FH | JOANNEUM
University of Applied Sciences

Best Practice Authentication and Authorization

in distributed Business Scenarios with OpenID Connect and OAuth 2.0

Master Thesis
submitted in conformity with the requirements for the degree of
Master of Science in Engineering (MSc)
Master's degree programme **IT & Mobile Security**

FH JOANNEUM (University of Applied Sciences), Kapfenberg

Supervisor: Elmar Krainz, FH JOANNEUM Kapfenberg

submitted by: Cornelia Rauch
personnel identifier: 1610419026

June 2016 <edit date!>

Assignment for the master thesis of
<your name>
Matr. no. 1400000000

Subject:
“<Title of Your Thesis>”

Abstract

Write your abstract here.

<place>, <date>

Academic adviser:
<firstname lastname>

<your name>

Formal declaration

I hereby declare that the present master's thesis was composed by myself and that the work contained herein is my own. I also confirm that I have only used the specified resources. All formulations and concepts taken verbatim or in substance from printed or unprinted material or from the Internet have been cited according to the rules of good scientific practice and indicated by footnotes or other exact references to the original source.

The present thesis has not been submitted to another university for the award of an academic degree in this form. This thesis has been submitted in printed and electronic form. I hereby confirm that the content of the digital version is the same as in the printed version.

I understand that the provision of incorrect information may have legal consequences.

Kapfenberg, 28.09.2018

Cornelia Rauch

Acknowledgement

Thanks to . . .

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This template shall provide some consdiderations and text examples for your Master's thesis.

Background. Describe the background, the prerequisites for your work ...

Objective. The aim of this master's thesis is ...

Terms and definitions. Technical terms ... abbreviations are summarised at the end (in "Acronyms"), e.g. application binary interface (ABI) or man-in-the-middle (MITM). If ABI is referenced again, only the acronym is printed (as hyperlink though).

Corre et al. (2017), *Why can't users choose their identity provider* (ibid.) Boyed (2012), Lynch (2011), Todorov (2007), Procházka, Kouřil, and Matyska (2010), Tomkins (2009), Corre et al. (2017), Neumann (2013), Foundation (2018), Tome et al. (2011), Grassi, Garcia, and L. (2017), Brooks et al. (2017), Dingle and Bary (2015), Xu (2015), Dasgupta, Arunava, and Abhijit (2017)

Harvard citation style is implemented in this template: Batina et al. (2012), Fernàndez-Mir et al. (2012), Li et al. (2008)

1.1 Some LaTeX Basics

This section is a *really very short* summary of LaTeX features. Do not forget to remove it after finishing your thesis.

Here you have an included graphic (figure 1.1).

