The World of Vulnerable and Outdated Components

This article will discuss Vulnerable and Outdated Components: what it is, how it happens, and why it’s dangerous.

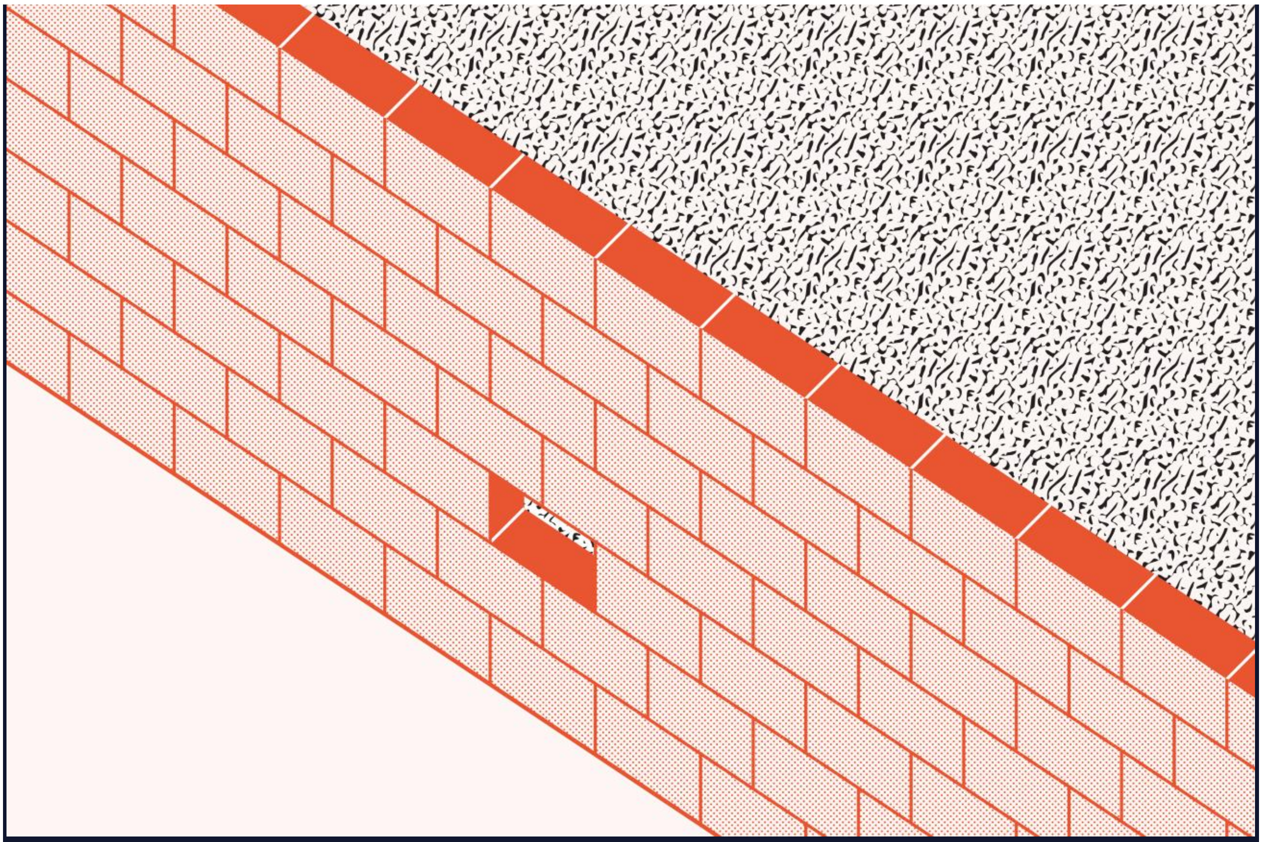
What are Vulnerable and Outdated Components?

What’s a Component, anyway?

In this context, a component is a discreet piece of software within a system. That software system might be a single piece of software or an entire tech stack, and components might be anything from a software library to an operating system to a database management system. Additionally, components can contain other components.

What does it mean for a component to be vulnerable or outdated?

A component is considered outdated when a more recently released stable version is available. Technically, this isn’t an issue on its own, but outdated components are more likely to be vulnerable. In addition, a component is considered vulnerable when there is at least one known vulnerability - one whose existence is public knowledge. Take, for example, the image below:



The image depicts a wall that has a hole in it. That hole makes this wall weak and vulnerable to attacks.

Why are Vulnerable and Outdated Components Dangerous?

Vulnerabilities Are Bad

The risk of outdated and vulnerable components is that they introduce other vulnerabilities into our environment, which is bad. If your environment already has bad security, having vulnerable components certainly won’t help its security. If the environment’s security is otherwise good, having vulnerable components can differentiate between an attack succeeding or failing.

Defenders have limited resources and time. If they constantly need to deal with vulnerabilities introduced by bad components, their attention may be spread too thin, causing the security of the software system overall to suffer.

Why Are Vulnerable and Outdated Components Still An Issue?

Updating isn’t always easy.

It might seem like there's an easy solution to vulnerable and outdated components: keep your components up to date. Unfortunately, this is easier said than done.

Firstly, it can be difficult to know what components you have. For instance, a single piece of modern software can have multiple layers of dependencies, all of which must be kept up to date. Now multiply the many different pieces of software you have in your system. The product of that multiplication will grow quickly.

Secondly, updating components can break compatibility. If a component changes its behavior with an update, it may break parts of the system. If you're lucky, the system will break in obvious ways and can be fixed immediately. If you're unlucky, the issues may be subtle and go unnoticed for long periods, waiting to pop up at the most inopportune moment.

Finally, components aren't maintained forever. There may be components in the system that can't be replaced but are no longer actively supported or maintained. For example, millions of computers still run Windows XP, even though it's a security nightmare.

Time is not on the defender's side.

More than anything else, Vulnerable and Outdated Components are an issue because new vulnerabilities are constantly being discovered. You can't just secure a system and leave it unattended because it might suddenly stop being secure without anything having changed within it. For a defender, the situation will worsen as time passes unless an effort is made to maintain the system.

Time poses another type of issue for defenders, too. When the maintainers of a component become aware of a vulnerability, it takes time for them to develop and deploy an update. If the maintainers learn about the vulnerability at the same time that it becomes public knowledge, then the component will be vulnerable until an update can be developed, released, and applied. Even if the maintainers are the first to know about the vulnerability, releasing an update for it starts a clock. Hackers will begin reverse-engineering the update to learn about what vulnerabilities it fixes. If an organization is too slow to apply updates, attackers may be able to use that knowledge against them.

Conclusion

Vulnerable And Outdated Components may not be the most glamorous class of vulnerability, but it is one of the most frustrating. It requires constant effort to combat and can render previously-secure systems vulnerable without warning. What’s more, updating components can introduce issues too, which also require effort to resolve.