I – Decomposing Application

Threat Model Information

Application Name	XML+BSEP
Application Version	1.0
Description	Web application made to represent Information system of the City Assembly of Novi Sad. Application provides issuing of digital certificates, voting procedure for passing acts and amendments and saving passed acts and amendments. Application users are: - President of the City Assembly of Novi Sad - Aldermen - Citizens. President of the CA issues self-signed certificates as the highest certification body, organizes voting sessions and has main word for passing acts and amendments. Aldermen can log on system and can receive certificates from CA, propose, make changes to acts and amendments. Citizens can view and search for passed acts and amendments.
Document Owner	Jovana Ikonic
Participants	Mirjana Curcin, Kristina Satara
Reviewer	Zoran Luledzija

External Dependencies

ID	**Description**
1	Application: Desktop application for issuing certificates, Web application for working with acts and amendments.
2	Database: MarkLogic NoSQL Database
3	Frontend: RESTfull Service + Angular JS
4	Middle layer: DAO Bean
5	Server: Apache TomEE Server
6	Browser-Server Communication: HTTPS Port

Entry Points

ID	Name	Description	Trust Levels
1	*HTTPS Port*		
1.1	Login Page	All users must log in before proceeding with their work/working with certificates.	(1) Anonymous Web User (2) User with Valid Login Credentials (3) User with Invalid Login Credentials (4) Aldermen (5) President of the City Assembly
1.1.1	Login Function	Compares provided credentials with database data.	(2) User with Valid Login C (3) User with Invalid Login Cr. (4) Aldermen (5) President of the City Assembly
1.2	Search Acts Page	Citizen searches for adopted (passed) acts, acts in procedure.	(2) User with Valid Login Credentials (4) Aldermen (5) President of the City Assembly
1.2.1	Search Acts Function	Processes provided data in search fields.	(2) User with Valid Login Credentials (4) Aldermen (5) President of the City Assembly
1.3	Proposing Acts Page	Alderman proposes acts.	(4) Aldermen
1.3.1	Proposing Acts Function	Submits desired act in electronic form.	(4) Aldermen
1.4	Proposing Amendments Page	Alderman proposes amendments on the count of proposed act.	(4) Aldermen
1.4.1	Proposing Amendments Function	Submits desired amendment.	(4) Aldermen
1.5	Export of Acts and Amendments Page	Export acts and amendments in desired format (XHTML, PDF) for	(2) User with Valid Login Credentials (4) Aldermen (5) President of the

		various legal purposes.	City Assembly
1.5.1	Export of Acts and Amendments Function	Converting already existing AAs from database into selected format.	(2) User with Valid Login Credentials (4) Aldermen (5) President of the City Assembly
1.6	Sending accepted (passed) acts to Historical Archive of City of Novi Sad Page	Acts that are valid go to HACNS for permanent archiving.	(4) Aldermen (5) President of the City Assembly
1.6.1	Sending acts to Historical Archive of City of Novi Sad Function	Sends passed acts through web to HACNS.	(4) Aldermen (5) President of the City Assembly
1.7	Opening/Closing Voting Session Page	While open aldermen can propose, edit, retrieve and deny acts and amendments.	(5) President of the City Assembly
1.7.1	Opening/Closing Voting Session Function	Notifying aldermen involved in session about period of time they can vote/down vote acts and amendments.	(5) President of the City Assembly
1.8	Voting Page	Aldermen can vote during active voting session.	(4) Aldermen (5) President of the City Assembly
1.8.1	Voting Function	If voting session is open, aldermen can vote. Votes will be processed and stored.	(4) Aldermen (5) President of the City Assembly

Assets

ID	**Name**	**Description**	**Trust Levels**
0	**Proposals, Acts and Amendments**	Proposals, acts and amendments that circle through app.	(4), (5), (6)
1	**App users**		
1.1	Citizen Login Details	Credentials of a citizen necessary for working with	(2) User with Valid Login C (7) Database Server

		application.	Admin (8) Database Read User (9) Database R/W User
1.2	Alderman Login Details	Credentials of an alderman.	(4) Aldermen (7) Database Server Admin (8) Database Read User (9) Database R/W User
1.3	President of the City Assembly Login Details	Credentials of a president of the City Assembly.	(5) President of the CA (7) Database Server Admin (8) Database Read User (9) Database R/W User
2	**System**		
2.1	Availability of the Website	Website should be available 24 hours a day, should grant access to registered users and be available to visitors.	(6) Website Administrator (7) Database Server Admin
2.2	Ability to Execute SQL as a Database Read User	Executing queries and retrieving information from database.	(7) Database Server Admin (8) Database Read User (9) Database R/W User
2.3	Ability to Execute SQL as a Database Read/Write User	Having read/write access to database data.	(7) Database Server Admin (9) Database R/W User
3	**Website**		
3.1	Login Session	Login session for all users' roles.	(2) User with Valid Login Credentials (4) Aldermen (5) President of the City Assembly

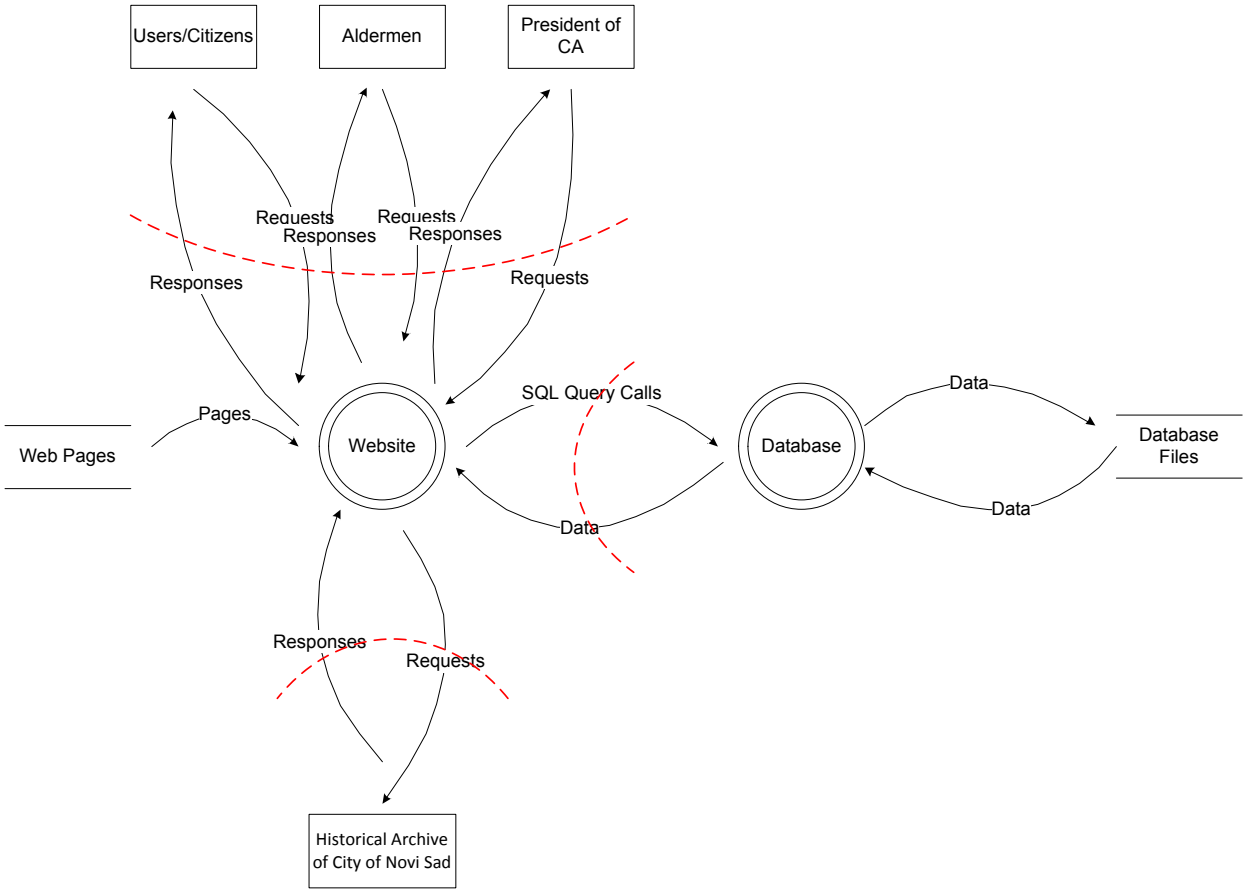
3.2	Access to the Database Server	Administering database and stored data.	(7) Database Server Admin
3.3	Ability to Create Users	Adding/Registering new users.	(6) Website Administrator
3.4	Ability to Add Acts	Adding new acts.	(4) Aldermen (5) President of the City Assembly
3.5	Ability to Add Amendments	Adding new amendments.	(4) Aldermen (5) President of the City Assembly
3.6	Ability to pass on acts to HCA	Sending acts through net.	(4) Aldermen (5) President of the City Assembly
3.7	Ability to open/close voting session	Determining specific period of time intended for voting.	(5) President of the City Assembly
3.8	Ability to vote for act/amendment	Aldermen vote for act/amendment currently proposed.	(4) Aldermen (5) President of the City Assembly

