
Cyber Aces

Module 1 – Operating Systems

Linux Applications and Services

By Tom Hessman, Tim Medin, Mark Baggett, Doug Burks,
Michael Coppola, Russell Eubanks, and Ed Skoudis

Presented by Tim Medin

v15Q1

This tutorial is licensed for personal use exclusively for students and teachers to prepare for the Cyber Aces competitions. You may not use any part of it in any printed or electronic form for other purposes, nor are you allowed to redistribute it without prior written consent from the SANS Institute.

Cyber Aces Online Module 1 - ©2015 The SANS Institute. Redistribution Prohibited.

1

Welcome to Cyber Aces Online Module 1 - Linux. In this session we will discuss Linux applications and services.

Applications and Services

- In general, there are two types of software: Applications and Services
 - On Linux, services are usually called daemons
 - Daemons is usually pronounced DEE-mons but some call them DAY-mons
- Applications are software started and interacted with by the user
 - Typically client software
 - Example: Firefox web browser
- Services are software started by the OS that run in the background
 - Typically server software
 - Example: Apache web server

Applications and Services

There are two categories of software that run on top of the operating system: applications and services (commonly called daemons on Linux). Applications are started by the user, interacted with, and then closed. For example, you sit down at your Linux VM, open your Firefox web browser, search for something on google.com, and then close your browser down. That's an application. Services, on the other hand, typically start when the computer boots up and run until the computer shuts down. An example of a service would be the HTTP service running on one of Google's servers that your web browser connects to. That HTTP service is started automatically by the operating system and will continue running until the operating system shuts down or an administrative user stops the service.

Linux Boot Process

- Typical Linux boot process:
 - BIOS starts the boot loader
 - Boot loader loads the kernel into memory
 - The kernel mounts disks/partitions & starts the init daemon
 - The init daemon starts services based on the runlevel
- A runlevel defines a set of services to run on startup
 - Runlevels are essentially profiles
- There are six runlevels on a typical Linux system, which define different modes for the system to run in
 - For example, runlevel 1 is typically Single User Mode, which is used for diagnostics
 - Most Linux distributions will normally run in either runlevel 3 or 5

Cyber Aces Online Module 1 - ©2015 The SANS Institute. Redistribution Prohibited.

3

Linux Boot Process

In a traditional Linux system, the boot process goes like this:

- The computer BIOS starts the boot loader, a program on the boot device (such as a hard drive or DVD)
- The boot loader loads the kernel into memory
- The kernel mounts the disk partitions needed to run the system and starts the init program, which is the first user-mode program on the machine
- Init starts services based on the runlevel

In the last step of the boot process, we said that init starts services based on the runlevel. What's a runlevel? A runlevel is a way of telling the operating system what you want to do with it so that it can start services appropriately. Most of the time, the runlevel will be 5, which tells Linux to start all normal services and the graphical user interface (GUI). An alternative would be runlevel 3, which is similar to runlevel 5, but without the GUI. However, for maintenance or emergency situations, we may choose to go to runlevel 1. This is a text-only mode where just the root user can login and no system services are started. Other runlevels usually have different sets of services that are started automatically.

CentOS Runlevels

- Varies based on the distribution, but this is a common setup:
- 0: System Halt (shutdown)
- 1: Single-User Mode, no GUI (no services)
- 2: Multi-User Mode, no GUI or networking
- 3: Multi-User Mode, no GUI
- 4: Not used, user definable
- 5: Multi-User Mode, load GUI
- 6: Reboot

Cyber Aces Online Module 1 - ©2015 The SANS Institute. Redistribution Prohibited.

