Hacking Articles

Raj Chandel's Blog

Linux For Pentester: tmux Privilege Escalation

posted in PRIVILEGE ESCALATION on AUGUST 9, 2019 by RAJ CHANDEL SHARE

In this article, we are going to describe "tmux" which is also known as a terminal multiplexer. It allows multiple terminal sessions to be retrieved concurrently in a single window. It is useful for running more than one command-line program at the same time.

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 criticize any kind of misconfiguration that a

network or system administrator does for providing higher permissions on any programs/binaries/files & etc.”

Table of Content

What is tmux?: tmux is also known as a terminal multiplexer which creates a host **server** on your Linode and connects to it with a client window. If the client is disconnected, the server keeps running and as you reconnect to your Linode after rebooting your computer, you can reattach to the tmux session and the files you were working with will still be open.

In other words, we can also say that this is a tool by the help of which we can open multiple windows and split views (called “*panes*” in tmux lingo) within one terminal window.

How to use tmux: Alike other tmux also supports many commands to perform its function. Now we will describe each of its major operations one by one.

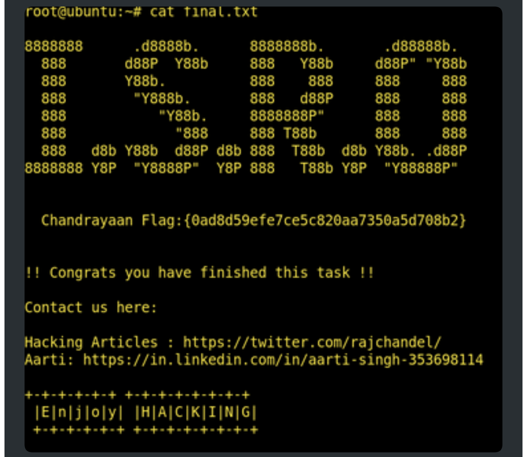
It can be attained by entering a key combination called the **prefix** and then typing a **letter**. There are many letters that are assigned to tmux for performing its task.

tmux framework: The entire operations that a tmux does can be easily understood by its hierarchical structure as shown below.

Hacking Articles Retweeted

Sorin Radu
@soriadu

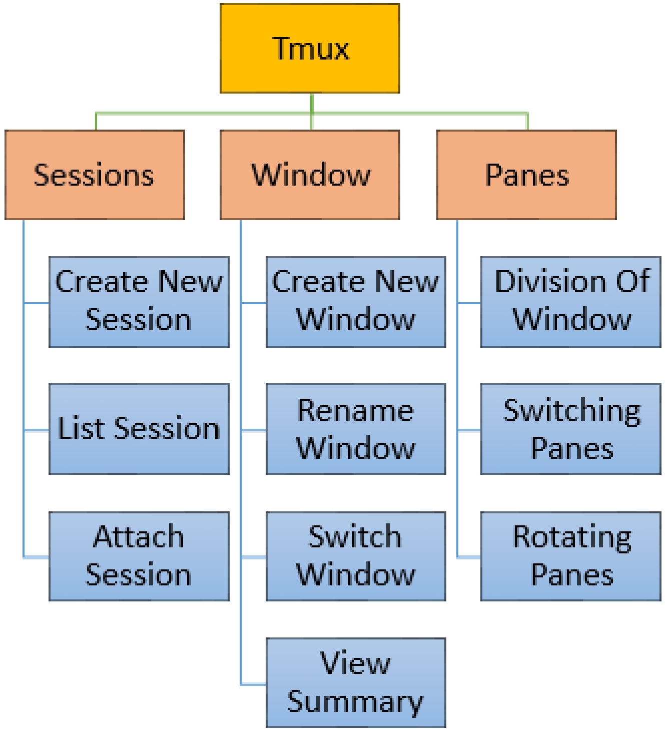
HA:ISRO @VulnHub @rajchandel
Nice experience !



```
root@ubuntu:~# cat final.txt

8888888      .d8888b.      8888888b.       .d88888b.
  888       d88P  Y88b     888   Y88b     d88P" "Y88b
  888       Y88b.          888    888     888     888
  888        "Y888b.       888   d88P     888     888
  888           "Y88b.     8888888P"      888     888
  888             "888     888 T88b       888     888
  888   d8b Y88b  d88P d8b 888  T88b  d8b Y88b. .d88P
8888888 Y8P  "Y8888P"  Y8P 888   T88b Y8P  "Y88888P"


  Chandrayaan Flag:{0ad8d59efe7ce5c820aa7350a5d708b2}


!! Congrats you have finished this task !!

Contact us here:

Hacking Articles : https://twitter.com/rajchandel/
Aarti: https://in.linkedin.com/in/aarti-singh-353698114

+-+-+-+-+-+ +-+-+-+-+-+-+-+
 |E|n|j|o|y| |H|A|C|K|I|N|G|
 +-+-+-+-+-+ +-+-+-+-+-+-+-+
```

10h

Embed View on Twitter



tmux commands: There are list of command that can help while working with tmux. Here in this article, we are running the major operation that can be performed by the help of tmux.

Very first we will start from its help command. For this we need to write "–help" on our kali terminal as shown below.

```
tmux --help
```

The tmux operations are categorized into 3 selection which I have described above in its framework. So now we will start from first step i.e "sessions"

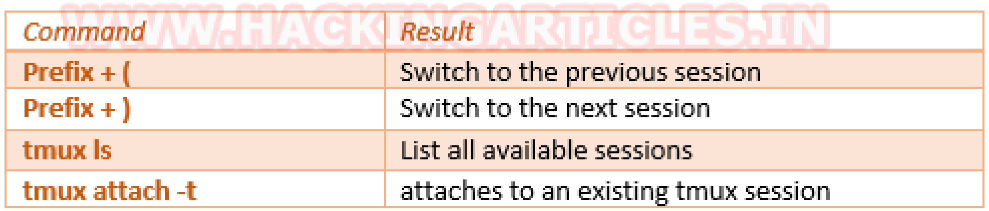
Operate tmux Sessions: Sometimes even multiple windows and panes aren't enough and you need to separate the layouts logically by grouping them into separate sessions.

