Attacking the FTP Service

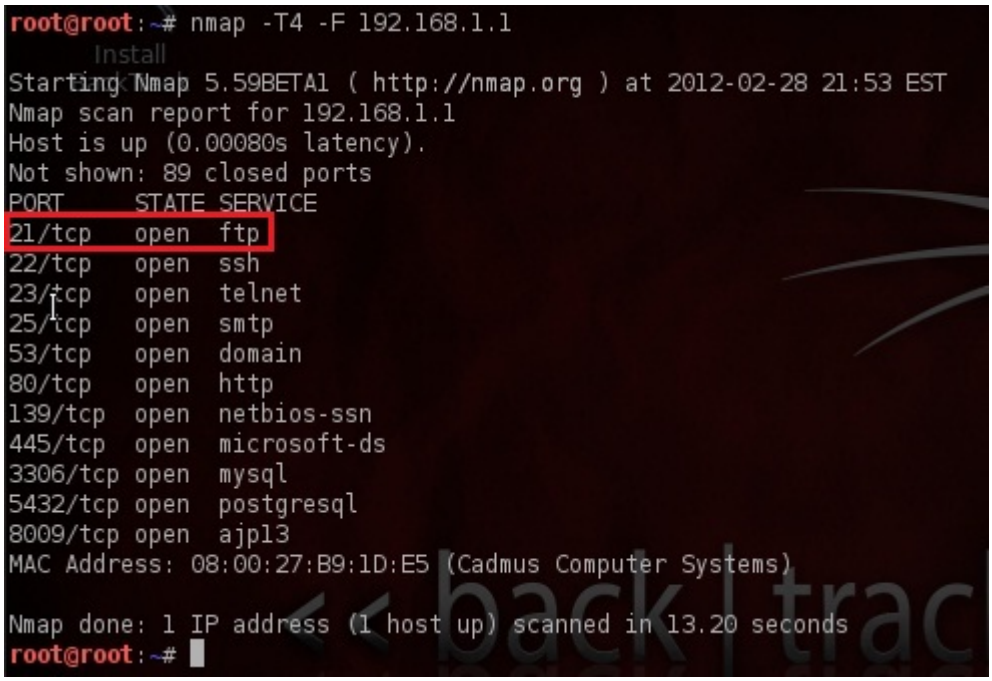
pentestlab.blog/category/exploitation-techniques/page/19

March 1, 2012

FTP is a service that is commonly used in Web Servers from Webmasters for accessing the files remotely.So it is almost impossible not to find this service in one of our clients systems during an engagement.

For that reason we will try to cover in this article a scenario of a possible attack against the FTP Server.

The first thing that we need to do is of course to identify which systems are running the FTP service (for the needs of this tutorial I have put only one system). We can do a simple scan with Nmap in order to find the open ports.



FTP Service Discovery

We can see that the FTP port is open. Now the next logical step that we have to do is to identify which version the FTP application is running by using a method which called FTP banner grabbing.

Of course we can use the Nmap for the discovery of the remote operating system and the service fingerprinting but in this tutorial we will not take advantage of that.

Banner Grabbing is a technique that someone can use in order to extract information from application banners.For example if the remote host is a web server,we can try to connect through telnet.The banner results will give us an indication about the operating system and the type of the web server (Apache or IIS).

Command: ***telnet target_IP 80***

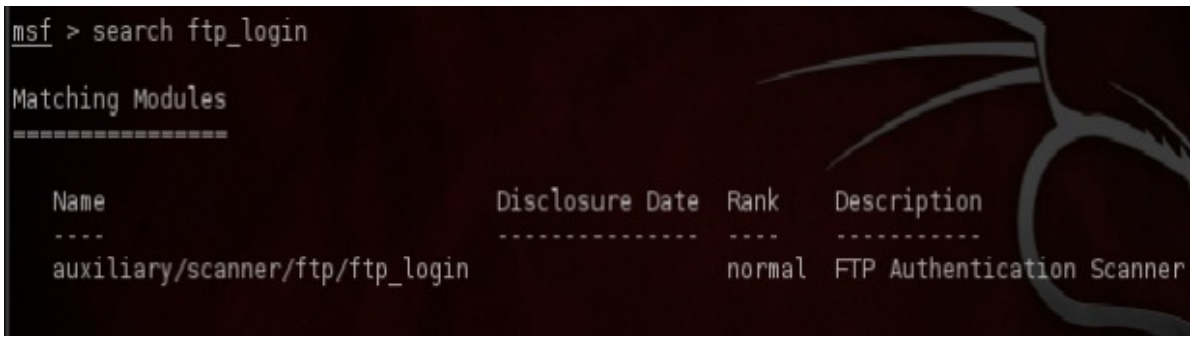
In order to do a banner grabbing in the FTP service we will just try to connect through our console to the FTP server.

