Backdoor Assignment

Design Document

Ian Lee and Luke Tao

Assignment 2 for COMP 8505

October 6, 2014

Table of Contents

Backdoor Features

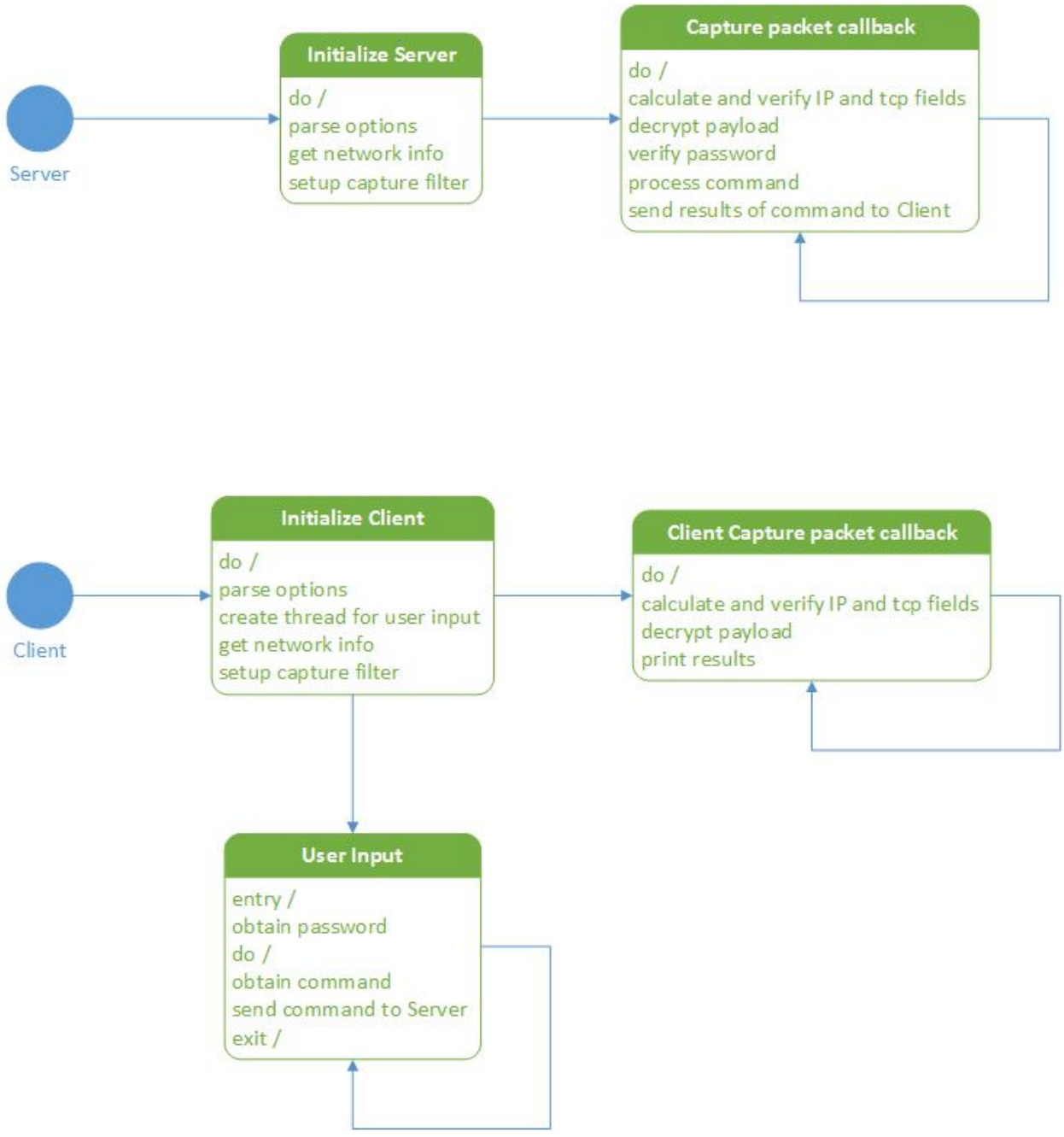
- Execute Commands
    - send results to client
- Process Masking
- Simple Encryption Scheme - XOR
- Using libpcap - Single Packet will be the main authentication protocol
    - Also use Password, hidden on client terminal
- Program in C

Design

Components

- Communication functions
    - Send Packet
    - Recv Packet Callback
    - XOR Encryption
        - Encode XOR = Decode XOR
- Execute commands
    - Parse commands from client
        - Process via popen()
        - Obtain results and send to client
- main
    - Daemon capable server
    - Client

State Transition Diagrams



Pseudo Code (Revised Version)

Main Server function

```
{
        Find a capture device (lookupdev or listalldevs)
        Get netmask and IP

        Print capture info
        Set filter expression
        Use pcap loop to callback

        Clean up stuff
}
```

Mask Process (Server)

```
{
        Set process name passed in and return
}
```

Parse Options (Server)

```
{
        Set struct options

        Set defaults if user doesn't specify them
        While parsing
        {
                Set Daemon to true if user wants it
                Display help if user requests it
        }
}
```

Print Usage (Server)

```
{
        Print the Following Options:
        Running as daemon with -d
        Masking process as (name)
}
```

Main Server

```
{
        Check to see if user is root otherwise exit
        Parse options

        Print Settings if required from -h
```

```
        Daemonize process if required from -d
        Mask process (function)

        Start the server
}
```

Callback function for packet Handling

```
{
        Calculate IP header offset
        Verify the IP header length
        Watch for packets defined by filter
        Calculate tcp header offset
        Verify TCP header length

        Calculate the payload offset and size
        Decrypt the payload

        Grab password and command field
        If incorrect password, return
        If we're the server and the password is correct
                execute the send command by sending the results back to the client
        Else is Client so print/ save command results

}
Send the command
{
        Execute the command
        Read the results and send them back to the client.
}
Start Client
{
        start capturing packets
        start thread to process user input
}
User Process
{
        Get password from user prompt
        loop until user types quit
                Get command from user prompt
                encrypt password and command as payload
                send packet to server
}
```