EITA25 - Project 2
Group 19

Computer Security - Department of Computer Science and Information Technology

Malin Åstrand, Axel Beke, André Frisk and Sebastian Malmström

Fig. 1: [1]

HIGH-LEVEL ARCHITECTURE OVERVIEW

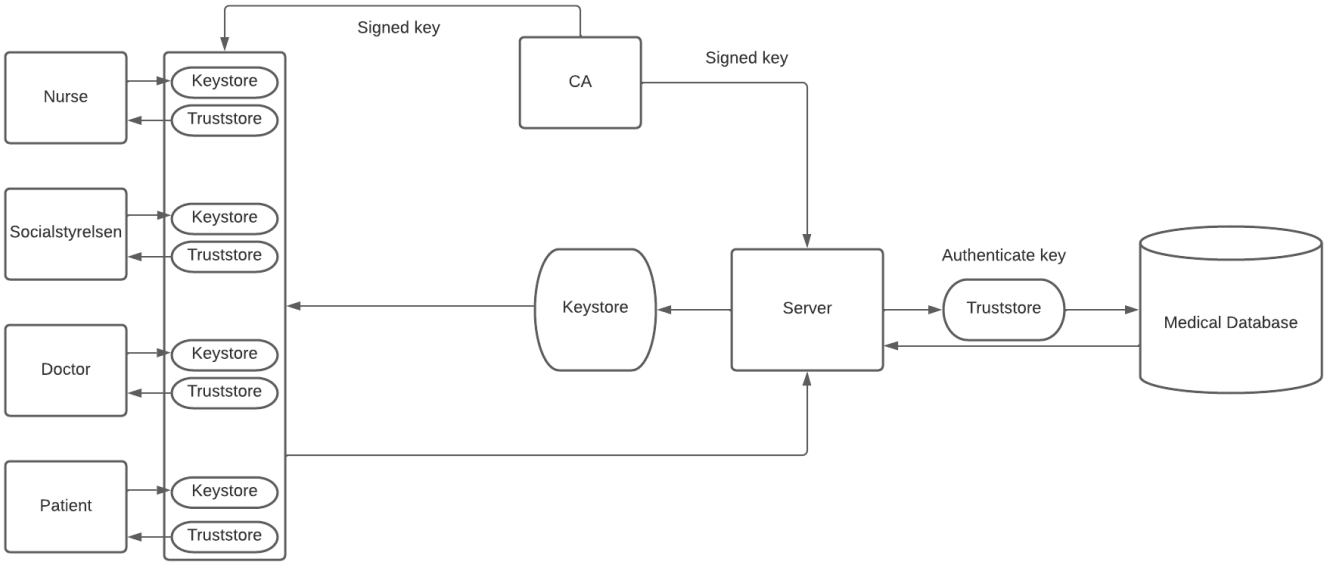

Fig. 2: Architecture of the hospital's medical database security

In order to acquire a system with confidentiality one need to build a secure program which denies access for unauthorized personnel or malicious people. This can be used with *Certification Authority (CA), Keystores & Truststores*. These objects and methods will protect the medical database which contains sensitive information about patients and why they are at the hospital.

A certificate authority (CA) is a trusted organization that verifies websites (and other entities) so that you know who you're communicating with online [2]. Their objective is to make the internet a more secure place for organizations and users alike. This means that they play a pivotal role in computer security. The CA works like a passport – but for websites and online activities. After the passport has been verified you get a stamp in your passport, in this case after a verification of a website the CA issues a digital certificate which proves who they are. Then a signed key (used for encryption and decryption) can be used to authenticate users. The CA certificate is installed on both the client and server truststore and all certificates in the system must be signed by the CA.

Keystores and truststores are repositories that contain cryptographic objects like certificates and private keys that are used for cryptographic protocols such as TLS [3]. A keystore contains personal certificates, plus the corresponding private keys that are used to identify the owner of the certificate. The truststore contains the signer certificates which tells which users the endpoint trusts. A signer certificate represents a certificate and public key associated with some personal certificate. The purpose of the signer certificate is to verify personal certificates. By accepting the signer certificate into an endpoint's truststore, you are allowing the owner of the private key to establish connections with this endpoint. The signer certificate explicitly trusts connections made to or by the owner of the associated personal certificate.

