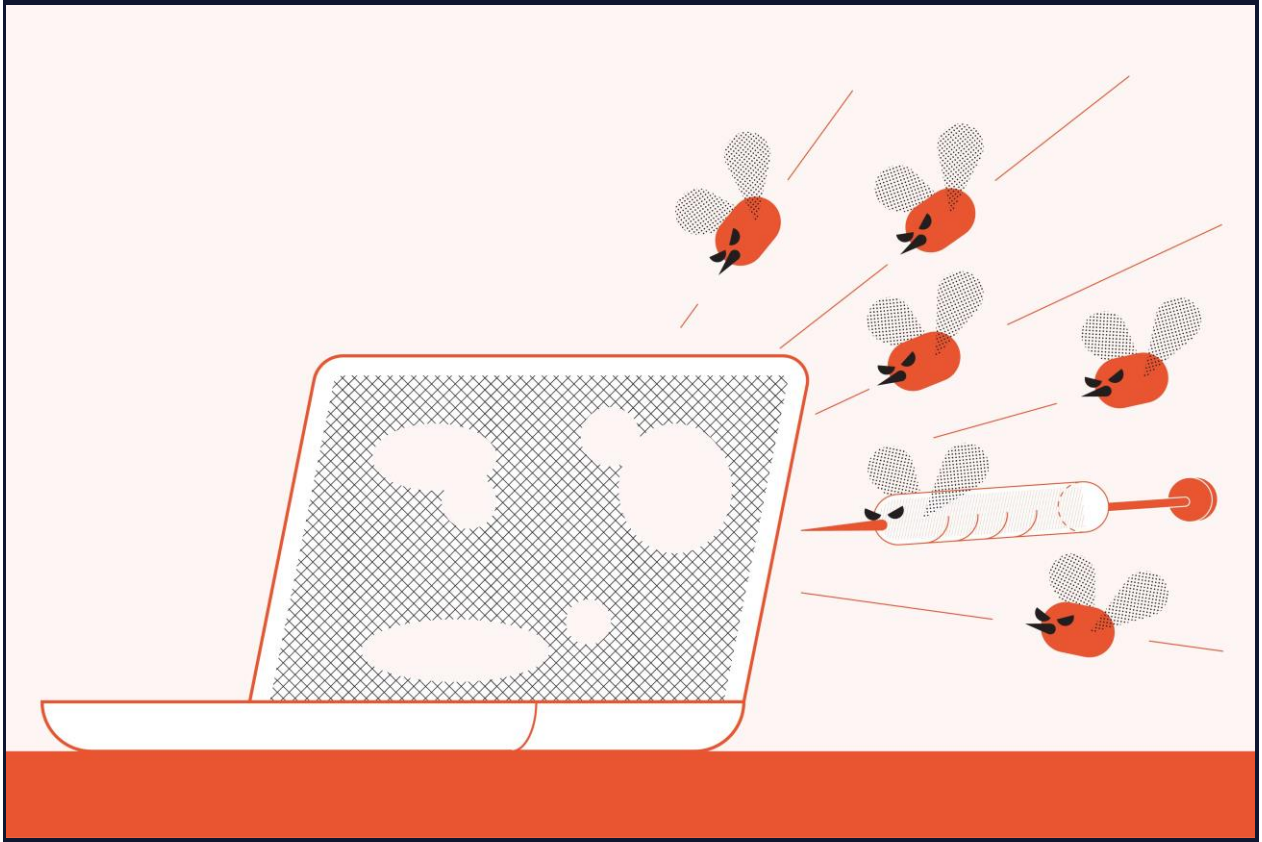
The World of Insecure Design

This article will discuss Insecure Design: what it is, how it happens, and why it's dangerous.

What Is Insecure Design?



The Insecure Design category deals with flaws in the structure of software or systems, which negatively impact security. Unlike other categories that deal with more specific vulnerability types, Insecure Design is more about the root cause.

Unlike vulnerabilities caused by bugs or oversights, vulnerabilities caused by insecure design are intended behavior in that they are working as designed (though the security consequences of the behavior are often unforeseen during the design process). If you want to remove the vulnerability, you need to redesign the portion of the system that causes it.

One example of an insecure design might be a website that will email you your password if you forget it. This is a *huge* security issue, but not one that can be easily fixed. The website needs to redesign not just its credential

recovery system but also its credential storage and authentication systems as well.

Why Does Insecure Design Happen?

So, what could have caused our example website to create such a terrible insecure design? Well, it probably wasn't on purpose. Generally speaking, people don't *choose* to design insecure systems. The issue is that insecure design is the default. In other words, secure design requires knowledge and effort, and unless both are applied sufficiently, a person risks creating an insecure design.

To create a secure design, you need to know the security requirements, have the background knowledge to interpret those requirements correctly, have the technical knowledge to ensure that your design meets those requirements, and have the time and resources to do everything correctly. Lacking any one of those things can result in an insecure design.

Bad design requirements can also lead to insecure design if the design requirements conflict with security. It's usually not the security team putting together the design requirements for a project, and it's very easy for someone without a security background to put forth a requirement that sounds reasonable at first but is a security nightmare.

Maybe the developers of the previously-mentioned website didn't know or understand the security requirements for authentication and credential management. Maybe they didn't have the time or resources to do it correctly. Maybe unsafe design requirements were what they had to work with. Regardless of how the website ended up designed as it was, the result is a massive security liability.

Why Is Insecure Design Dangerous?

The main issue with insecure design is that its insecurity is fundamental. We usually assume that vulnerabilities result from bugs or oversights that can be fixed, but this isn't the case with vulnerabilities caused by insecure design. These vulnerabilities aren't caused by issues implementing the design but by the design itself. Even if you could guarantee that the implementation worked exactly as designed with no bugs, the vulnerabilities would still be present.

Because the vulnerabilities are caused by the design, the design needs to be changed to fix them, which isn't always easy. Depending on where the design issue is, you might have to adjust large portions of the design or scrap the design entirely. For example, many protocols designed in the early days of the internet suffer from insecure design and have been replaced by modern, more secure protocols.

Conclusion

Insecure design may not be the flashiest category on the OWASP Top 10, but it can creep into any project if you're not watching out for it. The ease with which it can occur, combined with the fact that it can introduce difficult-to-fix vulnerabilities, means that everyone should keep it in mind when designing or developing. It may seem costly to invest the time and resources to create a secure design, but an insecure design may cost more in the long run.