Queens Medical Centre – Secure ASMIS Deployment

1. Introduction

1.1 Background

Queens Medical Centre is a community clinic that acts as the first point of contact for any resident within its catchment area and it is facing difficulties managing the high volume of calls. The high call-volume is leading to long waiting times for patients and disrupting access to on-time care. Furthermore, the clinic administration is facing issues planning required resources to account for the community's rate of growth.

To address and mitigate these critical issues, that are impacting the centre and residents, the clinic's management has acquired a web-based appointment and scheduling management information system (ASMIS). Once implemented, the system will enable prospective patients to book their appointments online.

This report aims to evaluate the proposed web-based ASMIS system in regards to potential cyber threats and how these can be mitigated.

1.2 Benefits of Implementing ASMIS

The introduction of an ASMIS system will streamline the process of allocating a specialist to patients for consultation. When applying for an appointment online, the system will request vital information from the patient upfront, with which it will be able to determine the best specialist to assign to a given case. Factors such as availability and pre-existing workloads will be taken into account. By asking for information beforehand, the system can cut down on waiting times for prospective patients and ensure access to on-time care. The system also provides ease of scalability to manage the growing population.

1.3 Desired Outcome

The goal of this report is to achieve an implementation of the system, provide the above benefits while keeping potential cyber threats in mind. A typical information security model that can guide an organisation's efforts and policies aimed at keeping its data secure is the CIA triad. The CIA triad model helps to determine whether a system is secure and is considered the industry standard for computer security. The triad consists of the following three principles (von Solms and van Niekerk, 2013):

Confidentiality: only authorised users and processes should be able to access or modify data

Integrity: data should be maintained in a correct state, and it should not be possible to modify it improperly, either accidentally or maliciously

Availability: authorised users should be able to access data whenever required

An organisation should incorporate these core principles when developing secure implementations. For this report, the CIA triad is used as a foundation to evaluate the overall implementation success.

2. Identifying Threats

2.1 Legitimate Use Cases and Actors

The identification of legitimate users of the systems and their respective use cases helps to establish required access levels and design the overall data flow of the system. In the case of the Queens Medical Centre's ASMIS system, there are three primary users: the prospective patients that will be using the system to book consultations, the specialists that will be allocated to said appointments, and the management to monitoring the system.

As depicted in figure 1, the general use cases of the system are as following:

Prospective patients need to be able to book their appointment via the system and submit vital information to ensure the best specialist is assigned to a given case. Once an appointment is submitted, patients need to be able to check their appointment status and view set dates for the consultation. Patients will also need to be able to add to or edit already submitted information.

Specialists need to be able to check cases that have been assigned to them. They further need to be able to check their patient's information before the consultation. Additionally, specialists might need to adjust their availability in the system, for instance, when requesting leave.

