

EXPLORING THE CAPABILITIES OF LYNIS

Introduction

We present to you a lab to learn about Lynis - an open source tool which is primarily used for security auditing. Some salient features of Lynis are:

- Security auditing
- Vulnerability detection and scanning
- Compliance testing (e.g. PCI, HIPAA, SOx) using plugins

In this lab we shall explore the security auditing feature, vulnerability detection and scanning feature of Lynis. Performing compliance testing using Lynis plugins is not open source, hence we will not be demonstrating compliance testing in this lab. Section 1 describes a scenario which requires host security auditing, vulnerability detection and scanning in order to be a safer host. Section 2 talks about installing Lynis while Section 3 talks about how to use Lynis. Sections 4 and 5 deal with exploring capabilities of Lynis and log analysis. The above sections talk about Lynis in general. From Section 6 onwards, we put Lynis into action in a specific scenario which was described in Section 1. Section 6 shows how Lynis can be used in performing a vulnerability scanning of a web server and trying to minimize the vulnerabilities. Section 7 deals with verifying that the web server is not prone to vulnerabilities any more. Section 8 of this lab focuses on hardening the webserver that we have based on the suggestions that Lynis gives in order to minimize the attack surface of the server. Finally, in section 9 we compare the system states before hardening and after hardening. We conclude with Section 10 in which we show that the attacks which were possible before are not possible anymore.

1. Scenario

In order to make this lab more practical instead of it being a descriptive document of tool capabilities, we have created a scenario.

We have two machines running Ubuntu 12.04.5 LTS as web servers - **C3PO** and **R2D2** and one machine running Kali Linux as a client machine.

C3PO is the server machine to be secured as this has a web server running on it. We also refer to this as the victim server in this document.

R2D2 is the machine which is patched and which has the final state that our web server should be in after resolving vulnerabilities and after incorporating Lynis suggestions.

Emperor is a compromised client machine (could be either within or external to our network) which can perform certain attacks on our web server. In this lab, we assume that it is within the network where the web server is hosted.

Luke is a rigorous and humble system administrator who is in charge of the web server. He believes that he has secured the web server by removing unwanted services and tools on the web server.

1.1. MACHINE NAMES AND IP ADDRESSES

1. Machine name: **R2D2**
IP address: **172.17.0.90**
2. Machine name: **C3PO**
IP address: **172.17.0.9**
3. Machine name: **Emperor**
IP address: **172.17.0.30**

1.2. USER CREDENTIALS

1. **R2D2**
User name: **Vader**
Password: **aiarocks**
2. **C3PO**
User name: **preeti**
Password: **aiarocks**
3. **Emperor**
User name: **root**
Password: **toor**



R2D2 IP: 172.17.0.9



C3PO IP: 172.17.0.90



Emperor IP: 172.17.0.30



LUKE - SYTEM ADMINISTRATOR

Figure 1: Scenario Map

***Note:** On all web servers the password is 'aiarocks'. When asked for permission to run as a sudo user, enter 'aiarocks' everywhere in the lab.*

1.3 TESTING THE SCENARIO:

From the machine farm pane on the left, right click on Emperor, click on Snapshot Manager and then click on Emperor-Final and 'Go To' to load the lab environment. Click on 'Yes' in the Warning box.

Login to Emperor with the credentials given in Section 1.2

Step 1. Open Iceweasel browser by double clicking on the Weasel and globe icon seen on the top left section in the menu bar.

Step 2. Open a new tab by clicking on the green colored plus button. In the website address bar, type 172.17.0.9/team1.html. You should be able to see a webpage served by C3PO as shown in Figure 2.

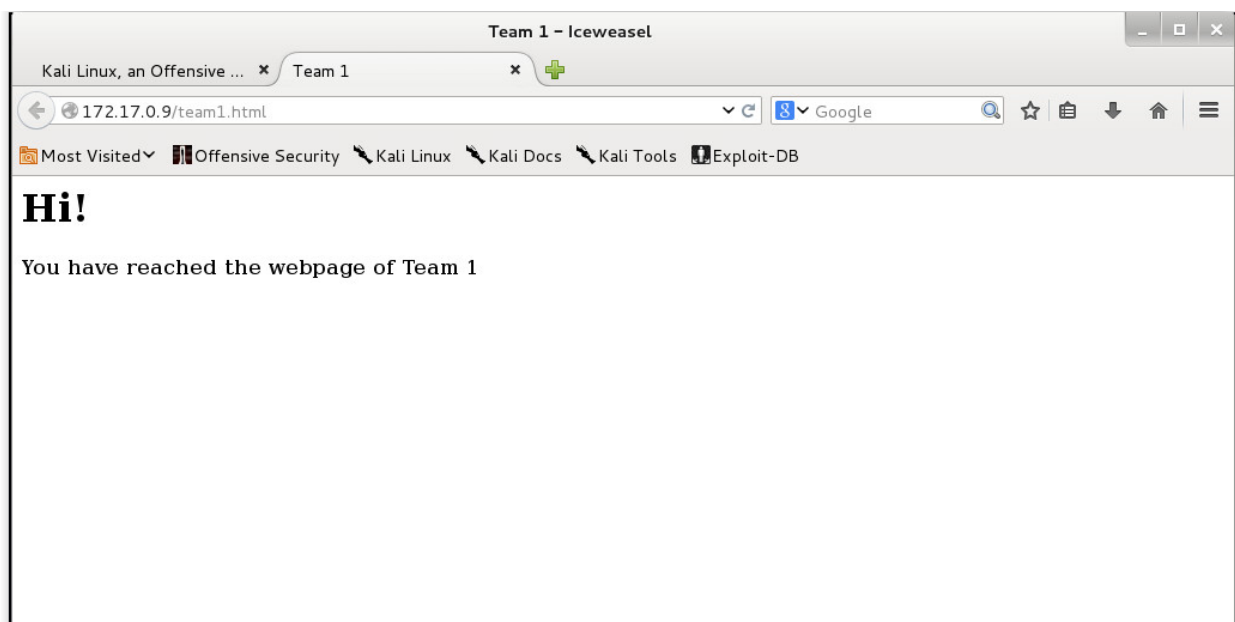


Figure 2: Checking Web-server functionality

Since Emperor is a compromised host, it tries to steal information from the web server or tries to bring it down. For now, assume that you are the Emperor.

The Emperor knows that the most common web server for Linux is Apache. So it tries to check for Apache security flaws. Let us now check if the web server leaks information when there is an error such as a page not found error. In the URL field, type 172.17.0.9/team2.html as team2.html is not hosted on our webserver.

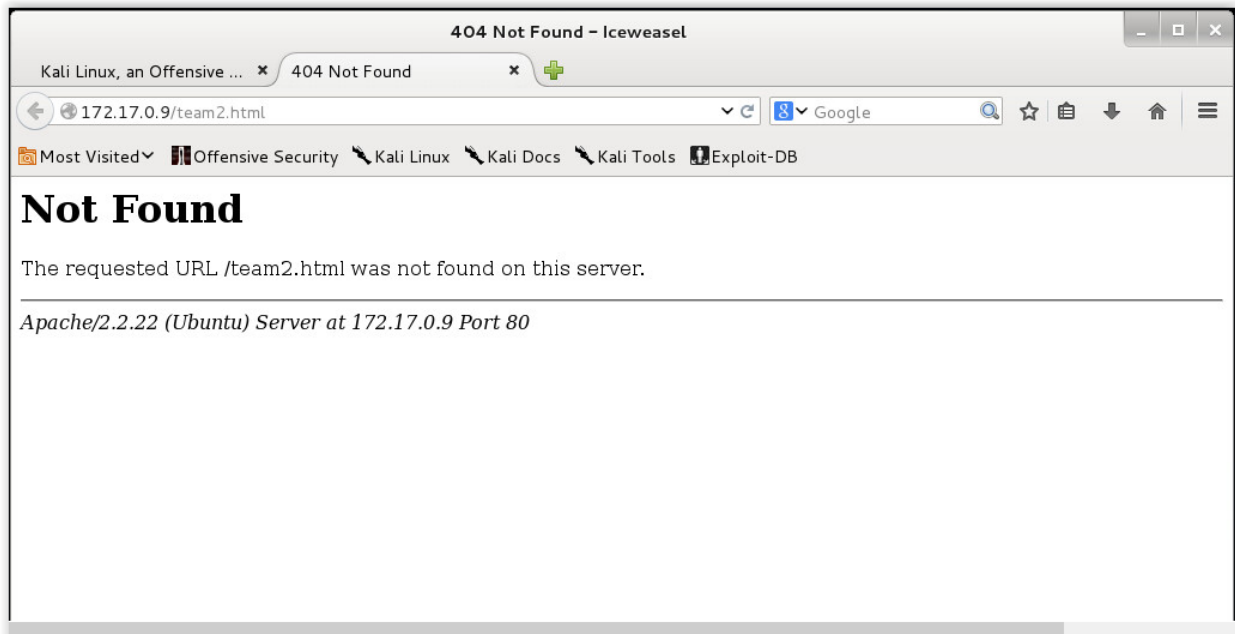


Figure 3: Checking for Information Leaks

Figure 3 shows that the web server leaked some information about itself. So, we now know that it is an Ubuntu web server running Apache 2.2.22.

Minimize the browser and open a terminal window from the top left of the menu bar. In the terminal, perform the following:

1. The first thing you would want is to gain root access to the webserver so that you have the highest privilege in the target machine which is C3PO.

Run port scan on C3PO by typing

```
nmap 172.17.0.9
```

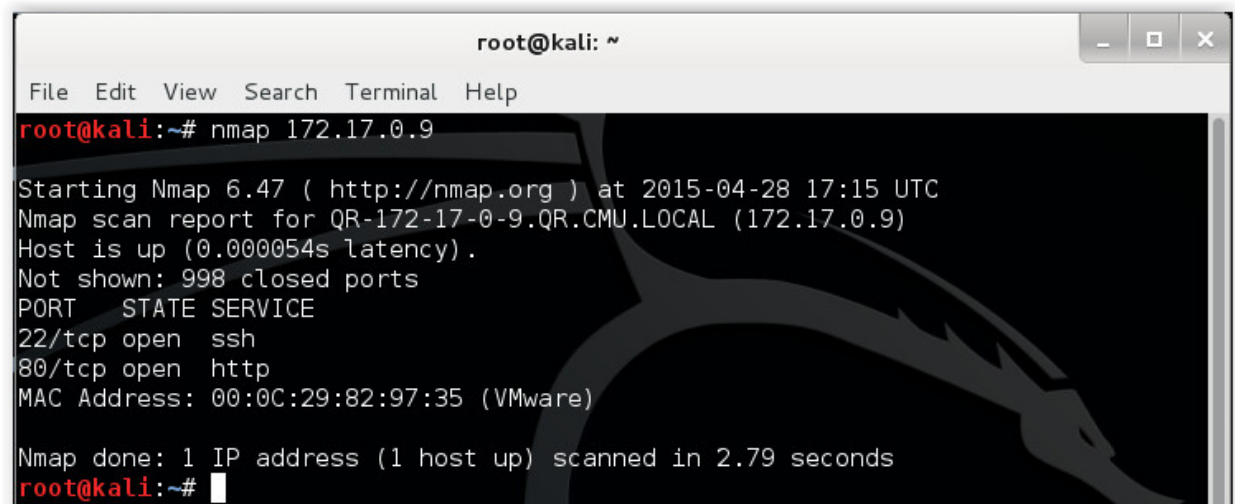


Figure 4: Portscan output

From Figure 4 we know that the webserver has other ports open apart from port 80. This is the SSH port 22

2. A web server need not have an SSH connection. Even if an SSH connection is possible, a remote user should not login as root. So, let us try to setup an SSH connection as root and see if it works. Enter the following command in the terminal

```
ssh root@172.17.0.9
```

The output should look like Figure 5. Type 'yes' when prompted to add ECDSA key fingerprint permanently. We are asked for a password. Let us assume that Emperor was able to brute force the password using Hydra or some other tool and found that the password is **root123**.

Type **root123** when prompted for a password. You will now notice that you are in C3PO's prompt where you are the root user.

