

Creación de un banner personalizado utilizando bash para la terminal

Buenas a todos en este tutorial les enseñare a crear banners personalizados con la finalidad de que logren crear sus propios banners sin la ayuda de algun script el cual pueden llegar a estar ofuscados y contener otro tipos de acciones a cambio de un simple banner que ustedes mismos pueden crear.

~ made by m4n14ck ~

```

1 matrix.txt |
1 echo -e "\e[38;2;255;0;0meax: 00000001 ebx: f77c8c00 ecx: 00000000 edx: f77f0001"
1 echo -e "\e[38;2;255;0;0mesi: 803bf014 edi: 8023c755 ebp: 80237f84 esp: 80237f60"
2 echo -e "\e[38;2;255;0;0nds: 0018 es: 0018 ss: 0018\e[0m\n"
3
4 echo -e "\e[38;2;98;255;90m;Alguna vez has tenido un sueño, Neo, del que estabas tan segur
5 echo "¿Qué pasaría si no pudieras despertar de ese sueño?"
6 echo -e "¿Cómo sabrías la diferencia entre el mundo de los sueños y el mundo real?\e[0m\n"
7
8 echo -e "\e[38;2;139;243;255mEsta es tu última oportunidad"
9 echo "Después de esto, no hay vuelta atrás"
10 echo "Tomas la pastilla azul, la historia termina"
11 echo "Te despiertas en tu cama y crees lo que quieras creer"
12 echo "Te tomas la pastilla roja, te quedas en el País de las Maravillas"
13 echo -e "y te muestro hasta dónde llega la madriguera del conejo.\e[0m\n"
14
15 echo -e "\e[38;2;255;233;73mCode: 00 00 00 00 M3 T4 SP L0 1T FR 4M 3W OR K! V3 R5 I0 N5 00
16 echo -e "\e[38;2;255;255;255mAiee, Killing Interrupt handler"
17 echo "Kernel panic: Attempted to kill the idle task!"
18 echo -e "In swapper task - not syncing\e[0m\n"
19 echo "\n"
20
1 NORMAL - 1.1k matrix.txt text | utf-8 1: 1 All
  
```

Para este tutorial se necesitará de:

- SO derivados de gnu/linux
- Editor de texto ya sea vi, vim, nano, emacs, vscode etc
- Fundamentos básicos de bash

Bien comencemos

En su terminal creamos una carpeta llamada **banners** utilizando el comando **mkdir** para poder crear la carpeta la cual vamos a utilizar para almacenar los banners que ustedes quieran.

```
m4n14ck@kali:~$ mkdir banners _
```

una vez creada la carpeta accedemos a ella con el comando **cd + <nombre de la carpeta>**

```
m4n14ck@kali:~$ cd banners/
```

una vez dentro utilizaremos un editor de código en mi caso utilizare **vim** para crear un archivo el cual voy a llamar **pacman.txt** ustedes pueden ponerle el nombre que quieran pero la razón por la cual le puse pacman es por que el ascii o banner que voy a utilizar esta relacionado con pacman.

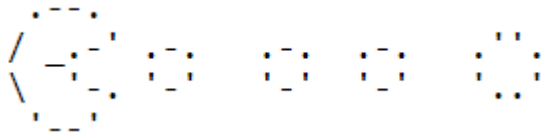
```
m4n14ck@kali:~/banners$ vim pacman.txt
```

Una vez dentro del archivo creado este no contendrá nada por lo cual nosotros vamos a tener que buscar en internet un ascii de pacman o de su personaje favorito en mi caso encuentre esta pagina:

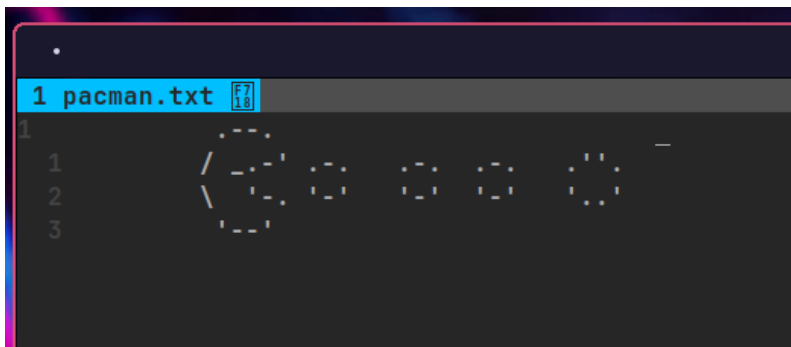
<https://www.asciiart.eu/video-games/pacman>

Dentro de esa página hay varios ascii art de pacman los cuales pueden seleccionar si están siguiendo los pasos que estoy haciendo pero como dije puede ser cualquier tipo de ascii art no tiene que afuerza ser pacman.

