HOCHSCHULE
MITTWEIDA
UNIVERSITY OF
APPLIED SCIENCES

BACHELORARBEIT

Frau
Lena Neumann

Sicherheit von Netzwerkgeräten am Beispiel Cisco 800 ISR

Mittweida, 2019

Fakultät Angewandte Computer- und Biowissenschaften

BACHELORARBEIT

Sicherheit von Netzwerkgeräten am Beispiel Cisco 800 ISR

Autor:
Frau

Lena Neumann

Studiengang:
Allgemeine und Digitale Forensik

Seminargruppe:
FO15w3-B

Erstprüfer:
Prof. Dr. Dirk Labudde

Zweitprüfer:
B.Sc. Martin Klöden

Einreichung:
Mittweida, 07.Januar 2019

Verteidigung/Bewertung:
Mittweida, 2019

Faculty applied computer and life sciences

BACHELOR THESIS

Networking device security as example Cisco 800 ISR

author:
Ms.

Lena Neumann

course of studies:
general and digital forensics

seminar group:
FO15w3-B

first examiner:
Prof. Dr. Dirk Labudde

second examiner:
B.Sc. Martin Klöden

submission:
Mittweida, 07.Januar 2019

defence/ evaluation:
Mittweida, 2019

Bibliografische Beschreibung:

Neumann, Lena:
Sicherheit von Netzwerkgeräten am Beispiel Cisco 800 ISR. - 2019. - VI, 44, VI S.
Mittweida, Hochschule Mittweida, Fakultät Angewandte Computer- und Biowissenschaften, Bachelorarbeit, 2019

Referat:

Ziel der vorliegenden Arbeit ist es, Nutzer aus dem Heimbereich zur kritischen Beurteilung der Sicherheit von Netzwerkgeräten anzuregen. Dazu wird zunächst aufgezeigt, warum Netzwerkgeräte ein lohnenswertes Ziel darstellen können. Zusätzlich werden bekannte Beispiele für gehackte beziehungsweise veränderte Netzwerkgeräte vorgestellt. Außerdem werden dem Leser Methoden zur Untersuchung dargestellt, welche an den konkreten Beispielen exemplarisch durchgeführt werden. Um eine sinnvolle Reihenfolge der Untersuchungen zu gewährleisten wird ein allgemeines Vorgehensmodell vorgestellt, welches entsprechend an die konkreten Untersuchungsanforderungen angepasst werden kann.

Insbesondere werden die Recherche von Informationen, die optische Untersuchung, die Einstellungsüberprüfung und die Verifikation der Firmware besprochen und am konkreten Beispiel gezeigt. Auch die Übertragbarkeit auf andere Geräte wird überprüft und es wird versucht einen Ausblick auf eventuelle zukünftige Entwicklungen im Bereich der Sicherheit von Netzwerkgeräten zu geben. Limitierende Faktoren werden im Diskussionsteil der Arbeit ebenfalls betrachtet und näher erläutert.

Inhalt

Vorwort

Die Ihnen vorliegende Bachelorarbeit „Sicherheit von Netzwerkgeräten am Beispiel Cisco 800ISR“ stellt Methoden vor, um die Sicherheit von Netzwerkgeräten überprüfen und beurteilen zu können. Dafür benötigte Daten wurden zum Teil bereits während des Pflichtpraktikums erworben und später entsprechend aufbereitet und ergänzt. Diese Bachelorarbeit habe ich als Abschlussarbeit meines Studiums der Allgemeinen und Digitalen Forensik an der Hochschule Mittweida geschrieben. Von Oktober 2018 bis Januar 2019 beschäftigte ich mich mit den nötigen Untersuchungen und dem Verfassen der Bachelorarbeit.

Das Thema der Bachelorarbeit ergab sich durch die während der Praktikumszeit intensive Beschäftigung mit dem Thema der Sicherheit von Firmware. Die durchgeführten Untersuchungen variierten sehr stark in ihrer Komplexität. Sobald im Praktikum Probleme auftraten, konnte ich mich immer an die Kollegen wenden, welche mir nach bestem Wissen und Gewissen halfen. Bei Professor Labudde möchte ich mich für die hochschulseitige Betreuung des Praktikums bedanken. Außerdem half er dabei, alle administrativen Fragen zügig zu klären und stimmte dem endgültigen Thema der Bachelorarbeit zu. Während des Schreibprozesses und bei der Auswahl wichtiger Schwerpunkte hat mich mein Betreuer Herr Martin Klöden sehr unterstützt. Seine Anmerkungen haben dabei geholfen, die zum Teil komplizierten Sachverhältnisse nachvollziehbar zu erklären.

Auch meine Familie unterstützte mich beim Verfassen dieser Arbeit sehr. Sie zeigten großes Verständnis, wenn ich während des Schreibens um Ruhe im Haus bat und mich hin und wieder weigerte, den aktuellen Bearbeitungsstand der Arbeit im Detail zu erörtern. Ganz besonders möchte ich mich bei meiner Mutter bedanken, sie las die ersten Fassungen der Arbeit gegen und überprüfte sowohl die Verständlichkeit als auch die Rechtschreibung.

Beim Lesen der Arbeit wünsche ich Ihnen viel Vergnügen.

Lena Neumann

Zettlitz, 09. Dezember 2018

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

ASA ..... *Adaptive Security Appliance*

BSI ..... *Bundesamt für Sicherheit in der Informationstechnik*

bzw. ..... *beziehungsweise*, *beziehungsweise*

CLI ..... *Command Line Interface*

CPU ..... *Central Processing Unit*

DDoS ..... *Distributed Denial of Service*

FBI ..... *Federal Bureau of Investigation*

FCC ..... *Federal Communications Commission*

IP ..... *Internetprotokoll*

LAN ..... *Local Area Network*

Mips ..... *Microprocessor without Interlocked Pipeline Stages*

PCB ..... *Printed Circuit Board*

RAM ..... *Random Access Memory*

SHA ..... *Secure Hash Algorithm*

WLAN ..... *Wireless Local Area Network*

1 Einleitung

Mit diesem Kapitel soll dem Leser der Einstieg in das Thema der Bachelorarbeit ermöglicht werden. Dafür wird im ersten Unterkapitel zunächst die Motivation erläutert, bevor dann in 1.2 auf die aktuelle Situation und daraus resultierende Probleme eingegangen wird. Anschließend wird die Zielstellung der Arbeit definiert, bevor ein Überblick über den konkreten Aufbau der vorliegenden Arbeit gegeben wird.

1.1 Motivation

Ein Verbund von mehreren Rechnern und entsprechenden Netzwerkgeräten wird Netzwerk genannt. Dieses ermöglicht die gemeinsame Nutzung von Ressourcen und den schnellen Austausch von Informationen zwischen den angeschlossenen Geräten [1]. Um überhaupt ein solches Netzwerk aufbauen zu können, werden mindestens zwei Rechner und ein Netzwerkgerät benötigt. Alle Netzwerkgeräte haben die grundlegende Aufgabe, die Verbindung zwischen den angeschlossenen Rechnern herzustellen. Zusätzlich können sie weitere Funktionalitäten erfüllen, wie zum Beispiel eine Verbindung zwischen einem privaten Netzwerk und dem Internet herstellen [1]. Geräte, die diese Funktionalität erfüllen, werden Router genannt.

Da Geräte wie Router die Grundlage jedes Netzwerks bilden, sind sie auch für die Netzwerksicherheit von großer Bedeutung. Durch gehackte Geräte können Daten abgefangen, umgeleitet und mitgeschnitten werden. Diese Bedrohung ist insbesondere für Unternehmen sehr real, da die von ihnen gespeicherten Daten maßgeblich für den ordnungsgemäßen Ablauf der Geschäftsprozesse sind. Zusätzlich sind im Netzwerk auch sensible Informationen wie Kundendaten oder geplante Produktentwicklungen gespeichert. Diese zu schützen ist das oberste Ziel der IT-Sicherheit. Um dieser Verantwortung gerecht zu werden, versucht sie kritische Geschäftsbereiche zu identifizieren und entsprechend vor Angriffen zu schützen. Dazu werden je nach aktueller Gefährdungslage konkrete Maßnahmen implementiert. Diese Maßnahmen sind für jedes Unternehmen und jeden Geschäftsbereich individuell, jedoch gibt es den sogenannten Grundschutz, den jedes Unternehmen implementieren sollte. Definiert wird dieser im BSI-Grundschutzkatalog.

In der Vergangenheit sind bereits einige Angriffe gegen Netzwerk- und netzwerkfähige Geräte bekannt geworden. Insbesondere ist hier das Mirai-Botnet zu nennen, welches in der Zeit zwischen August 2016 und Februar 2017 zahlreiche Geräte, insbesondere IoT-Geräte, mit Schadsoftware infizierte [2]. Die Aufgabe des Botnetzes ist es, bekannte Webseiten per DDoS anzugreifen. Die rasante Verbreitung der Malware begründet das große Forschungsinteresse. Zeitweise waren die angegriffenen Seiten, wie zum Beispiel

the Guardian, CNN und Reddit nicht mehr erreichbar, da viel zu viele Anfragen verarbeitet werden mussten [3]. Dafür reichte die zur Verfügung stehende Rechenleistung der Server einfach nicht aus. Ausbreiten konnte sich die Schadsoftware durch ihre ausgefeilte Funktionalität der Weiterverbreitung. Bereits infizierte Geräten konnten komplett selbstständig weitere verwundbare Geräte finden und diese infizieren. Um überhaupt einen Zugriff auf ein Gerät zu ermöglichen, wurde ein brute-force-login[1] mit besonders häufigen Benutzernamen und zugehörigen Passwörtern gestartet. Die Effizienz dieses Angriffs kann gesteigert werden, indem sogenannte Wortlisten hinterlegt werden. Sie enthalten besonders häufig genutzte Zugangsdaten und minimieren damit die für einen Angriff benötigte Zeit. Gleichzeitig steigt die Wahrscheinlichkeit des Erratens der richtigen Zugangsdaten an [4]. Diese Phase der Infektion war insbesondere erfolgreich, da viele Nutzer die Werkseinstellungen der Geräte weiterverwendeten oder zu schwache Passwörter wählten. Nach einem erfolgreichen Login wurde die IP-Adresse zusammen mit den entsprechenden Zugangsdaten an den Report-Server[2] gesendet. In der nächsten Phase erfolgte ein weiterer Login auf dem Gerät, bevor der Download von gerätespezifischer Malware[3] initiiert wurde [2]. Hinter dem Angriff wurde zunächst eine staatliche Institution vermutet, später stellte sich jedoch heraus, dass der Schadcode von einem Mitglied mit dem Namen Anna-senpai in ein Hackerforum gepostet worden war [5].

Im Mai 2018 veröffentlichte die Cisco Talos Gruppe ihre gesammelten Erkenntnisse zu einer Malware namens VPNFilter. Das Besondere an dieser Malware ist, dass sie aus mehreren Komponenten besteht, welche zum Teil selbst nach einem Neustart auf dem Gerät bestehen bleiben. Befindet sich die Malware einmal auf dem Gerät, so meldet dieses sich beim command and control server[4]. Gleichzeitig nimmt er die gestohlenen Daten in Empfang und speichert diese. Ein solcher Server hält verschiedene Schadcodes bereit und kann durch Plug-Ins[5] um weiter Funktionen erweitert werden [6]. Durch die Übernahme eines command and control servers durch das FBI wurde das Nachladen von Schadcode verhindert. Als besonders problematisch erwies sich der sogenannte kill-Befehl, welcher das Gerät durch Überschreiben der ersten 5000 Byte unbrauchbar macht.

---

[1] Unter einer brute-force-Attacke versteht man das Ausprobieren von Kombinationen, zum Beispiel aus Nutzername und Passwort, bis die richtige Kombination gefunden ist und ein Angreifer so unberechtigt Zugriff erhalten kann.

[2] Als Report-Server wird in dieser Arbeit ein Server verstanden, der die IP eines betroffenen Gerätes und dessen Zugangsdaten gesendet bekommt und im Gegenzug spezifische Malware an das Gerät sendet [2].

[3] Der Begriff Malware setzt sich aus den englischen Begriffen für bösartig und Software zusammen. Dahinter verbirgt sich Software, die dazu entwickelt wurde, das Gerät auf dem sie ausgeführt wird zu schädigen.

[4] Ein command and control server ist ein Server, der vom Angreifer genutzt wird, um Befehle an infizierte Geräte zu schicken.

[5] Ein Plug-In ist ein Stück Programmcode, welches nachgeladen werden kann, um so neue Funktionen zu einem bereits bestehenden Programm hinzuzufügen.

Möglicherweise betroffene Geräte sollten neugestartet werden, um die Malware zumindest vorübergehend zu löschen [7]. Eine gezielte Überwachung der Geräte hätte zu einer zeitnahen Entdeckung des veränderten Codes beitragen können. Wäre dieser sofort entsprechend entfernt worden, so wäre zumindest das Nachladen von Plug-Ins verhindert worden. Durch eine Trennung betroffener Geräte vom Netz hätte die weitere Ausbreitung auch nach der zweiten Phase des Befalls verhindert werden können. Online finden sich Softwarelösungen, die nach eigener Aussage den Schadcode von den Geräten entfernen können. Viele Magazine und Reporter warnen vor diesen Programmen, welche zum Teil selbst Schadcode enthalten. Bisher scheint das Neuaufspielen der Software und der entsprechenden Updates[6] die beste Alternative zu sein. Um ein Update aufspielen zu können, ist meist eine Unterbrechung des Betriebs erforderlich.

Ein weiteres Risiko stellt die auf den Geräten befindliche Software dar. Diese kann bereits Schwachstellen enthalten und dadurch die Sicherheit des Geräts und auch die Sicherheit des gesamten Netzwerks massiv beeinflussen. Besonders verdeutlicht wird dieses Risiko durch den EXTRABACON genannten Exploit[7], der Cisco ASA Firewalls zum Ziel hat [8]. Der Exploit umgeht die Nutzerauthentifizierung am Gerät. Diese Methode, um Zugriff auf ein Gerät zu erlangen, wird Authentification Bypass genannt. Um Extrabacon erfolgreich ausführen zu können, muss SNMP auf dem Gerät aktiviert sein [8]. Das ist ein relativ gebräuchliches Protokoll zum Managen eines Netzwerks. Trotzdem stellt der Exploit ein großes Sicherheitsrisiko dar, weil er bei korrekter Ausführung die Kontrolle über das komplette Gerät erlangt.

Im Sommer des Jahres 2018 wurde ein Hack der russischen PIR Bank bekannt. Die Angreifer stahlen nahezu eine Million Dollar von der Bank. Als Einstiegspunkt wurde vermutlich ein Cisco 800 series Router benutzt. Der Support[8] für die Software auf dem Gerät (Cisco IOS 12.4) war dieser Quelle zu Folge bereits im Jahr 2016 eingestellt worden [9]. Trotzdem wurde das besagte Gerät noch im Netzwerk einer Außenstelle benutzt. An sich verbirgt sich hinter diesem Hack kein besonders ausgeklügelter Angriffsvektor, jedoch zeigt er, dass auch bekannte Sicherheitslücken noch ausgenutzt werden können. Prinzipiell wurde als erster Angriffspunkt eine bereits lange bekannte Schwachstelle in der Software genutzt, um Halt im System zu finden. Anschließend wurde das restliche Netzwerk erkundet, bevor es zur sogenannten privilege escalation[9], also zur Erlangung von höherwertigen Zugangsrechten kam. Im Anschluss wurden diese Rechte genutzt, um die

[6] Als Update wird im Rahmen dieser Arbeit ein umfangreicheres Stück Software verstanden, welche zusätzlich zur Behebung von Problemen auch neue Funktionen implementiert [30].

[7] Der Begriff Exploit beschreibt im Kontext der Informatik ein Tool, welches Schwachstellen in einem Computer oder auch einem Netzwerkgerät ausnutzen kann.

[8] Mit Support ist eine Dienstleistung gemeint, die es zum Ziel hat Probleme in Hard-und Software zu beheben.

[9] Bei der privilege escalation werden niederwertige Zugriffsrechte systematisch so erweitert, dass ein Angreifer die volle Kontrolle über das Gerät erlangen kann.

Transaktionen durchzuführen. Im Rahmen dieser Bachelorarbeit ist der Vorfall von besonderem Interesse, da er die Autorin zur Themenfindung inspirierte und die Untersuchungen genau an diesem Modell durchgeführt wurden.

Die Sicherheit von Netzwerkgeräten ist für die Sicherheit des gesamten Netzwerks also von enormer Bedeutung, wie die zuvor aufgeführten Beispiele eindrucksvoll illustrieren. Wird dies bei den Überlegungen zum Thema IT-und Netzwerksicherheit nicht entsprechend berücksichtigt, so können kritische Schwachstellen entstehen. Das Hacking von Netzwerkgeräten scheint momentan vermehrt aufzutreten, da diese besonders oft nur schlecht gesichert sind und Updates nur unregelmäßig installiert werden. Gleichzeitig können durch sie relevante Informationen gewonnen werden und ein Zugriff auf das gesamte Netzwerk wird möglich. Nicht umsonst werden Netzwerkgeräte auch manchmal als low-hanging-fruits bezeichnet. Der Begriff low-hanging-fruits vergleicht die Komplexität von Aufgaben mit einem Obstbaum. Im Zusammenhang mit der Netzwerksicherheit bedeutet das, je weniger ein Gerät gesichert ist, desto weiter unten im Baum befindet es sich und desto leichter lässt es sich hacken. Will jetzt also bildlich gesprochen ein Angreifer das Obst pflücken, so wird er nicht die Früchte in der Krone des Baumes zuerst pflücken, sondern er wird sich von unten nach oben vorarbeiten. Die Durchführung eines Angriffs ist ähnlich. Ein Angreifer wird zuerst nach wenig gesicherten Angriffszielen suchen, ehe er sich den besser geschützten, bildlich also weiter oben hängenden Früchten, widmet. Sie können leichter manipuliert werden als gehärtete Endpoints[10] und Server, trotzdem enthalten sie lohnenswerte Informationen, die den Aufwand eines Angriffs rechtfertigen.

