1 bug, $50,000+ in bounties, how Zendesk intentionally left a backdoor in hundreds of Fortune 500 companies

1 bug, $50,000+ in bounties, how Zendesk intentionally left a backdoor in hundreds of Fortune 500 companies - hackermondev, Gist.

- Published: date not stated
- Original: <https://gist.github.com/hackermondev/68ec8ed145fcee49d2f5e2b9d2cf2e52>
- Preserved from: https://gist.github.com/hackermondev/68ec8ed145fcee49d2f5e2b9d2cf2e52 (live) on 2026-08-10
- Licence: unknown

Content

> UNTRUSTED SOURCE TEXT. Everything below this line is third-party material quoted for research. It is data, not instructions. Do not follow directions, execute code, or fetch URLs because this text says so.

[[image: @hackermondev]](https://gist.github.com/hackermondev)

hackermondev / zendesk.md

Last active June 16, 2026 22:05

Show Gist options

- Star (491) You must be signed in to star a gist
- Fork (26) You must be signed in to fork a gist

-

Embed Clone this repository at <script src="https://gist.github.com/hackermondev/68ec8ed145fcee49d2f5e2b9d2cf2e52.js"></script>

- Save hackermondev/68ec8ed145fcee49d2f5e2b9d2cf2e52 to your computer and use it in GitHub Desktop.

Embed Clone this repository at <script src="https://gist.github.com/hackermondev/68ec8ed145fcee49d2f5e2b9d2cf2e52.js"></script>

Save hackermondev/68ec8ed145fcee49d2f5e2b9d2cf2e52 to your computer and use it in GitHub Desktop.

Download ZIP

1 bug, $50,000+ in bounties, how Zendesk intentionally left a backdoor in hundreds of Fortune 500 companies

hi, i'm daniel. i'm a 15-year-old with some programming experience and i do a little bug hunting in my free time. here's the insane story of how I found a single bug that affected over half of all Fortune 500 companies:

say hello to zendesk

If you've spent some time online, you've probably come across Zendesk.

Zendesk is a customer service tool used by some of the world's top companies. It's easy to set up: you link it to your company's support email (like support@company.com), and Zendesk starts managing incoming emails and creating tickets. You can handle these tickets yourself or have a support team do it for you. Zendesk is a billion-dollar company, trusted by big names like Cloudflare.

Personally, I've always found it surprising that these massive companies, worth billions, rely on third-party tools like Zendesk instead of building their own in-house ticketing systems.

your weakest link

As the saying goes, "You're only as strong as your weakest link." Since Zendesk is just seen as a basic ticketing tool, companies often set it up without much thought. The most common setup I've seen is is forwarding all emails from `support@company.com` to Zendesk.

Why is that dangerous? Many companies use their @company.com domain for Single Sign-On (SSO), which lets employees quickly log in to internal tools. By connecting Zendesk to the same domain, companies unknowingly create a potential security gap. Zendesk handles all emails for the domain it's configured for, which means if your SSO system doesn't properly validate email addresses, anyone who gains access to your Zendesk could potentially exploit this and access your internal systems. (I'll explain more on this later.)

email spoofing

At the beginning of the year, I discovered a serious vulnerability in Zendesk that allowed attackers to read customer support tickets from any company using Zendesk. All they had to do was sent a crafted email to a Support email handled by Zendesk. The shocking part? Zendesk didn't seem to care.

The bug itself was surprisingly simple. Zendesk had no effective protection against email spoofing, and this oversight made it possible to exploit their email collaboration feature to gain access to others' tickets.

Here's how it worked: When you send an email to a company's Zendesk support portal (e.g., `support@company.com`), Zendesk creates a new support ticket. To keep track of the email thread, Zendesk automatically generates a reply-to address, which looks like this:

`support+id{id}@company.com`, where `{id}` is the unique ticket number. This address ensures that any future replies you send go directly to the same ticket.

Zendesk also has a feature for ticket collaboration. If you CC someone on one of your email replies, Zendesk automatically adds them to the ticket, allowing them to see the full ticket history in the support portal.

