# LINUX BACKDOOR

By:

Jeffrey Sasaki

British Columbia Institute of Technology

COMP 8505 – Assignment 3

Aman Abdulla

May 22, 2015

# Contents

# Introduction

A backdoor is perceived as a negative vulnerability because it allows an attacker to obtain access to a victim's machine without proper credentials. However, a backdoor is more than just a tool of exploitation, it is used far more commonly than one may think.

Generally speaking, the purpose of a backdoor is to allow access to a machine, implemented into the program by the programmer. This is without a doubt a security flaw, however, it is also a tool used for debugging and analytical purposes.

This assignment demonstrates a backdoor program where the attacker is capable of executing shell commands on the victim's machine and returns the response to the attacker.

## Requirements

- Backdoor must camouflage itself so as to deceive anyone looking at the process table.
- Application must ensure that it only receives (authenticate) those packets that are meant for the backdoor itself.
- The backdoor must interpret commands sent to it, execute them and send the results back.
- Incorporate an encryption scheme of your choice into the backdoor.

# Implementation

The program is written in python. There are two programs included in this assignment:

1. client.py (Attacker)
2. server.py (Backdoor Victim)

The client (attacker) program establishes a connection to the server (Victim) and will be able to execute Linux commands against the victim's machine. The messages will be encrypted using the AES encryption scheme while sending data to the server. When the victim sends the message back to the client, it will be encrypted once again; hence, the message will be decrypted to plaintext.

The server (victim) will acquire the encrypted data, decrypt it and execute the command. The command will not appear on the victim's message to emulate a hidden backdoor. The server then encrypts that data, again with AES, and transmit the data back to the client.

The program uses two libraries:

1. pycrypto 2.6.1 – For Encryption
2. setproctitle 1.1.8 – To masquerade process title

It is important to note that to ensure that the attacking machine is also authenticating the packets by utilizing the secret key used (for encryption) as a form of a flag. Any other traffic will be ignored altogether.

## Usage

Prior to running the program, the user must install pycrypto 2.6.1 and setproctitle 1.1.8. The two libraries can be downloaded at:

1. https://pypi.python.org/packages/source/p/pycrypto/pycrypto-2.6.1.tar.gz

2. https://pypi.python.org/packages/source/s/setproctitle/setproctitle-1.1.8.tar.gz#md5=728f4c8c6031bbe56083a48594027edd

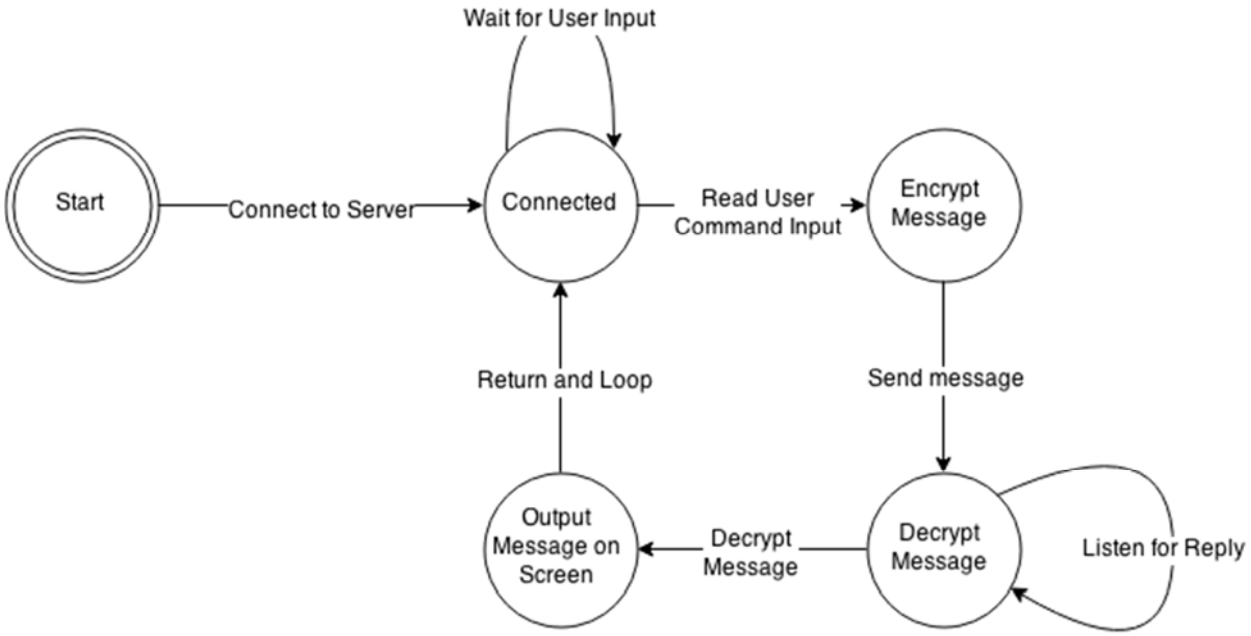
Attacker: $ python client.py -d <host_ip> -p <port>

