Kolloquium zur Abschlussarbeit
im Bachelorstudiengang Informatik

Untersuchung der Sicherheit von OpenWrt anhand der BSI TR-03148 mittels eines OpenWrt betriebenen Heim-Routers

von Henry Weckermann

Erstbetreuer:	Prof. Markus Ullmann
Zweitbetreuer:	Prof. Dr. Norbert Jung
Betreuer im BSI:	Florian Bierhoff

Agenda

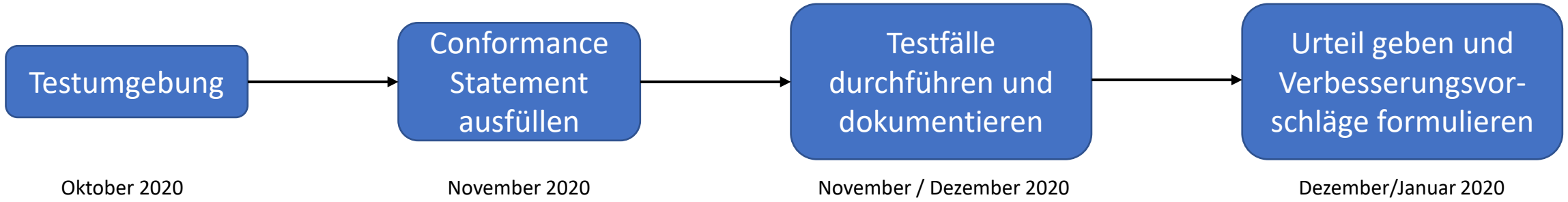
- Ziele der Arbeit
- Verwandte Arbeiten
- Methodik
  - Testumgebung und Rahmenbedingungen
  - Technische Richtlinie
  - Statische Code-Analyse mit FACT
- Ergebnisse
  - TR-Konformität von OpenWrt
  - Ergebnisse der Code-Analyse
- Limitationen und zukünftige Forschung

Agenda

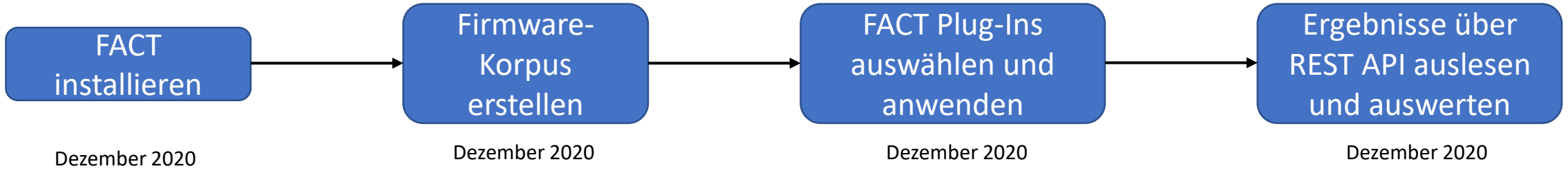
- **Ziele der Arbeit**
- Verwandte Arbeiten
- Methodik
  - Testumgebung und Rahmenbedingungen
  - Technische Richtlinie
  - Statische Code-Analyse mit FACT
- Ergebnisse
  - TR-Konformität von OpenWrt
  - Ergebnisse der Code-Analyse
- Limitationen und zukünftige Forschung

Ziele der Arbeit

1. TR-konformität von OpenWrt prüfen



2. Statischer Softwaretest von quelloffener Router-Firmware



Agenda

- Ziele der Arbeit
- **Verwandte Arbeiten**
- Methodik
  - Testumgebung und Rahmenbedingungen
  - Technische Richtlinie
  - Statische Code-Analyse mit FACT
- Ergebnisse
  - TR-Konformität von OpenWrt
  - Ergebnisse der Code-Analyse
- Limitationen und zukünftige Forschung

Verwandte Arbeiten

- **Ortega et al. (2009) [1]** — **ARP Cache Poisoning Angriffe verhindern (Technik auf OpenWrt implementiert)**
- **Palazzi et al. (2010) [2]** — **OpenWrt als vereinheitlichende Plattform für kabellose Geräte**
- **Andrew McDonnell (2014) [3]** — **Sicherheitsevaluation von OpenWrt Barrier Breaker (erschienen 2014)**

- **Viel Forschung am und mit dem Linux Kernel [4, 5]**
- **(Einsatz von /) Forschung an Softwarepaketen, welche auch in OpenWrt genutzt werden [6, 7]:**
  - **OpenSSL**
  - **dnsmasq**
  - **BusyBox**
  - **Dropbear**
  - **iptables**

Verwandte Arbeiten

- **Ortega et al. (2009) [1]** — **ARP Cache Poisoning Angriffe verhindern (Technik auf OpenWrt implementiert)**
- **Palazzi et al. (2010) [2]** — **OpenWrt als vereinheitlichende Plattform für kabellose Geräte**
- **Andrew McDonnell (2014) [3]** — **Sicherheitsevaluation von OpenWrt Barrier Breaker (erschienen 2014)**

- **Viel Forschung am und mit dem Linux Kernel [4, 5]**
- **(Einsatz von /) Forschung an Softwarepaketen, welche auch in OpenWrt genutzt werden [6, 7]:**
  - **OpenSSL**
  - **dnsmasq**
  - **BusyBox**
  - **Dropbear**
  - **iptables**

Verwandte Arbeiten

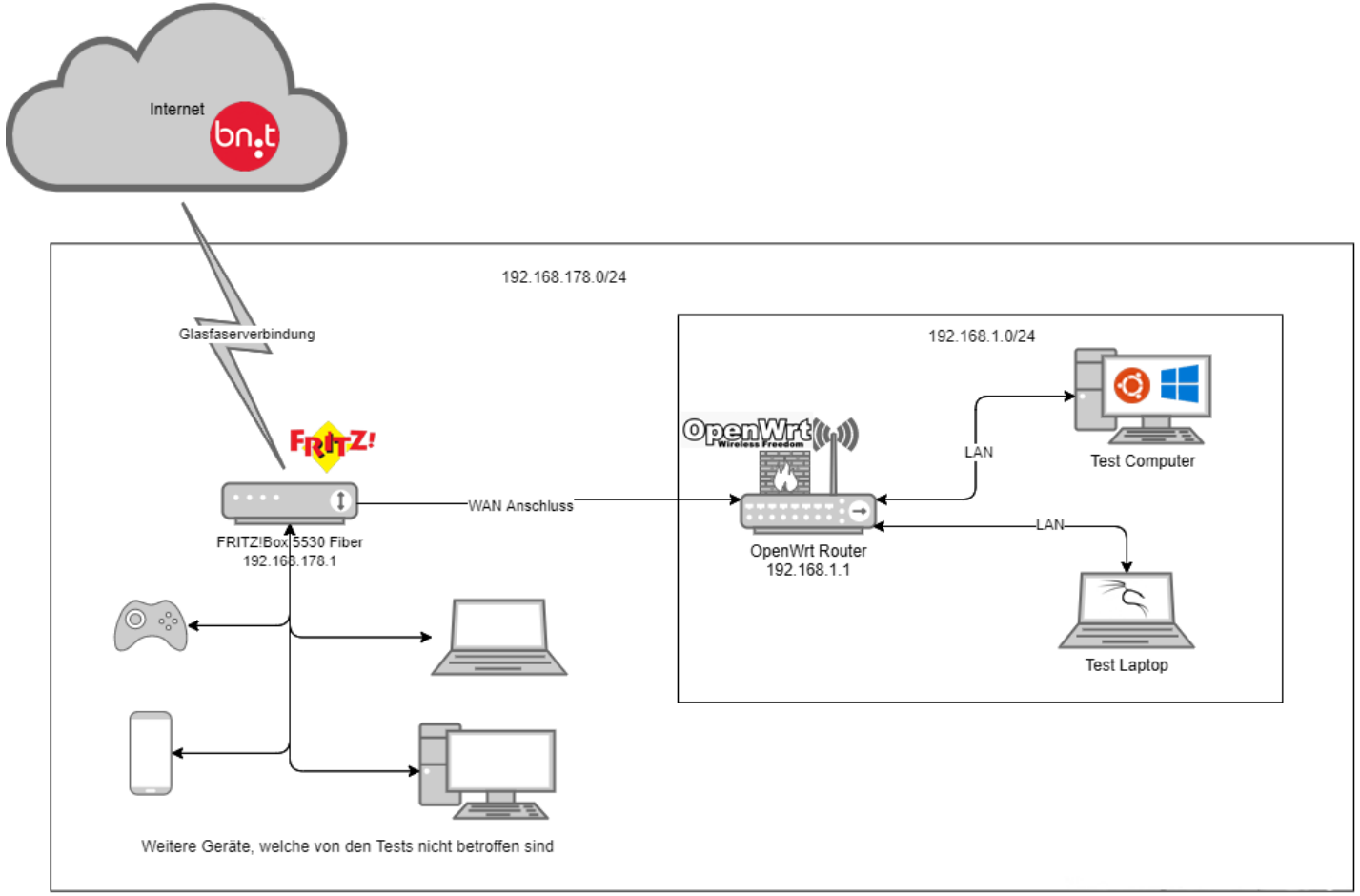
- **Ortega et al. (2009) [1]** — **ARP Cache Poisoning Angriffe verhindern (Technik auf OpenWrt implementiert)**
- **Palazzi et al. (2010) [2]** — **OpenWrt als vereinheitlichende Plattform für kabellose Geräte**
- **Andrew McDonnell (2014) [3]** — **Sicherheitsevaluation von OpenWrt Barrier Breaker (erschienen 2014)**

