Covert Communication Application

final project – design work

Danny Lieu

2018

Table of Contents

[1. BACKDOOR 2](#_Toc515474319)

[1.1. DIAGRAM 2](#_Toc515474320)

[1.2. DESCRIPTION 2](#_Toc515474321)

[1.3. PSEUDOCODE 3](#_Toc515474322)

[2. ATTACKER 4](#_Toc515474323)

[2.1. DIAGRAM 4](#_Toc515474324)

[2.2. DESCRIPTION 4](#_Toc515474325)

[2.3. PSEUDOCODE 5](#_Toc515474326)

[3. TIMELINES 5](#_Toc515474327)

[3.1. Milestone #1 (May 29 – June 05) 5](#_Toc515474328)

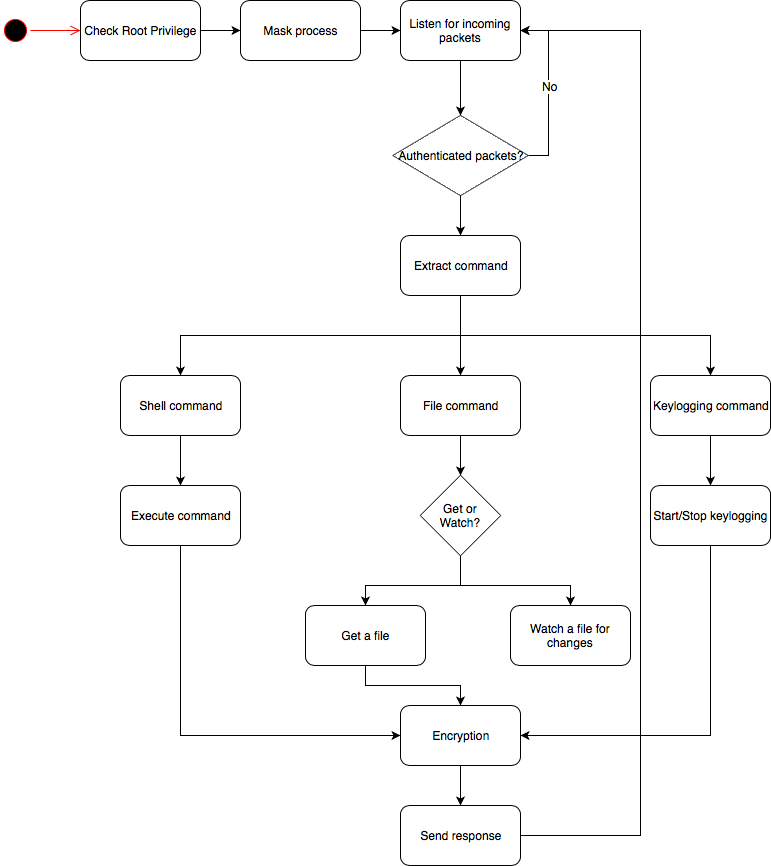
[3.2. Milestone #2 (June 05 – June 12) 6](#_Toc515474329)

[3.3. Milestone #3 (June 12 – June 19) 6](#_Toc515474330)

[3.4. Milestone #4 (June 19 – June 26) 6](#_Toc515474331)

# BACKDOOR

## DIAGRAM



## DESCRIPTION

* Check root privilege: the application requires root access in order to install necessary modules.
* Mask process: run the process as a disguised process that means it will not catch the victim’s attention when he looks at the process table.
* Listen for incoming packets: sniff the traffic for incoming packets
* Authenticated packets: the backdoor will only process packets that are authenticated. Authenticated procedure can be embedded the password in the payload.
* Extract command: extract the command in the payload
* Shell command: check if the command is a shell command
* Execute command: execute the command and return the result
* File command: check if the command is a file command
* Get or Watch: check if the user wants to watch a specific file/directory or to get a file
* Get a file: if the user wants to get a file, grasp the file and make it ready to send.
* Watch a file of changes: add a watch to look for any changes in the file/directory.
* Keylogging command: check if the command is a command for keylogging.
* Start/stop keylogging: start the keylogger if it is not started yet, stop the keylogger if it is started already.
* Encryption: encryption all data before sending them out
* Send response: send the packets to the attacker. If the command is a file command, send them on a different channel and run port knocker to create a connection.