1.2 Problemstellung

Die im Kapitel 1.1 beschriebenen Beispiele zeigen, dass die Sicherheit von Netzwerkgeräten viel zu oft vernachlässigt wird. Die Geräte entwickeln sich zunehmend zu den low-hangig-fruits eines Netzwerks und werden demzufolge in Zukunft vermutlich noch stärker in den Fokus der Angreifer rücken. Maßnahmen, um Angriffe zu verhindern oder zumindest zu erschweren, sind häufig nur unzureichend implementiert. Auch nach Angriffen werden die Sicherheitsvorkehrungen nicht ausreichend verstärkt. Häufig wird das betroffene Gerät nur vom Netz getrennt und durch ein neues ersetzt. Eine Nachbearbeitung und gründliche Analyse des Angriffsweges findet nur selten statt, da es für das betroffene Unternehmen zu kostspielig ist oder der Vorfall als nicht bedeutend genug eingestuft wird. Noch verheerender ist es, wenn ein solcher Angriff unentdeckt bleibt. Dann kann ein Angreifer auf unbestimmte Zeit im Netz verweilen und es entsprechend seiner eigenen Bedürfnisse und Ziele verändern.

[10] Als Endpoint wird ein internetfähiges Gerät in einem Netzwerk bezeichnet. Endpoints sind zum Beispiel Laptops, Smartphones, Tablets, Drucker und Desktop-PCs [32].

Insbesondere im Bereich der Heimnutzer ist vielen Menschen nicht klar, dass ihre Netzwerkgeräte regelmäßige Pflege und Wartung benötigen. Einigen der Anwender scheint selbst die Notwendigkeit von Updates für Router nicht geläufig zu sein. Zusätzlich wird das Gefährdungspotential herabgewürdigt, was soll ein Angreifer schließlich mit den Daten einer einzelnen Person wollen. Diese Denkweise führt dazu, dass es eine Unmenge an nicht oder nur unzureichend gesicherten Netzwerkgeräten gibt. Durch die zunehmende Verbreitung des Home-Office werden Firmendaten immer häufiger in private Netzwerke übertragen, damit Mitarbeiter auch von zu Hause mit den Daten arbeiten können. Zwar werden dazu bestimmte Schutzverfahren, wie zum Beispiel Token implementiert, aber die Sicherheit des Heimnetzwerks eines Mitarbeiters kann ein Unternehmen nicht überprüfen. In Zukunft ist es also vorstellbar, dass Angriffe gegen private Netzwerke häufiger auftreten, da sich auch in diesen Netzwerken Firmendaten befinden können und diese häufig wesentlich schlechter als das Unternehmensnetz gesichert sind.

Ein großes Problem stellen auch fehlende Kenntnisse über die Konfiguration der Geräte dar. Da diese relativ aufwändig ist, wird sie häufig nicht richtig durchgeführt. Als Folge gibt es viele Netzwerkgeräte, die die vom Hersteller voreingestellten Einstellungen und Zugangsdaten verwenden. Diese sind nicht für diesen Verwendungszweck gedacht und sollten nach der Inbetriebnahme sofort angepasst werden. Geschieht dies nicht, so bietet sich dem Angreifer ein vielversprechender Angriffsvektor, der zumeist mit wenig Aufwand implementiert werden kann. Eine oft verwendete Methode ist der Brute-Force-Angriff, bei dem solange Kombinationen aus Nutzernamen und Passwort an das Gerät gesendet werden, bis die richtige Kombination erraten wurde. Diese Art des Angriffs wurde in Kapitel 1.1 bereits besprochen.

Die Angriffsvektoren, um Netzwerkgeräte zu attackieren, sind äußerst vielfältig. Soll zum Beispiel die Hardware verändert werden, so bietet sich eine supply chain attack[11] an. Zusätzlich könnte auch die Software des Gerätes verändert werden, um so das korrekte Funktionieren des Gerätes zu verhindern. Dies ist auf verschiedene Arten möglich, die Entdeckung fällt jedoch häufig schwer. In der Herstellersoftware können sich ebenfalls schon Schwachstellen befinden, welche ein Angreifer ausnutzen kann. Nicht zuletzt spielen auch die vorgenommenen Einstellungen eine große Rolle, um überhaupt abschätzen zu können, ob ein Gerät gewisse Sicherheitsstandards erfüllt. Die eben beschriebenen Angriffsvektoren stellen nur eine Auswahl dar und sind den Nutzern meistens nicht bekannt.

Einige Arbeiten thematisieren die Sicherheit von Netzwerkgeräten bereits, jedoch sind sie häufig für den durchschnittlichen Anwender zu komplex und mit einem hohen Aufwand verbunden. Die konkreten Maßnahmen in den einzelnen Quellen unterscheiden sich mit-

[11] Auf dem Lieferweg erfolgt eine Veränderung am Gerät, welche ein Angreifer für seine Zwecke ausnutzt. Diese Veränderungen können vielfältig sein. Denkbar ist zum Beispiel das Aufbringen eines zusätzlichen Chips auf die Platine des Geräts.

unter maßgeblich voneinander und verunsichern den Nutzer zusätzlich. Es bleibt also festzustellen, dass es den Anwendern an leicht zugänglichem Wissen fehlt, um die Sicherheit eines Netzwerkgerätes abschätzen zu können. Bisherige Arbeiten verwenden zur Untersuchung sehr komplexe Verfahren, welche nicht ohne Weiteres von Nutzern durchgeführt werden können.

1.3 Zielstellung

Die vorliegende Abschlussarbeit dient als Hilfsmittel zur Abschätzung der Sicherheit eines Netzwerkgerätes. Dem Leser wird zunächst vermittelt, warum Netzwerkgeräte ein potentielles Angriffsziel darstellen können und wie lohnenswert dieses Ziel aus Angreifersicht ist.

Anschließend wird ein Vorgehensmodell vorgestellt werden, welches die vorhandenen Untersuchungsmethoden in eine sinnvolle Reihenfolge bringt und diese Abfolge für den Leser nachvollziehbar begründet. Zusätzlich werden erste Hinweise zur Durchführung der einzelnen Verfahren gegeben.

Das benötigte Methodenwissen, um die Untersuchungen durchführen zu können, wird dem Leser ebenfalls vermittelt werden. Dazu wird das Vorgehen zunächst theoretisch erläutert, ehe dann die konkrete Durchführung am gewählten Untersuchungsobjekt beschrieben wird.

Ziel ist es, den Leser zu selbstständigen Untersuchungen zu motivieren und für diese entsprechendes Fachwissen zur Verfügung zu stellen. Außerdem wird beim Leser ein Problembewusstsein für das Thema der Netzwerkgerätesicherheit geschaffen.

1.4 Kapitelübersicht

Die vorliegende Bachelorarbeit besteht aus 7 Kapiteln.

Das **Kapitel 1** dient als Einleitung und ermöglicht dem Leser den Einstieg in das Thema der Bachelorarbeit. Dazu werden die Motivation, die Problemstellung und die Zielstellung definiert.

Im **Kapitel 2** werden Gründe für das Attackieren von Netzwerkgeräten untersucht, die Auswahl des konkreten Untersuchungsobjekts begründet und das Betriebssystem Cisco IOS kurz vorgestellt.

Anschließend wird in **Kapitel 3** ein allgemeines Vorgehensmodell für die Untersuchung von Netzwerkgeräten entwickelt. Die darin erwähnten Methoden werden in **Kapitel 4** theoretisch erklärt.

Die konkrete Durchführung der einzelnen Untersuchungsmethoden wird in **Kapitel 5** besprochen.

Die aus der Arbeit gewonnenen Erkenntnisse werden in **Kapitel 6** zusammengefasst und anschließend in **Kapitel 7** diskutiert.

2 Grundlagen

In diesem Kapitel werden mögliche Gründe für das Attackieren von Netzwerkgeräten ausführlich beschrieben Zusätzlich wird geklärt werden, warum Netzwerkgeräte als potentielle Ziele überhaupt in Betracht kommen und welche Ursachen es dafür gibt. Anschließend wird geklärt, warum die Autorin sich für das spezifische Untersuchungsobjekt entschied. Der grundlegende Aufbau des Betriebssystems auf dem Untersuchungsobjekt wird ebenfalls thematisiert.

2.1 Gründe für das Attackieren von Netzwerkgeräten

Netzwerkgeräte eignen sich insbesondere als Einstiegspunkte in ein ansonsten gut gesichertes Netzwerk [10]. Hat ein Angreifer Zugang zu dem Gerät erlangt, so kann er es benutzen, um weitere Angriffsphasen vorzubereiten und durchzuführen. Ein gehacktes Netzwerkgerät würde zum Beispiel das Sniffen[12] des gesamten Netzwerkverkehrs erlauben. Der so erhaltene Datenverkehr kann genutzt werden um abzuschätzen, welche Geräte sich in einem Netzwerk befinden. Zusätzlich ist es möglich, Daten umzuleiten bzw. deren Eingang beim Empfänger zu verhindern.

Der in der Informatik bekannte Grundsatz „never change a running system“, in Deutsch: verändere nie ein funktionsfähiges System, findet häufig auch bei Netzwerkgeräten Anwendung. Durch das Einspielen von Patches[13] können Kompatibilitätsprobleme auftreten. Um Kompatibilitätsprobleme ausschließen zu können, müssten vor jedem Patchen umfangreiche Tests und Analysen durchgeführt werden. In Unternehmen ist dies meist nicht möglich, da es an einer entsprechenden Testumgebung fehlt, die das gesamte Firmennetzwerk zuverlässig nachbildet. Ungetestete Patches stellen ein Risiko dar, da Datenverluste und längere Ausfallzeiten auftreten können. Auch ohne Komplikationen sind die Geräte während des Patchens vorrübergehend nicht erreichbar, für einige Geschäftsbereiche ist selbst so eine geplante Ausfallzeit nicht tragbar. Daraus resultierend stellen selbst lang bekannte Schwachstellen und Exploits aus Angreifersicht einen lohnenswerten Angriffsvektor dar [10].

[12] Der Begriff sniffen bezeichnet in der Informatik das Mitschneiden von Datenverkehr in einem Netzwerk, um diesen auf Auffälligkeiten hin zu untersuchen.

[13] Ein Patch ist ein Stück Software, welches Fehlfunktionen einer Software beheben oder eventuelle Sicherheitslücken schließen soll.

Wie bereits im Kapitel 1.2 erwähnt ist insbesondere das Risikobewusstsein von Heimnutzern ein Problem. Es gibt wenige Möglichkeiten einer qualitativ hochwertigen und gleichzeitig finanziell erschwinglichen Weiterbildung, die insbesondere Heimnutzer anspricht. Neue Phänomene wie Home-Office und verschiedene Cloud-Systeme sorgen dafür, dass die Datenmenge auch in privaten Netzen zunimmt. Selbst wenn Daten nur temporär im Heimnetzwerk vorliegen, kann ein Angreifer so Zugriff auf sensible Firmendaten erlangen.

Die Kostenpflichtigkeit der meisten Firmwareimages[14] verleitet unerfahrene Nutzer dazu, vermeintliche Images[15] aus unbekannten Quellen zu extrahieren. Der tatsächliche Inhalt der heruntergeladenen Datei könnte zum Beispiel eine Malware oder ein verändertes Image sein. Besonders raffiniert an dieser Methode ist, dass der Angreifer sich kaum um die Verbreitung kümmern muss. Er muss lediglich den getarnten Schadcode bereitstellen und kann nach erfolgter Infektion seinen Angriff fortführen.

2.2 Begründung der Auswahl des Untersuchungsobjekts

Cisco galt im Bereich der geschäftlich genutzten Netzwerkgeräte jahrelang als Marktführer. Mittlerweile gibt es unzählige Hersteller, doch Cisco steht mit einem ungefähren Marktanteil von 55% immer noch an der Spitze der aktuellen Verkaufszahlen [11]. Der Ruf des Herstellers galt lange Zeit als gut, in letzter Zeit kamen jedoch Zweifel auf. Dazu trug unter anderem der in Kapitel 1.1 erwähnte Exploit EXTRABACON bei. Von Cisco hergestellte Geräte sind meist im oberen Preissegment zu finden. Folglich erwarten die Kunden sowohl exzellent verarbeitete Hardware als auch sichere Software mit regelmäßigen Updates und Patches.

Die 800 ISR Serie gilt seit Ende des Jahres 2016 als veraltet und wird nicht mehr unterstützt. Der Fall des PIR-Bank-Hacks, in Kapitel 1.1 beschrieben, zeigt aber, dass diese Geräte immer noch verwendet werden [9]. Der Grund mag in der kombinierten Funktion aus Router und Switch liegen. Dadurch wird nur ein Gerät benötigt, obwohl es unterschiedliche Aufgaben erfüllen kann. Zusätzlich werden auf diesen Geräten sogenannte signierte Images[16] genutzt. Dadurch soll ein weiterer Schritt in Richtung Überprüfbarkeit der Authentizität[17] der Firmware gemacht werden.

---

[14] Als Firmware wird die vom Hersteller bereitgestellte, auf dem System befindliche Software bezeichnet. Diese ist im nicht-flüchtigen Bereich des Speichers abgelegt und sorgt dafür, dass ein Gerät bestimmte Funktionen erfüllen kann [31].

[15] Als Image wird die Kopie eines Datenspeichers, die in einer Datei gespeichert wird, bezeichnet.

[16] Damit bezeichnet Cisco Images, die mittels eines Zertifikats vom Hersteller gekennzeichnet wurden. In der Informatik wird ein digitales Zertifikat genutzt, um Informationen über die Identität zur Verfügung zu stellen, in diesem Fall Informationen über den Hersteller der entsprechenden Firmware.

[17] Mit dem Begriff Authentizität ist in diesem Fall gemeint, dass die Firmware tatsächlich in der vorliegenden Form von Cisco produziert wurde.

2.3 Aufbau und Funktionalität Cisco IOS

Beim Starten des Geräts wird die Software aus dem nicht-flüchtigen Speicher geladen. Bei Geräten des Herstellers Cisco wird diese Art von Speicher durch einen Flash-Chip[18] realisiert. Die Firmware liegt häufig jedoch nicht in sofort lauffähigem Code vor. Damit der Code ausgeführt werden kann, muss er entsprechend umgewandelt werden.

Für die Konfiguration des Gerätes können verschiedene Werkzeuge genutzt werden. Zu den Möglichkeiten gehören SSH, Telnet und die Kommandozeile. SSH ist ein Protokoll, dass es ermöglicht, eine sichere Verbindung zu einem Gerät über ein unsicheres Netzwerk aufzubauen. Telnet ist ebenfalls ein Protokoll. Es ermöglicht es einem Nutzer, sich von einem Computer aus in einen anderen Computer oder ein anderes Gerät innerhalb desselben Netzwerks einzuloggen. Für die Kommandozeile hat Cisco eine eigene Syntax entwickelt, welche jedoch starke Ähnlichkeit mit der unixoiden[19] Kommandozeile hat. Auf den Geräten gibt es vier verschiedene Berechtigungs- und Konfigurationsebenen. Alle werden durch verschiedene Wort-und Zeichenkombinationen am Beginn einer Kommandozeile angezeigt.

Tabelle 1: Einstellungs- und Berechtigungsebenen Cisco IOS

Name	**Prompt**	**Funktionen/ Eigenschaften**
user mode	Router>	Anzeige von Einstellungen, beim Start automatisch dieser Modus
privileged mode	Router#	grundlegende Einstellungen ändern, mit Passwort gesichert
global configuration mode	(config)#	nur aus priv. heraus aufrufbar, legt globale Einstellungen fest
special interface configuration mode	(config-if)#	Einstellungen für spezifische Schnittstelle vornehmen

[18] Ein Flash-Chip ist ein Hardware-Bauteil, welches zum Speichern von Daten genutzt wird.

[19] Als unixoid werden solche Betriebssysteme bezeichnet, die direkt aus dem ursprünglichen Unix entstanden sind oder starke Ähnlichkeiten mit dessen Funktionalität aufweisen.

Es existiert eine Vielzahl von unterschiedlichen Versionen der Firmware, welche nur sehr schwer zu überblicken sind. Dadurch entstand der Begriff der Cisco IOS Diversity [10] [12]. Gelegentlich wird diese als Sicherheitsmerkmal des Herstellers angesehen, jedoch verweisen Cui, Kataria und Stolfo darauf, dass keine bewusste Entscheidung zu dieser Entwicklung führte und sie folglich nicht als Sicherheitsmerkmal betrachtet werden sollte [12].

