



# The Power of Linux “History Command” in Bash Shell

Narad Shrestha | Last Updated: March 15, 2024 | Read Time: 5 mins | [Linux Commands](#) | [32 Comments](#)

We frequently use the [history command](#) in our daily routine jobs to check the [history of commands](#) or to get information about commands executed by users.

In this post, we will see how we can use the history command effectively to extract the commands that were executed by users in the Bash shell. This may be useful for audit purposes or to find out which command was executed at a [specific date and time](#).

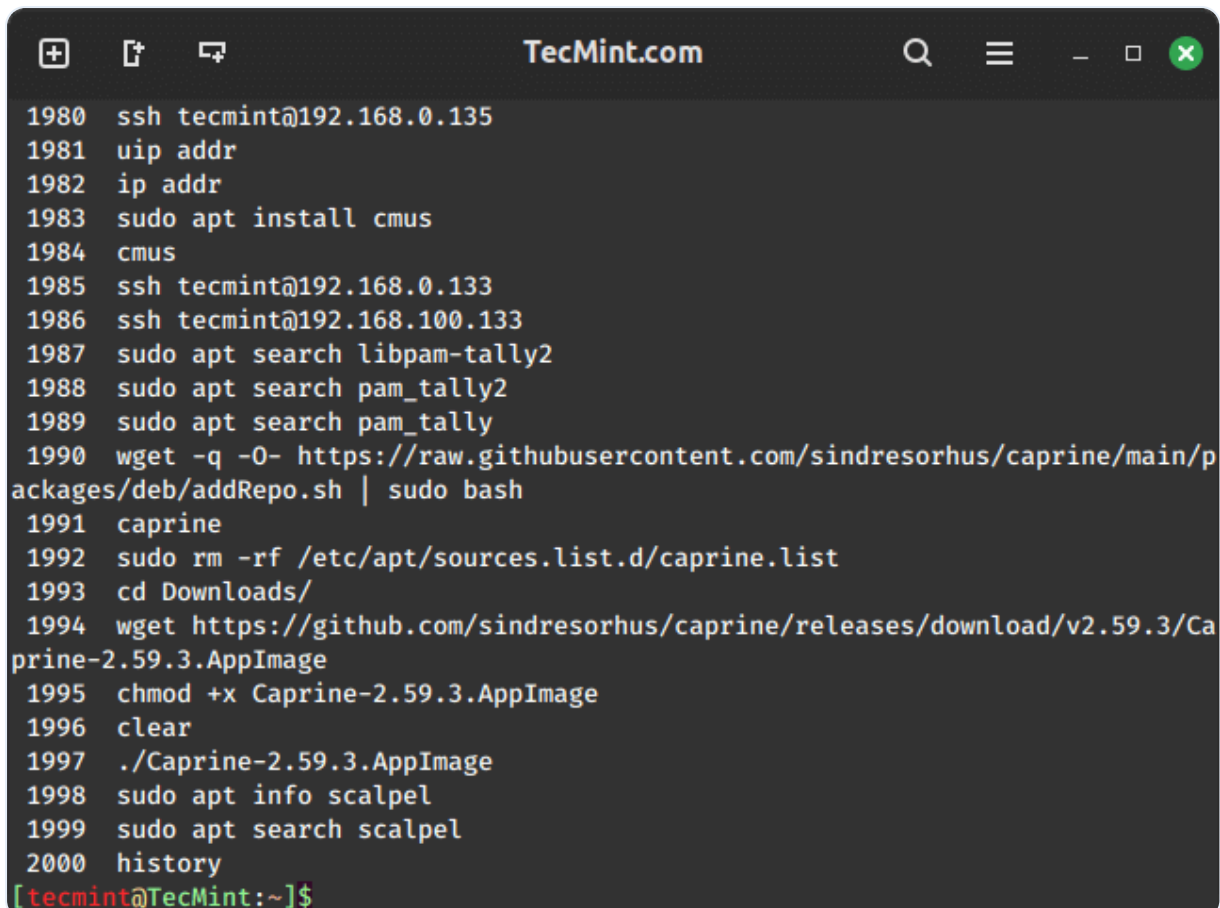
By default, the date and timestamp won't be visible when executing the history command. However, the bash shell provides command-line interface (CLI) tools for editing the user's command history.