- **Viel Forschung am und mit dem Linux Kernel [4, 5]**
- **(Einsatz von /) Forschung an Softwarepaketen, welche auch in OpenWrt genutzt werden [6, 7]:**
  - **OpenSSL**
  - **dnsmasq**
  - **BusyBox**
  - **Dropbear**
  - **iptables**

Agenda

- Ziele der Arbeit
- Verwandte Arbeiten
- **Methodik**
  - **Testumgebung und Rahmenbedingungen**
  - Technische Richtlinie
  - Statische Code-Analyse mit FACT
- Ergebnisse
  - TR-Konformität von OpenWrt
  - Ergebnisse der Code-Analyse
- Limitationen und zukünftige Forschung

Methodik - Testumgebung



Methodik – Device under Test

MODEL:	Archer C7 AC1750
VERSION:	v5
SUPPORTED SINCE REL:	18.06.0
SUPPORTED CURRENT REL:	19.07.6
TARGET:	ar71xx-ath79
SUBTARGET:	generic
PACKAGE ARCHITECTURE:	mips_24kc
CPU:	Qualcomm Atheros QCA9563
CPU CORES:	1
CPU MHZ:	750
FLASH MB:	16
RAM MB:	128
WLAN HARDWARE:	Qualcomm Atheros QCA9563, Qualcomm Atheros QCA9880
WLAN 2.4GHZ:	b/g/n
WLAN 5.0GHZ:	a/n/ac
WLAN DRIVER:	ath9k, ath10k

[8]

[9]

[9]

Methodik – Device under Test (2)

Gründe für dieses Gerät:

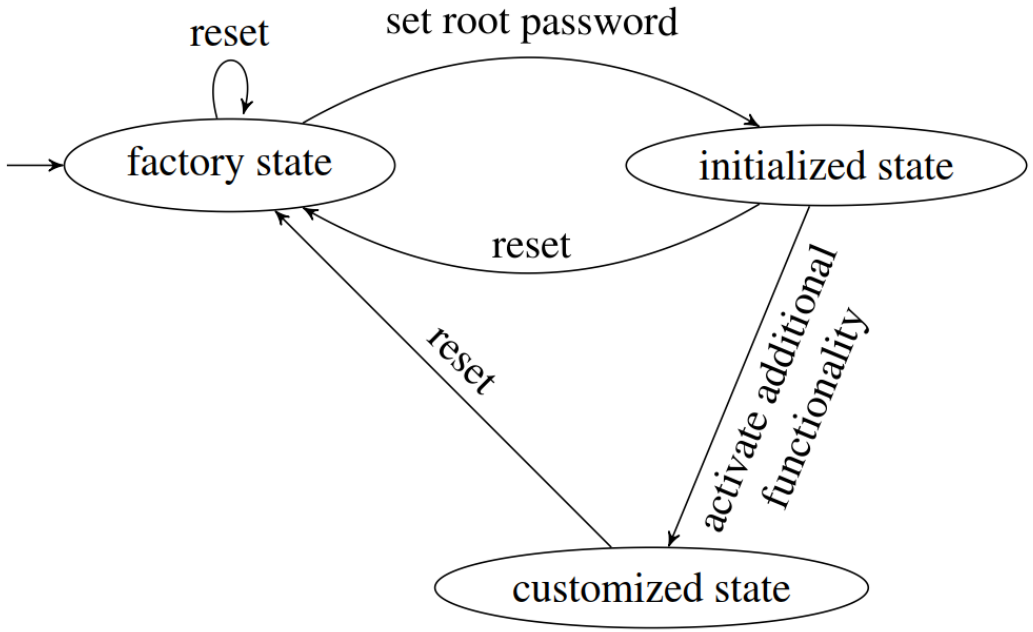
- Günstig (ca. 60€)
- Ausreichend Leistung für OpenWrt (s. Folie 21)
- Große Beliebtheit in der OpenWrt Community
  - Derzeit Platz 10 der Downloads [10]

OpenWrt Version:

- 19.07.4
- Veröffentlicht: 10. September 2020 [11]

Methodik – Device under Test (3)

Zustände des DUT



Agenda

- Ziele der Arbeit
- Verwandte Arbeiten
- **Methodik**
  - Testumgebung und Rahmenbedingungen
  - **Technische Richtlinie**
  - Statische Code-Analyse mit FACT
- Ergebnisse
  - TR-Konformität von OpenWrt
  - Ergebnisse der Code-Analyse
- Limitationen und zukünftige Forschung

Methodik – Technische Richtlinie

Test Documentation – Programme und Tools

- **nmap** **Port Scanner [12]**
- **Aircrack-ng** **Analyse von Wifi Netzwerk Sicherheit [13]**
- **Wireshark** **Netzwerkpaket Sniffer [14]**
- **testssl.sh** **Test der TLS/SSL Verschlüsselung [15]**
- **Python Skripte:**
  - Bruteforce check (SSH / web)
  - CSRF Token Einzigartigkeit
  - DNS Source Port / Transaction ID Zufälligkeit
- **DNS Rebinding Angriff: http://rebind.it/singularity.html [16]**

Agenda

- Ziele der Arbeit
- Verwandte Arbeiten
- **Methodik**
  - Testumgebung und Rahmenbedingungen
  - Technische Richtlinie
  - **Statische Code-Analyse mit FACT**
- Ergebnisse
  - TR-Konformität von OpenWrt
  - Ergebnisse der Code-Analyse
- Limitationen und zukünftige Forschung

Statische Code-Analyse mit FACT

Firmware Analysis and Comparison Tool (FACT) [17]

- **Vom Fraunhofer-Institut für Kommunikation, Informationsverarbeitung und Ergonomie (FKIE) entwickelt**
- In Python realisiert
- Software zur automatischen Durchführung einer statischen Firmwareanalyse
- Bietet verschiedene Analysen als Plug-Ins an
- Vergleich von analysierter Firmware möglich
- Datenexport durch REST API

Statische Code-Analyse mit FACT

Firmware Analysis and Comparison Tool (FACT) [17]

- **Vom Fraunhofer-Institut für Kommunikation, Informationsverarbeitung und Ergonomie (FKIE) entwickelt**
- **In Python realisiert**
- **Software zur automatischen Durchführung einer statischen Firmwareanalyse**
- **Bietet verschiedene Analysen als Plug-Ins an**
- **Vergleich von analysierter Firmware möglich**
- **Datenexport durch REST API**

