WebShellz

Presented By:
Joe McCray

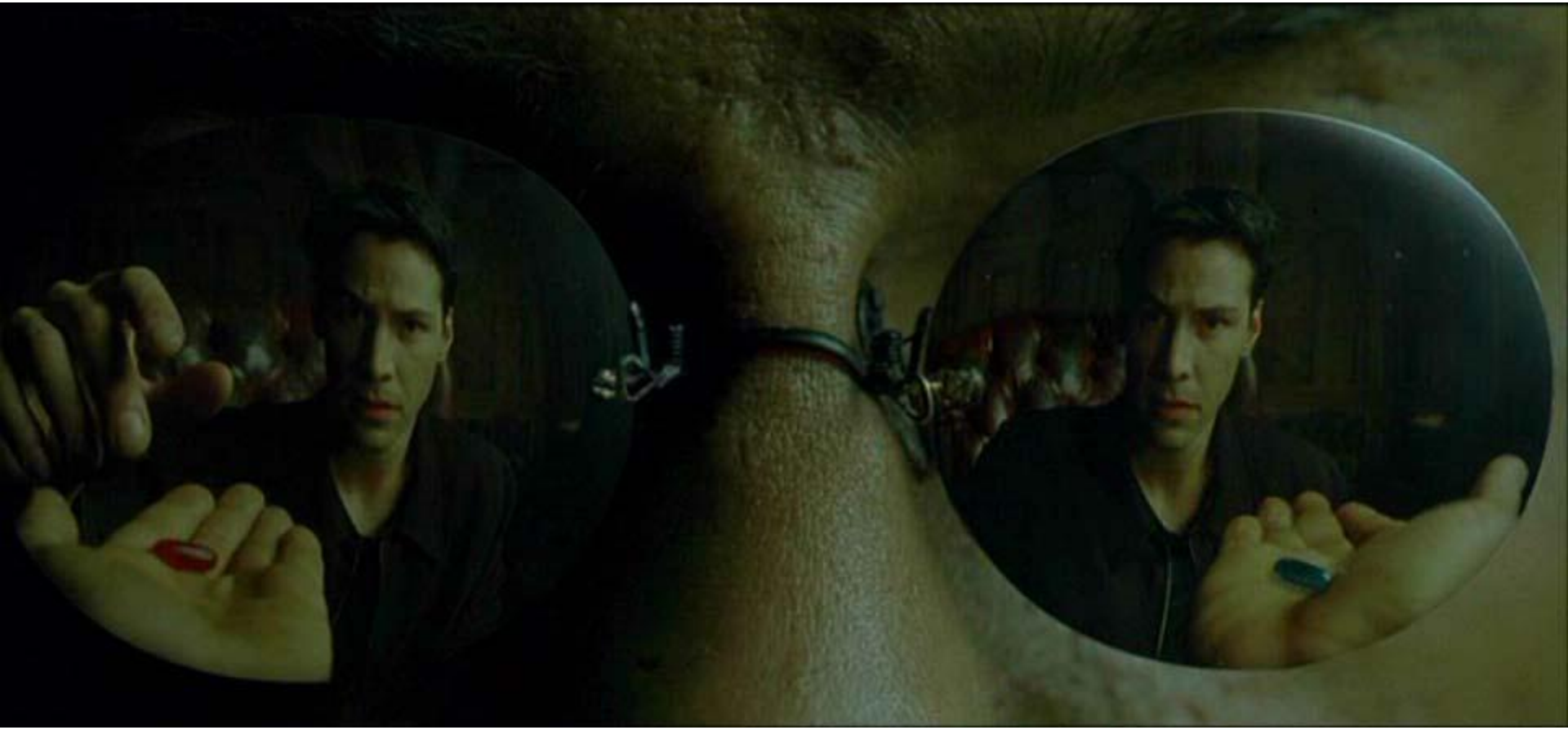
joe@strategicsec.com
http://www.linkedin.com/in/joemccray
http://twitter.com/j0emccray

Joe McCray.... Who the heck are you?

IT Security Consultant (Hacker)

AKA: The Black Guy at Security Conferences

I Only Offer You The Truth Neo...



What Are We Up Against?

News Flash:

Port Scanning is Dead & Your Firewall Is Useless!

Hackers today use client-side exploits to get into your network, or attack your company's web apps.

Deal with it!