The prompt is as shown below

```
root@preeti-virtual-machine:~#
```

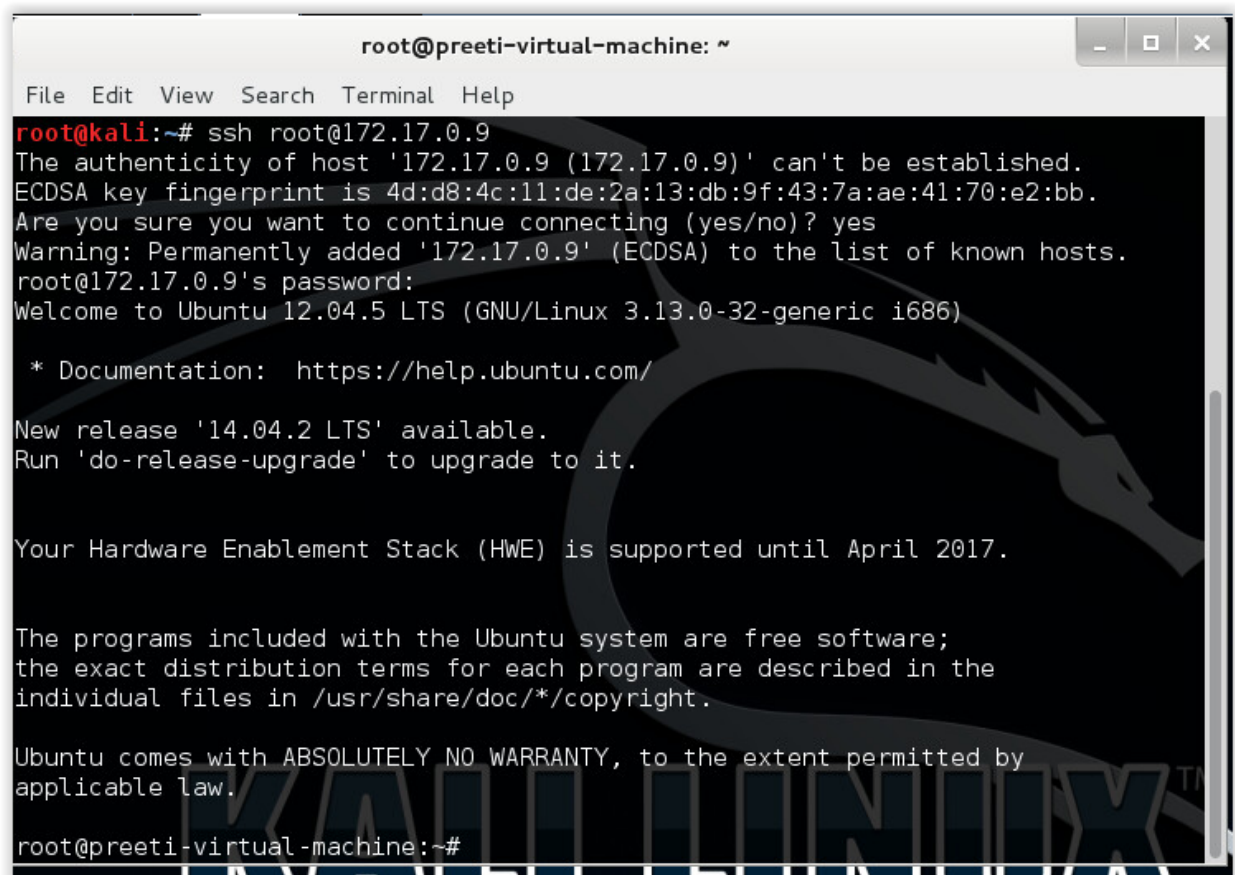


Figure 5: Successful SSH connection to the web server

We are now having root access on the web server and we can execute any command with root privileges.

3. Being an attacker, one tries to exploit as much vulnerability as possible. Since the Emperor learned that the server running is Ubuntu and that many Ubuntu servers' bash is not upgraded, the Emperor tries to see if a shellshock attack is possible.

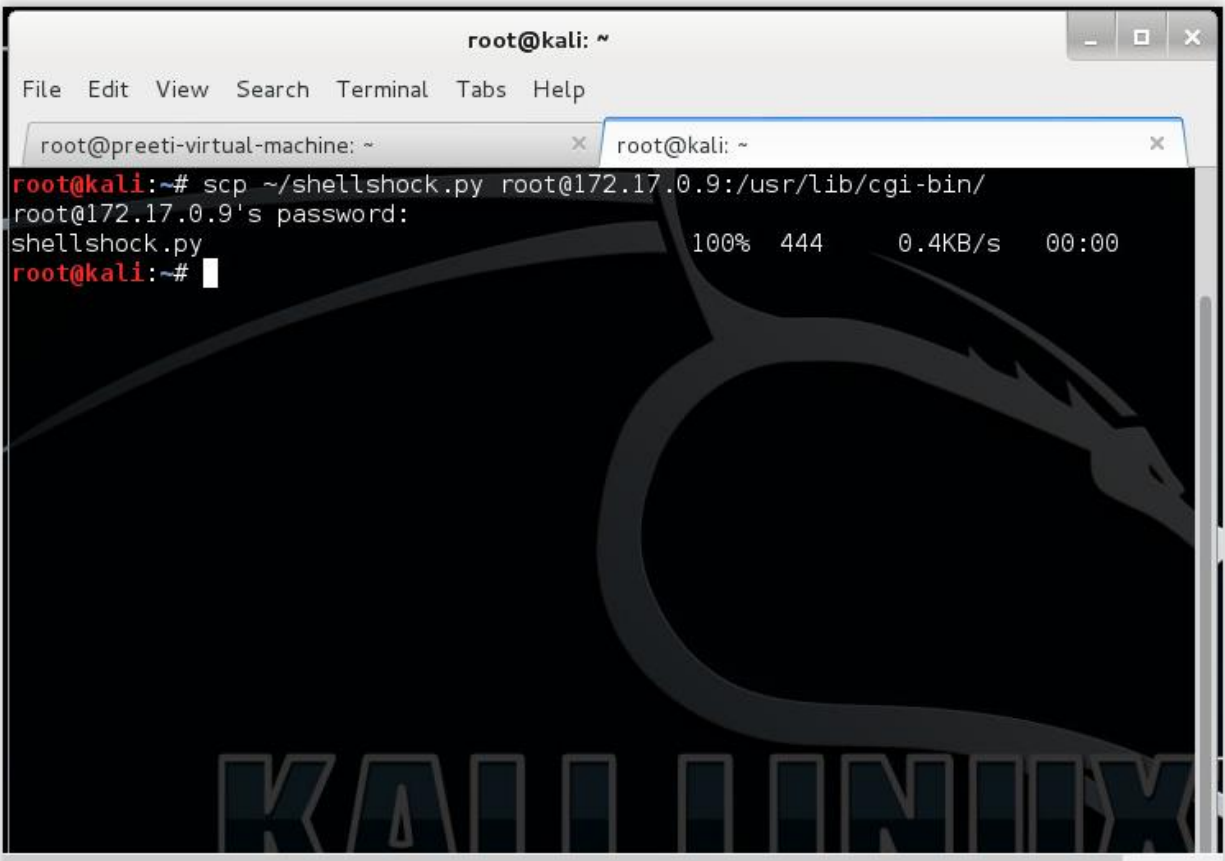


Figure 6: Sending malicious file to the web server

Open a new terminal tab by typing [Ctrl] + [Shift] + [T].

In the new terminal tab, type the command and press [Enter]

```
scp ~/shellshock.py root@172.17.0.9:/usr/lib/cgi-bin/
```

You may need to wait for some time before it prompts for the password and enter password as root123. The output of this command is as shown in Figure 6.

Switch to the other tab where you had established an SSH connection and type

```
chmod 755 /usr/lib/cgi-bin/shellshock.py
```

4. Switch back to the browser window. Let us see if we can execute bash commands remotely via a cgi-bin script. Shellshock.py has a script which will exploit the shellshock vulnerability by setting the environment variable in the shell to the value passed in the query parameter. Hence we can find out all the files in cgi-bin by passing ls as the query parameter.

In the browser window, type '172.17.0.9/cgi-bin/shellshock.py?COMMAND=ls' in the URL field as shown in Figure 7.

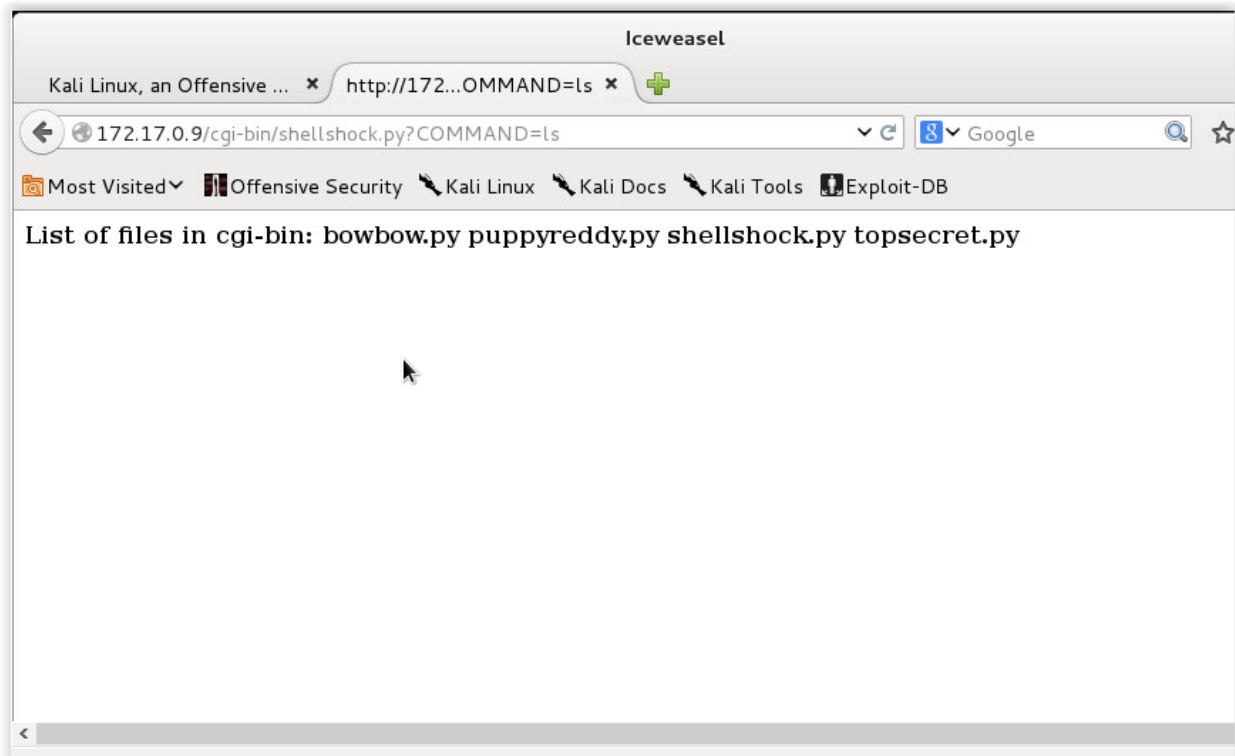


Figure 7. Listing files in cgi-bin directory

The attacker knowing all the files that are present in the cgi-bin of the web server is potentially dangerous because cgi-bin generally has executable scripts and the attacker can potentially cause a lot of damage with the information.

Type exit in both the tabs of the terminal and close the Terminal Window and the Browser Window after you have performed all the steps.

2. INSTALLING LYNIS ON HOST MACHINE:

You are now Luke, the administrator and you have heard about a tool called Lynis which is used for hardening systems and decide to use it.

In this section, let us see how you would go about checking if the web server is secure.

From the machine farm pane on the left, right click on C3PO, click on Snapshot Manager and then click on C3PO_FT1 and 'Go To' to load the lab environment. Click on 'Yes' in the Warning box. Use the credentials given below.

Username: preeti

Password: aiarocks

Lynis can be downloaded as a tarball. The required tarball lynis-2.1.0.tar.gz is present in the Downloads folder.

Step1. Open a terminal window from the launcher. If the terminal is not present in the launcher, click on Dash Home and search for Terminal in the Search bar. In the terminal, from the Home directory, change to Downloads directory by typing

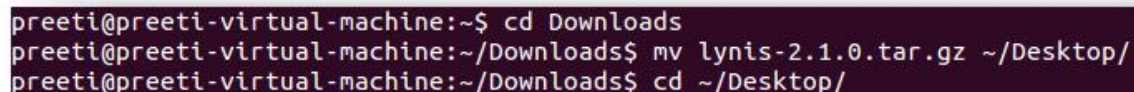
```
cd Downloads
```

Step2. Move the file lynis-2.0.1.tar.gz to the Desktop by typing the command

```
mv lynis-2.0.1.tar.gz ~/Desktop/
```

Step3. Change to the Desktop directory using the below command.

```
cd ~/Desktop/
```



```
preeti@preeti-virtual-machine:~$ cd Downloads
preeti@preeti-virtual-machine:~/Downloads$ mv lynis-2.1.0.tar.gz ~/Desktop/
preeti@preeti-virtual-machine:~/Downloads$ cd ~/Desktop/
```

Figure 8: Accessing the Lynis zip file

Step 4. It is always good practice to check the integrity of the file that is downloaded. Since we will be installing Lynis on the host machine, it is better to ensure that we have installed the untampered version of Lynis. So, we now check the SHA256 hash of the tar file that we downloaded with the hash value present in the Lynis website.

