderbar deklariert sind, durch die Dienste „AddListElement“ oder „RemoveListElement“ zu modifizieren
f) den Dienst „WriteProperty“ ausführen, wenn das Gerät Objekte enthält, welche als schreibbar gelten

4.1.1 Arten von Nachrichten (Primitive)

Die DIN EN ISO 16484-5 [3] definiert den Informationsaustausch zwischen zwei Punkten als Austausch von Daten mittels abstrakten, primitiven Diensten. BACnet nutzt hierfür vier Primitive:

- Anforderung (request)
- Anzeige (in der Norm als „indication“ bezeichnet. Anzeige ist hier im Sinne von Bekanntmachung zu verstehen)
- Antwort (response)
- Bestätigung (confirm)

Werden diese Primitiven nun mit den Möglichkeiten der bestätigten und unbestätigten Kommunikation (ähnlich TCP und UDP in der IT-Welt), dem Ablehnen von Paketen und der Option des kompletten Verbindungsabbruchs kombiniert, ergeben sich folgende mögliche Konstellationen von PDUs [3]:

Anforderungen (requests)

- CONF_SERV.request: Anforderung, bestätigter Dienst
- UNCONF_SERV.request: Anforderung, unbestätigter Dienst, einzelne Nachricht
- SEGMENT_ACK.request: Anforderung, Senden eines ACK als Bestätigung für den Erhalt einer PDU (auf die Details wird im Abschnitt 4.1.3 eingegangen)
- REJECT.request: Anforderung, Abweisen der APDU

Anzeigen/Bekanntmachungen (indications)

- CONF_SERV.indication: Bekanntmachung, dass eine CONF_SERV.request-PDU eingegangen ist
- UNCONF_SERV.indication: Bekanntmachung, dass eine UNCONF_SERV.request-PDU eingegangen ist

- SEGMENT_ACK.indication: Bekanntmachung, dass ein ACK empfangen wurde
- REJECT.indication: Bekanntmachung, dass eine PDU abgewiesen wurde
- ABORT.indication: Bekanntmachung, dass ein Verbindungsabbruch gefordert wurde
- SEC_ERR.indication: Bekanntmachung, dass eine PDU den Sicherheitsrichtlinien nicht entspricht. SEC_ERR.indication stellt einen Sonderfall dar, da diese Art von Nachricht in der Norm lediglich dahingehend spezifiziert wird, dass die entsprechende Behandlung der Nachricht lokal erfolgen muss

Antworten (responses)

- CONF_SERV.response: Antwortnachricht auf eine CONF_SERV.request-Anforderung

Bestätigungen (confirmations)

- CONF_SERV.confirm: Bestätigung des Eingangs einer Antwortnachricht vom Typ CONF_SERV.response

Je nach Art der in der APDU enthaltenen Informationen ändert sich damit nicht nur der Inhalt der Protokolldateneinheit, sondern auch der Aufbau des Headers (siehe Abschnitt 4.1.3).

4.1.2 Ablauf der Kommunikation

Die Kommunikation auf der BACnet-Anwendungsschicht läuft nach dem Client-Server-Prinzip ab. Einer der Kommunikationsteilnehmer nimmt dabei einen Dienst in Anspruch, den der andere Teilnehmer anbietet. Die Rolle des Servers und des Client sind hierbei nicht statisch festgelegt. Ein Teilnehmer bietet Dienste an, kann aber gleichzeitig auch Dienste anderer Geräte in Anspruch nehmen.

Die DIN EN ISO 16484-5:2012 [3] sieht vor, dass die kommunizierenden Parteien für jede Transaktion eine Tabelle mitführen, in der die Eckdaten der Transaktion erfasst werden (z.B. Anzahl der Wiederholungsversuche, letzte Sequenznummer, Überwachung von Timeouts). Die Norm bezeichnet diese Tabelle als „Transaction State Machine (TSM)“. Die TSM bleibt bis zum Ende der Transaktion bestehen und protokolliert den Nachrichtenaustausch.

Da das grundlegende Verständnis des Kommunikationsablaufs für das weitere Verständnis wichtig ist, wird an dieser Stelle der Kommunikationsfluss einiger Kommunikationssequenzen, angelehnt an Punkt 5.5 der DIN EN ISO 16484-5:2012 [3], abgebildet:

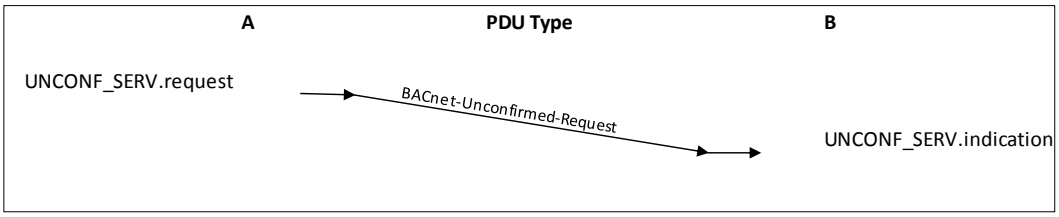


Abbildung 9: Unbestätigter Dienst, normaler Ausgang

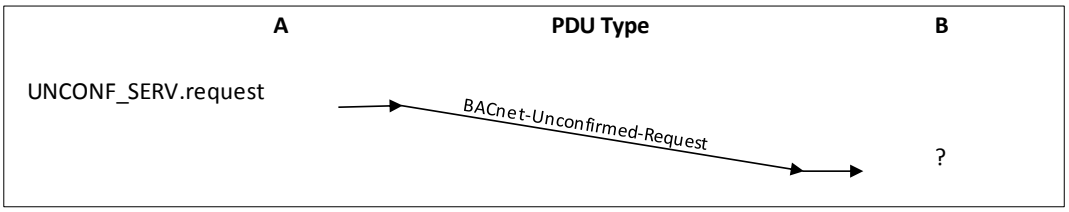


Abbildung 10: Unbestätigter Dienst, anormaler Ausgang

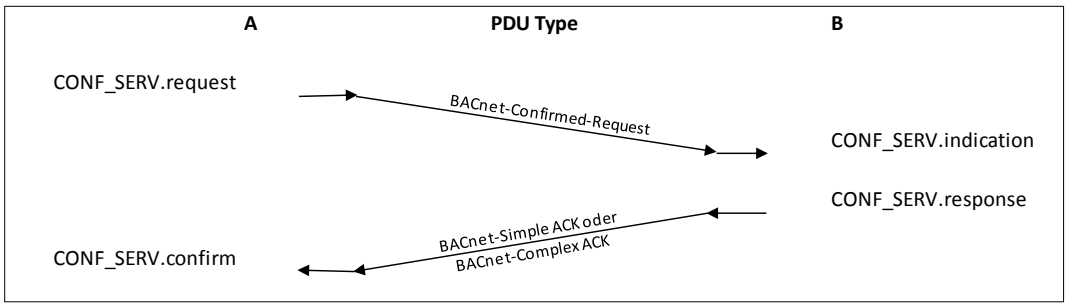


Abbildung 11: Bestätigter Dienst, normaler Ausgang

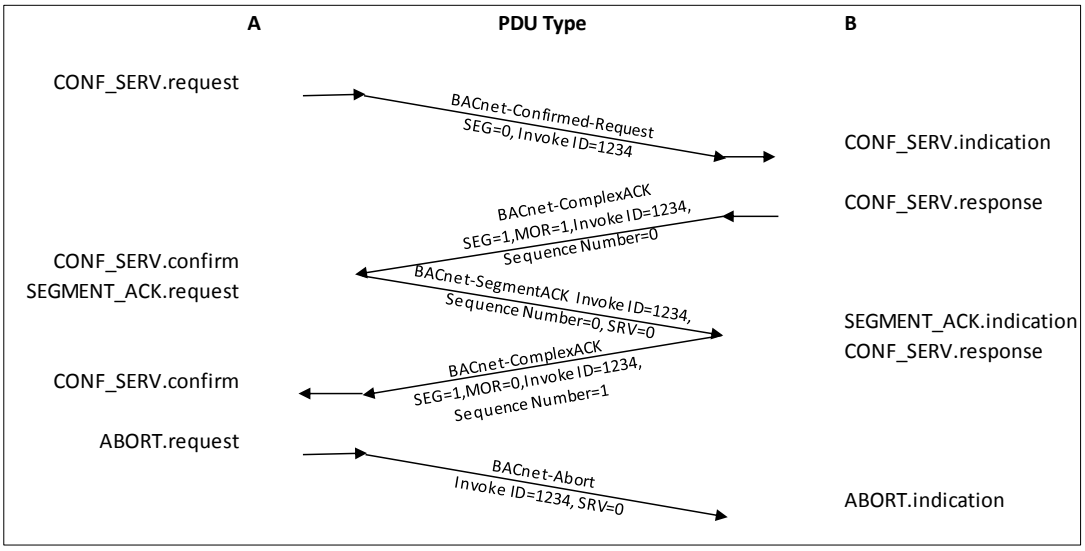


Abbildung 12: Bestätigter Dienst, normaler Ausgang, Flusskontrolle durch die Anwendung, Segmentierung der Antwort, Anfrage abgebrochen

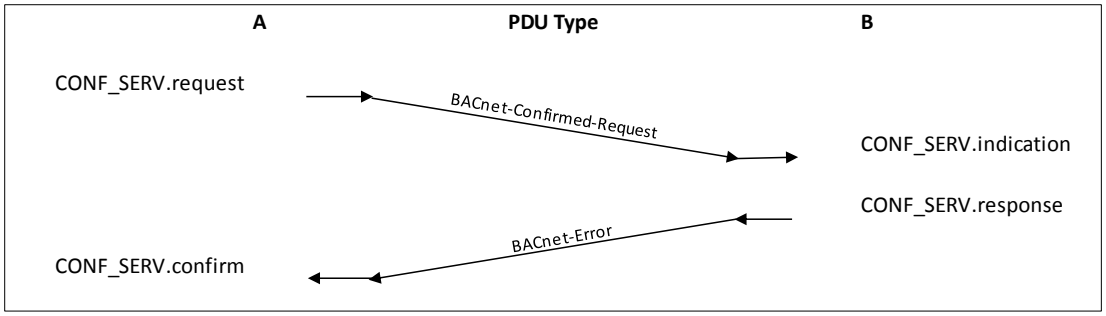


Abbildung 13: Bestätigter Dienst, Fehler

Die Norm stellt wesentlich mehr als die in dieser Arbeit abgebildeten Ablaufsequenzen vor. Für die folgenden Abschnitte werden die in Abbildung 9 bis Abbildung 13 dargestellten Szenarien aber für ausreichend erachtet. Die einzelnen PDUs werden in 4.1.3 erläutert.

Für das Zustandekommen einer Kommunikation genügen die in einer APDU enthaltenen Informationen allerdings noch nicht. Zusätzliche Netzwerk-, LLC- und MAC-bezogene Parameter, welche an die Vermittlungsschicht weitergegeben werden, sorgen dafür, dass u.a. Quell- und Zieladresse bekannt sind. Diese werden nicht in eine APDU integriert, sondern als Kontrollinformation (ICI, interface control information) an die Vermittlungsschicht übermittelt, welche diese (zum Teil) ihrerseits in den Header integriert.

4.1.3 Aufbau der Pakete auf der Anwendungsschicht

Die Zusammensetzung von Paketen auf der Anwendungsschicht geht aus Kapitel 20 der DIN EN ISO 16484-5:2012 [3] hervor.

Eine Protokolldateneinheit der Anwendungsschicht (englisch application layer protocol data unit, APDU) ist aus einem festen und einem variablen Bestandteil zusammengesetzt. Der feste Bestandteil der APDU (als APCI bezeichnet) enthält die Kontrollinformationen zu jeder APDU, während der variable Anteil dienstspezifische Inhalte umfasst. Die APCI fasst das zusammen, was in Abbildung 2 als Header der Anwendungs-, Sitzungs-, Darstellungs- und Transportschicht bezeichnet wurde. Der variable Teil entspricht der in Abbildung 2 als „Payload“ bezeichneten Komponente.

Fester Bestandteil (APCI)

Anders als in vielen anderen Protokollen ist der Aufbau des Headers eines BACnet-Paketes nicht statisch, sondern ändert sich je nach Art der übermittelten Information.

Es kann unterschieden werden in:

1) Protokolldateneinheit für bestätigte Anfragen (BACnet–Confirmed-Request-PDU)
2) Protokolldateneinheit für unbestätigte Anfragen (BACnet–Unconfirmed-Request-PDU)
3) Protokolldateneinheit für einfache Bestätigung (BACnet-SimpleACK-PDU)
4) Protokolldateneinheit für komplexe Bestätigung (BACnet-ComplexACK-PDU)
5) Protokolldateneinheit für Bestätigung bei segmentierten Nachrichten (BACnet-SegmentACK-PDU)
6) Protokolldateneinheit für Fehlermeldungen (BACnet-Error-PDU)
7) Protokolldateneinheit, in der das Abweisen von PDUs mitgeteilt wird (BACnet-Reject-PDU)
8) Protokolldateneinheit zum Abbrechen der Kommunikation (BACnet-Abort-PDU)

Um den Aufbau der einzelnen APCI verständlicher zu machen, wird jeweils ein Paket, welches mittels der GNU-GPL-Software Wireshark aufgezeichnet wurde, im Anhang A dargestellt [10]. Die Zusammensetzung der APCI bleibt für eine APDU derselben „Familie“ gleich. Die Bitwertigkeit für APDUs wird in dieser Arbeit – der Norm folgend – mit MSB (left) festgelegt.

BACnet–Confirmed-Request-PDU

Die Zusammensetzung einer BACnet-Confirmed-Request PDUs ist in Abbildung 14 dargestellt.

Für den Typus der PDU („PDU Type“) sind in einer BACnet-Confirmed-Request PDU die vier höchstwertigen Bit (Bit 7-4) des ersten Byte reserviert. Sind die vier Bit mit dem Wert 0 belegt, so kann die APDU eindeutig dem Typ BACnet-Confirmed-Request PDU zugeordnet werden.

Bit 3 des ersten Byte legt fest, ob es sich um eine segmentierte Nachricht handelt oder nicht. Die Belegung mit einer 0 steht für unsegmentierte Nachrichten, eine 1 für segmentierte.

Aus Bit 2 geht hervor, ob weitere Segmente folgen (MOR=1) oder ob es sich um das letzte (oder auch einzige) Segment einer Nachricht handelt (MOR=0).

Bit 1 definiert, ob segmentierte Antworten auf die Nachricht akzeptiert werden (SA=1) oder nur unsegmentierte (SA=0).

Bit 0 ist zurzeit ungenutzt und für eine spätere Verwendung durch die ASHRAE reserviert.

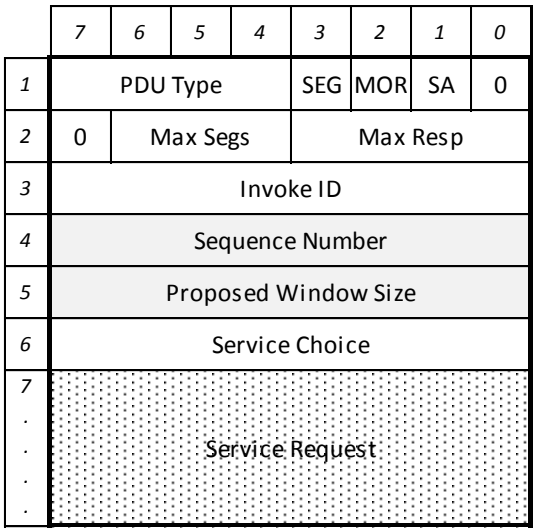


Abbildung 14: BACnet APDU Confirmed request

Das zweite Byte teilen sich die Informationen, wie viele Segmente maximal akzeptiert (Max Segs, Bit 4-6) werden und wie groß die Antwort-APDU maximal sein darf (Max Resp, Bit 0-3). Die Belegung dieser Bit spiegelt nicht – wie anzunehmen – die konkreten Werte wider, sondern ist folgendermaßen definiert:

Max Segs:	Max Resp:
000 = Anzahl an Segmenten nicht spezifiziert	0000 = bis zu 50 Oktette
001 = 2 Segmente	0001 = bis zu 128 Oktette
010 = 4 Segmente	0010 = bis zu 206 Oktette
011 = 8 Segmente	0011 = bis zu 480 Oktette
100 = 16 Segmente	0100 = bis zu 1024 Oktette
101 = 32 Segmente	0101 = bis zu 1476 Oktette
110 = mehr als 64 Segmente	0110 bis 1111: für spätere Verwendung durch ASHRAE reserviert

Byte Nr. 3 der Confirmed request-PDU entspricht der „InvokeID“. Die InvokeID ist eine Ganzzahl (0-255), welche die Transaktion identifiziert. Segmente können durch die ID eindeutig einer Transaktion zugeordnet werden, Antworten den ursprünglichen Anforderungsnachrichten.

Die Bytes 4 und 5 sind im APCI-Block nur dann vorhanden, wenn es sich um eine segmentierte Nachricht handelt. Die Sequenznummer („Sequence Number“) ist eine fortlaufende Nummer, welche inkrementell jedem Segment einer Transaktion zugeordnet wird. Dadurch können die APDUs wieder in der richtigen Reihenfolge zusammengefügt werden. „Proposed Window Size“ gibt an, wie viele Segmente einer Nachricht der Absender bereit ist zu übermitteln, bevor er ein ACK empfängt. Wird das Maximum erreicht, so wird der Absender in einen Wartezustand versetzt.

„Service Choice“ (Byte 6) definiert, welche Operation ausgeführt werden soll. Die Dienste und deren korrespondierende ID gehen aus Kapitel 21 der DIN EN ISO 16484-5:2012 [3] hervor.

Service Request enthält schließlich die Nutzinformationen und ist von variabler Größe.

Abbildung 14 des Anhangs A veranschaulicht den Aufbau.

BACnet–Unconfirmed-Request-PDU

Die Zusammensetzung einer BACnet–Unconfirmed-Request-PDU ist in Abbildung 15 dargestellt.

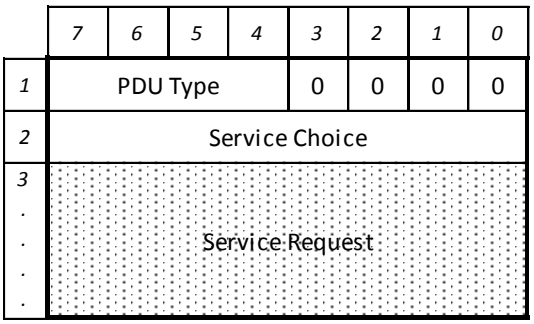


Abbildung 15: BACnet APDU Unconfirmed request

Diese Art von Nachricht wird durch die Belegung der ersten 4 Bit (7-4) mit 0001 identifiziert.

