# Linux for Pentester: CAT Privilege Escalation

June 15, 2019 By Raj Chandel

Today we are going to talk about CAT command and learn how helpful the cat command is for Linux penetration testing and how we'll progress cat to scale the greater privilege shell.

***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 CAT

In Linux, Cat stands for "catenate," which is one of Unix-like operating system most frequently used commands. It reads file information and displays its content as an output. It enables us to build, view and link files. So, we can not only see the content using CAT command; apart from this we can, copy the content of the file to some other file and view the files with numbers and so on. Not only this we will do such things which is not only new but is what we might have not thought of. We will perform Privilege Escalation using CAT command. That's sounds interesting. Isn't it? So, let's start-

**Major Functions of CAT command**

At first, we will run **cat -h** command which means help and which will tell you about all the options which are available in CAT command as we can see in the picture below.

```
cat --help
```

```
root@ubuntu:~# cat --help
Usage: cat [OPTION]... [FILE]...
Concatenate FILE(s) to standard output.

With no FILE, or when FILE is -, read standard input.

  -A, --show-all           equivalent to -vET
  -b, --number-nonblank    number nonempty output lines, overrides -n
  -e                       equivalent to -vE
  -E, --show-ends          display $ at end of each line
  -n, --number             number all output lines
  -s, --squeeze-blank      suppress repeated empty output lines
  -t                       equivalent to -vT
  -T, --show-tabs          display TAB characters as ^I
  -u                       (ignored)
  -v, --show-nonprinting   use ^ and M- notation, except for LFD and TAB
      --help     display this help and exit
      --version  output version information and exit

Examples:
  cat f - g  Output f's contents, then standard input, then g's contents.
  cat        Copy standard input to standard output.

GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Full documentation at: <http://www.gnu.org/software/coreutils/cat>
or available locally via: info '(coreutils) cat invocation'
root@ubuntu:~#
```

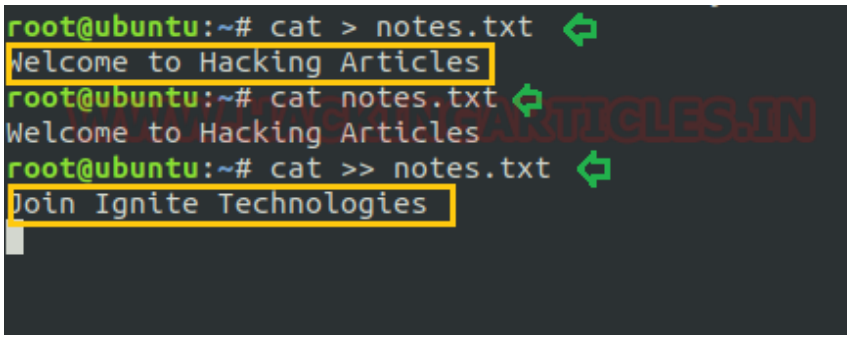
## Write and Read a file:

Our next step is to create a file using the cat command. And for this, we will use greater than sign (>) after cat command to generate a new file. So, we have created a new file named **notes.txt** by using **(>)** this sign after cat command and write the content which you want to keep in the file as in our case I have written **"Welcome to Hacking articles"** in the file **notes.txt**

```
cat > notes.txt
```

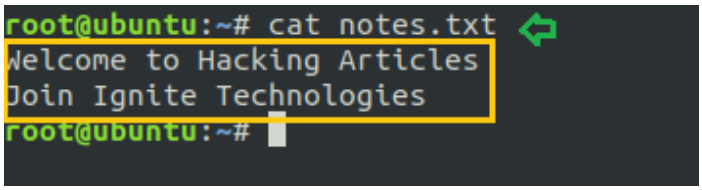
Not only this we can also edit the content of the existing file without opening the file by using greater than sign twice (>>) as you can see in the screenshot that we have added **"Join Ignite Technologies"** in notes.txt

```
cat >> notes.txt
cat notes.txt
```