In the Desktop directory, type the following command in the terminal

```
sha256sum lynis-2.1.1.tar.gz > ~/hash.txt
```

This command calculates the SHA256 hash of the Lynis tarball and saves the output in hash.txt in the home directory.

Step 5. Now, we navigate to the home directory. We then compare the hash calculated and stored in hash.txt with the hash taken from the Lynis website stored in lynis-hash.txt.

Type the following commands in the terminal

```
cd ~
```

```
diff hash.txt lynis-hash.txt
```

If the output of the diff command is blank, it means that the two files' contents are identical. If the two hashes do not match, in a normal scenario, one would try downloading the tarball again.

Step 6. Now that we have verified the integrity of the Lynis tarball that we have, let us install Lynis, in the terminal, type

```
cd ~/Desktop
```

Step 7. Extract files from the tarball lynis-0.0.tar.gz by typing this command in the terminal window.

```
tar xvzf lynis-2.1.0.tar.gz
```

Figure 9 below shows the start of the extraction

```
preeti@preeti-virtual-machine:~/Desktop$ tar xvzf lynis-2.1.0.tar.gz
lynis/CHANGELOG
lynis/CONTRIBUTORS
lynis/FAQ
lynis/INSTALL
lynis/LICENSE
lynis/README
lynis/db/
lynis/db/integrity.db
lynis/db/sbl.db
lynis/db/fileperms.db
lynis/db/malware-susp.db
lynis/db/malware.db
lynis/db/hints.db
lynis/default.prf
lynis/extras/
```

Figure 9: Extracting the tarball, beginning phase

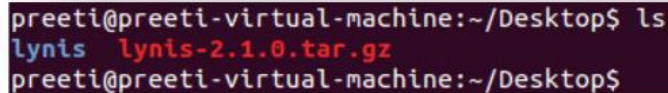
The last portion of extracting from tarball should look like the output in Figure 10.

```
lynis/include/tests_logging
lynis/include/tests_mail_messaging
lynis/include/tests_banners
lynis/include/tests_crypto
lynis/include/tests_kernel
lynis/include/tests_mac_frameworks
lynis/include/tests_solaris
lynis/include/tests_virtualization
lynis/include/tests_kernel_hardening
lynis/include/tests_snmp
lynis/include/tests_authentication
lynis/include/tests_filesystems
lynis/include/tests_storage
lynis/include/data_upload
lynis/include/tests_printers_spools
lynis/include/tests_php
lynis/include/consts
lynis/include/tests_tcpwrappers
lynis/lynis
lynis/lynis.8
lynis/plugins/
lynis/plugins/README
lynis/plugins/custom_plugin.template
preeti@preeti-virtual-machine:~/Desktop$
```

Figure 10: Results obtained on extracting the file lynis-2.1.0.tar.gz

Step 8: Upon listing the contents of that directory, we see that there is a new directory called 'lynis'. Type the following command in the terminal to list the contents

```
ls
```



```
preeti@preeti-virtual-machine:~/Desktop$ ls
lynis  lynis-2.1.0.tar.gz
preeti@preeti-virtual-machine:~/Desktop$
```

Figure 11: The contents of Desktop directory

Step 9: Then type the following command in the terminal within the Desktop directory and press the [Enter] key

```
cd lynis
```

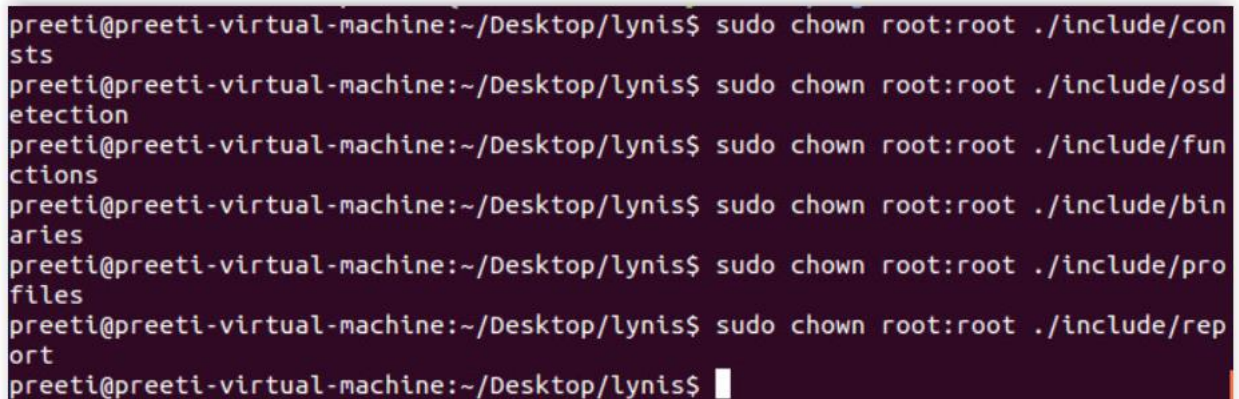
Step 10: We need to give root access to some of the elements within Lynis. Execute the following commands and if prompted for the password, enter 'aiarocks' and press [Enter] .

Hint: You can use the [up arrow] to avoid typing

```
sudo chown root:root ./include/consts
sudo chown root:root ./include/osdetection
sudo chown root:root ./include/functions
sudo chown root:root ./include/binaries
sudo chown root:root ./include/profiles
sudo chown root:root ./include/report
```

The output on executing each of the above commands is shown in the figure below.

Note: Observe that nothing would be printed in the screen



```
preeti@preeti-virtual-machine:~/Desktop/lynis$ sudo chown root:root ./include/consts
preeti@preeti-virtual-machine:~/Desktop/lynis$ sudo chown root:root ./include/osdetection
preeti@preeti-virtual-machine:~/Desktop/lynis$ sudo chown root:root ./include/functions
preeti@preeti-virtual-machine:~/Desktop/lynis$ sudo chown root:root ./include/binaries
preeti@preeti-virtual-machine:~/Desktop/lynis$ sudo chown root:root ./include/profiles
preeti@preeti-virtual-machine:~/Desktop/lynis$ sudo chown root:root ./include/report
preeti@preeti-virtual-machine:~/Desktop/lynis$
```

Figure 12: Changing ownership of some Lynis elements

Now, Lynis is installed and ready to be used.

3. USING LYNIS:

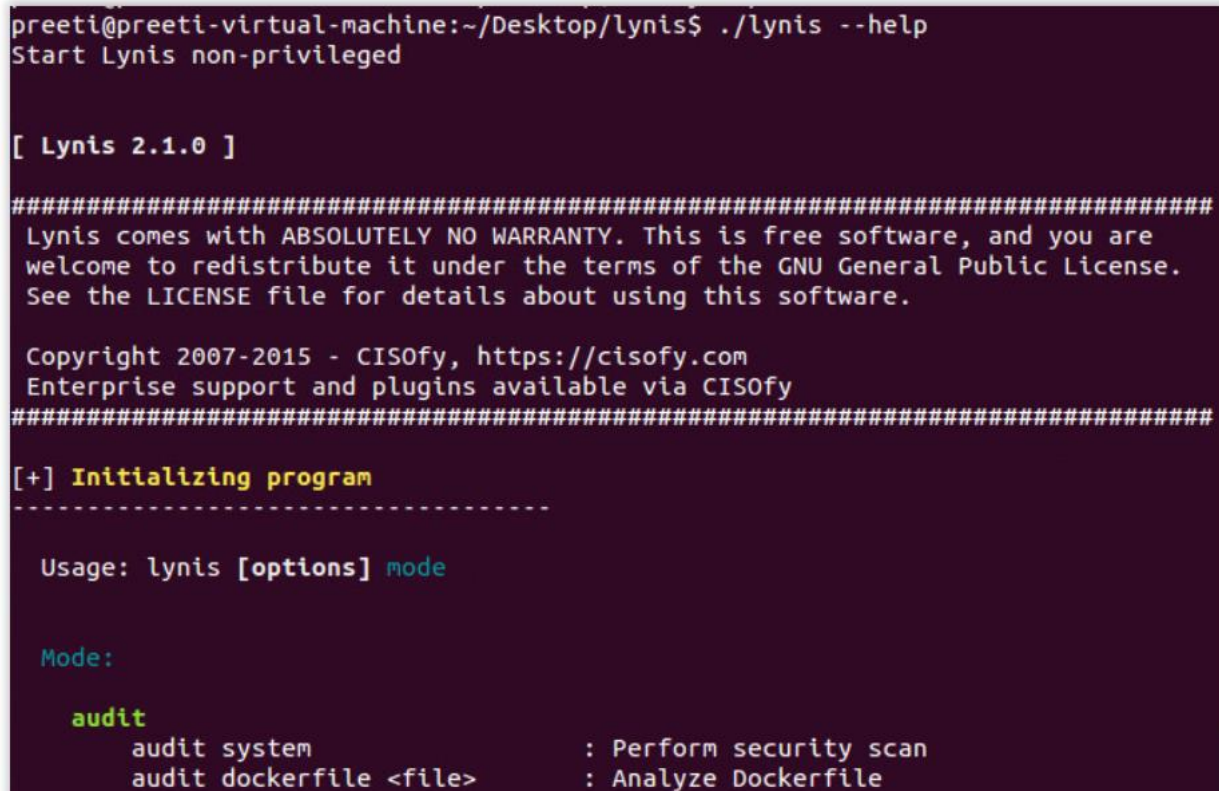
A key aspect of understanding what a tool does and for what purposes it can be used for requires exploring the capabilities. In this section, we shall take baby steps to see how to run the very first Lynis command as a user or as an administrator.

3.1 LYNIS OPTIONS:

1. In the machine C3PO, within the lynis directory, in the terminal, type in the command and press [Enter].

```
./lynis --help
```

The output will be as shown in Figures 13, 14. You need to scroll up to see the complete output. See in the screenshot below that Lynis starts in non- privileged mode.



```
preeti@preeti-virtual-machine:~/Desktop/lynis$ ./lynis --help
Start Lynis non-privileged

[ Lynis 2.1.0 ]

#####
Lynis comes with ABSOLUTELY NO WARRANTY. This is free software, and you are
welcome to redistribute it under the terms of the GNU General Public License.
See the LICENSE file for details about using this software.

Copyright 2007-2015 - CISOfy, https://cisofy.com
Enterprise support and plugins available via CISOfy
#####

[+] Initializing program
-----

Usage: lynis [options] mode

Mode:

  audit
    audit system          : Perform security scan
    audit dockerfile <file> : Analyze Dockerfile
```

Figure 13: Lynis help options without root permission - Part 1


```
Scan options:
  --auditor "<name>"           : Auditor name
  --dump-options               : See all available options
  --no-log                     : Don't create a log file
  --pentest                    : Non-privileged scan (useful for pentest)
  --profile <profile>          : Scan the system with the given profile file
  --quick (-Q)                 : Quick mode, don't wait for user input
  --tests "<tests>"             : Run only tests defined by <tests>
  --tests-category "<category>" : Run only tests defined by <category>

Layout options:
  --no-colors                  : Don't use colors in output
  --quiet (-q)                 : No output, except warnings
  --reverse-colors             : Optimize color display for light backgrounds

Misc options:
  --check-update               : Check for updates
  --debug                      : Debug logging to screen
  --view-manpage (--man)       : View man page
  --version (-V)               : Display version number and quit

Enterprise options:
  --plugin-dir "<path>"         : Define path of available plugins
  --upload                     : Upload data to central node

See man page and documentation for all available options.

Exiting..
preeti@preeti-virtual-machine:~/Desktop/lynis$
```

Figure 14: Lynis help options without root permission - Part 2

2. For an administrator to make the best use of Lynis, it needs to be run with superuser privileges. So, from here onwards, we will run Lynis commands with superuser privileges using the 'sudo' command. Type the following command in the terminal within the lynis directory and press [Enter].

```
sudo ./lynis --help
```

If prompted for a password, please type 'aiarocks' and press [Enter].