Statische Code-Analyse mit FACT

Firmware Analysis and Comparison Tool (FACT) [17]

- **Vom Fraunhofer-Institut für Kommunikation, Informationsverarbeitung und Ergonomie (FKIE) entwickelt**
- **In Python realisiert**
- **Software zur automatischen Durchführung einer statischen Firmwareanalyse**
- **Bietet verschiedene Analysen als Plug-Ins an**
- **Vergleich von analysierter Firmware möglich**
- **Datenexport durch REST API**

Statische Code-Analyse mit FACT

Firmware Analysis and Comparison Tool (FACT) [17]

- **Vom Fraunhofer-Institut für Kommunikation, Informationsverarbeitung und Ergonomie (FKIE) entwickelt**
- **In Python realisiert**
- **Software zur automatischen Durchführung einer statischen Firmwareanalyse**
- **Bietet verschiedene Analysen als Plug-Ins an**
- **Vergleich von analysierter Firmware möglich**
- **Datenexport durch REST API**

Statische Code-Analyse mit FACT

Firmware Analysis and Comparison Tool (FACT) [17]

- **Vom Fraunhofer-Institut für Kommunikation, Informationsverarbeitung und Ergonomie (FKIE) entwickelt**
- **In Python realisiert**
- **Software zur automatischen Durchführung einer statischen Firmwareanalyse**
- **Bietet verschiedene Analysen als Plug-Ins an**
- **Vergleich von analysierter Firmware möglich**
- **Datenexport durch REST API**

Statische Code-Analyse mit FACT

Firmware Analysis and Comparison Tool (FACT) [17]

- **Vom Fraunhofer-Institut für Kommunikation, Informationsverarbeitung und Ergonomie (FKIE) entwickelt**
- **In Python realisiert**
- **Software zur automatischen Durchführung einer statischen Firmwareanalyse**
- **Bietet verschiedene Analysen als Plug-Ins an**
- **Vergleich von analysierter Firmware möglich**
- **Datenexport durch REST API**

Statische Code-Analyse mit FACT (2)

Firmware Korpus

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

Statische Code-Analyse mit FACT (3)

- **Letztes Update der Firmware**
  - **Veröffentlichungsdatum**
- **Linux Version + Anzahl der CVE Einträge für diese Linux Version**
  - **Plug-Ins: CVE Lookup, CWE Checker, Software Components, Known Vulnerabilities**
- **Analyse der Exploit Mitigations (PIE, RELRO, NX, FORTIFY_SOURCE, Stack Canary)**
  - **Plug-In: Exploit Mitigations**
- **Privates kryptographisches Material**
  - **Plug-In: Crypto Material**
- **Hartkodierte Login-Daten (Nutzeraccounts) mit bekannten Passwörtern**
  - **Plug-In: Users and Passwords**

Statische Code-Analyse mit FACT (3)

- **Letztes Update der Firmware**
  - **Veröffentlichungsdatum**
- **Linux Version + Anzahl der CVE Einträge für diese Linux Version**
  - **Plug-Ins: CVE Lookup, CWE Checker, Software Components, Known Vulnerabilities**
- **Analyse der Exploit Mitigations (PIE, RELRO, NX, FORTIFY_SOURCE, Stack Canary)**
  - **Plug-In: Exploit Mitigations**
- **Privates kryptographisches Material**
  - **Plug-In: Crypto Material**
- **Hartkodierte Login-Daten (Nutzeraccounts) mit bekannten Passwörtern**
  - **Plug-In: Users and Passwords**

Statische Code-Analyse mit FACT (3)

- **Letztes Update der Firmware**
  - **Veröffentlichungsdatum**
- **Linux Version + Anzahl der CVE Einträge für diese Linux Version**
  - **Plug-Ins: CVE Lookup, CWE Checker, Software Components, Known Vulnerabilities**
- **Analyse der Exploit Mitigations (PIE, RELRO, NX, FORTIFY_SOURCE, Stack Canary)**
  - **Plug-In: Exploit Mitigations**
- **Privates kryptographisches Material**
  - **Plug-In: Crypto Material**
- **Hartkodierte Login-Daten (Nutzeraccounts) mit bekannten Passwörtern**
  - **Plug-In: Users and Passwords**

Statische Code-Analyse mit FACT (3)

- **Letztes Update der Firmware**
  - **Veröffentlichungsdatum**
- **Linux Version + Anzahl der CVE Einträge für diese Linux Version**
  - **Plug-Ins: CVE Lookup, CWE Checker, Software Components, Known Vulnerabilities**
- **Analyse der Exploit Mitigations (PIE, RELRO, NX, FORTIFY_SOURCE, Stack Canary)**
  - **Plug-In: Exploit Mitigations**
- **Privates kryptographisches Material**
  - **Plug-In: Crypto Material**
- **Hartkodierte Login-Daten (Nutzeraccounts) mit bekannten Passwörtern**
  - **Plug-In: Users and Passwords**

Statische Code-Analyse mit FACT (3)

- **Letztes Update der Firmware**
  - **Veröffentlichungsdatum**
- **Linux Version + Anzahl der CVE Einträge für diese Linux Version**
  - **Plug-Ins: CVE Lookup, CWE Checker, Software Components, Known Vulnerabilities**
- **Analyse der Exploit Mitigations (PIE, RELRO, NX, FORTIFY_SOURCE, Stack Canary)**
  - **Plug-In: Exploit Mitigations**
- **Privates kryptographisches Material**
  - **Plug-In: Crypto Material**
- **Hartkodierte Login-Daten (Nutzeraccounts) mit bekannten Passwörtern**
  - **Plug-In: Users and Passwords**

Agenda

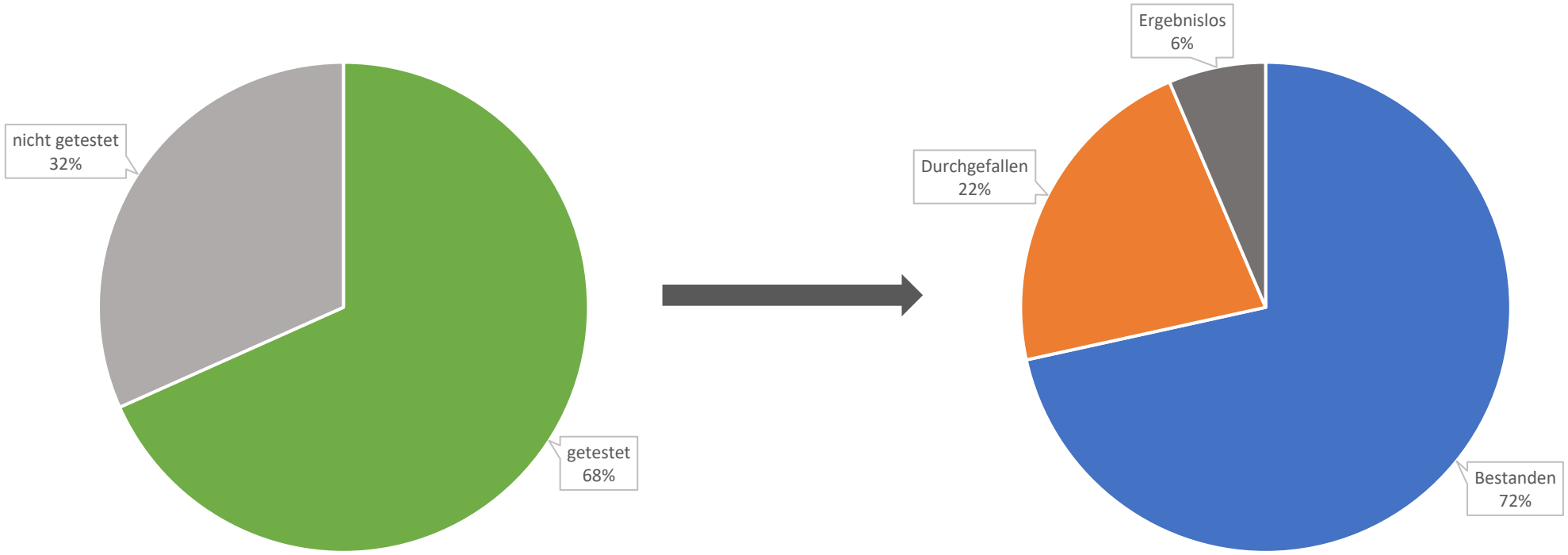
- Ziele der Arbeit
- Verwandte Arbeiten
- Methodik
  - Testumgebung und Rahmenbedingungen
  - Technische Richtlinie
  - Statische Code-Analyse mit FACT
- **Ergebnisse**
  - **TR-Konformität von OpenWrt**
  - Ergebnisse der Code-Analyse
- Limitationen und zukünftige Forschung