Now we can confirm this by reading the file once again.

```
cat notes.txt
```

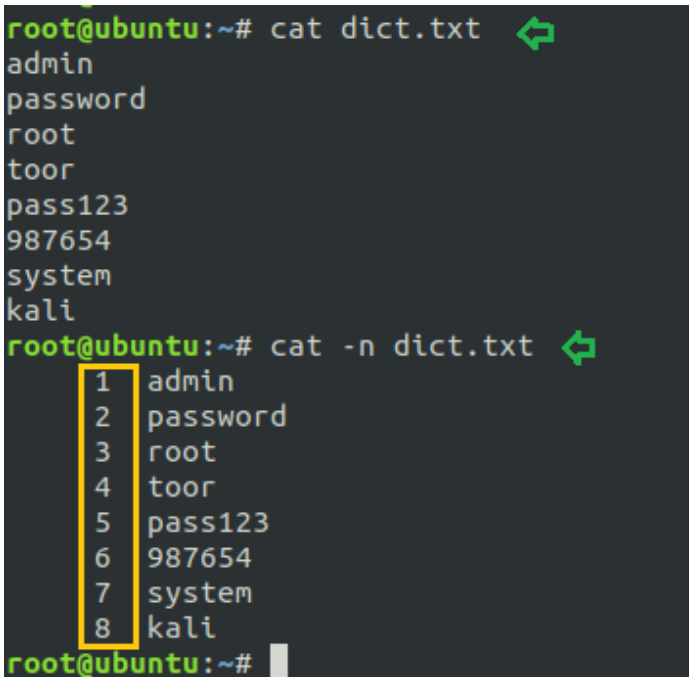



## Number all output lines:

Now let's say if we want to view file contents preceding line numbers or in other words you want to view the output serialized. So first we will create a new text file named **dict.txt** in which we have written some content which is going to be easily readable number wise with -n command.

```
cat dict.txt
cat -n dict.txt
```

As a result, this add a serial number column for every line as shown below:

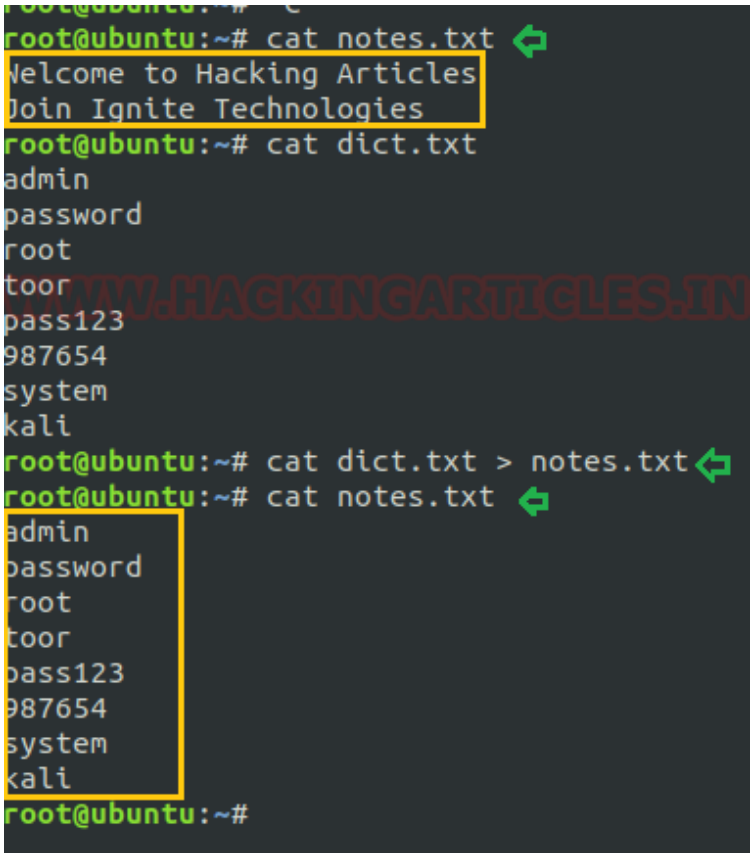



## Overwriting a file

Now we want to copy the content of file dict.txt into notes.txt or in other words we want to overwrite the file notes.txt. So in order to do, this first we write the file name from which the content is to be copied and then we will write the file name whose content we want to replace followed by greater than sign(>).

```
Syntax: cat [file1] > [file2]
cat dict.txt > notes.txt
```

