Linnéuniversitetet
Kalmar Växjö

Linnaeus University

Computer Security
Assignment 2
Threat Modelling document

Group Members: [redacted]
Firas Moussa

Table of Contents

1. Introduction

This document will present a Threat model for an e-voting system. It explains the current architecture of the e-voting system and shows how the data flows within and between the different parts of the voting system. The document will present current threats within the system as well as possible controls for the threats.

2. List of Assets and DFD Elements

This section describes the list of assets i.e., items or areas that the attacker would be interested in. Identifying entry points to see where a potential attacker could interact with the application.

1. *Web Server / API*
2. *User: External Entity*
3. *Authentication Process*
4. *Vote Counting Process*
5. *Application DBMS*
6. *Company DBMS*
7. *FIDO – Token*
8. *Data in Transit*

2.1 DFD Diagrams

2.1.1 Registration

The following diagram shows the flow of data during the registration process. It starts by the user requesting registration to the web server.

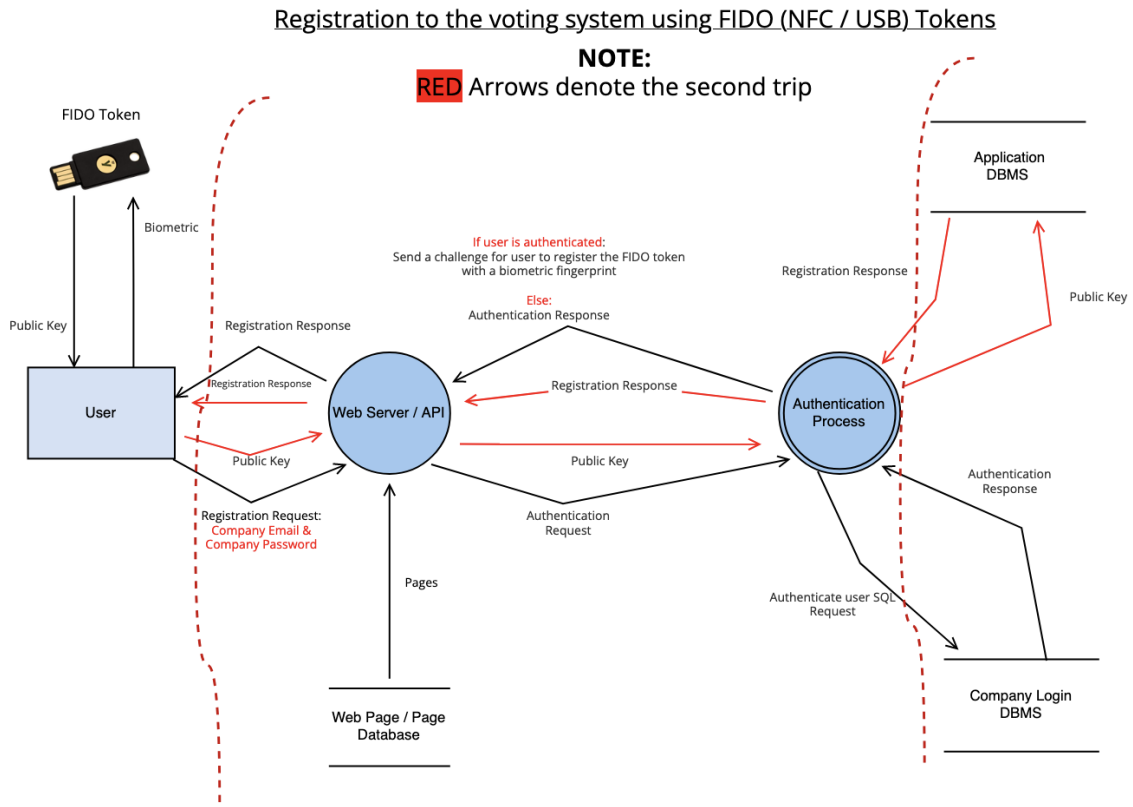

Figure 1 *Registration DFD*

When users visit the voting page for the first time they will have to register. The registration process will start by users logging in with their company email and password. The authentication process

will check with the company database to see if the credentials match to ensure the user is an allowed voter. If the authentication process is successful, the user will be prompted to input FIDO token. The public key of the FIDO token will be stored on the application database. By saving the public key on the application database the authentication process will not have any data exchange with the company database when user's login. This results in a vote which cannot be traced back to its voter and a reduction in the risks associated with communication to the company database.