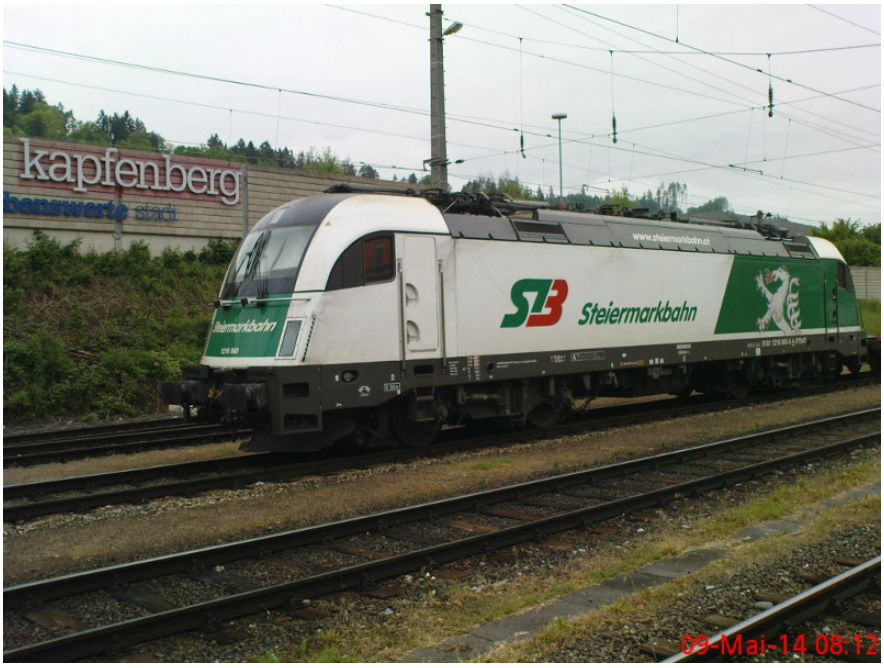

Figure 1.1: Train engine in Kapfenberg

Code listings require the *listings* package which, in turn, requires some settings[1]; see command `\lstset{}` in preamble of this template. Additionally the package *courier* should be used because the defaults do not provide for proper syntax highlighting.

```
1 void main(int argc, char *argv[])
2 {
3   printf("Hello world!");
4 }
```

Listing 1.1: Main programme

In order to see what's possible – here are two fancy tables: 1.1 and 1.2.

Version	Description	Author(s)	Date
1.0	Initial	Ohrt	July 15, 2014
1.1	Filled section "Open Issues"	Ohrt	July 16, 2014
1.2	Added section "Restrictions"	Ohrt	September 15, 2014

Table 1.1: Olive green heading

View also the preamble of this file for explanations.

[1] . . . because the defaults do not fit all purposes

Error	Solution
Java.lang.OutOfMemoryError: PermGen space *(32-/64-bit issue)*	-XX:MaxPermSize=1024M
Error occurred during initialization of VM *or* Could not reserve enough space for object heap	increase or remove -Xms value e.g. -Xms128m -Xmx512m (Eclipse default: -Xms40m -Xmx512m)

Table 1.2: A grey table

Here is a reference to listing 1.1.

Chapter 2

Authentication and Authorization

"Cloud-based services, the social Web, and rapidly expanding mobile platforms will depend on identity management to provide a seamless user experience." (Corre et al. (2017)).

Modern Devices are changing our every day life. They change the way how we access information, interact with each other and share content. With this change of user behavior also the way we think of authentication and authorization methods has to adjust. Users are find themselves struggling using multiple devices, accounts and services. The users burden of this site-by-site account management is putting security at risk. The goal of new authentication and authorization solutions is to help the user managing his accounts by providing single-sign-on, based on an exchange of identity-related assertion across security domains in a scalable way (Corre et al. (ibid.)).

2.1 Security Considerations

Before getting further into the topic of authentication and authorization, this section will shed light on some basic security Principles concerning authentication and authorization that help to understand the need for authentication and authorization mechanisms.

Some basic design principles formulated by Saltzer and Schroeder in 1975 where paraphrased by Neumann, 2013 and are still relevant today. The principles give a basic overview of what should be the focus when designing a secure system. The ten basic security principles are:

- The economy of mechanism: Keep the design as simple as possible.
- Fail-safe defaults: Access should be explicit permitted rather then explicit denied. The default should be that all access is denied, this is for example true for Access Control Lists (ACLs).
- Complete mediation: Every access to every object has to be checkt for authority without exceptions.
- Open design: The design of an application should not be secret, it can not be assumed that design secrecy will enhance security. This if for example true in cryptography, the design of cryptographic algorithms is available for the public, just the keys remain secret.
- Separation of privileges: Two keys should be used to protect resources if feasible and privileges should be separated.
- Least privilege: Every application and user should be provided with lest privileges they need to complete their job. The existence of overly powerful mechanisms such as superuser is inherently dangerous.
- Least common mechanism: Minimize the amount of mechanism common to more than one user and depended on by all users.
- Psychological acceptability: Keep it simple. The design of the interface for the user should be easy to understand so that the user routinely and automatically applies the protection mechanism correctly.
- Work factor: Make cost-to-protect commensurate with threats and expected risks. It should not be possible to circumvent the mechanismn with the resources of the attacker.
- Recording of compromises: Provide nonbypassable tamperproof trails of evidence.

Besides formulating these very important principles, Saltzer and Schroeder, 1975 also discus the terms "privacy" and "security". Those therms get frequently used in from authors writing about information storing systems, like in this paper, but are often used very differently. ibid. for example define "privacy" as the ability of an individual to specify whether, when and to whom sensible information is released. And the describe security as a technique that can control who is able to modify resources on a computer.

