Hunting down subdomain takeover vulnerabilities

BY BLACKBIRD-EU · APRIL 8, 2025 · LAST UPDATED ON APRIL 9, 2025

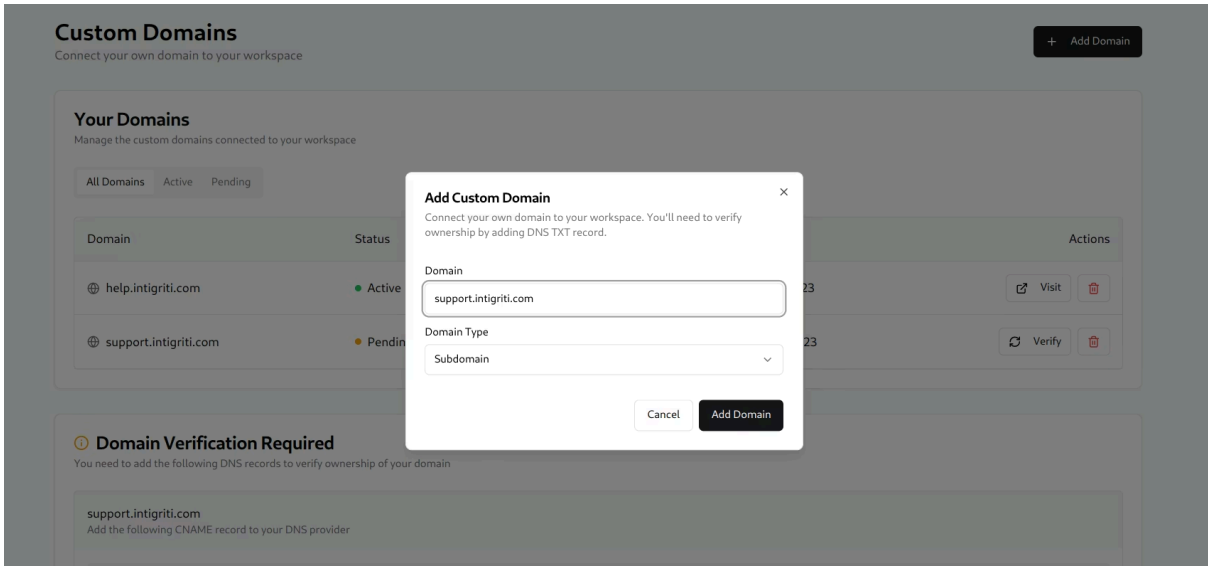
Subdomain takeovers are a well-documented security misconfiguration. Despite widespread awareness, developers still frequently forget to remove DNS records pointing to forgotten and unused third-party services, allowing these vulnerabilities to be present even today.

In this article, we will learn what subdomain takeover vulnerabilities are, we will cover ways on how to identify them (and distinguish non-vulnerable cases) and also document almost all possible exploitation vectors to help you escalate your initial finding.

Let's dive in!

What are subdomain takeover vulnerabilities

Companies often prefer third-party services because they provide multiple benefits: they reduce development time and costs, come from vendors with specialized industry expertise, and continuously improve through regular updates. Vendors adapt to customer needs by offering various personalization and branding options. These customization features range from logo placement to custom styling sheets and themes and even allow companies to point their own subdomains to the third-party service.



Example of a software vendor providing the possibility to connect custom domains

Subdomain takeover vulnerabilities occur when one of the company's subdomains points to a third-party service that no longer exists, allowing an attacker to claim that service and control what appears on the subdomain.

Identifying subdomain takeover vulnerabilities

But before you can claim a lost third-party service, you must meet the following conditions:

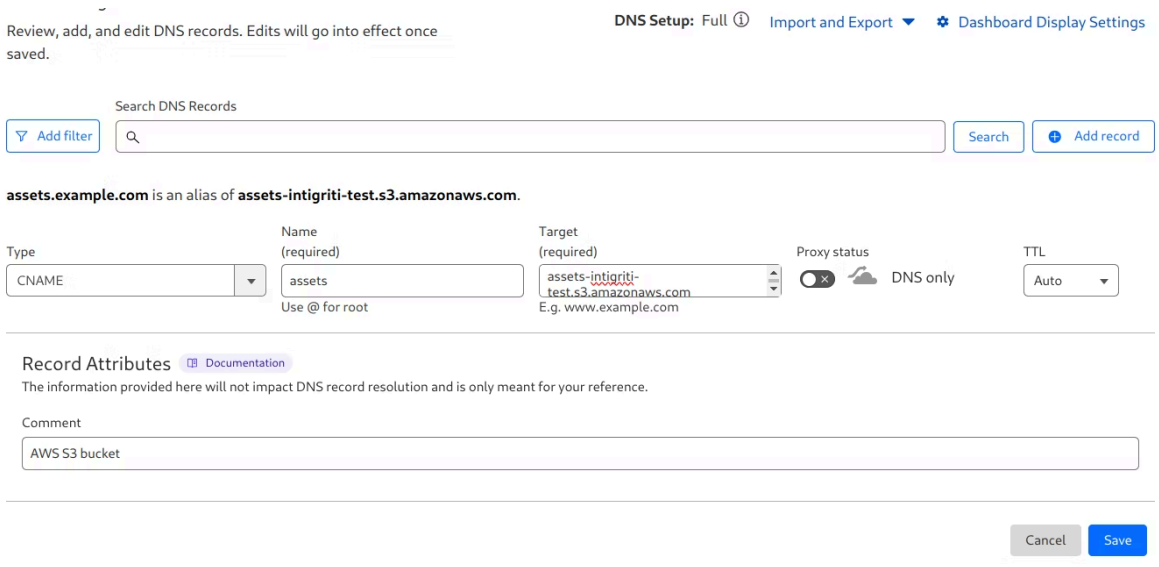
1. The third-party service (such as AWS S3) **must not require domain verification upon registration.**
2. The third-party service should **allow you to select the name when setting up the subdomain**. If it auto-generates a random subdomain name for you that you must add as a CNAME record, a subdomain takeover might not be possible.

Let's take a look at a few examples that clearly outline when a third-party service may allow subdomain takeovers. This will become more relevant when exploiting these issues, as most bug bounty programs do not accept potential or theoretical submissions.

AWS S3 (vulnerable case)

Companies across various industries use AWS S3 buckets to easily store data objects in the cloud. One common application involves using buckets to store publicly accessible content like JavaScript files. Suppose a software engineer at a company creates a new bucket to host all their client-side JavaScript code and CSS style sheets. During development, the developer discovers another third-party service that better fits their use case and decides to switch.

If the developer deletes the bucket but forgets to remove the corresponding DNS record, he'll have inadvertently introduced a new subdomain takeover vulnerability. An attacker can then create a new AWS S3 bucket with an identical name through the AWS console and take control of all content served at subdomain.company.com.



Setting up a DNS CNAME record pointing to an AWS S3 bucket using Cloudflare DNS

Later throughout this article in the exploitation section, we will go over why this is more problematic than it currently seems to be. Let's take a look at another example where proactive measures have been deployed to prevent subdomain takeover vulnerabilities.

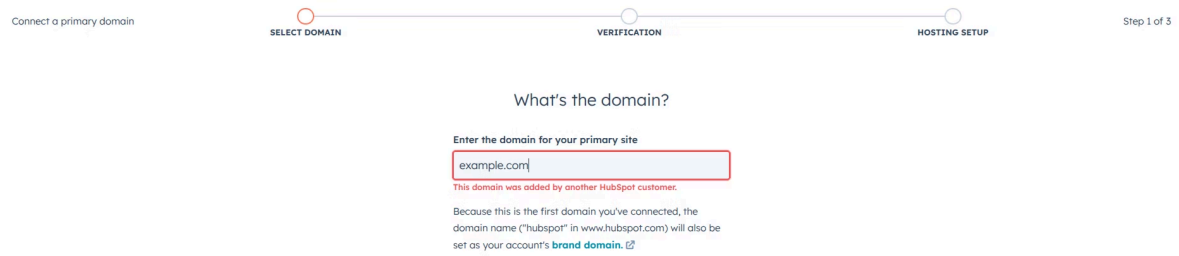
TIP! The example above documents a subdomain takeover vulnerability, which differs from an AWS S3 bucket misconfiguration issue. If you'd like to read more in-depth about AWS S3 bucket

misconfigurations, we recommend you read our detailed article '[Hacking misconfigured AWS S3 buckets.]'

HubSpot (non-vulnerable case)

HubSpot is another popular third-party software vendor enabling marketing and sales teams to work closely together. This third-party service also provides a feature to design and deploy static sites (such as landing pages, blogs, contact forms, etc.). As part of their service, they also allow customers to connect their own primary domain. However, in this case, HubSpot requires that the domain you're about to add is not connected to another HubSpot account. In addition to that, it also auto-generates a hostname for you.

