Benefits of web-based appointment and scheduling management information system (ASMIS)

Web-based ASMISs do have many benefits. They can reduce costs, errors and are available day and night. They can be configured, so that they find the best specialist based on information provided from the patient and the schedule of the doctor. Doctors can also benefit from such database driven systems in cases where they for example want to see the patient's history. This information can easily be retrieved within the second. On top of this, valuable meta data can also be created from an ASMIS. The hospital for example could get information from where or when they get most requests, which might help to improve their services. Classic systems having phone operators cannot easily be adapted to a growing community, whereas a web based ASMIS can easily be scaled to a higher workload by adding the necessary IT hardware in the backend or simply requesting more resources from the cloud provider with the ease of a mouse click.

Background & Justification for used diagrams & threat modelling techniques

The UML class diagram (figure 2) was used to get an overview of the structure of the hospitals' planned ASMIS domain to assist with the establishment of general requirements including also specific requirements such as threat mitigation measures, that should be implemented. (Ambler 2003)

STRIDE (compare table 1) established by Microsoft twenty years ago including a data flow diagram (DFD – compare figure 3) was selected since it is a very systematic method. Complex systems can be divided into data flows and those are analysed in both ways for any Spoofing, Tampering, Repudiation, Information Disclosure or Elevation of Privilege attacks, hence the word STRIDE. This ensures a high coverage of possible threats and their proposed mitigation even for complex scenarios, which might be a reason why STRIDE is well established within the industry (Howard & LeBlanc 2002)

A UML sequence diagram can be used to show the logic of a use case, to identify bottlenecks within your system architecture (Ambler 2003). Within this document the diagram (compare figure 4) was particularly chosen to visualize the fact, that actions should only be done by authenticated users and that any request altering or providing

data from the SQL backend should be checked for proper clearance, to ensure that only authorized users can get the desired functionality / data.

An attack tree (compare figure 1) was integrated to visualize all threats identified within the STRIDE threat model for protected health information (PHI), which is the most valuable asset within the ASMIS and therefore likely a target of potential cyber-attacks. The attack tree gives a comprehensive overview and visualizes threats in a very understandable way, which helps to communicate the requirements to non-security professionals.

Potential problems incl. cyber threats of a web based ASMIS

An ASMIS without operators does have its limitations. Chances are that a combination of data entered by the patient, will refer the patient to a wrong specialist, which is likely to be prevent by a skilled human operator. Another problem is that patients needing assistance such as elderly people, might want to speak to a human operator and are not willing or capable of using a web based ASMIS at all. Also, a web based ASMIS faces various threats since it contains very valuable assets such as PHI. Those threats are visualized in figure 1.

