

# Pranav Waghanna

Rochester, New York

pranav.waghanna@gmail.com | +1 585-537-9675 | Website | LinkedIn | Github

## Education

- |   |                     |
|---|---------------------|
| <b>University of Rochester</b> , Master of Science in Computer Science  | Aug 2024 – Dec 2025 |
| • CGPA: 3.59/4.0  |                     |
| • <b>Coursework:</b> Computer Security Foundations, Collaborative Programming, Intro to Cryptography              |                     |
| <b>Pune Institute of Computer Technology</b> , Bachelor in Information Technology                                 | Aug 2020 – Jun 2024 |
| • CGPA: 8.19/10   |                     |
| • <b>Coursework:</b> Operating Systems, Computer Networks, Processor Architecture, Data Structures and Algorithms |                     |

## Projects

- |  |                     |
|--|---------------------|
| <b>eBPF System and File Access Monitor</b> (View Project)  | Nov 2025 – Dec 2025 |
| • Designed and implemented a kernel-level security monitor with eBPF (BCC) to trace critical syscalls, capturing 14K+ events in 32 seconds (447 events/sec) with zero event loss.  |                     |
| • Developed heuristic-based detection mechanisms using advanced data structures and algorithms for identifying suspicious file access, directory traversal, and sensitive port activity; flagged malicious activities (2.8% of total events) in real time. |                     |
| • Evaluated runtime overhead using syscall tracing techniques, identifying a 2x latency increase in 'ls' execution (0.015s to 0.031s) while maintaining a minimal CPU and memory footprint.  |                     |

- |   |                      |
|---|----------------------|
| <b>Houdini: VFS-layer Rootkit for FreeBSD</b> (View Project)  | Sept 2025 – Dec 2025 |
| • Engineered kernel-mode rootkit within the VFS layer, strategically intercepting 'getdirentries64()' and process list operations to conceal 6 targeted files, demonstrating advanced low-level system manipulation techniques. |                      |
| • Evaded security tools by bypassing traditional syscall based monitoring using VFS function pointers, demonstrating deep kernel compromise and overcoming challenges with defined constants.                                   |                      |
| • Demonstrated expertise in kernel-mode programming on the FreeBSD operating system through VFS layer manipulation, investing 60 hours in rootkit design and implementation.  |                      |

- |  |                      |
|--|----------------------|
| <b>Roomsense: AR-Based Classroom Engagement System</b> (View Website) (Demo)   | Sept 2025 – Dec 2025 |
| • Proposed a full-stack AR system integrating Snap Spectacles, a web platform, a MongoDB Database and Flask backend services, supporting real-time visualization at 20 FPS with less than 1s UI update latency.        |                      |
| • Spearheaded the development of on-device engagement estimation, integrated facial landmark analysis across 10+ signals per frame with a team of 4, and preserved user privacy.                                       |                      |
| • Orchestrated user testing with eight participants to evaluate the software system, leading to a perfect 100% task completion rate and an average satisfaction score of 8.75/10, uncovering previously unknown needs. |                      |

- |   |                      |
|---|----------------------|
| <b>LHUPR: Modular Software Engineering Project</b>  | Sept 2025 – Dec 2025 |
| • Architected and led a team of four in developing a multi module distributed version control system (DVCS) using Rust language, creating formal design documentation and weekly reports over 8+ weeks. |                      |
| • Generated complete module specifications, SPMP, and weekly progress reports for 6+ software components over a 4 week development cycle, facilitating zero late-stage architectural refactors.         |                      |
| • Established rigorous interface definitions across the software modules, decreasing integration errors by 20% and improving code reusability system-wide, reducing debugging time by 7 hours weekly.   |                      |

## Skills

**Languages:** C/C++, Java, SQL, JavaScript, Python, Rust, HTML, CSS, Shell, Bash.

**Technologies:** MERN stack (MongoDB, Express, ReactJS, Node.js), Sockets, Cybersecurity, Cryptography, Computer Networks, Penetration Testing, Web Fuzzing, Footprinting, AWS(EC2, S3), Git, Github Copilot, CAD

## Publications

- |  |          |
|--|----------|
| <b>Effects of Adopting Industry 4.0 on a Manufacturing plant</b>                     | Mar 2024 |
| <b>P. Waghanna</b> , A. Reddy, S. Deshpande, S. Chavan, V. R. Jaiswal and V. Naranje |          |
| DOI: 10.1109/ICRITO61523.2024.10522189   |          |