Bit 0, 1, 2 und sind für eine spätere Verwendung durch ASHRAE reserviert und sollen mit 0 belegt werden.

Service Choice (Byte 6) definiert, welche Operation ausgeführt werden soll. Die Dienste und deren korrespondierende ID gehen aus Kapitel 21 der DIN EN ISO 16484-5:2012 [3] hervor.

BACnet-SimpleACK-PDU

Der Aufbau der BACnet-SimpleACK-PDU ist in Abbildung 16 wiedergegeben.

	7	6	5	4	3	2	1	0
1	PDU Type				0	0	0	0
2	Original Invoke ID							
3	Service ACK Choice							

Abbildung 16: BACnet APDU SimpleACK

Diese Art von Nachricht wird durch die Belegung der ersten 4 Bit (7-4) mit 0010 identifiziert.

Bit 3,2,1 und 0 des ersten Byte sind mit 0 belegt für spätere Verwendung durch die ASHRAE reserviert.

Eine BACnet SimpleACK-PDU wird als positive Bestätigung zum Erhalt einer Nachricht zurückgesendet. Um eine Verbindung zwischen der Nachricht, deren Empfang bestätigt wird und dem ACK herzustellen, enthält das zweiten Byte die „Original Invoke ID“, also die ID der Originalnachricht.

Das letzte Byte (Service Ack Choice) definiert – wie für die vorangegangenen APDUs – die Art der Operation laut Kapitel 21 der DIN EN ISO 16484-5:2012 [3] und sollte dem „Service Choice“ aus der Anforderungsnachricht entsprechen.

APDUs vom Typ SimpleACK weisen stets eine Größe von 3 Byte auf und enthalten keinerlei zusätzlichen Informationen.

BACnet-ComplexACK-PDU

Der Aufbau der BACnet ComplexACK PDU entspricht dem in Abbildung 17 gezeigten.

Eine BACnet ComplexACK PDU entspricht dem PDU Type 3. Bit 7 bis 4 des ersten Byte sind demnach mit 0011 belegt. Bit 3 gibt Auskunft darüber, ob es sich um eine segmentierte Antwort handelt (1) oder nicht (0). Entspricht Bit 2 einer 1, so folgen noch weitere Segmente mit derselben Invoke ID, bei einer 0 handelt es sich um das letzte Segment. Für Bit 1 und 0 schreibt die Norm eine Belegung mit 0 vor (spätere Verwendung durch ASHRAE).

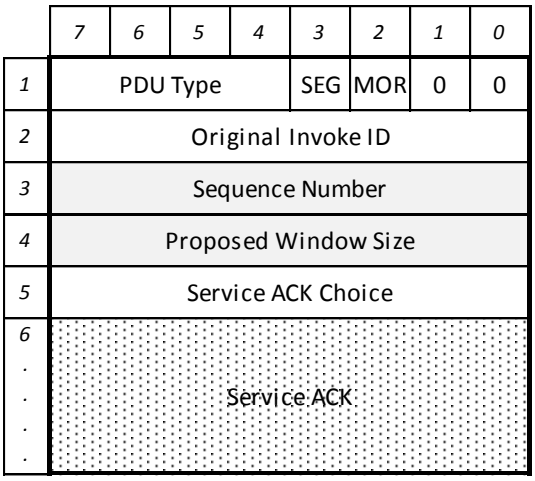


Abbildung 17: BACnet APDU ComplexACK

Die Original Invoke ID (Byte 2) gibt an, auf welche Invoke ID sich das im gegenständlichen Paket enthaltene ACK bezieht.

Die Angabe einer Sequenznummer (Sequence Number) und vorgeschlagenen Fenstergröße (Proposed Window Size, siehe BACnet-Confirmed-Request PDU) finden sich in einem ComplexACK-Paket nur im Falle einer segmentierten Nachricht.

Byte 5 (Service ACK Choice) sollte dem „Service Choice“ aus der korrespondierenden Anforderungsnachricht entsprechen.

BACnet-SegmentACK-PDU

Die APCI dieser Art von Nachricht entspricht der in Abbildung 18 dargestellten.

Dieser Typ von PDU wird dazu genutzt, um dem Sender das Ergebnis einer empfangenen PDU mitzuteilen (falls es sich um eine PDU gehandelt hat, welche eine Antwort erwartet).

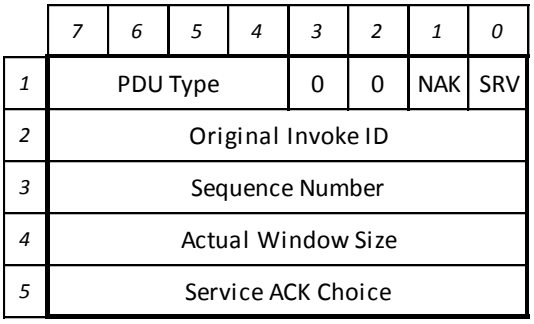


Abbildung 18: BACnet APDU SegmentACK

BACnet-SegmentACK-PDUs werden durch die Belegung der ersten 4 Bit des ersten Byte mit 0100 gekennzeichnet. Bit 3 und Bit 2 des ersten Byte sind für eine spätere Verwendung durch ASHRAE vorgemerkt und mit 0 zu belegen. Bit 1 (NAK, negati-

ve-ACK) gibt Informationen darüber, ob ein Segment außerhalb der richtigen Reihenfolge empfangen wurde. Bit 0 (SRV) ist TRUE, falls die SegmentACK-PDU als Antwort auf eine Confirmed Request PDU gesendet wurde. Falls die PDU als Antwort auf eine ComplexACK-PDU gesendet wurde, sollte das Bit auf FALSE gesetzt sein.

Byte 2 (Original Invoke ID), Byte 3 (Sequence Number) und Byte 5 (Service ACK Choice) entsprechen den Byte einer BACnet ComplexACK-PDU.

„Actual Window Size“ gibt an, wie viele Segmente mit derselben „Original Invoke ID) der Sender annimmt, bevor eine weitere SegmentACK-PDU übermittelt wird.

BACnet-Error-PDU

Eine BACnet-Error-PDU wird immer dann versandt, wenn eine Dienstanforderung fehlschlägt und gibt darüber Auskunft, warum die Dienstanforderung nicht ordnungsgemäß ausgeführt werden konnte. Abbildung 19 zeigt die Zusammensetzung einer BACnet-Error-PDU.

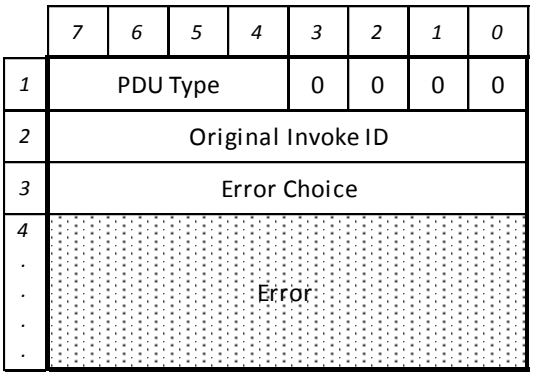


Abbildung 19: BACnet APDU Error

Eine BACnet-Error-PDU entspricht dem PDU Type 0101. Dieser ist durch Bit 7-4 des ersten Byte festgelegt. Bit 3 bis 0 sind für diese Art von PDU mit einer 0 belegt und für eine spätere Verwendung durch ASHRAE reserviert.

Die „Original Invoke ID“ (Byte 2) gibt an, auf welche Invoke ID sich die Fehlermeldung bezieht.

Byte 3 gibt Auskunft darüber, welcher Fehler beim Ausführen der Operation aufgetreten ist.

BACnet-Reject-PDU

Eine BACnet-Reject-APDU wird gesendet, falls eine Confirmed Request PDU abgelehnt wird. Ihr Aufbau wird in Abbildung 20 dargestellt.

	7	6	5	4	3	2	1	0
1	PDU Type				0	0	0	0
2	Original Invoke ID							
3	Reject Reason							

Abbildung 20: BACnet APDU Reject

Der PDU-Typ (PDU Type) einer BACnet-Reject-APDU ist mit 0110 festgelegt (Bit 7-4 des ersten Byte). Bit 3 bis 0 sind auch hier mit einer 0 belegt und für eine spätere Verwendung durch ASHRAE reserviert.

Byte 2 stellt wiederum den Bezug zur ursprünglichen Nachricht her (Original Invoke ID).

Byte 3 gibt Auskunft darüber, warum die Anforderung abgelehnt wurde. Die möglichen Gründe sind in Kapitel 21 der DIN EN ISO 16484-5:2012 [3] gelistet.

Eine BACnet-Reject-APDU hat eine statische Größe von 3 Byte.

BACnet-Abort-PDU

Als letzte Art von APDU definiert die Norm zum jetzigen Stand den Typ BACnet-Abort-PDU.

	7	6	5	4	3	2	1	0
1	PDU Type				0	0	0	0
2	Original Invoke ID							
3	Abort Reason							

Abbildung 21: BACnet APDU Abort

Eine BACnet-Abort-PDU kann durch eine Belegung der Bit 7-4 des ersten Byte mit 0111 identifiziert werden, während Bit 3 bis 0 wiederum mit 0 belegt sind.

Die Original Invoke ID bezieht sich auf die Ursprungsnachricht, auf welche mit der Abbruchnachricht geantwortet wird (Byte 2).

Die Gründe für den Abbruch sind in Kapitel 21 der DIN EN ISO 16484-5:2012 [3] gelistet und gehen aus Byte 3 hervor.

Variabler Bestandteil

Nachrichten der Typen „BACnet-Confirmed-Service-Request“, „BACnet-Unconfirmed-Service-Request“, „BACnet-Confirmed-Service-ACK und BACnet-Error weisen neben der APCI noch einen variablen Bestandteil auf, welcher Nutzinformationen enthält.

Die Kodierung der Bestandteile der APDU erfolgt nach der Notation ASN.1 (Abstract Syntax Notation One, welche in der Norm ISO 8824 definiert wird. Die Abstract Syntax Notation One stellt Elemente als abstrakte Datenstrukturen dar, die in der ISO 8824 als „Production“ bezeichnet wird.

Diese Datenstruktur soll anhand des Beispiels „I-Am-Request“ erklärt werden:

```
I-Am-Request::= SEQUENCE {
        iAmDeviceIdentifier             BACnetObjectIdentifier,
        maxAPDULengthAccepted           Unsigned,
        segmentationSupported           BACnetSegmentation,
        vendorID                        Unsigned
        }
```

„I-Am-Request“ stellt den Namen der Datenstruktur dar.

„SEQUENCE“ bedeutet, dass es sich um eine Sequenz von Attributen handelt, aus welchen sich die Datenstruktur zusammensetzt. Diese können eventuell mit „OPTIONAL“ gekennzeichnet sein.

Neben „SEQUENCE“ sind für BACnet noch die Strukturen „SEQUENCE OF“ (ein oder mehrere Attribute, „CHOICE“ (ein Attribut wird aus mehreren ausgewählt) und „ENUMERATED“ (Attribute, welchen ein bestimmter Wert zugewiesen wird) definiert.

Es folgen in geschweiften Klammern die Attribute mit den korrespondierenden Datentypen.

Der variable Anteil einer APDU wird durch ein Byte eingeleitet, welches folgendermaßen belegt ist:

7	6	5	4	3	2	1	0
Tag Number				Cl	Len/Val/Type		

Abbildung 22: BACnet APDU Tag

Bit 3, welches als „Class Bit“ bezeichnet wird, legt fest, ob es sich um einen anwendungsbezogenen (0) oder kontextbezogenen (1) Tag handelt. Anwendungsspezifische Tags sind vordefinierte Datentypen (z.B. Date, Time, Real, BACnetObjectIdentifier als ID eines BACnet-Objekts). Kontextbezogene Tags hingegen hängen vom Kontext ab, in welchem sie verwendet werden und sind im variablen Anteil der APDU näher spezifiziert.

Die Tag-Nummer (Tag Number, Bit 7 bis 4) gibt Auskunft über den Datentyp der APDU. Die Arten der anwendungsspezifischen Tags sind in Abschnitt 20.2.1.4 der DIN EN ISO 16484-5:2012 [3] definiert. Durch die Verwendung von lediglich 4 Bit können maximal 16 Tags kodiert werden. Bei Verwendung einer Tag-Nummer, welche größer als 15 ist, wird die Tag-Nummer auf 1111 gesetzt und im nachfolgenden Byte spezifiziert.

Die Bits 2 bis 0 (Length/Value/Type) variieren, je nachdem, ob es sich um einen primitive oder einen zusammengesetzten Datentyp handelt. Die Definition der beiden Datentypen ist 20.2.1.3 der DIN EN ISO 16484-5:2012 [3] definiert.

Auf den variablen Bestandteil der APDU wird in dieser Arbeit nicht näher eingegangen, für ein Basisverständnis werden die in diesem Abschnitt angeschnittenen Punkte für ausreichend erachtet.

4.1.4 Kennzeichnung sensibler Daten

Nachdem in den vorangegangen Abschnitten die Arten von Nachrichten und der Ablauf beim Austausch derselben erläutert wurden, ist die Basis geschaffen, um die Möglichkeit einer Kennzeichnung von Paketen, welche sensible Daten enthalten, zu untersuchen und aufzuzeigen.

In Abschnitt 2.5.4 wurden zwei Konzepte der Identifizierung sensibler Daten kurz vorgestellt. Es handelt sich dabei um inhaltsbasierte und kontextbasierte Lösungen. Da der variable Anteil einer APDU Informationen enthält, welche je nach Art der

durchgeführten Operationen sowohl in ihrer Zusammensetzung, als auch in ihrer Große stark schwanken können, erscheint eine Lösung, welche auf diesen Anteil abzielt, kaum handhabbar. Aus diesem Grund wird von der Untersuchung einer inhaltsbasierten Lösung abgesehen. Stattdessen wird ein Ansatz verfolgt, welcher Pakete durch eine bestimmte Kennzeichnung identifiziert. Diese Vorgehensweise wird allgemein als „Tagging“ bezeichnet.

Aus der in Abschnitt 4.1.3 erfolgten Analyse der APCI der in der DIN EN ISO 16484-5:2012 [3] definierten APDU kann gefolgert werden, dass im Grunde nur zwei Arten von APDU effektiv (möglicherweise sensible) Nutzdaten enthalten können. Es sind dies die Protokolldateneinheiten für bestätigte Anfragen (BACnet–Confirmed-Request-PDU) und Protokolldateneinheiten für unbestätigte Anfragen (BACnet–Unconfirmed-Request-PDU). Bei genauer Betrachtung der APCI der beiden Arten von Protokolldateneinheiten fällt auf, dass die Norm für Bit 0 des ersten Byte eine Belegung mit 0 vorsieht. Dieses Bit könnte für eine Kennzeichnung genutzt werden. Das „sensibel“-Tag könnte in diesem Fall durch die Belegung dieses Bit mit einer 1 realisiert werden. Gegen diese Variante spricht, dass die Norm dieses Bit allerdings nicht als verfügbar deklariert, sondern es für eine spätere Verwendung durch das ASHRAE reserviert hält.

Ähnlich verhält sich die Situation beim Ansatz, sensible Daten mittels Setzen von Bit 7 des ersten Byte auf 1. Bit 7 des ersten Byte entspricht dem höchstwertigen Bit des PDU-Type. Effektiv genutzt wird im Moment nur der Bitbereich 0000 bis 0111, welcher für die in Abschnitt 4.1.3 eingeführten PDU-Typen verwendet wird. Doch auch in diesem Fall ist der Bereich 1000 bis 1111 für eine spätere Verwendung reserviert und Bit 7 damit nicht für ein „sensibel“-Tag verfügbar.

Eine Realisierung der Kennzeichnung mittels des ersten Byte ist damit zwar möglich, aber nicht normkonform.

Die einzige Komponente, welche den beiden APDUs außer den bereits untersuchten gemein ist, ist der „Service Choice“-Teil. Dafür ist sowohl in der BACnet-Confirmed-Request- als auch in einer BACnet Unconfirmed-Request-PDU jeweils ein Byte vorgesehen. Dieses Byte erlaubt es, maximal 256 (2^8) Arten von Nachrichten darzustellen. Zurzeit sind in der Norm 29 Arten von Confirmed-Request-Nachrichten definiert, für Unconfirmed-Request-Nachrichten hingegen 9. Da mit 5

Bit 32 (2^5) Zustände dargestellt werden können, wäre damit der gesamte bereits adressierte Bereich abgedeckt. Dadurch ist es möglich, 3 Bit (Bit 7, 6 und 5) dieses Byte für andere Zwecke einzusetzen. In Anbetracht möglicher zukünftiger Weiterentwicklungen und für die Wahrung der Flexibilität sollten allerdings nicht alle 3 Bit verwendet werden. Für ein simples Tagging, welches ausschließlich der binären Unterscheidung „sensibel“/“nicht sensibel“ dient, ist ein einziges Bit ausreichend. Für das Tagging wird die Verwendung von Bit 7 vorgeschlagen.

Um den in dieser Arbeit vorgeschlagenen Ansatz anhand eines Beispiels zu verdeutlichen: die Confirmed-Request-Nachricht „readProperty“ wird in der DIN EN ISO 16484-5:2012 [3] mit 00001100 (12) definiert. Soll es sich nun nicht mehr um den Vorgang „readProperty“, sondern um ein „readSensitiveProperty“ handeln, so könnte dies durch die Belegung von Bit 7 mit einer 1 gekennzeichnet werden, in diesem Fall also 10001100.

4.2 Vermittlungsschicht

Nachdem im letzten Abschnitt die BACnet-Anwendungsschicht untersucht wurde, soll in in diesem Kapitel die BACnet-Vermittlungsschicht unter die Lupe genommen werden. Die Vermittlungsschicht ist für die Vermittlung von Paketen zwischen Netzwerken zuständig (siehe auch Abschnitt 2.1.1). Anders ausgedrückt: würden sich alle Endpunkte im selben Netzwerksegment befinden, könnte auf die Vermittlungsschicht verzichtet werden.