Trust Levels

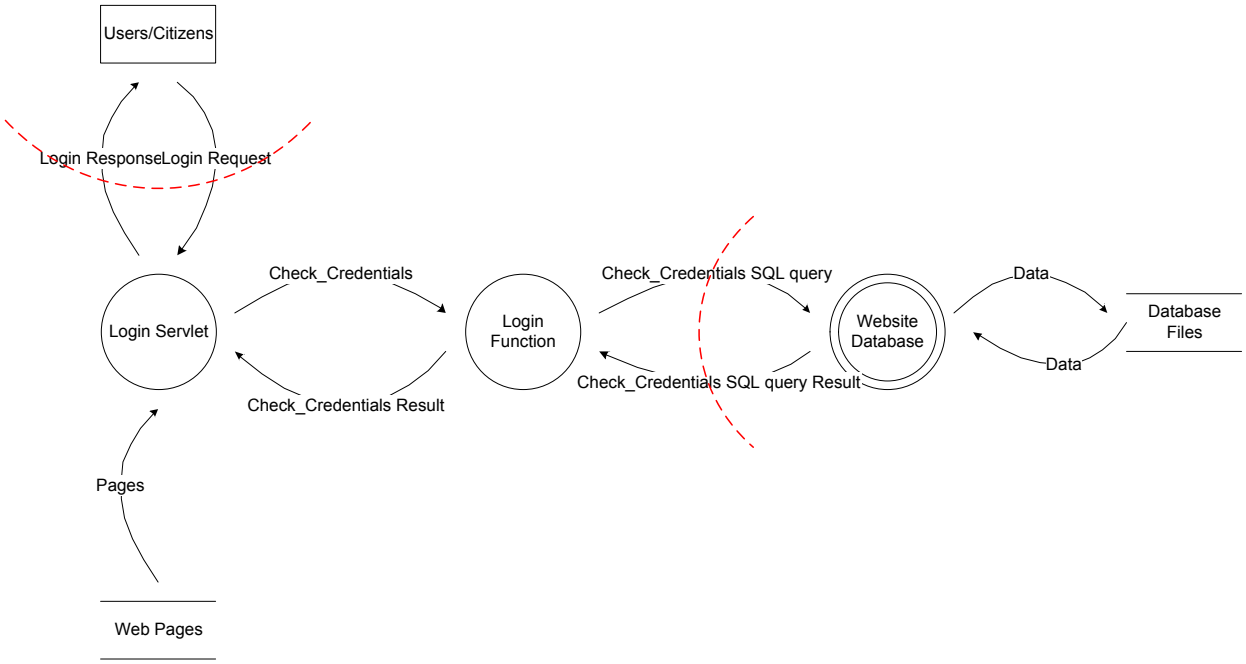
ID	**Name**	**Description**
1	Anonymous Web User	Visitors on the website.
2	User with Valid Login Credentials	User that is registered in system.
3	User with Invalid Login Credentials	User with expired, wrong or random credentials not found in database.
4	Aldermen	Can propose, retract, revise acts.
5	President of the City Assembly	Can approve acts and/or amendments, and manages City Assembly sessions. (can upgrade citizen to alderman)
6	Website Administrator	In charge of website availability, users work on the website and content.
7	Database Server Administrator	In charge of database and its data.
8	Database Read User	Database user account for read operations.
9	Database Read/Write User	Database user account for read/write operations.

