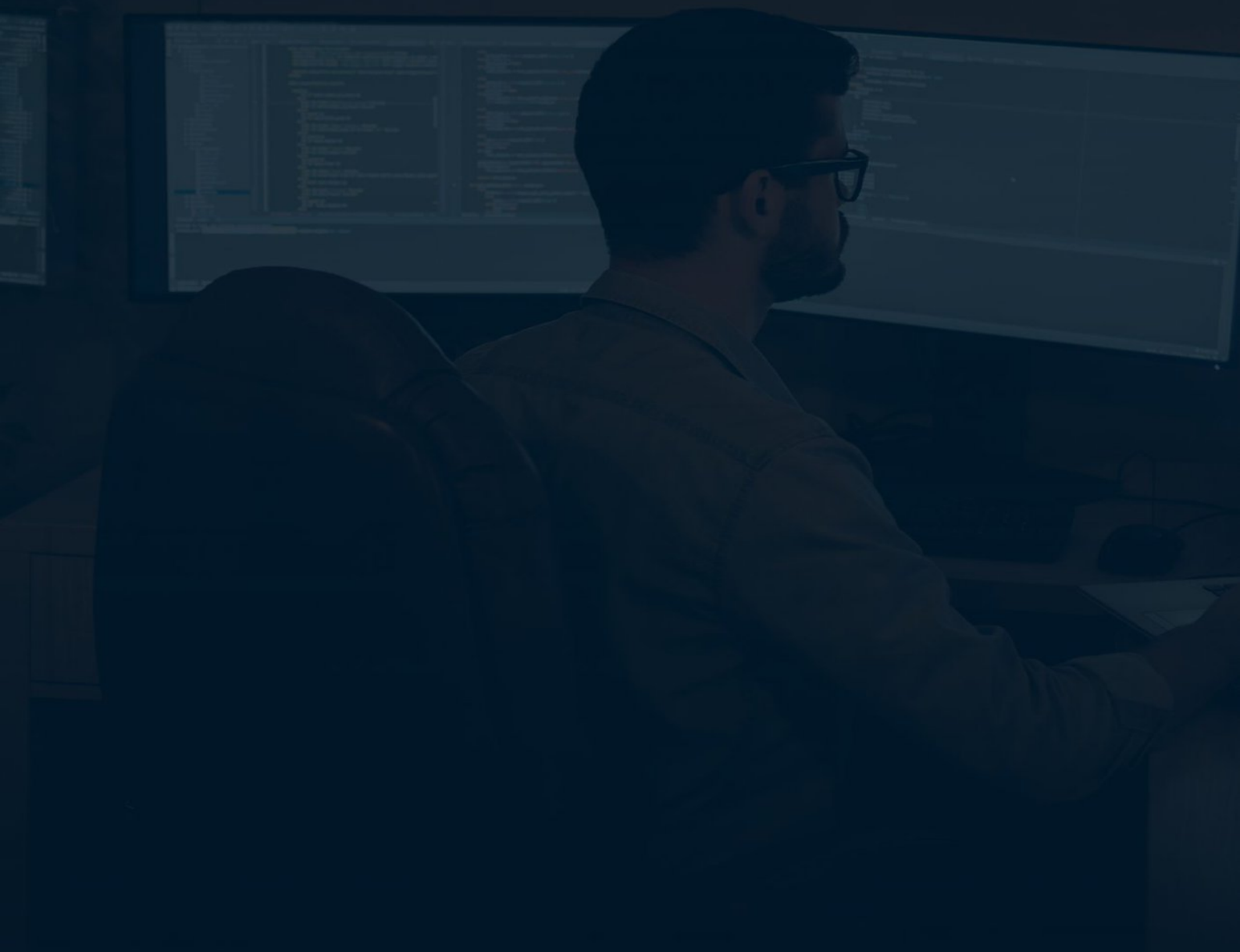
OPEN-SOURCE EBOOK

++101 Linux Commands



BOBBY ILIEV

Table of Contents

101 Linux commands Open-source eBook

This is an open-source eBook with 101 Linux commands that everyone should know. No matter if you are a DevOps/SysOps engineer, developer, or just a Linux enthusiast, you will most likely have to use the terminal at some point in your career.

The `who` command

The `who` command lets you print out a list of logged-in users, the current run level of the system and the time of last system boot.

Examples

1. Print out all details of currently logged-in users

```
who -a
```

2. Print out the list of all dead processes

```
who -d -H
```

Syntax:

```
who [options] [filename]
```

Additional Flags and their Functionalities

Short Flag	Description
`-r`	prints all the current runlevel
`-d`	print all the dead processes
`-q`	print all the login names and total number of logged on users

Short Flag	Description
-h	print the heading of the columns displayed
-b	print the time of last system boot

The `free` command

The `free` command in Linux/Unix is used to show memory (RAM/SWAP) information.

Usage

Show memory usage

Action: --- Output the memory usage - available and used, as well as swap

Details: --- The values are shown in kibibytes by default.

Command:

```
free
```

Show memory usage in human-readable form

Action: --- Output the memory usage - available and used, as well as swap

Details: --- Outputted values ARE human-readable (are in GB / MB)

Command:

```
free -h
```

Privacy Considerations

While the `finger` command is useful for retrieving information about system users, it may also expose sensitive details in shared or multi-user environments:

1. **Usernames and Login Times**: Displays login times, which can be used to track user activity.
2. **Home Directories**: Exposes paths to users' home directories.
3. **Idle Status**: Shows how long a user has been inactive, potentially signaling whether they are actively using their system.
4. **Mail Status**: Displays mail information, which may inadvertently reveal user engagement.

Potential Risks:

In environments with untrusted users, the information exposed by `finger` could be exploited for:

- **Social Engineering Attacks**: Malicious actors could use this information to craft personalized phishing attacks.
- **Timing Attacks**: Knowing when a user is idle or active could give attackers an advantage in timing their attempts.
- **Targeted Attacks**: Knowledge of user home directories can focus attacks on those locations.

Mitigating Privacy Risks:

To mitigate these risks, consider limiting access to the `finger` command in environments where user privacy is important.

The `in.fingerd` Service

It’s important to distinguish between the `finger` command and the **`in.fingerd` service**. The `finger` command is local, while `in.fingerd` is a network daemon that allows remote queries of user information. This service is typically disabled by default in modern systems due to potential security risks.

If enabled, the `in.fingerd` service can expose user information over the network, which could be exploited by attackers. To mitigate this risk, system administrators should ensure the service is disabled if it is not needed.

Disabling the `in.fingerd` Service:

If you are concerned about remote queries, you can disable the `in.fingerd` service:

```
sudo systemctl disable in.fingerd
sudo systemctl stop in.fingerd
```

By disabling the `in.fingerd` service, you prevent remote querying of user information, enhancing system security.

This is a sample from "101 Linux Commands" by Bobby Iliev and the Hacktoberfest community.

For more information, Click here.