CEH v12 Lesson 5 : Vulnerability Assessment Tools and Techniques

Learning Outcomes

In this module, you will complete the following exercises:

- Exercise 1 — Vulnerability Assessment Concepts
- Exercise 2 — Vulnerability Classification and Assessment Types
- Exercise 3 — Vulnerability Assessment Solutions and Tools
- Exercise 4 — Vulnerability Assessment Reports

After completing this module, you will be able to:

- Use Nikto for Vulnerability Scanning
- Use Lynis for System Vulnerability Scanning
- Create a Formatted Report with Nikto

After completing this module, you will have further knowledge of:

- Vulnerability and Vulnerability Assessment
- The Need for Vulnerability Research
- Mapping Vulnerabilities
- Vulnerability Management Lifecycle
- Common Vulnerabilities and Exposures (CVE)/Common Vulnerability Scoring System (CVSS)
- False Negative / False Positive
- Exploits
- Vulnerability Classification
- Types of Vulnerability Assessments

- Specialized Vulnerability Assessments
- Vulnerability Classification
- Types of Vulnerability Assessments
- Specialized Vulnerability Assessments
- Vulnerability Assessment Approaches
- Vulnerability Assessment Process
- Types of Vulnerability Assessment Tools
- The Vulnerability Assessment Report

Lab Duration

It will take approximately **1 hour** to complete this lab.

Exercise 1 — Vulnerability Assessment Concepts

A vulnerability is a weakness in a system that could be exploited and may be a defect or a bug present in software or part of an application's configuration. Vulnerability scans can list out said identified vulnerabilities before they can be exploited by an attacker.

To keep a watch on the network's health, every organization runs vulnerability scans regularly. Depending on various factors, the interval between the scans can differ for each organization. For example, an organization with thousands of servers may decide to run a vulnerability scan every six months. However, a smaller organization may run it every quarter — it entirely depends on the organization and their own risk assessments and circumstances.

In this exercise, you will learn about vulnerability scans.

Learning Outcomes

After completing this exercise, you will have further knowledge of:

- Vulnerability and Vulnerability Assessment
- The Need for Vulnerability Research
- Mapping Vulnerabilities

- Vulnerability Management Lifecycle
- Common Vulnerabilities and Exposures (CVE)/Common Vulnerability Scoring System (CVSS)
- False Negative / False Positive
- Exploits

Your Devices

This exercise contains supporting materials for **Ethical Hacker v11**.

Vulnerability and Vulnerability Assessment

A vulnerability is a defect or a bug present in software or part of an application's configuration; as such, the nature of vulnerabilities may vary.

A vulnerability assessment is a detailed review of existing applications, operating systems, or configurations to understand how they function and any security loopholes. For example, you may have several unnecessary applications and services running on a critical server. At the same time, a web server that you are running has a programming defect that could potentially be exploited. This is a vulnerability within the application and therefore is a configuration-related vulnerability.

After discovering vulnerabilities, you need to assign a severity level to each. Assigning severity is necessary because you need to first pay attention to the vulnerabilities that are high severity and can cause damage to a network, application, or system. Therefore, you need to remediate these vulnerabilities as soon as possible. Vulnerabilities with low severities can typically wait and be added to scheduled patching processes.

Depending on the assessment you are performing, you use an appropriate vulnerability scanner. For example, the Nikto tool identifies vulnerabilities for web servers and provides suggestions to close these vulnerabilities.

Vulnerability scanners can find a variety of vulnerabilities, such as:

- Weak passwords
- Unnecessary open ports and services running
- Incorrect permissions on files and folders

- Security misconfigurations in applications
- Missing updates and patches across operating systems and applications

Before running a vulnerability scan, you need to scope the run. For example, you specify whether to run the scan on a single application or the entire network.

The Need for Vulnerability Research

Vulnerabilities evolve from applications, services, protocols, and operating systems. When new versions are released, you can be sure that more vulnerabilities will be discovered.

An ethical hacker and a network administrator/system administrator should be well-versed with already released vulnerabilities. They need to know this information to:

- Gain knowledge about the new vulnerabilities and the threats emerging from them
- Alert the internal teams to patch the systems before it is too late proactively
- Understand how vulnerabilities can be exploited and the damage that can be caused
- Learn from others who are more knowledgeable about the vulnerabilities and share their findings in various forums
- Be knowledgeable about the mitigation process of these vulnerabilities

Ethical hackers can use various forums and communities for their research. Some of the key resources for research are:

- Microsoft Vulnerability Research
- Dark Reading
- Security Tracker
- Trend Micro
- Security Magazine
- PenTest Magazine
- Exploit Database

If you perform a search on Google, you will likely come across several resources.

Mapping Vulnerabilities

After generating a list of vulnerabilities, you need to map them. There can be several targets in a network environment that can have associated vulnerabilities. You may run more than one vulnerability scan. After you are done with the scans, you can collate the vulnerabilities in a single document and map them with the targets. You should update this document whenever you run a vulnerability scan.

Vulnerability Management Lifecycle

Identifying the vulnerabilities to closing them is a process in itself. The organizations that want to be secure and safeguard their data and applications must have the vulnerability management lifecycle and ensure it is implemented and adhered to.

The vulnerability management lifecycle has several key steps:

Identify keyassetsCreate abaselinePerform avulernabilityscanApplyremediationPerform riskassessmentPerformverificationMonitorsecurity

Figure 1.2 The Vulnerability Management Lifecycle, showing Identifying assests, creating baselines, vulnerability scans, risk assessments, remediation, verification and monitoring security.

First, you need to identify which assets you want to protect. Depending on the nature of the business, it could be a few servers or even the entire network. Then, it would help if you created a baseline with optimal performance parameters. It can also include updates, patches, configuration, ports, and services that must be either running or shut down.

For the next step, you need to perform a vulnerability scan. You should create a list of vulnerabilities and rank them according to their severity level. Next, you need to perform a risk assessment to define the risk levels for the assets with vulnerabilities. Beyond this point, you need to start patching or remediating the vulnerabilities. After the remediation is done, you need to verify that all vulnerabilities are closed. Finally, you need to monitor your network and its components.

This process should continuously examine your system or systems.