Categories

- BackTrack 5 Tutorials
- Cryptography & Stegnography
- CTF Challenges
- Cyber Forensics
- Database Hacking
- Footprinting
- Hacking Tools
- Kali Linux
- Nmap
- Others
- Penetration Testing

Sessions are useful for completely separating work environments.

There are many operations for the session using tmux which is shown in below image but I'm describing few of them.

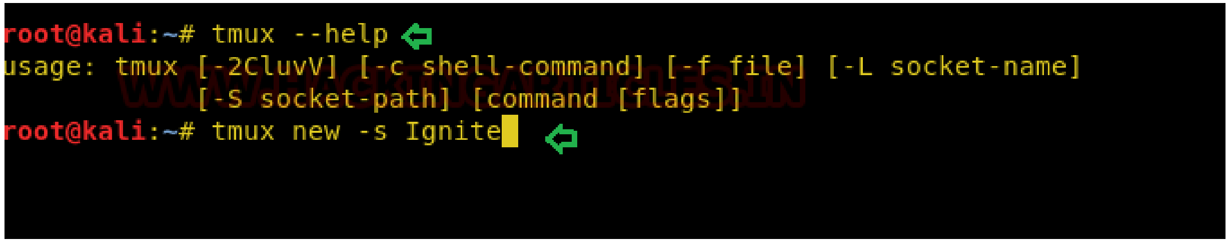


Command	Result
Prefix + (	Switch to the previous session
Prefix +)	Switch to the next session
tmux ls	List all available sessions
tmux attach -t	attaches to an existing tmux session

- ***Create a new session***: To create a new session we will frame command as shown in the below image.

```
tmux new -s Ignite
```

In the above command "-s" is used as an argument for a new session and "Ignite" is the name of the new session that I want to create.



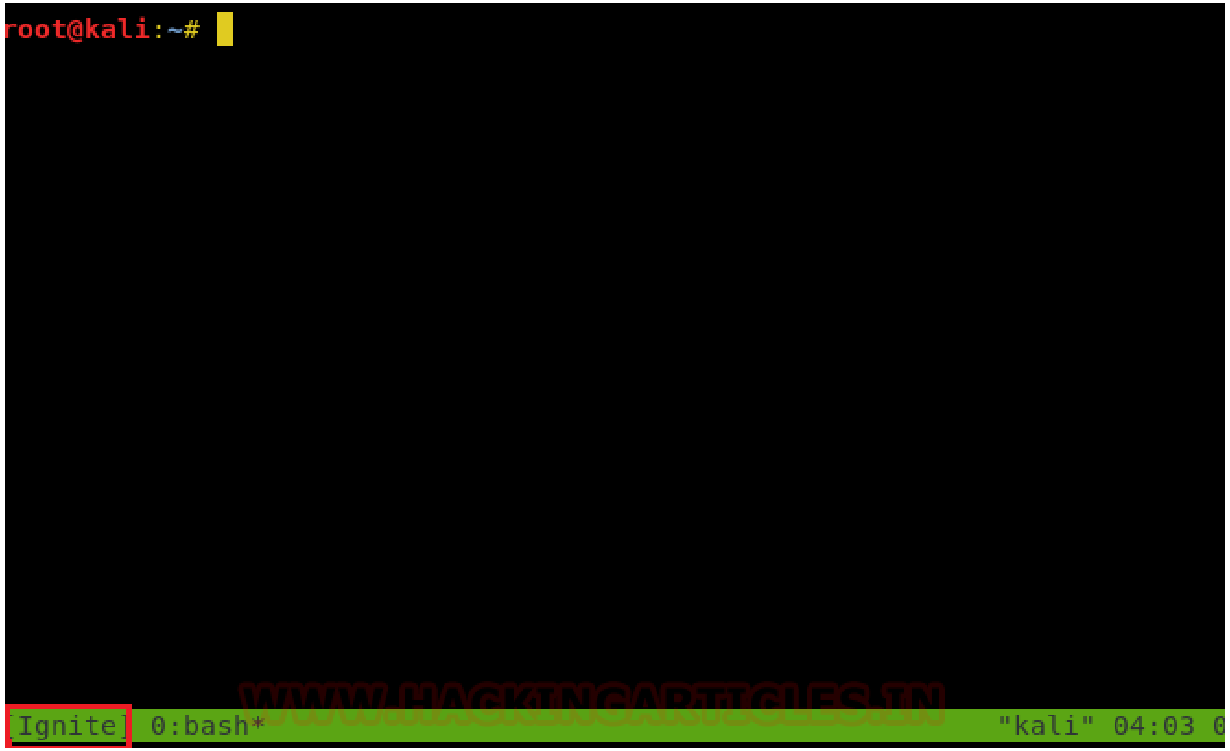
```
root@kali:~# tmux --help
usage: tmux [-2CluvV] [-c shell-command] [-f file] [-L socket-name]
            [-S socket-path] [command [flags]]
root@kali:~# tmux new -s Ignite
```

On framing above command tmux will create a new session by the name of Ignite which will highlight at the bottom of terminal. Similarly, one can create multiple session by a different name as per need.

