With the arrival of the information era, we generate an increasingly massive amount of data every second. Alongside, is the rapidly increased bandwidth of the internet. The high availability makes things that are impossible to imagine in the past possible. We’ve witnessed the boom of cloud services, IoT, and more recently, machine learning.

As a benefit of these lately introduced technology, we’re migrating our life services as well as research works to the cloud, and our data flow on the network is increasing by the day. However, since the network is connecting everyone, there is someone out there with unfriendly intentions. Cyber-attacks are a part of almost everyday life and to put a system on the internet, a high level of security precaution is defiantly always needed.

As a student in engineering, sometimes we need to put our own developed system on to the internet to provide services. While Kyushu university has a firewall across the entire campus, to provide services to the whole internet, there’s always a need to expose ports to the internet and by doing so, add security risk to the system itself as well as the entire campus network. Therefore, certain security concern is required.

Security should be tackled in a layered fashion. This layered security pattern provides distinct focus areas to secure, monitor, and tune.

Most often, the focus in this layered model starts at the lowest level and that is the physical hardware. Hardware must be physically secured and uniquely identifiable to other trusted systems. For instance, while any computer can be called a server, the real server that provide services to the massive should be placed in a secure room, instead of labs or offices. The server room should provide security sensors, access control as well as redundant power and network systems to provide high availability.

After the hardware is secured, the next security concern is the network. At this level, we deploy a number of devices to ensure the integrity of the security systems. This is also one of the largest investments besides the server itself. For instance, dedicated firewalls to analyze incoming and outgoing network packages to ensure network flow within a normal range. For distributed systems, special types of firewalls are required to handle the situation where outgoing flow and incoming flow are directed to different distributed servers. The next security implementation is network segmentation strategies like VLANs and security zones, this would ensure that even if the server itself is attacked or breached by attackers, other servers in the network are still in tacked and function normally. At the upper layer of the network, there are routers providing message forwarding across different subnets. While the security of the router itself is no doubt to be essential, the ACL inside the router and layer-3 switch is also required.

The next layer of security is operating systems. While most of the servers are running Linux or Unix systems and relatively immune to the virus, there're some servers running on NT architecture where most of the computer virus can be executed on those systems. For windows systems, anti-virus software is surely required and for Unix like systems, certain precaution is also required. In a more general way, an operating system security layer is where user access controls, patching policies, and software restrictions provide additional layers of security for our system as a whole.

Finally, the top layer of security is application layer security. And this is where good coding practices, proper data handling, and application user access controls play the most important part. This is also as a developer in general, where we should pay the most attention to. A good implementation of a system should always consider security as an essential part besides business logic. In addition to using security frameworks such as spring-security or other things like this, the security verification of a system is proved to be vital in most scenarios. From my previous working experience, during the period before the deploy stage, at least 5 steps of application security verification are required. Firstly, it’s the design review, it is the stage before coding is even started to ensure the entire design meets the security requirements. Next, is the tooling, where specialized software is used to find the security flaws of the application. After tooling is passed, the application then goes through a Whitebox security review where the code is passed to a 3rd-party to review the code manually and find its security flaws. When the procedure is done, the next is the Blackbox security audit where the application is delivered without code and the security flaw is detected via actual use. Finally, when the application gets delivered, a coordinated vulnerability platform is embedded where the mass could provide bug reports so that in the next version, these bugs could be repaired.

In all, as a researcher especially as one in the university, the security concern of physical device and network are not so urgent as of those of a commercial entity. However, also as a developer with needs to deploy applications onto the internet, it’s important to take security risks seriously and design our software with security concerns in mind.