A more recent description of the terms security and privacy comes form Brooks et al., 2017. ibid. state the importance of the distinction between privacy and security.

This distinction is between privacy and security are essential because there are security issues unrelated to privacy, just as there are privacy issues that are unrelated to security. While security concerns arise from illegal system behavior, privacy concerns arise from byproducts of authorized personally identifiable information (PII) processing. Even byproducts that are considered to protect PII can raise security concerns, for example it can be questioned to which degree a tool for persistent activity monitoring should reveal information about individuals that are related to security purposes. However, security and privacy have in commont that they want to protect personal information and resources or PII.

These security issues and privacy issues of course raise certain concerns for users as well as for companies offering authentication services. When it comes to protecting personal resources, there are three primary concerns. According to Todorov, 2007 those three concerns are:

- Confidentiality: The term confidentiality means that personal information will be protected from disclosure to unauthorized individuals and organizations.
- Integrity: The integrity of information is protecting information from accidental or intentional tampering. Modification of confidential data may affect the data validity.
- Availability: Availability is the need to be able to access information at the time a user requests it. This includes the availability of services that expose information.

In an ideal world companies offering those services will do everything to use the best technologies regarding countermeasures to protect confidentiality, integrity, and availability. Establishing countermeasures, however, can be expensive leading to a tradeoff between costs and level of production of information. A typical approach to establishing information security management is to analyze risks first and then form counter measurements (ibid.).

2.2 Authentication

“Digital identity is the unique representation of a subject engaged in an online transaction. The process used to verify a subject’s association with their real-world identity is called identity proofing” (Grassi, Garcia, and L. (2017))

A digital identity as explained above is the result of what we call the authentication

process. It is a way of identifying the user as whom he claims to be. A very typical authentication process is performed by asking the user for its username and password. If the user provides a correct user name and password, an application assumes the user is indeed the owner of the account he wants to log on (Boyed (2012)).

The evidence provided by the user in the authentication process is called credentials. Most of the time as mentioned above credentials are provided in the form of username and password. However credentials also may take other forms like PIN's, key cards, eye scanners and so on (Todorov (2007)).

Credentials, which prove the identity of an entity and are used as authenticators in authentication systems, are called factors. Grassi, Garcia, and L. (2017) categorize following types of factors:

- Something the user knows - Cognitive information the user has to remember. Examples include passwords, PIN, answers to secret questions.
- What the user has - something the user owns. Examples include a security token, driving license, one-time password (OTP). What the user is - biometric information of the user. Examples include fingerprint, voice, and face.
- What the user is - biometric information of the user. Examples include fingerprint, voice, and face.

Other types of information which are not considere athentication factors but can be used to enrich the authentication proceess according to Dasgupta, Arunava, and Abhijit (2017) are:

- Where the user is - the location was the user can be used as a fourth factor of authentication of a user. Examples include GPS, IP addresses.
- When the user logs on - Time can also be extracted as a separate factor. Verification of employee's identification in different office hours can prevent many kinds of grave data breaches. The time factor can easily prevent online banking fraud events to a great extent.

To secure a solution properly it should at least use two factors of the three listed above.To make use of more then one factor of a pool of potential credentials to verify the identity of a user is referred to as Multi-factor Authentication (MFA). The goal of multi-factor authentication is it to provide a layered defense and make it harder for unauthorized individuals to gain access. If one of the factors breaks, the service can still rely on the non-compromised authentication factors (Dasgupta, Arunava, and Abhijit (ibid.)).

Using just one factor is called Single Factor Authentication(SFA). Dasgupta, Arunava, and Abhijit, 2017 clearly describes the drawbacks SFA has compared to MFA, primarily the universal used password-based authentication.The user needs to remember different passwords for multiple accounts therefore the user often reuses one password also known as password fatigue.

In an Interview by Tomkins, 2009 with Jon Brody, he explains Password Fatigue like the following. An average user has 15 accounts; some people might even have up to 30 accounts - far too much to manage appropriately. Users then tend to adopt specific password patterns like using simple passwords for nontransactional sites and complex passwords for banking sites. Since many complex passwords are hard to remember users also often reuse passwords for different services at one point - this is called password fatigue.