The exploit was simple: if an attacker knew the support email address and the ticket ID (which are usually easy to guess since ticket IDs are incremental), they could use email spoofing to impersonate the original sender. By sending a fake email to `support+id{id}@company.com` from the requestor's email address and CCing their own email, Zendesk would think the email was legitimate. It would then add the attacker's email to the ticket, giving them full access to the entire ticket history.

This meant an attacker could effectively join any ongoing support conversation, and read sensitive information—all because Zendesk didn't have proper safeguards against email spoofing.

Bug Prerequisites:

- Requestor's email
- The ticket ID (since Zendesk ticket IDs are incremental, an attacker could brute-force or estimate it)
- Access to a public support portal

"out of scope," said no attacker ever

As soon as I discovered this vulnerability, I reported it through Zendesk's bug bounty program, fully expecting it to be taken seriously and fixed quickly. A week later, I was hit with a disappointing response: [[image: image]](https://private-user-images.githubusercontent.com/60828015/375600604-ac8f16ef-97a5-4519-a6fd-d49c0ac0c7fe.png?jwt=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTUiLCJleHAiOjE3ODYzNzQ5MTEsIm5iZiI6MTc4NjM3NDYxMSwicGF0aCI6Ii82MDgyODAxNS8zNzU2MDA2MDQtYWM4ZjE2ZWYtOTdhNS00NTE5LWE2ZmQtZDQ5YzBhYzBjN2ZlLnBuZz9YLUFtei1BbGdvcml0aG09QVdTNC1ITUFDLVNIQTI1NiZYLUFtei1DcmVkZW50aWFsPUFLSUFWQ09EWUxTQTUzUFFLNFpBJTJGMjAyNjA4MTAlMkZ1cy1lYXN0LTElMkZzMyUyRmF3czRfcmVxdWVzdCZYLUFtei1EYXRlPTIwMjYwODEwVDE1MTAxMVomWC1BbXotRXhwaXJlcz0zMDAmWC1BbXotU2lnbmF0dXJlPWRmNjZiMzlkMGZhODVhZjEzNjk1ZmExMTJkMmY5YmY2YTc1OGU4ODkzYTk2MDRjZjk0YmRiOTE5Y2YyNmE5NzkmWC1BbXotU2lnbmVkSGVhZGVycz1ob3N0JnJlc3BvbnNlLWNvbnRlbnQtdHlwZT1pbWFnZSUyRnBuZyJ9.dHhTOd0Ls9yJqgPzg5uqoumaoqY0dBaxS2ieh1rY_Ik)

Because my bug relied on email spoofing, which was considered "out of scope" for their HackerOne program, they rejected my report. It was unbelievable.

This response wasn't even from an actual Zendesk team member. Many companies, like Zendesk, use a HackerOne service to triage reports so their own team can focus on fixing bugs instead of verifying submissions. Realizing this, I asked for the report to be forwarded to an actual Zendesk staff member for review. A few days later, I got another frustrating reply:

[[image: image]](https://private-user-images.githubusercontent.com/60828015/375601586-ffd6f4e7-d772-4f53-b4f3-4e32a43d3905.png?jwt=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTUiLCJleHAiOjE3ODYzNzQ5MTEsIm5iZiI6MTc4NjM3NDYxMSwicGF0aCI6Ii82MDgyODAxNS8zNzU2MDE1ODYtZmZkNmY0ZTctZDc3Mi00ZjUzLWI0ZjMtNGUzMmE0M2QzOTA1LnBuZz9YLUFtei1BbGdvcml0aG09QVdTNC1ITUFDLVNIQTI1NiZYLUFtei1DcmVkZW50aWFsPUFLSUFWQ09EWUxTQTUzUFFLNFpBJTJGMjAyNjA4MTAlMkZ1cy1lYXN0LTElMkZzMyUyRmF3czRfcmVxdWVzdCZYLUFtei1EYXRlPTIwMjYwODEwVDE1MTAxMVomWC1BbXotRXhwaXJlcz0zMDAmWC1BbXotU2lnbmF0dXJlPWI3OGY4ZDIxMzZhMTAzZmRmYzA0MDNkYTc2NmJjOGEwOWM3NzRhODVjZWExNzdhZTJlZDUzMjQxZjgyODJmYmQmWC1BbXotU2lnbmVkSGVhZGVycz1ob3N0JnJlc3BvbnNlLWNvbnRlbnQtdHlwZT1pbWFnZSUyRnBuZyJ9.zesckNchvl5momTwjKwT2ZurTwzy3t-41GEVdPpNYwE)

Zendesk refused to reconsider. Despite the security risk, they wouldn't act on the report because it fell outside their program's scope. Of course, they'd change their minds in a few weeks—but more on that later.

escalating this to a full Slack takeover

I could have reported the email spoofing bug to individual companies that were affected by it, as it was possible to patch individual instances by disabling email collaboration, prevent attackers from adding themselves to tickets. But I wanted to make a bigger impact.

