Backchannel attack allows an attacker to fetch an ID token through an intercepted authorization code

Critical **sagikazarmark** published **GHSA-vh7g-p26c-j2cw** on Oct 3

Package	
dex (Go)	
Affected versions	**Patched versions**
<= 2.34.0	2.35.0

Description

Impact

Dex instances with public clients (and by extension, clients accepting tokens issued by those Dex instances) are affected by this vulnerability.

An attacker can exploit this vulnerability by making a victim navigate to a malicious website and guiding them through the OIDC flow, stealing the OAuth authorization code in the process. The authorization code then can be exchanged by the attacker for a token, gaining access to applications accepting that token.

Steps to reproduce

1. A victim navigates to a malicious website

2. The webserver initiates a connection with a Dex instance directly - https://dexexample.com/auth/https:%252F%252Faccounts.google.com?access_type=online&client_id=example&nonce=2AaJAimQU9CbeOFsNra1d7CJTWB&redirect_uri=http%3A%2F%2Flocalhost%3A40393%2Fauth%2Fcallback&response_type=code&scope=openid+email&state=2AaJAjhpUmsB25csCo5muvorMTl. In this example, the Dex instance is hosted on `dexexample.com`, and the connector is `accounts.google.com`.

3. Dex returns a 302 Redirect to the connector IDP, https://accounts.google.com/o/oauth2/v2/auth?client_id=237800849078-hri2ndt7gdafpf34kq8crd5sik9pe3so.apps.googleusercontent.com&redirect_uri=https%3A%2F%2Fdexexample.com%2Fauth%2Fcallback&response_type=code&scope=openid+email&state=g3dkmpontsr3ugocoddjx72ef. The attacker records the state parameter value g3dkmpontsr3ugocoddjx72ef which will be used as the request ID later on.

4. The malicious website redirects the victim's browser to the connector IDP.

5. The user authenticates to the connector IDP. If they have authenticated before, they may not be presented with an authentication challenge. The user will silently be taken through the following steps:

   Authentication with the connector IDP, which redirects the browser to the Dex callback with a code - https://dexexample.com/callback?state=g3dkmpontsr3ugocoddjx72ef&code=4%2F0AX4XfWizg1PQEQNl18hmP0_YQ3iUYIl2ed13n9ikKr_ZcV7uCZpZaPcIlxBzX5QwFIcs-w&scope=email+openid+https%3A%2F%[2Fwww.googleapis.com](http://2fwww.googleapis.com/)%2Fauth%2Fuserinfo.email&authuser=0&hd=google.com&prompt=none

   Dex handles the callback, fetching the user claims from the connector IDP, persisting them and generating an OAuth code. Then Dex redirects the browser to the approval endpoint https://dexexample.com/approval?req=g3dkmpontsr3ugocoddjx72ef. Note that the req parameter is the same as the attacker's recorded state parameter.

   Dex uses the request ID to look up the OAuth code, and builds a redirect to the original callback with the code - http://localhost:40393/auth/callback?code=bz5p3oov2wlh5k3rboa4atxas&state=2AaJAjhpUmsB25csCo5muvorMTI.

In step 2., when the webserver initiates the connection to Dex and receives the redirect to the connector IDP, the webserver will persist the connector state parameter (`g3dkmpontsr3ugocoddjx72ef`), which is used as the request ID to later look up the OAuth code. As the user goes through the authentication flow with the connector IDP, the webserver will repeatedly request `/approval?req=<state>` . Once the user has successfully authenticated, if the webserver is able to call /approval before the victim's browser calls `/approval` , then an attacker can fetch the Dex OAuth code which can be exchanged for an ID token using the `/token` endpoint.

Note that PKCE does not defend against this attack since the webserver initiates the request to Dex with a known code challenge.

Fix

The request has been made unpredictable with message authentication. This was accomplished by creating an HMAC using a randomly generated per-request secret. This secret is persisted between the initial login request and the approval request. Since the HMAC is derived using a secret key, its value cannot be known to an attacker, so they will be unable to poll `/approval` for the code.

Patches

Update to 2.35.0.

Workarounds

No known workarounds (without impacting behavior) for existing versions.

Disabling public clients is the only way to defend against attacks exploiting this vulnerability.

References

For more information

If you have any questions or comments about this advisory:

- Start a new discussion
- Email us at cncf-dex-maintainers@lists.cncf.io

Severity

Critical 9.3 / 10

CVSS base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	None
User interaction	Required
Scope	Changed
Confidentiality	High
Integrity	High
Availability	None

CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:H/A:N

CVE ID

CVE-2022-39222

Weaknesses

No CWEs

Credits

joernchen

bobcallaway

haydentherapper