Collage of Computers and Information Technology
Department: Computer Science

Distributed Bruteforce

Rayan Hatim Althobati

43307522

ريان حاتم عائض الثبيتي

Supervisor

Dr. Arif M. Bhatti

Document Approval:

The following Software Requirements Specification has been accepted and approved by the following:

Student Name	student ID	Date	Signature
Rayan Hatim Althobati	43307522		

REPORT SUMMARY

In recent days, my server has become a target to ever-more brute-force attacks on SSH service and they are succeeded to crack my password using a massive number of botnet that helps to reduce the complexity of the passwords. In other words, it was just one IP that attempted a login, then another IP would attempt the next password, then another for the next etc. This means that we can't rely on normal firewalls. The project goal is to figure out how to block them and how to apply attacks.

Table of Contents

Chapter 3 Linux Container

Chapter 4 Apply distributed brute force

Chapter 5 Konckd Service

5.1 Introduction.

5.2 Purpose.

5.3 How it’s works.

5.4 Install knockd service.

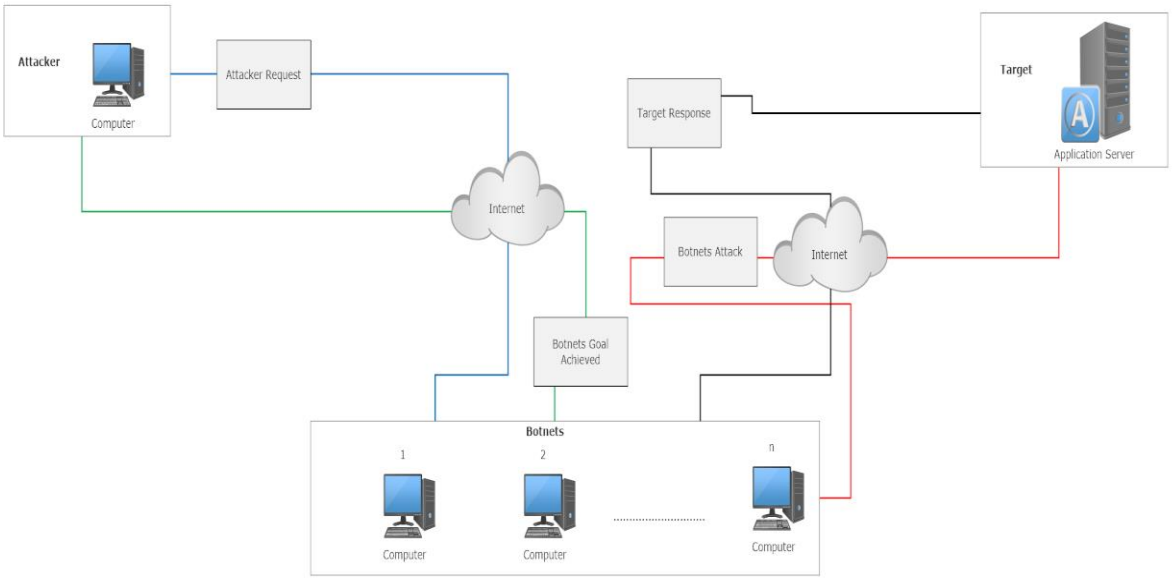
5.5 Attack on target that apply knockd service.

Chapter 1

Introduction

1.1 Introduction

In these days, everyone has a devices that are connected to the internet and at least everyone has more than one device and may they are vulnerable, misconfigured, misused, out of date, such as routers and IOT devices and software, that makes the hackers use those devices for a specific purpose like distributed brute-force or distributed denial of service or sniffing traffics or mass identity theft or spreading new malware. In this project, we focus on distributed brute force case which is



Assume that the SSH server password length is six characters contains 0-9 and a-z the complexity will be (36^6) = 2,176,782,336 possible password and suppose every botnet have one core at least and the SSH request takes 0.5 second, The time to crack the password based on the next Statistics

Botnet	complexity	Hours
500	2176782336	1209.32352
1000	2176782336	604.66176
1500	2176782336	403.10784
2000	2176782336	302.33088
2500	2176782336	241.864704
3000	2176782336	201.55392
3500	2176782336	172.7605029
4000	2176782336	151.16544
4500	2176782336	134.36928
5000	2176782336	120.932352
5500	2176782336	109.9385018
6000	2176782336	100.77696
6500	2176782336	93.02488615
7000	2176782336	86.38025143
7500	2176782336	80.621568
8000	2176782336	75.58272
8500	2176782336	71.13667765
9000	2176782336	67.18464
9500	2176782336	63.64860632
10000	2176782336	60.466176
10500	2176782336	57.58683429
11000	2176782336	54.96925091
11500	2176782336	52.57928348
12000	2176782336	50.38848
12500	2176782336	48.3729408
13000	2176782336	46.51244308
13500	2176782336	44.78976
14000	2176782336	43.19012571
14500	2176782336	41.70081103
15000	2176782336	40.310784
15500	2176782336	39.01043613
16000	2176782336	37.79136
16500	2176782336	36.64616727
17000	2176782336	35.56833882
17500	2176782336	34.55210057
18000	2176782336	33.59232
18500	2176782336	32.68441946
19000	2176782336	31.82430316
19500	2176782336	31.00829538
20000	2176782336	30.233088
20500	2176782336	29.49569561
21000	2176782336	28.79341714
21500	2176782336	28.12380279
22000	2176782336	27.48462545
22500	2176782336	26.873856
23000	2176782336	26.28964174
23500	2176782336	25.73028766
24000	2176782336	25.19424
24500	2176782336	24.68007184
25000	2176782336	24.1864704
25500	2176782336	23.71222588
26000	2176782336	23.25622154
26500	2176782336	22.81742491
27000	2176782336	22.39488
27500	2176782336	21.98770036

1.2 Purpose

Learn multiprocessing and multithreading programming.

