LibMatch Binary Ninja Plugin

David Jäggli

01. July 2025

Business project at Lucerne University of Applied Sciences and Arts - Computer Science

Title: Implement LibMatch Concepts as a Binary Ninja Plugin.
Student: Jäggli David
Program: BSc Computer Science, Cyber Security
Year: 2025
Supervisor: Eskhita Radwan
Expert: TBD
Client: armasuisse W+T
Classification of work: Public

Sworn declaration:

I hereby declare that I have prepared this thesis independently and without unauthorized external assistance, have cited all sources, literature and other aids used, have marked passages taken verbatim or in terms of content as such, will protect the confidentiality interests of the client and will respect the copyright regulations of the Lucerne University of Applied Sciences and Arts.

Place / Date, Signature: Rotkreuz, 19.07.2025 

I. Abstract

This paper presents the development of a Metasploit Framework (MSF) shellcode encoder module tailored for the ARM64 (AArch64) architecture. ARM64, which is widely used in modern IoT and embedded systems, previously lacked a dedicated encoder within the MSF toolkit. Shellcode encoding is essential for bypassing constraints such as NULL bytes in vulnerable applications. The primary objective was to implement an encoder that produces NULL-byte free payloads while preserving their functionality, alongside a decoder that reconstructs the payload at runtime.

A systematic literature review and agile development methods were used to explore state-of-the-art encoding and decoding techniques, including adaptation of existing frameworks and the unique requirements of AArch64. Challenges such as instruction set limitations, 64-bit address handling and payload size scalability were addressed. The research incorporated tools such as QEMU for testing and debugging in emulated environments, and used Python and Rust for automation and mapping of valid AArch64 assembly instructions.

Key contributions include

- Development of an encoder module integrated with MSF that converts payloads into NULL-byte free representations.
- Implementation of a decoder module that supports in-place decoding, reducing the risk of stack-related segmentation errors.
- Exploration of encoding polymorphism and extended character sets, improving evasion capabilities, although limitations such as address handling restricted full implementation of these features.

The resulting encoder supports various ARM64 payloads in MSF and meets the acceptance criteria for official repository integration. Future work could focus on polymorphic encoding and the development of a fully printable ASCII decoder for wider applicability. This work lays the foundation for improved exploitation tools in cyber security research and practical penetration testing.

II. Table of Contents

1 Problem, Question, Vision

2 State of Research

3 Ideas and Concepts

4 Methods

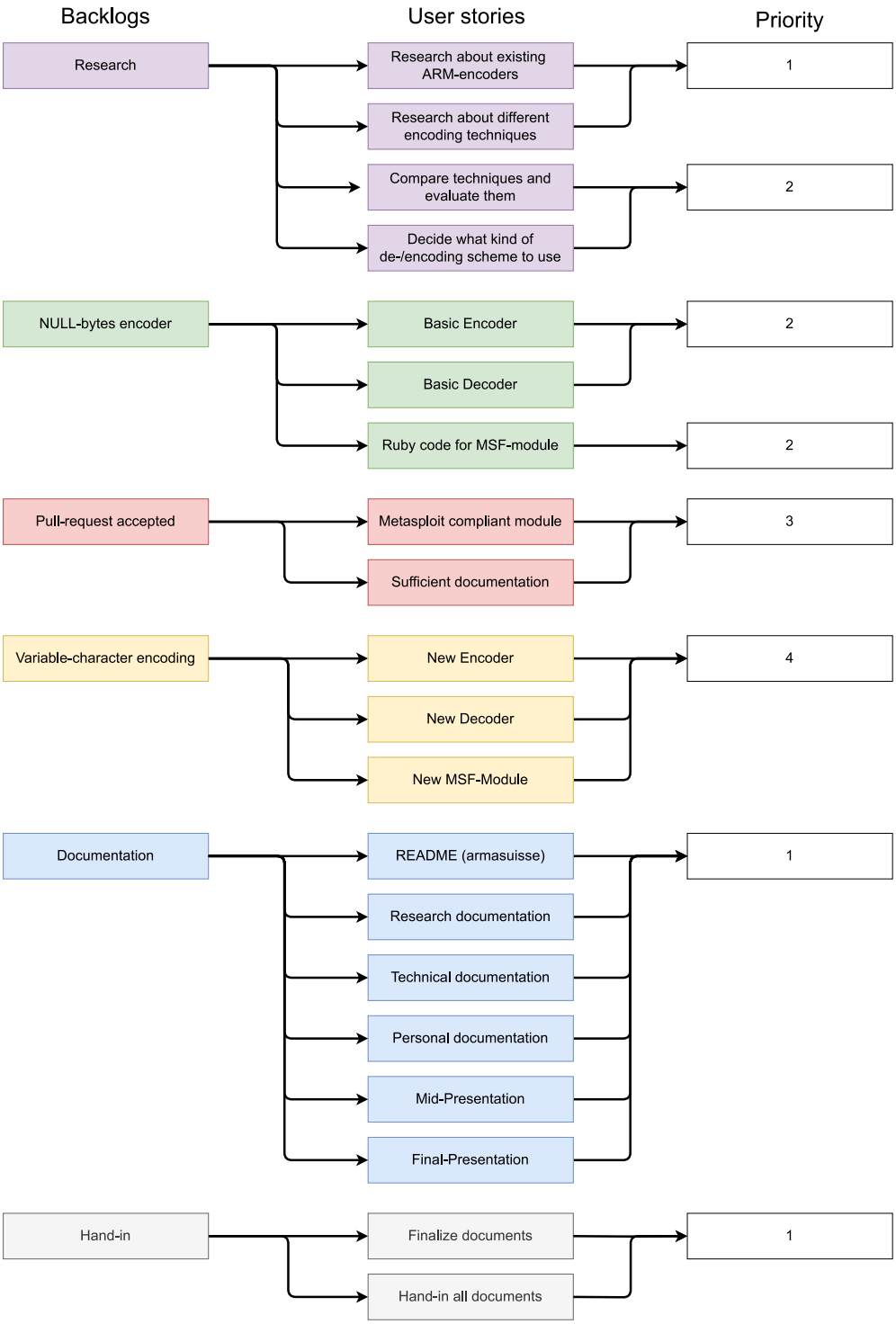
5 Implementation

arisntioaensrtienrst

6 Evaluation and Validation

7 Further Ideas

A. Backlogs and User Stories



B. Task definition

List of Abbreviations

Abbreviation	Definition
BAA	Bachelordiplomarbeit
CPU	Central Processing Unit
ASCII	American Standard Code for Information Interchange
GDB	GNU Project Debugger
LLM	Large Language Model
IO	Input/Output
SSH	Secure Shell
API	Application Programming Interface

Table 1: List of Abbreviations

References

Index of Figures

Index of Tables