

Ultimate Linux Hardening Bootcamp

Authors:

Omar Santos and Joseph Mlodzianowski

<https://hackinglinux.org>

LAB GUIDE

LAB GUIDE

Install RHEL

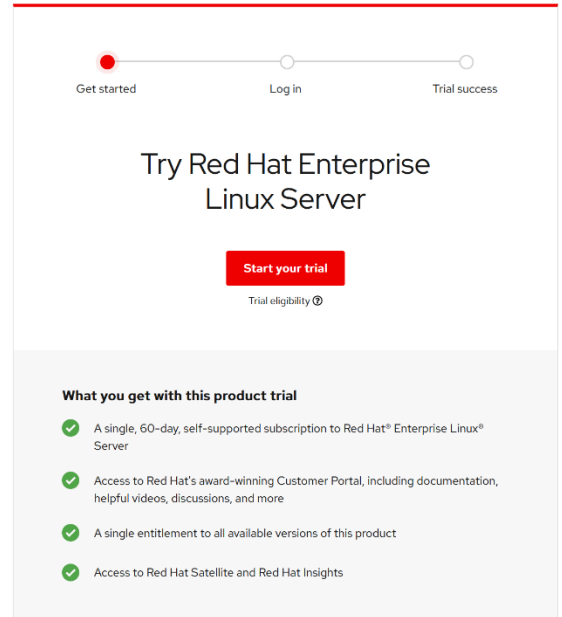
Lab 1

- Create Account/Login RHEL
- Locate, identify and Download RHEL 8.8
- Install RHEL 8.8 as a Virtual Machine

1. Create an account if you already don't have one register for one.

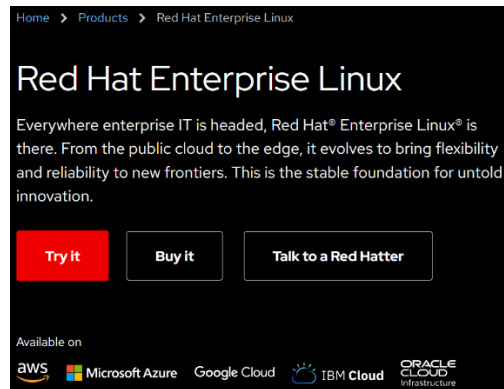


- <https://www.redhat.com>
- Select Login
- Register for 60-day "Trial"



2. After registration and login, clicking start trial, will automatically download RHEL-Baseos-9, however we want to be able to download and install rhel-8.8 so clicking on the link below will take you to the appropriate location provided you are logged in.

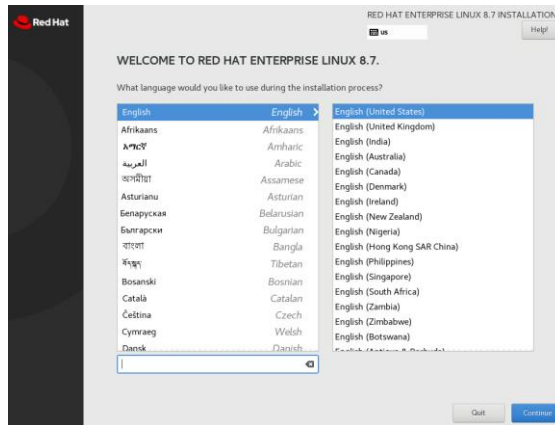
- [Red Hat Enterprise Linux for x86_64 8 - Red Hat Customer Portal](#) RHEL 8.8 dvd iso



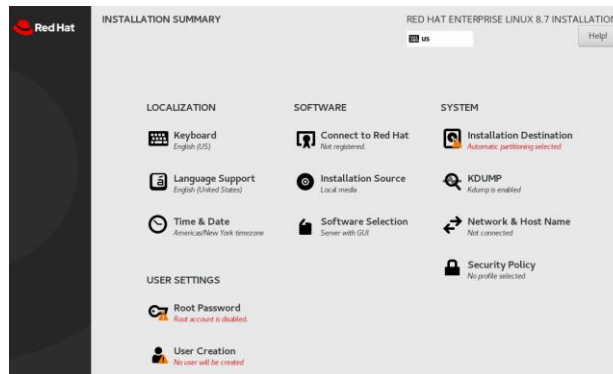
To read more about what's new and different about 8.8 from 8.5

3. Install RHEL 8.8 follow your normal process or for the beginner you can follow below:

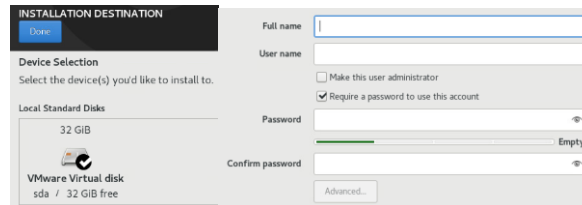
- A. Assuming you are using VMWare use your company's process to install/configure the base RHEL 8.7 system.



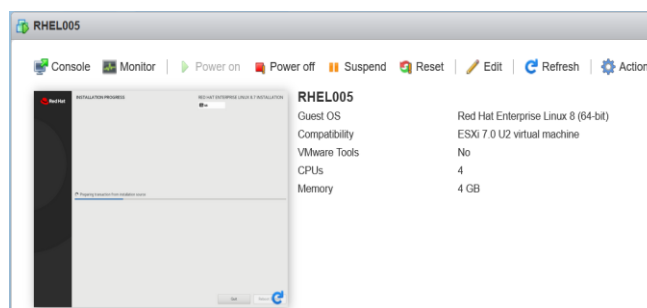
Select your language, here I am selecting English (US) and select continue to the next steps.

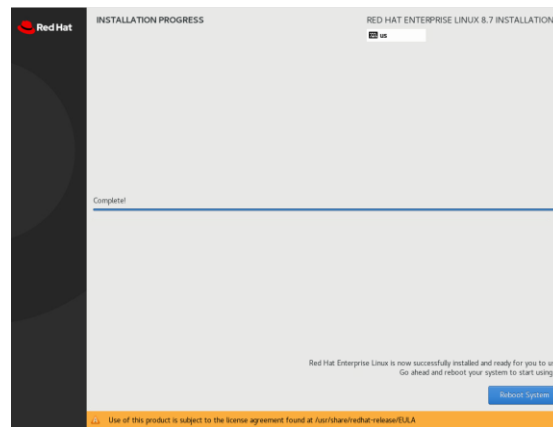
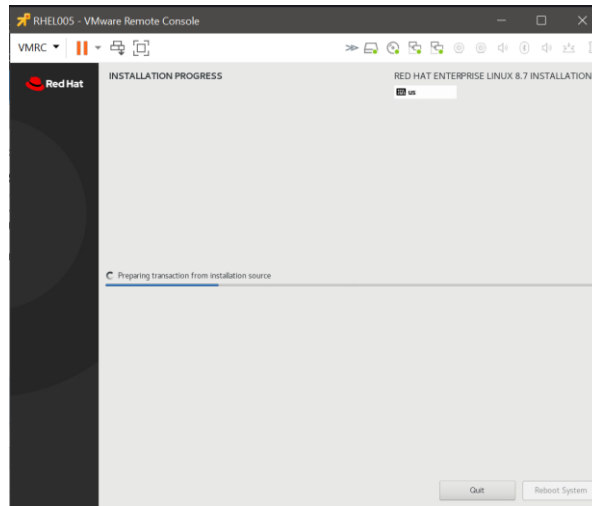


