SUBDOMAIN TAKEOVER

An Overview

Submitted by: Raunak K.Bagchi

What is a subdomain?

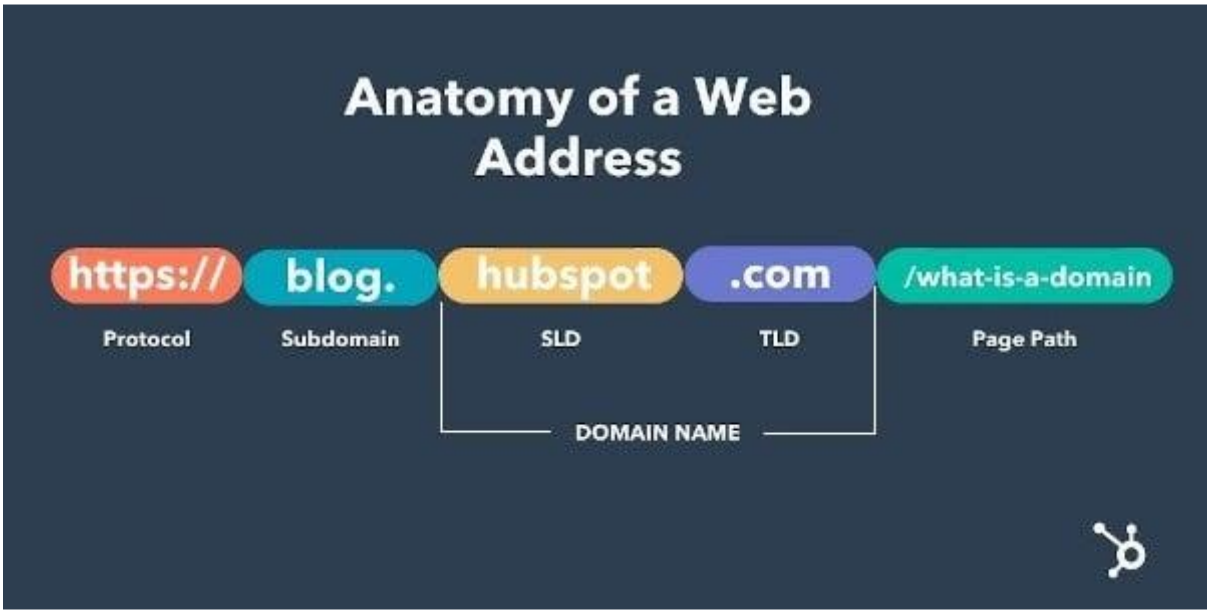
A subdomain name is a piece of additional information added to the beginning of a website's domain name. It allows websites to separate and organize content for a specific function — such as a blog or an online store — from the rest of your website.

A domain name typically has two parts: **The top-level domain (TLD)** is the extension, such as *.com* or *.org*, and the **second-level domain (SLD)** is the unique part of the domain name, often a business or brand name. In the *hubspot.com* example, *com* is the TLD and *hubspot* is the SLD.

The **subdomain** is what goes *before* the SLD. The most common subdomain is *www*, which stands for World Wide Web. This subdomain contains a website's homepage and its most important pages. The *www* subdomain is so widely used that most domain registrars include it with domain name purchases.

Subdomains are also commonly used to separate a section of a website from the main site. For example, *blog.hubspot.com* and *shop.hubspot.com* direct to our blog and online store respectively.

When we group our domain name and subdomain with a protocol at the beginning (HTTP or HTTPS for websites) and an optional file path at the end, we have a complete URL:



What is a subdomain used for?

Subdomains make it simple for you to organize the various functions of your website, while also making it easier for users to find these different functions.

Think about it this way: If you’re hosting a party, you need to provide guests with your address. The TLD would be the city you live in while the SLD would be your number and street name. If you live in an apartment building, you need to get even more specific so your guests know which apartment to ring. Your apartment number would be akin to a subdomain — a specific section of the greater building that's dedicated to your living space.

If you plan to add more functions to your website, such as a store, a forum, or a blog, you might add a subdomain to your domain to separate these functions off from your main website.

You can also use subdomains to create localized content. For example, if you run a restaurant chain with multiple locations, customers can visit *www.myrestaurant.com* for all-encompassing content. Or, customers looking for the menu at your Nashville, Tennessee location can access this information via *nashville.myrestaurant.com*.

What is subdomain takeover vulnerability?

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.

How do they happen?

If the process of provisioning or deprovisioning (removing) a virtual host is not handled properly, there can be an opportunity for an attacker to take over a subdomain.

During provisioning

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.

During deprovisioning

You take down your virtual host, but an attacker sets up a new virtual host using the same name and hosting provider.

You (or your company) decide that you no longer want to maintain a blog, so you remove the virtual host from the hosting provider. However, if you don't remove the DNS entry that points to the hosting provider, an attacker can now create their own virtual host with that provider, claim your subdomain, and host their own content under that subdomain.

Impact

A successful exploitation of this kind of vulnerability allows an adversary to claim and take control of the victim’s subdomain. This attack relies on the following:

1. The victim’s external DNS server subdomain record is configured to point to a non-existing or non-active resource/external service/endpoint. The proliferation of XaaS (Anything as a Service) products and public cloud services offer a lot of potential targets to consider.
2. The service provider hosting the resource/external service/endpoint does not handle subdomain ownership verification properly.

If the subdomain takeover is successful, a wide variety of attacks are possible (serving malicious content, phishing, stealing user session cookies, credentials, etc.). This vulnerability could be exploited for a wide variety of DNS resource records including: A, CNAME, MX, NS, TXT etc. In terms of the attack severity an NS subdomain takeover (although less likely) has the highest impact because a successful attack could result in full control over the whole DNS zone and the victim’s domain.

Mitigations

Preventing subdomain takeovers is a 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.

If you discover that a subdomain of your domain has been taken over, the first step, if possible, is to "cut power" by removing the DNS entry for the subdomain. If your site has multiple layers of virtualization (e.g., a CDN in addition to virtual hosting), you may need to examine each layer to see where exactly the attacker asserted their virtual host claim to take over your domain.

References

- https://developer.mozilla.org/en-US/docs/Web/Security/Subdomain_takeovers
- https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/02-Configuration_and_Deployment_Management_Testing/10-Test_for_Subdomain_Takeover