- Privilege Escalation
- Red Teaming
- Social Engineering Toolkit
- Trojans & Backdoors
- Website Hacking
- Window Password Hacking
- Wireless Hacking

Articles

Select Month



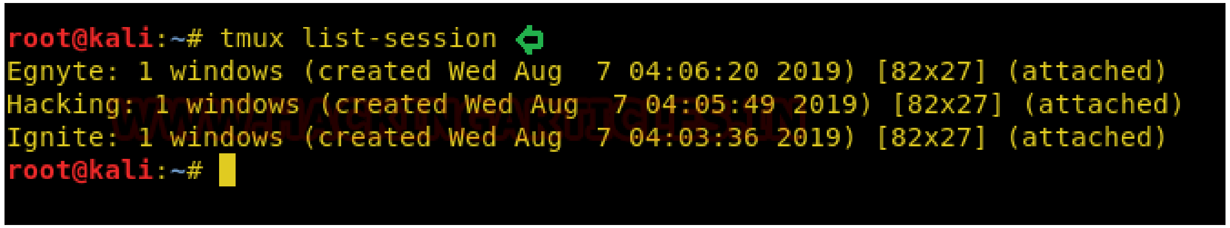
```
root@kali:~# 
```

```
[Ignite] 0:bash*                                          "kali" 04:03 0
```

- ***To list all created session***: once we have done with creating all session as per desire then we can check it by command as:

```
tmux list-session
```

This will list all session as output that have been created. In below image tmux has listed all session which I have created by following the same procedure as above.



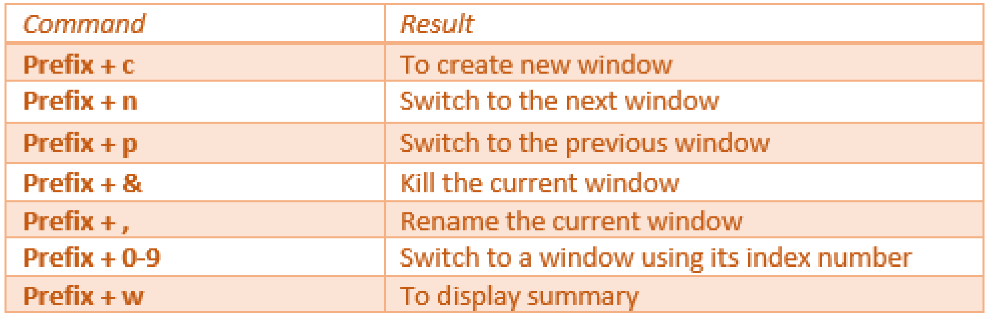
```
root@kali:~# tmux list-session
Egnyte: 1 windows (created Wed Aug  7 04:06:20 2019) [82x27] (attached)
Hacking: 1 windows (created Wed Aug  7 04:05:49 2019) [82x27] (attached)
Ignite: 1 windows (created Wed Aug  7 04:03:36 2019) [82x27] (attached)
root@kali:~# 
```

Operate tmux Window: When a tmux session starts, a single-window is fashioned by default but tmux also supports a utility to attach multiple windows to the same session and we can switch between them as needed. This can be supportive when you want to run numerous jobs in parallel.

Apart from creating multiple windows it also possesses many operations like rename any window, switch between window and many others.

At the initial phase, it shows "**0: bash***" by default in which **0** represents the index value of window **bash** is the window name which can be renamed as per need ***** denotes the working location and when we create new window tmux highlights all window at the bottom of the terminal.

Note: We know that working of tmux is done with joining prefix with any letter as per requirement. Find the below table to understand it clearly.



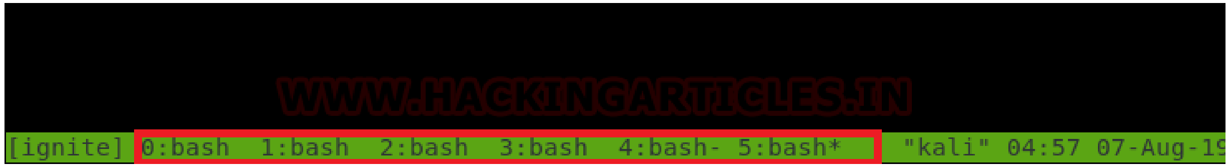
Command	*Result*
Prefix + c	To create new window
Prefix + n	Switch to the next window
Prefix + p	Switch to the previous window
Prefix + &	Kill the current window
Prefix + ,	Rename the current window
Prefix + 0-9	Switch to a window using its index number
Prefix + w	To display summary

In this article, I have created 5 windows as shown in the below image. We know that working of tmux is done with joining prefix with any letter as per requirement.

- ***Create new window***: For creating a new window we will use "-c" with the prefix (ctrl-b).

```
Prefix (ctrl-b) +c
```

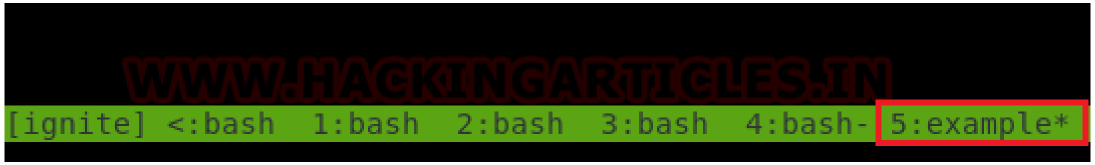
This will create a new window. You can use the same procedure for creating multiple windows as below image.



```
[ignite] 0:bash  1:bash  2:bash  3:bash  4:bash- 5:bash*   "kali" 04:57 07-Aug-19
```

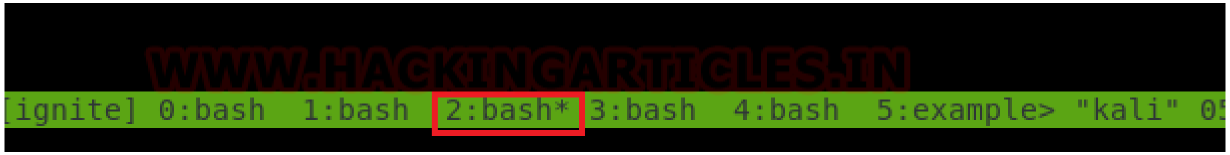
- ***Rename window:*** by default, tmux mention the window name as “bash” but we can also change it as per our wish. Here I’m renaming my last window as shown below.

```
Prefix (ctrl-b) + ,
```



```
[ignite] <:bash  1:bash  2:bash  3:bash  4:bash- 5:example*
```