Learn how to do LXD.

Learn socket programming.

Defend against distributed brute force.

Learn how to use Excel and write excel functions.

1.3 Scope

Defend against distributed brute force.

1.4 Definitions, Acronyms, and Abbreviations:

SSH: Secure Shell.

HTTP: Hyper Text Transfer Protocol.

Brute force: consists of an attacker trying many passwords or passphrases with the hope of eventually guessing correctly. The attacker systematically checks all possible passwords and passphrases until the correct one is found.

LXD: is a daemon which offers a REST API to drive full system containers just like you would drive virtual machines.

Botnet: is a collection of internet-connected devices, which may include PCs, servers, mobile devices and internet of things devices that are infected and controlled by a common type of Malware.

1.5 Related Subjects:

http://thehackernews.com/2017/05/cryptocurrency-mining-botnet.html

http://thehackernews.com/2016/09/ddos-attack-iot.html

http://thehackernews.com/2017/04/vigilante-hacker-iot-botnet_26.html

Chapter 2

Setup Environment

2.1 Install Ubuntu

Ubuntu: is a Debian-based Linux operating system for personal computers. Used for defending from the attacker.

Tutorial:
https://www.ubuntu.com/download/desktop/install-ubuntu-desktop

2.2 Install Python

sudo apt-get update
sudo apt-get install python3

library requirements:
Install pyqt, QT version 4.0
Install requests
Install colorama
Install pyfiglet
Install termcolor
Install argparse
Install paramiko
Install netaddr

2.3 Install LXD

sudo apt-get install lxd
Then either logout and login again to get your group membership refreshed, or use:
newgrp lxd

2.4 Install LXD

Sudo apt-get install apache2

Chapter 3

Linux Containers

3.1 Introduction

What is VM

VM stand for Virtual Machine, Virtual Machine is a software use a full copy of an operating system and create virtual copy of all the hardware that the operating system needs to run. that use a lot of Memory and CPU and storage space.

What is LXC

LXC stand for Linux containers, all containers is type of virtualization using one single kernel for all containers, it's just virtualize the software environment and you not virtualize or copy the hardware at all, and Linux containers shared all the same memory and storage, resources.

What is LXD

LXD is a daemon which offers a REST API to drive full system containers just like you would drive virtual machines.

The LXD daemon runs on every container host and client tools then connect to those to manage those containers or to move or copy them to another LXD.

Why LXC and Not VM

LXC is fast at boot, reboot.
LXC don't waste the memory space and storage space, CPU, like VM.

Why we need LXD with LXC

LXC has been around for about 7 years now, over those years a lot of security Issues effects on LXC, LXD add extra security features were added to the Linux kernel and LXC grew support for all of them. As we saw the need for people to build their own solution on top of LXC, the LXD developers developed a public API and a set of bindings.

Make LXC secure by default (rather than it being optional).

Completely rework the tools to make them simpler and less confusing to newcomers.

Rely on container images rather than using "templates" to build them locally.

Proper checkpoint/restore support (live migration).

When you need to use VM and when you need to use LXC

Use VM if you want to virtual an operating system that have different kernel that your host use.
Use LXC if you want to virtual an operating system that have same kernel that your host use.

3.2 Purpose

Avoid wasting memory and CPU and to handle more computers.

3.3 Creation and Management

Add Images to create Containers:

lxc remote add images images.linuxcontainers.org

List Available images:

lxc image list

Create a Container:

lxc launch images[imageName] [containerName]

List the Containers:

lxc list

List the Configuration of Container:

lxc config show [containerName] --expanded

Change the Configuration of Container:

lxc config edit [containerName]

Create A Snapshot:

lxc snapshot [ContainerName] [snapshot Name]

Change Container State (start, stop, pause):

lxc [state][containerName]

Enter Bash on a Container:

lxc exec [containerName] bash

Manage remote lxd servers:

lxc remote add [ContainerName] [IP]

Create a container inside a counter from the Main Container:

lxc launch images:[imageName]
[ParentContainerName]:[ChildContainerName]

Move A Child Container Inside Another Container:

lxc move [ParentContainer]:[ChildContainer] [newParentContainer]:

List the Containers Inside Container:

lxc list [containerName]:

Stop or Delete a Container:

lxc stop [containerName]
lxc delete [containerName]

Set and Unset a device to container:

lxc config device set [containerName] <device> <key> <value>

examples:

<Set the Ethernet interface 0 device to Container with maximum limit 100Mbit>

lxc config device set container1 eth0 limit.ingress 100Mbit

lxc config device unset container1 eth0 limit.ingress

<resize the home path to 10GB>

lxc config device set container1 home size 10GB

Push or Pull or Edit a file in the Container:

lxc file push [Hostfilename] [containername][FilePath]

lxc file pull [filename] [containername][FilePath]

lxc file edit [filename] [containername][FilePath]

3.4 References

An introduction to LXD, the container lighter-visor - Stéphane Graber

https://www.youtube.com/watch?v=yEr_EIZG0ZM

Linux Container (LXC) Introduction Eli The Computer Guy

https://www.youtube.com/watch?v=_KnmRdK69qM

Chapter 4

Apply distributed brute force

4.1 Introduction

I wrote a program called Simple botnet(Sbotnet) that use for collecting a botnet and use them to brute force a SSH application server.