Common Vulnerabilities and Exposures (CVE)/Common Vulnerability Scoring System (CVSS)

Common Vulnerabilities and Exposures (CVE) are vulnerabilities in published operating systems and applications software. These are publicly known cybersecurity vulnerabilities.

When you find several vulnerabilities within your infrastructure, you would probably not initially know how to rank them or assign scores to them. The Common Vulnerability Scoring System (CVSS) helps assign a score to each vulnerability. For example, you may have a vulnerability that risks your data's confidentiality, integrity, and availability. Using CVSS, you can determine the score. You can determine that the vulnerability has a high score and needs to be handled as a priority.

The scores are calculated based on several metrics. Once you define these metrics, you will determine the CVSS score. For example, a score of 10 would make a vulnerability severe. You can also use CVSS calculators that can help you calculate the scores.

False Positive

A false positive is a condition identified as a vulnerability when it, in fact, does not exist. or example, a vulnerability assessment may indicate that MySQL has a vulnerability, but it does not exist or is not considered a vulnerability in the context of the surrounding infrastructure configuration.

Because of this, vulnerability scan results can produce several false positives. A vulnerability scanner may show false positives due to several reasons:

- It is unable to recognize an executable or service.
- To cover up a vulnerability, you may have implemented a compensating control. Therefore, the vulnerability may be shown as a false positive even though a compensating control covers it.
- The vulnerability scanner does not have updated definitions.
- The scanner configurations are incorrect, so several services or configuration settings may be marked as false positives.

As an ethical hacker, you must be able to identify false positives. Each scan result should be researched and calculated whether it is a false positive or not. You will not know

about every vulnerability you discover, but researching can certainly prevent wastage of work hours.

False Negative

A false negative, an opposite of a false positive, is a vulnerability that exists within the system or applications but is missed by traditional scanning tools and processes. For example, you run a vulnerability scanner, and it lists only a few vulnerabilities within the system. Then, you test the operating system and applications with a second vulnerability scanner and find several new vulnerabilities missed by the first vulnerability scanner. The missed vulnerabilities are referred to as false negatives.

Exploits

Generally speaking, an exploit is a method of delivering a payload to a victim's system or device. A payload is code that runs on the victim's system and connects back to the attacker's system or acts in other malicious ways. For example, when you use Metasploit Framework, you deliver a payload to the victim's system, and then the listener on your system waits for the payload to be triggered. After it triggers, a connection is made to the listener.

Common payload examples are Meterpreter, backdoor, malicious DLLs, and trojans, although payloads can work differently. They can be self-dependent, which means that they get triggered independently or wait for the instructions from the attacker. In either case, they are present on the victim's system.

Exercise 2 — Vulnerability Classification and Assessment Types

Vulnerabilities are never good for the health of a network, applications, or operating system. If you have found one or more vulnerabilities, you should make an effort to close them.

After you perform a vulnerability assessment, you have to know the types of vulnerabilities that have been discovered. Therefore, as an ethical hacker, you need to perform vulnerability classification. You should also be aware of the types of vulnerability assessments you should perform.

In this exercise, you will learn about vulnerability classification and assessment types.

Learning Outcomes

After completing this exercise, you will have further knowledge of:

- Vulnerability Classification
- Types of Vulnerability Assessments
- Specialized Vulnerability Assessments

Vulnerability Classification

Vulnerabilities can be present in applications, networks, or operating systems. Each vulnerability is a specific type and is based on its type. They need to be categorized into different classifications.

Let's look at some of the common categories of vulnerabilities:

Misconfiguration

Misconfigurations can be intentional or unintentional, but human errors lead to one or more vulnerabilities. For example, an administrator installs a new server with Windows operating system but leaves unnecessary services running. You can consider this a misconfiguration.

Some examples of misconfigurations are:

- Outdated applications installed on a system
- Incorrect folder permissions
- Disabled antivirus and security settings
- Default user accounts

Default Configuration Some applications and devices come with factory configuration, which means that the vendor or manufacturer has configured the device or application with certain settings. The information about these settings is available on the vendor's website and other resources on the Internet. When an ethical hacker discovers a device or application, the ethical hacker may attempt to break into it using the default settings.**Buffer Overflows**Each buffer for a function within an application has a specific size. A buffer overflow is a coding error that occurs due to overflowing with more than the required information size. An ethical hacker can overflow the buffer with more information than the buffer can handle, which eventually leads to an application

crash or unexpected behavior.**Design Flaws**Applications can also be have design flaws, which is not unusual. If the developers are unaware of secure programming methods, an application can be designed with several inherent flaws, such as poor input validation, which an ethical hacker can exploit.**Applications and Ope rating System Flaws**

Users are often unaware of the applications and operating system flaws when released. In the context of an operating system, there can be several unwanted applications installed, or a user may install a malware-infected application. An ethical hacker can take advantage of an application of operating system flaw and can exploit it.

In the context of an application, it may be designed with poor authentication and authorization, which eventually can be exploited.

Open Services and PortsA server might be running with several services that are not required. A service may have vulnerabilities that an ethical hacker can exploit. Such situations occur when systems use operating systems that are not baselined or hardened.**Default Passwords**Several applications and network devices, including routers or switches, come with default passwords. In most cases, the network administrators or even the users, specifically at home, do not change the default password. It is easy for an ethical hacker to find the default password for a specific application or network device and use it to gain access.

Types of Vulnerability Assessments

You can conduct different types of vulnerability assessments. The selection of a type of assessment to conduct depends on your business needs and requirements. After defining the vulnerability scanning scope, you will choose the appropriate type of vulnerability assessment.

