**Inhaltsverzeichnis**

TODO:

Wifi / Wlan / DNS / DUT, Gerät, Router → einheitlich

Grafiken alle einheitlich

# Abbildungsverzeichnis

# Tabellenverzeichnis

# Abkürzungsverzeichnis

|  |  |
| --- | --- |
| Breitband-Internetzugang | Ein Begriff für verschiedene, moderne Internetzugangstechnologien. Es handelt sich hierbei um einen Nachfolger zu Telefonmodem. |
| Community WLAN | Ein WLAN, welches von einer größeren Gruppe Nutzern verwendet wird, welche dem Anwender unbekannt sind. Dieses WLAN ist logisch vom privaten WLAN separiert. |
| DUT | Device under Test bezeichnet den Router, wenn an diesem Tests im Rahmen der TR 03148 durchgeführt werden. |
| End-User | Der hauptsächliche Benutzer der Router Funktionen |
| Firewall |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Abstract

# Zusammenfassung

**Kapitel 1**

**Einleitung**

* 1. **Ausgangslage**

Das Internet wird ein zunehmend wichtigerer Teil des menschlichen Lebens. Öffentliche Hotspots, Internet fähige Alltags-Geräte (IOT-Geräte) und mobiles Arbeiten von Zuhause sind nur einige Beispiele für technologische Neuerungen, welche ohne das Internet nicht möglich wären. Die rund 35,5 Mio. Netzanbindung an DSL-, Kabel-, oder Glasfaser-Anschlüsse in Deutschland [SOURCE] werden in Heimnetzen und Kleinunternehmen überwiegend durch Netzwerkrouter realisiert. In vielen Fällen bildet der Router die direkte Schnittstelle zwischen dem Internet und dem privaten Netzwerk. So stellt dieser meist auch die einzige zentrale Sicherheitskomponente zum Schutz des Netzwerkes bereit. Ein erfolgreicher Angriff auf den Router bietet einem Angreifer unzählige Möglichkeiten in das Netz einzugreifen und so immensen Schaden anzurichten. Neben bekannten Zielen wie private Daten und Passwörtern kann der Router auch als Teil eines Bot-Netzwerks für Distributed Denial-of-Service (DDoS) verwendet werden [SOURCE] oder als Einfallstor auf weitere Geräte des Netzwerkes [SOURCE]. Die Korrelation mit stark steigenden Fällen von Cyberkriminalität [SOURCE LAGEBERICHT] zeigt wie wichtig ein inhärent geschützter Router mit sicherer Konfiguration ist.   
 Handelsübliche Router wie sie in Privathaushalten und Small Office, Home Office (SoHo) Umgebungen eingesetzt werden sind bereits mit einem proprietären Betriebssystem bespielt. Die Sicherheit dieser Distribution kann also nur mit großem Aufwand von Endnutzern verifiziert werden, sowie Sicherheitsupdates nur vom Hersteller veröffentlicht werden. Hersteller können in der zunehmend kürzer werdenden Zeit zwischen neuen Iterationen von Malware meist nicht in einer angemessenen Zeit reagieren, um Sicherheitsupdates zur Verfügung zu stellen. Quelloffene Router Firmware wie OpenWrt, DD-Wrt, Tomato oder LibreCMC bieten eine Alternative zu den vorinstallierten, proprietären Betriebssystemen der Router. Diese Projekte können vollständig eingesehen, modifiziert und kompiliert werden, sodass die Sicherheit des Produktes einfach evaluiert werden kann. Ebenfalls können aufgrund der hohen Zahl an Mitwirkenden Sicherheits- und Funktionsupdates schneller entwickelt und veröffentlicht werden. Umfangreiche Überprüfungen dieser Projekte, wie z.B. anhand der BSI TR-03148: Sichere Broadband Router, werden allerdings aufgrund des hohen Zeit- bzw. Kosten- Aufwands selten durchgeführt, sodass diese auch eine Zertifizierung nicht erlangen können. Eine solche Zertifizierung könnte ungeschulten Endnutzern auch diese quelloffenen Router-Betriebssysteme als Alternativen näherbringen und somit zu einem höheren Sicherheitsniveau in privater und SOHO Netzwerkinfrastruktur führen.

* 1. **Was ist OpenWrt?**

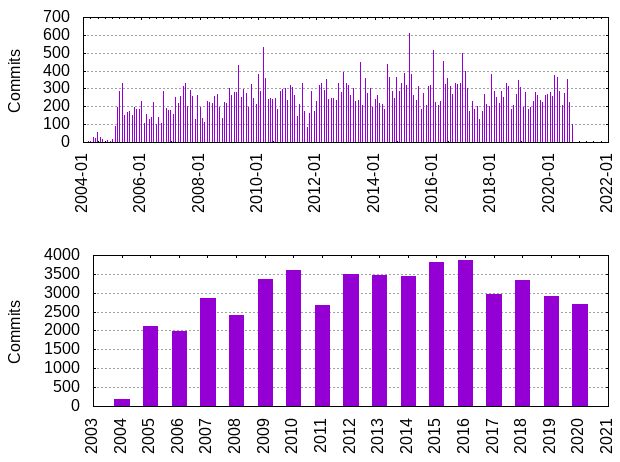
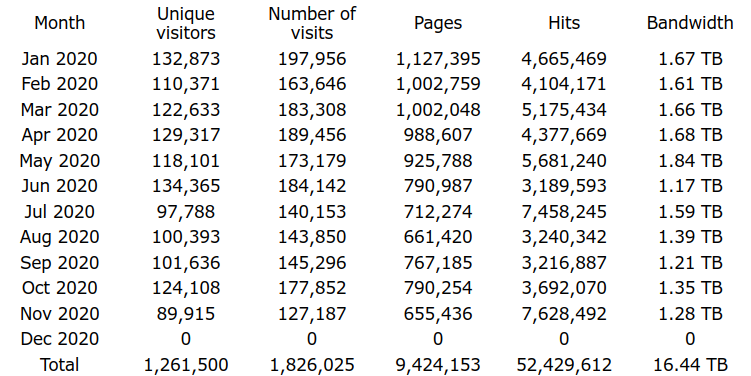
OpenWrt (**Open** **W**ireless **R**ou**T**er) ist ein quelloffenes Netzwerk-Betriebssystem für Router, welches auf GNU/Linux basiert und durch eine GNU General Public License (GPL) lizensiert ist. Die Installation umfasst einen bootloader, kernel, ein eigenes Dateisystem und ausgewählte Anwendungen. Es kann auf Routern, Switches und Accesspoints eingesetzt werden, um die vorinstallierte Firmware zu ersetzen. Es bietet neben standardmäßiger Router Funktionalität einen eigenen Paketmanager, über welchen ca. 3800 (Stand 01.11.20) weitere Pakete installiert werden können [Source]. Dies bietet viele weitere Einsatzmöglichkeiten und Funktionen, welche vom Hersteller nicht oder unzureichend unterstützt werden. Ebenfalls wird OpenWrt mit BusyBox, einem SSH Dienst, und Luci, einem Web-Interface, ausgeliefert, sodass der Nutzer vollständigen Zugriff auf das Gerät hat. Nach derzeitigem Stand werden über 1700 Geräte von ca. 270 Herstellern von OpenWrt unterstützt [Source]. Diese Anzahl Geräte kann unter anderem deshalb unterstützt werden, da OpenWrt nur minimale Ressourcen auf dem Endgerät benötigt. Nach eigenen Angaben kann die derzeitige Version, mit Einschränkungen, auf Geräten installiert werden, welche 4MB Flash Speicher und 32MB RAM besitzen. Ab der nächsten Version werden 8MB Flash und 64MB RAM vorausgesetzt [Source]. Diese Vorrausetzungen sind jedoch bei den meisten Geräten der letzten Jahre gegeben [Source].  
OpenWrt zeichnet sich ebenfalls dadurch aus, dass es sich nicht nur um eine statische Firmware handelt, sondern ebenfalls um ein komplettes Framework um angepasste Firmware Versionen zu erstellen. Ebenfalls zeichnet sich OpenWrt dadurch aus, dass Geräte solange unterstützt werden, wie sie diese Grundanforderungen erfüllen. Dies steht im Gegensatz zu den meisten proprietären Betriebssystemen, welche nur einige Jahre lang Funktions- und Sicherheitsupdates erhalten und nach ihrem sog. „End of Life“ (EOL) nicht mehr sicher betrieben werden können und ausgetauscht werden müssen. Auch wenn in der Entwicklungsgeschichte von OpenWrt viel für die Benutzerfreundlichkeit des Betriebssystems getan wurde ist es jedoch nicht für Laien geeignet. Auch wenn das Management rein über die Weboberfläche geschehen kann, erweist sich die Einrichtung ohne Grundkenntnisse als schwierig.   
 Die Entwicklung von OpenWrt begann 2004, nachdem der amerikanische Hersteller Linksys zuvor einen Router auf den Markt brachte, dessen Firmware ebenfalls unter der GPL Lizenz stand und somit öffentlich verfügbar sein musste. Die erste Veröffentlichung von OpenWrt erfolgte im Januar 2006 mit Version 0.9 (White Russian). Seitdem wurde das Projekt stetig weiterentwickelt (siehe Abbildung 1). 2016 spaltete sich eine Gruppe Mitwirkender aufgrund interner Diskrepanzen ab und gründete das LEDE Projekt. Jedoch wurde LEDE bereits 2018 wieder in OpenWrt integriert, sodass beide Projekte nun wieder zusammen unter einem Namen entwickelt werden. Die derzeit aktuelle Version ist 19.07.4, welche am 10.09.2020 veröffentlicht wurde.

Abbildung : Git commits pro Monat und pro Jahr. Die Datenreihe beginnt am 28.03.2004 und endet am 25.10.2020

* 1. **Relevanz und Verwendung von OpenWrt**

Die Webseite des OpenWrt Projektes verzeichnete im Jahre 2020 bis einschließlich November 1.261.500 einzigartige Besucher, sowie 52,4 Millionen Seitenaufrufe. Insgesamt wurden bereits 16,44TB Daten abgerufen [Source]. Die aktuelle Version von OpenWrt, 19.07.4, wurde dabei alleine im November 1981 Mal heruntergeladen. Ebenfalls wurde Version 18.06.8 noch 935 Mal angefragt. Zusammen wurden ca. 10000 Firmware-Abbilder im November heruntergeladen [Anhang]. Wie die Daten zeigen ist OpenWrt keinesfalls ein kleines Projekt mit nur wenigen Interessierten, sondern eine nachgefragt Alternative für Heimrouter, Unternehmen und Entwickler. Es lässt sich nur schwer abschätzen wie die Verteilung zwischen dem privaten und wirtschaftlichen Einsatz der Firmware ist, jedoch ist eine mehrheitliche Nutzung im privaten Umfeld zu vermuten. OpenWrt ist nicht nur für Heimrouter relevant, sondern zeichnet sich auch in seinem Nutzen für Unternehmen und Entwickler aus. Es bietet Unternehmen die Möglichkeit ein Netz zu betreiben, welches sie vollständig mit quelloffener Software realisieren und steuern können. Ebenfalls bietet es Dienstleisterunternehmen einen Weg hochgradig maßgeschneiderte Netzstrukturen für ihre Kunden zu entwerfen, welche quelloffen und leicht anpassbar sind. So können neue oder geänderten Funktionen über ein Paket bereitgestellt und verteilt werden.