FTP Banner Grabbing

From the above image we can see that the version is 1.3.1 and the operating system is Debian.There are many things that we can do from here.First we can try to find if there is any public exploit for the ProFTPD 1.3.1 version.If there is then we can launch it against the FTP Service.

If there is not any public exploit for the specific version then we can try to find a valid username and password by using a dictionary attack.We can use any tool like THC Hydra for this job but in this article we will see how it could be achieved through metasploit.

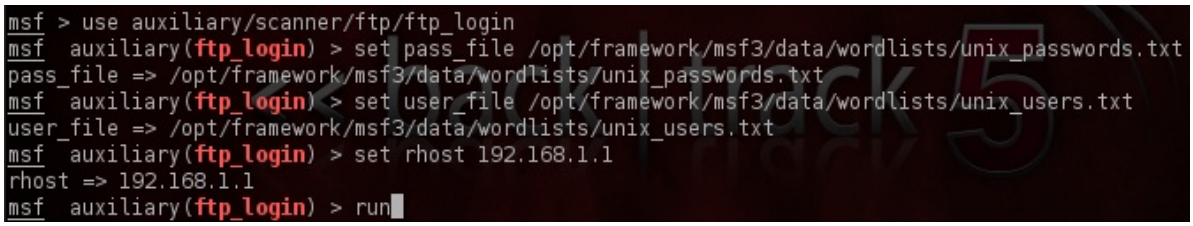
Metasploit Framework has a specific module for attacking FTP servers.So we will search on the metasploit for the module **ftp_login**.



FTP Login Module

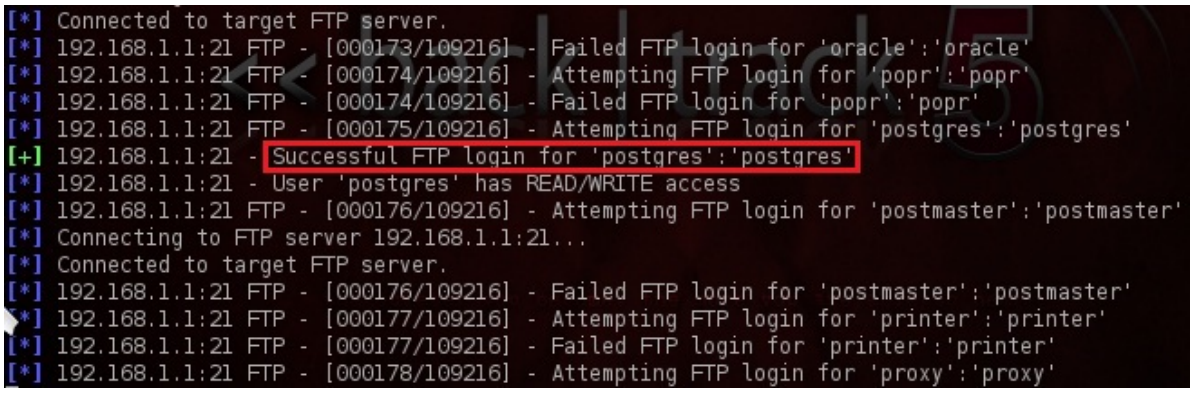
Now that we have found the FTP scanner it is time to configure it.Of course we will need some good wordlists for the usernames and the passwords.If we don't have then there is no problem because metasploit has a folder with various wordlists.Here we will use the wordlists that contains Unix usernames and passwords.