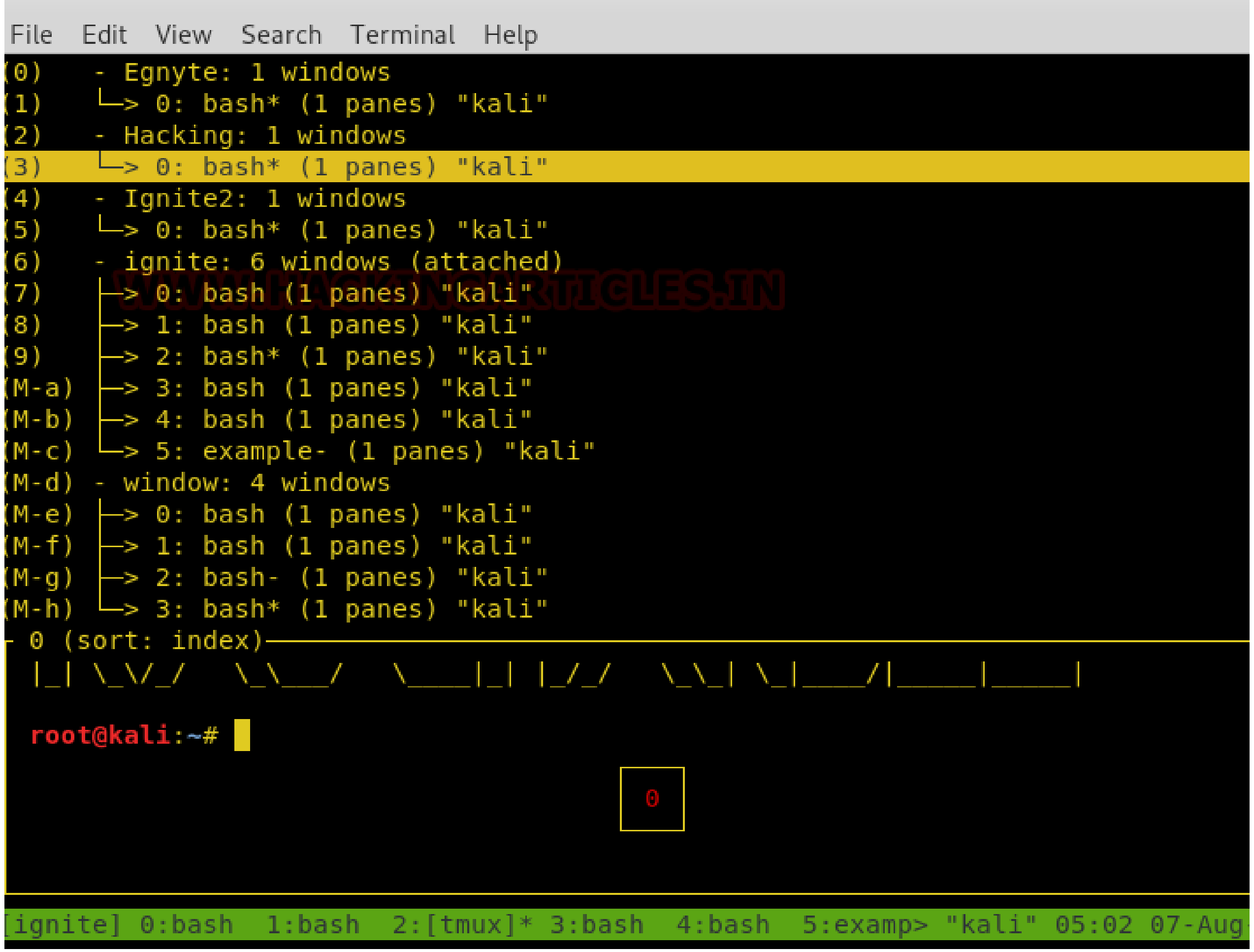
- ***To switch window***: we can also switch within multiple windows that help to provide the platform of working parallel. It can be done in many ways.



```
[ignite] 0:bash  1:bash  2:bash* 3:bash  4:bash  5:example> "kali" 05
```

- ***To display summary***: To see the entire summary for whatever we have done till now we will use tmux option as:

```
Prefix (ctrl-b) + w
```

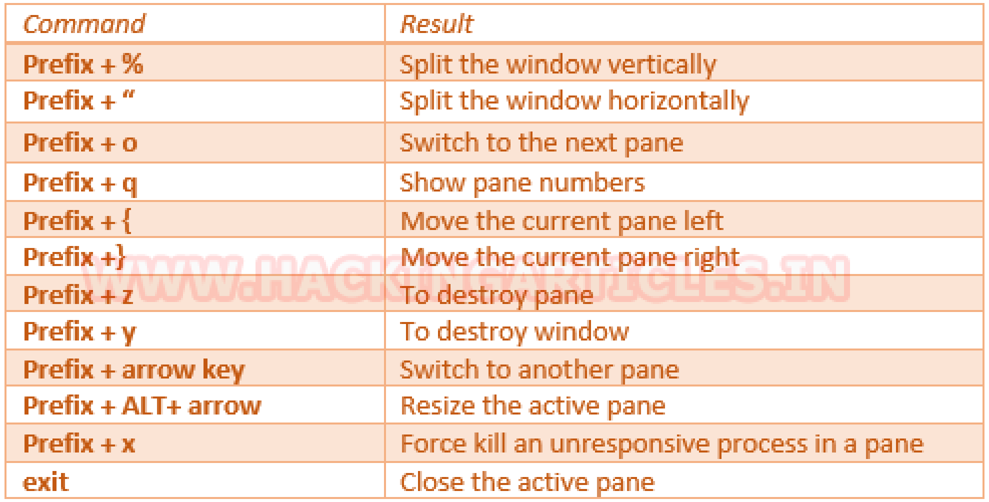


```
(0)   - Egnyte: 1 windows
(1)   └─> 0: bash* (1 panes) "kali"
(2)   - Hacking: 1 windows
(3)   └─> 0: bash* (1 panes) "kali"
(4)   - Ignite2: 1 windows
(5)   └─> 0: bash* (1 panes) "kali"
(6)   - ignite: 6 windows (attached)
(7)   ├─> 0: bash (1 panes) "kali"
(8)   ├─> 1: bash (1 panes) "kali"
(9)   ├─> 2: bash* (1 panes) "kali"
(M-a) ├─> 3: bash (1 panes) "kali"
(M-b) ├─> 4: bash (1 panes) "kali"
(M-c) └─> 5: example- (1 panes) "kali"
(M-d) - window: 4 windows
(M-e) ├─> 0: bash (1 panes) "kali"
(M-f) ├─> 1: bash (1 panes) "kali"
(M-g) ├─> 2: bash- (1 panes) "kali"
(M-h) └─> 3: bash* (1 panes) "kali"
- 0 (sort: index)
 |_| \_\/_/   \_\___/   \____|_| |_/_/   \_\_| \_|____/|_____|_____|

 root@kali:~#

                                    0

[ignite] 0:bash  1:bash  2:[tmux]* 3:bash  4:bash  5:examp> "kali" 05:02 07-Aug
```