2.1.2 Authentication

Following is a data flow diagram which shows the flow of data during the login process to the voting website.

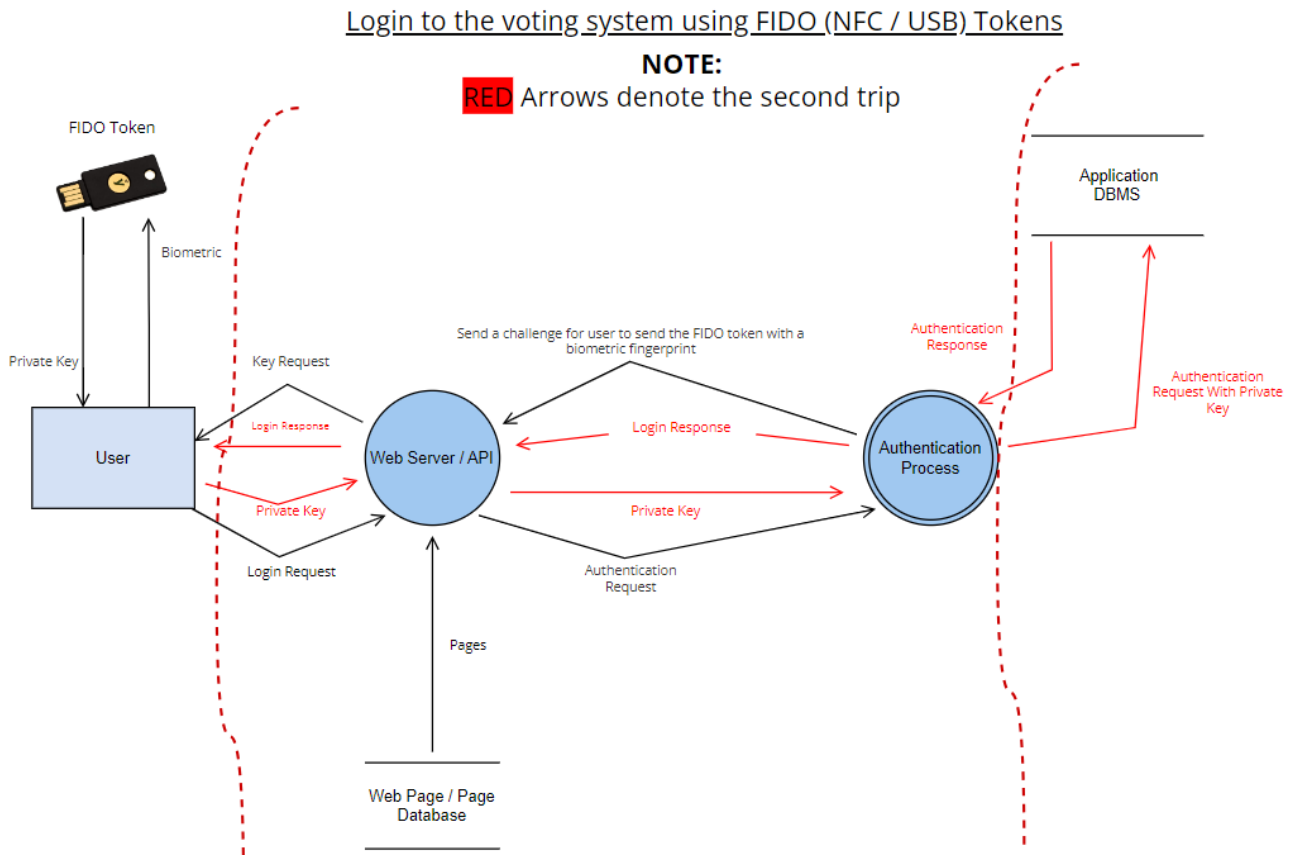

Figure 2 *Authentication DFD*

When the voter opens the webpage, they will use their biometric authentication on the FIDO-token as input. The web page will send the private key to the Vote DBMS which will check if the public key stored match with the private key. If they match the authentication process will be successful and the user will be logged in to the website.