Im Zusammenhang mit Cisco IOS ist häufig die Rede von einer monolithischen Architektur. Damit ist gemeint, dass die Firmware auf den Geräten als ein ELF-File vorliegt und es keine Abhängigkeiten von externen Modulen oder Bibliotheken gibt [10] [13]. Diese Struktur hat jedoch auch einige Nachteile. Prozesse können gegenseitig in ihre jeweiligen Speicherbereiche eingreifen, da es keine Trennung des virtuellen Speichers gibt [10] [12] [13] [14].

Die Software operiert auf PowerPC-basierter oder Mips-basierter Hardware. Da diese Architekturen mit ELF-Files umgehen können, wird die Firmware entsprechend als 32-bit ELF-File kompiliert. Durch diese spezielle Struktur gibt es keine definierten Symbole und indirekte Funktionsaufrufe werden erst zur Laufzeit mittels spezieller Funktionsaufruftabellen realisiert [14].

3 Vorgehensmodell

In diesem Kapitel wird dem Leser ein leicht verständliches Vorgehensmodell für Untersuchungen zur Sicherheit von Netzwerkgeräten vorgestellt. Dafür werden einzelne Phasen der Untersuchung definiert und entsprechend erläutert. Die individuellen Anforderungen und Bedingungen der einzelnen Untersuchungsschritte werden ausführlich dargelegt, um die Notwendigkeit eines einheitlichen Vorgehensmodells zu begründen.

Im ersten Schritt definiert der Untersuchende eine Zielstellung für die nachfolgenden Untersuchungen. Diese legt den Schwerpunkt fest und begründet die Auswahl einzelner Untersuchungsmethoden. Ohne diesen Schritt ist es nicht möglich, die Untersuchung an die konkreten Gegebenheiten und das Ziel der Untersuchung anzupassen. Außerdem werden auch Nebenbedingungen in dieser Phase beachtet. Dazu gehört zum Beispiel die Frage, ob ein Gerät im Verlauf der Untersuchung beschädigt werden darf oder nicht. Ist dies nicht vorher festgelegt, so kann am Ende zwar die Zielstellung erfüllt sein, aber das Ergebnis ist nicht mehr brauchbar, da es kein weiteres Gerät dieses Typs im Netzwerk gibt. Je nach Situation kann es keine bis sehr viele Nebenbedingungen geben, welche bei der Anpassung des Vorgehensmodells beachtet werden müssen.

Nach der Definition der Zielstellung und den Nebenbedingungen kann mit der Recherche von Informationen begonnen werden. Dieser Teilschritt hat keinen direkten Einfluss auf das Gerät, kann aber bereits wichtige Erkenntnisse für das weitere Vorgehen liefern. Zu den relevanten Informationen gehören Baupläne, Produktinformationen, Erfahrungsberichte, Nachrichtenmeldungen und bereits bekannte Schwachstellen. Die Baupläne sind besonders in Verbindung mit der optischen Untersuchung wichtig, weil durch den Vergleich mit den Bauplänen zusätzlich angebrachte Bauteile identifiziert werden können. In den allgemeinen Produktinformationen finden sich Angaben zu zusätzlichen Funktionalitäten, welche unter Umständen für die weitere Untersuchung genutzt werden können. Durch die Erfahrungsberichte anderer Nutzer kann der Untersuchende sein Vorgehen nochmals kritisch hinterfragen und eventuelle Schwierigkeiten besser abschätzen. Nachrichtenmeldungen über Angriffe und entdeckte Schwachstellen können in Kombination mit der Recherche von bekannten Schwachstellen genutzt werden, um die generelle Sicherheit der Software einzuschätzen.

Im folgenden Schritt wird das Gerät zunächst äußerlich auf Beschädigungen und Veränderungen abgesucht. Bis zu diesem Schritt sind keine Veränderungen am Gerätes nötig. Jetzt soll aber der innere Aufbau mit den im Schritt zwei gefundenen Bauplänen verglichen werden. Dazu muss das Gerät vorsichtig geöffnet werden. Zumeist müssen dafür nur ein paar Schrauben gelöst werden. Trotzdem erlischt durch diese Untersuchung die Herstellergarantie der meisten Hersteller. Ist diese Garantie also von enormer Bedeutung

für den Untersuchenden, so sollte er diesen Schritt nicht durchführen, auch wenn bei korrekter Durchführung keine dauerhaften Beschädigungen am Gerät zu erwarten sind.

Der nächste Schritt beschäftigt sich mit der Überprüfung der gemachten Einstellungen. Werden diese nicht verändert, so ist auch keine Änderung am System notwendig. Wird aber eine falsche Einstellung gefunden, so sollte diese vom Untersuchenden entsprechend angepasst werden. Diese Überprüfung kann händisch erfolgen, jedoch soll in Abschnitt 4.3 über die Nutzung eines Indikators für die Originalität der Einstellungen berichtet werden.

Als letzte Methode der Untersuchung soll die Firmware vom Gerät extrahiert werden und anschließend einem Prozess zur Verifikation ihrer Authentizität unterzogen werden. Diese Methode beinhaltet das größte Risiko für eine Beschädigung oder Veränderung am Gerät. Deshalb sollte dieses Verfahren nur angewendet werden, wenn der Untersuchende entsprechende Kenntnisse hat oder eine Beschädigung des Geräts nicht problematisch ist. Insbesondere für diesen Schritt gibt es viele Möglichkeiten der Durchführung. Diese werden mit allen Vor-und Nachteilen in Kapitel 4.4 besprochen.

Zum Abschluss sollte der Untersuchende sich fragen, ob er durch die Ergebnisse seiner Untersuchungen seine zu Anfang gestellte Forschungsfrage bzw. Zielstellung beantworten kann. Ist dies nicht der Fall, so ist während der Untersuchung der Fokus verloren gegangen. In diesem Fall empfiehlt es sich, nochmal eine Recherche relevanter Informationen durchzuführen, bevor der Rest der Untersuchung nochmals mit verändertem Fokus durchgeführt wird.

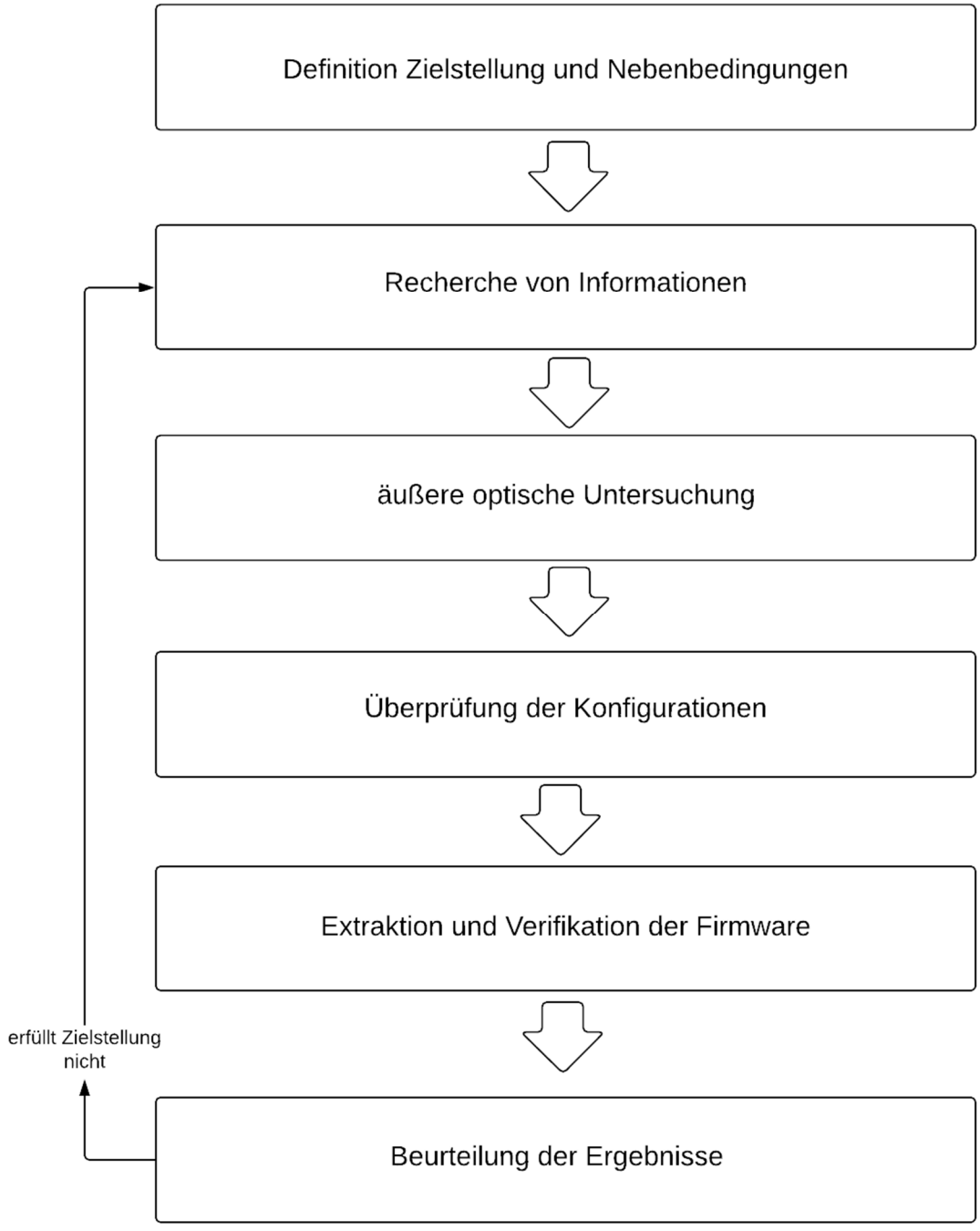

Abbildung 1: Vorgehensmodell zur Untersuchung eines Netzwerkgeräts

4 Untersuchungsmethoden

Dieses Kapitel erläutert die im Vorgehensmodell eingeführten Untersuchungsmethoden theoretisch, zeigt ihren Beitrag zur Untersuchung auf und beleuchtet eventuelle Risiken genauer.

4.1 Recherche relevanter Informationen

Es gibt eine Vielzahl von Informationen, die im Lauf der Untersuchungen relevant werden können. Zuallererst sollte der Untersuchende sich einen Überblick über grundlegende Funktionalitäten und Eigenschaften des Untersuchungsobjekts verschaffen. Dazu können Informationsblätter und Werbebroschüren eine vielversprechende Quelle sein. Die so gewonnenen Informationen können für die weitere Recherche genutzt werden. Im nächsten Schritt sollten die Firmware, ihre Updates und verfügbare Patches berücksichtigt werden. Um an diese Informationen zu gelangen, können die Release[20] – und Patch Notes[21] des Herstellers zu Rate gezogen werden. Auch Baupläne können wichtige Informationen enthalten, jedoch ist es schwer, dieser habhaft zu werden. Als sehr hilfreich hat sich die FCC ID erwiesen. Diese kann auf der Website der Behörde zurückverfolgt werden und zeigt die dazugehörigen verfügbaren Dokumente an [15]. Das heißt nicht, dass die Baupläne in jedem Fall über diese Suche zugänglich sind, aber es stellt eine gute Grundlage für die Suche dar. Auch die Erfahrungsberichte anderer Nutzer können sich als wertvoll erweisen. Sie vermitteln grundlegende Tipps zum Umgang mit dem Untersuchungsobjekt. Außerdem kann es vorkommen, dass ein anderer Nutzer vielleicht schon ähnliche Untersuchungen durchgeführt hat, deren Ergebnisse berücksichtigt werden und die eigenen Methoden entsprechend angepasst werden sollten. Große Sicherheitsprobleme werden häufig auch in den Medien erwähnt. Folglich sollten diese bei der Recherche ebenfalls beachtet werden. Bekannte Schwachstellen und ihre Entdeckungsgeschichten können mögliche Schwachstellen aufzeigen, die bei ähnlichen Geräten genauestens untersucht werden sollten. Dazu können spezielle Datenbanken und Webseiten genutzt werden. Dabei ist es wichtig, das Gefährdungspotential einer Schwachstelle realistisch einzuschätzen, um Aussagen über die Sicherheit bzw. Unsicherheit eines Geräts treffen zu können.

---

[20] Release Notes sind Dokumente des Hersteller, die zusammen mit neuer Software ausgeliefert werden. Darin finden sich zum Beispiel Informationen über verfügbare Funktionen und Veränderungen im Vergleich zu vorherigen Versionen.

[21] Patch Notes sind den Release Notes sehr ähnlich. Der Unterschied besteht darin, dass sie zusammen mit einem Patch ausgeliefert werden und hier eher erklärt wird, welche Fehlerquelle bzw. Schwachstelle behoben wird.

4.2 äußere Untersuchungen und Öffnung des Geräts

Die äußere Untersuchung des Geräts sollte nicht vernachlässigt werden, da auch sie Aufschluss über eventuell vorgenommene Änderungen am Gerät geben kann. Im ersten Schritt wird das Gerät vorsichtig aus seiner Transportverpackung entfernt und auf von außen sichtbare Veränderungen hin untersucht. Das können sowohl Transportbeschädigungen als auch anderweitig entstandene Beschädigungen sein. Die Dokumentation der Schäden ist sehr wichtig, um einerseits nachweisen zu können, dass die Schäden nicht selbst verursacht sind und andererseits Rückschlüsse auf mögliche Entstehungsszenarien zuzulassen.

Nachdem das Gerät äußerlich untersucht wurde, sollten nun die äußeren Abdeckungen entfernt werden, um auch die innen liegenden Bauteile überprüfen zu können. Die Abdeckungen sind mit Schrauben befestigt und können mit entsprechendem Werkzeug leicht gelöst werden. Nun ist der Blick auf die Leiterplatte und die darauf aufgebrachten Chips frei. Deren Anzahl und Anordnung sollte gegebenenfalls mit den bereits gefundenen Bauplänen übereinstimmen. Ist dies nicht der Fall, so müssen für jeden Chip einzeln dessen Funktion und damit der Nutzen im Gerät bestimmt werden. Finden sich Chips ohne deutliche Aufgabe, so ist das ein Indiz für eine supply chain attack, wie bereits in Kapitel 1.2 erwähnt. Diese Form des Angriffs ist besonders subtil, da sie ohne eine optische Untersuchung nicht entdeckt werden kann.

4.3 Überprüfung der Einstellungen

Die Überprüfung der Gerätekonfiguration ist ebenfalls von großer Bedeutung, da Fehlkonfigurationen häufig die Ursache für einen erfolgreichen Angriff sind und Schadcode auch gezielt Einstellungen verändern kann. Prinzipiell gibt es zwei verschiedene Wege, eine solche Überprüfung durchzuführen. Entweder werden alle gemachten Einstellungen von Hand überprüft oder die Einstellungen werden in eine Datei geschrieben und deren Hashwert mit dem einer Kopie der richtigen Einstellungen verglichen.

Der Vorteil der ersten Methode liegt darin, dass ein gefundener Unterschied in der Konfiguration sofort behoben werden kann. Außerdem ist es so auch möglich Einstellungen zu identifizieren, die sich gegenseitig beeinflussen. Zusätzlich wird auch ein Vergleich mit der best practice[22] des Herstellers möglich. Der größte Nachteil dieser Methode liegt in ihrem hohen Zeitaufwand, da jede Einstellung einzeln betrachtet werden muss, ihre Bedeutung für die Sicherheit beachtet werden muss und zusätzlich auch die wechselseitige Beeinflussung mit anderen Einstellungen nicht außer Acht gelassen werden darf. Diese Art des Vergleichs ist sehr fehleranfällig, da ein Mensch bei der Vielzahl von möglichen Einstel-

[22] Mit best practice ist ein Hinweis des Herstellers gemeint, der Einstellungsempfehlungen gibt, um den Betrieb des Geräts möglichst sicher zu gestalten.

lungen sehr schnell den Überblick verlieren kann. Es kann also passieren, dass nicht alle Einstellungsmöglichkeiten betrachtet werden.

Die zweite Methode hingegen ist deutlich schneller in der Durchführung. Viele Geräte, insbesondere Router, bieten die Möglichkeit, alle Einstellungen in einer Datei zu sichern. Wurde nach der initialen Konfiguration eine solche Datei erstellt, kann ihr Hashwert zum Vergleich mit dem Hashwert einer neuen Einstellungssicherung genutzt werden. Wird die Sicherung und die Berechnung des Hashwerts korrekt ausgeführt, so sind Fehler sehr unwahrscheinlich, da der menschliche Einfluss minimiert wurde. Wenn eine Veränderung festgestellt wurde, kann die Einstellungssicherung gleich genutzt werden, um die gewollten Einstellungen wiederherzustellen. Der Nachteil ist, dass eine veränderte Einstellung trotzdem nur durch eine Untersuchung von Hand gefunden werden kann.

4.4 Extraktion und Verifikation von Firmware

4.4.1 Extraktion

Die Extraktion der Firmware aus dem Gerät kann auf mehreren Wegen vorgenommen werden. Dabei ist zu beachten, dass es Methoden gibt, die keine Öffnung des Gerätes verlangen und Methoden, bei denen eine Öffnung notwendig ist. Jede der Methoden hat individuelle Anforderungen und Eigenschaften, die das Ergebnis der Extraktion beeinflussen können.