The communication between the client and the server must be encrypted and established, this is done using the network standard TLS. TLS is a cryptographic protocol which is used to provide secure communications over a computer network [4] to make it secure. TLS is designed to secure data against hackers and helps ensure that sensitive information are safe. TLS provides with confidentiality, integrity and authenticity with the use of certificates. The TLS protocol is popular in client-server applications to prevent eavesdropping and tampering, this is done through encryption of the data sent. This is important as the system used in this project also uses TLS to create a secure connection between the client (which is one of the actors at the hospital) and the server which handles the medical database. The secure connection

is established in a TLS handshake where the client and server together specifies which version of the TLS they will use, decide which cipher suites (encryption algorithms) they will use, authenticate the identity of the server via the server's public key and the SSL certificate authority's digital signature and generate session keys in order to use symmetric encryption after the handshake is complete.

In this system there are four different actors for the access control scheme: *patients, nurses, doctors & Socialstyrelsen*.

- **Patient**: The patients is only allowed to read their own list of records.
- **Nurse**: The nurses may read and write to all records associated with the nurse and also read the records which is associated with the same division at the hospital.
- **Doctor**: The doctor have the same access right as nurses but have the right to create new records for a patient provided that the doctor is treating this patient, a nurse is associated to this record by the doctor.
- **Government Agency**: The government agency called *Socialstyrelsen* is allowed to read and delete all records at the medical database.

Actors which is not one of the four titles won't gain any access at all.

The Access Control scheme, which is a Attribute Based Access Control (ABAC) scheme, is divided into different parts. Each actor have their own keystore and truststore. The CA is used to sign specific certificates which then are stored in the different stores. CA is stored in the truststore to then be used to decide if you could trust the incoming certificate. A certificate is only added to the truststore if it has been signed by the CA. The keystore as mentioned above contains the information about the actor (for example a doctor called Doctor Maria, which common name (CN) is Doctor.Maria) and their respective keys and certificates. These keystores communicates with the corresponding actors truststore which identifies and gains a trustworthy connection. This is done for every actor in the system.

The client and the server utilizes two-factor authentication. When starting the program the server requires a input of which keystore to check, in our case e.g *doctor1*, *patient2* & *government* as the program searches for the specific keystore with that name (e.g doctor1keystore). The server will look up the associated key from the clients keystore and check the associated certificate in the server truststore when a client is trying to access the medical database. If the certificate from the CA is in the servers truststore and the correct password is inserted to the keystore then the client will gain access, otherwise the connection won't be set up successfully and the client can't access the medical database. When the correct password is inserted the server and the client begins the TLS handshake to gain secure connection. When connection has been established the server checks which CN the keystore has and uses a separator to take out the first word of the CN, which is the role used in the ABAC (e.g Doctor.Maria is a Doctor). When the role is determined the client will gain a very simple text-based user interface with that specific roles actions listed. All actions that exist in this system is *read, write, delete, create, exit*. These actions are commands which is sent from the client to the server which is handled according to the specific action. When the action is done a result message is then sent back and printed in the terminal. If a role is trying to do a operation with a medical record which is not assigned to them or outside their department then access is denied. The program constantly checks permissions, files and the database to determine the permissions. Every command that is sent between the client and the server is logged with the timestamp, the actor who sent the command and which medical record was being read, modified, created or deleted.

Every actor has specified information in our database about their role and what conditions they have in a file called *Database* with collected medical records of every patient, e.g every patient has a medical record where it is stated what name they have, what department they are hospitalized in, which doctors and nurses which has been assigned to the patient and the status of sickness.

```
Name; Division; Doctor(s); Nurse(s); Sick
Sven; Heart; Holger; Fredrik; Very sick
```

Also, every nurse and doctor is assigned to a department. This is stated in the *UserAccounts* file. The third file is the *Logs* file which logs every command that is used in the communication between the client

and server. These three files build up the ”database” of this system. This is not a real database as we have not had a course in database technology yet.

Ethical Aspects

The trade-off between confidentiality and availability is an important one. By implementing a lot of different security systems such as passwords, two-factor authentication or tags you increase the confidentiality and security of the system. What has to be kept in mind is that this also increases the amount of hoops the end user has to jump through to use the system for their use-case. This might make the user try to skip out on a few of them to try to make their life easier. For example using shorter password so its quicker and less error-prone to enter, sharing accounts between users to make it so they do not have to go through the whole process to swap user or simply using an administrator account for the system. Therefore it is important to think about how and in what situations the user will access the system. Is it in a high stress environment where access has to be quick or can the access process be a bit slower and intricate but still be accepted?