BACnet bietet auf der Vermittlungsschicht ein verbindungsloses Protokoll an. Dies bedeutet, dass kein Verbindungsaufbau vor dem Austausch von Nachrichten stattfindet. Die Entscheidung fiel aus Gründen der Minimierung des Datenflusses auf ein verbindungsloses Protokoll, um auch bei der Verwendung von Übertragungsmedien mit einem niedrigen Durchsatz akzeptable Kommunikationszeiten zu erzielen.

4.2.1 Arten von Nachrichten (Primitive)

Wie für die BACnet-Anwendungsschicht definiert die DIN EN ISO 16484-5:2012 [3] in Kapitel 6.1 auch für die Vermittlungsschicht Primitive, welche unterteilt werden können in „Anforderungen“ (requests) und „Anzeigen“. Es können insgesamt drei Arten von Primitiven unterschieden werden:

- N-UNITDATA.request: Anforderung. Diese enthalten folgende Parameter: destination_address (Zieladresse), data (NSDU, Network Service Data Unit, im Grunde eine komplette APDU, welche von der Anwendungsschicht kommt), network_priority (Priorität), data_expecting_reply (wird eine Antwort auf die Nachricht erwartet?), security_parameters (Sicherheitsparameter)
- N-UNITDATA.indication: Anzeige (Bekanntmachung) eines Datenpaketes
- N_REPORT.indication: Anzeige (Bekanntmachung) einer Fehlermeldung

4.2.2 Aufbau der Pakete auf der Vermittlungssschicht

Wie auch die Nachrichten der Anwendungsschicht, haben die Nachrichten auf der BACnet-Vermittlungsschicht keine feste Zusammensetzung. Abbildung 23 stellt dar, aus welchen Bestandteilen sich eine NPDU zusammensetzen kann.

Die einzelnen Felder haben dabei folgende Bedeutungen: Version gibt die Version des verwendeten BACnet-Protokolls an. Aktuell wird die Version 1 eingesetzt. Bei Control handelt es sich um ein Kontrollbyte, dem entnommen werden kann, welche weiteren Komponenten in der NPDU vorkommen. Die Bedeutung der einzelnen Bit des Kontrollbyte ist in Abbildung 24 zusammengefasst. DNET ist das Zielnetzwerk des Paketes. DLEN gibt die Länge der Empfängeradresse der NPDU an (0 bedeutet, dass es sich um eine Broadcast-Nachricht handelt). Die Länge der Empfängeradresse unterscheidet sich je nach dem auf der Sicherungsschicht verwendeten Protokoll. DADR ist die MAC-Adresse des Empfängers der Nachricht. SNET gibt die Adresse des Quellnetzwerkes an. SLEN definiert die Länge der Absenderadresse (auch hier wird diese durch das Sicherungsschicht-Protokoll festgelegt). SADR entspricht der MAC-Adresse des Absenders. Hop Count ist ein mit dem Wert 255 initialisierter Zähler, welcher beim Passieren eines Routers jeweils um 1 vermindert wird. Erreicht er 0, so wird das Paket verworfen. Message Type legt fest, um welche Art von NPDU es sich handelt. Vendor ID entspricht einer eindeutigen ID, welche an den Hersteller eines Gerätes durch das ASHRAE vergeben wurde. APDU entspricht einem Paket, welches von der Anwendungsschicht durchgereicht wurde.

Die DIN EN ISO 16484-5:2012 [3] definiert in Abschnitt 6.4 insgesamt 20 Arten von Nachrichten auf der Vermittlungsschicht. Von diesen 20 Nachrichtentypen wird an dieser Stelle der Aufbau von 10 Arten dargestellt, welche im Anhang B als Wireshark-Caputures abgebildet sind [10].

Version	1 Byte
Control	1 Byte
DNET	2 Byte
DLEN	1 Byte
DADR	variabel
SNET	2 Byte
SLEN	1 Byte
SADR	variabel
Hop Count	1 Byte
Message Type	1 Byte
Vendor ID	2 Byte
ADPU	N Byte

Abbildung 23: Mögliche Bestandteile NPDU [3]

Control Byte							
7	6	5	4	3	2	1	0

Bit	Wert	Beschreibung
7	0	Feld Message Type nicht Bestandteil (Nachricht auf Anwendungsschicht)
	1	Feld Message Type folgt (Nachricht auf Vermittlungsschicht)
6	0	Reserviert
5		*Spezifikation zum Ziel der Nachricht*
	0	Felder DNET, DLEN, DADR und Hop Count nicht Bestandteil
	1	DNET, DLEN und Hop Count Bestandteil. Ist das Feld DLEN=0, dann handelt es sich um eine Broadcast-Nachricht und DADR ist nicht vorhanden. Falls DLEN>0 ist, so spezifiziert dies die Länge des DADR-Feldes
4	0	Reserviert
3		*Spezifikation zur Quelle der Nachricht*
	0	Felder SNET, SLEN und SADR nicht Bestandteil
	1	SNET, SLEN und SADR Bestandteil. Die Belegung SLEN=0 ist ungültig. Falls SLEN>0 ist, so spezifiziert dies die Länge des SADR-Feldes
2	0	Es handelt sich um eine andere Nachricht als eine BACnet-Confirmed-PDU, ein Segment einer BACnet-ComplexACK-PDU oder eine Nachricht auf der Vermittlungsschicht, welche eine Antwort erwartet.
	1	Eine BACnet-Confirmed-Request-PDU, ein Segment einer BACnet-ComplexACK-PDU oder eine Nachricht auf der Vermittlungsschicht erwarten eine Antwort
1, 0		*Spezifikation der Priorität der Nachricht*
	11	Lebenswichtig, höchte Priorität
	10	Kritischer Zustand eines Geräts
	01	Dringende Nachricht
	00	Normale Nachricht/normale Priorität

Abbildung 24: Bedeutung Bit Kontrollbyte [3]

Who-Is-Router-To-Network

Version	**00000001**	1 Byte
Control	**10000000**	1 Byte
Message Type	**00000000**	1 Byte
DNET	**var**	2 Byte

Abbildung 25: Who-Is-Router-To-Network-PDU

Who-Is-Router-To-Network-Nachrichten („Message Type“ 0) werden genutzt, um jenen Router zu finden, welcher Nachrichten in ein definiertes Zielnetzwerk vermitteln kann.

I-Am-Router-To-Network

Version	**00000001**	1 Byte
Control	**10000000**	1 Byte
Message Type	**00000001**	1 Byte
DNET	**var**	2 Byte
DNET	**var**	2 Byte

Abbildung 26: I-Am-Router-To-Network-PDU

I-Am-Router-To-Network („Message Type“ 1) ist eine Nachricht, welche eine Liste aller Netzwerke enthält, die dieser Router erreichen kann (Liste von DNET). Diese Nachricht wird in der Regel als Broadcast gesendet.

I-Could-Be-Router-To-Network

Version	**00000001**	1 Byte
Control	**10000000**	1 Byte
Message Type	**00000010**	1 Byte
DNET	**var**	2 Byte
Performance Index	**var**	1 Byte

Abbildung 27: I-Could-Be-Router-To-Network-PDU

Eine I-Could-Be-Router-To-Network-Nachricht wird („Message Type“ 2) wird als Antwort auf eine Nachricht des Typs 1 gesendet (Who-Is-Router-To-Network). Das „Could Be“ rührt daher, dass die PTP-Verbindung zum Netzwerk im Moment der Anfrage möglicherweise nicht aufgebaut ist. Das Feld „Performance Index“ ist ein Indikator für die Güte und Performance der Verbindung zum betreffenden Netzwerk.

Reject-Message-To-Network

Version	**00000001**	1 Byte
Control	**10000000**	1 Byte
Message Type	**00000011**	1 Byte
Rejection Reason	**var**	1 Byte
DNET	**var**	2 Byte

Abbildung 28:Reject-Message-To-Network-PDU

Reject-Message-To-Network („Message Type“ 3) wird versendet, wenn die Übermittlung einer Nachricht in ein Netzwerk abgelehnt wird. Der Grund für das Abweisen der Übermittlung der Nachricht in das Zielnetz DNET wird im Byte „Reject Reason“ codiert und ist in Abschnitt 6.4.4 der DIN EN ISO 16484-5:2012 [3] definiert.

Router-Busy-To-Network

Version	**00000001**	1 Byte
Control	**10000000**	1 Byte
Message Type	**00000100**	1 Byte
DNET	**var**	2 Byte
DNET	**var**	2 Byte

Abbildung 29: Router-Busy-To-Network-PDU

Router-Busy-To-Network-Nachrichten („Message Type“ 4) werden üblicherweise als Broadcast gesendet. Anhand dieser Nachrichten wird den Zielnetzwerken DNET

mitgeteilt, dass der Router ausgelastet ist und die Nachrichtenübermittlung deshalb eingeschränkt ist.

Router-Available-To-Network

Version	**00000001**	1 Byte
Control	**10000000**	1 Byte
Message Type	**00000101**	1 Byte
DNET	**var**	2 Byte
DNET	**var**	2 Byte

Abbildung 30: Router-Available-To-Network-PDU

Die Nachrichten der Art Router-Available-To-Network („Message Type“ 5) werden ebenso als Broadcast-Nachricht versendet. Sie teilen den Zielnetzwerken DNET mit, dass der Router (wieder) zur Verfügung steht.

Initialize-Routing-Table

Version	**00000001**	1 Byte
Control	**10000000**	1 Byte
Message Type	**00000110**	1 Byte
Data Portion	**var**	

Abbildung 31: Initialize-Routing-Table-PDU

Initialize-Routing-Table-PDUs („Message Type“ 6) stößt die Initialisierung der Routingtabelle eines Routers an. Im „Data Portion“-Block ist festgelegt, wie viele Ports in diesem Paket definiert werden, gefolgt von der eigentlichen Spezifikation. Ist die Anzahl der Ports mit 0 festgelegt, so gilt die Nachricht als Aufforderung an den Empfänger, seine Routingtabelle zurückzuliefern.

Initialize-Routing-Table-Ack

Version	**00000001**	1 Byte
Control	**10000000**	1 Byte
Message Type	**00000111**	1 Byte
Data Portion	**var**	

Abbildung 32: Initialize-Routing-Table-Ack-PDU

Dieser Typus von PDUs wird durch den „Message Type“ 7 identifiziert und informiert darüber, dass die Routingtabelle geändert wurde. Enthält die Nachricht zudem einen Datenblock (Data Portion), so handelt es sich um die Antwort auf die Anforderung der Routingtabelle. Die Routingtabelle selbst wird im Datenblock übermittelt.

Establish-Connection-To-Network

Version	**00000001**	1 Byte
Control	**10000000**	1 Byte
Message Type	**00001000**	1 Byte
DNET	**var**	2 Byte
Termination Time Value	**var**	1 Byte

Abbildung 33: Establish-Connection-To-Network-PDU

Eine Establish-Connection-To-Network-PDU („Message Type“ 8) ist als Anforderung an einen Halbrouter (Half Router) zu sehen, eine Punkt-zu-Punkt-Verbindung mit dem angegebenen Zielnetzwerk aufzubauen. „Termination Time“ gibt hierbei an, wie lange die Verbindung für die Übermittlung von Nachrichten aufrecht erhalten bleiben soll.

Disconnect-Connection-To-Network

Version	**00000001**	1 Byte
Control	**10000000**	1 Byte
Message Type	**00001001**	1 Byte
DNET	**var**	2 Byte

Abbildung 34: Disconnect-Connection-To-Network-PDU

Disconnect-Connection-To-Network-PDUs fordern einen Router auf, eine bestehende Punkt-zu-Punkt-Verbindung zum Netzwerk DNET abzubauen.

Andere NPDUs

Auf der BACnet-Vermittlungsschicht sind in der DIN EN ISO 16484-5:2012 [3] folgende weitere Nachrichten definiert:

„Message Type“ 10: Challenge-Request (spezifiziert in Kapitel 24 der Norm)
„Message Type“ 11: Security Payload (spezifiziert in Kapitel 24 der Norm)
„Message Type“ 12: Security Response (spezifiziert in Kapitel 24 der Norm)
„Message Type“ 13: Request-Key-Update (spezifiziert in Kapitel 24 der Norm)
„Message Type“ 14: Update-Key-Set (spezifiziert in Kapitel 24 der Norm)
„Message Type“ 15: Update-Distribution-Key (spezifiziert in Kapitel 24 der Norm)
„Message Type“ 16: Request-Master-Key (spezifiziert in Kapitel 24 der Norm)
„Message Type“ 17: Set-Master-Key (spezifiziert in Kapitel 24 der Norm)
„Message Type“ 18: What-Is-Network-Number (spezifiziert in Kapitel 24 der Norm)
„Message Type“ 19: Network-Number-Is (spezifiziert in Kapitel 24 der Norm)

4.2.3 Filtermöglichkeiten

Auf der Vermittlungsschicht werden Pakete ausgetauscht, welche zwischen Netzwerken vermittelt werden sollen. Auf dieser Ebene können die Nachrichten potentiell nach mehreren Kriterien gefiltert werden:

- Message Type
- Art der Nachricht (Unicast, Broadcast)
- Quellnetzwerk
- Zielnetzwerk

Im Hinblick auf Data Leakage und die in den vorangegangenen Abschnitten beleuchteten Aspekte kann eine Filterung der Pakete nach Quell- und/oder Zielnetzwerk bewirken, dass Pakete von/zu einem bestimmten Netzwerk verworfen bzw. blockiert werden können.

4.3 BACnet/IP

Einen Sonderstatus im BACnet-Protokollstapel nimmt BACnet/IP ein. BACnet/IP wird im Anhang J der DIN EN ISO 16484-5:2012 [3] definiert. Sollen zwei Gebäudenetzwerke, welche durch ein dazwischenliegendes IT-Netzwerk getrennt sind,

miteinander kommunizieren können, so bietet sich die Verwendung von BACnet/IP an. BACnet/IP kommt allerdings oftmals auch in der direkten Kommunikation zwischen Geräten, welche BACnet/IP unterstützen, zum Einsatz. IP wird in der Gebäudeautomation aus Performancegründen in Kombination mit UDP (normalerweise Port 47808) eingesetzt, da der Overhead bei der Verwendung von TCP wesentlich höher liegen würde. Im Unterschied zu konventionellen Netzwerken wird BACnet/IP nicht auf den ISO/OSI-Schichten 3 bzw. 4 angesiedelt, sondern als neue Zwischenschicht zwischen der BACnet-Vermittlungsschicht und der darunterliegenden Sicherungsschicht eingeführt. Diese Zwischenschicht wird als BACnet Virtual Link Layer (BVLL) bezeichnet.

4.3.1 Arten von Nachrichten

Die DIN EN ISO 16484-5:2012 [3] definiert im Abschnitt J.2 die Arten von Nachrichten, welche der BVLL anbietet. Ähnlich APCI auf der BACnet-Anwendungsschicht und NPCI auf der Vermittlungsschicht sind die Kontrollinformationen in einem Header enthalten. Dieser wird als BACnet Virtual Link Control (BVLC) bezeichnet. In der Norm sind 13 Nachrichten definiert

1. BVLC-Result
2. Write-Broadcast-Distribution-Table
3. Read-Broadcast-Distribution-Table
4. Read-Broadcast-Distribution-Table-Ack
5. Forwarded-NPDU
6. Register-Foreign-Device
7. Read-Foreign-Device-Table
8. Read-Foreign-Device-Table-Ack
9. Delete-Foreign-Device-Table-Entry
10. Distribute-Boradcast-To-Network
11. Original-Unicast-NPDU
12. Original-Broadcast-NPDU
13. Secure-BVLL

Im nächsten Abschnitt wird die Bedeutung der Nachrichten beschrieben und der Aufbau der Pakete dieser Nachrichten dargestellt.

4.3.2 Aufbau der Pakete

Die ersten 4 Byte der BVLC-Pakete geben Auskunft über den Typ der Nachricht, um welche Funktion es sich handelt und die Gesamtlänge des Pakets. Das erste Byte ist dabei stets mit 1000001 belegt. Diese Bitfolge steht für BACnet/IP. Das folgende Byte entspricht der Binärdarstellung der Nachrichtennummerierung in Abschnitt 4.3.1. Anschließend sind zwei Byte für die Längenangabe des gesamten Paketes reserviert.

BVLC-Result

Diese Nachricht dient dazu, das Ergebnis einer vorher übermittelten Nachricht mitzuteilen. Der Aufbau geht aus Abbildung 35 hervor.

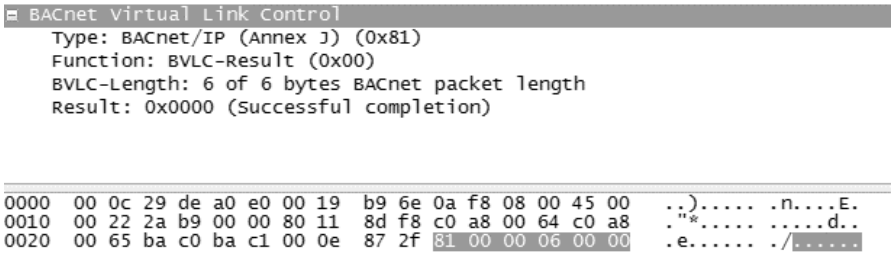


Abbildung 35: BVLC-Result

Die letzten zwei Byte enthalten hier die Information, wie das Ergebnis auf welche Nachricht lautet. Dabei kann es sich um folgende Arten handeln:

- 00000000 00000000: Erfolgreich abgeschlossen
- 00000000 00010000: Write-Broadcast-Distribution-Table-NAK
- 00000000 00200000: Read-Broadcast-Distribution-Table-NAK
- 00000000 00110000: Register-Foreign-Device NAK
- 00000000 01000000: Read-Foreign-Device-Table NAK
- 00000000 01010000: Delete-Foreign-Device-Table-Entry NAK
- 00000000 01100000: Distribute-Broadcast-To-Network NAK

Write-Broadcast-Distribution-Table

Eine Write-Broadcast-Distribution-Table wird gesendet, um die Broadcast Distribution Table (BDT) in einem BACnet Broadcast Management Device BBMD zu aktualisieren.