Este es el que voy a utilizar:



Una vez que hayan seleccionado su ascii art lo pegan en el editor.



Ahora lo que vamos a hacer es imprimir la imagen y para hacer esto vamos a usar **echo -e "texto"**.

El comando **echo** se utiliza en la terminal de Unix/Linux para imprimir texto en la pantalla. La opción **-e** que se le pasa a **echo** permite interpretar caracteres especiales en el texto que se imprime.

Para hacer esto se hace como en la imagen:

```
1 pacman.txt [F7]
3 echo -e ".--."
2 echo -e "/ _.-' .- .- .- ."
1 echo -e "\ _.-' .- .- .- ."
4 echo -e " ' _.-' .- .- .- ."
```

Una vez terminado ahora solo se tiene que guardar en mi caso como utilizo **vim** pues para guardar se hace lo siguiente: tecla **esc** seguido de **:wq** lo que hace es guardar y luego salir del editor.

```
1 COMMAND * pacman.txt text
```

Ahora lo que se tiene que hacer es entrar a **.bashrc** pero antes de todo ¿Que es **.bashrc**?

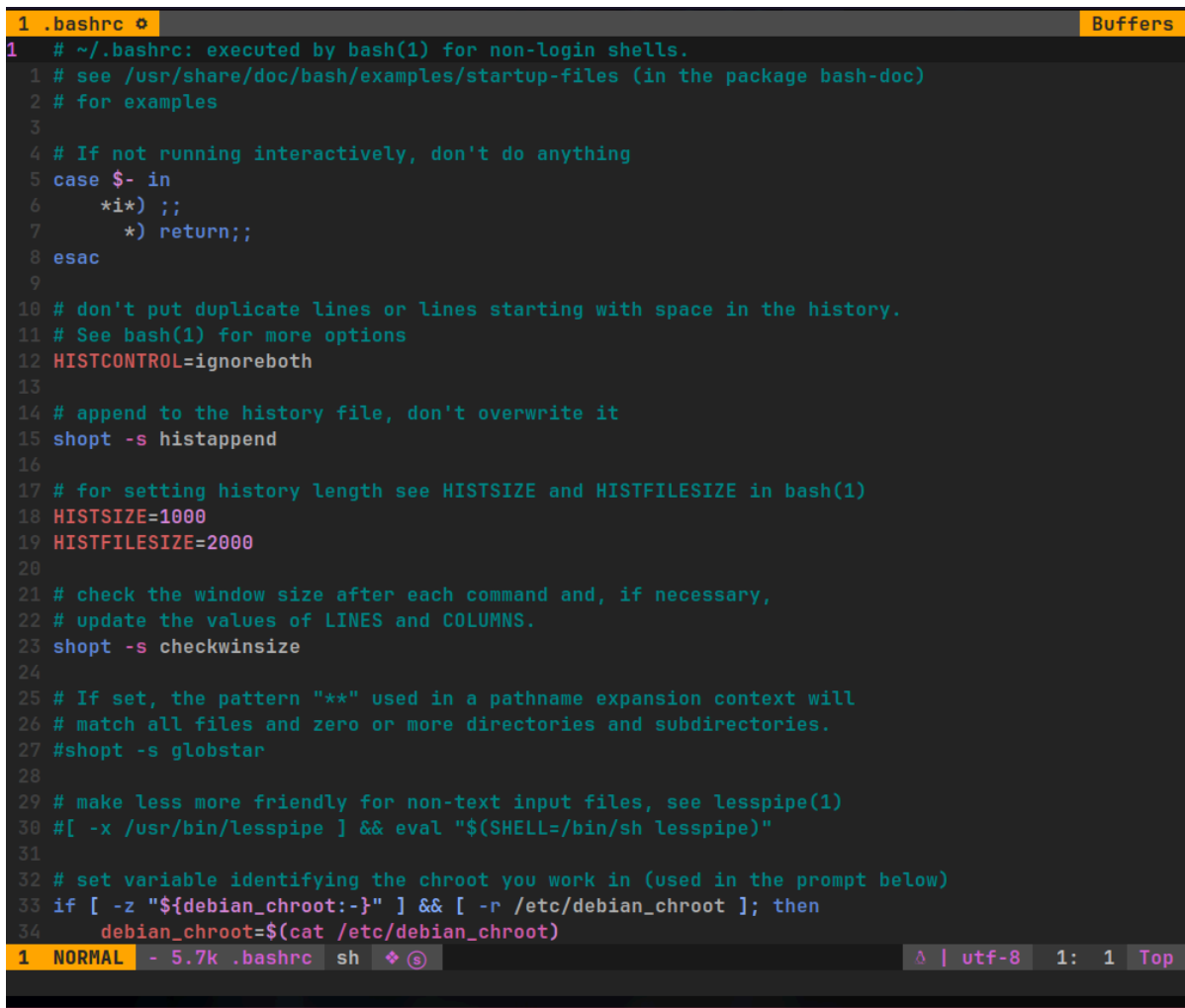
En términos simples, **.bashrc** es como una lista de instrucciones que Bash sigue cada vez que abres una nueva sesión de terminal. Puedes personalizar este archivo agregando comandos que quieras que Bash ejecute automáticamente al iniciar una sesión. Estos comandos pueden incluir configuraciones de entorno, alias de comandos, variables de entorno y funciones personalizadas, entre otras cosas.