TR-Konformität von OpenWrt (1)

Nicht anwendbare Testfälle:

- **Module K – Remote Configuration**
- **Module L – Voice over IP**
- **Module M – Virtual Private Network**
- **WPS Funktionalität**
- **Automatische Updates und Push Benachrichtigungen**

TR-Konformität von OpenWrt (2)



TR-Konformität von OpenWrt (3)

LAN Interface

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     Dropbear sshd (protocol 2.0)
53/tcp open  domain  Cloudflare public DNS
80/tcp open  http
```

TCP

TR-Konformität von OpenWrt (3)

LAN Interface

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     Dropbear sshd (protocol 2.0)
53/tcp open  domain  Cloudflare public DNS
80/tcp open  http
```

TCP

```
PORT     STATE SERVICE VERSION
53/udp open  domain  Cloudflare public DNS
```

UDP

TR-Konformität von OpenWrt (3)

LAN Interface

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     Dropbear sshd (protocol 2.0)
53/tcp open  domain  Cloudflare public DNS
80/tcp open  http
```

TCP

```
PORT     STATE SERVICE VERSION
53/udp open  domain  Cloudflare public DNS
```

UDP

WAN Interface

All 65535 scanned ports on OpenWrt.fritz.box (192.168.178.115) are closed (65494) or filtered (41)

TCP

TR-Konformität von OpenWrt (3)

LAN Interface

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     Dropbear sshd (protocol 2.0)
53/tcp open  domain  Cloudflare public DNS
80/tcp open  http
```

TCP

```
PORT     STATE SERVICE VERSION
53/udp open  domain  Cloudflare public DNS
```

UDP

WAN Interface

All 65535 scanned ports on OpenWrt.fritz.box (192.168.178.115) are closed (65494) or filtered (41)

TCP

All 65535 scanned ports on OpenWrt.fritz.box (192.168.178.115) are open|filtered (56258) or closed (9277)

UDP

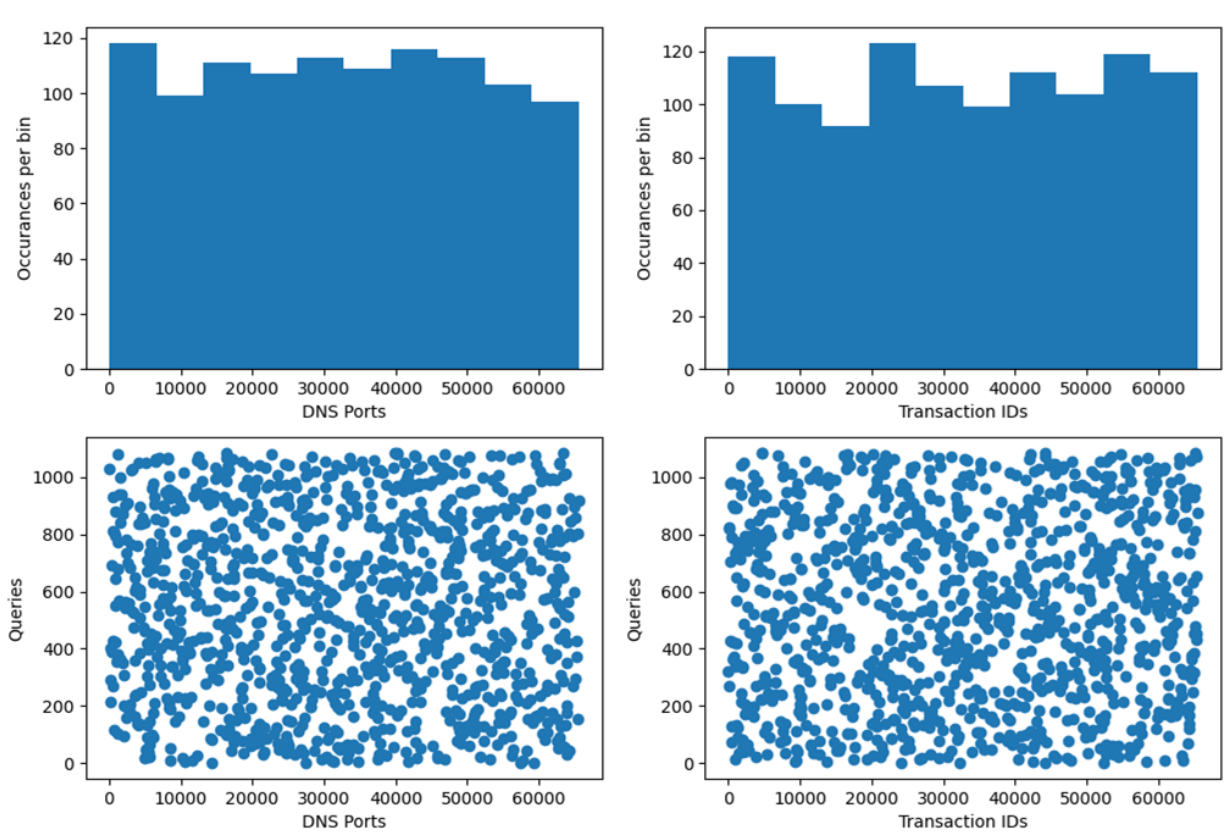
TR-Konformität von OpenWrt (4)

Test results for DNS port randomization:

Number of samples: 1086
Number of unique ports: 1086
Range: 61 - 65508
Standard Deviation: 18668.148912615263

Test results for transaction ID randomization:

Number of samples: 1086
Number of unique ports: 1037
Range: 45 - 65415
Standard Deviation: 19128.480563438716



