eLearnSecurity
Forging security professionals

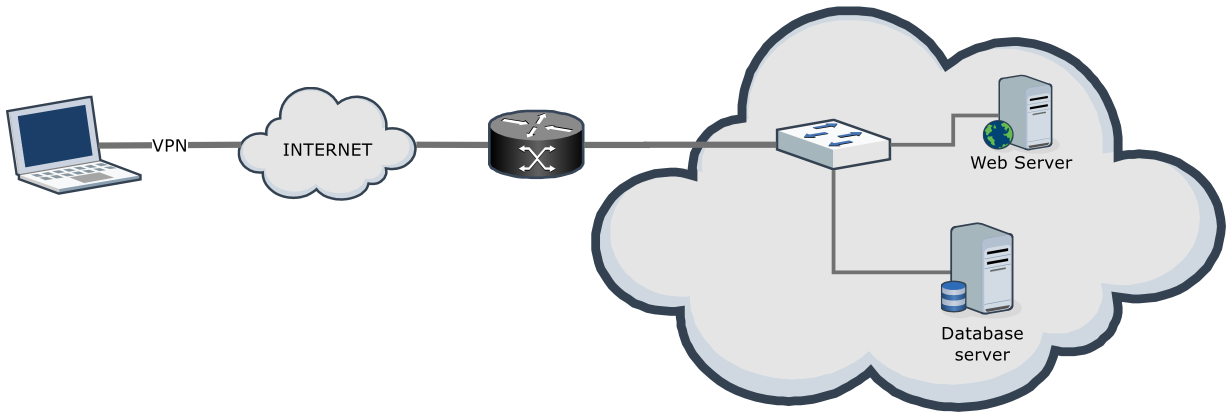
DIRBUSTER

PENETRATION TESTING | SECTION 3 MODULE 4 | LAB #10

LAB

1. Description

You are a Penetration Tester hired by the company *AwdMgmt* to perform security tests on their internal Web Application and machines. You are asked to perform the penetration test on the client premises. During this engagement you are not given a well-defined scope. You are sitting in the client corporate building, directly attached to the client network.



2. Goal

The goal of this lab is to first find the web servers in the network you are directly attached. Then to test the Web Application running on it in order to check if you can access restricted areas (such as the login page)!

3. Tools

The best tools for this lab are:

- *Dirbuster*
- *mysql*
- *Web browser*

4. Steps

4.1. Find all the machines in the network

Since we do not have any information about the network and related hosts, the first step is to find all alive hosts in the network.

4.2. Identify the machines role

Now that we know there is a host on the target network, let us scan it and gather as much information as we can about it. We are interested in web servers.

4.3. Explore the web application

Once we have found a web server explore the Web Application from a web browser and analyze it.

Remember that the goal of our tests is to access the restricted web area.

4.4. Find hidden files

Now that you have an idea of how the Web Application works, run *dirbuster* and check if there is any file that may be useful to access the *login* page!

4.5. Test the credentials found

You should have found two interesting files. Use the information stored in these two file to access the DMBS.

4.6. Retrieve the correct admin password

Now that you have access to the database, dump the administrator credentials and try to log into the Web Application.

SOLUTIONS

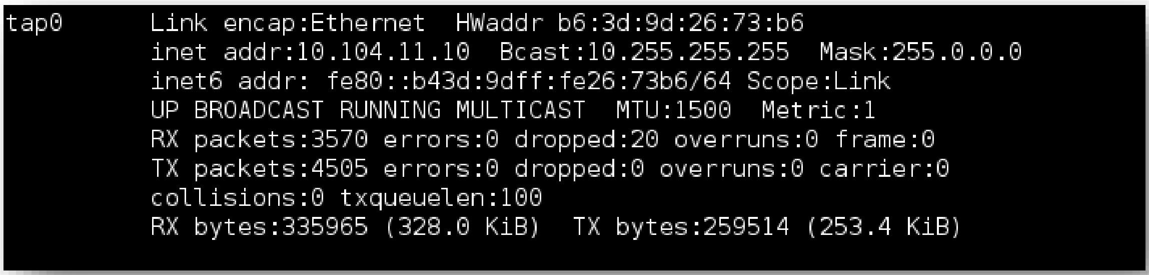
Please go ahead **ONLY** if you have **COMPLETED** the lab or you are stuck! Checking the solutions before actually trying the concepts and techniques you studied in the course, will dramatically reduce the benefits of a hands-on lab!

