URL shortener (Engineer 4)

Owner: Engineer 4
Reviewer:
Contributors:
Date Generated: Mon Dec 16 2024

Executive Summary

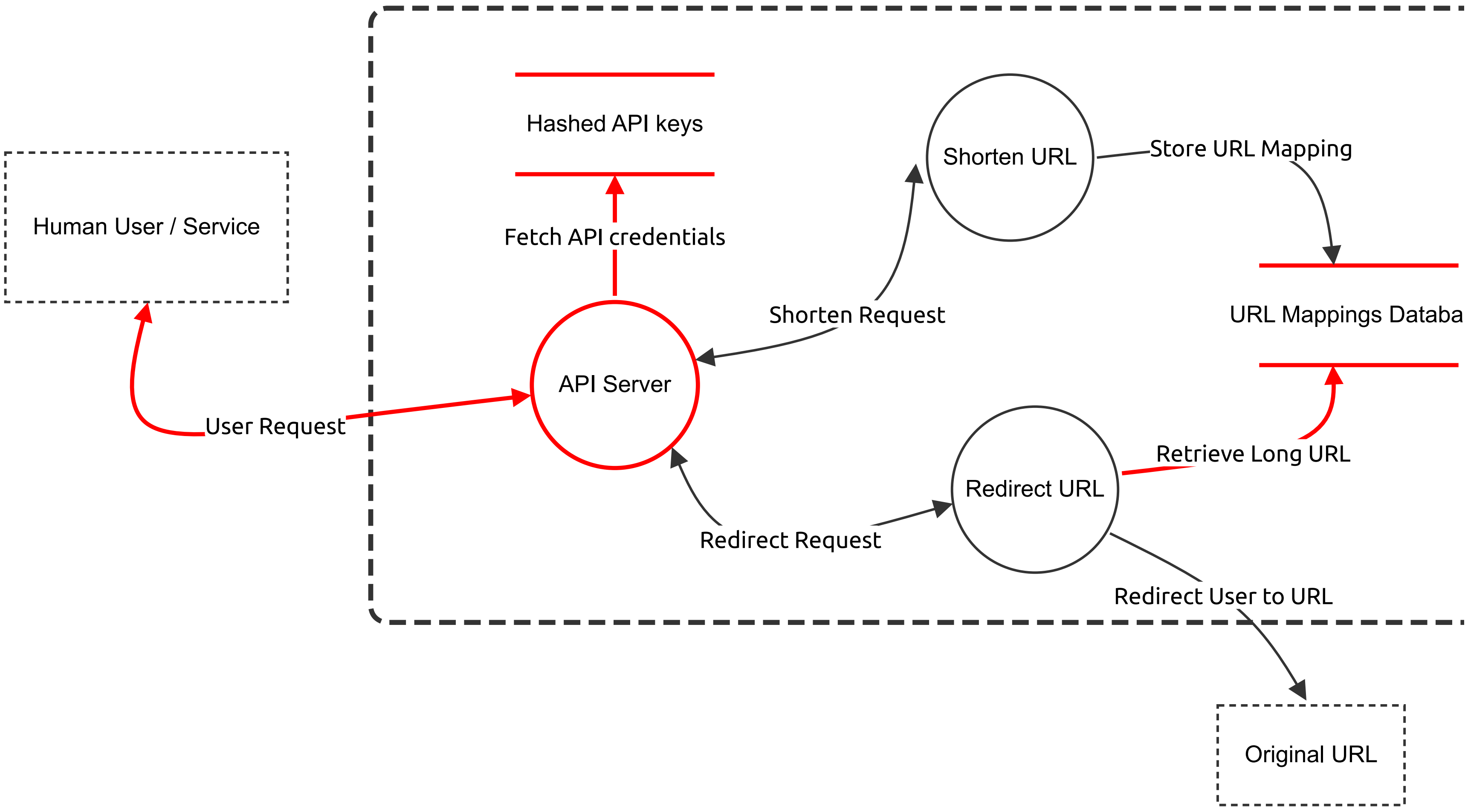
High level system description

The URL Shortener Service aims to provide a simple and efficient way for users to convert long URLs into short, easily shareable URLs. The service ensures functionality, reliability, and security while interacting with users and handling their data.

Summary

Total Threats	13
Total Mitigated	0
Not Mitigated	13
Open / High Priority	0
Open / Medium Priority	13
Open / Low Priority	0
Open / Unknown Priority	0

URL Shortener Service Architecture



URL Shortener Service Architecture

Human User / Service (Actor) *- Out of Scope*

Reason for out of scope: Incoming requests outside of application scope

Description: External services or human users may interact with the API for bulk operations or integrations.

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Redirect URL (Process)

Description: Enables users to resolve short URLs and be redirected to the corresponding long URLs via a dedicated API.