Let's explore some useful tips, tricks, and the power of the history command.

## 1. List All Last Executed Commands in Linux

To view the complete list of the last executed commands along with their line numbers, use the following command.

```
history
```



```
1980 ssh tecmint@192.168.0.135
1981 uip addr
1982 ip addr
1983 sudo apt install cmus
1984 cmus
1985 ssh tecmint@192.168.0.133
1986 ssh tecmint@192.168.100.133
1987 sudo apt search libpam-tally2
1988 sudo apt search pam_tally2
1989 sudo apt search pam_tally
1990 wget -q -O- https://raw.githubusercontent.com/sindresorhus/caprine/main/packages/deb/addRepo.sh | sudo bash
1991 caprine
1992 sudo rm -rf /etc/apt/sources.list.d/caprine.list
1993 cd Downloads/
1994 wget https://github.com/sindresorhus/caprine/releases/download/v2.59.3/Caprine-2.59.3.AppImage
1995 chmod +x Caprine-2.59.3.AppImage
1996 clear
1997 ./Caprine-2.59.3.AppImage
1998 sudo apt info scalpel
1999 sudo apt search scalpel
2000 history
[tecmint@TecMint:~]$
```

Show Last Executed Commands

## 2. List All Commands with Date and Timestamp

To list all commands with their date and timestamp in Linux, you can use the history command along with the HISTTIMEFORMAT environment variable as shown.