The output screen will be as shown in Figures 15, 16. Scroll to view the results as below

```

preeti@preeti-virtual-machine:~/Desktop/lynis$ sudo ./lynis --help

[ Lynis 2.1.0 ]

#####
Lynis comes with ABSOLUTELY NO WARRANTY. This is free software, and you are
welcome to redistribute it under the terms of the GNU General Public License.
See the LICENSE file for details about using this software.

Copyright 2007-2015 - CISOfy, https://cisofy.com
Enterprise support and plugins available via CISOfy
#####

[+] Initializing program
-----

Usage: lynis [options] mode

Mode:

audit
  audit system          : Perform security scan
  audit dockerfile <file> : Analyze Dockerfile

```

Figure 15: Shows the command used to show lynis options

```
Scan options:
  --auditor "<name>"           : Auditor name
  --dump-options               : See all available options
  --no-log                    : Don't create a log file
  --pentest                   : Non-privileged scan (useful for pentest)
  --profile <profile>         : Scan the system with the given profile file
  --quick (-Q)                : Quick mode, don't wait for user input
  --tests "<tests>"             : Run only tests defined by <tests>
  --tests-category "<category>" : Run only tests defined by <category>

Layout options:
  --no-colors                 : Don't use colors in output
  --quiet (-q)                : No output, except warnings
  --reverse-colors            : Optimize color display for light backgrounds

Misc options:
  --check-update              : Check for updates
  --debug                     : Debug logging to screen
  --view-manpage (--man)      : View man page
  --version (-V)              : Display version number and quit

Enterprise options:
  --plugin-dir "<path>"         : Define path of available plugins
  --upload                    : Upload data to central node

See man page and documentation for all available options.

Exiting..
preeti@preeti-virtual-machine:~/Desktop/lynis$ █
```

Fig 16: Various Lynis options

Note: You will need to scroll upwards to see the entire output.

3.2 Test categories in Lynis:

When auditing a server, it may be useful to only run a particular category of tests, like firewall related tests. In that case the `--tests-category` parameter can be used, together with the category name.

To determine what categories are available, Lynis has a built-in parameter `--view-categories` which lists all available files. Most of the names are self-explanatory on what tests they include.

Type the following command in the terminal in order to view the categories of Lynis.

```
sudo ./lynis --view-categories
```

If prompted for a password, please type 'aiarocks' and press [Enter].

The output is shown in Figures 17,18.

```
preeti@preeti-virtual-machine:~/Desktop/lynis$ sudo ./lynis --view-categories
[sudo] password for preeti:

[+] Available test categories
-----
- accounting
- authentication
- banners
- boot_services
- crypto
- databases
- file_integrity
- file_permissions
- filesystems
- firewalls
- hardening
- hardening_tools
- homedirs
- insecure_services
- kernel
- kernel_hardening
- ldap
- logging
- mac_frameworks
- mail_messaging
- malware
- memory_processes
- nameservices
- networking
- php
```

Figure 17: List of available categories - Part 1

```
- ports_packages
- printers_spools
- scheduling
- shells
- snmp
- solaris
- squid
- ssh
- storage
- storage_nfs
- tcpwrappers
- time
- tooling
- virtualization
- webservers

preeti@preeti-virtual-machine:~/Desktop/lynis$
```

Figure 18: List of available categories - Part 2

Note: You will need to scroll upwards to see the entire output.

4. DIFFERENT MODES OF LYNIS:

Lynis can be operated in the following modes Audit mode, Pentest mode and Full Check.

Let us see the operation of Lynis in each of the following modes.

4.1 PENTEST MODE:

Any administrator would be curious to know what details an outsider can find out about our machine. While system administrators and auditors have full access, pentesters will not. Hence running Lynis in pentest mode will give us an idea about what information an administrator is leaking about the machine to an unprivileged outside user. This is where pentest mode in Lynis comes handy. Although we run Lynis in the pentest mode with superuser privileges, Lynis will only report vulnerabilities that a non privileged attacker will be able to take advantage of. Lynis is run in pentest mode by typing the following command in the terminal and pressing [Enter].

```
sudo ./lynis --pentest
```

If prompted for a password, please enter 'aiarocks' and press [Enter]. Please observe the results that are displayed.

The output will be as shown in the Figure 19. The penetration testing mode is generally for exploring the system vulnerabilities assumed as a non-privileged user. At every prompt that says 'press [Enter] to continue or [Ctrl]+C to break', keep pressing the [Enter] key to completely view the Lynis auditing process. Press [Ctrl]+C in case you want to stop the Lynis scan. Once the scan has successfully completed, you should be able to see an output similar to the one in the figure shown below.

Note: Please be patient as it may take some time while testing for System Tools Test and Ports And Packages. If it still takes a long time restart the system and re-execute the pentest command.

```
Quick overview:
- Firewall [X] - Malware scanner [X]

Lynis Modules:
- Heuristics Check [NA] - Security Audit [V]
- Compliance Tests [X] - Vulnerability Scan [V]

Files:
- Test and debug information      : /var/log/lynis.log
- Report data                    : /var/log/lynis-report.dat

=====
Tip: Disable all tests which are not relevant or are too strict for the
     purpose of this particular machine. This will remove unwanted suggestions
     and also boost the hardening index. Each test should be properly analyzed
     to see if the related risks can be accepted, before disabling the test.
=====

I

Lynis 2.1.0
Auditing, hardening and compliance for BSD, Linux, Mac OS and Unix
Copyright 2007-2015 - CISOfy, https://cisofy.com
Enterprise support and plugins available via CISOfy
=====
preeti@preeti-virtual-machine:~/Desktop/lynis$
```

Fig 19: The output of the Lynis scan in pentest mode

4. 2. AUDIT MODE

Auditing is a process of review and analysis of management, operational, and technical controls. The auditor can obtain valuable information about activity on a computer system from the audit trail. Most of the times, Luke doubles as an auditor. With all the overwhelming tasks that he has at hand, it would be great if Luke was presented with a tool which can audit all his systems. The audit mode in Lynis precisely offers this. It performs a thorough system check which encompasses the following:

- System tools
- Boot loaders, startup services
- Kernel: run level, loaded modules, kernel configuration, core dumps
- Memory and processes: zombie processes, IO waiting processes
- Users, groups and authentication: group IDs, sudoers, PAM configuration, password aging, default mask
- File systems: mount points, /tmp files, root file system

- Storage: usb-storage, firewire ohci
- NFS Software: name services: DNS search domain, BIND
- Ports and packages: vulnerable/upgradable packages, security repository
- Software: firewalls: iptables, pf
- Software: Web servers: Apache, nginx
- SSH support: SSH configuration
- SNMP support Databases: MySQL, LDAP services
- Software: php: php options
- Scheduled tasks: crontab/cronjob, atd
- Time and synchronization: ntp daemon
- Cryptography
- Security frameworks
- Software: file integrity, malware scanners
- Home directories: shell history files

Within the lynis directory, execute the following command in the terminal. Replace <auditor name> with an auditor name of your choice in the command (without the Angle Brackets). You can give any name here. In this lab, we have given 'hap' as the auditor name. Refer screenshots for further clarity.

```
sudo ./lynis --auditor <auditor name>
```

If prompted for a password, please type 'aiarocks' and press [Enter].

This is shown in Figures 20, 21. At every prompt to continue, press [Enter] to completely view the Lynis auditing process.

```

aia@aia-virtual-machine:~/Desktop/lynis$ sudo ./lynis --auditor hap
[sudo] password for aia:

[ Lynis 2.0.0 ]

#####
Lynis comes with ABSOLUTELY NO WARRANTY. This is free software, and you are
welcome to redistribute it under the terms of the GNU General Public License.
See the LICENSE file for details about using this software.

Copyright 2007-2015 - CISOfy, https://cisofy.com
Enterprise support and plugins available via CISOfy
#####

[+] Initializing program
-----
- Detecting OS... [ DONE ]

```

Figure 20: The output on running the scan with auditor as the parameter - Part 1

```

[+] File Permissions
-----
- Starting file permissions check
  /etc/lilo.conf [ NOT FOUND ]
  /root/.ssh [ NOT FOUND ]

[ Press [ENTER] to continue, or [CTRL]+C to stop ]

[+] Home directories
-----
- Checking shell history files [ OK ]

[ Press [ENTER] to continue, or [CTRL]+C to stop ]

[+] Kernel Hardening
-----
- Comparing sysctl key pairs with scan profile
  - kernel.core_uses_pid (exp: 1) [ DIFFERENT ]
  - kernel.ctrl-alt-del (exp: 0) [ OK ]
  - kernel.kptr_restrict (exp: 1) [ OK ]
  - kernel.sysrq (exp: 0) [ DIFFERENT ]
  - net.ipv4.conf.all.accept_redirects (exp: 0) [ DIFFERENT ]
  - net.ipv4.conf.all.accept_source_route (exp: 0) [ OK ]
  - net.ipv4.conf.all.bootp_relay (exp: 0) [ OK ]
  - net.ipv4.conf.all.forwarding (exp: 0) [ OK ]

```

Figure 21: The output on running the scan with auditor as the parameter - Part 2

Go through each section of the output on the screen and observe that some of the sections are Kernel, File systems, Storage and so on by scrolling up and down. This shows that the auditing happens per categories as we mentioned in the beginning of this section

4.3. FULL CHECK

In full check mode, we run a comprehensive test of the entire system. It also includes auditing apart from the customized or pre-defined vulnerability scanning tests in the Lynis modules.

From the lynis directory, type the following command in the terminal

```
sudo ./lynis -c
```

If prompted for a password, please type 'aiarocks' and press [Enter].

-c (--check-all) is used to start the check/start the scan process with predefined test cases

This makes Lynis start the scan and the user will be prompted to press [ENTER] or [Ctrl]+C each time. This is shown in the Figure 22. Feel free to go through the screen output at your pace. The goal here was to introduce you to different modes of Lynis. We will be discussing what to infer from this vast output in the next section.



```
[+] Boot and services
-----
- Service Manager [ UNKNOWN ]
- Checking presence GRUB2 [ FOUND ]
- Checking for password protection [ WARNING ]
- Check services at startup (rc2.d) [ DONE ]
  Result: found 15 services
- Check startup files (permissions) [ OK ]

[ Press [ENTER] to continue, or [CTRL]+C to stop ]
```

Figure 22: The output of running a Lynis scan with -c option

We use 'quick' so that the user is spared the trouble of pressing [Enter] or [Ctrl]+C each time.

Type the following command in the terminal

```
sudo ./lynis -c --quick
```

If prompted for a password, please type 'aiarocks' and press [Enter].

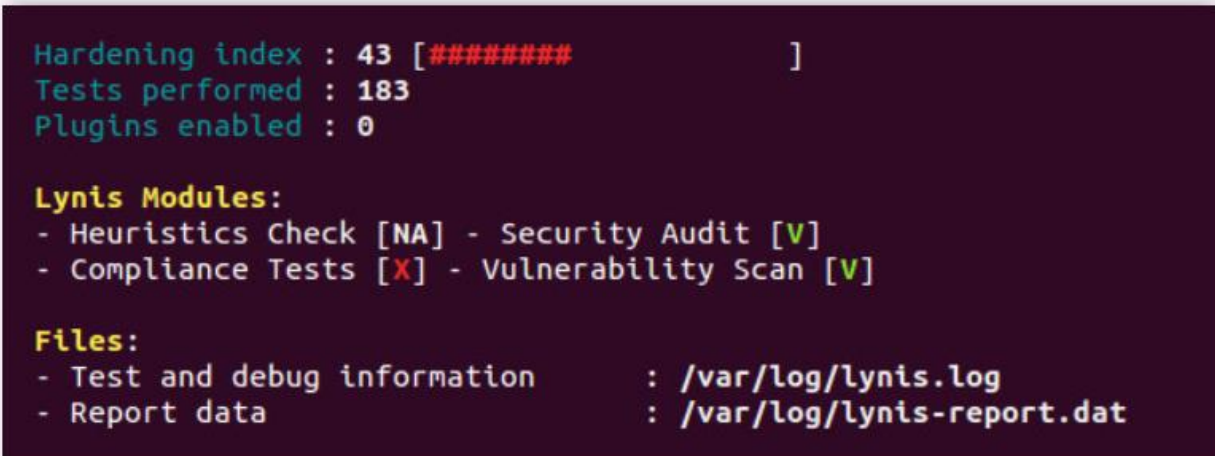
--quick indicates that we need not wait for user input, except on errors.