[This page intentionally left blank]

5. Solutions steps

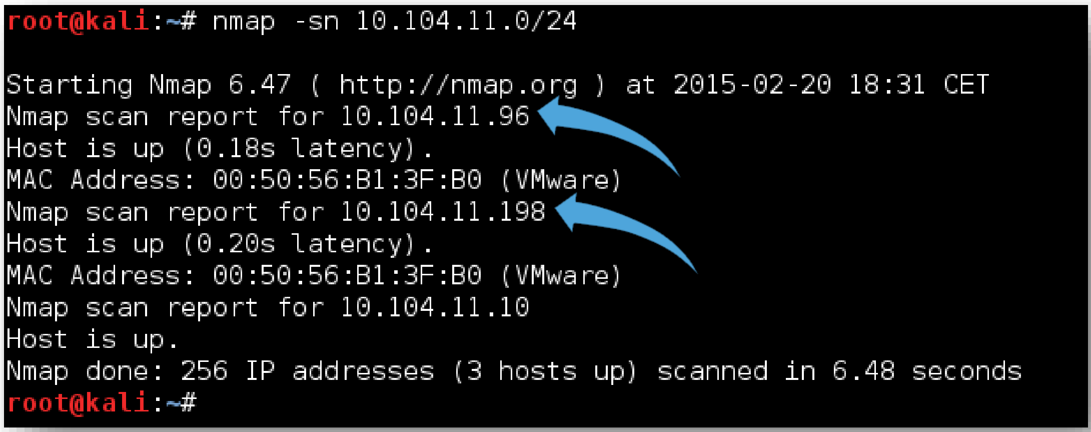
5.1. Find all the machines in the network

We first need to find the address of the corporate network we are connected to. We can do so by running `ifconfig` and check the IP address of our *tap0* interface.



```
tap0      Link encap:Ethernet  HWaddr b6:3d:9d:26:73:b6
          inet addr:10.104.11.10  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::b43d:9dff:fe26:73b6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3570 errors:0 dropped:20 overruns:0 frame:0
          TX packets:4505 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:335965 (328.0 KiB)  TX bytes:259514 (253.4 KiB)
```

As we can see the target network is 10.104.11.0/24. Let's run `nmap -sn` in order to discover all the available hosts on the network:

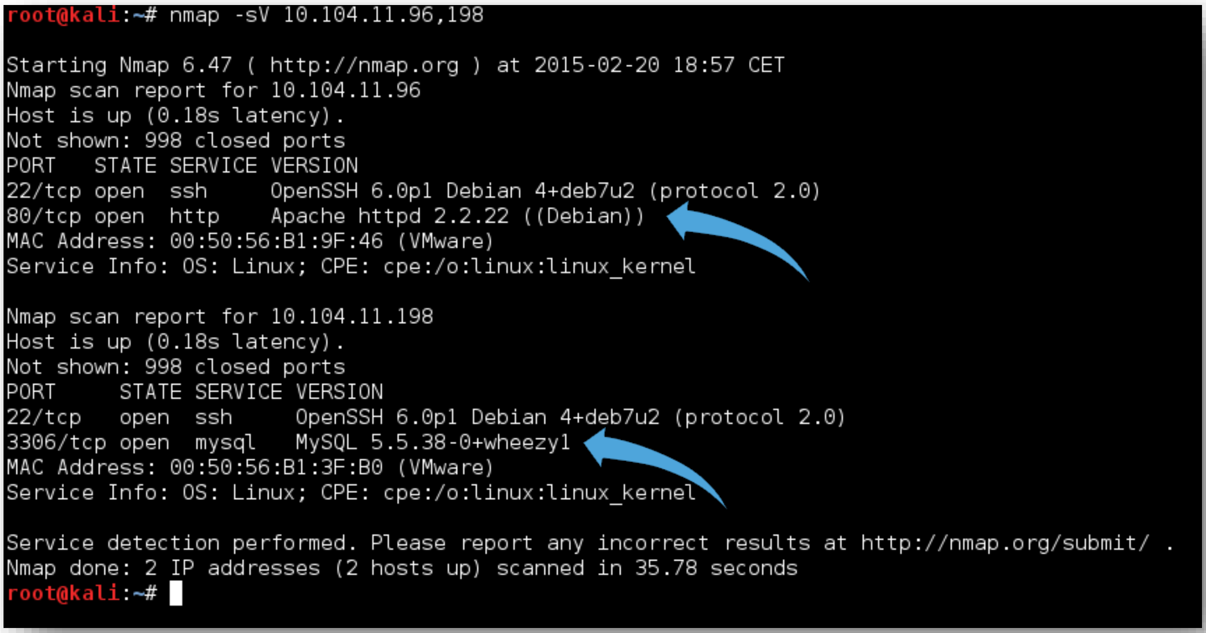


```
root@kali:~# nmap -sn 10.104.11.0/24

Starting Nmap 6.47 ( http://nmap.org ) at 2015-02-20 18:31 CET
Nmap scan report for 10.104.11.96
Host is up (0.18s latency).
MAC Address: 00:50:56:B1:3F:B0 (VMware)
Nmap scan report for 10.104.11.198
Host is up (0.20s latency).
MAC Address: 00:50:56:B1:3F:B0 (VMware)
Nmap scan report for 10.104.11.10
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 6.48 seconds
root@kali:~#
```