CredentialedTo run a credentialed scan, you need to have administrative access to a system and run the scan from administrative account credentials, as an authenticated scan probes deeper into a system and provides a more comprehensive vulnerability list. A credentialed scan is more thorough and takes longer to perform. However, it poses a risk of sharing all credentials of the network systems with the ethical hacker, who may find it difficult to maintain them. Later, after the vulnerability scan, the passwords must be changed.**Non-Credentialed**A non-credentialed vulnerability scan requires no specific credentials. It mainly focuses on finding open ports and the services or software using these ports. It is, also known as an unauthenticated scan, is a limitation because it cannot scan deep into the applications or systems requiring credentialed access. This

scan is quicker with a limited vulnerability list. It cannot discover the vulnerabilities with the applications and operating systems protected by firewalls.**Active**An active vulnerability assessment uses network scanners to perform vulnerability checks on the applications and operating systems. An active vulnerability assessment does not perform an in-depth intrusive scan. The scanner only looks for vulnerabilities. For example, it can scan for missing updates or look for open ports. It does not go beyond the scanning process.**Unpatched Systems and Servers**No system or server is free of vulnerabilities. An ethical hacker can discover vulnerabilities and then exploit them to steal confidential or private data, gain control over the application or server, or even disrupt the operations of the application. When vulnerabilities are discovered, the vendor for that specific application releases patches. However, in several cases, the systems or applications on the servers are left unpatched despite having the patches available.**Passive**The target system or application is not contacted in the passive vulnerability assessment. Rather, the network traffic is sniffed to locate systems, applications, and services.**Internal Assessment**The focus of an internal assessment is on an internal network. The assessment includes verifying the patching level of the systems and servers, open ports and services, and the vulnerabilities that may be present on the systems, devices, and overall network.**External Assessment**An external assessment intends to view the network from an ethical hacker's perspective and find the vulnerabilities. This assessment focuses on the network devices situated on the network boundaries. The external assessment can find the open ports and services, unpatched devices, or servers or obtain DNS information.**Manual**After gaining information from footprinting and network scanning, an ethical hacker can use this information to scan for vulnerabilities manually. An ethical hacker may rank vulnerabilities based on their criticality and exploit them with high criticality.**Automated**Alternatively, an ethical hacker may use a vulnerability scanner to find vulnerabilities within the network in this type of vulnerability assessment. In this vulnerability scan, the ethical hacker does not need to perform footprint or network scanning.

Specialized Vulnerability Assessments

There are specialized vulnerability assessment methods other than the above-mentioned vulnerability assessment methods.

Let's look at some of the key specialized methods:

Host-based

A host-based vulnerability assessment is conducted on dividual hosts. A host-based vulnerability assessment intends to find vulnerabilities due to:

- Incorrect registry permissions
- File and folder permissions
- Software configuration
- Excessive user privileges
- Unnecessary services running
- Unpatched operating system and applications

Network-based

A networked-based vulnerability assessment focuses on the entire network and its resources. It could be the network devices or servers catering to many users. Network-based vulnerability assessments intend to find vulnerabilities that can exist due to:

- Unpatched systems, servers, and network devices
- Use of weak encryption protocols
- Use of clear-text data transmissions
- Unnecessary services running
- Misconfigurations

Application

An application vulnerability assessment focuses on finding vulnerabilities within applications, client/server, or web applications. The application vulnerability assessment can focus on:

- Designing flaws within the application
- Unpatched applications
- Coding errors
- Misconfigurations

Database

Organizations use applications that store data in the databases running in the backend. Database vulnerability assessments scan databases and database servers for known vulnerabilities. Database vulnerability assessment can focus on:

- Locating the vulnerabilities in the database servers, such as MySQL or Microsoft SQL Server
- Locate known SQL vulnerabilities, such as SQL Injection
- Locate any misconfigurations in a database server

Wireless

A wireless network vulnerability assessment is about locating the vulnerabilities in an organization's wireless network. The wireless network vulnerability assessment can focus on:

- Locating the weak encryption protocol, such as Wired Equivalent Privacy (WEP)
- Locating rogue wireless access points
- Sniffing the network traffic to capture and crack the encryption keys
- Connecting with the wireless network as a rogue user

DistributedAn organization may have a distributed network across different geographical locations. A distributed vulnerability assessment needs to focus on all the network assets simultaneously. The vulnerability assessment must be synchronized to test the network assets simultaneously.

Exercise 3 — Vulnerability Assessment Solutions and Tools

A vulnerability assessment is performed to ensure that all the vulnerabilities are located and closed across all applications, operating systems, and networks. However, before proceeding with a specific vulnerability assessment, the organization needs to choose the correct solution to ensure that the vulnerabilities are located in the right manner.

In this exercise, you will learn about vulnerability assessment solutions and tools.

Learning Outcomes

After completing this exercise, you will be able to**:**

- Use Nikto for Vulnerability Scanning
- Use Lynis for System Vulnerability Scanning

After completing this exercise, you will have further knowledge of:

- Vulnerability Assessment Approaches
- Vulnerability Assessment Process
- Types of Vulnerability Assessment Tools

Your Devices

You will be using the following devices in this lab. Please power these on now.

PLABDC01Domain Controller192.168.0.1/24PLABWIN10Domain MemberWorkstation192.168.0.3/24PLABKALI01Domain MemberWorkstation192.168.0.5/24

- PLABDC01

Windows Server 2019 — Domain Server192.168.0.1/24

- PLABWIN10

Windows 10 — Workstation192.168.0.3/24

- PLABKALI01

Kali 2022.1 — Linux Kali Workstation192.168.0.5/24

Vulnerability Assessment Approaches

There are various vulnerability assessment approaches that an organization can choose from. Let’s look at some of them.

Product-based As the name suggests, a product is involved and is installed on the internal network. The product scans for the vulnerabilities within an internal network.**Service-based**A service-based approach is offered by third parties who use their proprietary services to scan a network. A service-based approach is installed within

the network or hosted on an external server to scan an internal network.**Tree-based**A tree-based approach uses multiple scanners. It can use one scanner for Windows and another scanner for Linux systems. An administrator provides the initial input to initiate the scan.**Inference-based**An inference-based approach first builds a pool of protocols running on a system. Then, it attempts to find the services running within a system. The intent is to map a protocol to the services. Finally, it detects the vulnerabilities and executes relevant tests based on these vulnerabilities.

Vulnerability Assessment Process

Broadly speaking, there are three steps in the vulnerability assessment process.

LocateNodesPerform OSand ServiceDiscoveryScan for VulnerabilitiesFind live hostson a networkFind open ports & running services on an operating systemFind vulnerabilities in running services

Figure 3.1 Screenshot of Vulnerability Assessment Process.

