Mathematical Problems of Computer Science 42, 81--84, 2014.

Research and Deployment of Improved Web Server Protection Methods

Arthur S. Petrosyan and Gurgen S. Petrosyan

Institute for Informatics and Automation Problems of NAS RA
e-mail: arthur@sci.am, gurgen@sci.am

Abstract

Since the World Wide Web service remains the most widely used internet service, the protection of Web Servers becomes more and more important, especially in view of vulnerability, being found from day to day. This article describes the research work done in Academic Scientific Research Computer Network of Armenia (ASNET-AM) managed by the Institute for Informatics and Automation Problems (IIAP) of the National Academy of Sciences of the Republic of Armenia (NAS RA), targeted to the research and deployment of the improved methods for Web server protection. Special attention was given to obtaining the best solution for Web server protection methods in Apache/PHP-based Shared Web Hosting Environment.

Keywords: WWW, Web Hosting Environment, Web Server, Apache, PHP.

1. Introduction

In a Shared Web Hosting Environment, different hosted websites share the whole server but each client has its own set of resources. A number of websites share a single server. It is an economic solution for those websites which do not have high traffic and high storage requirements. In such shared hosting environment everything that interacts with a server is a threat to any of the hosted websites.

The following analogy helps understanding the issues of shared hosting environment. A shared server is like an apartment building, where all the resources like water supply, power supply, parking lot, etc. are shared with other people in the apartment building. In case of water or power supply failure, every apartment faces the impact. Similarly in shared web hosting, the web server software (Apache HTTP Server Project [1]), requires a control over the files to be served to the client which immediately poses a security concern. If the domains have an ability to run scripts or if the domains have an access to the shell, then in shared hosting environment, one client can modify the files of another client. Though in a multiuser operating system like Linux read/write/execute, privileges can be provided to different user groups (user/group/other) yet through a simple PHP script, files outside the own home directory can be accessed. This is because in default web server configuration all hosted websites are served from the same username/UID and, thus, have the same privileges at the operating system level. Even when using pre-packaged software solutions, you need to allow the hosting server to have read, write and execute access to your files and, thus, expose vulnerability to other clients.

Moreover, though the PHP functions like exec(), shell_exec() provide flexibility to the developers, yet they pose adverse security problems. Most of the websites require some image uploads from the web and if the client on shared hosting does not have a server permission then these uploads will not move to the destination directory. The common solution is to give all the users 777 (read/write/execute) access to the destination directory. This is a common solution but what it has provided is an easy way to hack the files of other users sharing the same server.

2. Restriction on PHP Level

An important PHP feature, which can be helpful in shared hosting environment, is ***open_basedir***. It can be used to limit the files that can be accessed by PHP to the specified directory-tree, including the file itself. When some PHP script tries to access the filesystem, for example using include, or *fopen(),* the location of the file is checked. When the file is outside the directory-tree, specified within the '*open_basedir*', PHP will refuse to access it.

The typical '*open_basedir*' setting should be:

```
open_basedir = /tmp/:/var/www/
```

This will protect the other system directories from accessing by any PHP script.

But the problem is that the '*open_basedir*', which is generally set in the PHP global configuration file '*php.ini*', can be overridden in the local .htaccess file at the directory level, in case the Apache web server 'AllowOverride' setting is set to 'All' or at least 'Options', and even within the PHP script (using the function 'ini_set()'). Thus, the use of '*open_basedir*' in this configuration does not produce the desired effect.

Also setting '*open_basedir*' in the 'php.ini' file globally for the entire web server, will only help to protect external system directories. But the neighboring websites using the same shared web server will still be able to access each other's directories at PHP script level. It means that in case the website is broken, other website data will be potentially available to the intruder too.

The solution to the above issue is to define '*open_basedir*' as so called ***php_admin_value***. PHP documentation shows the settings defined as '*php_admin_value*' can't be overridden in '*.htaccess*'or '*ini_set()*' [2]. Even more, '*open_basedir*' can be defined as '*php_admin_value*' for each virtual host separately, so the issue mentioned above will be solved and we will get a separate '*open_basedir*' setting, determined for each website (virtual hosting), which can't be changed from '*.htaccess'* or '*ini_set()*'.

In this way it is possible to determine a separate PHP scripts execution area for each hosted website, so that it does not have access to the whole system, and even to files of neighboring websites using the same shared web server.

