Viestintäviraston julkaisun 004/2017 J LIITE 2

LIPPU-API: Security Considerations

Interoperability of ticket and payment systems project
27th of November 2017

Viestintävirasto

Contents

1 Introduction

This document outlines security considerations that API implementors should take into account when designing and implementing the Lippu-API.

The up-to-date version of this document is available on GitHub:
https://github.com/finnishtransportagency/lippu-api/blob/master/docs/Security.md

2 Threat modeling

Threat modelling is a good practise to approach security for a software application. In threat modelling, you try to identity, quantify and address security risks of the application. In short the aim is to identify and mitigate risks that have high probability or have severe consequences. There are different guides available for threat modeling, one is the Application Threat Modeling by the Open Web Application Security Project (OWASP). There is also an ongoing project to translate the guide to Finnish at Github: Uhka-analyysi. It is suggested to incorporate threat modelling practises into the software development process.

3 Layered security architecture and firewall considerations

Layered security architecture, sometimes referred as *onion model*, is a commonly used security architecture style to build software systems. Layered security architecture, the security is handled in different layers and the most critical information is located in the most inner layer. Then there are outer layers that protect the inner layers, the attacker must penatrate all of the outer layers before getting access to the most critical information. It is suggested to use layered security architecture when designing the software systems implementing the Lippu-API. It is also suggested to allow only required network connections in firewall policy and set the default policy to deny.

4 HTTP considerations

The Lippu-API is built on top of HTTP-protocol, all the security considerations of the HTTP-protocol as defined in the section-9 of the Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing -standard and the related specifications are relevant also for the Lippu-API.

5 TLS considerations

All traffic between the client and the server must be encrypted. The server side must support TLS 1.2 (RFC5246) encryption and may support additional transport-layer mechanisms. Client side should perform a TLS/SSL server identity check when using encrypted TLS connection. When the client application is not a private individual, but an organisational user, the implementors should seriously consider using client-side certification authentication to do TLS-level authentication. Additional information can be found on RFC6125 and RFC7525.

6 Authentication and authorization considerations

Requests to the secured endpoints must be protected with the authentication token received from the authentication service. Separate consideration must be placed on the authentication token's expiry time, shorter expire time reduces attacking time if the authentication token falls into hands of a malicious attacker. Also separate process to revoke authentication could be used, the downside is that is removes the stateless aspect of the authentication verification.

In addition to the authentication, the authorization model should be designed with *principle of least privilege* (also known as the principle of minimal privilege or the principle of least authority). In *principle of least privilege* access to the requested resources must be authorized accordingly and allow access only to the resources that are required to processing of the request.

7 Secure data storage

Service administrators should follow industry best practises on protecting credentials and sensitive data. See section-5.1.4.1 of the OAuth 2.0 Threat Model and Security Considerations for common practises to follow. Also see related items in the common web application vulnerabilities-section.

8 Privacy considerations

The sensitive and personal data stored by the service must be protected accordingly. If the service operates in EU it should take account the requirements from the General Data Protection Regulation (GDPR). Also separate considerations should be places on application logging and prevent logging of sensitive information to the application logs.

9 Automated log monitoring considerations

Automatically gathering and processing log information is useful approach to notice security anomalies. It is suggested to develop a baseline of the incoming trafic (what kind of requests are coming in, the amount of requests etc) and then use monitoring process to detect anomalies by comparing the realtime trafic to the baseline.

10 Automated security tools

There are different tools available to do automated security scanning for a web application, like Burp, OWASP Zap,Nmap etc. It is suggested to run atleast automatic security scans when developing a web application and before publishing it. There are also different kinds of static code analysing tools available for different programming languages to help detecting security and other bugs from code.

11 Common web application vulnerabilities

OWASP maintains a list of most used vulnerability categories. Implementors should go the through the list of the vulnerability categories and try mitigate and prepare to the relevant vulnerability categories. There is a separate cheat sheet to mitigate and protect against the common vulnerabilities. At least the following are relevant vulnerability categories from the 2013 top ten list (check also relevant ones from newer versions):

- Injection
- Broken Authentication and Session Management
- Sensitive Data Exposure
- Using Components with Known Vulnerabilities

12 Links

- RFC7230: Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing
- RFC5246: The Transport Layer Security (TLS) Protocol Version 1.2
- RFC6125: Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)
- RFC7525: Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)
- RFC6810: OAuth 2.0 Threat Model and Security Considerations
- OWASP Top Ten Project
- OWASP Top Ten Cheat Sheet
- OWASP Application Threat Modeling
- Burp Suite Scanner
- OWASP Zap
- Nmap Network Security Scanner

Viestintävirasto
PL 313
Itämerenkatu 3A
00181 Helsinki
puh: 0295 390 100
fax: 0295 390 270
www.viestintävirasto.fi