ymalhi@andrew.cmu.edu
(412) 499-4176
Pittsburgh, PA

Yuvraj Malhi

yuvraj-malhi.github.io
linkedin.com/in/yuvraj-malhi
github.com/yuvraj-malhi

Education

Carnegie Mellon University, *M.S. in Information Security* *Aug 2022 – Dec 2023*

- ○ **Coursework**: Software Reverse Engineering, Secure Software Systems, Network Security, Mobile & IoT Security, Embedded Systems, Computer Systems, Cyber Risk Modelling.

BITS Pilani University, *B.E. in Electronics Engineering* *Aug 2018 – May 2022*

Skills

- **Forte**: Network Security, Buffer Overflow, Malware, Web Security, Software Security, Machine Learning, Cryptography, CTF, Networking (Proxy, TCP/IP, OSI), Software Development, Linux, Security Automation, Application Security.
- **Languages**: C, C++, Python, MATLAB, LaTeX, HTML, Assembly language, SQL, Java, Shell, Dafny.
- **Tools and Technologies**: Ghidra, MobSF, Metasploit, WireShark, Tensorflow, Pytorch, Scikit-Learn Git, Nmap, Burp Suite, Kubernetes, Docker, Snort, Splunk b.

Work Experience

Amazon Seattle, WA

Security Engineering Intern *May 2023 – Aug 2023*

- ○ Created and put into practice a **numerical risk-assessment system** for evaluating the security of third-party applications, utilizing 22 risk indicators and correlation factors (such as code reviews, security incidents, SSO etc).
- ○ Automated **security compliance** through a AWS cloud-based system using 7 network and internal databases to detect unauthorized applications. Seamlessly integrated this system into the CI/CD pipeline of security teams.

Samsung Bangalore, IN

Network and Systems Intern *July 2021 – Jan 2022*

- ○ Worked on ML-based log analysis for **system compromise/fault detection** and **root cause analysis**.
- ○ Designed an **anomaly detection** system to monitor system background information and take pre-emptive action before hard failure. **Saved service teams 20 hrs/week** by automating 90% maintenance.

BITS Pilani Research Pilani, IN

Research Assistant: Mitigating DDoS Attacks in SDN Data Plane *Aug 2021 – Jan 2022*

- ○ Surveyed and analyzed methods used to **detect and mitigate** Denial-of-Service (**DoS**) and Distributed Denial-of-Service (**DDoS**) attacks at **Data Plane** level in Software Defined Networks (SDN) using P4 language.
- ○ Identified **limitations of P4** for attack detection/mitigation: no support for loops, complex numerical functions.

BITS Pilani Research Pilani, IN

Research Assistant: Machine Learning Intrusion Detection Systems for IoT *Jan 2021 – May 2021*

- ○ Designed and implemented **network IDS for IoT** to overcome few design flaws of existing IDS. This design can detect **22 attacks** with help of **3 ML-based detectors** using Random Forest, ANN, Decision Tree, XGBoost.
- ○ Central Module attack classification rate: **94.41%**. Edge modules attack detection rates: **99.98%** and **99.87%**.

Projects

- **Android Location Stealth**: A Kotlin-based Android application that finds device using (1) **WiFi Triangulation** (for API 19-25) with accuracy of **30 ft** and (2) **IP GeoLocation** (for API 26-31) with accuracy of **200 ft - 2 mi**.
- **Web Security Extension**: Built a Chrome extension using JavaScript to expose on **server design security** based on HTTP headers like CORS, SOP, X-Frame, and ubiquity that could help prevent XSS, CSRF, and code injection.
- **Ultra-fast URL Port Scanner**: Scans URL open ports up to **10X faster** than traditional scanners by using 100 child scanners **concurrently**. The scanner also **lists all IPv4 and IPv6 addresses** via DNS responses for each URL.
- **Concurrent TFTP Servers**: A TFTP single process server to handle multiple clients using **listen** call on multiple FDs with speed **25 Mbps**. Second, A TFTP **multi process server** to spawn a child server per client with speed **50 Mbps**.
- **C - Dynamic Memory Allocator**: Implemented a dynamic memory allocation library of malloc, realloc and free functions to minimize fragmentation while optimizing performance. Built footer-less segregated free lists with first-fit search and immediate coalescing of free space.
- **Simple Hadoop Implementation**: Replicated a simpler version of **Google File Storage** by creating client, data server and meta-data server. Client uploads files in chunks and distributed data servers store 3 separate copies of each chunk to ensure **availability**.