As you can observe in the picture below that we have replaced the content of **notes.txt** with **dict.txt**

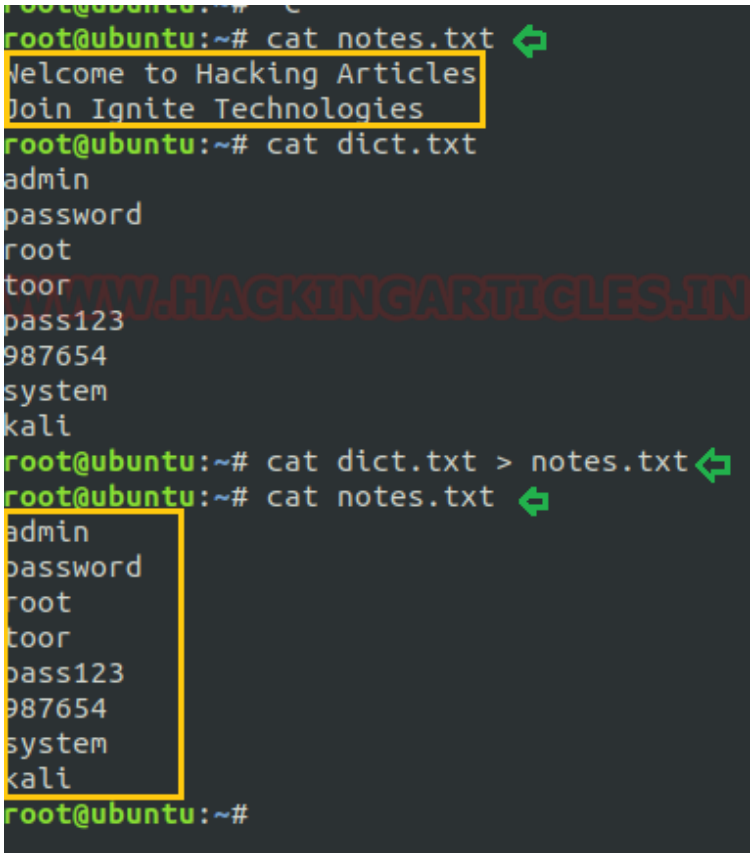



## Concatenating files:

Now we want to merge two files together or in other words, we want to combine two files. So, what will we do? Its again very simple; we will use greater than sigh here but now twice (>>) and the content will be replaced successfully. So here we have another new file which is **pass.txt** and then we will proceed towards merging two files for which we will use **(>>)** sign again as we have done in the image below. Now again we will use **-n** to put this content number wise which we have done above.

```
cat > pass.txt
cat dict.txt >> pass.txt
cat -n pass.txt
```

As result, you can observe that we have concatenate dict.txt in the pass.txt file.

