



Universidad Nacional  
Autónoma de México



1ª jornada para  
webmasters  
de la UNAM

# Buenas prácticas para el manejo de sitios y aplicaciones Web

Por: Ing. José Othoniel Chamú  
Arias

# Buenas Prácticas

- Las aplicaciones Web hoy día se encuentran expuestas a diferentes amenazas, por lo que es responsabilidad del administrador de aplicaciones buscar la manera de mitigar varias de ellas.  
Por ejemplo:
  - Ataques de fuerza bruta en aplicaciones Web.
  - Denegaciones de Servicio del tipo Half Open.
  - Command Injection.
  - Path Transversal.
  - Cross Site Scripting.
  - SQL Injection.
  - Uso de Exploits.
  - Entre otros.
- Es recomendable solicitar el apoyo de personal o áreas expertas en el tema, en caso de que no se cuente con estas competencias.

# Como mitigarlos



- Los ataques de fuerza bruta contra aplicaciones Web, intentan romper la contraseña por medio de múltiples intentos o uso de diccionarios.
- Se puede mitigar implementando herramientas tales como captcha ( Completely Automated Public Turing test to tell Computers and Humans Apart ).





- Para el caso de implementar un **captcha** por imagen, es recomendable verificar que el generador no sea legible por algún programa de reconocimiento de caracteres. Para ello se pueden apoyar de algún sitio tal como:

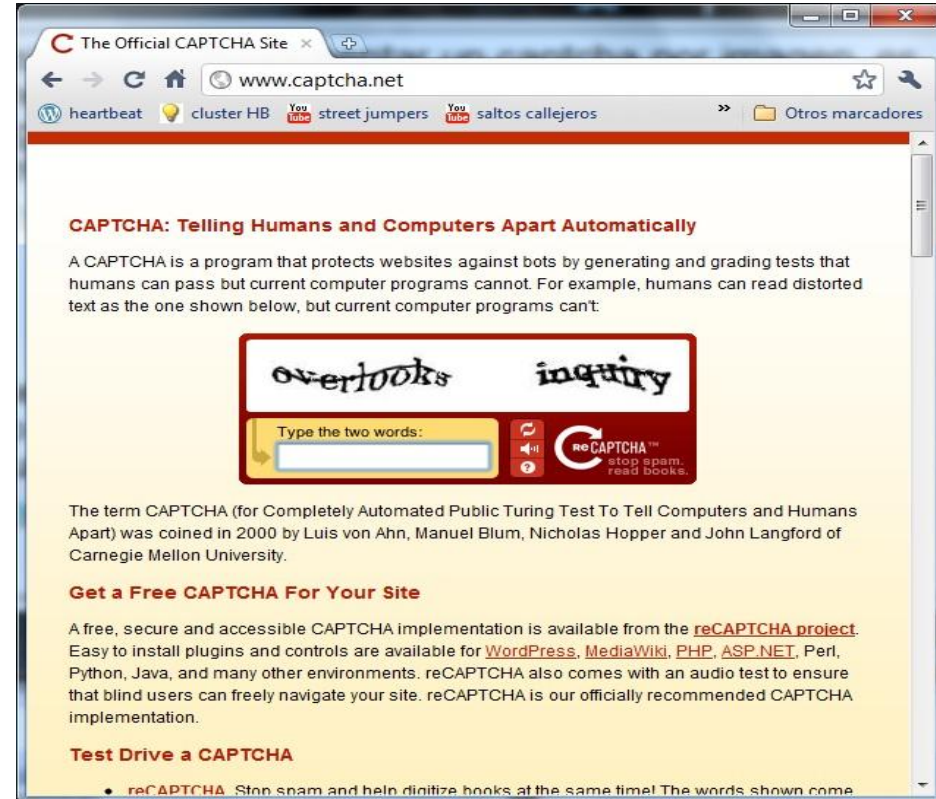
<http://www.free-ocr.com/>

- Ya que si el generador no es lo suficientemente bueno, el sitio puede quedar comprometido, como es el caso de la siguiente noticia:

<http://www.vsantivirus.com/16-04-08.htm>

# Ejemplo de captchas: vulnerable y robusto

Nombre *	<input type="text"/>	
Apellido paterno *	<input type="text"/>	
Apellido materno	<input type="text"/>	
Teléfono de casa *	<input type="text"/>	
Teléfono de oficina	<input type="text"/>	
Dirección de correo *	<input type="text"/>	<b>NO MODIFICABLE</b>
Contraseña *	<input type="password"/>	Máximo 8 caracteres
Confirme su contraseña *	<input type="password"/>	
Sobrenombre *	<input type="text"/>	Ej. Lic. González, Ana Mari, etc.
Pregunta secreta *	<input type="text"/>	
Respuesta secreta *	<input type="text"/>	
Marque la casilla si desea recibir información de la tienda en su correo	<input type="checkbox"/>	
	Para su seguridad escriba los caracteres que ve en la imagen siguiente.	
*Código de verificación	<input type="text"/>	
		



The screenshot shows a web browser window titled "The Official CAPTCHA Site" with the URL "www.captcha.net". The page content includes:

- CAPTCHA: Telling Humans and Computers Apart Automatically**
- A paragraph explaining that CAPTCHA is a program that protects websites against bots by generating and grading tests that humans can pass but current computer programs cannot.
- A CAPTCHA image showing two words: "overlooks" and "inquiry". Below the image is a text input field with the prompt "Type the two words:" and a "reCAPTCHA" logo.
- A paragraph explaining that the term CAPTCHA was coined in 2000 by Luis von Ahn, Manuel Blum, Nicholas Hopper and John Langford of Carnegie Mellon University.
- Get a Free CAPTCHA For Your Site**
- A paragraph explaining that a free, secure and accessible CAPTCHA implementation is available from the [reCAPTCHA project](#). It lists supported environments: WordPress, MediaWiki, PHP, ASP.NET, Perl, Python, Java, and many other environments.
- Test Drive a CAPTCHA**
- A bullet point: **reCAPTCHA** Stop spam and help digitize books at the same time! The words shown come

Las negaciones de Servicio tipo **Syn Flooding**, que tiene como finalidad consumir todas las conexiones del servidor, se ven de la siguiente manera:

It is very simple to detect SYN attacks. The netstat command shows us how many connections are currently in the half-open state. The half-open state is described as SYN\_RECEIVED in Windows and as SYN\_RECV in Unix systems.

```
# netstat -n -p TCP tcp          0          0 10.100.0.200:21
237.177.154.8:25882      SYN_RECV   - tcp          0          0 10.100.0.200:21
236.15.133.204:2577     SYN_RECV   - tcp          0          0 10.100.0.200:21
127.160.6.129:51748     SYN_RECV   - tcp          0          0 10.100.0.200:21
230.220.13.25:47393     SYN_RECV   - tcp          0          0 10.100.0.200:21
227.200.204.182:60427   SYN_RECV   - tcp          0          0 10.100.0.200:21
232.115.18.38:278      SYN_RECV   - tcp          0          0 10.100.0.200:21
229.116.95.96:5122     SYN_RECV   - tcp          0          0 10.100.0.200:21
236.219.139.207:49162   SYN_RECV   - tcp          0          0 10.100.0.200:21
238.100.72.228:37899    SYN_RECV   - ...
```

We can also count how many half-open connections are in the backlog queue at the moment. In the example below, 769 connections (for TELNET) in the SYN RECEIVED state are kept in the backlog queue.

- Para mitigar el impacto que tienen en la aplicación Web, se debe solicitar que se lleve a cabo un **TCP/IP HARDENING**, el cual consiste a grandes rasgos de disminuir el tiempo de vida de conexiones, rastreo de conexiones, los cuales muchas veces se modifican en los parámetros del kernel.
- Los siguiente enlaces muestran una serie de parámetros que se deben configurar:

<http://cromwell-intl.com/security/security-stack-hardening.html>

<http://www.javvin.com/networksecurity/TCPSYNAttack.html>



Para los ataques de tipo **command injection**, **SQL Injection**, **Path Transversal**, **Cross Site Scripting**, si bien son elementos que se deben contemplar en el desarrollo de la aplicación Web, también se pueden mitigar en el servidor, para lo cual es recomendable que se solicite la instalación del modulo de apache `mod_secure`.



En el módulo `mod_secure` se pueden implementar una serie de filtros que detectan los patrones propios de dichos ataques.