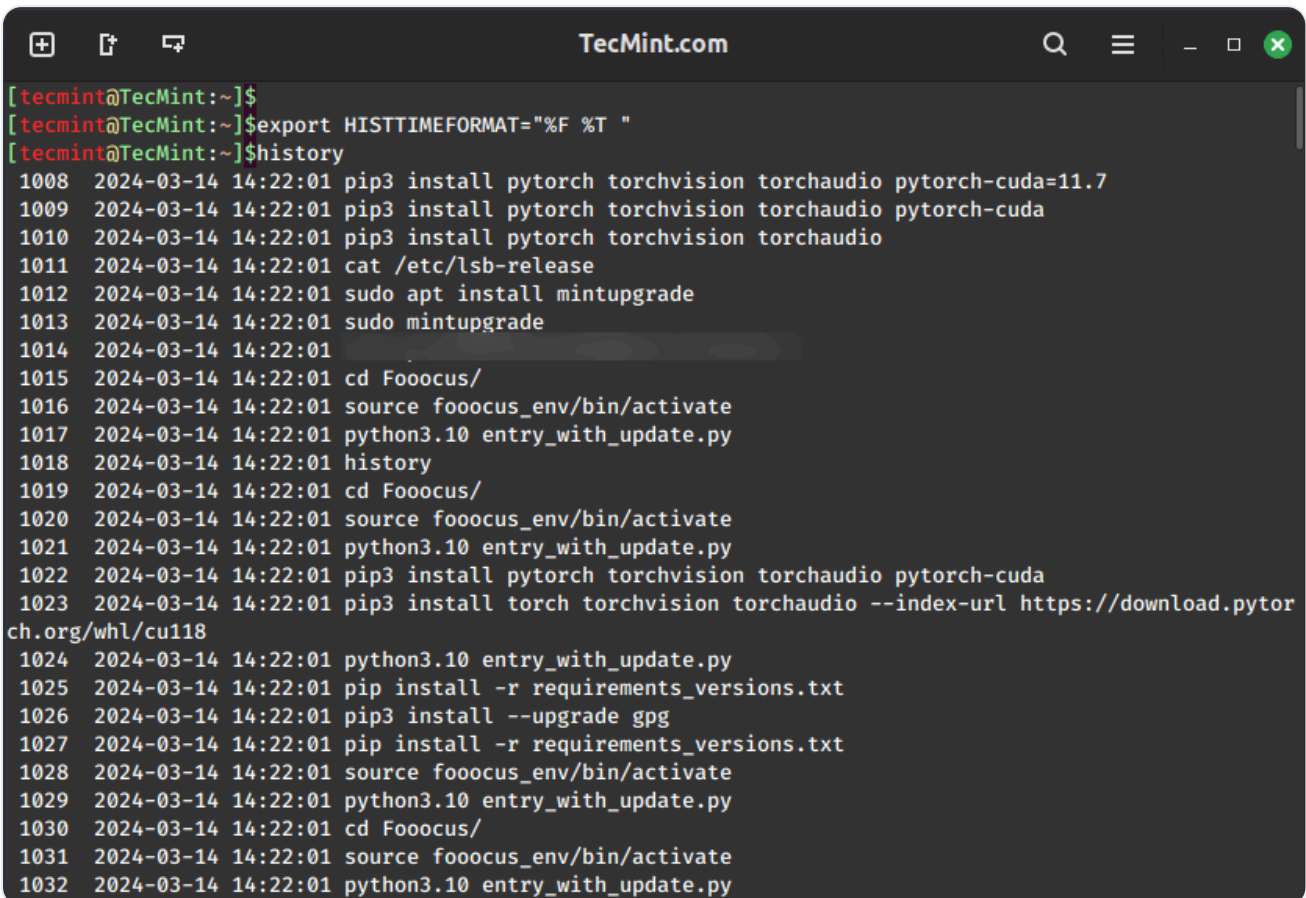
```
export HISTTIMEFORMAT="%F %T "
```

Now, you can use the history command to list all the commands with their respective date and timestamps.

```
history
```

Here's a breakdown of the format placeholders used in `HISTTIMEFORMAT='%F %T '`:

- `%F`: This signifies the full date in the format "YYYY-MM-DD".
- `%T`: This represents the time in the format "HH:MM:SS".



```
[tecmint@TecMint:~]$  
[tecmint@TecMint:~]$export HISTTIMEFORMAT="%F %T "  
[tecmint@TecMint:~]$history  
1008 2024-03-14 14:22:01 pip3 install pytorch torchvision torchaudio pytorch-cuda=11.7  
1009 2024-03-14 14:22:01 pip3 install pytorch torchvision torchaudio pytorch-cuda  
1010 2024-03-14 14:22:01 pip3 install pytorch torchvision torchaudio  
1011 2024-03-14 14:22:01 cat /etc/lsb-release  
1012 2024-03-14 14:22:01 sudo apt install mintupgrade  
1013 2024-03-14 14:22:01 sudo mintupgrade  
1014 2024-03-14 14:22:01  
1015 2024-03-14 14:22:01 cd Fooocus/  
1016 2024-03-14 14:22:01 source fooocus_env/bin/activate  
1017 2024-03-14 14:22:01 python3.10 entry_with_update.py  
1018 2024-03-14 14:22:01 history  
1019 2024-03-14 14:22:01 cd Fooocus/  
1020 2024-03-14 14:22:01 source fooocus_env/bin/activate  
1021 2024-03-14 14:22:01 python3.10 entry_with_update.py  
1022 2024-03-14 14:22:01 pip3 install pytorch torchvision torchaudio pytorch-cuda  
1023 2024-03-14 14:22:01 pip3 install torch torchvision torchaudio --index-url https://download.pytor  
ch.org/whl/cu118  
1024 2024-03-14 14:22:01 python3.10 entry_with_update.py  
1025 2024-03-14 14:22:01 pip install -r requirements_versions.txt  
1026 2024-03-14 14:22:01 pip3 install --upgrade gpg  
1027 2024-03-14 14:22:01 pip install -r requirements_versions.txt  
1028 2024-03-14 14:22:01 source fooocus_env/bin/activate  
1029 2024-03-14 14:22:01 python3.10 entry_with_update.py  
1030 2024-03-14 14:22:01 cd Fooocus/  
1031 2024-03-14 14:22:01 source fooocus_env/bin/activate  
1032 2024-03-14 14:22:01 python3.10 entry_with_update.py
```

Enable Timestamp in History Command

### 3. Ignore Commands in History

To ignore a command in the history command, you can use the `"HISTIGNORE"` environment variable, which is used to specify which commands should not be recorded in the command history.

You can set it by using the `"export"` command followed by the desired command to be ignored.

```
export HISTIGNORE='ls -l:pwd:date:'
```

In this example, `"ls -l"`, `"pwd"`, and `"date"` are specified, meaning that any commands containing these strings will not be saved in the history.

