

Open in app ↗

Sign up

Sign in

Medium

 Search Write

Stored XSS with Cloudflare WAF Bypass



Aland Dlshad (HexaPhp)

Follow

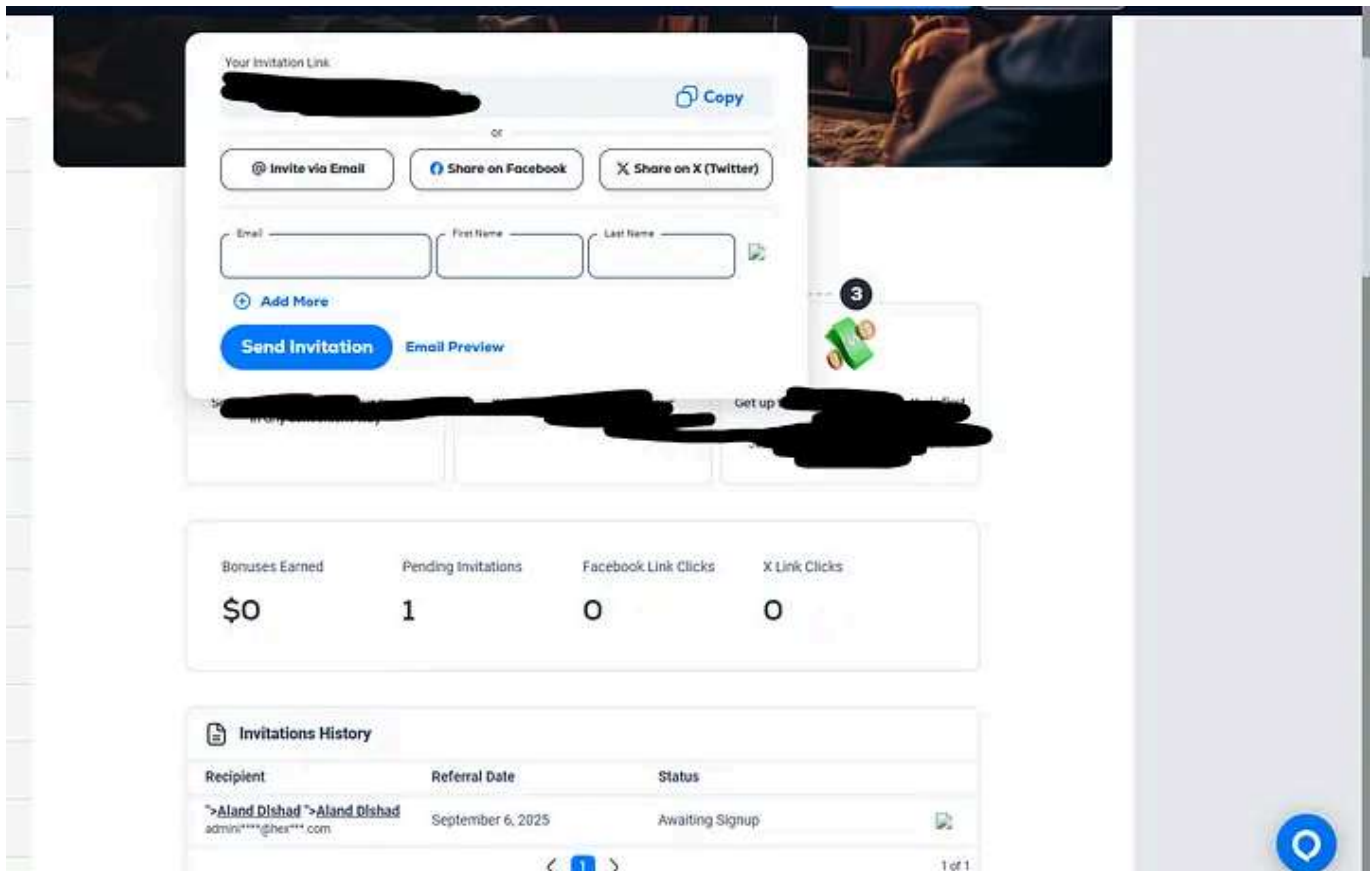
2 min read · Sep 6, 2025



While testing a site protected by **Cloudflare**, I came across some input fields that appeared to be storing user data. Naturally, I wanted to check if the stored data was being rendered unsanitized in the frontend.

