Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Московский государственный технический университет
имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)

Факультет: Информатика и системы управления
Кафедра: Теоретическая информатика и компьютерные технологии

Лабораторная работа №4
«Разработка SSH-сервера и SSH-клиента»
по курсу: «Компьютерные сети»

Выполнил:
Студент группы ИУ9-32Б
Гнатенко Т. А.

Проверил:
Посевин Д. П.

Москва, 2022

Цели

Изучить принципы работы протокола ssh

Часть 1. Разработка SSH-сервера

Задачи

Реализовать ssh сервер на языке GO с применением указанных пакетов и запустить его на localhost. Проверка работы должна проводиться путем использования программы ssh в ОС Linux/Unix или PuTTY в ОС Windows. Должны работать следующие функции: ◦ авторизация клиента на ssh сервере; ◦ создание директории на удаленном сервере; ◦ удаление директории на удаленном сервере; ◦ вывод содержимого директории; ◦ перемещение файлов из одной директории в другую; ◦ удаление файла по имени; ◦ вызов внешних приложений, например ping.

Решение

Исходный код

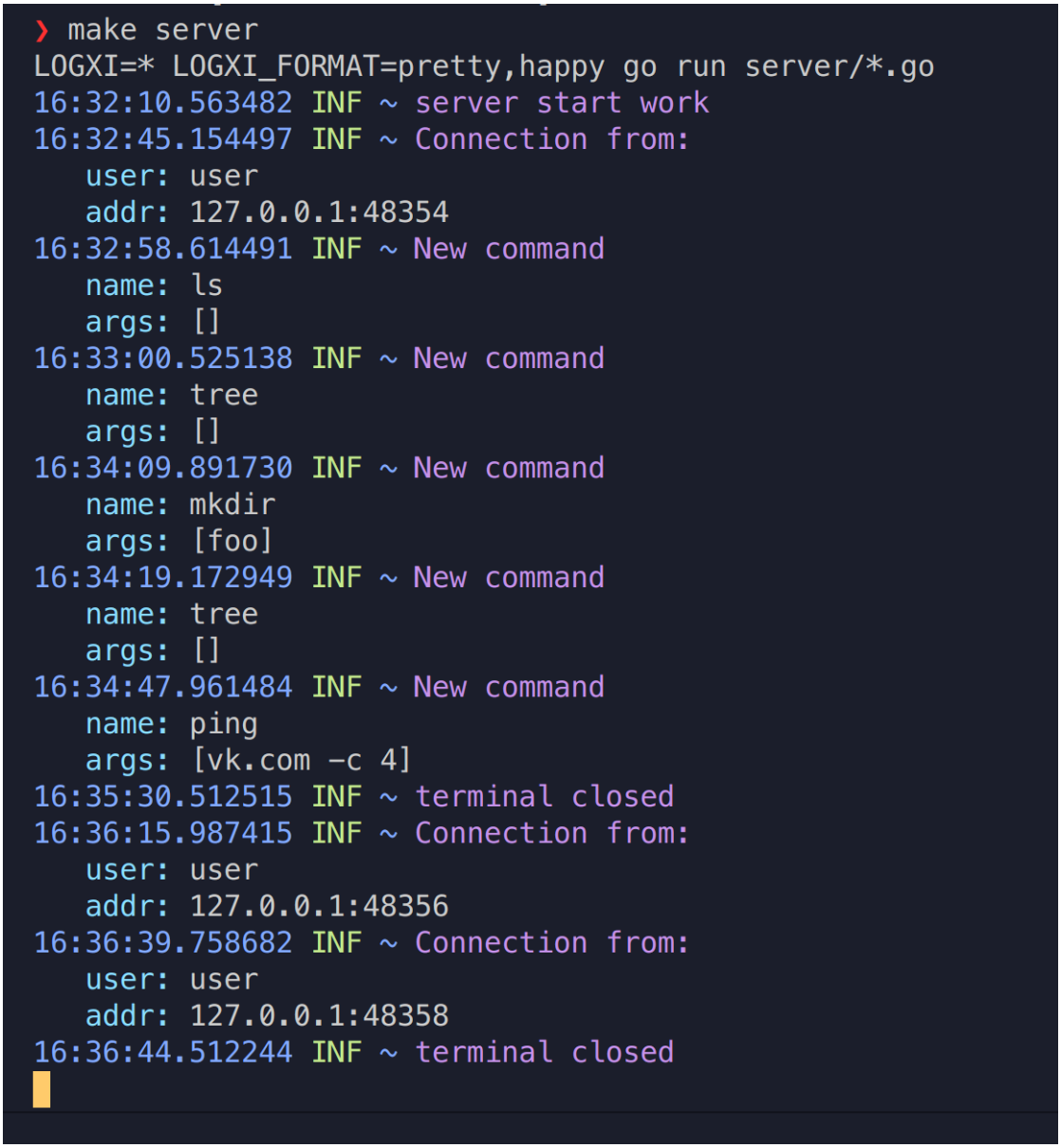
server.go

```
package main

import (
    "bytes"
    "log"
    "os/exec"
    "strings"

    "github.com/gliderlabs/ssh"
    log2 "github.com/mgutz/logxi/v1"
    "golang.org/x/term"
)

func handle(data []string) string {
```

```
    name := data[0]
    args := data[1:]

    log2.Info("New command", "name", name, "args", args)

    var buf bytes.Buffer
    cmd := exec.Command(name, args...)
    cmd.Stdout = &buf

    err := cmd.Run()
    if err != nil {
        log.Fatal(err)
    }

    return buf.String()
}

func main() {

    ssh.Handle(func(s ssh.Session) {
        log2.Info("Connection from: ", "user", s.User(),
↪   "addr", s.RemoteAddr())

        terminal := term.NewTerminal(s, "> ")
        for {
            line, err := terminal.ReadLine()
            if err != nil {
                break
            }
            answer := handle(strings.Split(line, " "))

            if answer != "" {
                terminal.Write(append([]byte(answer), '\n'))
            }
        }

        log2.Info("terminal closed")
    })

    log2.Info("server start work")
```

```
    log.Fatal(ssh.ListenAndServe(":2222", nil))
}
```

