Low-priority and Delegated Domains

In this lesson, we'll learn the importance of delegating domains and assigning domains to low-priority services properly.

We'll cover the following

- What is Google?
- Why low-priority services should be on a separate domain
- Delegating domains to external entities

What is Google?

You might say that Google is a search engine, but then you'd find yourself thinking about the vast amount of products they offer and quickly realize that Google is a conglomerate that offers a growing number of products, from Maps to services like Keep or Chrome Remote Desktop.

You might be wondering where we're headed, so let me clarify that right now, the organization you work for probably has more than one service it offers to customers, and those services might not really be related to each other. Some of them, for example, could be low-priority ones the company works on, such as a corporate or engineering blog, or a URL shortener your customers can use alongside other, larger services you offer. Often, these services, sit on a domain such as `blog.example.com`.

Why low-priority services should be on a separate domain

"What's the harm", you say? I would counter that using your main domain to store low-priority services can harm your main business, and you could be in a lot of trouble. Even though there's nothing inherently wrong with using subdomains to serve different services, you might want to think about offloading low-priority services to a different domain, the reasoning behind this choice is that, if the service running on the different TLD gets compromised, it will be much harder for attackers to escalate the exploit to your main service(s).

attackers to escalate the exploit to your main service(s).

As we’ve seen, cookies are often shared across multiple subdomains (by setting the *domain* attribute to something like `*.example.com`, `.example.com` or simply `example.com`), so a scenario could play out where you install a popular blogging software like WordPress on `engineering-blog.example.com` and run with it for a few months, forget to upgrade the software and install security patches as they get published.

Later, an XSS in the blogging platform allows an attacker to dump all cookies present on your blog somewhere in his control, meaning that users who are logged onto your main service (`example.com`) who visit your engineering blog could have their credentials stolen. If you had kept the engineering blog on a separate domain, such as `engineering-blog.example.io`, that would not have been possible.

Delegating domains to external entities

In a similar fashion, you might need to delegate domains to external entities, like email providers. This is a crucial step as it allows them to do their job properly. Sometimes, though, these providers might have security flaws on their interfaces as well, meaning that your users, on your domains, are going to be at risk.

Evaluate if you could move these providers to a separate domain, as it could be helpful from a security perspective. Assess risks and goals and decide accordingly, as always, there’s no one right answer.

In the next lesson, we’ll learn about the Open Web Application Security Project.