Dear Hiring Manager,

I am excited to apply for the Software Engineer - Ubuntu Server for Public Cloud position at Canonical. With my Ph.D. in Software and Information Systems and industry experience, I have honed my Software Development, Cyber Threat Hunting, Malware Analysis, and Machine Learning skills. I am a Graduate Assistant at the University of North Carolina at Charlotte, developing a distributed security analytics system for distributed threat hunting. My research has been funded by DOE and ONR. My work aims to deliver monitoring intrusiveness, reduce communication overhead among agents, and enable local decision-making while maintaining attacks and attack techniques detection accuracy high and in time.

As a Teaching Assistant, I teach, design, and prepare graduate courses in Principles of Information Security and Privacy, Network Infrastructure Security, and Data Mining. Before joining UNC Charlotte as a Ph.D. student, I worked as a Software Engineer and Team Lead at Kona Software Lab Ltd., Dhaka, Bangladesh, developing middleware libraries for PKI and CA systems. I also led a team of three software developers to design and develop NFC-based smart card authentication for Windows OS.

I am well-versed in programming languages like Python, Java, C++, C, and Prolog. Additionally, I have expertise in web development and scripting with Shell Scripting, PHP, JavaScript, HTML5, and SQL. I am proficient in using visualization tools such as UML, Weka, and Gephi and version control tools such as Git. I have experience with virtualization tools like VirtualBox, VMWare, Kubernetes, and Docker, and I am familiar with Scrum/Agile development. I am well-versed in TCP/IP networking, OSI models, CI/CD, and OOP. I have experience with machine learning libraries such as Stanford CoreNLP, AllenNLP, NLTK, Scikit-learn, Keras, and TensorFlow. I have worked extensively with cryptography, OpenSSL, MITRE ATT&CK framework, TCP/IP, ElasticSearch, RabbitMQ, IDAPro, Sysmon, OllyDbg, and Splunk.

During my Ph.D. journey and working as a software engineer in Fintech, I was involved in several projects. I am working on AUTO-Hunter, a Distributed Hierarchical Event Monitoring System for Attack Diagnosis through Active Investigation of Attacker Activities. I designed and implemented this system to reduce attack detection time, communication overhead, and resource usage. I also developed low-level log collecting agents for the Windows system (ETW, event logs, Syslog, NetFlow) and detectors to map low-level traces to the MITRE ATT&CK technique and evidential reasoning framework. Additionally, I worked on TTPDrill, which was an Automatic and Accurate Extraction of Threat Actions from Unstructured Text of CTI Sources and mapping of threat actions to MITRE ATT&CK techniques. I extensively used Python, Java, RabbitMQ, Elasticsearch, NLP, and Docker to develop the projects mentioned above.

One of my research areas is developing measures and metrics for Critical security control enforcement and validation. I have contributed to designing measures and metrics for the open-source **CIS controls assessment specification (CIS CAS)** project. In this Critical Security Control (CSC) Validation project, I am developing automated extraction of threat action, observables, and key measurement indicators (KMI) and metrics for the KMI of each CSC. During my collaboration with CIS open-source project CIS CAS, I feel the following areas to consider for

doing things right in open-source development- Efficient version controlling, good documentation, and regular maintenance.

At the beginning of my Ph.D., one of my research areas was malware analysis and Active Cyber Deception (ACD). In this project, I analyze and extract malware deception parameters using symbolic execution. For the malware analysis and symbolic execution, I use hypervisor technology **KVM/QMU, a Linux kernel module**. During this project, I gained hands-on experience with Linux packaging. Additionally, I implemented memory management and file system operating system using **Nachos**, which aided me in pursuing the rudimentary principles of Operating Systems during my undergraduate study. Moreover, I worked on embedded OS like **Java Card** as a software engineer in FinTech.

After completing my undergraduate study, I joined **Kona Software Lab Ltd.**, Dhaka, which is a part of KONA I, one of the smart card solution providers in the world. During this period, I have grown a deep insight into networking and security systems, which include cyber security, cryptography, digital signatures and certificates, authentication methods, and public key infrastructure. As my first project, I was assigned to the project PKCS#11 middleware. PKCS#11 middleware is a dynamic library that provides smart card personalization, key generation, and cryptographic operation service. My task was implementing smart card profile creation, key generation, cryptographic operation, multithreading, and multiprocessing in the middleware library. After that, I had to implement a CMS (Cryptographic Message Syntax), a PKCS#7 based toolkit to support CA during certificate issuing. It will be used during certificate signing request generation between the CA server and the client.