Similarly, on the DRE the user will be able to login through scanning the FIDO with the help of the NFC technology and using their fingerprint as usual.

2.1.3 Ballot generation and confirmation

Following is a data flow diagram which shows the flow of data when authentication is successful and user press vote. It starts with a ballot request from the user.

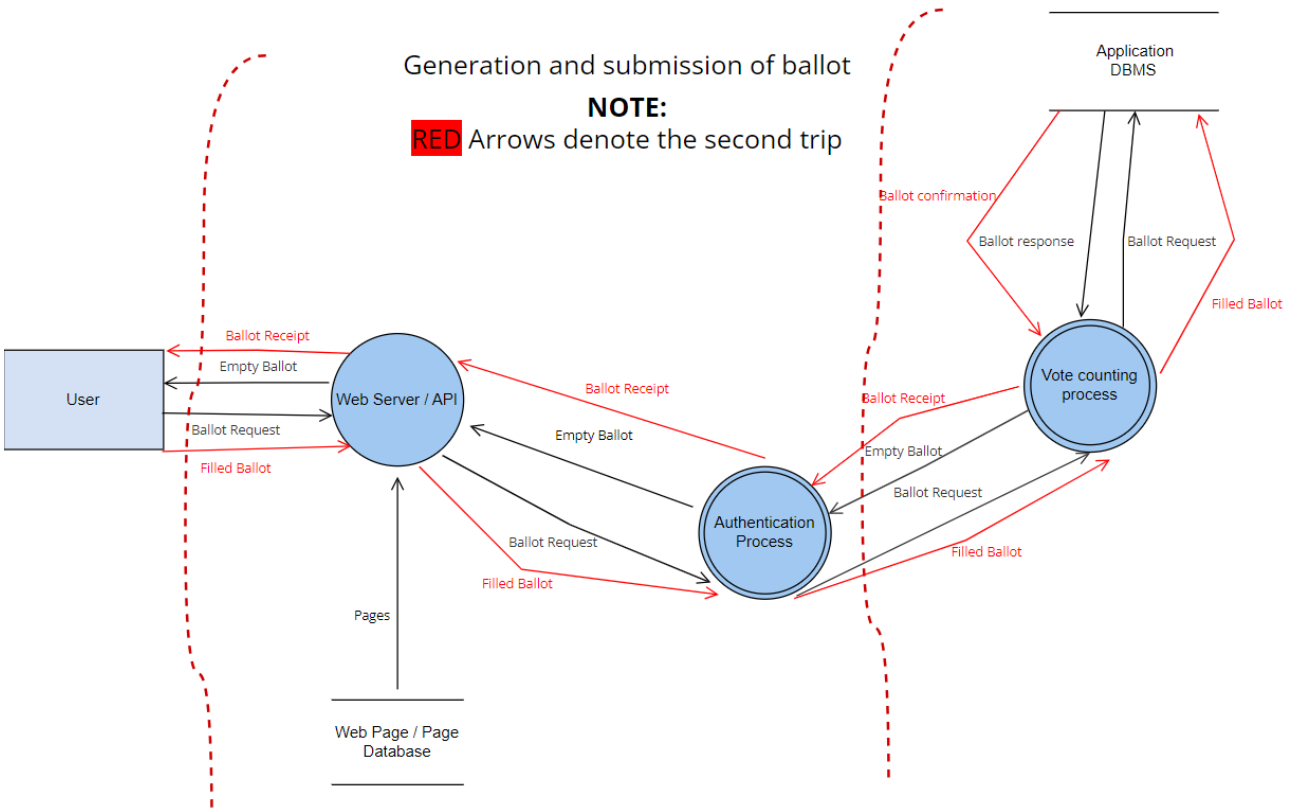

Figure 3 *Ballot generation and confirmation DFD*

After the login request has been granted the voter will send a request to vote. The request will go to the vote management process, which will check if there is a Ballot assigned to the user or not. If the user has voted already their submission will be sent back for viewing, if not, the vote management process will create an empty ballot and send it to the user through the web page. When the voter has filled the ballot, it will be sent back to the database and be stored. The vote management process will generate a receipt which will be sent to the voter as confirmation that the vote has been registered.