Вывод



```
❯ ssh user@localhost -p 2222
The authenticity of host '[localhost]:2222 ([127.0.0.1]:2222)' can't be established.
RSA key fingerprint is SHA256:QbDvOoQP7nk2ohKycfQZYT+9yvz8RcvAt4X6FtzbSgI.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[localhost]:2222' (RSA) to the list of known hosts.
> ls
client
go.mod
go.sum
Makefile
server

> tree
.
├── client
│   └── client.go
├── go.mod
├── go.sum
├── Makefile
└── server
    └── server.go

2 directories, 5 files

> mkdir foo
> tree
.
├── client
│   └── client.go
├── foo
├── go.mod
├── go.sum
├── Makefile
└── server
    └── server.go

3 directories, 5 files

> ping vk.com -c 4
PING vk.com (87.240.129.133) 56(84) bytes of data.
64 bytes from srv133-129-240-87.vk.com (87.240.129.133): icmp_seq=1 ttl=57 time=292 ms
64 bytes from srv133-129-240-87.vk.com (87.240.129.133): icmp_seq=2 ttl=57 time=111 ms
64 bytes from srv133-129-240-87.vk.com (87.240.129.133): icmp_seq=3 ttl=57 time=181 ms
64 bytes from srv133-129-240-87.vk.com (87.240.129.133): icmp_seq=4 ttl=57 time=294 ms

--- vk.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3378ms
rtt min/avg/max/mdev = 110.536/219.417/293.544/77.638 ms
```

Часть 2. Разработка SSH-клиента

Задачи

Реализовать ssh-клиент и запустить его на localhost. Протестировать соединение Go SSH-клиента к серверу реализованному в предыдущей задаче, а также к произвольному ssh серверу. Требования: SSH-клиент должен поддерживать следующие функции: ◦ авторизация клиента на SSH-сервере; ◦ создание директории на удаленном SSH-сервере; ◦ удаление директории на удаленном SSH-сервере; ◦ вывод содержимого директории; ◦ перемещение файлов из одной директории в другую; ◦ удаление файла по имени; ◦ вызов внешних приложений, например ping.