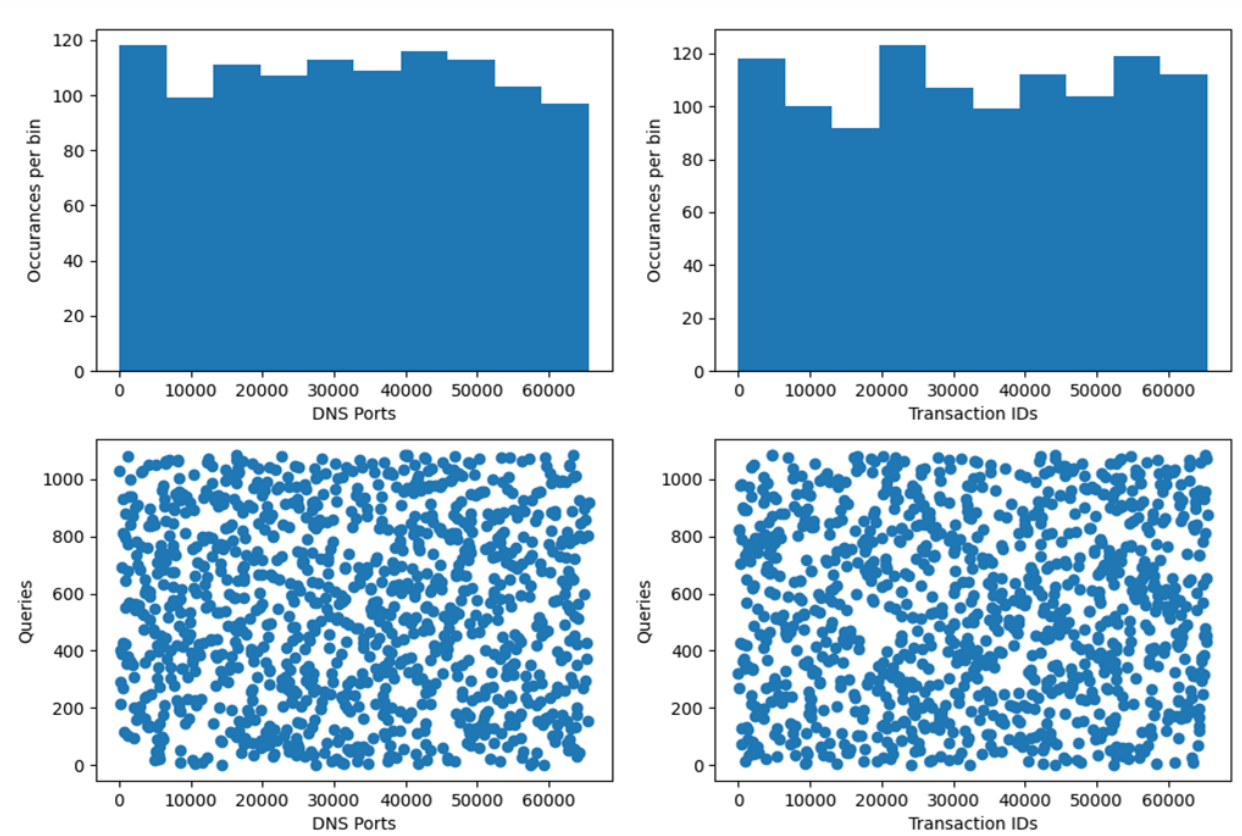
TR-Konformität von OpenWrt (4)

Test results for DNS port randomization:

Number of samples: 1086
Number of unique ports: 1086
Range: 61 - 65508
Standard Deviation: 18668.148912615263

Test results for transaction ID randomization:

Number of samples: 1086
Number of unique ports: 1037
Range: 45 - 65415
Standard Deviation: 19128.480563438716



Kolmogorow-Smirnow-Test

Source Port Randomization: statistic = 0.032, p = 0.626
Transaction ID Randomization: statistic = 0.028, p = 0.802

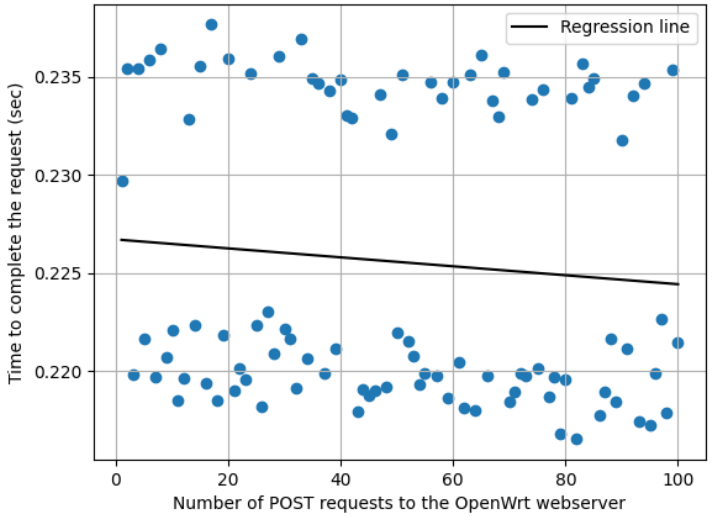
TR-Konformität von OpenWrt (5)

Notwendige Änderungen zum Bestehen der TR

- In den meisten Fällen mit wenig Aufwand verbunden
- Vor allem „MUST“ Kriterien betrachtet
- Nur einige „SHOULD“ Kriterien behandelt

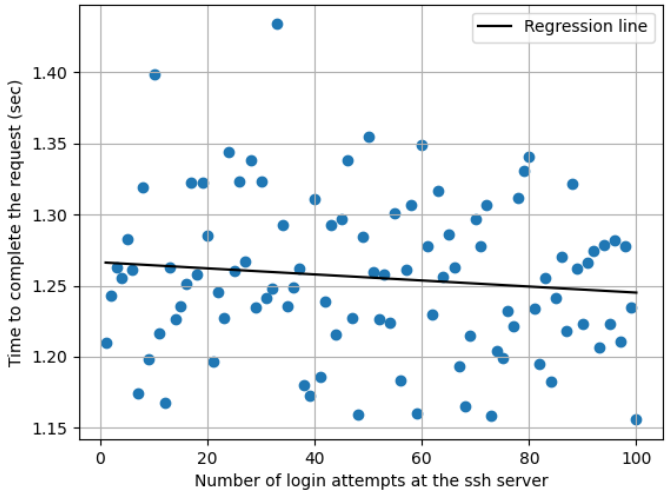
- Aufgrund der Natur von OpenWrt werden einige Testfälle auch in Zukunft fehlschlagen

TR-Konformität von OpenWrt (6)




```
Mean : 0.226
Median : 0.222
Regression coefficient :  -0.08936 (p = 0.377)
Standard error : 0.0
```

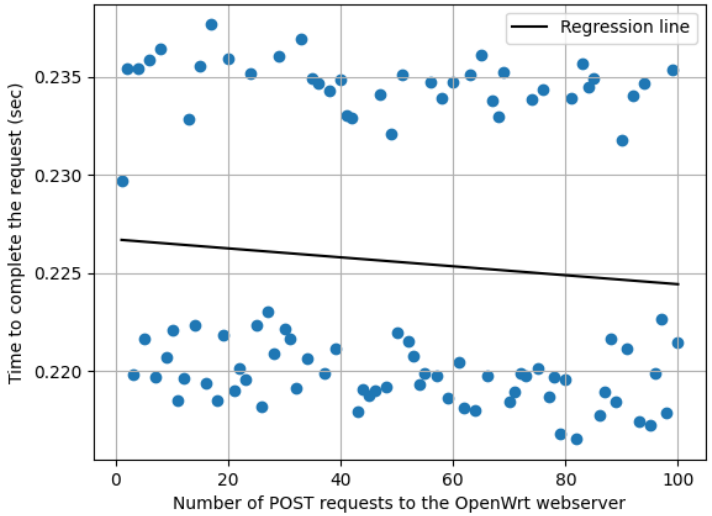
Webserver




```
Mean : 1.256
Median : 1.256
Regression coefficient :  -0.11312 (p = 0.262)
Standard error : 0.0
```

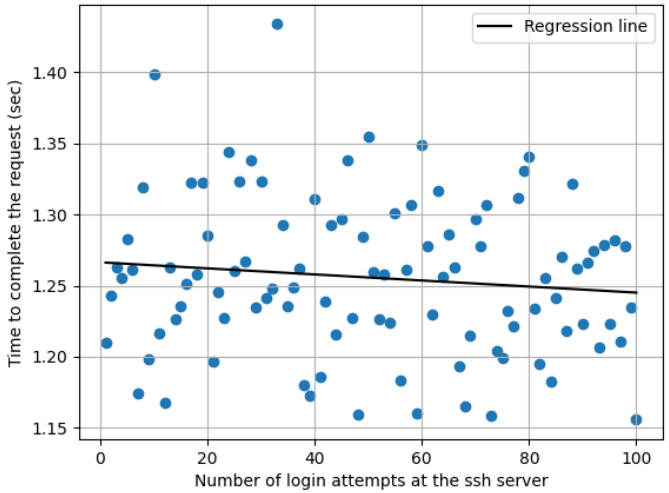
SSH Server

TR-Konformität von OpenWrt (6)




```
Mean : 0.226
Median : 0.222
Regression coefficient :  -0.08936 (p = 0.377)
Standard error : 0.0
```

Webserver




```
Mean : 1.256
Median : 1.256
Regression coefficient :  -0.11312 (p = 0.262)
Standard error : 0.0
```