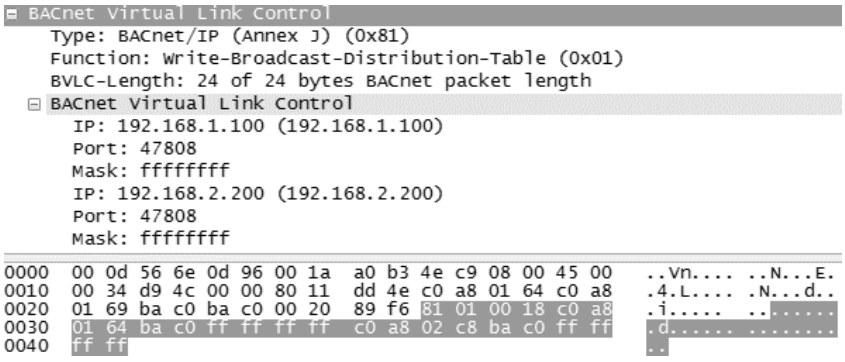


Abbildung 36: Write-Broadcast-Distribution-Table

Die Byte ab Byte fünf enthalten die Informationen, welche in die BDT geschrieben werden sollen, wobei es sich immer um ein Vielfaches von 10 Byte handelt (6 Byte für die BBMD-Adresse, 4 weitere Byte definieren, wie die Nachrichten im IP-Subnetz verteilt werden sollen).

Read-Broadcast-Distribution-Table

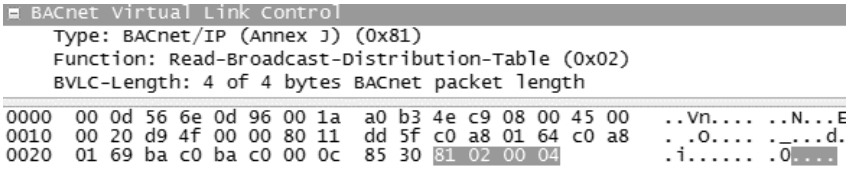


Abbildung 37: Read-Broadcast-Distribution-Table

Nachrichten der Art Read-Broadcast-Distribution-Table sind stets 4 Byte groß und werden versendet, um die Übermittlung der BDT anzustoßen.

Read-Broadcast-Distribution-Table-Ack

Diese Nachricht ist die Antwort auf eine Read-Broadcast-Distribution-Table-Anforderung und sendet dem Anforderer die BDT zurück. Die Tabelle selbst wird in 10-Byte-Blöcken übermittelt, wobei die Einträge wiederum 6 Byte BBMD-Adresse und umfassen und 4 Byte für die Art der Verteilung reserviert sind.

Forwarded-NPDU

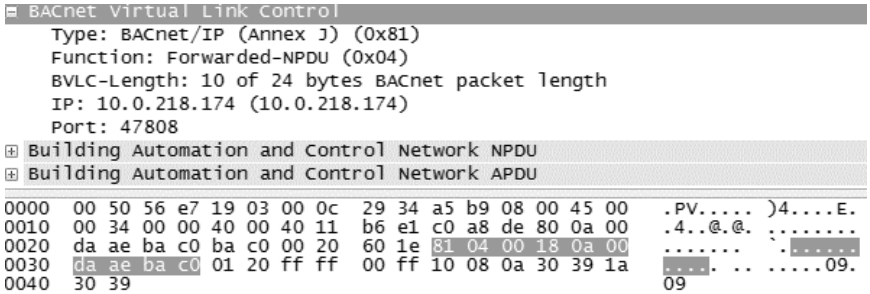


Abbildung 38: Forwarded-NPDU

Forwarded-NPDUs werden für das Weiterleiten von Unicast-Nachrichten an Geräte verwendet, welche an BBMDs oder an dem Router als erreichbar bekannte Empfänger gesendet werden. Byte fünf bis zehn sind für die Angabe der Quell-IP-Adresse (4 Byte) und des Port (2 Byte, Standard ist Port 47808) reserviert. Anschließend folgt die ursprüngliche NPDU.

Register-Foreign-Device

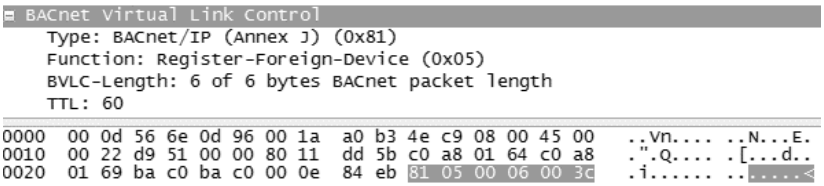


Abbildung 39: Register-Foreign-Device

Mit dieser Art von Nachricht kann ein Gerät ein BBMD dazu auffordern, in die BDT aufgenommen zu werden, um Broadcast-Nachrichten empfangen zu können. Byte fünf und sechs sind für einen TTL-Timer reserviert, welcher in Sekunden angegeben ist. Dieser Timer gibt an, innerhalb welcher Zeit sich das Gerät erneut am BBMD registrieren muss, um nicht aus der BDT entfernt zu werden.

Read-Foreign-Device-Table

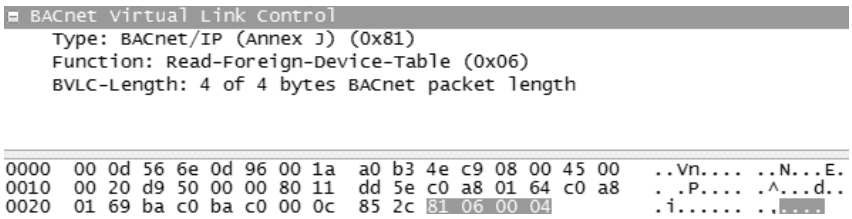


Abbildung 40: Read-Foreign-Device-Table

Read-Foreign-Device-Table stößt – ähnlich Read-Broadcast-Distribution-Table-Nachrichten – die Übermittlung der Foreign-Device-Tabelle (FDT) an und hat eine feste Größe von 4 Byte.

Read-Foreign-Device-Table-Ack

Read-Foreign-Device-Table-Ack ist die Antwort auf eine Read-Foreign-Device-Table und übermittelt die FDT an den Absender. Die übermittelten Einträge umfassen jeweils 10 Byte, wobei 6 Byte auf die B/IP-Adresse des registrierten Gerätes entfallen, 2 Byte auf die Zeitangabe, innerhalb der sich das Gerät neu registrieren muss und 2 Byte auf die noch verbleibende Zeit bis zur nächsten Registrierung des Gerätes.

Delete-Foreign-Device-Table-Entry

Delete-Foreign-Device-Table-Entry-Nachrichten dienen dazu, um Einträge aus der FDT zu löschen. In Byte fünf bis zehn wird die B/IP-Adresse angegeben, welche aus der Tabelle entfernt werden soll.

Distribute-Broadcast-To-Network

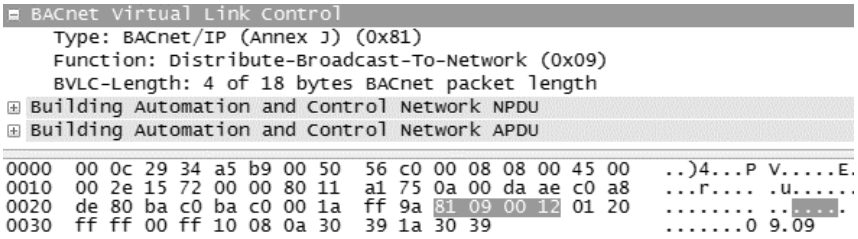


Abbildung 41: Distribute-Broadcast-To-Network

Mittels dieser Nachricht wird ein Broadcast an sämtliche in der BDT enthaltenen Einträge ausgesendet, wobei die ursprüngliche Nachricht in die Antwortnachricht integriert wird.

Original-Unicast-NPDU

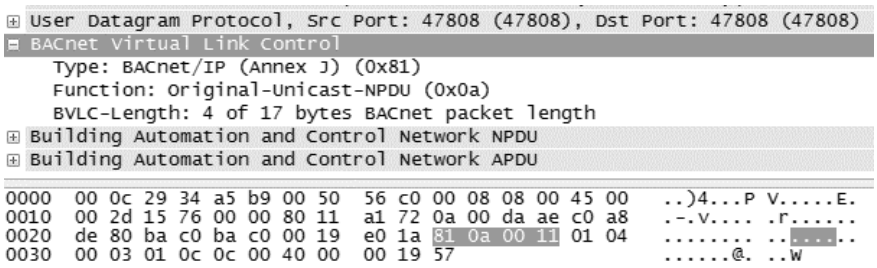


Abbildung 42: Original-Unicast-NPDU

Eine Unicast-PDU wird zur zielgerichteten Auslieferung einer NPDU an einen bestimmten Empfänger genutzt. Die NPDU ist in der Nachricht gekapselt.

Original-Broadcast-NPDU

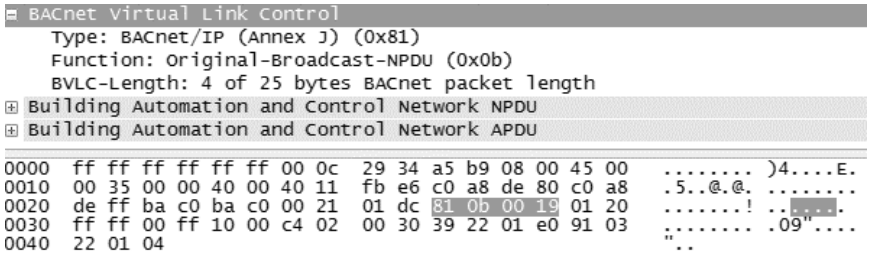


Abbildung 43: Original-Broadcast-NPDU

Original-Broadcast-NPDUs unterscheiden sich von Distribute-Broadcast-To-Network-Nachrichten darin, dass Original-Broadcast-NPDU-Nachrichten von B/IP-Geräten und Routern nur an das Netzwerksegment übermittelt werden, in dem sich der Absender selbst befindet.

Secure-BVLL

Secure-BVLL werden in Kapitel 24 der DIN EN ISO 16484-5:2012 [3] definiert. Die sichere Nachricht wird direkt in die BVLL-Nachricht integriert.

4.3.3 Filtermöglichkeiten

Von den im Anhang J der DIN EN ISO 16484-5:2012 [3] definierten und in Abschnitt 4.3.2 aufgezeigten Nachrichten werden neun für die Kommunikation zwischen BBMDs und deren Verwaltung benötigt. Lediglich vier Nachrichten (Forwarded-NPDU, Original-Unicast-NPDU, Original-Broadcast-NPDU und Secure-BVLL) enthalten neben den Kontroldaten auch NPDUs, welche potentiell ihrerseits wieder sensible Informationen in APDUs umfassen können.

Werden Nachrichten der Art Secure-BVLL außen vor gelassen, so sollte es demnach unter dem Aspekt der Data Leakage Protection möglich sein, Forwarded-NPDU-, Original-Unicast-NPDU- und Original-Broadcast-NPDU-Nachrichten zu filtern.

4.4 Untersuchung der bestehenden Implementierung des BFR

Für den BACnet-Protokollstapel wurde auf sourgeforge.net unter Public Domain Lizenz eine Implementierung des so genannten „BACnet Firewall Router“ (BFR) veröffentlicht. Der Quellcode des BFR wurde in der Programmiersprache C++ von Joel J. Bender verfasst [11].

Neben der Implementierung des BACnet-Vermittlungsschicht-Protokolls umfasst der Code auch die Implementierung der BACnet-Anwendungsschicht, sowie jene von BACnet/IP.

Der BFR kann sowohl als BACnet Router als auch als BBMD fungieren. Zusätzlich zu dieser „Grundfunktionalität“ bietet der BFR verschiedene Möglichkeiten der Paketfilterung an und ist in der auf sourceforge abrufbaren Version bereits sehr weit fortgeschritten. Die Konfiguration der Filterung kann komfortabel mittels einer in XML-Notation verfassten Datei eingelesen werden, welche durch einen Parser in Parameter für den BFR umgewandelt wird. Die Filterung kann hierbei nach ver-

schiedenen Kriterien erfolgen, wobei auch eine Kombination mehrerer Kriterien (auch auf verschiedenen Protokollschichten) möglich ist.

In diesem Kapitel wird untersucht, inwieweit die Implementierung die in den vorangegangen Abschnitten Filtermöglichkeiten abdeckt. Die Testumgebung wurde folgendermaßen eingerichtet:

- Drei virtuelle Maschinen, Betriebssystem Fedora 20
- Kommunikation mittels BACnet/IP
- Netzwerk 1: 192.168.100.0/24, BACnet Netzwerk 20
- Netzwerk 2: 192.168.200.0/24, BACnet Netzwerk 10
- Virtuelle Maschine 1 stellt den Client dar, IP-Adresse der Maschine 192.168.100.10
- Virtuelle Maschine 3 wird als Server betrachtet, IP-Adresse der Maschine 192.168.200.20
- Virtuelle Maschine 2 stellt den BFR dar, welcher die Routing-Funktionalität zwischen dem Client- und dem Servernetzwerk zur Verfügung stellt. Der BFR ist mit zwei virtuellen Netzwerkkarten ausgestattet, welche durch die IP-Adressen 192.168.100.20 und 192.168.200.10 angesprochen werden können. Netzwerk 20 ist über die NIC 1 (192.168.100.20) erreichbar, Netzwerk 10 über NIC 2. Der BFR wird im Folgenden so konfiguriert, dass bei Erweiterung der reinen Routing-Funktionalität auch eine Paketfilterung zwischen den beiden Netzwerken stattfinden kann.

Die in XML verfasste Datei zum Einlesen der Basiskonfiguration des BFR sieht folgendermaßen aus:

```
<BFR>
        <UDP address="192.168.100.20" server="net20" />
        <Debug client="net20" server="net20x" prefix="[20] " />
        <BIP client="net20x" server="net20y" />

        <UDP address="192.168.200.10" server="net10" />
        <Debug client="net10" server="net10x" prefix="[10] " />
        <BIP client="net10x" server="net10y" />

        <Router>
                <Adapter client="net20y" net="20" />
```

```
            <Adapter client="net10y" net="10" />
        </Router>
    </BFR>
```

Wie für XML-Dateien üblich weist die Konfiguration eine hierarchische Struktur auf, welche die einzelnen Protokollschichten widerspiegelt. Einzelne Elemente werden durch die Schlüsselwörter server/client zueinander zugeordnet, die Schichten dadurch untereinander verbunden. Für die oben angegebene, zur Konfiguration der Testumgebung verwendete XML-Datei, ergibt sich:

- UDP address="192.168.100.20" IP-Adresse 192.168.100.20. Es wird UDP verwendet (Standardport falls nicht anders definiert 47808)
- *Debug client=net20"* dient dazu, auf dieser IP-Adresse ankommenden Pakete in der Konsole auszugeben (reine Debugging-Zwecke). Das Schlüsselwort prefix definiert, dass Pakete, welche auf NIC 192.168.100.20 ankommen, als Folge von Hexadezimalwerten mit einer vorangestellten [20] ausgegeben werden
- BIP client="net20x": Es wird BACnet/IP verwendet (BVLL, siehe Abschnitt 4.3)
- Im Block Router wird festgelegt, welche Netzwerke über welchen Adapter erreichbar sind. Im vorliegenden Beispiel sind dies Netzwerk 20 über den Adapter mit der IP-Adresse 192.168.100.20 und Netzwerk 10 über den Adapter 192.168.200.10). Dadurch können Pakete zwischen den beiden Netzwerken geroutet werden.

Zur Veranschaulichung der Funktionsweise wird nun ein Paket vom Client (über den BFR, der als Router fungiert) an den Server übermittelt (Einfacher Unconfirmed „Who Is“ Request).

Die Ausgabe auf der Konsole liefert (Hex-Darstellung):

```
[20]  192.168.100.10 -> 192.168.100.20 : 81.0A.00.12.01.20.00.0A.06.C0.A8.C8.14.BA.C0.FF.10.08.

[10]  192.168.200.20 <- null : 81.0A.00.11.01.08.00.14.06.C0.A8.64.0A.BA.C0.10.08.
```

Das Paket wird also zum Netzwerk 20 weitergeroutet.

Wireshark loggt die Pakete folgendermaßen mit:

No.	Time	Source	Destination	Protocol	Length	Info
24	42.577825000	192.168.200.10	192.168.200.20	BACnet-APDU	59	Unconfirmed-REQ who-Is
25	42.577663000	192.168.100.10	192.168.100.20	BACnet-APDU	60	Unconfirmed-REQ who-Is

Abbildung 44: Wireshark-Capture Übermittlung Paket ohne Filterung

Client zu BFR/Router:

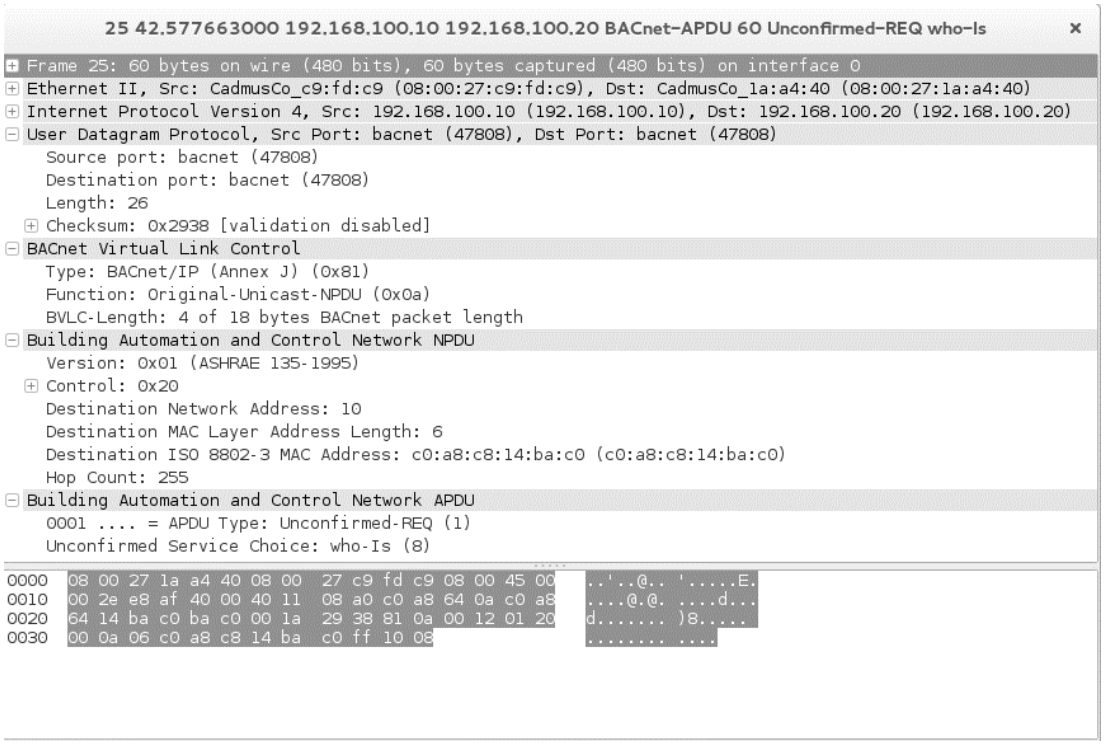


Abbildung 45: Detail des am BFR eingehenden Paketes aus Netzwerk 20

BFR/Router zu Server:

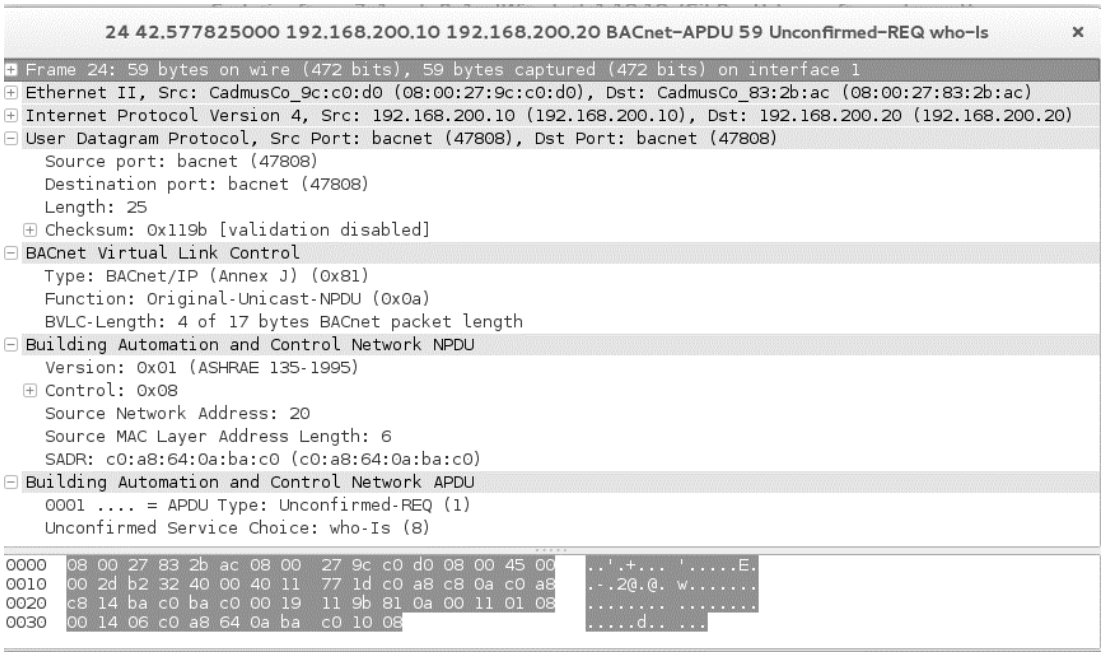


Abbildung 46: Detail des am BFR ausgehenden Paketes zum Netzwerk 10

In den folgenden Abschnitten soll die Basiskonfiguration nun um Filtermöglichkeiten auf den diversen Schichten erweitert werden. Sämtliche zur Konfiguration

verwendeten, in XML verfassten Dateien, sind zusätzlich im Anhang C - Konfiguration 1 bis Konfiguration 8 - dargestellt.

4.4.1 Filtermöglichkeiten auf der Anwendungsschicht

Der BFR bietet eine Vielzahl von Möglichkeiten zur Filterung an. Prinzipiell kann eine Filterung durch das Schlüsselwort Filter definiert werden. Wie in der IT üblich, werden die Regeln durch ein Accept (Paket darf passieren) oder Reject (Paket darf nicht passieren) festgelegt.

In der aktuellen Version stehen folgende Filtermöglichkeiten zur Verfügung (Keywords, welche für die Konfiguration verwendet werden können, sind in Klammern angeführt):

- Protokolldateneinheit für bestätigte Anfragen (CONFIRMED-REQUEST)
- Protokolldateneinheit für unbestätigte Anfragen (UNCONFIRMED)
- Protokolldateneinheit für einfache Bestätigung (SIMPLEACK)
- Protokolldateneinheit für komplexe Bestätigung (COMPLEXACK)
- Protokolldateneinheit für Bestätigung bei segmentierten Nachrichten (SEGMENTACK)
- Protokolldateneinheit für Fehlermeldungen (ERROR)
- Protokolldateneinheit, in denen das Abweisen von PDUs mitgeteilt wird (REJECT)
- Protokolldateneinheit zum Abweisen der Kommunikation (ABORT)

Der BFR deckt somit die gesamte Palette der im Standard DIN EN ISO 16484-5:2012 [3] definierten und in Abschnitt 4.1.3 angeführten Arten von Nachrichten auf der Anwendungsschicht ab.

Für die Veranschaulichung der Filterung auf der Anwendungsschicht wird die Konfiguration aus Abschnitt 4.4 erweitert und dasselbe Paket vom Client an den Server (via BFR) übermittelt. Die neue XML sieht folgendermaßen aus:

```
<BFR>
    <UDP address="192.168.100.20" server="net20" />
    <Debug client="net20" server="net20x" prefix="[20] " />
    <BIP client="net20x" server="net20y" />

    <UDP address="192.168.200.10" server="net10" />
    <Debug client="net10" server="net10x" prefix="[10] " />
```

```
        <BIP client="net10x" server="net10y" />

        <Filter client="net10y" server="net10yf">
            <Downstream>
                <Allow />
                <Reject function="UNCONFIRMED" />
            </Downstream>
            <Upstream>
                <Allow />
                <Reject function="UNCONFIRMED" />
            </Upstream>
        </Filter>

        <Router>
            <Adapter client="net20y" net="20" />
            <Adapter client="net10yf" net="10" />
        </Router>
    </BFR>
```

Durch die Filter-Regel werden auf Netzwerkseite 192.168.100.20 APDUs für unbestätigte Anfragen (Unconfirmed Request) verworfen. Die Schlüsselwörter „Upstream“ und „Downstream“ dienen der Spezifizierung, ob Pakete, welche den Protokollstapel von unten nach oben (Upstream) oder von oben nach unten (Downstream) durchlaufen, gefiltert werden sollen. Für den vorliegenden Anwendungsfall nimmt diese Unterscheidung keinen Einfluss auf das Ergebnis, da es sich hierbei um eine reine Inspektion mit anschließendem Routen der Pakete handelt.

Die Konsolenausgabe des BFR ergibt mit der angepassten Konfiguration:

```
[20]  192.168.100.10 -> 192.168.100.20 : 81.0A.00.12.01.20.00.0A.06.C0.A8.C8.14.BA.C0.FF.10.08.
```

Wireshark-Capture:

No.	Time	Source	Destination	Protocol	Length	Info
48	110.853410000	192.168.100.10	192.168.100.20	BACnet-APDU	60	Unconfirmed-REQ who-Is

Abbildung 47: Wireshark-Capture Filterung von „Unconfirmed Request“-Nachrichten

Daraus ist ersichtlich, dass das Paket nun nicht mehr an die Zieladresse weitergeleitet wird.

Ebenso wie für Pakete des Typs „Unconfirmed Request“ kann die Filterung für sämtliche Paketarten auf der Anwendungsschicht durchgeführt werden.

4.4.2 Filtermöglichkeiten auf der Vermittlungsschicht

Nachdem in Kapitel 4.4.1 festgestellt wurde, dass die Filterung auf der Anwendungsebene möglich ist, sollen nun die Filteroptionen auf der Vermittlungsschicht untersucht werden.

In Abschnitt 4.2.3 wurde vorgeschlagen, dass eine Appliance zur Paketfilterung auf Vermittlungsschicht-Ebene zumindest eine Filterung nach Quell- und/oder Zielnetzwerk bieten soll. Um dies zu gewährleisten, muss die Konfiguration aus dem vorangehenden Abschnitt abgeändert werden, sodass nun nicht mehr nach dem Pakettyp auf Anwendungsebene, sondern auf der Vermittlungsschicht gefiltert wird. Der BFR ordnet jedem Adapter (NIC) ein Netzwerk zu, welches über ihn erreichbar ist. Da in den Testbeispielen BACnet/IP verwendet wird, kann nach IP-Adresse und Port gefiltert werden, um Pakete von/an bestimmte Netzwerke zu verwerfen.

Zur Veranschaulichung wird das Testpaket mit drei unterschiedlichen Konfigurationen an das Netzwerk übergeben (die Zeitpunkte beziehen sich jeweils auf den Zeitpunkt, welcher im Wireshark-Caputure in Abbildung Abbildung 48 ersichtlich ist):

- Testfall 1 (Zeitpunkt 4.93): keine Filterung
- Testfall 2 (Zeitpunkt 79.3): Filterung mittels <Reject source="C0-A8-64-0A-BA-C0"/>. (IP-Adresse 192.168.100.10, Port 47808) Sinnvollerweise muss für diesen Testfall die Konfigurationsdatei derart angepasst werden, dass bei Eintreffen von Paketen der zu filternden Quelladresse diese direkt am eingehenden NIC gefiltert werden. Soll die Filterung nicht nur auf Port 47808 beschränkt sein, so bietet der BFR auch die Möglichkeit, die Quelladresse in Standardnotation „192.168.100.10“ anzugeben. Auch die Spezifizierung eines Netzwerkes in der Form „192.168.100.0/24“ ist möglich.
- Testfall 3 (Zeitpunkt 105.1): Filterung auf NIC 2 mittels <Reject destination="C0-A8-C8-14-BA-C0"/>. (192.168.200.20:47808).
- Testfall 4 (Zeitpunkt 124.7): Filterung auf NIC 2 mittels <Reject destination="C0-A8-FA-14-BA-C0"/> (192.168.250.20:47808. Es liegt keine Zieladresse in diesem Netzwerk, Pakete müssen also passieren können. Dieser Testfall dient der reinen Veranschaulichung der Funktionalität der Filterung).

Testfall 1 liefert auf dem BFR folgende Debug-Ausgabe:

[20] 192.168.100.10 -> 192.168.100.20 : 81.0A.00.12.01.20.00.0A.06.C0.A8.C8.14.BA.C0.FF.10.08.

[10] 192.168.200.20 <- null : 81.0A.00.11.01.08.00.14.06.C0.A8.64.0A.BA.C0.10.08.

Testfall 2:

[20] 192.168.100.10 -> 192.168.100.20 : 81.0A.00.12.01.20.00.0A.06.C0.A8.C8.14.BA.C0.FF.10.08.

Testfall 3:

[20] 192.168.100.10 -> 192.168.100.20 : 81.0A.00.12.01.20.00.0A.06.C0.A8.C8.14.BA.C0.FF.10.08.

Testfall 4:

[20] 192.168.100.10 -> 192.168.100.20 : 81.0A.00.12.01.20.00.0A.06.C0.A8.C8.14.BA.C0.FF.10.08.

[10] 192.168.200.20 <- null : 81.0A.00.11.01.08.00.14.06.C0.A8.64.0A.BA.C0.10.08.

Wireshark Capture:

No.	Time	Source	Destination	Protocol	Length	Info
5	4.931264000	192.168.200.10	192.168.200.20	BACnet-APDU	59	Unconfirmed-REQ who-Is
6	4.931052000	192.168.100.10	192.168.100.20	BACnet-APDU	60	Unconfirmed-REQ who-Is
35	79.360813000	192.168.100.10	192.168.100.20	BACnet-APDU	60	Unconfirmed-REQ who-Is
50	105.130276000	192.168.100.10	192.168.100.20	BACnet-APDU	60	Unconfirmed-REQ who-Is
59	124.722672000	192.168.200.10	192.168.200.20	BACnet-APDU	59	Unconfirmed-REQ who-Is
60	124.722522000	192.168.100.10	192.168.100.20	BACnet-APDU	60	Unconfirmed-REQ who-Is

Abbildung 48: Wireshark-Capture Testfälle Filterung NPDU

Die Filterung nach Quell- und Zielnetzwerk ist damit möglich.

Der BFR bietet allerdings nicht nur diese Option, sondern eine Vielzahl an weiteren Möglichkeiten. Konkret sind dies sämtliche im Standard DIN EN ISO 16484-5:2012 [3] definierten und in Abschnitt 4.2.2 angeführten Arten von Nachrichten auf der Vermittlungsschicht, welche durch folgende Schlüsselwörter mittels Accept/Reject function gefiltert werden können:

- WHO-IS-ROUTER-TO-NETWORK oder WHOS-RTN
- I-AM-ROUTER-TO-NETWORK oder IM-RTN
- I-COULD-BE-ROUTER-TO-NETWORK oder I-CLD-BE-RTN
- REJECT-MESSAGE-TO-NETWORK oder REJ-MTN
- ROUTER-BUSY-TO-NETWORK oder RBTN
- ROUTER-AVAILABLE-TO-NETWORK oder RATN
- INITIALIZE-ROUTING-TABLE oder IRT
- INITIALIZE-ROUTING-TABLE-ACK oder IRT-ACK
- ESTABLISH-CONNECTION-TO-NETWORK oder ECTN

- DISCONNECT-CONNECTION-TO-NETWORK oder DCTN

4.4.3 Filtermöglichkeiten bei Einsatz von BACnet/IP (BVLL)

Als letzter Punkt der Untersuchung der bestehenden Implementierung des BFR soll nun BACnet/IP betrachtet werden. In Kapitel 4.3.3 wird als Mindestanforderung eine Filterung von Forwarded-NPDU-, Original-Unicast-NPDU- und Original-Broadcast-NPDU-Nachrichten gefordert.

Ähnlich den zuvor erstellten Konfigurationsdateien kann auch für BACnet/IP eine Filterung mittels function eingerichtet werden. Kongruent mit untersuchten Filtermöglichkeiten auf der Anwendungs- und Vermittlungsschicht wird wieder dasselbe Paket vom Client zum Server übermittelt.

Die Filterung wird mittels <Reject function="ORIGINAL-UNICAST-NPDU" /> durchgeführt, da es sich im Testfall um eine Nachricht vom Typ Original-Unicast NPDU handelt.

Die Konsolenausgabe des BFR ergibt mit der angepassten Konfiguration:

[20] 192.168.100.10 -> 192.168.100.20 : 81.0A.00.12.01.20.00.0A.06.C0.A8.C8.14.BA.C0.FF.10.08.

Wireshark-Capture:

No.	Time	Source	Destination	Protocol	Length	Info
5	8.576627000	192.168.100.10	192.168.100.20	BACnet-APDU	60	Unconfirmed-REQ who-Is

Abbildung 49: Wireshark-Capture Filterung von „Original-Unicast“-Nachrichten

Durch die Filterung werden BACnet/IP-Pakete vom Typ „Unconfirmed Request“ damit nicht mehr zum Zielnetzwerk weitergeroutet. Eine Filterung von „Forwarded-NPDU“-, und „Original-Broadcast-NPDU“-Nachrichten ist in analoger Weise möglich.

4.4.4 Anmerkungen zur bestehenden Implementierung des BFR

Eine allgemeine Anmerkung zum BFR an dieser Stelle soll hier nicht unerwähnt bleiben: die Schlüsselwörter, welche für die Filterung eingesetzt werden können, werden nicht als String interpretiert. Vielmehr wird aus dem für die Konfiguration verwendeten Schlüsselwort ein Hashcode berechnet, welcher anschließend auf Übereinstimmung mit in der Klasse BFRFilter.cpp gelisteten Keywords geprüft wird. In mehr als einem Testfall gab es Abweichungen zwischen dem aus dem Schlüsselwort berechneten Hashcode und dem in der bestehenden Implementierung gelisteten Hashcode (beispielsweise wird ein „UNCONFIRMED-REQUEST“ mit einem Hashcode von 0xa17e3788 in der Klasse BFRFilter.cpp gelistet. Eine Neuberech-

nung des Hashcode ergibt allerdings 0xFFFFFFFFFF0E0FEB. Dadurch kann es zu „Unmatched Keyword“-Fehlern kommen. Für die vorliegende Arbeit wurden die Hashcodes punktuell neu berechnet, für den konkreten Anwendungsfall sollten diese allerdings vollständig neu berechnet werden.

4.5 Ansätze zur Erweiterung des BFR auf der Anwendungsschicht

Die Filterung der Pakete auf der Anwendungsschicht ist bereits für eine breite Palette von Optionen möglich. Sämtliche in DIN EN ISO 16484-5:2012 [3] definierten PDU-Typen sowie die dazu angebotenen Dienste (readProperty, writeProperty…) können bereits als Filterkriterium verwendet werden.

Die bestehende Implementierung wird nun um Filtermöglichkeiten für die in Abschnitt 4.1.4 vorgeschlagene Kennzeichnung (Tagging) erweitert. Dadurch soll die Option zur Filterung von getaggten Paketen umgesetzt werden. Für die vorliegende Arbeit beschränkt sich die Realisierung auf den Dienst „readSensitiveProperty“. Weitere Dienste können allerdings problemlos in ähnlicher Weise ergänzt werden.

Ein erster Ansatz, der im Zuge dieser Arbeit verfolgt wurde, war jener, dass ein Client seine Anfrage (z.B. readProperty) via BFR an der Server sendet. Der Server, welcher als „sensibel“ eingestuft ist, antwortet seinerseits mit einem „readSensitiveProperty“. Der BFR filtert nun diese diese „readSensitiveProperty“-Nachrichten dahingehend, dass diese nur auf bestimmten Interfaces des BFR erlaubt sind. Diese Vorgehensweise bringt jedoch einige Nachteile mit sich, welche sie für einen praktischen Einsatz als nicht akzeptabel darstellen: Beim Einsatz von BBMD werden Nachrichten weiterhin auch an Netzwerksegmente, auf welchen eigentlich nur sensible Daten verkehren sollen, weitergegeben. Eine weitere große Einschränkung ist jene, dass dem Server die „readSensitiveProperty“-Nachricht bekannt sein muss und er damit selbst das Taggen der Pakete durchführen muss. Das Konzept ist dadurch nicht auf bereits bestehende Geräte anwendbar.

Diese Probleme können durch einen alternativen Ablauf beseitigt werden. Das Konzept ist ähnlich der von der IEEE im Standard 802.1Q definierten VLAN-Technologie für Ethernet-Pakete. Bei diesem Ansatz wird das Netzwerk dadurch segmentiert, dass der Datenverkehr sämtlicher Geräte, welche Zugriff auf sensible Informationen haben, direkt bei der Übergabe an das Netzwerk getaggt werden. Der BFR, welcher unmittelbar hinter den Client geschaltet übernimmt das Tagging. Auf

ihrem Weg zum Server werden die Pakete nun auf dedizierten Ports weitergeleitet, auf welchen ausschließlich getaggte Pakete erlaubt sind. Der Ablauf geht aus Abbildung 50 hervor.