#### Null byte attack prevention

Null byte attacks try to confuse C/C++ based software and trick it into thinking that a string ends before it actually does. This type of an attack is typically rejected with a proper `SecFilterByteRange` filter. However, if you do not do this a null byte can interfere with ModSecurity processing. To fight this, ModSecurity looks for null bytes during the decoding phase and converts them into spaces. So, where before this filter:

```
SecFilter hidden
```

would not detect the word hidden in this request:

```
GET /one/two/three?p=visible%00hidden HTTP/1.0
```

it now works as expected.

#### Regular expressions

The simplest method of filtering I discussed earlier is actually slightly more complex. Its full syntax is as follows:

```
SecFilter KEYWORD [ACTIONS]
```

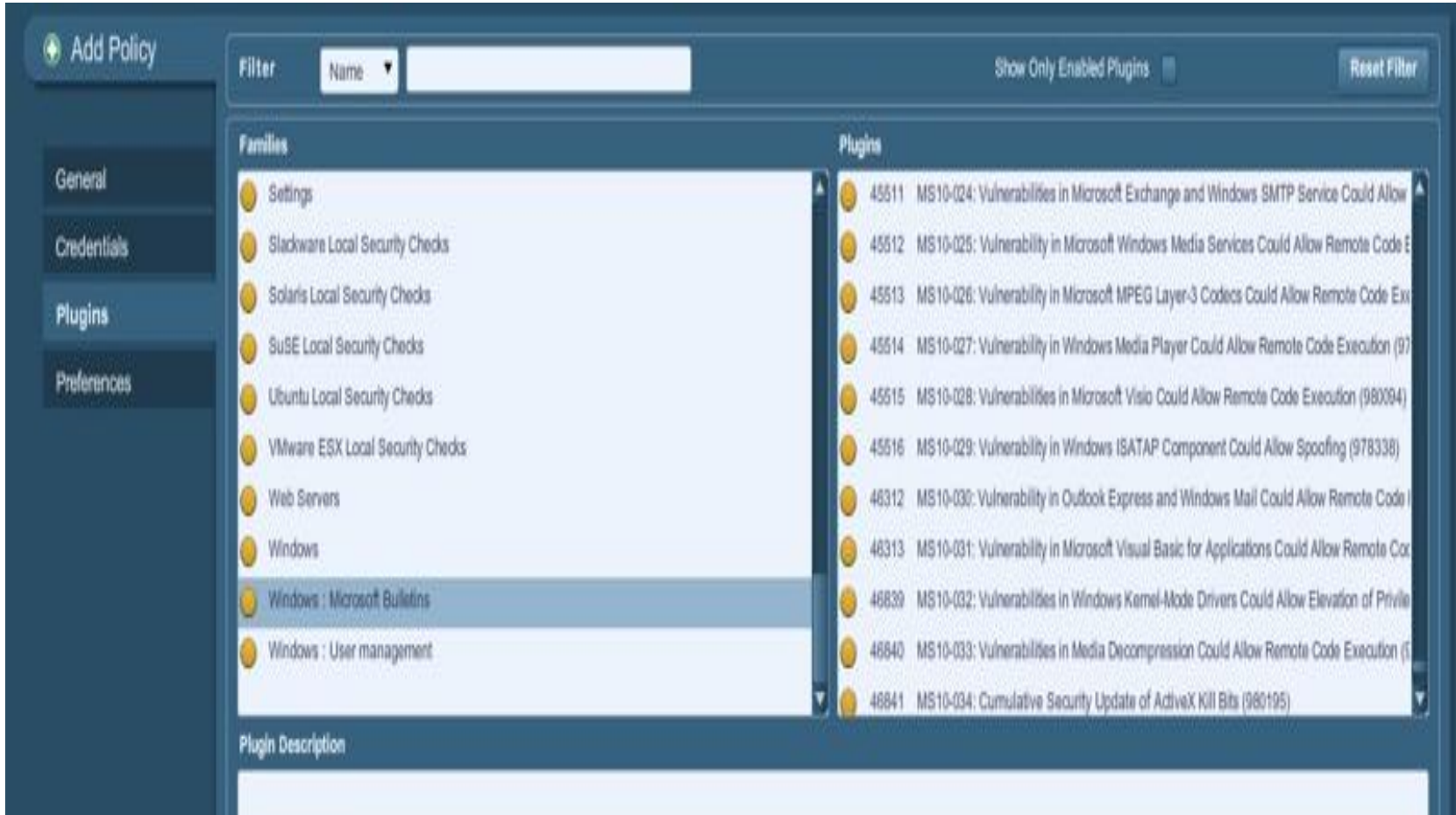
Por ejemplo ver:

<http://www.modsecurity.org/documentation/modsecurity-apache/1.9.3/html-multipage/04-rules.html>



- Para mitigar el uso de **exploits** en el servidor Web, es recomendable hacer uso de **scanners** de vulnerabilidades.
- La finalidad de dichos scanners es la de indicarnos que configuraciones o archivos se dejaron por default, así como encontrar huecos comunes en dichos servidores.
- Un scanner de uso común es nessus, el cual presenta una interfaz amigable y es útil para mitigar el uso de exploits conocidos.

## Vista del scanner de vulnerabilidades nessus



The screenshot displays the Nessus scanner interface. On the left, there is a sidebar with navigation options: "Add Policy", "General", "Credentials", "Plugins", and "Preferences". The "Plugins" section is currently selected. The main area shows a list of vulnerability plugins, organized into two columns: "Families" and "Plugins".

Families	Plugins
Settings	45511 MS10-024: Vulnerabilities in Microsoft Exchange and Windows SMTP Service Could Allow Remote Code Execution (978338)
Slackware Local Security Checks	45512 MS10-025: Vulnerability in Microsoft Windows Media Services Could Allow Remote Code Execution (978338)
Solaris Local Security Checks	45513 MS10-026: Vulnerability in Microsoft MPEG Layer-3 Codecs Could Allow Remote Code Execution (978338)
SuSE Local Security Checks	45514 MS10-027: Vulnerability in Windows Media Player Could Allow Remote Code Execution (978338)
Ubuntu Local Security Checks	45515 MS10-028: Vulnerabilities in Microsoft Visio Could Allow Remote Code Execution (980094)
VMware ESX Local Security Checks	45516 MS10-029: Vulnerability in Windows ISATAP Component Could Allow Spoofing (978338)
Web Servers	46312 MS10-030: Vulnerability in Outlook Express and Windows Mail Could Allow Remote Code Execution (978338)
Windows	46313 MS10-031: Vulnerability in Microsoft Visual Basic for Applications Could Allow Remote Code Execution (978338)
Windows : Microsoft Bulletins	46839 MS10-032: Vulnerabilities in Windows Kernel-Mode Drivers Could Allow Elevation of Privilege (978338)
Windows : User management	46840 MS10-033: Vulnerabilities in Media Decompression Could Allow Remote Code Execution (978338)
	46841 MS10-034: Cumulative Security Update of ActiveX Kill Bits (980195)

Below the list, there is a section for "Plugin Description" which is currently empty.

# Referencias

- Seguridad Informática: <http://hackdosx.blogspot.com/2008/12/como-identificar-ataques-por-fuerza.html>
- Mod security: <http://www.modsecurity.org/documentation/modsecurity-apache/1.9.3/html-multipage/04-rules.html>
- [http://webcache.googleusercontent.com/search?q=cache:IONhliwNdjQJ:cocinaphp.paleontologia.co.uk/index.php%3F/archives/174-mod\\_security-reglas-basicas-XSS,-SQL-injection-...html+Modsecure+apache+SQL+injection&cd=4&hl=es&ct=clnk&source=www.google.com](http://webcache.googleusercontent.com/search?q=cache:IONhliwNdjQJ:cocinaphp.paleontologia.co.uk/index.php%3F/archives/174-mod_security-reglas-basicas-XSS,-SQL-injection-...html+Modsecure+apache+SQL+injection&cd=4&hl=es&ct=clnk&source=www.google.com)
- Captcha: <http://www.captcha.net/> y <http://es.wikipedia.org/wiki/Captcha>
- Practicas de escaneo: <http://labs.dragonjar.org/laboratorios-hacking-tecnicas-y-contramedidas-escaneo-de-vulnerabilidades-iii>
- Hardening TCP/IP: <http://www.symantec.com/connect/articles/hardening-tcpip-stack-syn-attacks>
- SQLINJECTION: <http://www.thetechherald.com/article.php/200935/4314/SQL-Injection-attack-still-spreading-84000-and-counting>
- Scanner de seguridad: <http://www.digitalbond.com/wiki/index.php/Nessus> y <http://www.nessus.org/download/>