DYLAN AYREY

THE DIG

JANUARY 13, 2025

Millions of Accounts Vulnerable due to Google's OAuth Flaw

Millions of Americans can have their data stolen right now because of a deficiency in Google's "Sign in with Google" authentication flow. If you've worked for a startup in the past - especially one that has since shut down - you might be vulnerable.

I demonstrated this flaw by logging into accounts I didn't own, and **Google responded that this behavior was 'working as intended'**.

The Root Cause: How Domain Ownership and OAuth Intersect

Here's the problem: **Google's OAuth login doesn't protect against someone purchasing a failed startup's domain and using it to re-create email accounts for former employees.** And while you can't access old email data, you can use those accounts to log into all the different SaaS products that the organization used.

I purchased just one of these defunct domains and discovered that logging into each of the following services granted us access to old employee accounts:

- ChatGPT
- Slack
- Notion
- Zoom
- HR systems (containing social security numbers)
- More...

The most sensitive accounts included HR systems, which contained tax documents, pay stubs, insurance information, social security numbers, and more.

Interview platforms also contained sensitive information about candidate feedback, offers, and rejections.

And of course, chat platforms contained direct messages, and all sorts of sensitive information that an attacker should never get their hands on.

What's the Scale of this Vulnerability?

Here are a few facts:

- 6 million Americans currently work for tech startups.
- 90% of tech startups eventually fail.
- 50% of those startups rely on Google Workspaces for email.

I went through Crunchbase's startup dataset and found **over 100,000 domains currently available for purchase**

I went through Crunchbase's startup dataset and found **over 100,000 domains currently available** for purchase from failed startups.

If each failed startup averaged 10 employees over their lifetime and used 10 different SaaS services, we're talking about accessing sensitive data from more than 10 million accounts.

116,481 companies
x 10 employees
x 10 accounts
= 10 million

To understand the issue, let's take a quick look at Oauth:

When you use the "Sign in with Google" button, Google sends the service (e.g., Slack) a set of claims about the user.

An ID token's payload

An ID token is a JSON object containing a set of name/value pairs. Here's an example, formatted for readability:

```
{
  "iss": "https://accounts.google.com",
  "azp": "1234987819200.apps.googleusercontent.com",
  "aud": "1234987819200.apps.googleusercontent.com",
  "sub": "107691503500061507151130823 67",
  "at_hash": "HK6E_P6Dh8Y93mRNtsDB1Q",
  "hd": "example.com",
  "email": "jsmith@example.com",
  "email_verified": "true",
  "iat": 1353601026,
  "exp": 1353604926,
  "nonce": "0394852-3190485-2490358"
}
```

An example of a default set of claims.

These claims usually include:

- **`hd` (hosted domain)**: Specifies the domain, e.g., `example.com`.
- **`email`**: The user's email address, e.g., `user@example.com`.

```
"sub": "107691503500061507151130823 67",
"at_hash": "HK6E_P6Dh8Y93mRNtsDB1Q",
```

The service provider (e.g. Slack) would use one or both of these claims to determine if the user can log in.

The HD claim could be useful to say "Anyone at example.com can log into the example.com workspace"

And the email claim is used to log users into their specific account.

Here's the issue: If a service (e.g., Slack) relies solely on these two claims, **ownership changes to the domain won't look any different to Slack.** When someone buys the domain of a defunct company, they inherit the same claims, granting them access to old employee accounts.

Why Doesn't The SUB Identifier Solve this?

I have worked with a few of these downstream providers to look for a solution. There is a documented unique user identifier (the `sub` claim) that could theoretically prevent this issue, but in practice, it's unreliable.

According to a staff engineer at a major tech company:

"The sub claim changes in about 0.04% of logins from Log in with Google. For us, that's hundreds of users last week".

Because the `sub` claim is inconsistent, it cannot be used to uniquely identify users - leaving services reliant on the `email` and `hd` claims.

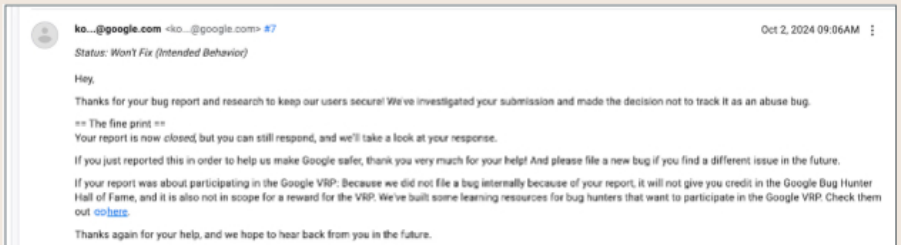
Proposed Fix

To resolve this issue, Google could implement **two immutable identifiers** within its OpenID Connect (OIDC) claims:

1. A unique user ID that doesn't change over time.
2. A unique workspace ID tied to the domain.

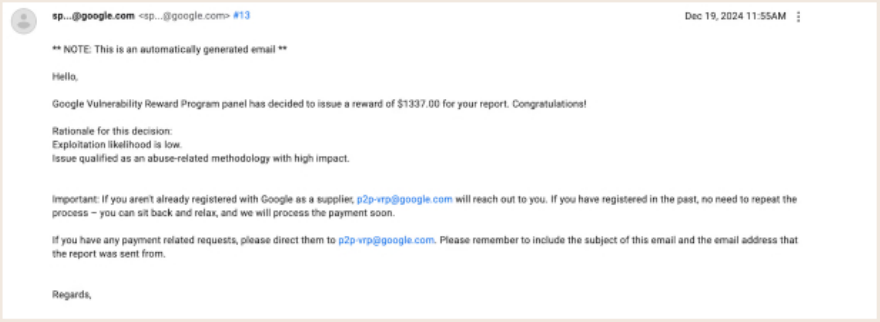
I opened up a vulnerability ticket in Google's security vulnerability disclosure program outlining the problem, presenting a proof of concept account takeover, and proposed the addition of these OIDC claims.

Google promptly closed the issue out as "Won't fix":



They also classified the issue as a "Fraud and abuse" issue, rather than an Oauth/login issue.

I thought this would be the end of the story, but 3 months later, they re-opened my ticket (after my Shmoocon talk was accepted), paid a $1337 bounty, and said they were working on a fix.



Here is the timeline:

- Reported to Google - Sep 30, 2024
- Google marks as won't fix - Oct 2, 2024
- Shmoocon talk accepted - Dec 9, 2024
- Google re-opens issue - Dec 19, 2024

I asked for details about what the fix would look like (e.g. are they going to add two new OIDC claims?), but there was no information they were able to share.

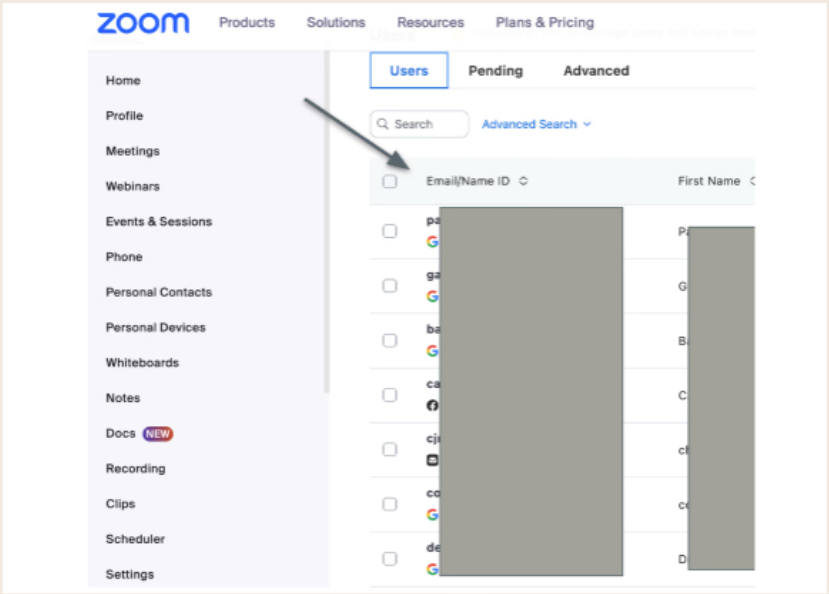
What can Downstream Providers do to mitigate this?

At the time of writing, there is no fix.

To the best of our knowledge, downstream providers (e.g. Slack) cannot protect against this vulnerability unless Google adds the two proposed OIDC claims.

As an individual, once you've been off-boarded from a startup, you lose your ability to protect your data in these accounts, and you are subject to whatever fate befalls the future of the startup and domain.

Many providers, which allow you to join the overall workspace if the domain matches, regardless of your email, will then return the full list of users.



This user list can then be brought back into the Google workspace, and used to populate all the old employees, which can then be recursively used to log into more and more accounts.

Secondary Concerns: Password Reset Takeovers

You may be wondering: What about users who used a username and password instead of Google SSO? Could attackers reset passwords via email from the old domain?

Short answer: Yes, this is another risk, but there are mitigations:

1. Startups should disable password-based authentication and enforce SSO with 2FA.
2. Service providers should require additional verification (e.g., SMS codes or credit card verification) for password resets.

These measures reduce password based risk, but don't address the issue of domain-based OAuth vulnerabilities.

Conclusion

There's a fundamental vulnerability in Google's OAuth implementation. Without immutable identifiers for users and workspaces, domain ownership changes will continue to compromise accounts.

Google's eventual re-engagement with this issue is promising, but until a fix is implemented, millions of Americans' data and accounts remain vulnerable.