In this scenario, a subdomain takeover is unlikely, even if the vulnerable company forgets to remove the DNS record that points to a HubSpot domain. They will have to remove the primary domain from their account altogether:



HubSpots proactive security measures against subdomain takeover vulnerabilities

Atlassian StatusPage (non-vulnerable case)

Atlassian StatusPage is an example where subdomain takeover is non-existent. StatusPage is a simple tool that helps companies keep their customers updated on possible service disruptions, planned or ongoing maintenance, and other system performance issues. StatusPage also allows customers to point one of their own (sub)domains to their status page.

However, StatusPage requires that you complete the DNS verification first, which involves adding a TXT record. As an attacker, you'll need to have access to their DNS registrar to alter existing records, which is unlikely to be the case. This scenario perfectly demonstrates where subdomain takeovers remain impossible.

TIP! '[Can I Take Over XYZ?]' is a popular community-powered resource that lists tens of vendors where subdomain takeovers are possible!

Automating subdomain takeover vulnerabilities

Automation is key when it comes to identifying subdomain takeover vulnerabilities. Luckily for us, there are several open-source tools that we can make use of. Our first focus should be to gather all possible subdomains of our bug bounty target. Tools like OWASP Amass and Subfinder (from Project Discovery) support both passive and active enumeration methods to help us gather a list of all subdomains.

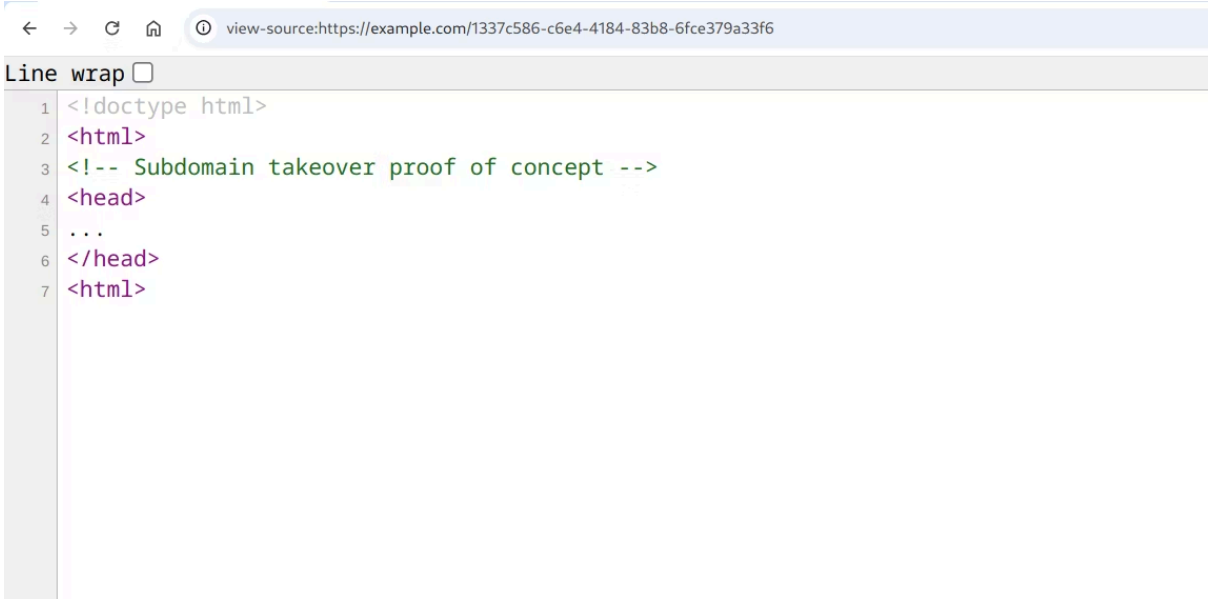
Next up, we will need to check the DNS records and HTTP responses of each individual subdomain to understand if they're pointing to a non-existing service and are susceptible to subdomain takeovers. You can even automate this step using open-source tools. Subjack and Subzy are a great fit as we can easily feed the tools our list of subdomains and they'd auto-check them against tens of different third-party services.

Subjack flagging potential subdomain takeover vulnerabilities

Once we've successfully identified a subdomain takeover vulnerability, we can proceed to the exploitation phase.