3 Primary Classes Of Web Vulnerabilities

- Injection Vulnerabilities
  - SQL, XPATH, LDAP, SSI
- Abuse of Trust Vulnerabilities
  - XSS, CSRF, HTTP Request Smuggling/Response Splitting
- File Handling Vulnerabilities
  - RFI/LFI/Unchecked File Upload

3 Questions To Ask Yourself

- Does the page talk to a DB?
  - If YES – try SQL, XPATH, LDAP, SSI
- Can you or someone else see what you type?
  - If YES – try XSS, CSRF, HTTP Request Smuggling/Response Splitting
- Does the page reference a file?
  - If Yes – try LFI/LFI/Unchecked File Upload

1st Question

- Does the page talk to a DB?
  - If YES – try SQL, XPATH, LDAP, SSI

- Look for parameter passing (something=something) in the HTTP request.

- Change it to:
  something=something’

2nd Question

- Can you or someone else see what you type?
  - Search box, contact us form, forum, blog, guestbook,

- Where ever you can type try:
  - <script>alert('xss')</script>
  - <script>alert(1)</script>

3rd Question

- Does the page reference a file?
  - If Yes – try LFI/LFI/Unchecked File Upload
- Look for a file references in the HTTP request
  - page=resume.pdf
  - file=resume.doc
- Change it to:

file=../../../../../../../../../../../../../etc/passwd
file=../../../../../../../../../../../../../etc/passwd%00

Demo

Ok – let's do a quick walkthrough each of these basic web vulnerabilities.

Web Shell Tricks (1: Content Type Check)

```
<?php
if($_FILES['userfile']['type'] != "image/gif") {
echo "Sorry, we only allow uploading GIF images";
exit;
}
$uploaddir = 'uploads/';
$uploadfile = $uploaddir . basename($_FILES['userfile']['name']);
if (move_uploaded_file($_FILES['userfile']['tmp_name'], $uploadfile)) {
echo "File is valid, and was successfully uploaded.\n";
} else {
echo "File uploading failed.\n";
}
?>
```

Bypass Technique:

Use your Proxy to change the HTTP header value from:

“Content-Type: text/plain”

to

"Content-Type" => "image/gif"

Web Shell Tricks (2: File Type Blacklist)

```
<?php
$blacklist = array(".php", ".phtml", ".php3", ".php4");
foreach ($blacklist as $item) {
if(preg_match("/$item\$/i", $_FILES['userfile']['name'])) {
echo "We do not allow uploading PHP files\n";
exit;
}
}
$uploaddir = 'uploads/';
$uploadfile = $uploaddir . basename($_FILES['userfile']['name']);
if (move_uploaded_file($_FILES['userfile']['tmp_name'], $uploadfile)) {
echo "File is valid, and was successfully uploaded.\n";
} else {
echo "File uploading failed.\n";
}
?>
```

Bypass Technique:

1. Some web applications may require that files with .gif or .jpeg extensions are interpreted by PHP (this often happens when images, for example graphs and charts, are dynamically generated on the server by a PHP script).

2. Try to upload a file with a null terminator shell.php%00.gif. Maybe the .gif file will be truncated by the application.

Web Shell Tricks (3: Valid Image Check)

```
<?php
$imageinfo = getimagesize($_FILES['userfile']['tmp_name']);
if($imageinfo['mime'] != 'image/gif' && $imageinfo['mime'] != 'image/jpeg') {
echo "Sorry, we only accept GIF and JPEG images\n";
exit;
}
$uploaddir = 'uploads/';
$uploadfile = $uploaddir . basename($_FILES['userfile']['name']);
if (move_uploaded_file($_FILES['userfile']['tmp_name'], $uploadfile)) {
echo "File is valid, and was successfully uploaded.\n";
} else {
echo "File uploading failed.\n";
}
?>
```

Instead of trusting the Content-type header a PHP developer might decide to validate the actual content of the uploaded file to make sure that it is indeed an image. The PHP getimagesize() function is often used for that. getimagesize() takes a file name as an argument and returns the size and type of the image.

Web Shell Tricks (3: Valid Image Check)

POST /webshell.php HTTP/1.1
TE: deflate,gzip;q=0.3
Connection: TE, close
Host: localhost
User-Agent: j0e-is-da-shiznit/1.1.1
Content-Type: multipart/form-data; boundary=xYzZY
Content-Length: 14835
--xYzZY
Content-Disposition: form-data; name="userfile"; filename="crocus.php"
Content-Type: image/gif
GIF89a(...some binary data...)<?php phpinfo(); ?>(...blah blah blah….)

Bypass Technique:
It is possible to create a perfectly valid image file that contains some PHP code in the comment. When getimagesize() looks at the file, it sees a proper GIF or JPEG image.

