Some basic website security rules

Over the past week, a series of websites and server systems that have been attacked, compromised and stolen data have raised concerns from end users. How response solutions?

You might think that your site has nothing to hack. But practically any website is at risk of being violated. When the website has vulnerabilities, hackers can easily penetrate, attack, and exploit data to make the website infected that is dangerous not only for website owners but also for visitors. The majority of site security violations are not to steal data or destroy site layout, but to use the site's server to forward spam emails or set up temporary, regular web servers to serving illegal files. The compromised machines can be turned into a part of the botnet, to dig Bitcoin, or use ransomware to attack victims.

Artwork: Internet

After the website has been decontaminated, if the administrator is still subjective not to take care of these vulnerabilities regularly, the website will be easily poisoned again at any time. To prevent this from happening, appropriate methods are needed to protect the web server as well as the administrator's computer when connecting to the server account.

The easiest way to protect the website is to ensure the safety of the website against the bad guys. But when you learn about web security vulnerabilities, you will face complex concepts and solutions. Understanding that, TipsMake.com has a collection of simple, basic but effective tips to help you improve the security of your website. Some of them are advice from Marta Janus, Network Security Research Expert, Kaspersky Lab.

Useful tips for website safety

1. Use strong passwords
2. Constantly updated
3. Create backups
4. Scan files regularly
5. Interested in computer security
6. Enhance the security level of the server
7. Human factors still play the most important role
8. Beware of SQL injection
9. Protection against XSS attacks
10. Watch out for error messages
11. Authenticate both sides
12. Avoid uploading files
13. Use HTTPS
14. Add website security tool
15. Encrypt login page
16. Use secure servers
17. Keep the site 'clean'
18. Hire a security specialist

Use strong passwords

At first glance, this rule is quite normal but using strong passwords is the basic foundation to help enhance the security of the server system. Passwords are not only required to change after an incident but need to change often, preferably once a month.

Password is the first gateway to sensitive data sources -*(Photo: Internet)*

A strong password needs to meet the basic criteria. Secure passwords must be integrated between letters, numbers and special characters but must be easy to remember so that they do not have to note this password to a notebook or computer, do not use the same password for multiple accounts different as email, bank account, etc. Users can refer to additional criteria for a strong password.

Constantly updated

In order to improve the security level, users need to update their website regularly, especially tracking new version information if they are using open source web software (CMS, portal, forum .).

All software that the user manages with the server account must be the latest version and all security patches need to be applied as soon as it is released. This will reduce the risk of an attack on data mining. A list of commonly vulnerable vulnerabilities can be viewed here.

Create backups

A backup of all server contents is not *"poisoned"* will certainly help users save a lot of time and effort when recovering. A recent copy will be very helpful in solving arising problems as well as in the case of a server or website being poisoned.

Scan files regularly

Scan even if no signs of poisoning are found. This is a very useful operation to protect the website, scan all files on the server for a certain period of time at least once.

Interested in computer security

Many malicious malware attacks websites and is spread by infecting computers. Therefore, the security of the server hosting the website is one of the most important aspects of website security. Keeping the server in a safe and uninfected state will improve the level of security as well as avoid malware attacks.

Enhance the security level of the server

If you own the server system, users need to pay attention to the configuration of the machine to ensure the safest level possible. Activities to enhance server security include the following sections:

1. Remove all unused software
2. Disable all unnecessary services and modules
3. Set appropriate policies for users and groups
4. Set access / restrict access to certain files and folders
5. Disable direct directory browsing
6. Collect activity log files, regularly check suspicious activities
7. Use encryption and secure protocols

Human factors still play the most important role

"Every network has its own problems and weaknesses. However, the critical weakness," said Kaspersky Lab's Southeast Asia security expert in an interview with Digital Life . *especially people, network administrators must constantly improve, learn new knowledge, increase awareness of threats and security to promptly identify technical gaps and avoidance must make mistakes in network administration ".*

