My site is Hacked – How do I secure it?

Update Web Applications Regularly

First thing you should do is to check vendor/developer websites for all of the web scripts/applications (e.g. WordPress, Joomla, Drupal etc.) used in your account for any updates. This includes any addon modules you may be using in any web applications. If you are using any open source web application, that may be the prime suspect. However, you must check all and keep them up to date.

Search on google or security related websites e.g. www.secunia.com for any known exploits in public knowledge for any web application in your use.

For example if you are using WordPress or Joomla, you must get yourself registered in their mailing lists and update to latest stable release or whenever they release a security patch. The window of opportunity for hackers should be kept to a minimum. We do this for a number of websites that we manage as managed accounts and they have never been hacked in 14 years.

Remove unwanted files and folders

Once you have verified that 100% of scripts are latest stable, you will need to go through all files of your account and make sure none is uploaded by hackers before you audited. Remove any unattended install of any applications. For example if you installed a web app to test it out and forgot to remove it.

Go through all files in your folders and check for timestamp of file changes. There may be files in folders you would never imagine. You can use ftp or cpanel file manager to go through all files under public_html and compare them with your local copy. [You should always maintain a local copy for this comparison as well as backup] – especially check any modified files for any code injected into it.

Typical locations where hackers install malicious scripts are in images folder or web app upload folder.

Password Strength

Make sure all passwords are mix of alpha-numeric and not a dictionary word. Any dictionary word is likely to be unsafe. Reset password if you are hacked, including email, database, web app logins, and any control panel. Hacker may have already scanned all your files to pick e.g. database password for future attacks.

Databases

The MySQL database access to all web application should be using separate db users. Do not ever use your main account user/pass for it. Your main user/pass should never be stored in any file in your account.

Raw Log Manager

In your control panel, activate archive option of your web logs in Raw Log Manager. This will give you the opportunity to check how the hacker exploited one of the scripts. Otherwise all raw logs are cleared after generating stats. If you have already been hacked, it's too late now but you can archive the logs for future attacks.

Customised Modules

If you have customized a web application with modifications or a mod (plugin), make sure it is also latest stable. Many popular web applications may be stable but one of the addon mods could be exploitable; or may not be maintained any more. Only use well maintained code in your account.

Check your own code

If you have written some code yourself, make sure all input variables are sanitised (checked for valid data before using it). Otherwise a single line of bad code can give access to your entire account. The usual mistake is to (a) include a file based on user input (b) passing the data as it is to database or other scripts without sanitising it. Again, make sure all input to a script is checked for valid data. All exploits are based on input data. If your site does not take any input, you are 100% safe from web exploits, i.e. if you run 100% static html site with no script whatsoever anywhere in your account.

Register_globals

For php, any application that uses register_globals to be active has more chances of being exploitable. Avoid such applications. In latest php, register_globals are no longer active so this type of exploit is going down.

eMail Script Practice

If you have some email script, make sure it is safe from header injection. In essence make sure that email address, subject and other part of data that is being submitted by user does not contain line breaks. If any line break comes in, the script should block such attempts. With such header injection spammers can use your account and server to send huge spam.

Open Source Updates

Using open source free web applications is great but you have to maintain it by regular updates or you can lose all your data and site if a new exploit is known about it. And as a hosting account owner, it is your responsibility that you maintain such applications and keep your account protected.

Luck

If your site has been running fine for years, it does not mean there were no security holes in it. It actually means that exploit was unknown or you were lucky that no one exploited it before.

Permissions

If you are on a shared hosting account, for added security, change the permissions of your configuration files (having database credentials etc.) to 660. You can do that via ftp or file manager. For example via ssh: *chmod 660 config.php*

Admin Restricted Access

Again for additional security, if you can block access to certain administrative sections of your site, do that by giving access to only authorized IP addresses and blocking access for everyone else, Or password protect it. This can be done using .htaccess file or Password Protected Folders.

File Uploads

If there is any file upload facility in your account, make sure that only authorized members can use it.

Also the uploaded file should not be accessible via web URL directly (i.e. stored outside of public_html) unless

a) It is only uploaded by a site admin (responsible person)

b) Checked and validated that it does not contain malicious data.

Test Sites

If you're exploring something by installing a test instance of a web app, or you are in the process of developing a new app, lock it behind password or IP access right away.

World Write Permissions

Since our servers come with suphp, you do not need any file or folder with world write permissions. The normal folder permissions should not exceed 755. And php/html files can be 644. CGI/perl scripts can be 755.

Educate Web Developers/Programmers About Security

Anyone who writes web application code should be familiar with security. Here is a book that covers the web application security particularly on php: http://www.oreilly.com/catalog/phpsec/ we recommend it to all. It covers all aspects of vulnerabilities found today in web applications. Remember, one single line of bad code can give access to your entire account. Writing code is easy but writing secure code needs awareness. This is not a problem of php or server. It is lack of security awareness and education. It should be high priority in a web development project.

Site Check

Log in to sucuri.net (or other site checker) to check your site for errors. (Contact us if you would like to use this software at our discounted rate)

P O Box 321 Brunswick Heads NSW 2483 Australia
ABN 37 175 432 807
Phone: 1300 301 990 / **Fax:** (02) 9475-0070
Email: support@oceaniawebhosting.com.au

Oceania
Business Solutions