The previous screenshot shows that there are two hosts alive in the network: *10.104.11.96* and *10.104.11.198*.

5.2. Identify the machines role

Let us run nmap in order to gather information about the services listening on our targets. To do this we will run a **-sV** scan as follow:



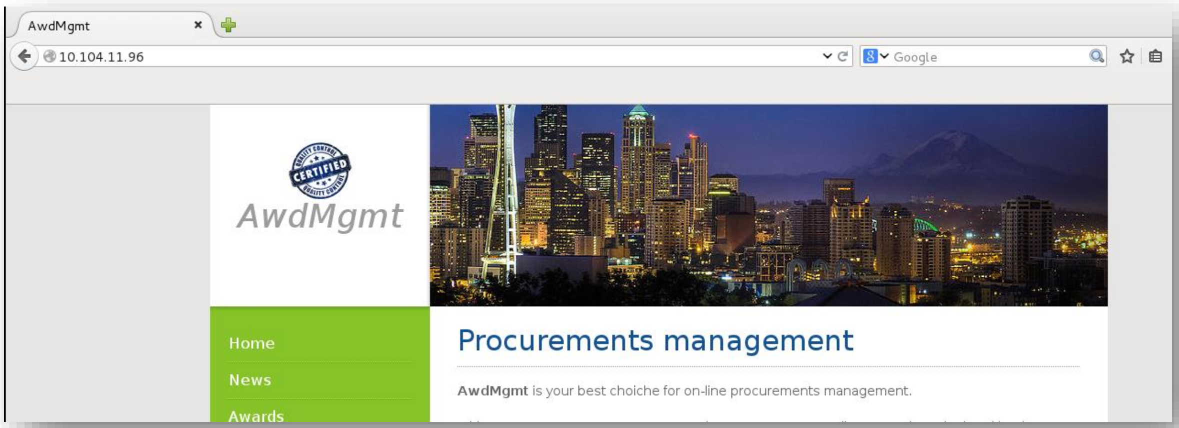
```
root@kali:~# nmap -sV 10.104.11.96,198

Starting Nmap 6.47 ( http://nmap.org ) at 2015-02-20 18:57 CET
Nmap scan report for 10.104.11.96
Host is up (0.18s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.0p1 Debian 4+deb7u2 (protocol 2.0)
80/tcp open  http    Apache httpd 2.2.22 ((Debian))
MAC Address: 00:50:56:B1:9F:46 (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 10.104.11.198
Host is up (0.18s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 6.0p1 Debian 4+deb7u2 (protocol 2.0)
3306/tcp open  mysql   MySQL 5.5.38-0+wheezy1
MAC Address: 00:50:56:B1:3F:B0 (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 2 IP addresses (2 hosts up) scanned in 35.78 seconds
root@kali:~# 
```

From the results, we can see that the machine with IP address 10.104.11.96 is running Apache on port 80, meaning that it is probably hosting the internal web application, while the other machine (10.104.11.198) is running MySQL.

Since the scope of the engagement is to check if an attacker can access restricted areas of the web application, let's focus our tests on the machine 10.104.11.96.

5.3. Explore the web application

In order to inspect the web application we just need to type the IP address of the target machine into our browser.