Data Flow Diagrams

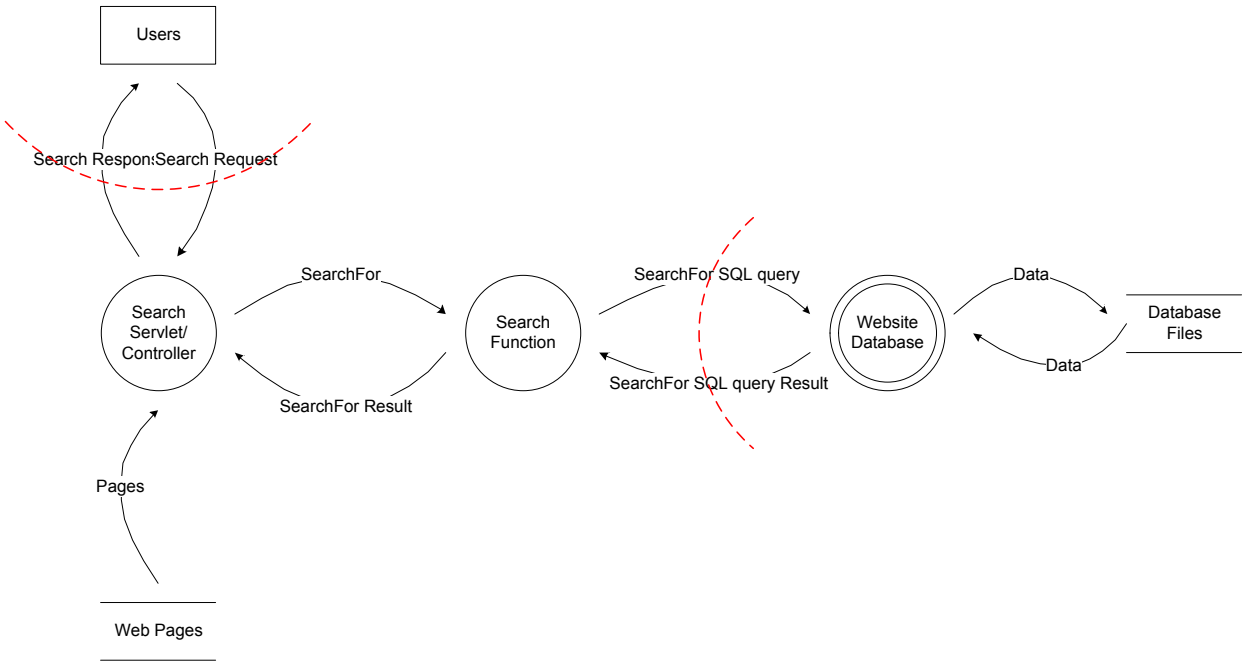
Website Data Flow Diagram



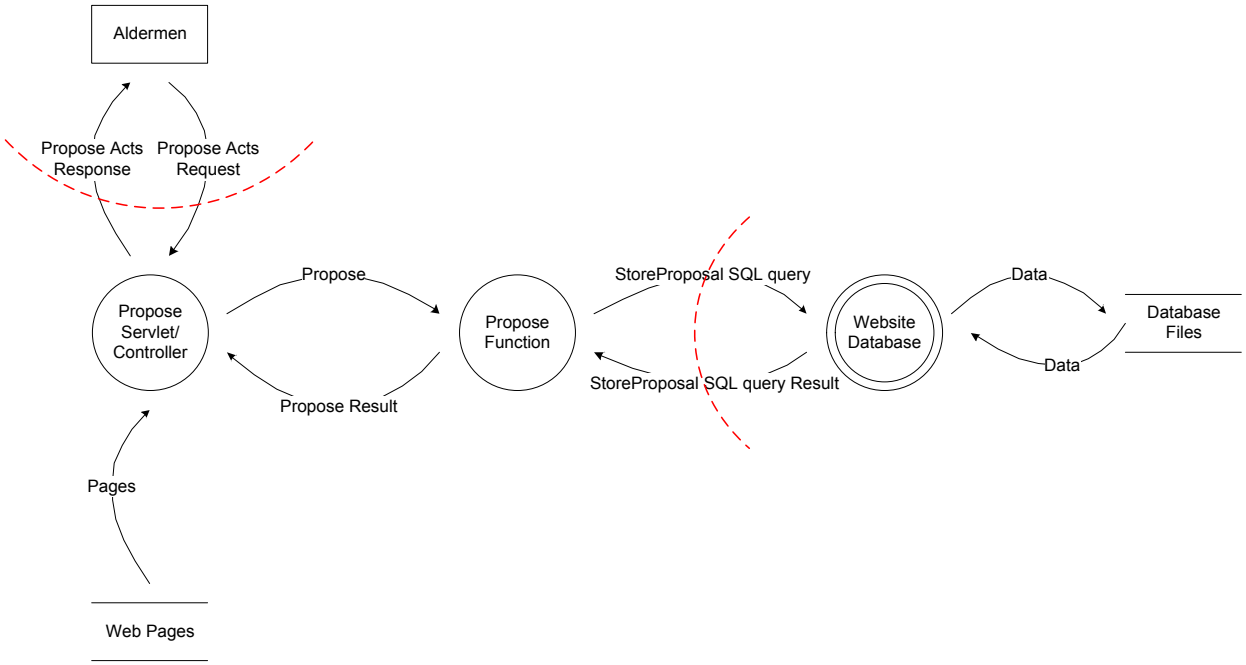
User/Citizen/Aldermen/President Login Data Flow Diagram



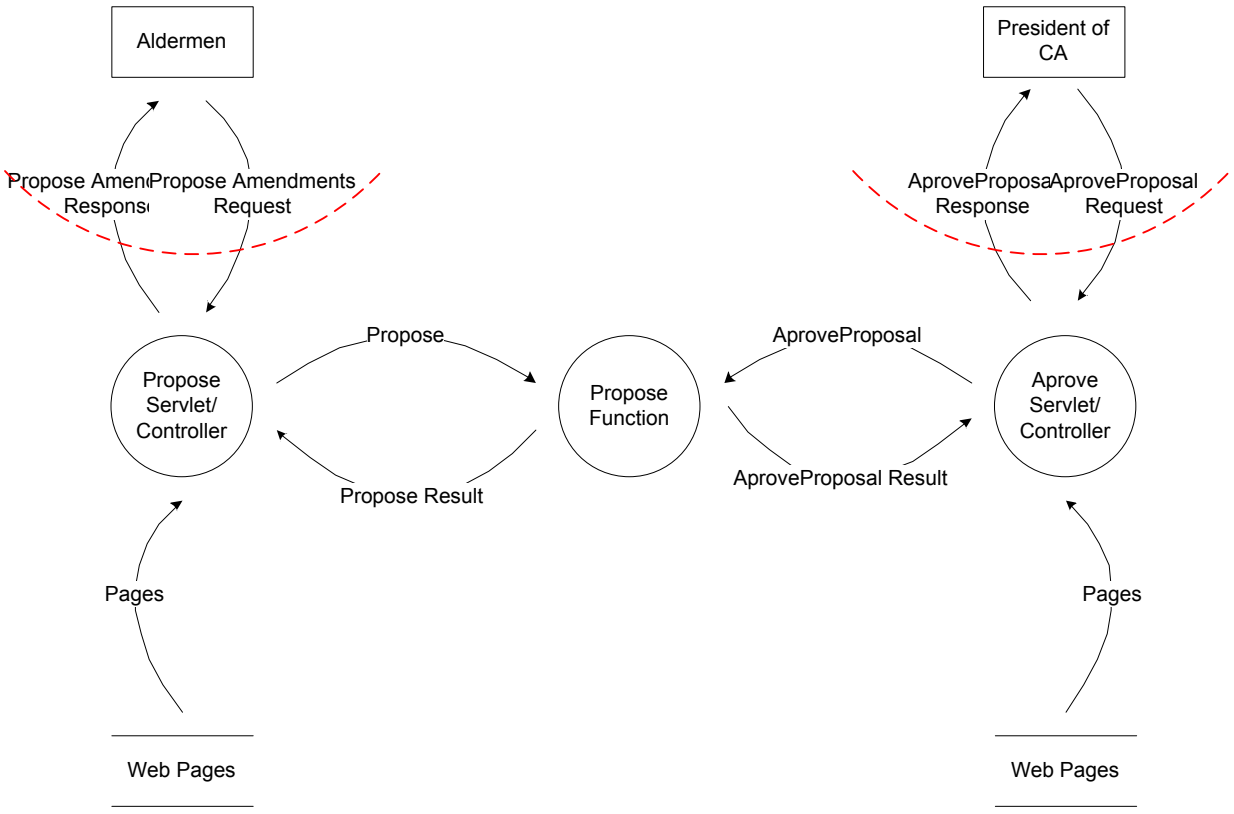
User/Citizen/Aldermen/President Search Data Flow Diagram