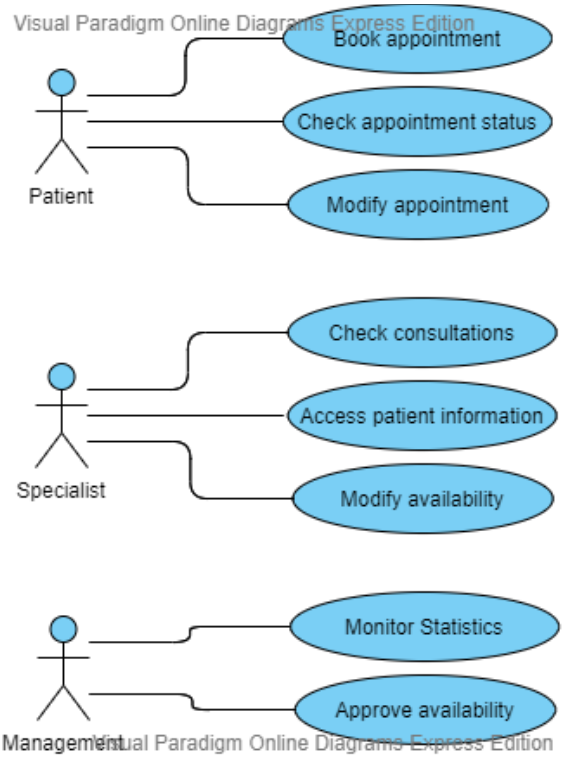

Figure 1 - UML Use Case Diagram for Queens Medical Centre

2.2 STRIDE Methodology

Now that the legitimate use cases have been established, it is essential to identify the illegitimate abuse cases as well. A useful tool to highlight such cases is the STRIDE methodology. The STRIDE method is considered a failure-oriented threat modelling approach which focuses on six core areas:

Spoofing, **T**ampering, **R**epudiation, **I**nformation Disclosure, **D**enial of Service, **E**levation of Privilege

When considering potential threats, it is useful to look at each of the above categories and evaluate the impact each could have on the system. The STRIDE methodology is considered the most mature threat-modelling method available and has been adopted by Microsoft in 2002 (Howard and LeBlanc, 2002).

2.3 Potential Abuse Cases

The goal of this chapter is to establish potential abuse cases by applying the STRIDE methodology. To do so, the individual categories will be applied to the proposed ASMIS system.

Spoofing describes attack vectors in which the attacker impersonates another user, a process or a system. In the case of the ASMIS, a likely case of a spoofing attack that may occur is the attacker authenticating with another user's credentials. This may happen via brute force or, if the user uses the same password elsewhere, having been obtained via an external database leak.

Data tampering involves the illegitimate modification of data. This often occurs if the system has set weak access controls. Tampering could result, for instance, in alteration of data in the patient database if the read, write and execute permissions have not been set up correctly.

Repudiation describes a user denying to have acted without other parties having the means to prove otherwise. This can occur if the system has no way of tracing performed actions of users. For instance, a specialist might illegitimately access a patient's file not currently under consultation. Without traces or logs, it would not be possible to prove the specialist has done so. This would also mean that an information disclosure threat has occurred. The information has been exposed to a party who should not have access to it.

Denial of Service (DoS) may occur if attackers deny the ASMIS service to valid users. This often occurs via distributed denial of service (DDoS) attacks that use botnets to flood the service. The consequence of a DDoS attack would be the disruption of regular traffic to the ASMIS, resulting in prospective patients not being able to book consultations (Howard and LeBlanc, 2002).

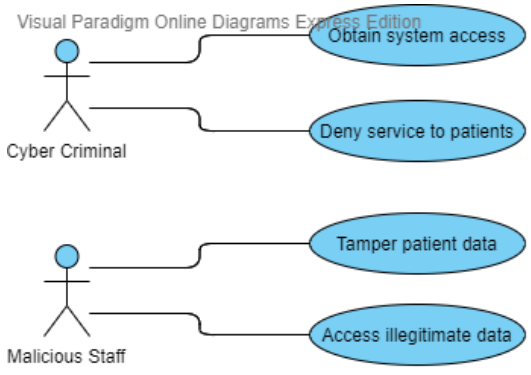

Figure 2 – UML Sample Abuse Case Diagram for Queens Medical Centre

2.4 ASMIS without Inbuilt Secure Design

Developers and product owners often use models to communicate their work to other stakeholders. The Unified Modelling Language (UML) has established itself as an industry-standard modelling language since its introduction in 1997. UML diagrams offers a straightforward way of communicating specific aspects of object-oriented systems.

UML class diagrams are used to depict the classes of a system, including their attributes and operations. Additionally, the inter-relationship between classes are also described. This makes class diagrams very useful for a variety of uses, from understanding requirements to describing systems designs in detail (Ambler, 2003).