First select the System – Installation Destination to make sure your disc/storage is properly assigned, you notice things in red require our attention. We select our disc and make sure it has a check mark next to the storage, you can see I have configure 32 GiB (SDA) here we also have options** to select additional disks, change partition (custom) or select Automatic, we can also select to encrypt your data on the disk with a password required at startup. We can select the full disk summary and bootloader that will allow is to remove or set boot device. next we select Done, we will see the installer verify that its available for use, and show (Automatic Partitioning Selected)

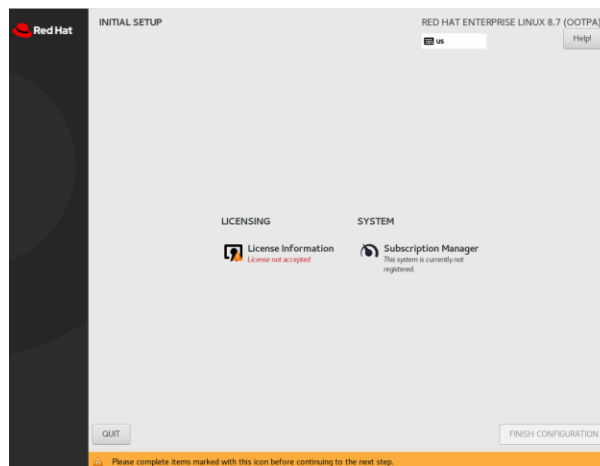


Next section is to create our local user that is not root. Selecting advanced allows you to change the home directory, select user and group id's, group membership, etc, generally you can leave those settings as is. Select done once you enter your Full/User name. Next Select Root Password and set the password. Next select [Begin Installation](#)

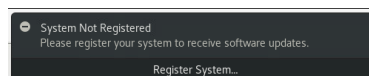




When installation is completed, reboot and notice the notification about the requirement to obtain a license to use rhel.



Licensing requires you accept the elua, so enter into that section and check the I accept box, and finish configuration, then you will be able to login with your user. You might see the “system not registered” lets put that aside for now.



After you select the language and keyboard again, select next and you will be prompted with three videos that will show you how to do certain tasks, launch applications, switch tasks, and use windows and workspaces. You can watch these videos and then close the “getting started”. Since rhel 8 uses gnome it might be familiar to you, check out your environment.

Lab 2

Harden RHEL to the CIS Benchmark

- Enable the Ansible Repositories for RHEL
- Install Ansible and the SCAP Security Guide
- Locate the Relevant Ansible Playbooks
- Harden the System using a playbook

check version we are running

```
$ cat /etc/os-release
```

```
Return: 8.5xxxx
```

Next lets enable the subscription service for this version of RHEL

```
$ sudo subscription-manager repos --enable ansible-2.9-for-rhel-8-x86_64-rpms
```

Specifically ansible 2.9 , we do sudo to obtain root privileges to be able to install. This takes some time because it goes out and checks to make sure I have a valid subscription. **

Return: Repository 'ansible-2.9-for-rhel-8-x86-rpms' is enabled for this system.

Now let's install ansible and scap security (scap "Security Content Automation Protocol") allows us to scan our system for vulnerabilities and configuration compliance. Includes ansible playbooks for many compliance baselines including CIS which we will be using in this exercise. which we will be using in this exercise.

Next let's install:

```
$ sudo dnf install -y ansible scap-security-guide
```

We should see:

```
updating subscription management repositories....
```

```
Red Hat Ansible Engine 2.9 for RHEL 8 x86_64 (RMPs)
```

```
Last metadata expiration check: 0:00:01 ago on October Fri 21, 2022 14:21:10 PST.
```

Should be installing three packages, not including SCAP which I already had installed.

So now everything is installed, lets switch to where the ansible playbook provided by the scap security guide package live: which is -> /usr/share/scap-security-guide/ansible/

So we will enter:

```
$ cd /usr/share/scap-security-guide/ansible/
```

```
:ansible $ we will check the directory with ls – we can see all kinds of security profile playbooks, we will focus on the CIS Playbooks
```

```
$ ls -la rhel8-playbook-cis*
```

Which should result in 4 playbooks

That cover both level 1 compliance for workstations, (l1) and servers. The level 2 playbooks are for servers. (if you cat the last file you will see it is a CIS level 2 playbook for servers)

So to begin, we will start with a level 1 baseline playbook which is less stringent, it will take less time to run with take a look at how to run this ansible, we will escalate our privileges with sudo, and run the playbook against the local host. We will make sure we select the correct yml file.

```
$sudo ansible-playbook -i "localhost," -c local rhel8-playbook-cis_server_l1.yml
```

Results: PLAY [all] ***** TASK [Gathering Facts] *****

This will take up to 10 minutes, it will make a large number of checks, and will change system configuration settings and parameters according to the CIS level 1 benchmark.

Results: PLAY RECAP *****

Now we can see the playbook is completed, let's go down to the PLAY RECAP bottom , we can see hundreds of checks were made by the ansible playbook , and the system had 80 configuration items change to harden it according to the CIS Level 1 for Servers. Changes to services, files and permissions and much more.

Now we can see our system it compliant with the CIS Level 1 benchmark.

Summary. – We installed ansible, we downloaded benchmarks, we ran our level 1 CIS benchmark, and is compliance and supported with Red Hat.

Reboot when completed.

Commands:

```
Sudo subscription-manager repos --enable \
```

```
Ansible-2.9-for-rhel-8-x86_64-rpms
```

```
Sudo dnf install -y ansible scap-security-guide
```

```
Cd /usr/share/scap-security-guide/ansible/
```

```
Sudo ansible-playbook -i "localhost," -c local \
```

```
Rhel8-playbook-cis_server_l1.yml
```

Lab 3

RHEL 8.8 SELinux

SELinux

Red Hat based on Discretionary Access Control (DAC) mechanism.

SELinux support three major states that we discussed earlier, it can be in the following states:

- Disabled
- Permissive
- Enforcing

The states are set in the /etc/selinux/config file

Check it on your system:

```
[adminx@localhost ~]$ egrep ^SELINUX= /etc/selinux/config
SELINUX=enforcing
[adminx@localhost ~]$
```

Execute `egrep ^SELINUX /etc/selinux/config`

To get more details you can use the -> "SEStatus" `sestatus` command

```
[adminx@localhost ~]$ sestatus
SELinux status:                enabled
SELinuxfs mount:              /sys/fs/selinux
SELinux root directory:       /etc/selinux
Loaded policy name:           targeted
Current mode:                 enforcing
Mode from config file:       enforcing
Policy MLS status:           enabled
Policy deny_unknown status:  allowed
Memory protection checking:  actual (secure)
Max kernel policy version:    33
```

You can toggle the SELinux state between "Permissive 0" and "Enforcing 1" without a reboot.