4

CentOS Runlevels

Most Linux system use a set of runlevels similar to the following (though some Linux distributions define runlevels 2-5 differently):

- 0: Shuts down all services and the operating system itself gracefully.
- 1: Enters Single-User Mode, which is typically an emergency rescue mode that simply loads a command shell as the root account (and no other services)
- 2: Boots into Multi-User Mode, but does not load any services related to networking or a windowing system (GUI)
- 3: Boots into Multi-User Mode with networking, but does not load any services related to a windowing system (GUI)
- 4: Not typically used for anything
- 5: Boots into Multi-User Mode, and loads the windowing system (GUI)
- 6: Shuts down all services and reboots the operating system gracefully

For additional information on this subject check out the link below:

[http://www.techotopia.com/index.php/Configuring CentOS 6 Runlevels and Services](http://www.techotopia.com/index.php/Configuring_CentOS_6_Runlevels_and_Services)

Service Management

- Each service (or daemon) contains an "init script" to manage starting and stopping that service gracefully
- On traditional Linux systems (including CentOS), the `/etc/rc.d` directory contains directories corresponding to each runlevel
- Each runlevel directory (`/etc/rc.d/rcN.d`) contains numbered symlinks (more on those later) to init scripts
 - The numbering represents the order in which the services should be started, allowing for one service to depend on another
 - Each symlink starts with an S (start) or a K (kill), indicating whether that service is enabled or disabled at that runlevel
- An important rule of computer security is to disable any unnecessary services
- There are command line and graphical tools to manage the `/etc/rc.d` directories (changing symlinks can be tedious!)

Cyber Aces Online Module 1 - ©2015 The SANS Institute. Redistribution Prohibited.

5

Service Management

How do we tell the operating system which services should be run in what runlevel? We do that via the `/etc/rc.d/` directory structure. Inside of `/etc/rc.d/` are subdirectories that correspond to each runlevel, such as `rc0.d`, `rc1.d`, etc. Inside of each of these subdirectories are symbolic links (similar to shortcuts in Windows) to all of the services that are available in the operating system. The first letter of the name determines whether the service will be started in that particular runlevel or not. If the first letter of the name is an S, then that service will be automatically started. If the first letter is a K, then that service will be killed when entering that runlevel (or simply not started). So, for example, if `/etc/rc.d/rc5.d/S25sshd` exists, then `sshd` (the Secure Shell daemon) will start automatically in runlevel 5.

On your LiveCD the `ssh` daemon is disabled at run level 5. To enable it we could run this command:

```
# mv /etc/rc.d/rc5.d/K25sshd /etc/rc.d/rc5.d/S25sshd
```

One rule of computer security is to disable any unnecessary services. So if we look at the `rc.d` directory for our current runlevel (`rc5.d`, for example) and see an unnecessary service, we can disable it by renaming the file and changing the first letter from an S to a K. For example:

```
# mv /etc/rc.d/rc5.d/S25sshd /etc/rc.d/rc5.d/K25sshd
```

Some tools are available for CentOS to make this easier than altering each and every S link. `chkconfig` and `ntsysv` are both command-line tools. The Service Configuration Tool (`system-config-services`) is a graphical tool but it is not installed by default in CentOS 6.5. To install it run the following command:

```
$ sudo yum -y install system-config-services
```

Note: Your VM is a LiveCD so the changes made with this tool, and the installation itself, will be gone after a reboot of your VM.

chkconfig

- The "chkconfig" command can be used to enable and disable services, as well as to determine which services are enabled and disabled
- To view service status (run without service name for all services):
 - # `chkconfig --list [service name]`
- To enable or disable a service at all applicable runlevels:
 - # `chkconfig <service name> <on|off>`
- Chkconfig can also enable or disable a service at particular runlevels:
 - # `chkconfig --level 5 <service name> <on|off>`

chkconfig

On Red Hat-based systems, the "chkconfig" command can be used to enable and disable services, as well as to determine which services are enabled and disabled at all runlevels.