4.2 Purpose

Its main goals are to crack a SSH server application using botnet.

4.3 Create penetration Environment

Step 1: create Five containers and install SSH server on those containers and create a username called (default) with password (default) and with install python requests library and paramiko, and cryptography.

create the first container which called botnet

```
root@diefunction:~# lxc launch ubuntu:14.04 botnet
Creating botnet
Starting botnet
root@diefunction:~# lxc list
+--------+---------+----------------------+----------------------------------------------+------------+-----------+
|  NAME  |  STATE  |         IPV4         |                    IPV6                      |    TYPE    | SNAPSHOTS |
+--------+---------+----------------------+----------------------------------------------+------------+-----------+
| botnet | RUNNING | 10.207.176.13 (eth0) | fd9:301:54d5:a33b:216:3eff:feac:c19d (eth0) | PERSISTENT | 0         |
+--------+---------+----------------------+----------------------------------------------+------------+-----------+
root@diefunction:~#
```

Login to container bash

```
root@diefunction:~# lxc exec botnet bash
root@botnet:~#
```

Add user that called default and password is default

```
root@botnet:~# adduser default
Adding user `default' ...
Adding new group `default' (1001) ...
Adding new user `default' (1001) with group `default' ...
Creating home directory `/home/default' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for default
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
root@botnet:~#
```

Install SSH and paramiko and requests

```
root@botnet:~# apt-get install ssh; apt-get install python-paramiko; apt-get install python-requests;
```

Install python cryptography requirements

```
root@botnet:~# apt-get install build-essential libssl-dev libffi-dev python-dev
```

```
root@botnet:~# python -m pip install pyparsing
Downloading/unpacking pyparsing
  Downloading pyparsing-2.2.0-py2.py3-none-any.whl (56kB):  79%  45kB
```

```
root@botnet:~# python -m pip install appdirs
Downloading/unpacking appdirs
```

```
root@botnet:~# apt-get install python-cffi
```

```
root@botnet:~# pip install --upgrade six
```

```
root@botnet:~# python -m pip install cryptography
Downloading/unpacking cryptography
  Downloading cryptography-1.8.1.tar.gz (423kB): 423kB downloaded
  Running setup.py (path:/tmp/pip_build_root/cryptography/setup.py) egg_info for package cryptography

    no previously-included directories found matching 'docs/_build'
    warning: no previously-included files matching '*' found under directory 'vectors'
Downloading/unpacking idna>=2.1 (from cryptography)
  Downloading idna-2.5-py2.py3-none-any.whl (55kB): 55kB downloaded
```

Configure SSH, change PasswordAuthentication no to yes.

```
root@botnet:~# nano /etc/ssh/sshd_config
```

```
# Change to no to disable tunnelled clear text passwords
PasswordAuthentication yes
```

now create from this container four containers to be total five containers

```
root@diefunction:~# lxc list
+--------+---------+----------------------+----------------------------------------------+------------+-----------+
|  NAME  |  STATE  |         IPV4         |                     IPV6                     |    TYPE    | SNAPSHOTS |
+--------+---------+----------------------+----------------------------------------------+------------+-----------+
| botnet | RUNNING | 10.207.176.13 (eth0) | fd9:301:54d5:a33b:216:3eff:feac:c19d (eth0) | PERSISTENT | 0         |
+--------+---------+----------------------+----------------------------------------------+------------+-----------+
root@diefunction:~# lxc copy botnet botnet1
root@diefunction:~# lxc copy botnet botnet2
root@diefunction:~# lxc copy botnet botnet3
root@diefunction:~# lxc copy botnet botnet4
root@diefunction:~# lxc start botnet1 botnet2 botnet3 botnet4
root@diefunction:~# lxc list
+---------+---------+----------------------+----------------------------------------------+------------+-----------+
|  NAME   |  STATE  |         IPV4         |                     IPV6                     |    TYPE    | SNAPSHOTS |
+---------+---------+----------------------+----------------------------------------------+------------+-----------+
| botnet  | RUNNING | 10.207.176.13 (eth0)  | fd9:301:54d5:a33b:216:3eff:feac:c19d (eth0) | PERSISTENT | 0         |
+---------+---------+----------------------+----------------------------------------------+------------+-----------+
| botnet1 | RUNNING | 10.207.176.254 (eth0) | fd9:301:54d5:a33b:216:3eff:feca:b408 (eth0) | PERSISTENT | 0         |
+---------+---------+----------------------+----------------------------------------------+------------+-----------+
| botnet2 | RUNNING |                      | fd9:301:54d5:a33b:216:3eff:fe5e:48ed (eth0) | PERSISTENT | 0         |
+---------+---------+----------------------+----------------------------------------------+------------+-----------+
| botnet3 | RUNNING |                      | fd9:301:54d5:a33b:216:3eff:fe58:3065 (eth0) | PERSISTENT | 0         |
+---------+---------+----------------------+----------------------------------------------+------------+-----------+
| botnet4 | RUNNING |                      | fd9:301:54d5:a33b:216:3eff:fe54:bac0 (eth0) | PERSISTENT | 0         |
+---------+---------+----------------------+----------------------------------------------+------------+-----------+
root@diefunction:~#
```

Step 2: create another container and install SSH server and create a username called (unsecure) with password (unsecure).