Using `setenforce` you can change SELinux state.

```
[adminx@localhost ~]$ getenforce
Enforcing
[adminx@localhost ~]$ sudo setenforce 0
[adminx@localhost ~]$ getenforce
Permissive
```

Or you can change the value in the `/etc/fs/selinux/enforce` file by echo'ing the mode

```
echo 0 > /sys/fs/selinux/enforce
```

Lab 4

Enabling Automatic updates for RHEL 8.8

1. `sudo dnf search dnf-automatic` [search packages that might be available]
2. `sudo dnf info dnf-automatic`
3. `sudo yum install dnf-automatic` [install package using dnf]
4. `sudo view /etc/dnf/automatic.conf` [change **apply_updates** from no to yes]
5. `sudo systemctl enable --now dnf-automatic.timer` [enable and run dnf-automatic]

To receive an email once a download/update has been completed,

- A. `sudo view /etc/dnf/automatic.conf` [fix "emit_via = email" and configure your email_to = joseph@mydomain.com

```

From: baseater@domain
Subject: Updates applied on 'rhel8-01'.
To: seater@domain

The following updates have been applied on 'rhel8-01':
=====
Package           Arch  Version              Repository              Size
=====
Upgrading:
bind-export-libs  x86_64 32:9.11.22-5.el8_3.1
                                     rhel-8-for-x86_64-baseos-rpms 1.1 M
grub2-common      noarch 1:2.02-90.el8_3.1  rhel-8-for-x86_64-baseos-rpms 885 k
grub2-efi-x64     x86_64 1:2.02-90.el8_3.1  rhel-8-for-x86_64-baseos-rpms 409 k
grub2-tools       x86_64 1:2.02-90.el8_3.1  rhel-8-for-x86_64-baseos-rpms 2.0 M
grub2-tools-extra x86_64 1:2.02-90.el8_3.1  rhel-8-for-x86_64-baseos-rpms 1.1 M
grub2-tools-minimal
x86_64 1:2.02-90.el8_3.1 rhel-8-for-x86_64-baseos-rpms 206 k
qemu-guest-agent x86_64 15:4.2.0-34.module+el8.3.0+9903+ca3e42fb.4
                                     rhel-8-for-x86_64-appstream-rpms
                                     232 k

Installing dependencies:
grub2-tools-efi  x86_64 1:2.02-90.el8_3.1  rhel-8-for-x86_64-baseos-rpms 471 k

Transaction Summary
=====
Install  1 Package
Upgrade  7 Packages

```

RHEL8	Cloud Access AWS
--------------	-------------------------

Lab 5

- Access the Red Hat Hybrid Cloud Console
 - *console.redhat.com*
- Provide our AWS Credentials
 - *Aws username & password*
- Select which Services we want to have to access as part of cloud access
 - *Select which images and services you want access to in AWS.*
- Check for “Gold Images” in AWS so we can BYO subscriptions
 - *Locate and utilize RHEL Gold Images in AWS*

First: Let’s browse to and login to console.redhat.com, we should be at the hybrid cloud console. We will scroll down to the bottom and select **configure** section (connect a new source) click **Connect to Sources->** this will allow us to connect our Red Hat account to a number of public provider cloud accounts.

We will select the blue button, “Add Source” we will see a number of options for public clouds, including AWS, Google Cloud and Microsoft Azure.

Select source type

To import data for an application, you need to connect to a data source. Start by selecting your source type.

Select a cloud provider *



We will select AWS, and next and be promoted for “Add a Cloud Source” and you will see “Name Source” now you will provide a name that is relevant to your project, name *

After you will see validating to make sure there isn’t a conflict with something you already have setup...

The screenshot shows a progress indicator on the left with three steps: 1. Select source type, 2. Name source (highlighted), and 3. Select configuration. The main content area is titled "Name source" and contains the instruction "Enter a name for your Amazon Web Services source." Below this is a text input field with the label "Name *" and the placeholder text "source_1".

Click next

Now you are “Select Configuration” and Select the radio button “Account Authorization” (Recommended) This will fully automate my connection between my two accounts.

You will be prompted for the Access Key ID* and the Secret Access Key * make sure you obtain those from your vault (private place you store secrets) and enter them in the appropriate fields.

The screenshot shows a progress indicator on the left with five steps: 1. Select source type, 2. Name source, 3. Select configuration (highlighted), 4. Select applications, and 5. Review details. The main content area is titled "Select configuration" and contains the instruction "Configure your source manually or let us manage all necessary credentials by selecting account authorization configuration." Below this is a section titled "Select a configuration mode *" with two radio buttons: "Account authorization" (checked and labeled "Recommended") and "Manual configuration". The "Account authorization" option has a sub-description: "A new automated source configuration method. Provide your AWS account credentials and let Red Hat configure and manage your source for you." Below this are two text input fields: "Access key ID *" with the placeholder "AKIAIOSFODNN7EXAMPLE" and "Secret access key *" with the placeholder "wJairXUtnFEMU/K7MDENG/bPxRfiCYEXAMPLEKEY". The "Manual configuration" option has a sub-description: "Configure and manage your source manually if you do not wish to provide account authorization credentials. You will set up sources the same way you do today."

Click Next

Next we will be prompted for “Select Applications” We will leave “Available Applications:”

The screenshot shows a progress indicator on the left with five steps: 1. Select source type, 2. Name source, 3. Select configuration, 4. Select applications (highlighted), and 5. Review details. The main content area is titled "Select applications" and contains the instruction "Configuring your cloud sources provides additional capabilities included with your subscription. You can turn these features on or off at any time after source creation." Below this is a section titled "Available applications" with three items: "Cost Management" (enabled), "RHEL management" (enabled and labeled "Bundle"), and "Red Hat gold images" (enabled). Each item has a sub-description: "Cost Management: Analyze, forecast, and optimize your Red Hat OpenShift cluster costs in hybrid cloud environments."; "RHEL management: Unlock cloud images in AWS and bring your own subscription instead of paying hourly."; "Red Hat gold images: View precise public cloud usage data in subscription watch."; "Autoregistration: Cloud instances automatically connect to console.redhat.com when provisioned."

Here we get to decided which applications we want to enable as part of cloud access, we leave cost management enabled as that will help right size our virtual machines in AWS. And we will leave RHEL management enabled, Red Hat Gold images – allows us to unlock red hat gold images, the BYO subscription. Allows us to auto register as part of the build process.

Click Next

Review data details which all looks good

Click Add.

This now validate credentials and create the link between red hat and amazon.

Validating credentials

This might take some time. You'll receive a notification if you are still in the Sources application when the process completes. Otherwise, you can check the status in the main sources table at any time.

In the meantime, you can close this window while the validation process continues.

Close

This will take some time, and as the configuration is in process, we will go back to our sources page



Configuration in progress

We are still working to confirm credentials and app settings. To track progress, check the Status column in the Sources table.

Go back to Sources

Add another source

On return to the Source page we can see the status of our (in process) changes to (available)

The screenshot shows the 'Sources' page in the AWS console. At the top, there are tabs for 'Cloud sources' and 'Red Hat sources'. Below this, there are three informational cards: 'Use gold images', 'Explore Red Hat Insights', and 'Track usage with Subscriptions'. The main part of the page is a table with the following columns: Name, Type, Connected applications, Date added, and Status. There is one entry in the table:

Name	Type	Connected applications	Date added	Status
RHEL01_AWS_Open	Amazon Web Services	Cost Management, RHEL management	1 minute ago	Available

Our AWS cloud configuration is available, it then gives us access to the cost management services and the RHEL management services with in aws, including allowing us to bring our own subscriptions into the public cloud.

Switch to AWS EC2 Dashboard:

Now in AWS we can see what it looks like from there, so let's go to AWS console. So next let's locate at the gold images that have been unlocked for us as part of the cloud access:

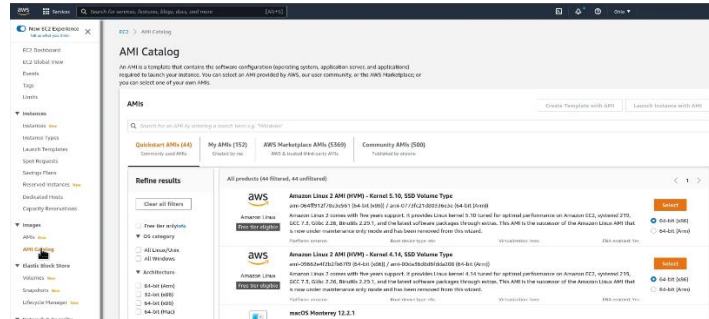
Click AMI Catalog on the left.