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Shorten URL (Process)

Description: Allows users to submit long URLs and receive shortened URLs via a dedicated API.

Number	Title	Type	Priority	Status	Score	Description	Mitigations

URL Mappings Database (Store)

Description: Data store containing the mappings between short URLs and their corresponding long URLs.

DATA STORED: Long URL, Short URL, UserID, CreationDate

Number	Title	Type	Priority	Status	Score	Description	Mitigations
7	No backup in case of data loss	Denial of service	Medium	Open		In case of no backup in place, the information from database may be lost	Perform regular backups, use solutions from cloud vendors like AWS Backup

Original URL (Actor) *- Out of Scope*

Reason for out of scope: Original URL not in scope of URL shortener application

Description: Destination server of the original URL

Number	Title	Type	Priority	Status	Score	Description	Mitigations

API Server (Process)

Description: The API Server is the central component of the URL Shortener Service. It manages all interactions with users and external services, handling requests for both shortening URLs and redirecting from short URLs to their original long URLs.

Number	Title	Type	Priority	Status	Score	Description	Mitigations
9	Ransomware attack	Denial of service	Medium	Open		Malicious link may lead to downloading a malicious file	Input validation and filtering
12	Missing access controls for POST, PUT and DELETE	Elevation of privilege	Medium	Open		Accessing API with missing access controls for POST, PUT and DELETE.	Use API Gateway and I AM roles?
16	APIs are no monitored for suspicious activity	Repudiation	Medium	Open		No monitoring for suspicious activity	

Hashed API keys (Store)

Description: Database containing hashed API keys for authentication purposes.

DATA STORED: Hashed API Key, UserID

Number	Title	Type	Priority	Status	Score	Description	Mitigations
1	Weak hashing algorithm	Information disclosure	Medium	Open		Usage of weak hashing algorithm (MD5, SHA-1) that is easy to decrypt the information	Use more secure algorithms like SHA-2 or SHA-3
4	No backup in case of data loss	Denial of service	Medium	Open		In case of no backup in place, the information from database may be lost	Perform regular backups, use solutions from cloud vendors like AWS Backup

Redirect Request (Data Flow)

Description: The user submits a short URL to the API and gets redirected to the original Long URL.

DATA EXCHANGED: Short URL, Long URL

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Redirect User to URL (Data Flow)

Description: The user is redirected to the Long URL.

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Fetch API credentials (Data Flow)

Description: Hashed API keys are pulled to authenticate the user before performing any operation.

DATA EXCHANGED: Hashed API key

Number	Title	Type	Priority	Status	Score	Description	Mitigations
10	Weak TLS algorithm	Tampering	Medium	Open		Weak TLS algorithms (TLS 1.0, 1.1, 1.2 (maybe also can be mentioned)) don't provide secure communication and may allow modification of the transferred data	Use TLS 1.3

Shorten Request (Data Flow)

Description: The user submits a long URL to the API for shortening.

DATA EXCHANGED: Short URL, Long URL, UserID

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Store URL Mapping (Data Flow)

Description: The API Server processes the request for shortening a URL, generates a short URL, and stores the mapping between the short URL and the long URL in the URL Database.

DATA EXCHANGED: Short URL, Long URL

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Retrieve Long URL (Data Flow)

Description: Long URL mapping is pulled from the database in order to redirect user.

DATA EXCHANGED: Short URL, Long URL, UserID

Number	Title	Type	Priority	Status	Score	Description	Mitigations
15	URL exposes session identifier	Information disclosure	Medium	Open		This may lead to cross-site scripting or sniffing attack, steal the session ID and hijack the user's session. It may also raises privacy concerns	Use TLS (1.3), generate a long, random session

User Request (Data Flow)

Description: The User sends a request to the API Server to shorten a long URL or to get the redirect URL.

DATA EXCHANGED: Long URL, Short URL, API key

Number	Title	Type	Priority	Status	Score	Description	Mitigations
1	Unsafe link sent by an attacker	Tampering	Medium	Open		Unsafe link may redirect to a malicious page (at the first sight this can look legitimate). Additionally, there might be a file download after clicking the page	Hover over the link and see what appears in the preview - check if the link is the same Input validation and filtering
5	Usage of compromised API keys	Information disclosure	Medium	Open		An attacker may use a compromised API keys that allow access	Rotate API keys on a regular basis
11	Parameter tampering	Tampering	Medium	Open		Manipulation of parameters in URL	Input validation, usage of HTTPS protocol

Number	Title	Type	Priority	Status	Score	Description	Mitigations
14	URL exposes session identifier	Information disclosure	Medium	Open		This may lead to cross-site scripting or sniffing attack, steal the session ID and hijack the user's session. It may also raises privacy concerns	Use TLS (1.3), generate a long, random session
19	Sensitive data sends in URL	Denial of service	Medium	Open		Sensitive and authentication information like tokens and passwords can be send in the URL	