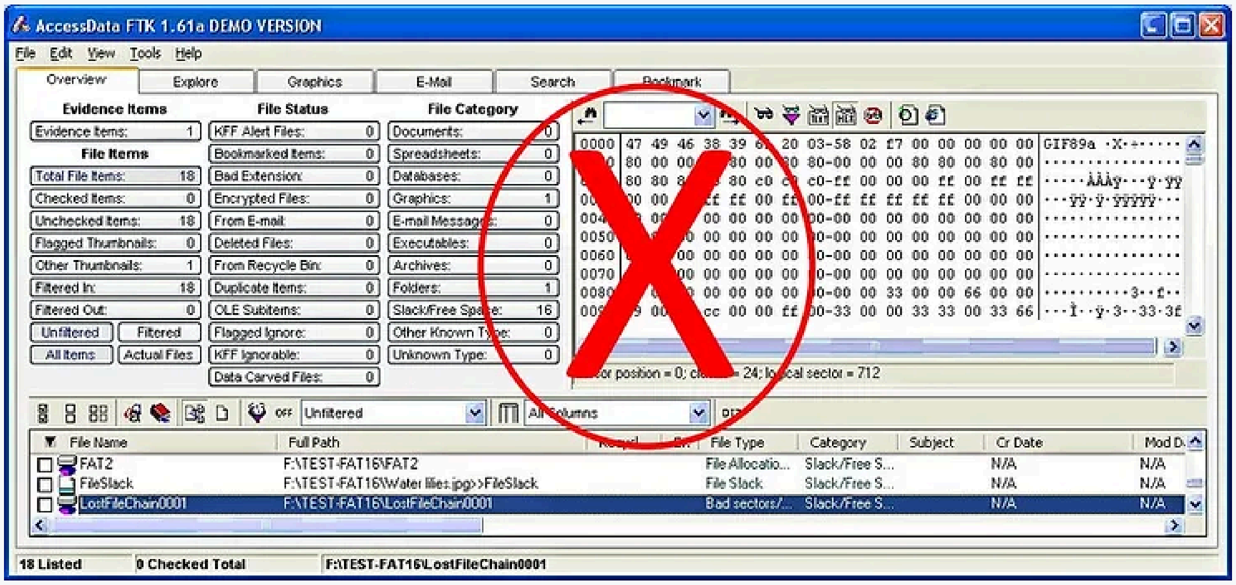
Hackers-Arise operates 2 parallel websites
Hackers-Arise.com & Hackers-Arise.net
STOP PUTIN NOW
HACKERS-ARISE
THE BEST PLACE TO LEARN CYBERSECURITY
Getting Started Becoming a Master Hacker
TOP 5 CYBERSECURITY BOOKS OF ALL TIME!
BECOMING A CYBER WARRIOR
Buy Now!

otw · Jun 20, 2016 · 4 min read

Covering your BASH Shell Tracks- Anti-Forensics

Updated: Dec 31, 2022

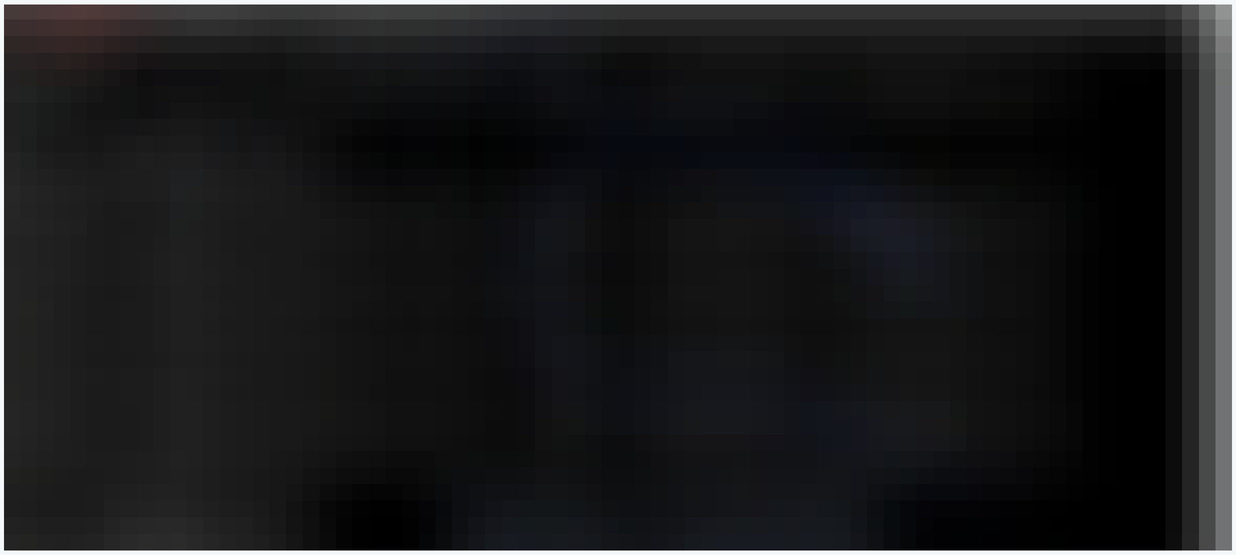
Those of us who use the BASH shell regularly, love the fact that our history can be recovered simply by using the up and down arrows. This saves us significant time in re-writing our commands. The BASH or **B**ourne **A**gain **Sh**ell saves the command history is a file called bash_history. At the same time, this convenient BASH history can be damning in the event of an investigation into our activities on the system. As nearly all hacking is done from the Linux command line, our activities can easily be deciphered and recovered from retrieving those commands.



