TruckMotion Threat Model

Owner: DESOFS 3
Reviewer: DESOFS 3
Contributors: Nuno Marmeleiro, Rogério Sousa, Óscar Folha, Rafael Oliveira, Rafael Faísca
Date Generated: Thu Apr 18 2024

Executive Summary

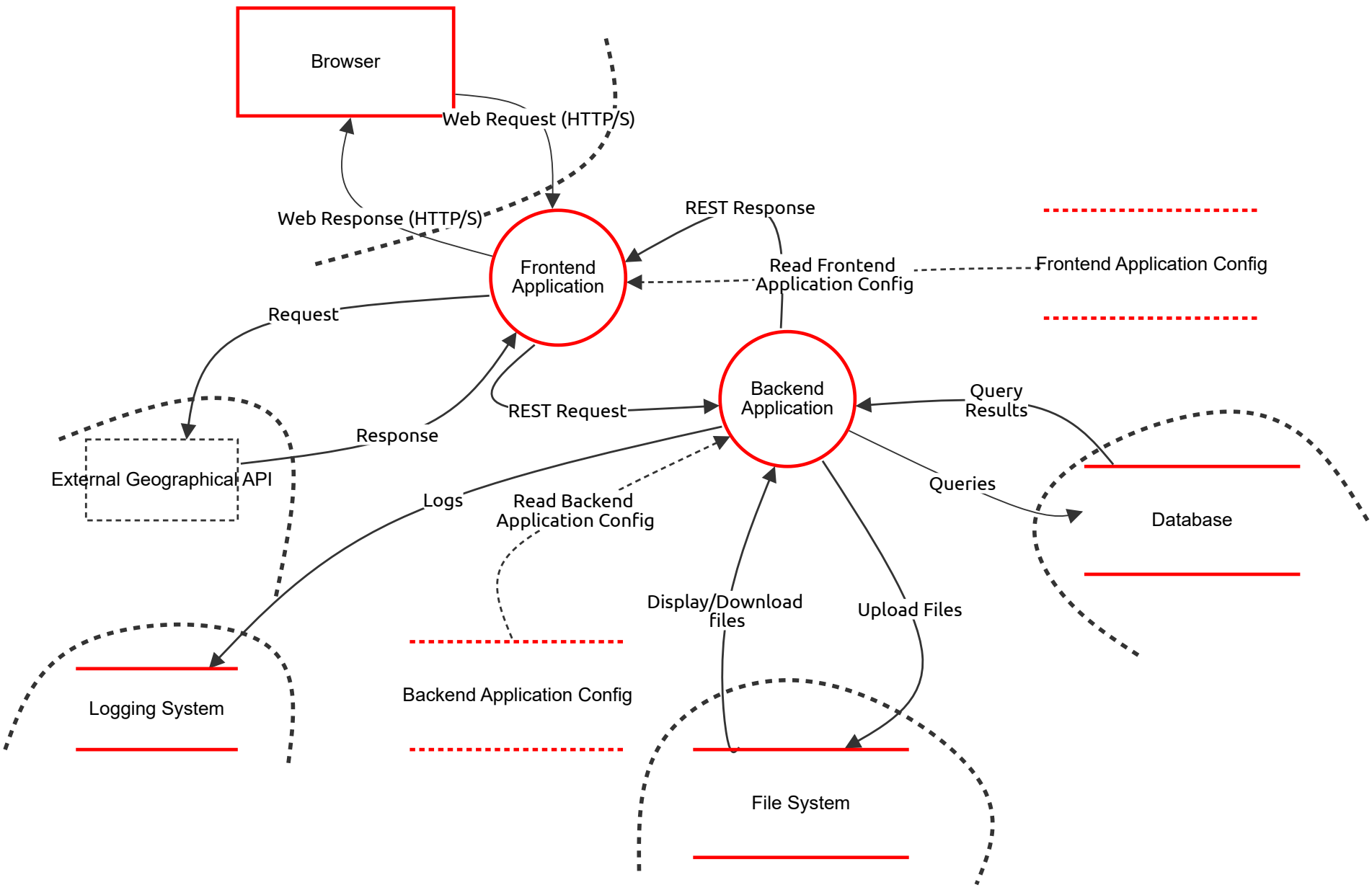
High level system description

Model for TruckMotion

Summary

Total Threats	23
Total Mitigated	10
Not Mitigated	13
Open / High Priority	8
Open / Medium Priority	5
Open / Low Priority	0
Open / Unknown Priority	0

Main Request Data Flow



Main Request Data Flow

Database (Store)

Number	Title	Type	Priority	Status	Score	Description	Mitigations
	Unauthorised access	Information disclosure	High	Mitigated		An attacker could make an query call on the DB.	Require all queries to be authenticated.
	Credential theft	Information disclosure	Medium	Open		An attacker could obtain the DB credentials and use them to make unauthorised queries.	Use a firewall to restrict access to the DB to only the Backend IP address.

Backend Application Config (Store) - *Out of Scope*

Number	Title	Type	Priority	Status	Score	Description	Mitigations
	Credentials should be encrypted	Information disclosure	High	Open		The Web Application Config stores credentials used by the Web App to access the message queue. These could be stolen by an attacker and used to read confidential data or place poison message on the queue.	The Message Queue credentials should be encrypted.