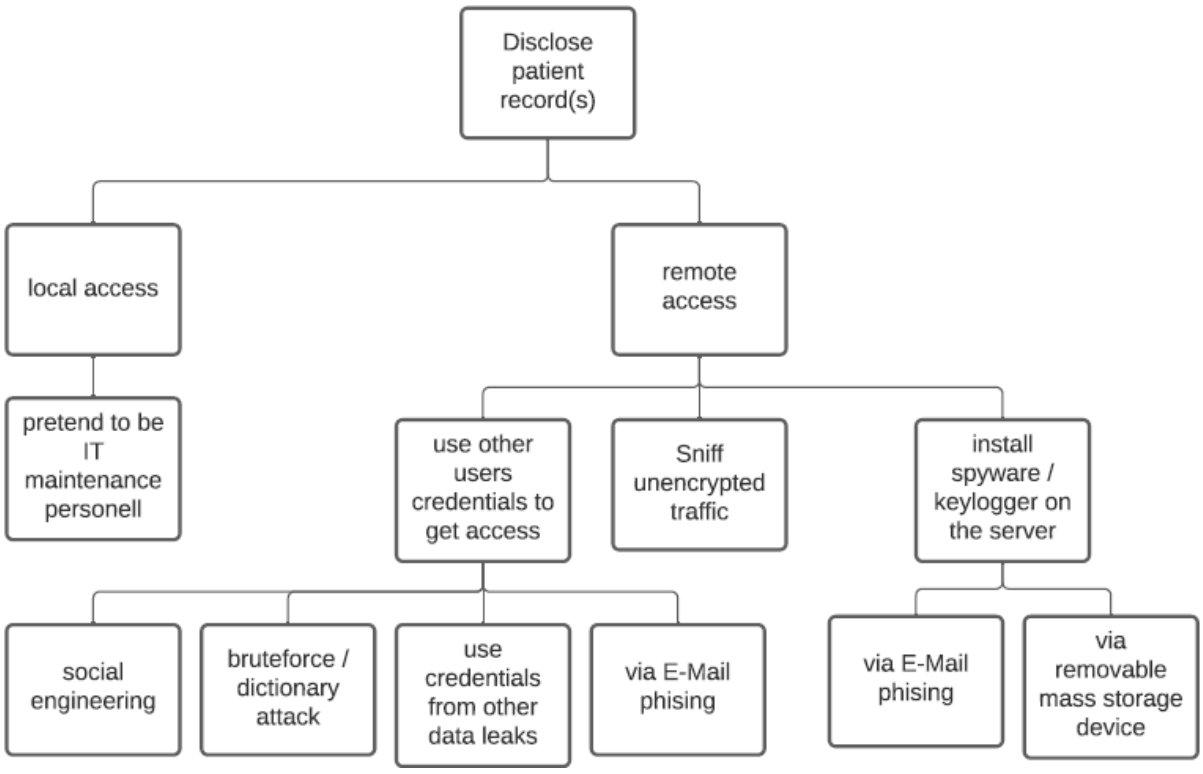

figure 1 – Attack Tree for ASMIS

Figure 1 includes local attack vectors, such as getting physical access to the server room, as well as remote attack vectors. The remote attacks are grouped in three categories within figure 1. The first group exploits credentials of other users of the system. The credentials of others can be obtained via e.g., social engineering, brute force / dictionary attack, credential stuffing (re-using credentials from other data breaches) or via E-Mail phishing. Another group of attack (called “install spyware / keylogger on the server” in figure 1) will aim at the hospital’s IT infrastructure. This infrastructure can either be compromised by staff installing malware send via E-Mail phishing or by plugging in rogue USB sticks, containing malware, into a computer connected to or running the ASMIS. The last group (called “sniff encrypted data” in figure 1) shows that an attacker could simply sniff traffic via the internet, if the traffic is not encrypted & authenticated. This section shows that there are various attack vectors to obtain PHI, but an ASMIS does also have other assets. To provide a list of all attack vectors and their mitigation STRIDE was chosen (compare table 1).

Threat Mitigation

The following section only highlights a few mitigations. A complete list of proposed mitigation can be found in table 1. The following paragraph references the mitigations to table 1 by providing the mitigation id, which starts with a hash tag.

#1 TLS authentication, **#2** TLS encryption, **#3** TLS integrity

TLS is a cypher suite used within the internet. Its predecessor is SSL. TLS can be implemented on the server side and is natively supported by all modern browsers. TLS will take care of encryption of messages transmitted between browser and the ASMIS web-application. This is important since confidential information is transported and should not be disclosed. To prevent man in the middle attacks, which circumvents encryption, TLS also comes with integrated pre-built authentication. This should also be used to prevent information disclosure. TLS also provides integrity checks, which help to prevent any tampering attack. TLS has many strengths such as relatively easy implementation & maintainability as well as a high level of security. Nonetheless TLS can also be very weak, if outdated protocols such as e.g., TLS 1.1, SSL V2.0 are supported by the server or if the implementation is not done correctly (e.g., broken authentication). TLS can also be weak, if the certificate chain is compromised and not mitigated by certificate pinning. TLS / SSL is used as a default

for secure communication by all major legitimate websites such as Amazon, eBay, PayPal. (Howard & LeBlanc 2002; Suttard & Pinto 2011; Yaworski 2017)