So, I started simple — a basic HTML injection to confirm output was being reflected. I submitted:

`<u>Aland Dlshad</u>`



And sure enough, it rendered just like that. ✓

This told me the input wasn't being properly encoded or sanitized before being displayed — definitely a red flag.

Next, I tried a classic stored XSS payload:

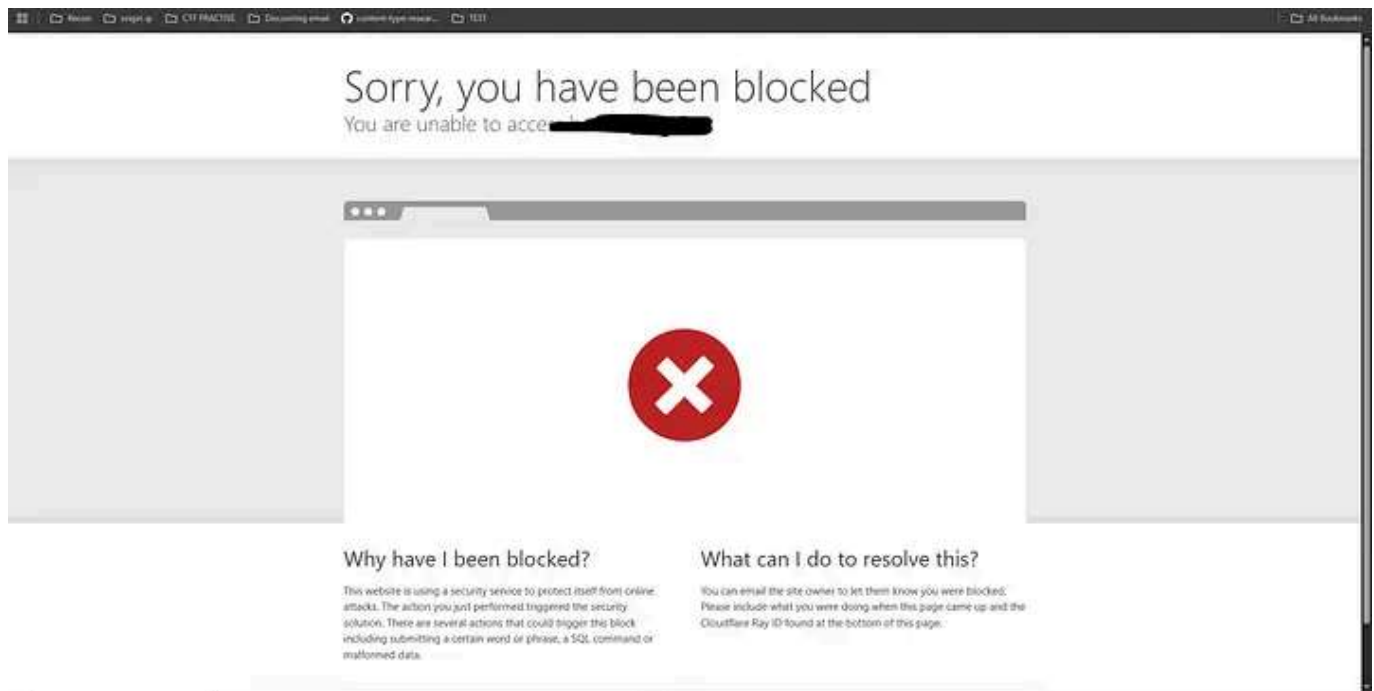
Get Aland Dlshad (HexaPhp)'s stories in your inbox

Join Medium for free to get updates from this writer.