SSH Server

Lösung:

- Fehlerzähler -> Login-Sperre nach x Fehlversuchen

TR-Konformität von OpenWrt (7)

Notwendige Änderungen zum Bestehen der TR - TR.D.2

"Access to the configuration of the DUT **MUST** at least be secured by a password in the initialized and customized state. The DUT **MAY** offer a higher level of security by providing alternative authentication mechanisms." [19]

- Man kann das DUT ohne Passwort nutzen
- Nur ein Passwort für alle Authentifizierungsmethoden ►LuCI Passwort = SSH Passwort = User Passwort
- Der Nutzer ist immer "root" Nutzer ► passwd setzte keine Restriktionen ein
- Man kann das Passwort ohne Eingabe des vorherigen Passwortes löschen

Lösung: Einführen eines dedizierten Nutzeraccounts mit der Möglichkeit das Programm „sudo" (o.Ä.) zu nutzen

- Würde auch TR.D.10 und TR.D.15 beheben

TR-Konformität von OpenWrt (8)

Notwendige Änderungen zum Bestehen der TR - TR.E.5 bis TR.E.8

- Authentizität eines Firmware Updates muss automatisch geprüft werden
- Sollte auf digitalen Signaturen beruhen
- Nicht signierte Firmware darf nicht automatisch installiert werden -> Vorher muss dem Nutzer eine Warnung gezeigt werden

Lösungen:

- Einsatz von digitalen Signaturen möglich -> gleiches (ähnliches) Prüfverfahren wie bei opkg Paketen [20]
    - usign Ed25519 Signaturen (elliptische Kurve)

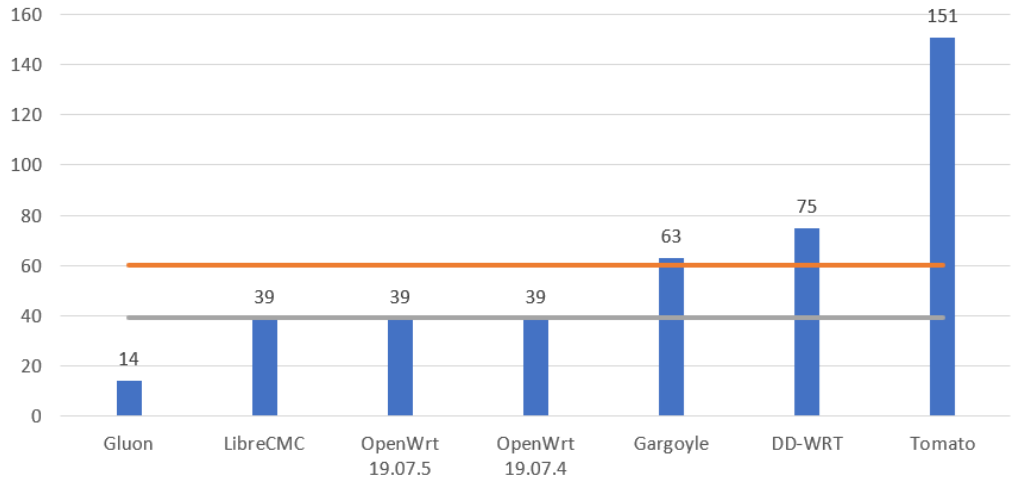
- Automatischer Download der digital signierten sha256sums Datei sowie der sha256sums.asc Datei
- Automatische Prüfung der Signatur
- Automatische Prüfung der Firmware

Agenda

- Ziele der Arbeit
- Verwandte Arbeiten
- Methodik
  - Testumgebung und Rahmenbedingungen
  - Technische Richtlinie
  - Statische Code-Analyse mit FACT
- **Ergebnisse**
  - TR-Konformität von OpenWrt
  - **Ergebnisse der Code-Analyse**
- Limitationen und zukünftige Forschung

Ergebnisse der Code-Analyse (1)

Betriebssystem und CVE Einträge



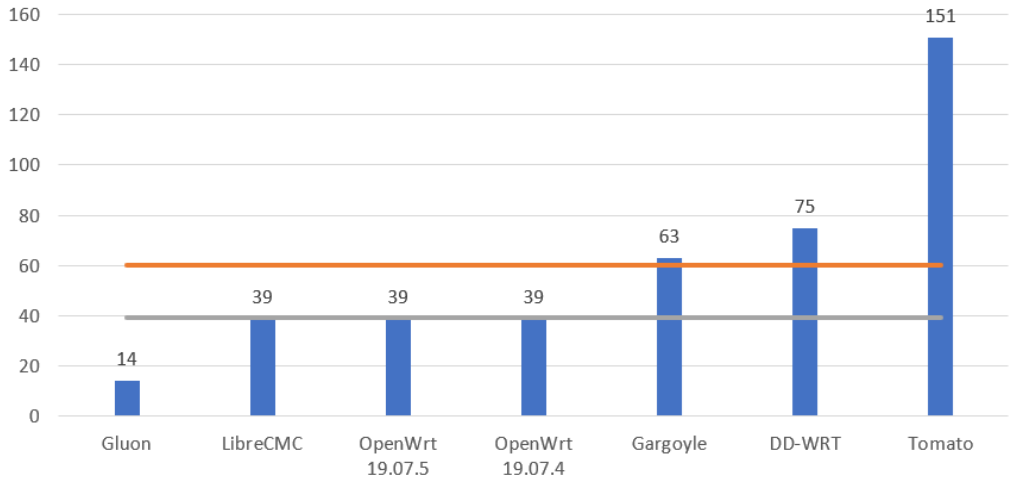
[21]

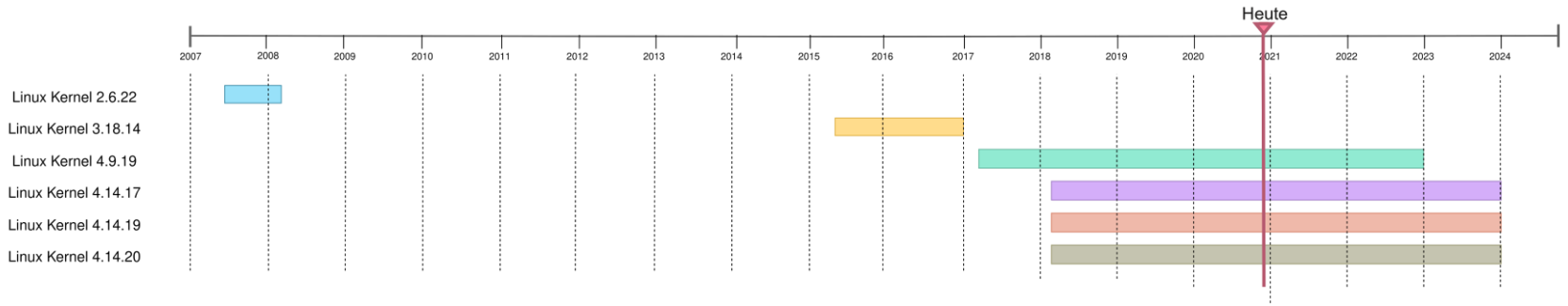
Figure 3.3: Number of Critical Severity CVEs in Linux Kernel per Firmware Image

Figure 3.4: Number of High Severity CVEs in Linux Kernel per Firmware Image

Ergebnisse der Code-Analyse (1)

