UNITECH 2017 GABROVO

INTERNATIONAL SCIENTIFIC CONFERENCE
17-18 November 2017, GABROVO

МЕТОДИ ЗА ЗАЩИТА НА WEB ПРИЛОЖЕНИЯ

METHODS FOR PROTECTION OF WEB APPLICATIONS

Assoc. Prof. Aldeniz Rashidov, PhD
Technical university of Gabrovo

Abstract

Proposed are methods for protection and enhancing the security of the applications in the Internet at Web server level. The methods cover the most popular Web server - Apache HTTP Server and they can be used alone or in combination. The presented methods provide optimal protection of the Web-based modules and applications of the University Information System (UMIS) of Technical University of Gabrovo (http://umis.tugab.bg).

Keywords: Web protection, DDoS, Flood, Brute Force, OpenSSL, SSL.

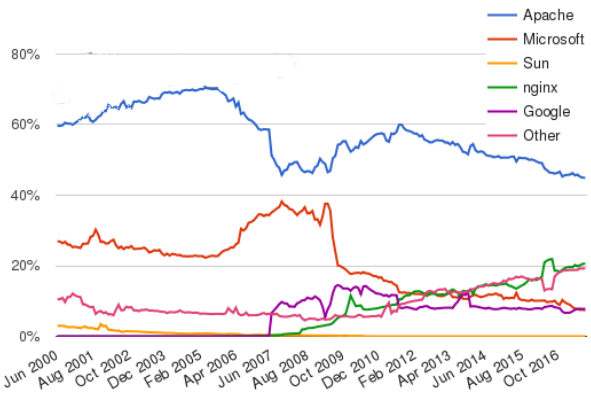
ВЪВЕДЕНИЕ

Онлайн заплахите застрашават ежедневно милиони Web сайтове и други Web базирани програмни приложения по света. Най-използваните и често срещани заплахи в глобалната мрежа са от DDoS атаки, зловредни софтуери, вируси и ботове, които търсят уязвимости в приложенията. Голяма част от Web сайтовете са разработени на базата на Content Management Systems (CMS) като WordPress, Drupal, Joomla, Blogger, Magento, phpBB и др. Това е предпоставка заплахите да бъдат съсредоточени върху сайтове базирани на тези системи. За да се осигури защита и повиши сигурността на един персонален блог, Web сайт или друго Web базирано приложение е необходимо да се търсят и периодично разработват разнообразни методи за защита.

Цел на настоящия доклад е да предложи методи за защита и повишаване на сигурността на програмните приложения в глобалната мрежа Интернет на ниво Web сървър.

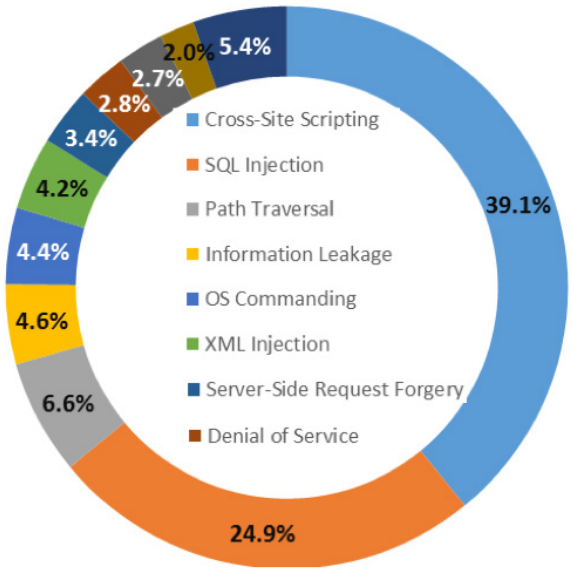
ИЗЛОЖЕНИЕ

Web сървърите заемат основно място при защитата на приложенията, тъй като те са основен компонент, който осигурява публичност на приложенията. Популярни Web сървъри са Apache HTTP Server, Nginx, Microsoft IIS, Sun Java System Web Server и др. (фиг. 1). Според Netcraft [1], която предоставя данни от изследвания и анализи за различни аспекти на Интернет, Apache HTTP Server (Apache) има най-висок пазарен дял на активните сайтове по света до 2017 г. включително (фиг. 1). Над 90% от CMS поддържат този вид сървър. Това е предпоставка да се търсят и разработват предимно методи за защита на програмните приложения достъпни посредством Apache.