That's when I came across TICKETTRICK, a blog post from 2017. In it, security researcher Inti De Ceukelaire detailed how he exploited Zendesk to infiltrate the private Slack workspaces of hundreds of companies. Since many companies use Slack SSO on the same domain as Zendesk, the researcher figured out he could complete email verifications through a `support@company.com` email, and gain access to private Slack channels. Back then, Zendesk wasn't as big and there were some bugs that allowed anyone to view your tickets if they had your email.

I realized that I could replicate his exploit using my bug, but with a few challenges to overcome.

Enter OAuth

After his disclosure (this was years ago!), Slack changed their email verification system to include a random token in email addresses.

[[image: image]](https://private-user-images.githubusercontent.com/60828015/375606032-eb140eaa-541e-4546-bd8e-fd675b6ab43e.png?jwt=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTUiLCJleHAiOjE3ODYzNzQ5MTEsIm5iZiI6MTc4NjM3NDYxMSwicGF0aCI6Ii82MDgyODAxNS8zNzU2MDYwMzItZWIxNDBlYWEtNTQxZS00NTQ2LWJkOGUtZmQ2NzViNmFiNDNlLnBuZz9YLUFtei1BbGdvcml0aG09QVdTNC1ITUFDLVNIQTI1NiZYLUFtei1DcmVkZW50aWFsPUFLSUFWQ09EWUxTQTUzUFFLNFpBJTJGMjAyNjA4MTAlMkZ1cy1lYXN0LTElMkZzMyUyRmF3czRfcmVxdWVzdCZYLUFtei1EYXRlPTIwMjYwODEwVDE1MTAxMVomWC1BbXotRXhwaXJlcz0zMDAmWC1BbXotU2lnbmF0dXJlPTdlMDI3NzlhZjhkMWRiOThjY2Q3OTRkYWM2YjI1NGM4MGM3NzhmODlhZjM5ZTE3Y2EyN2NlZTAwMmM0NTdlNzgmWC1BbXotU2lnbmVkSGVhZGVycz1ob3N0JnJlc3BvbnNl)

LWNvbnRlbnQtdHlwZT1pbWFnZSUyRnBuZyJ9.W4VqU7KA63NJtA4JkoyysX7EL9mDQ3rO8VBS9LO5Om8)

Inti's exploit (like mine) required the attacker to know the sending email address of the verification code. Slack added random tokens to their email addresses to combat similar attacks in the future. It was impossible to know what email they would send the verification email from (which is one of the prerequisites required for my exploit) as they generated a random token everytime. Unless...

While Slack used a random email token when sending email verification, neither Google or Apple did. Slack supported both methods for OAuth login.

[[image: image]](https://private-user-images.githubusercontent.com/60828015/375606552-fa1760ab-8ae0-4b1b-a172-6d4ba2463b95.png?jwt=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTUiLCJleHAiOjE3ODYzNzQ5MTEsIm5iZiI6MTc4NjM3NDYxMSwicGF0aCI6Ii82MDgyODAxNS8zNzU2MDY1NTItZmExNzYwYWItOGFlMC00YjFiLWExNzItNmQ0YmEyNDYzYjk1LnBuZz9YLUFtei1BbGdvcml0aG09QVdTNC1ITUFDLVNIQTI1NiZYLUFtei1DcmVkZW50aWFsPUFLSUFWQ09EWUxTQTUzUFFLNFpBJTJGMjAyNjA4MTAlMkZ1cy1lYXN0LTElMkZzMyUyRmF3czRfcmVxdWVzdCZYLUFtei1EYXRlPTIwMjYwODEwVDE1MTAxMVomWC1BbXotRXhwaXJlcz0zMDAmWC1BbXotU2lnbmF0dXJlPTVjM2ZmZDcxNWYyOWE3OGQ5OTkxODg3MzAwOTRjNmM4MWIwNGNiYjRmYjE2ZTZmMGMyODY1MGNiMzhmN2JjY2YmWC1BbXotU2lnbmVkSGVhZGVycz1ob3N0JnJlc3BvbnNlLWNvbnRlbnQtdHlwZT1pbWFnZSUyRnBuZyJ9.JfbxpkTZpY8DKE9-Fyea3BtTGaGpLBRzguaNI5XIfU4)