To view the status of all services at all runlevels, run:

```
# chkconfig -list
```

To view the status of a particular service, run:

```
# chkconfig --list [service name]
```

To enable or disable a service (e.g. SSH) at all applicable runlevels, run:

```
# chkconfig sshd off
```

Chkconfig can also enable or disable a service at particular runlevels using the "--level" option. Note that it is possible to specify more than one runlevel at once by simply combining the numbers. For example, to enable the SSHD service at runlevels 3 & 5, run:

```
# chkconfig --level 35 sshd on
```

ntsysv

- The command line tool "ntsysv" provides a semi-graphical interface for managing services at a particular runlevel
- Use the arrow keys to select a service, then use the space bar to enable or disable it



ntsysv

The "ntsysv" tool runs at the command line, but provides an ncurses-based interface for managing services at a particular runlevel (ncurses is a library for creating semi-graphical applications at the CLI). By default, it edits the current runlevel, but it also supports the same "--levels" syntax as chkconfig to edit one or more arbitrary runlevels at a time.

To manage services, simply use the arrow keys to select a service, and use the space bar to enable or disable it. Then, press the tab key to move the cursor to the "Ok" button and press the space bar to "click" it.

Service Configuration Tool

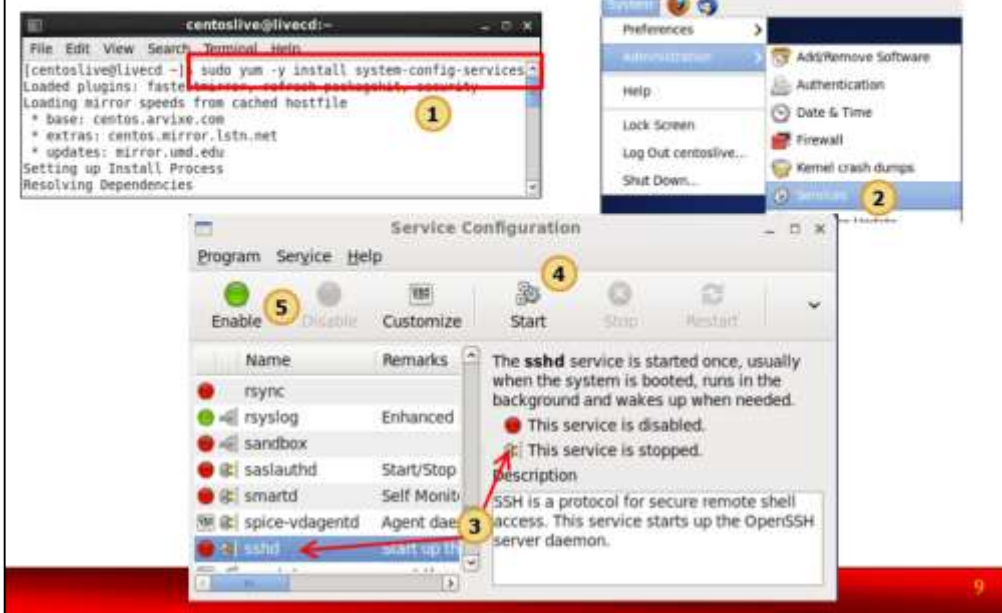
- Red Hat-based systems also provide a GUI-based tool for managing services known as system-config-services
- To enable or disable a service, simply check or clear its checkbox
- You can also start, stop, or restart a service immediately, and see its status and description



Service Configuration Tool

Red Hat-based systems also provide a GUI-based tool for managing services called "system-config-services" (based on its executable name). To run it, run "system-config-services" from a root shell prompt. To enable or disable a service, simply check or clear its checkbox. You can also start, stop, or restart a service immediately, and see its current status (whether it's currently running) and its description. It edits run levels 2-5 by default, but can be configured to edit any of the run levels.