***Фиг. 1**. Пазарен дял на активните сайтовете по света*

Positive Technologies [2] съпоставя (фиг. 2) и анализира актуалните заплахи и атаки при Web приложенията. Преобладаваща част от тях са свързани с Web сървъра и неговото конфигуриране по отношение на защита и сигурност.



***Фиг. 2**. Съпоставка на заплахите и атаките при Web приложенията*

- **Използване на символни връзки, като механизъм за защита**

Много от Web сайтовете показват изображения и други документи, които се намират на друго, различно от реалното местоположение във файловата система. В случай, че клиент разгледа първичния код на страница, за да открие дадено изображение, което го интересува, то той ще открие за него връзка от вида /ди-ректория0/директория1/файл.разширение. Така откритата връзка може да се използва за визуализиране на изображението от всеки браузър. Значително ще е по-трудно да се открие (и изтрие) изображението във файловата система на сървъра при вход в системата или чрез FTP сървър, в случай че не е известно реалното местоположение на изображението. Този механизъм на защита на приложенията в сървъра се реализира с помощта на създадените от операционната система символни връзки (Symlinks) или чрез деклариране на псевдо имена с директива Alias при Apache.

Примери за създаване на символни връзки при Linux и Windows:

```
#1. Създаване на символна връзка (/tmp/aldeniz)
# на GNU/Linux:
Ln -s –T /var/www/html  /tmp/aldeniz
#
#2. Създаване на символна връзка при Windows:
Junction –s C:/apache/htdocs/video  D:/video
```

Механизмът за защита с използване на символните връзки при Apache е възможен, ако се извършат допълнителни настройки в основния конфигурационен файл на Apache за следене на символните имена.

Конфигурирането на сървъра посредством създадената по-горе в текста символна връзка C:/apache/htdocs/video към реална физическа директория D:/video се извършва по следния начин:

```
<Directory C:/apache/htdocs>
        # Опция за следване на символни имена:
        Options +FollowSymLinks
        AllowOverride None
        Require all granted
</Directory>
#
# Създаване на допълнително псевдо-име:
Alias /v C:/apache/htdocs/video
#
# Заявките към сървъра с префикси -
# http://домейн/video или http://домейн/v,
# които реално, ще извличат файлове от D:/video.
```

- **Контрол на достъпа на ниво директория на сървъра**

Директива AccessFileName от основния конфигурационен файл на Apache задава име на конфигурационен файл на ниво директория. Ако сървърът открие файл с това име (по подразбиране .htaccess) в директорията, то той прилага конфигурационните команди, дефинирани в него, за директорията. Преди да бъде разрешен достъп до ресурсите на дадена директория се прове-рява дали съществува конфигура-ционен файл на ниво директория. С цел по-висока степен на защита е препоръчително във всеки конфигурационен файл на ниво директория да се зададат права, които забраняват неговото разглеждане:

```
# Защита на htaccess.conf
<Files htaccess.conf>
    Require all denied
</Files>
```

- **Ограничаване на достъпа само за определени хостове и IP адреси**

Директива Require на Apache позволява да се дефинира списък от хостове, на които е позволен достъп до ресурсите на директорията. Хостът може да се идентифицира с пълно или частично име, IP адрес или домейн.

Пример за задаване на адреси, които имат достъп до ресурсите на директория:

```
#1. Конфигуриране на поддиректория
# htdocs/joomla/admin и осигуряване на достъп
# само от хостовете tugab.bg и aldeniz.eu:
<Directory htdocs/joomla/admin>
    Require host tugab.bg aldeniz.eu
</Directory>
```

Пример за задаване на адреси, които нямат достъп до ресурсите на директория:

```
#1. Конфигуриране на поддиректория
# htdocs/joomla/admin и ограничаване на достъп
# от адреси 192.168.1.2 и 192.168.1.5:
<Directory htdocs/joomla/admin>
    Require not ip 192.168.1.2 192.168.1.5
</Directory>
```

- **Ограничаване на достъпа и поддържане на защитени връзки чрез модул mod_ssl на Apache**