The screenshot shows the AWS EC2 Dashboard. On the left, there is a navigation menu with 'AMI Catalog' selected. The main content area is divided into three sections: 'Resources', 'Launch instance', and 'Service health'. The 'Resources' section shows a summary of EC2 resources in the US East (Ohio) Region:

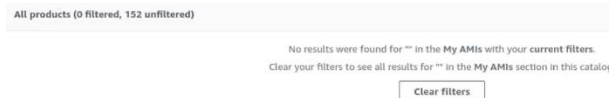
Resource	Count
Instances (running)	0
Dedicated Hosts	0
Elastic IPs	0
Instances	0
Key pairs	2
Load balancers	0
Placement groups	0
Security groups	3
Snapshots	0
Volumes	0

The 'Service health' section shows the status of the AWS service in the US East (Ohio) Region, which is 'This service is operating normally'.

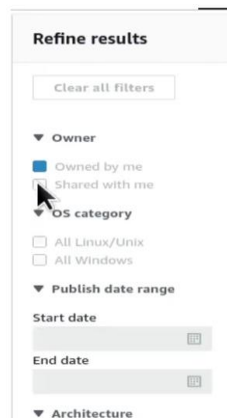
It will show us all AMI available to us whether provided by AWS, Community, Market Place, shared or owned by myself.



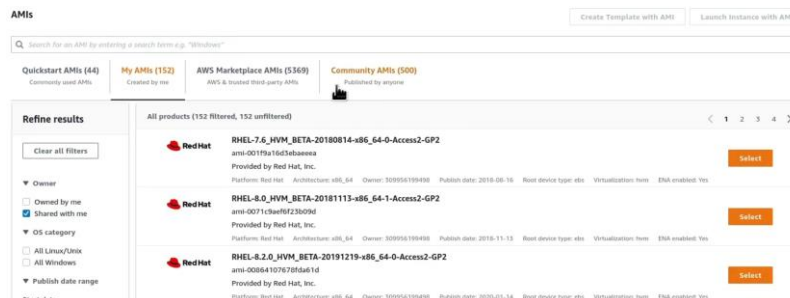
So next we will want to locate the gold image from Red Hat, click on “My AMI’s (152)” we will likely have no results found:



So what will need to do is filter – so we will change to the filter to be AMI that were shared with me, by selecting the boxes on the left under “Refine Results” which provides us with plenty of options to narrow down to the specific image we are looking for:



We will select “Shared with Me” when I click on that I can see a number of Red Hat Enterprise Linux images, the key item to remember is what they share in common is these images are provided by Red Hat and not AWS. This is the easiest way to identify a particular RHEL Image is a Gold image provided as part of cloud access.



Now you can select the image you want to setup a EC2, and provide licensing (subscription)

Summary:

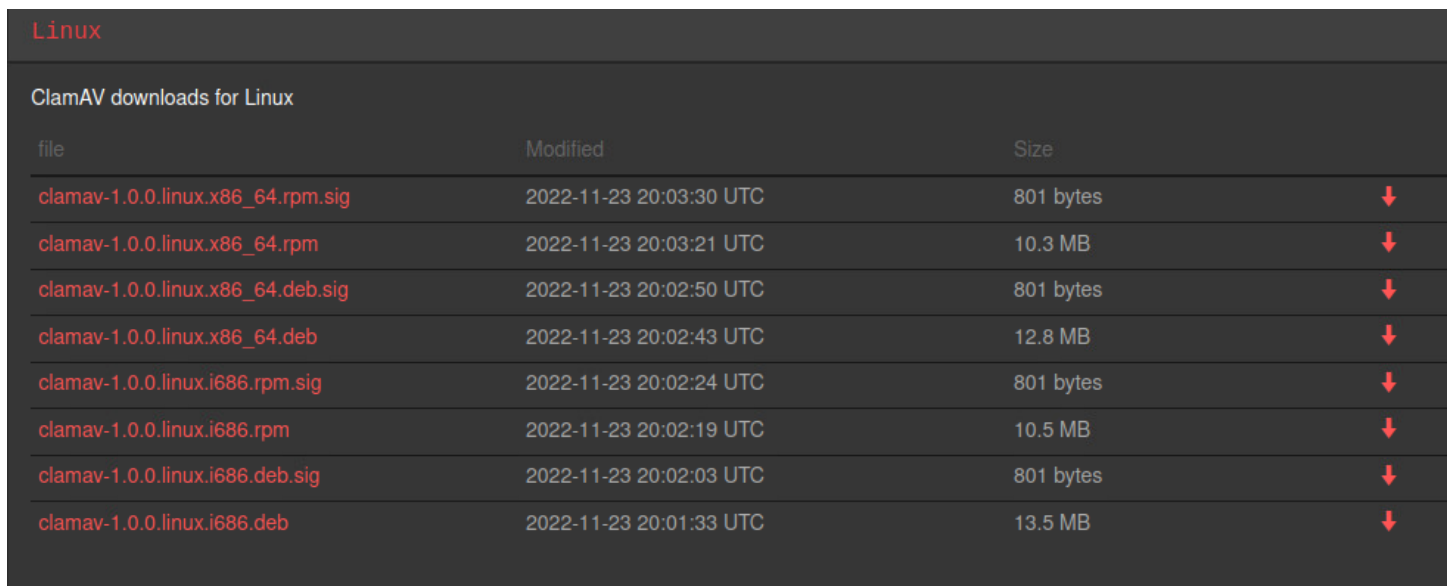
Disable Root Login:

To disable root login, open the `/etc/ssh/sshd_config` file and set `PermitRootLogin` to `no`. Then restart the SSH service to apply the change. Additionally, you should also consider changing the default port used by SSH to further increase security

Lab 7

ClamAV Linux

<https://www.clamav.net/download.html>

A screenshot of a web page showing ClamAV downloads for Linux. The page has a dark background with light text. At the top, it says "Linux" in red. Below that, it says "ClamAV downloads for Linux". There is a table with three columns: "file", "Modified", and "Size". Each row represents a different download package, and each row ends with a red downward arrow icon. The packages listed are: clamav-1.0.0.linux.x86_64.rpm.sig (801 bytes), clamav-1.0.0.linux.x86_64.rpm (10.3 MB), clamav-1.0.0.linux.x86_64.deb.sig (801 bytes), clamav-1.0.0.linux.x86_64.deb (12.8 MB), clamav-1.0.0.linux.i686.rpm.sig (801 bytes), clamav-1.0.0.linux.i686.rpm (10.5 MB), clamav-1.0.0.linux.i686.deb.sig (801 bytes), and clamav-1.0.0.linux.i686.deb (13.5 MB).

file	Modified	Size	
clamav-1.0.0.linux.x86_64.rpm.sig	2022-11-23 20:03:30 UTC	801 bytes	↓
clamav-1.0.0.linux.x86_64.rpm	2022-11-23 20:03:21 UTC	10.3 MB	↓
clamav-1.0.0.linux.x86_64.deb.sig	2022-11-23 20:02:50 UTC	801 bytes	↓
clamav-1.0.0.linux.x86_64.deb	2022-11-23 20:02:43 UTC	12.8 MB	↓
clamav-1.0.0.linux.i686.rpm.sig	2022-11-23 20:02:24 UTC	801 bytes	↓
clamav-1.0.0.linux.i686.rpm	2022-11-23 20:02:19 UTC	10.5 MB	↓
clamav-1.0.0.linux.i686.deb.sig	2022-11-23 20:02:03 UTC	801 bytes	↓
clamav-1.0.0.linux.i686.deb	2022-11-23 20:01:33 UTC	13.5 MB	↓

Yum install epel-release -y

```
yum -y install clamav-server clamav-data clamav-update clamav-filesystem clamav clamav-scanner-systemd clamav-devel clamav-lib clamav-server-systemd
```

freshclam -d

```
systemctl start freshclam.service
```

```
systemctl enable freshclam.service
```

```
systemctl status freshclam.service
```

```
clamscan --infected --remove -recursive /var/www/
```

Enable/Configure FIPS 140-2

```
[root@rhel005 ~]# fips-mode-setup --check
Installation of FIPS modules is not completed.
FIPS mode is disabled.
```

To avoid cryptographic key material regeneration and reevaluation of the compliance of the resulting system associated with converting already deployed systems, Red Hat recommends starting the installation in FIPS mode. Add `"fips=1"`

Procedure

1. With the boot menu open, press the `Esc` key on your keyboard.
2. The `boot:` prompt is now accessible.
3. Press the `Tab` key on your keyboard to display the help commands.
4. Press the `Enter` key on your keyboard to start the installation with your options.
To return from the `boot:` prompt to the boot menu, restart the system and boot from the installation media again.