As an ethical hacker, you need to perform these steps to ensure that the vulnerabilities are discovered and closed before an attacker exploits them.

Types of Vulnerability Assessment Tools

There are six types of vulnerability assessment tools selected and used depending on your requirement. For example, you may have a requirement to scan a single host. In such a case, you will choose one type of vulnerability assessment tool. On the other hand, you have a requirement to scan a database for vulnerabilities. The important point to remember is that a single tool may not fulfill all your requirements. Therefore, choose the correct tool.

Let's look at the vulnerability assessment tools:

Host-based Assessment ToolA host-based assessment tool is suited for a single system running one or more applications. For example, a single system may be configured as a webserver. The host-based assessment tools can thoroughly scan an operating system for known vulnerabilities and provide recommendations in closing them.**Scope Assessment Tool**Some scope assessment tools are targeted for specific applications, while the others are targeted at the operating systems and applications in general. The scope assessment tools allow the ethical hacker to select a specific scan, execute it, and generate a report.**Depth Assessment Tool**A depth assessment tools

test a system against previously unknown vulnerabilities in a system. It uses fuzzers that provide arbitrary input to the system to check its stability and identify vulnerabilities. It can use a set of signatures to identify vulnerabilities.**Active and Passive Tools**An active vulnerability assessment tools directly work on the target system and use its system resources to conduct a scan. On the other hand, passive tools do not work on the target systems and should be used with the critical systems as the target's system resources are not impacted.**Application-layer Assessment Tool**An application-layer assessment tool is used to discover vulnerabilities within webservers and databases. Most cyber-attacks are targeted at the application layer.**Location and Data Examination Assessment Tools**A location and data examination assessment tools can include various scanners, such as network-based scanners, agent-based scanners, proxy scanners, and cluster scanners.

Task 1 — Use Nikto for Vulnerability Scanning

Nikto is a vulnerability scanner that is part of Kali Linux. Ethical hackers, penetration testers widely use it, and hackers to find the vulnerabilities in web applications.

In this task, you will learn to use Nikto for vulnerability scanning.

Step 1

Ensure you have powered on all the devices listed in the introduction and connect to **PLABKALI01**.

Log in using the following credentials:

Username:

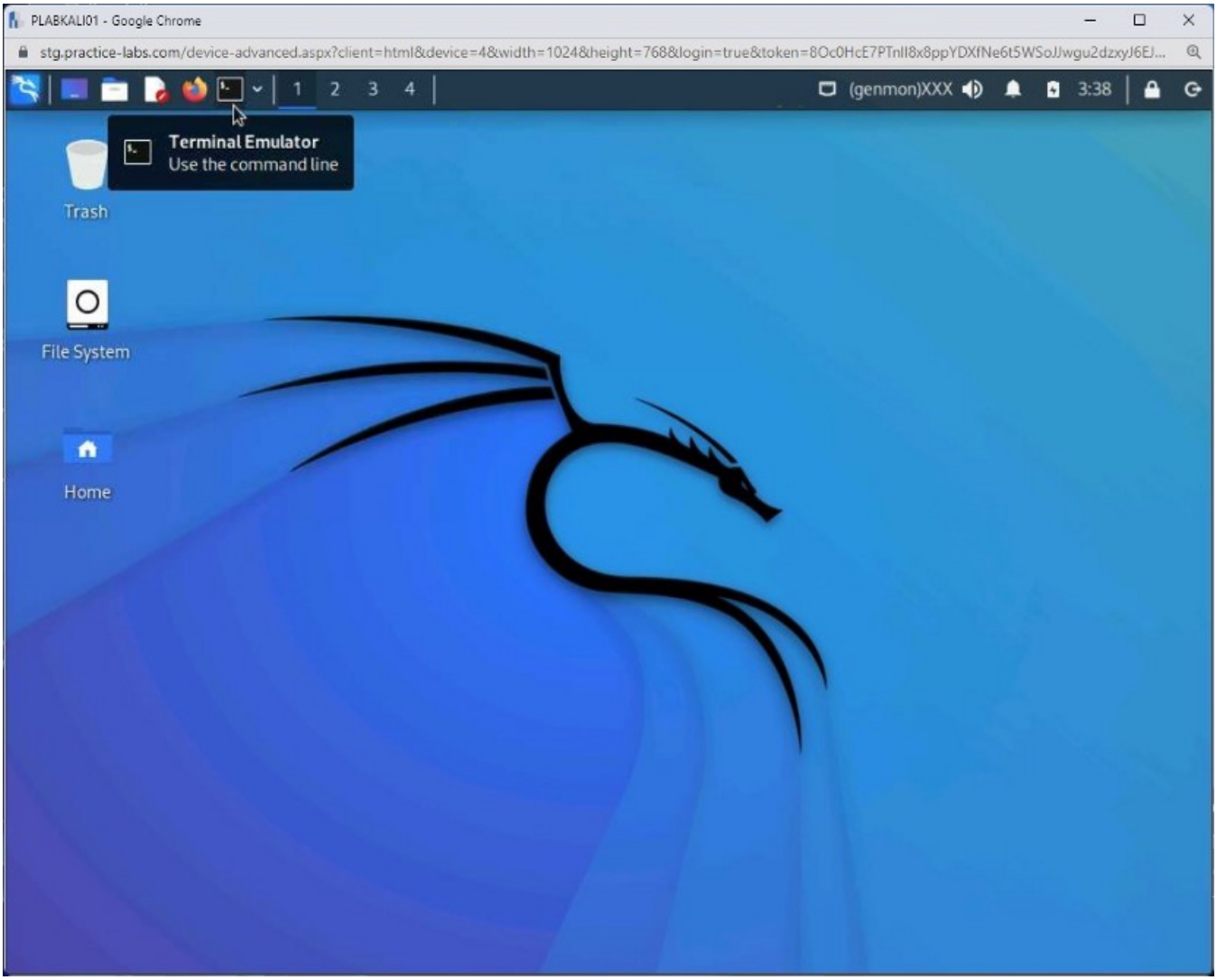
```
root
```

Password:

Password

The desktop of **PLABKALI01** is displayed.

Open a new terminal window by clicking the **Terminal Emulator** icon on the taskbar.