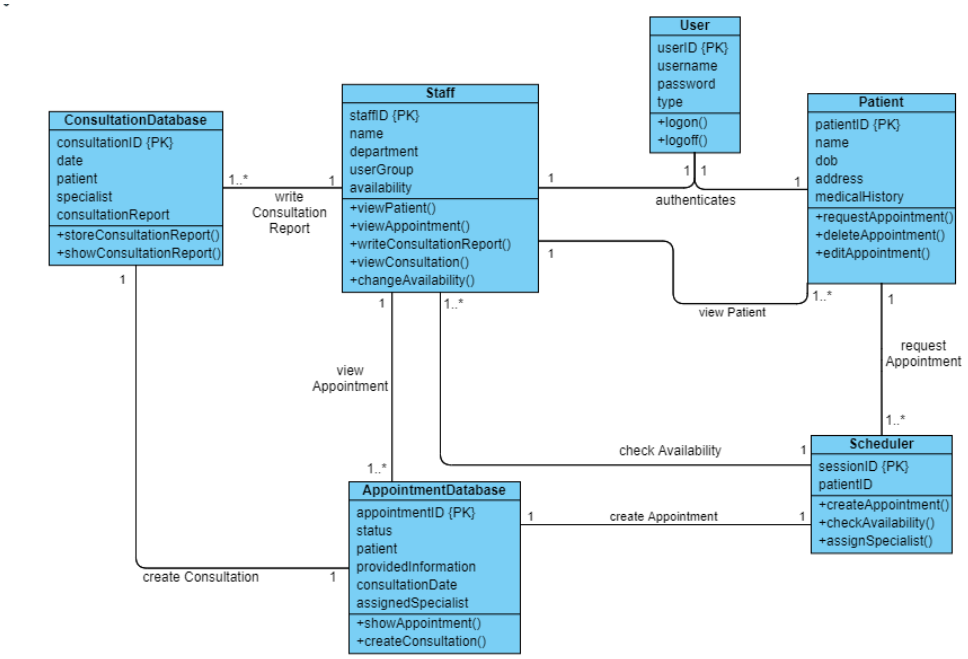

Figure 3 - ASMIS UML Class Diagram without inbuilt Secure Design

In a system without inbuilt secure design, staff can view any patient data, as well as their respective consultation data, such as reports. Figure 3 depicts the UML class diagram of such an insecure system. In addition to weak access controls and permissions, the system does not offer any traces or logs, thus making it vulnerable to repudiation. With this design, a malicious staff member could easily access any patient record and modify it.

UML sequence diagrams are a dynamic modelling technique that is often used to validate the logic and completeness of a usage scenario. They depict the sequential order of how an interaction is carried out step-by-step (Ambler, 2003). In the case of secure design, a sequence diagram can illustrate the steps a threat actor may utilise to achieve an abuse case and the results of such a case.