#6 traffic filtering

Traffic filtering is important to any device or service connected to the internet. Traffic filtering is typically done by firewalls, which either can be hardware appliances or software instances. Since the ASMIS is web-based a web application firewall (WAF) is a good solution to filter malicious packets from the traffic. WAFs come with many advantages. They can easily be scalable to higher load, they are easy to implement, they can do basic filtering, but they also can perform advanced filtering even of the encrypted portion of the packets received. Unfortunately, WAFs do also have weaknesses. They can be circumvented, and they cannot detect unknown attacks (also known as zero-day attacks). WAFs can be found at some web services nowadays. (Ross 2008; Yaworski 2017; Palka & Zachara 2011)

#9 Authorization

Users (patients, doctors, administrators) should have least privileges assigned to do their tasks. A patient for example should not be able to utilize administrator functions. A patient should also not be able to access other patient's data. To do this user with respective privileges must not only be created and maintained their actions also must be authorized when they act. Without proper validation of authorization, a malicious actor logged in as a user could for example request data, that should only be visible to doctors, such as e.g., a certain patient complete medical history (compare figure 4). To prevent this, it should be checked if the user has appropriate authorization to obtain the data. The validation of authorization should be done anytime a privileged action is done or privileged data is requested along the sequence of events, that define each individual use case (compare figure 4). The proper enforcement of authorization comes with the advantage that escalation of privilege attacks can be prevented, and potential information disclosure can be prevented. The implementation of least privilege is not easy to achieve since authorization validation must often be done within the source code. To achieve full coverage a good testing strategy including code review and automated testing should be used. The biggest weakness of least privilege and its enforcement however comes with maintenance. Anytime an update is created, the implementation and verification of authorization

validation must be realized within any new peace of code and should not be forgotten. (OWASP 2021; Howard & LeBlanc 2002)

#11 Sanitize input to SQL by web

“All input is evil until proven otherwise. That’s rule number one.” (Howard & LeBlanc 2002). This statement and the fact, that SQL injection (SQLi) is still on the top of the OWASP Top Ten list of web application security risks show, that input sanitization is important, but unfortunately not always done, which likely is the cause for various severe data breaches in the past. SQLi prevention can be achieved easily and can completely remedy SQLi as a threat. Input sanitization can be done by the database itself. You can parameterize a SQL database to replace certain characters such as e.g., a single quote with double quotes. The more common approach is however, that the web application replaces characters within an input string with other characters prior to forwarding it to the SQL backend. Sanitization of input to backend databases is employed in many web applications, but unfortunately not all. (OWASP 2021; Howard & LeBlanc 2002; Yaworski 2017; Suttard & Pinto 2011)

#18 Intrusion detection system (IDS)

Part of any IT infrastructure should be an intrusion detection system (IDS). A common IDS tool is a security Information and Event Manager (SIEM). SIEMs need to be configured by security specialists, some also utilize artificial intelligence (AI). They help identify suspicious behavior automatically and flag this to a security operator. The strength of SIEMs is that they can assist the security specialist to do his / her job more efficiently and to identify breaches early. However, most SIEMs are not maintained as they should and do have a negative effect on security in some instances, since they keep IT staff busy with too many false positives. If a SIEM is parameterized correctly and does have a good vendor support on malware signatures they can be very effective, however. SIEMs are deployed in some corporations nowadays. (Ikeda 2021)

#19 database data encrypted at rest

To add an effective layer of defense and to avoid hefty regulatory fines it is recommended, to encrypt data at rest. In this scenario the data can and should be encrypted at the database level with e.g., SQL Server Transparent Data Encryption (TDE) or functions that securely encrypt (e.g., AES256-GCM, AES256-CCM,

RSA2048) the data at the web application prior storage in the SQL database. This backend encryption will make it economically impossible for an attacker to decrypt the database data obtained after a breach. Weaknesses of data at rest encryption typically are key generation and key storage. The best encryption can easily be broken, if they key is hardcoded / integrated in the source code of the application. It is better to store the key securely e.g., within the windows certificate storage or within a hardware secure module. Decryption of PHI is nowadays widely used in a variety of web applications found on the internet. (OWASP 2021; Howard & LeBlanc 2002)