## 4. Ignore Duplicate Commands in History

To ignore duplicate commands in history, you can use the following command which will instruct the system to not save repeated commands in the history, which will help in keeping the history clean and free of redundant duplicate entries.

```
export HISTCONTROL=ignoredups
```

The option `ignoredups` tells the system to ignore duplicate commands when recording them in the history. If you execute the same command multiple times consecutively, only the first occurrence will be stored in the history.

## 5. Save Export Command Permanently

To save the configurations for `HISTTIMEFORMAT`, `HISTIGNORE`, and `HISTCONTROL` permanently in your Linux environment, you can add them to your `.bash_profile` file, which is executed every time you start a new shell session.

```
nano ~/.bash_profile
```

Add the following lines to the `.bash_profile` file to set the environment variables:

```
export HISTTIMEFORMAT="%F %T "  
export HISTIGNORE="some:commands:to:ignore"  
export HISTCONTROL=ignoredups
```

To apply the changes, either restart your terminal or run the following command in the terminal:

```
source ~/.bash_profile
```

## 6. Unset Export Command

To unset the settings for `HISTTIMEFORMAT` and `HISTCONTROL` environment variables, you can use the `unset` command to remove their values, which will revert these settings to their default configurations.

```
unset HISTTIMEFORMAT
unset HISTCONTROL
```

After executing these commands, the timestamp display in the command history will revert to the default setting, and duplicate commands will no longer be filtered out automatically.

## 7. List Specific User's Executed Commands

To list specific user's executed commands, you can use the `.bash_history` file of the user, which stores the history of commands executed in the Bash shell.

```
sudo cat /home/username/.bash_history
```

If you want to filter the command history based on a specific pattern, use the [grep command](#) along with the commands containing the word as shown.

```
sudo cat /home/username/.bash_history | grep "ls"
```

## 8. Disable Storing History of Commands

To disable the storing of command history in Linux, you can unset the `HISTFILE` variable, which is responsible for maintaining the history of commands in a file.

```
unset HISTFILE
```

By unsetting the `HISTFILE` variable, you prevent the system from storing command history, ensuring that commands executed in the terminal are not recorded for future sessions.

It's important to note that this change will only apply to the current session and will not persist across different terminal sessions. If you want to make this change permanent, you

can consider updating your shell's configuration `.bash_profile` file to unset the `HISTFILE` variable upon each login.

## 9. Delete or Clear History of Commands

You can use the up and down arrow keys to see previously used commands, which can be helpful or annoying. To delete or clear all the entries from the bash history list, you can use the `'-c'` option.

```
history -c
```

To clear the command history for all users, you can delete or truncate the history file located at `/home/[username]/.bash_history` for each user.

```
> /home/[username]/.bash_history  
OR  
rm /home/[username]/.bash_history
```

## 10. Search Commands in History Using Grep Command

To filter commands in the history command output, you can use the [grep command](#), which will only display the commands from the history that contain the keyword as shown.

```
history | grep "ls"  
history | grep "pwd"  
history | grep "date"
```

## 11. Search for Recent Commands in History

To search for recent commands in the history, you can use the `grep` command along with containing keyword such as "ssh" from the command history.

```
history | grep "ssh"
```

You can also search for previously executed commands using the '**Ctrl+r**' command. Once you've found the command you're looking for, press 'Enter' to execute it, or press 'esc' to cancel.

```
(reverse-i-search)`source ': source .bash_profile
```

## 12. Recall Last Executed Command in History

To recall the last executed command from a specific position in the command history in Linux, you can use the history command along with a specific line number.

To view the command history with line numbers, use the history command:

```
history
```

Next, identify the line number associated with the command you want to recall, in this case, command number 8, and then recall the command using **!** followed by the line number as shown.

```
!8
```

## 13. Recall Lastly Executed Specific Command

You can recall a previously used command, such as "netstat -np | grep 22", by using the  **'!'** symbol followed by some letters from that command.

For example, if you want to recall the "netstat -np | grep 22" command, you can type  **'!net'** and press Enter in the terminal, which will execute the most recent command that starts with "net" from your command history.

```
!net
```

## Conclusion

In conclusion, we've shown you the usefulness of the history command, but there's still more to explore. We'd love to hear about your experiences with the history command!

Share them with us in the comment box below.

