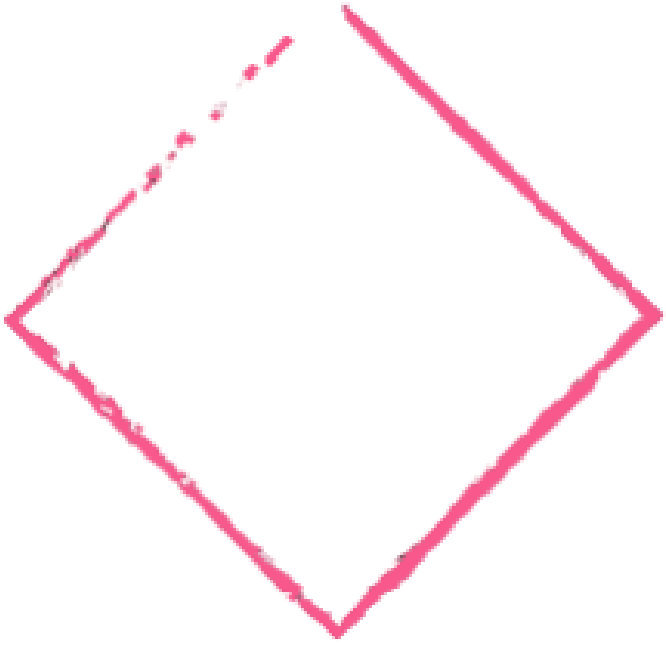


WALKTHROUGH

[GIFT MACHINE]

Disclaimer

This report is for **educational purposes only**. The author is **not responsible** for any misuse of the information. Perform security testing **only on systems you own or are authorized to test**.

Table of Contents

Introduction

This report documents the security assessment of the HackMyVM machine “Gift”, hosted locally in a controlled lab environment. The purpose of this walkthrough is twofold: to demonstrate the purpose of identifying and exploiting vulnerabilities in the target system and to provide remediation steps to secure the system against such attacks.

Machine Information

Gift is a Linux-based machine available on the HackMyVM platform. It was created by sml and is categorized as an Easy machine. The challenge focuses on basic enumeration and exploiting weak credentials to gain system access.

Recon and Enumeration

Target Discovery

Before beginning enumeration, it's important to identify the target machine’s IP address within the local network. This required knowing both the attacker’s IP address and the subnet range. In this lab, the network range was set to /24.

```
┌──(kali㉿kali)-[~/hackmyvm/gift]
└─$ ip a
```

Now that we know the range, we can scan the network with **Nmap** to look for active hosts.

```
┌──(kali㉿kali)-[~/hackmyvm/gift]
└─$ sudo nmap -sn 192.168.7.0/24
[sudo] password for kali:
```

The scan result shows the target machine’s IP. (I can’t share the actual output here because it contains sensitive information, but it would look something like the screenshot below.)

```
MAC Address: [redacted] (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.7.5
```

Hostname Configuration

To make things easier while working on the target, I edited the /etc/hosts file and mapped the machine's IP to the name gift.hmv. Now, instead of typing the IP address each time, I can just use gift.hmv in the browser or terminal, and it points directly to the target.

```
┌──(kali㉿kali)-[~/hackmyvm/gift]
└─$ nano /etc/hosts
```

When editing the file with nano, you just need to add a line with the IP address followed by the hostname you want to assign.

