

# Keeping Web and Database Servers Secure

Archived: 2026-04-06 03:10:25 UTC

Various high-profile hacking attacks have proven that [web security](#) remains the most critical issue to any business that conducts its operations online. Web servers are one of the most targeted public faces of an organization, because of the sensitive data they usually host. Securing a web server is as important as securing the website or web application itself and the network around it. If you have a secure web application and an insecure web server, or vice versa, it still puts your business at a huge risk. Your company's security is as strong as its weakest point.

Although securing a web server can be a daunting operation and requires specialist expertise, it is not an impossible task. Long hours of research and an overdose of coffee and take away food, can save you from long nights at the office, headaches and data breaches in the future. Irrelevant of what web server software and operating system you are running, an out of the box configuration is usually insecure. Therefore one must take some necessary steps in order to increase web server security. Below is a list of tasks one should follow when securing a web server.

## 1. Remove Unnecessary Services

Default operating system installations and configurations, are not secure. In a typical default installation, many network services which won't be used in a web server configuration are installed, such as remote registry services, print server service, RAS etc. The more services running on an operating system, the more ports will be left open, thus leaving more open doors for malicious users to abuse. Switch off all unnecessary services and disable them, so next time the server is rebooted, they are not started automatically. Switching off unnecessary services will also give an extra boost to your server performances, by freeing some hardware resources.

## 2. Remote access

Although nowadays it is not practical, when possible, server administrators should login to web servers locally. If remote access is needed, one must make sure that the remote connection is secured properly, by using tunneling and encryption protocols. Using security tokens and other single sign on equipment and software, is a very good security practice. Remote access should also be restricted to a specific number of IP's and to specific accounts only. It is also very important not to use public computers or public networks to access corporate servers remotely, such as in internet café's or public wireless networks.

## 3. Separate development / testing / production environment

Since it is easier and faster for a developer to develop a newer version of a web application on a production server, it is quite common that development and testing of web applications are done directly on the production servers itself. It is a common occurrence on the internet to find newer versions of a specific website, or some content which should not be available to the public in directories such as /test/, /new/ or other similar sub directories. Because such web applications are in their early development stages, they tend to have a number of

vulnerabilities, lack input validation and do not handle exceptions appropriately. Such applications could easily be discovered and exploited by a malicious user, by using free available tools on the internet.

To ease more the development and testing of web applications, developers tend to develop specific internal applications that give them privileged access to the web application, databases and other web server resources, which a normal anonymous user would not have. Such applications usually do not have any kind of restriction, since they are just test applications accessed that should be accessed from the developers only. Unfortunately, if development and testing is done on a production server, such applications can easily be discovered from a malicious user, which could help him compromise and gain access on the production server.

Ideally, development and testing of web applications should always be done on servers isolated from the internet, and should never use or connect to real life data and databases.

## **4 .Web application content and server-side scripting**

The web application or website files and scripts should always be on a separate partition or drive other than that of the operating system, logs and any other system files. Through experience we've learnt that hackers who gained access to the web root directory, were able to exploit other vulnerabilities, and were able to go a step further and escalate their privileges to gain access to the data on the whole disc, including the operating system and other system files. From there onwards, the malicious users have access to execute any operating system command, resulting in complete control of the web server.

## **5. Permissions and privileges**

File and network services permissions play a vital role in web server security. If a web server engine is compromised via network service software, the malicious user can use the account on which the network service is running to carry out tasks, such as execute specific files. Therefore it is very important to always assign the least privileges needed for a specific network service to run, such as web server software. It is also very important to assign minimum privileges to the anonymous user which is needed to access the website, web application files and also backend data and databases.

## **6. Install all security patches on time**

Although having fully patched software does not necessarily mean your server is fully secure, it is still very important to update your operating system and any other software running on it with the latest security patches. Up until this day, hacking incidents still occur because hackers took advantage and exploited un-patched servers and software.

## **7. Monitor and audit the server**

All the logs present in a web server, should ideally be stored in a segregated area. All network services logs, website access logs, database server logs (e.g. Microsoft SQL Server, MySQL, Oracle) and operating system logs should be monitored and checked frequently. One should always be on the lookout for strange log entries. Log files tend to give all the information about an attempt of an attack, and even of a successful attack, but most of the

times these are ignored. If one notices strange activity from the logs, this should immediately be escalated so the issue can be investigated to see what is happening.

## **8. User accounts**

Unused default user accounts created during an operating system install should be disabled. There is also a long list of software that when installed, user accounts are created on the operating system. Such accounts should also be checked properly and permissions need to be changed required. The built in administrator account should be renamed and is not to be used, same for the root user on a linux / unix installation. Every administrator accessing the web server should have his own user account, with the correct privileges needed. It is also a good security practice not to share each others' user accounts.

## **9. Remove all unused modules and application extensions**

A default Apache installation has a number of pre-defined modules enabled, which in a typical web server scenario are not used, unless they are specifically needed. Turn off such modules to prevent targeted attacks against such modules.

The same applies for Microsoft's web server; Internet Information Services. By default, IIS is configured to serve a large number of application types, e.g. ASP, ASP.NET and more. The list of application extensions should only contain a list of extensions the website or web application will be using. Every application extension should also be restricted to use specific HTTP verbs only, where possible.

## **10. Use security tools provided with web server software**

Microsoft released a number of tools to help administrators secure IIS web server installations, such as URL scan. There is also a module called mod\_security for Apache. Although configuring such tools is a tedious process and can be time consuming, especially with custom web applications, they do add an extra bit of security and piece of mind.

## **11. Stay informed**

Nowadays, information and tips on the software and operating system being used can be found freely on the internet. It is very important to stay informed and learn about new attacks and tools, by reading security related magazines and subscribing to newsletters, forums or any other type of community.

## **12. Use Scanners**

Scanners are handy tools that help you automate and ease the process of securing a web server and web applications. [Acunetix Web Vulnerability Scanner](#) is also shipped with a port scanner, which when enabled will port scan the web server hosting the web application being scanned. Similar to a network security scanner, Acunetix WVS will launch a number of advanced security checks against the open ports and network services running on your web server.

Acunetix Web Vulnerability scanner ensures website and web server security by checking for [SQL Injection](#), [Cross site scripting](#), web server configuration problems and other vulnerabilities. It checks password strength on authentication pages and automatically audits shopping carts, forms, dynamic Web 2.0 content and other web applications. As the scan is completed, the software produces detailed reports that pinpoint where vulnerabilities exist.

## Frequently asked questions

---

Source: <https://www.acunetix.com/websitesecurity/webserver-security/>