On running the above command, you will get an output in the same format as shown in Figure 23.

The output of the scan also has information regarding the logs. It gives us the hardening index which is indicative of the security of our system. The greater the value of the hardening index, the more secure our system is said to be.

The result also shows the number of tests that are performed. The system is scanned against the modules present in Security Audit and Vulnerability Scanning which is indicated by the Green 'V' in the screenshot. Since we are not testing the system for Compliance, Compliance Tests have been disabled indicated by 'X'. Heuristics Check is not applicable as indicated by NA.

***Note:** The Hardening Index and the number of tests performed need not be exactly as shown in the figure. You will need to scroll upwards to see the entire output.*



```
Hardening index : 43 [#####]
Tests performed : 183
Plugins enabled : 0

Lynis Modules:
- Heuristics Check [NA] - Security Audit [V]
- Compliance Tests [X] - Vulnerability Scan [V]

Files:
- Test and debug information      : /var/log/lynis.log
- Report data                    : /var/log/lynis-report.dat
```

Fig 23: Figure depicting the Hardening index of the system

From the above section we see that, when we run a lynis scan, lynis performs single target tests and outputs the result of every test performed to the screen.

The auditor or the system administrator should consult the log file and interpret the results. This is done by reading the report file about the technical background (often it contains a suggestion at the test), consult internet sources and documentation about what the impact of the change can be. In the next section we look at analyzing the report and log files.

5. ANALYZING LYNIS REPORTS AND LOGS

Apart from displaying the results on the output screen, every mode in which Lynis runs, generates a log file and a report file. They are:

1. lynis.log
2. lynis-report.dat

respectively. These files are located at **/var/log**. We can see this information under the files section in Figure 23 above.

The following information will help in understanding and analysing the result of a lynis scan better.

Warnings and vulnerabilities are displayed on the output screen. For most tests, Lynis will output [OK] or [WARNING] on the screen. [OK] is considered an expected (good) result, whereas [WARNING] is considered to be deviant from secure practices. If the result says "[OK]", it does not always mean the scanned target is correctly configured securely or according to best practice.

Conversely, every "[WARNING]" doesn't have to be 'bad', since every system (and its requirements) is different. However, as an auditor it is good to pay attention to them and check what influence the warning would have on system or policy. Warnings are of utmost importance, hence they appear on the output screen for every mode Lynis is run in and they also appear in the logs and reports.

Let us use this information to analyze the log files that are generated. We will look at lynis.log first.

Technical details regarding the scan are stored in the log file. The lynis.log file contains detailed information regarding each of the binaries in the system as well as the tests that are performed in each category (as mentioned in section 3.2) and the results of these tests.

Type the following command in the terminal from within the lynis directory in order to view the file lynis.log

```
sudo vim /var/log/lynis.log
```

The contents of the file will be as shown in the Figure 24 below.

```

[07:26:54] ### Starting Lynis 2.1.0 with PID 4933, build date 16 April 2015 ###
[07:26:54] =====
[07:26:54] ### Copyright 2007-2015 - CISOfy, https://cisofy.com ###
[07:26:54] Program version:          2.1.0
[07:26:54] Operating system:         Linux
[07:26:54] Operating system name:    Ubuntu
[07:26:54] Operating system version: 12.04
[07:26:54] Kernel version:           3.13.0
[07:26:54] Kernel version (full):    3.13.0-32-generic
[07:26:54] Hardware platform:        i686
[07:26:54] Hostname:                  preeti-virtual-machine
[07:26:54] Auditor:                   [Unknown]
[07:26:54] Profile:                   ./default.prf
[07:26:54] Log file:                  /var/log/lynis.log
[07:26:54] Report file:               /var/log/lynis-report.dat
[07:26:54] Report version:           1.0
[07:26:54] -----
[07:26:54] Include directory:         ./include
[07:26:54] Plugin directory:          ./plugins
[07:26:54] Database directory:        ./db
[07:26:54] =====
[07:26:54] Checking permissions of ./include/profiles
[07:26:54] File permissions are OK
[07:26:54] Reading profile/configuration ./default.prf
[07:26:54] Profile option set: profile_name (with value Default Audit Template)
[07:26:54] Profile option set: pause_between_tests (with value 0)
[07:26:54] Profile option set: show_tool_tips (with value 1)
[07:26:54] Set option to default value: MACHINE_ROLE --> server
[07:26:54] Set option to default value: NTPD_ROLE --> client
[07:26:54] =====
"/var/log/lynis.log" 5675L, 338099C                                1,1          Top

```

Fig 24: The contents of lynis.log

Figure 24 shows the technical details of the system.

In the log file, let us now look how categories come into play. Let us search for the tests performed under the category Software : file integrity.

To do this, press the [Esc] key and type '/category: Software: file integrity'. Then press [Enter].

Note: Please keep in mind that the commands are case sensitive.

The result is in Figure 25 below which shows some of the tests the tests performed in the category - Software: file integrity.


```

[12:51:29] ===-----=====
[12:51:29] Action: Performing tests from category: Software: file integrity
[12:51:29] ===-----=====
[12:51:29] Performing test ID FINT-4310 (AFICK availability)
[12:51:29] Test: Checking AFICK binary
[12:51:29] Result: AFICK is not installed
[12:51:29] ===-----=====
[12:51:29] Performing test ID FINT-4314 (AIDE availability)
[12:51:29] Test: Checking AIDE binary
[12:51:29] Result: AIDE is not installed
[12:51:29] ===-----=====
[12:51:29] Skipped test FINT-4315 (Check AIDE configuration file)
[12:51:29] Reason to skip: Prerequisites not met (ie missing tool, other type of
Linux distribution)
[12:51:29]

```

Figure 25: Looking at a Category

We see that the tests which check for AFICK and AIDE binaries are performed and the result shows that they have not been installed on the system.

The tests performed under various other categories and their results can also be found in the log file. Feel free to look at the various tests that are performed under each category and explore the results. Since tests are run for every category, more ground is covered and better insights regarding the system are obtained. The information that is collected from the log file can be used to determine weaknesses in unexpected areas.

Apart from this we can also examine Warnings in the file. To do this, press [Esc] then type “/Warning” and press the [Enter] key to find the warnings that Lynis found in the system. Once you find the first warning, press ‘n’ to go to the next warning. Notice that there are three warnings just like the ones in the report file.

Press [Esc] and type :q to exit the log file once you have finished examining the contents.

Now let us analyze the file lynis-report.dat. The lynis-report.dat will show recommendations on how to secure things. It is not as detailed as the lynis.log file and contains the findings(warnings, suggestions and data collections) that the user could follow in order to harden their system.

View the contents of the report by typing the following command from the lynis directory in the terminal.

```
sudo vim /var/log/lynis-report.dat
```

The output of this command will be similar to the output shown in Figure 26.

```

# Lynis Report
report_version_major=1
report_version_minor=0
report_datetime_start=2015-04-26 07:26:54
auditor=[Unknown]
lynis_version=2.1.0
os=Linux
os_name=Ubuntu
os_fullname=Ubuntu 12.04
os_version=12.04
linux_version=Ubuntu
hostname=preeti-virtual-machine
lynis_update_available=0
binary_paths=/bin,/sbin,/usr/bin,/usr/sbin,/usr/local/bin,/usr/local/sbin
binaries_count=1835
vm=1
vmtype=vmware
plugins_enabled=0
hostid=28058e62a28d75f356501bcb10fb208122897314
suggestion[]=BOOT-5122|Set a password on GRUB bootloader to prevent altering boot
configuration (e.g. boot in single user mode without password)|
uptime_in_seconds=1638450
uptime_in_days=18
boot_loader=GRUB2
service_manager=unknown
linux_default_runlevel=2
cpu_pae=1
cpu_nx=1
linux_kernel_release=3.13.0-32-generic
linux_kernel_version=#57-precise1-Ubuntu SMP Tue Jul 15 03:50:54 UTC 2014
"/var/log/lynis-report.dat" 356L, 62557C 1,1 Top

```

Fig 26: The contents of lynis-report.dat

In figure 26, we see that the report contains suggestions as well as general information about the system for example, the hostname, operating system version etc.

Let us examine some of the warnings in this report file. To do this, press [Esc] then type '/warning' and press [Enter] key to find the warnings that Lynis found in the system. Once you find the first warning, press the 'n' key to go to the next warning. Notice that there are three warnings. This will most likely be similar to the warnings found in lynis.log file as we saw previously.

Note: Please keep in mind that the commands are case sensitive.

The file also contains various suggestions that the user could implement in order to improve the security of their system. Feel free to explore the file and examine the suggestions in the file.

To do this, press [Esc] then type '/suggestion' and press [Enter] key to find the suggestions that Lynis found gives regarding the system. Once you find the first warning, press 'n' to go to the next suggestion.

Press [Esc] and type :q to exit the report file once you have finished examining the contents.

Now you are familiar with the different security auditing options of Lynis Tool and how to view the reports and logs that Lynis generates.

6. VULNERABILITY SCANNING WITH LYNIS:

Discovering weaknesses in IT security is named vulnerability scanning. It is the art of finding weaknesses, before malicious people do. These vulnerabilities may exist in essential parts of the operating system, software, or even configuration files.

In this section of the lab, let us explore the vulnerabilities that are reported by Lynis. Once we explore the vulnerabilities, we shall address them on our web server C3PO. Some of these patching procedures will be done on C3PO while a few others will be demonstrated on R2D2 which has the expected final state of the C3PO system after removing all vulnerabilities. If you haven't already logged into C3PO web server, please login using the credential mentioned in Section 1.3. Examine the Warnings generated in Section 4.3 when a full check was executed. If this is not possible, run Lynis again in full check mode as described in the steps below.

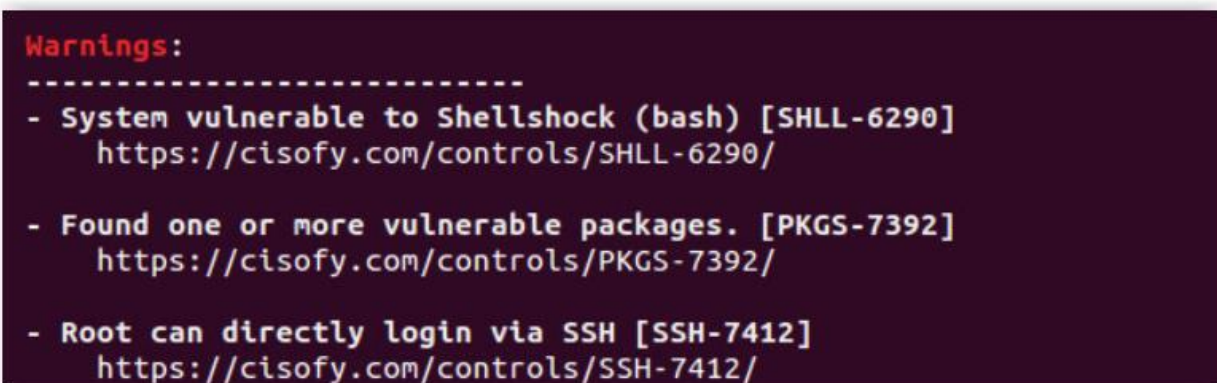
Step 1 : Navigate to the lynis directory by typing the following command in the terminal

```
cd ~/Desktop/lynis
```

Step 2 : Type the command to run Lynis in the check-all mode:

```
sudo ./lynis -c
```

On running the Lynis scan, we can see the warnings on the screen by scrolling upwards. This is shown in the Figure 27.

A terminal window with a dark purple background and light green text. The text displays the output of a Lynis scan, specifically the 'Warnings' section. It lists three vulnerabilities: Shellshock in bash, vulnerable packages, and direct root login via SSH. Each warning includes a reference URL from cisofy.com.

```
Warnings:
-----
- System vulnerable to Shellshock (bash) [SHLL-6290]
  https://cisofy.com/controls/SHLL-6290/

- Found one or more vulnerable packages. [PKGS-7392]
  https://cisofy.com/controls/PKGS-7392/

- Root can directly login via SSH [SSH-7412]
  https://cisofy.com/controls/SSH-7412/
```

Figure 27: The warnings generated on running a Lynis scan

Step 3 : Make a note of the hardening index which would have a value similar to that in Figure 28.

```
Hardening index : 43 [#####  
Tests performed : 183  
Plugins enabled : 0  
  
Lynis Modules:  
- Heuristics Check [NA] - Security Audit [V]  
- Compliance Tests [X] - Vulnerability Scan [V]  
  
Files:  
- Test and debug information      : /var/log/lynis.log  
- Report data                    : /var/log/lynis-report.dat
```

Figure 28: The hardening index and file information displayed as part of the lynis scan

6.1 OVERCOMING VULNERABILITIES:

Let us tackle the warnings shown in the figure above systematically

VULNERABILITY 1: Root can directly login via SSH

One of the biggest security holes you could open on your server is to allow directly logging in as root through ssh, because any cracker can attempt to brute force your root password and potentially get access to your system if they can figure out your password.