Besides password fatigue Todorov, 2007 draws attention to one of the significant challenges of secure user authentication represented by default passwords. Vendors often ship their devices with pre-configured standard passwords. Although vendors recommend changing default passwords, system architects and engineers often fail to do so because they are more focused on the business logic than on security causing security issues. Systems with default passwords are more straightforward to attack, for example by knowing or guessing the device the attacker has an easy time authenticating with the system accordingly.

Due to the problem of password fatigue and default passwords, one factor like a password might easily get compromised. Now the user can no longer use the service until the system is repaired which can lead to a delay of the user when trying to access necessary information. Also, there is a risk that the user does not notice that the single factor is compromised which can lead to devastating effects (Dasgupta, Arunava, and Abhijit, 2017).

When designing an authentication process with using multiple factors the designers of the process should be very aware of the type of application and the information that has to be secured. For example a solution for an international bank should have different standards then an app for a making an grocery list. On the one hand, difficult and complex authentication processes for trivial applications might scare away users. On the other hand simple methods for applications protecting sensitive data might drive users away as well. Application designers should try to always find a middle way that suits both parties, the application owners and the users (Grassi, Garcia, and L., 2017).

The factors are an important part of the authentication process which result should be

an authenticated user. However, to become an authenticated user the user has to go through certain steps on the way. These steps the user has to go through are enabled be typical components unique to the authentication process. Todorov, 2007 identifies three typical components that are part of the authentication:

- The Supplicant, is the party that provides the evidence to prove the identity of a user or client. The result of the authentication process should be the authenticated user or client.
- The Authenticator, also called server, is responsible for ascertaining the user identity. Once the identity is proved, the authenticator can authorize or audit the user access to resources.
- Security Authority Database, which is storage or a mechanism to check the user's credentials. The storage can be represented by as much as a flat file, a server on the network providing centralized user authentication or a distributed authentication server.

// insert picture here //

It is vital that all the components of a user authentication system can communicate independently of each other. Whether or not all communication channels are used depends on the authentication mechanism and the model of trust that it implements. For example, the Kerberos authentication protocol does not feature direct communication between the authenticator and security server (ibid.).

Application security is a complex task and developing a customized siloed identity solution can be expensive. A Stand-alone identity store can besides being expensive also causes information assurance and administrative problems for organizations (JerichoSystems, 2018).

A federate authentication or identity federation says Boyed, 2012 is a system that is maintaining its accounts, for example username and password databases, with the help of a third party service. Often big cooperate IT environments already use such solutions. Environment applications for example may trust an Active Directory server, an LDAP server or an SAML provider. Grassi, Garcia, and L., 2017 also claims that identity federation is preferred over some siloed identity solution that each serve a single agency or Relying Party (RPs). Furthermore ibid. lists certain benefits that come with using federated architectures, as can be examined before.

- Enhanced user experience. For example, an individual can be identity proofed once and reuse the issued credential at multiple RPs.

- Cost reduction to both the user (reduction in authenticators) and the agency (reduction in information technology infrastructure).
- Data minimization as agencies do not need to pay for collection, storage, disposal, and compliance activities related to storing personal information.
- Pseudonymous attribute assertions as agencies can request a minimized set of attributes, to include claims, to fulfill service delivery.
- Mission enablement as agencies can focus on the mission, rather than the business of identity management.

To reflect on authentication it can be said that authentication is a very important part of every applications. As users are getting more concerned on security the pressure on developers grows to provide a solution that secures sensible data while keeping up usability standards, which can often be a trade off. Complex applications need complex security which can mean high costs for individual developed solutions, therefor application developers should also think about using federate authentication solutions.

2.3 Authorization

The authentication and authorization process are very closely related to each other and for users often hard to separate. After the authentication process of a user the application has now proof that the user is who he claims to be, but not every user is the same. After the user authenticates, the user may want to access data or services. Based on the information provided by the authentication process the application has the possibility to allow or deny the user to access information or services. In other words we have to check if the user is authorized to access data of a service. Furthermore authorization, offers granular control to distinguish between read, write or execute access to individual resources, typically access control list (ACL) are used for each operation (Todorov, 2007, Boyed, 2012).

