AV-TEST
The Independent IT-Security Institute
Magdeburg Germany

AV-TEST-Studie

7 Smart-Home-Starter-Kits im Sicherheits-Test

Michael Schiefer M.Sc.
Ulf Lösche
Dipl.-Ing. Maik Morgenstern

Editing und Executive Summary:
Markus Selinger

28. April 2014
1.21

www.av-test.org

Executive Summary

Im ersten großen Test von Starter-Kits für Smart-Home-Lösungen können die Experten von AV-TEST leider keine Entwarnung geben. Nur 3 von 7 Kits sind gegen Angriffe gut gesichert, der Rest ist gegen interne und zum Teil externe Angriffe schlecht geschützt. Das bedeutet: melden sich ungeschützte Smart-Home-Geräte im Internet, wird das heimische Netz durch die Hintertür gekapert. Die minimal geschützten Smart-Home-Geräte wird man bald mit Trojanern angreifen, die sich dann bei einem erfolgreichen Angriff nicht im PC, sondern etwa im Speicher des Rauchmelders verstecken.

Ist das Smart-Home-Konzept völlig unsicher?

Die Grundidee hinter der Smart-Home-Technik ist natürlich sehr gut: in Zukunft sollen sich alle Komponenten im Haushalt überwachen und steuern lassen. Alle Komponenten, die Zugang zum heimischen Netz und dadurch vielleicht Zugriff auf das Internet haben, sind potentiell angreifbar. Das bedeutet aber nicht, dass sie auch potentiell unsicher sind.

Der aktuelle Test belegt, dass einige Anbieter ihre Geräte nicht blindlings auf den Markt geworfen, sondern ein Sicherheitskonzept mit eingearbeitet haben.

Im Test waren die folgenden Smart-Home-Kits, die primär durch große Firmen vertrieben oder auch exklusiv angeboten werden; es erfüllen zwar nicht alle Kits die gleichen Aufgaben im Haushalt, aber vergleichbar sind sie dennoch:

- iConnect von eSaver:
- tapHOME von EUROiSTYLE
- Gigaset Elements von Gigaset
- iComfort von REV Ritter
- RWE Smart Home von RWE
- QIVICON von der Telekom
- XAVAX MAX! von Hama

3 mal sicher vs. 4 mal unsicher

Das Ergebnis des Tests wird einige Hersteller hoffentlich zur Nachbesserung ihrer Produkte bewegen:

Die Sets Gigaset Elements, RWE Smart Home und QIVICON boten im Test eine gute Absicherung vor An- und Eingriffen.

Die Sets iComfort und tapHOME sind vor Angriffen im eigenen Netz ungeschützt. Die Kits iConnect und XAVAX MAX! sind gegen Angriffe im eigenen Netz und von außen recht machtlos.

Das können die Sets smart lenken

Die Kits haben diverse Aufgaben in einem smarten Haushalt und lassen sich folgendermaßen aufteilen:

- Kontrollsystem für Strom, Heizung & Sicherheit:
  RWE Smart Home und QIVICON
- Überwachungssystem für Fenster, Türen und Wohnräume: Gigaset Elements
- Kontrollsystem für Schaltsteckdosen: iComfort, tapHOME, iConnect
- Schaltsystem für Licht, Heizung und Strom: MAX!

Bei den verschiedenen Aufgaben lässt sich ein Horror-Szenario bei der Übernahme durch außen schnell finden: man könnte zum Beispiel eine gekaperte Heizungssteuerung so herunterregeln, dass im tiefsten Winter die Wasserleitungen einfrieren und platzen würden.

Da aber Angreifer in der Regel hinter Geld oder wertvollen Daten her sind, sind folgende Szenarien wahrscheinlicher: anhand der aktuellen An- und Abschaltpläne der Geräte lässt sich ersehen, wann die Bewohner eines Hauses da sind und wann nicht. Lässt sich dann auch noch die Sicherheitsüberwachung für Fenster und Türen abschalten, hat ein Einbrecher leichtes Spiel.

Ebenfalls wahrscheinlich: alle verbunden Geräte greifen über das heimische Netz weitere Geräte an und nehmen sie als Geisel – sperren den Zugriff darauf. Erst gegen eine Zahlung wird die Sperre vermeintlich aufgehoben. In dieser Art arbeitet etwa der BKA-Trojaner, der PCs samt den darauf befindlichen Daten sperrt.

Wo stecken die Schwachstellen?

In der Sicherheitsuntersuchung fahndeten die Tester nach den Schwachstellen. Daher untersuchten sie jedes Starter-Kit auf dessen Schutzkonzept. Dabei lag das Augenmerk auf der Verschlüsselung bei der Kommunikation, einer aktiven Authentifizierung, der Manipulation durch Externe und auf der gesicherten Fernsteuerung. Schließlich lassen sich die Sets entweder per Smartphone-App oder via Browser per WLAN oder aus dem Web ansprechen.

Verschlüsselte Kommunikation

Bei der Nutzung im lokalen Netzwerk zur Steuerung und bei einem Firmware-Update sollte die Kommunikation verschlüsselt erfolgen. Ebenso bei der Steuerung der Komponenten via Internet oder dem direkten Firmware-Update via Web.

Die Sets Gigaset Elements, RWE Smart Home und QIVICON erledigen die Kommunikation ohne Sicherheitsmanko. iConnect bietet eine Verschlüsselung, die aber leicht zu umgehen ist. Die Konzepte von iComfort, tapHOME und XAVAX MAX! versagen hier komplett – es findet keine gesicherte Kommunikation statt.

Aktive Authentifizierung

Selbst eine eigentlich zu erwartende Authentifizierung beim Zugriff auf die Geräte ist kein Standard bei den Produkten. Das Kit iComfort verlangt keine Authentifizierung. Die Sets iConnect und XAVAX MAX! verlangen zwar beim Zugriff via Web eine Authentifizierung, aber intern gibt es keinerlei Barrieren. Das ist fahrlässig.

Das Set von tapHOME verwendet intern zwar einen Benutzernamen und ein Passwort, aber das ist fast wertlos, da die Kommunikation unverschlüsselt abläuft und so jeder die Zugangsdaten abgreifen kann.

Nur die Sets Gigaset Elements, RWE Smart Home und QIVICON machen wieder einen fehlerfreien Job, wie bereits beim Punkt verschlüsselte Kommunikation.

Manipulation durch Externe

Da einige Geräte nicht verschlüsselt kommunizieren, lässt sich eventuell Schadcode injizieren oder noch perfider: den Geräten eine verseuchte Firmware unterschieben. In beiden Fällen hätten Hacker schnell Zugriff auf die Geräte und damit eine fest installierte Hintertür in das heimische Netzwerk.

Gesicherte Fernsteuerung

Der Fernzugriff ist gerade beim Smart-Home-Konzept eine wichtige Sache. Schließlich wollen die Nutzer ihre Komponenten via Smartphone-App oder Browser steuern. Wieder fielen im Test die beiden Kandidaten iConnect und XAVAX MAX! wegen eklatanter Schwächen durch. Die Kits von iComfort und tapHOME sind nicht gefährdet, da sie keinerlei Fernsteuerung bieten.

Die drei Sets Gigaset Elements, RWE Smart Home und QIVICON arbeiten auch in diesem Bereich gesichert und belegen damit ein insgesamt fehlerfreies Schutzkonzept.

Fazit: Hohe Sicherheit ist Trumpf

Wie eingangs bereits erwähnt, kann ein unsicheres Smart-Home-Kit die Hintertür in Ihr heimisches Netzwerk werden. Daher ist es unerlässlich, dass die Produkte ein hohes Maß an Sicherheit bieten. Leider gibt es noch keinen Sicherheits-Standard, wie er bei anderen Netzwerkgeräten, etwa WLAN oder Routern, vorhanden ist.

Die Testergebnisse lassen kein anderes Fazit zu, als dass im Moment nur die Sets Gigaset Elements von Gigaset, RWE Smart Home von RWE und QIVICON von der Telekom empfehlenswert sind.

Die Hersteller bzw. Vertreiber der Starter-Kits iConnect, tapHOME, iComfort und XAVAX MAX! müssen ihre Produkte in Sachen Sicherheit überarbeiten, um sie auf einen guten Stand zu bringen.

Blick in die Zukunft

Bei keinem der Sets gehört im Moment die Übertragung von Bild und Ton mit zur Ausstattung. Die Tester haben aber bei dem großen Test bereits die Vorbereitung von Schnittstellen für

Webcams mit Mikrofon gefunden. Auch der Zugriff auf passend ausgestattete Küchengeräte wie einen Kühlschrank oder eine Kaffeemaschine wird nicht lange auf sich warten lassen. Schließlich ist die Technik dazu leicht in der Lage. Der Gedanke, dass ein Fremder auf alle diese Geräte Zugriff hat, löst bei manchem Nutzer kalte Schauer aus.

Dazu kommt, dass Marktforscher einen weltweiten Umsatz mit Smart-Home-Produkten von über 15 Milliarden Dollar bis zum Jahr 2015 erwarten. Das klingt viel, aber erst vor kurzem hat Google bereits die Firma Nest Labs für 3,2 Milliarden Dollar gekauft. Der Suchmaschinen-Riese wird in dem Bereich Druck machen, den Nest Labs produziert: die steuerbaren Smart-Home-Komponenten Thermostate und Feuermelder.

Produkt	**Gigaset Elements**	**RWE Smart Home**	**QIVICON**	**iComfort**	**tapHOME**	**iConnect**	**XAVAX MAX!**
Anbieter	**Gigaset**	**RWE**	**Telekom**	**REV Ritter**	**EUROiSTYLE**	**eSaver**	**Hama**
Enthaltene Komponenten und Software	Gateway, Türsensor, Bewegungssensor, Smartphone-App "Gigaset Elements"	Gateway, Steckdosenschalter (Ein/Aus), Wandschalter (2 Tasten), Heizkörperthermostat, Online-Portal, mobiles Online-Portal, lokales Portal, Smartphone-App "RWE SmartHome"	Gateway, Steckdosenschalter (Ein/Aus), Heizkörperthermostat, Rauchmelder, Smartphone-App "Smart Home"	Gateway, 2 Steckdosenschalter (Ein/Aus), Smartphone-App "REV iComfort"	Gateway, Steckdosenschalter (Ein/Aus), Dimmbare Steckdose, Smartphone-App "tapHOME Hausautomatisierung"	Gateway, 2 Steckdosenschalter (Ein/Aus), Smartphone-App "eSaver Cloud"	Gateway, 2 Heizkörperthermostate, Eco-Wandschalter (Wechsel Eco/Auto), Fensterkontakt, MAX Desktop Software, Webportal
Enthaltene Schutzfunktionen							
Verschlüsselte Kommunikation	JA	JA	JA	NEIN	NEIN	JA	TEILWEISE
Aktive Authentifizierung	JA	JA	JA	NEIN	JA	nur bei Webzugriff	nur bei Webzugriff
Manipulation durch Externe	Keine Möglichkeit	Keine Möglichkeit	Keine Möglichkeit	Keine Möglichkeit	Keine Möglichkeit	Anfällig für Manipulationen	Anfällig für Manipulationen
Gesicherte Fernsteuerung	Wirksamer Schutz	Wirksamer Schutz	Wirksamer Schutz	Kein Fernzugriff möglich	Kein Fernzugriff möglich	Anfällig für Manipulationen	Anfällig für Manipulationen
Testergebnis	**Guter Schutz**	**Guter Schutz**	**Guter Schutz**	**Anfälliger Schutz**	**Anfälliger Schutz**	**Zu schwacher Schutz**	**Zu schwacher Schutz**
Erläuterung	Wirksames Schutzkonzept	Wirksames Schutzkonzept	Wirksames Schutzkonzept	Kein Schutz gegen interne Angriffe	Kein Schutz gegen interne Angriffe	Kein Schutz gegen in- und externe Angriffe	Kein Schutz gegen in- und externe Angriffe
Bemerkungen	keine	Konfiguration nur mit Web-Zugang möglich	Verschlüsselungsmethode könnte noch besser sein	Lokal eingedrungene Schadsoftware kann sich einnisten	Lokal eingedrungene Schadsoftware kann sich einnisten	Keine Hürden für Angreifer, Remote-Zugriff angreifbar	Lückenhafte Verschlüsselung lässt Angreifern freie Hand

Gesamtergebnis „7 Smart-Home-Starter-Kits im Sicherheits-Test“:

Im großen Sicherheitstest von AV-TEST konnten nur die Smart-Home-Lösungen Gigaset Elements, RWE Smart Home und QIVICON überzeugen. Die Sets kommunizieren verschlüsselt und bieten einen wirksamen Schutz gegen einen nicht autorisierten Zugriff.

Inhaltsverzeichnis

1 Einleitung

Seit einiger Zeit finden Smart-Home-Produkte eine immer größere Verbreitung und immer mehr Firmen drängen auf diesen Markt. Nicht selten werden dabei vorhandene Geräte gekoppelt und mit dem Internet verbunden. Die Folge sind über das Internet verwundbare Systeme. Mit speziellen Suchmaschinen wie „Shodan“ ließen sich so schon Ampelanlagen und Sicherheitskameras, aber auch Kontrollanlagen einer Badeanstalt und sogar Teile eines Nuklearkraftwerkes aufspüren[Goldman, 2013].

Natürlich bedeutet die Auffindbarkeit im Netz nicht automatisch eine Verwundbarkeit, doch es kommt immer wieder vor, dass die Sicherheit während der Entwicklung nicht bedacht oder vernachlässigt wird.

Der Begriff Smart Home ist dabei keineswegs einheitlich definiert. Für die Einen bedeutet er eine Verbesserung bestehender Funktionen, wie etwa noch effizientere Kühlschränke. Für die Anderen sind es ganz neue Geräte, die bestehende Funktionen mit dem Internet verbinden. Dies sind dann beispielsweise Kühlschränke, die über ihren Inhalt Bescheid wissen und vom Benutzer während des Einkaufs befragt werden können.

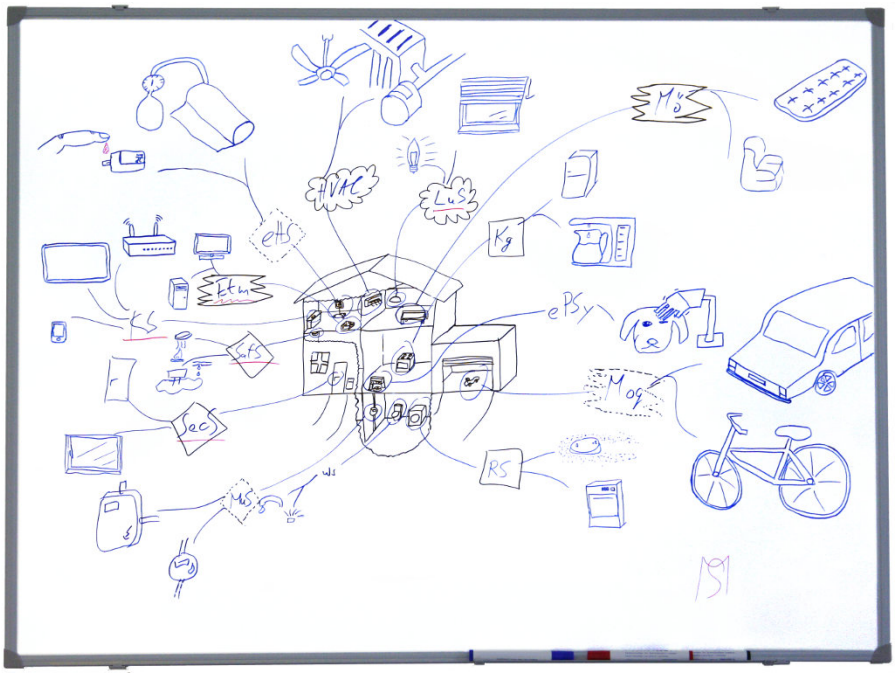

Abbildung 1: Beispielkategorien

Aufgrund dieser Ansicht mit der Verbindung zum Internet gibt es auch den alternativen Begriff Internet der Dinge bzw. Internet of Things. Basierend auf dem Fokus, der von den Beteiligten bei Smart Home gesehen wird, ergeben sich viele weitere Alias, allein [Brucke, 2009] zählt auf: Intelligentes Haus, Haus der Zukunft, Multimedia Home / Digital Home (Consumer Electronics), Internet Home, e-home und Intelligentes Wohnen, Smartes Wohnen. Wie auch bei den Firmen gibt es eine Vielfalt von Begriffen, die alle dasselbe beschreiben, aber doch

etwas leicht anderes meinen. Doch in einem sind sich alle einig: Smart Home sei die (unausweichliche) Zukunft.

Bild 1 zeigt, wo die Reise hingehen kann. Quasi das gesamte Haus ist vernetzt. Für diverse Bereiche gibt es schon die ersten Vorboten, seien es Bluetooth-Toiletten aus Japan oder Fütterungsautomaten für Tiere.

Das sind schon zwei gute Gründe, um sich einige Smart-Home-Starter-Kits genauer anzusehen. Hauptauswahlkriterien waren die Verfügbarkeit in Deutschland und der Heimanwender als Marktziel.

Die sieben untersuchten Systeme variieren stark in der Umsetzung, dem Funktionsumfang, aber auch in der IT-Sicherheit. Bei den Tests wurde nur an der Oberfläche der Möglichkeiten gekratzt und dennoch ergaben sich teils erhebliche Probleme.

2 Testübersicht

Im Folgenden werden die Tests beschrieben. Es wird zuerst auf den Aufbau eingegangen. Da die untersuchten Systeme unterschiedlich sind, sind auch die entsprechenden Testaufbauten unterschiedlich. Deshalb gibt es an dieser Stelle eine allgemeine Beschreibung.

