# **Deanonymizing OpenSea NFT Owners via Cross-Site Search Vulnerability**

Ron Masas, Red Team

## **TLDR**

Recently, a cross-site search vulnerability was discovered affecting the popular NFT marketplace OpenSea. When successfully exploited, this issue allows for the deanonymization of OpenSea users by linking an IP address, a browser session, or an email in certain conditions to a specific non-fungible token (NFT) and therefore a wallet address, potentially revealing a user's identity.

The root cause of this vulnerability was the use of the iFrame-resizer library by OpenSea, but it was due to a misconfiguration of the library rather than an inherent flaw in the library itself. The misconfiguration allowed for the cross-site search vulnerability to exist, leading to the potential exposure of user identities.

After the vulnerability was disclosed, OpenSea quickly released a patch to fix the issue. The patch restricted cross-origin communication, mitigating the risk of further exploitation. The fix was validated by the Imperva Red Team, who confirmed that the vulnerability had been properly addressed.

[embedded video]

<https://drive.google.com/file/d/18TZ6yEESsgSGepmyi_0cZUgcl7ctxcvy/view?usp=share_link>

## Introduction

The world of Web3 and decentralized applications (dApps) is rapidly expanding, bringing with it a host of new possibilities and challenges. As the popularity of dApps grows, so too does the potential for security breaches and vulnerabilities.

Recent years have seen several high-profile hacks and vulnerabilities affecting popular Web3 platforms, highlighting the need for developers to prioritize security and privacy. From the infamous DAO hack on the Ethereum blockchain to more recent hacks targeting cross-chain bridges, it is clear that the security of Web3 applications must be a top priority.

In this blog post, we will dive into the details of the OpenSea vulnerability and discuss the importance of proper cross-origin communication restrictions in protecting user anonymity. We will also explore the potential dangers of cross-site search attacks, and the need for Web3 developers to stay vigilant in ensuring the safety and security of their platforms.

## What is a cross-site search vulnerability?

[embedded video]

<https://drive.google.com/file/d/1jUYhCF6itqUUK6eWbEls4iPBAQmTHLgg/view?usp=share_link>

[Cross-site search](https://xsleaks.dev/docs/attacks/xs-search/) (XS-Search) is a vulnerability in web applications that use query-based search systems. It allows an attacker to extract sensitive information from a different origin by sending queries and observing differences in the behavior of the search system when it returns or doesn't return results. The attacker incrementally gathers information by sending multiple queries, using the distinguishable differences in the behavior of the system to extract more and more information. The XS-Leaks family of attacks was built upon the principles of XS-Search, using the similar underlying method to extract sensitive information from a web application in a more generic way.

## The iFrame-resizer library and the potential for cross-site search

The iFrame-resizer library is used to automatically resize iFrames to fit their content. This is useful in situations where an iFrame is embedded on a page and the content within the iFrame is dynamic and can change size. Without the iFrame-resizer library, the iFrame would not resize to fit its content, potentially leading to a poor user experience.

However, when the iFrame-resizer library is used in a situation where cross-origin communication is not restricted, it can lead to a cross-site search vulnerability. This is because the library broadcasts the width and height of the iFrame, which can be used as an oracle to detect when a given search query returns results. An attacker can exploit this vulnerability by continuously searching the victim's assets, which is done cross-origin, to leak an NFT name and associated wallet address. This can lead to the deanonymization of the user if the attacker is able to associate the leaked information with the user's identity.

In addition to the risk of cross-site search vulnerabilities, the iFrame-resizer library can also be exploited to leak sensitive information in other ways. For example, an attacker could use the library to leak parts of the URL of a cross-origin window, potentially exposing sensitive information such as authentication tokens or other sensitive data. As such, it is important to carefully consider the potential risks of using the iFrame-resizer library without origin restrictions and take steps to mitigate these risks.

## Vulnerability overview

OpenSea did not restrict cross-origin communication, making it possible for attackers to exploit this vulnerability through cross-site search attacks. The iFrame-resizer library broadcasts the width and height of the page, which can be used as an "oracle" to determine when a given search returns results, because the page is smaller when a search returns zero results. By continuously searching the user's assets, which is done cross-origin through the use of a tab or popup, an attacker can leak the name of an NFT that was created by the user, thereby revealing their public wallet address. This information can then be used to associate the user's identity with the leaked NFT and public wallet address.