## PSEUDOCODE

Backdoor

Check if user is root user

Set the process tittle to be something else

While listening for incoming packets

Receive a packet

If packet is authenticated

Extract command from packet payload

If command is shell command

Execute command

Encrypt the result

Send the result to attacker machine

Elif command is file command

If file command is watch

Add watch to a file

Elif file command is get

If file exits

Get the file

Encrypt all data

Start port knocker

Send the file

Endif

Endif

Elif command is keylogging command

If keylogger is active

Stop keylogger

Elif keylogger is not active

Start keylogger

Encrypt keystroke data

Send keystroke data

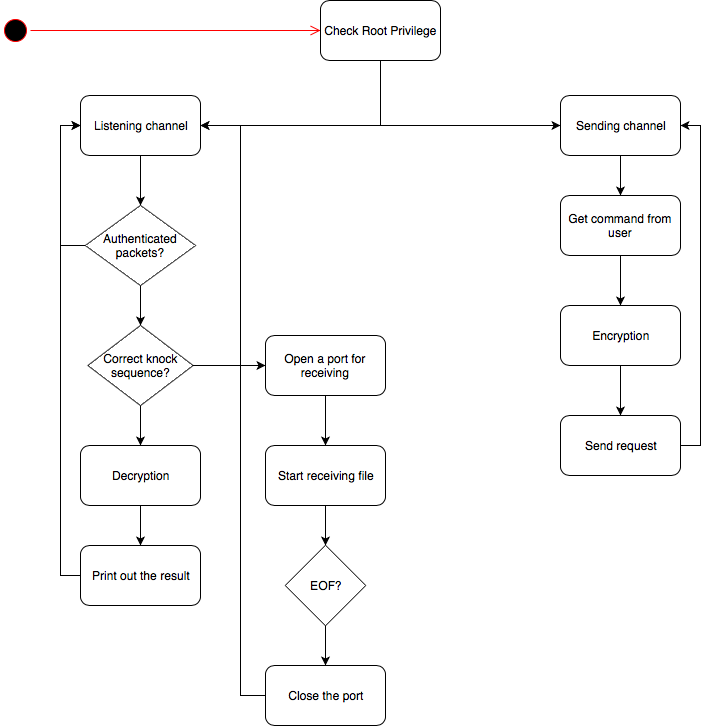
Endif

Endif

Endwhile

# ATTACKER

## DIAGRAM



## DESCRIPTION

* Check root privilege: the application requires root access in order to install necessary modules.
* Listening channel: sniff the traffic for incoming packets.
* Authenticated packets: check if the packets are intended for the attacker application.
* Correct knock sequence: if the knock sequence is correctly done, it means that the backdoor wants to send a file. If it is not, it means that the backdoor wants to send the result of executed commands.
* Decryption: decrypt the packet to get the result.
* Print out the result: print the result to standard output.
* Open a port for receiving: modify the iptables table to allow incoming traffic for a specific port.
* Start receiving file: after opening the port, the application will receive the file in chunk bytes.
* EOF: if the packet is EOF packet, stop receiving and write to a file.
* Close the port: remove the iptables rule to close the port.
* Sending channel: run the loop to get the user command.
* Get command from user: ask the user to get the command.
* Encryption: encrypt the command with password and put it in the payload.
* Send request: send the packet to the backdoor.

## PSEUDOCODE

Attacker

Check if the user is root user

Start listening thread

While listening for incoming packets

If packet is authenticated

If it does correct port knock sequence

Open a port to start receiving

While there is more data to receive

Write data to a file

If it reaches end-of-file

Break

Endwhile

Close the port

Else

Decrypt the payload to get the result

Print out the result

Endif

Endif

Endwhile

Start sending thread

While the application is still alive

Get the command from the user

Encrypt the command

Wrap encrypted command in packet’s payload

Send the packet

Endwhile

# TIMELINES

## Milestone #1 (May 29 – June 05)

* Implementing keylogger on the backdoor application.
* The backdoor will send every keystroke that the victim presses.
* The attacker application will receive the keystroke data and print to the standard output.

## Milestone #2 (June 05 – June 12)

* Implementing file monitor on the backdoor application.
* The attacker can specify to watch a file or a directory for changes such as modify, delete, create.

## Milestone #3 (June 12 – June 19)

* Implementing file transfer on both application.
* The backdoor application will use port knocking technique to make the attacker application open a port
* The attacker application will open a port and start receiving and close the port after finished the task

## Milestone #4 (June 19 – June 26)

* Testing and debugging
* Documentation

# 