Procurements management

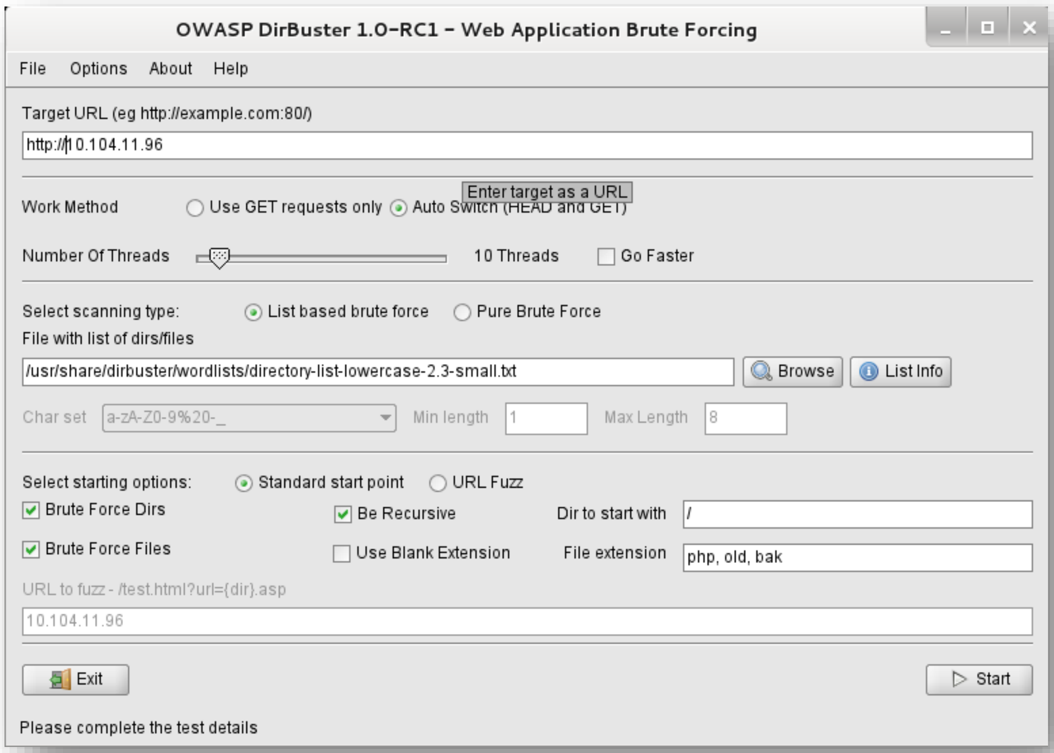
AwdMgmt is your best choiche for on-line procurements management.

If we inspect the web application, we can see that the "Sign up" page is not available, meaning that we cannot create a new user in order to access the restricted area.

Moreover, we do not have any valid credential to use and the form seems not vulnerable to any SQL injection attack.