Operate tmux Panes: By the help of tmux, we can divide each window into multiple panes. This is useful when you want outputs from multiple processes visible within a single window.

In this we have many options such as divide window into vertical, horizontal, rotating panes, switching to different panes. Now we will check each of this one by one.

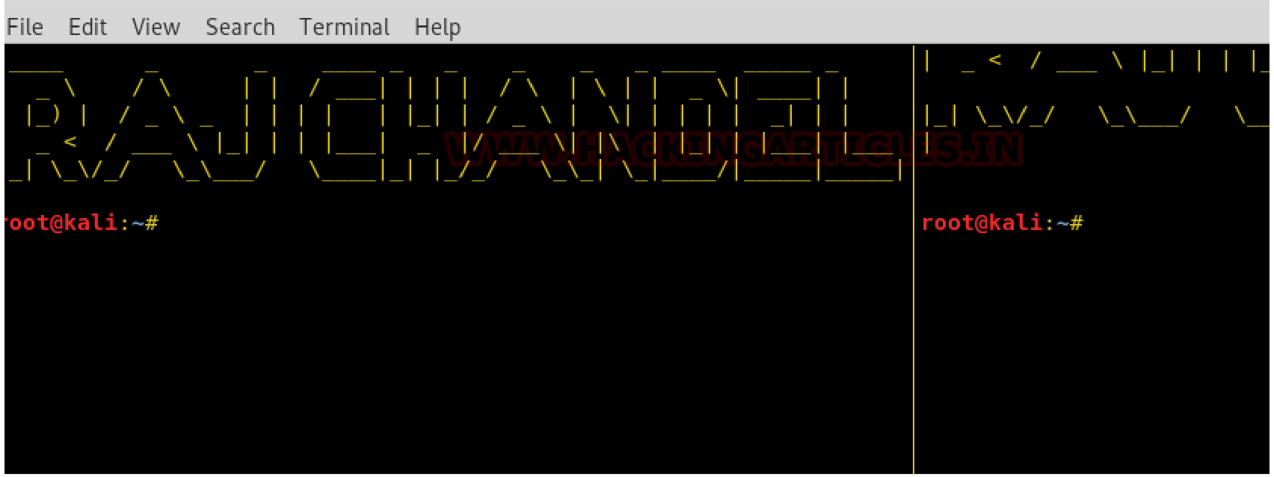
Note: use below table for your reference



Command	Result
Prefix + %	Split the window vertically
Prefix + “	Split the window horizontally
Prefix + o	Switch to the next pane
Prefix + q	Show pane numbers
Prefix + {	Move the current pane left
Prefix +}	Move the current pane right
Prefix + z	To destroy pane
Prefix + y	To destroy window
Prefix + arrow key	Switch to another pane
Prefix + ALT+ arrow	Resize the active pane
Prefix + x	Force kill an unresponsive process in a pane
exit	Close the active pane

Here I have divided my window into 2 panes vertically by the command as:

```
Prefix (ctrl-b) + %
```