Betriebssystem und CVE Einträge





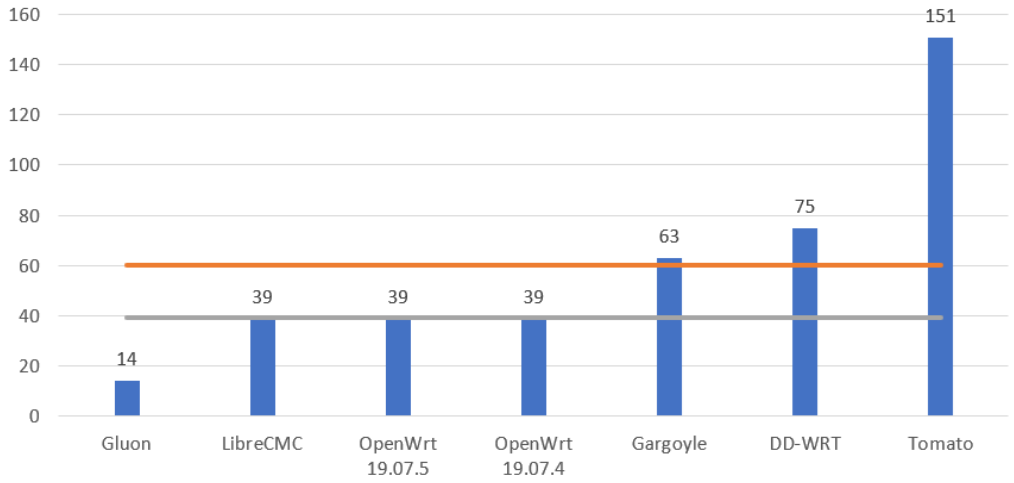
[21]

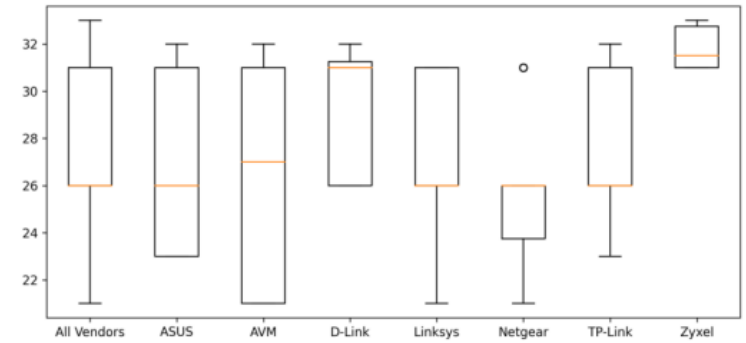
Figure 3.3: Number of Critical Severity CVEs in Linux Kernel per Firmware Image

Figure 3.4: Number of High Severity CVEs in Linux Kernel per Firmware Image

Ergebnisse der Code-Analyse (1)

Betriebssystem und CVE Einträge





[21]

Figure 3.3: Number of Critical Severity CVEs in Linux Kernel per Firmware Image

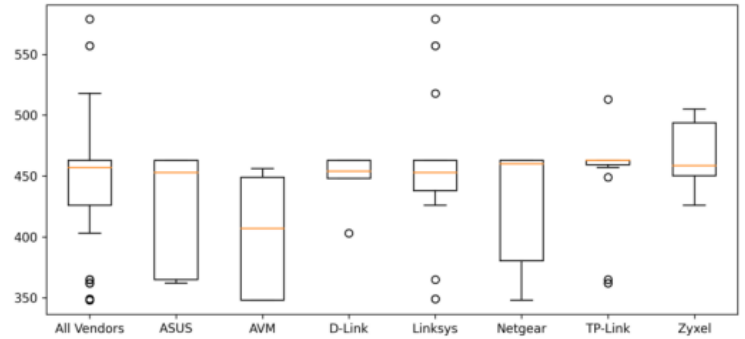
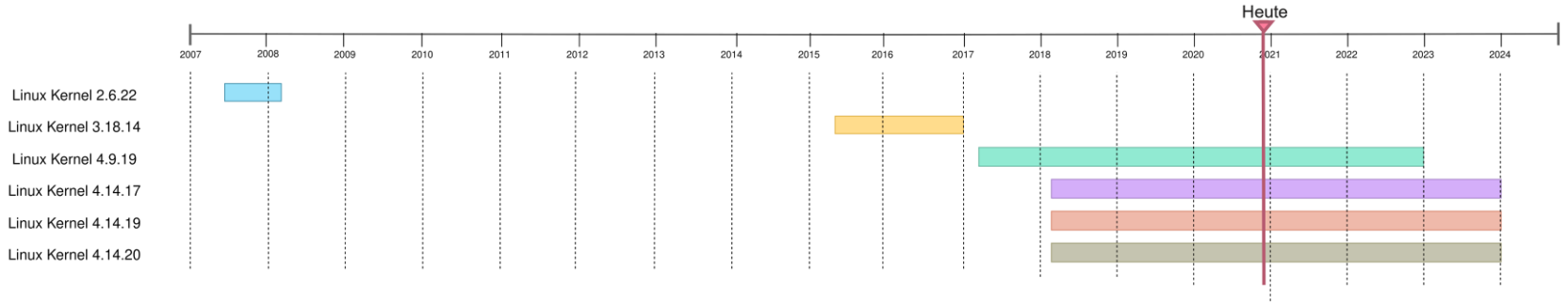
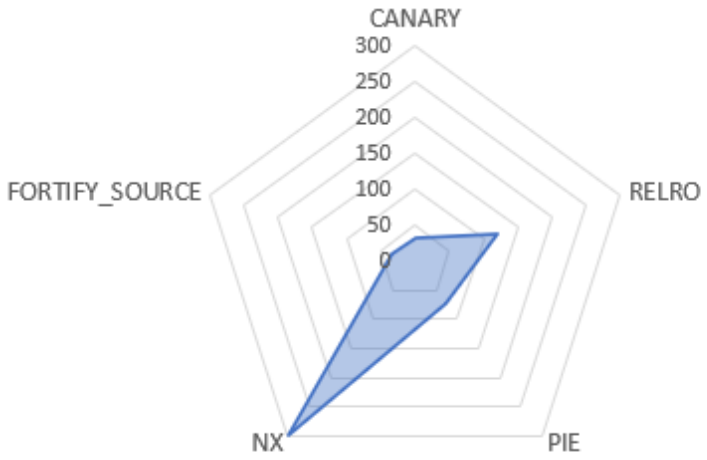


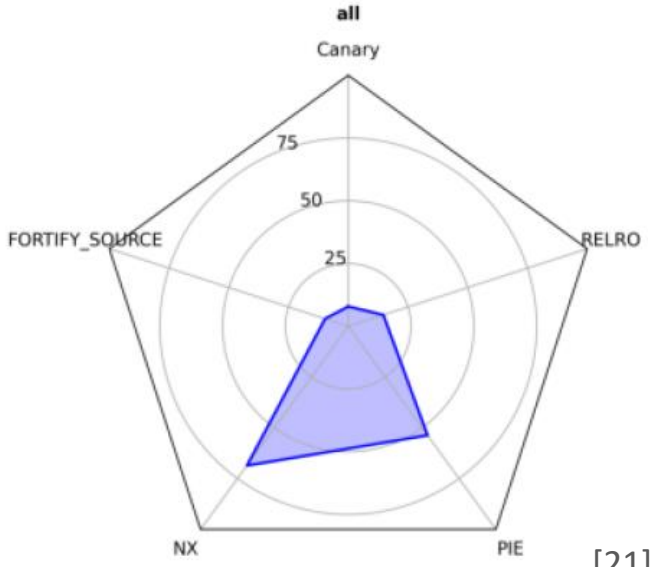
Figure 3.4: Number of High Severity CVEs in Linux Kernel per Firmware Image



Ergebnisse der Code-Analyse (2)

Exploit Mitigations





[21]

Ergebnisse der Code-Analyse (3)