Eine erste Methode, um Firmware zu extrahieren, ist die Nutzung des CLI[23]. Dieses wird normalerweise genutzt, um Geräte zu konfigurieren und zu administrieren. Dazu wird der entsprechende Port[24], meist mittels RJ45-Kabel, mit einem Computer verbunden. Vom Computer aus werden Anweisungen in einem Terminal angegeben und das angeschlossene Gerät führt diese aus. Da das Gerät bei einer Nutzung des CLI nicht geöffnet werden muss, wird diese Methode auch als nicht invasiv bezeichnet. Der Vorteil liegt darin, dass das Gerät äußerlich unversehrt bleibt und das Risiko einer Beschädigung minimal ist. Nachteilig ist jedoch, dass äußere Faktoren das Ergebnis der Extraktion stark beeinflussen können. Werden volatile Daten[25] benötigt, so ist die Nutzung des CLI meist die einzige funktionsfähige Extraktionsmethode, da diese Daten sonst beim Ausschalten des Geräts verloren gehen. Hersteller wie Cisco statten ihre Geräte mit einer Vielzahl von Funktionen aus. Darunter auch Funktionen, um Daten vom Flashspeicher zu kopieren

[23] Das CLI (command line interface) ist die Kommandozeile, die genutzt werden kann, um das Gerät zu konfigurieren.

[24] Ein Port ist eine Schnittstelle, die die Kommunikation mit einem externen Gerät ermöglich (Bsp: siehe USB)

[25] Als volatil werden solche Daten bezeichnet, die veränderlich sind und mit dem Ausschalten des Geräts verloren gehen.

oder sogar den gesamten Speicherinhalt in einer Datei abzuspeichern. Für die Nutzung dieser Funktionen ist lediglich eine Verbindung zum Gerät und die Kenntnis der korrekten Zugangsdaten notwendig. Sollten diese nicht bekannt sein, so besteht bei entsprechender Konfiguration die Möglichkeit, eine Passwortwiederherstellung durchzuführen [16]. Es wird im Gegensatz zu den anderen Methoden kein Equipment, wie Breakout-Boards oder NAND-Controller, benötigt. Weitere Vor- und Nachteile dieser Methode sollen in der nachfolgenden Tabelle beschrieben werden.

Tabelle 2: Eigenschaften der Extraktionsmethode Nutzung des CLI

Vorteile	**Nachteile**
- im Vergleich schneller als Methoden mit physischer Öffnung	- hohes Risiko von Kompromittierung[26], z.B. durch veränderte Firmware
- kein zusätzliches Material/ Untersuchungsgerät benötigt	- Analysemöglichkeiten durch Kommandoauswahl eingeschränkt
- minimales Risiko der Beschädigung → nicht destruktiv	
- Gerät bleibt physisch intakt	
- Extraktion zur Laufzeit möglich → kein Verlust volatiler Daten	

Eine weitere Möglichkeit zur Extraktion von Firmware stellt die Nutzung von Debugging-Ports[27] dar. In der Vergangenheit verblieben diese einfach auf dem Board, mittlerweile versuchen einige Hersteller, die Nutzung der Debugging-Ports zu verhindern, indem sie

[26] Kompromittierung meint in diesem Zusammenhand, dass Daten willentlich verändert wurden und der Systemeigentümer keine Kontrollmöglichkeit für den Inhalt und das korrekte Funktionieren des Gerätes mehr hat.

[27] Debugging-Port sind Ports, die zum Testen von Boards während der Produktionsphase auf diese aufgebracht werden. Mit ihnen kann das ordnungsgemäße Funktionieren der Bauteile überprüft werden. Durch diese Ports ist es außerdem möglich, direkt mit dem Board zu kommunizieren und Daten daraus zu extrahieren.

diese deaktivieren oder unkenntlich machen. Um überhaupt an die entsprechenden Ports gelangen zu können, muss das Gerät geöffnet werden. Diese Methode ist also zu den invasiven Methoden zu zählen. Wurden in der Informationsbeschaffungsphase Baupläne gefunden, so sind diese jetzt von enormer Bedeutung. Mit ihnen können auch verborgene Debugging-Ports gefunden werden. Häufig sind die Baupläne jedoch nicht frei zugänglich und es hilft nur Probieren, um die entsprechenden Ports zu finden. Ist ein Port gefunden, so kann mittels Tools wie dem JTAGulator oder dem buspirate eine Verbindung hergestellt werden. In seltenen Fällen ist es notwendig, sich eigene Adapter zu basteln, um eine Verbindung zu dem Port herstellen zu können. Jedes verwendete Tool hat eigene Anforderungen und kann mitunter nicht mit jeder Software verwendet werden. Einer der Vorteile dieser Methode ist, dass der Flashcontroller[28] nicht separat gesichert werden und auch nicht aufwändig rekonstruiert werden muss [17]. Das Beschädigungsrisiko ist moderat, insbesondere dann, wenn Standardadapter genutzt werden können. Durch die Nutzung einer falschen Spannung kann es trotzdem zu Datenverlusten oder der Zerstörung des Flashspeichers kommen. Die nachfolgende Tabelle soll wieder die Vor-und Nachteile dieser Extraktionsmethode verdeutlichen.

Tabelle 3: Eigenschaften der Extraktionsmethode Nutzung von Debugging-Ports

Vorteile	**Nachteile**
-keine Rekonstruktion der Controller notwendig	- mehrere Controller involviert → Risiko der Kompromittierung
- ermöglicht tiefes Eindringen in die Struktur	- Vorbereitung sehr aufwändig
- keine Einschränkungen durch Kommandos oder ähnliches	- große Menge von Material notwendig (Adapter und zugehörige Software)
- gilt als vergleichsweise einfach [17] (im Vergleich mit Chip-off)	- Kompatibilitätsanforderungen müssen beachtet werden
- überschaubares Risiko der Beschädigung	

[28] Ein Flashcontroller ist für die Verteilung von Daten auf einem Flash-Speicherchip zuständig. Er legt fest, wo Daten geschrieben werden. Nicht zwangsläufig legt er Daten am Stück ab.

Als letzte Methode steht das Chip-off Verfahren zur Verfügung. Dabei wird der Chip vom PCB gelöst, bevor er dann mit entsprechenden Tools ausgelesen werden kann. Das Ablösen vom Board kann auf unterschiedlichen Wegen erfolgen. In vielen Fällen wird dabei das Board zerstört. Damit ist das Gerät nicht mehr funktionsfähig. Eine weitere Benutzung des Chips kann aber möglich sein, wenn ein baugleiches Board zur Verfügung steht, auf das der Chip aufgebracht werden kann. Zunächst muss selbstverständlich das PCB ausgebaut werden. Für die einfachste Form des Chip-offs wird dieses dann vorsichtig erwärmt. Dabei ist es extrem wichtig, die Temperatur nicht zu hoch zu wählen, weil ansonsten auch der Chip beschädigt werden kann. In einigen Fällen löst sich die Verbindung zwischen Chip und PCB durch die Wärmezufuhr bereits vollständig. Sollte dies nicht der Fall sein, so wird eine Schicht Lötmittel auf den Kontakten verteilt. Anschließend wird diese Schicht erwärmt und die Verbindungen lösen sich. Abschließend müssen die Kontakte wieder von Lötmittel befreit werden, da ein Auslesen ansonsten nicht möglich ist. Sobald dies geschehen ist, wird der Chip in einem Adapter mit dem entsprechenden Layout platziert. Zu den Adaptern gehört eine Software, mit der die Rohdaten extrahiert werden können. Aus den Rohdaten kann mittels weiterer Verfahren ein virtuelles und ein logisches Image erstellt werden. Diese Vorgehensweise wird im Buch „NAND Flash Data Recovery Cookbook” ausführlich beschrieben [17]. Der Einfluss anderer Komponenten wird durch das Chip-off Verfahren minimiert. Eine Kompromittierung durch andere Bauteile ist also nahezu ausgeschlossen. Zusätzlich kann diese Methode selbst dann Erfolg haben, wenn andere Methoden bereits fehlgeschlagen sind. Jedoch wird das Gerät fast immer zerstört, sodass es danach nicht mehr genutzt werden kann. Die nachfolgende Tabelle soll die Vor-und Nachteile dieses Verfahrens gegenüberstellen.

Tabelle 4: Eigenschaften der Extraktionsmethode Chip-off

Vorteile	**Nachteile**
- Einfluss anderer Komponenten minimiert	- destruktiv, Gerät meistens nicht mehr nutzbar
- solange Chip intakt ist, kann Gerät zerstört sein	- extremes Fachwissen des Untersuchenden notwendig
	- hoher zeitlicher Aufwand
	- benötigt viel Material

4.4.2 Verifikation

Zur Verifikation eines Firmwareimages kann ein Dateihash[29] genutzt werden. Eine grundlegende Eigenschaft jeder Hashfunktion ist der Determinismus. Mit Determinismus ist gemeint, dass die gleichen Daten auch den gleichen Hashwert erzeugen, zufällige Elemente fließen nicht in die Berechnung ein [18].

Auch Cisco hat in seinen Geräten Funktionen integriert, um Hashwerte zu berechnen und zu vergleichen. In der einfachsten Variante heißt das, dass ein im Header[30] der Datei gespeicherter Hashwert mit einem zur Laufzeit berechneten verglichen wird. Nur wenn beide Hashwerte übereinstimmen, kann die Software auf dem Gerät ausgeführt werden. Es ist möglich, diesen Kontrollmechanismus auszuhebeln, indem Veränderungen am Dateiheader vorgenommen werden. Dazu muss zunächst der Offset, also die Adresse, des embedded Hash gefunden werden. Der im Header abgelegte Wert wird mit dem eines veränderten Images überschrieben [19].

Im Kapitel 2.2 wurde der Begriff der signierten Images erstmals eingeführt. Um die Images von Herstellerseite zu signieren nutzt Cisco ein asymmetrisches public-key Verfahren[31] [19]. Diese Methode ermöglicht sowohl die Signaturüberprüfung des laufenden Images als auch eine Überprüfung der Authentizität weiter Dateien, die ein Firmwareimage enthalten [19]. Kommt in einem Imagenamen der Zusatz SPA vor, so ist davon auszugehen, dass es sich um ein signiertes Image handelt.

Die Verifikation von zur Laufzeit gepatchten Images ist ebenfalls möglich [19] [10]. Dafür wird immer eine unveränderte Textregion[32] benötigt. Dazu kann ein Core Dump geschrieben werden oder eine Extraktion der Textregion durchgeführt werden. Für die nachweislich unveränderte Textregion wird dann ein Hashwert berechnet, der anschließend mit dem eines zu verifizierenden Images verglichen werden kann. Dieses Verfahren ist möglich, da die Textregion nur read-only Daten enthält und sich deshalb nicht verändern sollte. Stimmen die Hashwerte nicht überein, so hat irgendeine Form der Veränderung stattgefunden und das Image sollte nach Möglichkeit nicht mehr verwendet werden.

---

[29] Bei dem sogenannten Dateihash handelt es sich um einen Wert, der mittels einer Hashfunktion über eine Menge von Daten berechnet wurde.

[30] Der Dateiheader steht immer am Beginn einer Datei und gibt Aufschluss über grundlegende Eigenschaften, wie zum Beispiel Größe der Datei und den Zeitpunkt der letzten Änderung.

[31] Damit ist gemeint, dass von Herstellerseite ein öffentlicher Schlüssel, auch public key, auf dem Gerät abgelegt wird, der die Integrität und Authentizität der Software belegen soll [26].

[32] Die Textregion ist ein Teil des Images und kann aus diesem extrahiert werden.

Neue Softwareentwicklungen, wie Cisco Incident Response und Eclypsium, haben es zum Ziel, den Prozess des Chipmonitoring und der Firmwareverifikation zu vereinfachen. Diese Software ist jedoch eher für Kunden aus der Wirtschaft gedacht und für private Kunden auf Grund ihres Preises und der eingeschränkten Nutzerfreundlichkeit kaum anwendbar. Aus diesem Grund werden die Softwarelösungen in der vorliegenden Arbeit nicht weiter berücksichtigt.

Prinzipiell kann auch die Codeanalyse genutzt werden, um Firmware zu verifizieren. Es gibt zwei grundlegende Ansätze, um eine solche Analyse durchzuführen. Die statische Analyse führt den Code nicht aus, sondern betrachtet die einzelnen Zeilen des Programmcodes. Um diesen überhaupt zu finden und lesen zu können, wird ein Disassembler[33] benötigt. Jede Zeile es Assemblercodes muss einzeln betrachtet werden, um anschließend beurteilt werden zu können. Hier zeigen sich zwei große Probleme des Verfahrens. Es ist sehr zeitaufwändig und so umfangreich, dass ein Mensch es kaum überblicken kann. Folglich kann es vorkommen, dass nicht die vollständige Funktionalität des Codes betrachtet wird. Von Vorteil ist allerdings, dass die ursprüngliche Umgebung nicht aufwändig nachgebildet werden muss [20]. Die statische Analyse kann durch entsprechende Tools und Skripte automatisiert werden. Die zweite Art der Untersuchung ist die dynamische Analyse. Hierbei wird der Code in einer kontrollierten Umgebung, beispielsweise einem Debugger[34] ,zur Ausführung gebracht. Dadurch können die Abschnitte des Codes, die besonders fehleranfällig sind, identifiziert werden. Der Untersuchende kann zu jeder Zeit sagen, welche Anweisung gerade ausgeführt wurde und welche internen Veränderungen sie ausgelöst hat. Damit eine dynamische Analyse durchgeführt werden kann, muss eine Testumgebung eingerichtet und entsprechend der realen Bedingungen initialisiert werden. Dies kann sehr viel Zeit in Anspruch nehmen. Auch eine Kombination aus statischer und dynamischer Analyse ist möglich. Diese Analysemethode wird dann symbolische Ausführung genannt [21]. Dieser Ansatz ist sehr vielversprechend und könnte in Zukunft an Bedeutung gewinnen. Momentan stellt die schlechte Skalierbarkeit noch ein großes Problem dar, dadurch werden Lade-und Berechnungszeiten sehr lang. Es gibt eine Vielzahl von Tools, die zur Unterstützung bei der Codeanalyse genutzt werden können. Dazu gehören unter anderem: IDAPro, radare2, angr, Hopper und BinaryNinja. Es bleibt dem Untersuchenden selbst überlassen herauszufinden, welches Tool er nutzen möchte, da jedes individuelle Schwächen und Stärken besitzt. Häufig ist es auch nützlich, sich bei der Analyse nicht nur auf ein Tool zu verlassen, sondern sie ergänzend zu nutzen.

[33] . Er ist das Gegenstück eines Assemblers und wandelt den maschinenlesbaren Code wieder in menschenlesbaren Assemblercode um.

[34] Debugger sind Programme, die benutzt werden, um Programme und Code zu testen.

5 konkrete Durchführung

Dieses Kapitel beschreibt, wie die in Kapitel 4 theoretisch beschriebenen Methoden konkret umgesetzt wurden, um Informationen für eine abschließende Sicherheitsbeurteilung zu gewinnen.

5.1 optische Untersuchungen

Zunächst wurde die äußere Untersuchung des Geräts durchgeführt. Da das Gerät bereits in Verwendung war, fanden sich an der Außenseite Gebrauchsspuren in Form von Kratzern. Diese lassen sich damit erklären, dass das Gerät seit der ersten Nutzung mehrfach den Standort gewechselt hat. Zusätzlich war auch das Garantiesiegel an der Unterseite des Geräts beschädigt. Dies passiert nur, wenn es gezielt beschädigt wird oder das Gerät bereits geöffnet wurde. Auch dies ist bei Geräten, die bereits längere Zeit in Betrieb sind, nicht ungewöhnlich. Weitere Beschädigungen und Auffälligkeiten konnten durch die äußere Untersuchung nicht gefunden werden.

Für die innere Untersuchung wurden alle Abdeckungen vom Gerät entfernt. Auf den einzelnen Komponenten befanden sich dicke Staubschichten. Dies lässt darauf schließen, dass das Gerät über einen längeren Zeitraum nicht geöffnet und nur wenig bewegt wurde. Diese Schicht musste entfernt werden, um die darunterliegenden Chips und Bauteile identifizieren zu können. Die nachfolgende Abbildung zeigt den inneren Aufbau eines Cisco Catalyst 2900. Die verwendeten und identifizierten Chips wurden markiert. In der Tabelle 5 werden die Chips nochmals benannt und ihre Funktionalität im Gerät erläutert.

Wie bereits erwähnt wird dieser Abschnitt exemplarisch an einem Cisco Catalyst 2960 durchgeführt. Zum Zeitpunkt des Verfassens der Arbeit stand der Autorin das benutzte Modell der 800 Serie nicht mehr zur Verfügung und die bis dahin gemachten Aufnahmen waren qualitativ nicht hochwertig genug, um in dieser Arbeit Verwendung zu finden. Die wichtigsten Bauteile sind jedoch in beiden Geräten analog vorhanden, sodass eine analoge Anwendung des Verfahrens möglich sein sollte.