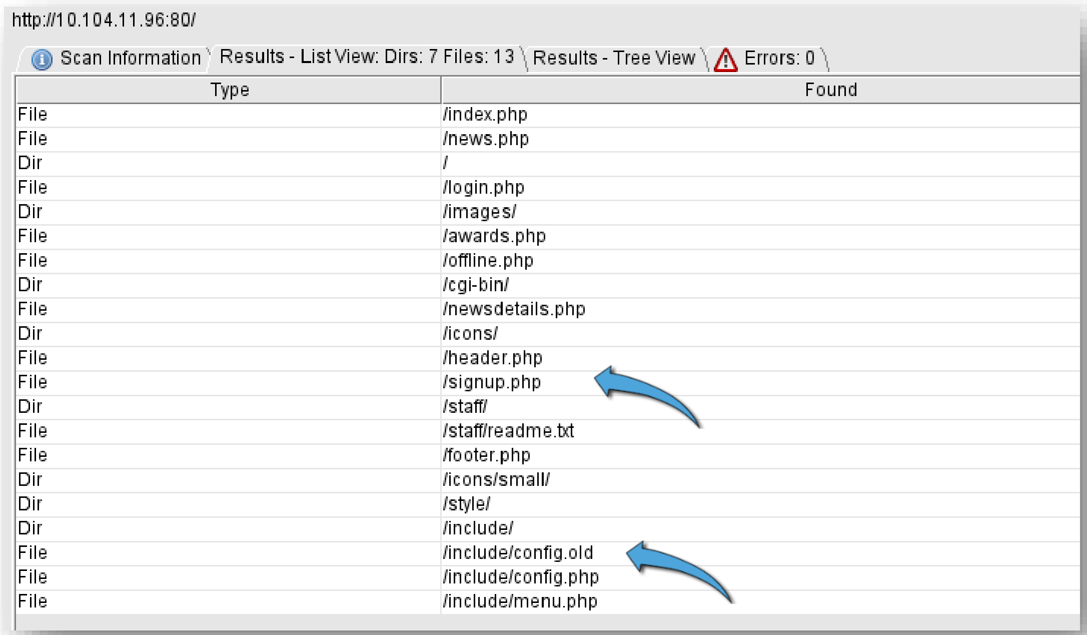
5.4. Find hidden files

Since we do not want to bruteforce the login form, we can try to run discovery tools such as *dirbuster* in order to find hidden files that may help us with our goal.



Target URL (eg http://example.com:80/)
http://10.104.11.96
Enter target as a URL
Work Method: Use GET requests only [✓ Auto Switch (HEAD and GET)]
Number Of Threads: 10 Threads
Go Faster
Select scanning type: [✓ List based brute force] Pure Brute Force
File with list of dirs/files: /usr/share/dirbuster/wordlists/directory-list-lowercase-2.3-small.txt
Char set: a-zA-Z0-9%20-_ Min length: 1 Max Length: 8
Select starting options: [✓ Standard start point] URL Fuzz
[✓ Brute Force Dirs] [✓ Be Recursive] Dir to start with: /
[✓ Brute Force Files] Use Blank Extension File extension: php, old, bak
URL to fuzz - /test.html?url={dir}.asp
10.104.11.96
Please complete the test details

Let us start dirbuster and run a scan using the *directory-list-2.3.-small.txt* file. After a minute or two, we should start getting some interesting results:

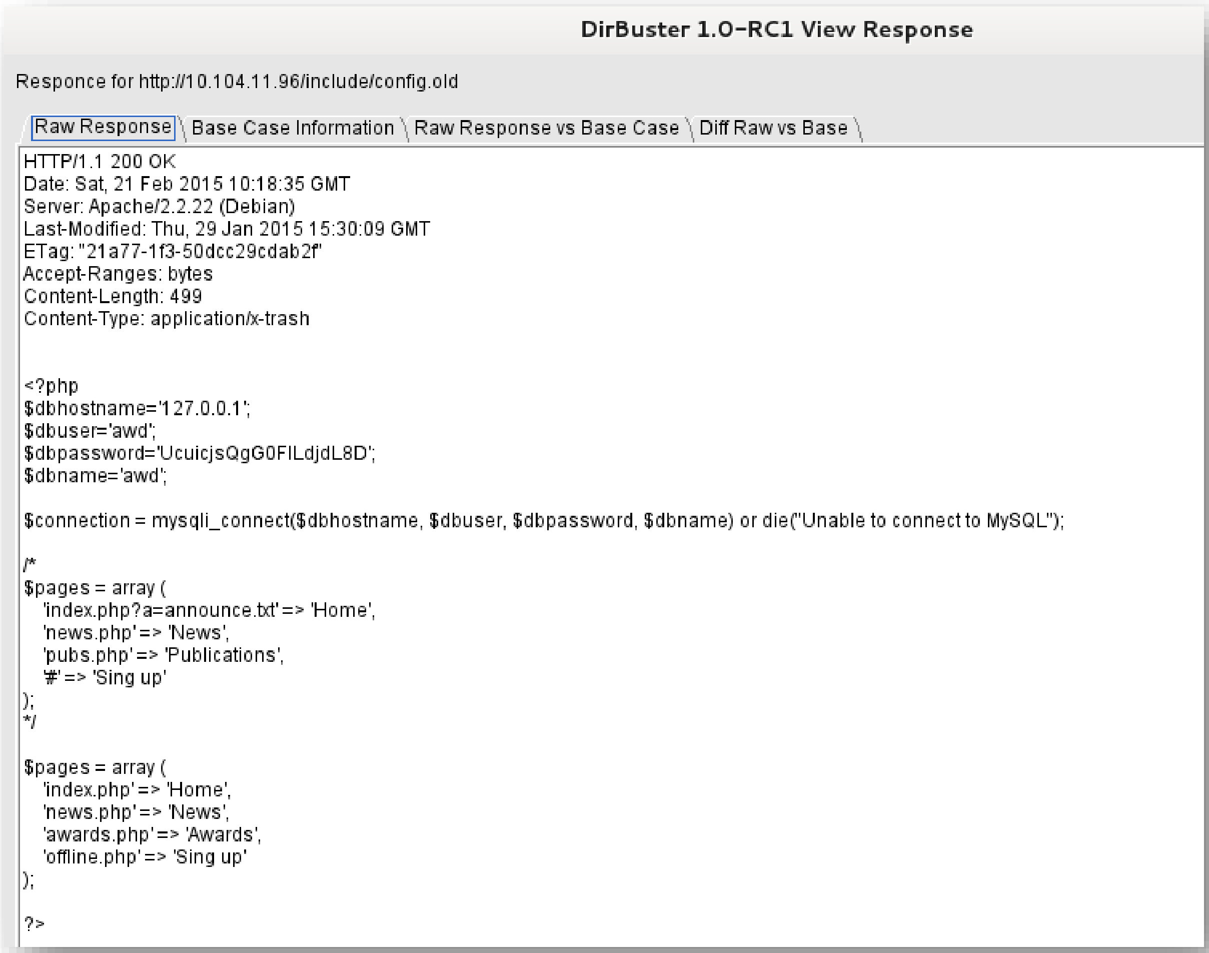


http://10.104.11.96:80/

Scan Information | Results - List View: Dirs: 7 Files: 13 | Results - Tree View | Errors: 0

Type	Found
File	/index.php
File	/news.php
Dir	/
File	/login.php
Dir	/images/
File	/awards.php
File	/offline.php
Dir	/cgi-bin/
File	/newsdetails.php
Dir	/icons/
File	/header.php
File	/signup.php
Dir	/staff/
File	/staff/readme.txt
File	/footer.php
Dir	/icons/small/
Dir	/style/
Dir	/include/
File	/include/config.old
File	/include/config.php
File	/include/menu.php

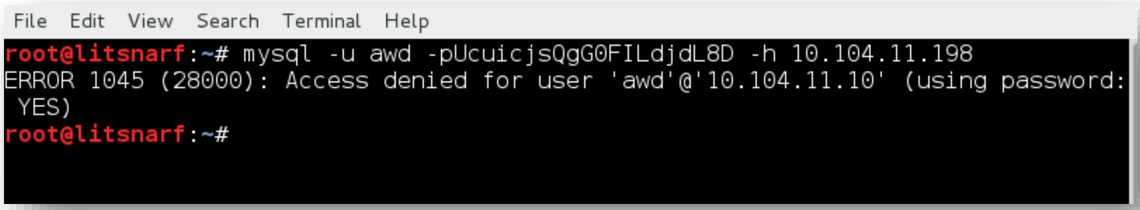
Here we can see that in the *include* folder there is a file named *config.old*. Let us inspect it and see if there is anything interesting in it:



Responce for http://10.104.11.96/include/config.old

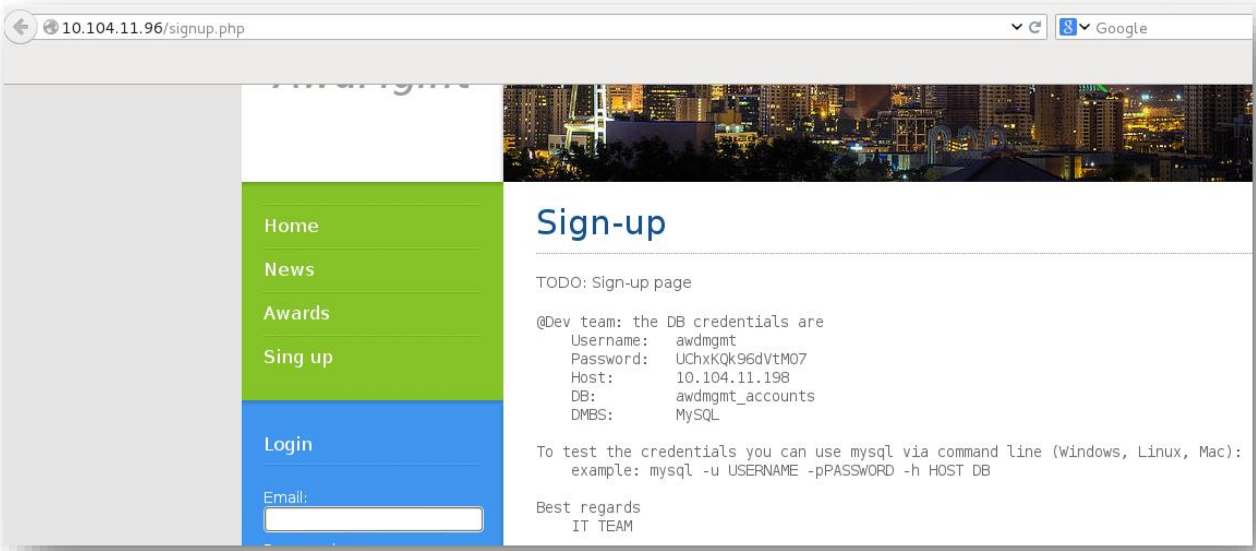
```
HTTP/1.1 200 OK
Date: Sat, 21 Feb 2015 10:18:35 GMT
Server: Apache/2.2.22 (Debian)
Last-Modified: Thu, 29 Jan 2015 15:30:09 GMT
ETag: "21a77-1f3-50dcc29cdab2f"
Accept-Ranges: bytes
Content-Length: 499
Content-Type: application/x-trash


<?php
$dbhostname='127.0.0.1';
$dbuser='awd';
$dbpassword='UcuicjsQgG0FILdjdL8D';
$dbname='awd';

$connection = mysqli_connect($dbhostname, $dbuser, $dbpassword, $dbname) or die("Unable to connect to MySQL");

/*
$pages = array (
   'index.php?a=announce.txt' => 'Home',
   'news.php' => 'News',
   'pubs.php' => 'Publications',
   '#' => 'Sing up'
);
*/

$pages = array (
   'index.php' => 'Home',
   'news.php' => 'News',
   'awards.php' => 'Awards',
   'offline.php' => 'Sing up'
);

?>
```

As we can see, the file contains some database credentials! If you recall, in the previous steps we had found a machine running MySQL. Let us try a DB connection to this machine with the credentials just found:



```
root@litsnarf:~# mysql -u awd -pUcuicjsQgG0FILdjdL8D -h 10.104.11.198
ERROR 1045 (28000): Access denied for user 'awd'@'10.104.11.10' (using password:
 YES)
root@litsnarf:~#
```

Unfortunately, it seems that the credentials are not working. Let us keep investigating the files found with *dirbuster*. If we check the previous screenshot, we can see that there is a page named *signup.php* that we were not able to access from the links in the web application:



Sign-up

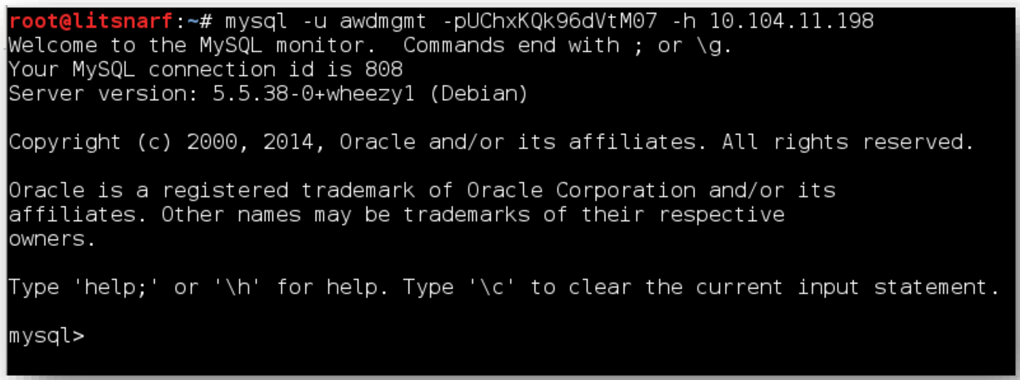
TODO: Sign-up page

```
@Dev team: the DB credentials are
    Username:   awdmgmt
    Password:   UChxKQk96dVtM07
    Host:       10.104.11.198
    DB:         awdmgmt_accounts
    DMBS:       MySQL

To test the credentials you can use mysql via command line (Windows, Linux, Mac):
    example: mysql -u USERNAME -pPASSWORD -h HOST DB

Best regards
    IT TEAM
```

This is even better of the previous file found.

5.5. Test the credentials found

Let us try the credentials found in the *signup.php* file and see if we are able to access the DB!