Port Scanning & Service Enumeration

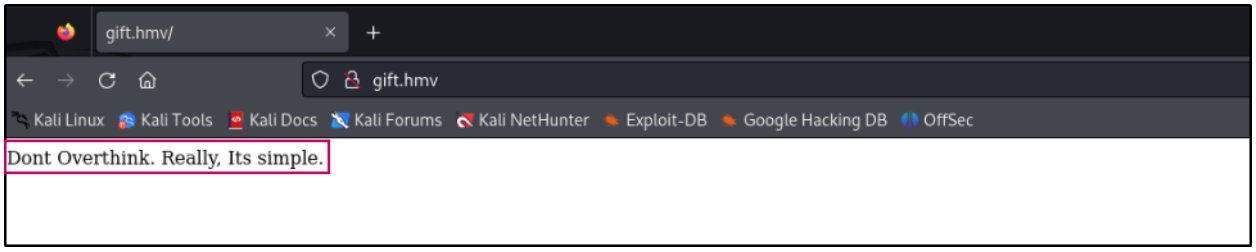
The next step is to scan the target machine to open ports and see what services are running. For this, I like to use Rustscan since it's much faster at detecting open ports compared to other tools. Once I get the list of open ports, I feed them into Nmap for deeper enumeration to gather service details and versions. Rustscan makes this very simple, it's just a one-line command.

```
┌──(kali㉿kali)-[~/hackmyvm/gift]
└─$ rustscan -a gift.hmv -- -A
```

```
Open 192.168.7.6:80
Open 192.168.7.6:22

PORT   STATE SERVICE REASON  VERSION
22/tcp open  ssh     syn-ack OpenSSH 8.3 (protocol 2.0)
| ssh-hostkey:
|   3072 2c:1b:36:27:e5:4c:52:7b:3e:10:94:41:39:ef:b2:95 (RSA)
| ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQCwvhffyA9Z9cqVhVe0GuixD3HU4XTTTf1CQnN9PbBFckBHxypueBuI9N0WkAOvZLGKI9Jkj
xzxgQ5vIdzr83IoyrbUBw/nFLwRzsVhBM+JMUqSZ9OHMhg8qQpFIAcdNprgB40DgER+hMrU+yUAqwbNISQC/aE+DCdHNjNqF
w6Pf2/+7bp8CbntJAxdh4DtHZAmneKy/2JGKzpJcDxU2L8B5pY9uvajkKVSDXVFe1bJZV9ZirBalgYGgke4sTz5kpIeT3CyEefJie
6r7wloIH4CiWtyXDsYGMt5mD2UBCa4GDQaJO5U9F0qjYFa8YdVCOTWdyQvOlFOgqydvAl0LRf6tZKNqVOb/peNf9K8Ucrg4n+Ieva
Gmivh
yGXnwbuCuHN1QH/9dzbNbnZwXn2GYtwYdjBy6AmHRX9Jcsdorj4b/r+eCEPvFIm4ESc7qsn4ShtQr9R8fTgrWArJkfLKhr4KdwMZo
ifAbjrR/G/lj524dS20mbbVLdhjy/8rH/42dN0=
|   256 93:c1:1e:32:24:0e:34:d9:02:0e:ff:c3:9c:59:9b:dd (ECDSA)
| ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBEK4YVSVfGAFEwIJqSel1n33seZLyN+AgGU4rUu5Xrf2LnzQm
ntddLtLtc1Soqu6SpOi/A6vefQzI+a867uJ3Tw=
|   256 81:ab:36:ec:b1:2b:5c:d2:86:55:12:0c:51:00:27:d7 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAID5tIogpq9Eky8MaFF10Cq48d+nTRmXk0OwWl8J8CNIq
80/tcp open  http    syn-ack nginx
| http-methods:
|_  Supported Methods: GET HEAD
|_http-title: Site doesn't have a title (text/html).
```

We found two open ports: 22 SSH and 80 HTTP. Let's check the web service on port 80.



First, I tried to connect to the machine using SSH.

```
┌──(kali㉿kali)-[~/hackmyvm/gift]
└─$ ssh root@gift.hmv
The authenticity of host 'gift.hmv (192.168.7.6)' can't be established.
ED25519 key fingerprint is SHA256:dXsAE5SaInFUaPinoxhcuNloPhb2/x2JhoGVdcF8Y6I.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'gift.hmv' (ED25519) to the list of known hosts.
root@gift.hmv's password:
Permission denied, please try again.
root@gift.hmv's password:
Permission denied, please try again.
root@gift.hmv's password:
root@gift.hmv: Permission denied (publickey,password,keyboard-interactive).
```

This allowed me to instantly observe whether the SSH service is accessible and what type of authentication was required (password or key-based).

Exploitation

Credential-Based Access (Weak Password Guessing)

The website message was *Don't overthink, it's simple.* So, I thought, why not try simple? I tested logging in as 'root' with a basic password 'simple', and it worked. I got in.

```
┌──(kali㉿kali)-[~/hackmyvm/gift]
└─$ ssh root@gift.hmv
root@gift.hmv's password:
IM AN SSH SERVER
gift:~# id
uid=0(root) gid=0(root)
groups=0(root),0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel),11(floppy),20(dialout),26(tape),27(video)
gift:~# ls -lah
total 20K
drwx------    2 root     root        4.0K Sep 24  2020 .
drwxr-xr-x   22 root     root        4.0K Sep 18  2020 ..
-rw-------    1 root     root          20 Aug 24 00:35 .ash_history
----------    1 root     root          12 Sep 24  2020 root.txt
-rw-rw----    1 root     root          12 Sep 24  2020 user.txt
gift:~#
```

Flags

User and Root flags

```
gift:~# cat root.txt
[redacted]
gift:~# cat user.txt
[redacted]
gift:~#
```