Модул mod_ssl е интерфейс на Apache към OpenSSL [3,4]. Той може да се използва както за кодиране на комуникацията между сървъра и клиента, удостоверяване на идентичността на сървъра, така и за ограничение на достъпа до ресурсите на сървъра. Следва преглед на основните етапи при конфигуриране на сървъра за работа с модул mod_ssl при Windows.

```
#1. Изтегляне на Apache с модул mod_ssl.

#2. Инсталиране на Apache.

#3. Копиране на файлове ssleay32.dll и libeay32.
# dll в директория %SystemRoot%\system32.

#4.  Изтегляне и разархивиране на пакета
# OpenSSL (от http://www.openssl.org/source/ в
# произволна временна директория.

#5. Копиране на файла openssl.cnf от временната
# поддиректория \apps в поддиректория \bin на
# Apache.
```

Важен етап при осигуряването на защитени връзки е създаването на заявката за сертификат (Certificate Signing Request):

```
#6. Създаване на сървърен ключ (private key) на
# сертификат чрез openssl.exe във файл server.key:
bin/openssl.exe genrsa -des3 -out server.key 1024
# При създаване на ключа се въвежда:
Enter pass phrase for server.key: ******
Verifying – Enter pass phrase for server.key: ******

#7. Създаване на не криптиран ключ
# (unencrypted key) на сертификата от #6
# във файл server.pem:
bin/openssl.exe rsa –in server.key –out server.pem
# При създаване на ключа се въвежда:
Enter pass phrase for server.key: ******
writing RSA key
# Създаденият ключ трябва да се защити
# внимателно и да е достъпен единствено на
# администраторите на Apache,
# тъй като ще се използва при обмен на ключове!

#8.  Създаване на неподписан сертификат
# (unsigned certificate, CSR) във файл server.csr:
bin/openssl.exe req -config openssl.cnf -new –key
    server.key -out server.csr
# Въвеждат се последователно данните за CSR:

Enter pass phrase for server.key: ******
…
Country Name (2 letter code) [AU]:BG
Locality Name (eg. city) [ ]:Gabrovo
Organization Name (eg., company) [Internet Widgits
    Pty Ltd]:TU Gabrovo
Organizational Unit Name (eg., section) [ ]:
    Automation
Common Name (eg., Your Name) [ ]:Aldeniz
(Common name съответства на името на сървъра)
```

```
Email Address [ ]:admin@domain.com
…
A challenge password [ ]:
An optional company name [ ]:
```

След създаване на неподписания CSR сертификат, той следва да бъде изпратен за подписване от упълномощена за това страна (Certificate authority), която ще гарантира, че собственикът на сертификата е този, за когото се представя. Възможно е създаване на саморъчно подписан (self-signed certificate, SSC) сертификат без да се използват услугите на компания за сертифициране. Много от сайтовете използват Self-signed за проверка на входните данни на потребителите, с цел да се осигури достъп към сървъра само на тези, които имат право за това.

```
#9.  Създаване на саморъчно подписан
# сертификат във server.crt, валиден за 60 дена:
openssl x509 -req -days 60 -in server.csr -signkey
      server.key -out server.crt
# При създаване на сертификата се въвежда:
Enter pass phrase for server.key: ******

#10. Копиране на файловете server.crt и
# server.pem в поддиректория \conf\ssl на Apache.

#11. Редактиране на конфигурационния файл от
# директория \conf\httpd.conf.
# Добавяне на директиви за зареждане на модул
# mod_ssl и четене на конфигурационния файл на
# Apache за SSL връзки.
LoadModule ssl_module modules/mod_ssl.so
<IfModule mod_ssl.c>
      Include conf/ssl.conf
</IfModule>

#12. Редактиране на конфигурационния файл
# ssl.conf от поддиректория \conf\ssl\ и
# конфигуриране на защитен виртуален хост:
<VirtualHost domain.com:443>
…
      DocumentRoot "C:/Apache/htdocs2"
      ServerName domain.com:443
…
</VirtualHost>

#13. Стартиране (рестартиране) на Apache с SSL.

#14. Тестване на SSL връзката:
https://domain.com
```

- **Осигуряване на защита от атаки с модул mod_dosevasive на Apache**

Модул mod_dosevasive [5] на Apache осигурява възможност за ограничаване на достъпа до ресурсите на сървъра за определено време от определени потребители. Това ограничаване се налага, както в случаи на злонамерени действия, така и в ситуации при които от сървъра се извлича прекалено често един и същ ресурс. Посредством mod_dosevasive може да се укаже например, че ако от даден IP адрес се получават повече от 15 заявки за една секунда, то този адрес да бъде блокиран за определено време.