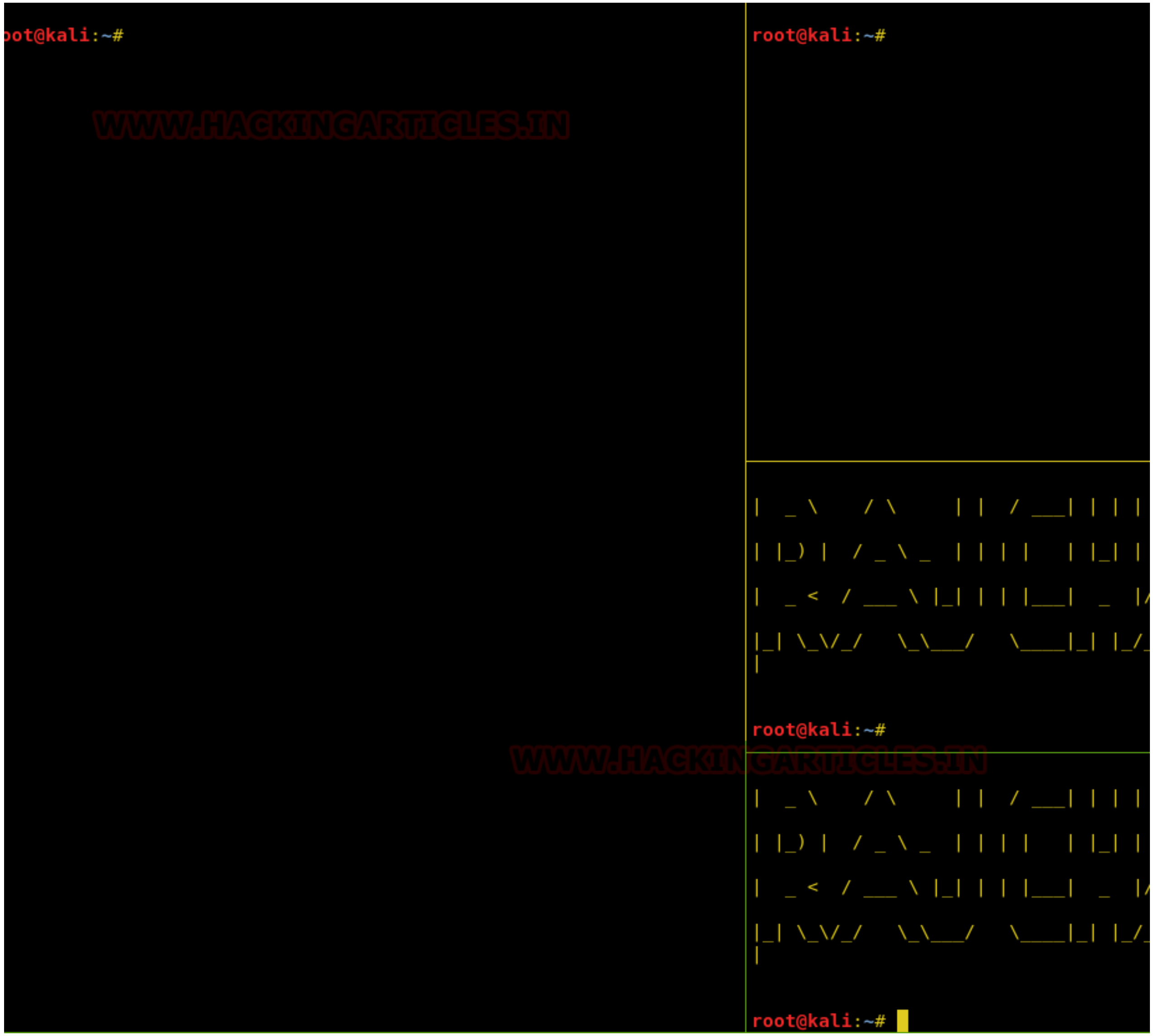


```
oot@kali:~#
```

```
root@kali:~#
```

In the below image, I have further sub-divide my window horizontally.

```
Prefix (ctrl-b) + "
```



```
oot@kali:~#
```

```
root@kali:~#
```

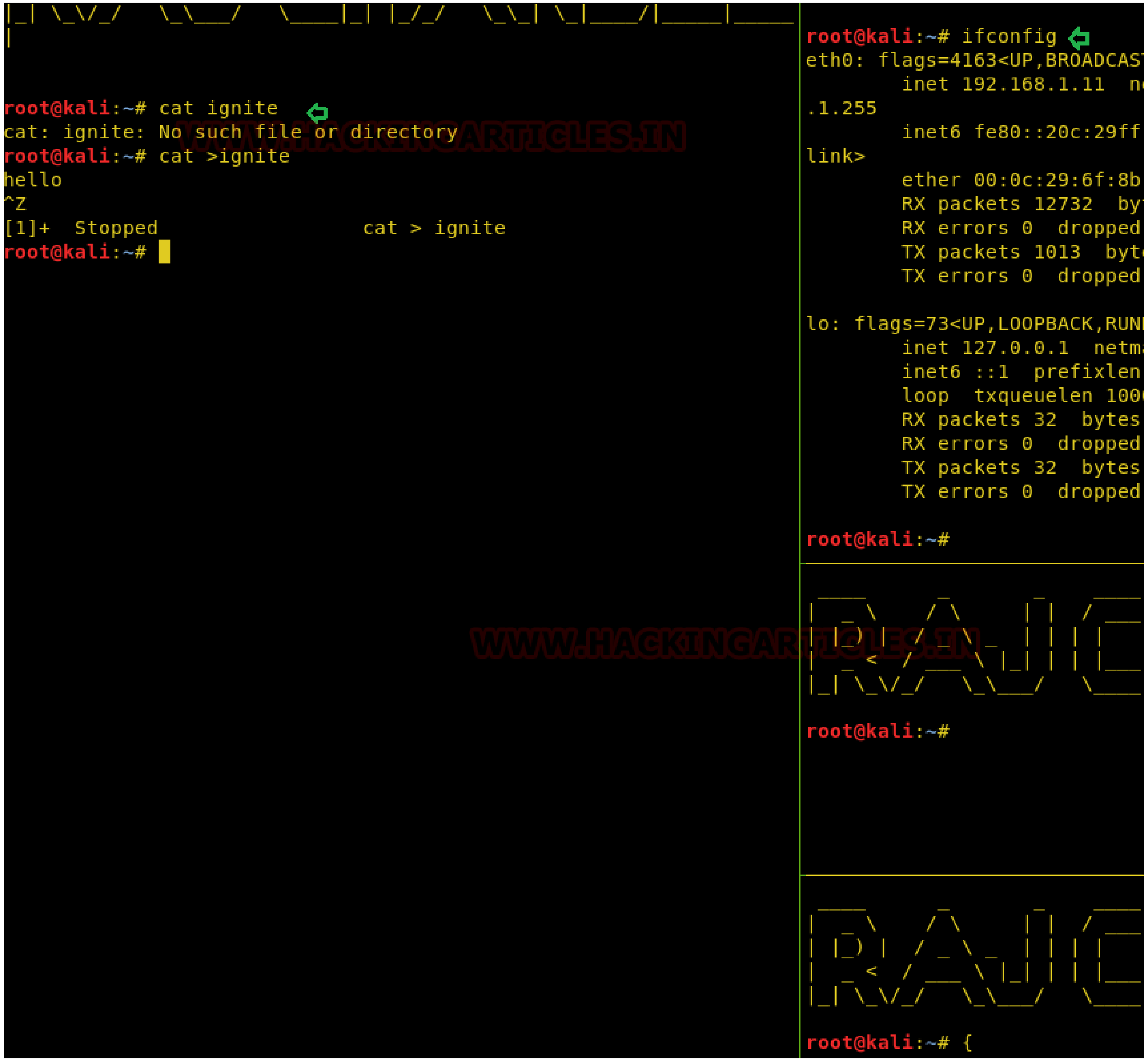
```
|  _ \    / \     | |  / ___| | | |
| |_) |  / _ \ _  | | | |    | |_| |
|  _ <  / ___ \ |_| | | |___|  _  |/
|_| \_\/_/   \_\___/   \____|_| |_/_
|

root@kali:~#
```

```
|  _ \    / \     | |  / ___| | | |
| |_) |  / _ \ _  | | | |    | |_| |
|  _ <  / ___ \ |_| | | |___|  _  |/
|_| \_\/_/   \_\___/   \____|_| |_/_
|

root@kali:~#
```