#25 cross-site request forgery (CSRF)

CSRF is an attack, where a malicious website triggers a request from the user's browser such as request patient medical history to the ASMIS. This request will get blocked since the user is not logged in (compare figure 4). But if the user has the ASMIS application running in another browser tab and is logged in, the request will not get blocked, since the ASMIS cannot distinguish from which browser's tab the request was sent from. To prevent CSRF you can use built in CSRF protection available e.g., for Java and AngularJS or implement tokens (e.g., via nonce) to all requests, that cause data to be displayed from SQL backend or changed at SQL backend and verify the validity of the token (compare figure 4). An advantage of token based CSRF protection is that CSRF as a threat is completely mitigated. A disadvantage of token based CSRF protection is however, that it needs to be implemented on various places within the code and might be forgotten. Thorough test scripts and code review, which are time consuming need therefore be conducted. CSRF token-based protection is implemented in many major websites these days. (OWASP 2021; Howard & LeBlanc 2002; Yaworski 2017; Suttard & Pinto 2011)

Summary

In conclusion it can be said that cyber security is not a limitation of a web based ASMIS, if implemented well. This is due to the fact, that all threats can be mitigated to an acceptable level by technological or organisational means and are likely to be more cost efficient than a solution with human operators. Since ASMISs do have the very valuable protected health information (PHI) as an asset, they are likely to face many attacks such as e.g., information theft & ransomware attacks. It is therefore recommended to have a very high cyber security standard. The recommend standard

for cyber security of the discussed ASMIS (compare figure 2, 3 & 4) are outlined in table 1.