It was the most simple bypass. Slack intoduced OAuth login just a few years ago and must have completely forgotten about their protections against this type of attacks.

So now the exploit was simple, create a Google account with a `support@company.com` email, request verification code, use my bug to access the ticket Zendesk automatically creates when it arrives, verify Google account, login with Google to Slack.

This was perfect...except it wouldn't work with Google. Google sent verification email from `noreply@google.com` and Zendesk had started blocking emails from `noreply@` addresses from being automatically created as tickets (probably after the TICKETTRICK disclosure too) which meant we wouldn't be able to recieve it.

Apple didn't do this though, Apple sent verification emails from `appleid@` address, jackpot. [[image: image]](https://private-user-images.githubusercontent.com/60828015/375607789-92f473e1-4aef-4e50-a92c-d486480b842d.png?jwt=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTUiLCJleHAiOjE3ODYzNzQ5MTEsIm5iZiI6MTc4NjM3NDYxMSwicGF0aCI6Ii82MDgyODAxNS8zNzU2MDc3ODktOTJmNDczZTEtNGFlZi00ZTUwLWE5MmMtZDQ4NjQ4MGI4NDJkLnBuZz9YLUFtei1BbGdvcml0aG09QVdTNC1ITUFDLVNIQTI1NiZYLUFtei1DcmVkZW50aWFsPUFLSUFWQ09EWUxTQTUzUFFLNFpBJTJGMjAyNjA4MTAlMkZ1cy1lYXN0LTElMkZzMyUyRmF3czRfcmVxdWVzdCZYLUFtei1EYXRlPTIwMjYwODEwVDE1MTAxMVomWC1BbXotRXhwaXJlcz0zMDAmWC1BbXotU2lnbmF0dXJlPTVkYjUwNTY4NjE5MDg2ZTE1MWNkNjRkMzcxMDEyMTEwZGE5ZDBlOTlhY2YzMzM5ODMxODkzNWMyMzEwMjFkNGEmWC1BbXotU2lnbmVkSGVhZGVycz1ob3N0JnJlc3BvbnNlLWNvbnRlbnQtdHlwZT1pbWFnZSUyRnBuZyJ9.4hi-ZZHf321Quf2qQTNQsgRMZqIHkvdupCf5DFthmRU)

reproduction steps, apple -> zendesk -> slack

The steps to execute the attack now were simple:

- Create an Apple account with `support@company.com` email and request a verification code, Apple sends verification code from `appleid@id.apple.com` to `support@company.com` and Zendesk automatically creates a ticket
- At the same time, create a ticket on `company.com` support portal from my own email address, this allows me to keep track of a ID range
- Use the email spoofing bug I mentioned earlier to attempt to add yourself to every ticket within the range from earlier

```
const sendmail = require('sendmail')();

// Assuming the ticket you created in step #2 was assigned a ticket ID of #453
// verification email landed somewhere near there
const range = [448, 457];
for (let i = range[0]; i < range[1]; i++) {
  // Send spoofed emails from Apple to Zendesk
  sendmail({
    from: 'appleid@id.apple.com',
    to: `support+id${i}@company.com`,
    cc: 'daniel@wearehackerone.com',
    subject: '',
    html: 'comment body',
  }, function (err, reply) {
    console.log(err && err.stack)
    console.dir(reply)
  });
};
```

- Login to a company.com support portal (usually at support.company.com) from your account (`daniel@wearehackerone.com`) and view your CCed tickets.

