Top 5 Web Application Security Tips

Ben Sandison. Penetration Tester, Ambersail Ltd.

Read our 5 step guide to better web application security.

Here are the most common problems we find when reviewing web applications for security weaknesses.

Use The Right Technologies

Build applications using proven technologies.
Use development tools that have been put through their paces and tested throughly.
Where weaknesses have been discovered and patched.

Keep It Contained

Don't put all your eggs in one basket.
Distribute your web application estate. Reduce the risk of a catastrophic breach by not hosting all web applications on a single server.

Never Trust Defaults

A major cause of cyber security breaches.
Application frameworks that come out of the box have widely published configuration details.
Details that provide easy, low hanging fruit to criminals.

Web Application Firewalls

These are proven, highly effective tools to control and filter malicious communications.
They intercept and control web traffic before it even reaches web applications.

Keep Updated

It sounds obvious. We've all heard this advice...
Always keep web application software and services up-to-date.
Regularly patched application are far less likely to be successfully breached. Fact.

ambersail

www.ambersail.com/web-application-penetration-testing

Easy To Find. Easy To Attack.

Thanks to advancements in modern frameworks and cloud technology, creating a web application is now easier than ever.

Web applications can be found everywhere. Supporting vital business functions such as direct selling, managing projects, B2B communications and supporting staff.

This widespread use draws unwanted attention from attackers for several reasons.

Web applications are easy to find. Whether through a simple Google search or DNS bruteforce, web applications are easily discovered.
Web applications are exploitable. A vast number of public exploits are available for almost every web stack. These issues are well known and attacks can often be automated.
Web applications are valuable. Web applications can provide access to huge amounts of confidential data. Often the primary target for modern cyber-criminals.

Fortunately, many common attacks can be easily avoided.

Here are Ambersail's top 5 tips for web application security.

Use The Right Technologies

Don't make work for yourself. Build applications using proven technologies.

Tools such as WordPress, Joomla and Laravel have been around for years. They are the target of countless security reviews and penetration tests. As a result, obvious security issues are highly likely to have been discovered and patched.

Look at it as an attacker would. Which is more likely to contain an easy-to-find vulnerability? A thoroughly vetted framework, or an untested custom implementation…

Of course, frameworks can't provide every feature. Custom functionality will always be required. At a minimum, you want to make sure that the following features are provided (and used).

Authentication. Some authentication schemes such as username and password may seem straightforward and easy to implement, but they have nuances. For example, how would you handle password resets? Bruteforce protections? Preventing time-based user enumeration? The list goes on and on.

Secure data storage. This ties in with authentication, as often the most secure storage is required for user data.
If you consider passwords: are you confident in selecting and implementing a strong hashing algorithm, managing salts and the associated database? Far easier to let a well-tested framework do it.

Input sanitisation. User input has long been a cause of headaches for webmasters, as it leads to issues such as XSS and SQL injection. Often custom implementations are based on the concept of a list of allowed or banned characters. However, creative bypasses are discovered all the time. Again, the nature of a regularly tested framework helps limit this risk.

2 Keep It Contained

What would be worse than a data breach? TWO data breaches…

Lumping all your web applications and services together on one server might make economic sense. It can reduce maintenance effort and costs with no obvious downsides. It simplifies things by putting everything in one place.

However, an attacker would see that server as a goldmine. Not only has all the sensitive data been moved to one place, the available attack surface has massively increased. A single vulnerability in one service is all it would take to compromise every other service, application and data store on the server.

Cross-site contamination... When one application is compromised, others on the same server can be as well.

This type of attack also complicates incident response, as discovering which services have been subsequently affected can be very difficult. It can also lead to re-infection, as subtle backdoors may be planted in the other applications.

Fortunately, virtualisation technology is common and aspects of it are designed for this very purpose. By using solutions such as Docker containers or carefully planned AWS networks, it is possible to make the most of your server space without compromising security.

3 Never Trust Defaults

Everyone knows that default passwords are a no-go. They are publicly available, attacks are easy to automate and they are likely the first thing an attacker will try.

When it comes to web applications, there are plenty of other defaults that are often overlooked. Unfortunately, some can have catastrophic consequences. Each depends on the web stack you are using. However, in general, keep an eye on:

Default plugins. Certain CMS and web servers come with plugins or modules pre-installed. These are often harmless when not in use, but in some cases can provide attackers with the foothold they need. Make sure to verify every enabled plugin or module is in active use and disable those that are unnecessary.

TLS configurations. Transport Layer Security is a family of encryption protocols used to prevent HTTPS traffic being monitored and/or modified. Enabling it is easy, but ensuring it is configured as securely as possible needs care. For specifics on how to make these changes, look at the advice provided by your web server's maintainers.

Logs. Every framework and web server have some provision for log keeping. You will find that many are disabled by default. Hopefully, you will never need them. However, in the event of a security breach or catastrophic error, they can be a huge time saver.

It is not possible to list every insecure default and recommendation here. To be safe, we recommend doing a search for hardening guides for each piece of software in your stack. A good example for Apache can be found here.

Make sure to stay vigilant and check your sources.

4 Web Application Firewalls

A Web Application Firewall, or WAF, is one of the most effective methods of filtering malicious traffic.
A WAF will typically sit between your web server and the Internet. Its primary function is to intercept and assess traffic before it even reaches your web server.

WAFs can provide many protections, including:

Filtering of traffic containing payloads for attacks such as XSS & SQL injection.

Distributed Denial of Service (DDoS) protection, ensuring your web application remains available.

Bruteforce protections, implementing blocking at the IP level.

In some cases, WAFs are updated even before vulnerable software is. Ensuring affected web applications are protected automatically. Take, for example, the Drupalgeddon 2 vulnerability, which Cloudflare's WAF was able to mitigate even when the underlying framework was vulnerable.

WAFs are often simple to implement. In many cases they require a DNS change and a small amount of configuration. Alternatively, custom hardware or software implementations can be created.

5 Keep Updated

We have left the most important point till last.

Keep up-to-date with updates. A good place to start is to go to the software vendor's online support area.

This goes for everything in the stack. From CMS plugins to your server's operating system.

A lack of updates is perhaps the most common reason web applications are compromised. Certainly, some of the most successful worms and automated attacks exploit issues which were patched long ago, and in a perfect world would not be exploitable.

The importance of this can perhaps be demonstrated by the number of security updates released by Wordpress.

The use of a WAF may help mitigate issues caused by out-of-date software but, as with all defence-in-depth measures, it is best to make sure you're protected at every possible level.