



Module 1 - Operating Systems Linux

Session 7 - Installing Software

Presented by Tim Medin

© SANS, Cyber Aces, Red Siege. All Rights Reserved. Redistribution Prohibited.

YOUR GATEWAY TO CYBERSECURITY SKILLS AND CAREERS

Welcome to Cyber Aces Online Module 1. In this session we will cover the ways to install software on Linux and in CentOS.

Content in this session has been developed by Tom Hessman, Tim Medin, Mark Baggett, Doug Burks, Michael Coppola, Russell Eubanks, Ed Skoudis, and Red Siege.

SANS CYBER ACES ONLINE TUTORIALS

YOUR GATEWAY TO CYBERSECURITY SKILLS AND CAREERS

1. Introduction to Operating Systems

- 01. Linux
- 02. Windows

2. Networking

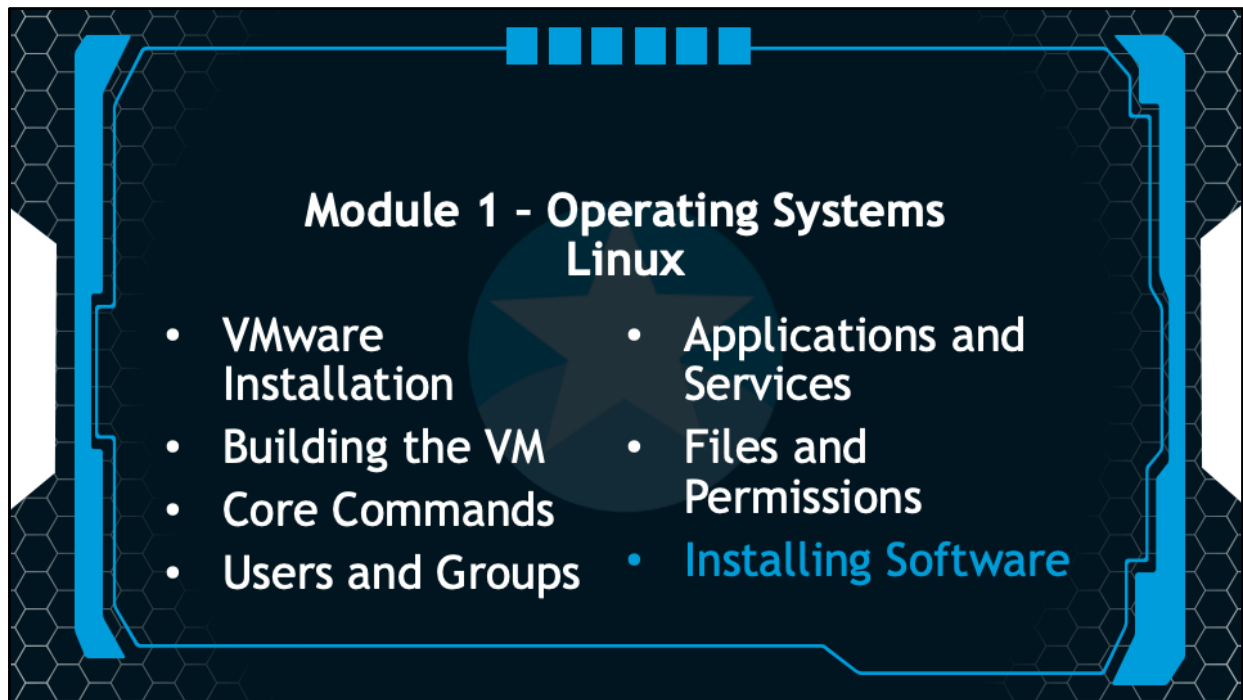
3. System Administration

- 01. Bash
- 02. PowerShell
- 03. Python

This training material was originally developed to help students, teachers, and mentors prepare for the Cyber Aces Online Competition. This module focuses on the basics of what an operating systems is as well as the two predominant OS's, Windows and Linux. This session is part of Module 1, Introduction to Operating Systems. This module is split into two sections, Linux and Windows. In this session, we will continue our examination of Linux.

