

# Forum: [Hack Forums](#)

Thread: [\[Help!\] Bash Shell Scripting Functions](#)

**Blackhat.r1sh** - December 5, 2012, 3:20 pm

## MISSION:

I'm creating bash scripts that will do a number of tasks as an amateur server admin (Root).

Any tips on achieving any the different functions are greatly appreciated. I could use all the help I can get! I'll have a rough notes/sources section and will be making mini-scripts to test each function so I can then add them to the final master scripts.

## BRIEFING:

-All scripts must be accessed by root only (See Solved! :D)

-Create two new CLI commands 'lock' and 'unlock' from scripts named 'lock' and 'unlock' in /usr/local/sbin/. The 2 master scripts can be made up of multiple scripts as long as there are two main ones which root operates (lock/unlock).

-lock/unlock has the option to work with input from files

## Lock/Unlock File Input Example

Spoiler [\(Click to View\)](#)

-Create an accounts file (and optional path, absolute or relative) with the read permission where each user account id (one per line) is locked or unlocked with the master scripts.

```
xxx# lock -f accounts.text
Accounts user567 user678 locked
xxx#
```

Where accounts.text is:

```
user567
user678
```

## LOCK/UNLOCK NECESSARY REQUIREMENTS

-Prompt for and read in any number of account ids from stdin (one per line) to be locked or unlocked; detect end-of-file (Control-D) in order to terminate the script. This is the no arguments case.

#Example:

```
Spoiler \(Click to View\)xxx# lock
Enter Account: user123
Enter Account: ^D (This is ctrl-d)
Accounts user123 locked
xxx#
```

-Use of the passwd -l and passwd -u commands to lock accounts in the script.

-Use any number of arguments as account ids to be locked or unlocked with no practical limit to the number of accounts that can be supplied as long as the command line does not exceed 256 characters total.

#Example

Spoiler ([Click to View](#))xxx# unlock user123 user234 user345 user456  
Accounts user123 user234 user345 user456 unlocked  
xxx#

-Use command-line arguments to determine input method:

If no arguments, ask for an account or list of accounts one at a time from stdin (Solved)

If 2 arguments and the first is -f, read the list of accounts one at a time from the second argument as a file

Otherwise, use the arguments as the list of accounts

In each case, process each item of the list of accounts as follows:

- Accounts must exist

- Examine password for valid-locked or valid-unlocked ("man lock/unlock" command mentions this, will add to rough notes for explanation)

- If command is lock and password is valid-unlocked, lock it

- If command is unlock and password is valid-locked, unlock it

- Issue messages and log information as required.

\*(echo, logger/rsyslog.conf edit)

SOLVED

Root access only to scripts:

Spoiler ([Click to View](#))# Authenticate User

```
if [[ $EUID -ne 0 ]]; then
```

```
echo "Access Denied" 2>&1
```

```
exit 1
```

```
#Else continue the script!
```

```
else ..
```

```
exit 0
```

```
fi
```

ROUGH NOTES

To check if username exists, search /etc/passwd file

Example:

```
grep ^user /etc/passwd
```

Default arguments \$0,\$1,\$2,etc..

\$0= file name

\$1= first argument

\$2= second argument

etc...

\$@ and \$#= all arguments. They have subtle differences

## FINISHED SCRIPTS

This space is reserved for finished scripts. They have been tested and work 100% of the time.

### Example Test Script 1: Root Access

Spoiler ([Click to View](#))#!/bin/bash

```
#Filename: rootacc
```

```
# Authenticate User: If their EUID is 0(root) proceed. If not, exit script.
```

```
if [[ $EUID -ne 0 ]]; then
```

```
echo "Access Denied" 2>&1
```

```
exit 1
```

```
#Else continue the script
```

```
else echo "Welcome Root!"
```

```
exit 0
```

```
fi
```

### Example Test Script 2: Verify User Existence

Spoiler ([Click to View](#))#!/bin/bash

```
#filename ver
```

```
read -p "Username: " username
```

```
if ! egrep -q "^$username:" /etc/passwd; then
```

```
echo "Error: Username doesn't exist"
```

```
else echo "Username Verified"
```

```
fi
```

**Gent** - December 5, 2012, 3:29 pm

[spoiler] [/ spoiler]

[code] [/ code]

Those two are for code snippets and spoilers.

**Blackhat.r1sh** - December 5, 2012, 3:32 pm

Thanks! Editing now..

I'll get the hang of forums sooner or later..

Updated the rough notes with some useful info.

Feel free to post any hints, tips or suggestions!

Adding to scripts now... two have been made so far for testing phase