Private Key Material

- **DD-WRT und Gargoyle jeweils 2 krypographische Schlüssel**

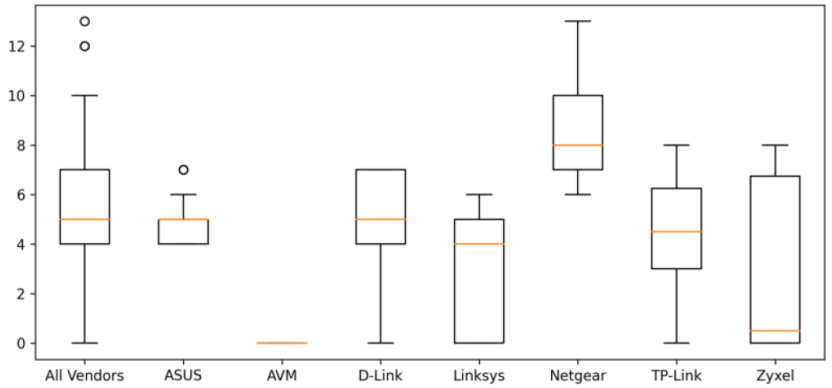
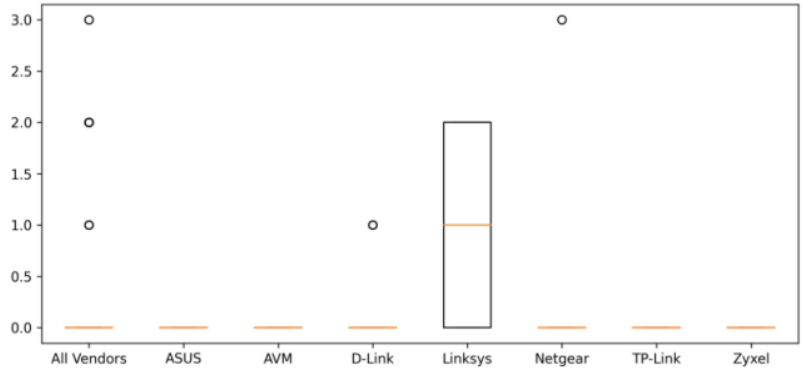


Figure 3.12: Number of Private Keys per Firmware Image [21]

Hard-coded Login Credentials

- **Ein Benutzeraccount mit Passwort bei Gargoyle**



[21]

Figure 3.14: Number of Well Known Hard-coded Passwords per Firmware Image

Agenda

- Ziele der Arbeit
- Verwandte Arbeiten
- Methodik
  - Testumgebung und Rahmenbedingungen
  - Technische Richtlinie
  - Statische Code-Analyse mit FACT
- Ergebnisse
  - TR-Konformität von OpenWrt
  - Ergebnisse der Code-Analyse
- **Limitationen und zukünftige Forschung**

Limitationen und zukünftige Forschung

Limitationen

- Nativer Internetanschluss vs. double NAT -> weniger fehleranfällig und einfachere Konfiguration
- Zu wenige Systeme für einige Testanforderungen
- Conformance Statement wurde vom Tester ausgefüllt -> ggf. Voreingenommenheit (confirmation bias)
- Keine wirkliche Vergleichbarkeit von CVE Analyse

Limitationen und zukünftige Forschung

Zukünftige Forschung

- Vergleiche der TR Ergebnisse von Open Source Projekten mit Closed Source Router Firmware
- Vollständige Durchführung aller Module der TR bei OpenWrt mit vollem Funktionsumfang von OpenWrt (mit zusätzlichen Paketen aus dem Paketmanager)
- Ausführlichere statische Code-Analyse ggf. mit eigenen Plug-Ins und größerem Korpus
- Entwicklung einer TR-konformen OpenWrt Version (z.B. mit dem OpenWrt Image Builder)

Fragen

FACT Impressionen

Literaturverzeichnis

[1] A. P. Ortega, X. E. Marcos, L. D. Chiang, and C. L. Abad, “Preventing ARP cache poisoning attacks: A proof of concept using OpenWrt,” in 2009 Latin American Network Operations and Management Symposium, pp. 1–9, IEEE, 19.10.2009 - 21.10.2009. [Abgerufen am: 08.11.2020].
[2] C. E. Palazzi, M. Brunati, and M. Roccetti, “An OpenWRT solution for future wireless homes,” in 2010 IEEE International Conference on Multimedia and Expo, pp. 1701–1706, IEEE, 19.07.2010 - 23.07.2010. [Abgerufen am: 08.11.2020].
[3] Andrew McDonnell, “Evaluating the security of OpenWRT.” https://blog.oldcomputerjunk.net/2014/evaluating-the-security-of-openwrt-part-1/, 2014. [Abgerufen am: 08.11.2020].
[4] Linus Torvalds, “Linux—a free unix-386 kernel.” https://tech-insider.org/linux/research/acrobat/911010.pdf, 1991. [Abgerufen am: 08.11.2020].
[5] G. K.-H. Jonathan Corbet, “Linux Kernel Development: How Fast It is Going, Who is Doing It, What They Are Doing and Who is Sponsoring the Work.” https://www.linuxfoundation.org/wp-content/uploads/linux-kernel-report-2016.pdf, 2016. [Abgerufen am: 11.11.2020].
[6] M. Jimenez, M. Papadakis, and Y. Le Traon, “An Empirical Analysis of Vulnerabilities in OpenSSL and the Linux Kernel,” in 2016 23rd Asia-Pacific Software Engineering Conference (APSEC), pp. 105–112, IEEE, 06.12.2016 - 09.12.2016. [Abgerufen am:08.11.2020].
[7] J. Viega, M. Messier, and P. Chandra, Network Security with OpenSSL: Cryptography for Secure Communications. Sebastopol: O’Reilly Media Inc, 2009.
[8] OpenWrt Webseite, “Techdata: TP-Link Archer C7 AC1750 v5.” https://openwrt.org/toh/hwdata/tp-link/tp-link_archer_c7_v5. [Abgerufen am: 03.02.2021].
[9] TP-Link Corporation Limited. „AC1750-Dualband-Gigabit-WLAN-Router“https://www.tp-link.com/de/home-networking/wifi-router/archer-c7/ [Abgerufen am: 03.02.2021].
[10] OpenWrt Webseite, “OpenWrt Download Statistik November 2020,” 29.11.2020. [Abgerufen am: 30.11.2020].
[11] OpenWrt Webseite, “OpenWrt Version History.” https://openwrt.org/about/history, 13.12.2020. [Abgerufen am: 28.10.2020].
[12] G. Lyon, nmap network scanning. Sunnyvale, CA: Insecure.Com LLC, 2008.
[13] Thomas d’Otreppe de Bouvette. https://github.com/aircrack-ng/aircrack-ng, 25.01.2020. [Abgerufen am: 25.11.2020].
[14] C. Sanders, Practical packet analysis: Using Wireshark to solve real-world network problems. San Francisco: No Starch Press, 3rd edition ed., 2017.
[15] D. Wetter, “testssl.sh.” https://testssl.sh, 2020. [Abgerufen am: 26.11.2020].
[16] R. M. Gérald Doussot, “State of DNS Rebinding: Attack & Prevention Techniques and the Singularity of Origin.” https://docs.google.com/presentation/d/1O7MxvbIfRcPSlbyZbFxD-fAR34XlquQSlRAHPb2kR4E/edit#slide=id.p, 2019. [Abgerufen am: 02.12.2020].
[17] Fraunhofer FKIE, “FACT Core.” https://github.com/fkie-cad/FACT_core, 2020. [Abgerufen am: 26.10.2020].
[18] F. J. M. Jr., “The kolmogorov-smirnov test for goodness of fit,” Journal of the American Statistical Association, vol. 46, no. 253, pp. 68–78, 1951. [Abgerufen am: 15.01.2021].

Literaturverzeichnis (2)

[19] Bundesamt für Sicherheit in der Informationstechnik, “BSI TR-03148:Secure Broadband Router: Requirements for secure Broadband Routers.” https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03148/TR03148.pdf?__blob=publicationFile&v=3, 2020. [Abgerufen am: 26.10.2020].
[20] OpenWrt Webseite, “Release Signing.” https://openwrt.org/docs/guide-user/security/release_signatures, 2019. [Abgerufen am: 05.01.2021].
[21] Peter Weidenbach, Johannes vom Dorp, “Home Router Security Report 2020.” https://www.fkie.fraunhofer.de/content/dam/fkie/de/documents/HomeRouter/HomeRouterSecurity_2020_Bericht.pdf, 2020. [Abgerufen am: 27.10.2020].