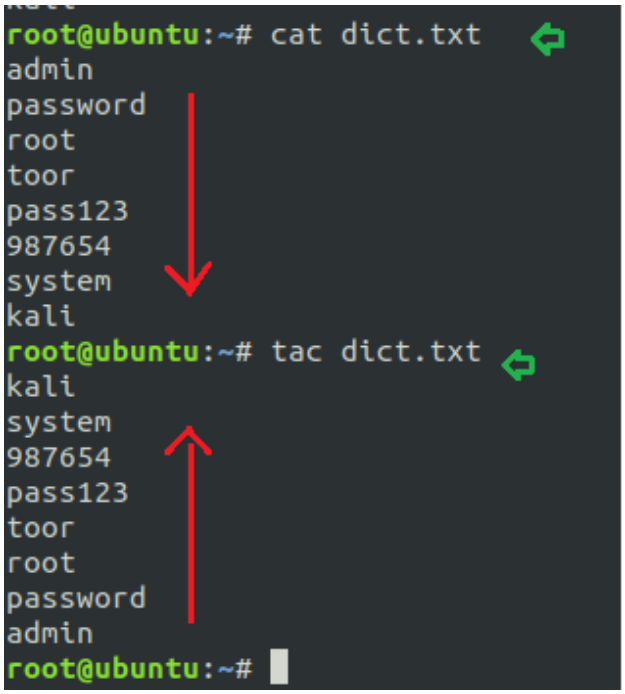



## Reverse order

As the name suggests and we can reverse all the content using **tac command which** is just a reverse of **cat** command and it works for this purpose only.

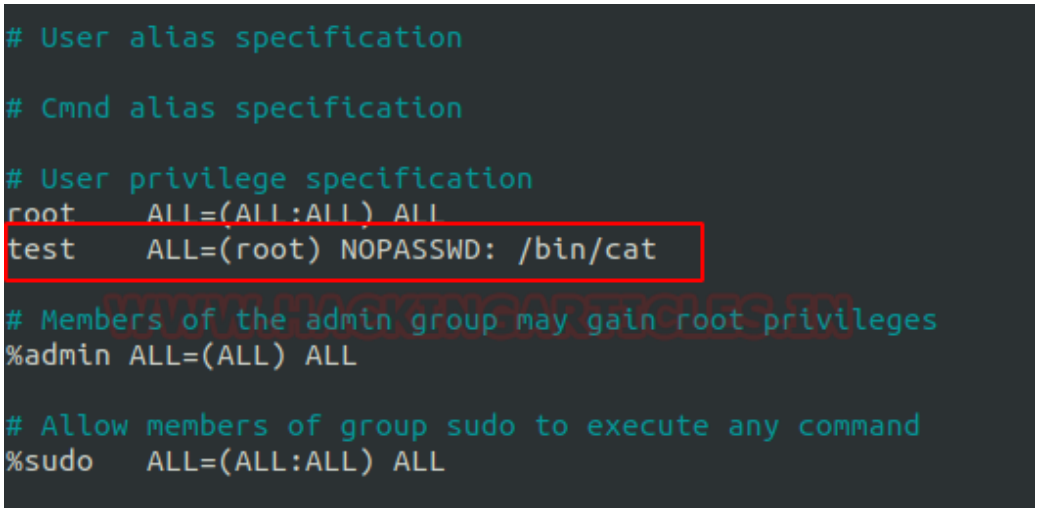
```
cat dict.txt
```

With the help of tac command, we try to reverse the file by making a vertical flip as shown below.




## Sudo rights Lab setups for Privilege Escalation

Now here our next step is to set up the lab of Sudo rights or in other words to provide Sudo privileges to a user for cat executable. Here we are going to add a user by the name of the **test** in the suoders files and here we have given permission to user test to run **cat** command as **root** user.