Enter your email

Subscribe

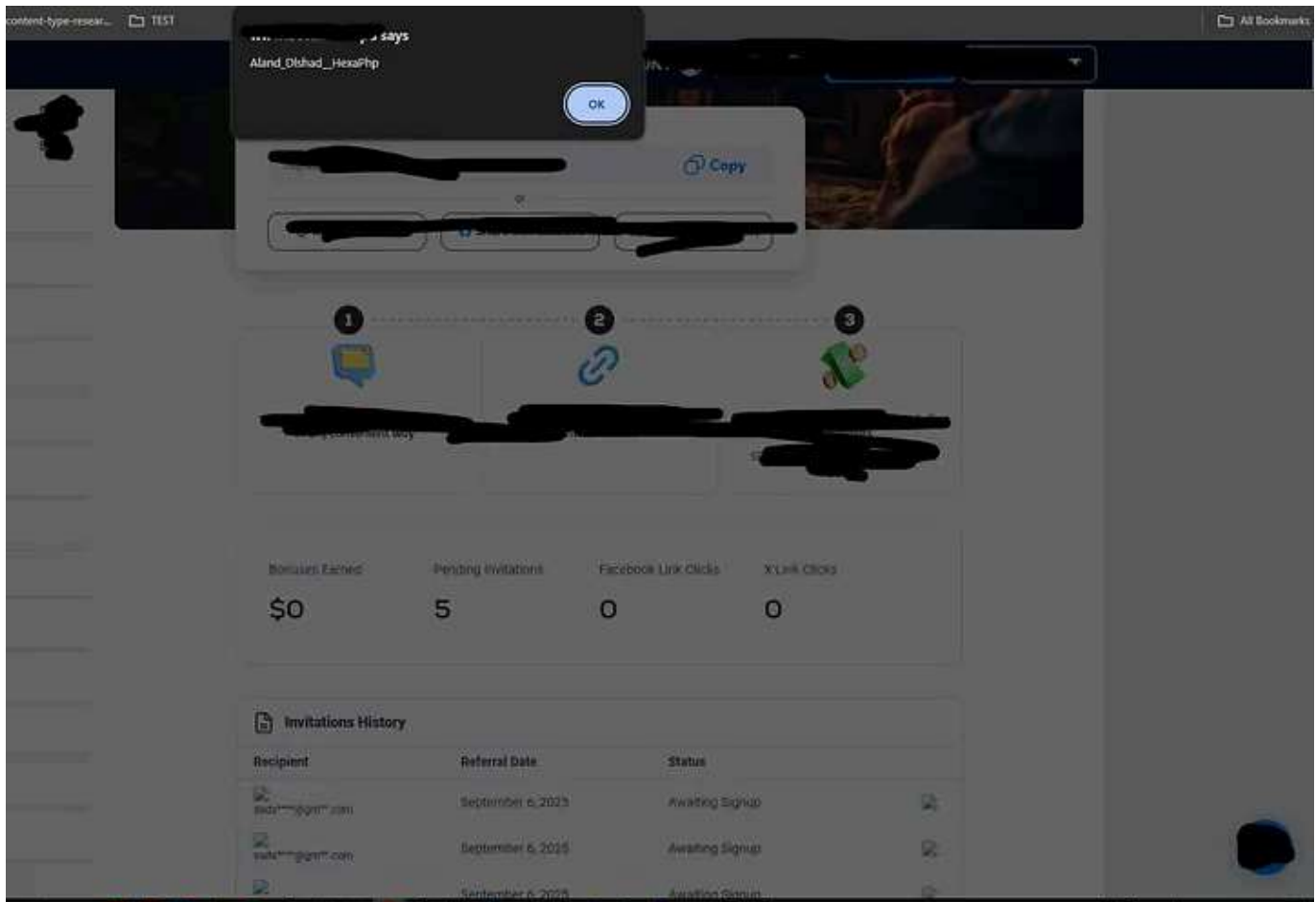
``



But this time, **Cloudflare's WAF blocked the request**, showing a 403 error. So I knew I'd need to get creative if I wanted to bypass it.

After trying a few different obfuscation techniques, I landed on this payload:

```
<img src=OnErRor OnErRor=(alert)("Aland_Dlshad__HexaPhp")>
```



To my surprise — it worked.