Im Anschluss wird der Ablauf der Tests kurz beschrieben, auch hier wird eine eher abstrakte Form gewählt.

Die hier angegebenen und die durchgeführten Tests stellen erste Betrachtungen dar. Sie bieten einen ersten Überblick und ermöglichen bei späteren Untersuchungen einen tieferen Einblick. Im Kern werden die Verbindungen des Produktes in das lokale Netzwerk und ins Internet überprüft und bzgl. Verschlüsselung und Authentifizierung betrachtet. Darauf aufbauend werden die Folgen von Sicherheitsproblemen betrachtet, das Mitlesen von Daten und Manipulieren der Geräte. Der Missbrauch der Geräte für andere Zwecke, wie dem Versand von Spam, wurde nicht betrachtet.

2.1 Testaufbau

Das Bild 2 zeigt schematisch einen allgemeinen Testaufbau. Das zu untersuchende Gerät ist an einen Router angeschlossen, was in bisher allen Fällen per Ethernet stattfand. Ein Mobiltelefon für eventuelle Apps ist per WLAN – hier 2,4 GHz – ebenfalls an diesen Router angeschlossen, außerdem zwei weitere Computer per Kabel. Alle beschriebenen Geräte befinden sich im selben Netz. Für normale Anwendungen und browserbasierte Webseiten stehen ein Windows-XP- und ein Windows-7-Computer zur Verfügung.

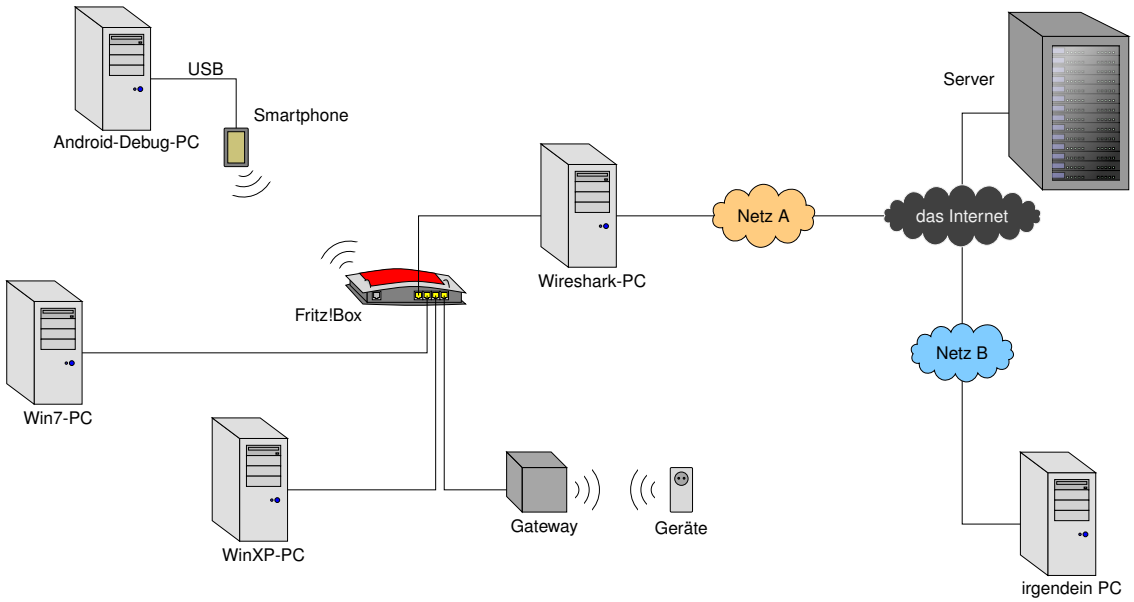

Abbildung 2: Schematischer Testaufbau

Der Router ist so konfiguriert, dass er sein Internet über einen LAN-Anschluss holt. Ein Computer mit der Mitschnittsoftware Wireshark ist dazwischen eingebracht. Auf diese Weise kann jede Kommunikation ins Internet, beispielsweise zu den Servern des untersuchten Systems, identifiziert werden. Es muss einfach nur das entsprechende Kabel vom oder zum Wireshark-PC entfernt werden, um das Internet aus dem Versuchsaufbau zu entfernen, ohne das durch den Router aufgebaute Netzwerk zu zerstören.

Ein weiterer Mitschnitt von Netzwerkpaketen geschieht auf dem Router selbst. Aufgrund von Optimierung in diesem sind leider nicht alle abgreifbar. Der Download geschieht auf einem der angeschlossenen Computer.

Um vom Telefon Screenshots erstellen zu können, läuft dieses im Debug-Modus und ist an einen Computer außerhalb des Netzwerkes angeschlossen. Für zeitkritische Testabschnitte werden diese notfalls wiederholt und beim Smartphone der Debugmodus deaktiviert, um so die ermittelten Ergebnisse nicht zu verfälschen.

Außerhalb des beschriebenen Netzwerkes gibt es einen weiteren Computer. Dieser besitzt im Internet eine andere IP als alle Geräte des beschriebenen Netzwerkes. Er stellt damit einen beliebigen Rechner auf der Welt dar. Er wird für die Simulation eines unberechtigten Zugriffes auf die Geräte durch einen potentiellen Angreifer im Internet verwendet.

2.2 Testablauf

Ziel der Untersuchung ist die IP-basierte Kommunikation des Gateways bzw. der Basisstation. Diese Hardware dient zur Überführung der Befehle des Smartphones, der Anwendungen oder des Webportals zu den einzelnen Geräten. Jeder Befehl läuft über die Basisstation, was sie auch für Angreifer besonders interessant macht. Deshalb erscheinen Authentifizierung und Verschlüsselung als notwendig und deren Existenz wird untersucht. Ferner sind mögliche Angriffsvektoren seit Jahren bekannt und wurden verfeinert. Deshalb wird auch eine automatisierte Version der Penetration-Software Metasploit in Form von Armitage für Vulnerability-Scans verwendet.

Der Ablauf ist anfangs von den Herstellern vorgegeben, es wird versucht, die Installationsanleitung so genau wie möglich zu befolgen. Allerdings wird das Internet zu Beginn aus dem Testaufbau verbannt, um einer automatisierten Firmware-Aktualisierung entgegenzuwirken und auch einen möglichen Internetzwang zu identifizieren. Folgerichtig muss die Anleitung des Herstellers abgeändert werden, beispielsweise wird der Download der Smartphone-App mittels Internet an den Anfang gestellt.

Sobald sich die Testmöglichkeiten erschöpfen, was bei Systemen mit Internetpflicht relativ schnell geschieht, wird dieses hinzugefügt und das System erst einmal nur beobachtet. Falls notwendig, wird dann das System wie vom Hersteller vorgesehen eingerichtet oder bekannte Internetfeatures getestet.

Ist das alles erledigt, so werden die Protokolle analysiert und nach theoretischen Eingriffsmöglichkeiten gesucht, die falls möglich auch real umgesetzt werden. Dies stellt den letzten Schritt dar. Bei SSL-Verbindungen findet jedoch keine Überprüfung von Man-in-the-Middle-Erkennung statt. Dafür wird nachgesehen, welche Version von SSL und welche Verschlüsselungen verwendet werden. Bei einigen Produkten werden viele SSL-Verbindungen hergestellt, so dass diese nur stichprobenartig untersucht werden.

2.3 Verschlüsselungen

Für sichere Übertragungen sind Verschlüsselungen oft das Mittel der Wahl. Die bloße Existenz bietet allerdings keinen Schutz. So gibt es Verfahren, die nach heutiger Sicht als sicher eingestuft werden, aber auch sehr viele unsichere. Dabei gilt es auch zu unterscheiden, ob es theoretische oder praxisrelevante Angriffe gibt. Diverse Verfahren sind theoretisch gebrochen, doch die dafür notwendigen Voraussetzungen sind in der Realität meist nicht gegeben. Solche Verfahren gelten dann je nach Schwere der Schwäche trotzdem noch als sicher.

Für SSL kommen häufig RC4, 3DES oder AES basierte Verschlüsselungen zum Einsatz. Beim RC4-Verfahren ist anzunehmen, dass mindestens ein Geheimdienst dieser Welt dieses brechen kann. Auch wenn der dafür nötige Aufwand oder auch nur eine Bestätigung dessen nicht bekannt sind, sollte eine Verwendung vermieden werden. 3DES oder Triple-DES gilt für die Privatperson noch als sicher, doch die geringe effektive Schlüssellänge von 112 Bit wird oft als Grund genannt, dieses Verfahren nicht mehr zu verwenden. Es ist davon auszugehen, dass Firmen oder Einrichtungen mit großen Rechenkapazitäten schon heute 3DES brechen können. Durch Cloud-Computing steht dies theoretisch auch Einzelpersonen zur Verfügung. Die bessere Wahl ist AES. Für dieses Verfahren sind mit 128, 192 und 256 Bit drei verschiedene Schlüssellängen vorgesehen, nach heutiger Sicht sind die 256 Bit auch die sichersten.

Der eigentliche Verschlüsselungsalgorithmus ist für die Sicherheit aber nicht alleinig entscheidend. Auch die Version von SSL und der verwendete Modus sind wichtig. Mit BEAST von 2011 und Lucky Thirteen von 2013 gibt es Angriffe gegen AES im CBC-Modus in Verbindung mit der SSL-Version TLS 1.0. Die 2006 erschienene Version TLS 1.1 verhindert bereits BEAST, doch bis heute sind TLS 1.1 und die aktuelle Version TLS 1.2 von 2008 kaum verbreitet.

Viele Entwickler-Bibliotheken für SSL versuchen von sich aus die Angriffe zu verhindern, z.B. sollten die meisten aktuellen Versionen BEAST verhindern, auch für TLS 1.0.

Eine Aussage über die Sicherheit ist folgerichtig nicht alleine aufgrund der exakten Verschlüsselung und SSL-Version möglich. Zusammen mit den verwendeten Bibliotheken kann eine Einschätzung vorgenommen werden, ob ein System möglicherweise verwundbar ist oder ob es wahrscheinlich als sicher angesehen werden kann. Für präzisere Aussagen ist die Erprobung notwendig, was im Rahmen dieser Arbeit nicht geschah. Es wurden keine Angriffe durchgeführt.

2.4 Erprobung von Theorien

Im Rahmen der Untersuchungen tauchten immer wieder mögliche Schwachstellen in den Produkten auf. Individuelle Schwächen erfordern individuelle Lösungen zur Verifizierung, doch meist galt es bestimmte Netzwerkpakete an das Gateway des entsprechenden Produktes zu schicken. Für die Übertragung wurde meist TCP/IP von den Herstellern verwendet. Zum Versenden und Empfangen von einfachen Textnachrichten über dieses Transportprotokoll bietet sich das Programm netcat an. Dieses ist in vielen Linux-Distributionen enthalten und ist auch sonst weit verbreitet.

Das Programm verfügt über keine grafische Oberfläche und wird per Konsole verwendet. Beim Start werden die Adresse und der Port des Ziels angegeben. Daraufhin baut netcat eine Verbindung zum Empfänger auf. Ist dies erfolgreich, so können Pakete durch Eingabe in die

Konsole versendet werden. Nach jedem Druck auf Enter wird an dem eingegebenen Text ein Zeilenumbruch angefügt, alles in ein TCP/IP-Paket gepackt und zum Ziel übermittelt. Vom Ziel eintreffende Pakete werden direkt in der Konsole ausgegeben.

Alle textbasierten, auf TCP/IP aufbauenden Protokolle, wie z.B. HTTP, können so leicht getestet werden. Prinzipiell ist es auch möglich, nahezu beliebige Binärdaten zu versenden, da jeder eingegebene Text – auch wenn er unleserliche oder ungültige Zeichen enthält – übertragen wird. Allerdings beschränken Betriebssystem und Desktopoberfläche, welche Bytefolgen eingegeben oder aus der Zwischenablage eingefügt werden können. Oft wird beispielsweise das Byte mit dem Wert 0 als Ende betrachtet und alles Nachfolgende ignoriert. Da netcat aber beim Start der Inhalt einer Datei übergeben werden kann und netcat diesen versendet, kann zumindest das erste Paket mit beliebigem Inhalt versendet werden. Alles andere danach ist auf Text beschränkt.

3 Getestete Produkte

Zum aktuellen Zeitpunkt wurden sieben Produkte untersucht. Es war dabei wichtig, dass es sich um Starter-Kits für Privatpersonen handelt. Ferner mussten die Produkte in Deutschland verfügbar sein.

Zu Beginn werden die für die Untersuchungen relevanten Eigenschaften der Produkte aufgelistet. Im Anschluss erfolgt eine kurze Beschreibung, welche die Anwendungsgebiete der Starter-Kits darlegt. Diese lassen sich mit weiteren Komponenten teils erheblich ausbauen.

3.1 Funktionsübersicht

Im Folgenden werden die für die Untersuchung wichtigen Eigenschaften der Starter-Kits tabellarisch dargelegt.

	eSaver, iConnect	EUROiSTYLE, tapHOME	Gigaset, Gigaset elements	REV Ritter, iComfort	RWE, RWE Smart Home	Telekom, QIVICON	XAVAX, MAX!
Umfang							
Smartphone-App	✔	✔	✔	✔	✔[1]	✔[1]	✘[2]
Lokale Software	✘	✘	✘	✘	✘	✘	✔
Webportal (im Internet)	✘	✘	✔[3]	✘	✔	✔	✔
Webportal (auf Gateway)	✘	✔[4]	✘	✘	✔[5]	✔	✘
Fernzugriff	✔	✘[6]	✔	✘	✔	✔	✔
Verbindungen							
Smartphone ↔ Gateway	✔	✔	✘	✔	✘	✘	⌀
Smartphone ↔ Internet	✔	✔	✔	✔	✔	✔	⌀
Software ↔ Gateway	⌀	⌀	⌀	⌀	⌀	⌀	✔
Software ↔ Internet	⌀	⌀	⌀	⌀	⌀	⌀	✔
Gateway ↔ Internet	✔	✔	✔	✘[7]	✔	✔	✔

✔vorhanden ✘nicht vorhanden ⌀nicht möglich

1 Nur mit Abonnement nutzbar, anfangs kostenlos.
2 Es gibt mehrere Apps (von eQ-3 und ELV), aber in den beigefügten Dokumenten wurde keine spezifiziert.
3 Dies ist in den FAQs erwähnt, auf die in der Schnellstartanleitung verwiesen wird.
4 In den beigefügten Dokumenten konnte kein Hinweis darauf gefunden werden.
5 Nur mit Silverlight und nach Installation von einem Modul nutzbar.
6 Dies war für Ende 2013 angekündigt.
7 Mit Ausnahme von NTP konnte keine Internetverbindung beobachtet werden.

Festgehalten wird, ob das System über eine Smartphone-App verfügt, Software existiert, die lokal installiert werden muss, oder ob es ein Webportal gibt. Das Webportal kann sowohl als im Internet gehostete Version vorliegen oder direkt auf dem Gateway des untersuchten Starter-Kits laufen. Auch hält die Tabelle fest, ob ein Fernzugriff für das Produkt existiert.

Danach werden die existierenden Verbindungen aufgelistet. Ob die lokale Software mit dem Gateway kommuniziert oder nicht, wird allerdings nur angegeben, wenn es eine lokale Software gibt. Ähnlich verhält es sich mit den anderen Verbindungen.

Es werden nur prinzipielle Angaben gemacht. Nur weil das Gateway eine Verbindung ins Internet hat, bedeutet dies nicht, dass diese immer besteht. Es kann auch sein, dass sie nur während eines Remote-Zugriff aktiv ist.

3.2 Produktbeschreibungen

Untersucht wurde die Einbruchsdetektions-Lösung „Gigaset elements“. Das Set besteht aus einer Basisstation, einem Bewegungssensor und einem Türsensor. Das System soll erkennen, wenn versucht wird, gewaltsam in ein Haus einzudringen. In dem Fall soll es einerseits Alarm geben und andererseits den Bewohner per Nachricht in der Smartphone-App oder per E-Mail informieren.

Weiterhin im Test waren die Steckdosenschalter-Systeme iComfort von REV Ritter, eSavers iConnect und tapHOME von EUROiSTYLE. Während das erstgenannte System über zwei Steckdosenschalter verfügt, hat das eSavers iConnect vier im Paket. In tapHOME gibt es neben einem Schalter auch einen Dimmer für gewöhnliche Glühlampen. Über die Schalter lassen sich angeschlossene Geräte beispielsweise per App schalten. Dazu ist das Zielgerät in die Zwischensteckdose einzustecken und einzuschalten. Wird dann die Steckdose eingeschaltet geht auch das Gerät mit an. So lassen sich z.B. Fernseher, Lampen oder eine alleinstehende Kochplatte aus der Ferne aktivieren. Eine Fernsteuerung – außerhalb der eigenen Wohnung – ist bisher nur bei der Lösung iComfort vorgesehen.

Das MAX! System, von Xavax vertrieben, dient der Einsparung von Heizkosten. Dafür gibt es neben Heizthermostaten auch Fensterkontakte und Schalter. Wird das Fenster geöffnet, regelt das Thermostat automatisch die Temperatur herunter, auch Zeitprofile können erstellt bzw. verändert werden. Eine Steuerung ist auch über die lokale Software oder nach Aktivierung per Webportal möglich.

RWE bietet seinen Kunden eine spezielle Form des „Starterpakets“ von RWE Smart Home kostenlos an. Es besteht aus einer Zentrale, einem Heizkörperthermostat, einem Wandtaster und einem Steckdosenschalter. RWE sieht in seinen Produkten eine hohe Konfigurierbarkeit vor. So ist es beispielsweise möglich, basierend auf der vom Thermostat gemessenen Temperatur den Steckdosenschalter zu bedienen. Laut RWE sollen in Zukunft auch Produkte weiterer Hersteller unterstützt werden. Das Produkt erfordert für die Einrichtung und Konfiguration sowohl Internet, als auch das Flash-Player-Pendant Silverlight von Microsoft.