* 1. **Beschreibung der BSI TR-03148**

Bei der Technischen Richtlinie „Sichere Breitband Router“ (BSI TR-03148) des Bundesamtes für Sicherheit in der Informationstechnik handelt es sich um eine Sammlung von grundlegenden Sicherheitsanforderungen für Breitband Router. Der Schwerpunkt der Richtlinie liegt hierbei auf Heimroutern, sowie auf Geräten, welche im sogenannten SOHO (Small Office, Home Office) Umfeld eingesetzt werden. Das Dokument wird durch die Dokumente „BSI TR-03148 Implementation Conformance Statement (ICS)“ sowie „BSI-TR-03148-P ICS and Test Documentation” ergänzt. In diesen Dokumenten sind Testfälle und Dokumentation zur Durchführung einer Prüfung festgehalten. Die Technische Richtlinie definiert 101 Test Requirements, welche 164 Test Procedures beschreiben. Ein Test Requirement wird als fehlgeschlagen gewertet, wenn ein zugehöriges Test Procedure nicht bestanden wird. Nach Angaben des Bundesamtes für Sicherheit in der Informationstechnik richtet sich die Technische Richtlinie vor Allem an Hersteller von Routern, sie kann jedoch auch für Endnutzer relevant sein, wenn diese einen neuen Router anschaffen möchten und sich im Zuge dessen über den Stand der Technik informieren wollen [SOURCE]. Es werden Anforderungen für ein Mindestmaß an verpflichtenden und einigen optionalen IT-Sicherheitsmaßnahmen definiert, um ein grundlegendes Niveau für die Sicherheit dieser Geräte zu schaffen [Source].   
 Das Dokument entstand aus einer Zusammenarbeit des BSIs mit verschiedenen deutschen Herstellern von Heimroutern, Wirtschaftsvertretern, sowie mit Vertretern des OpenWrt Projektes und dem Chaos Computer Club. Diese trugen ihre Ideen und Vorstellungen zur Sicherheit von Routern zusammen und suchten Lösungen für Interessenkonflikte. Nach Veröffentlichung der Richtlinie im Jahre 2018 wurde diese allerdings unter Anderem von Vertretern des OpenWrt Projektes, sowie vom Chaos Computer Club (CCC), kritisiert. Nach Meinung dieser Interessengruppe sind die definierten Maßnahmen in der Technischen Richtlinie nicht ausreichend, um tatsächliche Angriffe auf Router zu verhindern.

* 1. **Bisherige Forschung**

Während der Einsatz von OpenWrt für spezialisierte Netzwerkumgebungen und zur Vereinheitlichung unterschiedlicher Netzwerkprotokolle beliebt zu sein scheint sind derzeit keine aktuellen Arbeiten zur Sicherheit von OpenWrt verfügbar. Ortega et al. veröffentlichte 2009 eine Arbeit über eine quelloffene Methode zum Verhindern von sogenannten ARP Poisoning Attacken. Sie nutzen in diesem Kontext OpenWrt lediglich als vielseitig unterstützte Testplattform [source]. Palazzi et al. nutzen den Funktionsumfang und die Anpassbarkeit der Firmware, um einen verbesserten Datendurchsatz in Heimnetzen mit verschiedenen W-Lan Geräten zu erreichen. Keine der derzeitigen Veröffentlichungen beschäftigt sich mit der Sicherheit von OpenWrt als Betriebssystem. Einzig Andrew McDonnell veröffentlichte in seinem Blog 2014 zwei Einträge über eine Sicherheitsanalyse von OpenWrt mittels des Tools checksec.sh[source] und entwarf eine verbesserte Version, in welcher bedeutend mehr Härtungsmaßnahmen aktiviert waren [source]. McDonnells Ergebnisse basierten jedoch auf Version 14.07 (Barrier Breaker) von OpenWrt, welche stark veraltet ist.   
 Die Forschung an Komponenten die OpenWrt ausmachen ist jedoch keinesfalls so eingeschränkt wie zuvor aufgezeigt. Der Linux Kernel, welcher einen grundlegenden Teil des OpenWrt Betriebssystems ausmacht, ist seit seiner Veröffentlichung 1991 ein andauerndes Gebiet der Forschung und Entwicklung, so auch in der IT-Sicherheit. Ebenso definiert sich OpenWrt über seine ca. 3800 zusätzlichen quelloffenen Pakete. Viele dieser Software-Erweiterungen existieren schon seit Jahrzehnten und ihre Integrität und Vertraulichkeit sind von den unzähligen Nutzern auf verschiedensten Plattformen anerkannt. Abschließend kann man feststellen, dass es zwar durchaus Forschung an Komponenten von OpenWrt gibt, jedoch OpenWrt selbst noch nicht oft mit Mittelpunkt der Forschung stand und die Sicherheitslage weitestgehend ungeklärt bleibt.

* 1. **Zielsetzung**

Ziel dieser Arbeit ist es die aktuelle Version von OpenWrt (19.7.04) anhand der BSI TR-03148 zu analysieren. Hierbei soll ein handelsüblicher, moderner Heimrouter, welcher vermehrt von OpenWrt Nutzern eingesetzt wird, genutzt werden. Es sollen die grundsätzlichen Sicherheitsmerkmale von OpenWrt mittels der technischen Richtlinie evaluiert werden. Ebenso soll die Anwendbarkeit der technischen Richtlinie auf quelloffene Netzwerk-Betriebssysteme ermessen werden. In einem weiteren Schritt werden die Ergebnisse der Untersuchung im Kontext anderer quelloffenen und proprietären Router-Betriebssysteme betrachtet. Darüber hinaus sollen statische Software Tests aller betrachteten Betriebssysteme als weitere Metrik dienen und einen differenzierteren Einblick in die Sicherheitslage gewähren. Abschließend muss sich kritisch mit den Ergebnissen, sowie der technischen Richtlinie, auseinandergesetzt werden. Die Ergebnisse der Arbeit können sowohl der Entwicklung von OpenWrt als auch unerfahrenen Endnutzern weitere Einblicke in die Sicherheit des Projektes liefern und somit langfristig die Resilienz der Heim- und SoHo Netzinfrastruktur stärken.

Kapitel 2

Grundlagen

2.1 Something

Kapitel 3

Methodik

3.1 Übersicht und Begründung der verwendeten Methodik

Die Methodik der Arbeit ist in großen Teilen durch die Technische Richtlinie vorgegeben. Auf eine genaue Durchführung wurde großer Wert gelegt. Die Testfälle wurden aufgrund ihrer Gruppierung in thematische Module in chronologischer Reihenfolge erarbeitet. Einzig solche Testfälle, welche spezifizierten, dass sie erst zum Ende der Testphase durchgeführt werden sollten, wurden nach Hinten gestellt. Da es in erster Linie um die Technische Richtlinie 03148 gehen soll, wurden weitere Tests, wie ein statischer Test mit dem Tool „FACT“[SOURCE], erst nach Vollendung der Richtlinie begonnen.   
 Die Testfälle der Technischen Richtlinie wurden, soweit möglich, mit den Programmen durchgeführt, welche in der TR selbst spezifiziert wurden. Die aufgeführte Software ist für die Überprüfung der Testanforderungen geeignet, sowie die Ergebnisse derselben seit vielen Jahren weitestgehend als korrekt akzeptiert. Hierzu zählt vor Allem das Programm nmap, welches aufgrund von verschiedenen Testrechnern in den Versionen 7.80, 7.90 und 7.91 verwendet wurde. Die Änderungshistorie von nmap gibt allerdings keinen Anlass zur Annahme, dass dies die Ergebnisse invalidiert [SOURCE]. Ebenso wurde airmon-ng / airodump-ng zum Prüfen verwendet. Diese Softwarepacket ist ebenfalls seit vielen Jahren angesehen. Zur Aufzeichnung von Netzwerkpaketen wurde Wireshark verwendet, welches neben der Kommandozeilenanwendung tcpdump häufig Verwendung findet. Im Rahmen der Tests wurde des Weiteren auf einige zweckspezifische Skripte in der Programmiersprache Python zurückgegriffen. Bei der Entwicklung wurde Wert auf einfache Ausführbarkeit, sowie eine geringe Zahl an externen Abhängigkeiten, gelegt, um eine wiederholbare Ausführbarkeit auch in der Zukunft zu gewährleisten.

3.2 Aufbau und Beschreibung der Testumgebung

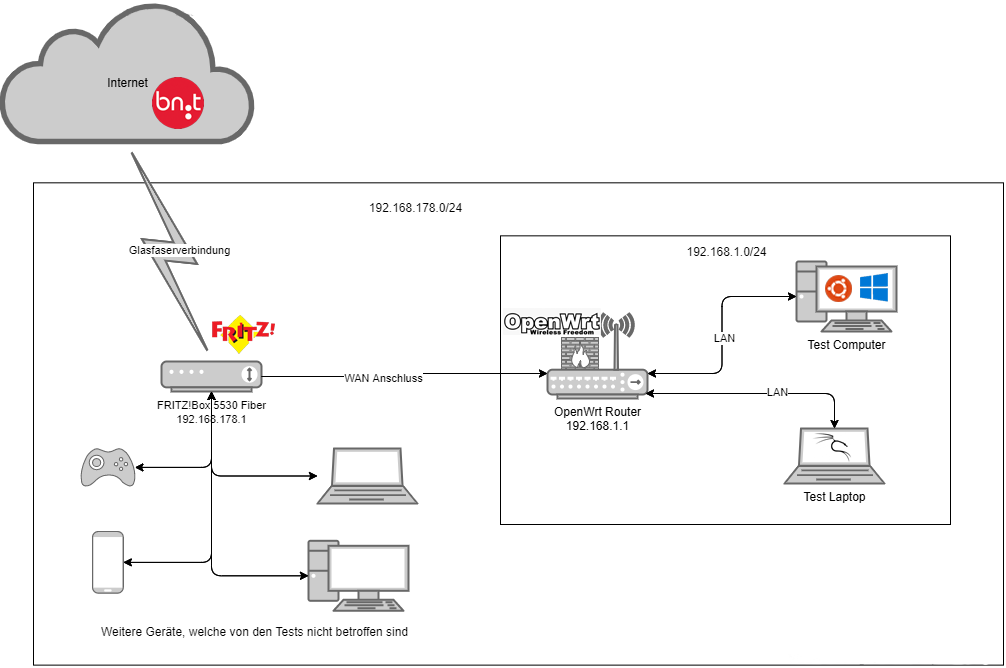
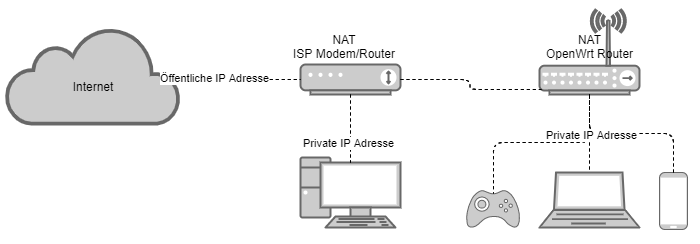
Der genutzte Testaufbau soll einen reibungslosen Ablauf der Testfälle erlauben, sowie einfach reproduzierbar sein. Der Internetanschluss wurde durch den Internet Service Provider (ISP) bn:t Blatzheim Networks Telecom GmbH zur Verfügung gestellt. Der Glasfaseranschluss des ISP mündete in eine FRITZ!Box 5530 Fiber, welche das Subnetz 192.168.178.0/24 bereitstellt. Der WAN Port des mit OpenWrt 19.7.04 bespielten Heimrouters, ein TP-Link Archer C7 v.5, wurde mit dieser FRITZ!Box verbunden, sodass der OpenWrt fähige Router das Subnetz 192.168.1.0/24 aufspannen konnte. Die Erstinstallation von OpenWrt auf dem TP-Link Router erfolgte über die zur Verfügung stehende Anleitung [SOURCE]. Zunächst wurde das Firmware-Abbild heruntergeladen, daraufhin wurden die Hashwerte mit den veröffentlichten und signierten Hash Werten abgeglichen. Nachdem sichergestellt wurde, dass diese übereinstimmten konnte die Datei über das Web-Interface des TP-Link Routers aufgespielt werden. Die Datei wird hierzu über die Firmware-Update Funktion hochgeladen und automatisch vom Gerät installiert. Das Gerät startet daraufhin persistent mit OpenWrt anstelle des Betriebssystems von TP-Link. Alternativ besteht die Möglichkeit das Firmware-Abbild von OpenWrt über die „Trivial File Transfer Protocol“ (TFTP) Funktionalität des Routers aufzuspielen.   
 Ein Testcomputer wurde über LAN angeschlossen, ein weiterer Laptop per WLAN verbunden (siehe Abbildung 2). Der Testcomputer wurde wahlweise mit Windows 10 Version 20H2 (Build 19042.685) oder Ubuntu 20.04 LTS betrieben. Auf dem Laptop kam Kali Linux zum Einsatz. Dieser Aufbau gibt dem Tester eine flexible Arbeitsumgebung, in welcher die Tests ungestört durchgeführt werden können. Durch die automatische Abtrennung des Netzes in das 192.168.1.0/24 Subnetz durch den OpenWrt Router sind Geräte des allgemeinen Heimnetzes von Portscans und Netzwerkpaketmitschnitten ausgeschlossen wodurch Tests performanter durchgeführt werden können, während andere Teilnehmer des Netzes ungestört weiterarbeiten können. Ebenso bietet der beschriebene Aufbau einfach die Möglichkeit weitere Netzteilnehmer oder Geräte hinzuzufügen. Die verwendeten Linux-Distributionen, Ubuntu 20.4 LTS und Kali Linux, bieten dabei die notwendigen Programme und Möglichkeiten zur Durchführung der Testfälle. Dieser sogenannte „double NAT“ (Network Address Translation) [SOURCE] Aufbau stellt praktisch keinen Nachteil dar. Obwohl der direkte Anschluss des OpenWrt-fähigen Routers präferiert eingesetzt werden sollte, können alle Tests ohne Integritätsverlust durchgeführt werden. Die Tests bezüglich des WAN Anschlusses können über die IP-Adresse des Routers durchgeführt werden, welche durch die FRITZ!Box vergeben wurde. Weiterhin wurde der DNS Resolver der FRITZ!Box auf die IP-Adresse des OpenWrt Routers geändert, ebenso wurden alle verfügbaren Firewall und Filter Einstellungen deaktiviert.