Mr. Jimmy said that any engineer can make mistakes in the start-up phase due to lack of experience and advice for young engineers that " *don't be afraid to learn from the previous one, if you don't learn from people with experience and mistakes, you can hardly develop yourself in your career. "*

Beware of SQL injection

SQL injection attacks are when an attacker uses a web form field or a URL parameter to have access to or manipulate the victim's database. When using standard Transact SQL, it's easy to insert fake code into the user's query, thereby changing the table, getting information and deleting the data. This can be easily prevented by always using parameterized queries. Most web languages ??have this feature and it is very easy to implement.

Consider the following query:

```
"SELECT * FROM table WHERE column = '" + parameter + "';"
```

If the attacker changes the URL parameter to **'or' 1 '=' 1** , that will cause the query to look like this:

```
"SELECT * FROM table WHERE column = '' OR '1'='1';"
```

Since **'1' is** equal to **'1',** this will allow the attacker to add an additional query to the end of the SQL statement and it will also be executed.

You can fix this query by parameterizing it explicitly. For example, if you are using MySQLi in PHP, the query will become:

```
$stmt = $pdo->prepare('SELECT * FROM table WHERE column = :value'); $stmt->execu
```

Protection against XSS attacks

Cross-site scripting (XSS) transmits malicious JavaScript to web pages, then runs in the user's browser and can change the page content, or steal information to send back to the attacker. For example, if a comment is displayed on a page without authentication, the attacker can send comments that contain script and JavaScript tags, run in every other user's browser and steal their login cookie, then take control of the account of every user who has viewed the comment. Make sure that users cannot include active JavaScript content on websites.

This is of particular interest in modern web applications, where pages currently built primarily from user content and creating HTML are also compiled by front-end frameworks like Angular and Ember in many cases. . These frameworks provide many protection against XSS, but combining server and client rendering also creates new and more complex attacks, not only effectively transmitting JavaScript into HTML, but You can also transfer content that will run code by inserting Angular commands or using Ember helper.

The key here is to focus on how user-generated content is. This is similar to protection against SQL injection. When creating dynamic HTML, use functions that explicitly perform the changes that are looking for (for example, use the **.setAttribution** and **Element.textContent** elements to automatically **exit** the browser, instead of the manual element.innerHTML implementation. or use automatic escape functions when appropriate, instead of concatenating strings or placing raw HTML content.

Another powerful tool against XSS is the Content Security Policy (CSP). CSP is a returnable server header, allowing the browser to limit how and what JavaScript is executed in the page, for example, not allowing any script not stored on the domain, not allowing JavaScript inline or disable **eval ().** Mozilla has a great tutorial with some example configurations (refer to developer.mozilla.org/en-US/docs/Web/HTTP/CSP). This makes the attacker's scripts harder to operate, even if an attacker can put them into the site.

Watch out for error messages

Be careful with the amount of information you provide in error messages. Only provide minimal errors to the user, to ensure they do not leak secrets that are on the server (for example, API key or database password). Do not provide complete exception details because these can make complex attacks like SQL injection made much easier. Keep detailed errors in server logs and show users only the information they need.

Authenticate both sides

Authentication must always be performed both on the browser and on the server side. The browser may encounter simple errors such as when the required fields are blank or enter text in the field just for entering numbers. However, these can be ignored and should ensure the verification of deeper server-side authentication. Failure to do so may result in malicious code or scripts being inserted into the database or may cause undesired results in the site.

Avoid uploading files

Allowing users to upload files to the site can bring great risks to the site, even if simply changing the avatar. The risk lies in any file that is uploaded, although it looks harmless, it may contain a script that when executed on the server will "expose" the site completely.

If uploading files is a must, be wary of everything. If you allow users to upload images, it is not possible to rely solely on the file extension to verify that it is an image file because they can be easily tampered with. Even opening files and reading titles or using functions to check image sizes cannot be absolutely safe. Most image formats that allow storing a comment section can contain PHP code executed by the server.