PATCH PROCEDURE:

1. Open the file sshd_config by typing the command in the terminal from the lynis directory

```
sudo vim /etc/ssh/sshd_config
```

If prompted for a password, please type 'aiarocks' and press [Enter].

The file looks as follows:

```

# See the sshd_config(5) manpage for details

# What ports, IPs and protocols we listen for
Port 22
# Use these options to restrict which interfaces/protocols sshd will bind to
#ListenAddress ::
#ListenAddress 0.0.0.0
Protocol 2
# HostKeys for protocol version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes

# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 768

# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 120
PermitRootLogin yes
StrictModes yes

"/etc/ssh/sshd_config" 87L, 2489C 1,1 Top

```

Figure 29: The contents of sshd_config

Once the file opens, press [Esc] key and search for the text PermitRootLogin by typing '/PermitRootLogin'. Press [Insert] to go to insert mode.

Once you find the line, comment out the line by putting a '#' symbol at the beginning of the line. After making the changes, the changed part of the file should look as follows

```

# Authentication:
LoginGraceTime 120
#PermitRootLogin yes
StrictModes yes

```

Figure 30: Commenting out the line PermitRootLogin in the file sshd_config

Press [Esc] key and type :wq and press [Enter] to save your changes and quit.

As an extra precaution, let us disable the root account as well. Type the following command in the terminal

```
sudo passwd -dl root
```


If prompted for a password, please type 'aiarocks' and press [Enter]

Password expiry information changed would be the output.

Then type the following command in the terminal

```
sudo service ssh restart
```

If prompted for a password, please type 'aiarocks' and press [Enter].

This restarts the ssh service with the changes that we incorporated and the output mentions the SSH process which is running.

VULNERABILITY 2: System vulnerable to Shellshock

Shellshock:

The issue, which is currently being called the "Bash" bug or "Shellshock," was discovered by the security team at software company Red Hat. The Bash bug is a vulnerability that can allow a hacker to issue remote commands to web servers. Since the bug makes it possible for a hacker to tell a server to do anything they want, there's a risk that confidential information can easily be stolen from affected servers. The exploit affects servers and systems that use a language interpreter called Bash to process commands. Linux, UNIX and MAC OS commonly use bash. Hence a lot machines are affected. Servers using the bash shell (or interpreter) are only vulnerable if they're capable of passing commands remotely over the internet which makes it relevant in the context of web servers.

Why It's Dangerous:

Bash allows hackers to remotely execute commands. Just like what we saw the attacker machine (Emperor) do, If a hacker can identify a vulnerable server, they can perform a number of different types of requests. The most important one is to setup a reverse shell and can get that server to send back information from it via network. One of the biggest problems, however, is that many systems that don't get updated regularly won't receive the necessary patch to fix the vulnerability. This can include things like routers, which aren't updated very frequently.

What You Can Do To Be Safe:

The complete patch for this bug is still underway, but there are some precautions we can take some of them being to make the servers offline without it posing too much of a risk to live systems. Using a firewall for your website might help, but there's no guarantee. Also, administrators should look through the server logs to make sure no suspicious commands have been given.

The U.S.-CERT's advisory includes a simple command line script that users can run to test for the vulnerability. Let us write a shell script that can check if the machine is vulnerable to shell shock.

Ensure that you are still on the C3PO machine.

Step 1. Make a new directory called scripts in the home directory by typing the following command in the terminal:

```
mkdir ~/scripts
```

Step 2. Navigate to the directory using

```
cd ~/scripts
```

Step 3. From within the scripts directory, type the following command in the terminal and press the [Enter] key to make a shell script called check_shellshock.sh.

```
vim check_shellshock.sh
```

Step 4. Once the vim editor opens, press the 'i' key or [Insert] to access the insert mode and type the following into the script:



```
#!/bin/bash
env x='() { :}; echo vulnerable' bash -c "echo this is a test"
```

Figure 31: Contents of check_shellshock.sh

Step 5. Press [Esc] and type :wq to save and exit the script.

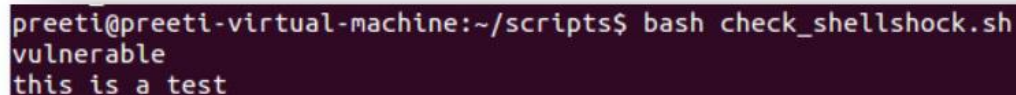
Step 6. Now, we need to provide execute permission to the script. We do this by typing the following command in the terminal

```
sudo chmod +x check_shellshock.sh
```

We then run the script using the following command in the terminal

```
bash check_shellshock.sh
```

The output will be as shown in Figure 32 below.



```
preeti@preeti-virtual-machine:~/scripts$ bash check_shellshock.sh
vulnerable
this is a test
```

Figure 32: Running check_shellshock.sh

This shows that the machine is vulnerable to Shell Shock attack.

PATCH PROCEDURE:

R2D2 machine has some previously installed packages which would help us in addressing these vulnerabilities as we are limited by lack of Internet connection to get the packages. To patch the bash shell to prevent ShellShock, the following commands are run from the terminal.

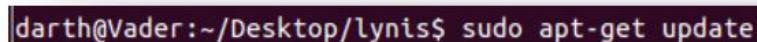
Note: Steps 1 and 2 need not be run and are just given for reference. These have already been executed on R2D2 because internet connection would not available in the lab environment to execute these commands

Step 1. Make sure the list of packages and dependencies from all repositories and Personal Package Archives are up to date. This can be done by using the command shown below

```
sudo apt-get update
```

If prompted for a password, please type 'aiarocks' and press [Enter].

As shown in Figure 33 below.



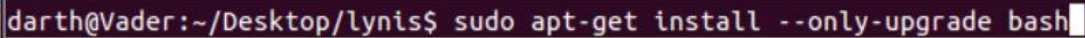
```
darth@Vader:~/Desktop/lynis$ sudo apt-get update
```

Fig 33: Updating list of packages and dependencies

Step 2. The bash that is vulnerable to Shell Shock can be upgraded using the following command

```
sudo apt-get install --only-upgrade bash
```

This is shown in Figure 34 below. The command is used to upgrade bash.



```
darth@Vader:~/Desktop/lynis$ sudo apt-get install --only-upgrade bash
```

Figure 34: Upgrading bash

VULNERABILITY 3: Found one or more vulnerable packages

This is the only warning which needs to be addressed now. If you are not logged into the C3PO machine, login again using the following credentials

Username: preeti

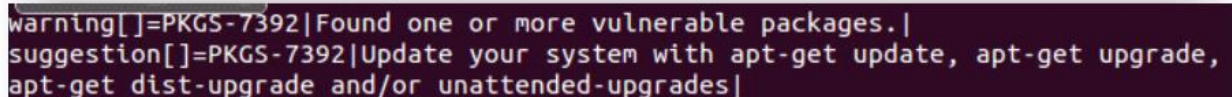
Password: aiarocks

Open a terminal window and examine the log file /var/log/lynis.log on C3PO by typing

```
sudo cat /var/log/lynis.log | grep "PKGS-7392"
```

If prompted for a password, please type 'aiarocks' and press [Enter].

We should be able to see the recommendation made by Lynis which is shown in Figure 35 below.



```
warning[]=PKGS-7392|Found one or more vulnerable packages.|  
suggestion[]=PKGS-7392|Update your system with apt-get update, apt-get upgrade,  
apt-get dist-upgrade and/or unattended-upgrades|
```

Figure 35: Lynis warning for vulnerable packages

PATCH PROCEDURE:

Note: The following commands have already been executed on R2D2 because internet connection would not available in the lab environment to execute these commands.

The vulnerable packages need to be upgraded. As suggested by Lynis, we can resolve this warning by running the following commands:

```
sudo apt-get update  
sudo apt-get upgrade  
sudo apt-get dist-upgrade  
sudo unattended-upgrades
```

If prompted for a password, please type 'aiarocks' and press [Enter].

7. VERIFICATION:

In this section of the lab let us verify if the patches that we made in the previous sections have been effective

VERIFICATION 1: Root can directly login via SSH

We completed the patch on C3PO. Hence to verify, log into C3PO if you are not logged in yet using the following credentials

Username: preeti

Password: aiarocks

Perform the following steps:

Step 1. Open a terminal window if you closed it in the previous sessions and navigate to the lynis directory typing the following command:

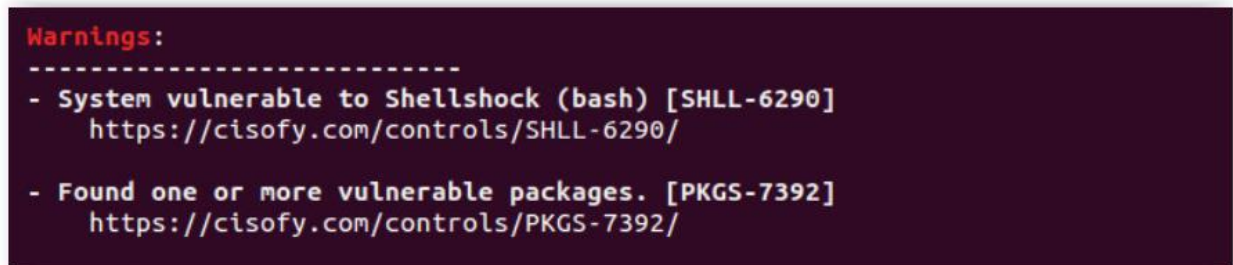
```
cd ~/Desktop/lynis
```

Step 2. Run Lynis using the command

```
sudo ./lynis -c --quick
```

If prompted for a password, please type 'aiarocks' and press [Enter].

Scroll down to the warnings section. On comparing the results from the last Lynis run to this one, we notice that the warning corresponding to the root login ssh is no longer reported as we patched the vulnerability. But the other warnings still remain unresolved. This is shown in Figure 36 below.



```
Warnings:
-----
- System vulnerable to Shellshock (bash) [SHLL-6290]
  https://cisofy.com/controls/SHLL-6290/
- Found one or more vulnerable packages. [PKGS-7392]
  https://cisofy.com/controls/PKGS-7392/
```

Figure 36: The warnings that result from the lynis scan on the machine C3PO

VERIFICATION 2: System vulnerable to Shellshock:

From the machine farm pane on the left, right click on R2D2, click on Snapshot Manager and then click on R2D2_FT1 and 'Go To' to load the lab environment. Click on 'Yes' in the Warning box.

Log in to the patched machine R2D2 using the following credentials

Username: Vader

Password: aiarocks

We need to have Lynis set up on this machine. Install Lynis in R2D2 following the steps from section 2.1. Then, we can verify that shellshock vulnerability does not exist by running a script.

Let us write a shell script that can check if the machine is vulnerable to shell shock.

1. Make a new directory called scripts in the home directory by typing the following command in the terminal:

```
mkdir ~/scripts
```

2. Navigate to the directory using


```
cd ~/scripts
```

Once in the directory, create a shell script to check if the system is vulnerable to Shellshock.

3. From within the scripts directory, type the following command in the terminal and press the [Enter] key to make a shell script called check_shellshock.sh.

```
vim check_shellshock.sh
```

4. Once the vim editor opens, press the 'i' key to access the insert mode and type the following into the script:



```
#!/bin/bash
env x='() { :}; echo vulnerable' bash -c "echo this is a test"
```

Figure 37: Contents of check_shellshock.sh

5. Press the [Esc] key and type :wq to save and exit the script.
6. Now, we need to provide execute permission to the script. We do this by typing the following command in the terminal

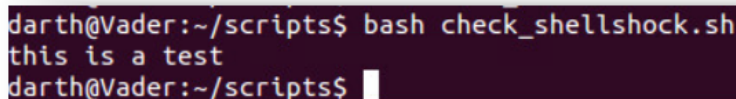
```
sudo chmod +x check_shellshock.sh
```

If prompted for a password, please type 'aiarocks' and press [Enter].

