VULNERABILITY DECODER

Encapsulation

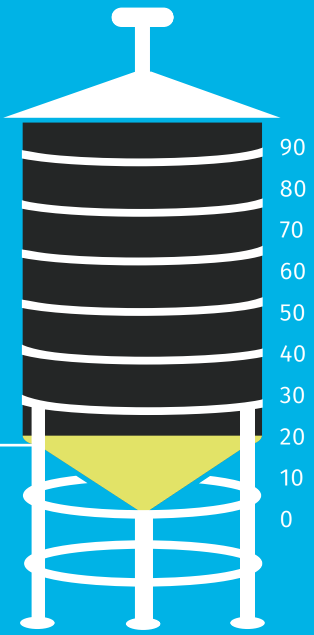
The Vulnerability

Encapsulation refers to a programming approach that revolves around data and functions contained, or encapsulated. Applications become vulnerable to an encapsulation attack when they fail to differentiate data or functionality within components. Without clearly defined boundaries between control spheres, bad code can creep across components and attackers can gain unauthorized access to data and functions.

29%

of applications have an encapsulation vulnerability on initial scan.

Source: SOSS v11



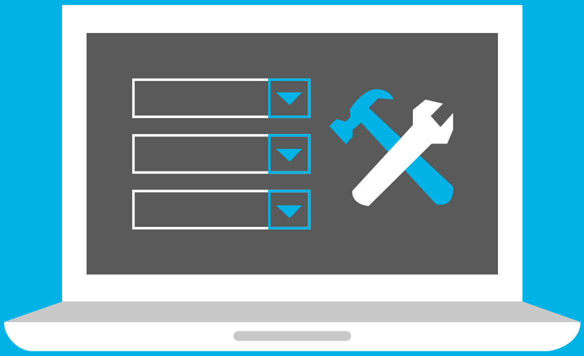
The Risks

An encapsulation attack can lead to service denial, cause security mechanisms to fail, or crash a server. There's also the risk of damage to applications and other tools and resources, or a ransomware attack.

Example Breach

In 2016, San Francisco's Municipal Rail (MUNI) suffered a significant ransomware attack based on an encapsulation vulnerability. MUNI had neglected to patch a Java deserialization flaw that had existed for more than a year.

Prevention & Remediation



The discovery of an encapsulation vulnerability requires an organization to block access to the affected application, database, or system until it can be fully protected. In the case of ransomware, a backup allows the organization to return to a working state quickly and without paying the fee demanded by the attacker.

Recommendations

Nobody writes perfect code the first time around. You can avoid vulnerabilities and prevent breaches when you:

☑ Get training in secure coding best practices through on-demand eLearning courses, in-person security consultations, and professional development certifications and conferences.

☑ Scan early and often to detect flaws while you code. Use application security tools that allow you to scan small batches of code instantaneously, and provide remediation guidance within your development workflow.

Download the Secure Coding Best Practices Handbook

Join the Veracode Community
community.veracode.com

VERACODE