QIVICON liefert nur das multifunktionale Herzstück für Smart Home: eine Basisstation, an der dann Produkte von zugelassenen bzw. unterstützten Herstellern angeschlossen werden können. Das Starter-Kit der Telekom enthält neben der Station von QIVICON noch vier weitere HomeMatic-Produkten: einen Rauchmelder, einen Steckdosenschalter und zwei Thermostaten. Zur Steuerung bietet die Telekom eine App für die Basisstation und eine für die ge-

koppelten Produkte. Die Telekom nennt ihre Lösung, bestehend aus der Basisstation, den vier HomeMatic-Produkten und der App „Smart Home“. Zur besseren Unterscheidung von RWE und anderen Produkten wird sie in diesem Dokument mit QIVICON bezeichnet.

Bei einer genaueren Betrachtung fällt auf, dass die Firma eQ-3 bei mindestens drei Systemen mitgewirkt hat. So stammt das sowohl bei MAX! als auch bei QIVICON eingesetzte Funkprotokoll BidCoS von eQ-3. Auch das Funkprotokoll von RWE haben sie entwickelt [eQ-3 AG]. Genauso gab es auch einen Einfluss auf die Hardware der drei Systeme.

3.3 Tabellarische Komponentenübersicht

Tabelle 2 listet die sieben Systeme zusammen mit den für die Untersuchung verwendeten Geräten und der verwendeten Software auf.

eSaver, iConnect	• Gateway • 2 Steckdosenschalter (Ein/Aus) • Smartphone-App ”eSaver Cloud”
EUROiSTYLE, tapHOME	• Gateway • Steckdosenschalter (Ein/Aus) • Dimmbare Steckdose • Smartphone-App ”tapHOME Hausautomatisierung”
Gigaset, Gigaset elements	• Gateway • Türsensor • Bewegungssensor • Smartphone-App ”Gigaset elements”
REV Ritter, iComfort	• Gateway • 2 Steckdosenschalter (Ein/Aus) • Smartphone-App ”REV iComfort”
RWE, RWE Smart Home	• Gateway • Steckdosenschalter (Ein/Aus) • Wandschalter (2 Tasten) • Heizkörperthermostat • Online-Portal • mobiles Online-Portal • lokales Portal • Smartphone-App ”RWE SmartHome”
Telekom, QIVICON	• Gateway • Steckdosenschalter (Ein/Aus) • Heizkörperthermostat • Rauchmelder • Smartphone-App ”Smart Home”
XAVAX, MAX!	• Gateway • 2 Heizkörperthermostate • Eco-Wandschalter (Wechsel Eco-Auto) • Fensterkontakt • MAX Desktop Software • Webportal

Tabelle 2: Übersicht über die getesteten Produkte und ihre Komponenten

4 Testergebnisse

Basierend auf den Untersuchungen werden hier die Ergebnisse präsentiert. Den Anfang bilden dabei zwei zusammenfassende Tabellen, die die Sicherheitsaspekte bzw. die daraus resultierenden Gefahren aufzeigen. Im Anschluss werden die Geräte einzeln betrachtet und die Probleme bzw. Eigenschaften mehr im Detail dargestellt.

4.1 Übersicht

Die Tabelle 3 stellt die verschiedenen Sicherheitsaspekte der Produkte dar. Dies betrifft sowohl das Vorhandensein von Verschlüsselung bei der Kommunikation, als auch die Notwendigkeit einer Authentifizierung und eines aktiven Internets. An dieser Stelle findet noch keine Bewertung statt. Ist eine Verschlüsselung vorhanden, so ist irrelevant, ob diese sicher oder unsicher ist – es wird nur das Vorhandensein festgehalten.

	eSaver, iConnect	EUROiSTYLE, tapHOME	Gigaset, Gigaset elements	REV Ritter, iComfort	RWE, RWE Smart Home	Telekom, QIVICON	XAVAX, MAX!
Kommunikationsverschlüsselung							
lokal, Nutzung (z.B. Steuerung)	✔	✘	∅	✘	✔	✔	✘
lokal, Rest (z.B. Firmware-Update)	✘	✘	∅	✘	∅	?	✘
Web, Nutzung (z.B. Steuerung)	✔	∅	✔	∅	✔	✔	✘
Web, Rest (z.B. Firmware-Update)	✘	✘	✘	✘	✔	?	✘
Authentifizierung							
lokal, Nutzung	✘	✔	∅	✘	✔	✔	✘
Web, Nutzung	✔	∅	✔	∅	✔	✔	✔
Weiteres							
Internet ist verzichtbar	✔	✔	✘	✔	✘	✘	✔
offene Ports im Gateway	1	4	0	1	3	10	1

✔gegeben ✘nicht gegeben ∅nicht möglich ? unbekannt

Tabelle 3: Sicherheitsaspekte der Produkte

In der Tabelle 4 wird basierend auf den Eigenschaften der Produkte eine Einschätzung vorgenommen, ob Angreifer Geräte manipulieren oder Daten mitlesen können. Da die SSL-Verbindungen nicht direkt untersucht wurden, sondern es nur einige Indizien zur Sicherheit oder Anfälligkeit gibt, wurde für alle erst einmal der Optimalfall angenommen, sie seien sicher. Gleiches gilt für Verschlüsselungen, die nicht identifiziert werden konnten.

Mit erfasst wird, ob Daten zum Hersteller bzw. weiteren Dienstleistern gesendet werden. Dies schließt nicht-personenbezogene Daten mit ein, wie etwa die momentan laufende Version des Geräts für eine Updatekontrolle oder zum Beispiel den Standort der Wohnung. Lediglich die Zeitsynchronisation mittels NTP-Kommunikation wurde ignoriert.

In der Tabelle wurde das Stattfinden einer Kommunikation festgehalten. Daraus resultieren die potentiellen Sicherheitsprobleme der Produkte. Es findet aber keine Aussage darüber statt, welche Hürden potentielle Angreifer zu überwinden haben.

	eSaver, iConnect	EUROiSTYLE, tapHOME	Gigaset, Gigaset elements	REV Ritter, iComfort	RWE, RWE Smart Home	Telekom, QIVICON	XAVAX, MAX!
Anbieter							
Übermittlung Daten zu einem Anbieter	✔	✔	✔	✘	✔	✔	✔
Angreifer							
lokal Daten mitlesen	✔	✔	⌀	✔	✘	✘	✔
lokal Geräte manipulieren	✔	✔	⌀	✔	✘	✘	✔
remote Daten mitlesen	✔	✔	✘	⌀	✘	✘	✔
remote Geräte manipulieren	✔	⌀	✘	⌀	✘	✘	✔

✔anfällig ✘nicht anfällig ⌀nicht möglich

Tabelle 4: potentielle sicherheitsrelevante Probleme der Produkte

Neben dem Mitlesen und der Gerätemanipulation ist auch der Missbrauch der Geräte für andere Zwecke ein mögliches Motiv der Angreifer. Dies erfordert die Injizierung von eigenem Code bzw. die Infizierung der Geräte, beispielsweise per Überspielung einer eigenen Firmware auf den Geräten. Dies wurde wie bereits erwähnt in dieser Arbeit nicht weiter betrachtet und ist damit in der Tabelle auch nicht aufgeführt. Lediglich bei einem Produkt wurde die Firmware-Manipulation erprobt, in den Details ist es auch beschrieben.

4.2 Details

Jedes Produkt ist einzigartig und ist so auf eine individuelle Analyse angewiesen. Die Analysen von Metasploit bzw. Armitage waren oft ergebnislos und die Suche nach funktionierenden Exploits schlug immer fehl. Lediglich die Analyse von nmap brachte die offenen Ports der Tabelle 3 hervor und in seltenen Fällen ein paar weitere Informationen. Aus diesem Grund werden diese Angaben in den folgenden detaillierteren Aussagen zu den Produkten oft nicht erwähnt.

Festgehalten sind für jedes Produkt die Versionen der Firm- bzw. Software, sofern sie bekannt sind. Sie waren zum Zeitpunkt der Untersuchungen die jeweils aktuellsten. Von einigen gibt es mittlerweile neuere Varianten.

4.2.1 eSaver, iConnect

In den Untersuchungen kamen die Version 0.11.1 der Smartphone-App und Version R0.9.17 des Gateways zum Einsatz. Die Angaben wurden der App entnommen.

In den Funktionstests ließen sich die Steckdosen meist problemlos schalten. Nur in seltenen Fällen musste der Befehl ein zweites Mal gesendet werden.

Nach dem Start holt sich das Gateway seine IP per DHCP. Im Anschluss beginnt das Gateway periodisch Pakete per UDP-Broadcast, Port 25122, zu versenden. Es wird dabei rund alle 1,5 Sekunden ein Paket versendet.

Die versendeten Bytes sind immer gleich, es sind sechs Stück, sie lauten `07 12 10 ** ** **`. Es ist nicht bekannt, was diese Bytes bedeuten sollen. Es ist nicht die MAC des Gerätes. Es gibt zwar eine Übereinstimmung bei den letzten zwei Byte, dies wird allerdings als Zufall angesehen.

Was?	Datenblock vom Paket							⊕ (xor)				∑
	Len	Num	Key				CS					
13:28:00; Ein	07	B0	7C	68	7D	7D	6A	14	01	01	16	2FF
13:29:00; Aus	07	B3	A1	B5	A0	A1	AE	14	01	00	1D	3FF
13:30:00; Ein	07	B9	82	96	83	83	21	14	01	01	A3	2FF
13:31:00; Aus	07	BD	32	26	33	32	7E	14	01	00	4C	3FF
13:32:00; Ein	07	C1	C0	D4	C1	C1	21	14	01	01	E1	2FF
13:33:00; Aus	07	C5	0F	1B	0E	0F	EC	14	01	00	E3	3FF

Tabelle 5: Testweise Analyse von einzelnen Paketen für den ersten Schalter. Bei ⊕ sind die dem Schlüssel (Key) folgenden Bytes, als Ergebnis einer XOR-Verknüfung mit dem Schlüssel, aufgelistet.

Nach einem Verbindungsaufbau geschieht die Netzwerkkommunikation per TCP an den Port 25123. Bei der Analyse erschienen die Netzwerkpakete erst relativ zufällig, dann waren erste Muster sichtbar. Innerhalb kurzer Zeit konnte ein mögliches Protokoll identifiziert werden, siehe Tabelle 5. Das Paket besteht dabei nicht nur aus dem Befehl an sich, sondern auch aus einigen Verwaltungsbytes. Dazu gehören eine Checksumme am Ende und eine Art Schlüssel an der dritten Stelle. Der eigentliche Befehl ist mittels einer byteweisen Exklusiv-Oder-Verknüpfung (XOR) unkenntlich gemacht, in der Tabelle 3 wurde dies als Verschlüsselung gewertet. Jedes Paket hat einen eigenen Schlüssel, welcher immer mitgeschickt wird. Ansonsten beginnt jedes Paket mit der Länge in Bytes und einem hochzählenden Wert, vermutlich der internen Paketnummer.

Die erste Erprobung des potentiellen Protokolls geschah mittels netcat und Linux. Beim Start des Programms konnte per Datei ein Paket übergeben werden, welches direkt nach dem Verbindungsaufbau zum Gateway gesendet wurde. Da das vorliegende Protokoll nicht textbasiert ist, konnte netcat nur für die Versendung von einem Paket verwendet werden. Für eine Folge von Befehlen wurde deshalb ein Programm entwickelt, welches ähnlich arbeitet, aber beliebige

Binärdaten versenden kann. Es hat allerdings nur den benötigten Funktionsumfang und ist kein kompletter netcat-Ersatz.

Für das Einschalten des ersten Steckdosenschalters wurde der eigentliche Befehl `14 01 01` identifiziert, wie schon in Tabelle 5. Als Schlüssel wurde das Byte `00` gewählt, so dass die Exklusiv-Oder-Verknüpfung keine Änderung durchführt. Das um die Verwaltungsbytes erweiterte Paket wurde an das Gateway versendet, und der Schalter und damit auch das angeschlossene Gerät gingen an.

Um dies durchzuführen, mussten nur das Gateway aktiv und im selben Netzwerk, der Steckdosenschalter eingesteckt und angelernt sein. Für den Verbindungsaufbau zum Gateway war kein Passwort notwendig und auch sonst konnten keine Absicherungen bis auf das Exklusive-Oder identifiziert werden.

Nach dem Erfolg mit dem Einschalten wurden auch das Deaktivieren und die Steuerung eines zweiten Steckdosenschalters erfolgreich erprobt. Aufgrund der Beschränkung von netcat auf Textnachrichten war es allerdings notwendig, für jeden Befehl netcat zu beenden und erneut zu starten. Die Bündelung von Befehlen in ein Paket war nicht erfolgreich, weil das Gateway dann immer nur den ersten bearbeitete.

Der Befehl `14 01 00` besteht vermutlich aus dem eigentlichen Befehl `14`, hier für ”Schalten” oder so etwas, der Gerätenummer `01`, basierend auf dem Pairing, und dem neuen Wert `00`, was hier ”aus” bedeuten soll.

Byte	Typ	I-Net	Beschreibung
01	unbekannt	✔	fast immer nur das erste Pakete-Paar im Mitschnitt
02	unbekannt	✔	nur bei zwei Paaren in Mitschnitt 20
04	unbekannt	✔	?
10	unbekannt		?
12	unbekannt	✔	oft vorhanden, viele `02`-Bytes in Antwort
13	unbekannt	✔	oft vorhanden, viele `00`-Bytes in Antwort
14	Steuerung	✔	Gerätezustand setzen
A1	remote	✔	neues Gerät beim Server registrieren
A2	remote		Gateway Login-Daten für Server mitteilen
A5	remote	✔	beim Server einloggen
A6	remote		Login-Daten werden angegeben oder `FF FF FF FF 00`
FB	Fehler?	✔	Fehler?

Tabelle 6: gesehene potentielle Befehlsbytes; I-Net steht für ”im Internet gesehen”

Neben `14` gibt es wohl u.a. noch die Befehle in Tabelle 6. Es kann nicht ausgeschlossen werden, dass es weitere Befehle gibt oder dass Befehle auch aus mehr als einem Byte bestehen können.

Das System bietet optional die Steuerung aus der Ferne, mittels Remote-Zugriff, an. Dieser muss erst aktiviert werden, dabei ist ein Passwort einzugeben. Im Anschluss kann mittels App der Remote-Zugriff erfolgen. Findet die App nach dem Start kein Gateway des Produktes im Netzwerk, bietet sie den Fernzugriff als Alternative an.

Bei der Einrichtung baut die App scheinbar eine Verbindung zu einem Server im Internet – remote1.netinfostation.com, 103.1.173.1 – auf. Das vom Benutzer per App angegebene Passwort wird im selben Protokoll zum Server geschickt – Befehlsbyte `A1` – und als Antwort meldet dieser eine ID, die die App zusammen mit dem Passwort an das Gateway weiterreicht (`A2`). Dieses verbindet sich dann mit dem Server – Befehlsbyte `A5`. Bei Nutzung des Remote-Zugriffes verbindet sich auch die App mit dem Server. Beide, App und Gateway, geben am Anfang die Gateway-ID und das Passwort an – offenbar als eine Art Login. Erst danach kann die App Befehle an den Server schicken, der diese dem Gateway übermittelt.

Mit Hilfe eines selbstgeschriebenen Programmes war es möglich, den an das Gateway angelernten Steckdosenschalter per Remote zu steuern. Das Programm lief dabei auf einem Computer im Netz B, während das Gateway und Smartphone im Netz A angesiedelt sind. Im Internet haben beide Netze eine andere IP. Somit stellt der Computer einen beliebigen Rechner mit Internetzugang, irgendwo auf der Welt, dar.

Zusammenfassend kann gesagt werden, dass das System keine wirkliche Sicherheit bietet. Die Exklusiv-Oder-Verknüpfung ist schnell identifiziert und danach gibt es keine richtigen Hürden mehr. Da mit Hilfe des Protokolls auch Daten über das Internet gesendet werden, ist das System mit aktiviertem Fernzugriff angreifbar.

4.2.2 EUROiSTYLE, tapHOME

Während der Untersuchungen wurde die Version 1.0.139 der Smartphone-App verwendet. Laut App war die Firmware-Version des Gateways anfangs die 1.0.0 vom 29.01.2013. Ein Update auf 1.0.5 wurde im Laufe der Zeit angeboten, doch per App ließ sich dieses nicht überspielen. Über einen anderen Weg konnte es doch eingespielt werden, danach war es laut App aber immer noch 1.0.0, jedoch vom 05.11.2013. Es ist damit unklar, ob die anfangs angegebene Version wirklich 1.0.0 war. Es konnten keine Unterschiede vor und nach dem Update in Hinblick auf die Evaluation festgestellt werden.

Die App findet das Gateway mittels UPnP. In den beiliegenden Dokumenten wird erwähnt, dass das Gateway an einen Router mit UPnP-Fähigkeit angeschlossen werden muss. Das ”Universal Plug and Play”-Protokoll war in der Vergangenheit immer wieder Grund für Sicherheitsprobleme. Die Schnittstelle meldet sich mit „Intel MicroStack“. Es fand keine Überprüfung statt, ob auch das UPnP hier bekannte Probleme, wie Buffer-Overflows ([Cisco Security Advisory , 2013]), aufweist. Neben der App können auch Angreifer im lokalen Netzwerk – aber auch nach Hause telefonierende Fernseher und ähnliches – auf diese Weise von der Existenz des Gateways erfahren.

```
1 HTTP/1.1 200 OK
2 LOCATION: http://192.168.[...]:56598/
3 EXT:
4 SERVER: POSIX, UPnP/1.0, Intel MicroStack/1.0.1868
5 USN: uuid:cf[...]::upnp:rootdevice
6 CACHE-CONTROL: max-age=1800
7 ST: upnp:rootdevice
8
```

Listing 1: Antwort vom Gateway, 1; bei den Stellen mit `[...]` sind die genauen Werte entfernt

Der Verbindungsaufbau geschieht mittels mehrstufiger Kommunikation. Das Smartphone sendet eine Suchanfrage per Broadcast in das Netzwerk auf Port 1900. Darauf antwortet das Gateway mit drei Antworten, dargestellt in Listing 1, 2 und 3. Die Antwort kommt aber nicht vom Port 1900, sondern beispielsweise 1039. Dies ändert sich je nach Start.

```
HTTP/1.1 200 OK
LOCATION: http://192.168.[...]:56598/
EXT:
SERVER: POSIX, UPnP/1.0, Intel MicroStack/1.0.1868
USN: uuid:cf[...]
CACHE-CONTROL: max-age=1800
ST: uuid:cf[...]