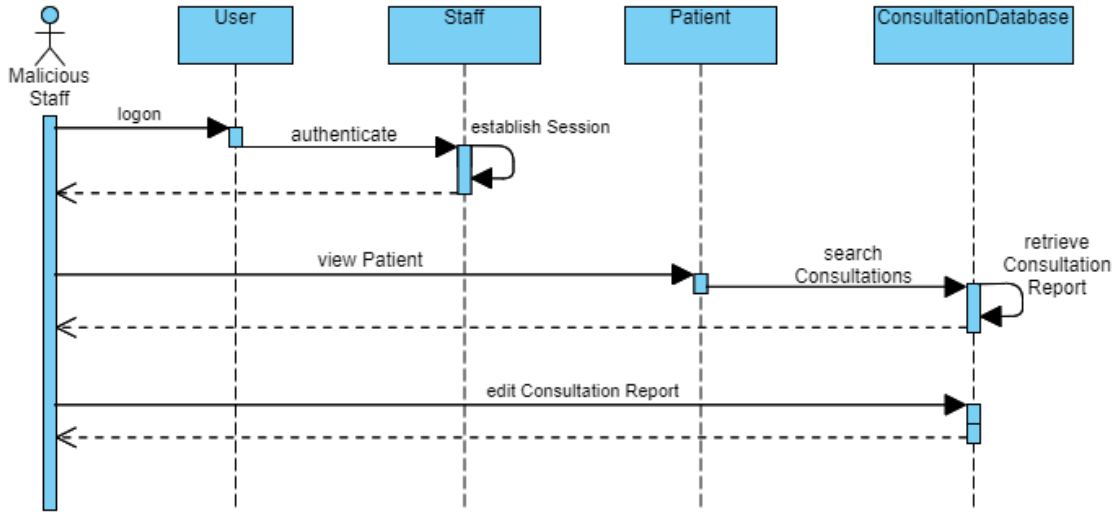

Figure 4 - ASMIS UML Sequence Diagram without inbuilt Secure Design

Figure 4 depicts a UML sequence diagram for the same insecure system discussed above. The interaction described in the diagram is the case of a malicious staff viewing patient data and subsequently accessing and modifying the patient's consultation report. There are no access controls in place that prohibit the staff from viewing and modifying the data they should not have access to. Again, there is no logging in place either. In the above scenario, the malicious staff member would not have any obstacle in achieving the abuse case and would not leave any traces with which the actor could be identified.

3. Proposed Secure ASMIS Deployment

3.1 Secure Design Practices and Technologies

The threats discussed above pose a severe threat to the ASMIS implementation. It is essential to deploy secure design practices and technologies to mitigate the risk of a threat.

To reduce the risk of having a cybercriminal obtain system access via another user's credentials, it is vital to enforce strict password conventions. Having a combination of a minimum password length and complexity restrictions in place often makes brute-forcing the password very difficult due to the extended time required. As an example, increasing the password complexity to an 8-character full alpha-numeric password increases the required time to brute force it to more than 252 days at 10 million attempts per second (Cheswick, 2012). This should be the recommended minimum restriction to implement. Another aspect that helps improve authentication security is to add a notice for users to use a unique password when registering for the service. This would eliminate the risk of having authentication compromised by an external data breach of the same password.

The risk of data tampering can be mitigated by setting up secure access controls. In a Linux system, this can be done by setting up secure file permissions via the *chmod* command. It is vital to approach permissions from a bottom-up approach. Users and groups should only be granted a privilege if they cannot accomplish their tasks without it. This holds especially true for database access controls (Connolly and Begg, 2015). In the case of the ASMIS system, it should be ensured that specialists can

only access and modify the files of the patients they consult. This serves as an added layer of security to prevent information disclosure and data tampering.

Furthermore, tracing all actions conducted by each user on the system via log files is an integral part of a secure implementation. Nonrepudiation is essential to account for cases where data is illegitimately accessed or tampered with by malicious staff. By having a log created whenever a patient file is created, viewed or changed, it is possible to trace back the event to whoever actioned it.

An effective measure against DoS and DDoS threats is the implementation of a cloud-based Web Application Firewall (WAF). The WAF is deployed and managed by a service provider on a SaaS-basis and entirely located in the cloud. This offers a straightforward solution with no physical maintenance required; however, this solution only allows a low degree of customisation and flexibility. WAFs represent an affordable security solution that can prevent the expense and loss of trust that result from common attacks and breaches (PentaSecurity, 2020).

3.2 ASMIS with Inbuilt Secure Design

Considering the above improvements, the class model of the UML class diagram of the previously discussed system needs to be updated to reflect the aforementioned secure practices. Figure 5 depicts the UML class diagram of an ASMIS system with inbuilt secure design.