The three modules of Cyber Aces Online are Operating Systems, Networking, and System Administration.

For more information about the Cyber Aces program, please visit the Cyber Aces website at <https://CyberAces.org/>.



In this session we will discuss different ways to install software on Linux, both from source and from a package.



Installing Software



Before we install software, we must obtain it
We can obtain it from a Linux distribution CD or from the Internet

- Most packages are on the Internet today

Most Linux software comes in two different forms: source or package

- Source has to be compiled
- Packages are generally distro-specific, but install easily

Before we can install software, we must obtain the software itself. We can obtain software from a Linux distribution CD or directly from the Internet. Nowadays, most packages are retrieved from the Internet. Most Linux software downloaded from the Internet comes in two different forms: source or package. Source packages contain source code, and therefore have to be compiled. Packages are generally distro-specific, but they are much easier to install.



Installing Software from Source (1)



Installing from source is the traditional way to install software in the UNIX world

This is typically done with the following commands:

```
$ ./configure
$ make
$ sudo make install
```

- "configure" examines the OS environment and configures the Makefile.
- "make" uses the Makefile to compile the software.
- "make install" copies the software to the appropriate system directories; sudo is used to ensure the files can be copied to the protected locations

The traditional method of software installation in the UNIX world is manually compiling source code into executable form. This is usually done with the following commands:

```
$ ./configure
```

```
$ make
```

```
$ sudo make install
```

The source tarball downloaded from the Internet contains a file called "configure", which we execute from the current directory by calling "./configure". This command examines the operating system and the software already installed on it and configures the Makefile which will be used in the next step. The "make" command uses the compiler that is already installed in the operating system to create binary executable programs. It references the Makefile created by the "./configure" step and compiles the source code into binary form. Finally, "make install" copies the newly-compiled binaries from the current directory to the appropriate system directories. This last step will usually copy the binaries (executables) to directories only writeable by root, as such you will need root permissions to perform this step.



Installing Software from Source (2)



These commands are often combined into a single command:

```
# ./configure && make && make install
```

- The double ampersand is a conditional operator that says "IF the first command succeeds, THEN execute the second command"

To remove software installed from source, enter the original source directory and run:

```
# make uninstall
```

- This will not always work

These three commands are commonly joined together on a single line using double ampersands (&&) like this:

```
# ./configure && make && make install
```

The double ampersand is a conditional operator that says "IF the first command succeeds, THEN execute the next command".

To uninstall a program that was compiled from source, enter the original source directory and type "make uninstall". Unfortunately, not all software contains this feature in its Makefile.



Package Managers



Most Linux distros use some form of package manager to speed up the installation process and make it less error prone

- Red Hat-based distros use RPM
- Debian-based distros use APT

Packages contain pre-compiled software for your distribution and processor type

Package managers can be used from the GUI or CLI

To speed up the installation process and make it less prone to errors, most modern Linux distributions use some form of package manager. Distributions based on Red Hat use RPM, the RPM Package Manager (note the recursive acronym!). Distributions based on Debian (including Ubuntu) use the DEB package format with APT. In either case, the package is a single file that contains the entire application, pre-compiled for your distribution and processor. The package can be installed using a package manager with a graphical user interface, or from the command-line.



RPM Examples



Use "rpm" to install RPM files:

```
# rpm -Uvh NewApplication-3.2.1.rpm
```

"rpm" can also download and install an RPM in a single step:

```
# rpm -Uvh http://site.example.com/NewApp-3.2.1.rpm
```

To delete an application using RPM, use the "-e" option and the package name:

```
# rpm -e NewApplication
```

Most package managers can validate installed packages to make sure they haven't been tampered with. The following command can help detect tampered files on a Red Hat system:

```
# rpm -Va | sort
```

To install an RPM package on a Red Hat-based system, use the "rpm" command as follows:

```
# rpm -Uvh NewApplication-3.2.1.rpm
```

The "U" means "install or upgrade", the "v" means to print more verbose information, and the "h" means to print a progress bar during the install.