Victim: $ python server.py -p <port> &

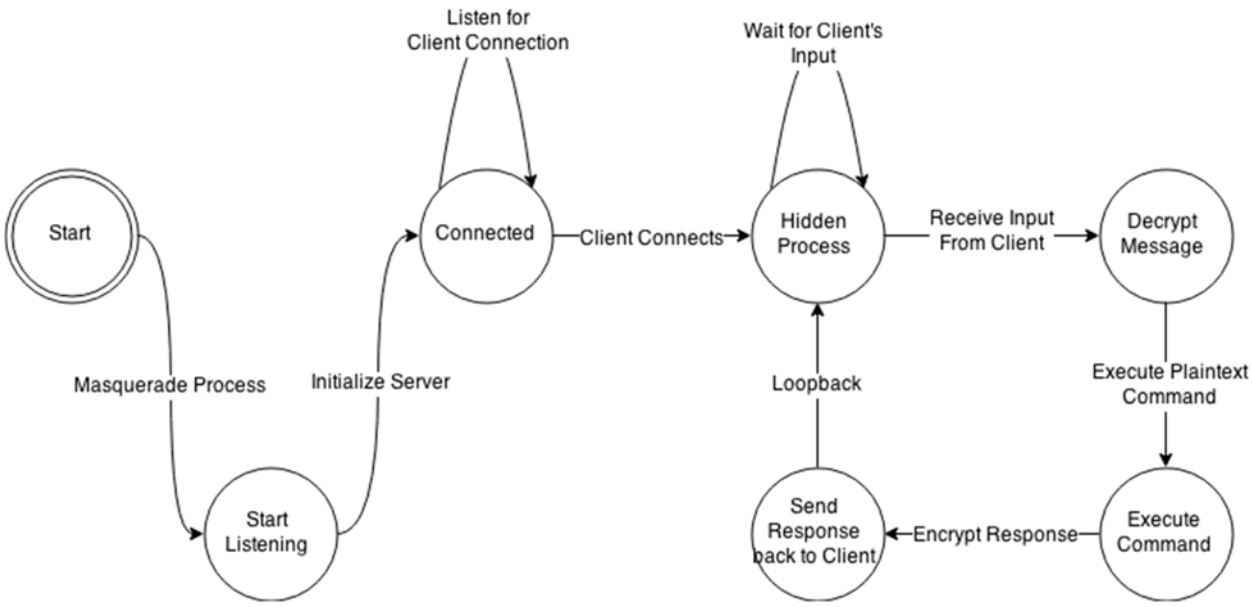
The ampersand (&) will denote that the backdoor will be executed in the background.