## Exploiting Sudo Rights

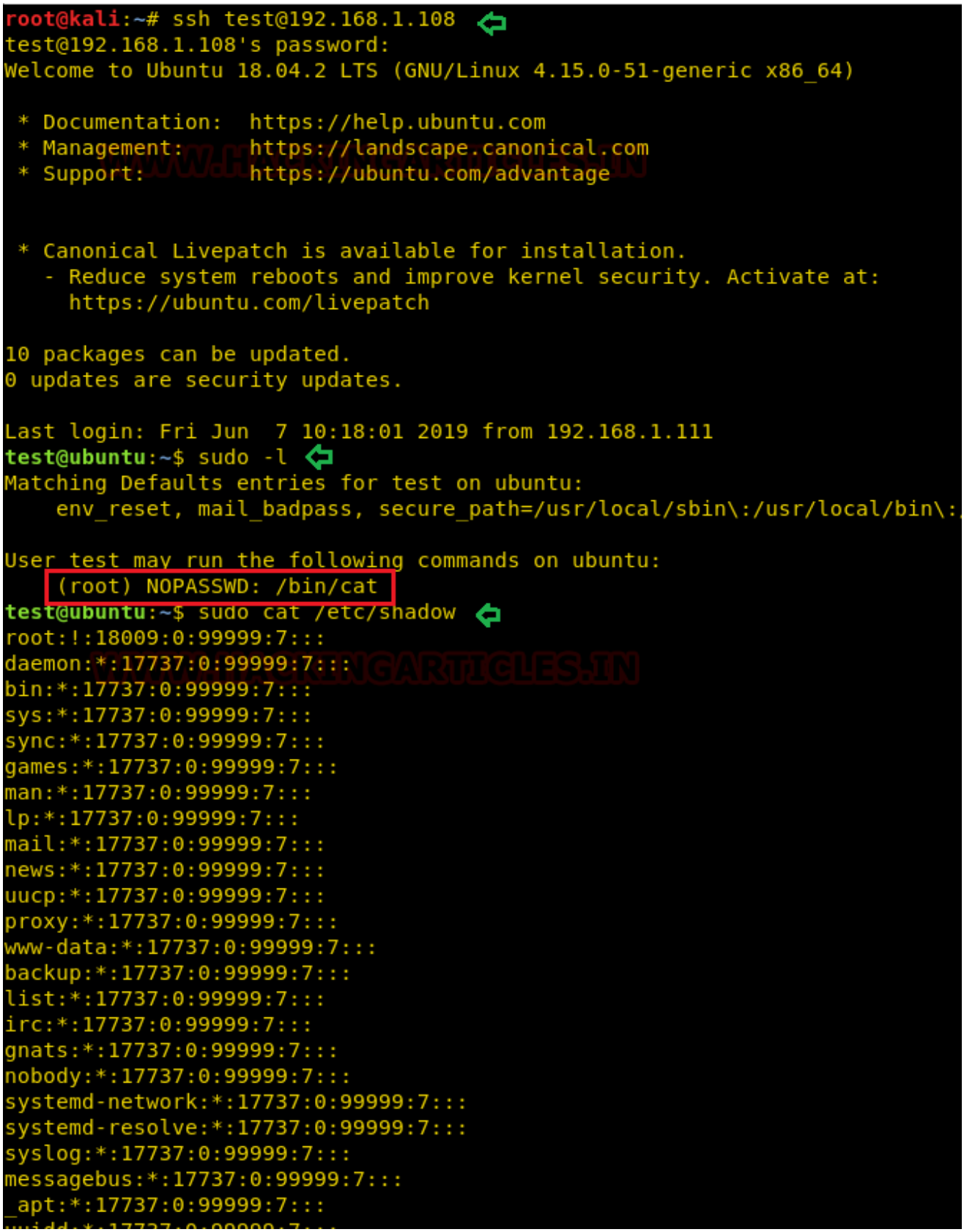
Now we will connect through **ssh** in kali and after that, we will run **sudo -l** which is sudo list and through which we can see that user test has the permission to run cat as root user.

```
ssh test@192.168.1.108
```

Now our next step is to exploit sudo rights through cat command. So, we will run cat /etc/shadow command to see all the users and their respective passwords hashes.

```
sudo -l
sudo cat /etc/shadow
```

Wonderful! We have got all the user's list and their passwords' hash value.



## Cracking the Hash Password

Now our next step is to crack the hash value so that we are going to use "**John the Ripper"** tool to crack this hash value in order to get the password in decrypted form. So first we have taken one user whose password we want to check. So, run the following command in the terminal-

```
john hash --show
```


Great! We have cracked the password successfully. Now we will switch user **raj** to check if we can log in through that password and we can see that we have successfully logged in as raj user.

Now we will run **sudo -l** command to check if user **raj**, and found he has all the root permissions.

```
sudo -l
sudo su
```

Now, we will again try to switch to user **root** and we are logged in as **root** and then we run **id** command we get to know that we got a root shell.

So, we have performed **privilege escalation** through **cat** command successfully.