Yiğit Çolakoğlu

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

Education

TU Delft — Delft, Netherlands
MSc in Computer Science (Cyber Security & Distributed Systems) — *Sept 2024 – Present*

TU Delft — Delft, Netherlands
BSc in Computer Science and Engineering, Cum Laude (8.1/10) — *Sept 2021 – June 2024*

Experience

Developer & Offensive Cyber Threat Intelligence Analyst — Sept 2022 – Present
PRODAFT — *Den Haag, Netherlands*

- Maintain OSS threat intelligence correlation platform (CRADLE) 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 (PageRank, risk propagation) in Neo4j for threat classification across 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 — Sept 2022 – Present
TU Delft — *Delft, Netherlands*

- Instructed 500+ students across 6 computer science courses including Computer Organization, Distributed Systems, and Embedded Software

Digital Forensics Automation Developer Intern — Apr 2023 – July 2023
Police Department — *Rotterdam, Netherlands*

- Developed Python-based forensic tool detecting encrypted containers in 5TB+ evidence files, eliminating manual analysis tasks
- Automated field data extraction from desktop devices allowing officers to quickly acquire data from live systems with minimal training

Projects

CTF Challenge Designer & Organizer

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

Tilikum – DAG-based Consensus Protocol with Fair Ordering

- Developed fair-ordering algorithm preventing MEV attacks using ordering linearizability and batch-order-fairness
- Optimized BFT consensus achieving 12,000 tx/s throughput with <2s latency, major improvement for fair ordering protocols

LLVM Fence Optimization – Memory Ordering Optimization

- Implemented LLVM pass optimizing fence placement in concurrent programs using min-cut max-flow algorithm
- Fully eliminate unnecessary fences in LLVM programs while ensuring memory consistency

Sanctum – Process-Bound Disk Encryption

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

massurl – High-Performance URL Reconnaissance Tool

- Built URL aggregator processing 250K+ URLs/second

Technical Skills

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