Das Tagging selbst wird am ersten BFR dadurch durchgeführt, dass eine „Filter"-Regel „Allow" für „Confirmed-Request" und „Unconfirmed" gesetzt wird. Bei der Überprüfung der Pakete werden diese gleichzeitig getaggt. Dies wird durch die zusätzlichen Codezeilen

```
int TagPos = (pduData[0] & 0x08) ? 5 : 3;
pduData[TagPos]=pduData[TagPos]+0x80;
```

der Switch-Anweisung der Filterung für „Confirmed-Requests" bzw.

```
pduData[1] = pduData[1] + 0x80;
```

für „Unconfirmed-Requests" in der Quelldatei BFRFilter.cpp erreicht.

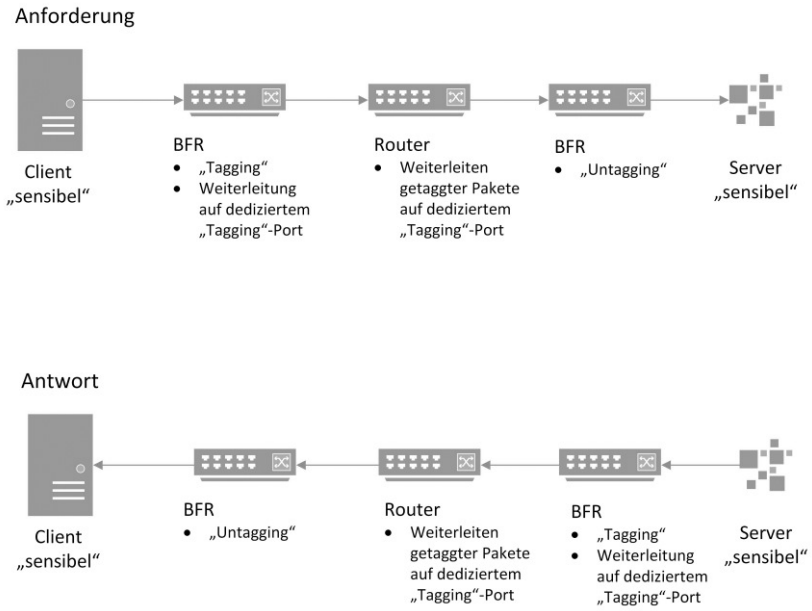


Abbildung 50: BACnet APDU Confirmed request

Die Schlüsselwörter für die Filterung werden in einer in XML verfassten Datei durch die Quelldatei BFRParser.cpp identifiziert. Für die Überprüfung, ob ein Schlüsselwort als Filterkriterium zulässig ist, wird jedoch nicht die Zeichenfolge, sondern ein daraus berechnet Hashcode berechnet. Die Funktion zur Berechnung des Hashwertes kann der Quelldatei BFRFilter.cpp entnommen werden, in welcher auch das Keyword bzw. dessen daraus berechneter Hashwert erfasst ist. Die Berechnung des Hashwertes für das Schlüsselwort „READSENSITIVEPROPERTY" ergibt

0x1D32ABAE. Für die Filterung wird nun dieser Hashcode, sowie der Wert des Byte, welches den Dienst definiert, herangezogen. Für ein laut Abschnitt 4.1.4 getaggtes „ReadProperty“-Paket, welches durch den Tag zu einem „ReadSensitiveProperty“ wird, entspricht dieser Wert 140 (Binär 10001100, siehe Abschnitt 4.1.4). Der BFR kann nun durch Reject function="READSENSITIVEPROPERTY" angewiesen werden, Pakete des Dienstes „ReadSensitiveProperty“ zu filtern. Das eingehende Paket selbst wird daraufhin überprüft, ob das „ServiceChoice“-Byte dem im Paar {(Hashcode, Wert „ServiceChoice“-Byte)} (im konkreten Fall {0x1D32ABAE, 140}) definierten Wert entspricht und gegebenenfalls verworfen. Für den praktischen Einsatz sollte die Erweiterung der Schlüsselwort-Tabelle unter dem Aspekt der Usability dahingehend umgesetzt werden, dass ein einziges Keyword „Sensitive“ eingeführt wird (Hascode 0x2A28B3C8), welchem als rechter Partner der Wert des ungetaggten „ServiceChoice“-Bit + 128 zugewiesen wird.

Zur Veranschaulichung wird die Testumgebung um zwei weitere Server erweitert, welche unter der IP-Adresse 192.168.220.40 (Server 2) und 192.168.220.50 (Server 3) erreichbar sind. Damit einhergehend muss der BFR um einen NIC 3 erweitert werden, welcher mit der statischen IP-Adresse 192.168.220.30 konfiguriert wird. Diese wird in den <Router>-Block der Konfigurationsdatei aufgenommen, damit Pakete an das zusätzliche Netzwerk weitergeleitet werden können (die vollständige Konfigurationsdatei kann Anhang C - Konfiguration 8 entnommen werden.

Als Filter-Regel wird nun für das Zielnetzwerk 192.168.200.0/24 definiert):

```
<Filter client="net10y" server="net10yf">
        <Downstream>
                <Allow function="READSENSITIVEPROPERTY" />
        </Downstream>
        <Upstream>
                <Allow function="READSENSITIVEPROPERTY" />
        </Upstream>
</Filter>
```

Für das Zielnetzwerk 192.168.220.0/24 wird die Filterung folgendermaßen festgelegt:

```
<Filter client="net30y" server="net30yf">
        <Downstream>
            <Allow function="READSENSITIVEPROPERTY" destination="192.168.220.50" />
```

```
        <Reject function="READSENSITIVEPROPERTY" destination="192.168.220.40" />
    </Downstream>
    <Upstream>
        <Allow function="READSENSITIVEPROPERTY" destination="192.168.220.50" />
        <Reject function="READSENSITIVEPROPERTY" destination="192.168.220.40" />
    </Upstream>
</Filter>
```

Mit der gewählten Konfiguration ist es nun möglich, Pakete des Dienstes „ReadSensitiveProperty“ an alle Teilnehmer im Netzwerk 192.168.200.0/24 weiterzuleiten. Im Netzwerk 192.168.220.0/24 werden „ReadSensitiveProperty“-Pakete an den Server 3 übermittelt, jene an Server 2 hingegen verworfen.

Zur Überprüfung der Funktionsweise der Filterung wird nun je ein getaggtes Paket an jeden Server übermittelt.

Für den Testfall 1 (Zeitpunkt 51.3 der Abbildung 51), der Übermittlung eines getaggten Paketes an den Server 1 (192.168.200.20), ergibt die Konsolenausgabe des BFR:

```
[20] 192.168.100.10 -> 192.168.100.20 : 81.0A.00.1B.01.20.00.0A.06.C0.A8.C8.14.BA.C0.FF.02.03.C1.8C.0C.02.00.00.6F.19.4B.
[10] 192.168.200.20 <- null : 81.0A.00.1A.01.08.00.14.06.C0.A8.64.0A.BA.C0.02.03.C1.8C.0C.02.00.00.6F.19.4B.
```

Das Paket wird also vom BFR an den Server 1 weitergeleitet.

Für Testfall 2 (Zeitpunkt 68.3 der Abbildung 51) wird ein getaggtes Paket an Server 2 (192.168.220.40) gesendet. Die Konsolenausgabe liefert lediglich

```
[20] 192.168.100.10 -> 192.168.100.20 : 81.0A.00.1B.01.20.00.1E.06.C0.A8.DC.28.BA.C0.FF.02.03.C1.8C.0C.02.00.00.6F.19.4B.
```

Richtigerweise wird das Paket nicht an Server 2 weitergeleitet.

Im dritten und letzten Testfall (Zeitpunkt 83.4 der Abbildung 51) wird dasselbe Paket nun an den Server 3 (192.168.220.50) gesendet, was zu folgendem Ergebnis auf dem BFR führt:

```
[20] 192.168.100.10 -> 192.168.100.20 : 81.0A.00.1B.01.20.00.1E.06.C0.A8.DC.32.BA.C0.FF.02.03.C1.8C.0C.02.00.00.6F.19.4B.
[30] 192.168.220.50 <- null : 81.0A.00.1A.01.08.00.14.06.C0.A8.64.0A.BA.C0.02.03.C1.8C.0C.02.00.00.6F.19.4B.
```

Die „ReadSensitiveProperty“-APDU wird korrekt an Server 3 weitergegeben.

Das Wireshark-Capture des Testvorganges sieht folgendermaßen aus (da getaggte „ReadSensitiveProperty“-Pakete von Wireshark nicht als solche identifiziert werden können, werden diese als „unknown service“ bezeichnet):

No.	Time	Source	Destination	Protocol	Lengtl	Info
28	51.365586000	192.168.100.10	192.168.100.20	BACnet-APDU	69	Confirmed-REQ unknown service[193]
29	51.365791000	192.168.200.10	192.168.200.20	BACnet-APDU	68	Confirmed-REQ unknown service[193]
42	68.386180000	192.168.100.10	192.168.100.20	BACnet-APDU	69	Confirmed-REQ unknown service[193]
52	83.425511000	192.168.220.30	192.168.220.50	BACnet-APDU	68	Confirmed-REQ unknown service[193]
53	83.425363000	192.168.100.10	192.168.100.20	BACnet-APDU	69	Confirmed-REQ unknown service[193]

Abbildung 51: Wireshark-Capture Testfälle getaggte „ReadSenstive-Property“-APDU

Die Erweiterung des BFR um die Filterung getaggter Pakete ist somit funktionstüchtig und erledigt die Filterung zuverlässig. Es wird vorausgesetzt, dass sämtliche Geräte zur Vermittlung von Paketen den in dieser Arbeit vorgeschlagenen Tag erkennen können (zutreffend, falls es sich um den erweiterten BFR handelt) und der Tag am letzten Router vor dem Server entfernt wird, sodass dem Client eine ungetaggte Nachricht zugestellt wird.

Ein ähnlicher Ansatz wurde bereits 2012 in der Arbeit von Wenzel, Kahler und Rist [12] verfolgt. Dieser unterscheidet sich allerdings darin, dass die Konfiguration der Filterung des BFR nach Netzwerken durchgeführt wird und nicht auf der Anwendungsebene. Dadurch gestaltet sich die Konfiguration der einzelnen BFR recht umständlich. Ein zusätzlicher Vorteil der Filterung auf der Anwendungsebene ist jener, dass der Datenverkehr nun nicht mehr über einen zentralen BFR (im Paper von Wenzel, Kahler und Rist [12] als “first level router” bezeichnet) laufen muss, sondern auch andere Netzwerktopologien als baumförmige (z.B. Vermaschung) eingesetzt werden können.

4.6 Ansätze zur Erweiterung des BFR auf der Vermittlungsschicht

Die bestehende Implementierung des BFR wird auf der Vermittlungsschicht für erschöpfend erachtet. Sämtliche in Abschnitt 4.2.3 geforderten Filtermöglichkeiten sind bereits abgedeckt. Selbiges gilt für die in Abschnitt 4.3.3 für BACnet/IP vorgeschlagenen Optionen.

5 Evaluierung

Nachdem in Kapitel 2 eine Einführung in die verschiedenen Komponenten gegeben wurde, welche bei der Untersuchung von Data Leakage in der Gebäudeautomation eine Rolle spielen, hat die Analyse in Kapitel 3 zum Schluss geführt, dass auch in diesem Bereich Handlungsbedarf besteht. Ansätze für eine Eindämmung – oder bestenfalls sogar Vermeidung – eines Datenabflusses wurden in Kapitel 4 aufgezeigt. Auch wenn keine konkreten Zahlen vorliegen, welche das wirtschaftliche Ausmaß von Data Leakage in der Gebäudeautomation beziffern, so muss davon ausgegangen werden, dass das Schadenspotential auch in diesem Bereich als relevant einzustufen ist.

Doch wie sind die in dieser Arbeit verfolgten Ansätze zu bewerten? Grundlegend muss sicherlich erwähnt werden, dass das in 4.1.4 eingeführte Tagging nicht dazu geeignet ist, sämtliche Sicherheitsprobleme, welche es in der Gebäudeautomation noch lösen gilt, zu beseitigen. Wird allerdings der reine Aspekt ‚Data Leakage' betrachtet, so kann durch das Tagging kombiniert mit dem Routen von Paketen über ein eigenes, als vertrauenswürdig geltendes Netzwergsegment, durchaus eine Verbesserung der bestehenden Situation, in welcher sensible Daten im gesamten Netzwerk exponiert sind, erzielt werden. Daten in Bewegung folgen auf diese Art und Weise ausschließlich klar definierten Wegen. Ein unbefugter Gewinn von sensiblen Informationen wird erschwert - allein schon aus dem Grund, dass die „Angriffsfläche" für einen unerwünschten Datenabfluss erheblich reduziert wird und sich nur mehr auf ausgewählte Segmente beschränkt.

Oftmals scheitert die Einführung einer Sicherheitslösung an den Investitionskosten, welche mit deren Umsetzung verbunden sind. Die in dieser Arbeit aufzeigten Optionen können ohne größere Investitionen umgesetzt werden. Die Geräte auf der Feld- und Managementebene bleiben von den Anpassungen völlig unberührt, Adaptionen sind ausschließlich auf der Automationsebene angesiedelt. Trotz der begrenzten Zusatzkosten kann für ein nennenswertes Plus an Sicherheit gesorgt werden.

Die Skalierbarkeit eines bestehenden Systems erfährt durch die Tagging-Lösung keine Einbußen. Im Gegenteil: die Tagging-Lösung selbst lässt noch Spielraum zur Erweiterung. Soll zu einem späteren Zeitpunkt eine Verfeinerung von Sicherheitsstufen (vgl. hierzu auch das Paper von Wendzel, Kahler und Rist [12]) oder die Einfüh-

rung weiterer VLANs angestrebt werden, so kann dies durch eine Erweiterung des Tagging erreicht werden. Zu diesem Zweck könnte die in Abschnitt 4.1.4 vorgestellte Methode durch die Hinzunahme von Bit 6 des „ServiceChoice“-Byte auf insgesamt 3 Abstufungen/VLANs (2^2-1) oder bei zusätzlichem Einsatz von Bit 5 auf 7 (2^3-1) Optionen ausgedehnt werden.

Ein positiver Nebeneffekt, welcher sich durch die Einführung des Tagging ergibt: versucht ein Angreifer, Pakete in das Netzwerk einzubringen, welche an als sensibel eingestufte Endpunkte gerichtet sind, so werden diese nur dann weitergeleitet, wenn sie direkt an einen BFR übergeben werden, welcher diese markiert. Das Weiterleiten ungetaggter Pakete wird durch die in Abschnitt 4.5 skizzierte Vorgehensweise unterbunden.

Die Markierung der Pakete könnte des Weiteren dazu genutzt werden, um Seitenkanäle in einem Gebäudenetzwerk aufzudecken: werden getaggte Pakete an Positionen im Netzwerk angetroffen, welche sie eigentlich nicht passieren sollten, so ist dies ein klarer Hinweis auf einen Seitenkanal und bedarf einer Analyse der Konfiguration der Netzwerkkomponenten. Sollte ein Extrusion Detection System zum Einsatz kommen, können hierbei getaggte Pakete berücksichtigt werden.

Erwähnt werden soll auch, dass sich das Tagging von Paketen nicht auf die in 2.3.4 erwähnte Konformität der Geräte auswirkt.

Bei allen Vorteilen, die die vorgeschlagene Vorgehensweise mit sich bringt, soll aber auch auf Aspekte hingewiesen werden, welche bei der Umsetzung zu berücksichtigen sind: in verschiedenen Arbeiten (z.B. in [13]) wird die Einführung eines Traffic-Normalizers vorgeschlagen, welcher Pakete auf ihre Entsprechung mit der Norm hin überprüft. Beim Einsatz eines Normalizers muss sichergestellt werden, dass getaggte Pakete in ihrem Aufbau zwar nicht der aktuellen Norm DIN EN ISO 16484-5:2012 [3] entsprechen, eine Normalisierung aber ein „Untagging“ bedeuten würde, was einen Rückschritt zur bereits vorherrschenden, nicht eingeschränkten Situation nach sich zieht.

6 Fazit und Ausblick

Die Untersuchung der verschiedenen Protokollschichten und das Aufzeigen der Unterschiede, welche die Gebäudeautomation von der Realität in der „konventionellen“ IT abhebt, sollten klar gemacht haben, dass Vorkehrungen, welche zur Vermeidung – oder zumindest Eindämmung - von Data Leakage in der Gebäudeautomation führen, unverzichtbar sind.

Auf Standard-Sicherheitsmaßnahmen darf in der Gebäudeautomation trotz der in dieser Arbeit eingeführten, zusätzlichen Mechanismen nicht vergessen werden. So kann unter anderem eine Trennung von Implementierung des Automationsnetzwerkes und anschließender Konfiguration des BFR zur zusätzlichen Sicherheit beitragen.

Verschlüsselung, Traffic Normalizer und viele andere Ansätze, auf welche sich die Forschung im Bereich Gebäudeautomation konzentriert, stehen nicht im Widerspruch zum Konzept des Tagging.

Letztlich muss aufgrund der Einstufung der Kritizität der im Netzwerk übertragenen Daten entschieden werden, ob ein reines Tagging für ein ausreichendes Mehr an Sicherheit sorgen kann, ob weitere Maßnahmen ergriffen werden müssen oder ob – wie es im Falle von hochsensiblen Daten der Fall sein könnte – sogar ein isoliertes Gebäudenetzwerk für sensible Komponenten implementiert werden sollte.

Es ist davon auszugehen, dass die Gebäudeautomation, deren Wachstumspotential noch lange nicht ausgeschöpft ist, auch in den nächsten Jahren ein stark wachsender Markt bleiben wird. Damit einhergehend wird auch die Menge der zur Verfügung stehenden Daten stark anwachsen, was seinerseits wieder das Interesse der Wirtschaft daran verstärken kann. Insbesondere im Gesundheitswesen kann das Risiko, welches für den unbefugten Informationsgewinn eingegangen wird, schnell durch lukrative Geschäftsmöglichkeiten aufgewogen werden. Die Notwendigkeit, dass sich die Forschung im Bereich der Gebäudeautomation laufend mit dem Thema Sicherheit beschäftigt, liegt damit auf der Hand, nicht zuletzt aufgrund der Tatsache, dass sich der ortsunabhängige Zugriff auf die Daten des Automationsnetzwerkes immer größerer Beliebtheit erfreut.