2.1.4 Vote processing

When the election is finished, the vote management process will automatically post the result to the web page to avoid any middle hand which would create more threats. The following diagram shows the flow of data when the results are viewed. The dataflow starts with a result request from the user.

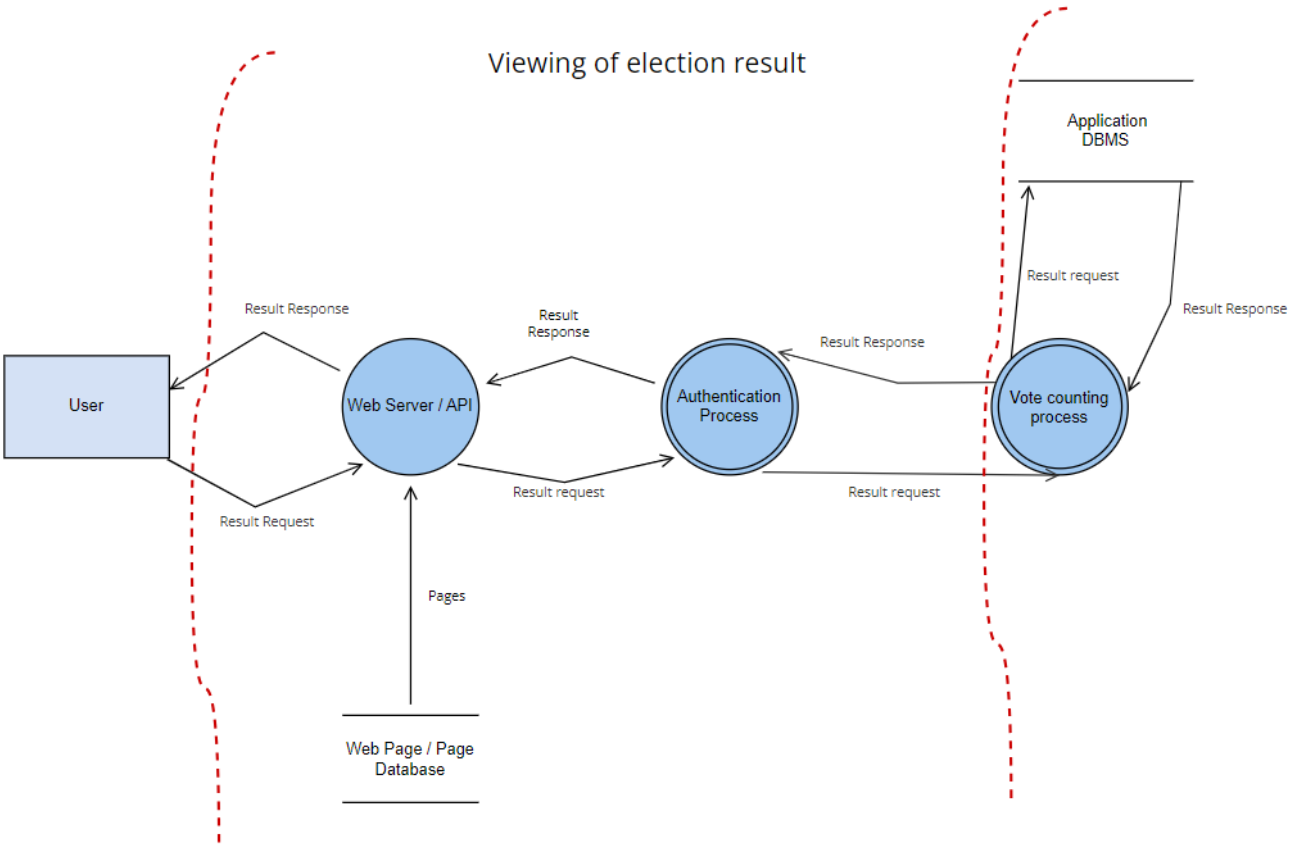

Figure 4 *Vote Processing*

When a user login on the website after the election is finished, the user can view results. The request will be sent from the website to the database with votes and the vote management process will request the result from the database and send it to the voter.