Este se encuentra en nuestra carpeta home.

```
m4n14ck@kali:~$ pwd
/home/m4n14ck

m4n14ck@kali:~$ vim .bashrc
```

Entramos con su editor de texto preferido:



```
1 # ~/.bashrc: executed by bash(1) for non-login shells.
1 # see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
2 # for examples
3
4 # If not running interactively, don't do anything
5 case $- in
6     *i*) ;;
7     *) return;;
8 esac
9
10 # don't put duplicate lines or lines starting with space in the history.
11 # See bash(1) for more options
12 HISTCONTROL=ignoreboth
13
14 # append to the history file, don't overwrite it
15 shopt -s histappend
16
17 # for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
18 HISTSIZE=1000
19 HISTFILESIZE=2000
20
21 # check the window size after each command and, if necessary,
22 # update the values of LINES and COLUMNS.
23 shopt -s checkwinsize
24
25 # If set, the pattern "**" used in a pathname expansion context will
26 # match all files and zero or more directories and subdirectories.
27 #shopt -s globstar
28
29 # make less more friendly for non-text input files, see lesspipe(1)
30 #[ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh lesspipe)"
31
32 # set variable identifying the chroot you work in (used in the prompt below)
33 if [ -z "${debian_chroot:-}" ] && [ -r /etc/debian_chroot ]; then
34     debian_chroot=$(cat /etc/debian_chroot)
35
36 1 NORMAL - 5.7k .bashrc sh  🔍 | utf-8  1: 1 Top
```

Esto es lo que nos tendrá que aparecer una vez que hayamos logrado acceder al archivo **.bashrc**.

Si son muchas configuraciones y están en bash pero por ahora no vamos a cambiar algo de eso lo que vamos hacer es algo muy simple y para esto solo nos movemos hasta la parte casi final del archivo.

una vez estando en la parte casi final del archivo en un espacio que encuentre van a poner esto:

echo -e "\$(~/banners/pacman.txt)"

```
164 echo -e "$(~/banners/pacman.txt)"
```

y bien esto lo que hace es lo siguiente:

Juntando todo, la línea de código **echo -e "\$(/banners/pacman.txt)"** hace lo siguiente:

- Busca el archivo pacman.txt en la carpeta banners dentro del directorio de inicio del usuario.
- Lee el contenido de ese archivo.
- Utiliza el comando echo para imprimir el contenido del archivo en la terminal.
- La opción -e permite que echo interprete cualquier secuencia de escape que pueda estar presente en el texto del archivo, como colores o saltos de línea.

ya ahora solo nos queda hacer 2 cosas: guardar los cambios y recargar el archivo.

Guardo los cambios con **esc** seguido de **:wq** una vez fuera del archivo se tiene que hacer lo siguiente en la terminal se tiene que poner **source .bashrc**

Cuando ejecutas el comando **source .bashrc**, lo que estás haciendo es decirle a la terminal que vuelva a leer el archivo .bashrc y ejecute todas las configuraciones y comandos que contiene. Esto es útil cuando realizas cambios en el archivo .bashrc y deseas que esos cambios se reflejen inmediatamente en tu sesión de terminal actual, en lugar de tener que cerrar y volver a abrir la terminal.

```
m4n14ck@kali:~$ pwd
/home/m4n14ck
m4n14ck@kali:~$ source .bashrc
```

Una vez que se ejecute el comando **source** seguido de **.bashrc** pasara esto:

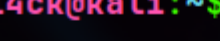
```
m4n14ck@kali:~$ source .bashrc
bash: /home/m4n14ck/banners/pacman.txt: Permission denied
m4n14ck@kali:~$
```

y eso paso por que no le dimos permiso al archivo xD se me olvido mencionarlo pero es bueno que vean este error por si les llega a pasar ustedes ya logran cómo solucionarlo y es muy sencillo lo único que tiene que hacer es ir denuevo a su carpeta de **banners** y una vez dentro de la carpeta ponen este comando:
chmod +x <texto>

```
m4n14ck@kali:~/banners$ chmod +x pacman.txt _
```

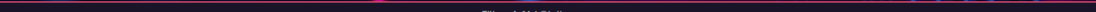
[illegible]

```
m4n14ck@kali:~$ source .bashrc
```



```
m4n14ck@kali:~$ _
```