So what can be done to prevent this? Prevent users from executing any files they upload. By default, web servers will not try to execute image extension files, but cannot rely solely on checking file extensions, because a file named **image.jpg.php** can be easy. 'breaking rule'.

Some options are to rename the file when uploading to make sure the file extension is correct or to change the file permissions, for example, chmod 0666 so that it cannot be executed. If using * nix, users can create a **.htaccess** file **,** allowing only access to a set of files that prevent the use of the dual extension mentioned earlier.

```
deny from all order deny,allow allow from all
```

Finally, the proposed solution is to prevent direct access to uploaded files. In this way, all files uploaded to the website are stored in a folder outside the webroot or in a blob database. If the files are not directly accessible, you will need to create a script to fetch files from your own directory (or the HTTP handler in .NET) and send them to the browser. The **img tag that** supports the src attribute is not a URL directly to the image, so the src attribute can point to the file distribution script, providing the correct content type in the HTTP header. For example:

Good web hosting providers handle the server configuration for you, but if you are hosting a website on your own server, there are a few things to check.

Be sure to have a firewall set up and all unnecessary ports are blocked. If possible, set the DMZ (Demilitarized Zone) to allow access only to ports **80** and **443** from the outside. Although this may not be possible without access to the server from the local network, ports will need to be opened to allow uploading files and remote login to the server via SSH or RDP.

If you allow files to be uploaded from the Internet, use only safe transport methods to your server such as SFTP or SSH.

If possible, the database will run on a different server than the web server. Doing this means that the database server cannot be accessed directly from the outside, only the web server can access it, minimizing the risk of data being exposed.

Finally, don't forget to restrict physical access to the server.

Use HTTPS

HTTPS is a protocol used to provide security over the Internet. HTTPS ensures that users are talking to the server they expect and no one else can block or change the content they are seeing in the process.

If there is any information that the user wants to be private (such as credit cards and login pages, as well as incoming URLs), only HTTPS should be used to transmit information. For example, a login form will set a cookie, which is sent along with all other requests to the site that the user has logged in, and then used to

authenticate those requests. An attacker who steals this can impersonate the user perfectly and take over the login session. To beat these types of attacks, use HTTPS for your entire website.

That is no longer difficult or expensive. Let's Encrypt provides free and automatic certificates, activates HTTPS and has existing community tools for many common platforms and frameworks to automatically set this up for users.

Notably, Google has announced that it will increase the rankings in the search rankings if it uses HTTPS (which also benefits SEO). Unsafe HTTP is slowly disappearing and now is the time to upgrade.

If you already use HTTPS, consider setting up HTTP Strict Transport Security (HSTS), a simple header to add server feedback to prevent unsafe HTTP from running across the entire domain.

Add website security tool

When I thought I had done everything I could, it was time to check the website security. The most effective way to do this is through the use of some site security tools, often called penetration testing (pen testing).

There are many free and commercial products that support this. They work on the same basis as hackers' scripts in that they check all exploitation activities and try to attack the site, using some of the previously mentioned methods such as SQL Injection.

Some free tools worth considering are:

1. **Netsparker.com** (Free community version and trial version available). Suitable for checking SQL injection and XSS
2. **OpenVAS.org:** Self-recognition is the most advanced open source security scanner. Suitable for checking known vulnerabilities (more than 25,000). But it can be difficult to install and requires installing OpenVAS server only running on * nix. OpenVAS is a branch of Nessus before it becomes a closed source commercial product.
3. **SecurityHeaders.com** (check online for free). A tool to quickly report the above mentioned security headers (such as CSP and HSTS), a domain has been turned on and configured correctly.
4. **Xenotix XSS Miningit Framework (xenotix.in)** . A tool from OWASP (Open Web Application Security Project) includes a lot of examples of XSS attacks, users can run to quickly confirm whether the site inputs are vulnerable in Chrome, Firefox. and IE or not.