Abbildung 2: Aufbau der Testumgebung mit Einteilung in Subnetze

Abbildung 3: Beispiel einer allgemeinen "double NAT" Umgebung. Basiert auf: https://kb.netgear.com/30186/What-is-Double-NAT [abgerufen: 13.12.2020]

3.3 Durchführung der Testfälle

3.3.1 Conformance Statement

Bevor die eigentlichen Tests, welche in der Technischen Richtlinie 03148: Sichere Broadband Router beschrieben sind, durchgeführt werden können, muss zunächst ein sogenanntes „Implementation Conformance Statement“ (ICS) ausgefüllt werden. In diesem werden maßgebende Informationen über das zu testende Gerät festgehalten. Bei einer Durchführung der Technischen Richtlinie im Kontext einer Zertifizierung würde dieses Conformance Statement zunächst vom Hersteller bzw. Auftraggeber ausgefüllt und eingereicht. Die angegebenen Informationen unterstützen den Tester, sind aber auch selbst Teil der Testprozedur. Zu diesen Informationen gehören neben dem Namen und der betrachteten Software Version auch eine Übersicht über die zur Verfügung stehende Dokumentation des Gerätes. Hierzu wird auch technische Dokumentation gezählt, welche normalerweise nicht für Endnutzer und Verbraucher zur Verfügung steht. Des Weiteren werden relevante Informationen zu allen Modulen zusammengetragen, welche bei der Durchführung der Tests von Relevanz sind. So werden zum Beispiel für Modul A – Privates Netzwerk alle Dienste gesammelt, welche im privaten Netz zur Verfügung stehen, sowie die dazugehörigen Interfaces und Ports. Im Falle dieser Arbeit wurde das Conformance Statement als Teil der Richtlinie betrachtet und ordnungsgemäß mit den in der Dokumentation von OpenWrt beschriebenen Informationen ausgefüllt. Darüber hinaus konnte der Quellcode Aufschluss über in der Dokumentation ungeklärte Fragestellungen geben. OpenWrt bietet eine vergleichsweise geringe Anzahl an Diensten im Ausgangs- sowie initialisierten Zustand an. Lediglich der Web-Server uHTTPd auf Port 80, der SSH Server auf Port 22 und der von dnsmasq zur Verfügung gestellte DNS-Dienst auf Port 53 liegen vor. Funktionen wie das Session Initiation Protocol (SIP) für Voice over IP-Telefonie oder Protokolle zur externen, automatischen Konfiguration des Geräts, welche oft bei handelsüblichen Routern verwendet werden, fehlen vollends. Ebenso kann ohne die Installation von zusätzlicher Software nicht das veraltete und als unsicher geltende Wi-Fi Protected Setup (WPS) Verfahren zur Verbindung von Geräten mit dem Router verwendet werden. Dies ist auf vielen aktuellen Geräten in den Standardeinstellungen aktiviert [SOURCE]. Aus dieser eingeschränkten Menge an Diensten wird ersichtlich, dass das Gerät nur über die Netz-Schnittstelle oder per ssh eingerichtet und bedient werden kann. Jedoch steht dem Nutzer standardmäßig der sogenannte „root“ Benutzer zur Verfügung, sodass uneingeschränkter Zugriff auf alle Funktionen und Einstellungen des Gerätes gewährleistet ist. Eine weitere Besonderheit zeigt sich in auch in der Vorkonfiguration des W-Lan Netzes von OpenWrt. Dies ist zunächst deaktiviert und wird standardmäßig ohne Password initialisiert. Dies ist jedoch leicht damit zu begründen, dass OpenWrt nicht mit gerätespezifischer Dokumentation ausgeliefert werden kann, wie sonst üblich. Ein Schriftstück mit einzigartigem Passwort für das Gerät, sowie das voreingestellte W-Lan, kann nicht erstellt werden. So muss jedes Passwort, welches für ein OpenWrt Gerät verwendet wird vom Benutzer selbst erstellt werden. Dies kann sowohl positive als auch negative Implikationen für die Sicherheit des Gerätes haben.   
 Schon im zweiten Abschnitt des Conformance Statements, welcher sich auf das öffentliche Netz bezieht, wird erkenntlich, dass auch auf Seiten des Internets nur eine minimale Anzahl an Diensten verwendet wird. Die Dokumentation von OpenWrt enthält keinen Dienst, welcher nach außen angeboten wird. Ein vergleichbarer Trend kann auch bei den angebotenen Funktionen des Geräts beobachtet werden. Lediglich sehr grundlegende Funktionen wie das Dynamic Host Configuration Protocol (DHCP), ssh, secure copy (scp), IPv6 Unterstützung und eine Firewall werden angeboten. Lediglich die eigens für OpenWrt entwickelte, quelloffene Packet-Management Software „opkg“, über welche zusätzliche Funktionalität installiert werden kann, sticht heraus. Der geringe Umfang an Funktionen lässt sich in zweierlei Hinsicht begründen. Durch den Packet Manager opkg kann gewünschte Funktionalität leicht vom Benutzer selbst installiert und eingerichtet werden, ohne schon im Vorhinein Speicherplatz für Funktionen zu nutzen, welche unter Umständen nicht verwendet werden. Darüber hinaus kann OpenWrt so auch auf Geräten mit limitieren persistenten Speicher oder Arbeitsspeicher installiert werden. So kann selbst das Web-Interface von der Installation ausgeschlossen sein, wenn ein Gerät nicht über genügend Speicher verfügt. Dadurch ist eine minimale Installation auf Geräten mit 4MB Flashspeicher und 32MB RAM möglich, jedoch lediglich bis einschließlich Version 19.07.   
 Ein Defizit von OpenWrt lässt sich jedoch bereits im Comformance Statement finden. Es besteht keine Möglichkeit sicherheitsrelevante Updates automatisch einzuspielen. Über den Paket Manager bereitgestellte Funktionen könnten zwar mittels CronJobs aktualisiert werden, dies würde jedoch nur periodisch nach Einstellung des Nutzers geschehen. Dies bietet keine Sicherheit, wenn die Periode zu groß gewählt wurde. Sicherheitslücken im Linux Kernel können jedoch nur über vollständige Firmware-Upgrades behoben werden und erfordern das aktive Eingreifen des Nutzers. Dies setzt das Engagement und fachliche Verständnis des Nutzers voraus, über den aktuellen Stand informiert zu bleiben und das Upgrade zeitnah durchzuführen. Gleichermaßen ist die Überprüfung des Firmware-Upgrades, bzw. des aufzuspielenden Abbildes von OpenWrt, auf Integrität und Authentizität nicht vollständig automatisiert. Für einige Abbilder stehen digitale Signaturen zur Verfügung, welche vom integrierten Tool fwtool beim Aufspielen des Updates geprüft werden, jedoch steht diese Option nicht immer zur Verfügung. So auch im Falle der für diese Arbeit verwendeten Firmware. Zur Unterstützung des Nutzers beim Upgrade-Prozess stehen dann lediglich die eingebetteten Metadaten bereit, welche ausschließlich sicherstellen, dass es sich überhaupt um ein unterstütztes Gerät handelt, und die berechneten Hash-Werte, welche durch den Benutzer mit den signierten Werten des Download-Servers abgeglichen werden können.   
 Die folgenden Module des Conformance Statements zeigen gleichwohl eine weitere Besonderheit von OpenWrt. Die für Firewall, DNS und DHCP verwendete Implementierung ist vollständig quelloffen und schon seit vielen Jahren verfügbar. Die Firewall wird durch ein für OpenWrt gestaltetes Programm firewall3 bereitgestellt. Es handelt sich hier um eine einfache Möglichkeit netfilter/iptable Regeln zu gestalten. Iptables sowie ip6tables sind Bestandteil des Kernels und schon seit Version 2.4 mitgeliefert wird [Source]. Der DHCP und DNS-Dienst wird von dnsmasq ermöglicht. Dies ist ebenfalls ein weitverbreitetes Programm, welches bereits 2001 veröffentlicht wurde und seitdem kontinuierlich weiterentwickelt wurde. Da OpenWrt keine Fernwartungs-, VoIP- oder Virtual Private Network (VPN) Funktionalität bereitstellt, ohne die entsprechenden Pakete über den Paketmanager zu installieren, werden diese im weiteren Verlauf nicht betrachtet und dieses Ergebnis im Conformance Statement vermerkt.

3.3.2 Test Dokumentation

Die Testdokumentation wurde in Form der bereitgestellten Tabellenkalkulationsdatei „20200430\_BSI-TR-03148-P\_ICS\_and\_Test\_Documentation\_for\_v1.1.xlsx“ ausgefüllt. Die Anforderungen mit Kriterien zum Bestehen des Testes finden sich im Dokument „20200430\_BSI\_TR-03148-P\_Router-TR\_Test\_Specification\_v1.1.pdf“. Die Testdokumentation definiert die folgenden Kategorien: Eine durchlaufende Nummerierung und eine Angabe, ob es ein „muss“ oder „soll“ Kriterium ist, eine Beschriebung des Testfalls, die Angabe des Testers, ob der Testfall anwendbar ist, oder nicht. Ebenso steht „N/A“ (not applicable) als Option zur Verfügung. Darauf folgen Felder für die jeweiligen Ergebnisse der Test einer jeden Testreihe, gefolgt von der Möglichkeit für Notizen, Referenzen, benutze Tools, Zugriffsmethoden und einer Referenz für weitere Daten, wie Bilder.   
 Die in der Richtlinie spezifizierten Zustände des DUT wurden vor Beginn der Test wie folgt festgelegt: Das Gerät ist im Auslieferungszustand (factory state), wenn er initial in Betrieb genommen wurde und nach jedem vollständigen Zurücksetzen. Der erste Start nach einem solchen Zurücksetzen des Geräts versetzt dieses in den Auslieferungszustand. Der initialisierte Zustand (initialized state) ist erreicht, wenn das Gerät im Auslieferungszustand gestartet und ein Passwort für den Benutzer vergeben wurde. Dies ist vom Nutzer selbst vorzunehmen und nicht verpflichtend. Für alle Testfälle, die den initialisierten Zustand oder den kundenspezifischen (customized state) Zustand vorrausetzen, wurde diese Aktion vom Tester durchgeführt. Das Gerät befindet sich im kundenspezifischen Zustand, wenn zusätzliche Einstellungen vom Nutzer aktiviert oder angepasst wurden.