Suppose we have multiple panes containing some of the information in each and we want to rotate our panes if we desire. Then will follow the step as:

```
Prefix (ctrl-b) + {
```

On framing above command tmux will simply move the current pane to left.



```
|_| \_\/_/   \_\___/   \____|_| |_/_/   \_\_| \_|____/|_____|_____
|


root@kali:~# cat ignite
cat: ignite: No such file or directory
root@kali:~# cat >ignite
hello
^Z
[1]+  Stopped                 cat > ignite
root@kali:~# 
```

```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAS
        inet 192.168.1.11  ne
.1.255
        inet6 fe80::20c:29ff
link>
        ether 00:0c:29:6f:8b
        RX packets 12732  by
        RX errors 0  dropped
        TX packets 1013  byt
        TX errors 0  dropped

lo: flags=73<UP,LOOPBACK,RUN
        inet 127.0.0.1  netm
        inet6 ::1  prefixlen
        loop  txqueuelen 100
        RX packets 32  bytes
        RX errors 0  dropped
        TX packets 32  bytes
        TX errors 0  dropped

root@kali:~#
```

```
 ____      _      _   ____
|  _ \    / \    | | / ___
| |_) |  / _ \ _ | || |
|  _ <  / ___ \ |_| || |___
|_| \_\/_/   \_\___/  \____

root@kali:~#
```

```
 ____      _      _   ____
|  _ \    / \    | | / ___
| |_) |  / _ \ _ | || |
|  _ <  / ___ \ |_| || |___
|_| \_\/_/   \_\___/  \____

root@kali:~# {
```

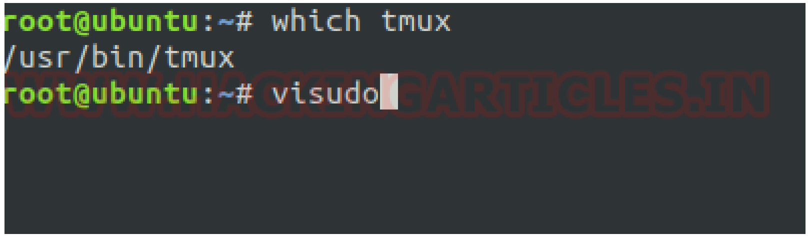
Assigning Sudo Rights

Sudo right is a type of permission that allows users to execute a file with super user permissions. Now we will start to perform privilege escalation for “tmux”. For doing so we need to set up our lab of tmux command with administrative rights. After that, we will check for the “tmux command” that what effect it has after getting sudo rights.

After that, we will give Sudo permission on tmux so that a local user can take the privilege of tmux as the root user.

Hence type following for enabling Sudo:

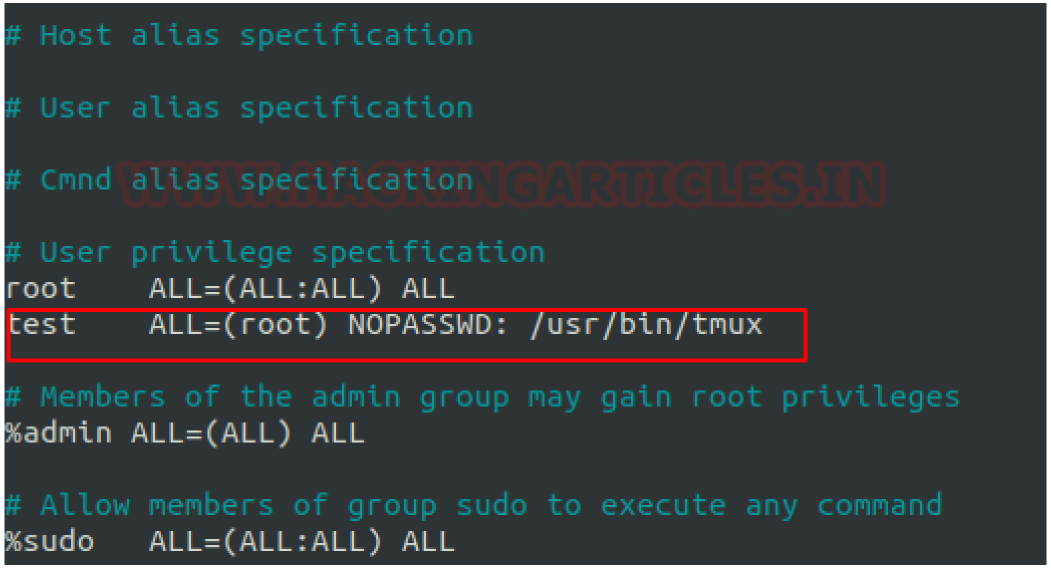
```
which tmux
visudo
```



```
root@ubuntu:~# which tmux
/usr/bin/tmux
root@ubuntu:~# visudo
```