Примерно конфигуриране за работа с модул mod_dosevasive:

```
# Зареждане на модул mod_dosevasive:
LoadModule dosevasive_module
                modules/mod_dosevasive.so
#
<IfModule mod_dosevasive.c>
# Размер (в байтове) на хеш таблицата, която
# обработва заявките подадени към сървъра:
      DOSHashTableSize          16384
# Брой заявки към една страница от един и същ
# IP адрес в определен интервал от време:
      DOSPageCount   5
# Брой заявки към различни страници от един и
# същ адрес в определен интервал от време:
      DOSSiteCount    50
# Интервал от време за DOSPageCount:
      DOSPageInterval 1
# Интервал от време за DOSSiteCount:
      DOSSiteInterval  1
# Интервал от време (секунди) за блокиране на
# IP адрес, нарушил ограниченията на
# DOS..Count:
      DOSBlockingPeriod  5
# E-mail адрес за уведомяване при нарушение:
      DOSEmailNotify none
# Команда за изпълнение при установено
# нарушение:
      DOSSystemCommand      ""
# Списък от адреси, за които не са валидни
# директиви DOS..Count:
      DOSWhiteList    10.3.1.*
# Затваря връзката на блокиран IP адрес (on/off):
      DOSCloseSocket On
 </IfModule>
```

Модулът mod_dosevasive е полезен при защита от следните видове атаки:

- Блокиране на работата на сървъра с голямо количество заявки (Flood);

- Ограничаване или цялостно блокиране на определени услуги (DoS);
- Разпознаване на пароли и други данни на потребителите (Brute Force).

Тъй като mod_dosevasive не е включен в стандартния пакет на Apache, то той трябва да се компилира от първичен код на C и след това копира в директория \modules на Apache.

- **Подмяна на URL адреси чрез модул mod_rewrite на Web сървъра**

Модулът mod_rewrite [6] е включен в стандартния пакет на Apache и реализира съпоставяне, филтриране и подмяна на URL адреси. Той трябва да се зареди посредством директива LoadModule в основния конфигурационен файл на Apache. Модул mod_rewrite намира най-често приложение в ситуации, при които URL адресите трябва да се променят според определени условия или външна за адреса информация, а също така и при търсене на файлове и други данни. Модулът оперира на базата на предварително зададени правила, които се прилагат при получаване на заявка и пренасочват към променен URL адрес. Конфигурирането на Apache с използване на mod_rewrite се извършва с помощта на директивите RewriteEngine, RewriteCond и RewriteRule. Стартирането на модула се извършва в следната последователност:

```
#1. Зареждане на модул mod_rewrite:
LoadModule rewrite_module modules/mod_rewrite.so
#
#2. Разрешаване на механизма за подмяна на URL
RewriteEngine On
```

Съществуват следните приложения на модул mod_rewrite при блокиране и пренасочване на достъпа до сървъра:

- *Блокиране на достъпа според начина на заявка за ресурс:*

Този вид блокиране на достъпа се използва, когато неправомерно дадена Web страница на сървъра се заяви от друга страница, разположена на друг сървър. Обикновено, идентифицирането на такъв сървър се извършва след преглед на журналните (log) файлове.

Пример за блокиране на неправомерен достъп през определен домейн:

```
…
#3. Прихващане на имената на домейните,
# използващи неправомерно сайт на сървъра:
RewriteCond %{HTTP_REFERER}
      cracks\.eu [NC,OR]
RewriteCond %{HTTP_REFERER}
      cracks\.am [NC]
#4. Пренасочване не достъпа до всички ресурси
# към съобщение за забранен достъп – Forbidden:
RewriteRule .* [F]
```

- *Блокиране на достъпа при директно използване на ресурси:*

Този вид блокиране на достъпа (известно като hot linking и bandwidth stealing) се използва, когато неправомерно отделни ресурси (част от Web страница) на сървъра като изображения, скриптове на JavaScript и други се заявят самостоятелно и станат част от друга страница, разположена на друг сървър. При такива случаи връзките към тези ресурси се блокират или пренасочват към невалидни или умишлено заместващи ги ресурси.