As different parts of being a doctor or nurse requires different kind of information and how fast access is needed you would ideally make different authentication ways for different situations. For example having a tag or biometric system for the information that is needed at a higher speed, while keeping most of the information safely behind more levels of authentication. This would help protect the integrity of the patients by protecting most of their records behind a much more tougher means of access while also helping the doctors and nurses access the information needed for quick decisions in the heat of the moment. Even if someone for example drops their tag the amount of information an outside threat could get access to is limited.

When choosing between a biometric system or tag system the trade-off is between the availability of the system and the security. Someone can steal a tag but stealing someones biometric data is a much harder task. Todays biometric scanners have an error rate where the biometric data can overlap enough between two users that one that should not be authenticated is allowed access. There is also a problem of not letting an user access a system by mistake as the biometric data scanned is not always perfect. In a situation where the doctor or nurse need fast access to patient information the choice should with todays technology be tags. Even if there is a risk of losing it or having it stolen it outweighs the risk of not getting access in the heat of the moment because you have a hard time placing your finger correctly on the scanner or holding your eyes still for it to be read by the scanner. Even more so in a stressful situation.

To implement a system like this Attribute Based Access Control (ABAC) would be the best system. It has the flexibility to both check what role, what record and how you are trying to access it before giving access. If the implementation would go the way of having different kind of information for different kind of access methods the flexibility would be key for making it work. Systems like mandatory access control (MAC) or discretionary access control (DAC) is either way to rigid and too few options for users and resources or they leave the option for resource owners to choose who has access. To not have any confidential information regarding patients leak the choice of who gets access should be automatic and centralized.

Our system is much more focused on the security. We have a keystore with a password to access it. So the advantage for our system is that someone can not steal or find a tag and then get into the system. If you work with biometric locks you instead have the problem of false-positives and false-negatives, so that is more accessible than our system in most cases but with the outliers it could instead be to the detriment of the user. As the data in the system is necessary for a doctor to do their job it can become a problem if someone always have a hard time to access it. What a tag system misses is the authentication of the person. Our log in require the use of a signed certificate that makes sure of the identity of the user. With a tag or a normal password process this is not something you can guarantee. To use someones certificate you both need access to their system and know their password to unlock the keystore.

When looking at the three different actors administration, engineer/security expert and hospital staff everyone tries to work towards the same goal, a secure system. The difference is their other agendas.

While the administration can be concerned with keeping the project inside of the budget meanwhile the engineer want to build a safe system. During all of this the staff want a system that is easy and quick to use for their work. All of these opinions need to be taken into account when designing the system.

A. Hospital administration

The administration is concerned with following the current laws regarding how to store patient data. As this is generally one of the kinds of data people are the most concerned about to keep their integrity it is of the utmost importance to the administration to keep it safe. If the data is leaked, either by accident or on purpose, it could have large consequences for the hospital itself. Because of this the concern of staying inside the budget while also having a secure system is what the administration has to balance. Even though they could build a very secure system by spending a lot of money this could get them into trouble with for example investors or the board, because they went over budget.

B. Engineer/Security Expert

The engineer wants to build a system that is both secure but also easy to use for the staff. As most security implementations makes it more time consuming and/or harder to use for the staff these two need to be balanced. To come up with the balance you have to compare the sensitivity of the data to the measures being taken to secure it. If you make the staff jump through too many hoops to get to data that they deem not so sensitive you could get users that work against the user. As talked about previously they could then start sharing passwords or not switch users cause the process takes too long. Then your security implementation is useless anyway. By going too far the other direction you instead risk data breaches which could get you and the hospital in problem with the regulators or the administration of the hospital. It ends up in a balance of the privacy, availability and system vulnerabilities.

C. Hospital Staff

The hospital instead is concerned with patient health, availability and privacy. If the staff needs some information regarding the patient quickly the more privacy is a concern the harder and more time consuming it could be to access it. For example with allergies and the likes, if you print it on a chart and leave it out in the open the availability is high but the privacy of the patient is lowered. What you could do is implement physical security so that only vetted people can access that space. Also following policy and routines to protect privacy and health of the patients could lower the availability by requiring a process to happen before you can do what you want to.

SECURITY EVALUATION

D. Two-factor authentication

In this project we have implemented basic two-factor authentication when connecting to the server by requiring both a key signed by the CA contained in the server's truststore, as well as a password protected personal keystore. In theory this means that the potential intruder needs to have access to the client's keystore and know the password in order to establish a connection to the server. Downsides to this type of authentication is that the employees might choose weak passwords or that the time consumption of entering a password could have negative consequences in a hospital. Other possible solutions, like biometric or tag-based, are discussed in the Ethical Aspects.