Red Hat systems can download and install an RPM package in one step using a command like this:

```
# rpm -Uvh http://site.example.com/NewApp-3.2.1.rpm
```

That same application could then be removed from the system with the following command:

```
# rpm -e NewApplication
```

If an attacker compromises a machine and modifies a file that belongs to an RPM package, then the following command can help detect that:

```
# rpm -Va | sort
```

The "V" means to verify packages, and the "a" means "all packages".



Package Repositories



Linux vendors maintain online repositories of all software included in their distribution

This makes it easy to install software after installing your system, straight from the online repository

- Rather than needing to go to a vendor's website like in the Windows world, you can get almost all software you need from one place

Linux distros generally have tools for automating this process

- On Red Hat systems, this tool is called "yum"

Linux vendors maintain online repositories of all of the software they've decided to include in their distribution. This means that if you did a default installation (which doesn't include every single package in the repository) and later decide that you need one of those packages, you can simply install it straight from the repo instead of having to locate it at a third-party site.

Linux distributions generally have tools for automating this process. On Red Hat systems, this tool is called "yum".



Exercise: Package Repositories



First, we need to update the repositories on our VM

You will need to connect to the internet and you will need an IP address to do so

```
[cyberaces@localhost ~]$ sudo dhclient ens33
[cyberaces@localhost ~]$ ifconfig ens33
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet X.X.X.X netmask 255.255.255.0 broadcast X.X.X.255
    ether 00:0c:29:6a:0a:c4 txqueuelen 1000 (Ethernet)
    RX packets 4373 bytes 854721 (834.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 40 bytes 4575 (4.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

In this lab we are going install software on our system from the internet. To do so, you will need to obtain an IP address using the command below:

```
$ sudo dhclient ens33
```

The `sudo` is required to run the `dhclient` command as root. The `dhclient` executable will dynamically obtain an IP address on the `ens33` interface. In the text above, our place holder for our IP address is `X.X.X.X`, your IP address will be different. You should be able to get to the internet at this point.



Exercise: Package Repositories (1)



```
[cyberaces@localhost ~]$ nmap
bash: nmap: command not found...
Failed to search for file: Cannot update read-only repo
[cyberaces@localhost ~]$ rpm -qa | grep nmap
nmap-ncat-7.70-4.el8.x86_64
[cyberaces@localhost ~]$ yum search nmap
CentOS-8 - AppStream                1.1 MB/s | 6.3 MB    00:05
CentOS-8 - Base                     1.5 MB/s | 7.9 MB    00:05
CentOS-8 - Extras                   687 B/s | 2.1 kB     00:03
===== Name Exactly Matched: nmap =====
nmap.x86_64 : Network exploration tool and security scanner
===== Name & Summary Matched: nmap =====
nmap-ncat.x86_64 : Nmap's Netcat replacement
nmap-ncat.x86_64 : Nmap's Netcat replacement
```

1. Try to start a network scanning tool called nmap:

```
$ nmap
```

You should receive a "command not found" error.

2. Let's verify that it's not installed by querying the RPM database:

```
$ rpm -qa | grep nmap
```

There is output showing "nmap-ncat" which is Nmap's variant of Netcat called "Ncat". Nmap is not installed.

3. Use the "yum search" command to verify that the repositories have nmap:

```
$ yum search nmap
```

You should find one tool that matched exactly.



Exercise: Package Repositories (2)



```
[cyberaces@localhost ~]$ sudo yum install nmap
CentOS-8 - AppStream 1.7 MB/s | 6.3 MB 00:03 A
CentOS-8 - Base 1.3 MB/s | 7.9 MB 00:06
CentOS-8 - Extras 970 B/s | 2.1 kB 00:02
Dependencies resolved.
=====
Package Arch Version Repository Size
=====
Installing:
nmap x86_64 2:7.70-4.el8 AppStream 5.8 M
Transaction Summary
=====
Install 1 Package
Total download size: 5.8 M
Installed size: 24 M
Is this ok [y/N]: y
```

4. Use the "yum install" command to install the nmap RPM from the CentOS repositories:

```
$ sudo yum install nmap
```

Note: This also requires a working internet connection

5. You will be asked if this is okay before continuing; type "y" to continue. Note how it shows you a list of exactly what actions will be performed, including any dependencies that will be affected.



Exercise: Package Repositories (3)



```
Downloading Packages:
nmap-7.70-4.el8.x86_64.rpm                12 MB/s | 5.8 MB    00:00
-----
Total                                    3.9 MB/s | 5.8 MB    00:01
warning: /var/cache/dnf/AppStream-a520ed22b0a8a736/packages/nmap-7.70-
4.el8.x86_64.rpm: Header V3 RSA/SHA256 Signature, key ID 8483c65d: NOKEY
CentOS-8 - AppStream                     1.6 MB/s | 1.6 kB    00:00
Importing GPG key 0x8483C65D:
  Userid      : "CentOS (CentOS Official Signing Key) <security@centos.org>"
  Fingerprint: 99DB 70FA E1D7 CE22 7FB6 4882 05B5 55B3 8483 C65D
  From        : /etc/pki/rpm-gpg/RPM-GPG-KEY-centosofficial
Is this ok [y/N]: y
```

This is more of the output from the "yum install nmap" command on the previous slide.

6. You will be asked if you want to import a key; type "y" to continue. Note how it shows you a list of exactly what actions will be performed.



Exercise: Package Repositories (4)



```
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                                1/1
  Installing     : nmap-2:7.70-4.el8.x86_64      1/1
  Running scriptlet: nmap-2:7.70-4.el8.x86_64      1/1
  Verifying      : nmap-2:7.70-4.el8.x86_64      1/1
Installed:
  nmap-2:7.70-4.el8.x86_64

Complete!
[cyberaces@localhost ~]$
```

This is the rest of the output from the "yum install nmap" command. It has now been successfully installed!



Exercise: Package Repositories (5)



```
[cyberaces@localhost ~]$ nmap -h
Nmap 7.70 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  ...
```

6. The nmap RPM is installed. Verify you can execute nmap:

```
$ nmap -h
```

You should see the usage options for nmap. This verifies that it was installed correctly.



Exercise: Package Repositories (6)



```
[cyberaces@localhost ~]$ sudo yum erase nmap
```

```
Dependencies resolved.
```

```
=====
```

Package	Arch	Version	Repository	Size
=====				
Removing:				
nmap	x86_64	2:7.70-4.el8	@AppStream	24 M

```
Transaction Summary
```

```
=====
```

Remove	1 Package
--------	-----------

```
Freed space: 24 M
```

7. Remove the nmap RPM with the "yum erase" command:

```
$ sudo yum erase nmap
```

Note how it shows you a list of exactly what actions will be performed, including any dependencies that will be affected.



Exercise: Package Repositories (7)



```
Is this ok [y/N]: y
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                                1/1
  Erasing       : nmap-2:7.70-4.el8.x86_64        1/1
  Running scriptlet: nmap-2:7.70-4.el8.x86_64      1/1
  Verifying      : nmap-2:7.70-4.el8.x86_64        1/1