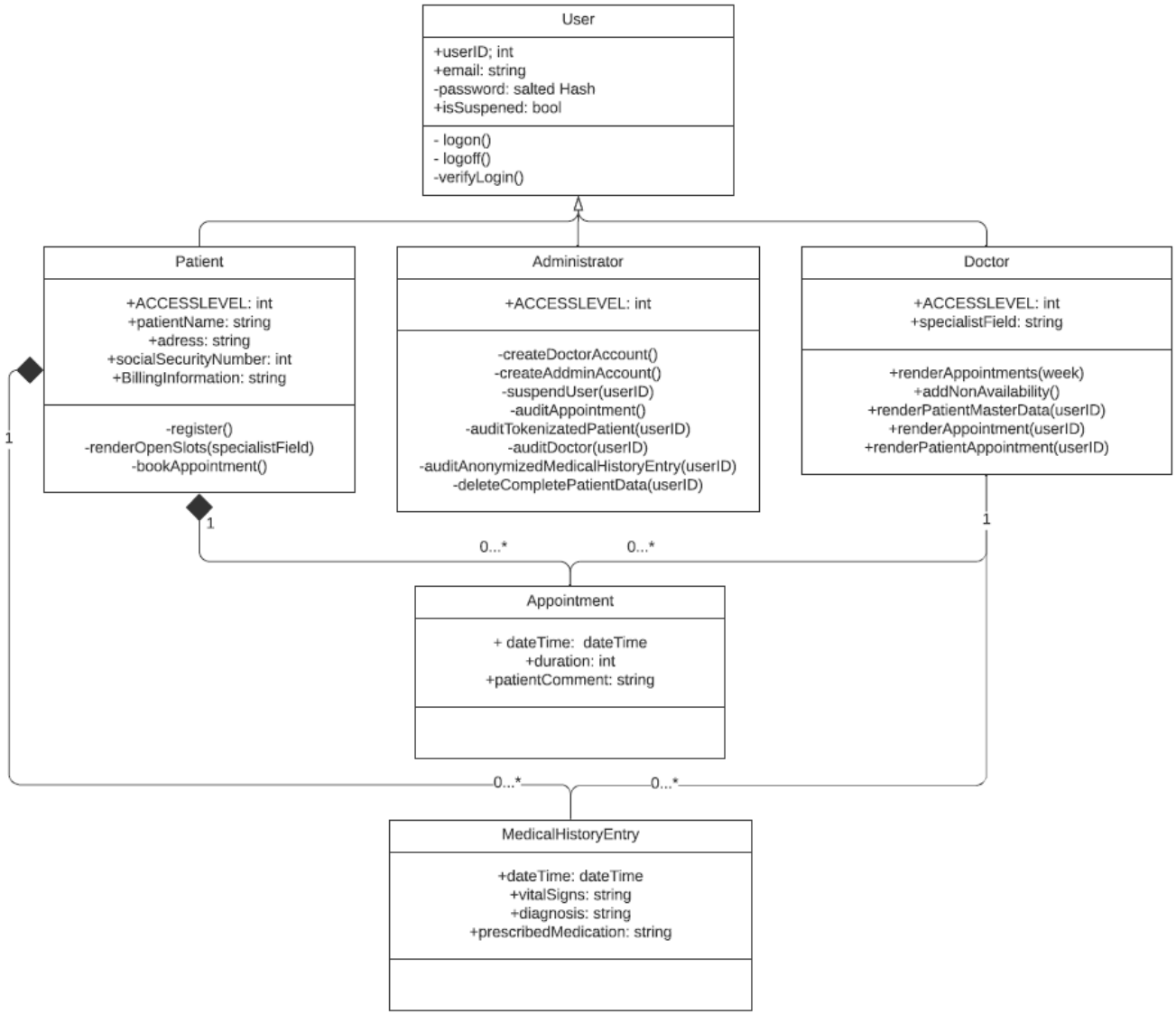

figure 2 – ASMIS UML class diagram

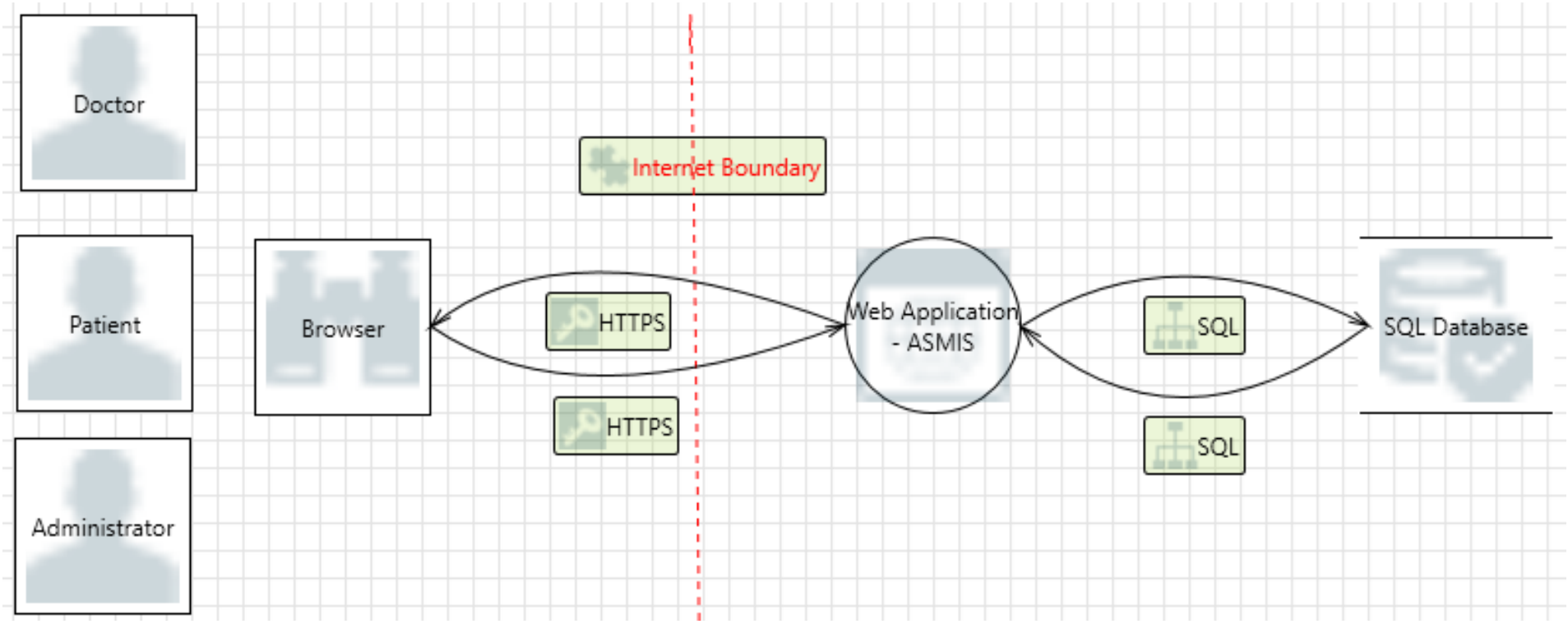

figure 3 – data flow diagram (DFD) of the ASMIS supportive to table 1

Table 1 – STRIDE table for the ASMIS (Microsoft 2021)

#	Interface	Category	Description	Mitigation
1	browser <--> web application (ASMIS)	Spoofing	The client side can easily be spoofed. Someone could e.g. pretend to be a legitimate user and easily intercepted all packages from ASMIS	#1: TLS authentication #8: user management incl. password policy #24: user authentication via registration by user E-Mail, using CAPTCHA
2		Spoofing	Access via stolen user credentials by • Social engineering • Brute forcing / dictionary attack • Credential stuffing • Phishing	#8: user management incl. password policy #6: traffic filtering #26: training of ASMIS staff
3		Tampering	Unencrypted data with no integrity control can very easily be tampered with.	#2: TLS encryption #3: TLS integrity
4		Tampering	ASMIS vulnerable to (stored) XSS	#4: Sanitize input in the web application ASMIS for XSS
5		Tampering	Using known exploits of the underlying ASMIS framework	#16: patch management of ASMIS
6		Repudiation	Web Application - ASMIS can deny having received any info from client	#5: logging source, time transmitted data

#	Interface	Category	Description	Mitigation
7		Repudiation	Browser denies having received data from ASMIS	#5: logging source, time transmitted data
8		Information Disclosure	Unencrypted data can easily be sniffed, since it is transported in the clear	#2: TLS encryption
9		Information Disclosure	Data leakage via e.g., compromised IT infrastructure	#18: Intrusion Detection System (IDS)
10		Denial of Service	Denial of Service attacks	#6: traffic filtering #7: load balancing