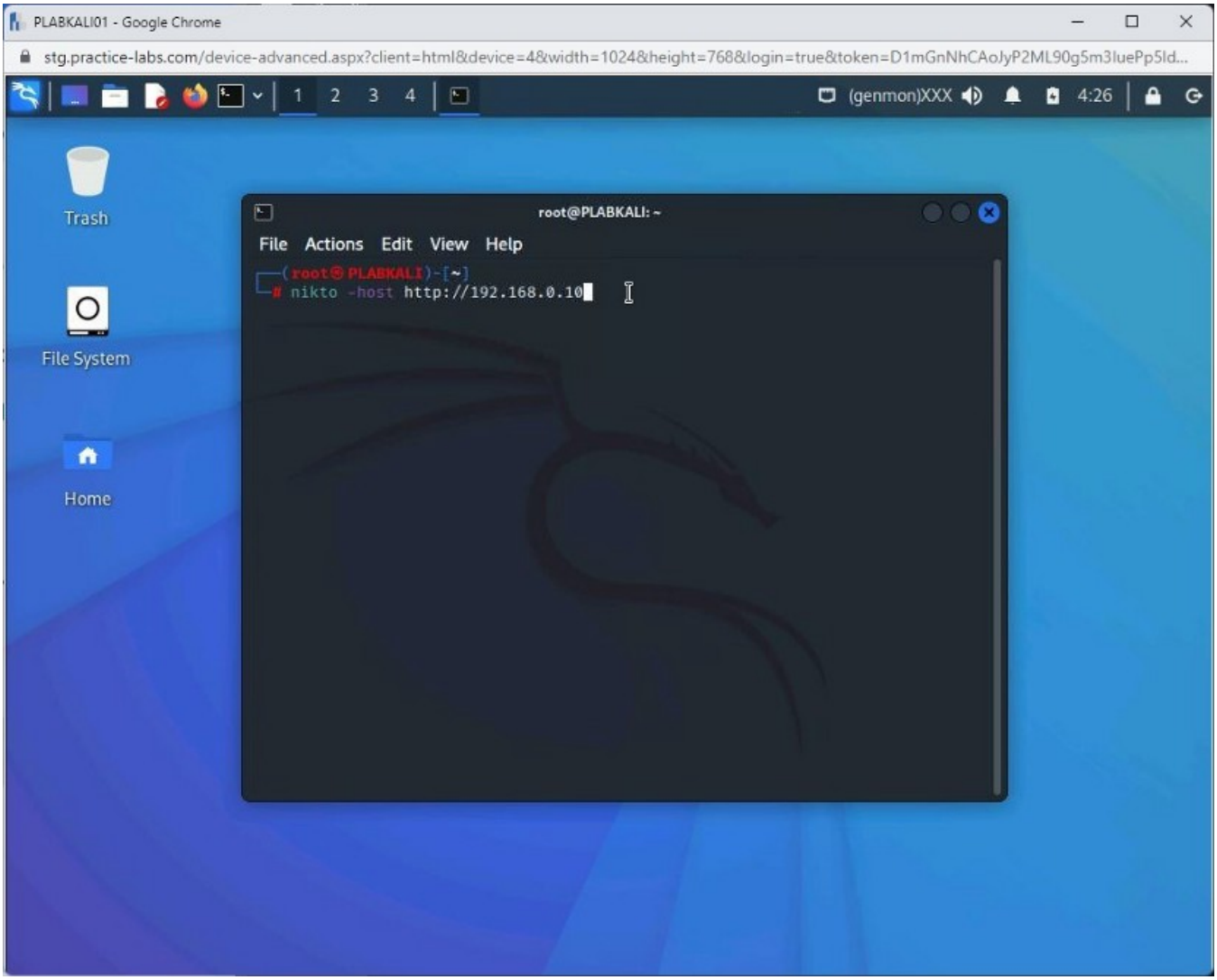
Step 2

To scan a website for vulnerabilities, type the following command:

Note: *Instead of the* ***-host*** *parameter, you can also use the* ***-h*** *parameter. Both parameters provide the same result.*

```
nikto -host http://192.168.0.10
```

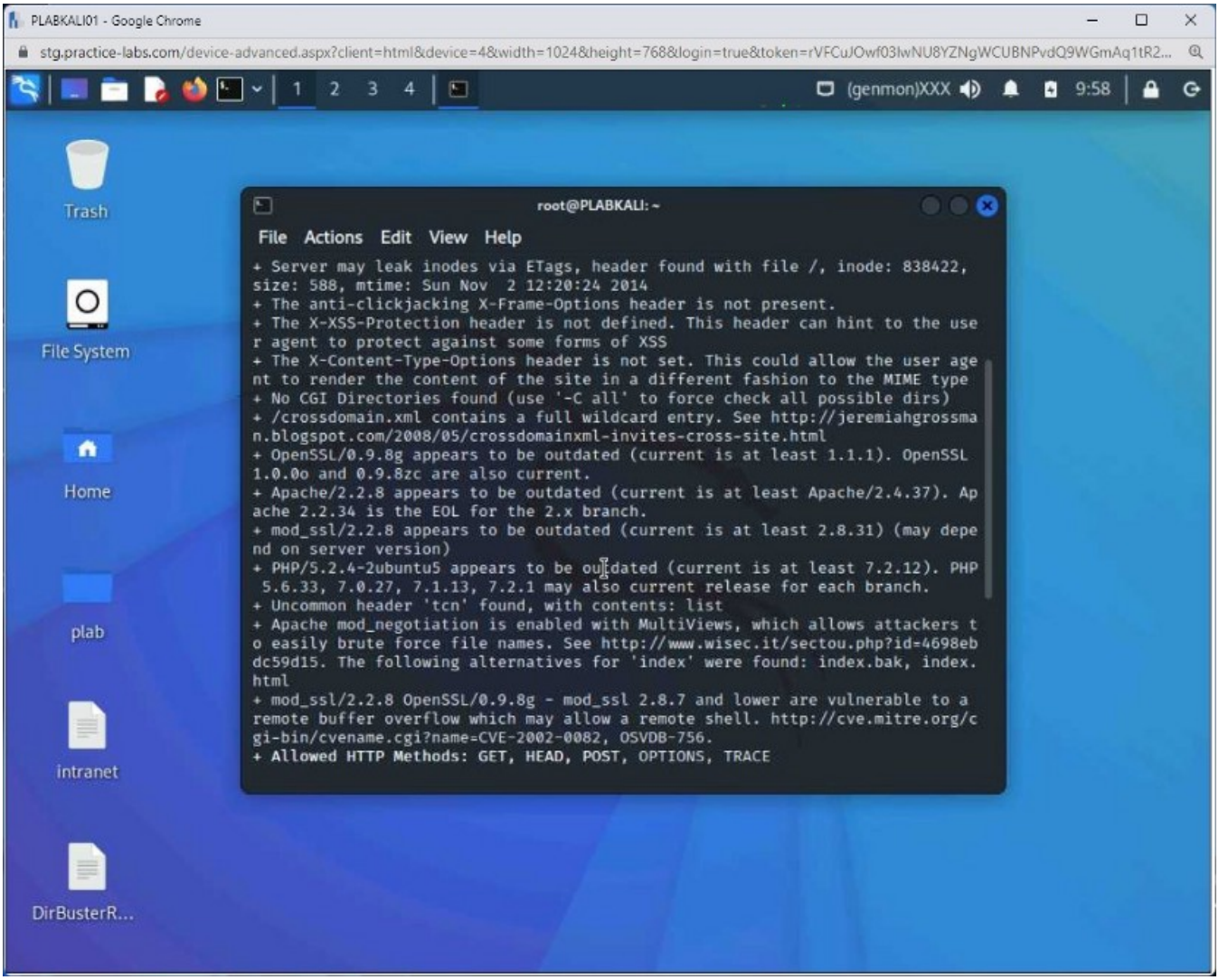
Press **Enter**.



