Yiğit Çolakoğlu

root@yigit.run | yigit.run | github.com/arg3t

Education

TU Delft – MSc in Computer Science (Cyber Security & Distributed Systems) Sept 2024 – Present
TU Delft – BSc in Computer Science and Engineering, Cum Laude (8.1/10) Sept 2021 – June 2024

Experience

Developer & Intelligence Analyst, PRODAFT – Den Haag, NL Sept 2022 – Present

- Maintain OSS threat intelligence correlation platform (CRADLE) using Python/Django processing 15,000+ weekly artifacts from 5 data sources, reducing analysis time by 45%
- Lead team of 5 engineers developing open source threat intelligence platform CRADLE with 200+ GitHub stars
- Implemented graph-based correlation algorithms using Neo4j for threat propagation of 50000 daily signals, in a graph with 1M+ nodes
- Automated 8 manual threat intelligence workflows using Python, saving 20 hours weekly
- Conducted threat hunting operations, authoring 2 published reports and presenting at ONE Security Summit

Teaching Assistant, TU Delft – Delft, NL Sept 2022 – Present

- Instructed 500+ students across 6 computer science courses using Java/C/Python including Object Oriented Programming, Computer Organization, Distributed Systems, Embedded Software and Software Project

Software Engineering Intern Dutch Police, Team Zeden – Rotterdam, NL Apr 2023 – July 2023

- Managed a 10-week team project with 4 other students, distributed tasks and ensured code quality
- Developed a Python forensicss tool to detect encrypted containers in evidence files upo to 5TB
- Designed a parallelized scanning engine using multiprocessing to spawn processes and schedule tasks
- Automated field data extraction from live Windows/Linux systems in time constrained response situations

Projects

CTF Challenge Designer & Organizer

- Organized 2 CTF competitions using Docker/Kubernetes with 300+ participants, TU Delft's largest CTF event
- Designed 5 CTF challenges using C/Python ranging from web exploitation to binary/kernel exploitation

Tilikum – DAG-based Consensus Protocol with Fair Ordering

- Implemented a batch-order-fairness, fair ordering protocol using Rust and Tokio to prevent MEV attacks
- Optimized it 12,000 tx/s throughput with <2s latency, big improvement for fair ordering protocols

LLVM Fence Optimization – Memory Ordering Optimization

- Implemented C++ LLVM pass optimizing fence placement in concurrent programs using min-cut a algorithm
- Fully eliminate unnecessary fences in LLVM IR programs while ensuring memory consistency

Sanctum – Process-Bound Disk Encryption

- Developed a Linux kernel module enabling transparent per-process encryption with <5% performance overhead

massurl – High-Performance URL Reconnaissance Tool

- Built URL parser & aggregator in C, processing 250K+ URLs/second

Skills

Languages: Python, C, Rust, Go, Java, JavaScript, Bash
Technologies: Django, Tokio, Neo4j, Docker, Kubernetes, Git, PostgreSQL, React, Qt, Tokio
Security: Penetration Testing, Threat Intelligence, Binary Exploitation, BFT Consensus, Concurrency