E. Cipher suites?

When you establish a TLS-connection you are able to chose from many different supported key agreements, authentication methods and ciphers, and you can specify which ones to use in a cipher suite. The server and the client will have lists of available cipher suites and during the handshake, the first one that is featured in both lists will be chosen. The cipher suite identifier consist of multiple field separated by underscores. One example could be *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384*, where TLS is the protocol, ECDHE is the key exchange algorithm, ECDSA is the authentication mechanism during handshake, AES is the session cipher, 256 is the encryption key size of the cipher, CBC is the type of encryption, and SHA384 is the hash function (in this case the digest size is 384 bits). Some cipher suits are not recommended because they contain outdated means of encryption. Examples of bad cipher suites are ones that use key sizes that are smaller than 128 bits or if they use an already cryptographically broken algorithm such as MD5. To change which cipher suite that is chosen you need to either disable the cipher suites that you want to avoid, or also enable a better cipher suites to your server and client's lists. Another way of controlling which cipher suite that is chosen is by using TLS1.3, which does not have the same legacy support as TLS1.2. This means it is not compatible with older, more unsafe, versions.

F. Attacks

1) Brute force and dictionary attacks: A brute force attack is a potential threat to our system. By using two-factor authentication in the authentication process the system is protected against this type of attack. The only way for an attacker to gain access to the server is by being in possession of a valid certificate and the password to it. Brute forcing the password to a certificate is possible and is thereby a security threat to us. In order to add protection to our system against this, another scheme has to be added with usernames and passwords. These would then be hashed and stored in our database. This is one way to further increase security, though we have not implemented this.

In the same way that the system is protected against brute force attacks, dictionary attacks fall under same protection. The difference between the two being that dictionary attacks use a set of commonly used passwords and dictionaries to guess the password. Dictionary attack software also use variations of the words. By using a set of words reduces the amount of phrases that is being tested significantly and makes it faster than brute force, given that the password is not complex. In the situation of a dictionary attack, our implementation of two factor authentication will protect very well.

2) Rainbow Table attacks: A rainbow table is a very powerful attack which can be devastating if not properly protected against. The best and most widely used way to protect against these types of attacks is by using salts on passwords before hashing them. A salt is a string that gets appended to the beginning of the password making the hash unique. Subsequently making the hash for two identical passwords different. Thus, a rainbow table for each salt is required which is computationally exhaustive which in turn makes salting an effective counter measure. In rainbow table attacks, it is the creation of rainbow tables that takes much time and memory. Generally one have to be in possession of the hash to use this type of attack. The password an attacker would have to crack is connected to a key store and the hash is not available. Because the hash is not available, the only attacks possible would be brute force- and dictionary attacks because of the possibility of entering passwords repeatedly.

3) Man in the middle: Another potential threat is a man in the middle attack. A man in the middle attack is when an attacker is positioned between the user and server, intercepting or altering messages between them. The messages intercepted could potentially be usernames and passwords, making this attack a security issue. The attacker could also pretend to be the server, fooling the user into believing they are communicating with the right one, making leaking passwords a possibility. The way to defend against it, is to use some type of authentication, so the server can prove it is the server and a can user prove it is the user. In our network system we require the use of digital certificates issued by a trusted certificate authority. That way, a user can prove his authenticity by being in possession of the certificate and knowing the password to it. The same goes for the server. [5]

4) Denial of service: In a denial of service attack the attacker is restricting users access to the service. The attacker does this by flooding the network with traffic until the server either crashes or cannot respond. DDos-attack, more widely known, is a denial of service attack where the attacker uses a network of infected computers to simultaneously flood a target. As of now, no simple and at the same time efficient counter measure exists, making it a troublesome threat for smaller systems. Because there are no reasonable protection against this, we were unable to protect our system against this. [6]

5) Sniffing: Sniffing attack is when you capture the traffic being sent over a network. If the traffic is not encrypted in anyway the person capturing the data can read it as plain-text and by using a sniffertool the attacker can in the end gain information enough to crash or gain access to the system. To stop this you encrypt the data being sent over a network. In our application we use TLS to encrypt the data. To decrypt the attacker need a private key that should be kept safely at the users machine. If the attacker succeed in gaining access to the private key it can decrypt and read all the data being sent.