Step 3

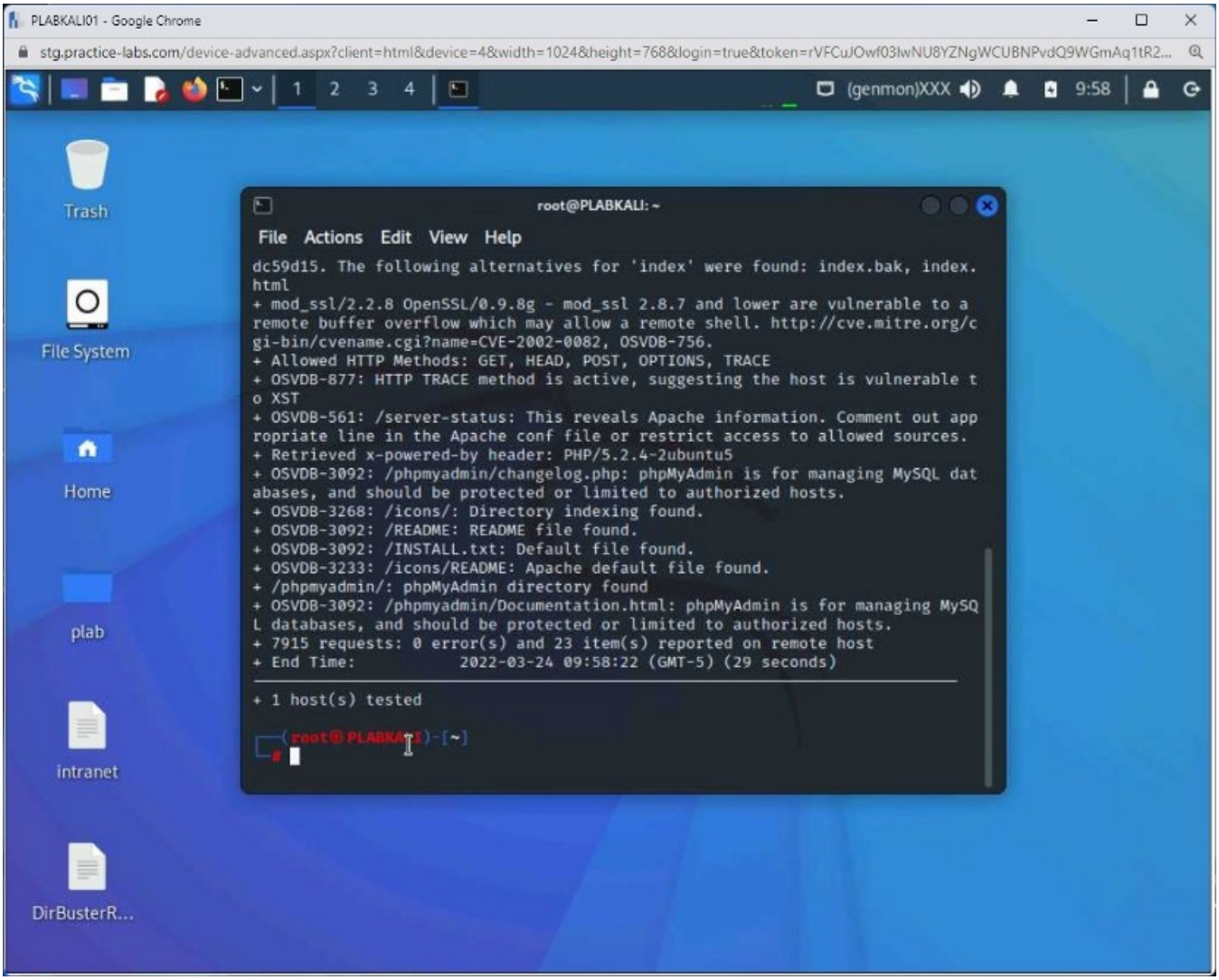
The vulnerability scanning process starts.

Depending on the number of vulnerabilities, the process may run for a few minutes.



Step 4

A detailed list of vulnerabilities is listed as the output.



Task 2 — Use Lynis for System Vulnerability Scanning

Lynis is a built-in multi-purpose tool in Kali Linux. It is designed to perform the following tasks:

- Security auditing
- Compliance testing
- Penetration testing
- Vulnerability detection
- System Hardening

It can perform several types of system auditing, such as system binaries, boot loaders, startup services, run level, loaded modules, kernel configuration, core dumps, and so on.