Results from automated tools can be daunting, because they show a lot of potential problems, focusing on important issues first. Each reported issue often comes with a clear explanation of the potential vulnerability. Users may find that some low / medium alert issues are not a concern for the site.

There are several ways you can try to attack your own website, such as changing **POST / GET** values **.** A debugging proxy can be well supported in this case because it allows blocking the HTTP request values ?? between the browser and the server. Fiddler software is a good starting point for doing this.

If the website has a page that is only visible to logged-in users, try changing the URL parameters such as user ID or cookie value to try to view details of other users. Another area of ??worth testing is forms, changing POST values ??to try to send code that executes XSS or uploads to server-side scripts.

Encrypt login page

Use SSL encryption on login pages. SSL allows sensitive information such as credit card numbers, social security numbers and login information to be transmitted securely. The information entered on a page is encrypted so that it is meaningless to any third party. This helps prevent hackers from accessing login information or other private data.

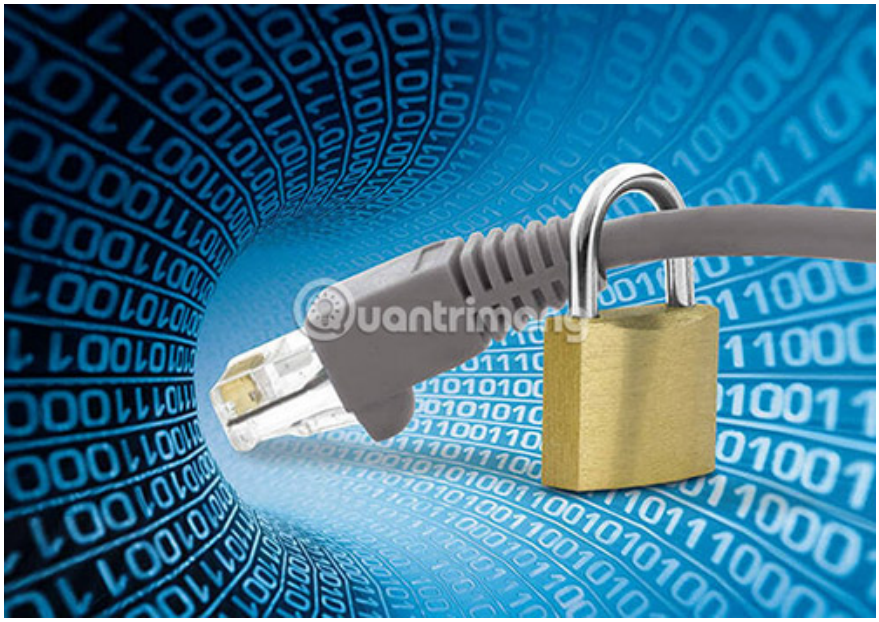
Use secure servers

Choosing a safe and reputable web hosting company is very important for website security. Make sure the server you choose recognizes threats and can keep the site safe. The server should also back up the data to the remote server and can easily be restored in case the site is hacked. Please choose a server that provides ongoing technical support regardless of day and night.

Keep the site 'clean'

Every database, application or plugin on the website is an exploit point in the eyes of hackers. Users should delete files, databases or applications that are no longer used from the site. It is important to keep the file structure well organized to track changes and make it easier to delete old files.

Hire a security specialist



Developing a relationship with a security service company can be a relief in protecting the site. Although users can handle small things themselves, there are many security measures that need to be handled by an expert. Security service companies can often scan websites to find vulnerabilities, perform full site security checks, monitor malicious activity and be present whenever problems need to be fixed. cure.

Users must always be vigilant in protecting websites and these practical advice are just the most basic methods. Never stop looking for protection measures for the site. Don't let the bad guys win!

See more:

1. Don't ignore these 10 security tips when creating a new website
2. Enhance the effectiveness and security of Website with CloudFlare
3. How to evaluate and improve security for a website