

# RISC-V

## Präsentation des Softwareprojekts

**Felix Manuel Peterka    Niklas Pauli    Niclas Schwarzlose**

19.01.2021

- ▶ Felix Manuel Peterka (fptk@zedat.fu-berlin.de)
- ▶ Niklas Pauli (niklap97@zedat.fu-berlin.de)
- ▶ Niclas Schwarzlose (nischw@zedat.fu-berlin.de)

GitLab-Repo: <https://git.imp.fu-berlin.de/nischw/risc-v-wifi>

ReportingPad: <https://git.imp.fu-berlin.de/nischw/risc-v-wifi/-/boards>

Erhaltene Hardware:

- ▶ 4x SiFive HiFive1 Rev B
- ▶ 1x Xilinx Arty 7
- ▶ 1x ARM-USB-TINY-H Debugger
- ▶ Einige Kabel

- ▶ Erinnerung: Was ist unser Thema
- ▶ Unsere Vision des Endprodukts
- ▶ Meilensteine
- ▶ Zeitplan
- ▶ Dokumentation
- ▶ Stand: Arty Implementierung
- ▶ Stand: HiFive (mit openMZ)
- ▶ Stand: HiFive (ohne Multizone)

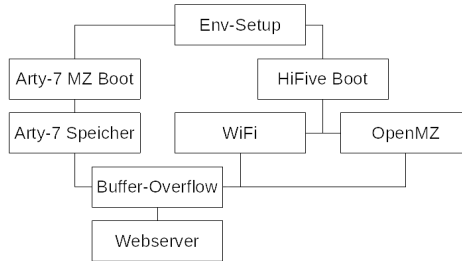
## Erinnerung: Was ist unser Thema

---

- ▶ Multizone erlaubt verschiedene Softwaremodule mithilfe von Hardware zu trennen
- ▶ Wir wollen RISC-V Hardware verwenden um den Nutzen von Multizone aufzuzeigen
- ▶ Wir wollen einen Angriff entwickeln, der normalerweise funktioniert, jedoch durch Multizone gestoppt werden kann
  - ▶ Dieser soll in ein realistisches Szenario eingebettet werden
- ▶ Testen mit unterschiedlicher Hardware (SiFive HiFive1 Rev B & Xilinx Arty 7 FPGA)

- ▶ Sensible Daten (z.B. Passwort) liegen irgendwo im Speicher.
- ▶ Parallel läuft ein HTTP-Server mit einer Schwachstelle (z.B. buffer overflow).
- ▶ Ein Angriff auf eine Variante ohne Multizone kann das Passwort auslesen.
- ▶ Die Variante mit Multizone verhindert das Auslesen des Passworts durch einen Angriff.

## ► Aktuelle Meilensteine



- ▶ Über Issues verfeinert
- ▶ 08.11.2020: Meilensteine angelegt, Zeitplan erstellt, Arbeit an Zielen bis 2020 begonnen
- ▶ 19.01.2021: Funktionierender Overflow, mindestens erster Server-Entwurf
  - ▶ Wenn bis 31.12.2020 OpenMZ nicht funktioniert, dann abbrechen

- ▶ 02.02.2021:
  - ▶ Zwischenstand Angriff - wie weit mit Server-Client-Szenario? Erweiterung möglich?
  - ▶ Dokumentation vollständig
  - ▶ Getting started guide vollständig
  - ▶ Abschlusspräsentation beginnen
- ▶ 02.03.2021: Abschlusspräsentation
- ▶ 29-31.03.2021 Übergabe des Repos
- ▶ 31.03.-15.04.2021 Abgabe der Hardware (inkl. Lieferschein)



- ▶ Hauptsächlich über Wiki und Code
- ▶ Je ein Kapitel pro “Projekt”
  - ▶ HiFive + WiFi
  - ▶ Arty + Multizone
  - ▶ Arty + lwIP
  - ▶ HiFive + openMZ
  - ▶ Angriffsszenario
  - ▶ Troubleshooting / Aufgetretene Probleme

## Stand: Arty Implementierung

- ▶ 2 Zonen erstellt
  - ▶ Zonen können anhand von Speicheradressen definiert werden
  - ▶ Zugriffsrechte können per Zone/ Speicherbereich gesetzt werden
- ▶ Webserver ließ sich nicht direkt übertragen
  - ▶ Bearbeitung auf Grundlage der Beispielanwendung aus multizone-iot-sdk
  - ▶ Nutzung des Netzwerk-Interface-Treibers von HexFive
  - ▶ Nutzung des lwIP Stacks
  - ▶ Nutzung des selben HTTP-Parsers wie auf dem HiFive1
- ▶ Dokumentation hängt etwas hinterher
  - ▶ Sobald Implementierung erfolgt, Doku im Wiki anhand Code-Snipptes
  - ▶ Zusätzlich Dokumentation der genutzten lwIP Anteile und des zonings

## Stand: HiFive (mit openMZ)

- ▶ OpenMZ:
  - ▶ durch eine Masterarbeit entstandene Implementierung der Multizone sdk für das Hifive1 rev b
  - ▶ schützt den Speicher der Zonen vor direktem Zugriff mit dem PMP-Mechanismus von RISC-V (wie Multizone)
  - ▶ deutlich schlanker als Multizone
- ▶ OpenMZ auf dem Hivefive1 rev b
  - ▶ Anfangs Probleme mit Projekt Importierung und hex-file-Erstellung
  - ▶ OpenMZ läuft auf dem Hifive1 rev b
- ▶ OpenMZ mit der Wifi-Funktion Erweiterung
  - ▶ Zone X soll Verbindung zum Webserver aufbauen
  - ▶ Zone Y soll nicht einsehbar sein

## Stand: HiFive (ohne Multizone)

---

- ▶ Die Verbindung zwischen dem SiFive HiFive1 Rev B und dem WiFi-Router ist möglich.
- ▶ Ein einfacher Webserver wurde entwickelt und ist funktionstüchtig.
- ▶ Eine vulnerable Anwendung für diesen Webserver wurde entwickelt.
- ▶ Ein passender Angriff fehlt noch.
- ▶ Das Angriffsziel (z.B. Passwort) fehlt auch noch.