The 10 Most Common Website Security Attacks

1. Cross-Site Scripting (XSS)

Cross-site scripting targets the users of a site instead of the web application itself. The malicious hacker inserts a piece of code into a vulnerable website, which is then executed by the website’s visitor. The code can compromise the user’s accounts, activate Trojan horses or modify the website’s content to trick the user into giving out private information.

You can protect your website against XSS attacks by setting up a web application firewall (WAF). WAF acts as a filter that identifies and blocks any malicious requests to your website. Usually, web hosting companies already have WAF in place when you purchase their service, but you can also set it up yourself.

2. Injection Attacks

The Open Web Application Security Project (OWASP) in their [latest Top Ten research](https://owasp.org/www-project-top-ten/2017/) named injection flaws as **the highest risk factor for websites**. The SQL injection method is the most popular practice used by cyber criminals in this category.

The injection attack methods target the website and the server’s database directly. When executed, the attacker inserts a piece of code that reveals hidden data and user inputs, enables data modification and generally compromises the application.

Protecting your website against injection-based attacks mainly comes down to how well you’ve built your codebase. For example, the number one way to mitigate a SQL injection risk is to always use parameterized statements where available, [among other methods](https://www.hacksplaining.com/prevention/sql-injection). Furthermore, you can consider using a third-party authentication workflow to out-source your database protection.

3. Fuzzing (or Fuzz Testing)

Developers use [fuzz testing](https://searchsecurity.techtarget.com/definition/fuzz-testing) to find coding errors and security loopholes in software, operating systems or networks. However, attackers can use the same technique to find vulnerabilities in your site or server.

It works by initially inputting a large amount of random data (fuzz) into an application to get it to crash. The next step is using a *fuzzer* software tool to identify the weak spots. If there are any loopholes in the target’s security, the attacker can further exploit it.

The best way to combat a fuzzing attack is by keeping your security and other applications updated. This is especially true for any security patches that come out with an update that the perpetrators can exploit if you haven’t made the update yet.

4. Zero-Day Attack

A zero-day attack is an extension of a fuzzing attack, but it doesn’t require identifying weak spots per se. The most recent case of this type of attack was identified by [Google’s study](https://www.zdnet.com/article/google-discloses-windows-zero-day-exploited-in-the-wild/), where they identified potential zero-day exploits in Windows and Chrome software.

There are two scenarios of how malicious hackers can benefit from the zero-day attack. The first case is if the attackers can get information about an upcoming security update, they can learn where the loopholes are before the update goes live. In the second scenario, the cyber criminals get the patch information and target users who haven’t yet updated their systems. In both cases, your security gets compromised, and the subsequent damage depends on the perpetrators’ skills.

The easiest way to protect yourself and your site against zero-day attacks is to update your software immediately after the publishers prompt a new version.

5. Path (or Directory) Traversal

A path traversal attack isn’t as common as the previous hacking methods but is still a considerable threat to any web application.

[Path traversal](https://owasp.org/www-community/attacks/Path_Traversal) attacks target the web root folder to access unauthorized files or directories outside of the targeted folder. The attacker tries to inject movement patterns within the server directory to move up in the hierarchy. A successful path traversal can compromise the site’s access, configuration files, databases, and other websites and files on the same physical server.

Protecting your site against a path traversal attack comes down to your input sanitization. This means keeping the user’s inputs safe and unrecoverable from your server. The most straightforward suggestion here is to build your codebase so that any information from a user isn’t passing to the filesystem APIs. However, if that’s not possible, there are other [technical solutions](https://portswigger.net/web-security/file-path-traversal).

6. Distributed Denial-of-Service (DDoS)

The DDoS attack alone doesn’t allow the malicious hacker to breach the security but will temporarily or permanently render the site offline. [Kaspersky Lab’s IT Security Risks Survey in 2017](https://usa.kaspersky.com/about/press-releases/2018_ddos-breach-costs-rise-to-over-2m-for-enterprises-finds-kaspersky-lab-report" \t "_blank) concluded that a single DDoS attack costs small businesses $123K and large enterprises $2.3M on average.

The DDoS attack aims to overwhelm the target’s web server with requests, making the site unavailable for other visitors. A botnet usually creates a vast number of requests, which is distributed among previously infected computers. Also, DDoS attacks are often used together with other methods; the former’s goal is to distract the security systems while exploiting a vulnerability.