```

Listing 2: Antwort vom Gateway, 2; bei den Stellen mit `[...]` sind die genauen Werte entfernt

```
HTTP/1.1 200 OK
LOCATION: http://192.168.[...]:56598/
EXT:
SERVER: POSIX, UPnP/1.0, Intel MicroStack/1.0.1868
USN: uuid:cf[...]::urn:schemas-upnp-org:device:Basic:1
CACHE-CONTROL: max-age=1800
ST: urn:schemas-upnp-org:device:Basic:1

```

Listing 3: Antwort vom Gateway, 3; bei den Stellen mit `[...]` sind die genauen Werte entfernt

Danach nimmt das Telefon eine Verbindung mit dem dort angegebenen Port auf dem Gateway auf, dieser kann ebenfalls bei jedem Start des Gateways ein anderer sein. Dieses antwortet mit dem in 4 angegebenen XML-Block. Dort stehen dann u.a. das Modell (ZE100-A-MS), der Produzent (Cybertan Technology Inc.) und der nächste Port. Eine besser lesbare Variante befindet sich in Listing 5.

```
HTTP/1.0 200 OK
CONTENT-TYPE:  text/xml; charset="utf-8"
Server: POSIX, UPnP/1.0, Intel MicroStack/1.0.1868

<?xml version="1.0" encoding="utf-8"?><root xmlns="urn:schemas-upnp-org:device-1-0"><↩
 specVersion><major>1</major><minor>0</minor></specVersion><device><deviceType>urn:schemas-↩
 upnp-org:device:Basic:1</deviceType><friendlyName>EASYGate</friendlyName><manufacturer>↩
 Cybertan Technology Inc.</manufacturer><manufacturerURL>http://www.cybertan.com.tw</↩
 manufacturerURL><modelDescription>Zwave Bridge</modelDescription><modelName>ZE100-A-MS</↩
 modelName><modelNumber>1.0</modelNumber><serialNumber>0000001</serialNumber><UDN>uuid:cf↩
 [...]</UDN><serviceList><service><serviceType></serviceType><serviceId></serviceId><↩
 controlURL></controlURL><eventSubURL></eventSubURL><SCPDURL></SCPDURL></service></↩
 serviceList><presentationURL>http://192.168.[...]:80</presentationURL></device></root>
```

Listing 4: Antwort vom Gateway auf direkte Anfrage. Bei den Stellen mit `[...]` sind die genauen Werte entfernt

Auf dem nun angegebenen Port 80 läuft ein Webserver, mit dem die eigentliche Funktionalität des Gateways ermöglicht wird.

Ist eine Verbindung aufgebaut, so kommuniziert die App mit einem auf dem Gateway laufenden Webserver – per HTTP und dabei mittels CGI-Skripten. Sie erfährt auf diese Weise, welche Geräte es gibt, welche Funktion diese haben, und steuern kann die App sie so auch. Die Verbindung ist nicht verschlüsselt, aber bei jeder Anfrage muss eine Authentifizierung mit

den HTTP-Headern mitgeschickt werden. Aufgrund des Fehlens einer Verschlüsselung können potentielle Angreifer diese mitlesen. Zudem sind der übermittelte Benutzername und das Passwort nicht vom Benutzer generierte Daten, sondern in der App hinterlegt. Es handelt sich dabei um eine Standardkombination von Benutzername und Passwort und kann daher leicht erraten werden.

```
<?xml version="1.0" encoding="utf-8"?>
<root xmlns="urn:schemas-upnp-org:device-1-0">
  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <device>
    <deviceType>urn:schemas-upnp-org:device:Basic:1</deviceType>
    <friendlyName>EASYGate</friendlyName>
    <manufacturer>Cybertan Technology Inc.</manufacturer>
    <manufacturerURL>http://www.cybertan.com.tw</manufacturerURL>
    <modelDescription>Zwave Bridge</modelDescription>
    <modelName>ZE100-A-MS</modelName>
    <modelNumber>1.0</modelNumber>
    <serialNumber>0000001</serialNumber>
    <UDN>uuid:cf[...]</UDN>
    <serviceList>
      <service>
        <serviceType></serviceType>
        <serviceId></serviceId>
        <controlURL></controlURL>
        <eventSubURL></eventSubURL>
        <SCPDURL></SCPDURL>
      </service>
    </serviceList>
    <presentationURL>http://192.168.[...]:80</presentationURL>
  </device>
</root>
```

Listing 5: lesbare Version vom XML des Gateways; bei den Stellen mit `[...]` sind die genauen Werte entfernt

Basierend auf diesen Informationen wurde versucht, den Steckdosenschalter und den Steckdosendimmer per textbasiertem netcat zu steuern. Da Benutzername und Passwort bekannt sind, stellt es keine große Herausforderung dar. Wie bereits angedeutet lassen sich nicht nur die Geräte steuern, sondern auch Informationen ermitteln. Wird beispielsweise die Webseite `http://[IPdesGateways]/cgi-bin/Get.cgi?get=SET` aufgerufen, so werden in der Antwort alle angelernten Geräte inklusive deren ID, deren Types und der von ihnen unterstützten Funktionalität aufgelistet.

Liste 7 zeigt einige bekannte Scripte mit ihrer vermutlichen Funktionalität. Der Webserver bietet noch mehr bedeutend CGI-Scripte an, wie spätere Analysen der Firmware zeigten.

Die Suche nach offenen TCP-Ports mit nmap brachte vier verschiedene Ports zum Vorschein. Zwei davon dienen wohl dem UPnP und die anderen beiden, 80 und 443, gehen zu einem Webserver. Das SSL auf Port 443 liefert ein vor Jahren abgelaufenes Zertifikat für die Webseite von CyberTAN Technology Inc., folglich wird es aus mehreren Gründen als nicht vertrauenswürdig gemeldet.

CGI	Beschreibung
`Get.cgi?get=SET`	angelernte Geräte mit ID auflisten
`getMAC.cgi`	Informationen zum Gateway ermitteln
`Status.cgi?ZID=...,...`	Informationen zu den angegebenen Geräten ermitteln
`Switch.cgi?ZID=...&OP=...`	Switch an- oder ausstellen (`OP=1` oder `=0`)
`Switch.cgi?ZID=...&VALUE=...`	Dimmer einstellen (`0-10`)
`Mode.cgi?MODE=A`	? Antwort ”Success” → anlernen?
`Mode.cgi?MODE=C`	meldet Fehler ”URL incomplete”
`command_class_switch_all.cgi?COMMAND=...`	alle Geräte auf einmal an- oder ausstellen (`SWITCH_ALL_ON` oder `SWITCH_ALL_OFF`)
`GetSchedule.cgi?ID=...`	? Zeitplan vom Gerät ermitteln ?

Tabelle 7: gesehene potentielle Befehle mit der entsprechenden Bedeutung

Beim Aufruf der Webseite auf Port 80 mittels eines Browsers erscheint eine Authentifizierungsabfrage. Werden die bereits ermittelten Login-Daten angegeben, so erscheint eine Webseite. Diese erweckt den Eindruck, bedeutend umfangreicher zu sein, als es das tapHOME-Produkt aktuell ist. So sind unter anderem offenbar auch Geräte für Überwachung der Wohnung und Verfolgung bzw. Ortung von Tieren vorgesehen. Zum Zeitpunkt der Tests bestand das tapHOME-System aus Schalter und Dimmer. Doch auch für die gab es teils mehr Funktionalität auf der Webseite als in der Smartphone-App. Der Steckdosenschalter konnte auch den Energieverbrauch messen, was auf der Webseite eingesehen werden konnte.

Neben der Einbindung von Flickr auf der Webseite gibt es beispielsweise auch Google Maps. Ferner gibt es auch Korrespondenz zu MO-SOFT. So gibt es beispielsweise den Aufruf von `http://reg.mo-soft.com/register.html?kind=meterreg&url=192.168.[...]&homeid=C75[...]&mac=D0:D6:CC:**:**:**`, wobei reg.mo-soft.com zu dem Zeitpunkt die IP 211.78.87.203 hatte. Der Aufruf ist vermutlich Teil von Ajax, darauf deutet auch der Header-Eintrag ”`Referer: http://192.168.[...]/newAjax/outerHTML/autoSend.html?L=1&HOMEID=C75[...]&MAC=D0:D6:CC:**:**:**`” hin. Die Quelle der Anfrage ist der Computer, auf dem die Webseite des EASYGATE betrachtet wird.

Beim Aufruf des Ports 443 mit einem Browser erscheint erst die Zertifikatswarnung. Nach dem die ignoriert wurde, kommt die Passwortabfrage und danach wird erst die bereits bekannte Webseite angezeigt. Es gibt also auch eine verschlüsselte Variante des Systems, doch sie reagiert wesentlich langsamer, vermutlich ist der gewählte Mikrocontroller mit der Verschlüsselung überfordert.

Bei näherer Betrachtung der Webseite fallen zwei Dinge auf. Auf der Hauptseite prangt groß ein vermutlich altes SwitchDIY-Logo der Firma Wintop. Das hierzulande nicht erhältliche Produkt iGate sieht dem Gateway von tapHOME physisch sehr ähnlich. Auf Unterwebseiten ist das Logo von Infairy der Firma MO-SOFT zu sehen. Deren Hausautomatisierungs-Websystem sieht dem auf dem tapHOME-Gateway sehr ähnlich, sie haben aber mehr Geräte als nur Steckdosenschalter und Dimmer im Angebot, die ebenfalls vom Webportal unterstützt werden.

Trotz Suche nach Hinweisen konnten bei den Dokumenten zum Produkt keine Informationen zum oder auf das Webportal gefunden werden. Viele Benutzer werden davon nichts mitbekommen und sich somit auch keine Gedanken um Absicherung oder irgendwelche Probleme machen.

Sowohl die App, als auch die Weboberfläche bieten eine Möglichkeit der Zeitschaltung an. Anfangs wurde nicht die Version in der App verwendet. Nach der Neuinstallation der App waren die Profile weg und auch in der Weboberfläche wurden keine aufgelistet. Es scheint, als würde die Funktionalität des Gateways an dieser Stelle nicht genutzt werden.

Eine Analyse der Webzugriffe ergab, dass das System kaum Kontakt ins Internet aufnimmt, wenn nicht gerade der „Bildschirmschoner“ des Webportals vom Gateway aktiv ist. Beim Start jedoch nimmt das Gateway Kontakt zu Seiten von MO-SOFT auf und überträgt dabei Daten wie MAC und lokale IP.

Bei der verwendeten Firmware handelt es sich wohl um ein Linux von CyberTAN. Hinweise auf die GPL oder den Lizenztext selber wurden vergebens gesucht. Basierend auf den ELF-Dateien handelt es sich um eine ARM-Architektur, laut dem z-Wave-Modul bzw. der Linux-Bezeichnung ARMv4. Der Webserver unter `/usr/bin/webs` stellt vermutlich die CGI-Scripte zur Verfügung. Dazu zählen auch `webtest.cgi`, `CameraView.cgi`, `putHealthData.cgi`, `command_class_protection.cgi`, `newUser.cgi`, `getPW.cgi` und `SetBurglar.cgi`. Es scheint OpenSSL 0.9.7g von 2005 verwendet zu werden.
Das Passwort des allmächtigen root-Benutzers kann leicht erraten werden.

Zusammenfassend muss festgestellt werden, dass das System nahezu keinen Schutz vor Angreifern im lokalen Netzwerk bietet. Da es keinen Fernzugriff gibt, muss der Angreifer allerdings erst einmal ins lokale Netzwerk gelangen, beispielsweise per Schadsoftware auf den PC. Ist der Zugang aber erst einmal da, kann der Angreifer machen, was er will. Im schlimmsten Fall gibt der Angreifer den Port des Webportals ins Internet frei und kann dann zu jeder Zeit die Geräte nutzen – zu denen laut Funktionalität prinzipiell auch bereits angekündigte Überwachungskameras gehören. Es ist davon auszugehen, dass viele Benutzer diesen Schaden nicht rückgängig machen, selbst wenn sie z.B. die Schadsoftware entdeckt und entfernt haben; schließlich wissen sie nichts vom Webportal.

Im Januar 2014 wurde eine neue Version, 2.0.141, der Smartphone-App veröffentlicht. Diese unterstützt nun, laut Liste der Änderungen, auch die Energiemessung. Die Veröffentlichung kam für die Berücksichtigung bei den Tests zu spät.
Ebenfalls im Januar wird ein aktualisierter Fahrplan auf der Webseite angegeben. Nach diesem soll u.a. im März/April des Jahres 2014 der Fernzugriff aktiviert werden. Die Verzögerung wird mit der Vermeidung von Sicherheitslücken begründet. [EUROiSTYLE GmbH , 2014a]
Im März wurde eine neue Version, 2.0.146, der App veröffentlicht. Unter anderem wurde die Möglichkeit hinzugefügt, das Passwort des Gateways zu verändern. In Klammern wurde ”mehr Sicherheit” angegeben. [EUROiSTYLE GmbH , 2014b]

4.2.3 Gigaset, Gigaset elements

Die Tests für Gigaset elements erfolgten mit der Smartphone-App-Version 1.1.3. Das Gateway hatte eine Vorgängerversion der Firmware 001.000.026, welche genau, ist unklar.

Schnell fiel auf, dass das System zwangsweise Internet-Zugang benötigt. Am ersten Testtag war das nötige Rechenzentrum nicht erreichbar. Infolgedessen mussten die Tests abgebrochen und am nächsten Tag fortgesetzt werden.

Die Verbindungen zu den von Gigaset genutzten Servern verwenden SSL zur Verschlüsselung, genauer TLS 1.0. In den betrachteten Stichproben wurde vom Server immer `TLS_RSA_WITH_AES_128_CBC_SHA` als Verfahren ausgewählt. Mit BEAST-Attack und Lucky-Thirteen-Attacke gibt es potentielle Angriffe gegen den CBC-Modus bei TLS 1.0 ([Böck, 2013]). Gängige Implementationen von SSL-Bibliotheken sind allerdings auch gegen diese abgesichert. Basierend auf diesen Informationen kann die Verbindung prinzipiell als sicher angesehen werden.

Auch für die von der Smartphone-App aufgebauten Verbindungen zu den Servern von Gigaset gelten diese Aussagen. Allerdings sind die Ziel-IPs andere. Alle IPs stammten aus dem 217.150.144.0/24-Pool.

In den Funktionstests wurde nach entsprechendem Aufbau das Öffnen und Schließen der Tür zuverlässig erkannt. Bei den starken Erschütterungen eines simulierten Einbruchs wurde dies entsprechend gemeldet. Teilweise gab der Türsensor auch eine Minute lang ein Piepen von sich.

Auch der Bewegungsmelder reagierte auf Veränderungen im Raum, dies wurde aber gefiltert – wie es auch aus den FAQs zu dem Produkt hervorgeht –, so dass nicht jede Bewegung auch eine Meldung auslöste.

Basierend auf den Zeitstempeln wurden alle diese Meldungen immer zeitnah weitergegeben. Es war höchstens eine Differenz von einer Sekunde auszumachen. Jedoch war es teils notwendig, die App manuell zu aktualisieren, damit diese die neuen Informationen auch mitbekam und beispielsweise den Alarm auch anzeigte.

Anders verhält es sich mit den Offline-Meldungen. Antwortet ein Gerät einige Zeit lang nicht, so generiert dies ebenfalls eine Meldung. In den Untersuchungen lagen in der Regel zwischen der Entfernung der Batterie und der Meldung über den abwesenden Sender um die 40 Minuten. In seltenen Fällen waren es auch nur 30 Minuten, in anderen dauerte es deutlich länger. Ein Angreifer könnte theoretisch die Funkverbindung stören und dann ins Haus einbrechen. Er hätte so fast eine dreiviertel Stunde zum Ausräumen der Wohnung, bevor der Besitzer überhaupt erst die Möglichkeit hätte, Verdacht zu schöpfen. Lediglich die Abwesenheit der Basisstation wurde schon nach wenigen Minuten vermeldet. Ein Kappen der Internetverbindung führt also nicht zum gewünschten Erfolg.

Während der Tests unternahm die Basisstation den Versuch, Firmwareupdates einzuspielen. Dies geschah anders als erwartet nicht per HTTPS, sondern mit HTTP für den Erhalt der Dateien. Diese Update-Verbindungen waren auch die einzigen unverschlüsselten, die in dem Test gesehen werden konnten.

Basierend auf späteren Untersuchungen konnte der wahrscheinliche Ablauf des Updates rekonstruiert werden. Das System besteht hauptsächlich aus zwei Partitionen, auf der einen Partition sind die ausführbaren Dateien hinterlegt, auf der anderen Daten wie die Konfigurationsdateien. Von beiden Partitionen gibt es „Kopien“. Diese beherbergen eine andere Version der Firmware. Beim Update werden die Kopien auf den neusten Stand gebracht und danach ein interner Schalter geändert. Dieser entscheidet, welches Partitionspaar aktiv ist und welches für das nächste Update genutzt werden soll.