Create unsecure container

```
root@diefunction:~# lxc launch ubuntu:14.04 unsecured
Creating unsecured
Starting unsecured
root@diefunction:~# lxc list
+-----------+---------+----------------------+----------------------------------------------+------------+-----------+
|   NAME    |  STATE  |         IPV4         |                    IPV6                      |    TYPE    | SNAPSHOTS |
+-----------+---------+----------------------+----------------------------------------------+------------+-----------+
| botnet    | RUNNING | 10.207.176.13 (eth0) | fd9:301:54d5:a33b:216:3eff:feac:c19d (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+----------------------------------------------+------------+-----------+
| botnet1   | RUNNING | 10.207.176.254 (eth0)| fd9:301:54d5:a33b:216:3eff:feca:b408 (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+----------------------------------------------+------------+-----------+
| botnet2   | RUNNING | 10.207.176.206 (eth0)| fd9:301:54d5:a33b:216:3eff:fe5e:48ed (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+----------------------------------------------+------------+-----------+
| botnet3   | RUNNING | 10.207.176.227 (eth0)| fd9:301:54d5:a33b:216:3eff:fe58:3065 (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+----------------------------------------------+------------+-----------+
| botnet4   | RUNNING | 10.207.176.25 (eth0) | fd9:301:54d5:a33b:216:3eff:fe54:bac0 (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+----------------------------------------------+------------+-----------+
| unsecured | RUNNING | 10.207.176.61 (eth0) | fd9:301:54d5:a33b:216:3eff:feb9:1e2e (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+----------------------------------------------+------------+-----------+
root@diefunction:~#
```

Add unsecure user with unsecure password

```
root@unsecured:~# adduser unsecure
Adding user `unsecure' ...
Adding new group `unsecure' (1001) ...
Adding new user `unsecure' (1001) with group `unsecure' ...
The home directory `/home/unsecure' already exists.  Not copying from `/etc/skel'.
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for unsecure
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
root@unsecured:~#
```

Install SSH

```
root@diefunction:~# lxc exec unsecured bash
root@unsecured:~# apt-get install ssh
Reading package lists... Done
Building dependency tree
Reading state information... Done
ssh is already the newest version.
The following packages were automatically installed and are no longer required:
  libfreetype6 os-prober
Use 'apt-get autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@unsecured:~# nano /etc/ssh/sshd_config
```

Enable SSH PasswordAuthentication

```
# Change to no to disable tunnelled clear text passwords
PasswordAuthentication yes
```

Step 3: create another container and install SSH server and create a username called (secure) with password (secure).

Create secure container

```
root@diefunction:~# lxc launch ubuntu:14.04 secure
Creating secure
Starting secure
root@diefunction:~#
```

Add secure user with secure password

```
root@secure:~# adduser secure
Adding user `secure' ...
Adding new group `secure' (1001) ...
Adding new user `secure' (1001) with group `secure' ...
Creating home directory `/home/secure' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for secure
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
root@secure:~#
```

Install SSH

```
root@diefunction:~# lxc exec secure bash
root@secure:~# apt-get install ssh
```

Enable SSH PasswordAuthentication

```
root@botnet:~# nano /etc/ssh/sshd_config
```

```
# Change to no to disable tunnelled clear text passwords
PasswordAuthentication yes
```

At the end the list all containers

```
root@diefunction:~# lxc list
+-----------+---------+----------------------+-----------------------------------------------+------------+-----------+
|   NAME    |  STATE  |         IPV4         |                     IPV6                      |    TYPE    | SNAPSHOTS |
+-----------+---------+----------------------+-----------------------------------------------+------------+-----------+
| botnet    | RUNNING | 10.207.176.13 (eth0) | fd9:301:54d5:a33b:216:3eff:feac:c19d (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+-----------------------------------------------+------------+-----------+
| botnet1   | RUNNING | 10.207.176.254 (eth0) | fd9:301:54d5:a33b:216:3eff:feca:b408 (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+-----------------------------------------------+------------+-----------+
| botnet2   | RUNNING | 10.207.176.206 (eth0) | fd9:301:54d5:a33b:216:3eff:fe5e:48ed (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+-----------------------------------------------+------------+-----------+
| botnet3   | RUNNING | 10.207.176.227 (eth0) | fd9:301:54d5:a33b:216:3eff:fe58:3065 (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+-----------------------------------------------+------------+-----------+
| botnet4   | RUNNING | 10.207.176.25 (eth0)  | fd9:301:54d5:a33b:216:3eff:fe54:bac0 (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+-----------------------------------------------+------------+-----------+
| secure    | RUNNING | 10.207.176.118 (eth0) | fd9:301:54d5:a33b:216:3eff:feb1:20c4 (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+-----------------------------------------------+------------+-----------+
| unsecured | RUNNING | 10.207.176.61 (eth0)  | fd9:301:54d5:a33b:216:3eff:feb9:1e2e (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+-----------------------------------------------+------------+-----------+
root@diefunction:~#
```

Step 4: download our software Sbotnet.