Решение

Исходный код

```
package main

import (
    "bufio"
    "fmt"
    "log"
    "os"

    "golang.org/x/crypto/ssh"
)

func main() {

    username := "test"
    password := "SDHBCXdsedfs222"
    hostname := "151.248.113.144"
    port := "443"
    // username := "user"
    // password :=
     ↪  "S4EmRoIhDy/w1bIzlOU51lphBN4=|+5jIYH84ai5UcFexe6VSFCaMQV0=
     ↪  ssh-rsa
     ↪  AAAAB3NzaC1yc2EAAAADAQABAAABAQC10H6D+szTfU4FZvGlzUNYRYfrSNGPfOSHP9
```

```
    // hostname := "localhost"
    // port := "2222"

    // SSH client config
    config := &ssh.ClientConfig{
        User: username,
        Auth: []ssh.AuthMethod{
            ssh.Password(password),
        },
        // Non-production only
        HostKeyCallback: ssh.InsecureIgnoreHostKey(),
    }

    // Connect to host
    client, err := ssh.Dial("tcp", hostname+":"+port,
    config)
    if err != nil {
        log.Fatal(err)
    }
    defer client.Close()

    // Create sesssion
    sesssion, err := client.NewSession()
    if err != nil {
        log.Fatal("Failed to create session: ", err)
    }
    defer sesssion.Close()

    // StdinPipe for commands
    stdin, err := sesssion.StdinPipe()
    if err != nil {
        log.Fatal(err)
    }

    // Enable system stdout
    // Comment these if you uncomment to store in variable
    sesssion.Stdout = os.Stdout
    sesssion.Stderr = os.Stderr

    // Start remote shell
```

```
    err = sesssion.Shell()
    if err != nil {
        log.Fatal(err)
    }

    // send the commands
    for {
        reader := bufio.NewReader(os.Stdin)
        line, _, err := reader.ReadLine()
        if err != nil {
            break
        }
        cmd := string(line)
        if cmd != "exit" {
            _, err = fmt.Fprintf(stdin, "%s\n", cmd)
            if err != nil {
                log.Fatal(err)
            }
        } else {
            break
        }
    }

    // Wait for sesssion to finish
    err = sesssion.Wait()
    if err != nil {
        log.Fatal(err)
    }

}
```

Вывод

```
❯ make client
LOGXI=* LOGXI_FORMAT=pretty,happy go run client/*.go
Linux izobretarium.posevin.com 2.6.32-042stab145.3 #1 SMP Thu Jun 11 14:05:04 MSK 2020 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
ls
test
mkdir foo
ls
foo
test
rmdir foo
ls
test
ping vk.com -c 4
PING vk.com (87.240.137.164) 56(84) bytes of data.
64 bytes from srv164-137-240-87.vk.com (87.240.137.164): icmp_seq=1 ttl=57 time=8.09 ms
64 bytes from srv164-137-240-87.vk.com (87.240.137.164): icmp_seq=2 ttl=57 time=8.18 ms
64 bytes from srv164-137-240-87.vk.com (87.240.137.164): icmp_seq=3 ttl=57 time=8.03 ms
64 bytes from srv164-137-240-87.vk.com (87.240.137.164): icmp_seq=4 ttl=57 time=7.93 ms

--- vk.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 7.935/8.061/8.188/0.128 ms
```