Or if already installed:

```
[root@rhel005 ~]# fips-mode-setup --enable
Kernel initramdisks are being regenerated. This might take some time.
Setting system policy to FIPS
Note: System-wide crypto policies are applied on application start-up.
It is recommended to restart the system for the change of policies
to fully take place.
FIPS mode will be enabled.
Please reboot the system for the setting to take effect.
```

LAB 9

Cockpit/Web Console RHEL 8

Procedure

```
# yum install cockpit
```

Enable and start the `cockpit.socket` service, which runs a web server:

```
# systemctl enable --now cockpit.socket
```

The screenshot displays the Cockpit web console interface for RHEL 8. The browser address bar shows `https://localhost:9090/system`. The interface includes a search bar, a sidebar with navigation options (System, Overview, Logs, Storage, Networking, Podman containers, Accounts, Services, Tools, Applications, Diagnostic reports, Kernel dump, SELinux), and a main content area with sections for Health, Usage, System information, and Configuration.

Health

- Not registered
- Not connected to Insights
- Last successful login: Jan 08, 07:21 PM on tty2
- [View login history](#)

Usage

- CPU: 5% of 4 CPUs
- Memory: 1.9 / 3.6 GiB
- [View metrics and history](#)

System information

Model	VMware, Inc. VMware7,1
Asset tag	VMware-56 4d fe 8c ea ee b3 07-8d fe 93 4e 91 d7 4f 86
Machine ID	668d4a8203244554be2c72799b8dda2e
Uptime	8 minutes

[View hardware details](#)

Configuration

Hostname	rhel005.localdomain edit
System time	Jan 8, 2023, 7:33 PM !
Domain	Join domain
Performance profile	virtual-guest
Crypto policy	FIPS
Secure shell keys	Show fingerprints

If the web console was not installed by default on your installation variant and you are using a custom firewall profile, add the `cockpit` service to `firewalld` to open port 9090 in the firewall:

Procedure

```
# firewall-cmd --add-service=cockpit --permanent
```

```
# firewall-cmd --reload
```

With Firefox browse to: <https://localhost:9090>

FAIL2BAN Lab

Fail2ban can only be used to protect services that require username/password authentication. In this lab we will protect the SSHD Daemon (SSHD) from a brute force attack. From here you can setup and protect almost any service listening port on your server.

```
[adminx@rhel8x ~]$ sudo dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

```
[adminx@rhel8x ~]$ sudo dnf -y install fail2ban
```

The fail2ban configuration files are located in the `/etc/fail2ban/` directory and filters are stored in the `/etc/fail2ban/filter.d/` directory (the filter file for sshd is `/etc/fail2ban/filter.d/sshd.conf`).

The global configuration file for the fail2ban server is `/etc/fail2ban/jail.conf`, however, it is not recommended to modify this file directly, as it will probably be overwritten or improved in case of a package upgrade in the future.

As an alternative, it is recommended to create and add your configurations in a `jail.local` file or separate `.conf` files under the `/etc/fail2ban/jail.d/` directory. Note that configuration parameters set in `jail.local` will override whatever is defined in `jail.conf`.

We will enable fail2ban

```
# sudo systemctl enable fail2ban
```

```
# sudo systemctl start fail2ban
```

configure a few basic things in fail2ban to protect the system without it interfering with itself. Copy the `/etc/fail2ban/jail.conf` file to `/etc/fail2ban/jail.local`. The `jail.local` file is the configuration file of interest for us.

```
$ sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
```

```
$ sudo view /etc/fail2ban/jail.local
```

Look for the setting “ignoreip” and all IP Addresses to this line that will have access without the potential of lockout. By default you should add the loopback address, and all IP Address local to the protected system.

```
ignoreip = 127.0.0.1/8 192.168.1.5 192.168.1.10
```

Adding entire networks takes away the intended protections that fail2ban provides. Make sure you follow the golden rule – keep it simple. Save the file and then restart the fail2ban service.

```
# sudo systemctl restart fail2ban
```

Filtered Services in fail2ban

To setup filtered services, you need to create a corresponding “jail” file under /etc/fail2ban/jail.d directory. For SSHD, create a new file named “sshd.local” and enter the service filtering instructions into it.

```
[sshd]
enabled = true
port = ssh
action = iptables-multiport
logpath = /var/log/secure
maxretry = 3
bantime = 600
```

DEBIAN LABS

LAB 1

Let's update the Kali Linux Kernel –

To see what your running perform the following command as a general user or root.

➤ hostnamectl | grep Kernel

```
root@dw00k0:~# hostnamectl | grep Kernel
Kernel: Linux 5.18.0-kali7-amd64
```

Next lets check our linux distribution level

➤ uname -a

```
mrxrdp@dw00k0:~$ uname -a
Linux dw00k0 5.18.0-kali7-amd64 #1 SMP PREEMPT_DYNAMIC Debian 5.18.16-1kali1 (2022-08-31) x86_64 GNU/Linux
```

Next lets check what is publicly available for our distribution

➤ apt search linux-headers | grep headers


```

mrxrdp@dw00k0:~$ apt search linux-headers | grep headers

WARNING: apt does not have a stable CLI interface. Use with caution in sc
ripts.

linux-headers-6.0.0-kali3-amd64/kali-rolling 6.0.7-1kali1 amd64
linux-headers-6.0.0-kali3-cloud-amd64/kali-rolling 6.0.7-1kali1 amd64
linux-headers-6.0.0-kali3-common/kali-rolling 6.0.7-1kali1 all
linux-headers-6.0.0-kali3-common-rt/kali-rolling 6.0.7-1kali1 all
linux-headers-6.0.0-kali3-rt-amd64/kali-rolling 6.0.7-1kali1 amd64
linux-headers-amd64/kali-rolling 6.0.7-1kali1 amd64
linux-headers-cloud-amd64/kali-rolling 6.0.7-1kali1 amd64
linux-headers-rt-amd64/kali-rolling 6.0.7-1kali1 amd64

```

So we have quite a list, we look for our current match which is linux-headers line 1 linux-headers-6.0.0-kali3-amd64/kali-rolling 6.0.7-1kali1 amd64

**Warning this could bork your system up, be sure you are using a test machine, that you have backups, and snapshots. If you do not have your system up to date, or are missing dependencies its highly likely you will bork your system.

Next, we will install linux-headers based on our

```

sudo apt install linux-headers-$(uname -r | sed 's,[^-]*-[^-]*-,,')

```

➤ sudo apt install linux-headers-\$(uname -r | sed 's,[^-]*-[^-]*-,,')

You should see the below message and another 10 lines, explaining which packages will be installed or updated.