Backend Application (Process)

Process that contains all the interactions needed to the backend API application

Number	Title	Type	Priority	Status	Score	Description	Mitigations
32	Attacker sends too many requests	Denial of service	High	Open		An attacker send too many requests to API endpoint so that the service goes down.	Add a strategy that blocks requests when the amount is high and the interval of time is low.
33	Attacker sends API Request via Tool to steal data	Information disclosure	High	Mitigated		Attacker sends a request using another tool (i.e.: postman) with the known endpoint to gather data.	Add tokens or some type of authentication inside backend.
35	Attacker sends an API POST/Delete Request	Tampering	High	Mitigated		Attacker uses another tool instead of browser (i.e.: postman) to update or delete data inside the application.	Add token/authentication inside the backend. Do not allow delete endpoints.
36	Attacker steals Authentication Token	Spoofing	Medium	Open		Attacker successfully steals or even generates a valid authentication token, with this, gaining access to data through API endpoints.	Implement techniques to not use one static token, such as the refresh token.
37	User pretends to be someone else changing Header Parameter	Repudiation	Medium	Mitigated		The user, by manipulating the API Request changing the Header Parameter username, pretends to be someone else in the logging system and the actions taken are associated wrongly, making the logging invalid.	Implement logging based on the token used to get the user in question and not the header user name parameter.
39	User by using their token successfully performs actions outside their authorization	Elevation of privilege	Medium	Mitigated		A user, by sending an API Request to an End Point using their token, can successfully perform some action that is beyond their authorization.	Implement authorization based on token and more inside the backend API.

Browser (Actor)

Number	Title	Type	Priority	Status	Score	Description	Mitigations
16	GPS Spoofing	Spoofing	Medium	Open		Users can use GPS Spoofing applications to fake the geographical location.	Apply mechanisms to detect suspicious changes of location that can be associate with GPS Spoofing, for example, if a driver changes from a location to another distancing more than 1km faster than a second, it triggers a warning.

Queries (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Upload Files (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Query Results (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Web Request (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations
	Data flow should use HTTP/S	Information disclosure	High	Mitigated		These requests are made over the public internet and could be intercepted by an attacker.	The requests should require HTTP/S. This will provide confidentiality and integrity. HTTP should not be supported.

Web Response (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations
	Data flow should use HTTP/S	Information disclosure	High	Mitigated		These responses are over the public internet and could be intercepted by an attacker.	The requests should require HTTP/S. This will provide confidentiality and integrity. HTTP should not be supported.

Read Backend Application Config (Data Flow) *- Out of Scope*

Number	Title	Type	Priority	Status	Score	Description	Mitigations

REST Request (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

REST Response (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Read Frontend Application Config (Data Flow) *- Out of Scope*

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Display/Download files (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Request (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Response (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Logs (Data Flow)

Number	Title	Type	Priority	Status	Score	Description	Mitigations

File System (Store)

Number	Title	Type	Priority	Status	Score	Description	Mitigations
2	Upload Malicious Files	Tampering	High	Open		By letting the application have a file upload system, the files need to be stored somewhere, in a physical system. This allows malicious users to upload files that are bad to tamper the server.	Add security processes such as define the maximum upload file or even only accept a certain type of extension.
31	Execute command line to access other files	Information disclosure	High	Open		User access the filesystem by the application and by running some command can access other files.	Make the filesystem have files separated by folders or other strategy so that users can not access other files.

Frontend Application (Process)

Process that contains all the interactions needed with the front end application

Number	Title	Type	Priority	Status	Score	Description	Mitigations
21	Credentials Spoofing	Spoofing	High	Open		Get users credentials to access frontend application and pretend to be user with the stolen credentials.	Add strong password policies.
22	XSS Injection	Tampering	High	Open		Cross Site Injection	Never use unsafe functions, such as eval, and verify every input that can be inserted by the user
25	Change user sensible data	Tampering	Medium	Open		User with stolen credentials changing sensible data such as Bank Information, Address and more.	Add notification for each change or even add 2 step confirmation, with e-mail or SMS Token.
26	Customer/Driver being able to perform Manager Actions	Repudiation	Medium	Mitigated		An user with less privilege being able to perform prohibited actions, such as a customer/driver being able to create users.	Add logging and roles verification.
27	Hacker send too much requests	Denial of service	Medium	Mitigated		Hacker send too much requests to the frontend page so that denies service.	Add configuration such as a limit of connections per machine or IP, so it blocks after a suspicious amount of requests in a short interval of time
28	Accessing others information altering accessible data	Information disclosure	Medium	Mitigated		A user, by changing something accessible, such as the request query string of the link, access other information, such as a customer accessing a service from another customer.	Added authorization and data protection in every page needed.
30	Execute client functions via console to access unauthorized pages	Elevation of privilege	Medium	Open		User executes client functions via console that redirects the browser to a page without privilege.	Do not render all client functions into the users browser so that the user doesn't execute them.

Frontend Application Config (Store) *- Out of Scope*

Number	Title	Type	Priority	Status	Score	Description	Mitigations
	Credentials should be encrypted	Information disclosure	High	Open		The Web Application Config stores credentials used by the Web App to access the message queue. These could be stolen by an attacker and used to read confidential data or place poison message on the queue.	The Message Queue credentials should be encrypted.

External Geographical API (Actor) *- Out of Scope*

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Logging System (Store)

Number	Title	Type	Priority	Status	Score	Description	Mitigations
40	Attacker successfully gathers credentials User with write privilege	Repudiation	High	Open		Attacker successfully gains control of the User with write privilege, now being able to alter the logging or even deleting it.	Create strong password policy for this particular user.