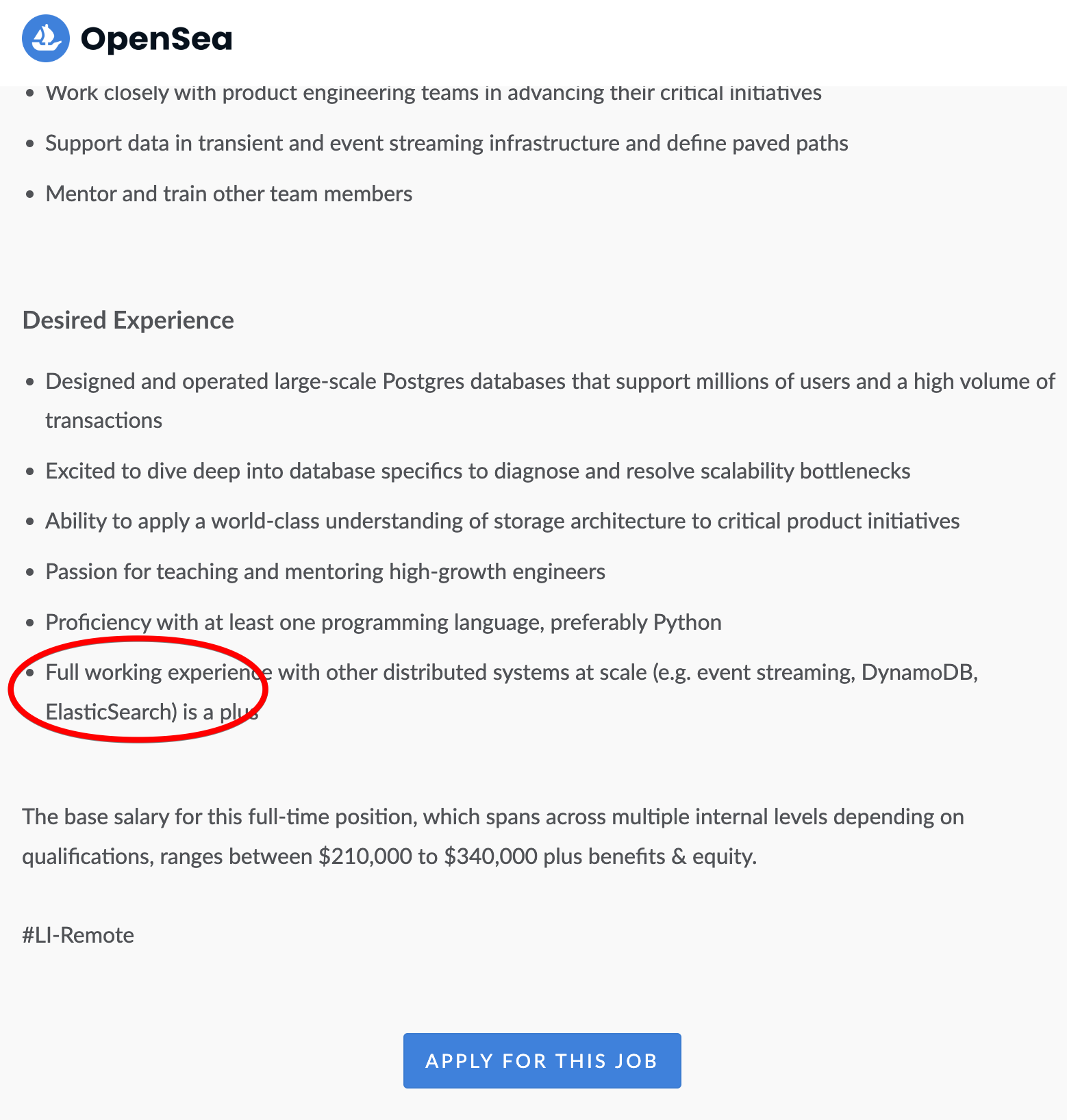
The exploitation steps for this vulnerability are as follows:

* The attacker would send the victim a link through various communication channels such as email or SMS.
* When the victim clicks on the link, it reveals valuable information, such as the victim's IP address, user agent, device details, and software versions.
* The attacker's service would then exploit the cross-site search vulnerability and extract one of the victim's NFT names.
* Finally, the attacker would associate the leaked NFT/public wallet address with the victim's identity, which is the email or phone number that the link was initially sent to.