Literaturverzeichnis

[1] H. Merz, T. Hasemann und C. Hübner, Building Automation, Berlin Heidelberg: Springer-Verlag, 2009.

[2] DIN EN ISO, Systeme der Gebäudeautomation (GA) - Teil 2: Hardware (ISO 164842:2004): Deutsche Fassung EN ISO 16484-2:2004.

[3] DIN EN ISO, Systeme der Gebäudeautomation (GA) - Teil 5: Datenkommunikationsprotokoll (ISO 16484-5:2012): Englische Fassung EN ISO 16484-5:2012, 2012.

[4] C. P. Pfleeger und S. L. Pfleeger, Security in Computing, New Jersey: Prentice Hall, 2003.

[5] A. S. Tanenbaum und M. van Steen, Verteilte Systeme, Prinzipien und Paradigmen, München: Pearson Education Deutschland GmbH, 2008.

[6] A. Shabtai, Y. Elovici und L. Rokach, A Survey of Data Leakage Detection and Prevention Solutions, New York Heidelberg Dordrecht London: Springer, 2012.

[7] R. Mogull, „Understanding and Selecting a Data Loss Prevention Solution," [Online]. Available: https://securosis.com/assets/library/reports/DLP-Whitepaper.pdf.

[8] InfoWatch, „Globale Untersuchung von Datenlecks im Jahr 2013," 2014. [Online]. Available: http://infowatch.com/sites/default/files/report/infowatch_global_report_2013_de.pdf. [Zugriff am 30 05 2014].

[9] W. Granzer, F. Praus und W. Kastner, „Security in Building Automation Systems," IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, Vol. 57, NO 11, pp. 3622-3630, November 2010.

[10] S. Karg, „BACnet captures," [Online]. Available: http://kargs.net/captures/. [Zugriff am 08 08 2014].

[11] J. J. Bender, „BACnet Firewall Router (Beta) auf sourceforge.org,“ 2004-2014. [Online]. Available: https://sourceforge.net/projects/bfr/.

[12] S. Wendzel, B. Kahler und T. Rist, „Covert Channels and their Prevention in Building Automation Protocols - A Prototype Exemplified Using BACnet,“ 2012 IEEE Conference on Green Computing an Communications, Conference on Internet of Things, and Conference on Cyber, Physical an Social Computing, pp. 731-736, 2012.

[13] S. Szlósarczyk, S. Wendzel, J. Kaur, M. Meier und F. Schubert, „Towards Suppressing Attacks on and Improving Resilience of Building Automation Systems - an Approach Exemplified Using BACnet,“ GI Sicherheit, Wien 2014, pp. 407-418, 2014.

Anhang A: BACnet APDUs – Wireshark-Captures

```
Building Automation and Control Network APDU
    0000 .... = APDU Type: Confirmed-REQ (0)
  .... 0010 = PDU Flags: 0x02
      .... 0... = Segmented Request: Unsegmented Request
      .... .0.. = More Segments: No More Segments Follow
      .... ..1. = SA: Segmented Response accepted
    .000 .... = Max Response Segments accepted: Unspecified (0)
    .... 0011 = Size of Maximum ADPU accepted: Up to 480 octets (fits in an ARCNET frame) (3)
    Invoke ID: 54
    Service Choice: atomicReadFile (6)
  ObjectIdentifier: file, 0
    Application Tag: BACnetObjectIdentifier, Length/Value/Type: 4
        .... 0... = Tag Class: Application Tag
        1100 .... = Application Tag Number: BACnetObjectIdentifier (12)
        Length Value Type: 4
      0000 0010 10.. .... .... .... .... .... = Object Type: file (10)
      .... .... ..00 0000 0000 0000 0000 0000 = Instance Number: 0
  stream access
    {[0]
        .... 1... = Tag Class: Context Specific Tag
        0000 .... = Context Tag Number: 0
        .... .110 = Named Tag: Opening Tag (6)
    File Start Position: (Signed) 0
      Application Tag: Signed Integer (2's complement notation), Length/Value/Type: 1
          .... 0... = Tag Class: Application Tag
          0011 .... = Application Tag Number: Signed Integer (2's complement notation) (3)
          Length Value Type: 1
    Requested Octet Count: (Unsigned) 440
      Application Tag: Unsigned Integer, Length/Value/Type: 2
          .... 0... = Tag Class: Application Tag
          0010 .... = Application Tag Number: Unsigned Integer (2)
          Length Value Type: 2
    }[0]
        .... 1... = Tag Class: Context Specific Tag
        0000 .... = Context Tag Number: 0
        .... .111 = Named Tag: Closing Tag (7)
```

```
0000  00 0c 6e b0 3c 15 00 60  2d 00 15 d5 08 00 45 00   ..n.<..` -.....E.
0010  00 36 6c a7 00 00 40 11  8c ad c0 a8 00 05 c0 a8   .6l...@. ........
0020  00 0d ba c0 ba c0 00 22  fe 1f 81 0a 00 1a 01 0c   ......." ........
0030  00 0d 01 3d 02 03 36 06  c4 02 80 00 00 0e 31 00   ...=..6. ......1.
0040  22 01 b8 0f                                       "...
```

Abbildung 52: BACnet APDU Confirmed request

```
Building Automation and Control Network APDU
    0001 .... = APDU Type: Unconfirmed-REQ (1)
    Unconfirmed Service Choice: who-Is (8)
  Device Instance Range Low Limit: 9999
    Context Tag: 0, Length/Value/Type: 2
        .... 1... = Tag Class: Context Specific Tag
        0000 .... = Context Tag Number: 0
        Length Value Type: 2
  Device Instance Range High Limit: 9999
    Context Tag: 1, Length/Value/Type: 2
        .... 1... = Tag Class: Context Specific Tag
        0001 .... = Context Tag Number: 1
        Length Value Type: 2
```

```
0000  ff ff ff ff ff ff 00 19  b9 6e 0a f8 08 00 45 00   ........ .n....E.
0010  00 2e 6b f9 00 00 80 11  4c 6e c0 a8 00 08 c0 a8   ..k..... Ln......
0020  00 ff ba c0 ba c0 00 1a  34 4c 81 0b 00 12 01 20   ........ 4L..... 
0030  ff ff 00 ff 10 08 0a 27  0f 1a 27 0f               ....... ..'.
```