Hey TecMint readers,

Exciting news! Every month, our top blog commenters will have the chance to win fantastic rewards, like free Linux eBooks such as RHCE, RHCSA, LFCS, Learn Linux, and Awk, each worth \$20!

Learn [more about the contest](#) and stand a chance to win by [sharing your thoughts below!](#)



# GIVEAWAY!

## Win eBooks



[www.tecmint.com](http://www.tecmint.com)

PREVIOUS ARTICLE:

**[Is Linux Operating System Virus Free?](#)**

NEXT ARTICLE:

**[Shellngn – Best Online SSH Client with SFTP, VNC, RDP](#)**



## Narad Shrestha

He has over 10 years of rich IT experience which includes various Linux Distros, FOSS and Networking. Narad always believes sharing IT knowledge with others and adopts new technology with ease.

*Each tutorial at TecMint is created by a team of experienced Linux system administrators so that it meets our high-quality standards.*

Join the [TecMint Weekly Newsletter](#) (More Than 156,129 Linux Enthusiasts Have Subscribed)

Was this article helpful? Please [add a comment](#) or [buy me a coffee](#) to show your appreciation.

## Related Posts

```
tecmint@tecmint ~/testing $ find . -type f \( -name "*.txt" -o -  
name "*.sh" -o -name "*.c" \)  
./emails.txt  
./script-1.sh  
./header.c  
./examples.txt  
./script.sh  
./expenses.txt
```