When the PHP interpreter looks at the file, it sees the executable PHP code inside of some binary garbage.

Here A Shell, There A Shell, Everywhere A Shell

- C99madShell v. - C99madShell v. 2.0 madnet edition 2.0 madnet edition
- c99-safe-mode - C99-safe-mode
- c99edit - C99edit
- c99shell - C99shell
- DownloaderToFTP – DownloaderToFTP
- GFS Web-Shell ver 4.0.0.0 - GFS Web-Shell ver 4.0.0.0
- NetworkFileManager – NetworkFileManager
- NiX Remote Web Shell™ - NiX Remote Web Shell ™
- r57MySQL_FileViewer - R57MySQL_FileViewer
- r57shell - R57shell
- MySQLBackUpAll – MySQLBackUpAll
- MySQLBackUpOnce – MySQLBackUpOnce
- webadmin – Webadmin
- cihshell – Cihshell
- Sql – Sql
- a_gedit - A_gedit
- Antichat – Antichat
- bk – Bk

Here A Shell, There A Shell, Everywhere A Shell

- c2007 - C2007
- Casus15 - Casus15
- CmdAsp – CmdAsp
- Csh – Csh
- Ctt_sh - Ctt_sh
- Cybershell – Cybershell
- DxShell – DxShell
- gfs_sh - Gfs_sh
- grp-2018 - Grp-2018
- Hidshell – Hidshell
- iMHaPFtp – IMHaPFtp
- Load_shell - Load_shell
- NFM – NFM
- NGH – NGH
- Nixrem – Nixrem
- NST – NST
- Phvayvv – Phvayvv
- Predator – Predator
- r0t - R0t
-

Here A Shell, There A Shell, Everywhere A Shell

- Rashell v.1.31 - Rashell v.1.31
- Xoce 1.5 - Xoce 1.5
- Xoce 1.7 - Xoce 1.7
- img – Img
- mailer3 - Mailer3
- myshell – Myshell
- mysql_tool - Mysql_tool
- mysql – Mysql
- network – Network
- nshell – Nshell
- ru24_post_sh - Ru24_post_sh
- pHpINJ – PHpINJ
- PHP Shell - PHP Shell
- Pws – Pws
- KA_uShell - KA_uShell
- Sincap – Sincap
- telnet – Telnet
- telnetd – Telnetd

Here A Shell, There A Shell, Everywhere A Shell

- Indexer.asp - Indexer.asp
- Klasvayv.asp - Klasvayv.asp
- NTdaddy.asp - NTdaddy.asp
- Reader.asp - Reader.asp
- RemExp.asp - RemExp.asp
- Zehir4.asp - Zehir4.asp
- Ajan.asp - Ajan.asp
- EFSO_2.asp - EFSO_2.asp
- Elmali Seker.asp - Elmali Seker.asp
- Server Variables.asp - Server Variables.asp
- Tool.asp - Tool.asp
- WebShell.pl - WebShell.pl
- phpRemoteView – PhpRemoteView
- PHP Backdoor Connect.pl - PHP Backdoor Connect.pl
- perlbot.pl - Perlbot.pl
- shellbot.pl - Shellbot.pl
- r57pws.pl - R57pws.pl
- lurm_safemod_on.pl - Lurm_safemod_on.pl
- Asmodeus v0.1.pl - Asmodeus v0.1.pl
- connectback2.pl - Connectback2.pl
- Java Shell.js - Java Shell.js

Defense

Check FileType/Valid Files/File Extensions
Use things like content-type header, the PHP getimagesize(), and filetype verification to all files being uploaded.

Restrict PUT Method
Particular care has to be taken with regards to writable web directories if you are running PHP on Microsoft IIS. As opposed to Apache, Microsoft IIS supports "PUT" HTTP requests, which allow users to upload files directly, without using an upload PHP page.

PUT requests can be used to upload a file to the web server if the file system permissions allow IIS (which is running as IUSR_MACHINENAME) to write to the directory and if IIS permissions for the directory allow writing.

Indirect access to the uploaded files
The solution is to prevent the users from requesting uploaded files directly. This means either storing the files outside of the web root or creating a directory under the web root and blocking web access to it in the Apache configuration or in a .htaccess file.

Contact Me....

Toll Free: **1-866-892-2132**

Email: joe@strategicsec.com

Twitter: http://twitter.com/j0emccray

LinkedIn: http://www.linkedin.com/in/joemccray

Slideshare: http://www.slideshare.net/joemccray