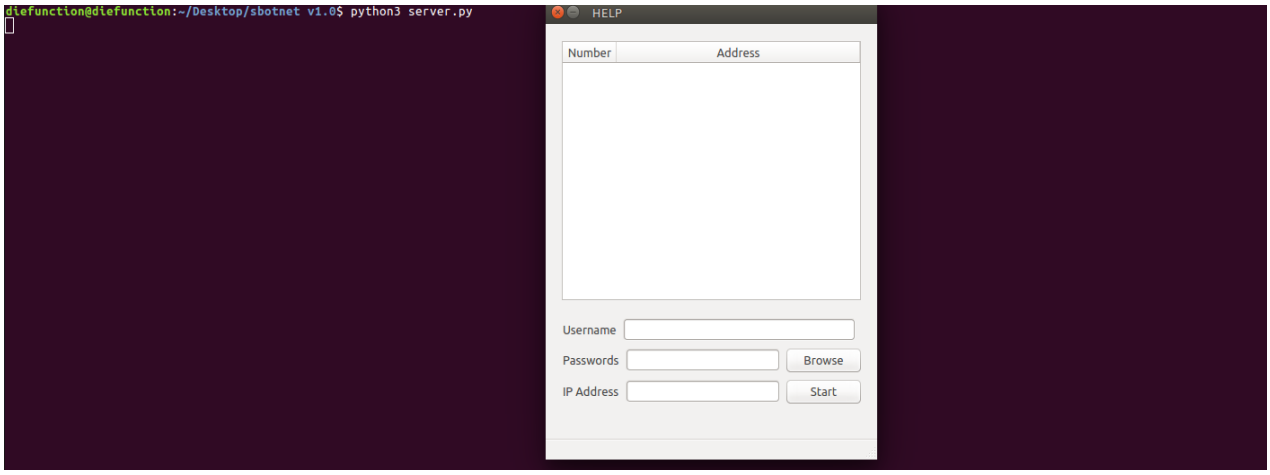
Download link :.

Create sbotnet folder in /var/www/html/.

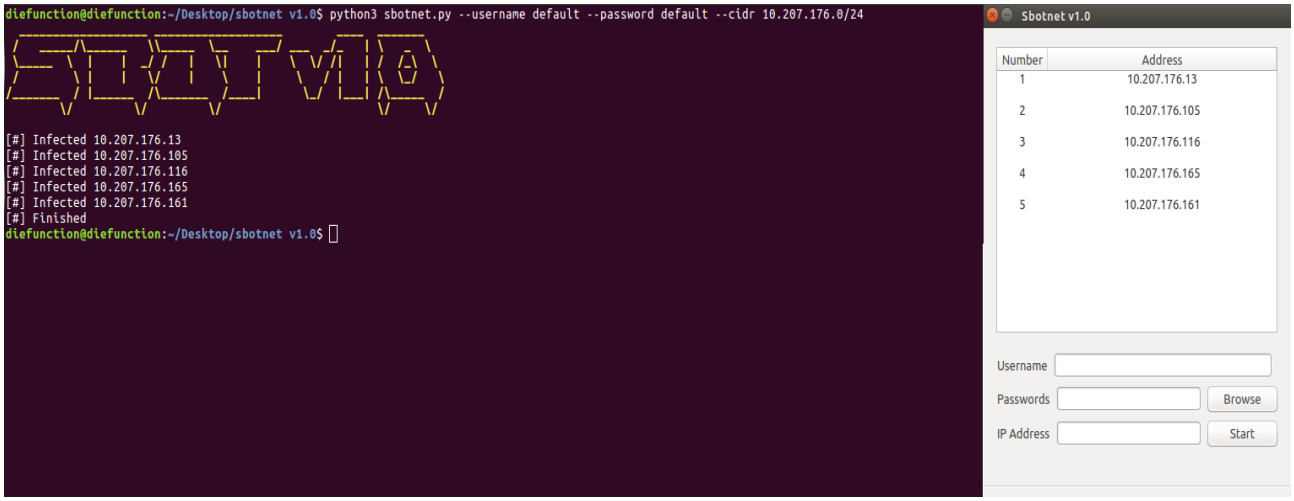
Move index.php to /var/www/html/sbotnet/

4.4 Collect botnet

Step 1: Run server.py



Step 2: In new Terminal execute python –username default –password default –cidr 10.207.176.0/24. 10.207.176.0/24 is an example you can change the it.