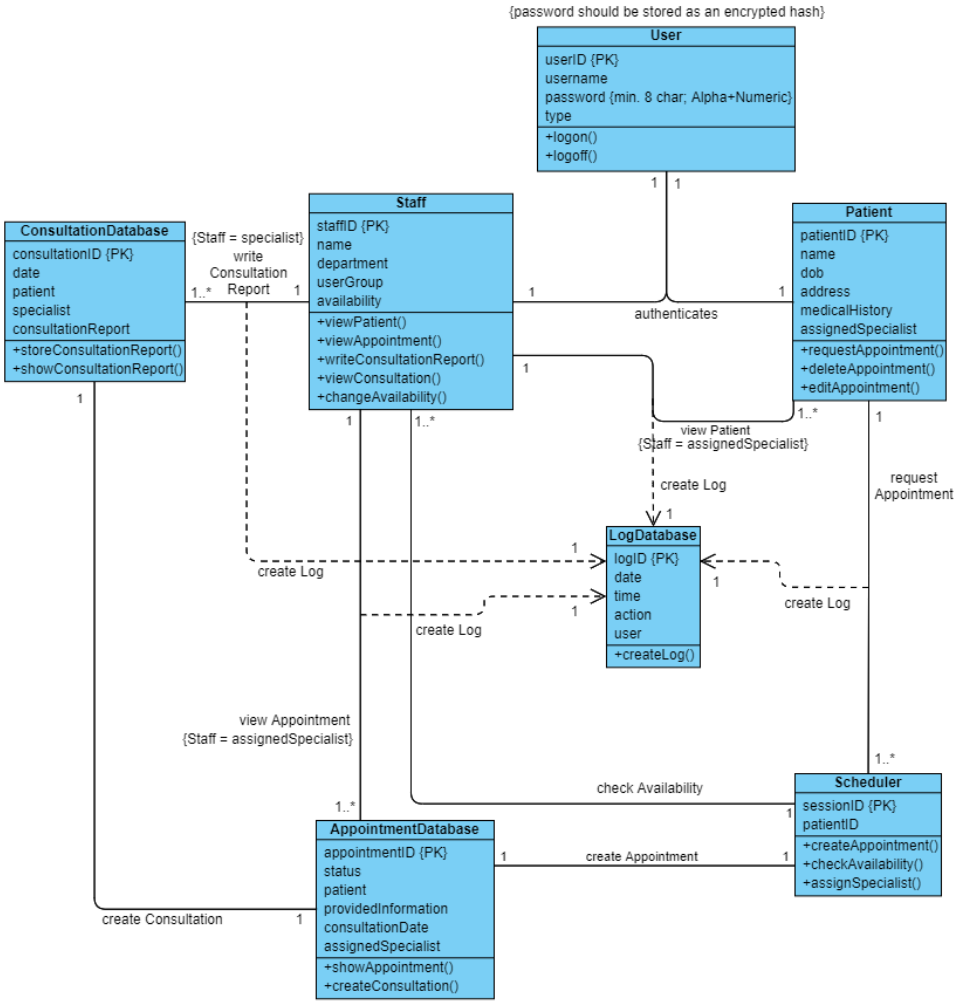

Figure 5 - ASMIS UML Class Diagram with inbuilt Secure Design

As discussed, a log database has now been included in the system that traces all actions users undertake on it. This includes viewing, creating and editing data in the consultation, appointment and patient databases. Furthermore, restrictions have been added to limit what data staff can view and modify. Specialists can now only view patients that have been assigned to them. The same applies to consultation data: a specialist can only write a consultation report for a consultation he is assigned to. The password convention has also been updated to reflect the 8-character full alpha-numeric minimum length. Additionally, the password should never be stored as a plaintext string; instead, the password should be encrypted upon registration and stored as a hash to compare against when a login attempt is made.

When revisiting the abuse case highlighted in chapter 2.4, there are now severe obstacles in place for the threat actor to achieve his abuse case.