Services Exercise (1)



Services Exercise (1)

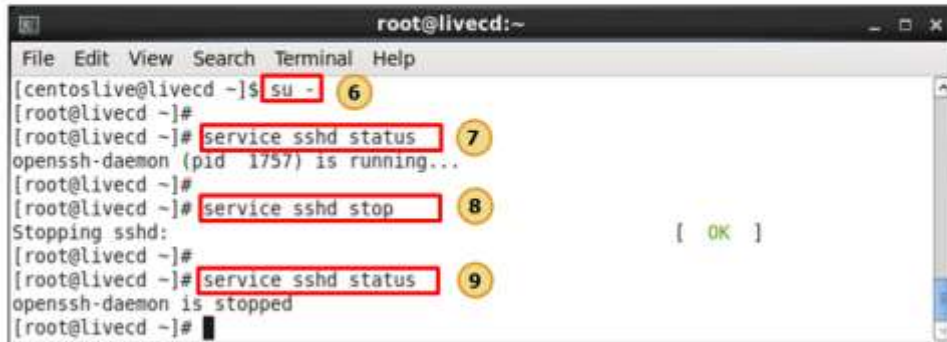
1. Install the system-config-services tool by typing the command below.

```
$ sudo yum -y install system-config-services
```

The "yum" utility is the package manager that will install the tool for us. The -y option will tell yum to answer "yes" to all prompts.

2. In your CentOS VM, click System, click Administration, and then click Services. This new option is added after the above tool is installed.
3. The "Service Configuration" window appears. On the left side, notice the list of services. Scroll down and click on "ssh". Notice on the right side the status box that says "This service is stopped" and "This service is disabled".
4. Click the Start button. Notice that the status box now says "This service is running", the "Start" button is disabled, and the "Stop" button is now available.
5. Notice the icon to the left of sshd is red. This means that sshd is not configured to start automatically on boot. Click the "Enabled" button. SSHD is now configured to start automatically on boot. (Note that since this is a LiveCD, this setting will not be saved as it would be on a normal Linux system).

Services Exercise (2)



The screenshot shows a terminal window titled 'root@livecd:~'. The user starts at a prompt '[centoslive@livecd ~]\$' and enters 'su -' (step 6). The prompt changes to '[root@livecd ~]#'. The user then enters 'service sshd status' (step 7), which returns 'openssh-daemon (pid 1757) is running...'. Next, the user enters 'service sshd stop' (step 8), which returns 'Stopping sshd: [OK]'. Finally, the user enters 'service sshd status' (step 9), which returns 'openssh-daemon is stopped'.

```
root@livecd:~
File Edit View Search Terminal Help
[centoslive@livecd ~]$ su - 6
[root@livecd ~]#
[root@livecd ~]# service sshd status 7
openssh-daemon (pid 1757) is running...
[root@livecd ~]#
[root@livecd ~]# service sshd stop 8
Stopping sshd: [ OK ]
[root@livecd ~]#
[root@livecd ~]# service sshd status 9
openssh-daemon is stopped
[root@livecd ~]#
```

Services Exercise (2)

6. Close the Service Configuration window. Open a terminal window and become root by typing the following command. Please note that the dollar sign (\$) is your terminal prompt and you shouldn't type that. Only type the letters in bold.

\$ **su -**

If you successfully became root, then your terminal prompt should have changed from a dollar sign (\$) to a pound sign (#), also known as an octothorpe. The root prompt (#) reminds you that you are root and have full privileges over the entire system. However, with power comes responsibility, so remember that any mistakes you make as root could have disastrous consequences!

7. Check the status of the sshd service with the following command:

service sshd status

It should show that the openssh-daemon is running.

8. Stop the sshd service with the following command:

service sshd stop

9. Press the Up arrow twice to recall the status command and press Enter. It should show that openssh-daemon is now stopped.

