HACKMANIT

Penetration Test Report: *DENIC ID Relying Party - Member Login*

Version: 1.2

25.07.2019

Dr. Juraj Somorovsky

Phone: (+49)(0)234 / 45930961 | Email: Juraj.Somorovsky@hackmanit.de

Table of Contents

Project Information

Customer:	DENIC eG Kaiserstraÿe 75 - 77 60329 Frankfurt am Main, Deutschland
Contact:	Marco Sanz
Comissioned to:	HackmanIT GmbH Universitätsstraÿe 150 44801 Bochum, Germany
Project executive:	Dr. Juraj Somorovsky Phone: (+49)(0)234 / 45930961 Fax: (+49)(0)234 / 45930960 Email: Juraj.Somorovsky@hackmanit.de
Project members:	Mario Korth (Hackmanit GmbH) Mario Korth (Hackmanit GmbH) Mario Korth (Hackmanit GmbH) Mario Korth (Hackmanit GmbH)
Project period:	June 4, 2019 - June 11, 2019
Version of the report:	1.2

General Condition and Scope

The scope allowed was the following:

- `https://cours.*.epitech.eu/*`
- https://123.033.33.11
- https://0.929.22.11
- https://33.33.33.33
- `http://cours.*.epitech.eu/*`
- http://123.033.33.11
- http://0.929.22.11
- http://33.33.33.33
- `ftp://svn.epitech.eu/*`
- ftp://123.033.33.11
- ftp://0.929.22.11
- ftp://33.33.33.33
- `http://intranet.epitech.*/admin/*`
- `https://intranet.epitech.*/admin/*`
- `http://intranet.epitech.*/student/*`
- `https://intranet.epitech.*/admin/*`

Summary

DENIC ID is the first widely-deployed implementation of the ID4me protocol [1]. ID4me is a novel protocol for federated identity management whose two main goals are to provide (1) Authorization of a user for access to any third party accepting ID4me identifiers and (2) Controlled communication of the user's personal information to the third parties accessed by the user [1]. ID4me is based on well-established standards such as OpenID Connect [8] and the domain name system (DNS) [4].

Hackmanit GmbH was commissioned to perform a penetration test on a relying party in the context of DENIC ID - the new DENIC Member Login page. The penetration test was performed remotely with a total expense of 11 PT.

Weaknesses. During the penetration test, three weaknesses classified as Medium were identified. Two of these weaknesses relate to the insufficient protection against cross-site request forgery (CSRF) attacks. First, the login page does not contain CSRF protection mechanisms like CSRF tokens, which allows an attacker to force a victim to start an authentication flow without its consent. Second, the presence of the state parameter, which is used to protect against CSRF attacks in the OpenID Connect protocol, is not enforced by the relying party. This enables an attacker to log a victim into an account controlled by the attacker which might result in the victim revealing personal information or files to the attacker. The third weakness could allow an attacker to compromise the account of a victim due to faulty session and cookie management when the victim logs in again after a successful logout using the same browser. Some of the weaknesses identified during the penetration test are weaknesses in the library OpenID-Connect-PHP[1] which the tested relying party is based on. We responsibly disclosed these weaknesses to the library developers in June 2019 and supported them by implementing security fixes.

Structure. The report is structured as follows: In Section 2, the timeline of the penetration test is listed. Section 3 introduces our methodology, and Section 4 explains the general conditions and scope of the penetration test. In section 5, the scenario of the penetration test is described in detail. Section 6 provides an overview of the identified weaknesses and further recommendations. In Section 7, all identified weaknesses are discussed in detail and specific countermeasures are described. Section 8 summarizes our recommendations resulting from observations of the application. Finally, Section 9 lists additional tests that did not reveal any weaknesses.

References

In the following sections, we list the identified weaknesses. Every weakness has an identification name which can be used as a reference in the event of questions, or during the patching phase.

R01 Valid OpenID Connect Flow with a Missing `state` Parameter

General Description. The OpenID Connect standard suggests to use the nonce parameter to associate a Client session with an ID token, and to mitigate replay attacks [8]. The relying party randomly chooses a value for the nonce parameter and sends it to the identity authority in the authentication request. The identity authority later adds this value to the issued ID token. The relying party must verify that the nonce parameter is present when it receives the ID token, and contains the same value which was chosen earlier for this specific protocol flow.

Recommendation. We recommend to further increase the security of the relying party and the protection against well-known attacks, such as CSRF and replay attacks, by adding a binding between the authentication request and the ID token. This is achieved using the OpenID Connect parameter nonce in the way described above.

Weaknesses

In the following sections, we list the identified weaknesses. Every weakness has an identification name which can be used as a reference in the event of questions, or during the patching phase.

M01 Valid OpenID Connect Flow with a Missing `state` Parameter

Exploitability Metrics		**Impact Metrics**	
Attack Vector (AV)	**Netword**	Confidentiality Impact (C)	**Low**
Attack Complexity (AC)	**Low**	Integrity Impact (I)	**Low**
Privileges Required (PR)	**Low**	Availability Impact (A)	**None**
User Interaction	**Required**	Scope (S)	**Unchanged**
Subscore: **2.1**		Subscore: **2.5**	
Overall CVSS Score for M01		**4.6**	

General Description. cross-site request forgery (CSRF) is an attack in which an attacker tricks his victim into performing authenticated commands changing the application state [5] without the victim's consent. In OAuth and OpenID Connect the state parameter is used to mitigate cross-site request forgery (CSRF) attacks. It is randomly generated by the relying party at the beginning of each authentication flow. The redirect, which is used to send the code generated by the identity authority to the relying party, also contains the state parameter. This enables the relying party to verify that the authentication flow was triggered by the user.

Weaknessess. The relying party does not enforce the presence of a state parameter. If the state parameter is missing and only a valid code is provided, the relying party redeems the code at the identity authority and uses the issued ID token to successfully log in the user; see also Figure 2. 7

Counter-measures. The relying party must enforce the presence of the state parameter and validate that its value matches the value choosen at the beginning of the authentication flow.

Figure 2: Successful session initialization with a missing `state` parameter

Recommendation

In the following sections, we list the identified weaknesses. Every weakness has an identification name which can be used as a reference in the event of questions, or during the patching phase.

R01 Valid OpenID Connect Flow with a Missing `state` Parameter

General Description. The OpenID Connect standard suggests to use the nonce parameter to associate a Client session with an ID token, and to mitigate replay attacks [8]. The relying party randomly chooses a value for the nonce parameter and sends it to the identity authority in the authentication request. The identity authority later adds this value to the issued ID token. The relying party must verify that the nonce parameter is present when it receives the ID token, and contains the same value which was chosen earlier for this specific protocol flow.

Recommendation. We recommend to further increase the security of the relying party and the protection against well-known attacks, such as CSRF and replay attacks, by adding a binding between the authentication request and the ID token. This is achieved using the OpenID Connect parameter nonce in the way described above.