In this task, you will learn to use Lynis for system vulnerability scanning. To do this, perform the following steps:

Step 1

Ensure you have powered on all the devices listed in the introduction and connect to **PLABKALI01**.

Clear the screen by entering the following command:

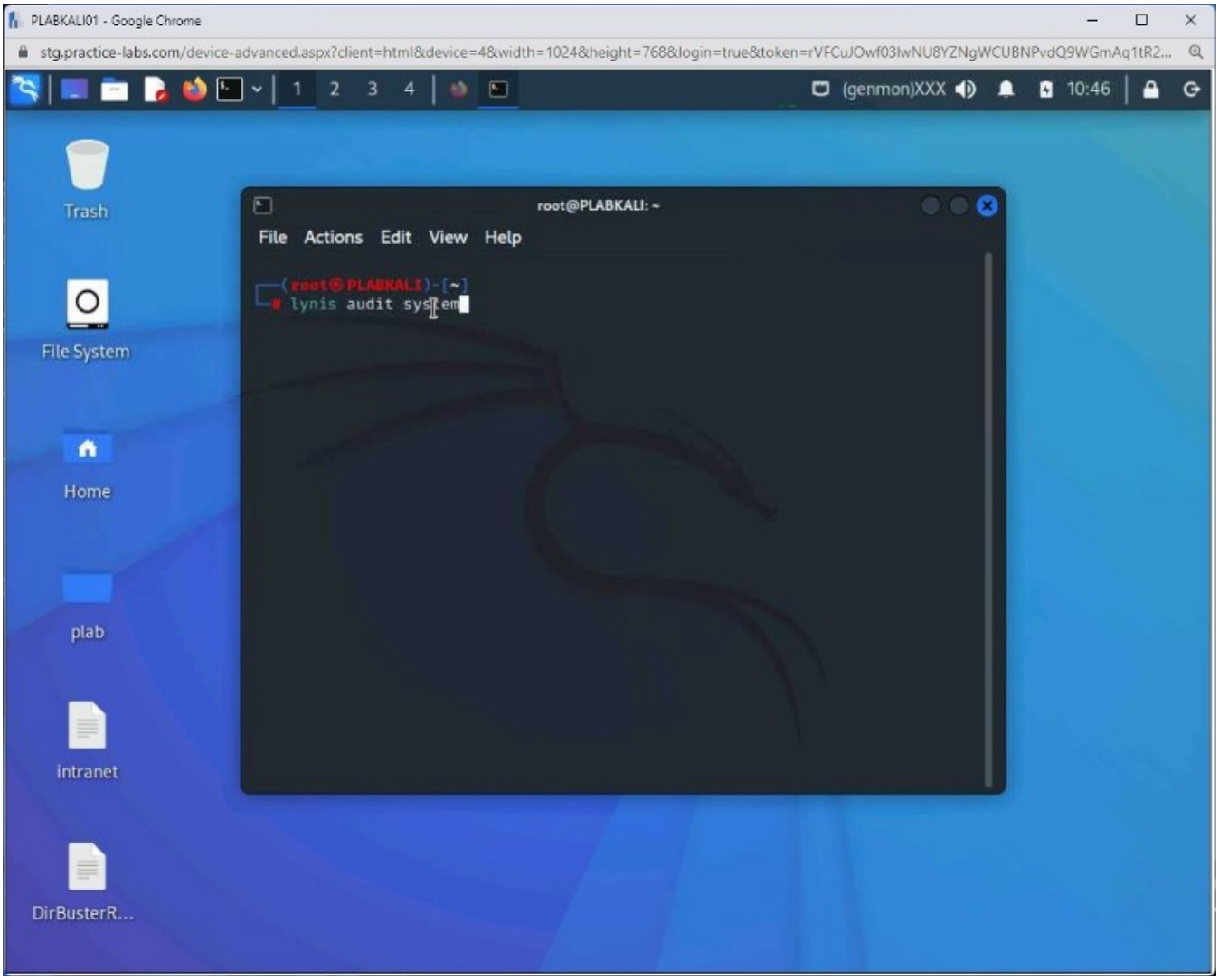
```
clear
```

By default, Lynis will perform a local system scan. You have the option to run a normal audit scan or can run the entire system scan.

Let's first run the normal audit scan. Type the following command:

```
lynis audit system
```