3.3.2.1 Modul A – Private Network

Wie in TP.A.1 nachgewiesen, unterstützt die betrachtete Version von OpenWrt zwei Arten das Gerät in Betrieb zu nehmen. Zum einen den ssh Zugang, zum anderen den Web-Server, welches das Web-Interface „luci“ bereitstellt. Zur Prüfung des verlangten vollständigen Internetzugangs im initialisierten Zustand wurde zur Überprüfung der DNS-Funktionalität das bei Windows 10 standardmäßig installierte Kommandozeilenprogram nslookup verwendet. Der FTP-Funktionsumfang wurde ebenfalls mittels des Kommandozeilenprogramms getestet. Hierzu wurde der FTP-Downloadserver von DD-WRT genutzt [ftp.dd-wrt.com], da dieser ohne Passwort genutzt werden kann. HTTP, sowie HTTPS Unterstützung konnten mittels des Programms „curl“ nachgewiesen werden. Hierbei handelt es sich um ein quelloffenes Programm, welches neben http und https viele verschiedene Protokolle unterstützt und zur Übertragung von Daten über diese Protokolle gedacht ist. Das „Simple Mail Transfer Protocol“ (SMTP) kann ebenfalls mit Hilfe von curl getestet werden. Die geforderte IPv4 und IPv6 Konnektivität kann ebenfalls trivial mit den Kommandozeilenapplikationen ping bzw. ping6 geprüft werden. Zur Sicherstellung der SSH Verbindung kann zum Beispiel der öffentliche Server ssh.sdf.org genutzt werden. Ein eigens bereitgestellter SSH-Server kommt ebenfalls in Frage. Das Telnet Protokoll muss unter Windows zunächst aktiviert werden, es steht jedoch auch auf vielen Linux Distributionen zur Verfügung. Ein Test kann über die URL „towel.blinkenlights.nl“ durchgeführt werden. Die verwendeten Programme stehen unter den meisten aktuellen Betriebssystemen standardmäßig zur Verfügung und die spezifizierten Server sind weltweilt kostenlos zu erreichen. Ebenfalls kann angenommen werden, dass die angegebenen URLs längerfristig zu erreichen sind.   
 Ein wichtiger Aspekt der Technischen Richtlinie wird ebenfalls durch TR.A.2 bis TR.A.5 spezifiziert. Diese Test Requirements behandeln die durch das Gerät zur Verfügung gestellten Dienste. Es wird vorausgesetzt, dass die angebotenen Dienste durch den Hersteller dokumentiert sind und ebenfalls, dass es sich um eine volldefinierte, minimale Menge handelt. Die Überprüfung kann mit Hilfe des Tools nmap durchgeführt werden. Nmap ist ein quelloffenes Port-Scanning Programm, welches ursprünglich von Gordon Lyon entwickelt wurde [SOURCE]. Es wird genutzt, um offene Ports und die darauf lauschenden Dienste zu identifizieren. Die TCP Ports des DUT wurden mit dem Kommando

nmap -sS -sC -sV -p- -Pn -oN <Dateiname.txt> 192.168.1.1

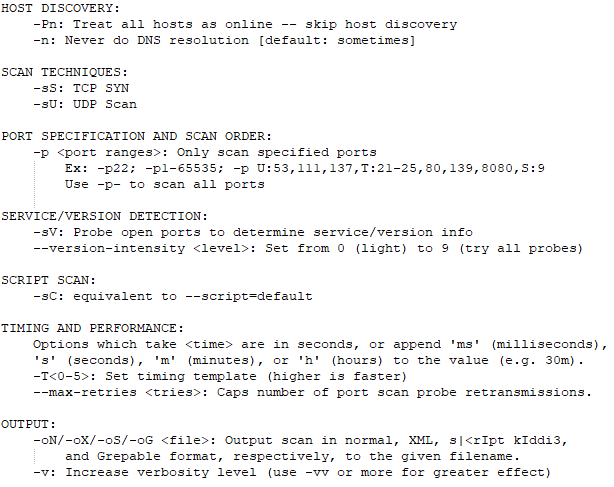
oder verkürzt

nmap -sSCV -p- -Pn -oN <Dateiname.txt> 192.168.1.1

überprüft. Ebenfalls kann der Schalter „-T4“ hinzugefügt werden, um die Geschwindigkeit zu erhöhen. UDP Dienste wurden mittels des Kommandos

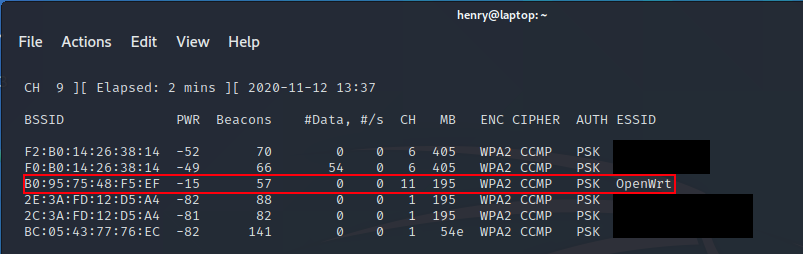
nmap -n -sUV --version-intensity 0 -p- --max-retries 1 -v -oN <Dateiname.txt> 192.168.1.1

Die optionale Erweiterung „-v“ erhöht die Verbosität und liefert bei den zeitintensiven UDP-Scans Informationen über den Fortschrittsgrad. Eine genaue Übersicht über die Funktion der gewählten Kommandos liefert Abbildung 4. Die beiden verwendeten Kommandos, bzw. leichte Abwandlungen von diesen, wurden vor allem aufgrund ihrer detaillierten Ausgabe, sowie Performanz gewählt. Aufgrund unterschiedlicher Testcomputer wurde für einige Test Prozeduren Version 7.91 des nmap Tools verwendet, für andere Version 7.8. Das Änderungsprotokoll der Versionen 7.90 und 7.91 von nmap, welche seit Version 7.8 veröffentlicht wurden, gibt jedoch keinen Anlass zur Annahme, dass dies die Ergebnisse invalidiert.   
 Zur Prüfung der W-Lan Schnittstelle wurde auf die Programmsuite aircrack-ng zurückgegriffen. Es handelt sich hierbei um eine frei verfügbare Sammlung von Programmen zur Analyse der Sicherheit von Wi-Fi Netzwerken [SOURCE]. Zunächst muss das Programm airmon-ng, um die W-Lan Karte in den sogenannten Monitor-Modus zu versetzen.

airmon-ng start wlan0

Daraufhin kann airodump-ng verwendet werden, um Informationen zu allen verfügbaren W-Lan Netzen bereitzustellen.

airodump-ng wlan0mon

Vor allem die Spalte „ENC“, welche für encyrption steht, ist interessant. Sie zeigt an, dass das Gerät durch Wi-Fi Protected Access 2 (WPA2) geschützt ist. Dies unterstützt die Annahme, dass das Gerät WPA2 nach dem IEEE802.11i Standard bereitstellt. 

3.3.2.2 Modul B – Public Network

Die Teststrategie, welche für Modul B – Public Network eingesetzt wurde, ist nahe an Modul A – Private Network orientiert. Jedoch wird nun die IP des OpenWrt Geräts im Kontext des übergeordneten Netzes 192.168.178.0/24 verwendet. So wird nicht die Local Area Network (LAN) Schnittstelle des Gerätes angesprochen, sondern die Wide Area Network (WAN) Schnittstelle, also die öffentliche IP.

nmap -sSCV -p- -Pn -oN <Dateiname.txt> 192.168.178.115

nmap -n -sUV --version-intensity 0 -p- --max-retries 1 -v -oN <Dateiname.txt> 192.168.178.115

Auch die VoIP Funktionalität kann effektiv mit nmap getestet werden. Zusätzlich zu den vollständigen Scans des Geräts können auch die standardmäßig für VoIP verwendeten Ports 5060 und 5061 separat gescannt werden.

nmap -sSCV -p 5060,5061 -Pn -oN <Dateiname.txt> 192.168.178.115

nmap -n -sUV --version-intensity 0 -p 5060,5061 --max-retries 1 -v -oN <Dateiname.txt> 192.168.178.115

Jedoch ist eine vollständige Prüfung aller Ports durchaus zu bevorzugen, da diese Ports nicht zwingend genutzt werden müssen.

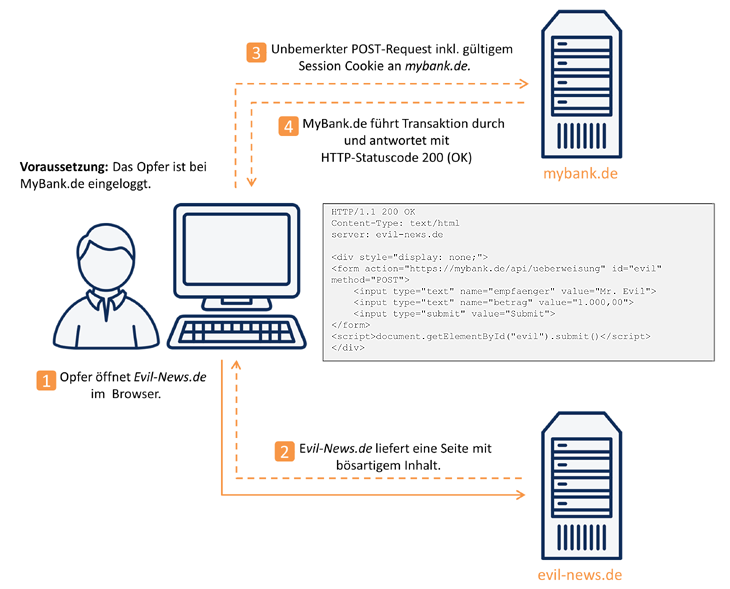
3.3.2.3 Modul C - Functionalities

Das Test Requirement TR.C.2 beschreibt die Anforderung, dass dem Endnutzer keine Funktionalität verheimlicht werden darf. Dies ist eine durchaus schwierig zu prüfende Anforderung, welche erst zum Ende des Tests durchgeführt werden sollte. Im Falle von OpenWrt und dem somit vollständig verfügbaren Quellcode, sowie dem vollumfänglichen root Zugriff auf das Gerät per ssh ist dies vereinfacht, jedoch aufgrund des Funktionsumfangs immer noch eine Herausforderung. Es muss sich hier auf die Eindrücke und Erfahrungen des Testers zum Ende der Testphase verlassen werden.