```

The following additional packages will be installed:
linux-compiler-gcc-12-x86 linux-headers-6.0.0-kali3-amd64 linux-headers-6.0.0-kali3-common
linux-image-6.0.0-kali3-amd64 linux-image-amd64 linux-kbuild-6.0 linux-libc-dev

```

After some time you will see progress, and a popup – Newer Kernel Available



```

Setting up linux-image-amd64 (6.0.7-1kali1) ...
Setting up linux-headers-6.0.0-kali3-amd64 (6.0.7-1kali1) ...
Setting up linux-headers-amd64 (6.0.7-1kali1) ...
Scanning processes ...
Scanning processor microcode ...
Scanning linux images ...

```

Then next you reboot

➤ sudo shutdown -r now

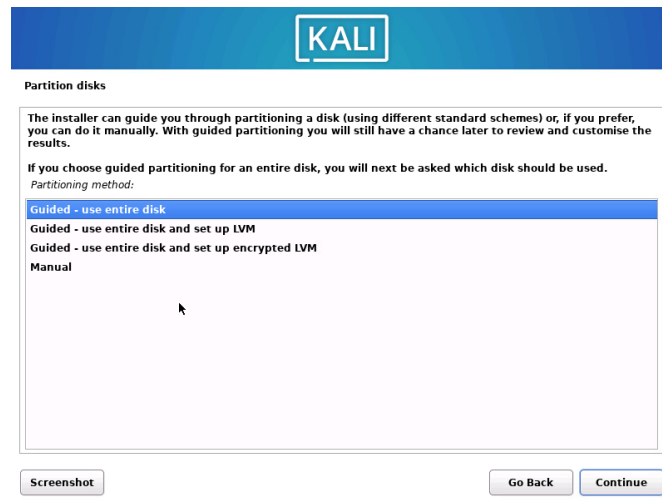
```

mrxrdp@dw00k0:~$ uname -a
Linux dw00k0 6.0.0-kali3-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.0.7-1kali1 (2
022-11-07) x86_64 GNU/Linux

```

And now as you can see your linux-kernel is updated 6.0.7-1kali1

Install LVM (Logical Volume Manager) There is a tool you can use called



You can select Encrypted LVM on installation

If you already have your Debian system setup, you can install LVM by starting with

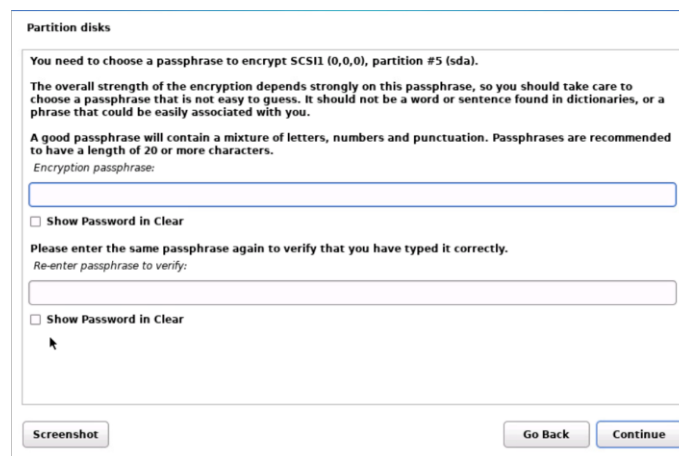
```
➤ apt-get install lvm2
```

To start it

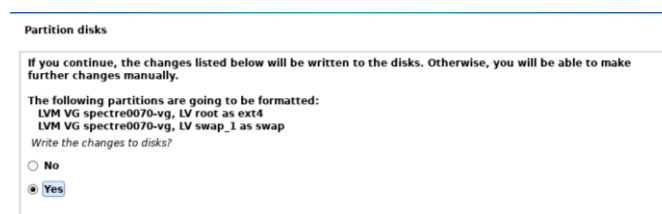
```
➤ test -x '/etc/init.d/lvm2' && /etc/init.d/lvm2 start
```

I found this to be one of the best resources for setting up LVM after the fact, and it makes more sense to provide you a link to Linux Help - <https://www.linuxhelp.com/how-to-manage-lvm-in-debian>

During Kali setup, continue with partitioning the disks as shown below:



Remember the max size allowed on Debian is 16gb, and everytime you boot you will need to enter the password to decrypt the drive (on boot)





`vgdisplay` will provide you details about your volume group and VG size, and status.

`lvdisplay` will provide you detailed output of the logical volumes.

`lsblk` will display your block devices, it reads the filesystem and udev db to gather the information

`blkid /dev/vg1/*` Is used to obtain the UUID Universally Unique Identifier Value, and also the file system type which is required to use, start and setup persistent mounting.

LAB³

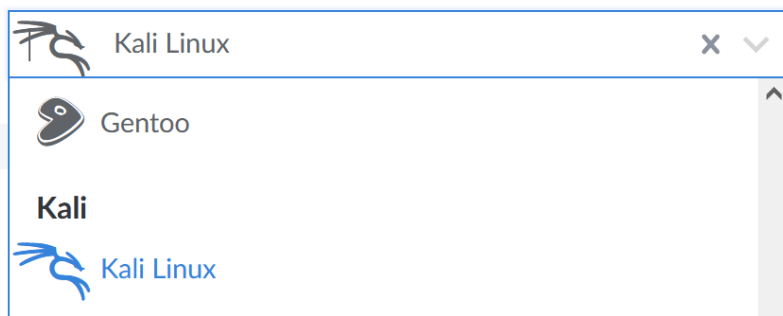
Cloud Linux

Lineode, Digital Ocean and AW have a \$100 credit, so lets use that

Here I will demo Lineode, after we signup, we select create a host.

Choose a Distribution

Images



Lineode has kali as well, so let's select that to build our cloud machine

Next our choices for the machine, our selection lowest level is \$30 a month on dedicated CPU, but since I am just testing this as a lab machine, I select "Shared CPU" and its only \$5.00 a month, and for that I get a fully functional linux host.

Choose a Distribution

Images



Region

You can use our [speedtest page](#) to find the best region for your current location.

Region



Linode Plan

Choosing a Plan

[Dedicated CPU](#) [Shared CPU](#) [High Memory](#) [GPU](#)

Shared CPU instances are good for medium-duty workloads and are a good mix of performance, resources, and price.

	Monthly	Hourly	RAM	CPUs	Storage	Transfer	Network In / Out
Nanode 1 GB	\$5	\$0.0075	1 GB	1	25 GB	1 TB	40 Gbps / 1 Gbps

I selected my region, Dallas which is close to me

There is an option to label the node, by default it will call it "kali-us-central" ie: my timezone. You can also add tags (like what this was build for)

Next enter a complex and secure root password: k\$LAAn8frdvQKLX7

Next option is to use ssh keys. Attach a Vlan, and Add-ons, You can skip for now.

You will see it in the Provision phase for a good 3-5 min,

But then it will turn green and your ready to go.

Select the line that says "SSH Access":

➤ ssh root@ipaddress.x.x

```
mxrxdp@dww00k0:~$ ssh root@[redacted].114.26
The authenticity of host '[redacted].114.26 ([redacted].114.26)' can't be established.
ED25519 key fingerprint is SHA256:r/J7l[redacted]:jTEnfwaf+0EsQcreTslNM.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[redacted].114.26' (ED25519) to the list of known hosts.
root@[redacted].114.26's password:
Linux kali 5.18.0-kali7-amd64 #1 SMP PREEMPT_DYNAMIC Debian 5.18.16-1kali1 (20
(Message from Kali developers)
This is a minimal installation of Kali Linux, you likely
want to install supplementary tools. Learn how:
=> https://www.kali.org/docs/troubleshooting/common-minimum-setup/
(Run: "touch ~/.hushlogin" to hide this message)
(root@kali)-[~]
# touch ~/.hushlogin
```

We enter our password, get a nice message about stopping this banner from printing every time we login so we can create the hushlogin by doing a touch.

- touch ~/.hushlogin

First thing we want to do is bring this distribution UpToDate , and that is run initially manually, but later we will set the system to update auto-magically

- apt-get update // sudo apt update //
- sudo apt dist-upgrade (may take some time) we are talking about 10-15 minutes.

you might see “Configuring libc6:amd” restart services during package upgrades (select yes) you might get kicked off and have to log back in as certain services restarted.

Next we will configure dpkg to automatically update our system to do that we enter

- dpkg-reconfigure --default-priority unattended-upgrades