11		Elevation of Privilege	Hacker could exploit the logic of the ASMIS to gain higher privileges. Hacker could also exploit poor password policies to gain higher privileges	#1: TLS authentication #8: user management incl. password policy #9: Authorization (e.g., patient cannot use admin functions. #23 Least privilege (Web server does not run as root)
12		Elevation of Privilege	CSRF is possible w/o prior mitigation	#25 Cross-Site Request Forgery (CSRF) tokens
13	web application (ASMIS) <--> SQL database	Spoofing	Someone could install a 2nd fake and malicious SQL database (unlikely, though)	#10: authentication via username & password at the SQL backend #13: lock server room

#	Interface	Category	Description	Mitigation
14		Tampering	SQLi	#11: Sanitize input to SQL by web application ASMIS for SQLi #21: database entries hashed #22: regular database backup
15		Tampering	Using known exploits of SQL server	#17: patch management of SQL server
16		Tampering	Persistent XSS attack, since output is not sanitized	#18: Sanitize output from SQL server by ASMIS web application for any (left) XSS
17		Repudiation	Wrong log data could be fed to the system. SQLi having log data could be fed to the system.	#8: user management incl. password policy #12: encrypted logs
18		Repudiation	Some attacks might need other data to understand the issue	#5: logging source, time, and summary of the received / send data of users within the system

#	Interface	Category	Description	Mitigation
19		Information Disclosure	Data on SQL not encrypted or otherwise secured	#13: lock server room #14: tokenization of data (admin cannot see patient names) #9: Authorization (e.g., patient can only see their data; doctors can only see patient data of patients having an appointment with them.) #19: database data encrypted at rest #20: data minimization: delete expired user data #21: user password stored as salted hash