## State Diagram

client.py – Attacker



server.py – Victim




## Pseudocode

### client.py

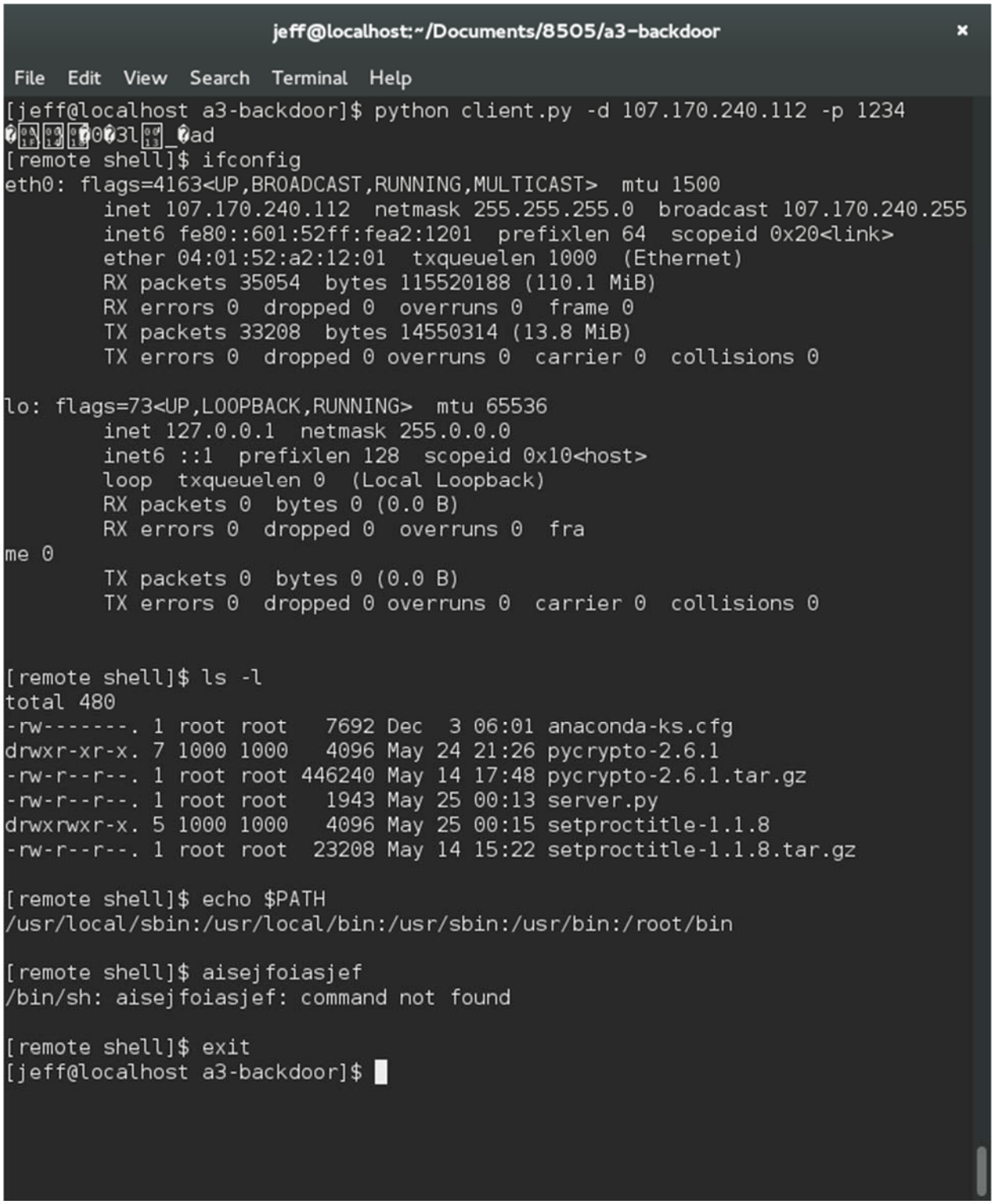
```
Parse command-line argument

Connect to server

While client is connected to the server

      Input command

      Encrypt and Send command

      Decrypt Response
```