Although PHP Developer Team is mentioning that the use of '*open_basedir*' feature is "a convenience to system administrators and should in no way be thought of as a complete security framework" [3], using the '*open_basedir*' helps to limit the potential threat area.

3. Filesystem Permissions Level Protection

PHP level restrictions described above are good enough in case of proper configuration. But to obtain the best solution for Web server protection in shared hosting environment it would be important to have additional protection on the filesystem permissions level. This task can be implemented in different ways and as a result of research work done in ASNET-AM to deploy the improved methods for Web server protection the package Apache 2 ITK MPM was chosen to be most effective [4]. Apache 2 ITK MPM (just *mpm-itk* for short) is a Multi-Processing Module (MPM) for the Apache web server. *mpm-itk* allows to run each of the virtual websites under a separate UserID (UID and GroupID (GID). This means, that a desired additional protection on the filesystem permissions level can be obtained.

The typical ‘mpm_itk_module’ setting for each virtualhost in Apache configuration should include:

```
<IfModule mpm_itk_module>
AssignUserId user1 group1
</IfModule>
```

This will force Apache wer server to fork a process with 'user1' and 'group1' for serving requests to this website.

As a result the neighboring websites using the same shared web server will not be able to access either each other’s directories, or the system directories, based on the filesystem permissions.

This solution can be counted much more effective, since it provides protection on a system level and, thus, not only PHP, but any scripts and configuration files for one virtual website no longer is accessible for all the others. It means that in case any website is broken, only this website data will be available to the intruder.

An important note on *mpm-itk* usage is that it is based on *prefork* method and not *threads*, (i.e., extra fork is done per request). On one hand, it means that *mpm-itk* can support running a non-thread-aware code (like many PHP extensions) without problems. On the other hand, *mpm-itk* performance is lower when compared with threads. But our decision was to have less performance benefits and gain a more powerful method of protection on the filesystem permissions level.

4. Conclusion

Deployment of the improved methods for Web server protection includes implementation of multilevel security means. Thus, the best solution for Web server protection in Apache/PHP-based Shared Web Hosting Environment can be described as a combination of properly configured *open_basedir* PHP level restrictions and additional protection on the filesystem permissions level with modules like *mpm-itk*.

References

[1] Apache HTTP Server Project, [Online]. Available: http://httpd.apache.org

[2] Description of core php.ini directives, [Online]. Available: http://php.net/manual/en/ini.core.php

[3] A Note on Security in PHP, [Online]. Available: http://php.net/security-note.php
[4] Apache 2 ITK MPM, [Online]. Available: http://mpm-itk.sesse.net/

Submitted 29.07.2014, accepted 27.11.2014.

Վեբ սերվերների պաշտպանության բարելավված մեթոդների հետազոտություն և մշակում

Ա, Պետրոսյան և Գ. Պետրոսյան

Ամփոփում

Քանի որ World Wide Web (Համաշխարհային սարդոստայնի) ծառայությունը առանցքային դեր ունի ներկայիս ցանցային տեխնոլոգիաների ոլորտում, վեբ սերվերների պաշտպանությունը կենսական նշանակություն ունի: Հաշվի առնելով վերջին տարիներին լայն տարածում ստացած վեբ սերվերներին ուղղված հարձակումները, շատ կարևոր են դառնում World Wide Web ծառայության պաշտպանության արդյունավետ մեթոդների մշակումն ու ներդրումը:

Հոդվածում նկարագրված են ASNET-AM Հայաստանի ակադեմիական գիտահետազոտական կոմպյուտերային ցանցում կատարված ուսումնասիրության արդյունքները, որոնց նպատակն է մշակել վեբ սերվերների պաշտպանության արդյունավետ մեթոդներ:

Исследование и реализация улучшенных методов защиты веб серверов

А. Петросян и Г. Петросян

Аннотация

Услуга World Wide Web (Всемирной паутины) является ключевой в современном мире сетевых коммуникаций, поэтому защита веб серверов имеет жизненно важное значение. В связи с различными формами атак на веб сайты, внедрение эффективных методов защиты веб серверов становится очень важным.

В статье описаны исследования, направленные на определение эффективных способов защиты от атак на веб сайты. Представлены результаты научно-исследовательской работы, проделанной в Академической научно-исследовательской компьютерной сети Армении (ASNET-AM), направленные на применение улучшенных методов защиты веб серверов. Особое внимание уделено определению наилучшего решения для защиты веб серверов в среде виртуального разделяемого хостинга, на основе программных пакетов Apache/PHP.