Exploiting subdomain takeovers

There are several ways you can actively exploit subdomain takeover vulnerabilities, which we will document below in detail. But before we proceed, let us remind you that generally, most companies prefer you to cease testing after having proven that you were able to take control of a subdomain. This usually involves creating a simple hidden page with a non-guessable URL key on the subdomain with a hidden HTML comment:



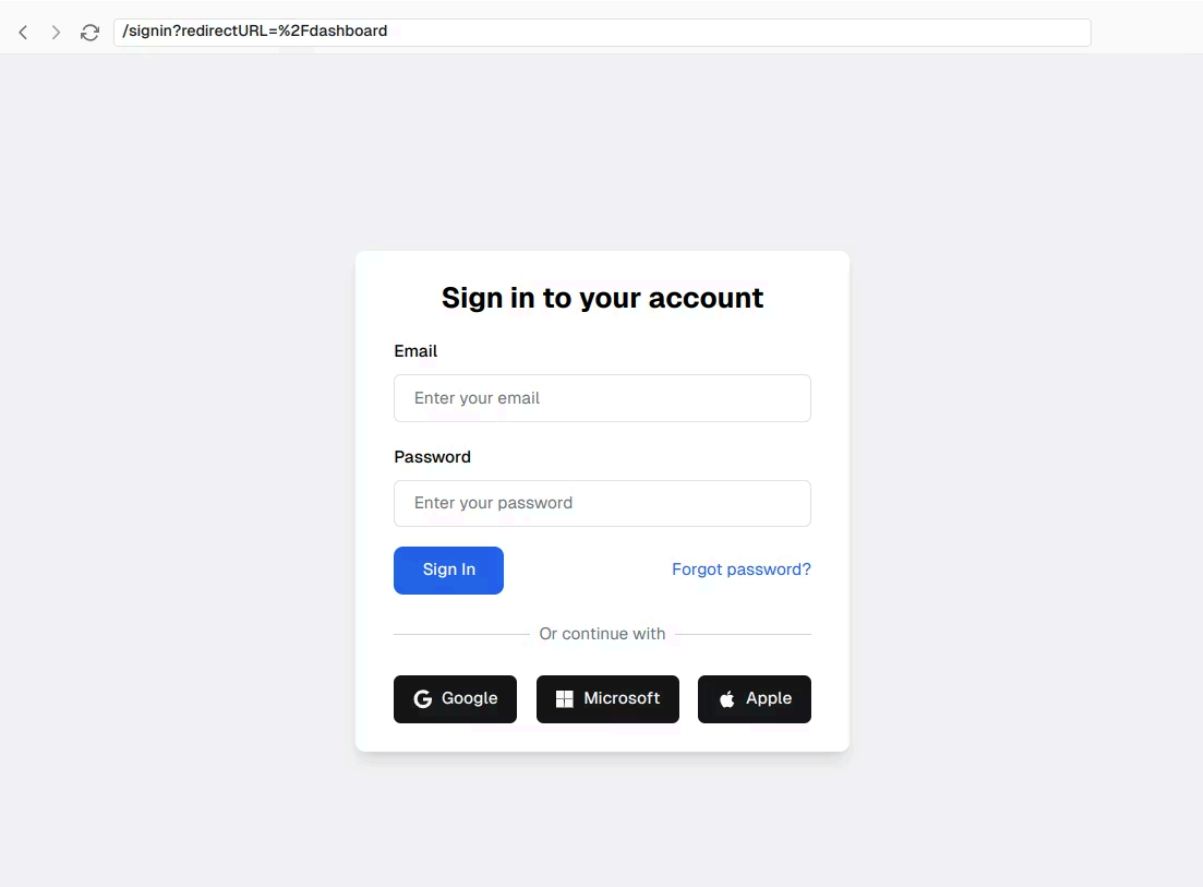
Example of a subdomain takeover proof of concept

Let's take a look now at some of the exploitation methods in detail.

OAuth/SSO token leak via open URL redirects

Most companies implement OAuth or SSO authentication to let customers sign in with their Microsoft or Google accounts. When developers configure these authentication flows to allow redirects to any subdomain of the primary domain, attackers can exploit subdomain takeovers to grab session tokens.