Services Exercise (3)

The screenshot shows a terminal window titled 'root@livecd:~'. It contains the following commands and output, with yellow circles and red boxes highlighting specific parts:

```
root@livecd:~# chkconfig --list sshd 10
sshd      0:off  1:off  2:on   3:on   4:on   5:on   6:off

root@livecd:~# chkconfig sshd off 11
root@livecd:~# chkconfig --list sshd 12
sshd      0:off  1:off  2:off  3:off  4:off  5:off  6:off

root@livecd:~# ls -hal /etc/rc.d/rc5.d/*sshd 13
lrwxrwxrwx. 1 root root 14 Jul 17 08:18 /etc/rc.d/rc5.d/K25sshd -> ../init.d/sshd

root@livecd:~#
```

Lowercase L

Services Exercise (3)

10. Type the following command to see if sshd is set to start automatically at boot:

```
# chkconfig --list sshd
```

The "5:on" verifies that sshd is configured to automatically start in runlevel 5.

11. Type the following to set sshd to not automatically start:

```
# chkconfig sshd off
```

12. Press the Up arrow key twice to recall the previous command and press Enter. The "5:off" verifies that sshd is configured to not automatically start.

13. Type the following command (note that "alh" contains a lower case L):

```
# ls -hal /etc/rc.d/rc5.d/*sshd
```

You should see that the sshd link starts with a "K" which means it will not start automatically on boot. If it was set to start automatically on boot, the sshd link would start with an "S".

14. Exit the root account by typing the following command:

```
# logout
```

Alternatively, you could type "exit" or just press Ctrl-D.

15. Notice that your terminal prompt changes back to a dollar sign (\$), signifying that you are no longer root and you just have standard user privileges.

16. Close the terminal window.

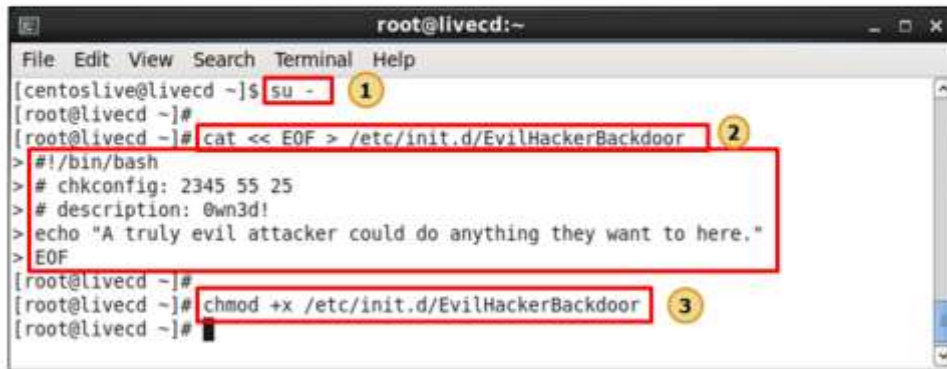
Init Backdoors

- One technique attackers use is to create a backdoor in `/etc/rc.d/init.d/`
- By creating their own service and configuring it at the default runlevel, the attacker's service will start every time the machine boots
- You can use the tools in the previous section (such as `chkconfig`) to look for suspicious services

Init Backdoor

One technique used by computer attackers is to create a backdoor in `/etc/rc.d/init.d/`. Recall that this is the directory that specifies system services. If attackers can install their own service in this directory (by writing a simple script) and create an entry in the appropriate `rc.d` directory (such as `rc5.d` if we're in runlevel 5), then the attackers' service will start every time the machine boots and the bad guys will always have access to the box. You can use the tools listed in the previous section to find services that look suspicious.

Init Backdoors Exercise (1)



```
root@livecd:~  
File Edit View Search Terminal Help  
[centoslive@livecd ~]$ su - 1  
[root@livecd ~]#  
[root@livecd ~]# cat << EOF > /etc/init.d/EvilHackerBackdoor 2  
> #!/bin/bash  
> # chkconfig: 2345 55 25  
> # description: 0wn3d!  
> echo "A truly evil attacker could do anything they want to here."  
> EOF  
[root@livecd ~]#  
[root@livecd ~]# chmod +x /etc/init.d/EvilHackerBackdoor 3  
[root@livecd ~]#
```

