Stefan Mladić

Belgrade, Serbia | stefan@mladicstefan.com | github.com/mladicstefan | mladicstefan.com

About Me

I approach technology with a fundamentally curious mindset. In equal parts, I love building things from first principles and breaking things because breaking things is like solving a puzzle. I'm currently studying Computer Science in university and for the OSCP certification by Offsec.

Education

Računarski Fakultet (RAF) — Belgrade, Serbia
1st Year Computer Science Student — *Aug 2025 — Current*

XIII Belgrade Gymnasium — Belgrade, Serbia
Prirodno Matematički Smer (Mathematics and Natural Sciences) 4.8/5 GPA — *2021 — 2025*

Skills

- **Programming Languages**: C/C++, Python, Rust, TypeScript, Bash
- **Technologies**: Linux (Arch, Ubuntu, Debian, Fedora, Kali), Docker, AWS (DynamoDB, Lambda, S3), Redis, SQL, NGINX
- **Security Tools**: nmap, netcat, Burp Suite, objdump, gdb, Wireshark, proxychains, Elastic EDR

Work Experience

Software & Devops Engineer — Jun 2025 — Present
AlgoDev Professionals — *Belgrade, Serbia (Hybrid)*

- Performed comprehensive security audit for production web application, implementing XSS prevention, enhanced form validation, and hardening of Nginx deployment
- Developed React components while optimizing performance through lazy loading, Next.js chunking, and strategic caching implementation
- Delivered measurable performance improvements, reducing application load times from 1.2s to 0.6s
- Applied TypeScript security patterns and modern frontend optimization techniques

Co-Founder — Dec 2024 — May 2025
Redzlab — *Belgrade, Serbia (Remote)*

- Founded and built algorithmic trading startup platform with 5-person engineering team
- Engineered high-frequency trading and historical strategy testing system using Python (NumPy, Pandas) with data science-driven strategy evaluation
- Designed and maintained scalable AWS infrastructure using S3 for data storage and Lambda functions for compute operations
- Implemented DynamoDB for database operations and contributed to React TypeScript frontend development

Information Technology Security Engineer — May 2024 — Present
ImpactLTD — *Belgrade, Serbia (On-site)*

- Designed and implemented comprehensive Linux server security hardening protocols including SSH key-only authentication, root access restrictions, fail2ban integration, and custom port configurations to minimize attack surface
- Developed automated security maintenance systems using bash scripting to block Tor exit nodes via UFW/iptables with scheduled cron jobs, significantly reducing reconnaissance and port enumeration risks
- Performed successful web-server penetration test for client, achieving data exfiltration of SQL database, obtaining administrator credentials and secret for CSRF exploit. Wrote a detailed report and mitigated the vulnerability.
- Applied systematic approach to threat mitigation by analyzing attack vectors and implementing countermeasures that increase attacker operational costs and detection probability

Projects

Rust, Jester C2 Framework

- Advanced work in progress malware payload for Linux designed to bypass cutting-edge EDRs by implementing MITRE ATT&CK evasion techniques including encrypted C2 communications via TLS mimicry and kernel API bypasses through asynchronous I/O syscalls. Static analysis is avoided through advanced compiler flags and Rust's complex memory allocation patterns. Successfully deployed and tested in lab environment with Elastic EDR running.
- Source code is private, but I would be happy to talk implementation.

C, Icepick https://github.com/mladicstefan/icepick

- 802.11 Radiotap Frame Analyzer for deep packet inspection to map personal data flows from home networks
- Intercepts and parses raw packet frames using libpcap in C with RTL8812AU monitor mode RF antenna, stores results for analysis

C++, Matrix https://github.com/mladicstefan/matrix

- Built multithreaded Linux-native epoll-based web server capable of handling 50k requests/s on 8-core machine
- Deployed using multistage Docker build process on hardened Debian image.

Python, DWG-JSON https://github.com/mladicstefan/dwg-json

- Built Python geometry semantic engine to infer asset-to-object relationships and generate structured JSON from unstructured AutoDesk DWG file format
- Implemented optimization algorithm preventing engine overfitting using NumPy and Numba for performance

Bash, DevOps Scripts https://github.com/mladicstefan/arch

- Developed automation scripts for Arch Linux workflows including system updates, installations, and hardening with LUKS2 encryption, SSH key-only authentication, microcode updates, filesystem permissions, and AppArmor profiles.
- Created bash script to fetch Tor exit nodes IPs and putting them in custom fail2ban jails that rotate every few hours via cronjob. Implementing defenses against MITRE ATT&CK reconnaissance techniques and significantly reducing port enumeration risks.

C, NanoLisp https://github.com/mladicstefan/nanolisp

- Built minimal Lisp interpreter in C demonstrating parser design, recursive evaluation, and memory management

Extracurricular Activities

Language Proficiency

- English: Cambridge C2 Proficiency Certificate (native-level fluency)
- French: B1 Intermediate (high school qualification)

Baltic Sea World Philosophy Event - Essay Writing Finalist Mar 2023 — Mar 2024

- Achieved finalist status (top 30 of 10,000+ participants) writing philosophy essays analyzing quotes from prominent philosophers including Kant, Dostoyevsky, Camus, and Jean-Paul Sartre

AIESEC Youth Entrepreneurship Competition - 2nd Place Jun 2022 — Jul 2022

- Developed ChatGPT-powered legal advisory application "Legal or Not" using Python, leading 5-person team through product development
- Created promotional content in Adobe After Effects achieving thousands of views

Open Source Contributions & Technical Documentation

- Audited multiple PKGBUILDs on the Arch User Repository (AUR) for security vulnerabilities and build script correctness
- LearnC: Comprehensive GitHub compilation of self-study notes covering advanced C programming, memory management, computer architecture differences, and network communication protocols (HTONS/HTONL, endianness)
- Arch Linux System Hardening Guide: Complete security configuration documentation