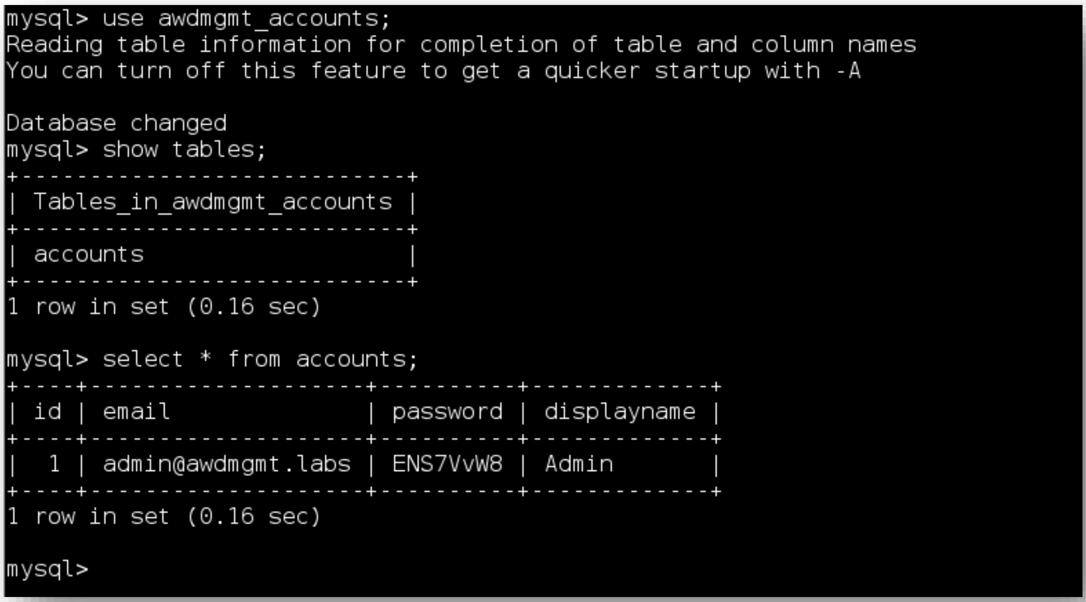


```
root@litsnarf:~# mysql -u awdmgmt -pUChxKQk96dVtM07 -h 10.104.11.198
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 808
Server version: 5.5.38-0+wheezy1 (Debian)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

As we can see, this time we are successfully logged into the database! Let us inspect it!

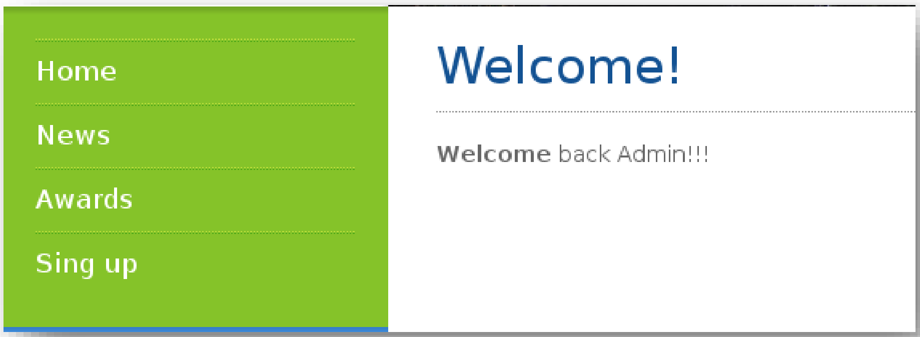
5.6. Retrieve the correct admin password

Let us use some simple *mysql* commands to navigate the database and check if there is anything interesting in it. First, we will have to select the database to use and then inspect its tables and data:



```
mysql> use awdmgmt_accounts;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+----------------------------+
| Tables_in_awdmgmt_accounts |
+----------------------------+
| accounts                   |
+----------------------------+
1 row in set (0.16 sec)

mysql> select * from accounts;
+----+--------------------+----------+-------------+
| id | email              | password | displayname |
+----+--------------------+----------+-------------+
|  1 | admin@awdmgmt.labs | ENS7VvW8 | Admin       |
+----+--------------------+----------+-------------+
1 row in set (0.16 sec)

mysql>
```

With the information just obtained, let us try to log into the web application:



Welcome!

Welcome back Admin!!!

We are logged in!!