To receive the information necessary to make access decisions, systems offer a user login process or sign-in process. The process initiates the authentication process between user and the system. As a result of this process, the user receives a system or application specific structure called an access token. The access token holds information about the user which will indicate what kind of resources the user can access. For every action the user then has to provide the access token and based on the information provided within the access token is then either granted or denied access Todorov, 2007.

Access Control lists

Delegated Authorization Delegated authorization is granting access to another person or application to act on behalf of the user. Boyed, 2012

2.4 Single Sign-On

The aim of Single Sign-On(SSO) is to design an authentication system that serves the interests of the user as well as the interests of the service provider. Whereas the user prefers a a simple process, the service provider requires a complicated authentication procedure. Ironically trying to make the authentication procedure more save often leads to weakening the whole system due to the user always finding new ways to bypass it. An example mentioned before is password fatigue. Another challenge that SSO is trying to takel is that classic web authentication solutions that require the user to login with a password, only authenticate the user and are not capable of providing access control or revealing additional information about the user. Most SSO solutions try to combine the authentication process and authorization (Procházka, Kouřil, and Matyska, 2010).

According to Lynch, 2011 two SSO solutions gained broad acceptance. On the one hand, SAML-based federations using SOAP, focusing on large enterprises also including governments and educational networks. On the other hand the Web Authorization Protocol was introduced which is a combination of the Protocols OpenID and OAuth. SAML federations have been customized to address the security concerns of those institutions that typically have a large user base, significantly protected resources, complex authorization patterns and data and services spread across multiple domains. However in a Web 2.0 world, the SAML solutions where seen as too rigid and too severe to mantain; a lightweight SSO was needed. This was when the Web Authorization Protocol comes into play. The approach of this protocol is taking advantage of the lightweight RESTful APIs which are reusing the existing HTTP architecture features and the JavaScript Object Notation(JSON).

One of the very well-known solutions based on Security Assertion Markup Language version 2 (SAML2) is Shibboleth. Shibboleth is one of the leading middlewares for building identity federations in a higher education sphere. To offer authentication, authorization and attribute assertion between entities. Procházka, Kouřil, and Matyska, 2010 identifies following entities defined by Shibboleth are:

- Identity Provider (IdP)

- Service Provider (SP)
- Discovery Service
- Metadata Operator
- Federation Operator

The discovery service is used to find the users organization IdP. The IdP defines an attribute release policy and releases diffrent sets af the user's attributes to different SPs. Users are only able to agree or disagree with the whole set of attributes because the decision comes from the IdP. When the user tries to log on, to a service of a Service Provider, he gets redirected to a page where he can choose and IdP previously found by the Discovery Service. If an SP wants to provide its service to multiple federations it has to negotiate policy and technical detail with each federation operator. The federation operator manages the federation policies and introduces SPs and IdPs. Furthermore the federations operator has to maintain and manage the information about all entities in an federation which is contained in the Metadata. A problem with this architecture is that the SP has to keep track of changes of the technical specification of various federation operators to maintain the configuration for each federation operator. A solution would be a significant federation registration, but this is indeed not possible because of technical and administrative severity and political will. This solution is also somewhat misleading for users since they have to maintain multiple credentials, select from multiple identifiers and so on Procházka, Kouřil, and Matyska, 2010.

According to ibid. Shibboleth Shibboleth is too restrictive, a solution with a centrally managed point of IdPs and SPs is preferred. Also, users should not have to deal with redundant accounts.

Chapter 3

Conclusion and Outlook

Your text here ...

Acronyms

ABI application binary interface

ACL access control list

GUI graphical user interface

KISS keep it small and simple

MITM man-in-the-middle

OS operating system

UART universal asynchronous receiver/transmitter

UID unique identifier

Bibliography

Batina, Lejla, Stefaan Seys, Dave Singelée, and Ingrid Verbauwhede (2012). "Hierarchical ECC-Based RFID Authentication Protocol". In: *Proceedings of the 7th International Conference on RFID Security and Privacy*. RFIDSec'11. Amherst, MA: Springer-Verlag, pp. 183–201. ISBN: 978-3-642-25285-3. DOI: 10.1007/978-3-642-25286-0_12. Available from: <http://dx.doi.org.acm.perm.fh-joanneum.at/10.1007/978-3-642-25286-0_12>.