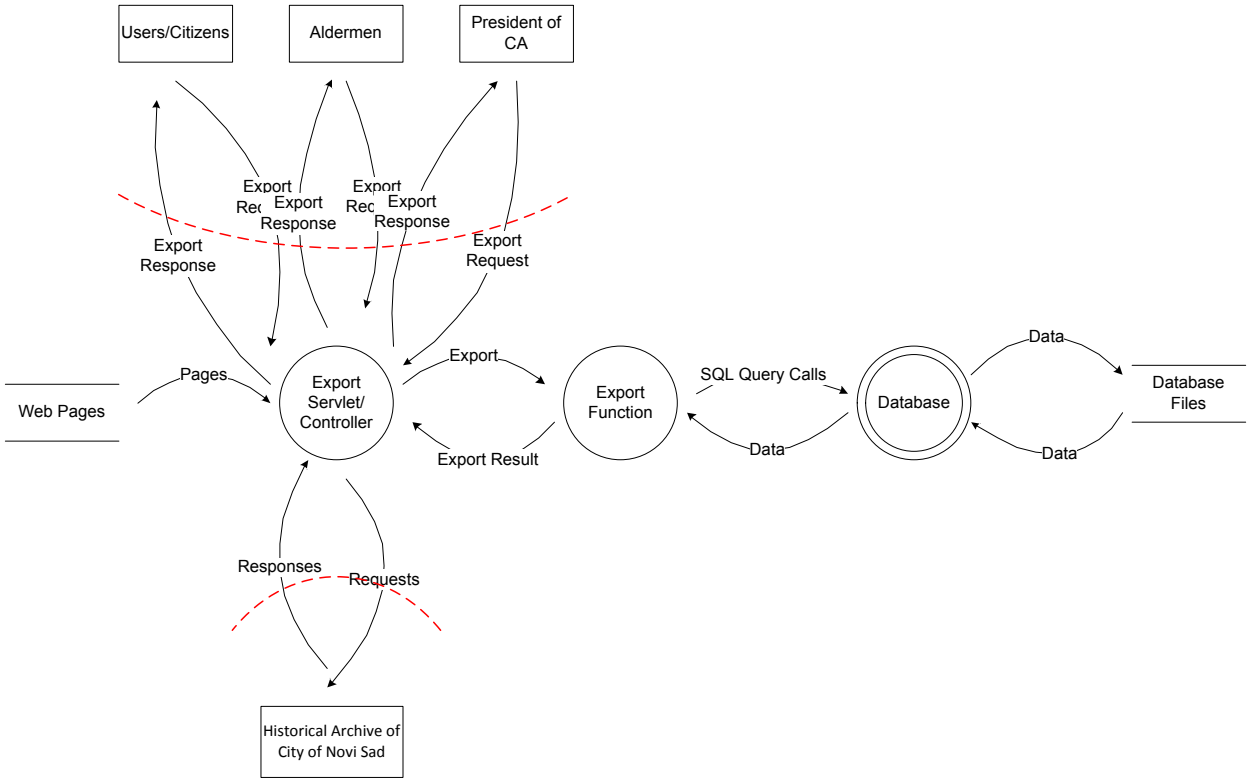
Aldermen Proposing Acts Data Flow Diagram



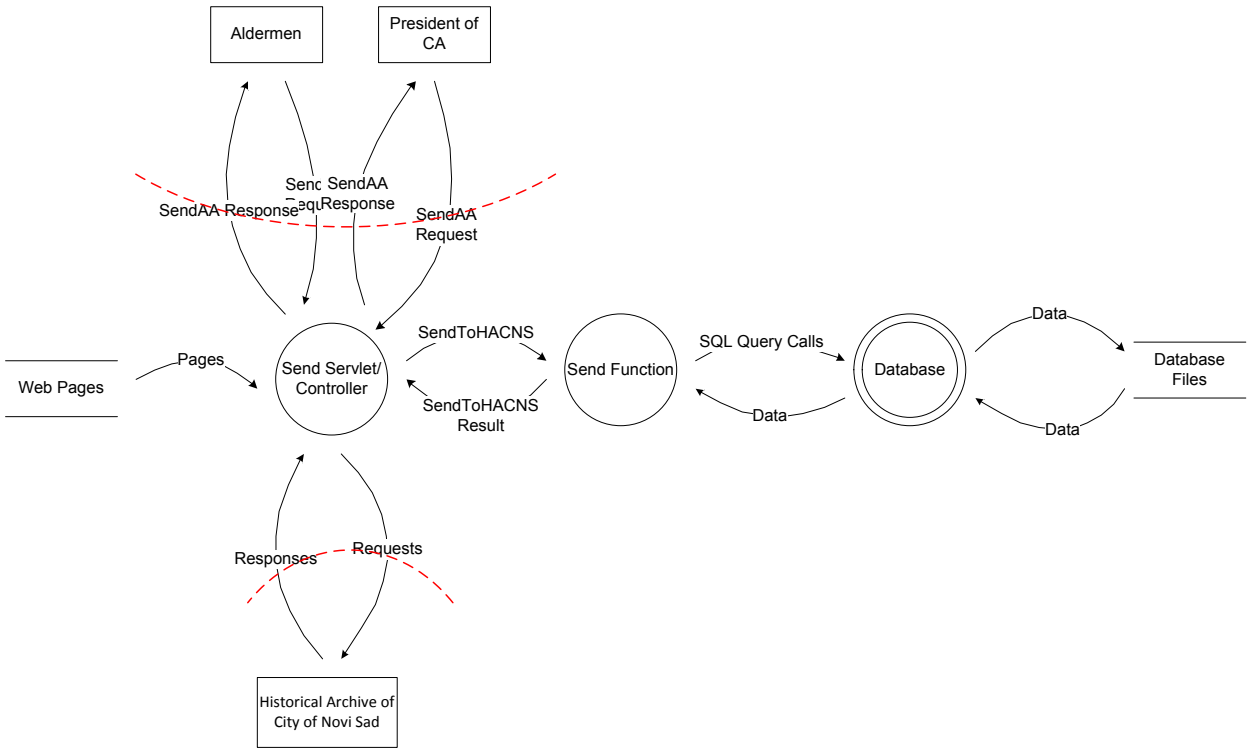
Aldermen Proposing Amendments Data Flow Diagram