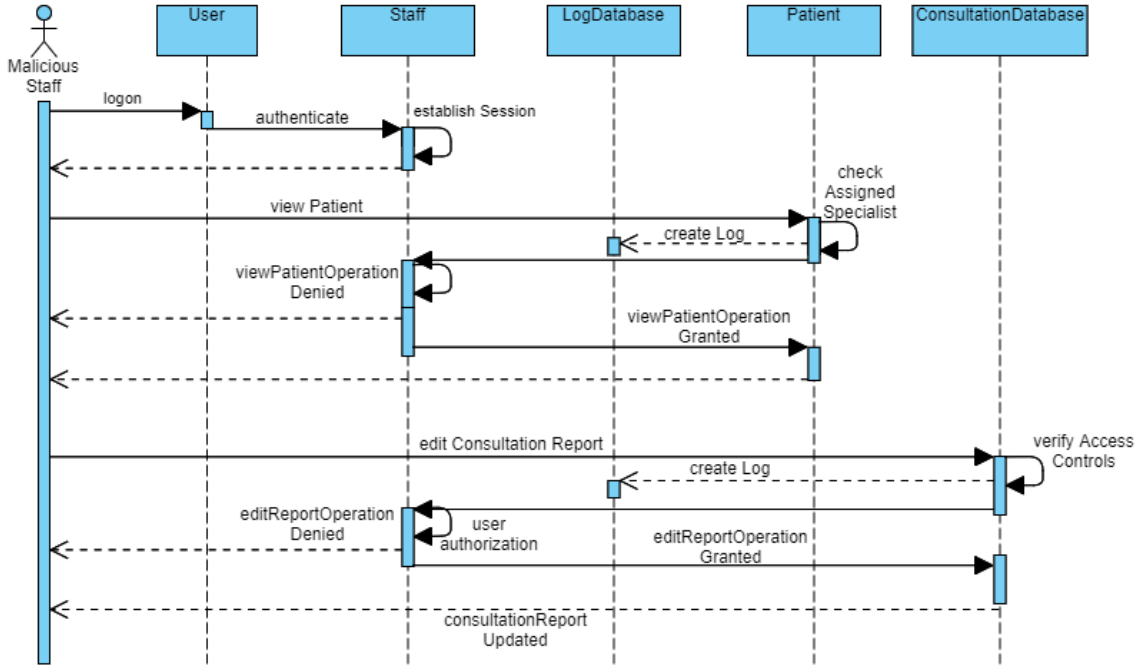

Figure 6 - ASMIS UML Sequence Diagram with inbuilt Secure Design

As a malicious staff member, viewing a patient is now tied to the assigned specialist of the patient. If another staff member attempts to view the patient's data, the view operation will be denied. The same goes for modifying the consultation reports. The staff is only able to edit a report with the appropriate access controls, else, the edit operation is denied. Furthermore, the logging is now in place and takes action whenever an interaction occurs.

4. Conclusion

As mentioned in chapter 1.3, the CIA triad serves as a foundation for the evaluation of the secure implementation. By enabling strict information disclosure restrictions, as well as a password convention that ensures a certain level of complexity, the confidentiality of the system is ensured. Robust access controls, coupled with a tracing capability, ensure the integrity of the system. Lastly, the implementation of a cloud-based WAF ensures the availability of the service.

In conclusion, when applying the proposed security practices and technologies, the system will be guarded against the most crucial threats. All aspects of the CIA triad are fulfilled, and the security measures do not interfere with the legitimate users and their respective use cases.

Reference List

Ambler, S. (2003) *The elements of UML style*. Cambridge University Press.

Cheswick, W. (2012) Rethinking Passwords. *Association for Computing Machinery (ACM)* 10(12). Available from: https://queue.acm.org/detail.cfm?id=2422416 [Accessed 21 November 2020].

Connolly, T. & Begg, C. (2015) *Database systems: a practical approach to design, implementation and management*. 6th ed. Pearson Education. Available via the Vitalsource Bookshelf [Accessed 21 November 2020].

Howard, M. & LeBlanc, D. (2002) *Writing secure code: practical strategies and proven techniques for building secure applications in a networked world.* 2nd ed. Microsoft Press. Available via the Vitalsource Bookshelf [Accessed 21 November 2020]

PentaSecurity (2020) 3 Types of Web Application Firewalls: How to Choose? Available from: https://www.pentasecurity.com/blog/3-types-web-application-firewalls/ [Accessed 22 November 2020].

von Solms, R. & van Niekerk, J. (2013) From information security to cyber security. *Computers & Security* 38:92 -102.