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;

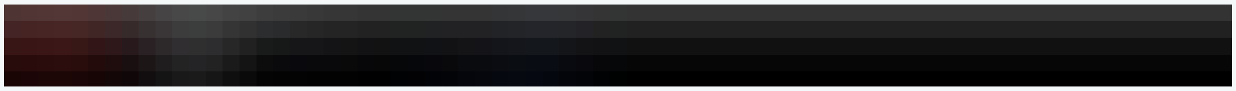
kali > export HISTSIZE=0

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

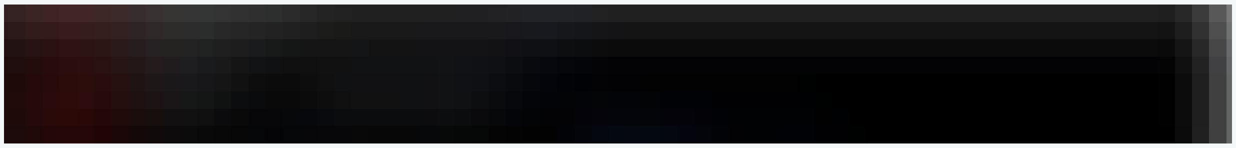
This only clears the history of the **current shell.** Commands run in other shells will remain, so we need to use this command in every shell to completely remove our command history. probably not the most convenient way of clearing our history and it requires us to remember and go back and clear each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

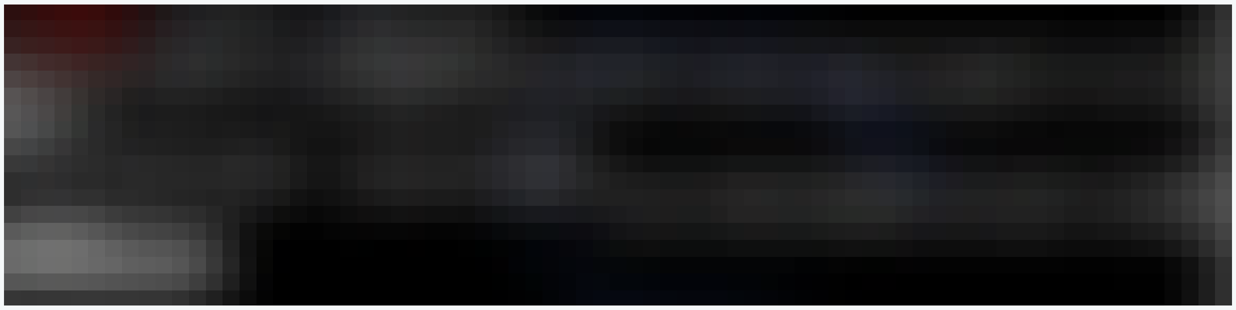
Now, when we look at the shell history or the bash_history, we can see that they are empty.



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.

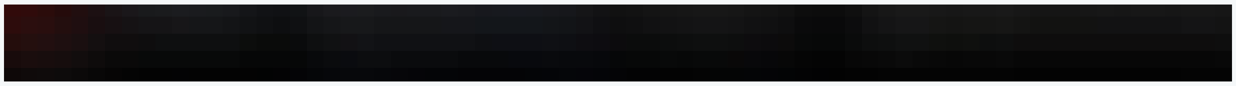
kali > shred ~/.bash_history



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!

8,187 views 17

Recent Posts See All

Python Basics for Hacker is Now Available for Pre-Order

35

So...You Thought Your VPN Was Keeping you Safe and Secure?

892 5

Data Science Analytics for Cybersecurity, December 17-19

288 6