Table 2 - STRIDE table for the web based ASMIS (Microsoft 2021)

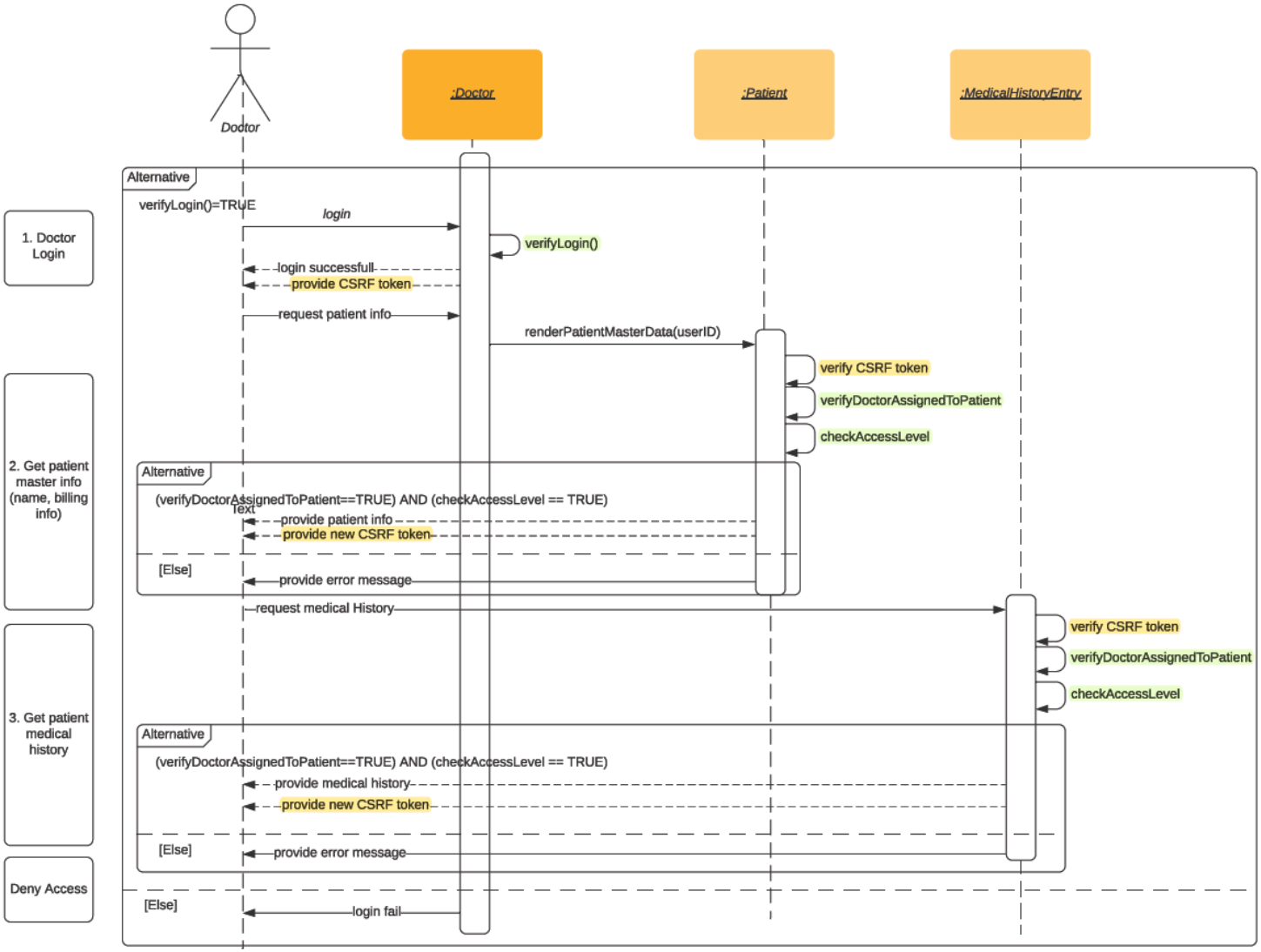

figure 4 – ASMIS UML sequence diagram

References:

Ambler, W. S. (2003) *The Elements of UML™ Style*. First Edition. Cambridge: Cambridge University Press

Howard, M. LeBlanc, D. (2002) *Writing Secure Code*. Second Edition. Redmond: Microsoft Press

Ikeda, S. (2021) *SIEM Rules' Threat Coverage Is Far Less Than What's Expected; 84% of MITRE ATT&CK Threats Are Not Covered*. Available from: https://www.cpomagazine.com/cyber-security/siem-rules-threat-coverage-is-far-less-than-whats-expected-84-of-mitre-attck-threats-are-not-covered/ [Accessed on 18.03.2021]

Microsoft (2021) *Threat Modelling Tool*. Available from https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling [Accessed on 17.03.2021]

OWASP (2021) *Cheat Sheet Series*. Available from: https://cheatsheetseries.owasp.org/Glossary.html [Accessed on 24.03.2021]

Palka, D. Zachara, M. (2011) *Learning Web Application Firewall - Benefits and Caveats*. Available from: https://link.springer.com/content/pdf/10.1007/978-3-642-23300-5_23.pdf [Accessed on 27.02.2021]

Ross, A. (2008) *Security Engineering: A Guide to Building Dependable Distributed Systems*. Second Edition. Indianapolis: Wiley Publishing Inc.

Suttard, D. Pinto, M. (2011) *The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws*. Second Edition. Indianapolis: Wiley Publishing Inc.

Yaworski, P. (2017) *Web Hacking 101 - How to Make Money Hacking Ethically*. First edition. Victoria: Leanpub

Bibliography:

Alder, S. (2021) *What are the Penalties for HIPAA Violations?* Available at: https://www.hipaajournal.com/what-are-the-penalties-for-hipaa-violations-7096/ [Accessed 12.02.2021]

Damodaran, M. (2006) Secure software development using use cases and misuse cases in Issues in Information Systems. Available from https://iacis.org/iis/2006/Damodaran.pdf [Accessed on 17.03.2021]

Fox, C. (2019) *Google hit with £44m GDPR fine over ads*. Available from: https://www.bbc.com/news/technology-46944696 [Accessed 26 Jan 2021]

GDPR (2016) *REGULATION (EU) 2016/679*. Available from: https://eur-lex.europa.eu/legal-content/DE/TXT/?uri=CELEX%3A32016R0679 [Accessed 13.02.2021]

Khan, R. et al (2017) STRIDE-based threat modelling for cyber-physical systems. Available from https://pureadmin.qub.ac.uk/ws/portalfiles/portal/140549879/2017_STRIDE_based_threat_modeling_for_cyber_physical_systems.pdf [Accessed on 17.03.2021]

Koret, J. Bachaalany E. (2015) *The Antivirus Hacker's Handbook*. First Edition. Indianapolis: Wiley Publishing Inc.

Lucas, M. et al. (2006) *Firewall Policies and VPN Configurations*. First Edition. Rockland: Syngress Publishing

Microsoft (2019) *Transparent Data Encryption (TDE*). Available from: https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/transparent-data-encryption?view=sql-server-ver15 [Accessed on 24.03.2021]

Mustafa, A.S., Mohammed, J. M. & Thajeel, A.M. (2015). *Layered Defence Approach: Towards Total Network Security*. International Journal of Computer Science and Business Informatics. 15. Available from: https://www.researchgate.net/publication/321622160_Layered_Defense_Approach_Towards_Total_Network_Security [Accessed on 20.03.2021].

Tschider, C. (2018) *Deus ex Machina: Regulating Cybersecurity and Artificial Intelligence for Patients of the Future*. 5 Savannah L. Rev. 177. Available from: https://ssrn.com/abstract=3070000 [Accessed: 12.02.2021]

Williams, L. et al. (2019) *Secure Software Lifecycle Knowledge Area Issue 1.0*. Crown. Available from:

https://www.cybok.org/media/downloads/Secure_Software_Lifecycle_issue_1.0.pdf
[Accessed 26 Jan 2021]