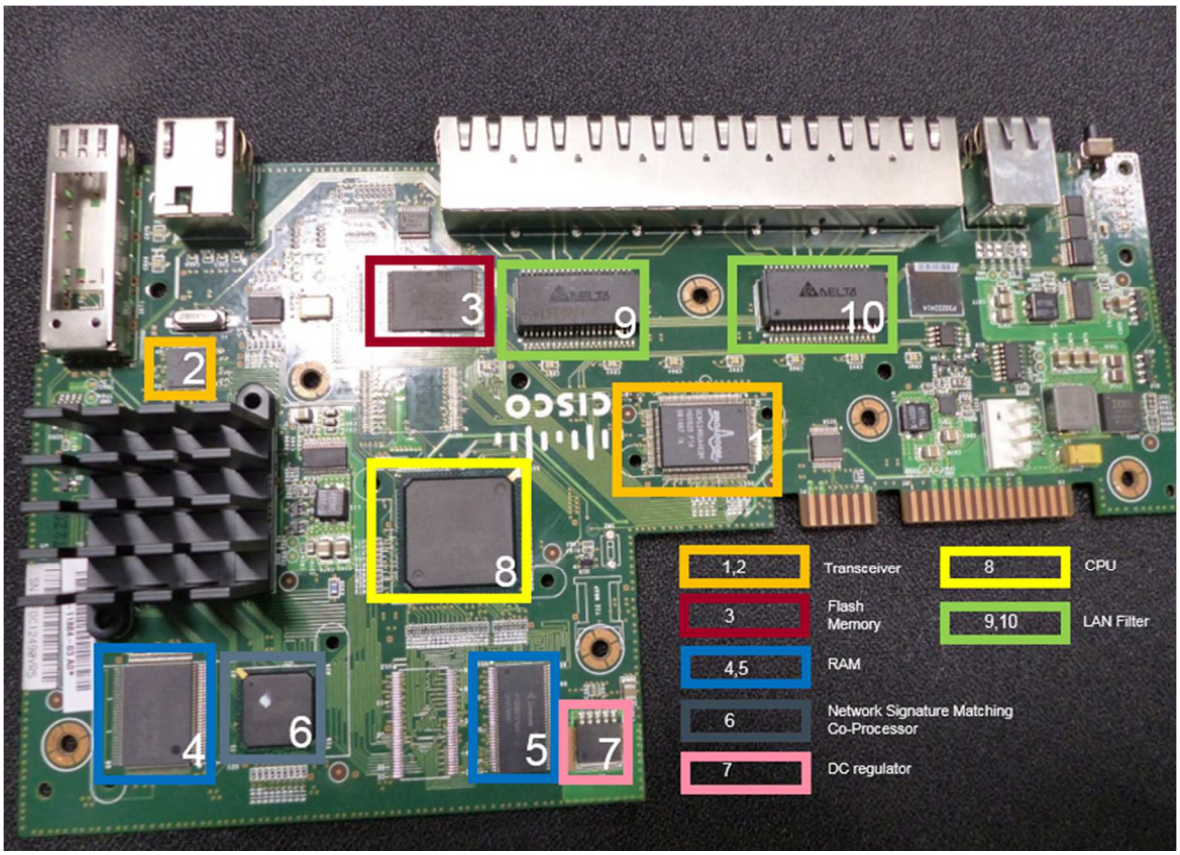

Abbildung 2: innerer Aufbau eines Cisco Catalyst 2960

Die Autorin begann mit der Identifikation bei solchen Bauteilen, auf denen noch eine Beschriftung erkennbar war. Diese gab sie in entsprechende Suchmaschinen ein. Dadurch konnten wiederum Baupläne der einzelnen Chips gefunden werden, welche auch Informationen über den geplanten Verwendungszweck und die Funktion des Bauteils enthielten . Zusätzlich konnte ein Video gefunden werden, welches sehr detailreich zeigt, wie ein solches Gerät geöffnet werden kann und die einzelnen sichtbaren Komponenten benennt [22].

Tabelle 5: innere Untersuchung- Chips und Eigenschaften

Ziffer	**Name**	**Funktion im Gerät**
1,2	Transceiver	Senden und Empfangen von Signalen aus einem Übertragungsmedium (z.B. Licht und elektr. Signale)
3	Flash Speicher	Speicherung des Betriebssystems (Cisco IOS)

4,5	RAM	enthält flüchtige Informationen (z.B. laufende und nicht gespeicherte Konfigurationen), ARP Cache, Routing Tabellen
6	Network Signature Matching Co-Prozessor	Entlastung des Prozessors bei Vergleich von Netzwerksignaturen
7	DC Regulator	Regulation der Stromspannung
8	CPU	Ausführung von Anweisungen, beinhaltet laufende Prozesse
9,10	LAN Fiter	Filterung von Datenübertragungen per LAN

5.2 Einstellungsüberprüfung

Auch diese Untersuchung konnte nicht an einem Gerät der Cisco 800 Serie durchgeführt werden, weil im Untersuchungszeitraum kein entsprechendes Gerät zur Verfügung stand. Deshalb hat sich die Autorin entschlossen, die Untersuchung an einer FRITZ!BOX Fon WLAN 7170 durchzuführen. Bei diesem Gerät handelt es sich um einen Router. In Heimnetzwerken sind Router dieser Firma nicht ungewöhnlich. Die meisten Router für die Nutzung im Heimbereich besitzen ein Konfigurationsinterface, so auch das untersuchte Gerät.

Eine Verbindung zum Gerät herzustellen war während der Untersuchung denkbar einfach. Das Gerät musste nur mit Strom versorgt werden. Anschließend war eine Verbindung per WLAN möglich. Im nächsten Schritt wurde das Konfigurationsinterface aufgerufen. Dies ist ohne Internetzugang aus dem Browser möglich. Dazu wird in die Adresszeile http://fritz.box eingegeben [23]. Dann öffnet sich die Benutzeroberfläche des Geräts.

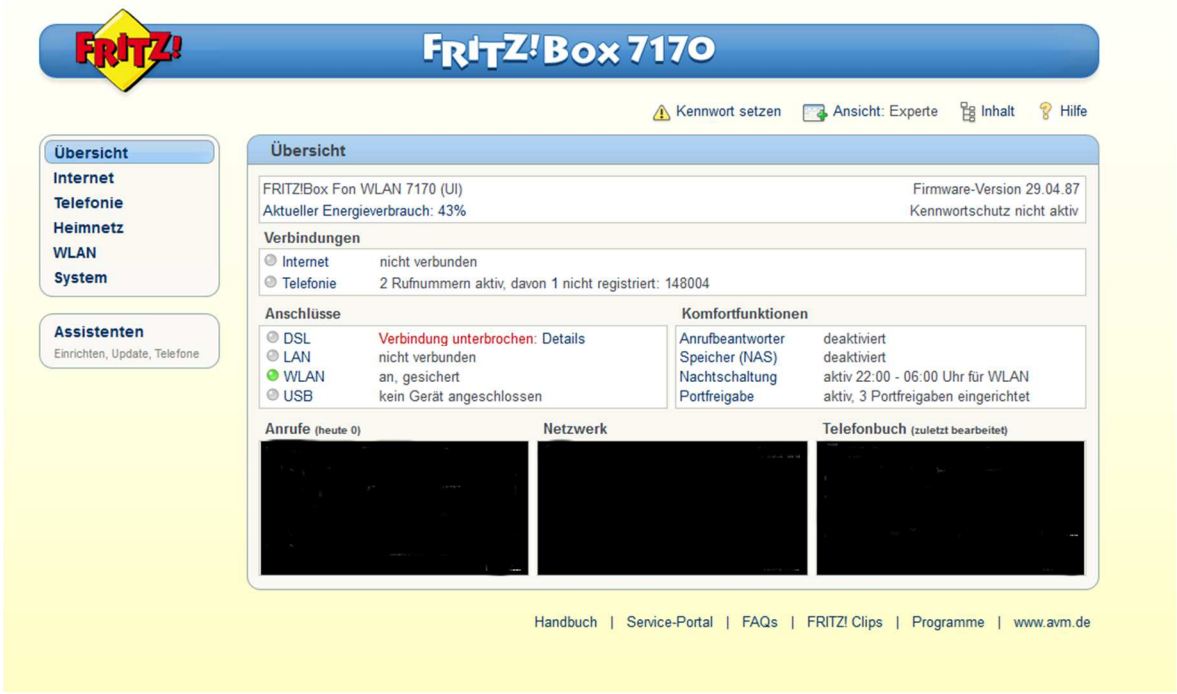

Abbildung 3: Benutzeroberfläche der Fritz!Box WLAN 7170

Hier muss nach dem Unterpunkt System gesucht werden. Es öffnet sich eine neue graphische Oberfläche. Auf der linken Seite findet sich der Unterpunkt Einstellungen sichern. Nachdem dieser angeklickt wurde, wird der Nutzer gebeten, ein Passwort für die Sicherungen zu vergeben. Dies wird jedoch nur benötigt, wenn die Einstellungen aus dieser Datei später für eine Wiederherstellung benutzt werden sollen [24]. Ansonsten werden die Einstellungen in eine Datei geschrieben, welche anschließend gedownloadet wird.

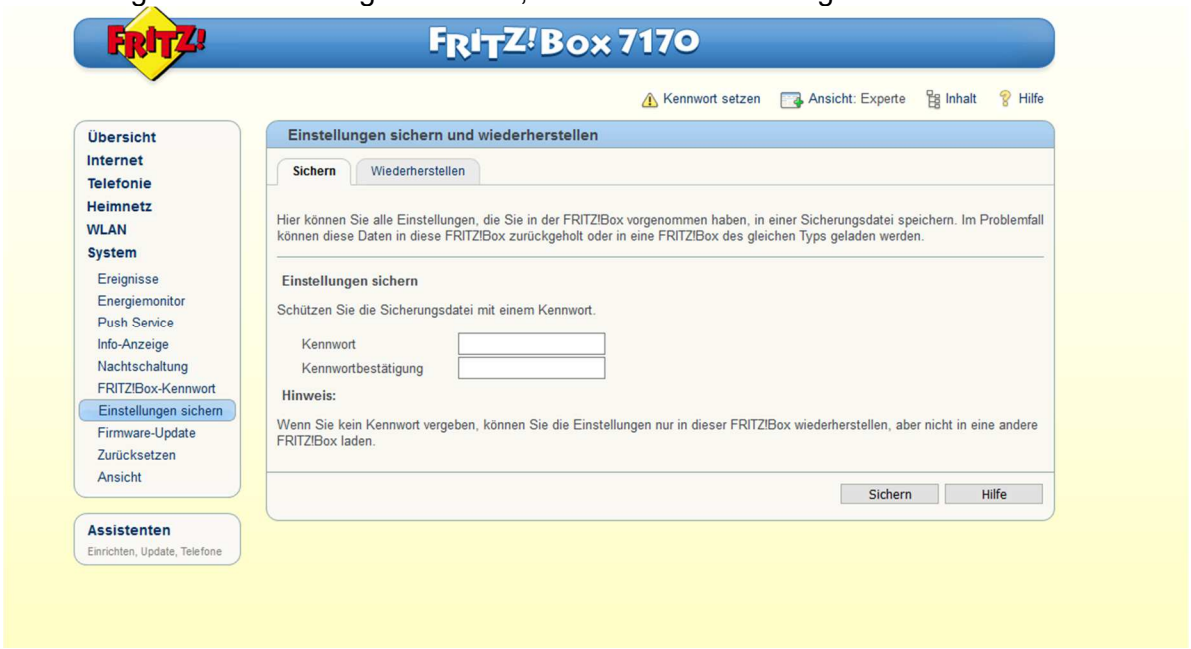

Abbildung 4: Menü zur Datensicherung

Für die so erhaltenen Dateien kann ein Hashwert berechnet werden. Fand zwischen den Einstellungssicherungen eine Veränderung an den Einstellungen statt, so stimmen die Hashwerte nicht überein. Die Autorin hat diese Theorie getestet, indem sie vor der zweiten Datensicherung den Namen des Netzwerks änderte. Als Folge stimmten die Hashwerte nicht mehr überein. Wichtig ist es auch zu wissen, dass die Vergabe von Passwörtern Einfluss auf den Hashwert hat. Demzufolge empfiehlt es sich, kein Passwort für die Datensicherungen zu vergeben, außer sie sollen für eine Wiederherstellung genutzt werden. Dann sind die Einstellungssicherungen aber für einen Vergleich der Hashwerte nicht mehr geeignet. In Anlage, Teil 1 finden sich die Kurznotizen der Autorin und die entsprechend berechneten Hashwerte für die Einstellungssicherung.

5.3 Extraktion und Verifikation der Firmware

Dieser Abschnitt und die zugehörigen Unterkapitel wurden in Anlehnung an den Praxisbericht der Autorin verfasst. Die genutzten Bilder sind dieser Quelle entnommen [25].

5.3.1 Verbindungsaufbau

Um eine Extraktion und Verifikation der Firmware vorzunehmen, muss zunächst eine Verbindung zum Gerät hergestellt werden. Dazu stehen die in Kapitel 2.3 beschriebenen Methoden zur Verfügung. Zusätzlich ist es außerdem möglich, eine webbasierte GUI[35] zu nutzen. Die Autorin entschied sich gegen dieses Vorgehen, da sie Kenntnisse über verfügbare Kommandos und Funktionen des CLI erlangen wollte. Stattdessen wurde eine serielle Verbindung zwischen Gerät und dem Computer hergestellt.

Dazu werden ein Konsolenkabel (RJ45) und ein Terminal beziehungsweise ein entsprechender Emulator[36] benötigt. Im Rahmen dieser Arbeit wurde PuTTY genutzt, weil sich ausführliche Anleitungen für den Verbindungsaufbau zu einem Cisco Gerät fanden. Die Verwendung eines anderen Emulators ist mit entsprechenden Kenntnissen ebenfalls möglich. Für den Emulator mussten die in der nachfolgenden Abbildung dargestellten Einstellungen konfiguriert werden.

[35] Eine GUI ist eine graphische Benutzeroberfläche, die dem Nutzer das Bedienen des zugehörigen Geräts erleichtern soll.

[36] In der Informatik wird ein Emulator benutzt, um die Funktionsweise eines anderen Systems oder Geräts im Ganzen oder zu Teilen nachzuahmen.

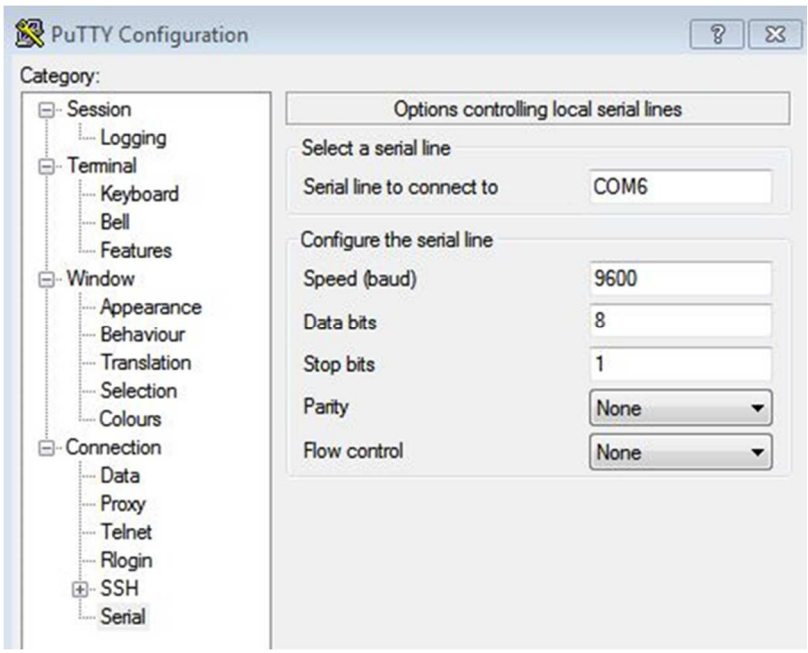

Abbildung 5: Einstellung des Terminalemulators

Eine Verbindung mit diesen Spezifikationen kann auch über ein Skript etabliert werden. Dazu kann Userinput erforderlich sein. Nachdem die entsprechenden Einstellungen vorgenommen wurden, wird die PuTTY-Session gestartet.

5.3.2 Passwortwiederherstellung

Anschließend kann das Gerät mit Strom versorgt werden. Innerhalb der ersten 60 Sekunden nach dem Stromanschluss (dieser Wert kann sowohl nach oben als auch nach unten abweichen) muss die Break-Taste (oder die äquivalente Tastenkombination) auf der Tastatur des Computers gedrückt werden. Dadurch wird der Bootvorgang unterbrochen. Der rommon-prompt wird angezeigt. Um ohne die Eingabe eines Passworts Zugang zum Gerät zu erlangen, muss das configuration register geändert werden. Die Adresse dieses Registers wird dafür auf 0x2142 gesetzt. An dieser Stelle befinden sich aber keine Konfigurationen, also auch kein Passwort. Nachdem dieser Schritt erfolgt ist, wird ein Reset durchgeführt. Das Gerät startet ganz normal, nach dem Booten wird gefragt, ob die initiale Startkonfiguration ausgeführt werden soll, diese Frage muss verneint werden. Die ursprüngliche Konfiguration kann wieder hergestellt werden, indem die startup-config in die laufende Konfiguration geschrieben wird [16].

```
IOS Image Load Test
___________________

System Bootstrap, Version 15.4(1r)T1, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 2014 by cisco Systems, Inc.

Total memory size = 512 MB
Field Upgradeable ROMMON Integrity test
_____________________________________
ROM: Digitally Signed Production Software
C891F-K9 platform with 524288 Kbytes of main memory
Main memory is configured to 32 bit mode

Upgrade ROMMON initialized

monitor: command "boot" aborted due to user interrupt
rommon 1 > confreg 0x2142


rommon 2 > reset
```

Abbildung 6: Unterbrechung des Boot-Prozesses

In diesem Abschnitt wurden die folgenden Kommandos verwendet:

```
\‘Break key pressed‘
confreg 0x2142
reset
no
copy startup-config running-config
```

5.3.3 Erstellung eines virtuellen FTP-Servers

Für den nächsten Schritt wird ein FTP-Server benötigt, da Core-Dumps via FTP auf einen Server übertragen werden können. Im Rahmen der vorliegenden Arbeit wurde xlight genutzt. Die dafür im ersten Schritt vorgenommenen Einstellungen verdeutlicht die nachfolgende Abbildung.