3.3.2.4 Modul D – Configuration and Information

Für die meisten modernen Heimrouter ist die Konfiguration durch ein Web-Interface die prominenteste Methode, so auch für OpenWrt. Die Sicherung der Datenintegrität und Vertraulichkeit auf dem Transportweg wird durch HTTPS erreicht. Diese Transportwegverschlüsselung verhindert, dass eine böswillige Dritte Partei die übertragenen Daten auslesen oder verändern kann. Es ist also keine Überraschung die Anforderung an eine durch HTTPS gesicherte Verbindung zum Webserver in der Technischen Richtlinie zu finden. Zur Überprüfung des Test Requirements TR.D.3 bietet sich ein Skript wie testssl.sh an, welches von Dr. Wetter IT-Consulting frei zur Verfügung gestellt wird [SOURCE]. Dieses Skript zeigt detaillierte Informationen zu allen vom Webserver unterstützten Protokollversionen sowie Verschlüsselungsmethoden. Des Weiteren kann auch ein Netzwerkpacketsniffer wie Wireshark eingesetzt werden, um die unverschlüsselten Pakete zu betrachten. Wenn HTTPS aktiv ist, so sollten keine menschenlesbaren Daten in den Pakten gefunden werden. Zu Letzt ist es ebenfalls möglich Informationen zu HTTPS und dem dazugehörigen Zertifikat in den meisten modernen Browsern in der Nähe der URL-Leiste zu finden.   
 Nichtsdestoweniger müssen auch andere Angriffsvektoren auf Heimrouter betrachtet bzw. getestet werden. So muss der Log-In auf dem Gerät gegen Bruteforce Angriffe geschützt sein [ERKLÄRUNG]. Eine mögliche Schutzmaßnahme kann ein Fehlerzähler sein, welcher die fehlgeschlagenen Versuche protokolliert und das Aufschalten auf das Gerät nach einer gewissen Anzahl Versuche unterbindet oder entschleunigt. Ebenso könnte die Eingabe auf Muster geprüft werden, um automatische Login-Versuche zu erkennen. Die Prüfung dieses Test Requirements wurde durch ein Skript in der Programmiersprache Python umgesetzt. Durch den Aufruf python3 OpenWrt\_Bruteforce\_Check.py web wird der Web-Server getestet. Alternativ kann durch python3 OpenWrt\_Bruteforce\_Check.py ssh der SSH Server getestet werden. Vor der Nutzung können der korrekte Benutzername, sowie das korrekte Password, die Anzahl der Versuche, die IP des Geräts, sowie der SSH Port festgelegt werden. Für den Test des SSH Servers wurden 40 Versuche eingestellt, wobei die Zeit für die Antwort des Servers gemessen wird. Das Python Modul „SSHLibrary“ wird genutzt, um die Verbindungen mit dem SSH Server zu handhaben. Zunächst wird geprüft, ob der spezifizierte Server erreichbar ist. Daraufhin werden die spezifizierten Login Versuche durchgeführt und die Zeit bis zu Antwort des Servers gemessen. Dies ist bei falschen Daten der Abbruch der Session durch eine SSHLibrary Exception. Nachdem die Daten gesammelt wurden wird eine lineare Regression auf den Daten durchgeführt, um einen Trend in den Antwortzeiten zu erkennen. Wenn ein linearer Anstieg zu erkennen ist, dann werden die Versuche verlangsamt, wenn die Regressionslinie jedoch zur X-Achse parallel ist, so werden die Versuche in konstanter Zeit durchgeführt. Neben der grafischen Darstellung der Antwortzeiten, sowie der Regressionslinie, werden dem Nutzer der Mittelwert, der Median, der Regressionskoeffizient und der Standardfehler angezeigt. Nachdem die Analyse durchgeführt wurde werden die korrekten Login Daten verwendet, um eine arbeitende Verbindung herzustellen. Wenn das OpenWrt SSH-Banner korrekt angezeigt wird, lässt der SSH-Server trotz der vorherigen fehlgeschlagenen Versuche noch weitere zu, ohne erkennbare Entschleunigung. Der Test des Webservers wurde durch die POST Anfrage

http://192.168.1.1/cgibin/luci/admin/status?luci\_username={USERNAME}&luci\_password={PASSWORD}

realisiert. Wenn ein falscher Benutzername, oder ein falsches Passwort verwendet wird, so antwortet der Webserver mit dem Statuscode 403 [SOURCE]. Nach der ersten Überprüfung der Verbindung wurden 100 Versuche eingestellt. Der weitere Ablauf der Analyse verläuft wie bereits beschrieben. Nach der Auswertung der Daten werden die korrekten Login-Daten an den Server geschickt. Ein einfacher Regex-Ausdruck überprüft, ob ein erfolgreicher Login möglich war und es wird dem Benutzer anschließend angeboten eine eingeloggte Session im Browser zu öffnen [ANHANG].   
 Neben dem Bruteforce Angriff auf den Webserver ist auch Cross-Site-Request-Forgery (CSRF) ein üblicher Angriffsvektor. Wenn keine adäquaten Schutzmaßnahmen vom Server getroffen werden, kann ein Angreifer über eine präparierte Website oder einen Phishing Link, schädlichen Code auf Seiten eines authentifizierten Nutzers ausführen. Dieser Code versetzt den Angreifer in die Lage Befehle auf der Webseite, oder dem Webserver, auszuführen, auf welchem der Nutzer angemeldet ist. Es könnte zum Beispiel ein neuer Benutzer durch den Angreifer angelegt werden, oder Einstellungen und Sicherheitsparameter an den Angreifer gesendet werden [siehe Grafik]. Eine häufig verwendete Sicherheitsmaßnahme gegen CSRF Angriffe ist ein Anti-CSRF Cookie. Dieser wird im http-Header der Website deklariert und besteht aus einer zufälligen Zeichenkette. Dieser Cookie wird für jede http-Methode benötigt, welche nach dem setzen des Cookies aufgerufen wird, und vom Server validiert. Zur Überprüfung der Anforderung TR.D.12 wird zunächst festgestellt, ob es einen Anti-CSRF Cookie gibt. Zunächst kann der Speicher des Webbrowsers angezeigt werden, um zu prüfen, ob überhaupt ein Cookie eingesetzt wird. Daraufhin wird die Web-Proxy Funktionalität von Burp Suite genutzt, um den Ablauf des Logins und der Erstellung einer gültigen Session zu beobachten. Alle nachfolgenden http-Methoden sollten nach Initialisierung des Cookies diesen als Sicherheitsmerkmal mitversenden. Der Quellcode von OpenWrt gibt darüber hinaus weiteren Aufschluss über die Implementierung der Anti-CSRF Tokens. Die Datei „dispatcher.lua“ des Luci Interfaces, welche die Erstellung und Validierung der Benutzersessions handhabt, zeigt in diesem Falle eindeutig, dass es sich um Anti-CSRF Cookies handelt und das diese durch den als sicher anerkannten Zufallszahlengenerator /dev/urandom [SOURCE] generiert werden. Abschließend wurde ein einfaches Python Skript verwendet, welches 100 gültige Sitzungen am Web Server des OpenWrt Routers anmeldet und mittels eines Regex-Ausdruckes den Wert des Cookies ausließt. Dazu wird das Request Modul von Python verwendet, sowie die POST-Anfrage, welche bereits für das Bruteforce-Skript verwendet wurde. Abschließend wird geprüft, ob die 100 verschiedenen Sitzungen einzigartige SessionIDs und Anti-CSRF Token besitzen.

3.3.2.5 Modul E – Firmware Updates

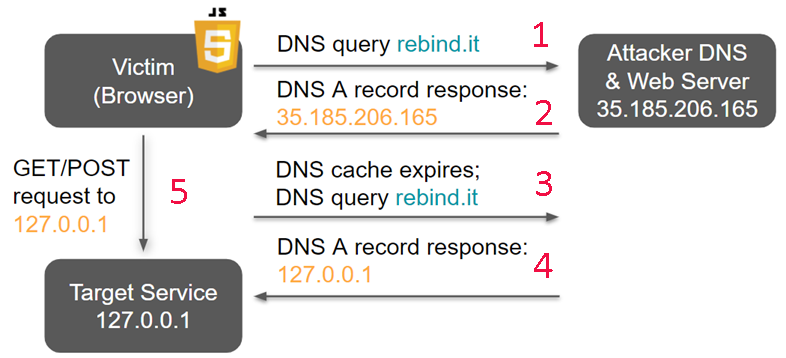
Modul E der Technischen Richtlinie prüft die Firmware Update Funktion des Geräts. Hier ist vor allem der Mechanismus der Firmware-Validierung von Interesse. Nach Angaben der Entwickler werden einige Firmware Dateien signiert. OpenWrt liefert standardmäßig ein Kommandozeilenprogramm mit welchem Signaturen und Metadaten aus den Firmwareabbildern extrahiert werden können. Der Aufruf   
$ fwtool -s - <Dateiname.bin>   
zeigt die Signatur an, wenn diese vorhanden ist. Ebenso muss ermessen werden, wie lange der Hersteller benötigt, um Sicherheitslücken zu beheben. Die sogenannten „Git Hashes“, genaue Identifizierungsmerkmale eines git commits sind hier förderlich, da sie einen genauen Zeitstempel tragen. Des Weiteren ist der entsprechende git commit, welcher eine Sicherheitslücke behebt, in den Sicherheitsnotizen auf der OpenWrt Website spezifiziert, sodass das Erstellen einer Zeitleiste mit Sicherheitsvorfällen und deren Beheben einfach realisierbar ist.

3.3.2.6 Modul G – Domain Name System (DNS)

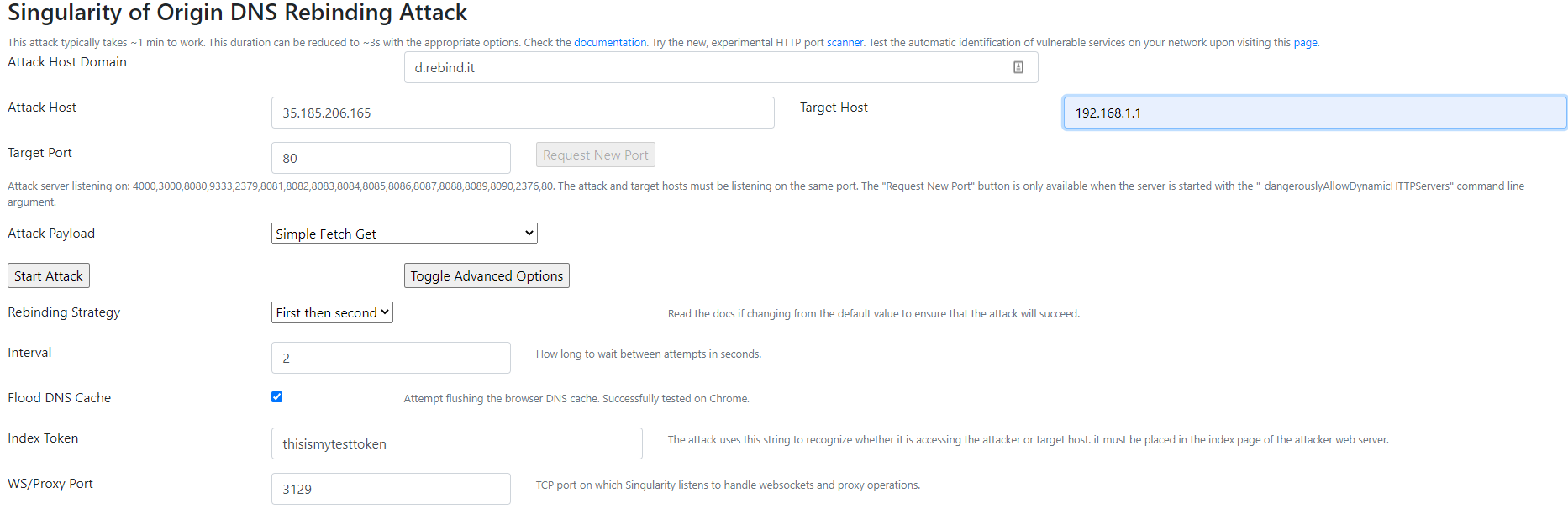
Zur weiteren Einschränkung der Angriffsoberfläche wird in Modul G die Implementierung des DNS-Dienstes des DUT geprüft. Ein Angriff auf DNS-Dienste ist eine sogenannte DNS Rebinding Attacke. Bei dieser Art von Angriff wird die vom Browser durchgesetzte „Same Origin Policy“ ausgehebelt, um arbiträre Anfragen an das lokale Netzwerk des Opfers zu stellen. Die Herkunft („Origin“) eines Web Dokumentes ist dabei wie folgt definiert:



Zwei Dokumente haben also die gleiche Herkunft („same origin“), wenn sie identische „scheme“, „host“ und „port“ Komponenten haben. Die „Same Origin Policy“ setzt also durch, dass Skripte, oder auch Cascading Style Sheets (CSS), nur auf Daten von anderen Webseiten zugreifen können, wenn diese sich dieselbe Herkunft teilen. Wenn diese Richtlinie nicht implementiert wäre, dann wäre eine bösartige Webseite zum Beispiel in der Lage auf ein Bankkonto zuzugreifen, auf dem ein Opfer ebenfalls eingeloggt ist. Dort könnten Daten ausgelesen, oder Aktionen ausgeführt werden.   
 Bei einem DNS Rebinding Angriff ruft das Opfer zunächst eine kompromittiere, oder bösartige, Website auf. Für diesen Aufruf wird ein DNS-Server beauftragt mit der IP-Adresse des angefragten Web-Servers zu antworten. Der vom Angreifer kontrollierte DNS-Server antwortet mit einem DNS A Record, welcher auf die Angreifer-Webseite verweist und den Browser des Opfers anweist, die DNS-Daten nur für eine geringe Zeit im Cache zu behalten. Ein Skript, welches auf der Webseite des Angreifers platziert wurde, wartet nun darauf, dass die DNS-Daten aus dem Cache verfallen, sodass der Browser eine neue Anfrage stellen muss. Diesmal antwortet der DNS-Server allerdings nicht mit der eigenen IP-Adresse, sondern mit einer IP-Adresse des lokalen Netzwerks des Opfers. Nun kann das Skript Anfragen an den lokalen Dienst stellen, z.B. Daten exfiltrieren, oder weitere Angriffe starten [siehe Grafik].