We are setting the scanner according to the following image and we type **run** in order to the scanner to start:

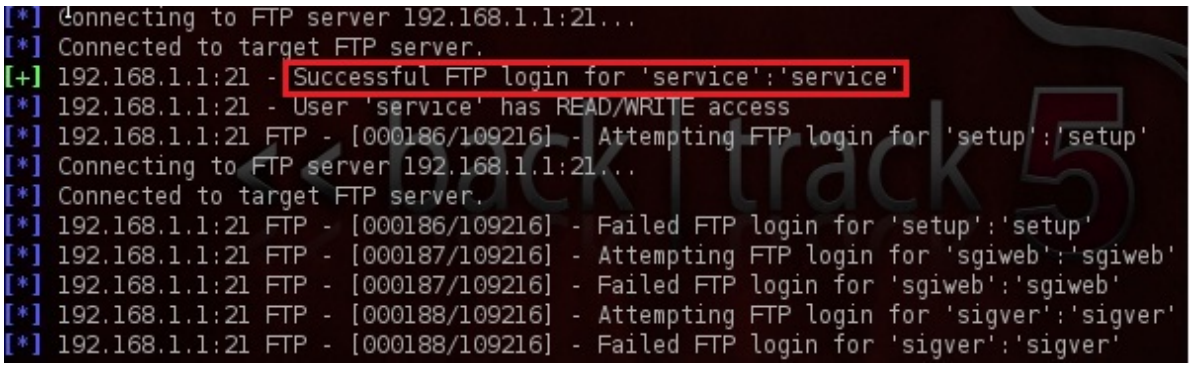


FTP Scanner Settings

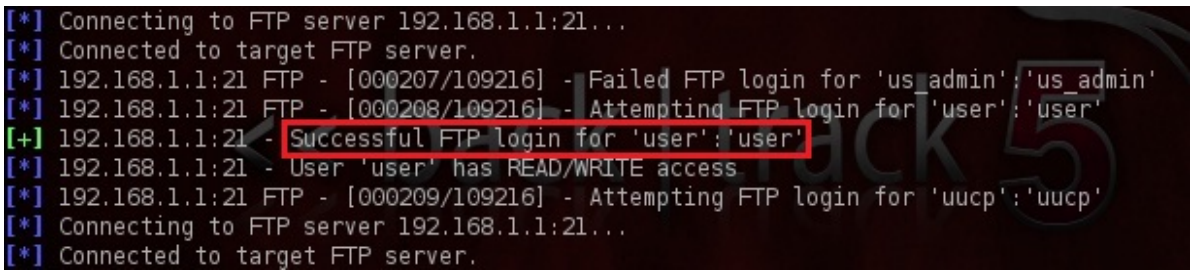
The scanner has discovered 3 valid login credentials as you can see from the next 3 images.