Removed:
  nmap-2:7.70-4.el8.x86_64

Complete!
[cyberaces@localhost ~]$
```

8. This is the rest of the output from the "sudo yum erase nmap" command. Note that when you are asked if this transaction is okay, you should type "y".



Review



Which of the following commands is used to install software from an online repository?

- `install`
- `yum install`
- `rpm install`
- `installrpm`

Which of the following commands is used to query the RPM database to determine if a package (such as `tcpdump`) is installed?

- `yum find tcpdump`
- `rpm tcpdump`
- `yum search tcpdump`
- `rpm -qa | grep tcpdump`

Which of the following commands is used to install software from an online repository?

- `install`
- `yum install`
- `rpm install`
- `installrpm`

Which of the following commands is used to query the RPM database to determine if a package (such as `tcpdump`) is installed?

- `yum find tcpdump`
- `rpm tcpdump`
- `yum search tcpdump`
- `rpm -qa | grep tcpdump`



Answers



Which of the following commands is used to install software from an online repository?

- `yum install`

Which of the following commands is used to query the RPM database to determine if a package (such as `tcpdump`) is installed?

- `rpm -qa | grep tcpdump`
- "`rpm -qa`" generates a list of all installed packages, and piping that into "`grep tcpdump`" searches the list for `tcpdump`

Which of the following commands is used to install software from an online repository?

`yum install`

Which of the following commands is used to query the RPM database to determine if a package (such as `tcpdump`) is installed?

`rpm -qa | grep tcpdump`

"`rpm -qa`" generates a list of all installed packages, and piping that into "`grep tcpdump`" searches the list for `tcpdump`



Software Updates



Online repositories are particularly convenient for software updates

- Keeping software fully patched is very important for system security!

On Red Hat-based systems, you can update all software by running:

```
# yum update
```

To update a specific package:

```
# yum update tcpdump
```

To update all packages except a specific package:

```
# yum update --exclude tcpdump
```

Distribution repositories are not only handy for installing new software, but also for keeping your existing software updated. We all know that we have to keep our software updated so that we minimize our vulnerabilities and therefore reduce our risk of compromise. Keeping all of your repo software updated is as simple as running "yum update". You can also update just a specific package by specifying its name, or leave out certain packages by using the "--exclude" option ("-x" for short). You can also use the * as a wildcard when specifying package names. For example, the following command would update all software except anything starting with "kernel":

```
# yum update --exclude kernel*
```



Exercise: Software Updates (1)



```
[cyberaces@localhost ~]$ sudo yum update
Dependencies resolved.
```

Package	Arch	Version	Repository	Size
Installing:				
kernel	x86_64	4.18.0-80.11.2.el8_0	BaseOS	424 k
kernel-core	x86_64	4.18.0-80.11.2.el8_0	BaseOS	24 M
kernel-modules	x86_64	4.18.0-80.11.2.el8_0	BaseOS	20 M

Upgrading:				
anaconda-core	x86_64	29.19.0.43-1.el8_0	AppStream	2.1 M
...trimmed for brevity...				
vdo	x86_64	6.2.0.298-10.el8_0	BaseOS	682 k

Installing dependencies:				
xorg-x11-drv-fbdev	x86_64	0.5.0-2.el8	AppStream	27 k
...trimmed for brevity...				

Transaction Summary