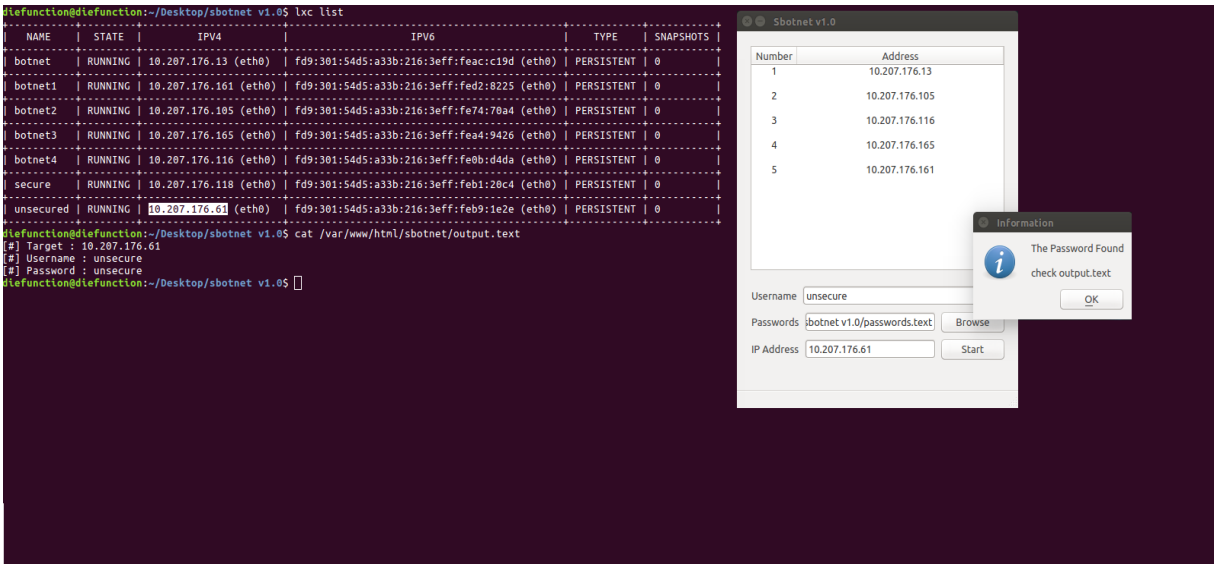


4.5 Attack on target

Step 1: Put the IP of the unsecure container that we create in the server application.

Step 2: Choose the password file that contain the unsecure container password.

Step 3: Put the username of the unsecure container.



The distributed brute-force cracked the unsecure container.

Chapter 5

Knockd Service

5.1 Introduction

Port Knocking is one method of obscuring the services that you have running on your machine, until you ask for a port to be opened by attempt to connect on a specific sequence of ports then Open the port to the IP address that supplied the correct knock.

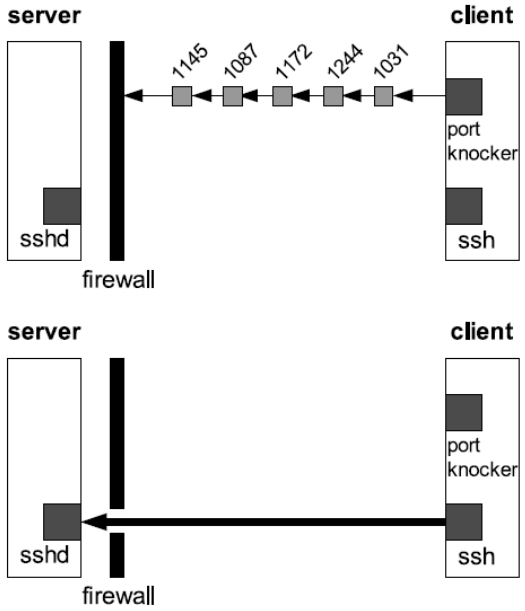
5.2 Purpose

to Hide your SSH Daemon from Attackers

5.3 How it's works

Port knocking will not open any ports on the server by default.

just waiting for a sequence of port traffic.



5.4 Install knockd Service

Install knockd

```
diefunction@diefunction:~$ lxc exec secure bash
root@secure:~# apt-get install knockd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libfreetype6 os-prober
Use 'apt-get autoremove' to remove them.
The following NEW packages will be installed:
  knockd
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 28.9 kB of archives.
After this operation, 176 kB of additional disk space will be used.
0% [Connecting to archive.ubuntu.com (2001:67c:1360:8001::17)]
```

First, we flush existing firewall rules and ensure that outgoing connections don't get dropped.

Second, block incoming port 22 (SSH).

Once you have established your iptables rules, you can automate the restore process at reboot with iptables-persistent.

```
root@secure:~# iptables --flush
root@secure:~# iptables -t nat --flush
root@secure:~# iptables -t mangle --flush
root@secure:~# iptables --policy OUTPUT ACCEPT
root@secure:~# iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
root@secure:~# iptables -A INPUT -p tcp --destination-port 22 -j DROP
root@secure:~# apt-get install iptables-persistent
```

```
root@secure:~# iptables-save
# Generated by iptables-save v1.4.21 on Mon May  8 14:31:07 2017
*mangle
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
COMMIT
# Completed on Mon May  8 14:31:07 2017
# Generated by iptables-save v1.4.21 on Mon May  8 14:31:07 2017
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
COMMIT
# Completed on Mon May  8 14:31:07 2017
# Generated by iptables-save v1.4.21 on Mon May  8 14:31:07 2017
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m tcp --dport 22 -j DROP
COMMIT
# Completed on Mon May  8 14:31:07 2017
root@secure:~#
```

to configure the sequence port that the client must knock to open the ssh service.

And choose the sequence timeout so that the client must knock the sequence < timeout

The choose cmd timeout that if the client not connect to the SSH after opened it 10 second. will be close.

```
root@secure:~# nano /etc/knockd.conf
```

```
[options]
        UseSyslog

[SSH]
    sequence = 3000,4000,5000,6000
    tcpflags = syn
    seq_timeout = 15
    start_command = /sbin/iptables -I INPUT 1 -s %IP% -p tcp --dport 22 -j ACCEPT
    cmd_timeout = 10
    stop_command = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 22 -j ACCEPT

File Name to Write: /etc/knockd.conf
```

start Knockd service

```
root@secure:~# nano /etc/default/knockd
```

```
################################################
#
# knockd's default file, for generic sys config
#
################################################

# control if we start knockd at init or not
# 1 = start
# anything else = don't start
#
# PLEASE EDIT /etc/knockd.conf BEFORE ENABLING
START_KNOCKD=1

# command line options
#KNOCKD_OPTS="-i eth1"

File Name to Write: /etc/default/knockd
```

```
root@secure:~# sudo service knockd restart
```