Abbildung 53: BACnet APDU Unconfirmed request

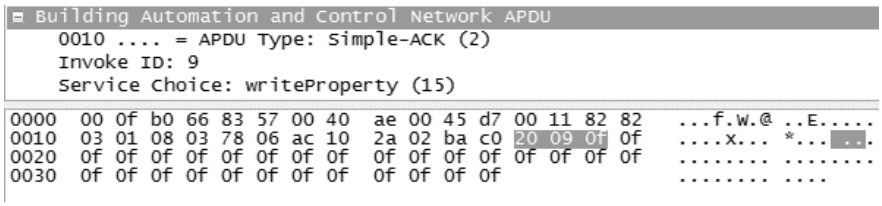


Abbildung 54: Bacnet APDU SimpleACK

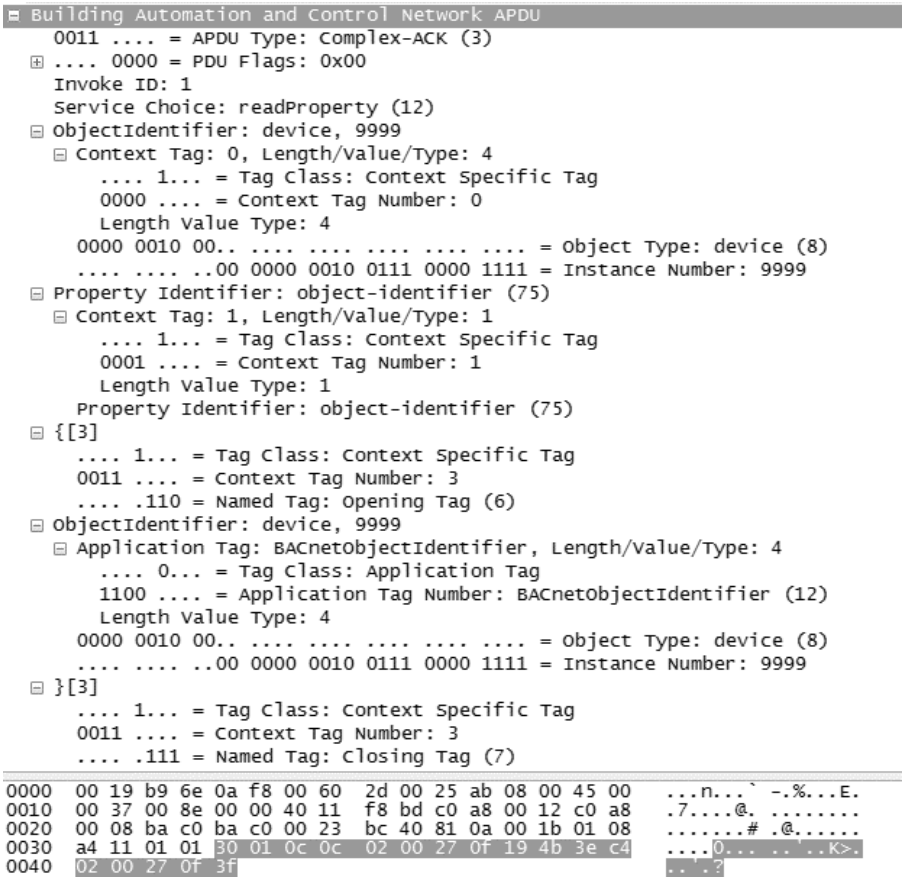


Abbildung 55: BACnet APDU ComplexACK

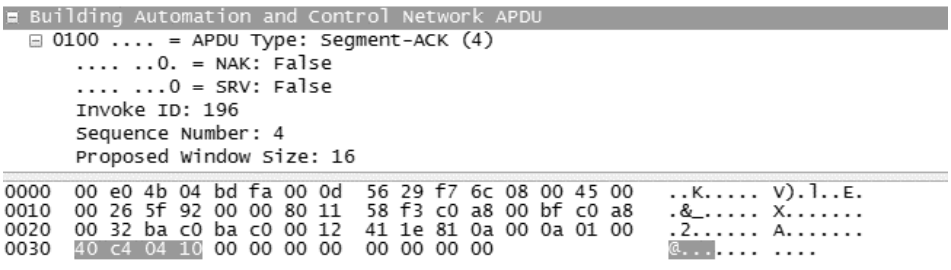


Abbildung 56: BACnet APDU SegmentACK

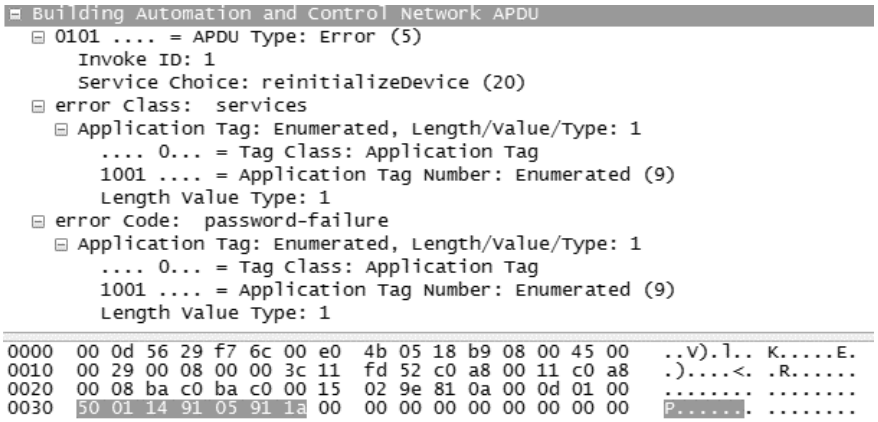


Abbildung 57: BACnet APDU Error

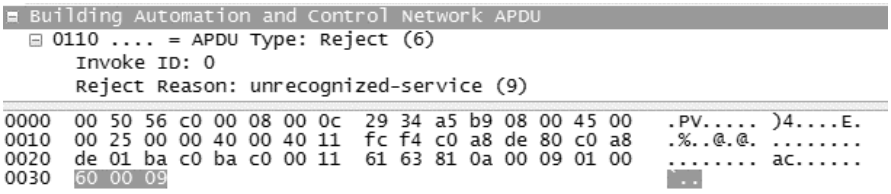


Abbildung 58: BACnet APDU Reject

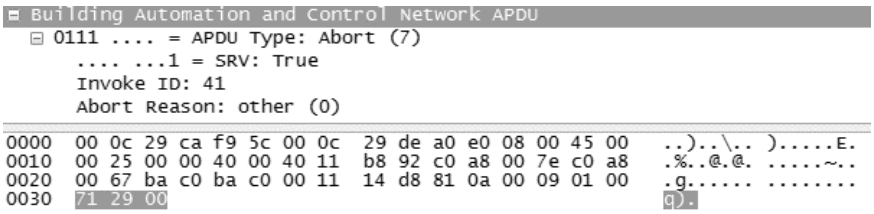


Abbildung 59: BACnet APDU Abort

Anhang B: BACnet NPDUs – Wireshark-Captures

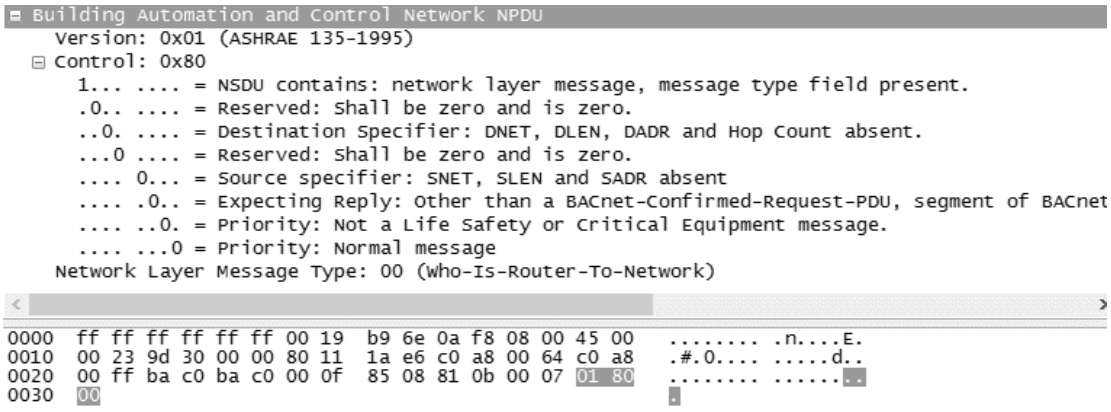


Abbildung 60: BACnet NPDU Who-Is-Router-To-Network

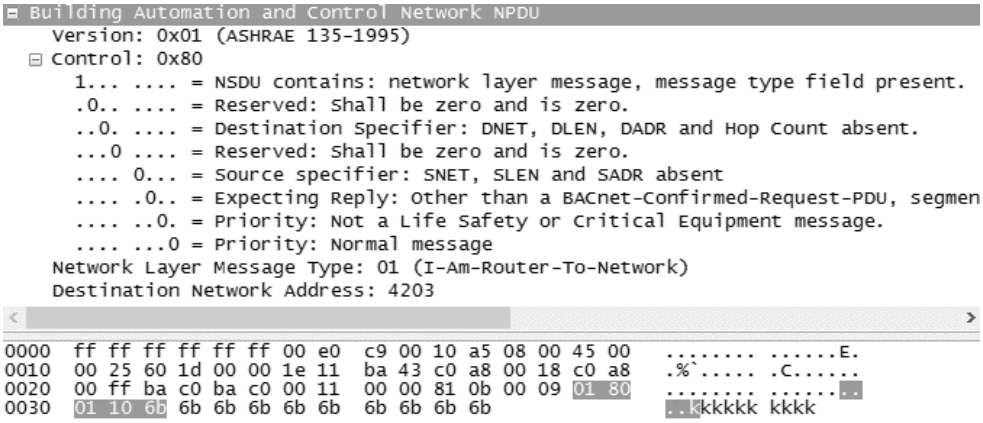


Abbildung 61: BACnet NPDU I-Am-Router-To-Network

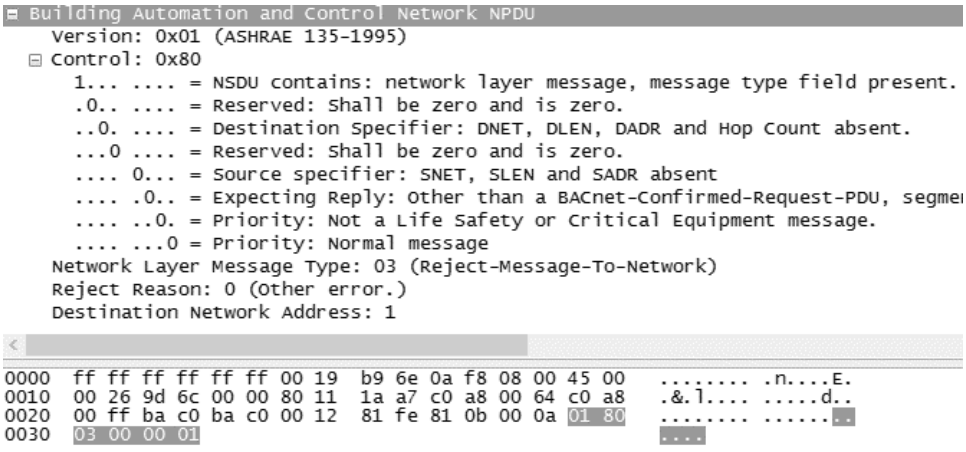


Abbildung 62: BACnet NPDU Reject-Message-To-Network

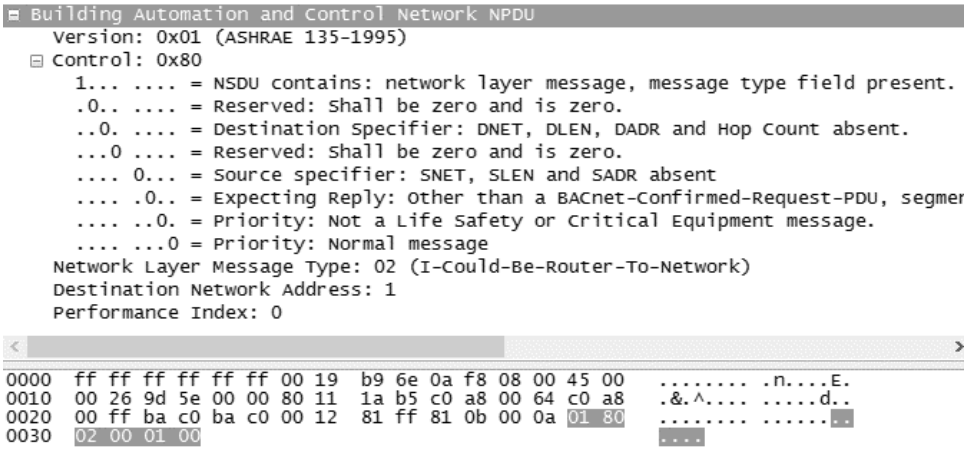


Abbildung 63: BACnet NPDU I-Could-Be-Router-To-Network

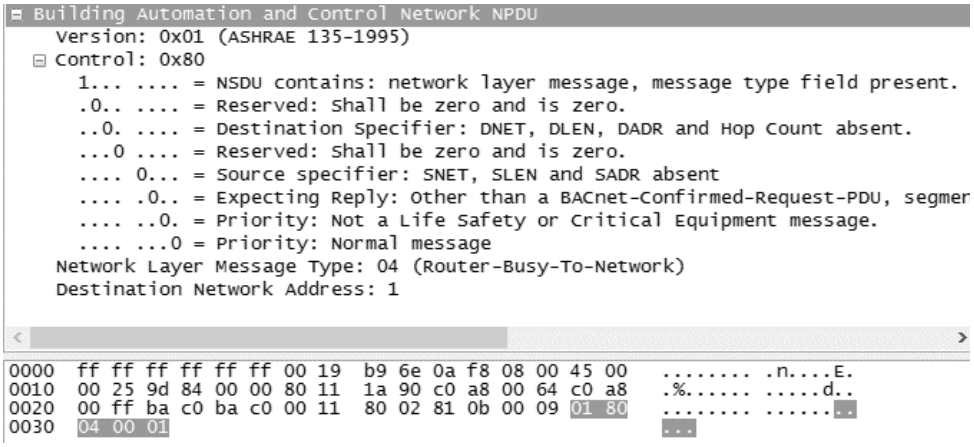


Abbildung 64: BACnet NPDU Router-Busy-To-Network

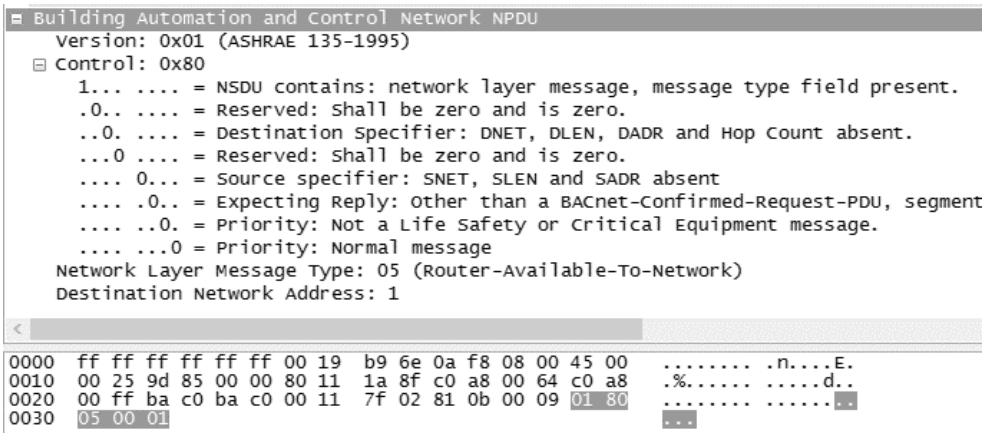


Abbildung 65: BACnet NPDU Router-Available-To-Network

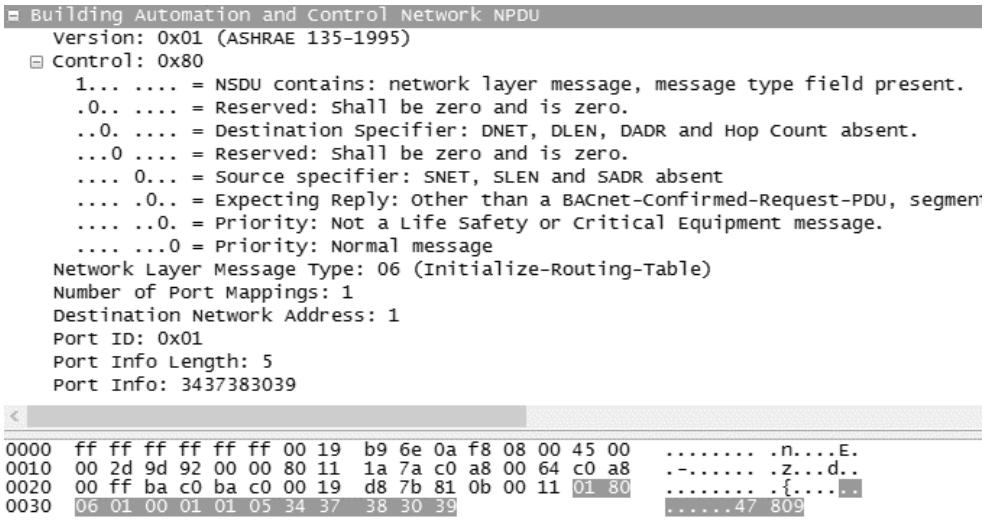


Abbildung 66: BACnet NPDU Initialize-Routing-Table

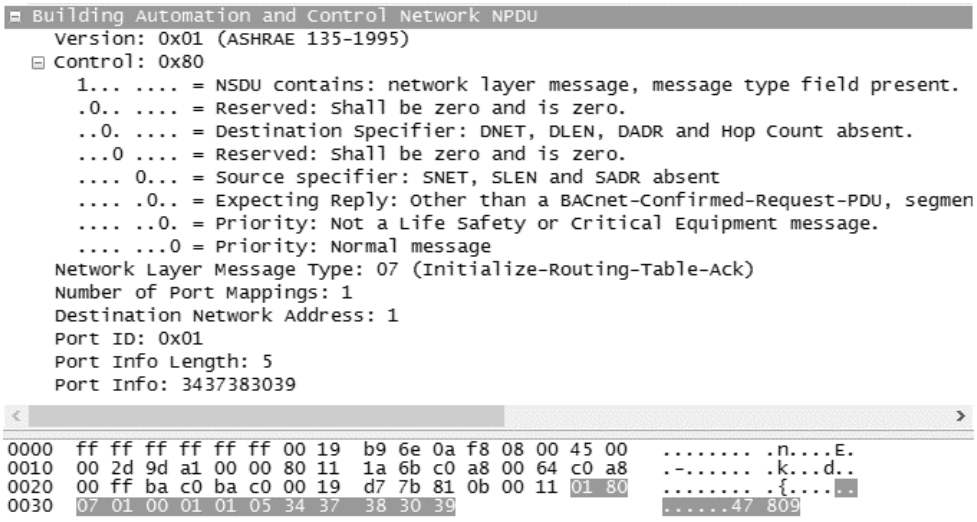


Abbildung 67: BACnet NPDU Initialize-Routing-Table-Ack

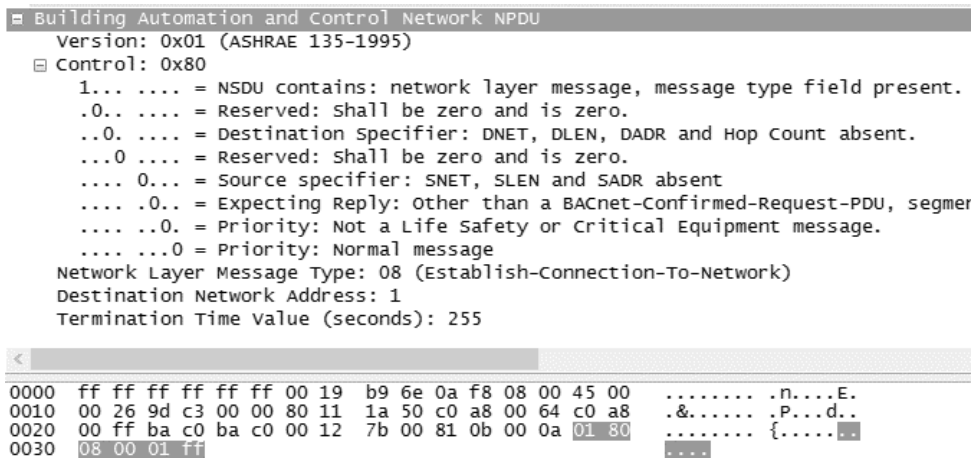


Abbildung 68: BACnet NPDU Establish-Connection-To-Network

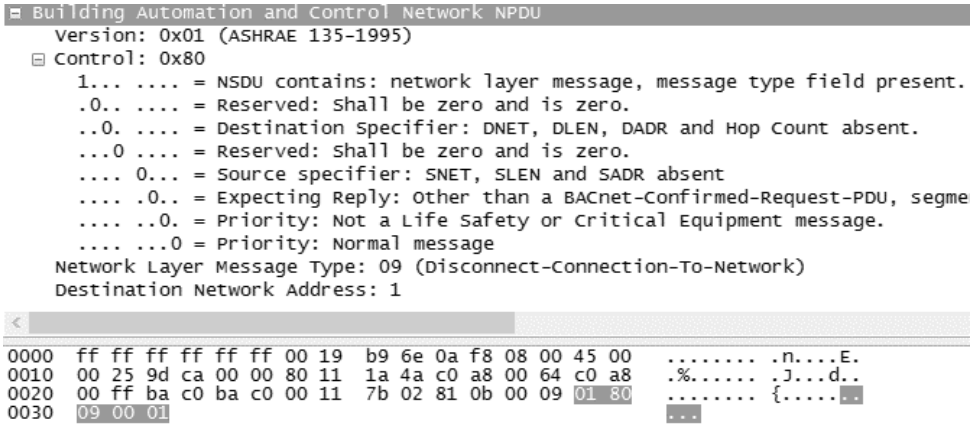


Abbildung 69: BACnet NPDU Disconnect-Connection-To-Network

Anhang C: Konfiguration der Testfälle des BFR

```
<BFR>
        <UDP address="192.168.100.20" server="net20" />
        <Debug client="net20" server="net20x" prefix="[20] " />
        <BIP client="net20x" server="net20y" />

        <UDP address="192.168.200.10" server="net10" />
        <Debug client="net10" server="net10x" prefix="[10] " />
        <BIP client="net10x" server="net10y" />

        <Router>
                <Adapter client="net20y" net="20" />
                <Adapter client="net10y" net="10" />
        </Router>
</BFR>
```

Konfiguration 1: Konfiguration BFR als reiner Router, keine Filterung

```
<BFR>
        <UDP address="192.168.100.20" server="net20" />
        <Debug client="net20" server="net20x" prefix="[20] " />
        <BIP client="net20x" server="net20y" />

        <UDP address="192.168.200.10" server="net10" />
        <Debug client="net10" server="net10x" prefix="[10] " />
        <BIP client="net10x" server="net10y" />

        <Filter client="net10y" server="net10yf">
                <Downstream>
                        <Allow />
                        <Reject function="UNCONFIRMED" />
                </Downstream>
                <Upstream>
                        <Allow />
                        <Reject function="UNCONFIRMED" />
                </Upstream>
        </Filter>

        <Router>
                <Adapter client="net20y" net="20" />
                <Adapter client="net10yf" net="10" />
        </Router>
</BFR>
```

Konfiguration 2: Konfiguration BFR für Filterung von „Unconfirmed-Request“

```
<BFR>
	<UDP address="192.168.100.20" server="net20" />
	<Debug client="net20" server="net20x" prefix="[20] " />
	<BIP client="net20x" server="net20y" />

	<UDP address="192.168.200.10" server="net10" />
	<Debug client="net10" server="net10x" prefix="[10] " />
	<BIP client="net10x" server="net10y" />

	<Filter client="net10y" server="net10yf">
		<Downstream>
			<Allow />
		</Downstream>
		<Upstream>
			<Allow />
		</Upstream>
	</Filter>

	<Router>
		<Adapter client="net20y" net="20" />
		<Adapter client="net10yf" net="10" />
	</Router>
</BFR>
```

Konfiguration 3: Konfiguration BFR für Filterung NPDU Testfall 1 (ohne Filterkriterien)

```
<BFR>
	<UDP address="192.168.100.20" server="net20" />
	<Debug client="net20" server="net20x" prefix="[20] " />
	<BIP client="net20x" server="net20y" />

	<UDP address="192.168.200.10" server="net10" />
	<Debug client="net10" server="net10x" prefix="[10] " />
	<BIP client="net10x" server="net10y" />

	<Filter client="net20y" server="net20yf">
		<Downstream>
			<Allow />
			<Reject source="C0-A8-64-0A-BA-C0" />
		</Downstream>
		<Upstream>
			<Allow />
			<Reject source="C0-A8-64-0A-BA-C0" />
		</Upstream>
	</Filter>

	<Router>
		<Adapter client="net20yf" net="20" />
		<Adapter client="net10y" net="10" />
	</Router>
</BFR>
```

Konfiguration 4: Konfiguration BFR für Filterung NPDU Testfall 2 (Quelle 192.168.100.10 Port 47808)

```
<BFR>
	<UDP address="192.168.100.20" server="net20" />
	<Debug client="net20" server="net20x" prefix="[20] " />
	<BIP client="net20x" server="net20y" />

	<UDP address="192.168.200.10" server="net10" />
	<Debug client="net10" server="net10x" prefix="[10] " />
	<BIP client="net10x" server="net10y" />

	<Filter client="net10y" server="net10yf">
		<Downstream>
			<Allow />
			<Reject destination="C0-A8-C8-14-BA-C0" />
		</Downstream>
		<Upstream>
			<Allow />
			<Reject destination="C0-A8-C8-14-BA-C0" />
		</Upstream>
	</Filter>

	<Router>
		<Adapter client="net20y" net="20" />
		<Adapter client="net10yf" net="10" />
	</Router>
</BFR>
```

Konfiguration 5: Konfiguration BFR für Filterung NPDU Testfall 3 (Ziel 192.168.200.20 Port 47808)

```
<BFR>
	<UDP address="192.168.100.20" server="net20" />
	<Debug client="net20" server="net20x" prefix="[20] " />
	<BIP client="net20x" server="net20y" />

	<UDP address="192.168.200.10" server="net10" />
	<Debug client="net10" server="net10x" prefix="[10] " />
	<BIP client="net10x" server="net10y" />

	<Filter client="net10y" server="net10yf">
		<Downstream>
			<Allow />
			<Reject destination="C0-A8-FA-14-BA-C0" />
		</Downstream>
		<Upstream>
			<Allow />
			<Reject destination="C0-A8-FA-14-BA-C0" />
		</Upstream>
	</Filter>
	<Router>
		<Adapter client="net20y" net="20" />
		<Adapter client="net10yf" net="10" />
	</Router>
</BFR>
```

Konfiguration 6: Konfiguration BFR für Filterung NPDU Testfall 4 (Ziel 192.168.250.20 Port 47808)

```
<BFR>
	<UDP address="192.168.100.20" server="net20" />
	<Debug client="net20" server="net20x" prefix="[20] " />

	<Filter client="net20x" server="net20y">
		<Downstream>
			<Allow />
			<Reject function="ORIGINAL-UNICAST-NPDU" />
		</Downstream>
		<Upstream>
			<Allow />
			<Reject function="ORIGINAL-UNICAST-NPDU" />
		</Upstream>
	</Filter>

	<BIP client="net20y" server="net20yf" />
	<UDP address="192.168.200.10" server="net10" />
	<Debug client="net10" server="net10x" prefix="[10] " />
	<BIP client="net10x" server="net10y" />

	<Router>
		<Adapter client="net20yf" net="20" />
		<Adapter client="net10y" net="10" />
	</Router>
</BFR>
```

Konfiguration 7: Konfiguration BFR für Filterung BAcnet/IP von „Original-Unicast-NPDU“-Nachrichten

```
<BFR>
	<UDP address="192.168.100.20" server="net20" />
	<Debug client="net20" server="net20x" prefix="[20] " />
	<BIP client="net20x" server="net20y" />

	<UDP address="192.168.200.10" server="net10" />
	<Debug client="net10" server="net10x" prefix="[10] " />
	<BIP client="net10x" server="net10y" />

	<Filter client="net10y" server="net10yf">
		<Downstream>
			<Allow function="READSENSITIVEPROPERTY" />
		</Downstream>
		<Upstream>
			<Allow function="READSENSITIVEPROPERTY" />
		</Upstream>
	</Filter>

	<UDP address="192.168.220.30" server="net30" />
	<Debug client="net30" server="net30x" prefix="[30] " />
	<BIP client="net30x" server="net30y" />

	<Filter client="net30y" server="net30yf">
		<Downstream>
			<Allow function="READSENSITIVEPROPERTY"
			destination="192.168.220.50" />
			<Reject function="READSENSITIVEPROPERTY"
			destination="192.168.220.40" />
		</Downstream>

		<Upstream>
			<Allow function="READSENSITIVEPROPERTY"
			destination="192.168.220.50" />
			<Reject function="READSENSITIVEPROPERTY"
			destination="192.168.220.40" />
		</Upstream>
	</Filter>

	<Router>
		<Adapter client="net20y" net="20" />
		<Adapter client="net10yf" net="10" />
		<Adapter client="net30yf" net="30" />
	</Router>
</BFR>
```

Konfiguration 8: Konfiguration BFR für Filterung von „ReadSensitiveProperty“-Nachrichten

Versicherung

Name: Eva Maria Anhaus

Matrikel-Nr.: 7694687

Fach: Informatik

Modul: Abschlussarbeit

Hiermit versichere ich an Eides statt, dass ich die vorliegende Abschlussarbeit mit dem Thema

Data Leakage Protection für Gebäude

selbstständig und ohne Inanspruchnahme fremder Hilfe angefertigt habe. Ich habe dabei nur die angegebenen Quellen und Hilfsmittel verwendet und die aus diesen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen. Ich erkläre mich damit einverstanden, dass die Arbeit mit Hilfe eines Plagiatserkennungsdienstes auf enthaltene Plagiate überprüft wird.

Datum: _____ **Unterschrift:** _____