Discovery of the postgres username/password

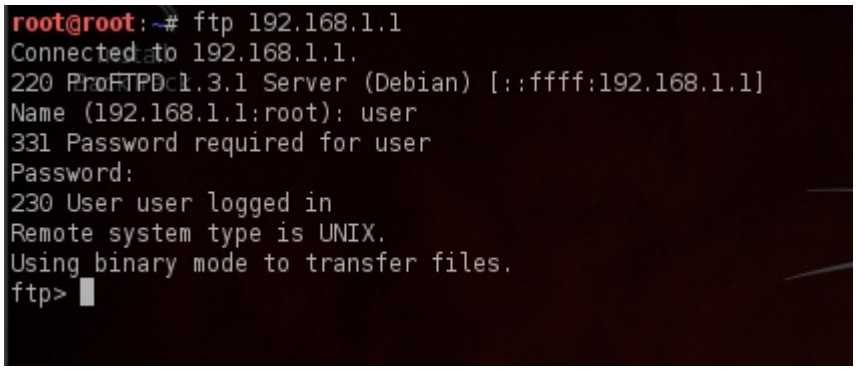


Discovery of the service username/password



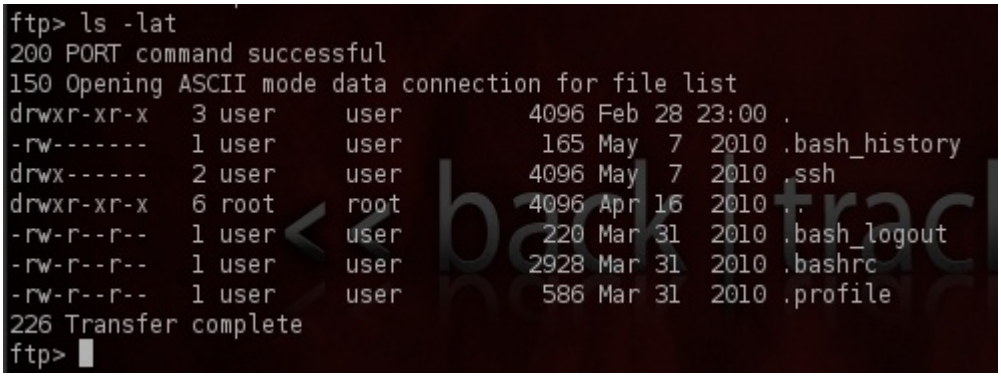
Discovery of the user username/password

So now we have three valid logins to choose in order to connect to the FTP server.Lets try the last one which is the user as username and user as password.



Login with a valid account on the FTP server

We can see that we have successfully managed to login to the FTP server.Now we can execute the command **ls -lat** to the server in order to display the list with the current directories and subdirectories and the permissions that we have on the directories.



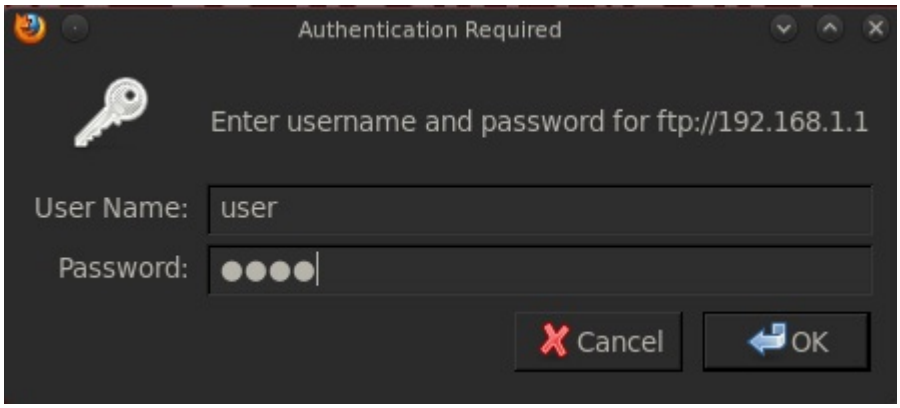
Execution of the command ls -lat

There are two directories that are important here.The SSH because it may contain private SSH keys and the bash_history because it keeps a history of all the commands that the user has run.For example you can find information about user ID,passwords,confidential file names,locations,server names and shared folders.

We will download the bash_history file to our computer with the command **get** as you see it in the image below:

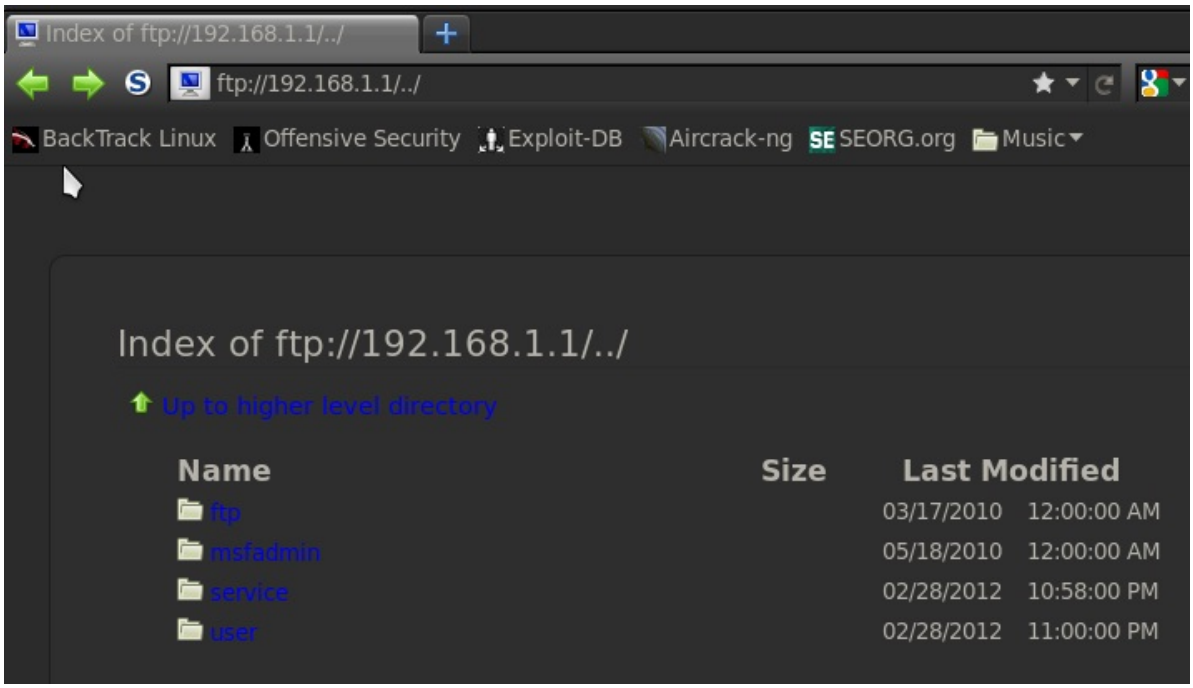
Download the file to our computer

Except of the console for the connection to the FTP server we can use also our browser.We will try to login with the same credentials user/user



Login to the FTP server via browser

After some searching in the directories we have found a directory which contained the following:



Directory of usernames

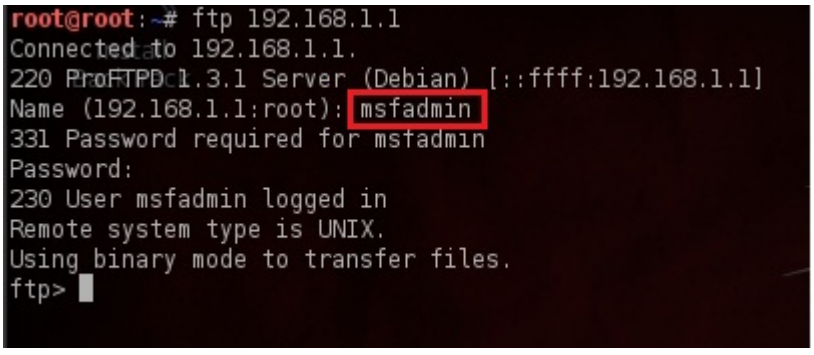
We can see that there are 4 folders.A folder named **user**, a folder named **service** and a folder named **msfadmin**.

This is an indication that another account exists under the username **msfadmin** which probably is an administrator's account and has more privileges.The reason that we assumed that is because the folders names are the same with the usernames that we have discovered previously.

The previous accounts had passwords same with the usernames.So we will try to login with the following credentials:

Username: msfadmin

Password: msfadmin



Login with msfadmin

The image above is showing that our try to login with the username/password msfadmin was successful.If the password was different then we could have tried another dictionary attack against the FTP server in order to find and the password.

So we have managed to login to the FTP server with an administrator's account.

Conclusions

There are some conclusions that we can make regarding this scenario.First of all the banner grabbing allow us to discover valuable information about the FTP server and the target operating system.This means that if the administrator had changed the FTP banner then it would be much harder for us to disclose these information.

In addition we have noticed the weak credentials of the 3 accounts that we discovered.Also the administrator’s account password is the same with the username.This account policy is unacceptable in most of the companies and probably you will not meet something similar.However even large organizations are suffering from weak passwords so eventually it can happen.It is important for that reason to have a good password policy.

On the other hand if a malicious user was trying brute force or dictionary attacks (like this scenario) against the FTP server then it would probably flooded the log files.A security solution that would block the IP address after 3 unsuccessful logins would be the most effective.