No WAF block, and the alert popped clean when the stored input was rendered. ✓

Looks like the mixed casing and duplicated `onerror` attributes confused Cloudflare's WAF, allowing the payload to slip through. The backend didn't sanitize the data, so the browser just executed it.

💡 Takeaway

Cloudflare caught the obvious stuff — but **it doesn't matter how strong your WAF is if your backend blindly trusts user input**. Stored XSS was possible because of poor output encoding, and the WAF was bypassed with minor tweaks.

Cybersecurity

Bug Hunting

Bug Bounty

Bugs

Bugbounty Writeup



Written by Aland Dlashad (HexaPhp)

Follow

28 followers · 127 following

Web Application Penetration Tester | Securing Web Applications | Certified
#EWPTX #EJPT #CAP #CAPEN

Responses (2)



Write a response

What are your thoughts?



Aybora Ünveren

Sep 7



Who would have thought they blindly trust WAF :)

1 reply [Reply](#)

Hacker Write-Ups

Sep 7



If you don't mind.

Bro I used this write-ups for my website where I gather bug bounty and hacking related write-ups.

Website : <https://hacker-writeups.github.io>

If there is any problem tell me I will delete the write-ups.

Thank You for contributing.



1 reply

[Reply](#)

More from Aland Dlashad (HexaPhp)



Aland Dlashad (HexaPhp)

How I Got a Four-Digit Bug Bounty From Grammarly

Sometimes, big bugs hide in the simplest places. While testing Grammarly's document upload feature, I noticed a tiny detail in...



Oct 31



59



1



[See all from Aland Dlashad \(HexaPhp\)](#)

Recommended from Medium



 In InfoSec Write-ups by SIDDHANT SHUKLA

One Click to All Basic Recon for Bug Bounty

All Recon with One Click

🌟 Oct 8 🖱️ 605 💬 5

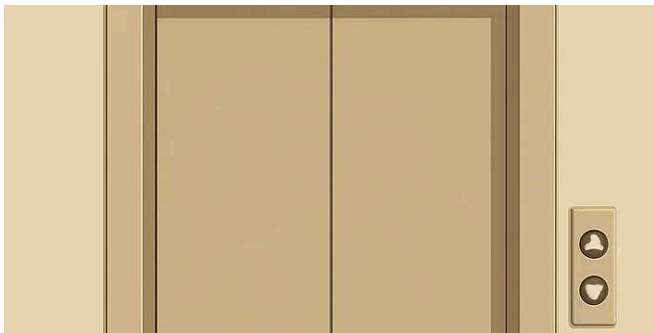


 Very Lazy Tech 🤖

Mastering Subdomain Takeover: Step-by-Step Guide with Real Too...

🌟 Link for the full article in the first comment

🌟 Oct 30 🖱️ 26 💬 2



 Jerry Shah (Jerry)

HTML Injection - Return of the P2 Elevator

Summary

Oct 31 🖱️ 81 💬 1



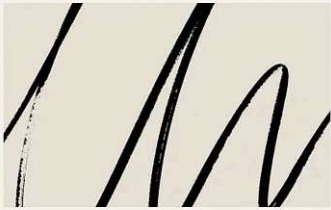
 Abhishek meena

🔥 🌵 Web Cache Poisoning—Part 1: Understanding the Beast

“If you can make the cache remember your payload, you control what everyone else...”

🌟 Oct 29 🖱️ 106



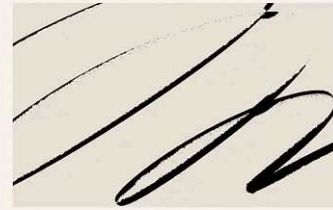


JS In JavaScript in Plain En... by Narendar Battula (n...

I Used usedJS to Find 100+ Vulnerabilities—Here's How

“The JavaScript files told me everything—I just had to listen.”

★ Jul 27 🖱 84 💬 1



 Be nice insabat

How i found account takeover in private bug bounty program of...

email@gmail.com.burpcolab.com Assalam o
alaikum for muslim brothers and hello for no...

★ Oct 30 🖱 95 💬 3



See more recommendations