E-Portfolio: https://sam-tselapedi.github.io/CyberSecurity/ssa.html

Secure Systems Architecture: Reflection Essay

The security system architecture is a system that defines the entire security of an IT organisation based on specifications, processes, and standard operating procedures, which is part of the prevention, mitigation, and investigation of different threats. In our final project, we came with systems that aimed at detecting and controlling the home-based IoT nodes vulnerability management through automated services.

We implemented a frequent scanning of IoT nodes, and the vulnerable devices were prohibited access to the internet connection. The server allows clients to send and receive responses through the RESTful API interface. We shared tasks guided by the group leader and later merged, using the bottom-up approach to develop the more prominent solution to securing systems. We were working to identify threats affecting computer systems and possible ways of securing them, especially online threats.

We developed the solution with Python since it is a good language for data analysis in IoT systems. MongoDB was used to structure the Database features due to its document-oriented architecture, which is more secure. To strengthen security, we used OTSHA for authentication

Operating systems and web applications are prone to threats such as Denial-of-Service(DoS) attacks. To strengthen the security of the architecture, we researched vulnerabilities and best practices following the OWASP project methodology. Additionally, we factored in methods for code refactoring to produce a secure code that reduces vulnerabilities (Kayaalp et al.,2012).

Using the knowledge we acquired during the attack defence tree(ATD), we sampled the sources of most threats and countered them. For example, we implemented a non-repudiation measure(accounting) by recording security transactions logs. We had challenges in identifying the code errors that arose during the implementation. We tested the system security feature we identified to uncover security implications. Some of the critical remedies we identified include; Software Composition Analysis, Input Validation, encryption, and rate-limiting to minimise the security risks.

We researched smart doorbells during the ADT exercises. We discovered that due to the internet connection architecture in their implementation, they are vulnerable to authorisation creep. The authorisation issue resulted from the conflict between the access control list local to the node and the control server. After reading through blockchain research papers, I believe we can use blockchain architecture to validate the correct access list.

Microservices architecture was incorporated in our solution to protect the system's availability. This architecture reduced the vulnerabilities posed through direct data and resource allocation. The architecture also ensured that objects were accessed only by the authorised subjects or processes.

I found the whole exercise educative, knowledge improving, and confidence improving. Group work enhanced the mutual understanding of how vital group work is and taught us that group work improves one's knowledge where one member adds on insight well unknown to others, and they learn from your ideas. I realised that I could incorporate security measures in other organisational architecture. Consulting from more

professional people would have brought more ideas and expert solutions to improve our knowledge.

We documented each process and shared copies with each group to improve our future skills. We made it a routine to get advanced knowledge and share it with different computer science-related professionals to strengthen our skills. I also learned new architecture designs that would help improve my skills, such as Message Queuing Telemetry Transport Publish/Subscribe Architecture which I did not know before this module.

Please see my e-portfolio for my contributions to the group activities at **E-Portfolio:** https://sam-tselapedi.github.io/CyberSecurity/ssa.html

Appendix:

Samples of application of the knowledge and skills I have learned in this module.

I have designed and deployed Microsoft Exchange 2019 at the National Institute of Communicable Diseases (NICD), my employer. The system was deployed as distributed architecture over two towns data centres. The solution features microservices where the EDGE server are used for receiving and sending external emails, and the mailbox access servers are used as proxy servers to access user mailbox. A reverse proxy has been deployed to protect the internet-facing server, and an application load balancer is deployed internally for accessing the internal servers

1 Exchange 2019 – Logical Architecture

The proposed logical architecture for NICD is shown below. Microsoft Exchange 2019 will be deployed in the primary and Secondary Data-center's to provide a centralized and highly available site resilience Exchange messaging infrastructure. All elements of the architecture require dedicated virtual servers for a multi-role Exchange deployment. These are required to provide security and integration of internal infrastructure services from services exposed for public access via the Internet. It is also important to note that Reverse Proxy solutions, an Internal PKI Solution, a third-party Unified Communications Certificate solution and an Office Web Apps infrastructure in each data centre are prerequisites for the below environment and will be catered for as part of the project. A Load balancing infrastructure is also required to use Multi-Role exchange servers, and NICD will cater to this requirement in each DMZ and Datacenter. Microsoft Exchange Edge servers will be deployed in the DMZ one in each data centre. The server in the Sandringham DMZ will be the primary Edge server; if the Disaster recovery is Invoked, the Edge Server in Braamfontein will be configured to act as the primary Edge Server