Protecting your site against a DDoS attack is [generally multi-faceted](https://aws.amazon.com/shield/ddos-attack-protection/). First, you need to mitigate the peaked traffic by using a Content Delivery Network (CDN), a load balancer and scalable resources. Secondly, you also need to deploy a Web Application Firewall in case the DDoS attack is concealing another cyberattack method, such as an injection or XSS.

7. Man-In-The-Middle Attack

The [man-in-the-middle attacks](https://en.wikipedia.org/wiki/Man-in-the-middle_attack) are common among sites that haven’t encrypted their data as it travels from the user to the servers. As a user, you can identify a potential risk by examining if the website’s URL begins with an *HTTPS*, where the “S” implies that the data is being encrypted.

Attackers use the man-in-the-middle type of attack to gather (often sensitive) information. The perpetrator intercepts the data as it’s being transferred between two parties. If the data isn’t encrypted, the attacker can easily read personal, login or other sensitive details that travel between two locations on the Internet.

A straightforward way to mitigate the man-in-the-middle attack is to install a Secure Sockets Layer (SSL) certificate on your site. This certificate encrypts all the information that travels between parties so the attacker won’t easily make sense of it. Typically, most modern hosting providers already feature an SSL certificate with their hosting package.

8. Brute Force Attack

A brute force attack is a very straightforward method for accessing the login information of a web application. It’s also one of the easiest to mitigate, especially from the user’s side.

The assailant tries to guess the username and password combination to access the user’s account. Of course, even with multiple computers, this can take years unless the password is very simple and obvious.

The best way of protecting your login information is by creating a strong password or using [two-factor authentication](https://searchsecurity.techtarget.com/definition/two-factor-authentication) (2FA). As a site owner, you can require your users to set up both to mitigate the risk of a cyber criminal guessing the password.

9. Using Unknown or Third-Party Code

While not a straight-up attack on your site, [using unverified code](https://www.tripwire.com/company/press-releases/2017/06/tripwire-research-reveals-danger-of-using-freelance-web-developers) created by a third-person can lead to a severe security breach.

The original creator of a piece of code or an application has hidden a malicious string inside the code or unknowingly left a backdoor. You then incorporate the “infected” code to your site, and then it’s executed or the backdoor exploited. The effects can range from simple data transfer to getting administrative access to your site.

To avoid risks surrounding a potential breach, always have your developers research and audit the code’s validity. Also, make sure that the plugins you use ([especially for WordPress](https://www.tripwire.com/state-of-security/security-data-protection/13-reasons-wordpress-hacks-successful/)) are up to date and regularly receive security patches – research shows that [over 17,000 WordPress plugins](https://websitesetup.org/news/wordpress-statistics-facts/) (or about 47% of WordPress plugins at the time of the study) hadn’t been updated in two years.

10. Phishing

Phishing is another attack method that isn’t directly aimed at websites, but we couldn’t leave it off the list, either, as it can still compromise your system’s integrity. The reason being that phishing is, according to [the FBI’s Internet Crime Report](https://www.ic3.gov/Media/PDF/AnnualReport/2019_IC3Report.pdf), the most common social engineering cybercrime.

The standard tool used in phishing attempts is email. The assailants generally mask themselves as someone they’re not and try to get their victims to share sensitive information or make a bank transfer. These types of attacks can be outlandish as the 419 scam (a part of an [Advance Fee Fraud](https://www.investor.gov/protect-your-investments/fraud/types-fraud/advance-fee-fraud) category) or more sophisticated involving spoofed email addresses, seemingly authentic websites and persuasive language. The latter is more widely known as Spear phishing.

The most effective way to mitigate the risk of a phishing scam is by training your staff and yourself to identify such attempts. Always check if the sender’s email address is legit, the message isn’t odd and the request isn’t bizarre. And, if it’s too good to be true, it probably is.

In Conclusion

The attacks on your website can take many forms, and the attackers behind them can be amateurs or coordinated professionals.

The key takeaway is not skipping on security features when creating or running your site because it can have dire consequences.

While it’s not possible to completely eliminate the risk of a website attack, you can at least mitigate the possibility and the severity of the outcome.