Web Applications and Vulnerabilities: What to Look Out For

ARTICLE 09.07.22 BOP-ADMIN

Recent years have seen a dramatic increase in the amount of publicly accessible web applications. As more organizations have expanded their internet presence, web application attacks have become increasingly profitable for threat actors. Recent vulnerabilities such as Log4j have also brought more intense scrutiny to web applications. If your organization hosts business-critical applications or is allowing customers to access their data through the web, it is no longer sufficient to rely simply on traditional external security assessments.

Why network testing is not enough

While traditional external vulnerability testing may include some light unauthenticated web application scanning, automated scanning cannot be relied upon to validate the security of web applications in the same way that it can be for network and host vulnerabilities. Every web application is unique, and automated scanning tools lack the context to catch many vulnerabilities. The only effective way to test web applications involves manual testing, supplemented with the targeted use of automated tools.

Testing from an unauthenticated context is also a problem. According to a review of 2021 breaches performed by Verizon, around 80% of the web application breaches in 2021 were attributed to stolen credentials rather than technical vulnerabilities. This represents a significant increase since 2017, when the number was 50%. The increase in this can be attributed to two attack methods: phishing and credential stuffing.

Phishing continues to be a major and successful attack vector for stealing credentials. Despite improvements in detection and response capabilities, threat actors have continued to find success with this technique. Credential Stuffing leverages large sets of usernames and passwords, usually stolen in data breaches and sold on the dark web.

An attacker will take these sets of credentials and use automated tools to test them against a wide array of websites, looking for sites where the user reused the same username and password combination. Because these methods involve the attacker gaining authenticated access to the application, it is important to ensure that any security testing is performed from an authenticated perspective.

How to perform an authenticated web application assessment

When evaluating if your web application is sufficiently comprehensive, a good resource to use is the Offensive Web Application Security Project (OWASP) Top 10. OWASP monitors web application vulnerabilities and breaches and compiles the most common types of vulnerabilities. The current list was updated in late 2021 and includes the following categories from most to least prevalent:

1. Broken Access Control
2. Cryptographic Failures
3. Injection

4. Insecure Design
5. Security Misconfiguration
6. Vulnerable and Outdated Components
7. Identification and authentication Failures
8. Software and Data Integrity Failures
9. Security Logging and Monitoring Failures
10. Server-Side Request Forgery

If your organization is hosting externally accessible web applications that store sensitive data, it is important to ensure that your security assessment methodology is covering at least the areas covered in the OWASP Top 10.

Penetration testing can be used to cover the majority of the OWASP Top 10 categories. The goal of the penetration test is to identify vulnerabilities from an external perspective using manual testing and targeted automated tools. The test should be performed with a white or grey box perspective, where the tester is given access to the application and one or two accounts of each role in order to sufficiently cover access control issues both between users of the same privilege level and between users of different privilege levels. With a white box approach, the tester could also be provided the source code of the application. Additionally, the tester should be provided with information about the architecture of the application and the software suites and tools in use on the back end to better understand how to attack the application.

Why penetration testing is not enough

Not all areas of the OWASP Top 10 can be covered sufficiently by a penetration assessment, so the testers should also meet with the developers to discuss the areas of Insecure Design, Software and Integrity Failures, and Logging and Monitoring. These areas cannot be observed from an external perspective so a collaborative approach should be used to ensure coverage.

With this approach, you can ensure that your application is secure against the most prevalent web application vulnerabilities in the threat landscape right now and ensure that your user’s data is secure.

Cyber Security Services

Gui Cozzi
Cybersecurity Practice Lead
gcozzi@ddaftech.com • 859.425.7649