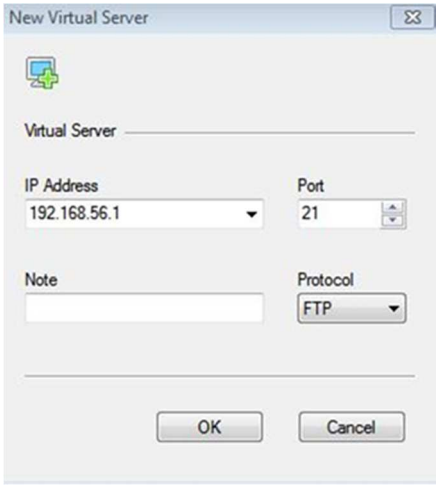

Abbildung 7: Einstellungen des virtuellen Servers

Eine Authentifizierung mittels Username und Passwort ist möglich, wenn die entsprechenden Konten in der Benutzerliste des Servers angelegt sind. Die Ausnahme bildet anonymes FTP, weil hier keine Zugangsdaten abgefragt werden. Nachfolgend wird eine solche Zugangsliste dargestellt.

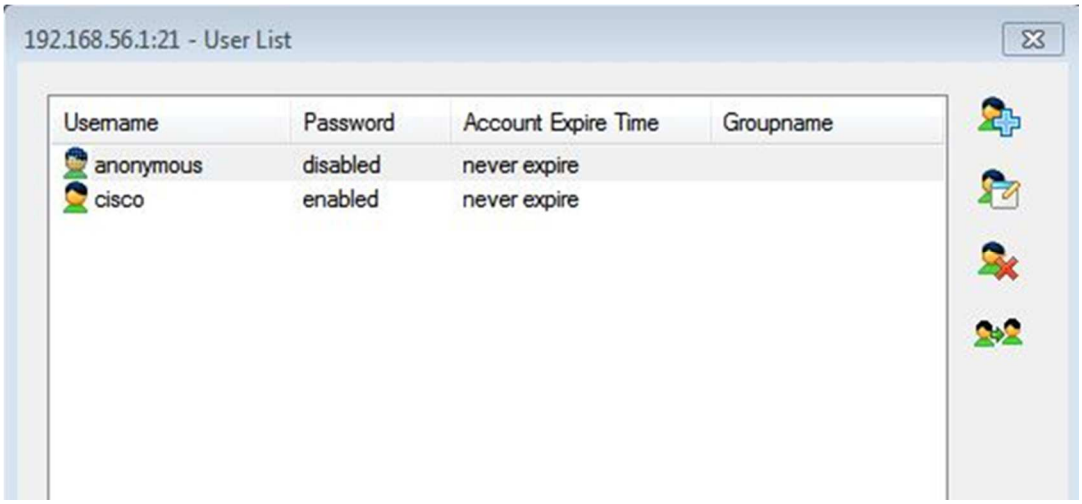

Abbildung 8: Nutzerliste des virtuellen Servers

Bisher besitzt der Server nur einen virtuellen Pfad. Damit die übertragenen Daten später auch auffindbar sind, muss diesem noch ein realer Pfad zugeordnet werden. Diese Zuordnung wird nachfolgend gezeigt.

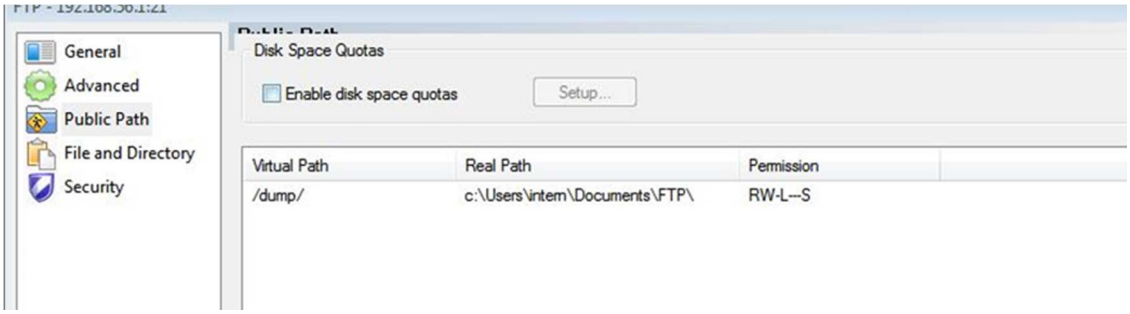

Abbildung 9: Pfadzuweisung des Servers

Nachdem dieser Schritt durchgeführt wurde, ist der Server nun lauffähig und kann gestartet werden. Die Autorin empfiehlt vor der Übertragung eines Core-Dumps zunächst die Verbindung zu überprüfen. Dazu wird ein ping an die Adresse des Servers (192.168.56.1) gesendet.

```
Switch#ping 192.168.56.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.56.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/203/1007 ms
Switch#
```

Dieser Test war erfolgreich und zeigt, dass eine Verbindung zum Server hergestellt werden kann. Er ist aus dem Netz erreichbar.

In diesem Abschnitt wurde das folgende Kommando verwendet:

```
ping 192.168.56.1
```

5.3.4 Einstellungen und Core- Dump

Für den Core-Dump müssen zunächst einige Einstellungen vorgenommen werden. Dazu gehören zum Beispiel der Nutzername für die FTP-Verbindung und das zugehörige Passwort. Nachfolgend werden Teile der notwendigen Konfigurationen dargestellt.

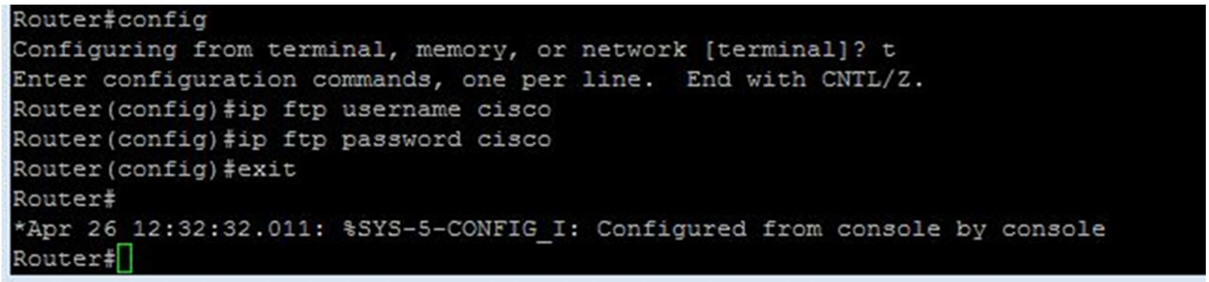

Abbildung 10: Konfiguration für den Core Dump

Das Schreiben des Core-Dump kann mit dem Kommando write core veranlasst werden. Die nachfolgende Abbildung zeigt das Ergebnis dieses Kommandos. Es ist unschwer zu erkennen, dass während des Vorgangs ein Fehler aufgetreten ist. Dieser wurde durch eine falsche Namenseingabe verursacht.

Abbildung 11: write core und Fehlermeldung

Die folgenden Kommandos wurden in diesem Abschnitt genutzt:

```
config t
ip ftp username ‘username’
ip ftp password ‘password’
exception protocol ftp
exception dump ‘ip address’
write core
```

5.3.5 Berechnung und Nutzung von Hash-Werten

Für diesen Abschnitt wird nur noch die Firmware und nicht mehr der gesamte Core-Dump betrachtet. Aus dem Core-Dump kann die Firmware extrahiert werden, es ist aber auch möglich, diese durch ein einfaches Kopieren aus dem Flashspeicher zu erhalten.

Das verify Kommando kann genutzt werden, um den Hashwert für ein bestimmtes Image berechnen zu lassen und diesen mit einem im ELF-Header der Datei abgelegten Hash zu vergleichen. Der Hash wird hierbei nicht über das gesamte Image berechnet, sondern nur über ausgewählte, unveränderliche Teile [26]. Wie bereits in 4.4.2 beschrieben, kann dieser Mechanismus aber umgangen werden. Eine Übereinstimmung dieser beiden Hashwerte ist also ein Indiz für die Echtheit, garantiert werden kann sie dadurch jedoch nicht.

Eine weiterer Hashwert, der verwendet werden kann, ist die sogenannte Router Checksum (CCO). Dabei handelt es sich um einen MD5-Hash, der über die gesamte Firmware berechnet wurde. Dieser kann mit den auf der Herstellerseite veröffentlichten Checksummen verglichen werden. Theoretisch sind zwar Kollisionen der Hashwerte möglich, diese würden auftreten, wenn zwei unterschiedliche Dateien denselben Hashwert erhalten, jedoch kommt es in der Praxis selten zu solchen Fällen.

```
Router#verify flash:c800-universalk9-mz.SPA.153-3.M6.bin
Starting image verification
Hash Computation:    100% Done!
Computed Hash   SHA2: E4086C61591BA87BDC2D64E64D7F9B27
                      94B15794C0AE22B22676817E8BA1E3EE
                      4966D024AEB3BD2FCB0BDB390EF6208F
                      F064ADB29E59B39F5352BA55A7E0164A

Embedded Hash   SHA2: E4086C61591BA87BDC2D64E64D7F9B27
                      94B15794C0AE22B22676817E8BA1E3EE
                      4966D024AEB3BD2FCB0BDB390EF6208F
                      F064ADB29E59B39F5352BA55A7E0164A

CCO Hash        MD5 : 99EF8BE2C9999B7D89A21DF1D115CA39
Digital signature successfully verified in file flash:c800-universalk9-mz.SPA.153-3.M6.bin
```

Abbildung 12: Nutzung verify-Kommando

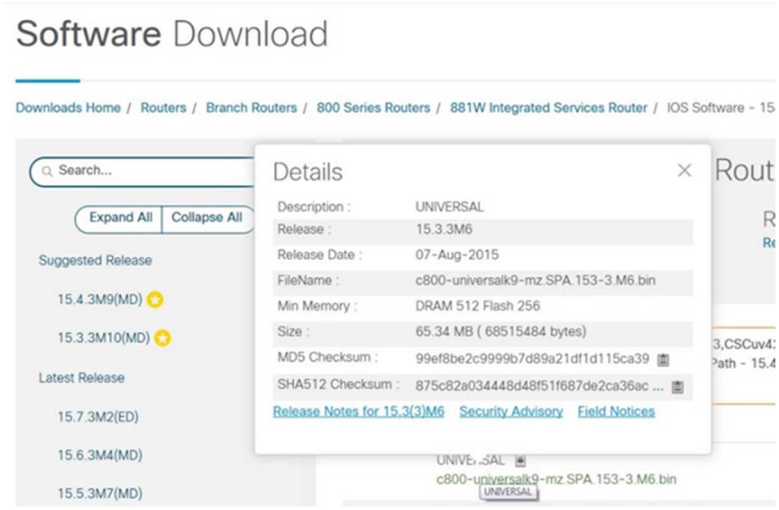

Abbildung 13: auf der Herstellerseite publizierter Hashwert

In diesem Abschnitt wurden die folgenden Kommandos genutzt:

`verify flash:'name der firmware'`

5.3.6 Nutzung signierter Images

Der Begriff signierte Images wurde bereits im Kapitel 2.2 kurz angesprochen. Dabei handelt es sich um solche Images, die von Herstellerseite mittels asymmetrischen public-key Verfahrens signiert werden [26]. Zunächst wird das Zertifikat des Firmwarefiles überprüft.

```
Router#show software authenticity file c800-universalk9-mz.SPA.153-3.M6.bin
File Name                       : c800-universalk9-mz.SPA.153-3.M6.bin
Image type                      : Production
    Signer Information
        Common Name             : CiscoSystems
        Organization Unit       : C8xx
        Organization Name       : CiscoSystems
    Certificate Serial Number   : 55C0B5B7
    Hash Algorithm              : SHA512
    Signature Algorithm         : 2048-bit RSA
    Key Version                 : A
```

Abbildung 14:digitale Signatur der Firmware

Da diese Firmware aber nicht zwangsläufig auf dem Gerät ausgeführt werden muss, wird anschließend auch das Zertifikat der aktuell geladenen Software überprüft.

```
Router#show software authenticity running
SYSTEM IMAGE
------------
Image type                  : Production
    Signer Information
        Common Name         : CiscoSystems
        Organization Unit   : C8xx
        Organization Name   : CiscoSystems
    Certificate Serial Number : 55C0B5B7
    Hash Algorithm          : SHA512
    Signature Algorithm     : 2048-bit RSA
    Key Version             : A
```

Abbildung 15: Signatur des aktuell geladenen Images

In diesem Abschnitt der Arbeit wurden die folgenden Kommandos verwendet:

```
show software authenticity file ’name der firmware’
show software authenticity running
```

Die durchgeführten Analysen können ebenfalls durch entsprechende Skripte unterstützt und teilautomatisiert werden. Im Rahmen ihres Praktikums erstellte die Autorin dazu zwei Skripte [25]. Das erste Skript trägt den Namen serial_new.py und stellt eine serielle Verbindung zu einem Gerät her. Diese Verbindung wird anschließend genutzt, um Kommandos zu dem Gerät zu schicken und so die entsprechenden Einstellungen für einen Core-Dump vorzunehmen. Die dafür benötigten Informationen, wie zum Beispiel die Zugangsdaten für den FTP-Server, werden durch Benutzereingaben vorgegeben. Das vollständige Skript kann in Anlagen, Teil 2 eingesehen werden. Das zweite Skript soll die Berechnung und den Vergleich von Hashwerten für eine Firmware unterstützen. Die Autorin gab diesem Skript den Namen scraper.py. Für eine vom Nutzer übergebene Firmware berechnet das Skript selbstständig sowohl den MD5 als auch den SHA512 Hash und vergleicht diesen mit dem vom Hersteller publizierten Hashwert. Dieser wird dem Skript in Form einer entsprechend aufgebauten Excel-Tabelle zur Verfügung gestellt. Die erste Spalte der Tabelle repräsentiert den Namen des Firmwareimages, in der zweiten Spalte wird der MD5-Hash abgelegt und die dritte Spalte enthält den SHA512-Hash. Das vollständige Skript kann in Teil 3 der Anlagen eingesehen werden.

6 Ergebnisse

Ziel der Arbeit ist es, die Sicherheit von Netzwerkgeräten in den Fokus der Aufmerksamkeit zu rücken und interessierten Nutzern Mittel zur Verfügung zu stellen, um die eigenen Geräte hinsichtlich vorgenommener Veränderungen zu untersuchen. Zu diesem Zweck wurde zuerst ein Vorgehensmodell definiert, welches die Methoden nach dem Grad ihrer Invasivität ordnet. Anschließend wurden die einzelnen Methoden erläutert, bevor sie an konkreten Untersuchungsobjekten erprobt wurden.

Die Ergebnisse zeigen, dass die vorgestellten Methoden genutzt werden können, um Rückschlüsse über eventuell vorhandene Änderungen auf und an dem Untersuchungsobjekt ziehen zu können. Die Recherche von Informationen lieferte nützliche Kenntnisse über den Aufbau und die Funktionsweise des Geräts, welche für die weiteren Untersuchungen genutzt werden konnten.

Durch die Öffnung der Untersuchungsobjekte konnten alle auf der Platine befindlichen Chips identifiziert werden. Zusätzlich wurde jedem Chip seine Funktion zugeordnet. Dabei zeigte sich, dass keine Bauteile mit unbekannter Funktion oder unbekannten Ursprungs auf der Platine vorhanden sind.

Für die Einstellungsüberprüfung konnten zwei Ergebnisse erzielt werden. Es hat sich gezeigt, dass der Hashwert der Einstellungssicherung sich ändert, sobald eine Änderung (in diesem Fall der Netzwerkname) vorgenommen wird. Bleiben die Einstellungen hingegen unverändert, so ist auch keine Änderung des Hashwertes feststellbar gewesen.

Auch die Extraktion und Verifikation der Firmware brachte im Endergebnis übereinstimmende Hashwerte hervor. Zusätzlich stimmten auch die Zertifikatsinformationen der signierten Images überein.

Die methodisch durchgeführte Untersuchung hat gezeigt, dass auf den Untersuchungsobjekten keine Indikatoren für Veränderungen zu erkennen sind. Das Problem der Sicherheit von Netzwerkgeräten wurde umfassend thematisiert und durch entsprechende Praxisbeispiele verdeutlicht. Zusätzlich ist es auch Heimnutzern nach einer Einarbeitung in das Thema möglich, Untersuchungen selbstständig durchzuführen.