After that, I worked on a custom CSP project, a cryptographic security provider like Microsoft's base CSP with additional cryptographic operation support. Using this Custom CSP, I have to implement Microsoft's crypto API. Using these crypto APIs, I have to support key pair generation, signature generation, and verification, which will eventually be used for document signing, mail signing, and smart logon.

My next research project was Smart Windows logon through Bluetooth authentication. In this project, I gained detailed knowledge of the Windows logon system, Bluetooth, and BLE stack. I have to implement the credential provider and Authentication package, two Microsoft APIs used during Windows logon. For the device authentication, I have to implement an Android service that will access smart cards in a wearable device to retrieve Key-Pair and certificate, which will be used in signature generation and verification during device authentication. Besides study and research, I led this project alongside Custom CSP, as mentioned above, by which I learned the value of teamwork. A team must always aspire to become more significant than the sum of its parts, and the leader should inspire others by setting an example.

I am a Ph.D. candidate at **UNC Charlotte** with a 3.93 out of 4.0 GPA. Though GPA is not an assessment factor for Ph.D., I published my research in security conferences like ACSACS, ESORICS, and HOTSOS. I completed my B.Sc. from **Bangladesh University of Engineering and Technology**, the topmost university in Bangladesh. I completed my secondary school certificate exam (high school, S.S.C) and Higher secondary school certificate exam (College, H.S.C) with a GPA of 5.0 out of 5.0. During my high school and college studies, I was the topmost

student in the school. During those days, I considered myself highly competent in mathematics and science subjects (Physics, Chemistry, Biology). Though I was not topmost in arts during high school, I was above average in arts and top in language class. Because of my passion for science and good background in science during high school and college, I chose computer science and engineering for undergrad study. Next, my extensive software development experience using cryptography in the payment industry encourages me to pursue my Ph.D. in cybersecurity. Besides my academic journey, I am a voracious non-fiction reader and a soccer fan, and I play soccer whenever an opportunity comes. I was an active blood donor and blood organizer during my undergrad.

From my experience of working in Industry, the best practices to follow for improvement in software development are agile and incremental development to adapt to changing requirements, regular code reviews, automated testing like unit tests, integration tests, and deployment tests to ensure reliability and stability of the software, comprehensive and up-to-date documentation to promote collaboration and maintainability and reduce knowledge gaps, efficient use of version control system like Git to track code changes and avoid conflicts and mistakes. Another practice to follow is CI/CD pipeline to separate integration and delivery from the software development. In addition to the practices to track quality software development, the best practices for improving security in software development are conducting secure code reviews such as looking for potential vulnerabilities and weaknesses, providing guidelines and checklists to identify common security issues, implementing security testing through vulnerability scanning, penetration testing, promoting threat modeling to introduce developers to the concept of identifying potential threats, assessing their impacts and devising strategies to mitigate them. Above all, we must train the developers to create awareness about security by providing best practices like OWASP's top ten. Though I have yet to gain any industry experience in large-scale cloud estate management, from my exposure to distributed systems and cloud services, centralized management and compliance with regulatory requirements are the first things to consider for effective management of large-scale public cloud estate. Centralized management enables efficient policy and security control enforcement, resource provisioning, monitoring, and standard management task automation. To automate resource provisioning, configuration, and management, we can promote the use of infrastructure-as-code tools like Terraform and Ansible. Finally, collaboration and knowledge sharing among the developers is essential to improve the developer experience in large-scale estate management. The version control tool Git and issue tracking system will help in this aspect. Moreover, providing training and skill development programs to developers about managing large-scale systems will improve the developer experience.

Since Canonical is known for the development and support of Ubuntu, I think the mission of Canonical is to provide user-friendly, secure, and versatile operating systems and software platforms for individuals, enterprises, and cloud environments to enable widespread adaptation of open-source technologies and collaboration within the open-source community. Though canonicals' revenue model is around providing cloud services, the dependency on community contribution is unappealing. Though open-source communities are usually vibrant and supportive, some risks may arise from addressing critical issues and conflicting opinions on project directions. A balance must be established between commercial interests and community engagement. Since Canonical's revenue comes from cloud services to my knowledge, the main competitor of

Canonical is Amazon Web Service, Microsoft Azure, and Google Cloud Platform. To compete with those cloud service providers, Canonical should highlight the Ubuntu server's advantage and invest in providing enterprise-grade support, timely assistance, security updates, and proactive problem-solving for mission-critical deployments.

In an era of cloud computing and containerization, the chance to work on cloud services provides career growth opportunity. The opportunity to be involved in the open-source development of Ubuntu features and services for cloud platforms and to work on virtualization, containerization, and package management is exciting. I hope to bring value to Canonical with my software development expertise, security research, and virtualization knowledge.