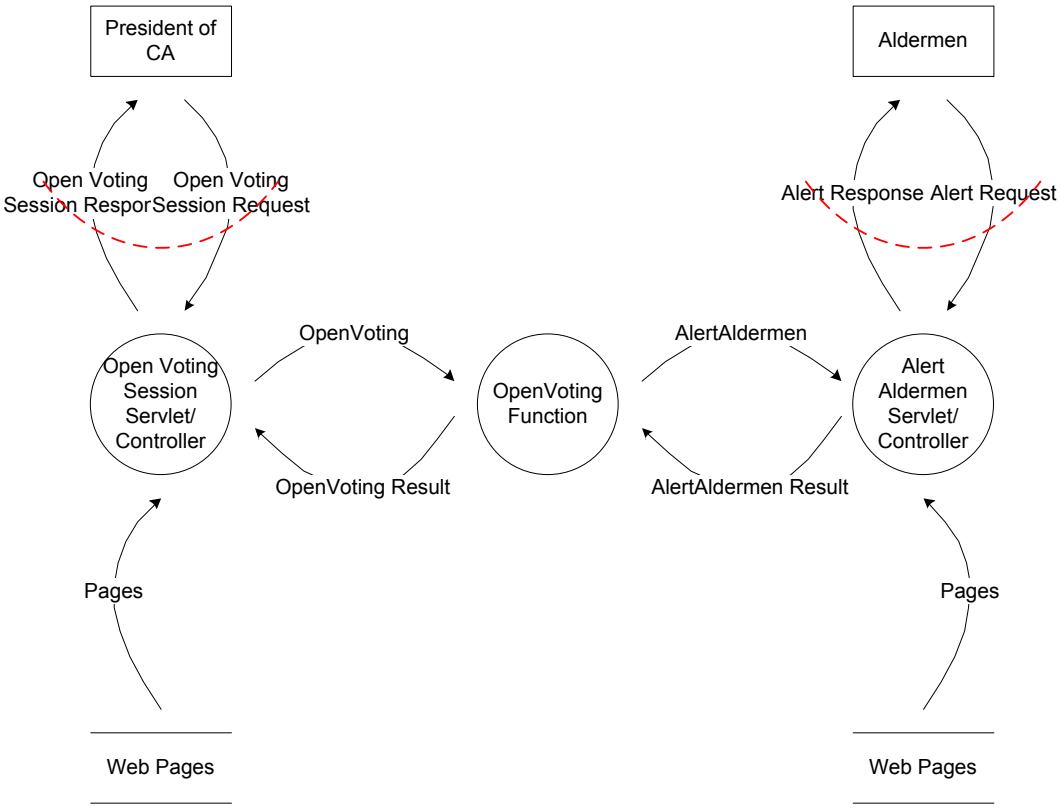
User/Citizen/Alderman/President Export AA Data Flow Diagram



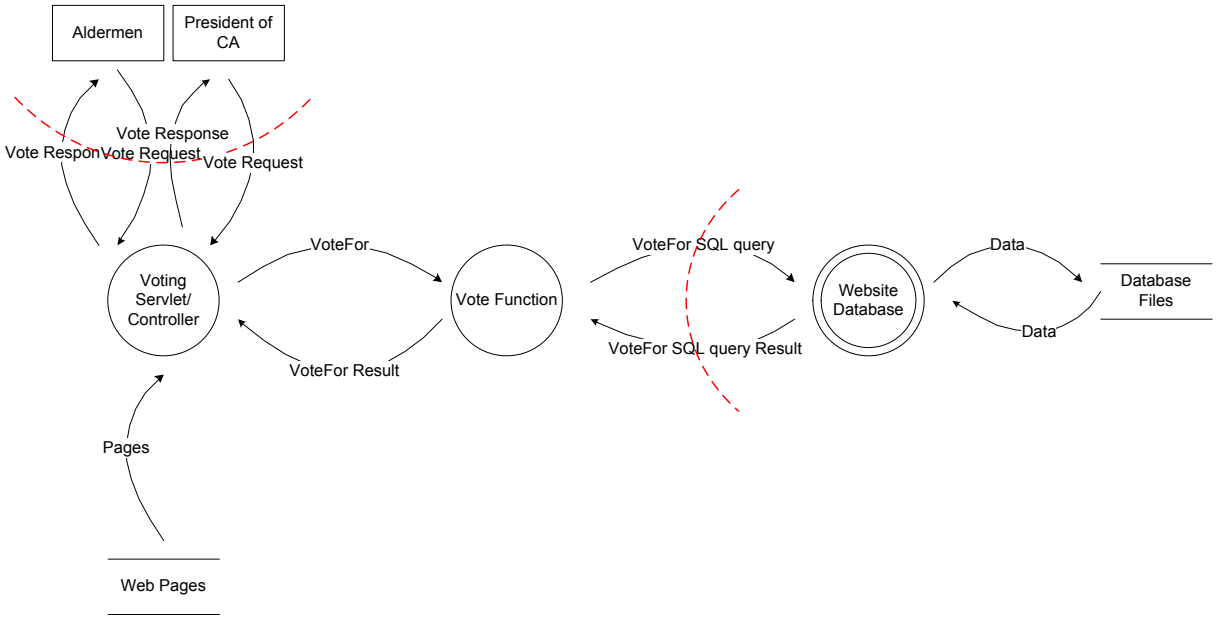
Alderman/President Send AA to HACNS



President Open Voting Data Flow Diagram



Aldermen/President Voting Data Flow Diagram



II – Determine and Rank Threats

Threat Categorization

Threat categorization that will be adopted for this application is STRIDE.

Type	Security Control	Description
Spoofing	Authentication	Threat action aimed to illegally access and use another user's credentials, username and password.
Tampering	Integrity	Threat action aimed to maliciously change/modify persistent data, such as persistent data in a database, and the alteration of data in transit between two computers over an open network, such as the Internet.
Repudiation	Non-repudiation	Threat action aimed to perform illegal operations in a system that lacks the ability to trace the prohibited operations.
Information disclosure	Confidentiality	Threat action to read a file that one was not granted access to, or to read data in transit.
Denial of service	Availability	Threat aimed to deny access to valid users, such as by making a web server temporarily unavailable or unusable.
Elevation of privilege	Authorization	Threat aimed to gain privileged access to resources for gaining unauthorized access to information or to compromise a system.

Spoofing. Is a type of attack where attacker gains access to information and/or system by tricking user into revealing sensitive data. Usually sensitive data are users credentials for their bank accounts, company account etc. Spoofing can come in various shapes: email spoofing or phishing, caller ID, URL spoofing, IP spoofing, etc.

Phishing is form of attack where users get tricked by email that resembles a lot like an email they used to get from their bank, friends, associates, coworkers, family etc. Those emails usually ask for passwords, account numbers, personal information, credit card numbers etc. Used mechanism: forged header, spams, commercials, offers, etc. Caller ID spoofing involves attack where users receive calls from numbers that are same as numbers they usually get call from. Those calls usually have someone representing themselves as worker in user associated place and they have some storyline about resetting accounts, emergencies, confirmation of an account etc. Used mechanism: manipulating caller ID recognition, forged caller ID. URL spoofing is an attack that steals users sensitive data by imitating websites user often uses. User gets redirected to attackers copy website and is prompted with login form and/or confirmation dialog that installs viruses in user's machine. Used mechanism: forged websites.

Tampering. Is a type of attack where attackers modify data from database. They tamper with data which is usually sent in forms, from client to server and vice versa. This is achieved by manipulating parameters from GET and POST methods in HTTP communication protocols. Attackers bypass web interface and send altered data to server side application. Those changes can have malicious effect on backend of the application. Attackers use tampering tools such as malwares that they trick user into installing.

Repudiation. Is a type of attack that is not traced properly and therefore malicious users/attackers can access and modify data. Attackers can change log files so it would seem that other user did the modifications. They can plant other users the malicious deeds they did.

Information disclosure. Is a type of attack where attackers get view of sensitive data about users, web application developers, and website. Attackers also can gain information about location of backup and temporary files. Types of information disclosure attack are: directory indexing, information leakage, path traversal, predictable resource location.

Directory indexing is an attack that exploits web server function that enlists all files within a requested directory if the requested file is not available. That way attacker gets information about hidden files, backup files, configuration files etc. Information leakage is an attack that reveals websites sensitive data like developer comments (where can sometimes be stored login credentials) and error messages. Path traversal is an attack that can reveal access to files, directories and commands that are located outside the web document root directory. Attackers exploit URL by using special characters (../, ..\, %2e%2e%2f, etc.). Predictable resource location is an attack that uncovers hidden website content and functions.