## Find Multiple Filenames (File Extensions) Using 'find' Command in Linux

How to Search Files by Name or Extension Using find Command



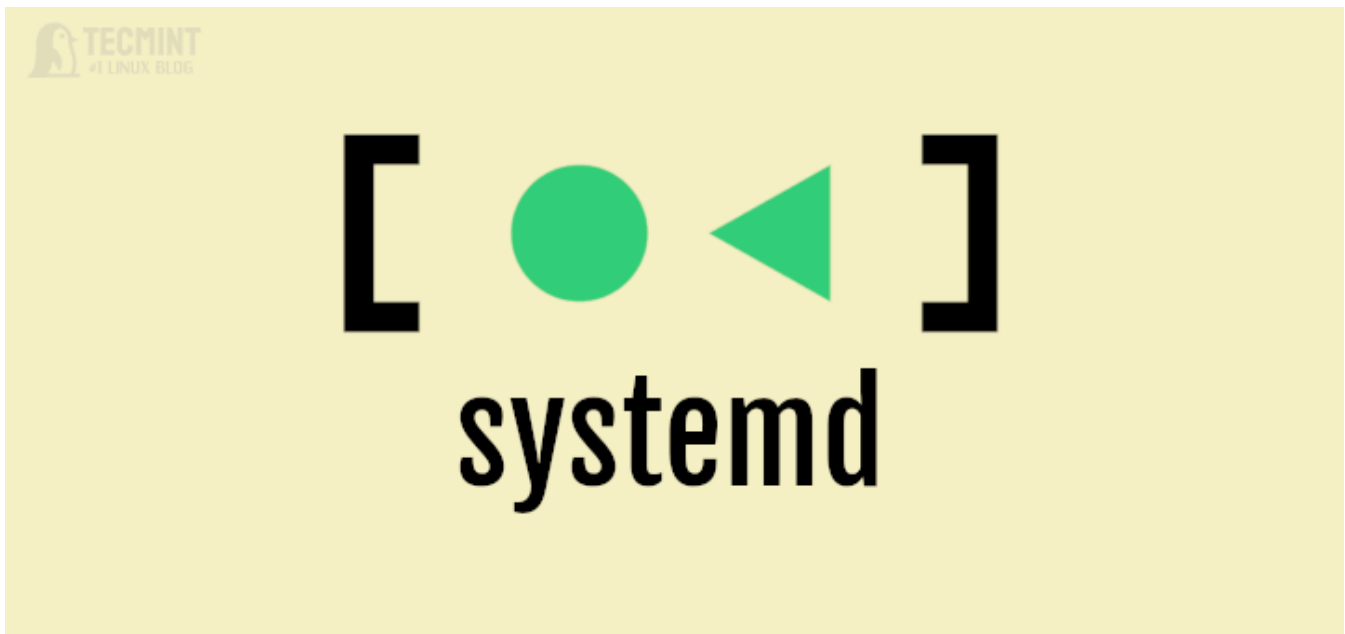
10 Lesser Known Linux Commands – Part 2



## 11 Lesser Known Useful Linux Commands



## 26 Security Hardening Tips for Modern Linux Servers



## How to Remove Systemd Services on Linux

```
TecMint.com
ravi@TecMint:~/glibc-2.39/build$
ravi@TecMint:~/glibc-2.39/build$ ../configure --prefix=/usr/local/glibc-2.39
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking for g++... g++
checking whether the compiler supports GNU C++... yes
checking whether g++ accepts -g... yes
checking for g++ option to enable C++11 features... none needed
checking whether g++ can link programs... yes
checking for sysdeps preconfigure fragments... aarch64 alpha arc arm csky hppa i386 loong
arch m68k microblaze checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
mips nios2 orlk powerpc riscv s390 sh checking for grep that handles long lines and -e...
(cached) /usr/bin/grep
```

## How to Install and Run Multiple glibc Libraries in Linux

 **32 Comments**

[Leave a Reply](#)

Narendra

February 3, 2021 at 5:09 pm

How I can delete history between two-time stamps.

[Reply](#)

**jimm**

February 4, 2021 at 12:46 pm

<https://askubuntu.com/questions/708906/how-to-remove-command-line-history-from-a-specific-time-period>

[Reply](#)

**jimm**

February 4, 2021 at 12:49 pm

Then again, the \*only\* reason to delete history between two points is to hide nefarious, embarrassing, or even illegal, activity on someone's machine. :P

[Reply](#)



**jimm**

February 4, 2021 at 12:51 pm

The only reason to delete history is to hide something you are doing.

[Reply](#)

**Edward**

December 4, 2019 at 6:59 am

How can i display the last five commands when i login like this :

The last five commands you executed last login were:

```
566 2019-10-25 20:30:19 > ls
567 2019-10-25 20:30:22 > ls -l
568 2019-10-25 20:30:27 > clear
569 2019-10-25 20:30:32 > test
```

[Reply](#)**jack**

November 29, 2018 at 3:53 pm

How can I filter a many of commands in history? for example i don't want to show me "ls" command in history file.

```
HISTIGNORE='ls' command don't work for me
```

[Reply](#)**the digitalmouse**

December 1, 2018 at 2:30 pm

Perhaps the easiest is to just filter what you are looking \*for\* instead of what you are not looking for. Just a standard 'history | grep (thing you want to find)' is

enough? Otherwise you should be able to construct a regex in the grep to search what you want \*and\* ignore the ls lines found.

[Reply](#)



**the digitalmouse**

December 1, 2018 at 2:32 pm

remember to 'export' that command before using it

[Reply](#)



**the digitalmouse**

August 30, 2016 at 11:19 am

Great stuff! I would suggest updating the title text for points 11, 12, and 13 to read: "Recall Last...", not "Recall Lastly..." just to make the tips a bit more readable and could possibly improve search engine results. It is also proper grammar. :) Keep up the good work!

[Reply](#)

**SGTItlog**

March 29, 2016 at 4:51 pm

To clear (totally) all the command history of a certain user (e.g root)



1. Log-in to that user
2. cat "" > .bash\_history (same as cat /dev/null > .bash\_history)
3. history -c

Logout and re-login.

[Reply](#)

Admin



**Ravi Saive**

March 29, 2016 at 4:53 pm

@Sugat,

Thanks for the tip on history command, hope it will help other Linux users..

[Reply](#)

**Dragos Alexe**

September 8, 2015 at 4:58 pm

Dear Narad,

If you allow me one suggestion. The ignoredups remove consecutive duplicates; erasedups- eliminate duplicates across the whole history.

So, if I type multiple times-but not consecutive- the same commands in bash:

1.ls -l

2.dir -a

3.ls -l- the way to remove duplicates from history is : export

HISTCONTROL=erasedups.

Great work!!

Thanks!!!!!!!!!!!!!!

[Reply](#)**Milton Reyes**

July 29, 2015 at 5:33 am

HISTSIZE=0 is nice if you don't want to leave traces of history increasing security, however your shell is your working tool and sometimes you need to view the commands you typed previously which you can't with this limitation.