## Exploit development

Once we confirmed that our basic exploit techniques were working, we began investigating the search function on OpenSea.

We noticed that the company mentioned using ElasticSearch in their job listings, indicating that this is likely the engine they use for their search feature. We then focused on better understanding how search works on the platform in order to improve our exploit.



Understanding ElasticSearch

ElasticSearch is a powerful search engine that can be used to quickly search and analyze large volumes of data. One of the key features of ElasticSearch is its ability to normalize language through the use of language-specific analyzers and stemmers. These algorithms are designed to break down text into individual words and tokens, and to remove inflectional endings such as "-s" or "-ed". This normalization process helps to improve the accuracy and relevance of search results by allowing ElasticSearch to match words with similar meanings, regardless of their inflection.

For example, a search for "run" would also match words like "ran" and "running" because they share the same base form.

We also learned that the order of 2 leaked words can be confirmed by searching for them as one word. For example, if we leaked the words “amazing” “cat” and “hacker” we can confirm the order by searching for “amazingcat”, which only returns if the words’ order is correct.

ElasticSearch is a highly efficient search engine that allows you to search and analyze large volumes of data with ease. One of the key features of ElasticSearch is its ability to normalize language through the use of language-specific analyzers and stemmers. These algorithms are designed to break down text into individual words and tokens, and to remove inflectional endings such as "-s" or "-ed". This normalization process helps to improve the accuracy and relevance of search results by allowing ElasticSearch to match words with similar meanings, regardless of their inflection. For example, a search for "run" would also match words like "ran" and "running" because they share the same base form. This understanding also allows us to anticipate common inflectional endings, such as "s", when we are piecing together leaked information.

This understanding also allows us to anticipate common inflectional endings, such as "s", when we are piecing together leaked NFT names.

iFrame-resizer postMessage communication protocol

Since the iFrame-resizer is a relatively small and simple [open source project](https://github.com/davidjbradshaw/iframe-resizer) we were able to quickly understand the messages we needed to send in order to leak the width and height of the page.

We explored [several modes](https://github.com/davidjbradshaw/iframe-resizer/blob/master/docs/parent_page/options.md#heightcalculationmethod) for computing the page size using iFrame-resizer and used the iFrame-resizer library to generate post message content for each mode until we identified a suitable one for our attack.



## Speeding up the attack

The attack was functional but slow, taking tens of minutes to reveal one word. To expedite the attack, we added a basic word predictor that filtered possible next characters using an English dictionary. This led to a ~10x improvement.

## Hardening recommendations

One of the most effective measures to take is to utilize the [Cross-Origin-Opener-Policy (COOP)](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Cross-Origin-Opener-Policy) header. This header enables website owners to specify whether their pages can be accessed in cross-origin contexts, like when an attacker attempts to open their page within a malicious website.

By setting this header, website owners can prevent attackers from gaining access to cross-site leaks “widgets”, such as a reference to the window object of your website, which is often used in cross-site leaks attacks.

It's important to note that adding the "Cross-Origin-Opener-Policy" header to your website is a defense-in-depth technique and not a complete solution to cross-site leaks.

## OpenSea’s response

In response to this vulnerability, OpenSea quickly took action to address the issue and implement proper cross-origin communication restrictions. Within a few days of the vulnerability being reported, the team at OpenSea had fixed the issue and ensured that their platform was no longer at risk of such attacks.

## Final Thoughts

This vulnerability highlights the dangers of cross-origin communication, which can lead to XS-Leaks and other vulnerabilities. We appreciate OpenSea's prompt response in addressing the security issue and working with us to mitigate it.

Our team is dedicated to identifying and reporting vulnerabilities and collaborating with software providers to improve the safety and security of their platforms, just as we do for our customers.