```
Install 6 Packages
Upgrade 166 Packages
```

```
Total download size: 252 M
Is this ok [y/N]: y
```

1. In your CentOS VM, open a terminal and search for updates using the command "yum update". You have to have root level access, so we prefix this command with "sudo". Note that this requires a working Internet connection.

\$ **sudo yum update**

2. After a few seconds, Yum will report that a certain number of packages need to be updated and ask if it's OK to continue. Answer "y" to continue installing the updates. This will take a while to complete and will download a large amount of data.

After you have completed this step, you can close the terminal window.



Review Questions



Which of the following commands is used to install software updates?

- rpm update
- update
- yum update
- installupdate

Which of the following commands is used to update all installed RPM packages EXCEPT for httpd?

- yum update --exclude httpd
- yum dont-update httpd
- yum update --without-httpd
- yum update -httpd

Which of the following commands is used to install software updates?

- rpm update
- update
- yum update
- installupdate

Which of the following commands is used to update all installed RPM packages EXCEPT for httpd?

- yum update --exclude httpd
- yum dont-update httpd
- yum update --without-httpd
- yum update -httpd



Answers



Which of the following commands is used to install software updates?

- `yum update`

Which of the following commands is used to update all installed RPM packages EXCEPT for httpd?

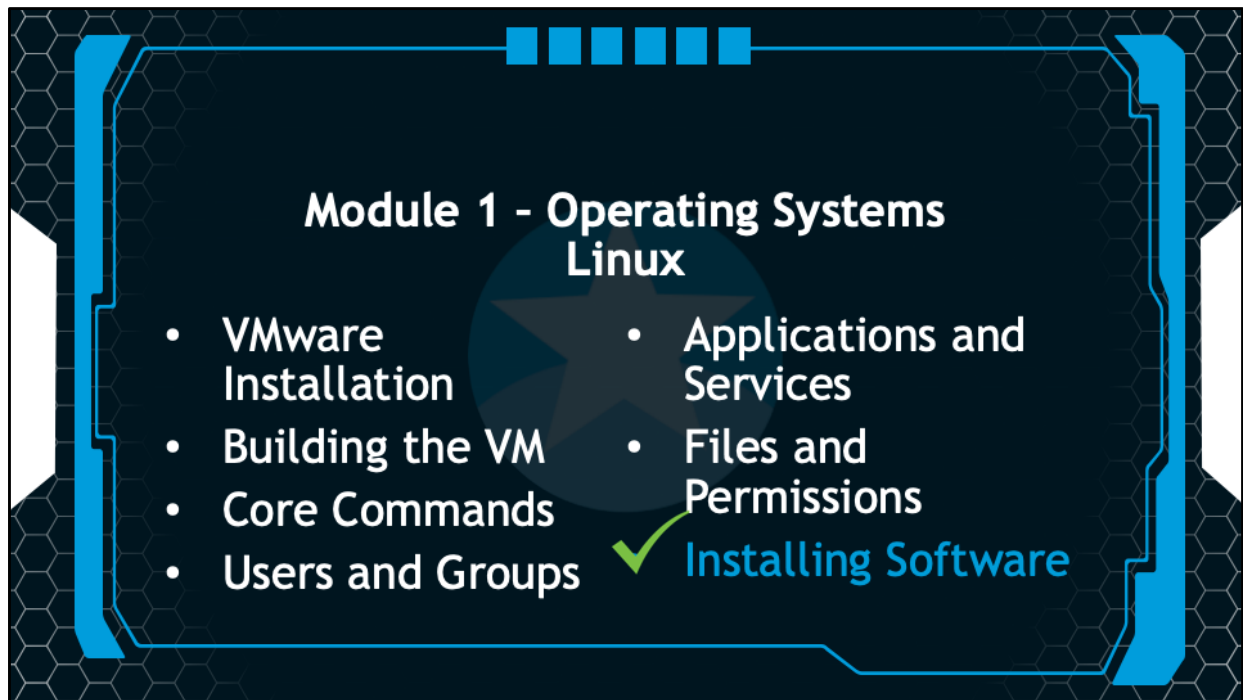
- `yum update --exclude httpd`

Which of the following commands is used to install software updates?

`yum update`

Which of the following commands is used to update all installed RPM packages EXCEPT for httpd?

`yum update --exclude httpd`



You have successfully completed the session on installing software in Linux. This was Session 6 of Module 1, Operating Systems – Linux.



Linux Conclusion



This concludes our whirlwind tour of the Linux operating system
The CentOS manuals are a great resource to learn more
You may also wish to continue experimenting with Linux by
downloading Ubuntu, Fedora, or Backtrack

This concludes our whirlwind tour of the Linux operating system. Once again, the CentOS manuals are very comprehensive and can be found here:

<https://redsiege.com/ca/centos-guide>

You may wish to continue experimenting with Linux by downloading other free distributions such as Ubuntu, Fedora, and Kali:

<http://www.ubuntu.com/>

<http://fedoraproject.org/>

<http://www.kali.org/>

SANS CYBER ACES ONLINE TUTORIALS

YOUR GATEWAY TO CYBERSECURITY SKILLS AND CAREERS

1. Introduction to Operating Systems

- ✓ 01. Linux
- 02. Windows

2. Networking

3. System Administration

- 01. Bash
- 02. PowerShell
- 03. Python

You have successfully completed the Linux portion of Module 1, Introduction to Operating Systems. In the next series of sessions we will discuss Windows.