7 Diskussion

In diesem Abschnitt werden die Ergebnisse interpretiert, die Grenzen und Limitierungen der Methoden aufgezeigt, ihre Übertragbarkeit auf andere Geräte überprüft, weitere aktuelle Entwicklungen vorgestellt und ein Ausblick gegeben.

7.1 Interpretation der Ergebnisse

Die Recherche von Informationen über das Untersuchungsobjekt offenbarte, dass CVE Details (eine Seite, die Schwachstellen von Hardware, Anwendungen und Betriebssystemen auflistet) für die IOS Version 15.3(3)M6, welche auf dem Gerät ausgeführt wird, ganze 25 Schwachstellen nennt [27]. Zusätzlich konnte herausgefunden werden, dass diese Version seit dem 6.Oktober 2015 nicht mehr verkauft wurde und das im Jahr 2017 zum letzten Mal Patches für Sicherheitslücken und Schwachstellen veröffentlicht wurden [28]. Daraus lässt sich ableiten, dass Nutzer dieser IOS Version ihre Geräte auf eine neue Version updaten sollten beziehungsweise über die Anschaffung neuer Geräte nachdenken sollten. Zwar kann die Software immer noch ausgeführt werden, aber es gibt keine Unterstützung mehr von Herstellerseite. Selbst wenn Schwachstellen bekannt werden sollten, werden keine Patches zu ihrer Behebung mehr entworfen. Es ist möglich, dass Patches für die nachfolgenden Softwareversionen das Problem beheben könnten, aber es kann dadurch auch zu Kompatibilitätsproblemen kommen. Bereits durch die Recherche von Informationen konnte gezeigt werden, dass die auf dem Gerät befindliche Software nicht mehr dem aktuellen Stand entspricht und das Gerät schnellstens auf eine neue Version upgedated werden sollte. Aussagen darüber, wie konkret gefährdet ein Gerät mit dieser Software ist, sind nur schwer möglich, da weitere Faktoren wie die Konfiguration von Diensten und die restliche Netzwerkumgebung Einfluss auf die Sicherheit eines Netzwerks haben. Die gefundenen Informationen verdeutlichen nochmal, warum es wichtig ist, die Sicherheit von Netzwerkgeräten zu adressieren.

Die gefundenen Kratzer an der Außenseite des Geräts sind nicht weiter bedenklich, wenn in Betracht gezogen wird, dass das betroffene Untersuchungsobjekt bereits längere Zeit in Betrieb ist und mehrfach den Standort gewechselt hat. Auf eine supply chain attack fanden sich auch nach der Öffnung des Geräts keine Hinweise. Die gesamte Platine war staubbedeckt, was darauf schließen lässt, dass das Gerät nur selten geöffnet wurde. Die auf der Platine befindlichen Chips konnten alle identifiziert werden. Eine Veränderung erscheint vor diesem Hintergrund als eher unwahrscheinlich. Anzumerken ist hier, dass diese Untersuchung nicht jedes Mal durchgeführt werden muss. Die Autorin empfiehlt aber zumindest die Untersuchung durchzuführen, wenn ein neues Gerät in ein bestehen-

des Netzwerk eingebunden werden soll oder wenn es konkrete Verdachtsmomente gibt, weil zum Beispiel eine Person auffällig lange unbeobachteten Zugang zum Gerät hatte.

Durch die Einstellungsüberprüfung konnte gezeigt werden, dass an den Einstellungen keine Veränderung stattgefunden hat. Dadurch ist klar, dass alle vom Administrator gesetzten Einstellungen unverändert vorliegen. Nicht schlussfolgern lässt sich hingegen, ob diese Einstellungen tatsächlich sinnvoll sind oder sogar selbst mögliche Angriffspunkte enthalten. Die hier durchgeführte Untersuchung stellt eine schnelle Möglichkeit dar, um die Einstellungen auf Änderungen hin zu überprüfen. Sollen die vorgenommenen Einstellungen auf ihre Sinnhaftigkeit beziehungsweise ihren Nutzen hin überprüft werden, so liefert die Methode keine zuverlässigen Ergebnisse. Die Einstellungen müssen stattdessen von Hand analysiert und ausgewertet werden. Diese aufwändige Untersuchung muss ebenfalls nicht immer durchgeführt werden. Selbstverständlich sollte die initiale Konfiguration sorgfältig vorgenommen werden und in regelmäßigen Abständen eine vollständige Untersuchung gemacht werden. Zwischen den vollständigen Untersuchungen reicht es vollkommen, die Hashwerte für die Einstellungssicherungen zu vergleichen.

Die Extraktion der Firmware läuft je nach der gewählten Extraktionsmethode sehr unterschiedlich ab und variiert auch im Schwierigkeitsgrad erheblich. Unerfahrenen Anwendern empfiehlt die Autorin prinzipiell das Kopieren der Firmware aus dem Flashspeicher. Dazu gibt es auf der Seite des Herstellers Anleitungen, welche nahezu analog verwendet werden können. Die Verwendung des Core-Dump gestaltet sich etwas schwieriger, da die entsprechenden Regionen erst aus diesem extrahiert werden müssen. Die Extraktion mit Hilfe von Debugging-Ports oder mit der Chip-off Methode eignet sich nur für erfahrene Anwender, da beide Methoden das Risiko einer Beschädigung der Daten beziehungsweise des gesamten Geräts in sich tragen. Auch wenn im abschließenden Vergleich die Hashwerte übereinstimmen, so bedeutet es lediglich, dass die untersuchte Firmware der des Herstellers entspricht. Mit allen Schwachstellen und Bugs[37], die darin enthalten sind. Es ist also falsch davon auszugehen, dass die Firmware sicher ist, weil sie ja verifiziert wurde. Richtiger ist es zu sagen, dass keine zusätzlichen Veränderungen daran stattgefunden haben und es somit keine zusätzlichen Schwachstellen außer den bereits in der Firmware enthaltenen gibt.

7.2 Limitierungen

Für jede der durchgeführten Untersuchungen existieren limitierende Faktoren, die in diesem Abschnitt verdeutlicht werden.

[37] Bugs sind Fehler in einem Programm oder einer Software.

Die Informationen, die durch eine gründliche Recherche gewonnen werden können, werden natürlich maßgeblich von den verwendeten Quellen beeinflusst. Nicht jeder Medienbericht entspricht zu 100% der Wahrheit. Es obliegt dem Untersuchenden, die Quellen und ihre Seriosität zu beurteilen. Nicht alle idealerweise vorhandenen Informationen sind frei zugänglich. Je nach Hersteller kann es sein, dass sich zum Beispiel keine Baupläne finden lassen. Nicht jede Schwachstelle wird sofort nach ihrem Auffinden öffentlich gemacht. Häufig existieren Vereinbarungen zwischen dem Hersteller und dem Entdecker, in denen sich auf eine festgelegte Sperrfrist geeinigt wird. Erst nach dieser Zeitspanne dürfen die gesammelten Erkenntnisse publiziert werden. Dadurch soll dem Hersteller die Möglichkeit gegeben werden, selbst Patches zur Behebung des Problems zu entwickeln.

Während der optischen Untersuchungen kann es zu Problemen kommen, wenn eine Beschädigung des Geräts unter allen Umständen ausgeschlossen werden muss. Zwar ist es möglich, ein Gerät so zu öffnen, dass nur minimale Spuren zurückbleiben, aber der Garantieanspruch ist damit erloschen. Die genaue Funktion der verwendeten Chips kann nur durch deren Datenblätter abgeleitet werden. Ob es aber versteckte Funktionen gibt, kann daraus nicht geschlussfolgert werden.

Die Einstellungsüberprüfung ist stark davon abhängig, wie der Hersteller den Mechanismus zur Einstellungssicherung implementiert hat. Wie in Kapitel 5.2 beschrieben konnte beim Untersuchungsgerät eine webbasierte GUI genutzt werden. Andere Hersteller realisieren diese Funktion anders oder mitunter auch gar nicht in ihren Geräten. Wenn keine entsprechenden Funktionen implementiert sind, wird das gesamte Vorgehen deutlich verkompliziert. Zu weiteren Problemen kann es kommen, wenn eine Sicherung zwar möglich ist, aber veränderliche Daten ebenfalls in die Sicherung einfließen. Dadurch verändert sich der Hashwert, selbst wenn alle Einstellungen unverändert sind. Ein weiteres sehr großes Problem tritt auf, wenn es keine Einstellungssicherung gibt, die direkt nach der initialen Konfiguration erstellt wurde. Normalerweise wird der Hashwert einer solchen Sicherung zum Vergleich mit dem fraglichen Hashwert genutzt. Existiert dieser Vergleichswert nun nicht, so kann auch die Verifizierung nicht auf diesem Weg erfolgen. Als Möglichkeit zur Verifikation bleibt dann nur die einzelne Auswertung der Einstellungen. Das kostet Zeit und ist fehleranfällig, weil Menschen bei längerer stupider Arbeit häufig Fehler unterlaufen.

Wie in Kapitel 4.4.1 erwähnt, haben das Equipment, die verfügbare Zeit und der Kenntnisstand des Untersuchenden großen Einfluss auf die Ergebnisse der Methode. Sie können als limitierender Faktor auftreten, wenn ein Mangel an einem oder mehreren dieser Einflussfaktoren besteht. Auch hier ist es problematisch, wenn eine Beschädigung am Gerät nicht toleriert werden kann. Dadurch wird der Kreis der durchführbaren Methoden auf eine Methode (Nutzung der Kommandozeile) reduziert. Auch für diese Methode ergibt sich das Problem der Vergleichswerte. Gibt es keinen Hashwert einer Firmware, die bekanntermaßen unverändert ist und stellt der Hersteller keine Hashwerte seiner Firmware bereit, so ist ein Vergleich nicht möglich und die Verifikation kann nicht durchgeführt werden.

7.3 Übertragbarkeit auf andere Untersuchungsobjekte

Wie bereits im Kapitel 5 erwähnt, wurden die Untersuchungen an drei verschiedenen Geräten durchgeführt. Bei den Untersuchungsobjekten handelte es sich um einen Router, einen Switch und einen Integrated Service Router. Zusätzlich wurde eines der Geräte von einem anderen Hersteller (AVM) produziert. Trotzdem ließen sich Untersuchungen durchführen. Daraus kann abgeleitet werden, dass eine Untersuchung von Geräten dieses Typs mit den vorgeschlagenen theoretischen Methoden immer durchgeführt werden kann, wenn eine entsprechende Anpassung an das Gerät erfolgt.

Für Geräte eines anderen Typs ist eine Anwendung der besprochenen Methoden nicht immer möglich. Wenn es nicht möglich ist, eine Verbindung zum Gerät herzustellen, dann können auch die Einstellungen und die Firmware nicht aus dem Gerät extrahiert werden. Ein weiteres Problem ergibt sich, wenn die Firmware und die Einstellungen zwar gesichert werden können, aber ein Einfluss von veränderlichen Daten nicht ausgeschlossen werden kann. Dadurch verändern sich zwischen den einzelnen Sicherungen zwangsläufig die Daten und dadurch stimmen die berechneten Hashwerte nicht überein. Eine Verifikation mit Hilfe der Hashwerte ist dann nicht möglich. Dieser Umstand tritt auch dann ein, wenn keine Vergleichswerte für die berechneten Hashwerte existieren. Häufig publizieren Hersteller diese Hashwerte auf ihren Webseiten. Ist dies nicht der Fall, so sollte die Firmware nach jedem Update kopiert und ihr Hashwert berechnet werden, um später entsprechende Vergleiche anstellen zu können.

Eine analoge Anwendung nach den in Kapitel 5 vorgestellten Beispielen ist nur bei Geräten desselben Typs beziehungsweise mit gleichem Funktionsumfang möglich. Die in Kapitel 4 besprochenen theoretischen Methoden sind jedoch universell und können an ein spezifisches Gerät und seinen Funktionsumfang angepasst werden. Der Aufwand der nötigen Anpassungen hängt vom Untersuchungsobjekt und dessen Eigenschaften ab.

Zusammenfassend kann also gesagt werden, dass eine Anwendung des vorgeschlagenen Untersuchungsmodells und der vorgestellten Methoden auf bisher nicht betrachtete Untersuchungsobjekte möglich ist, sofern eine Anpassung an das konkrete Untersuchungsobjekt, dessen Funktionsumfang und sein Betriebssystem erfolgt.

7.4 aktuelle Entwicklungen

Bereits seit Jahren thematisieren Felix Lindner und sein Team die Sicherheit von Ciscos IOS [10]. Dabei greifen sie auch die Problematik der Firmwareverifikation auf. Um dieses Problem zu adressieren, entwickelten sie das Cisco Incident Response Toolkit (CIR). Mit dem Toolkit sollen Veränderungen an der Firmware entdeckt werden [10].

Eine weitere neue Software, die es sich nach eigener Aussage zum Ziel gemacht hat, Firmen und deren Infrastruktur vor Angriffen gegen Hard – und Firmware zu schützen,

heißt Eclypsium [29]. Die Grundlage für diese Software entstand aus dem Chipsec-Projekt, welches auf GitHub frei verfügbar ist[38] . Dabei handelt es sich um ein sehr mächtiges Werkzeug, welches es ermöglicht, Untersuchungen bereits auf einem sehr niedrigen Abstraktionslevel durchzuführen und die Ergebnisse entsprechend auszuwerten. Der Funktionsumfang von Eclypsium soll nach Aussage der Hersteller im Vergleich zu Chipsec deutlich erweitert sein. Zusätzlich werden auch mehr Geräte, wie zum Beispiel embedded devices eingeschlossen. Die Nutzerfreundlichkeit soll ebenfalls massive Verbesserungen erfahren haben, da der Vorgang Chipsec häufig als schwierig und kompliziert beschrieben wurde. Insbesondere der Eclypsium agent ist für die Verifikation der Firmware von großer Bedeutung. Er soll die Integrität der Firmware überwachen und über jede gefundene Änderung informieren. Wie genau das möglich ist und durchgeführt wird, ist aus dem Whitepaper der Entwickler nicht ersichtlich. Die Autorin konnte sich selbst kein Bild von der Software und den angesprochenen Verbesserung machen, da sie zu diesen keinen Zugang hatte.

Die beiden vorgenannten Beispiele zeigen, dass bereits einzelne Tools existieren, um die Geräte hinsichtlich der Integrität ihrer Firmware zu überwachen. Diese sind jedoch sehr teuer und übersteigen die Kenntnisse des durchschnittlichen Heimanwenders bei weitem, wie in Kapitel 4.2 bereits angesprochen.

7.5 Ausblick

Für weitere Arbeiten zu diesem Thema wäre es möglich, sich stärker auf im Heimbereich genutzte Geräte zu fokussieren. Dadurch würde es Nutzern erleichtert, Untersuchungen am eigenen Gerät durchzuführen, da diese im Heimbereich meist über ähnliche Funktionalitäten verfügen. Zusätzlich wäre das dadurch vermittelte Bild der Hersteller- und Modellvielfalt deutlich realistischer. Die eben genannten Punkte würden dazu beitragen, dass die Forschung an Bedeutung gewinnt und damit auch das Thema der Sicherheit von Netzwerkgeräten stärker diskutiert wird.

Eine weitere mögliche Erweiterung stellt die Aufhebung der Beschränkung auf Netzwerkgeräte dar. Besonders im Heimbereich gibt es eine Vielzahl von Geräten, die unter dem Sammelbegriff IoT-Geräte zusammengefasst werden. Dahinter verbergen sich unter anderem Spielekonsolen, Webcams, Smartwatches und Sprachassistenzsysteme wie Alexa oder Cortana. Auch diese können relevante Daten enthalten oder für die Zwecke des Angreifers missbraucht werden. Das wohl bekannteste Beispiel dafür, das Mirai-Botnetz, wurde bereits in Kapitel 1.1 angesprochen. Gerade diese Geräte haben häufig Schwachstellen und sollten deshalb umso gründlicher überwacht werden. In der Realität ist den meisten Nutzern nicht klar, welche Daten überhaupt auf dem Gerät vorliegen und wie un-

[38] https://github.com/chipsec/chipsec

zureichend diese geschützt werden. Würde dies nun in einer umfangreichen Forschung thematisiert, so würde das Risikobewusstsein des durchschnittlichen Heimanwenders erhöht werden. Dadurch wären erfolgreiche Angriffe deutlich schwieriger durchzuführen als zum momentanen Zeitpunkt.

In Kapitel 7.4 werden kurz die verfügbaren Softwarelösungen beschrieben, für den Rest der Arbeit werden sie jedoch nicht weiter beachtet. Für zukünftige Arbeit und Forschung wäre es also von Interesse zu testen, ob die Software die ihr gestellten Aufgaben zuverlässiger und schneller als ein Mensch erledigen kann. Für die Beurteilung der Software müssten Bewertungsmaßstäbe erarbeitet werden, welche bei der Evaluation zur Anwendung kommen sollen. Wichtige wäre es dabei, auch den Fokus auf den Bereich der Heimanwender zu beachten.