Denial of service. Is a type of attack where attackers have only one purpose and that is to make your website unavailable, for as long as they can, to regular users. Attack is performed usually by sending too many useless requests to your server. Attackers usually send messages that ask

network or server to authenticate request that have invalid return addresses. Server or network will not be able to find return address and therefore it will cause server or network to wait before closing connection. When one connection is closed attackers send more of those messages that keep server or network busy waiting and keeping it out of service for valid messages. DoS attack usually manifests as clustering network traffic, as disrupting connection between two machines, as preventing targeted victim from accessing service, etc. Forms of Dos attacks are: Buffer overflow attack, Teardrop attack, Smurf attack, viruses.

Buffer overflow is an attack where attackers send more traffic that it is anticipated that will come to network. This attack can be based on a weakness of the system, that attackers somehow got to knowing or it can be tried blindly. Teardrop attack is an attack where attacker's IP address puts offset value, in second or later fragment, that is strange to package causing system to crash when trying to reassemble large package sent over network. Smurf attack is an attack where attacker sends an IP ping request to receiving site. Package indicates that the request is from victim's site and specifies its broadcasting to a number of hosts inside the receiving site's local network. Meaning, attackers are spoofing the return address which results in huge amount of ping replies directed to victim's website which leads to non-distinguishing real traffic. Viruses can be transmitted through network in a DoS attacks. Then you have casualty victims that are usually not targeted, but simply a host for a while. That time is enough for virus to spread.

Elevation of privilege. Is a type of attack where attackers aim to gain access to information, data they don't have granted access. This is achieved by tricking system into thinking that they have rights to access certain data, information or part of the application. This type of the attack is usually an inside job, because it demands high set of skills of an attacker to access victim's network directly through dial-up or by hacking VPN. It can be done by using bug or flaw in system or by attacking and altering sensitive data that shows what privilege which kind of user has.

Threat Analysis and Identification

SPOOFING in our application

Possible attacks:

1. Attacking Login page in order to steal credentials and get into application

    1.1 Tricking victims by email, call, URL, false representation online

    1.2 Victims use shared computer and they store they're credentials in a browser

    1.3 Victims forget to log off from shared computer

1.4 Unauthenticated access: SQL injection in form fields, login form field manipulation

1.5 Password cracking/stealing

1.6 Brute force attack on password field

Exploitation: attackers can get different access to the system based on which user's credentials they have. They can get access to sensitive application data (laws, acts and amendments). If they get presidents login credentials, they can get rights to change results of voting, daily schedule of a voting session, already adopted laws, acts and amendments passed to consideration. If they get alderman login credentials they can get access to acts and amendments and they can change proposition, concepts, alderman voting, exporting format of acts, amendments. If they get citizen login credentials they get access to searching and viewing adopted laws. If they get web application administrators login credentials then they have full access to application.

Solution:

1.1 Educate users about types of spoofing attacks
1.2 Educate users to not save their credentials in browsers in general, especially in browsers that are used by multiple users
1.3 Educate users to log off always, especially if they use shared computers
1.4 Validate every input field in login form by length, format, type; Make strong checking credentials functions
1.5 Strong password policies; Safe passwords storing
1.6 Encrypt passwords using salt and hash technique.

<u>TAMPERING</u> in our application

Possible attacks:

1. Attacking the Proposing Acts/Amendments page in order to tamper with acts/amendments

1.1 Planting malicious program to client using email and URL spoofing

1.1.1 Malware modifies keyboard input while inputting acts/amendments

1.1.2 Malware tampers with data (acts/amendments) client sends to server

1.2 Intercepting server communication with tampering tools

1.2.1 Malware tampers with messages sent from server to client

1.3 SQL Injection (insert/update/delete attack on database data)

2. Attacking the Exporting Acts/Amendments page in order to tamper with acts/amendments

2.1 Planting malicious program to client using email and URL spoofing

2.1.1 Malware modifies keyboard input while configuring export options for acts/amendments

2.1.2 Malware tampers with data that is exporting

2.2 Intercepting server communication with tampering tools

2.2.1 Malware tampers with messages sent from server to client

2.3 SQL Injection (insert/update/delete attack on database data)

3. Attacking the Sending Acts/Amendments to HACNS page in order to tamper with acts/amendments

3.1 Planting malicious program to client using email and URL spoofing

3.1.1 Malware modifies keyboard input while inputting sender information acts/amendments

3.1.2 Malware tampers with data (acts/amendments) client sends to server

3.2 Intercepting server communication with tampering tools

3.2.1 Malware tampers with messages sent from server to client

3.3 SQL injection and Code injection

Exploitation: data that attackers can tamper with are acts and amendments, which mean they can tamper with future laws.

Solution:

1.1, 2.1, 3.1 Firewalls, malware detection programs from client side

1.2.1-3.2.1 Using key cryptography to prevent tampering with messages sent between server and client and vice versa

1.1.2-3.1.2 Strong access control for ensuring that only authorized users can access and/or modify data; Role-based security for determination of users allowed functionalities

1.1.1-3.1.1, 1.3-3.3 Integrity checks (checksum, HMAC, encryption, digital signature), data validation, business rule validation; Validation on backend side as well as in the web interface

REPUDIATION in our application

Possible attacks:

1. Attacking Proposing Acts/Amendments page

1.1 Changing proposed acts/amendments

1.2 Changing log records

1.2.1 Deleting log records

1.2.2 Messing up log records by deleting some records, combining others, etc.

2. Attacking Exporting Acts/Amendments page

2.1 Changing/Recalling export options

2.2 Changing/Recalling which act/s and/or amendment/s is exporting

2.3 Changing log records

2.3.1 Deleting log records

2.3.2 Messing up log records by deleting some records, combining others, etc.

3. Attacking Sending Acts/Amendments to HACNS page

3.1 Changing/Recalling adopted law that is being sent to HACNS

3.2 Changing log records

3.2.1 Deleting log records

3.2.2 Messing up log records by deleting some records, combining others, etc.

4. Attacking Opening/Closing Voting Session page

4.1 Changing duration of voting session

4.2 Changing daily schedule for voting sessions

4.3 Changing log records

4.3.1 Deleting log records

4.3.2 Messing up log records by deleting some records, combining others, etc.

5. Attacking Voting page

5.1 Changing voting results

5.2 Changing log records

5.2.1 Deleting log records

5.2.2 Messing up log records by deleting some records, combining others, etc.

Exploitation: log records of application which helps attacker cover up their trails and allows them anonymous data tampering.