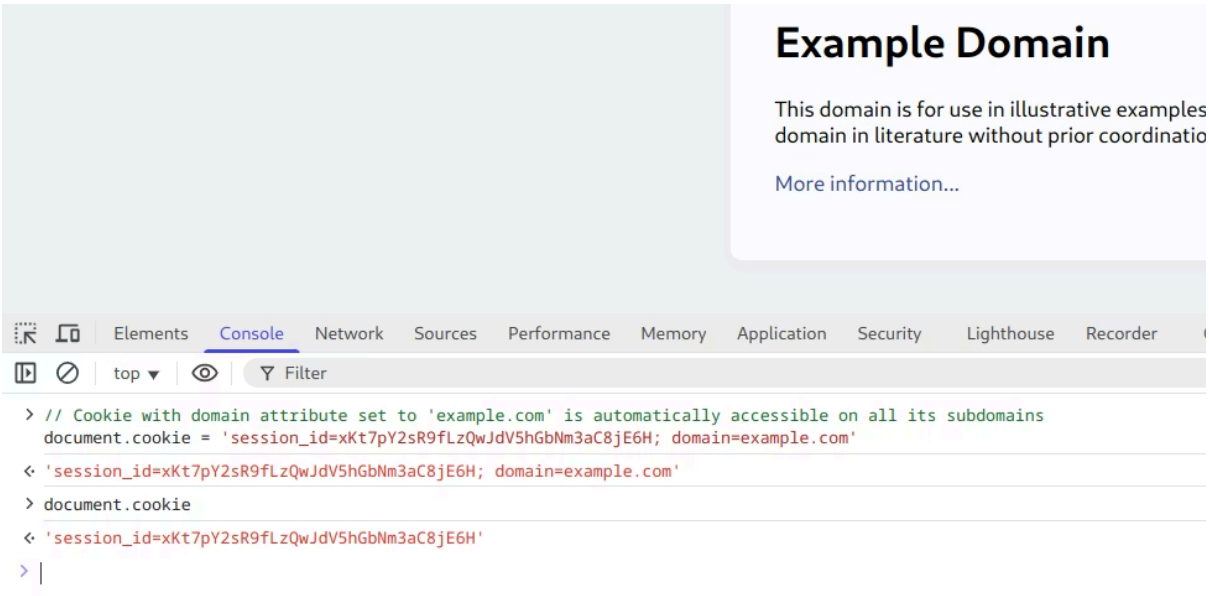
Since OAuth/SSO flows typically return tokens as URL parameters or fragments, attackers can set up listeners on the vulnerable subdomain to capture and forward these secret tokens to their own servers.



Example of an authentication form supporting OAuth/SSO

Cookie leaking through misconfigured cookie policies

When websites configure cookies with a domain attribute of `*.example.com` or `.example.com`, these cookies automatically become available to all subdomains of example.com, including any subdomain vulnerable to takeover. This loose cookie policy gives attackers access to these cookies when they successfully exploit a subdomain takeover vulnerability. If any cookies contain authentication tokens or session IDs, attackers can use them to access victims' accounts on the main domain.



Cookie with domain attribute set to 'example.com' is automatically accessible on all its subdomains

Cross-site request forgery (CSRF) attacks

Most web browsers by default set the same site cookie policy on cookies to 'Lax,' which enables including them in cross-site requests to any subdomain of the main domain. This enables us, in some conditions, to bypass existing security measures that have been put in place to mitigate cross-site request forgery attacks and conduct actions on behalf of the victim on the main application.

Cross-origin resource sharing (CORS) attacks

Subdomain takeover vulnerabilities can also help us exploit cross-origin resource sharing (CORS) issues to leak sensitive data available on the main application, provided that our subdomain is whitelisted for CORS requests.

Some applications whitelist all subdomains by default to allow seamless integration between all services. There are several open-source tools that can help you auto-detect CORS issues on your list of subdomains.

Content security policy (CSP) bypass

Content Security Policy (CSP) is a browser security feature that allows developers to take control of all resources that are evaluated on their application. This can hold back cross-site scripting (XSS) attacks. We can leverage our subdomain takeover to bypass any existing CSP measures.

You should specifically look for if the vulnerable subdomain is whitelisted in the content security policy set on the main application.

TIP! When you discover a subdomain takeover vulnerability involving a content delivery network (CDN) or cloud storage service (like AWS S3), check if the main website loads resources (especially

JavaScript files) from this subdomain. This could indicate a potential supply chain attack vector, where the hijacked subdomain would allow you to inject malicious code into the main application!

Conclusion

Subdomain takeovers are relatively easy to find, however they only do pose a more significant impact to the vulnerable organization when exploited in the right conditions. In this article, we learned how to identify potential subdomain takeovers and several ways on how to exploit these vulnerability types to escalate them to higher severity issues. We've also concluded that automation plays a key role in staying on top of finding new subdomain takeovers before others do!

You've just learned how to hunt for subdomain takeover vulnerabilities... Right now, it's time to put your skills to the test! Browse through our 70+ public bug bounty programs on Intigriti, and who knows, maybe your next bounty will be earned with us!

START HACKING ON INTIGRITI TODAY

REQUEST A DEMO
intigriti.com/demo

VISIT THE WEBSITE
intigriti.com

GET IN TOUCH
hello@intigriti.com