

How to Use Full Disk Encryption on OpenBSD 7.0

Learn how to implement full disk encryption on OpenBSD 7.0 to secure your data. This step-by-step guide covers installation, configuration, and best practices.

Contents

01	Introduction	3
02	Prerequisites	3
03	1. Upload the OpenBSD Amd64 7.0 Image to Vultr	3
04	2. Deploy a New Server Instance with the Newly Added OpenBSD Amd64 7.0 Image	4
05	3. Install OpenBSD Amd64 7.0 to Your Server Instance from the noVNC Console	4
06	4. Next Steps	8
07	More Information	9

Introduction

When deploying a new server instance, one of the security best practices is to enable full disk encryption on the machine, adding a second layer of security to your data. This article explains how to set up an OpenBSD 7.0 server instance with full disk encryption at Vultr.

Prerequisites

- An active Vultr user account.
- A modern web browser, such as Chrome or Firefox, that supports noVNC connections

1. Upload the OpenBSD Amd64 7.0 Image to Vultr

Because the OpenBSD amd64 7.0 image is not in the public Vultr ISO Library yet, you need to upload it to Vultr by yourself.

1. Point your web browser to the [official OpenBSD download page](#), and then find the download link to the OpenBSD amd64 7.0 image named [install70.iso](#).
2. Log in to the [Vultr Control Panel](#) in a new browser window, navigate to the **ISOs** tab, and click the **Add ISO** button.
3. Copy the download link mentioned above, paste it into the text box on the page, and click the **Upload** button to start uploading.
4. If nothing goes wrong, the MD5 value of the uploaded OpenBSD amd64 7.0 image should be **a6a8091dceaa6a88972902b6616b38e8**. Delete the image and upload it again if you find a different MD5 value.

2. Deploy a New Server Instance with the Newly Added OpenBSD Amd64 7.0 Image

Having the OpenBSD amd64 7.0 image in place, hover your mouse over the blue + icon on the upper right corner of the Vultr Control Panel page, and then click **Deploy New Server** to deploy a new server instance.

Among other settings, in the **Server Type** section, be sure to choose **install70.iso** from the **My ISOs** line within the **Upload ISO** tab.

After your machine gets up and running, collect its networking information from the **Settings** tab in the **Server Details** page for later use. The IP addresses listed below are for demonstration purposes only.

- IPv4 address: 203.0.113.100
- Netmask: 255.255.254.0
- Gateway: 203.0.113.1
- Nameserver: 108.61.10.10 (Find it in the **networking configuration** page.)

3. Install OpenBSD Amd64 7.0 to Your Server Instance from the noVNC Console

Note: Some browser extensions, including but not limited to Vimium, could interfere with your input in the noVNC console window. Disable those extensions when using a noVNC console.

Setup Full Disk Encryption on the Server Instance

Click the **View Console** button on the upper right corner of the **Server Details** page to open a noVNC console window. Use this console window to access your server instance and begin the installation of OpenBSD amd64 7.0 with full disk encryption:

```
Welcome to the OpenBSD/amd64 7.0 installation program.  
(I)nstall, (U)pgrade, (A)utoinstall or (S)hell? s:key_enter:
```

In the OpenBSD shell, use the following command to find any hard disks on your server instance:

```
# dmesg | grep "^[sw]d":key_enter:  
sd0 at scsibus0 targ 0 lun 0: <VirtIO, Block Device, >  
...
```

As you see, **sd0** stands for a single Small Computer System Interface (SCSI) hard disk on this machine, which is the most common configuration on Vultr.

To make the OpenBSD 7.0 system recognize that SCSI hard disk, create a device file with the same name **sd0** in the system using the **MAKEDEV** script:

```
# cd /dev && sh MAKEDEV sd0:key_enter:
```

Optionally, overwrite the whole disk with random data to prevent unauthorized space usage deductions:

```
# dd if=/dev/urandom of=/dev/rsd0c bs=1m:key_enter:
```

Use the **fdisk** command to write a default Master Boot Record (MBR) boot code to the **sd0** disk:

```
# fdisk -iy sd0:key_enter:  
Writing MBR at offset 0.
```

Create a partition layout on **sd0** with the **disklabel** command, allocating all available disk space:

```
# disklabel -E sd0:key_enter:  
Label editor (enter '?' for help at any prompt)  
sd0> a a:key_enter:  
offset: [64] :key_enter:  
size: [52420031] *:key_enter:  
FS type: [4.2BSD] RAID:key_enter:
```

```
sd0*> w:key_enter:  
sd0> q:key_enter:
```

Build an encrypted device named **softraid0** with the **sd0a** partition:

```
# bioctl -c C -l sd0a softraid0:key_enter:  
New passphrase: YourOwnPassphrase:key_enter:  
Re-type passphrase: YourOwnPassphrase:key_enter:  
sd1 at scsibus2 targ 1 lun 0: <OPENBSD, SR CRYPTO, 006>  
...  
softraid0: CRYPTO volume attached as sd1
```

Important: Don't lose the passphrase you input here, or you won't be able to access the data within the machine.