Solution:

1., 2., 3., 4., 5. Log activities on web, database and application server; Log everything

1.1, 2.1, 2.2, 3.1, 4.1, 4.2, 5.1 Role-based security; Database transaction logs; Authentication

1.2, 2.3, 3.2, 4.3, 5.2 Better log actions capturing that includes identity, action, date, time without capturing sensitive data in logs

<u>INFORMATION DISCLOSURE</u> in our application

Possible attacks:

1. Not properly configured web server (allowing directory listing)

2. Not properly handling special characters (backslash, null bytes)

3. Not securing cashed or historical data contained in online databases

4. Malware for obtaining sensitive data through screenshots, keystroke logging

Exploitation: backup files, configuration files, directory listing, password files, database files

Solution:

1. Administrators will configure server more complex so that directory listing is available for specific directories and/or sub-directories

2. Properly handle most of the special characters

3. Authenticating on every page because of cached or historical data

4. Firewalls, malware detection programs from client side

DENIAL OF SERVICE in our application

Possible attacks:

1. Attacking Proposing Acts/Amendments page with too much requests

1.1 Buffer overflow attack (sending too many requests for proposing/documenting acts/amendments)

1.2 Teardrop attack (tampering with fragments of a too big act/amendment)

2. Attacking Exporting Acts/Amendments page with too much requests

2.1 Buffer overflow attack (sending too many requests for exporting same act/amendment and/or different acts/amendments)

2.2 Teardrop attack (tampering with fragments of act/amendment that's being exported)

3. Attacking Login page with too many requests

3.1 Buffer overflow attack

4. Attacking Search Acts page with too many requests

4.1 Buffer overflow attack (sending too many requests for searching same act/amendment and/or different acts/amendments)

5. Attacking Voting page with too many requests

5.1 Buffer overflow attack (sending too many voting requests for same and/or different act/amendment)

6. Smurf attack (ping requests)

Exploitation: application vulnerabilities, database vulnerabilities, sensitive data

Solution:

1.1-5.1 Input validation; Limit application's use of unmanaged code; Determination of memory boundaries; Memory bound checking

6. Configuring routers, limiting ICMP traffic

ELEVATION OF PRIVILEGE in our application

Possible attacks:

1. Buffer overflow attack on database and then exploiting that vulnerability to gain highest privileges in application

2. Malicious code injections through URL, email spoofing that install key loggers

Exploitation:

Solution:

1. Query level access control; Intrusion prevention systems

2. Firewalls; Malware detection tools; Detecting changes in OS Server core files (registries)

Ranking of threats

OWASP methodology:

Possible attackers/Threat agents				
Ranking	**Skill level**	**Motivation**	**Opportunity**	**Size**
0			Full access or exp. res. required	
1	No technical skills	Low or no reward		
2				Developers, system admin.

3	Some tech. skills			
4		Possible reward	Special access or res. required	Intranet users
5	Adv. comp. user			Partners
6	Network and programming skills			Authenticated users
7			Some access or res. required	
8				
9	Security and penetration skills	High reward	No access or res. required	Anonymous users

Security flaw				
Ranking	**Ease of discovery**	**Ease of exploit**	**Awareness**	**Intrusion detect.**
0				
1	Practically impossible	Theoretical	Unknown	Active detect. in application
2				
3	Difficult	Difficult		Logged and reviewed
4			Hidden	
5	Medium	Easy		
6			Obvious	
7	Easy			
8				Logged without review
9	Automated tools available	Automated tools available	Public knowledge	Not logged

Technical impact				
Ranking	**Loss of confidentiality**	**Loss of integrity**	**Loss of availability**	**Loss of accountability**
0				
1		Minimal slightly corrupt data	Minimal secondary services interrupted	Fully traceable
2	Minimal non-			

	sensitive data disclosed			
3		Minimal seriously corrupt data		
4				
5		Extensive slightly corrupt data	Minimal primary services interrupted, extensive secondary services interrupted	
6	Minimal critical data disclosed, extensive non-sensitive data disclosed			
7	Extensive critical data disclosed	Extensive seriously corrupt data	Extensive primary services interrupted	Possibly traceable
8				
9	All data disclosed	All data totally corrupt	All services completely lost	Completely anonymous

Business impact				
Ranking	**Financial damage**	**Reputation damage**	**Non-compliance**	**Privacy violation**
0				
1	Less than the cost to fix the vulnerability	Minimal damage		
2			Minor violation	
3	Minor effect on annual profit			One individual
4		Loss of major accounts		
5		Loss of goodwill	Clear violation	Hundreds of people
6				
7	Significant effect on annual profit		High profile violation	Thousands of people

8				
9	Bankruptcy	Brand damage		Millions of people

Likelihood and Impact Levels	
0 to <3	LOW
3 to <6	MEDIUM
6 to 9	HIGH

Overall Risk Severity = Likelihood x Impact				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	**Low**	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

Risk = Likelihood x Impact (technical or business, depending on goodness level)

Stealing user credentials

Risk: *Tricking victims by email, call, URL, false representation online* = **LOW**

Likelihood								
Possible attackers					**Security flaw**			
Skill level	**Motive**	**Opportunity**	**Size**		**Ease of discovery**	**Ease of exploit**	**Awareness**	**Intrusion detection**
1	1	7	9		5	5	4	9
Overall likelihood: 5.125, MEDIUM								

Impact								
Technical impact					**Business impact**			
Loss of confidentiality	**Loss of integrity**	**Loss of availability**	**Loss of accountability**		**Financial damage**	**Reputation damage**	**Non compliance**	**Privacy violation**
2	1	0	7		1	1	0	3
Overall impact: 1.875, LOW								

Risk: *Victims use shared computer and they store they're credentials in a browser* = **HIGH**

Likelihood								
Possible attackers					Security flaw			
Skill level	**Motive**	**Opportunity**	**Size**		**Ease of discovery**	**Ease of exploit**	**Awareness**	**Intrusion detection**
3	4	9	5		7	5	9	9
Overall likelihood: 6.375, HIGH								

Impact								
Technical impact					Business impact			
Loss of confidentiality	**Loss of integrity**	**Loss of availability**	**Loss of accountability**		**Financial damage**	**Reputation damage**	**Non compliance**	**Privacy violation**
6	3	1	7		1	1	2	3
Overall impact: 3.000, MEDIUM								

Risk: *Victims forget to log off from shared computer* = **MEDIUM**

Likelihood								
Possible attackers					Security flaw			
Skill level	**Motive**	**Opportunity**	**Size**		**Ease of discovery**	**Ease of exploit**	**Awareness**	**Intrusion detection**
3	1	9	4		7	5	9	9
Overall likelihood: 5.875, MEDIUM								

Impact								
Technical impact					Business impact			
Loss of confidentiality	**Loss of integrity**	**Loss of availability**	**Loss of accountability**		**Financial damage**	**Reputation damage**	**Non compliance**	**Privacy violation**
6	3	1	7		1	1	2	3
Overall impact: 3.000, MEDIUM								

Risk: *Brute force attack on password field* = **MEDIUM**

Likelihood		
Possible attackers		Security flaw

Skill level	Motive	Opportunity	Size		Ease of discovery	Ease of exploit	Awareness	Intrusion detection
6	4	4	9		5	9	4	1
Overall likelihood: 5.250, MEDIUM								

Impact								
Technical impact					Business impact			
Loss of confidentiality	Loss of integrity	Loss of availability	Loss of accountability		Financial damage	Reputation damage	Non compliance	Privacy violation
6	5	1	7		1	1	5	7
Overall impact: 4.125, MEDIUM								

<u>Planting malicious program to client using email and URL spoofing in order to tamper with data</u>

<u>Risk</u>: *Mending with acts and/or amendments* = **MEDIUM**

Likelihood								
Possible attackers					Security flaw			
Skill level	Motive	Opportunity	Size		Ease of discovery	Ease of exploit	Awareness	Intrusion detection
6	4	4	6		3	3	4	3
Overall likelihood: 4.125, MEDIUM								