Der Updatevorgang besteht aus zwei Schritten. Im ersten Schritt werden die eigentlichen Firmware-Dateien aus dem Internet geladen und in die Update-Partitionen überspielt. Im zweiten Schritt werden die Daten der aktiven in die Update-Partitionen kopiert und die Konfiguration notfalls angepasst. Zum Schluss findet der Wechsel der aktiven und der Update-Partitionen statt.

Die Basisstation lädt in regelmäßigen Abständen ein Tar-Archiv aus dem Internet herunter und validiert es mit einer hinterlegten Checksumme. Ist diese korrekt, so wird das Paket – bestehend aus ausführbaren Dateien – entpackt und eine aktive Komponente ausgeführt. Diese benutzt dann wiederum die anderen Dateien des Archivs. Diese Dateien dienen der Durchführung des Updates. Für jede Version gibt es ein eigenes Tar-Archiv.

Es werden weitere Dateien aus dem Internet heruntergeladen. Bei diesen handelt es sich um Checksummen-Dateien mit den Namen `up_vmlinuz.bin.sum` und `up_fs.bin.sum`. Mit diesen werden die für das Update hinterlegten Partitionen kontrolliert. Stimmen die Checksummen nicht, werden die eigentlichen Firmware-Dateien aus dem Internet heruntergeladen und in die Partitionen kopiert. Beim nächsten Durchgang stimmen dann damit die Checksummen und es ist kein erneuter Download notwendig.

Unabhängig davon, ob die Firmware tatsächlich oder nur deren Checksummen heruntergeladen wurden, meldet das System, dass eine neue Firmware geladen wurde. Der Nutzer muss dann noch manuell die Firmware installieren, dabei werden hauptsächlich ein paar Dateien kopiert und intern auf die neue Firmware gewechselt.

Basierend auf diesem Vorgang wurde eine Möglichkeit entwickelt, dem System eine andere Firmware statt der echten unterzuschieben. Dazu wurde ein eigener DNS-Server aufgesetzt. Für die Domain des Update-Servers gibt dieser die IP eines eigenen Webservers heraus, ansonsten wird die korrekte Antwort erteilt. Dem System wurde danach dieser DNS-Server gegeben. Fortan liefen die SSL-Verbindungen der Basisstation noch über die originalen Server, für das Update nahm die Station aber Kontakt zum eigenen Webserver auf. Auf diesem wurde zwar die original Tar-Datei des Updates 001.000.026 hinterlegt, die Firmware-Dateien waren aber andere. Im Detail handelt es sich um die der Vorgänger Version 001.000.023. Die dafür notwendigen Checksummen-Dateien wurden selber erstellt.

Beim nächsten Update-Versuch lud die Basisstation erst die Tar-Datei, danach die Checksummen-Dateien und im Anschluss die realen Firmware-Dateien. Beim nächsten Durchgang wurden nur noch die Tar-Datei und die Checksummen-Dateien heruntergeladen. Das System hatte die falsche Firmware akzeptiert.

Die Firmware kann auch per Browser heruntergeladen werden. Es handelt sich dabei um die Dateien `up_vmlinuz.bin` und `up_fs.bin`. Die erste Datei entpuppte sich als ein Linux mit 2.6.19-Kernel – Ende 2006 veröffentlicht – welcher den Zusatz ”uc1reef-dirty” trägt. Zwar mel-

det die Firmware M68K als verwendete Prozessorarchitektur, aber nach ersten Untersuchungen erscheint Xilinx MicroBlaze oder CR16C+ als wahrscheinlicher. Die Information bzgl. MicroBlaze entstammt der einzigen ELF-Datei, die auf dem System gefunden wurde. Bei `/lib/modules/2.6.19-uc1reef-dirty/kernel/drivers/dect/rtxdect452.ko` handelt es sich wohl um einen proprietären DECT-Treiber von RTX A/S, Morten Laursen. Es lassen sich dort auch die Begriffe NatalieV3 und REEF finden. In den ELF-Dateien wird die Architektur mit hinterlegt. Die Angaben zu CR16C+ entstammen der Datei `/usr/bin/coco`, in der der Build-Befehl hinterlegt ist. Dieser beginnt mit `cr16-uclinux-gcc` und hat auch das `-mcr16cplus` Flag gesetzt. Ferner ist `-DSC14450` gesetzt, was auf den SC14450 als Mikrocontroller hinweisen könnte, dies wäre nicht das erste Mal, dass Gigaset diesen wählen würde [Mobile Dev & Design, 2009].

Nach Extraktion des Dateisystems konnte dieses ebenfalls betrachtet werden. Viele Programme werden mittels busybox, in Version 1.20.2 von Mitte 2012, umgesetzt. Zu ihnen zählt auch telnet, ein entsprechender Dienst wird aber nur bei ”unlocked”-Systemen gestartet, bei der untersuchten Hardware war dies nicht der Fall. Ein als ”coco” hinterlegtes Programm beinhaltet vermutlich OpenSSL 1.0.1b vom April 2012. Damit enthält die Firmware einen veralteten Kernel und leicht veraltete Programme. Besonders bei OpenSSL könnte sich dies als Fehler erweisen. Zusätzlich zu weiteren Dateien wurde auch ein Bündel an SSL-Zertifikate gefunden. Auf diese Weise ist es prinzipiell möglich, dass die Basisstation klassisches Man-in-the-Middle bei den SSL-Versionen mitbekommen kann.

In der Datei `up_fs.bin` befindet sich ein fast leeres JFFS2-Dateisystem. In ihm befindet sich nur die leere Datei `valid_fs`. Bei einem Update wird also ein leeres Dateisystem angelegt, in das dann die Daten wie Konfigurationsdateien hineinkopiert werden.

Zusammenfassend kann gesagt werden, dass Gigaset elements eine gute Basis in Punkto IT-Sicherheit bietet. Es gibt jedoch einige potentielle Gefahrenquellen, die an dieser Stelle nicht weiter untersucht und bei weitergehenden Untersuchungen betrachtet werden können. Einige Bedenken könnte Gigaset durch die Umstellung auf TLS 1.2 von 2008 beheben. Da sie theoretisch Client und Server kontrollieren, sollte eine Umstellung auch kein Problem darstellen. Einzig die potentielle Anfälligkeit gegen Einspielung falscher Firmware ist bedenklich. Die dafür notwendigen Bedingungen werden in der Praxis allerdings wohl selten erfüllt sein, ein Risiko bleibt es aber dennoch.

Mittlerweile gibt es neue Versionen der Firmware. Ende Januar erschien die Version 1.0.28 und Anfang Februar 1.0.29. Nun wird in der Tar-Datei der Download der Firmware-Versionen über HTTPS durchgeführt, außer es laufen die Versionen 1.0.17, 1.0.26 oder 1.0.13. In Listing 6 ist der entscheidende Part dargestellt.

Die dafür nötige Angabe von `BAS_TAG` wird vermutlich im Initialisierungs-Script `/etc/init.d/S40reef.sh` erstellt. Entsprechender Ausschnitt ist in Listing 7 dargestellt.

Somit sollte die Gefahr nur noch bei den installierten Versionen 1.0.13, 1.0.17 und 1.0.26 oder beim Upgrade auf eine Version vor 1.0.28 bestehen. Dabei wird immer vorausgesetzt, dass die neue SSL-Verbindung ebenfalls sicher ist, also das Programm `coco` beispielsweise das Zertifikat überprüft.

```
154 ######################################
155 #Download requested file
156 #Use wget when ver.13 or 17 or 26, coco otherwise
157 #$1 - name of file which is requested to be downloaded
158 #$2 - path where downloaded file will be saved
159 DownloadFile()
160 {
161     rm -f ${2}
162     if [ ${BAS_TAG} = "bas-001.000.017" ] || [ ${BAS_TAG} = "bas-001.000.026" ] || [ ${↩
      BAS_TAG} = "bas-001.000.013" ];then
163         LOG "Basestation␣${BAS_TAG}␣detected."
164         LOG "Downloading␣file␣from␣'${BASE_URL_HTTP}/${1}'␣to␣'${2}'"
165         wget ${BASE_URL_HTTP}/${1} -O ${2}
166     else
167         LOG "Downloading␣file␣from␣'${BASE_URL}/${1}'␣to␣'${2}'"
168         coco --wget -i ${BASE_URL}/${1} -o ${2}
169     fi
```

Listing 6: Ausschnitt von fw_upgrade_func.sh aus Version 001.000.029

```
92          export BAS_TAG=`cat /proc/version | grep -w "Reef␣BS␣version" | cut -d \' -f 2`
93          export BAS_HASH=`cat /proc/version | grep -w "Reef␣BS␣hash" | cut -d \' -f 2`
```

Listing 7: Ausschnitt von /etc/init.d/S40reef.sh aus der Firmware 001.000.026

4.2.4 REV Ritter, iComfort

In den Untersuchungen wurden die Version 1.5 der Smartphone-App, sowie die Gateway-Versionen 1.0 und 1.1 verwendet. Die Angaben entstammen der App.

Die App und das Gateway finden sich durch UDP-Broadcast-Pakete über Port 5556. Dabei sendet das Gateway Nachrichten, die aus ”REV ICOMFORT”, der MAC-Adresse und eventuellen Statusmeldungen bestehen. Eine Beispielnachricht ist in Listing 8 dargestellt.

```
1 REV␣ICOMFORT
2 F8-D7-BF-**-**-**
3 DHCP/Power␣event␣occurred
```

Listing 8: Beispiel UDP-Broadcast vom Gateway, Teile der MAC wurden durch ** ersetzt

Sollte das Gateway nicht gerade gestartet sein, so kann auch eine Broadcast-Nachricht von der App, mit dem Inhalt ”D” gesendet werden, woraufhin das Gateway dann antwortet. Angreifer im Netzwerk können so auch versuchen, das Gateway zu finden.

Haben sich App und Gateway erst einmal gefunden, so kommunizieren sie über TCP-Port 5555.

Fast alle gesendeten Nachrichten haben einen URL-ähnlichen Aufbau. Der allgemeine Aufbau ist

`[Buchstabe]?[[Parameter]=[Wert]][&[Parameter]=[Wert]][...]\n`

Ein Beispiel dafür ist `S?d=3F&a=01234567`, es beginnt mit einem Buchstaben, hier `S` als den eigentlichen Befehl, gefolgt von einem Fragezeichen und danach eine Liste von Parametern, die jeweils durch ein Kaufmanns-Und voneinander getrennt sind. In diesem Beispiel wird der

Parameter d mit dem Wert 3F und der Parameter a mit dem Wert 01234567 übergeben. Den Abschluss bildet ein Linux-artiger Zeilenumbruch \n bzw. hexadezimal 0x0A.

Die Kommunikation scheint immer textbasiert zu sein. Als gültige Buchstaben konnten A, B, E, F, G, I, R, S und T gesehen werden. Es gibt auch W, allerdings endet das Paket dann nicht auf \n.

Mit B kann das Gateway neu gestartet werden. Tritt ein Fehler auf, so wird dies mit F kenntlich gemacht. Durch G ist es möglich, den Status eines Gerätes abzufragen, auch beginnt die Mitteilung des Status mit diesem Buchstaben. Mit I können Informationen zum Gateway ermittelt werden, wie z.B. die Netzwerkkonfiguration. Durch R können Informationen zu einem Gerät angefragt werden. S ermöglicht das Setzen eines neuen Gerätezustandes. T könnte evtl. das Setzen des Zeitstempels vom Gateway sein und W wird möglicherweise für die Zeitschaltung verwendet. Das Frame ist zu groß und wird deshalb geteilt.

Verschiedene Befehle konnten ermittelt und mit Hilfe von netcat genutzt werden, um die Geräte zu schalten. Wichtig war dabei, dass die App nicht läuft. Das Gateway erlaubt nur eine TCP-Verbindung gleichzeitig, der Zugriff ist exklusiv. Der bereits angegebene Befehl S?d=3F&a=01234567 würde die Steckdosenleiste mit der ID 01234567 einschalten. Das System antwortet zweimal darauf und gibt den aktuellen Zustand mit einer G-Nachricht zurück. Es ist zu vermuten, dass die erste Antwort vom Gateway und die andere vom geschalteten Gerät kommt. Dafür gibt es mehrere Indizien. Eines ist die Korrektur des Wertes. Wird beispielsweise d=20 hingesendet, so liefert die erste Antwort diesen Wert, die zweite aber den alten, und das Gerät bleibt ebenfalls unverändert.

```
R?h=0
R?a=FFFFFFFF&d=FF&h=0&y=65535&z=1111111
R?h=1
R?a=00012[...]&d=3F&h=1&y=794&z=0001100
R?h=2
R?a=0000D[...]&d=3F&h=2&y=796&z=0000100
R?h=3
R?a=FFFFFFFF&d=FF&h=3&y=65535&z=1111111
```

Listing 9: Beispiel Geräte abfragen; bei den Stellen mit [...] sind die genauen Werte entfernt

Mit Hilfe eines anderen Buchstabens ist es möglich, die installierten Geräte abzufragen. Dies wird mit R?h=... durchgeführt. Die App geht dabei die Zahlen ab 0 durch. Die Zahl gibt vermutlich die Nummer in der internen Liste der Geräte an. Es wurde nie gesehen, dass bei 0 ein sinnvoller Wert zurückgegeben wurde. In den Tests fing es immer erst bei 1 an. Listing 9 stellt eine beispielhafte Abfrage mit den entsprechenden Antworten dar. Dabei sind h=1 und h=2 mit einem Gerät versehen. Die App fragt mindestens bis h=31 ab.

```
W?d=00&z=0000000&y=794&h=1&a=00012[...]
W?d=00&z=0001100&y=794&h=1&a=00012[...]
W?d=3f&z=0001100&y=794&h=1&a=00012[...]
W?d=00&z=0000000&y=795&h=2&a=0000D[...]
W?d=3f&z=0000000&y=795&h=2&a=0000D[...]
W?d=3f&z=0000000&y=796&h=2&a=0000D[...]
W?d=3f&z=0000100&y=796&h=2&a=0000D[...]
```

Listing 10: Beispiel für potentielle Zeitschaltungen; bei den Stellen mit [...] sind die genauen Werte entfernt

Während der Tests mit der Zeitschaltung ergab sich, dass die App Befehle mit `W` versendet. Beispiele sind in Listing 10 zu sehen. Das Gateway gibt an, dass seine TCP-Fenster-Größe 20 Byte groß ist. Die Angaben sind aber größer, also teilt wohl TCP/die App das Ganze in mehrere Teile. So wird z.B. zuerst `W?d=00&z=0000000&y=7` gesendet und Wireshark gibt dort "TCP Window Full" an – 20 von 20 Byte sind belegt. Danach wird `94&h=1&a=00012[...]` gesendet. So erschien es, dass einige Pakete nicht mit einem Buchstaben anfangen und `W` als einziger ohne endenden Zeilenumbruch gesichtet wurde. Bei den Parametern gibt das z möglicherweise den Tag an, jede Ziffer steht dann für einen Tag der Woche.

Ein weiterer existierender Befehl ist `B`. Mit diesem lässt sich das Gateway neu starten. Ein Angreifer könnte somit prinzipiell das Gateway ständig neu starten und dem Käufer so die Benutzung unmöglich machen. Da allerdings der Zugriff auf das Gateway exklusiv ist, reicht es auch aus, dass der Angreifer eine Dauerverbindung aufbaut. Somit ist die App des Benutzers nicht in der Lage, Kontakt zum Gateway aufzunehmen, seine Geräte kann er so nicht mehr steuern.

Ein Fernzugriff existiert nicht, ein Angreifer ist also gezwungen, erst einmal das lokale Netzwerk zu infiltrieren.

Die App bot die Möglichkeit eines Firmware-Updates an. Auch dieses wurde betrachtet, die Firmware wird in vielen kleinen Paketen per UDP an das Gateway geschickt, nachdem dieses neu gestartet wurde. Eine Absicherung dieser Übertragung konnte nicht beobachtet werden. Lediglich die Firmware selbst scheint verschlüsselt zu sein. Die Struktur lässt ein Verfahren vermuten, welches heutzutage nicht mehr als sicher, aber für rechenschwache Geräte noch in Software umsetzbar ist. Dies wurde nicht weiter untersucht, so dass es reine Spekulation bleibt.

Zusammenfassend kann gesagt werden, dass sich dieses Produkt auf die Sicherheit des lokalen Netzwerkes verlässt. Hat der Angreifer erst einmal diese Hürden überwunden, kann er mit dem System machen, was er will, es gibt wohl keinerlei Absicherung. Lediglich die Firmware kann scheinbar nicht trivial ersetzt werden.

4.2.5 RWE, RWE SmartHome

Zum Einsatz kamen die Version 1.60.426.0 der Software und 1.60.356.0 von der Zentrale. Diese Angaben stammen aus dem Webportal. Die Bezeichnung Software stammt vom Hersteller. Der Benutzer bedient an sich nur Webportale oder eine App.

Die Einrichtung bzw. Installation beginnt über ein Webportal. Zu diesem Zeitpunkt muss die Zentrale bereits im Internet und mit einem Server von RWE verbunden sein. Sie zeigt dann auf dem Display eine Nummer an, die im Webportal zusammen mit der Seriennummer der Zentrale angegeben werden muss. Beim ersten Start der Zentrale mit Internet lud diese sofort ein Update und war deshalb mehrere Minuten lang nicht zu benutzen. Vor der ersten Einrichtung kann die Box lokal nicht genutzt werden, da ein Benutzername und Passwort erwartet werden – beides wird erst während der Einrichtung der Box mitgeteilt.