Nos dirigimos a la carpeta donde tenemos nuestro banner y lo editamos para poder agregarle los colores que queremos y para hacer esto se necesita agregar esto `\e[38;2;r;g;b` se utiliza para especificar el color RGB, donde r, g y b son valores enteros entre 0 y 255 que representan la cantidad de rojo, verde y azul respectivamente. Las líneas que comienzan con `\e[38;2;` establecen el color del texto, y `\e[0m` se utiliza al final de cada línea para restablecer el color al predeterminado.



```

1 pacman.txt
2 echo -e '\x38;2;255;252;85m - - - - -'
3 echo -e '/'
4 echo -e '\
echo -e '\x0m'

```

m4n14ck@kali:~\$

- `\e`: Este es el carácter de escape ASCII, que indica el comienzo de una secuencia de escape ANSI en el texto.

- **[38;2:** Esta parte indica que se está seleccionando un color de texto (en lugar de un color de fondo) y que se usará el formato de color RGB de 24 bits.
- **255;252;85:** Estos son los valores RGB del color que se quiere definir. En este caso, representa un color que tiene una intensidad máxima de rojo (255), una intensidad muy alta de verde (252) y una intensidad media de azul (85).

Bien tal vez esté pensando en cómo es que yo se cual es el color amarillo y pues esta claro que yo no soy alguien con una gran capacidad para recordar todo así que para esto se utiliza una página la cual te da el color en RGB:

<https://htmlcolorcodes.com/>

le agradezco a [zegit] por recomendar la página

ya con eso pueden ponerle cualquier color que les guste.

Por mi parte ya termine de personalizar mi banner y este sería el resultado:

```

1 pacman.txt
2 echo -e "\e[38;2;255;252;85m  .--.  "
3 echo " / _.' .-. .-. .-. .-' "
4 echo " \ '._' '._' '._' '._' "
5 echo -e " '._'          \e[0m\n"
6
7 echo -e "\e[38;2;255;252;85m Code: 00 00 00 00 M3 T4 SP L0 1T FR 4M 3W OR K! V3 R5 IO N5 00 00 00 \e[0m"
8 echo -e "\e[38;2;255;255;255m Aiee, Killing Interrupt handler\e[0m"
9 echo -e "\e[38;2;255;0;0m Kernel panic: Attempted to kill the idle task!"
10 echo -e " In swapper task - not syncing\e[0m\n"
11
12 echo -e "\e[38;2;98;234;255m ----->>>\e[0m"
13 echo "\n"

```

```

echo -e "\e[38;2;255;252;85m  .--.  "
echo " / _.' .-. .-. .-. .-' "
echo " \ '._' '._' '._' '._' "
echo -e " '._'          \e[0m\n"

echo -e "\e[38;2;255;252;85m Code: 00 00 00 00 M3 T4 SP L0 1T FR 4M 3W OR K! V3 R5 IO N5 00 00 00 \e[0m"
echo -e "\e[38;2;255;255;255m Aiee, Killing Interrupt handler\e[0m"
echo -e "\e[38;2;255;0;0m Kernel panic: Attempted to kill the idle task!"
echo -e " In swapper task - not syncing\e[0m\n"

echo -e "\e[38;2;98;234;255m ----->>>\e[0m"
echo "\n"

```

Básicamente este sería el resultado final:

```

  .--.
 /  .  .  .  .  .
 \  .  .  .  .  .
  '--'

Code: 00 00 00 00 M3 T4 SP L0 1T FR 4M 3W OR K! V3 R5 IO N5 00 00 00 00
Aiee, Killing Interrupt handler
Kernel panic: Attempted to kill the idle task!
In swapper task - not syncing

----->>>

m4n14ck@kali:~$ _
```

Ahora usted puede hacer diferentes tipos de banners

```

m4n14ck@kali:~/banners$ echo -e "$(~/banners/matrix2.txt)"

-----
|                                     |
|               3Kom SuperHack II Logon               |
|-----|
| ¿Alguna vez has tenido un sueño, Neo, del que estabas tan seguro de que era real? |
| ¿Qué pasaría si no pudieras despertar de ese sueño? |
| ¿Cómo sabrías la diferencia entre el mundo de los sueños y el mundo real? |
|
| Esta es tu última oportunidad |
| Después de esto, no hay vuelta atrás |
| Tomas la pastilla azul, la historia termina |
| Te despiertas en tu cama y crees lo que quieras creer |
| Te tomas la pastilla roja, te quedas en el País de las Maravillas |
| y te muestro hasta dónde llega la madriguera del conejo. |
|-----|
| [ ] Select Option or ESC to Abort | M4tr1x |
|-----|

m4n14ck@kali:~/banners$ _
```

Bien eso sería todo por este tutorial espero y le haya gustado hasta luego.