Consider this:

```
ln -s -f /dev/null ~/.bash_history
```

This approach will keep the history through your session and erase it when you close shell or logout

[Reply](#)**Jack**

June 19, 2015 at 5:43 am

Thanks for this nice summary on the "history" command. I've been using the method with grep as

in paragraph 10 for years. Now I've upgraded my linux system and get the following:

```
$ history | grep pwd
```

Binary file (standard input) matches

Any idea what might be misconfigured, making grep consider the output of history to be binary, even when it clearly isn't?

[Reply](#)

**ilya**

April 20, 2015 at 2:46 pm

Good article! Just one question: can anyone explain why all commands in the first example have the same time stamp, 2013-06-09 10:40:12 ?

Even if you cut and pasted them, commands "ping google.com" would take more than one second, would not they?

Occasionally I see the same anomaly in my history logs; I wonder if these time stamps are reliable at all?

[Reply](#)**Frank**

April 7, 2017 at 9:30 am

Good question, I really want to know as well.

[Reply](#)**unclez**

May 15, 2020 at 11:41 am

I am confused too

[Reply](#)

**charm aliros**

December 12, 2014 at 5:31 pm

This article saved tons of my worries. I am not a linux cli expert and I am asked to configure an l2tp server. Did not know how to add iptables rules and so checking the previous commands in a different server, I was able to copy and execute the proper commands!

Cheers!

[Reply](#)**Shubham**

June 25, 2020 at 6:16 pm

Good article, I really need this. Thanks

[Reply](#)**Satyendra Jaiswal**

July 21, 2014 at 7:16 pm

Hi,

I really very thankful for your post , it's very useful for me , I am a beginner in Linux , we can delete all history using "\$ history -c" command, but I want to delete my history from last 1 month, or any specified duration , how can we delete by command ,

[Reply](#)**Martin**

July 15, 2014 at 9:59 am

Perhaps you could consider BASH history suggest box <https://github.com/dvorka/hstr> that greatly simplifies navigation through the history and its management.

[Reply](#)**shabu**

May 7, 2014 at 7:10 pm

Hi,  
while using this commnad "export HISTSIZE=0" it wil delete all users history or related to me(the commands which has executed by me) in the global user account.

[Reply](#)**Subu**

April 1, 2014 at 5:28 pm

Hi,

We need the ssh session commands to be redirected to a file everytime they login.

We have added the below lines under bashrc.

```
HISTFILE=/var/tmp/history/$USER.$(date +%d-%m-%Y:%H:%M)
```

```
readonly HISTFILE
```

```
HISTSIZE=5000
```

```
HISTTIMEFORMAT="%d/%m/%y %T "
```

Its logging all the commands but, when we run history it shows only session history where we want that to show the complete history..

Please help. thanks in advance.

Subu.

[Reply](#)

**rehan**

March 24, 2014 at 9:01 am

var/log/secure provides us below information :

```
Mar 20 08:07:07 testing sshd[29749]: Accepted password for oracle from 10.51.1.12  
port 49239 ssh2
```

```
Mar 20 08:07:07 testing sshd[29749]: pam_unix(sshd:session): session opened for  
user oracle by (uid=0)
```

```
Mar 20 08:12:16 testing sshd[29759]: Received disconnect from 10.51.1.12: 11:  
Disconnect requested by Windows SSH Client.
```

```
Mar 20 08:12:16 testing sshd[29749]: pam_unix(sshd:session): session closed for  
user oracle
```

But what commands that oracle user executed we can not get from the above information?

/home/oracle/.bash\_history provides us information like this about what commands are executed:

```
exit
cat /home/oracle/ashfaq/Auto_Clone/Auto_Clone.sh
cat /u01/down.sh
cd /u01/oracle/inst/apps/TESTING_testing/admin/scripts/
./adstpall.sh apps/apps
./adstrtal.sh apps/apps
cat /home/oracle/ashfaq/Auto_Clone/Auto_Clone.sh
ls -ltr
cat main_exec.sh
```

but does not tell us that user from which IP or hostname and at what time executed these commands.

how we can store all this information in one file with time, IP, user name (root, oracle, ricky ) and commands executed by these users ?