Nach Einrichtung eines Benutzerkontos gibt es eine Weiterleitung auf ein Webportal mit Silverlight. Dort ist ein Login mit dem zuvor eingerichteten Account notwendig. Danach kann auch eine Version der nötigen „Software“ heruntergeladen werden, welche dann auch ohne Internetanbindung läuft. Die Zentrale benötigt aber weiterhin immer dann einen Internet-Zugang, wenn Änderungen an der Konfiguration, wie die Installation eines Gerätes oder die Erstellung bzw. Änderung von Automatisierungsregeln, durchgeführt werden sollen.

Zusätzlich zu dem beschriebenen Webportal gibt es noch eine mobile Version und eine Smartphone-App. Darin ist es nicht möglich, die bestehende Konfiguration zu ändern. Es können nur die bestehenden Geräte und Regeln benutzt werden. Zusätzlich ist wohl ein kostenpflichtiges Abonnement für die Benutzung dieses Dienstes notwendig. In dieser Variante ist dann auch kein Silverlight mehr erforderlich.

Die Verbindungen der einzelnen Portale wurden überprüft. Es wird zumeist SSL in der Version TLS 1.0 verwendet. Es konnten in den Stichproben vier verschiedene Verschlüsselungen festgestellt werden. Die drei häufigsten waren `TLS_RSA_WITH_`**`3DES`**`_EDE_CBC_SHA`, `TLS_RSA_WITH_`**`AES_128_CBC`**`_SHA` und `TLS_RSA_WITH_`**`RC4`**`_128_SHA`. Auch das Firmware-Update wird über eine entsprechend gesicherte Verbindung übertragen.

Doch es wurden auch unverschlüsselte Verbindungen zu `relay.rwe-smarthome.de` entdeckt. In regelmäßigen Abständen von rund 15 Minuten wird dort eine Webseite aufgerufen und der Anfrage vermutlich eine Zeitangabe beigefügt. Der vermeintliche Zeitstempel könnte die Angabe der verstrichenen Zeit in 100 ns-Intervallen ab dem 01.01.0000, 00:00 Uhr sein. Dies ergibt sich aus der Differenz zweier Werte – $\frac{9169730000}{10 \cdot 1000 \cdot 1000} = 916,973$ bei einem zeitlichen Unterschied von 917 Sekunden – und der groben Rückrechnung eines absoluten Wertes auf das Jahr – $\frac{63525384570285}{(1000 \cdot 60 \cdot 60 \cdot 24 \cdot 365)} \approx 2014,38$. Es scheint sich bei der HTTP-Verbindung um eine Art Kontrolle einer Internetverbindung zu handeln, da der entsprechende Server offenbar bei einer ausbleibenden Verbindung mittels eines ICMP-basierten Pings auf Erreichbarkeit überprüft wird. Erhielt die Zentrale von dem Webserver eine Antwort, so konnte bei den Paketmitschnitten kein Ping-Versuch erkannt werden.

```
# cat /var/flash/ar7.cfg > /var/tmp/ar7.cfg
# vi /var/tmp/ar7.cfg
# cat /var/tmp/ar7.cfg > /var/flash/ar7.cfg
# ar7cfgchanged
[...]
```

Listing 11: telnet-Befehle zur Anpassung

Basierend auf der für diese Untersuchung gestellten Annahme, dass SSL sicher sei, gibt es hier nur die eine unverschlüsselte Verbindung als Ansatzpunkt. Aus diesem Grund wurde der verwendete Router so konfiguriert, dass alle Anfragen abgeblockt wurden, die an die IP hinter `relay.rwe-smarthome.de` geschickt wurden. Dies geschah, indem telnet auf der Fritz!Box per Fake-Firmware-Update aktiviert wurde. Im Anschluss wurde sich mit der Box verbunden und die Datei `/var/flash/ar7.cfg`, wie in Listing 11 und 12 beschrieben, verändert und die Änderung im laufenden Betrieb übernommen. Dabei wurde unter `dsldpconfig`, `highoutput`, `accesslist` der letzte Eintrag mit der IP 193.25.80.73 hinzugefügt. Auf diese Weise schlagen sowohl die HTTP-, als auch Ping-Versuche fehl. Die ganze andere, SSL-

basierte Kommunikation zu anderen Servern kann nach wie vor durchgeführt werden und findet laut Mitschnitten auch weiterhin statt. Mit 193.25.80.70 – services.rwe-smarthome.de – wurde kommuniziert. Nachdem die Relay-Verbindung nicht aufgebaut werden konnte, endet vorerst auch die Kommunikation. Der Port wird aber nicht geschlossen. Nachdem das lokale Portal nach drei Minuten geöffnet wurde, wird die TCP-Verbindung in drei Schritten, ausgehend von der Zentrale, ordnungsgemäß mit FIN beendet.

```
dsldpconfig {
        security = dpsec_firewall;
        filter_teredo = yes;
        filter_netbios = yes;
        lowinput {
                policy = "permit";
                accesslist =
                        "deny ip any 242.0.0.0 255.0.0.0",
                        "deny ip any host 255.255.255.255";
        }
        lowoutput {
                policy = "permit";
        }
        highinput {
                policy = "permit";
        }
        highoutput {
                policy = "permit";
                accesslist =
                        "reject ip any 242.0.0.0 255.0.0.0",
                        "deny ip any host 255.255.255.255",
                        "reject ip any 169.254.0.0 255.255.0.0",
                        "reject ip any 193.25.80.73 255.255.255.255";
        }
        forwardrules = "[...]";
}
```

Listing 12: Ausschnitt der geänderten ar7.cfg-Datei; bei den Stellen mit `[...]` sind die genauen Werte entfernt

Die Zentrale reagiert auf die Blockierung so, als sei kein Internet vorhanden, eine Konfiguration ist nun nicht mehr möglich. Es können also nur noch vorhandene Geräte und bestehende Automatisierungen genutzt werden. Neue Geräte und Automatisierungsregeln können aber nicht mehr hinzugefügt werden. Vorhandene Regeln können nicht mehr verändert werden. Ein Angreifer könnte auf diese Weise die Zentrale zu einer Offline-Version degradieren, ohne dass das Internet komplett weg ist. Für den Angreifer wäre es wohl leichter, die komplette Internetkommunikation der Zentrale zu unterbinden und so denselben Effekt zu erzielen.

Um die Zentrale zu finden, fragt das lokale Portal offenbar nach lokalen DNS-Auflösungen der Client-Namen smarthome01 bis smarthome10 ab. Bei smarthome09.fritz.box kam die IP der Zentrale zurück. Im Anschluss wird per TCP eine SSL-Verbindung zu 443 aufgebaut.

Während der Tests wurden immer wieder diverse Domainnamen aufgelöst. Neben einigen mit RWE im Namen gab es auch welche von Microsoft, Amazon-Cloud, Symantec, GeoTrust und etracker. Während GeoTrust wohl vom Smartphone stammt, konnte etracker nur bei dem Test dieses Produktes gesichtet werden. Es ist deshalb berechtigt anzunehmen, dass beim RWE Smart Home auch die Dienste des Datensammlers genutzt werden.

Zur Überprüfung der gesperrten Konfiguration bei fehlender Internetverbindung wurde die lokale Version des Webportals gestartet und auf die Zentrale eingeloggt. Im Anschluss wurde das Netzwerkkabel physisch gezogen und das Portal danach weiter bedient. Während einer kurzen Zeit war es noch möglich, nach neuen Geräten suchen zu lassen und auch neue Regeln anzulegen. Erst nachdem die Zentrale per Display die Abwesenheit des Internets anmerkte, verweigerte auch das Portal weitere Konfigurationsversuche. Die Restriktionen scheinen damit vom Hersteller vorgegeben und nicht direkt durch das gewählte Design hervorgerufen zu sein. Eine Konfiguration ist möglich, solange die Zentrale nur denkt, sie hätte Internet-Zugang.

Die ohne Internetverbindung angelegte Regel blieb nicht nur nach einem Neustart erhalten, sondern überlebte auch die Entfernung der Stromversorgung. Nachdem die Zentrale wieder Internetzugriff hatte, scheint sie die neue Regel auch dem Server mitgeteilt zu haben.

Zusammenfassend kann gesagt werden, dass RWE Smart Home basierend auf den Annahmen dieser Untersuchungen im Großen und Ganzen als sicher angesehen werden kann. Einzig die unverschlüsselte Verbindung mit dem Relay-Server stellt ein theoretisches Problem dar.

4.2.6 Telekom, QIVICON (Smart Home)

Die während der Tests wohl aktuelle Version 1.3.4s der Gateway-Firmware war auf dem QIVICON Home Base genannten Gerät installiert. Die Smartphone-App der Telekom gab sich mit der Version 1.2.822/1.3.3 zu erkennen. Der originale Name des Starter-Kits der Telekom lautet „Smart Home“.

Für die Einrichtung des Systems ist eine Internetverbindung notwendig. Sonst ist keine Nutzung möglich. Zu diesem Zweck ist eine Registrierung der Person inkl. Wohnanschrift und des Gateways beim Anbieter Pflicht. Während dieses Vorganges können Geräte auch gleich angelernt werden. Den Abschluss der Einrichtung ermöglicht die Installation der Telekom-App auf der Home Base. Das System sieht also Möglichkeiten der Erweiterung vor. Laut Hersteller lassen sich nur signierte Anwendungen auf der Home Base installieren. Dies wurde aber im Rahmen dieser ersten Untersuchungen nicht überprüft.

Die Nutzung der Telekom-App auf dem Gateway ist zeitlich begrenzt. Zusätzlich gibt es auch noch eine passende Smartphone-App, die aber separat aus dem entsprechenden Store heruntergeladen und installiert werden muss.

Während der kompletten Untersuchungen konnte weder ein Firmware-Update, noch eine unverschlüsselte Verbindung ins Internet oder ins lokale Netzwerk gesehen werden. Es wurde scheinbar immer TLS 1.0 eingesetzt. Als Verschlüsselungsverfahren kam in allen Stichproben `TLS_RSA_WITH_RC4_128_SHA` zum Einsatz.

Ist das System erst einmal eingerichtet, kann es sowohl per Webportal als auch per lokaler Anbindung gesteuert und konfiguriert werden. Die lokale Variante ist nach der Einrichtung auch benutzbar, wenn das System komplett – also auch das Gateway – ohne Internetzugriff ist. Bei der lokalen Anbindung handelt es sich um ein Webportal, dessen Server das Gateway ist. Um es nutzen zu können, muss z.B. die IP des Gateways im Browser eingegeben werden.

Sollte die unverschlüsselte Webseite auf Port 80 abgerufen werden, so wird automatisch auf die verschlüsselte HTTPS-Version auf Port 443 umgeleitet.

Das Zertifikat der lokalen Webseite ist kein offizielles, sondern ein selbst signiertes Zertifikat. Es wird von den Browsern als ungültig gemeldet, lässt sich aber akzeptieren. Die Ersetzung bei einer lokalen Man-in-the-Middle-Attacke sollte somit theoretisch nicht auffallen. SSL wurde aber wie bereits gesagt nicht näher untersucht.

Die beiden Portale, das Webportal im Internet als auch das lokale, sehen offenbar gleich aus und haben dieselbe Funktionalität. Im Vergleich zu den anderen Produkten war der Funktionsumfang zu der Zeit der Untersuchungen eher marginal. Die Smartphone-App mit anderem Aussehen und anderer Funktionalität hatte schon mehr Umfang. Doch die Zeitprofile schienen nur zu greifen, wenn die App zum Zeitpunkt der Umschaltung läuft. Ob dies Zufall oder die tatsächliche Funktionalität war, ist unklar. Auch die Verbindungen der Telekom-Smartphone-App sind verschlüsselt.

In der Standardkonfiguration sieht das System ein Netzwerk vor, in dem das Gateway seine IP automatisch zugewiesen bekommt. Der Hersteller hat allerdings auch vorgesehen, dass eine feste IP verwendet werden kann. Zu diesem Zweck gilt es den Computer und das Gateway direkt über ein Netzwerkkabel, aber ohne Router und ohne Switch, zu verbinden. Sind Computer und Home Base auf diese Art verbunden, spielt die Basisstation DHCP- und DNS-Server. Sie hat die IP 192.168.5.5 und der Computer bekommt 192.168.5.10. Jede DNS-Anfrage wird mit der IP 192.168.5.5 beantwortet, im Test war dies für fritz.box, google.de, www.google.de, www.google.com und www.qivicon.com der Fall. Die eigentliche Konfiguration findet über die Webseite `http://www.qivicon.com/cgi-bin/luci` statt, auf die automatisch umgeleitet wird. Die Kommunikation ist unverschlüsselt.

Auf dem Websystem ist es nicht nur möglich, eine IP-Adresse fest zu vergeben, sondern auch die vermutlich komplette Konfiguration von der Home Base zu speichern und wieder einzuspielen. Die Daten scheinen den Entwicklern so wichtig zu sein, dass für die Datei ein Passwort mit Mindestqualität an Sicherheit – Buchstaben, Zahlen und Sonderzeichen, sowie mindestens sechs Zeichen – vergeben werden muss. Vermutlich wird die Datei mit dem Passwort verschlüsselt.

Dies ist auch für einen Angreifer interessant, könnte er doch so geheime Daten herunterladen oder die Einstellungen der Box ändern. Doch zur Nutzung der Webseite ist ein auf der Box der Home Base aufgedruckte Kennwort anzugeben. Dieses kann später noch verändert werden. Dieses könnte, da die Webseite nicht verschlüsselt ist, bei der Eingabe abgefangen werden. Allerdings ist der Computer physisch direkt mit der Home Base verbunden, ein Mithören auf dem Kabel ist schwer, ein Injizieren von Netzwerkkommunikation kaum möglich. In einer normalen Netzwerkkonfiguration konnte die Webseite nicht aufgerufen werden. Tiefer gehende Versuche wurden nicht unternommen.

Eine Ausnutzung ist mit einer Kombination von Schadsoftware und Social Engineering aber dennoch theoretisch denkbar. Das Opfer wird dazu gebracht, den Computer und die Basisstation zu verbinden sowie sich auf der Webseite anzumelden.

Die Untersuchungen mit nmap ergaben Linux 2.6.X – bei den Details 2.6.32-2.6.35 – als Betriebssystem des Gateways. Linux klingt plausibel, auch da im herunterladbaren Handbuch u.a.

GPLv2 erwähnt wird. In dieser, `http://www.gnu.org/licenses/gpl-2.0.html`, gibt es den Textabschnitt »For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights. «. Im Handbuch gibt es zwar Auflistungen der Lizenzen, doch es gibt nur Verweise auf die QIVICON-Support-Webseite `http://www.qivicon.de/support`. Der Lizenztext und der Quelltext konnten aber, anders als versprochen, nicht auf der verwiesenen Webseite gefunden werden. Auch eine als normaler Benutzer gestellte Supportanfrage diesbezüglich blieb auch nach über einem Monat unbeantwortet und der Zustand unverändert. Eine Überprüfung der Ergebnisse war so nicht möglich.

Ferner brachte nmap anfangs sieben und am Ende zehn offene Ports zutage. Diese sind 53, 80 (http), 139 (netbios-ssn), 443 (ssl/http), 445 (netbios-ssn), 7547, 8081, 8444 (ssl), 40044 und 49152. Nicht allen konnten Dienste zugeordnet werden. Jeder offene Port stellt ein potentielles Sicherheitsrisiko dar. Doch die bloße Existenz der Ports ist nicht ausreichend. Abseits der Automatisierung von Armitage wurde nicht versucht, potentielle Sicherheitslücken bei den Diensten hinter den Ports zu finden.

Zusammenfassend kann gesagt werden, dass das QIVICON System basierend auf den Annahmen der Untersuchungen sicher ist. Alle Verbindungen sind verschlüsselt, mit der Ausnahme eines extrem speziellen Falles. Die genutzte Version von SSL und die gewählten Verschlüsselungsmethoden könnten aber noch einmal überdacht werden. Auch die vielen offenen Ports geben Grund für Bedenken. Es bleibt aber, dass während dieser Untersuchungen nichts Praktikables für einen Angriff gefunden werden konnte.

4.2.7 Xavax, MAX!

Die bei den Untersuchungen verwendete Software hatte die Version 1.4.0. Das Gateway benutzte die in der Software hinterlegte Version.

Die Software ist an sich ein Java-basierter Webserver und die Oberfläche ist eine Webseite. Bevor allerdings die Einrichtung des Systems beginnen kann, muss die Firmware des Gateways aktualisiert werden. Zwar muss der Benutzer dies anstoßen, doch das Update ist die einzige Möglichkeit, die der Benutzer hat. Aus diesem Grund fanden die ganzen Tests mit der in der Software hinterlegten Firmware-Version statt. Die Versionsnummer ist nicht bekannt, aufgrund verschiedener Hinweise wird allerdings 1.19 vermutet – vorher war dann 1.0 installiert. Bei der Suche antwortete das Gateway. Vor und nach der Aktualisierung hat sich nur das letzte Byte der Antwort verändert. Die letzten zwei Bytes waren hexadezimal `01 00` und danach `01 13`.

Bei dem Update selber wird die Firmware per UDP in vielen kleinen Paketen über Port 23272 übertragen. Nach dem Update konnten die eigentlichen Tests beginnen.

Damit die Software und das Gateway zusammenfinden, sendet die Software mehrfach UDP-Broadcast-Pakete an jeden im Netzwerk an Port 23272. Das Gateway antwortet ebenfalls per UDP auf demselben Port. Natürlich könnten auch Angreifer, die es in das lokale Netzwerk

schaffen, diese Methode verwenden, um auf einen MAX! Cube, wie das Gateway heißt, zu testen.