3. STRIDE per element of DFD

This section presents the STRIDE threats and mapping to elements of DFD. Also, the reasoning of mapping. Table 2 shows the mapping table.

Element	S	T	R	I	D	E
User	X		X			
Data in Transit		X		X		
Application DBMS	X	X	X	X	X	
Company DBMS	X	X	X	X	X	
Authentication Process	X		X			X
Vote Process		X	X	X		
Web Server / API	X	X		X	X	
Fido - Token	X					

Table 1 *Mapping of STRIDE per DFD Element*

3.1 Reasoning of Mapping

3.1.1 Spoofing

Password disclosure
The login component of the e-voting system is secure, utilizing FIDO authentication and biometrics. These methods significantly reduce the risk of identity spoofing for registered voters. However, vulnerabilities exist during the registration phase. If a malicious actor gains access to a voter's password, they could register using the voter's company email and password, then link their own FIDO authentication. This allows the malicious actor to cast votes under the legitimate voter's identity, potentially remaining undetected if the real voter doesn't register.

Database spoofing
A malicious person could create their own database server potentially connecting to the *database* to spy on other clients making requests to the database. For this to happen a malicious person would have to replace the actual database server with their own to alter incoming requests and intercept data.

Process requests
Authentication is crucial when processes across trust boundaries communicate. A malicious actor could potentially deceive the application or web server into believing false information is legitimate. Such deception could lead to system malfunctions or unauthorized actions.

FIDO-Token
A FIDO-Token holds information such as the private key which can be used to login to the application. If it is not protected or if a user manages to lose it in some place, someone could be able to duplicate the information saved in the key and in that way use the information to login with the users account.

3.1.2 Tampering

Data flow tampering
Throughout the voting process, *data transmission* between various elements is vulnerable to tampering. Malicious actors could intercept and modify this data, compromising its integrity.

Database and process tampering
Data being stored is also vulnerable to tampering. If a malicious user gets access to a *data store* like the *application* or *company database*, they can alter information like votes, passwords, and other information. Getting access to a database can be rather difficult for an attacker. An easier target for a malicious user is the website. The architecture of the *web server* can allow attackers to modify information using various programs [1]. Information stored in hidden fields or URLs can be exploited and manipulated by an attacker to make the website preform desirable actions. The *vote counting process* also needs to be correctly designed to correctly handle all possible input. It needs to dismiss any invalid input to hinder exploitation.

3.1.3 Repudiation

User Repudiation
Because the e-voting system is designed to not be able to derive a vote to a user, repudiation attacks can be difficult to detect. If an attacker succeeds in making a vote in another *user's* name and it is not detected, the error will likely go unnoticed [2].

Server Repudiation
Because the users vote is designed to be untraceable the *web server* and *application server* need to be designed in a way that hinders attackers to perform repudiation attacks. This problem has a lot to do with *tampering* as protection against both entails proper *web server* design that hinder any attacker from manipulating any information.

Database Repudiation
If databases lack logging of admin and user actions a person with malicious intentions that has access to the *database* could make changes and *tamper* or *disclose* information without being traced. This could happen to the *application* and *company database* if they lack logging of user or admin actions.

3.1.4 Information Disclosure

Data flow disclosure
Because of the nature of election, information disclosure is a very serious threat. If *data flow* does not implement adequate encryption, there is a risk that an attacker might see the information. If the attacker obtains the encrypted information and cracks the encryption, confidential information about the voter can be disclosed. This is especially bad if the information contains company passwords which can lead to *spoofing*.

Database and server disclosure
If the *web server* fails to implement protections like TLS for its pages, an attacker might exploit this to steal information from other visitor's sessions [3]. The *vote counting process* needs to be designed to handle the votes in a proper and secure manner avoiding any outside person affecting the process.

The *application database* saves the votes and the public key of the user's authentication resulting in low desire for attacks to disclose information from the adequately protected database. The *Company database* on the other hand will be a desirable goal for an attacker because the information from the database could be used to perform spoofing attacks.