Init Backdoors Exercise (1)

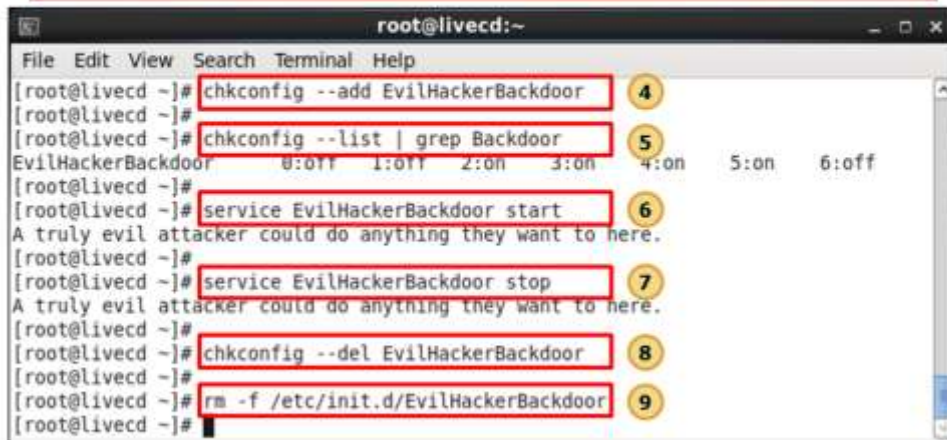
1. In your CentOS VM, start a terminal (using either the panel or desktop icon you created earlier) and become root using the following command:
`$ su -`
2. Now that you are root, type the following command to create a backdoor in /etc/init.d. Remember that the first pound sign (#) you see is the root prompt and you shouldn't type that. The pound signs on lines 2-4 are part of the EvilHackerBackdoor script and you SHOULD type those (they are important to the script). Type everything below to create the script. This techniques uses a "heredoc" to create the script.

```
cat << EOF >> /etc/init.d/EvilHackerBackdoor  
#!/bin/bash  
# chkconfig: 2345 55 25  
# description: 0wn3d!  
echo "A truly evil attacker could do anything they want to here."  
EOF
```

3. Make the backdoor executable:

```
# chmod +x /etc/init.d/EvilHackerBackdoor
```

Init Backdoors Exercise (2)



```
root@livecd:~  
File Edit View Search Terminal Help  
[root@livecd ~]# chkconfig --add EvilHackerBackdoor 4  
[root@livecd ~]#  
[root@livecd ~]# chkconfig --list | grep Backdoor 5  
EvilHackerBackdoor 0:off 1:off 2:on 3:on 4:on 5:on 6:off  
[root@livecd ~]#  
[root@livecd ~]# service EvilHackerBackdoor start 6  
A truly evil attacker could do anything they want to here.  
[root@livecd ~]#  
[root@livecd ~]# service EvilHackerBackdoor stop 7  
A truly evil attacker could do anything they want to here.  
[root@livecd ~]#  
[root@livecd ~]# chkconfig --del EvilHackerBackdoor 8  
[root@livecd ~]#  
[root@livecd ~]# rm -f /etc/init.d/EvilHackerBackdoor 9  
[root@livecd ~]#
```

Init Backdoors Exercise (2)

4. Add the backdoor to all runlevels:
chkconfig --add EvilHackerBackdoor
5. Verify that the backdoor was added (grep is used to search for a particular string of text):
chkconfig --list | grep Backdoor
6. Start the backdoor:
service EvilHackerBackdoor start
You should see the following text:
A truly evil attacker could do anything they want to here.
7. Stop the backdoor:
service EvilHackerBackdoor stop
8. Remove the backdoor from all runlevels:
chkconfig --del EvilHackerBackdoor
9. Remove the backdoor script:
rm -f /etc/init.d/EvilHackerBackdoor
10. Close the terminal window.

Exercise Complete!

- Congratulations! You have completed the Linux Applications and Services Session