Die UDP-Kommunikation ist noch umfangreicher als es bisher den Anschein erweckt. Die fragende Broadcast-Nachricht enthält die 19 Byte `eQ3Max*\0**********I`, wobei `\0` das Byte mit dem Wert 0 ist. Die Antwort vom Gateway ist dann `eQ3MaxApIHA002[...]>I...`, wobei anstelle des `[...]` eigentlich die letzten vier Stellen der Seriennummer stehen. Weitere Anfragen haben dann den Aufbau `eQ3Max*\0IHA002[...]` gefolgt von einem Buchstaben und evtl. weiteren Werten. Die Antwort beginnt mit `eQ3Max..IHA002[...]` gefolgt von `>`, dem Buchstaben der Anfrage und weiteren Werten. Die .. werden durch `Ap` oder `Bl` ersetzt. Das `Ap` nach `eQ3Max` steht wohl für Application, alternativ ist `Bl` für Bootloader.

Die jeweils einzelnen Buchstaben scheinen anzugeben, worum es bei der Nachricht geht. So scheint `I` nach Informationen zu fragen, `B` für einen Neustart zu sorgen, `U` ein Update einzuleiten und `W` die einzelnen Updatepakete einzuleiten. Zusätzlich wurde noch `N` gesehen.

```
1 /*  3:   */ public enum CubeUdpOpcode
2 /*  4:   */ {
3 /*  5:10 */   Identify('I'), GetNetworkAddress('N'), EnterBootloader('B'), InitUpdate('U↩
   '), WriteFrame('W'), Reboot('R'), GetNetworkConfiguration('c'), ↩
   SetNetworkConfiguration('C'), GetUrl('h'), SetUrl('H');
4 /*  6:   */
5 /*  7:   */   private char opcodeChar;
```

Listing 13: Ausschnitt der Rekonstruktion von CubeUdpOpcode.java

In der Datei `MaxLocalBackend-1.4.0.jar/de.eq3.max.al.local.update.CubeUdpOpcode.class` werden die verschiedenen Buchstaben aufgelistet. Beispielsweise steht `I` für Identify, `N` für GetNetworkAddress und `W` für WriteFrame. Zusätzlich zu den gesehenen Buchstaben soll es noch `R`, `c`, `C`, `h` und `H` geben.

Ist das Gateway erst einmal gefunden, so wird eine TCP-Verbindung über den Port 62910 aufgebaut. Über diese Verbindung wird dann in Textform kommuniziert. Eine Nachricht besteht dabei aus einem Buchstaben, der den Befehl oder die Art der Antwort angibt, gefolgt von einem Doppelpunkt, eventuellen Parametern und am Ende einem Windows-Zeilenumbruch. Die Parameter können je nach Befehl auch aus beliebigen Binärdaten bestehen. Diese werden dann aber mittels Base64 in normale, druckbare Zeichen umgewandelt und so in Textform übertragen.

Wie bei UDP gibt es auch für TCP eine Datei, `MaxEssentialsBackend-1.4.0.jar/de.eq3.max.al.core.IOperationCodes.class`, in der die verschiedenen Buchstaben definiert sind. Doch meist findet in den benutzenden Klassen erneut eine Definierung statt, so dass die Vollständigkeit dieser Liste in Frage steht. Der Quellcode – genauer ein Interface – lässt vermuten, dass es eine Administrations-Version der Software gibt, bei der beispielsweise auch das Geheimnis für die Verschlüsselung ausgelesen und verändert werden kann. Ein entsprechender Buchstabe konnte in der Liste nicht entdeckt werden.

```
3  public abstract interface IOperationCodes
4  {
5    public static final char DEVICE_LIST = 'L';
6    public static final char CONFIGURATION = 'C';
7    public static final char METADATA = 'M';
8    public static final char SET_CREDENTIALS = 'B';
9    public static final char GET_CREDENTIALS = 'G';
10   public static final char SET_REMOTEACCESS = 'J';
11   public static final char SET_USER_DATA = 'P';
12   public static final char GET_USER_DATA = 'O';
13   public static final char CHECK_PRODUCT_ACTIVATION = 'V';
14   public static final char ACTIVATE_PRODUCT = 'W';
15   public static final String INCOMING_HELLO = "H:";
16   public static final String INCOMING_NTP_SERVER = "F:";
17   public static final String INCOMING_DEVICE_LIST = "L:";
18   public static final String INCOMING_CONFIGURATION = "C:";
19   public static final String INCOMING_METADATA = "M:";
20   public static final String INCOMING_NEW_DEVICE = "N:";
21   public static final String INCOMING_ACKNOWLEDGE = "A:";
22   public static final String INCOMING_ENCRYPTION = "E:";
23   public static final String INCOMING_DECRYPTION = "D:";
24   public static final String INCOMING_SET_CREDENTIALS = "b:";
25   public static final String INCOMING_GET_CREDENTIALS = "g:";
26   public static final String INCOMING_SET_REMOTEACCESS = "j:";
27   public static final String INCOMING_SET_USER_DATA = "p:";
28   public static final String INCOMING_GET_USER_DATA = "o:";
29   public static final String INCOMING_CHECK_PRODUCT_ACTIVATION = "v:";
30   public static final String INCOMING_ACTIVATE_PRODUCT = "w:";
31   public static final String INCOMING_SEND_DEVICE_CMD = "S:";
32   public static final String OUTGOING_URL = "u:";
33   public static final String OUTGOING_INTERVAL = "i:";
34   public static final String OUTGOING_SEND = "s:";
35   public static final String OUTGOING_METADATA = "m:";
36   public static final String OUTGOING_INCLUSION_MODE = "n:";
37   public static final String OUTGOING_CANCEL_INCLUSION_MODE = "x:";
38   public static final String OUTGOING_MORE_DATA = "g:";
39   public static final String OUTGOING_QUIT = "q:";
40   public static final String OUTGOING_ENCRYPTION = "e:";
41   public static final String OUTGOING_DECRYPTION = "d:";
42   public static final String OUTGOING_SET_CREDENTIALS = "B:";
43   public static final String OUTGOING_GET_CREDENTIALS = "G:";
44   public static final String OUTGOING_SET_REMOTEACCESS = "J:";
45   public static final String OUTGOING_SET_USER_DATA = "P:";
46   public static final String OUTGOING_GET_USER_DATA = "O:";
47   public static final String OUTGOING_CHECK_PRODUCT_ACTIVATION = "V:";
48   public static final String OUTGOING_ACTIVATE_PRODUCT = "W:";
49   public static final String OUTGOING_SEND_DEVICE_CMD = "s:";
50   public static final String OUTGOING_RESET = "a:";
51   public static final String OUTGOING_RESET_ERROR = "r:";
52   public static final String OUTGOING_DELETE_DEVICES = "t:";
53   public static final String OUTGOING_SET_PUSHBUTTON_CONFIG = "w:";
54   public static final String OUTGOING_GET_DEVICE_LIST = "l:";
55   public static final String OUTGOING_SET_URL = "u:";
56   public static final String OUTGOING_GET_CONFIGURATION = "c:";
57   public static final String OUTGOING_TIME_CONFIG = "v:";
58   public static final String OUTGOING_NTP_SERVER = "f:";
59   public static final String OUTGOING_SEND_WAKEUP = "z:";
60 }
```

Listing 14: Ausschnitt der Rekonstruktion von IOperationCodes.java

Aufgrund des textbasierten Protokolls ist es möglich, netcat für die Kommunikation zu verwenden. Ein Verbindungsaufbau war erfolgreich. Auch die Steuerung der Heizung gelang ohne

Probleme. Es ist keinerlei Authentifizierung notwendig. Ist der Angreifer erst einmal ins Netzwerk gelangt, so kann er das System verändern wie er möchte.

Zur Steuerung von Geräten oder Gerätegruppen ist eine ID notwendig, die die Geräte eindeutig bestimmt. Sowohl die lokal auf einem Computer installierte MAX! Software, als auch Angreifer müssen IDs ermitteln, um sie verwenden zu können. Diese Arbeit nimmt das Gateway selber ab. Sobald eine Verbindung mit dem MAX! Cube aufgenommen wurde, schickt dieses ungefragt so einige Daten, unter anderem die IDs der Geräte und deren Konfiguration. Ein Zutun des Angreifers oder der Software ist nicht notwendig. Ein potentieller Angreifer braucht somit nur Zugang zum lokalen Netzwerk und hat dann direkt alle Informationen, die er zur Steuerung des Smart-Home-Systems benötigt.

```
1 Anfrage: s:AARAAAAAA***AWw=\r\n
2 Antwort: S:01,0,31\r\n
```

Listing 15: Heizung steuern; einige Byte wurden zur Unkenntlichmachung durch * ersetzt

Die in Listing 15 angegebene Anfragezeile stellt einen Befehl dar, der zum Cube gesendet wurde. Er beginnt mit `s:` und danach kommt ein Binärfeld in Base64, welches die eigentlichen Daten enthält. Die elf Byte Binärdaten sind `00 04 40 00 00 00 03 ** ** 01 7B`. Tabelle 8 stellt den Aufbau des Befehls dar. Wenn es sich um eine Gruppe handelt, dann ist egal,

00	unbekannt
04	Bit 7-3 → unbekannt Bit 2 → Gruppe Bit 1 → nicht Gruppe Bit 0 → unbekannt
40	Befehl, hier 64 → `ControlMode`
00	unbekannt
00	
00	
03	
**	Geräte-Funk-Adresse
**	
01	Raum-ID
6C	Bit 7-6 → Mode; hier 01 → Permanent Bit 5-0 → Temperatur; hier 2C → 22,0 °C

Tabelle 8: Aufbau Binärdaten des Geräte-Befehls, wobei die beiden **-Bytes die eigentlichen Bytes ersetzen

welcher Wert bei der Geräte-Funk-Adresse steht. Je nachdem, was für ein Befehl es ist, werden verschiedene Angaben gesetzt, oder eben nicht. Bei den unbekannten Feldern ist nicht klar, ob diese bei anderen Befehlen als dem Setzen der Temperatur mit dem „ControlMode“ einen Wert haben.

Die Antwort besteht aus drei Teilen, durch Komma getrennt. Der Datei `MaxLocalBackend-1.4.0.jar/de.eq3.max.il.local.commands.DeviceSendCommand.class` kann entnommen werden, was die Bedeutung der Werte ist. Der erste Wert ist eine Hex-Zahl, der „DutyCycle“, es folgt, ob der Befehl verworfen wurde, und den Abschluss bildet die Angabe der freien Speicherslots, ebenfalls hexadezimal. Der Befehl wurde also angenommen und entsprechend umgesetzt.

```
POST /login HTTP/1.1
Host: max.eq-3.de
Connection: keep-alive
Referer: http://max.eq-3.de/login.jsp
Content-Length: 64
Cache-Control: max-age=0
Origin: http://max.eq-3.de
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (Linux; U; Android 4.0.4; de-de; SonyEricsson[...] Build/4.1.B.0.431) AppleWebKit/534.30 (KHTML, like Gecko) Version/4.0 Mobile Safari/534.30
Accept-Encoding: gzip,deflate
Accept-Language: de-DE, en-US
Accept-Charset: utf-8, iso-8859-1, utf-16, *;q=0.7
x-wap-profile: http://wap.sonyericsson.com/UAprof/[...].xml
Cookie: JSESSIONID=[...]

user=[...]&passwd=[...]&submit=&mobile=false&productKey=
```

Listing 16: Login im Webportal, hier per Smartphone; bei den Stellen mit `[...]` sind die genauen Werte entfernt

Neben dem lokalen Zugriff gibt es auch ein Webportal, welches optional verwendet werden kann. Dazu gilt es in der Software einen Benutzernamen und ein Passwort zu vergeben. Die Software kommuniziert dann mit dem Cube und dem Server. Am Ende baut das Gateway selbst eine Verbindung zum Server auf. Die Übertragung der Daten geschieht zwar mittels HTTP und damit im Prinzip unverschlüsselt, die eigentlichen, übertragenen Daten sind allerdings verschlüsselt. Mehrere Indizien – wie die Nennung von AES128 im Pseudo-Quelltext, dass in jeder Meldung vom Cube die ersten 16 Byte immer gleich sind und jeder Block durch 16 teilbar ist – sprechen für AES128 als Verfahren; doch dies wurde nicht näher untersucht. Die Software ver- und entschlüsselt die Daten nicht selbst. Sie sendet Befehle an das Gateway, welches dann die Arbeit übernimmt. Die dafür verwendeten Buchstaben sind `d`/`D` bzw. `e`/`E`. Es wurde nicht untersucht, ob alle Cubes eines Anbieters denselben Schlüssel haben oder nicht. Sollten alle den gleichen nutzen, so bräuchte ein Angreifer nur einen Cube und könnte jede Kommunikation mitlesen.

Das große Problem stellt allerdings die andere Seite der Verbindung dar. Um das Portal von unterwegs nutzen zu können, muss sich der Anwender auf dem Webportal einloggen und kann dann dort seine Änderungen vornehmen. Dazu wird ein normaler Webbrowser verwendet. Auch hier geschieht die Kommunikation per HTTP und diesmal ist kein Cube vorhanden, der den eigentlichen Inhalt verschlüsseln könnte. Dies läuft also komplett unverschlüsselt ab. Ein Angreifer kann die Login-Daten abgreifen und später verwenden. Listing 16 stellt den mitgeschnittenen Login dar. Aber auch die Sitzungs-IDs können gelesen werden, um so die aktuelle Sitzung des Benutzers gleich mitzunutzen. Der Mitschnitt in Listing 17 ergab sich beispielsweise beim Setzen der Temperatur. Die Mitschnitte sind bei den immer verbreiteteren öffentlichen WLAN-Hot-Spots besonders einfach, muss dort doch nur gelauscht werden.

Die Basis der vorhandenen lokalen Software ist Java. Die Programme dieser Sprache liegen als Bytecode vor, aus dem der Quelltext wieder gewonnen werden kann. Im Großen und Ganzen fehlen dann nur die Kommentare der originalen Quelltexte. Einige Ausschnitte wurden in diesem Kapitel zwecks Demonstration bereits geliefert. Eine Verschleierung des Quellcodes

mittels Obfuscation stellt für Experten kaum ein Problem dar und wurde bei der vorliegenden Software auch nicht durchgeführt. Das Verbot des Dekompilierens durch die der Software beiliegenden Lizenzbedingungen wird potentielle Angreifer sicher nicht davon abhalten.

Nach Gewinnung des Quelltextes könnte ein Angreifer diesen nutzen, um für seinen Angriff alle notwendigen Befehle in Erfahrung zu bringen.

```
POST /dwr/call/plaincall/MaxRemoteApi.setClientCommands.dwr HTTP/1.1
Host: max.eq-3.de
Connection: keep-alive
Referer: http://max.eq-3.de/index.html
Content-Length: 355
Origin: http://max.eq-3.de
Content-Type: text/plain
User-Agent: Mozilla/5.0 (Linux; U; Android 4.0.4; de-de; SonyEricsson[...] Build/4.1.B↩
 .0.431) AppleWebKit/534.30 (KHTML, like Gecko) Version/4.0 Mobile Safari/534.30
Accept-Encoding: gzip,deflate
Accept-Language: de-DE, en-US
Accept-Charset: utf-8, iso-8859-1, utf-16, *;q=0.7
x-wap-profile: http://wap.sonyericsson.com/UAprof/[...].xml
Cookie: JSESSIONID=[...]

callCount=1
page=/index.html
httpSessionId=[...]
scriptSessionId=[...]
c0-scriptName=MaxRemoteApi
c0-methodName=setClientCommands
c0-id=0
c0-e2=number:1
c0-e3=number:16.5
c0-e1=Object_MaxSetRoomPermanentMode:{roomId:reference:c0-e2, temperature:reference:c0-e3}
c0-param0=Array:[reference:c0-e1]
batchId=7
```

Listing 17: Aufruf zum Setzen der Temperatur; bei den Stellen mit `[...]` sind die genauen Werte entfernt

Der Zugriff auf das Gateway ist exklusiv, ein zeitgleicher Zugriff lokal und per Webportal ist nicht möglich. Ein Angreifer könnte so theoretisch lokal eine Verbindung aufbauen und jeglichen Zugriff blockieren. Das Gateway meldet sich selber nur ca. alle fünf Minuten beim Server. Es dauert deshalb einige Zeit, bis eine Fehlermeldung im Webportal erscheint, und ein Benutzer sieht so anfangs nicht einmal, dass er keinen Zugriff mehr hat.

Zusammenfassend kann gesagt werden, dass sich bei dem System wohl durchaus Gedanken um die Sicherheit beim Zugriff über das Internet gemacht wurde. Allerdings hat man nicht den kompletten Weg mit einer Verschlüsselung bedacht, so dass die Bemühungen letztendlich vergebens waren. Bei der Nutzung im lokalen Netzwerk verlässt sich die Lösung komplett auf die durch den Benutzer geschaffene Sicherheit. Es gibt offenbar keine weiteren Schutzvorrichtungen.

4.3 Resultierende Gefahren

Die untersuchten Systeme offenbaren ungewollt eine Vielfalt an Angriffsmöglichkeiten. Die vielen verschiedenen Lösungen, die alle irgendwie anders arbeiten, erschweren zumindest Angreifern das Fertigen von Multitools für den Angriff. Smart Home ist bisher noch kaum verbreitet, was sich allerdings langsam ändert. Momentan lohnt es sich für Angreifer noch nicht, die Systeme zu unterwandern. Gezielte Angriffe sind aber trotzdem denkbar.

Keines der getesteten Systeme konnte direkt, ohne Zutun des Benutzers, über das Internet gesteuert werden. Doch eine Entwarnung kann dennoch nicht gegeben werden. Bei einigen Systemen besteht beispielsweise die Gefahr, dass bei der Benutzung der Geräte über das Internet Login-Daten unbemerkt abgefangen werden. Angreifer sind danach in der Lage, die Systeme über das Internet zu steuern.

Dieses und weitere Szenarien werden in allgemeiner Form im Folgenden angerissen.