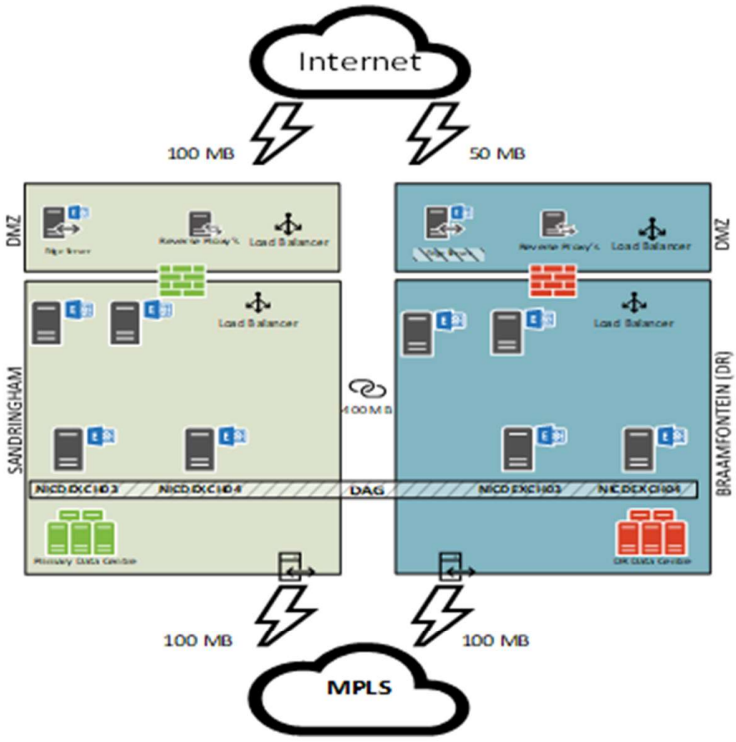
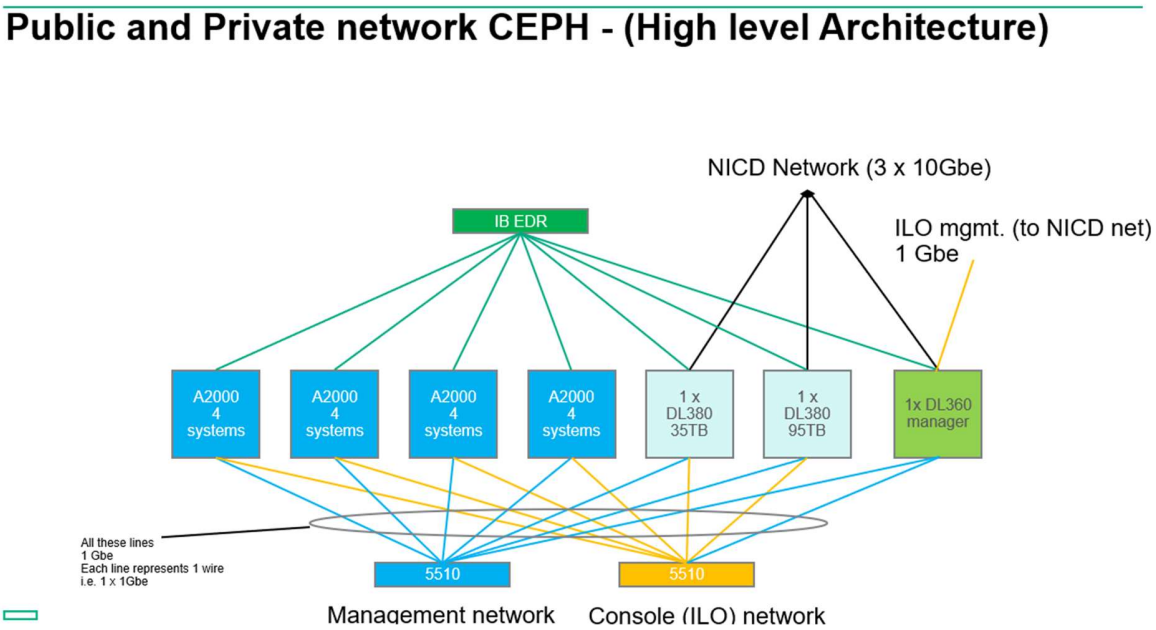


Figure 1: NICD - Exchange 2019 Topology Diagram

I have also designed and worked with my colleagues in deploying High-Performance Cluster for Genome Sequencing. The architecture consists of four computing nodes and two management nodes for scheduling management in parallel computing. We deployed SLURM Workload manager software for scheduling management.



References:

Alves-Foss, J., 1998, January. The architecture of secure systems. In *Proceedings of the Thirty-First Hawaii International Conference on System Sciences* (Vol. 3, pp. 307-316). IEEE. https://ieeexplore.ieee.org/abstract/document/656293/

Cohen, F.B., 1993. Operating system protection through program evolution. *Comput. Secure.*, *12*(6), pp.565-584. http://all.net/books/tech/evolve.pdf

Huth, C., Zibuschka, J., Duplys, P. and Güneysu, T., 2015, April. We are securing systems on the Internet of Things via the physical properties of devices and communications. In *2015 Annual IEEE Systems Conference (SysCon) Proceedings* (8-13). IEEE. https://ieeexplore.ieee.org/abstract/document/7116721/

Kayaalp, M., Ozsoy, M., Ghazaleh, N.A. and Ponomarev, D., 2012. Efficiently securing systems from code reuse attacks. *IEEE Transactions on Computers*, *63*(5), pp.1144-1156. https://ieeexplore.ieee.org/abstract/document/6355533/