We then run the script using the following command in the terminal

```
bash check_shellshock.sh
```

If the output has the text Vulnerable, then it is susceptible to Shellshock.



```
darth@Vader:~/scripts$ bash check_shellshock.sh
this is a test
darth@Vader:~/scripts$
```

Figure 38: The output on running check_shellshock.sh

We see that the output shown in Figure 39 does not have the text 'Vulnerable' in it. Hence, it is no longer vulnerable to shellshock.

VERIFICATION 3: Found one or more vulnerable packages

Open a terminal window and navigate to the lynis directory typing the following command:

```
cd ~/Desktop/lynis
```

Run the following command in the terminal

```
sudo ./lynis -c --quick
```

If prompted for a password, please type 'aiarocks' and press [Enter].

This time, the output is free of all warnings. This means that our system is safe against Shell Shock. It is also devoid of vulnerable packages.



```
-[ Lynis 2.0.0 Results ]-  
  
No warnings  
  
Suggestions:
```

Fig 39: The output of the lynis scan on R2D2

8. SYSTEM HARDENING BASED ON SUGGESTIONS GIVEN BY LYNIS:

So far, we have explored the vulnerabilities mentioned by Lynis. That apart, Lynis also makes suggestions to improve the security of the system. As seen in the logs and reports, Lynis mentions a lot of suggestions, implementation of which may not be relevant in our context

Let us go through some of those suggestions which are applicable for a web server and try to implement these suggestions as part of system hardening. Login to our web server C3PO using 'preeti' as username and 'aiarocks' as password. Go through the log files again for suggestions. The logs can be found at /var/log/lynis.log. The suggestions can be found by using grep tool.

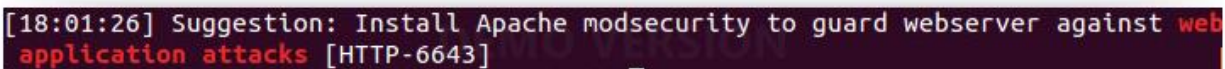
SUGGESTION 1: APACHE-MODSECURITY:

In the terminal, type the following command and press [Enter].

```
sudo cat /var/log/lynis.log | grep "web application attack"
```

If prompted for a password, please type 'aiarocks' and press [Enter].

You should be able to see results as shown in Figure 40 below



```
[18:01:26] Suggestion: Install Apache modsecurity to guard webserver against web  
application attacks [HTTP-6643]
```

Fig 40: Output of searching for "web application attack" in the log file.

So, Lynis tells us that the server may be vulnerable to web attacks such as information leakage through errors as demonstrated in Section 1.3 using the Emperor machine. We can prevent this by making changes in the configuration files of modsecurity or by changing security settings in Apache. Let us change the security settings in Apache configuration files. Open a new tab by pushing down on the [Ctrl]+[Shift]+[t] keys at once.

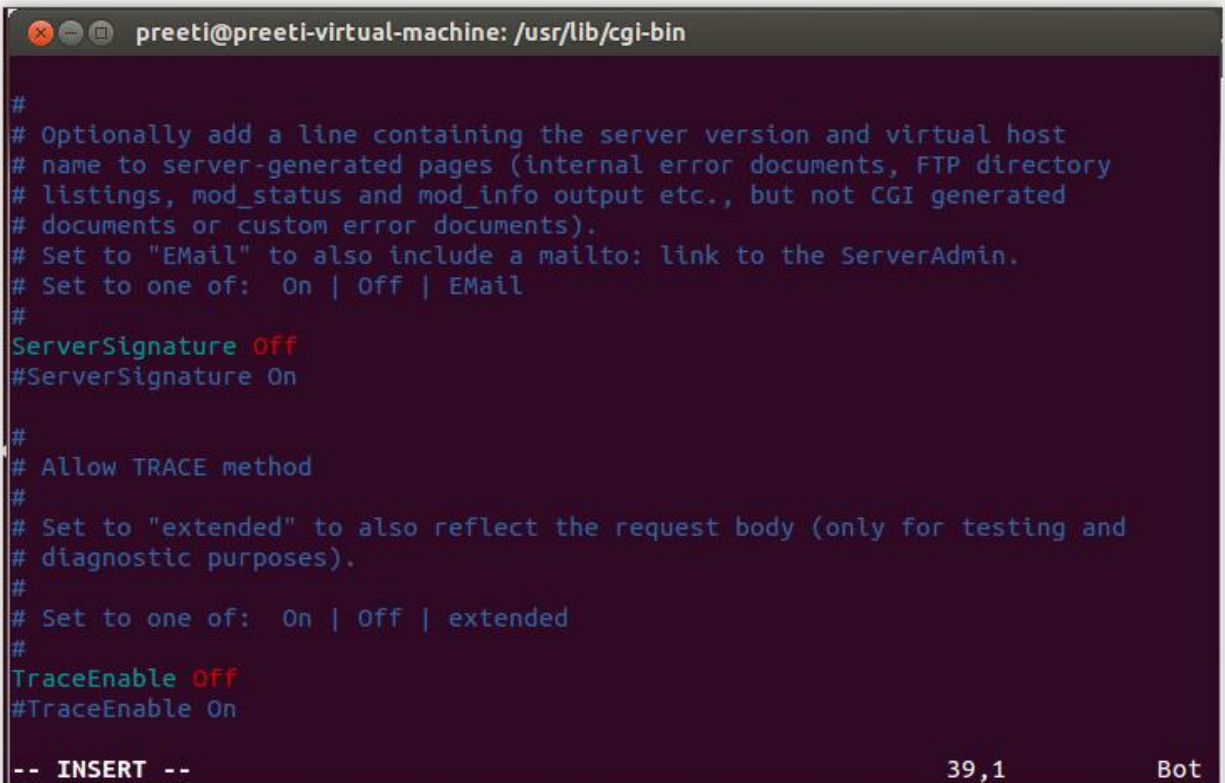
In the terminal, type

```
sudo vim /etc/apache2/conf.d/security
```

If prompted for a password, please type 'aiarocks' and press [Enter].

Within the file search for the word 'Signature' by pressing the [Esc] key and then typing '/Signature'.

Go to insert mode by pressing pressing [Insert]. Uncomment ServerSignature Off by removing '#' and comment ServerSignature On by placing '#' in the beginning of the line. The Final output should look as shown in Figure 41.

A terminal window titled 'preeti@preeti-virtual-machine: /usr/lib/cgi-bin' showing the contents of a file. The text is as follows:

```
#
# Optionally add a line containing the server version and virtual host
# name to server-generated pages (internal error documents, FTP directory
# listings, mod_status and mod_info output etc., but not CGI generated
# documents or custom error documents).
# Set to "EMail" to also include a mailto: link to the ServerAdmin.
# Set to one of:  On | Off | EMail
#
ServerSignature Off
#ServerSignature On

#
# Allow TRACE method
#
# Set to "extended" to also reflect the request body (only for testing and
# diagnostic purposes).
#
# Set to one of:  On | Off | extended
#
TraceEnable Off
#TraceEnable On

-- INSERT --
```

The bottom right corner of the terminal shows '39,1' and 'Bot'.

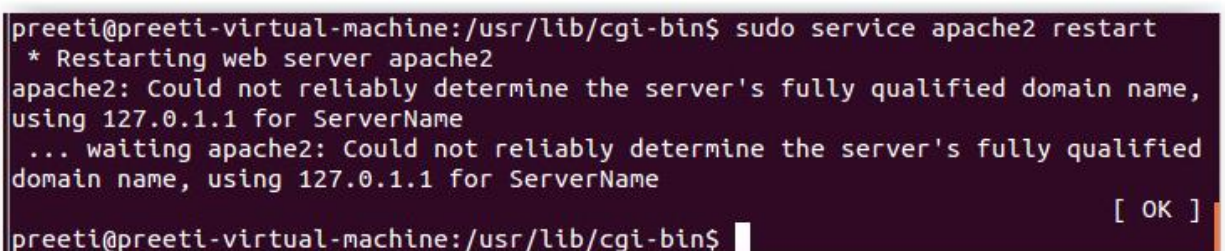
Figure 41: The modified contents of the file 'security'

Press the [Esc] key and then type :wq to save and exit the file.

Once you're back on the terminal type the following command and the output of this command should as shown in Figure 42.

```
sudo service apache2 restart
```

If prompted for a password, please type 'aiarocks' and press [Enter].

A terminal window showing the output of the command 'sudo service apache2 restart'. The text is as follows:

```
preeti@preeti-virtual-machine:/usr/lib/cgi-bin$ sudo service apache2 restart
* Restarting web server apache2
apache2: Could not reliably determine the server's fully qualified domain name,
using 127.0.1.1 for ServerName
... waiting apache2: Could not reliably determine the server's fully qualified
domain name, using 127.0.1.1 for ServerName
[ OK ]
preeti@preeti-virtual-machine:/usr/lib/cgi-bin$
```

Figure 42: The output obtained on restarting the apache server

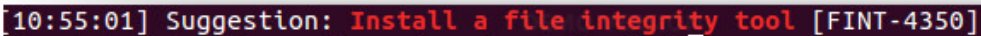
SUGGESTION 2: FILE INTEGRITY TOOL:

In the terminal, type the following command and press [Enter].

```
sudo cat /var/log/lynis.log | grep "Install a file integrity tool"
```

If prompted for a password, please type 'aiarocks' and press [Enter].

You should be able to see results similar to Figure 43.



```
[10:55:01] Suggestion: Install a file integrity tool [FINT-4350]
```

Fig 43: The suggestion in the lynis.log file to install a file integrity tool.

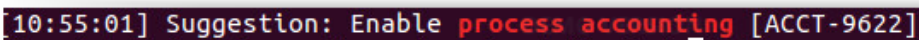
The suggestion shown in the log was that a File Integrity Tool should be present on the system, such that data integrity is preserved.

SUGGESTION 3: PROCESS ACCOUNTING:

In the terminal, type the following command and press [Enter].

```
sudo cat /var/log/lynis.log | grep "process accounting"
```

If prompted for a password, please type 'aiarocks' and press [Enter].



```
[10:55:01] Suggestion: Enable process accounting [ACCT-9622]
```

Fig 44: The suggestion in the lynis.log file to enable process accounting.

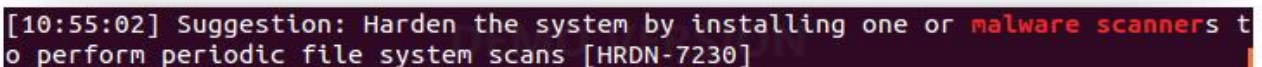
Lynis suggests that there should be a process accounting tool so that we can keep track of processes running on the system.

SUGGESTION 4: MALWARE REMOVER:

In the terminal type the following command and press [Enter].

```
sudo cat /var/log/lynis.log | grep "malware scanner"
```

If prompted for a password, please type 'aiarocks' and press [Enter].



```
[10:55:02] Suggestion: Harden the system by installing one or malware scanners to perform periodic file system scans [HRDN-7230]
```

Fig 45: The suggestion in the lynis.log file to install a malware scanner.

Lynis recognizes that there is no malware scanning tool and suggests that a malware scanner should be installed.

SUGGESTION 5: FIREWALL

In the terminal type the following command and press [Enter].

```
sudo cat /var/log/lynis.log | grep "a firewall/packet filter"
```

If prompted for a password, please type 'aiarocks' and press [Enter].

```
[18:15:03] Suggestion: Configure a firewall/packet filter to filter incoming and outgoing traffic [FIRE-4590]
```

Fig 46: The suggestion in the lynis.log file to configure a firewall.

Lynis recognizes that there is no firewall and suggests that the administrator should configure a firewall.

8.1 VERIFICATION OF SYSTEM HARDENING:

Let us address some of these suggestions one by one.

Log onto R2D2 with the username 'Vader' and password 'aiarocks'.

In the terminal, type the following command to [Enter] the lynis directory.

```
cd Desktop/lynis
```

From within the lynis directory, run lynis by typing the command

```
sudo ./lynis --auditor <auditor-name> --quick
```

You can mention any name as the auditor. If prompted for a password, please type 'aiarocks' and press [Enter].

When we run Lynis, we see that the suggestions pertaining to File Integrity, websecurity, process accounting and antivirus are not present in the log file. We can do this by typing the following commands at the command prompt

```
sudo cat /var/log/lynis.log | grep "web application attack"
sudo cat /var/log/lynis.log | grep "Install a file integrity
tool"
sudo cat /var/log/lynis.log | grep "process accounting"
sudo cat /var/log/lynis.log | grep "a firewall/packet filter"
```

If ever prompted for a password, please type 'aiarocks' and press [Enter].