Die mehrfach vorgekommenen exklusiven Zugriffe ermöglichen es Angreifern auf leichte Weise ein System lahmzulegen. Es wird eine TCP-Verbindung zu einem Port aufgebaut und gehalten und schon können die Besitzer die „Komfortfunktionen“, für die sie die Geräte gekauft haben, nicht mehr benutzen. Hier ist ein herstellerübergreifender Angriff gut denkbar. Eine mögliche Schadsoftware muss nur eine Liste von Ports haben, die sie durchprobiert.

Schwerer umzusetzen ist das Befallen des Gateways, zumeist indem die komplette Firmware ersetzt wird. Dies muss zwangsweise gerätespezifisch geschehen. Bei der momentanen Vielfalt und geringen Verbreitung ist das eher unwahrscheinlich.

Ein lästiges Ärgernis wäre Schadsoftware, die Systeme im lokalen Netzwerk steuert, beispielsweise die Heizung in regelmäßigen Intervallen herunterregelt und so die Wohnung auskühlen lässt oder für Frostschäden an Leitungen sorgt. Theoretisch könnten sie auch einen Steckdosenschalter einschalten, an dem eine eingeschaltete Herdplatte hängt und ein Topf mit Essen steht und dieser verkohlt und im schlimmsten Fall einen Brand auslöst. Hier müssten allerdings schon mehrere ungünstige Umstände zusammen kommen.

Das Abgreifen von Login-Daten für Fernzugriffe stellt eine weitere Gefahr dar, besonders in öffentlichen WLANs. Hat der Angreifer die Daten erst einmal erlangt, kann er sie nutzen um sich selber jederzeit als berechtigte Person auszugeben und „legal“ auf das System zuzugreifen.

Ein weiteres Potential haben Geräte, die undokumentierte Dienste beinhalten. Da die Benutzer die Dienste nicht kennen, können sie diese auch nicht absichern. Hat ein Angreifer sich die Kontrolle darüber verschafft, stehen die Besitzer vor einem Problem. Sie sehen zwar die Ergebnisse der Übernahme, wie etwa eine kalte Wohnung, können sich aber den Grund nicht erklären, die Ursache nicht ausmachen und somit nicht beheben.

Durch das Mitlesen von Daten können, genau wie bei der Übermittlung von Daten an die Hersteller, Profile über die Personen im Haushalt erstellt werden. Angreifer können diese Daten nutzen, um Zeiten zu ermitteln, in denen niemand im Haus ist und teils auch das Haus zu lokalisieren. Die Profile der Anbieter wiederum sind für die Werbeindustrie äußerst interessant und könnten für den Anbieter einen monetären Vorteil bieten. Bei Smart-TVs konnte beispielsweise schon einen Datenfluss nach Google beobachten werden [Eikenberg, 2014].

5 Informierung der Hersteller

Die Meldung sicherheitsrelevanter Ergebnisse der Untersuchung an die Hersteller, sowie deren Reaktion wird hier festgehalten.

5.1 Zeitleiste

In Tabelle 9 wird die zeitliche Abfolge der Meldung der Sicherheits-Evaluation an die Hersteller und deren Reaktion darauf festgehalten.

Datum	Bemerkung
2014-02-03	Hersteller wurden per E-Mail und PDF-Datei in Kenntnis gesetzt
2014-02-05	EUROiSTYLE antwortet
2014-04-25	bisher gab es noch keine weiteren Antworten

Tabelle 9: zeitliche Abfolge der Informationsweitergabe an die Hersteller

5.2 Antworten der Hersteller

5.2.1 EUROiSTYLE, tapHOME - 2014-02-05

Es folgt eine gekürzte Version der Antwort von EUROiSTYLE:

> [...]Folgende Änderungen haben wir bereits in der Umsetzung[...]:
>
> - Änderung des Gateway-Passworts. [...]
> - Verzicht auf UPnP [...]
>
> An der grundsätzlichen Kommunikation mit dem Gateway werden wir vorerst festhalten müssen[...]. Auf die Möglichkeit, das Gateway via Webinterface zu bedienen und mehr Funktionen zu verwenden, haben wir bewusst in der Dokumentation verzichtet. [...]
>
> Langfristig denken wir darüber nach, ein eigenes Gateway zu entwickeln, [...]

6 Einschätzung

Die Smart-Home-Starter-Kits wurden sowohl automatisiert als auch manuell untersucht. Die automatisierten Tests mit Nmap und Metasploit führten zu keinen direkten Erfolgen. Es wurden zwar in mehreren Fällen offene Ports und möglicherweise ausnutzbare Schwachstellen gemeldet, eine tatsächliche Ausnutzung war aber nicht möglich. Dieser Umstand könnte darauf hindeuten, dass die Hersteller sich zumindest rudimentär mit Sicherheit beschäftigt haben und Ihre Systeme gegen automatisierte Angriffe geprüft und gesichert haben. Wahrscheinlicher aber ist, dass es sich lediglich um Zufall handelt und die Systeme nicht vorsätzlich gesichert wurden.

Diese Annahme wird dadurch gestützt, dass die manuellen Untersuchungen in nahezu allen Fällen ausnutzbare Sicherheitslücken zu Tage gebracht haben. Wird dabei bedacht, dass die bisher durchgeführten Tests relativ oberflächlicher Natur waren, macht dies umso deutlicher, wie wenige Ressourcen in die Sicherheit der Systeme investiert wurden.

Zurzeit erscheint die daraus folgende Gefährdungslage noch harmlos. Zum einen sind die Geräte in keinen nennenswerten Stückzahlen verbreitet und zum anderen existiert kein lohnenswertes Geschäftsmodell für Kriminelle, mit welchem sie schnell und einfach finanziellen Vorteil aus einem Angriff ziehen könnten. Da die Angriffe aber extrem einfach durchführbar sind, ist vorstellbar, dass sie als weiteres Modul in bestehende Schadsoftware integriert werden. AV-TEST sammelt täglich mehrere hunderttausend neue Varianten von Schadsoftware ein. Es ist also leicht vorstellbar, dass in einigen dieser Varianten auch Module zum Angriff auf Smart-Home-Anlagen integriert werden. Da sich Schadsoftware ohnehin oft aus einem „Baukasten“ von Modulen bedient, wäre es lediglich nötig, einen Angriff auf ein bestimmtes Smart-Home-Gerät einmal zu programmieren. Dieser Programmcode könnte dann einfach in verschiedensten Varianten von Schadsoftware eingesetzt werden. Sobald die Verbreitung der Smart-Home-Geräte ansteigt und ein Geschäftsmodell für Angreifer existiert, kann von einem Tag auf den anderen ein massenhafter Angriff erfolgen.

Schon jetzt sind lohnende Angriffsszenarien denkbar:
Smart-Home-Systeme, die die Überwachung des Wohnraums – insbesondere das berechtigte und unberechtigte Öffnen von Türen und Fenstern – leisten, können ein Angriffsziel für Einbrecher darstellen. Zum einen kann durch Zugriff auf solche Systeme herausgefunden werden, ob sich jemand im Wohnraum befindet oder ob gefahrlos eingebrochen werden kann. Zum anderen können die Systeme manipuliert werden, so dass sie dem Nutzer keinen Alarm senden.

Systeme, welche Steckdosen, Schalter und Heizungsthermostate steuern, bieten einen großen Komfortgewinn für den Nutzer. Diesen Komfort könnte eine Schadsoftware problemlos stören, indem sie die Anlage unbenutzbar macht oder die Aktionen des Nutzers verändert. Zum Beispiel könnten Thermostate herunter gedreht werden, so dass die Wohnung auskühlt oder Frostschäden entstehen. Es ist auch denkbar, dass Schaltvorgänge an Lichtschaltern und Elektrogeräten verhindert werden und so angeschlossene elektronische Geräte und Lampen nicht mehr benutzt werden können. Das Haus wird gewissermaßen als Geisel genommen und betroffene Nutzer müssen es gegen ein Lösegeld an den Angreifer freikaufen, ähnlich wie die sogenannten BKA/GEMA/GVU-Trojaner heute den PC als Geisel nehmen.

Die Schwachstellen und möglichen Angriffsszenarien, die hier beispielhaft an Smart-Home-Systemen nachgewiesen wurden, sind auch in anderen Geräteklassen zu erwarten. Sie kön-

nen dort ebenfalls Schaden anrichten, der sogar über das Potential der Smart-Home-Systeme hinausgeht. An dieser Stelle ist der Begriff des „Internets der Dinge“ zu nennen. Nahezu jedes elektronische Gerät hat das Potential einen Mehrwert zu liefern, wenn es an das Internet angeschlossen ist. Viele Geräteklassen sind es heute schon und in Zukunft werden es noch mehr:

- Hi-Fi und TV
- Wetterstationen
- Haushaltsgeräte
- Smart Meter
- Smart-Home-Systeme
- Heizungen
- Überwachungskameras

Selbst wenn die bisher genannten Schwachstellen in Smart-Home-Systemen an der einen oder anderen Stelle nur eine theoretische Gefahr darstellen, sind sie ein Beispiel dafür, was möglich ist und woran Hersteller von Geräten, die mit dem Internet oder einem lokalem Netzwerk verbunden sind, rechnen müssen.

Die Schwierigkeit hier ist, dass es viele verschiedene Geräte von unterschiedlichen Herstellern gibt. Alle diese Hersteller müssen sich um Sicherheit kümmern. Es ist aber leider unrealistisch zu erwarten, dass dies geschehen wird, außer es wird entsprechende gesetzliche Vorschriften oder verbindliche Sicherheitsstandards geben.

Wie die beispielhaften Untersuchungen gezeigt haben, ignorieren manche Hersteller selbst grundlegende Regeln einer sicheren Implementierung. So werden Netzwerkverbindungen nicht verschlüsselt, sondern im Klartext übertragen, was es zum Beispiel ermöglicht, Nutzernamen und Passwörter mitzulesen. Für Verbindungen ins Internet ist dies besonders kritisch. Als Steigerung wird von manchen Systemen selbst auf eine Anmeldung durch Nutzernamen und Passwort verzichtet, so dass jeder, der Zugriff auf das lokale Netzwerk hat, die Geräte beliebig abfragen und steuern kann.

Einzelne Botnetze infizieren hunderttausende PCs und haben damit Zugriff auf das Netzwerk, in dem sich der PC befindet. Derart schlecht geschützte Smart-Home-Systeme – oder Geräte anderer Geräteklassen – sind dann eine leichte Beute für die Schadsoftware. Erschwerend kommt hinzu, dass Sicherheitslücken in eingebetteten Geräten nur schwer zu schließen sind. Bei nicht wenigen Geräten ist eine nachträgliche Erhöhung der Sicherheit aufgrund der begrenzten Ressourcen nicht praktikabel. So ist beispielsweise die Rechengeschwindigkeit vieler Geräte zu gering, um nachträglich eine bessere oder gar asymmetrische Verschlüsselung einzuführen, wie sie etwa bei SSL vorgesehen ist. Für gefundene Lücken gilt es Sicherheitsupdates zu entwickeln und einzuspielen. Auf normalen Computern und selbst Smartphones sind regelmäßige Updates an der Tagesordnung und vom Nutzer akzeptiert. Es ist allerdings fraglich, ob ein Nutzer es auch akzeptieren würde, regelmäßig seine Heizung, sein

Festnetztelefon, seinen Kühlschrank, seinen Fernseher, sein Smart-Home-System usw. mit Sicherheits-Updates manuell zu versorgen. Ein komfortabler und vor allem automatischer, aber auch sicherer Updateprozess wäre notwendig. Leider ist es wahrscheinlich, dass es für viele Geräte nie Updates geben wird. Falls doch, werden sie automatisch oder sogar nur manuell einzuspielen sein. Allerdings steht fest, dass jegliche Art des Updates einen neuen Angriffspunkt darstellen kann.

Es wird demnach auch eine andere Frage aufgeworfen: Wenn nicht zu erwarten ist, dass alle Geräte zumindest ein Mindestmaß an Sicherheit – wie Authentifizierung mit Nutzernamen und Passwort, sowie verschlüsselte Kommunikation – bereitstellen, wie können diese Geräte und das Netzwerk dann geschützt werden? Es können nicht auf jedem einzelnen Gerät Schutzlösungen installieren werden. Dies wäre weder vom Durchschnittsnutzer handhabbar, noch bieten diese Geräte genügend Ressourcen. Es sind also neue Konzepte notwendig, die die bereits erwähnte sichere Implementierung von Anmeldung, Kommunikation und Update ergänzen. Denn selbst wenn das Gerät grundlegend sicher implementiert wird, ist dies keine Garantie, dass nicht doch ausnutzbare Schwachstellen auftreten. Es ist also denkbar, dass Sicherheitssoftware nicht mehr nur den Client, wie PC oder Smartphone, sondern das gesamte Netzwerk bzw. allgemeiner die Verbindung vom Internet zum Netzwerk schützt.

Schon jetzt gibt es Meldungen, dass eingebettete Geräte, wie Ampelanlagen und Radarfallen, aber auch Industriesteuerungen, über das Internet infiziert und von Kriminellen in ihrer Funktionalität manipuliert wurden. Selbst Meldungen von infizierten Kühlschränken, die für Spamversand benutzt werden, fanden sich. Nicht alle diese Meldungen sind mit ausreichend Fakten belegt, aber sie zeigen was schon heute möglich und denkbar ist.

7 Ausblick

Schon diese einfachen Untersuchungen brachten bei einigen der untersuchten Geräten größere potentielle Sicherheitsprobleme zu Tage. Tiefer gehende Evaluationen sollen folgen. Nachdem die grundlegende IP-basierte Kommunikation zu den Gateways betrachtet wurde, ist der nächste Schritt die genauere Analyse der SSL-Verbindungen. Sind diese, wie hier vorerst angenommen wurde, wirklich sicher oder gibt es Anfälligkeiten für Man-in-the-Middle- oder gängige SSL-Angriffe? Da noch nicht zweifelsfrei feststeht, ob die RC4-Verschlüsselung wirklich gebrochen ist, geschweige denn wie dies durchgeführt wird, kann dieser spezielle Punkt leider nicht getestet werden.

Immer mehr Hersteller drängen mit ihren Lösungen zu Smart Home auf den Markt. Schon jetzt stellen die untersuchten Geräte nur einen sehr kleinen Teil der verfügbaren Produkte da. In Zukunft wird es Tests mit weiteren Geräten anderer Hersteller geben.

Für ein Produkt wurde schon die Möglichkeit des Ersetzens der Firmware erprobt. Für Angreifer ist dies äußerst interessant, da sie so das Gerät komplett und dauerhaft unter ihre Kontrolle bringen können. Bei komplexen Systemen wie QIVICON ist sogar Schadsoftware, die das Gerät befällt, theoretisch denkbar. Es ist daher sinnvoll, solche Betrachtungen zukünftig zu verstärken. Damit einher geht auch die Folge des Missbrauchs von Geräten für andere Zwecke, wie den Versand von Spam oder Bitcoin-Mining.

Die Verbindungen des Gateways gehen nicht nur ins lokale Netzwerk bzw. Internet. Die Geräte selber sind meist per Funk an das Gateway angeschlossen. Da diese stark lokal eingegrenzt sind und Angreifer sich somit in der Nähe der Geräte aufhalten müssten, ist diese Art des Angriffs unattraktiv. Doch vor allem für Sicherheitslösungen wie die von Gigaset ist eine möglichst unverwundbare Kommunikation wichtig. AV-TEST plant zukünftig auch dieses Feld abzudecken.

Literatur

[EUROiSTYLE GmbH , 2014a] EUROiSTYLE GmbH (2014a). Fahrplan der Entwicklung der nächsten Monate. Webseite, Stand: 2014-02-18. Link: http://www.taphome.eu/de/news-detail/items/Fahrplan.html.

[EUROiSTYLE GmbH , 2014b] EUROiSTYLE GmbH (2014b). Neue Android App, Version 2.0.146 verfügbar. Webseite, Stand: 2014-03-14. Link: http://www.taphome.eu/de/news-detail/items/neue-android-app-version-20146-verfuegbar.html.

[Böck, 2013] Böck, H. (2013). SSL/TLS Schwächen in RC4 ausnutzbar. Webseite, Stand: 2014-01-23. Link: http://www.golem.de/news/ssl-tls-schwaechen-in-rc4-ausnutzbar-1303-98164-2.html.

[Brucke, 2009] Brucke, M. (2009). Smart Home mit intelligenten Assistenten. Link: https://www.dke.de/de/Wirueberuns/MitteilungenderDKEGeschaeftstelle/Documents/Brucke1.pdf.

[Cisco Security Advisory , 2013] Cisco Security Advisory (2013). Portable SDK for UPnP Devices Contains Buffer Overflow Vulnerabilities. Webseite, Stand: 2014-01-28. Link: http://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20130129-upnp.

[Eikenberg, 2014] Eikenberg, R. (2014). Spion im Wohnzimmer: c't entdeckt Sicherheitslücken in zahlreichen Smart-TVs. Webseite, Stand: 2014-01-30. Link: http://www.heise.de/security/meldung/Spion-im-Wohnzimmer-c-t-entdeckt-Sicherheitsluecken-in-zahlreichen-Smart-TVs-2097287.html.

[eQ-3 AG] eQ-3 AG. RWE Smarthome. Webseite, Stand: 2014-01-30. Link: http://www.eq-3.de/rwe-smarthome.html.

[Goldman, 2013] Goldman, D. (2013). Shodan: The scariest search engine on the Internet. Webseite, Stand: 2014-01-21. Link: http://money.cnn.com/2013/04/08/technology/security/shodan/.

[Mobile Dev & Design, 2009] Mobile Dev & Design (2009). SiTel Processor Picked For DECT Phones. Webseite, Stand: 2014-02-12. Link: http://mobiledevdesign.com/news/sitel-processor-picked-dect-phones.