Die Überprüfung der Anforderung TR.G.2 basiert auf der Untersuchung der verwendeten Methoden zur Mitigation von DNS Rebinding Attacken und einem funktionalen Test dieser Umsetzung. Da OpenWrt DNS-Dienste mittels dnsmasq anbietet muss geprüft werden, ob die Option „--stop-dns-rebind“ aktiviert ist. Dies ist sowohl über die Kommandozeile als auch über das Luci Web-Frontend möglich. Ein funktionaler Test dieser Sicherheitsmaßnahme kann mittels des Singularity of Origin Web-Toolkits der NCC Group getestet werden. Als Target Host wird dabei die IP-Adresse des OpenWrt Routers spezifiziert. Desweiterten wurde das Intervall auf zwei reduziert und die Option „Flood DNS Cache“ aktiviert, da der Test mit einem Google Chromium basierten Browser durchgeführt wurde. Es bietet sich ebenfalls an verschiedene „Attack Payloads“ und Strategien zu testen. [http://rebind.it/manager.html]



Eine ebenso relevante Sicherheitsfunktion von DNS-Diensten ist die sogenannte „Source Port Randomization“ und „Transaction ID Randomization“, also die zufällige Wahl eines Quell-Ports, sowie einer Transaktions-ID für eine DNS-Anfrage. Diese Werte, welche vom DNS-Client generiert werden, dienen als Synchronisationsmethode zwischen dem DNS-Server und Client. Wenn der Quell-Port und die Transaktionsidentifikationsnummer von einem Angreifer berechnet oder geraten werden können, dann kann ein Angreifer diese nutzen um dem Opfer manipulierte DNS Antworten zu senden. Der DNS-Client würde diese aber als korrekt akzeptieren und eine potenziell schädliche Verbindung zu einem dritten Server aufbauen [Source]. Für einen funktionalen Test werden zunächst mithilfe des Python Skriptes send\_dns\_requests.py 1000 verschiedene DNS-Anfragen generiert. Dazu wird eine Liste mit 1000 häufig besuchten Webseiten genutzt [SOURCE]. Dies bietet sich an, da so sichergestellt wird, dass es sich wirklich um 1000 verschiedene DNS anfragen handelt und zum anderen ist es wahrscheinlich, dass diese Webseiten verfügbar sind. Während die DNS-Anfragen gestellt werden wird ein Mitschnitt aller Netzwerkpakete durch das Programm Wireshark gemacht. Die so erstellte Datei wird in einem weiteren Schritt analysiert. Dazu ließt das Python Skript „analyze\_pcap.py“ diese ein und selektiert im ersten Schritt alle DNS-Pakete, welche vom OpenWrt Router gesendet wurden. Daraufhin werden der DNS-Quell-Port sowie die Transaktions-ID aus diesen Paketen ausgelesen. Im letzten Schritt werden die Anzahl der DNS Anfragen, die Anzahl der einzigartigen Ports und Transaktions-IDs, die jeweiligen minimalen und maximalen Werte, die Standardabweichung und die häufigsten Werte angezeigt. Des Weiteren wird ein Kolmogorow-Smirnow-Test durchgeführt, um zu prüfen, ob die Verteilung der Daten mit einer Gleichverteilung übereinstimmt. Schlussendlich werden noch jeweils zwei Grafiken generiert, welche die Daten in einem Säulendiagramm und einen Streudiagramm darstellen. Auf diese Art kann der Tester prüfen, ob Muster in den Darstellungen zu erkennen sind.

3.3.2.7 Modul I – Factory Reset

Das Testen der Zurücksetzfunktion des OpenWrt Routers fällt aufgrund des uneingeschränkten Systemzugriffs einfach. Es können verschiedene Methoden eingesetzt werden. Zunächst sollte eine Leitlinie (Baseline) erstellt werden. Dazu dient ein Konfigurationsbackup, welches direkt nach dem ersten Einschalten des Geräts erstellt wurde. Diese wird anschließend mittels des Kommandozeilenprogramms diff mit einem Backup verglichen, welches nach der Nutzung des Routers und einem anschließenden Zurücksetzen des Geräts, nach Anleitung der OpenWrt Dokumentation, erstellt wurde. Alternativ kann das ebenfalls auf OpenWrt zur Verfügung stehende Kommandozeilenprogramm md5sum verwendet werden, um die Hash-Werte aller Dateien auf dem System zu generieren und diese zu exportieren. Diese sollten nach dem Zurücksetzen des Geräts übereinstimmen. Eine Datei mit den initialen Hash-Werten der betrachteten Version ist im Anhang enthalten.

3.3.3 Nicht anwendbare Test Prozeduren

Ebenso wie die Natur des OpenWrt Projektes ein einfaches Testen vieler Test Requirements ermöglicht, so werden einige Aspekte der Firmware anders gehandhabt als bei handelsüblichen Heimroutern. So sucht man vergeblich nach einem initial verfügbaren Wlan-Netz, nachdem der Router gestartet und eingerichtet wurde. Ebenso sind viele Funktionen, die ein Nutzer vielleicht von anderen Geräten gewöhnt ist, nur als zusätzliches Software-Paket verfügbar, oder durch aufwendige Konfiguration. Beispiele sind Wi-Fi Protected Setup (WPS), ein Community WLAN, Fernwartung, automatische Firmware-Updates oder Meldungen zu neuen Firmware-Updates, Voice over IP und Virtual Private Network Funktionen.

3.4 Statische Code-Analyse einiger quelloffenen Router Firmware Alternativen mittels FACT

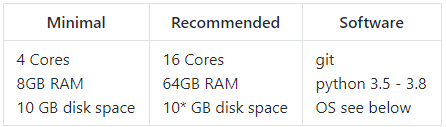
Neben der Methodik der Technischen Richtlinie des BSI gibt es noch viele weitere, um Aspekte einer Software zu evaluieren. Die Sicherheit einer betrachteten Software, in diesem Fall OpenWrt, lässt sich unter anderem durch sogenannte dynamische Tests oder statische Tests abschätzen. Diese Verfahren gehören zu den analytischen Softwaretests und unterscheiden sich darin, dass bei einem dynamischen Test die Software während der Laufzeit (execution based) getestet wird, während sie bei einem statischen Test nicht ausgeführt wird (non-execution based). Es wird sich für die Durchführung einer statischen Code-Analyse von quelloffener Router Firmware an der Methodik des „Home Router Security Reports 2020“ [SOURCE] des Fraunhofer-Institut für Kommunikation, Informationsverarbeitung und Ergonomie (FKIE) orientiert. In dieser Veröffentlichung des FKIE wurden 127 verschiedene, aktuelle Firmware-Abbilder von sieben Herstellern automatisch durch das ebenfalls vom FKIE entwickelte Firmware Analysis and Comparison Tool (FACT) analysiert und ausgewertet.

3.4.1 Installation und Testumgebung

FACT, welches vom FKIE auf github.com zur Verfügung gestellt wird, wurde lokal auf einem Desktop Computer installiert. Es handelt sich hierbei um ein System mit 12 Prozessoren, welche jeweils auf einer Taktfrequenz von 4.2GHz betrieben werden, sowie 16GB RAM. Ebenfalls stehen dem System 256GB persistenter Speicher zur Verfügung. Da die Installation auf Ubuntu 16.04, 18.04, 20.04 (stable) empfohlen wird, wurde Ubuntu 20.04 als aktuellster Vertreter des Ubuntu-Betriebssystems ausgewählt. Die zum Zeitpunkt der Arbeit aktuelle Version von FACT, FACT\_core v3.1.1 [https://github.com/fkie-cad/FACT\_core/archive/v3.1.1.zip], wurde mittels der bereitgestellten Anleitung installiert [https://github.com/fkie-cad/FACT\_core/blob/master/INSTALL.md].

|  |
| --- |
| $ sudo apt update && sudo apt upgrade && sudo apt install git  $ git clone https://github.com/fkie-cad/FACT\_core.git ~/FACT\_core  $ ~/FACT\_core/src/install/pre\_install.sh && sudo mkdir /media/data && sudo chown -R USER /media/data  $ sudo reboot  $ ~/FACT\_core/src/install.py  $ ~/FACT\_core/start\_all\_installed\_fact\_components |

Da das System den minimalen Software Anforderungen von FACT entspricht ist die Installation und Nutzung des Programms prinzipiell möglich, jedoch empfiehlt sich ein System mit mehr RAM, da dies die Performanz der Analyse erhöht. Ebenfalls kam es bei dem eingesetzten System vermehrt dazu, dass kein RAM mehr zur Verfügung stand und der Rechner während der Analyse nicht anderweitig genutzt werden konnte. Der Einsatz eines separaten Test Computers, oder eines Virtuellen Privaten Servers (VPS) ist zu empfehlen.



3.4.2 Erstellung des Firmware-Corpus

Der zu testende Firmware-Corpus besteht aus sieben verschiedenen, quelloffenen Router-Firmwares. Neben dem für die Technische Richtlinie verwendeten Abbild von OpenWrt Version 19.7.04 für den betrachteten TP-Link Archer C7 v5 Router, wurden noch sechs weitere Alternativen gewählt, von denen fünf spezifisch für das gewählte TP-Link Model Archer C7 v5 kompiliert sind. Zu den betrachteten Firmwares gehören DD-WRT, Gargoyle Router Management, Gluon, LibreCMC, AdvancedTomato, sowie Version 19.7.05 von OpenWrt. Einzig AdvancedTomato bietet keine Version für den getesteten Router an, weshalb auf eine Version für einen NETGEAR WNDR3700v3 Dual-Gigabit-WLAN-Router zurückgegriffen wurde, da dieser Router ebenfalls eine MIPS Architektur nutzt.

|  |  |  |
| --- | --- | --- |
| Hersteller | Geeignetes Produkt | Firmware Version |
| OpenWrt | TP-Link Archer C7 v5 | 19.07.4 |
| OpenWrt | TP-Link Archer C7 v5 | 19.07.5 |
| LibreCMC | TP-Link Archer C7 v2 | v1.5.3:2020-10-02 |
| Freifunk Gluon | TP-Link Archer C7 v5 | V2-v2020.2.1 |
| Gargoyle Router Management | TP-Link Archer C7 v5 | 1.12.0 (stable) |
| AdvancedTomato | NETGEAR WNDR3700v3 | 3.4-138 |
| DD-WRT | TP-Link Archer C7 v5 | 12-18-2020-r45036 |

Die beschriebenen Firmwares wurden gewählt, da sie in Funktion und Umfang OpenWrt ähnlich sind und die Projekte, denen sie entstammen, ebenfalls mehrere Heimrouter unterstützen. Es wurden keine Firmware-Alternativen gewählt, die auf Desktop Computern oder Servern installiert werden, da diese aufgrund der zur Verfügung stehenden Rechenkapazitäten im Leistungsumfang nicht vergleichbar sind. Das analysierte Korpus wurde am 21.12.2020 erstellt. Es wurde für jede analysierte Firmware die aktuellste Version für den TP-Link AC1750-Dualband-Gigabit-WLAN-Router genutzt, mit Ausnahme des Abbildes von OpenWrt Version 19.07.4. Diese Version wurde ebenfalls getestet, da es sich um die mittels der Technischen Richtlinie geprüfte Version handelt.