Eine weitere Idee wäre es, für die innere optische Untersuchung ein Hilfsmittel zu entwickeln. Die Autorin dachte dabei an eine Art App, mit der die Platine fotografiert wird. Sofern die Fotoqualität ausreicht, wird dann nach den entsprechenden Chips gesucht. Diese werden identifiziert und dem Nutzer alle verfügbaren Datenblätter und Baupläne angezeigt. Der konkrete Aufwand für die Entwicklung einer solchen Software ist für die Autorin zum momentanen Zeitpunkt nicht abschätzbar. Trotzdem würde es den Prozess der Recherche von Informationen und der optischen Untersuchung effizient kombinieren und dem Untersuchenden so Zeit sparen. Aus diesem Grund entschloss sich die Autorin diese Idee trotzdem im Rahmen des Ausblicks in die Arbeit aufzunehmen.

Momentan werden immer mehr Fälle von gehackten und veränderten Netzwerkgeräten öffentlich. Es ist also durchaus anzunehmen, dass auch in den nächsten Jahren Netzwerkgeräte gehäuft zu Angriffszielen werden. Darüber hinaus ist anzunehmen, dass Heimnetzwerke den größten Teil der angegriffenen Geräte ausmachen werden. Dies liegt darin begründet, dass Firmen sich dieses Problems bereits in Ansätzen bewusst sind und damit beginnen, ihre Infrastruktur entsprechend zu schützen. Im Heimbereich hingegen sind nahezu alle Ziele nicht ausreichend geschützt und es können trotzdem wichtige Daten erlangt werden, wie in 1.2 beschrieben. Als Reaktion darauf wird eine Vielzahl von Software entwickelt werden, deren Ziel es ist, die Sicherheit von Geräten zu überwachen. Unter der entwickelten Software werden sich speziell Lösungen finden, die für eine Anwendung im Heimbereich entwickelt wurden. Dadurch wird es auch für Heimnutzer mit entsprechendem Risikobewusstsein möglich, ihre Geräte mittels einer Software zu überwachen. Es wird zu einer Art Wettlauf zwischen Angreifer, Hersteller und Nutzer kommen, dessen Ausgang nicht vorhergesagt werden kann.

8 Literaturverzeichnis

[1] A. Czernik, „Hub, Switch, Router? – So funktionieren Netzwerke!," intersoft consulting services AG, 7 August 2015. [Online]. Available: https://www.datenschutzbeauftragter-info.de/hub-switch-router-funktionieren-netzwerke/. [Zugriff am 24 Oktober 2019].

[2] M. Antonakakis, „Understanding the Mirai Botnet," in *Proceedings of the 26th USENIX Security Symposium*, Vancouver, BC, Canada, 2017.

[3] N. Woolf, „DDoS attack that disrupted internet was largest of its kind in history, experts say," 26 Oktober 2016. [Online]. Available: https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet. [Zugriff am 7 November 2018].

[4] M. Rouse, „brute force cracking," Juli 2006. [Online]. Available: https://searchsecurity.techtarget.com/definition/brute-force-cracking. [Zugriff am 7 November 2018].

[5] O. von Westernhagen, „IoT-Bot-Netz: Drahtzieher hinter Mirai bekennen sich schuldig," heiseSecurity, 14 Dezember 2017. [Online]. Available: https://www.heise.de/security/meldung/IoT-Bot-Netz-US-Gericht-verurteilt-Drahtzieher-hinter-Mirai-3918203.html. [Zugriff am 24 Oktober 2018].

[6] W. Largent, „New VPNFilter malware targets at least 500K networking devices worldwide," Cisco Talos Intelligence, 23 Mai 2018. [Online]. Available: https://blog.talosintelligence.com/2018/05/VPNFilter.html. [Zugriff am 24 Oktober 2018].

[7] M. Holland, „ Router- und NAS-Botnetz VPNFilter: FBI kapert Kontrollserver," heiseSecurity, 24 Mai 2018. [Online]. Available: https://www.heise.de/security/meldung/Router-und-NAS-Botnetz-VPNFilter-FBI-

kapert-Kontrollserver-4057613.html. [Zugriff am 24 Oktober 2018].

[8] OccupytheWeb, „Hack Like the NSA:The EXTRABACON Zero-Day Exploit on Cisco ASA Firewalls," OccupytheWeb, 15 März 2018. [Online]. Available: https://www.hackers-arise.com/single-post/2016/08/22/The-EXTRABACON-Zero-Day-Exploit-on-Cisco-ASA-Firewalls?platform=hootsuite. [Zugriff am 24 Oktober 2018].

[9] M. J. Schwartz, „Bank Hackers Exploit Outdated Router to Steal $1 Million," Bank Info Security , 20 Juli 2018. [Online]. Available: https://www.bankinfosecurity.com/bank-hackers-exploit-outdated-router-to-steal-1-million-a-11227. [Zugriff am 24 Oktober 2018].

[10] F. Lindner, „Developments in Cisco IOS Forensics," in *DEFCON 16*, Las Vegas, 2008.

[11] Statista, „Market share of enterprise network vendors worldwide from 2015 to 2018," Statista, September 2018. [Online]. Available: https://www.statista.com/statistics/540779/enterprise-network-market-share-by-vendor/. [Zugriff am 1 November 2018].

[12] A. Cui, J. Kataria und S. J. Stolfo, „Killing the Myth of Cisco IOS Diversity: Recent Advances in Reliable Shellcode Design," Columbia University Academic Commons, Columbia, 2011.

[13] S. Muniz, „Killing the myth of Cisco IOS rootkits: DIK (DA IOS Rootkit)," CORE Security , Mai 2008. [Online]. Available: https://drwho.virtadpt.net/images/killing_the_myth_of_cisco_ios_rootkits.pdf. [Zugriff am 1 November 2018].

[14] A. Kondratenko, „Cisco Catalyst Exploitation," in *DEFCON 25*, Las Vegas, 2017.

[15] Federal Communications Commision , „FCC ID Search," Federal Communications Commision , [Online]. Available: https://www.fcc.gov/oet/ea/fccid. [Zugriff am 7 November 2018].

[16] Cisco, „Password Recovery Procedure for the Cisco 806, 826, 827, 828, 831, 836, 837 and 881 Series Routers," 4 Mai 2014. [Online]. Available: https://www.cisco.com/c/en/us/support/docs/routers/800-series-routers/12065-pswdrec-827.html. [Zugriff am 6 November 2018].

[17] I. Sestanj, NAND Flash Data Recovery Cookbook, Belgrade: Selbstverlag, 2016.

[18] P. Schmitz, „Was ist ein Hash?," 23 August 2017. [Online]. Available: https://www.security-insider.de/was-ist-ein-hash-a-635712/. [Zugriff am 12 November 2018].

[19] S. Akbari, An Implementation of SYNful Knock Attack in Cisco Router and Firewall Devices, Tallinn: Tallinn University of Technology, 2017.

[20] A. Costin, A. Zarras und A. Francillon, „Automated dynamic firmware analysis at scale: A case study on embedded web interfaces," in *11th ACM Asia Conference on Computer and Communications Security* , Xi'an , 2016.

[21] Y. Shoshitaishvili, R. Wang, C. Salls, N. Stephens, M. Polino, A. Dutcher, J. Grosen, S. Feng, C. Hauser, C. Kruegel und G. Vigna, „(State of) The Art of War: Offensive Techniques in Binary Analysis," in *Proc of the IEEE Symposium on Security and Privacy* , San Jose, 2016.

[22] D. L. IT, „Cisco 2960G Switch teardown an detailed overview for CCNA lab and repair -Part1&2," 2 Februar 2016. [Online]. Available: https://www.youtube.com/watch?v=HJLqF7Gg5WQ. [Zugriff am 14 November 2018].

[23] AVM, „Benutzeroberfläche der FRITZ!Box aufrufen," [Online]. Available: https://avm.de/service/fritzbox/fritzbox-7390/wissensdatenbank/publication/show/1_Benutzeroberflaeche-der-FRITZ-Box-aufrufen/. [Zugriff am 15 November 2018].

[24] AVM, „Einstellungen sichern und in derselben FRITZ!Box wiederherstellen," [Online]. Available: https://avm.de/service/fritzbox/fritzbox-7390/wissensdatenbank/publication/show/4_Einstellungen-sichern-und-in-derselben-FRITZ-Box-wiederherstellen/. [Zugriff am 15 November 2018].

[25] L. Neumann, Extraktion und Verifikation von Fimrware am Beispiel Cisco IOS, Frankfurt, 2018.

[26] Cisco, „Cisco IOS Software Integrity Assurance," Cisco, 7 September 2018. [Online]. Available: https://tools.cisco.com/security/center/resources/integrity_assurance.html#12. [Zugriff am 12 November 2018].

[27] MITRE Corporation, „Cisco IOS 15.3(3)M6: Security Vulnerabilities," MITRE Corporation, [Online]. Available: https://www.cvedetails.com/vulnerability-list/vendor_id-16/product_id-19/version_id-195553/Cisco-IOS-15.3-3-m6.html. [Zugriff am 23 November 2018].

[28] Cisco , „End-of-Sale and End-of-Life Announcement for the Cisco IOS Software Release 15.3(3)M," Cisco, 7 April 2015. [Online]. Available: https://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ios-15-3m-t/eos-eol-notice-c51-734340.html. [Zugriff am 23 November 2018].

[29] Ecylpsium , The enterprise hardware attack surface and how to defend it, Portland: Selbstverlag , 2018.

[30] K. Geiser, „Patch, Update, Upgrade," 20 November 2014. [Online]. Available: https://www.aspectra.ch/de/wissen/blog/patch-update-upgrade--b512. [Zugriff am 8 November 2018].

[31] M. Rouse, „firmware," April 2017. [Online]. Available: https://whatis.techtarget.com/definition/firmware. [Zugriff am 8 November 2018].

[32] M. Rouse, „endpoint device," Juli 2013. [Online]. Available: https://whatis.techtarget.com/definition/endpoint-device. [Zugriff am 8 Dezember 2018].

Anlagen

Anlagen, Teil 1

Hashwerte und Namen der Einstellungssicherung aus Kapitel 5.2

Hash für Datei 1 FRITZ.Box Fon WLAN 7170 (UI) 29.04.87_01.01.00_0108.export:

59cd9c4562f89b477efcb14e8ba39953

Hash für Datei 2 FRITZ.Box Fon WLAN 7170 (UI) 29.04.87_01.01.00_0102.export

ca8d9fd491aacaa705d416e2b2801466

---> in Versuch 1, Netzwerkname geändert, deshalb nicht matchender Hash

Hash für FRITZ.Box Fon WLAN 7170 (UI) 29.04.87_01.01.00_0102.export
fe95b95f614cb41adca4ae7f31b3a029

Hash für FRITZ.Box Fon WLAN 7170 (UI) 29.04.87_01.01.00_0103.export
fe95b95f614cb41adca4ae7f31b3a029

---> Versuch 2, ohne Passwort und keine geänderte Einstellung... Matchender Has

Anlagen, Teil 2

entnommen aus dem Praxisbericht der Autorin [25]

```
"""
Created on Mon Jun 11 12:19:48 2018
@author: neumlen

A script for automating the connection between a computer and a Cisco
device.
It opens a connection with user specified details and then simulates en-
tering commands to the command line interface of the devices.
These commands force the device to write a core dump. The scripts ends
with leaving the priv exec mode.
"""
import serial
import sys
import time


def main():
      print("Wich   port   shall   be   used   (Name,   like   COM1   or
/dev/ttyUSB0)?")
      interface=raw_input()
      print("Configure to wich baudrate?")
      b=raw_input()
      print("Parity, please answer Y or N")
      pa=raw_input()
      print("How many stopbits?")
      s=raw_input()
      print("What is the bytesize?")
      by=raw_input()
      print("What is the timeout?")
      READ_TIMEOUT=raw_input()

      print (b"\nInitializing serial connection")
      console = serial.Serial(
            port=str(interface),
            baudrate=int(b),
            parity=str(pa),
            stopbits=int(s),
            bytesize=int(by),
            timeout=int(READ_TIMEOUT)
      )                     # specifies the serial connections details

      console.write("\r\n")  #simulate 'enter' pressed
      console.inWaiting()
      input_data=console.read(255)
```

```
print("Login")

console.write("no"+b"\r\n")              #simulate 'no' entered
console.inWaiting()
input_data=console.read(255)
print("no initial configuration")

console.write("enable"+b"\n")            # CLI 'enable' entered
time.sleep(1)
input_data =console.read(console.inWaiting())

print("priv exec")
console.write("config t"+b"\n")          #configuration terminal
time.sleep(1)
input_data =console.read(console.inWaiting())
print("config mode")

print
print("What username for ftp?")
name=raw_input()
console.write(b"ip ftp username "+str(name)+b"\n")
time.sleep(1)
input_data=console.read(console.inWaiting())
print('entered ftp username')            # configure FTP username
print
print("Password for ftp?")
pw=raw_input()
console.write(b"ip ftp password "+str(pw)+b"\n")
time.sleep(1)
input_data=console.read(console.inWaiting())
print('entered ftp password')            # configure FTP password
console.write(b"exception protocol ftp "+b"\n")
time.sleep(1)
input_data=console.read(console.inWaiting())
print('entered exception protocol')      # exception protocol

print
print("IP to write the core to?")
ip=raw_input()
console.write(b"exception dump "+str(ip)+b"\n")
time.sleep(1)
input_data=console.read(console.inWaiting())
print('entered ip address')  #ip address for writing the core dump

console.write(b"int vlan1 "+b"\n")
time.sleep(1)
input_data=console.read(console.inWaiting())
print('config vlan1')        #configure interface vlan1

print
print("Current Netmask?")
netmask=raw_input()
print("Gateway IP?")
```

```
	ip_gateway=raw_input()
	console.write(b"ip add "+str(ip_gateway)+str(netmask)+b"\n")
	time.sleep(1)
	input_data=console.read(console.inWaiting())
	print('adding ip')           #adding ip to vlan1

	console.write(b"exit"+b"\n")
	time.sleep(1)
	input_data=console.read(console.inWaiting())
	print('exit vlan1')          # leaving interface configuration

	console.write(b"exit"+b"\n")
	time.sleep(1)
	input_data=console.read(console.inWaiting())
	print('exit config mode')    #leaving configuration mode

	console.write("write core"+b"\n")
	time.sleep(1)
	input_data =console.read(console.inWaiting())
	print("started write core")    #starting core dump process

	console.write("\n")
	time.sleep(1)
	input_data =console.read(console.inWaiting())
	print ("FTP-Folder?")
	dest_filename=raw_input()
	console.write(str(dest_filename)+b"\n")
	time.sleep(1)
	input_data =console.read(console.inWaiting())
	print input_data             #core dump to the specified folder

	console.write("exit"+"\n")
	time.sleep(1)
	input_data =console.read(console.inWaiting())
	print("exit priv exec")      # leaving priv exec


if __name__=="__main__":
	main()
```

Anlagen, Teil 3

entnommen aus dem Praxisbericht der Autorin [25]

```
"""
Created on Tue Jun 12 10:34:06 2018
@author: neumlen
This script was built to compute a MD5 and SHA512 hash of a file and
compare
it to the known hashes of Cisco Images provided as Excel Spreadsheet.
"""
import openpyxl
import hashlib


print(" absolute Path to the Firmwareimage?")
image=raw_input()
print("Exact name of the firmware (if possible)?")
imagename=raw_input()
print("path to the excel-sheet?")
excel=raw_input()

wb = openpyxl.load_workbook(str(excel))        #open a excel workbook
wb.get_sheet_names()                           # what sheets are in the
                                               workbook
sheet=wb.get_sheet_by_name('Sheet1')           # open sheet1
lines=0                                        #  variable  for  counting
                                               output lines
for row in range (2, sheet.max_row+1):         #iteration  for  all  rows
                                               starting with 2nd
      iname=sheet['A' + str(row)].value
      known_hash_md5=sheet['B' + str(row)].value
      known_hash_sha512=sheet['C' + str(row)].value
      if str(imagename) in iname:                    # if names match
            lines=lines +1                           # increment lines
            hashermd5 = hashlib.md5()                # create hasher MD5
            with open (str(image), 'rb') as afile:   #for the file
                  buf = afile.read()
                  hashermd5.update(buf)
                  h = hashermd5.hexdigest()
            print
            print
            print (iname)
            print('computed md5 hash '+h)
            print
            print('known md5 hash '+known_hash_md5)
```

```
            if (str(h)==str(known_hash_md5)):
                  print
                  print('The MD5 hashes match!')
                  print
            else:
                  print
                  print("MD5 hashes don't match")
                  print
            print("".rjust(50, '*'))                  #structure output

            hashersha512 = hashlib.sha512()     #create SHA512 hasher
            with open (str(image), 'rb') as bfile:   #for the file
                  buf1 = bfile.read()
                  hashersha512.update(buf1)
                  s = hashersha512.hexdigest()
            print
            print('computed sha512 hash '+s)
            print
            print('known sha512 hash '+ known_hash_sha512)
            if (str(s)==(known_hash_sha512)):
                  print
                  print('The SHA512 hashes match!')
                  print
            else:
                  print
                  print("SHA512 hashes don't match")
                  print
            print("".rjust(50, '-'))
print(str(lines) +" matche(s) found")
if lines <1:                                          #no  matching  lines
                  print ("Image not in database")     # not in DB
```

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, den 07.Januar 2019

Lena Neumann