### server.py

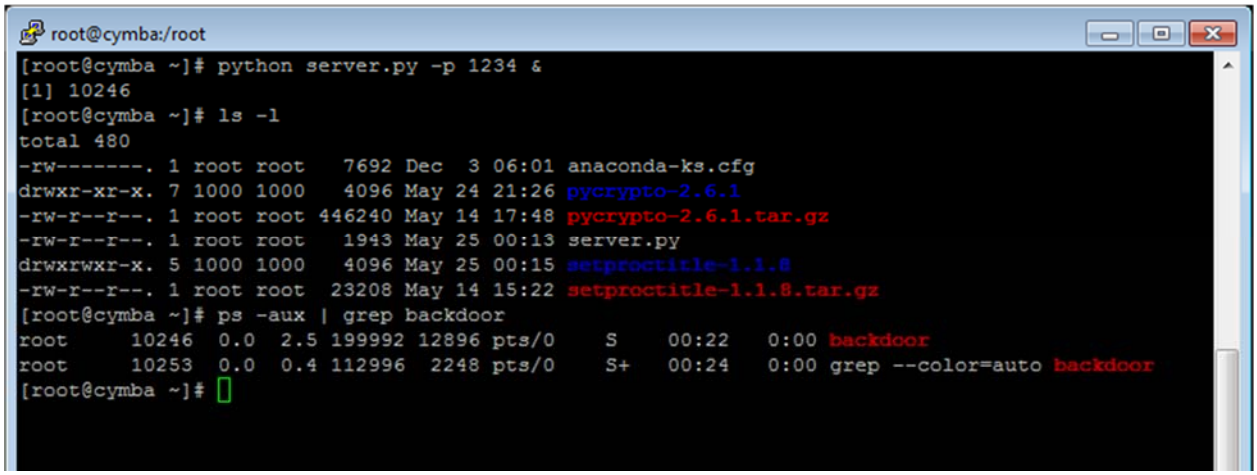
```
Masquerade process

Parse command-line argument
```

```
Listen for connection
While client is connected to the server
      Listen for client’s Input
      Decrypt message and execute command
      Send  command’s response to client
```