```
(root@kali)-[~]
└─# dpkg-reconfigure --default-priority unattended-upgrades
```

You will be prompted to “Automatically download and install stable updates” Select yes and continue.

There are other options you can select as well, like telling it to not reload, if this requires specific window of downtime.

- dpkg-reconfigure --help

```
(root@kali)-[~]
└─# dpkg-reconfigure --help
Usage: dpkg-reconfigure [options] packages
-u, --unseen-only           Show only not yet seen questions.
--default-priority         Use default priority instead of low.
--force                    Force reconfiguration of broken packages.
--no-reload                Do not reload templates. (Use with caution.)
-f, --frontend             Specify debconf frontend to use.
-p, --priority             Specify minimum priority question to show.
--terse                    Enable terse mode.
```

LAB⁴

Next we will create some users and limit access, but allow sudo / access (sudo group)

- adduser mrxdp
- enter password
- the rest of the fields can be blank

```

└─# adduser mrxrdp
Adding user `mrxrdp' ...
Adding new group `mrxrdp' (1000) ...
Adding new user `mrxrdp' (1000) with group `mrxrdp (1000)' ...
Creating home directory `/home/mrxrdp' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for mrxrdp
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] Y
Adding new user `mrxrdp' to supplemental / extra groups `users' ...
Adding user `mrxrdp' to group `users' ...

```

Now we have a user named mrxrdp, you can name your user whatever you want.

Next we want to make sure our user can execute the sudo command, that is root equivalent, and only used for escalated/privileged commands.

- `usermod -aG sudo mrxrdp`

```

└─(root@kali)~# usermod -aG sudo mrxrdp

```

Another way of doing this is to edit the sudoers file.

- `sudo view /etc/sudoers` (but for now lets do it view usermod)

```

# User privilege specification
root    ALL=(ALL:ALL) ALL

```

Disable Root Login:

To disable root login, open the `/etc/ssh/sshd_config` file and set `PermitRootLogin` to `no`. Then restart the SSH service to apply the change. Additionally, you should also consider changing the default port used by SSH to further increase security

LAB⁵

Next we move on to using a public/private key (ssh key) so that we wont have to use passwords

First login as your new user account, via ssh and the password you used.

Next create a directory under your user profile, and give yourself rights, this is where we will store the public key for that user.

- `mkdir ~/.ssh && chmod 700 ~/.ssh` (in some cases this may already be created)

Next we will run the ssh key gen tool to create a new key pair, the larger the key the longer it takes to generate it.

➤ `ssh-keygen -b 8096`

```
mrxrdp@dw00k0:~$ ssh-keygen -b 8096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/mrxrdp/.ssh/id_rsa):
```

I do not enter a passphrase, but you can, just make sure you remember it.

Finally, it asks where to store the key, in my case its `/home/mrxrdp/.ssh/` which is acceptable to me.

So lets check out our keys, change to the directory holding your keys, in my case its listed above, yours are likely (will) be different so pay attention to where it stored them and change to that directory.

➤ `cd ~/.ssh`

```
mrxrdp@dw00k0:~$ cd ~/.ssh
mrxrdp@dw00k0:~/.ssh$ ls
id_rsa  id_rsa.pub  known_hosts  known_hosts.old
```

We see two keys “id_rsa” and “id_rsa.pub” should be easy to figure out which one is public – correct ?

We want to upload that public key to our linux server, there are several ways to accomplish this, this first method really simple, make sure your in the directory with the keys and enter:

➤ `ssh-copy-id mrxrdp@ipaddress.x.x`

enter your password and then your done, note says try logging in.... that’s it, your in.

```
mrxrdp@dw00k0:~/.ssh$ ssh-copy-id mrxrdp@.114.26
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/mrxrdp/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already
installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install
the new keys
mrxrdp@.114.26's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'mrxrdp@.114.26'"
and check to make sure that only the key(s) you wanted were added.
```

Now that your in , try doing a sudo command like `sudo apt-get update` and make sure it properly responds.

```
(mrxrdp@kali) - [~]
└─$ sudo apt-get update
sudo: unable to resolve host kali: Name or service not known
[sudo] password for mrxrdp:
Hit:1 http://kali.download/kali kali-rolling InRelease
Reading package lists... Done
```

Next lets lockdown accounts that use passwords to log in. to do this we will edit the `sshd_config` file, you can do this with what ever your favorite edit is, mine is VI.

➤ `Sudo view /etc/ssh/sshd_config`

Scroll down to find “PermitRootLogin yes” and change this to “no” (if your using VI select i) to insert


```
#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

Next scroll down to find “PasswordAuthentication yes” and change this to no

After you make your changes (save) if your using view or vi – enter

- esc : wq! On nano you can use Ctrl x y enter

to make this take effect now you will need to restart the ssh demon, east with the following:

- sudo systemctl restart sshd

Make sure you test it via a new connection before you exit to make sure you didn’t bork something up.

Next we will enable the UFW firewall, Uncomplicated Firewall, making it easy to add/remove rules.

To see what ports are in use we will use

- sudo ss -tupln

```
(mrxdp@kali)-[~]
└─$ sudo ss -tupln
sudo: unable to resolve host kali: Name or service not known
Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port Process
udp UNCONN 0 0 127.0.0.1:323 0.0.0.0:* users:(("chronyd",pid=14261,fd=5))
udp UNCONN 0 0 [::]:323 [::]:* users:(("chronyd",pid=14261,fd=6))
tcp LISTEN 0 128 0.0.0.0:22 0.0.0.0:* users:(("sshd",pid=14768,fd=3))
tcp LISTEN 0 128 [::]:22 [::]:* users:(("sshd",pid=14768,fd=4))
```

Here we can see our ssh sessions on tcp (port 22)

- sudo apt-get install ufw

after its installed it still is not running, we can check that by doing

- sudo ufw status (should say “inactive”) confirming its installed but not running

Creating rules is pretty straight forward – we use the “allow” or the “deny” commands to allow or deny certain traffic through the firewall, its best to start with the allow, and make sure you accounted for all the ports (operating, like DNS, RDP, SSH, etc) otherwise you can break something including access to your system.

Lets create a rule to allow SSH access inbound to our host

- sudo ufw allow ssh (or 22) or if you have a custom port select that port

so now lets enable the firewall

- sudo ufw enable (you will be prompted, are you sure ?)

```
(mrxdp@kali)-[~]
└─$ sudo ufw enable
sudo: unable to resolve host kali: Name or service not known
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
```

If you are running an ftp or apache or other webserver, you might want to enable traffic inbound, so you would enter the following command:

- `sudo ufw allow 443/tcp` (you can also allow port 80/tcp)

Another best practice would be to disable icmp (pings) note that this could break certain applications so make sure you test this thoroughly - to do this you will need to edit the `ufw/before.rules`

- `sudo view /etc/before.rules`

scroll down to `# ok icmp codes for INPUT` and add a line to the beginning of the section, its echo-request drop, and remember this line is case sensitive.