3.1.5 Denial of Service

Server and database attacks
Denial of service attacks can be used by an attacker to make the *website server* or *application server* unavailable. These attacks can exploit vulnerabilities in the architecture of the system including the *processes* and *databases.* The systems need to be designed with controls for traffic overflow. If this is not implemented the systems can crash even without the presence of an attack [4]. The website also needs controls which prevent users from doing actions which will occupy excessive amounts of memory.

User account attacks
A kind of DoS attack which targets the availability of the user accounts is problematic. To hinder brute force attacks on the company password during the registration process, controls need to be implemented which impedes an attack to repeatedly try different password. This control must be designed in a way which does not allow attackers to lock the user accounts. For example, can the control not lock the account for 10 minutes if the incorrect password is used 3 times. This control could lead to attackers trying password repeatedly just to make all accounts unavailable.

3.1.5 Elevation of Privilege

To prevent elevation of privilege attacks the *web server* needs a design which hinders such actions. More importantly, the *company* and *application system* need adequate protection against elevation of privilege attacks to hinder administrators or other users to gain access to information or functions they should not have access to. If the systems lack controls an administrator or user could potentially find flaws in the design of the system which lead to them being able to give themselves elevated access rights [5]. If the administrator or user has malicious intents the consequences of an attack could be catastrophic. If the *application process* and *web server* lack proper design a user could exploit this to gain elevated access or privileges which could lead to tampering, information disclosure and other attacks.

4. Risks and Controls

This section describes the top risks associated with the created DFD and mapping of STRIDE in the previous section. It also suggests controls that should be implemented to mitigate and eliminate the risks.

1. Asset: *User,* ***Risk:*** *Hacker registers with user company credentials on the voting system.*

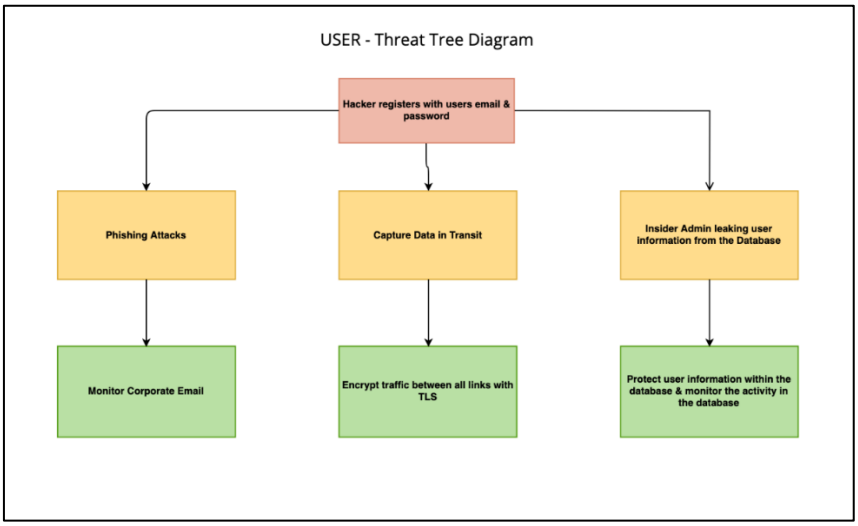

Figure 5 *User: Threat Tree Diagram*

Vulnerabilities
If a hacker manages to register with an employee's corporate credentials, this might go unnoticed if the employee never takes time to register, which opens the door to bigger threats to the system. This could happen in different ways:

- ***Phishing emails***: The hacker could be sending phishing emails to the company employees exploiting their company credentials.
- ***Capture and modify data in transit****:* The hacker could also capture and modify the data in transit during the registration process allowing them to redirect the registration to themselves with their own FIDO-token.
- ***Internal information disclosure****:* Another way is if an IT administrator that has access to the databases and user credentials discloses the user information to a malicious person.

Recommended Controls
To eliminate the threat of a hacker registering with an employee's credential, the suggested controls are as follows:

- ***Monitor corporate email****:* The company should monitor the corporate email server and using system such as email quarantine to block any suspicious emails from entering the mailbox of any employee, as well as educating users to reduce the impact of any phishing email that manages to slip through the monitoring.
- ***Encrypt all traffic between all elements****:* The company should use a secure encryption system such as TLS to mitigate the threat of a hacker capturing any data sent within the system.

