

RISC-V

Präsentation des Softwareprojekts

Felix Manuel Peterka Niklas Pauli Niclas Schwarzlose

08.12.2020

- ▶ 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

- ▶ Einführung in das Thema
- ▶ Vision
- ▶ Projektplanung
- ▶ Tools und Frameworks
- ▶ Struktur der Teamarbeit
- ▶ Dokumentation des Arbeitsergebnisses

Was ist RISC-V?

Eine Open Source Instruction Set Architecture (ISA).

Warum RISC-V?

- ▶ Flexibilität (verschiedene Hardwareimplementierungen mit unterschiedlichen Zielen)
- ▶ Stabilität (ISA bleibt garantiert gleich)
- ▶ Vertrauen (verschiedene Open Source Implementierungen)

Mehr Infos: <https://riscv.org/why-risc-v/>

- ▶ Ermöglicht das Unterteilen von Software in verschiedene “Zonen”
- ▶ Die Zonen sind Hardware-seitig voreinander geschützt
 - ▶ Braucht deshalb Hardwareunterstützung (Usermode)
- ▶ Open Source API

Mehr Infos: <https://hex-five.com/>

Ziel: Wir wollen den Nutzen von Multizone zeigen.

- ▶ Entwicklung eines Angriffs, der normalerweise funktioniert, jedoch durch Multizone gestoppt werden kann.
- ▶ Einbettung in ein realistisches Szenario.

- ▶ 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.

Umfang des Themas

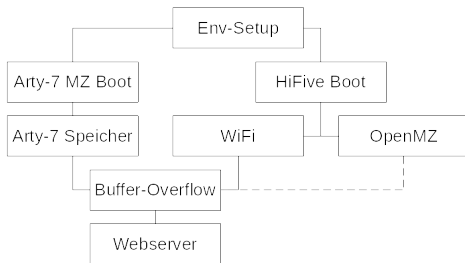
- ▶ Entwicklung der Software für das RISC-V Board
- ▶ Verwendung von Multizone
 - ▶ Eventuell mit verschiedenen Implementierungen
- ▶ Ausarbeitung des Angriffs
- ▶ Testen des Angriffs auf beide Varianten

Was nicht dazu gehört:

- ▶ Keine Hardwareanpassungen
- ▶ Kein Angriff auf Hardwareebene
- ▶ Keine Schwachstellenanalyse



► 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)

- ▶ Funktionierender Angriff
 - ▶ Speicherzugriffsüberschreitung
 - ▶ Server/ Netzbasiert (IoT Szenario)
 - ▶ Gegensatz zwischen MZ und nicht-MZ demonstrieren
- ▶ Implementierungsziele:
 - ▶ Bufferoverflowszenario auf Arty-7 (MZ) und HiFive
 - ▶ Server-/Client-basiertes Szenario
 - ▶ Getting Started Guide
 - ▶ Dokumentation via Wiki

Tools und Frameworks

- ▶ Risc-V Toolchain (inkl. gcc, make)
- ▶ Xilinx Vivado WebPack (Für das flashen des Artys)
- ▶ Vim, GitLab-WebIDE, Freedom-Studio
- ▶ GitLab Wiki für Dokumentation
- ▶ GitLab Repo/ Readme für Getting Started Guide
- ▶ GitLab Issues für Zeitplan, Milestones und ReportingPad (Kommentierfunktion, DueDates etc.)
- ▶ Signal, Webex für Kommunikation

- ▶ mindestens 1 Mal pro Woche:
 - ▶ Austauschen über den aktuellen Stand
 - ▶ Probleme und Erfolge werden genannt und beschrieben
 - ▶ Wissensabgleich (Überblick über die Arbeitsthemen der Anderen)
- ▶ durchgehende Kommunikation (per Signal-chatroom):
 - ▶ Austausch über Probleme und Unverständlichkeiten, die die weitere Arbeit verhindern
 - ▶ kurzfristige Meetings
 - ▶ => schnelle Lösungsfindung

Struktur der Teamarbeit (2)

- ▶ Aufteilung der Arbeitsthemen:
 - ▶ flexible Aufteilung der Themen
 - ▶ Mix aus- “was getan werden muss” und “was ich gerne machen möchte”
 - ▶ Jeder arbeitet an Implementierung und der dazugehörigen Dokumentation
 - ▶ Zwei arbeiten mit dem Hifive
 - ▶ hifive + wlan
 - ▶ hifive + evt. Multizone
 - ▶ Einer arbeitet mit dem Arty FPGA + Multizone
- ▶ Ausschließung von Überschneidungen:
 - ▶ regelmäßige und gründliche Kommunikation
 - ▶ => jeder weiß genau was die Anderen machen
- ▶ Überschneidungen:
 - ▶ Hifive1 rev b
 - ▶ Set up des dev boards
 - ▶ Freedom SDK/ FreedomStudio

- ▶ Dokumentation im Gitlab
- ▶ Geplant: jeweils eigene Dokumentation der 3 Varianten
- ▶ Gewährleistung der Qualitätssicherung und Verständlichkeit durch Teammitglieder
- ▶ Dokumentation möglichst begleitend zur Implementierung
- ▶ Momentan: getting Started für das Hifive and Arty FPGA + Multizone