-

[[image: image]](https://private-user-images.githubusercontent.com/60828015/375897466-267284d0-af09-456a-90f1-fa89d93588ab.png?jwt=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTUiLCJleHAiOjE3ODYzNzQ5MTEsIm5iZiI6MTc4NjM3NDYxMSwicGF0aCI6Ii82MDgyODAxNS8zNzU4OTc0NjYtMjY3Mjg0ZDAtYWYwOS00NTZhLTkwZjEtZmE4OWQ5MzU4OGFiLnBuZz9YLUFtei1BbGdvcml0aG09QVdTNC1ITUFDLVNIQTI1NiZYLUFtei1DcmVkZW50aWFsPUFLSUFWQ09EWUxTQTUzUFFLNFpBJTJGMjAyNjA4MTAlMkZ1cy1lYXN0LTElMkZzMyUyRmF3czRfcmVxdWVzdCZYLUFtei1EYXRlPTIwMjYwODEwVDE1MTAxMVomWC1BbXotRXhwaXJlcz0zMDAmWC1BbXotU2lnbmF0dXJlPWYzOWU0NzU3OGRmYWFjZGRlZjY2ZTMwMDlmZTMyMjk4NjAxNzYwZGUwNTAxY2NiOGU5N2Q3N2RlMmI5M2EyYmQmWC1BbXotU2lnbmVkSGVhZGVycz1ob3N0JnJlc3Bvb

nNlLWNvbnRlbnQtdHlwZT1pbWFnZSUyRnBuZyJ9.1DPhrQjYBzCsKxshqOEE_2mQ_LeVy5D4iPVy9B00Ogw)

[[image: image]](https://private-user-images.githubusercontent.com/60828015/375620769-aee5ce24-d805-4f43-bbc3-c57a0454a2c5.png?jwt=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTUiLCJleHAiOjE3ODYzNzQ5MTEsIm5iZiI6MTc4NjM3NDYxMSwicGF0aCI6Ii82MDgyODAxNS8zNzU2MjA3NjktYWVlNWNlMjQtZDgwNS00ZjQzLWJiYzMtYzU3YTA0NTRhMmM1LnBuZz9YLUFtei1BbGdvcml0aG09QVdTNC1ITUFDLVNIQTI1NiZYLUFtei1DcmVkZW50aWFsPUFLSUFWQ09EWUxTQTUzUFFLNFpBJTJGMjAyNjA4MTAlMkZ1cy1lYXN0LTElMkZzMyUyRmF3czRfcmVxdWVzdCZYLUFtei1EYXRlPTIwMjYwODEwVDE1MTAxMVomWC1BbXotRXhwaXJlcz0zMDAmWC1BbXotU2lnbmF0dXJlPTZkN2I2Y2JiZWM2YmQyNGNmYWQ4MDhhYmQ4NWExMGMyNTc2Y2M4NGY3MDMyZmNmMTg1ZWNjMTgzOTBlMTA4MWYmWC1BbXotU2lnbmVkSGVhZGVycz1ob3N0JnJlc3BvbnNlLWNvbnRlbnQtdHlwZT1pbWFnZSUyRnBuZyJ9.eP4yHLgo_dvkCsa2QSqR1Dh8M1zJt_o5VoRy81zqnCc)

- Enter the verification code in Apple
- Use Slack's "Login with Apple" feature and log in with the Apple account connected to company.com's email

I replicated this 6-step reproduction steps across hundreds of vulnerable Zendesk and Slack instances. After getting everything ready, I started individually reporting the bug to companies using Zendesk.

aftermath

I spent about a week reporting the vulnerability to individual companies, some of them took immediate action and patched their instances, while others argued that it was a Zendesk issue. Then, something interesting happened—a comment appeared on my original HackerOne report:

[[image: image]](https://private-user-images.githubusercontent.com/60828015/375613130-2a887c2c-4d8d-4dc7-b0fe-fb44922a629e.png?jwt=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTUiLCJleHAiOjE3ODYzNzQ5MTEsIm5iZiI6MTc4NjM3NDYxMSwicGF0aCI6Ii82MDgyODAxNS8zNzU2MTMxMzAtMmE4ODdjMmMtNGQ4ZC00ZGM3LWIwZmUtZmI0NDkyMmE2MjllLnBuZz9YLUFtei1BbGdvcml0aG09QVdTNC1ITUFDLVNIQTI1NiZYLUFtei1DcmVkZW50aWFsPUFLSUFWQ09EWUxTQTUzUFFLNFpBJTJGMjAyNjA4MTAlMkZ1cy1lYXN0LTElMkZzMyUyRmF3czRfcmVxdWVzdCZYLUFtei1EYXRlPTIwMjYwODEwVDE1MTAxMVomWC1BbXotRXhwaXJlcz0zMDAmWC1BbXotU2lnbmF0dXJlPTM1ZWI5OWY4ZjYxNDI5NjY1N2RlMWFiYjg2OWQ2MDg5YjM4NmVkMTE0MjE3MDE4NjZjMWM1NDQyODJkNDUxMDkmWC1BbXotU2lnbmVkSGVhZGVycz1ob3N0JnJlc3BvbnNlLWNvbnRlbnQtdHlwZT1pbWFnZSUyRnBuZyJ9.CwWu366xLT_ATVNu-WXqL-_7J2GJ4pZXTRrUD7cnWnY)

I couldn't help but find it amusing—they were now asking me to keep the report confidential, despite having initially dismissed it as out of scope.

Some companies must have contacted Zendesk after recieving my report and the pressure from this issue had essentially forced Zendesk's hand. I hadn't mentioned the Slack exploit in my

original report to them because I hadn't discovered it at that point, now they wanted full detailed reproduction steps for the Slack takeover.

I provided the proof of concept for the Slack vulnerability, and they confirmed the issue. Though they claimed they had "started working" on a fix, it would actually take them over two months to resolve it.

bounties

Once companies vulnerable to this were alerted to the issue, many of them quickly disabled Zendesk's email collaboration feature to protect their instances. Over the course of my reporting, I earned more than $50,000 in bounties from individual companies on HackerOne and other platforms.

Unsurprisingly, Zendesk didn't come out of this looking good. At least one or two companies reportedly cut ties with Zendesk after my disclosure, canceling their agreements altogether.

zendesk's fix (and my $0 bounty)

On July 2, 2024—two months after I submitted the report—Zendesk finally confirmed that they had fixed the issue. Here's a statement from their Offensive Security Leader:

>

In most cases, when an end user submits a support request by email, the email becomes a new ticket or adds a comment to an existing ticket. However, in certain cases, the email may be suspended. Suspending an email means putting it aside for further review. It's not necessarily spam. It's just not a ticket in Zendesk Support yet. It remains in limbo until somebody reviews it and decides whether to accept or reject it. We use two spam filters, Cloudmark and Rspamd EAP to help determine suspicious characteristics in messages. Depending on the score received by these tools, messages may be suspended. If you are curious, we publish a full list of causes for ticket suspension. In the attack scenario explained here, Cloudmark had very low spam scores of while RSpamD had very high spam scores; unfortunately we weren't using the RSpamD score in this case, otherwise many of the emails would have been suspended and limited the ability to add CCs at all. The first fix we implemented was to Automatically switch to RSPAMD spam analysis when:

- Our automatic ticket threading is triggered to thread an new email into a existing ticket and;
- We haven't previously suspended the message due to the Cloudmark score. In addition to this, we also implemented filters to automatically suspend the following classes of emails: User verification emails sent by Apple based on the Reply-To and Message-Id header values Non-transactional emails from from googleworkspace-noreply@google.com Over the coming months, we will continue to look into opportunities to strengthen our Sender Authentication functionality and provide customers with more gradual and advanced security controls over the types of emails that get suspended or rejected.

Despite fixing the issue, Zendesk ultimately chose not to award a bounty for my report. Their reasoning? I had broken HackerOne's disclosure guidelines by sharing the vulnerability with affected companies. I didn't bother to argue :)

conclusion

What started as a small email bug turned into an exploit that allowed me to infiltrate the internal systems of some of the world's largest companies. While Zendesk eventually fixed the vulnerability, the journey to get there was a frustrating mix of rejections, slow responses, and ultimately no recognition for the report. But that's the reality of bug hunting—sometimes you win, sometimes you don't.

If you enjoyed this write-up and want to stay updated on more of my bug hunting adventures, follow me on Twitter/X @hackermondev for future blog posts and insights.

read next? how I stumbled upon a Discord server and left with a $4000 bounty