- ***Protect & monitor database activity****:* User information should be handled according to GDPR, and any access to the database with sensitive user information should be monitored and logged to assure non-repudiation.

---

2. Asset: Application Database, **Risk**: Vote tampering

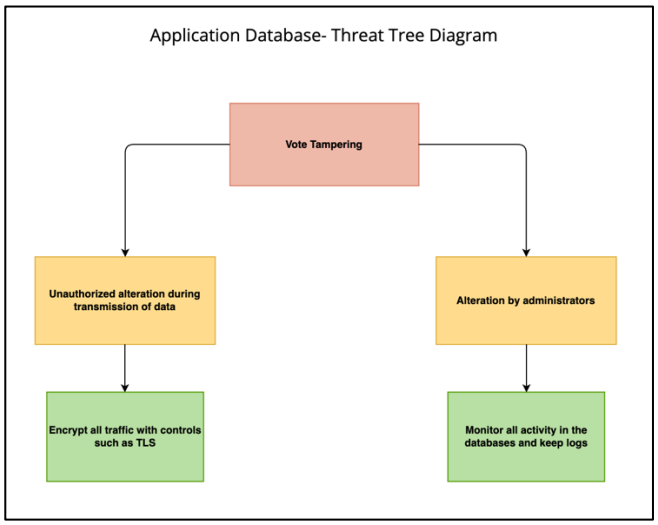

Figure 6 *Application Database: Threat Tree Diagram*

Vulnerabilities

- ***Alteration of vote during transmission of data****:* When a user has sent the vote to the system an attacker could potentially capture and alter the data during transmission.
- ***Alteration of votes by administrators****:* Authorized personnel that have access to the database such as IT administrators might have the ability to alter the data within the database.

Recommended Controls

To mitigate the threat of any vote data being altered we recommend the following controls:

- ***Encrypt all traffic that flows within the system***: Use encryption standards such as TLS to eliminate the threat of an attacker being able to capture and alter the data.
- ***Protect & monitor database activity****:* Any access to the database with sensitive user information should be monitored and logged to assure non-repudiation.

---

3. Asset: *Databases & Web Server,* ***Risk:*** *Denial of Service*

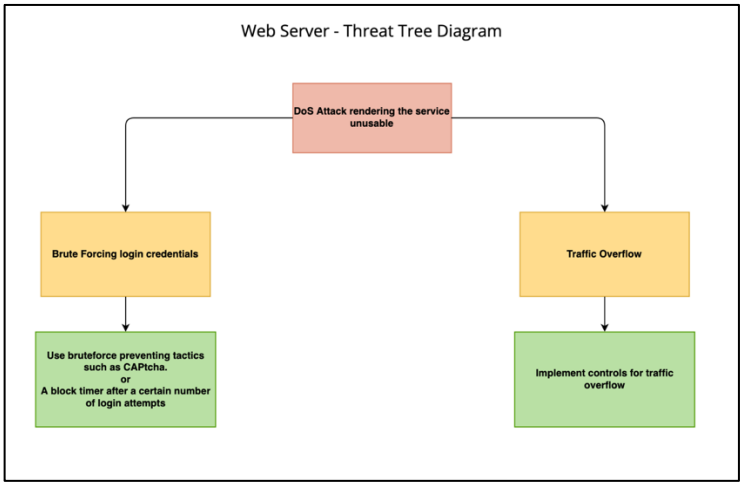

Figure 7 *Web Server/Database: Threat Tree Diagram*

Vulnerabilities
An attacker could perform a DoS attack making the application unusable. This could be critical specially if the attack is done during the end of the voting period as we consider users to postpone their voting until then.

- ***Brute forcing credentials during registration****:* An attacker could on the initial registration page be brute-forcing employee emails and passwords at a rate that the system cannot handle.
- ***Traffic overflow****:* An attacker could shut down the service by performing a DDoS attack flooding the system with traffic that it cannot handle.