Пример за блокиране на достъпа при директно използване на ресурси:

```
…
#3. Изключване на имената на домейните,
# използващи правомерно ресурси на сървъра:
RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{HTTP_REFERER}
      !^http://(www\.)?unitech.tugab.bg/.*$ [NC]

#4. Пренасочване на достъпа до файлове с
# разширения js и css към съобщение с код 403:
RewriteRule \.(js|css)$ [F]

#5. Пренасочване на достъпа до файлове с
# разширения png и jpg към друго изображение:
RewriteRule \.(png|jpg)$
      http://unitech.tugab.bg/null.png [R]
```

- *Блокиране на достъп при трасиране и претърсване на ресурси:*

В Интернет пространството съществуват множество програми, които автоматично претърсват съдържанието на документи и събират от тях данни или трасират сайта и изтеглят цялото му съдържание за други цели. Една част от тях използват събраната

информация добронамерено (индексиране в търсачки), а друга (известни като bad bot) за нанасяне на вреди (спам, откриване на пробиви в защитите). В двата случая сървърът се натоварва допълнително и това може да понижи неговото бързодействие. За предотвратяване на натоварването на сървъра е необходимо програмите, извършващи злонамерено събиране на дан-ни да се идентифицират и блокират.

Пример за блокиране на достъпа при трасиране и претърсване на ресурси:

```
…
#3. Прихващане на имена на програми,
# използващи неправомерно трасиране на ресурси:
RewriteCond %{HTTP_USER_AGENT}
      ^Superbot [OR]
RewriteCond %{HTTP_USER_AGENT}
      ^WebReaper

#4. Пренасочване на достъпа до всички ресурси
# към съобщение за забранен достъп – Forbidden
RewriteRule  ^.* [F]
```

- *Блокиране на достъпа в определен интервал от време:*

В определени случаи се налага сървърът да прекрати приемане на заявки в определен интервал от време. Предпоставка за това може да бъде натоварването на сървъра или засичането на умишлено злонамерени действия в този интервал от време.

Пример за блокиране на достъпа в определен интервал от време:

```
…
#3. Задаване на условие за правилото - интервал
# извън 20÷22 часа:
RewriteCond %{TIME_HOUR}%{TIME_MIN}
      <2000 [OR]
RewriteCond %{TIME_HOUR}%{TIME_MIN}
      >2200

#4. Пренасочване на достъпа до всички ресурси
# към съобщение за забранен достъп – Forbidden:
RewriteRule ^.*$ - [F,L]
```

ЗАКЛЮЧЕНИЕ

Предложени са методи за защита и повишаване на сигурността на програмните приложения в глобалната мрежа Интернет на ниво Web сървър:

- *Използване на символни връзки;*
- *Контрол на достъпа на ниво директория на сървъра;*
- *Ограничаване на достъпа само за определени хостове и IP адреси;*
- *Ограничаване на достъпа и поддържане на защитени връзки посредством OpenSSL;*
- *Защита от атаки посредством модул mod_dosevasive на Apache;*
- *Блокиране на достъп според начина на заявка за ресурс;*
- *Блокиране на достъп при директно използване на ресурси;*
- *Блокиране на достъп при трасиране и претърсване на ресурси;*
- *Блокиране на достъп в определен интервал от време.*

Методите обхващат най-разпространеният сървър - Apache HTTP Server и могат да бъдат използвани както самостоятелно така и в комбинация. За осигуряване на оптимална защита от различни заплахи и атаки на Web приложенията е препоръчително да бъдат включени по-голяма част от представените методи за защита. Разгледаните методи осигуряват оптимална защита на Web базираните модули и приложения на Университетска информационна система (UMIS) на Технически университет – Габрово (http://umis.tugab.bg).

ЛИТЕРАТУРА

[1] September 2017 Web Server Survey, Netcraft, https://news.netcraft.com/archives/2017/09/11/september-2017-web-server-survey.html, Accessed on 09.2017.

[2] Web Application Attack Statistics: Q2 2017, Positive Technologies - learn and secure, http://blog.ptsecurity.com/2017/09/web-

application-attack-statistics-q2.html, Accessed on 09.2017.

[3] OpenSSL, https://www.openssl.org/, Accessed on 08.2017.

[4] Apache Module mod_ssl, Apache, https://httpd.apache.org/docs/2.4/mod/mod_ssl.html, Accessed on 08.2017.

[5] mod_dosevasive for apache 2.2, Apache Lounge, https://www.apachelounge.com/viewtopic.php?t=917, Accessed on 08.2017.

[6] Apache Module mod_rewrite, Apache, http://httpd.apache.org/docs/current/mod/mod_rewrite.html, Accessed on 08.2017.