Create the required **sd1** pseudo-device file:

```
# cd /dev && sh MAKEDEV sd1:key_enter:
```

Clear the first megabyte of **sd1** for storing MBR data later:

```
# dd if=/dev/zero of=/dev/rsd1c bs=1m count=1:key_enter:
```

Exit the OpenBSD shell and return to the welcome prompt:

```
# exit:key_enter:
```

Install OpenBSD Amd64 7.0 in the Encrypted Partition

Start the installation:

```
Welcome to the OpenBSD/amd64 7.0 installation program.  
(I)nstall, (U)pgrade, (A)utoinstall or (S)hell? i:key_enter:
```

Go through the installation program as follows to install OpenBSD amd64 7.0 with full disk encryption for production. Main points you should know during the installation:

1. Replace all example networking configurations with your own.

2. For security purposes, set strong passwords for the root user and the newly created user, and do not allow root to log in from SSH.
3. The X Window System and games are unnecessary for building a production website. De-select those sets to save disk space.
4. Be sure to input **sd1** as the root disk.
5. Because the matching SHA256 signature is not uploaded, input **yes** to continue the installation.

```
Choose your keyboard layout ('?' or 'L' for list) [default] :key_enter:

System hostname? (short form, e.g. 'foo') openbsd:key_enter:

Available network interfaces are vio0 vlan0.
Which network interface do you wish to configure? (or 'done') [vio0] :key_enter:
IPv4 address for vio0? (or 'autoconf' or 'none') [autoconf]
203.0.113.100:key_enter:
Netmask for vio0? [255.255.255.0] 255.255.254.0:key_enter:
IPv6 address for vio0? (or 'autoconf' or 'none') [none]:key_enter:
Available network interfaces are: vio0 vlan0.
Which network interface do you wish to configure? (or 'done') [done]:key_enter:
Default Ipv4 route? (IPv4 address or none) 203.0.113.1:key_enter:
add net default: gateway 203.0.113.1
DNS domain name? (e.g. 'example.com') [my.domain] example.com:key_enter:
DNS nameservers? (IP address list or 'none') [none] 108.61.10.10:key_enter:

Password for root account? (will not echo) YourOwnRootPassword:key_enter:
Password for root account? (again) YourOwnRootPassword:key_enter:
Start sshd(8) by default? [yes] :key_enter:
Do you expect to run the X Window System? [yes] no:key_enter:
Setup a user? (enter a lower-case loginname, or 'no') [no] johndoe:key_enter:
Full name for user johndoe [johndoe] John Doe:key_enter:
Password for user johndoe? (will not echo) YourOwnUserPassword:key_enter:
Password for user johndoe? (again) YourOwnUserPassword:key_enter:
WARNING: root is targeted by password guessing attacks, pubkeys are safer.
Allow root ssh login? (yes, no, prohibit-password) [no] :key_enter:
What timezone are you in? ('?' for list) [US/Pacific] :key_enter:

Available disks are: sd0 sd1.
Which disk is the root disk? ('?' for details) [sd0] sd1:key_enter:
No valid MBR or GPT.
```

```
Use (W)hole disk MBR, whole disk (G)PT or (E)dit? [whole] :key_enter:
...
Use (A)uto layout, (E)dit auto layout, or create (C)ustom layout?
[a] :key_enter:
...
Available disks are sd0.
Which disk do you wish to initialize? (or 'done') [done] :key_enter:
...

Let's install the sets!
Location of sets? (cd0 disk http nfs or 'done') [cd0] :key_enter:
Pathname to the sets? (or 'done') [7.0/amd64] :key_enter:

Select sets by entering a set name, a file name pattern or 'all'. De-select sets
by
prepending a '-', e.g.: '-game*'. Selected sets are labelled '[X]'.
...
Set name(s)? (or 'abort' or 'done') [done] -game* -x*:key_enter:
Set name(s)? (or 'abort' or 'done') [done] :key_enter:
Directory does not contain SHA256.sig. Continue without verification? [no]
yes:key_enter:
...
Location of sets? (cd0 disk http nfs or 'done') [done] :key_enter:
...
CONGRATULATIONS! Your OpenBSD install has been successfully completed!
...
Exit to (S)hell, (H)alt or (R)eboot? [reboot] :key_enter:
```

Having OpenBSD 7.0 installed, you need to make the system boot from the hard disk other than the OpenBSD amd64 7.0 ISO image. To do so, Return to the **Server Details** page of your machine in the Vultr Control Panel, click the **Remove ISO** button in the **Custom ISO** column in the **Settings** tab.

4. Next Steps

1. Open a new noVNC console window.
2. Input the passphrase you set up earlier to start the fully encrypted system.
3. Log in as **root** from the same noVNC console window.
4. Grant **doas** permissions to the newly created **john** user:

```
# user mod -G wheel johndoe:key_enter:  
# echo "permit persist :wheel" >> /etc/doas.conf:key_enter:
```

5. Patch and then reboot the system:

```
# syspatch:key_enter:  
# shutdown -r now:key_enter:
```

After the system gets up and running again, input your passphrase to start the system.

6. On your desktop machine, create a public/private RSA key pair and then upload the public key to your OpenBSD server instance:

```
$ ssh-keygen -t rsa -b 4096 -f ~/.ssh/johndoe.openbsd.key -C "John Doe on  
OpenBSD":key_enter:  
$ ssh-copy-id -i ~/.ssh/johndoe.openbsd.key.pub johndoe@203.0.113.100:key_enter:
```

7. Login in to your OpenBSD as **johndoe** from an SSH console, and then disable password authentication as follows:

```
$ ssh johndoe@203.0.113.100:key_enter:  
$ doas sed -i 's/#PasswordAuthentication yes/PasswordAuthentication no/g' /etc/  
ssh/sshd_config:key_enter:  
$ doas /etc/rc.d/sshd restart:key_enter:
```

8. From now on, log in from SSH with your private key:

```
$ ssh -i ~/.ssh/johndoe.openbsd.key johndoe@203.0.113.100:key_enter:
```

More Information

Learn more about OpenBSD by visiting the following pages:

- Detailed explanations to [doas](#).
- [OpenBSD FAQs](#)
- [OpenBSD Manual](#)



VULTR