6) Code Injection: Injection attacks exploits potential vulnerabilities in the software where the system asks for user input and the hacker provides an input that the system interprets as code. In order to protect your system against this type of attack it is important to always assume that the input is malicious. Examples of ways of protecting the system could be to only allow certain inputs or to use prepared statements which forces commands and user input to be written separately. A program written in a static typed programming language also provides security since it has language separation. Our server and database is written in Java which is a static typed language, which means no further protection is needed.

7) Data leak: Data leaks can be divided into intentional data leaks and unintentional data leaks. Intentional data leaks are malicious and committed in order to bring sensitive data outside of the organization. Intentional attacks can be made by both cyber criminals as by people working inside the company. Unintentional leaks are also made by insiders but this time there is no malicious intent. It is important to protect our system against both types of leaks. In our system we use a role based system which restricts access. This in theory means no un-authorized users should be able to access information outside of it's scope. We have log files which can be used to investigate suspicious behaviour. There is currently no encryption of the files in the database, but as stated in the assignment, the server is stored behind locked doors. This means it is important to establish policies in the hospital about how the room is accessed. There is still a risk that someone with a high access level leaks classified information. This is hard to combat, but logs makes it easier to find out who is responsible of the leak.

8) Social Engineering: Social engineering is the act of manipulating users of the system in a way that grants the hacker access to confidential information. In our case this could be done by impersonating an authority, in order to be let in into the server room, or by phishing which could give the hacker login credentials to an authorized user in the system. Social engineering is not really applicable in our case, but the solution could involve training of the hospital employees, and good policies surrounding door access.

PEER-REVIEWS

Peer-review from group 20 on the report written by group 19

Overall a good report but there are some improvements that could be made. Sources 5 and 6 are not referenced to in the text and should therefore not be included in the report. The language in the report is for the most part formal but the instances of informal language makes the report feel less professional. For instance on page one “that they are the real deal” sounds very unprofessional and affects the whole feeling of the report.

High-level architecture overview

In this part of the report you describe the usage of certificates well but miss the technical architecture of the program. How does information flow between client and server? A short description on how your server and client are implemented would provide this. Some more specific things that could be improved. In the paragraph starting with “The communication between the client” sentence two is incomplete. In the same paragraph you state that TLS must be used but not explanation why this is the case.

In the paragraph starting with “The access control scheme is divided into different parts” you say that the key- and truststores are signed by CA. This is not true, CA is used to sign specific certificates that then are stored in the different stores. CA is stored in the truststore to then be used to decide if you could trust the incoming certificate.

Ethical overview

“Even though they could build a completely secure system by spending a lot of money this could get them into trouble in other ways” – Is it possible to build a completely secure system, especially one that can be accessed by a large number of people? There is no explanation to what kind of trouble you refer to. This should be added.

“As different parts of a doctors or nurses job requires both different speed of access to and amount of information you would ideally split it up in separate parts.” This sentence is difficult to understand. In the paragraph starting with “Our system is much more focused on the security.” you discuss the pros and cons of the described system and yours system. But in the middle you start to discuss pros and cons of biometric locks, not comparing this to your system.

Security overview

It is unclear how the login process works and how two factor authentication is implemented. In the security evaluation it is mentioned that the user needs the username and password that belongs to the used certificate. However, there is no mention of this when the login process is described in the architecture overview.

It is outside the scope of the project but not impossible to protect a system against a DDoS attack. See DDoS mitigation - Wikipedia

Social engineering: Even though there are no real employees that can be trained or door access policies that can be put in use, it could be interesting to discuss how this could be done instead of simply dismissing it.

Unclear language or typos: With double password, To use someone certificate, inside of the budge, you could get users that work against the user.

Feedback Group 19

Structure:

Overall the structure is good and it is easy to find where certain information is. No reference to figure 2.

Language:

Please read through the report twice when submitting it as there are several grammatical and spelling mistakes. The language does not feel formal enough for a report at this level. Some notable mistakes are:

- "The real deal" should not be used in a report.
- "dictionary attacks are fall under"
- "A certificate authority (CA), also known as a certification authority"
- "standard TLS. is a"
- "cracked, the two factor authentication" - sentence cut off

Readability:

The margins are very thin, and combining that with the small font hurts the readability of the report.

Possible improvements:

The authentication process is not well described in the high level architecture. The description of how the keystores communicate does not tell the reader how or when this is done.

It would be nice if you gave some examples of why availability is more important than confidentiality.