3.4.3 Durchgeführte Tests und Metriken

Um einen Vergleich mit den Ergebnissen des Home Router Security Reports 2020 des FKIE zu ermöglichen wurden die gleichen Aspekte auch bei den quelloffenen Firmwares analysiert. Es wurden die folgenden sicherheitsrelevanten Aspekte betrachtet:

* Wann wurde das letzte Update für das Gerät veröffentlicht?
* Welches Betriebssystem wird verwendet und wie viele bekannte kritische Schwachstellen sind für dieses bekannt?
* Welche Methoden zur Vereitelung von Angriffen werden eingesetzt und wie häufig sind diese aktiviert.
* Ist privates kryptografisches Schlüsselmaterial enthalten?
* Können Login Daten in dem Firmware-Abbild gefunden werden?

Die einzelnen Komponenten des Firmware Analysis and Comparison Tools (FACT) werden mittels des Befehls   
$ ~/FACT\_core/start\_all\_installed\_fact\_components   
gestartet. Nachdem der lokale Server gestartet ist, werden die Firmware-Abbilder einzeln über die Upload-Funktion hochgeladen. Die folgenden Analyse-Methoden wurden gewählt:

* CPU Architecture
* Crypto Material
* CVE Lookup
* CWE Checker
* Exploit Mitigations
* Known Vulnerabilities
* Software Components
* Source Code Analysis
* Users and Passwords

Die Ergebnisse der automatischen Analyse werden anschließend durch die REST API von FACT ausgelesen und als Grafiken dargestellt, sodass eine direkte Gegenüberstellung der Ergebnisse des FKIE mit den erhobenen Daten möglich ist.

Kapitel 4

Ergebnisse

4.1 Ergebnisse der TR

4.2 Notwendige Änderungen zum Bestehen der TR

4.2.1 Problemerkennung

4.2.2 Problemlösung

4.3 Ergebnisse der statischen Code-Analyse sowie Gegenüberstellung mit ausgewählten Ergebnissen des Home Router Security Reports 2020

Im Rahmen dieser statischen Code-Analyse durch das Firmware Analysis and Comparison Tool (FACT) des FKIE wurden sieben verschiedene quelloffene Router-Firmware Alternativen analysiert. Dabei waren fünf Fragen von besonderem Interesse.

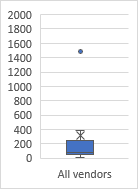
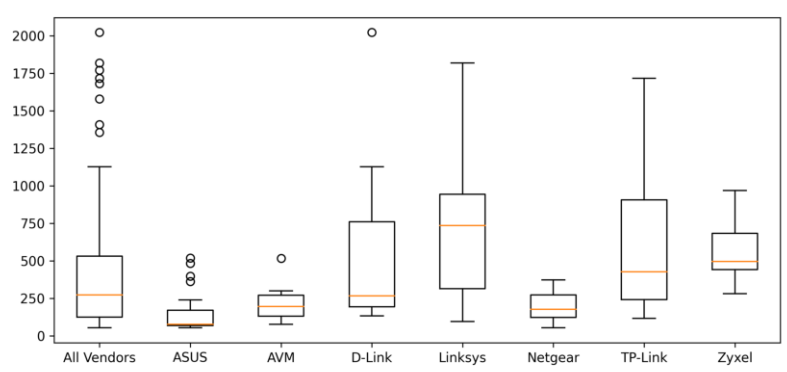
* Wann wurde das letzte Update für das Gerät veröffentlicht?
* Welches Betriebssystem wird verwendet und wie viele bekannte kritische Schwachstellen sind für dieses bekannt?
* Welche Methoden zur Vereitelung von Angriffen werden eingesetzt und wie häufig sind diese aktiviert.
* Ist privates kryptografisches Schlüsselmaterial enthalten?
* Können Login Daten in dem Firmware-Abbild gefunden werden?

FACT konnte während der Analyse erfolgreich 92,73% der Daten aus den Firmware-Abbildern extrahieren. Bei allen betrachteten Firmware wurde durch Analyse von Metadaten eine MIPS 32-bit Architektur mit „big-endian“ Byte-Reihenfolge festgestellt werden. Für die Analyse der „Critial Vulnerabilities and Exposures“ (CVE) wurde aufgrund einiger Fehler in FACT nicht das Ergebnis der automatischen Analyse gewählt. Stattdessen wurden die Ergebnisse durch die Webseite [www.cvedetails.com](http://www.cvedetails.com), welche wiederum auf die Daten der „National Vulnerability Database, zugreift, bereitgestellt. Da cvedetial.com ausschließlich CVSS v2 Bewertungen bereitstellt, wurden einzig diese für die Analyse verwendet. Ein CVE-Eintrag hat einen Schweregrad von „Hoch“, wenn der CVSS v2 Wert sieben oder höher beträgt. Um Vergleichbarkeit mit den Ergebnissen des FKIE zu gewährleisten wurden lediglich CVE-Einträge mit einem Schweregrad von „Hoch“ gezählt.

4.3.1 Vergangene Tage seit der letzten Veröffentlichung eines Firmware-Updates

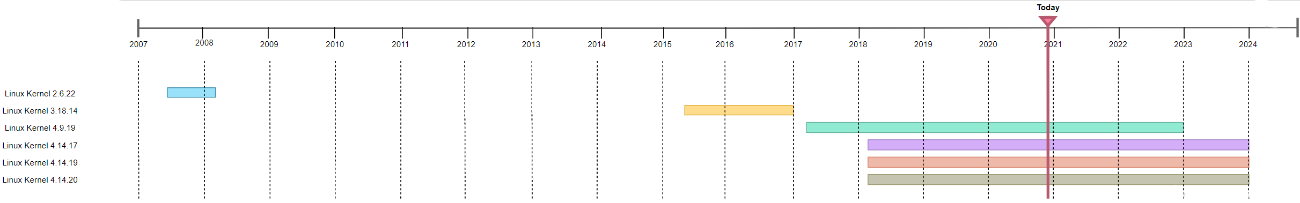
In diesem Abschnitt soll evaluiert werden, wann für die betrachteten Firmware-Abbilder das letzte Mal eine neue Version seit dem 24.12.2020 veröffentlicht wurde. Alle Abbilder des Firmware-Corpus spezifizierten das Veröffentlichungsdatum im Dateinamen selbst oder auf der Webseite. Dieses Kriterium wurde untersucht, da es zum einen die Bereitschaft der Entwickler andeutet, ihr Projekt regelmäßig mit Funktions- und Sicherheitsupdates zu unterstützen, zum anderen ist weniger Zeit vergangen, in denen sicherheitsrelevante Lücken gefunden werden konnten. Da die Unterstützer der quelloffenen Projekte in vielen Instanzen auf weitere Software zurückgreifen und auch diese Updates erfährt, ist es wahrscheinlich, dass Firmware bekannte Lücken hat, wenn diese längere Zeit nicht erneuert wurde.   
 Grafik [NUMMER] zeigt, dass es für fünf von sieben untersuchten Firmware in den letzten 365 Tagen eine neue Version gab. Die verwendete Stichprobe hat einen zu geringen Umfang, um dem Mittelwert besondere Bedeutung zukommen zu lassen. Der Median ist in diesem Falle besser geeignet. Es ergibt sich, dass die Router-Betriebssysteme nach Median-Berechnung alle 83 Tage und im Schnitt alle 309 Tage eine neue Version erhalten. Ebenfalls muss erwähnt werden, dass bei der Veröffentlichung einer neuen Version meist alle von dem jeweiligen Projekt unterstützen Geräte diese neue Version zur Verfügung gestellt bekommen. So werden bei einer neuen Version von OpenWrt alle ca. 1700 Geräte von diesem neuen Update unterstützt und erfahren somit alle Sicherheitsupdates, die bereitgestellt werden. Gargoyle Router Management wurden nicht in den letzten 365 Tagen erneuert und das Tomato Betriebssystem hat in den letzten 1480 Tagen kein Update erfahren. Der Zyklus von im Median 83 Tagen ist höher als der Update-Zyklus von Desktop- oder Server-Betriebssystemen, jedoch noch im Rahmen der 90 Tage, welche normalerweise das Zeitfenster darstellen, in dem Entwickler Zeit haben auf Sicherheitslücken und Probleme zu reagieren (responsible disclosure) [SOURCE]. Darüber hinaus muss bedacht werden, dass zumindest im Falle von OpenWrt ein Paketmanager zur Verfügung steht, über welchen während der Laufzeit Updates für Pakete installiert werden können. Somit sind Updates der Firmware nur notwendig, um Kernfunktionalität zu erweitern oder Fehler in dieser zu beheben, sowie um den Kernel zu updaten. Nur eine von acht bisher veröffentlichten Sicherheitslücken im Jahr 2020 konnte ausschließlich durch ein Update auf eine neuere Version von OpenWrt behoben werden, wobei alle weiteren durch ein einfaches Update des betroffenen Paketes verhindert werden konnten. Falls die anderen analysierten Firmware eine ähnliche Funktion bereit stellen, so gelte die Aussage ebenso.

Verglichen mit den Ergebnissen des Home Router Security Reports 2020 zeigt sich, dass für die quelloffenen Betriebssysteme häufiger neue Versionen veröffentlicht werden. Wenn man die analysierten Abbilder als Gruppe betrachtet, dann schneidet diese vergleichsweise gut ab. Einzig Tomato fällt als Ausreißer heraus. Einzig ASUS, AVM und Netgear, als Hersteller von handelsüblichen Routern, können mithalten.

Ebenso wie im Home Router Security Report 2020 festgestellt, muss zusätzlich beachtet werden, dass alle betrachteten Produkte kleinere Updates auch über die Geräte selbst zu Verfügung stellen könnten, sodass die aktuellste Version nicht im Internet veröffentlicht wird. Darüber hinaus handelt es sich bei den hier festgestellten Daten ausschließlich um eine Momentaufnahme, die keine Aussagekraft darüber hat, ob regelmäßig Updates bereitgestellt werden, oder ob diese Sicherheitslücken adressieren [SOURCE FKIE PAPER].