It can be clearly understood by the below image in which I have created a local user (test). To add sudo right open /sudoers file and type following as user Privilege specification.

```
test All=(root) NOPASSWD: /usr/bin/tmux
```



```
# Host alias specification

# User alias specification

# Cmnd alias specification

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

# 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 tmux service by taking the privilege of sudoer's permission. For this, we need sessions of the victim's machine that will assist 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.

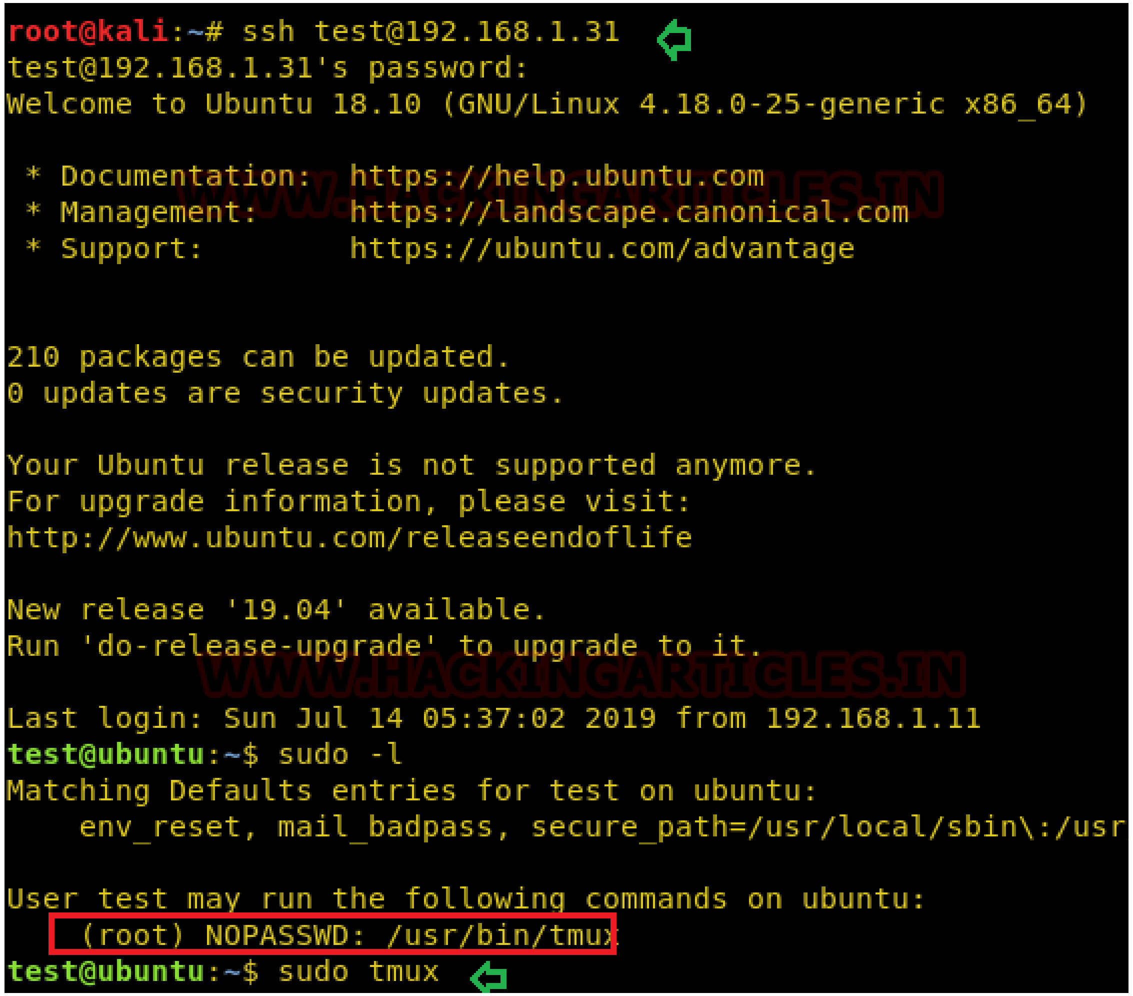
```
ssh test@192.168.1.31
```

Then we will look for sudo right of "test" user (if given) and found that user "test" can execute the tmux command as "root" without a password.

```
sudo -l
```

Now after knowing the fact that test user attains sudo rights so, taking this benefit here, we can use tmux command to escalate the privileges of the test user.

```
sudo tmux
```



```
root@kali:~# ssh test@192.168.1.31
test@192.168.1.31's password:
Welcome to Ubuntu 18.10 (GNU/Linux 4.18.0-25-generic x86_64)

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


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

Your Ubuntu release is not supported anymore.
For upgrade information, please visit:
http://www.ubuntu.com/releaseendoflife

New release '19.04' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Jul 14 05:37:02 2019 from 192.168.1.11
test@ubuntu:~$ sudo -l
Matching Defaults entries for test on ubuntu:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr

User test may run the following commands on ubuntu:
    (root) NOPASSWD: /usr/bin/tmux
test@ubuntu:~$ sudo tmux
```

Conclusion: This will launch a new terminal with root privilege shell.

```
root@ubuntu:~#
```

Author: Komal Singh is a Cyber Security Researcher and Technical Content Writer, she is completely enthusiastic pentester and Security Analyst at Ignite Technologies.
Contact **Here**

Share this:

Like this:

Loading...

ABOUT THE AUTHOR

RAJ CHANDEL

Raj Chandel is a Skilled and Passionate IT Professional especially in IT-Hacking Industry. At present other than his name he can also be called as An Ethical Hacker,

A Cyber Security Expert, A Penetration Tester. With years of quality Experience in IT and software industry

PREVIOUS POST

← CLAMP 1.0.1 VULNHUB WALKTHROUGH

NEXT POST

LINUX FOR PENTESTER: SCP PRIVILEGE ESCALATION →

2 Comments → LINUX FOR PENTESTER: TMUX PRIVILEGE ESCALATION

STEWART August 10, 2019 at 9:18 am

Great post, explained very well.

REPLY ↓

STEM August 10, 2019 at 10:37 pm

Really great information.

Thanks..

REPLY ↓

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

□ Notify me of follow-up comments by email.

□ Notify me of new posts by email.

POST COMMENT