Recommended Controls
- ***Use brute force prevention techniques*** *such as CAPTCHA and a registration/login block after a certain number of login/registration attempt.*
- ***Use controls such as Packet filtering technology*** *and other controls to reduce the risk of a DDoS attack on the application, session, or network.*

---

4. Asset*: Vote Counting Process,* ***Risk****: Inaccurate tally of votes*

Vulnerabilities
- ***Software Bugs****:* Errors in the vote counting software could lead to incorrect vote tallies.
- ***Malicious Manipulation****:* Deliberate manipulation of the vote counting process by insiders or hackers.

Recommended Controls
- ***Testing****:* Conduct extensive testing of the vote counting software to ensure accuracy.
- ***Access Control and Monitoring****:* Restrict and monitor access to the vote counting process to prevent unauthorized manipulation.

---

5. Asset: *Company DBMS,* ***Risk:*** *Disclosure of sensitive company data*

Vulnerabilities
- ***Lack of monitoring*** **and logs** of database activities could allow malicious action to go unnoticed leading to information disclosure

Recommended Controls

- ***Audit Trails and Monitoring:*** Establish comprehensive logging and monitoring for all database activities.

---

6. Asset: *FIDO-Token,* ***Risk:*** *Token duplication*

Vulnerabilities

- ***Token Cloning****:* FIDO-tokens could potentially be duplicated if they are not properly secured or lost by a user.

Recommended Controls

- ***Enhance Authentication Process****:* The login process could be even more secure by adding a 2FA requiring a user to authenticate with both a FIDO-token and PIN-Code or OTP (One-Time Passcode).
- ***Regular Security Audits****:* Conduct regular reviews of the FIDO system to detect any vulnerabilities or anomalies.

4.1 Summary of Controls

The recommended controls will overall improve the security of the voting system and application. Implementing these controls will protect the company and the service from the mentioned threats and many more that have not been mentioned. Below you will find a summary of all recommended controls mentioned.

- **Monitor corporate email**: The company should monitor the corporate email server and using system such as email quarantine to block any suspicious emails from entering the mailbox of any employee, as well as educating users to reduce the impact of any phishing email that manages to slip through the monitoring.
- **Encrypt all traffic that flows within the system**: Use encryption standards such as TLS to eliminate the threat of an attacker being able to capture and alter the data.
- **Protect & monitor database activity**: Any access to the database with sensitive user information should be monitored and logged to assure non-repudiation.
- **Use brute force prevention techniques** such as CAPTCHA and a registration/login block after a certain number of login/registration attempt.
- **Use controls such as Packet filtering technology** and other controls to reduce the risk of a DDoS attack on the application, session, or network.
- **Testing**: Conduct extensive testing of the vote counting software to ensure accuracy.
- **Access Control and Monitoring**: Restrict and monitor access to the vote counting process to prevent unauthorized manipulation.
- **Enhance Authentication Process**: The login process could be even more secure by adding a 2FA requiring a user to authenticate with both a FIDO-token and PIN-Code or OTP (One-Time Passcode).
- **Regular Security Audits**: Conduct regular reviews of the FIDO system to detect any vulnerabilities or anomalies.

Reference

[1] OWASP Foundation, "Web Parameter Tampering," OWASP, 2023. [Online]. Available: https://owasp.org/www-community/attacks/Web_Parameter_Tampering. [Accessed: Dec. 15, 2023].

[2] OWASP Foundation, "Repudiation Attack," OWASP, 2023. [Online]. Available: https://owasp.org/www-community/attacks/Repudiation_Attack. [Accessed: Dec. 15, 2023].

[3] OWASP Foundation, "A3:2017-Sensitive Data Exposure," in OWASP Top Ten 2017, OWASP, 2023. [Online]. Available: https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure. [Accessed: Dec. 15, 2023].

[4] OWASP Foundation, "Denial of Service," OWASP, 2023. [Online]. Available: https://owasp.org/www-community/attacks/Denial_of_Service. [Accessed: Dec. 16, 2023].

[5] OWASP Foundation, "Testing for Privilege Escalation," in Web Security Testing Guide, OWASP, 2023. [Online]. Available: https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/05-Authorization_Testing/03-Testing_for_Privilege_Escalation. [Accessed: Dec. 16, 2023].