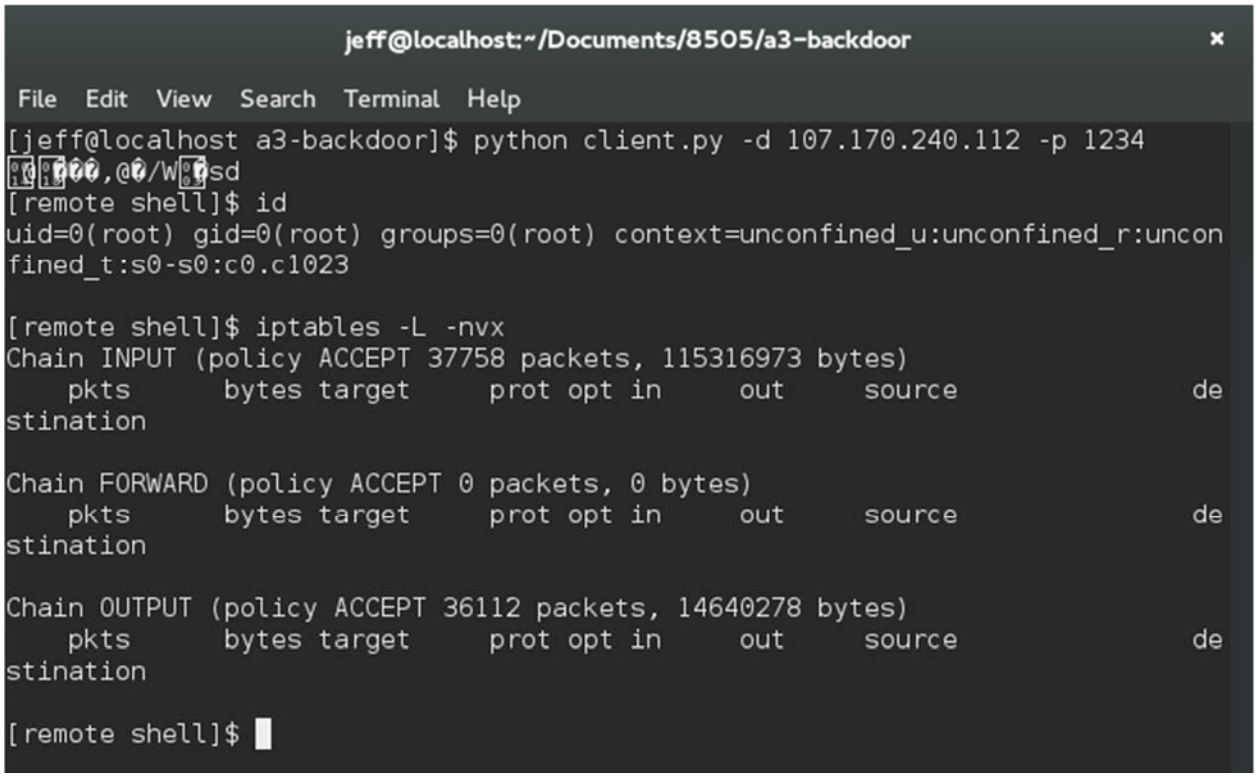
## Testing



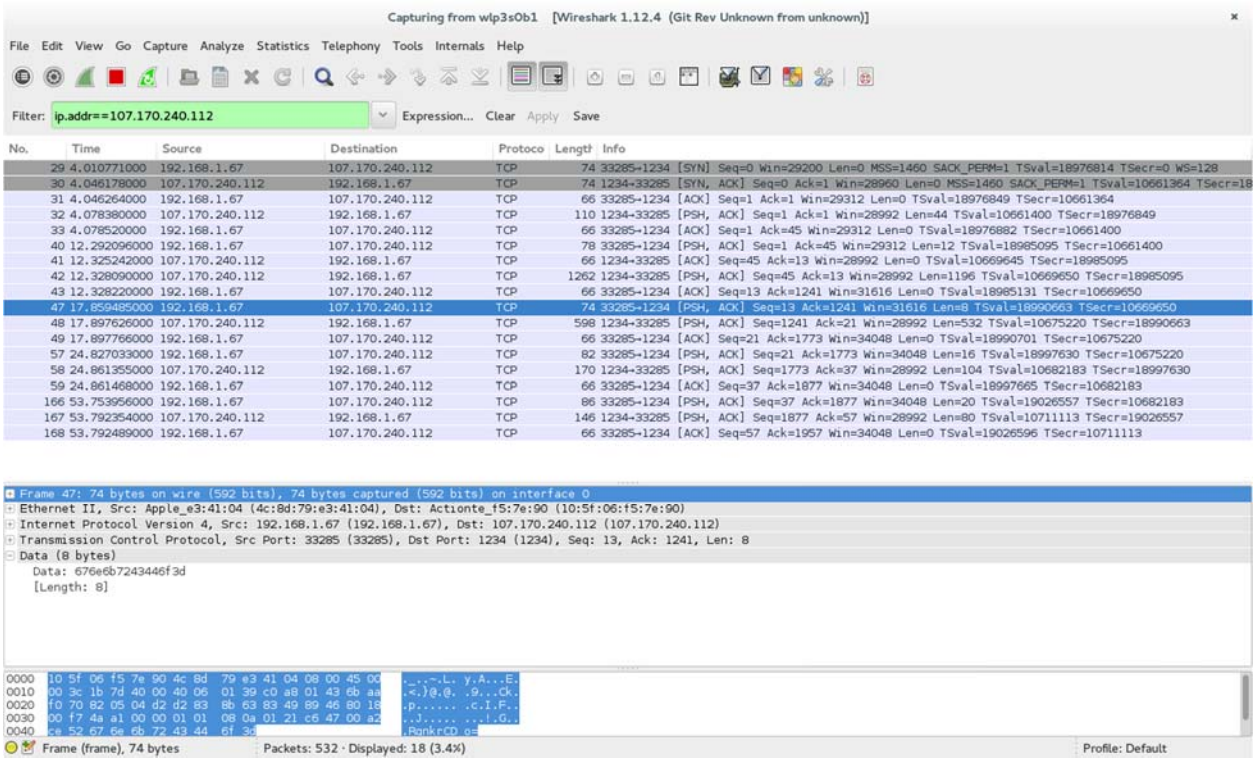

Screenshot of the attacker's screen (client.py)




Screenshot of the victim's machine (server.py)




The id command shows that the attacker has root privilege, and the iptables command (which needs super user privilege to execute) to support it. It also implies that the targeted machine ran the server program as a super user.




Wireshark capture of the data being encrypted during transmission

## Conclusion

The backdoor implementation only works with Linux-based systems. With further enhancements to the program, it is capable of backdooring other operating systems. In addition to that, the “exit” command was implemented into both the client and server program. That confirms that the backdoor program is capable of much more powerful exploitation techniques, such as screenshotting, file transfer, data sniffing, and other malicious activities to the victim’s machine. It is important to note that this is only the basics of a backdoor, where it executes basic commands of a backdoor program.