[Reply](#)

**Lee**

December 23, 2013 at 5:39 pm

We can modify History of command ..?

[Reply](#)

**jack**

December 5, 2013 at 1:44 pm

Hi

I want history like below. is it possible ?

year time command username

05/12/13 11:07:36 ls -lah

[Reply](#)

**Tom**

March 9, 2014 at 1:10 am

Hi Jack, of course it's possible.

Edit your ~/.bashrc and set the line

HISTTIMEFORMAT

to:

HISTTIMEFORMAT="%F %T ".

Exit all your terminals and now it should write a timestamp in the history file.  
(Maybe you have to log out!)

[Reply](#)

**Rahul**

November 24, 2013 at 8:11 am

Good one, I got a lot from this

[Reply](#)

**VIVEK**

June 22, 2013 at 4:56 pm



Sir

I am linux trainer. these cmds are very useful

THANKYOU

WITH REGARDS

VIVEK

[Reply](#)



**Ron**

June 17, 2013 at 9:16 am

Thanks for #2.

[Reply](#)



**Chris F.A. Johnson**

June 15, 2013 at 6:19 pm

5. unset export HISTCONTROL

This command unsets two variables, export and HISTCONTROL. It does not just remove HISTCONTROL from the environment; it unsets it entirely.

[Reply](#)

**Purandaran**

June 16, 2013 at 9:28 am

Yes. It's true.

unset HISTCONTROL command is enough for deletion.

[Reply](#)

### Got Something to Say? Join the Discussion...

*Thank you for taking the time to share your thoughts with us. We appreciate your decision to leave a comment and value your contribution to the discussion. It's important to note that we moderate all comments in accordance with our [comment policy](#) to ensure a respectful and constructive conversation.*

*Rest assured that your email address will remain private and will not be published or shared with anyone. We prioritize the privacy and security of our users.*

☐ Save my name, email, and website in this browser for the next time I comment.

Post Comment

## Do You Enjoy My Blog?

Support from readers like YOU keeps this blog running. Buying me a cup of coffee is a simple and affordable way to show your appreciation and help keep the posts coming!

[Buy Me a Coffee](#)

## Linux Commands and Tools

[How to Change Runlevels \(targets\) in SystemD](#)

[How to Use 'dir' Command with Different Options and Arguments in Linux](#)

[How to Fix "bash: curl: command not found" Error](#)

[6 Useful Tools to Troubleshoot DNS Name Resolution Problems](#)

[How to Pipe Command Output to Other Commands in Linux](#)

[7 Interesting 'sort' Command Examples in Linux – Part 2](#)

## Linux Server Monitoring Tools

[How to Add Windows Host to Nagios Monitoring Server](#)

[Install Nagios Core on openSUSE 15.3 Linux](#)

**SARG – Squid Analysis Report Generator and Internet Bandwidth Monitoring Tool**

**Installing “PHP Server Monitor” Tool using LEMP or LAMP Stack in Arch Linux**

**Watchman – A File and Directory Watching Tool for Changes**

**Cockpit – A Powerful Tool to Monitor and Administer Multiple Linux Servers via Browser**

## **Learn Linux Tricks & Tips**

**How to Configure Custom SSH Connections to Simplify Remote Access**

**How to Disable Shutdown and Reboot Commands in Linux**

**How to Delete Root Mails (Mailbox) File in Linux**

**How to Encrypt and Decrypt Files and Directories Using Tar and OpenSSL**

**Understanding Shutdown, Poweroff, Halt and Reboot Commands in Linux**

**How to Check Remote Ports are Reachable Using ‘nc’ Command**

## **Best Linux Tools**

**5 Open Source Log Monitoring and Management Tools for Linux**

**12 Best Notepad++ Alternatives for Linux in 2024**

**11 Best GUI Tools for Linux System Administrators in 2024**

**5 Best Skype Alternatives for Linux Desktop**

**10 Best Linux File and Disk Encryption Tools (2024)**

**17 Best RSS Feed Readers for Linux in 2024**

Tecmint: Linux Howtos, Tutorials & Guides © 2024. All Rights Reserved.

The material in this site cannot be republished either online or offline, without our permission.

Hosting Sponsored by : [Linode Cloud Hosting](#)