Boyed, Ryan (2012). *Getting Started with OAuth 2.0 - Programming Clients for Secure Web API Authorization and Authentication*. O'Reilly Media.

Brooks, Sean, Michael Garcia, Naomi Lefkovitz, Suzanne Lightman, and Ellen Nadeau (Jan. 2017). *An Introduction to Privacy Engineering and Risk Management in Federal Systems*. NISTR 8062. NIST (National Institute of Standards and Technology).

Corre, Kevin, Oliver Barais, Gerson Sunyé, Frey Vincent, and Jean-Michel Crom (2017). *Why can't users choose their identity provider*. Available from: <https://petsymposium.org/2017/papers/issue3/paper18-2017-3-source.pdf> [May 2018].

Dasgupta, Dipankar, Roy Arunava, and Nag Abhijit (2017). *Advances in User Authentication*. Springer International Publishing AG. ISBN: 978-3-319-58808-7.

Dingle, P. and T. Bary (Sept. 2015). *OpenID AccountChooser Basic API Profile 1.0*. draft-accountchooser-basic-profile-07. Google Inc.

Fernàndez-Mir, Albert, Rolando Trujillo-Rasua, Jordi Castellà-Roca, and Josep Domingo-Ferrer (2012). "A Scalable RFID Authentication Protocol Supporting Ownership Transfer and Controlled Delegation". In: *Proceedings of the 7th International Conference on RFID Security and Privacy*. RFIDSec'11. Amherst, MA: Springer-Verlag, pp. 147–162. ISBN: 978-3-642-25285-3. DOI: 10.1007/978-3-642-25286-0_10. Available from: <http://dx.doi.org/10.1007/978-3-642-25286-0_10>.

Foundation, OpenID (2018). *Account Cooser - Developer's Guide*. Available from: <http://www.accountchooser.net/developers> [June 2018].

Grassi, Paul A., Michael E. Garcia, and Fenton James L. (July 2017). *Digital Identity Guidelines*. Special Publication 800-63-3. NIST (National Institute of Standards and Technology).

JerichoSystems (2018). *Identity Silo*. Available from: <https://www.jerichosystems.com/technology/glossaryterms/identity_silo.html> [June 2018].

Li, Yingjiu, Robert H. Deng, Junzuo Lai, and Changshe Ma (Dec. 2008). "On Two RFID Privacy Notions and Their Relations". In: *ACM Trans. Inf. Syst. Secur.* 14.4, 30:1–30:23. ISSN: 1094-9224. DOI: 10.1145/2043628.2043631. Available from: <http://doi.acm.org.acm.perm.fh-joanneum.at/10.1145/2043628.2043631>.

Lynch, Lucy (Sept. 2011). "Inside the Identity Game". In: *IEEE Internet Computing* 15.5, pp. 78–82.

Neumann, Peter G. (2013). *Principled Assuredly Trustworthy Composable Architectures*. Available from: <http://www.csl.sri.com/users/neumann/chats4.pdf> [May 2018].

Procházka, Michal, Daniel Kouřil, and Luděk Matyska (May 2010). "User centric authentication for web applications". In: *Collaborative Technologies and Systems (CTS)*. Chicago, IL, USA: IEEE. ISBN: 978-1-4244-6622-1.

Saltzer, Jerome H. and Michael D. Schroeder (1975). *The Prodection of Information in Computer Systems*. Available from: <http://www.cs.virginia.edu/~evans/cs551/saltzer/> [June 2018].

Todorov, Dobromir (2007). *Mechanics of User Identification and Authentication - Fundamentals of Identity Managment*. Auerbach Publications. ISBN: 978-1-4200-5219-0.

Tome, Basheer, John Bradley, Nat Sakimura, Long Kevin, Janrain P. Dingle, Axel Nennker, Andrew Csinger, Eric Sachs, Chuck Sievert, Wei Tu, Andrew Daheley, and Chris Messina (2011). *Account Chooser Working Group Chater proposal*. Available from: <http://ac.openid.net/wgproposal> [June 2018].

Tomkins, Benjamin (2009). "Dealing With Password Fatigue". In: *Forbes*.

Xu, Fangyuan (2015). *Security and Privacy Concern for Single Sign-on Protocols*. Tech. rep. Tufts University.