Subdomain Takeover

Subdomain:

A subdomain is a prefix added to a domain name to separate a section of your website. Site owners primarily use subdomains to manage extensive sections that require their own content hierarchy, such as online stores, blogs or support platforms.

Subdomains function as a separate website from its domain. This distinction enables you to develop a section of your website without muddling your site’s overall intent. As a result, you increase your chances to build a niche authority and gain organic traffic.

Regardless of your purpose for creating a subdomain, optimize your string to clearly signal your standalone site’s purpose. Plus, ensure you’re working with an SSL certificate provider that provides subdomain protection. While choosing a domain name is one of the first steps you’ll make when creating a website, you can create a subdomain at any time. Registering a domain name gives you the rights to unlimited subdomains, including anything from “abc.” to “xyz.”

Subdomain Takeover:

A subdomain takeover occurs when an attacker gains control over a subdomain of a target domain. Typically, this happens when the subdomain has a canonical name (CNAME) in the Domain Name System (DNS), but no host is providing content for it. This can happen because either a virtual host hasn't been published yet or a virtual host has been removed. An attacker can take over that subdomain by providing their own virtual host and then hosting their own content for it.

If an attacker can do this, they can potentially read cookies set from the main domain, perform cross-site scripting, or circumvent content security policies, thereby enabling them to capture protected information (including logins) or send malicious content to unsuspecting users. A subdomain is like an electrical outlet. If you have your own appliance (host) plugged into it, everything is fine. However, if you remove your appliance from the outlet (or haven't plugged one in yet), someone can plug in a different one. You must cut power at the breaker or fuse box (DNS) to prevent the outlet from being used by someone else.

Process of Subdomain Takeover:

An attacker sets up a virtual host for a subdomain name you bought on the hosting provider, before you get to do it. Suppose you control the domain example.com. You want to add a blog at blog.example.com, and you decide to use a hosting provider who maintains a blogging platform. (For "blog", you can substitute "e-commerce platform", "customer service platform", or any other "cloud-based" virtual hosting scenario.) The process you go through might look like this:

1. You register the name "blog.example.com" with a domain registrar.
2. You set up DNS records to direct browsers that want to access blog.example.com so that they go to the virtual host.
3. You create a virtual host at the hosting provider.

Unless the hosting provider is very careful to verify that the entity who sets up the virtual host actually is the owner of the subdomain name, an attacker who is quicker than you could create a virtual host with the same hosting provider, using your subdomain name. In such a case, as soon as you set up DNS in step 2, the attacker can host content on your subdomain.

Types of Subdomain Takeover:

The majority of subdomain takeover vulnerabilities happen because of stale DNS CNAME entries. The example above illustrates the typical attack scenario, where a CNAME record of the organization points to an empty site hosted by a third-party service.

CNAME Chains:

But sometimes a CNAME record does not point directly to a domain available for takeover. Instead, it points to another domain with a CNAME, forming a CNAME chain. If the end of that CNAME chain is available for takeover, attackers can take over that site and achieve the same results.

```
a.example.com -> b.example.com -> example.github.io
```

In the above scenario, attackers can take over both `a.example.com` and `b.example.com` by registering `example.github.io`.

MX Takeovers:

Takeovers can also happen if there are misconfigurations with other kinds of DNS records. MX records are used to receive emails for a domain name. If an attacker is able to gain control over the domain name pointed to by your MX records, she will be able to receive emails addressed to your domain. This can lead to the exposure of sensitive information to outsiders.

NS Takeovers:

Finally, NS takeovers occur when there are stale NS records. If an attacker is able to take over the base domain of an NS entry, she can return any DNS response when your site is requested, and redirect the user anywhere she pleases.

Second-Order Takeovers:

There is also another class of takeover attacks, called "second-order takeovers" or "broken link hijacking". They happen when a website uses a resource that is hosted on an external domain, which in turn is vulnerable to takeover. For example, a website uses a JavaScript file hosted on an external domain. When that external domain is deleted and becomes available for registration to anyone, an attacker can register that external domain and host an arbitrary JS file that would be included on the website.

The Impact of Link Hijacking:

The impact of second-order takeovers depends on the type of link that was hijacked. If the broken link is embedded in a script tag, attackers will be able to cause a persistent XSS on the page.

```
<script src=”http://external.com/render.js”></script>
```

The HTML link tag is used for linking external stylesheets. If the broken link is in an HTML link tag, attackers could link the site to a malicious stylesheet, and cause clickjacking or website defacement.

```
<link rel="stylesheet" type="text/css" href="http://external.com/theme.css">
```

And if the broken link is located in an HTML anchor tag, attackers could redirect your users to a malicious site. Since the site is linked from your legitimate website, this makes for a very convincing phishing attack!

```
<a href=“http://external.com”>Visit our new website!</a>
```

To make matters worse, if your anchor tag is missing the rel=”noopener noreferrer” attribute, the hijack can have additional security consequences.

```
<a href=“http://external.com” rel=”noopener noreferrer”>Visit our new website!</a>
```

If the attacker hijacks an anchor tag that is missing the “noopener” attribute, they might be able to change the content and location of the originating page. On the other hand, if the anchor tag is missing a “noreferrer” attribute, the hijacked link might leak sensitive info like object IDs to the malicious site.

Social Media Hijacking:

Another way attackers can impersonate your site is through social media hijacking. Let’s say you register a username on a social media site and link to it from your website.

```
<a href=“https://twitter.com/vickieli7” rel=”noopener noreferrer”>Visit our Twitter page!</a>
```

When you later delete that account or change your username, social media sites usually free up that username for others to register. If you don’t update that social media link on your site, an attacker could register your old username and impersonate your social media presence. Still, another consideration when using social media is the integrity of the links that you embed in your posts. When you link to a site from the official social media account of your company, that website could become expired and available for registration. Now anyone can register that site and hijack the content that you are posting on your accounts! For examples of this, check out this blog post, where a hacker talks about how he was able to hijack the Tweets of celebrities.

Security Impact:

A successful subdomain takeover enables an attacker to serve content on the subdomain. If the subdomain is a child domain of the service’s basename, then the attacker can read and set cookies on the basename too – subdomain.example.com can set cookies for example.com.

Preventing subdomain takeovers:

It’s matter of order of operations in lifecycle management for virtual hosts and DNS. Depending on the size of the organization, this may require communication and coordination across multiple departments, which can only increase the likelihood for a vulnerable misconfiguration.

- Define standard processes for provisioning and deprovisioning hosts. Do all steps as closely together as possible.
    - Start provisioning by claiming the virtual host; create DNS records last.
    - Start deprovisioning by removing DNS records first.
- Create an inventory of all of your organization's domains and their hosting providers, and update it as things change, to ensure that nothing is left dangling.
- Put pressure on hosting vendors to close gaps; ask how they verify that someone claiming a virtual host actually has a legitimate claim to the domain name. Work within your organization to make this part of the vendor qualification process.

Reference:

- https://www.wix.com/blog/2020/10/what-is-a-subdomain/
- https://developer.mozilla.org/en-US/docs/Web/Security/Subdomain_takeovers
- https://www.honeybadger.io/blog/subdomain-takeover/
- https://hacker101.linuxsec.org/vulnerabilities/subdomain_takeover.html