

STOP PUTIN NOW

**Getting Started
Becoming a
Master
Hacker**

Hacking is the Most Important
Skill Set of the 21st Century!

Occupytheweb

Buy Now!

 Log In

Training Schedule

⋮

Updated: Dec 31, 2022

AccessData FTK 1.61a DEMO VERSION

File Edit View Tools Help

Overview Explore Graphics E-Mail Search Bookmark

Evidence Items

File Items: 18
Total File Items: 18
Checked Items: 0
Unchecked Items: 18
Flagged Thumbnails: 0
Other Thumbnails: 1
Filtered In: 18
Filtered Out: 0
Unfiltered: 18
All Items: 18
Actual Files: 18

File Status

KFF Alert Files: 0
Bookmarked Items: 0
Bad Extension: 0
Encrypted Files: 0
From E-mail: 0
Deleted Files: 0
From Recycle Bin: 0
Duplicate Items: 0
OLE Subitems: 0
Flagged Ignore: 0
KFF Ignorable: 0
Data Carved Files: 0

File Category

Documents: 0
Spreadsheets: 0
Databases: 0
Graphics: 1
E-mail Messages: 0
Executables: 0
Archives: 0
Folders: 1
Slack/Free Space: 16
Other Known Type: 0
Unknown Type: 0

File Name: FAT2, FileSlack, LostFileChain0001
Full Path: F:\TEST-FAT16\FAT2, F:\TEST-FAT16\Water files.jpg>FileSlack, F:\TEST-FAT16\LostFileChain0001
File Type: File Allocation..., File Slack, Bad sectors/..., Category: Slack/Free S..., Subject: N/A, Cr Date: N/A, Mod D.: N/A

18 Listed 0 Checked Total F:\TEST-FAT16\LostFileChain0001

Saved Command History

Before we explore ways to cover of BASH tracks, we need to know a little about this command history. Linux stores your command history in the hidden file, `~/.bash_history`. We can see its contents by typing;

```
kali > more ~/.bash_history
```



This would be the same way a skilled forensic investigator would find our history. This may not be a good thing.

Disabling History

If we want to keep the BASH shell from saving our history of commands, we can set the environment variable **HISTSIZE** to zero. HISTSIZE determines how many commands are being stored. By default, it is set to 500 or 1000.

We can keep the BASH shell from saving our commands by typing;

```
kali > export HISTSIZE=0
```



Now, the shell will no longer save our command history, but we will lose the convenience of retrieving our commands while working on our Linux system. This can increase the amount of time required for hacking as we re-write each command. I hate doing that.

Clearing the History

Rather than disabling command history, we can clear the history on the current BASH shell by simply using the **history** command with the **-c** (clear) switch.

```
kali > history -c
```



Then, to make certain the changes are written to disk, we need to tell the history command to do so with the `-w` switch such as;

```
kali > history -w
```

This only clears the history of the **current shell**. Commands run in other shells will remain, so we need to use this command in every shell to completely remove our command history. probably not the most convenient way of clearing our history and it requires us to remember and go back and clear each shell we were using. That must be a better way.

Clearing the User's Complete History

Probably a more effective and convenient way to erase our command history is to write `/dev/null` to the **bash_history** file and then clear the current shell with `history -c`. We can construct a command that does exactly that and then exits the shell. This can be done by typing;

```
kali > cat /dev/null > ~/.bash_history && history -c && exit
```



Now, when we look at the shell history or the `bash_history`, we can see that they are empty.



Shredding the History

Although its great to delete these files, as we know, a knowledgeable forensic investigator can still recover these deleted files. BASH has a command, **shred**, that does just what it implies, it shreds the target file. In this case, even if the forensic investigator is able to locate the history file, it will be unreadable.

```
kali > shred ~/.bash_history
```



We can see here that when we take a look at this file, it is simply a bunch of binary data that is nonsense.

The fact that the file is shredded may in itself be considered evidence of malicious activity. We can eliminate even that by combining the **shred** command with writing `/dev/null` to the `bash_history`. This would first, shred the history file, then empty it

and finally clear the evidence of the command itself by using the `history -c` command and exiting.

```
kali > shred ~/.bash_history && cat /dev/null > .bash_history && history -c && exit
```



Now, when someone goes to view our command history , it will be empty and when they go to the trouble of trying to recover the deleted file they will only find the shredded file. Success! We left no evidence behind and successfully covered our tracks!

Automating the Clearing of Command History

Finally, we might want to automate this process so that our command history is deleted each day. In this way, if we forget to remove our history (I'm sure I will often), the system will do it for us at the end of the day, automatically.

First, open the crontab table in edit mode by typing; .

```
kali > crontab -e
```

Using the crontab, we can navigate to the end of the file and add the following line.

```
1 * * * shred ~/.bash_history && cat /dev/null > .bash_history
```

This command will execute each morning at 1am, first shredding the `bash_history` and then erasing the `bash_history`. Note, that I did not include the `history -c` command as it is an internal BASH shell command and can not be used in crontab.

Keep coming back my tenderfoot hackers as we explore the most valuable skillset of the 21st century -- Hacking!

[!\[\]\(5ffb4f7d2924e9e409dde5575185e8f2_img.jpg\)](#) [!\[\]\(8e7454729608bc53ed3dac8a0f06d27a_img.jpg\)](#) [!\[\]\(658c7e8a7edd6256da377bc92a57edaa_img.jpg\)](#) [!\[\]\(531b5a23d50a5e1548063f8d7899a87f_img.jpg\)](#)

8,189 views

17 



71

2

Available for Pre-Order

201

2

Keeping you Safe and Secure?

965

6