4.3.2 Betriebssysteme

Da es sich bei allen analysierten Firmware-Abbildern um quelloffene Projekte handelt, ist es nicht verwunderlich, dass der Linux-Kernel dominant vertreten ist. Der Linux-Kernel, welcher 1991 von Linus Tovalds entwickelt wurde und seither stetig weiterentwickelt wird stellt einen der am häufigsten genutzten Betriebssysteme für IOT Geräte dar [SOURCE]. Die geringe Größe des Kernels, der große Funktionsumfang sowie die umfangreiche Dokumentation und Verbreitung sind für eine Community-getriebene Entwicklung auf Speicher- und Rechenleistungslimitierten Geräten, wie Heim-Routern, gut geeignet. Grafik [NUMMER] zeigt, dass alle untersuchten Projekte einen Linux Kernel verwenden. Dieser Trend deckt sich ebenfalls mit den Ergebnissen des Home Router Security Reports des FKIE. In den untersuchten Produkten des Verbrauchermarktes wurde Linux in 91% der Fälle verwendet [SOURCE].   
 Aufgrund der unzureichenden Ergebnisse der FACT-Analyse bezüglich der vorhandenen CVE-Einträge für die verwendeten Linux-Kernel wurden die Ergebnisse in diesem Fall direkt über [www.cvedetails.com](http://www.cvedetails.com) abgerufen. FACT erstellt zunächst eine „Common Platform Enumeration“ (CPE) der Software Version und stellt mit dieser CPE eine Anfrage an die „National Vulnerability Database“. Da die zurückgegebenen Ergebnisse allerdings auch Schwachstellen beinhalten, welche nur für bestimme Geräte mit der jeweiligen Linux-Kernel Version gelten, wurden die jeweiligen Schwachstellen des Kernels über die Website cvedetail.com abgefragt. Diese Webseite nutzt ebenfalls die „National Vulnerability Database“, stellt jedoch noch zusätzliche Informationen und Statistiken bereit. Auf diese Art wurde sichergestellt, dass die betrachteten Schwachstellen spezifisch für den Kernel sind und nicht für ein bestimmtes Gerät, welches diesen Kernel nutzt. Da nicht alle eingetragenen CVEs eine direkte Bedrohung darstellen, wurden die Ergebnisse weiter eingeschränkt. So wurden lediglich solche CVEs betrachtet, welche mit einen CVSS2 (Common Vulnerability Scoring System) Wert von sieben oder höher eingestuft wurden. Dies ist ein Bewertungssystem, mit welchem CVE Einträge kategorisiert werden, sodass der Schweregrad der Sicherheitslücke durch einen Wert definiert werden kann. Da das neuere Format, CVSS3, nicht durch cvedetails.com bereitgestellt wird, wird es in der Analyse vernachlässigt. Wie Grafik [NUMMER] zeigt, stehen für alle betrachteten Geräte einige CVE Einträge des Linux Kernels zur Verfügung. Ebenfalls kann man sehen, dass der in DD-WRT verwendete Kernel mehr CVE Einträge hat als der von Gargoyle Router Management, obwohl bei DD-WRT die geringste Zeit seitdem letzten Firmware Update vergangen ist. Tomato schneidet erneut als letzter ab. Grafik [NUMMER] zeigt zusätzlich, dass für zwei der sechs verschiedenen Linux Kernel schon seit einigen Jahren keine Sicherheitsupdates entwickelt werden. Sowohl der von Tomato verwendete Kernel, 2.6.22, als auch Linux Kernel 3.8.14, welcher von DD-WRT verwendet wird, werden nicht mehr unterstützt. Dies spiegelt sich auch in der hohen Anzahl CVE Einträge wider (siehe Grafik).   
 Die Ergebnisse sind aufgrund der unterschiedlichen Beschaffung sowie der fehlenden CVSS3 Werte nicht wirklich mit denen des Home Router Security Reports 2020 vergleichbar. Jedoch kann man sagen, dass die quelloffenen Router-Betriebssysteme im Schnitt modernere Linux-Kernel Versionen nutzen. Lediglich zwei der betrachteten Firmware nutzen einen Kernel, der nicht mehr unterstützt wird. Der Security Report 2020 gibt an, dass ein drittel der betrachteten Geräte einen Kernel vor Version 3 nutzen und lediglich ca. 22% einen aktuellen Kernel aus der 4. Version. Im Gegensatz dazu nutzen ca. 70% der betrachteten quelloffenen Software-Projekte einen Linux-Kernel der Version 4.9.19 oder höher (siehe Grafik).   
 Im Gegensatz zu den Ergebnissen des Security Reports können falsch positive Ergebnisse bei der Erkennung der Kernel Version beinahe ausgeschlossen werden, da diese ebenfalls von den Entwicklern auf der Webseite oder in den Veröffentlichungsdokumenten der jeweiligen Version veröffentlicht wird. Jedoch besteht die Möglichkeit, dass die Entwickler eigene Korrekturen für Sicherheitslücken des Kernels entwickeln und veröffentlichen. Dies ist bei dieser Art Community-getriebener Entwicklung nicht unwahrscheinlich, da hier keine Entwickler bezahlt werden müssen, welche zusätzlich zu ihren anderen Aufgaben für das Beheben von Sicherheitslücken im Kernel eingesetzt werden. Ebenfalls ist es möglich, dass aufgrund der uneindeutigen CPE Spezifikation eine CVE Einträge nicht von cvedetails.com gelistet werden [SOURCE Router report 7].

4.3.3 Exploit Mitigations

Die betrachteten Firmware Abbilder wurden auf die folgenden Exploit Mitigationen getestet:

* Stack Canary: Es handelt sich hierbei um eine zufällig gewählte Byte-Sequenz, welche vor die „return“-Adresse auf den Stack geschrieben wird, um Overflows zu erkennen. Wenn es zu einem Buffer-Overflow kommt, würde diese Sequenz überschrieben und kann somit nicht vor dem Zurückkehren (return) nicht korrekt verifiziert werden [SOURCE].
* FORTIFY\_SOURCE ist eine zusätzliche Option der GCC Compiler Collection. Wenn diese Option bei der Kompilierung von Dateien ausgewählt wird, werden verschiedene Funktionen zur Manipulation von Zeichenketten und Speicher (memcpy, memset, strcpy, strcat, sprintf, gets, …) während der Ausführung auf buffer overflows geprüft. Dies schützt meistens nicht vor gezieltem Ausnutzen dieser Funktionen [SOURCE https://man7.org/linux/man-pages/man7/feature\_test\_macros.7.html].
* Non-Executable Bit (NX): Dieses besondere Bit markiert Bereiche des Speichers als reine Datenspeicherbereiche. Dadurch wird sichergestellt, dass in diesen Bereichen, in denen kein Code ausgeführt werden sollte, auch kein Code ausgeführt werden kann. Diese Separierung findet sich sonst nur in Harvard-Architekturen [SOURCE] [SOURCE].
* Position-Independent Executable (PIE) (positionsunabhängiges ausführbares Programm) bezeichnet eine Technik, bei welcher Programm-Code an einer zufälligen Speicheradresse geladen wird. Hierbei wird nicht mit absoluten, sondern relativen Speicheradressen gearbeitet. Dies erschwert zwar Angriffe, da ein Angreifer zunächst die absolute Speicheradresse finden muss, jedoch verlangsamt diese Technik unter Umständen auch die Ausführung des Codes [SOURCE] [https://www.research-collection.ethz.ch/bitstream/handle/20.500.11850/68932/eth-5699-01.pdf].
* RELocation Read-Only (RELRO) schützt den “Global Offset Table” (GOT) gegen Manipulationen während der Laufzeit. Der Global Offset Table beinhaltet die Speicheradressen von gemeinsam genutzten Libraries oder globalen Variablen, sodass diese von einem Programm genutzt werden können [SOURCE ÜBERSETZT REPORT]. Wenn die RELRO Option beim Kompiliervorgang ausgewählt wurde, dann wird nach dem Start des Programms ein reiner Lesezugriff auf den GOT festgelegt.

Sowohl RELRO als auch das NX-Bit werden vermehrt bei den quelloffenen Router-Betriebssystemen eingesetzt. Außer Tomato nutzen alle betracheteten Firmware zu beinahme 100% NX. Mit Ausnahme von Tomato und DD-WRT nutzen im Schnitt ca. 50% aller ausführbarer Dateien der Firmware-Abbilder RELRO. Tomato und DD-WRT setzen hingegen gar nicht auf RELRO. PIE wird hingegen im Schnitt zu ca. 40% genutzt von allen Firmware. Tomato scheint bevorzugt auf PIE zu setzen (siehe Grafik). Die Nutzung von Stack Canaries und FORTIFY\_SOURCE verhält sich beinahe identisch. Gargoyle Router Management, LibreCMC und OpenWrt nutzen es bei ca. 19% aller ausführbarer Dateien, Gluon bei ca. 8%, während DD-WRT und Tomato beinahe vollständig auf diese Techniken verzichten.

Wie Grafik [NUMMER] zeigt, ist die Verbreitung von PIE vergleichbar. Ebenso wie im Security Report berichtet, nutzen auch die quelloffenen Betriebssysteme beinahe alle vollumfänglich NX-Bits. Dies lässt sich leicht durch den vergleichsweisen guten Schutz, bei infinitesimalen Geschwindigkeitseinbußen erklären. Die Daten des FKIE zeigten, dass RELRO nur selten von allen Herstellern, außer AVM, eingesetzt wird. Dem steht eine Nutzung von ca. 50% bei den freien Firmware-Produkten gegenüber. Ebenso wie die betrachteten Firmware der Markthersteller, wird nur selten auf Stack Canaries und FORTIFY\_SOURCE gesetzt. Obwohl Stack Canaries keinen merkbaren Einfluss auf die Geschwindigkeit eines Systems hat, scheint diese Technik nur bei absolut Systemkritischen Dateien angewendet worden zu sein. Dies gilt ebenso für die FORTIFY\_SOURCE Option (siehe Grafik).

Zusammenfassend kann man sagen, dass vor allem auf NX und RELRO für den großteil der Dateien gesetzt wird. PIE, Stack Canaries und FORTIFY\_SOURCE wird nur bei wenigen ausführbaren Dateien genutzt. Es lässt sich vermuten, dass es sich bei diesen Dateien um systemkritische Funktionen handelt, welche oft das Ziel von Angriffen sind.

4.3.4 Private Keys

Wenn private kryptographische Schlüssel in den Firmware Abbildern enthalten sind, so haben diese keine Sicherheitsfunktion mehr. Wenn privaten Schüssels unabdingbar in eine Firmware integriert werden müssen, um die korrekte Funktionaität zu gewährleisten, so sollten die Vorgaben der OWASP eingehalten werden

|  |
| --- |
| “Do not hardcode secrets such as passwords, usernames, tokens, private keys or similar variants into firmware release images. This also includes the storage of sensitive data that is written to disk. If hardware security element (SE) or Trusted Execution Environment (TEE) is available, it is recommended to utilize such features for storing sensitive data. Otherwise, use of strong cryptography should be evaluated to protect the data. If possible, all sensitive data in clear-text should be ephemeral by nature and reside in a volatile memory only.” [https://owasp.org/www-project-embedded-application-security/#div-project 26.12] |

Die Einhaltung dieser Vorgaben ist jedoch deutlich erschwert, wenn die Firmware nicht spezifisch für ein Gerät geschrieben ist. Ebenso stehen den Entwicklern der quelloffenen Firmware nicht alle Entwicklerwerkzeuge der Hersteller zur Verfügung um z.B. auf ein „Hardware Security Element“ zuzugreifen. Ebenso wird für den Zugriff physischer Zugang zu dem Gerät benötigt.   
 Trotz dieser Probleme konnte FACT nur in DD-WRT und Gargoyle Router Management private Schlüssel extrahieren. Bei beiden Betriebssystemen wurden jeweils ein Pkcs8PrivateKey sowie ein SSLPrivateKey gefunden. Da PKCS#8 ein Container-Format für private kryptographische Schlüssel ist, kann man ohne weitere Nachforschung nicht bestimmen, welchen nutzen diese Schlüssel für die Systeme haben. Die gefundenen SSL Schlüssel dienen dazu die vom Webbrowser an den Webserver gesendeten Session-Key zu entschlüsseln [SOURCE <https://ssl.de/ssl-glossar/private-key.html>]. Es lässt sich also vermuten, dass DD-WRT und Gargoyle Transport Layer Security verwenden, jedoch kann ein Man-in-the-Middle [SOURCE] Angriff einfach durchgeführt werden, wenn der private SSL Schlüssel bekannt ist. Die genauen Details der Implementierung und Nutzung der gefundenen Schlüssel ist jedoch vollkommen unbekannt. Es könnte sich ebenso um ungenutztes oder veraltetes Material handeln. Darüber hinaus könnte der SSL Schlüssel auch nur für die initiale Konfiguration des Gerätes genutzt werden, um danach durch einen neuen ersetzt zu werden.

4.3.5 Login Credentials

Kapitel 5

Diskussion

5.1 Zusammenfassung der Ergebnisse

5.2 Limitationen

5.3 Implikationen

Kapitel 6

Fazit

Literaturverzeichnis

Anhang