You could also go into more detail regarding your suggested access control. What type of access control is it? DAC, MAC, BAC or ABAC? How does the suggested access control differ from the one you used?

Is the system really protected against brute force attacks though? What if the attacker uses some other software to brute force the keystore password, assuming they have the keystore? They would gain access to the system unnoticed.

Is it certain that the best cipher suite featured in both lists will be chosen? It could be worthwhile manually setting the cipher suite. Otherwise one of the unsafe cipher suites still available in TLS 1.2 could be used, which would be a security issue.

Positives:

The thoughts about using biometric authentication are interesting. But how will it be implemented according to the access control methods used?

The explanations on keystores, truststores and certificates are well written, and allows the intended audience to understand that part.

IMPROVEMENT-SHEET

Here is a summary of the improvement to the report which is based on the peer-reviewed received from other groups.

G. High-level architectural overview

Added some clarification about what TLS is and why it is used in the project. Also corrected some information about the CA's connection with the key- and truststores.

Fixed the explanation about the authentication process and added information about the access control scheme.

H. Ethical discussion

Made it clearer the meaning of some sentences. Also made it more clear how some parts fit into the larger meaning of the paragraph. Added parts about DAC, MAC and ABAC when designing the optimal system.

I. Security evaluation of the design

Added some clarification on the subject of cipher suites. Added missing part about two-factor authentication. Changed some regarding brute force attacks against our system as well as how rainbow tables could be a potential risk.

FUNCTIONALITY REVIEW FORM

LUNDS UNIVERSITET
Lunds Tekniska Högskola

EITA25 Computer Security
Functionality Review Form
Spring 2022

How to perform the review

The group whose work is being reviewed (reviewee) is denoted 'Group E' in the sequel. The group that is performing the review (reviewer) is denoted 'Group R'. The review process is as follows.

1. Group E demonstrates that their TLS connection is fully functional.
2. Group E demonstrates their two-factor authentication.
3. Group E demonstrates that their access control scheme is implemented correctly.
   Group R specifies at least one positive and one negative usage scenario per user type. That is, one usage scenario that should work (doctor adding a record to his patient) and one that should not (patient deleting a record).
4. Group E demonstrates their audit log functionality.

Review report

Group R fills out the relevant information in this section.

Group R number: 20 Group E number: 19

Group R hereby confirms that the project implementation of Group E is of sufficient quality and complies with the project requirements.

Group R signatures:

Lund, 2022- 03 - 02

CONTRIBUTION STATEMENT FORM

LUNDS UNIVERSITET
Lunds Tekniska Högskola

EITA25 Computer Security
Contribution Statement Form
Spring 2022

Individual project contributions

Group number:

name	contribution
André Frisk	Programming (minor) peer-review High-level architecture overview
Malin Åstrand	Programming (minor) peer-review Security evaluation of design
Axel Beke	Programming (major) Security evaluation of design
Sebastian Malmström	Programming (major) Ethical discussion

All group members have actively taken part in and contributed to the project in sufficient part.

Member signatures:

André Frisk Sebastian Malmström

Malin Åstrand Axel Beke

Lund, 2022- 03-02

REFERENCES

[1] Digital Uppercut, *The "Less is More" Approach To Computer Security*. 2018. [Online image]. Available: https://www.digitaluppercut.com/2018/03/the-less-is-more-approach-to-computer-security/. [Accessed: Feb. 14, 2022]

[2] *Casey Crane*, Security Boulevard, *What Is a Certificate Authority (CA) and What Do They Do?*. 2020. [Online]. Available: https://securityboulevard.com/2020/08/what-is-a-certificate-authority-ca-and-what-do-they-do/. [Accessed Feb. 22, 2022]

[3] IBM, *Keystores and truststores*. 2022. [Online]. Available: https://www.ibm.com/docs/en/zosconnect/3.0?topic=ee-keystores-truststores. [Accessed Feb. 22, 2022]

[4] Wikipedia, *Transport Layer Security*. 2022. [Online]. Available: https://en.wikipedia.org/wiki/Transport_Layer_Security. [Accessed Feb 22, 2022]

[5] Nist, *man-in-the-middle attack(MitM)*. [Online]. Available: https://csrc.nist.gov/glossary/term/man_in_the_middle_attack. [Accessed Feb 22, 2022]

[6] Cybersecurity & infrastructure security agency, *Understanding Denial-of-service Attacks*. 2019 [Online]. Available: https://www.cisa.gov/uscert/ncas/tips/ST04-015. [Accessed Feb 22, 2022]