Test SSH Service after knockd service installed and configured.

```
diefunction@diefunction:~$ lxc list
+-----------+---------+----------------------+----------------------------------------------+------------+-----------+
|   NAME    |  STATE  |         IPV4         |                    IPV6                      |    TYPE    | SNAPSHOTS |
+-----------+---------+----------------------+----------------------------------------------+------------+-----------+
| botnet    | RUNNING | 10.207.176.13 (eth0) | fd9:301:54d5:a33b:216:3eff:feac:c19d (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+----------------------------------------------+------------+-----------+
| botnet1   | RUNNING | 10.207.176.161 (eth0) | fd9:301:54d5:a33b:216:3eff:fed2:8225 (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+----------------------------------------------+------------+-----------+
| botnet2   | RUNNING | 10.207.176.105 (eth0) | fd9:301:54d5:a33b:216:3eff:fe74:70a4 (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+----------------------------------------------+------------+-----------+
| botnet3   | RUNNING | 10.207.176.165 (eth0) | fd9:301:54d5:a33b:216:3eff:fea4:9426 (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+----------------------------------------------+------------+-----------+
| botnet4   | RUNNING | 10.207.176.116 (eth0) | fd9:301:54d5:a33b:216:3eff:fe0b:d4da (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+----------------------------------------------+------------+-----------+
| secure    | RUNNING | 10.207.176.118 (eth0) | fd9:301:54d5:a33b:216:3eff:feb1:20c4 (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+----------------------------------------------+------------+-----------+
| unsecured | RUNNING | 10.207.176.61 (eth0)  | fd9:301:54d5:a33b:216:3eff:feb9:1e2e (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+----------------------------------------------+------------+-----------+
diefunction@diefunction:~$ ssh secure@10.207.176.118
```

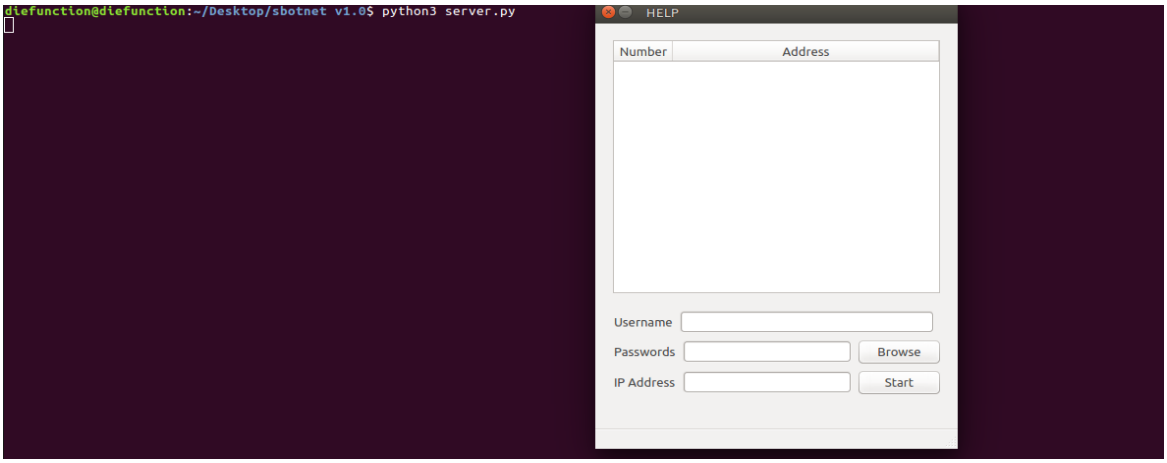
no responding it's mean the knockd service is working. Try to knock then connect.

```
diefunction@diefunction:~$ lxc list
+-----------+---------+----------------------+----------------------------------------------+------------+-----------+
|   NAME    |  STATE  |         IPV4         |                    IPV6                      |    TYPE    | SNAPSHOTS |
+-----------+---------+----------------------+----------------------------------------------+------------+-----------+
| botnet    | RUNNING | 10.207.176.13 (eth0) | fd9:301:54d5:a33b:216:3eff:feac:c19d (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+----------------------------------------------+------------+-----------+
| botnet1   | RUNNING | 10.207.176.161 (eth0) | fd9:301:54d5:a33b:216:3eff:fed2:8225 (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+----------------------------------------------+------------+-----------+
| botnet2   | RUNNING | 10.207.176.105 (eth0) | fd9:301:54d5:a33b:216:3eff:fe74:70a4 (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+----------------------------------------------+------------+-----------+
| botnet3   | RUNNING | 10.207.176.165 (eth0) | fd9:301:54d5:a33b:216:3eff:fea4:9426 (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+----------------------------------------------+------------+-----------+
| botnet4   | RUNNING | 10.207.176.116 (eth0) | fd9:301:54d5:a33b:216:3eff:fe0b:d4da (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+----------------------------------------------+------------+-----------+
| secure    | RUNNING | 10.207.176.118 (eth0) | fd9:301:54d5:a33b:216:3eff:feb1:20c4 (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+----------------------------------------------+------------+-----------+
| unsecured | RUNNING | 10.207.176.61 (eth0)  | fd9:301:54d5:a33b:216:3eff:feb9:1e2e (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+----------------------------------------------+------------+-----------+
diefunction@diefunction:~$ ssh secure@10.207.176.118
^C
diefunction@diefunction:~$ ssh secure@10.207.176.118
\^C
diefunction@diefunction:~$ knock 10.207.176.118 3000 4000 5000 6000 && ssh secure@10.207.176.118
secure@10.207.176.118's password:
```

Success it's ask for the password after knocking the sequence that we choose.