We see that the search is unsuccessful and there is nothing returned for any of these commands.

```
sudo cat /var/log/lynis.log | grep "malware scanner"
```

If prompted for a password, please type 'aiarocks' and press [Enter].

In the output, we see that Lynis recognizes that there is at least 1 malware scanner installed.

To address suggestion 1 from the previous section i.e. to prevent web attacks on the Apache2 Server, we have installed Apache2-modsecurity on R2D2.

In order address suggestion 2, we installed the Tripwire tool which ensures file integrity on R2D2.

In order to address suggestion 3, we installed and configured the tool “Acct” on R2D2. This is done so that we can have process accounting and keep track of all the resources available to a particular process to aid forensic analysis.

In order to address suggestion 4 given by Lynis, Clamav Anti-Virus is running on R2D2.

In this way, we can use Lynis to audit a system, find vulnerabilities and perform host hardening to address the vulnerabilities.

In order to address suggestion 5 given by Lynis, the Ubuntu Firewall has been enabled. It was previously turned off, we have just enabled it. System administrators can further enhance their firewall by implementing specific firewall rules depending on the requirements.

9. COMPARISON OF SYSTEM STATES:

If not already logged in, log into the R2D2 machine using the following credentials and open a terminal window from the dashboard.

Login: Vader

Password: aiarocks

In the terminal in R2D2 machine,

Step 1. Run Lynis again by using the following command

```
sudo ./lynis --auditor hap --quick
```

Note: You can use any name in place of 'hap'.

If prompted for a password, please type 'aiarocks' and press [Enter].

Step 2. Make a note of the hardening index. This will be present in the result of the above scan in

the Lynis Scanner details section.

Compare the hardening index obtained in this run with the hardening index that you had made note of previously i.e., after running the scan on C3PO. The score must have improved significantly.

10. FINAL VERIFICATION

Now that we have hardened our systems and minimized vulnerabilities, let us see if the Emperor can still carry out the attacks shown in Section 1.3

Login to Emperor using the credentials given below

Username: root

Password: aiarocks

1. Let us see if the web server functionality is still intact.

2. If not already opened, open Iceweasel browser by double clicking on the Weasel and globe icon seen on the top left section in the menu bar.

3. Open a new tab by clicking on the green colored plus button. In the website address bar, type 172.17.0.9/team1.html. You should be able to see a webpage served by C3PO as shown in Figure 47. We see that the webpage is still intact.

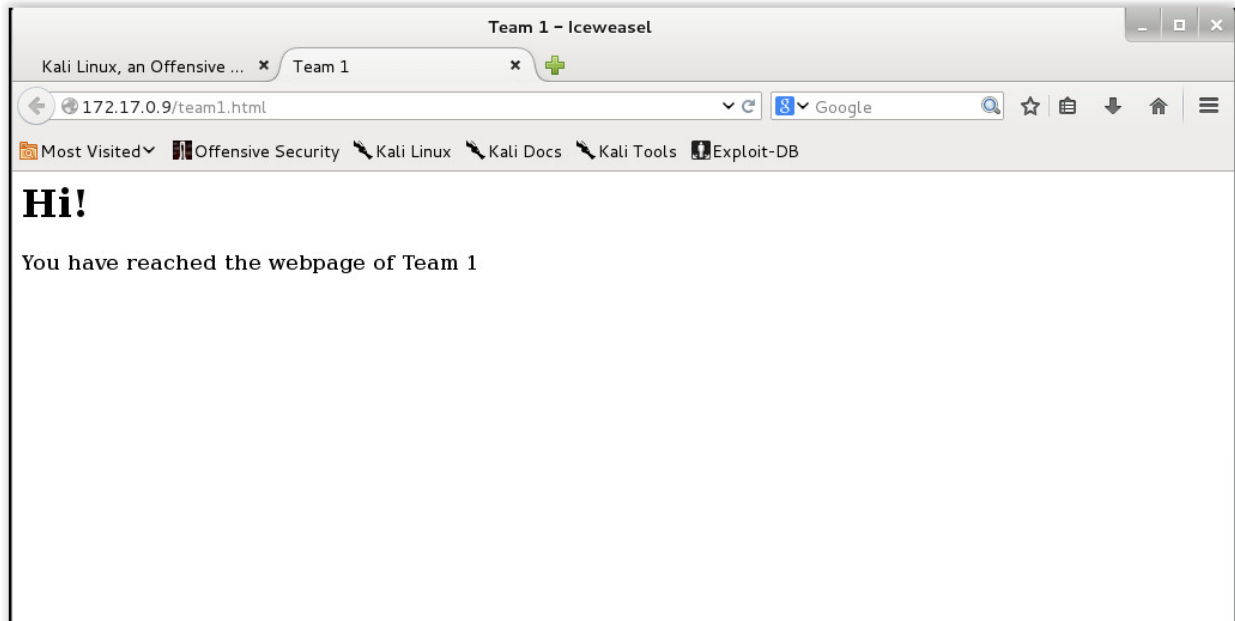


Figure 47: Checking Web-server functionality

4. Now, however if we give an invalid page by typing 172.17.0.9/team2.html, notice that there is no information leakage in the form of errors. Hence the attacker cannot gain any information about the server through information leaks. This is shown in Figure 48.

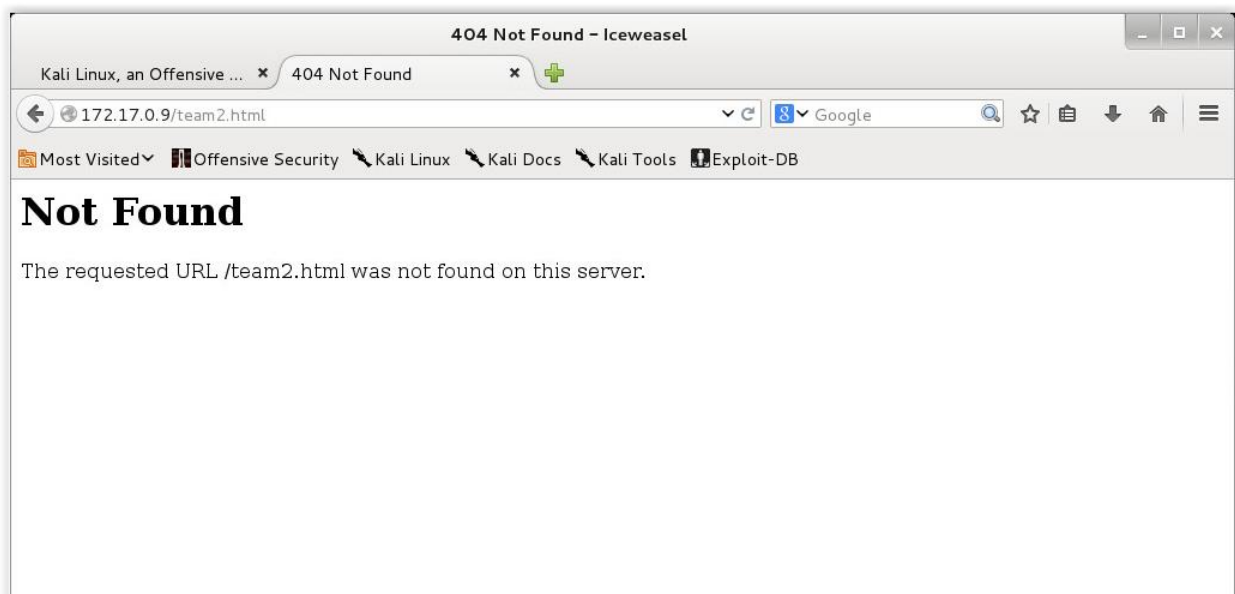


Figure 48: Demonstrates that there is no leakage about the server

5. Now, let us try to establish an SSH connection to the web server as root. Minimize the browser and open a terminal window from the top left of the menu bar if not already opened. In the terminal, perform the following:

```
ssh root@172.17.0.9
```

When prompted for a password, enter 'root123'. Permission is denied because we disabled root from accessing via SSH.

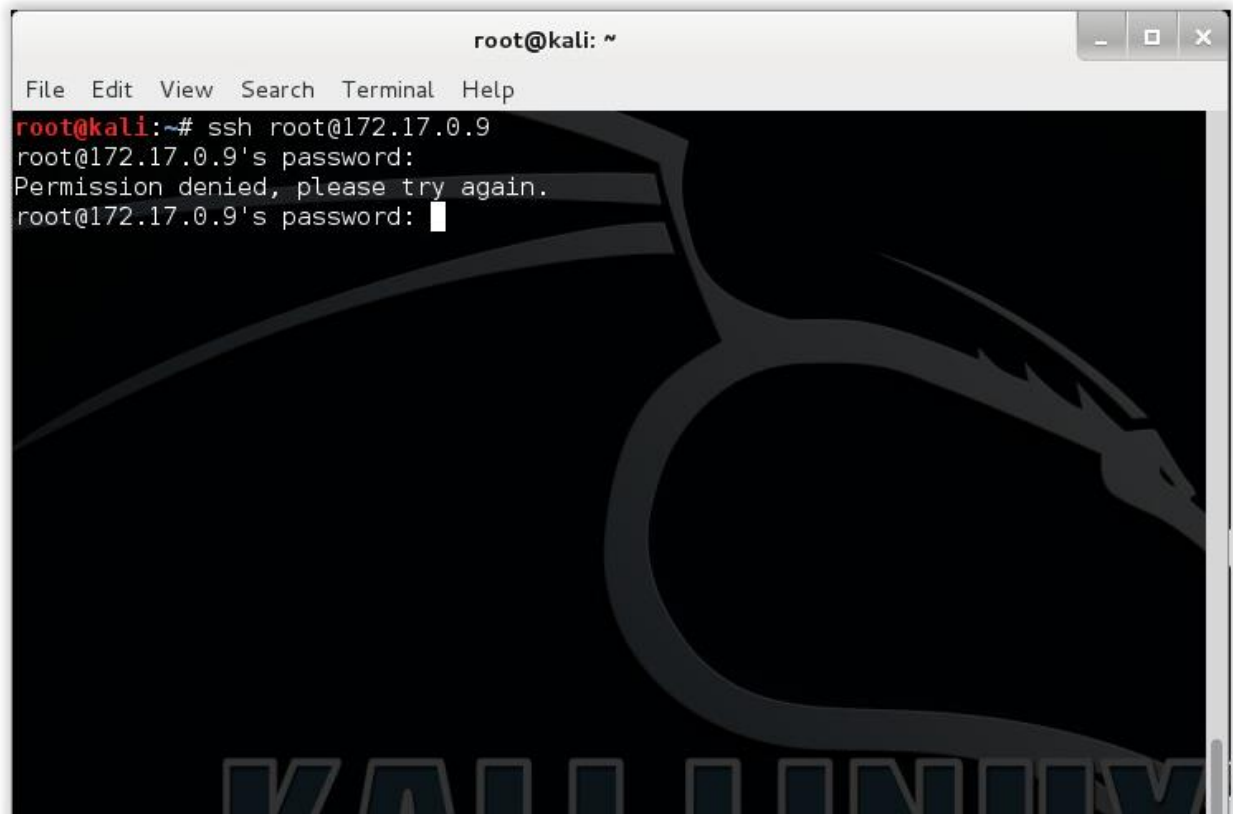


Figure 49: Demonstrates that root access is denied to the web server via ssh.

References:

1. <https://cisofy.com/documentation/lynis/>
2. <http://www.maketecheasier.com/audit-linux-security-with-lynis/>
3. <http://www.howtogeek.com/howto/linux/security-tip-disable-root-ssh-login-on-linux/>
4. <http://ubuntuguide.net/install-and-enable-telnet-server-in-ubuntu-linux>
5. <http://krebsonsecurity.com/2014/09/shellshock-bug-spells-trouble-for-web-security/>
6. [https://cisofy.com/documentation/lynis/\\$installation-package](https://cisofy.com/documentation/lynis/$installation-package)

7. <http://notes.sagredo.eu/node/36>
8. http://wiki.opensource-excellence.com/index.php?title=How_to_install_ClamAV
9. <https://help.ubuntu.com/community/OpenSSL>
10. <http://www.kitploit.com/2014/08/lynis-159-security-auditing-tool-for.html>
11. <http://linux-audit.com/viewing-available-test-categories-in-lynis/>