Yuvraj Singh Malhi

Carnegie Mellon University
Email: ymalhi@andrew.cmu.edu | Phone: +91 6362384360
Portfolio: yuvraj-malhi.github.io/ | **GitHub:** github.com/yuvraj-malhi/ | **LinkedIn:** linkedin.com/in/yuvraj-malhi/

Education

Masters of Science in Information Security — *Pittsburgh, PA*
Carnegie Mellon University — *GRE – 331/340* — *2022 - Present*

Bachelor of Engineering in Electronics and Instrumentation — *Pilani, India*
Birla Institute of Technology & Science, Pilani — *GPA – 8.33/10 JEE – 99.38 Percentile* — *2018 - 2022*

Skills & Proficiency

Courses — Intro to Computer Systems, Network security, Network programming, Data structures, Ethical hacking
Interests — Network security, System security, Software Security, Machine Learning, Deep Learning
Positions — Industry Internships (**02**), TAships (**02**), RAships (**05**), Publications (**03**)
Languages — **Proficient –** C, C++, Python, MATLAB, HTML, LaTeX. **Basics –** Java, Assembly language, Spice
Tools — Metasploit, WireShark, Tensorflow, Scikit-Learn Git, GitHub, VScode, Jupyter

Technical Experience

Samsung Research & Development Institute — *Bangalore, India*
Network and Systems Intern — *July 2021 - Dec 2021*

- Worked on **ML-based log analysis** for **system fault detection** and post-mortem **root cause analysis**.
- Worked on **anomaly detection** by monitoring system background information to take preventive action before hard failure.

Birla Institute of Technology and Science, Pilani — *India | Read Paper*
Project Assistant – **Intrusion Detection Systems for IoT using ML** *(Prof V. Shekhawat)* — *Fall 2021*

- Designed and implemented **network IDS for IoT devices** to overcome design flaws of existing intrusion detection systems. This IDS can detect 22 types of attacks with help of three ML based modules using **Random Forest, ANN, Decision Tree, and XGBoost**.
- Central Module used for attack detection & classification with F1 Score **94.41%**. One among two edge modules used for only attack detection at IoT edge with F1 scores of **99.98%** and **99.87%**.

Birla Institute of Technology and Science, Pilani — *Pilani, India*
Project Assistant – **Mitigating DoS/DDoS attacks in SDN Data Planes** *(Prof H. Babu)* — *Spring 2021*

- 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 and mitigation such as: No support for loops and for complex functions, and minimal support for mathematical analysis. Results are further being used to develop a defense solution at data plane level.

IIT Kanpur, c3i Cybersecurity Division — *Kanpur, India*
Research Intern — *May 2021 - July 2021*

- Among **top 5** students from India selected to be a part of the **Intrusion Detection Team** of IIT's cybersecurity division.
- Surveyed and categorized **non-encrypted and encrypted traffic analysis** solutions based on application and mechanism.

ioT-ioT — *Pune, India | See Project*
Linux Automation Intern — *Dec 2019 - Jan 2020*

- **Automated** the process of notifying user on occurrence of a specific event.
- Created an **SMTP client** with CLI in **C++** to send TLS encrypted emails using **cURL** library.

Projects

Ultra Fast Trace-route — *See Project*

- A concurrent server runs traceroute on multiple domains and give results within **3 sec** – up to **10X faster** than standard traceroute.
- A TCP client runs on a separate window to find the **longest common routing path** among given set of domains.

Concurrent TFTP Servers — *See Projects*

- Created a TFTP single process server to handle multiple clients using `listen` call on multiple FDs. Speed: **~25 Mbps**.
- Created a TFTP **multi process server** to handle clients by spawning a child server for each client. Speed: **~50 Mbps**.

Ultra-fast URL Port Scanner — *See Project*

- Scans URL open ports upto **10X faster** than traditional scanners by using upto 100 of child scanners **concurrently**.
- The scanner also **lists all IPv4 and IPv6 addresses** allotted to each URL.

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** stores 3 separate copies of each chunk to ensure **availability** in case of a server crash.
- All inter process communication for download, upload, permissions, and **security** is facilitated by the **meta-data server**.