5.5 Attack on target that apply knockd service

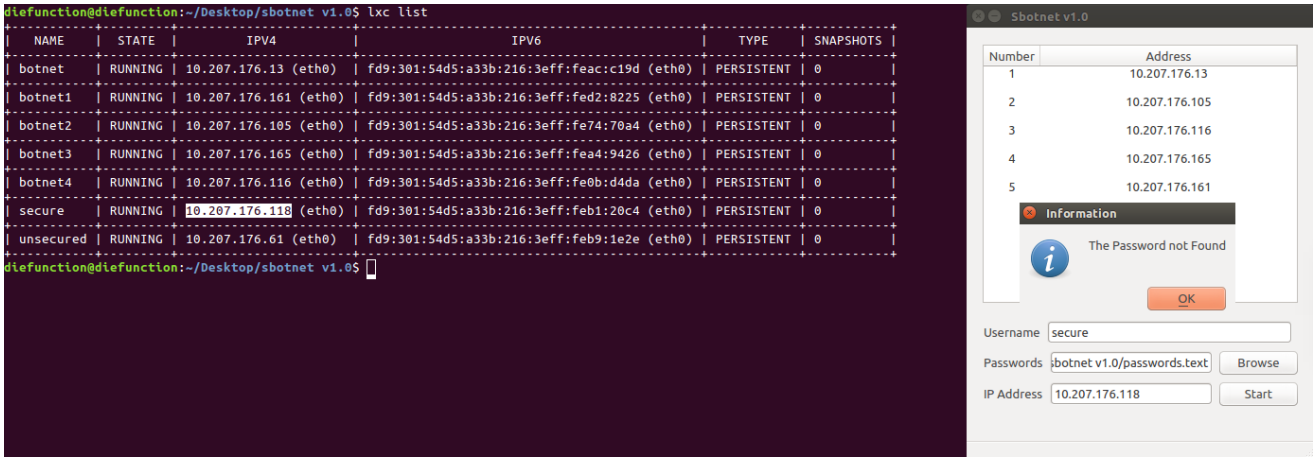
Step 1: Run server.py.



Step 2: put the IP of the secure container that we create.

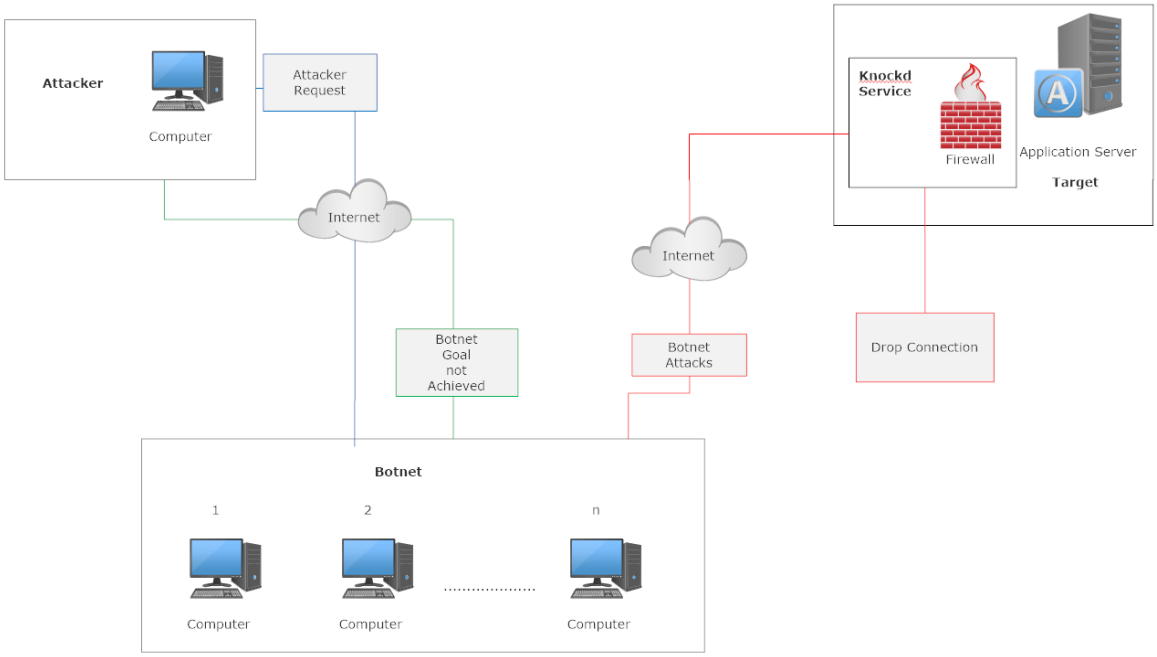
Step 3: choose the password file that contain the secure container password.

Step 4: put the username of the secure container.



the password failed even when the password is in the passwords list.

After we apply the knockd service the problem solved.



Conclusion

We will continue to see how to secure the LXD containers.