Defeating scanners

Php-Brute-Force-Attack-Detector

By Aung Khant (d0ubl3_h3lix)
Version: Beta
Date: Saturday, June 28, 2008
URL: http://yehg.net/lab/#tools

Overview

Nowadays, in web application mapping, attackers use brute force tools in finding default files, hidden directories and known vulnerabilities as well as fingerprinting third-party components, web server backend components and alike. Almost all web hacking and security assessment tools such as famous Nikto, Nessus, JBroFuzz, OWASP DirBuster, WFuzz, and W3Af have built-in flat databases for relatively comprehensive footprinting web applications.

At some time, developers may tend to use flawed third-party components without checking their vulnerabilities which may have been mentioned in milw0rm.com. Knowing someone is attacking your application is a sound alarm for waking up your security sense. You can proactively watch their tracks or you can stop their activities before they can mess up your critical applications.

How it works

Php-Brute-Attack-Detector notifies you in event of brute forcing attacks by looking at 404 errors per 5 minute routine. During that 5-minute period, there must be no more than 100 requests for 404. 100 is maximum request limit which you can change via config.php. It doesn't look for IP address which attackers can easily spoof and iterate using lists of public proxy servers. It looks only for how many 404 requests have been made.

Notification can be achieved via methods 1) emailing, 2) file logging, 3) logging in system log server. Option 1 and 2 are enabled by default. The third option forces system security auditor notice attack logs and to carry out appropriate actions. The first and the third one take place only once per day no matter how many there are flooded requests. This is to **prevent mail bombing** and **messing up system log servers**.

Unpredictable file names are used in file logging so that attackers cannot retrieve files by other means.

For example, the log file name will be like:

```
2008-06-24,19,30@cc2ef06a87f3dfaaeeaf
```

That means:

```
YYYY-MM-DD,HH,MM@MD5Hash
```

Note: 24-Hour format is used.

To fool most brute-force tools, I applied 200:

```
header("HTTP/1.1 200 OK");
```

In addition, the size and contents of generated page are intentionally randomized to defeat tools like WFuzz which can provide the number of lines and words for assisting attackers in differentiating existent and non-existent stuffs.

This would give attackers a heap of **false positives**.

For defensive purpose, the application can sleep for several minutes, causing some of attackers' tools timing out.

Upon notification and knowing attacks, you SHOULD deny attackers' IPs via .htaccess. You can hack the code if you want to change the logic. For example, you can also implement it for 403 error.

Objectives

- To detect as many web hacking tools processing as possible
- To give false positives to attackers in web application mapping phase
- To lengthen attackers' web application mapping time
- To deter attackers before they go any further

Requirements

- PHP 5
- MySQL 4 >

Installation Guide

Pretty easy.

1) Extract, Upload to your root folder

2) In .htaccess,

- Comment/Remove line like the following:

```
ErrorDocument 404 /missing.html
```

- Write

```
ErrorDocument 404 /yehgdetect/index.php
```

Note: If you rename yehgdetect to something you want, change it accordingly.

3) Open PhpMyAdmin, Export yehgdetect.sql, Then Delete yehgdetect.sql.

4) In yehgdetect/db_info.php

- Modify the following according to your database server, username, password, database name, database table name

```
$db_server = 'localhost';
$db_user_login = 'username';
$db_user_pass = 'password';

$db_name = 'test';
$db_table = 'yehgdetect';
```

5) In yehgdetect/config.php

- Modify Maximum request according to your will.

It means that if BruteForce requests are above max_request number, then notify you or your web server admin.

@line 22

```
$max_request = 100; // per 5 minutes
```

- Modify your/your web server admin's email address

@line 29

```
$to = 'youremail@yourdomain.net,yourwebserveradmin@yourdomain.net';
```

- Modify notify option

@line 35

```
// 0 to no emailing
$emailnotify = 1;
// Log attacks in file, 1 to yes, 0 to no
$logfile = 1;
// Log attacks in system log mechanism or server, 1 to yes,
// 0 to no
// You should discuss with your server admin to enable
$logsys = 0;
```

- Enable anti_attack or not

@line 48

```
// if set to 1, sleep the application for several minutes, causing attackers' tools timing out
$anti_attack = 0;
$sleep_time = 5; // minute
```

Time-out value in some tools can be configured easily but it takes much longer time to complete the scan and make you take defensive measures in advance. But be careful with this option. It may cause DOS to your web server.

7) Make “l0gz” folder writable (Chmod 777)

Warning

As a possible counter-attack, attackers can flood the database within minutes. You need to take immediate action by denying their IPs in .htaccess at least.

Attackers can fool search engines to attack your web sites by listing non-existent links of your site in their sites. That way, search engines see those lists of links pointing to your site and start indexing; hence they may get banned. That is why, before banning certain IPs, be sure to look them up in whois database such as http://networktools.com. If the IPs belong to trustworthy corporations like Google, Yahoo...etc., don’t do banning. However, the countermeasure for this attack is to frequently check in google like link:yoursite.com.

Regularly you need to clean up logs (after backing them up) as Php-Brute-Force-Attack Detector doesn’t automate cleaning process after the records reach to certain number. Although it aims to defeat scanning, it is not suitable for all situations.

Don’t forget to disable it during valid (in-house) security assessment by setting $isdisable to 1 at line 18 of config.php.

Further Help

Send help, bugs, suggestions, and comments to yehg.org[xat]gmail[d0t]c0m.

Thanks for downloading.