- `-A ufw-before-input -p icmp --icmp-type echo-request -j DROP`

```
# ok icmp codes for INPUT
-A ufw-before-input -p icmp --echo-request -j DROP
-A ufw-before-input -p icmp --icmp-type destination-unreachable -j ACCEPT
```

You may need to stop and start the firewall to make sure this rule takes effect, since its pre-processed it loads before everything. You may even have to reboot the host.

LAB⁶

Password Audit'

To audit passwords in linux (unix) you must compare the on-system encrypted passwords with a known set of passwords – there are a number of wordlists you can use, the number of lists use will dictate how long your process will run, a good rule of them is 10k passwords an-hour. The tradition location of unix passwords was `/etc/passwd`

Common format

Username:

Encrypted Unix password:

User number:

Group number:

Home Director:

Shell:

Tool: Salt Scanner

LAB⁷

Installing clamav

<https://www.clamav.net/download.html>

Linux			
ClamAV downloads for Linux			
file	Modified	Size	
clamav-1.0.0.linux.x86_64.rpm.sig	2022-11-23 20:03:30 UTC	801 bytes	↓
clamav-1.0.0.linux.x86_64.rpm	2022-11-23 20:03:21 UTC	10.3 MB	↓
clamav-1.0.0.linux.x86_64.deb.sig	2022-11-23 20:02:50 UTC	801 bytes	↓
clamav-1.0.0.linux.x86_64.deb	2022-11-23 20:02:43 UTC	12.8 MB	↓
clamav-1.0.0.linux.i686.rpm.sig	2022-11-23 20:02:24 UTC	801 bytes	↓
clamav-1.0.0.linux.i686.rpm	2022-11-23 20:02:19 UTC	10.5 MB	↓
clamav-1.0.0.linux.i686.deb.sig	2022-11-23 20:02:03 UTC	801 bytes	↓
clamav-1.0.0.linux.i686.deb	2022-11-23 20:01:33 UTC	13.5 MB	↓

Step1 - Download .deb for appropriate architecture –

1. `Dpkg -I filename.deb`
or
2. `apt install clamav`
3. `apt install clamav-daemon`

Systemctl status clamav-freshclam

Setup Scanning:

1. Open the clamD configuration file
 - a. View `/etc/clamav/clamd.conf`
2. Add the following lines:
 - a. `ScanOnAccess yes`
 - b. `OnAccessIncludePath /home`
 - c. `OnAccessIncludePath /etc`
 - d. `OnAccessIncludePath /var`
 - e. `OnAccessPrevention yes`
3. What do these settings mean:

ScanOnAccess yes: enable On-Access scanning

OnAccessIncludePath: the folders defined here (`/home/etc/var`) are recursively scanned.

OnAccessPrevention yes: blocks viruses, if you use **'no'**, only a notification is made but no action is taken.

4. Changes to ClamAv can only be done when the clamav-daemon service is stopped
 - a. `Systemctl stop clamav-daemon`
5. Specifically on Debian, the ClamAV user has no rights to the `/var/run/` (a symbolic link to the `/run` folder)
 - a. `mkdir /run/clamav/`
 - b. `chmod 777 -R /run/clamav/`

6. Start ClamD first then associate the service
 - a. clamd
 - b. `systemctl start clamav-daemon`
-

LAB 8 – PAM on Debian

Warning: Messing around with PAM is certainly likely to get you locked out of your system.

Pam documents: <https://github.com/linux-pam/linux-pam>

1. Install the necessary libraries for `pam_cracklib`.

```
sudo apt-get install libpam-cracklib
```

2. Edit `/etc/ssh/sshd_config`:

```
PasswordAuthentication yes
```

3. Enable password authentication and delete `authorized_keys`:

```
# rm /home/ec2-user/.ssh/authorized_keys
```

4. Edit `/etc/pam.d/common-password`.

Change the existing `pam_cracklib` configuration to the following:

```
password requisite pam_cracklib retry=3 minlen=10
```

5. Edit `/etc/pam.d/common-auth`.

Add a `pam_tally` configuration before the default block by adding the following text:

```
auth required pam_tally2.so deny=2 unlock_time=600
```

-
6. You can add several parameters to the module (do `man pam_pwcheck` for complete documentation) for extra rules, such as:
 - `minlen=aNumber`: specifies the minimum length (by default, five characters) for the new password. If you set it to zero, all password lengths are accepted.
 - `cracklib=pathToDictionaries`: allows use of the cracklib library for password checks. If the new password is in a dictionary, a simple brute-force attack quickly will guess it.
 - `tries=aNumber`: sets how many attempts to allow, if previous attempts were rejected because they were too easy.
 - `remember=aNumber`: defines how many previous passwords will be remembered.

LAB 9:

Install and use John the Ripper

Installed size: 77.31 MB

How to install: `sudo apt install john [or] git clone git://github.com/magnumripper/JohnTheRipper -b bleeding-jumbo john`

Dependencies: `sudo apt-get install build-essential libssl-dev yasm libgmp-dev libpcap-dev libnss3-dev libkrb5-dev pkg-config libbz2-dev zlib1g-dev`

- a. `cd ~/src/john/src /&/ ./configure && make -s clean && make -sj4`
- b. `../run/john -test`

USAGE:

`john --wordlist=/usr/share/john/password.lst --rules unshadowed.txt`

Using a wordlist (`--wordlist=/usr/share/john/password.lst`), apply mangling rules (`--rules`) and attempt to crack the password hashes in the given file (`unshadowed.txt`):

```
lit@kali$ john --wordlist=matkweb7-orp10000.txt --format=Raw-md5 md5-passwords.txt
Using default input encoding: UTF-8
Loaded 18765 password hashes with no different salts (Raw-MD5 [MD5 512/512 AVX512BW 16x3])
Remaining 12764 password hashes with no different salts
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:00 DONE (2022-09-26 23:27) 0g/s 124937p/s 124937c/s 1594MC/s gribouille.. starstar
Session completed.
```

Using verbose mode (`-v`), read a list of passwords (`-inp=allwords.txt`) and save only unique words to a file (`uniques.txt`):

There is also a commercial version of John the Ripper “pro” at openwall.com and a cloud version as well.

HYDRA:

```
sudo apt-get install hydra-gtk
sudo apt-get purge hydra-gtk && sudo apt-get autoremove && sudo apt-get autoclean
```

sometimes requires removing and fixing issues prior to new installation:

```
sudo apt-get purge hydra-gtk && sudo apt-get autoremove && sudo apt-get autoclean
sudo apt-get install libssl-dev libssh-dev libidn11-dev libpcre3-dev \ libgtk2.0-dev libmysqlclient-dev libpq-dev libsvn-dev
\ firebird-dev libncp-dev
```

```
git clone https://github.com/vanhauser-thc/thc-hydra.git
cd thc-hydra
./configure
sudo make install
```

```
hydra -help
```

`-l` specifies a username during a brute force attack.

`-L` specifies a username wordlist to be used during a brute force attack.

-p specifies a password during a brute force attack.

-P specifies a password wordlist to use during a brute force attack.

-t set to 4, which sets the number of parallel tasks (threads) to run.

Trying to bruteforce usernames and passwords for SSH

```
hydra -L user.txt -P pass.txt 192.168.29.135 ssh -t 4
```

Bruteforce the password when you have the user:

```
hydra -l msfadmin -P pass.txt 192.168.29.135 ssh -t 4
```

Bruteforce and change the port number

```
hydra -s 22 -L user.txt -P pass.txt 192.168.29.229 ssh -t 5
```

There is a gui - sudo apt-get install hydra-gtk

Xhdra

Ultimate Linux Hardening Bootcamp

Authors:

Omar Santos and Joseph Mlodzianowski