Impact								
Technical impact					Business impact			
Loss of confidentiality	Loss of integrity	Loss of availability	Loss of accountability		Financial damage	Reputation damage	Non compliance	Privacy violation
5	5	5	7		7	5	5	0
Overall impact: 4.875, MEDIUM								

<u>Changing log files</u>

<u>Risk</u>: *Changing log files after changing export options for acts/amendments* = **CRITICAL**

Likelihood								
Possible attackers					Security flaw			
Skill	Motive	Opportunity	Size		Ease of	Ease of	Awareness	Intrusion

level					discovery	exploit		detection
9	9	7	9		1	1	6	8
Overall likelihood: 6.250, HIGH								

Impact								
Technical impact					Business impact			
Loss of confidentiality	Loss of integrity	Loss of availability	Loss of accountability		Financial damage	Reputation damage	Non compliance	Privacy violation
7	7	7	9		7	5	7	7
Overall impact: 7.000, HIGH								

Breach through not well configured server

Risk: *Accessing backup files* = **HIGH**

Likelihood								
Possible attackers					Security flaw			
Skill level	Motive	Opportunity	Size		Ease of discovery	Ease of exploit	Awareness	Intrusion detection
9	4	4	9		3	3	4	9
Overall likelihood: 5.625, MEDIUM								

Impact								
Technical impact					Business impact			
Loss of confidentiality	Loss of integrity	Loss of availability	Loss of accountability		Financial damage	Reputation damage	Non compliance	Privacy violation
7	9	9	7		7	9	7	7
Overall impact: 7.750, HIGH								

Risk: *Accessing configuration files* = **HIGH**

Likelihood								
Possible attackers					Security flaw			
Skill level	Motive	Opportunity	Size		Ease of discovery	Ease of exploit	Awareness	Intrusion detection
9	4	4	9		3	3	4	9
Overall likelihood: 5.625, MEDIUM								

Impact								
Technical impact					Business impact			
Loss of confidentiality	Loss of integrity	Loss of availability	Loss of accountability		Financial damage	Reputation damage	Non compliance	Privacy violation
7	9	9	7		7	9	7	7
Overall impact: 7.750, HIGH								

Risk: *Accessing directory listing* = **HIGH**

Likelihood								
Possible attackers					Security flaw			
Skill level	Motive	Opportunity	Size		Ease of discovery	Ease of exploit	Awareness	Intrusion detection
9	4	4	9		3	3	4	9
Overall likelihood: 5.625, MEDIUM								

Impact								
Technical impact					Business impact			
Loss of confidentiality	Loss of integrity	Loss of availability	Loss of accountability		Financial damage	Reputation damage	Non compliance	Privacy violation
7	9	9	7		7	9	7	7
Overall impact: 7.750, HIGH								

Risk: *Accessing hidden files* = **HIGH**

Likelihood								
Possible attackers					Security flaw			
Skill level	Motive	Opportunity	Size		Ease of discovery	Ease of exploit	Awareness	Intrusion detection
9	4	4	9		3	3	4	9
Overall likelihood: 5.625, MEDIUM								

Impact								
Technical impact					Business impact			
Loss of confidentiality	Loss of integrity	Loss of availability	Loss of accountability		Financial damage	Reputation damage	Non compliance	Privacy violation
7	9	9	7		7	9	7	7

Overall impact: 7.750, HIGH

Overloading specific page with too many requests

Risk: *Buffer overflow attack on proposing acts/amendments* = **MEDIUM**

Likelihood								
Possible attackers					**Security flaw**			
Skill level	**Motive**	**Opportunity**	**Size**		**Ease of discovery**	**Ease of exploit**	**Awareness**	**Intrusion detection**
6	4	7	6		5	3	4	3
Overall likelihood: 4.750, MEDIUM								

Impact								
Technical impact					**Business impact**			
Loss of confidentiality	**Loss of integrity**	**Loss of availability**	**Loss of accountability**		**Financial damage**	**Reputation damage**	**Non compliance**	**Privacy violation**
6	5	9	7		3	4	5	5
Overall impact: 5.500, MEDIUM								

Overall risk rating

RISK NAME	**RISK IMPACT**
Tricking victims by email, call, URL, false representation online	Low
Victims use shared computer and they store they're credentials in a browser	High
Victims forget to log off from shared computer	Medium
Brute force attack on password field	Medium
Mending with acts and/or amendments	Medium
Changing log files after changing export options for acts/amendments	Critical
Accessing backup files	High
Accessing configuration files	High
Accessing directory listing	High
Accessing hidden files	High
Buffer overflow attack on proposing acts/amendments	Medium

III – Determine countermeasures and reduce risks

Countermeasure Identification

Threat Type	Countermeasure	Grouping	Mitigation techniques
Spoofing	Appropriate authentication Protecting/Hiding sensitive data Validation of inputs Encryption HTTPS		
Tricking users by false representation in person, through email, phone…		Not solved	Inform about the risk Mitigate Accept the risk
Users use shared devices		Not solved	Inform about the risk Mitigate Accept the risk
Unauthorized access/Sensitive data exposure (injections, manipulations…)		Solved	
Tampering	Appropriate authorization Hashes MACs Digital signatures Timestamps Key cryptography Tamper resistant protocols HTTPS		
Planting malicious programs		Partially solved	Inform about the risk Mitigate Accept the risk
Intercepting communication		Partially solved	Inform about the risk Mitigate Accept the risk
Injections (SQL, code)		Partially solved	Inform about the risk Mitigate Accept the risk

Repudiation	Digital signatures Timestamps Audit trails Token based authentication Role-based access		
Changing data		Solved	
Changing user's requests		Solved	
Changing loggers		Partially solved	Inform about the risk Mitigate Accept the risk
Information disclosure	Authorization Privacy-enhanced protocols Encryption Protect sensitive data HMACs Role-based access HTTPS		
Viewing app configuration		Solved	
Accessing sensitive data through not protected cached and history data		Solved	
Special characters attack		Partially solved	Mitigate Accept the risk
Keystroke logging		Not solved	Inform about the risk Mitigate Accept the risk
Denial of service	Appropriate authentication Appropriate authorization Token based authorization Filtering Throttling Quality of service		
Overload with requests		Partially solved	Mitigate Accept the risk
Intercepting communication		Partially solved	Inform about the risk Mitigate Accept the risk

Overload router with ping requests		Not solved	Do nothing Accept the risk
Elevation of privilege	Run with least privilege		
Buffer overflow for exploiting vulnerabilities		Partially solved	Inform about the risk Mitigate Accept the risk
Malicious code injections		Partially solved	Inform about the risk Mitigate Accept the risk

** took in consideration environment influences which differ in the majority of users, as well as that this application is only tested locally with one user at the time, that's why there is many partially solved threats*

Mitigation strategies overview

NOT SOLVED	***PARTIALLY SOLVED***	***SOLVED***
Tricking users by false representation in person, through email, phone…	*Planting malicious programs*	*Unauthorized access/Sensitive data exposure (injections, manipulations…)*
Users use shared devices	*Intercepting communication*	*Changing data*
Overload router with ping requests	*Injections (SQL, code)*	*Changing user's requests*
Keystroke logging	*Changing loggers*	*Viewing app configuration*
	Special characters attack	*Accessing sensitive data through not protected cached and history data*
	Overload with requests	
	Intercepting communication	
	Buffer overflow for exploiting vulnerabilities	
	Malicious code injections	