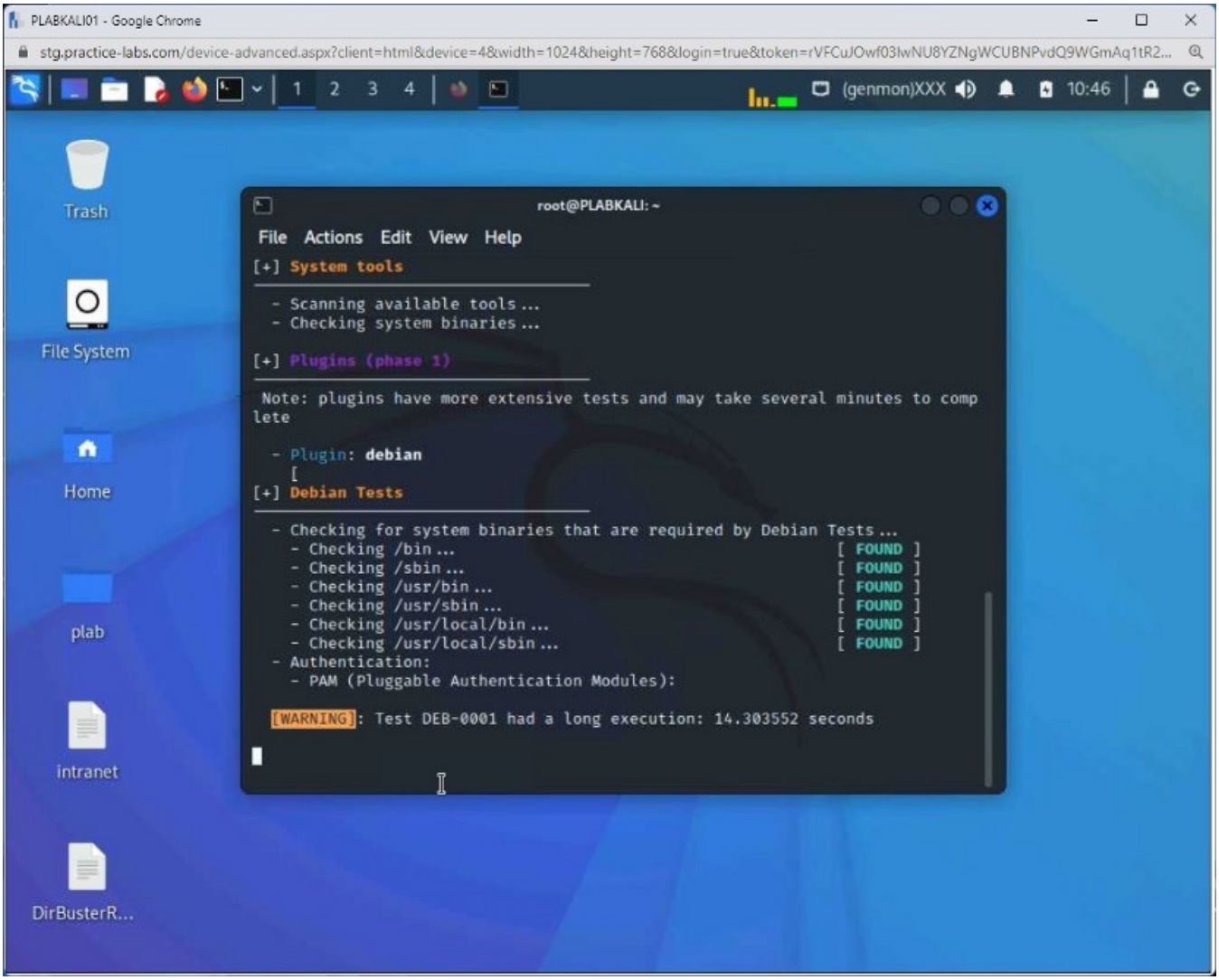
Press **Enter**.



Step 2

The auditing process starts. Notice that it has already detected the operating system version, its hostname, and so on.

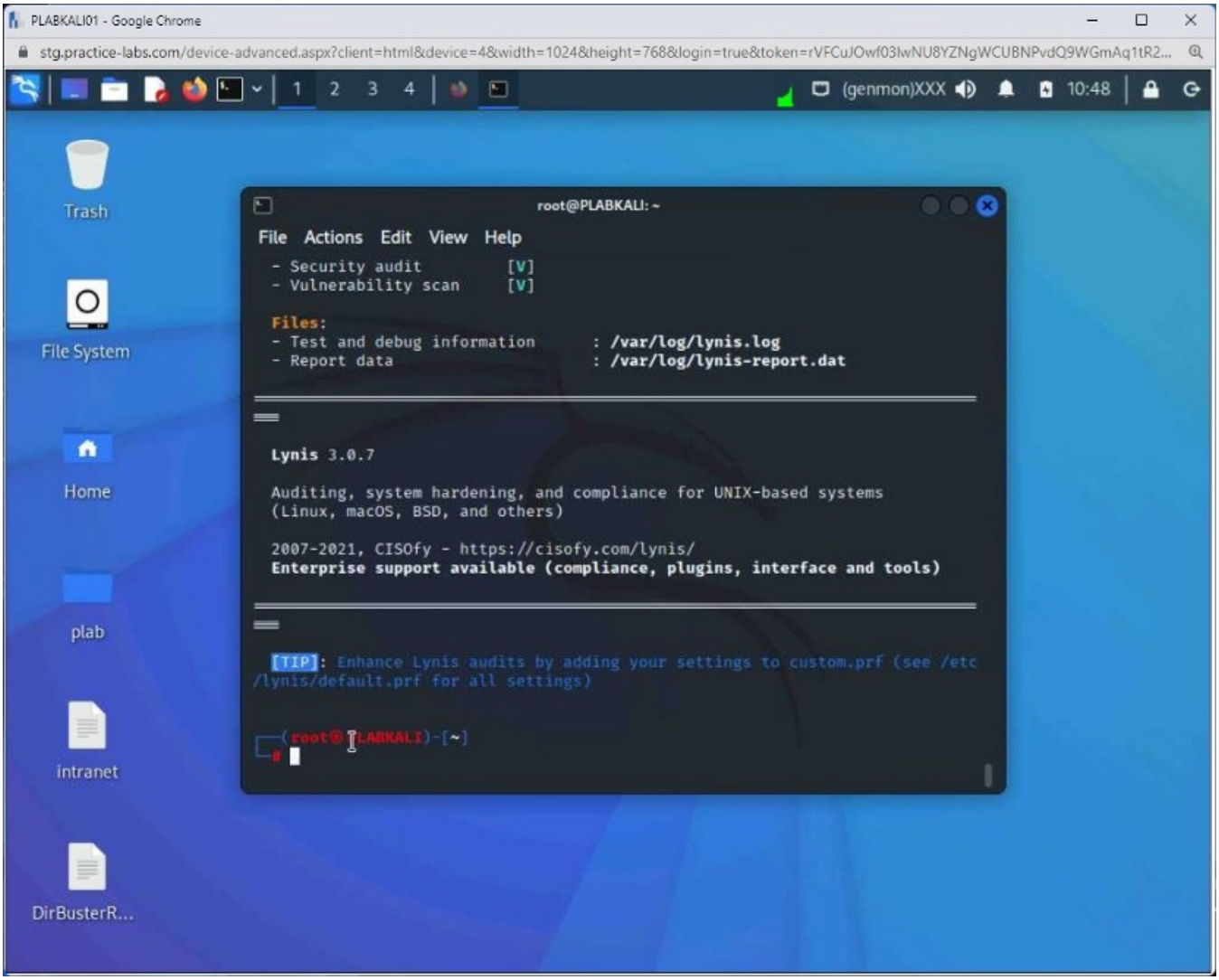
Note: *The audit process will take a few minutes to complete.*



Step 3

During the scan process, you will notice that the results are categorized under different categories.

The audit process completes.

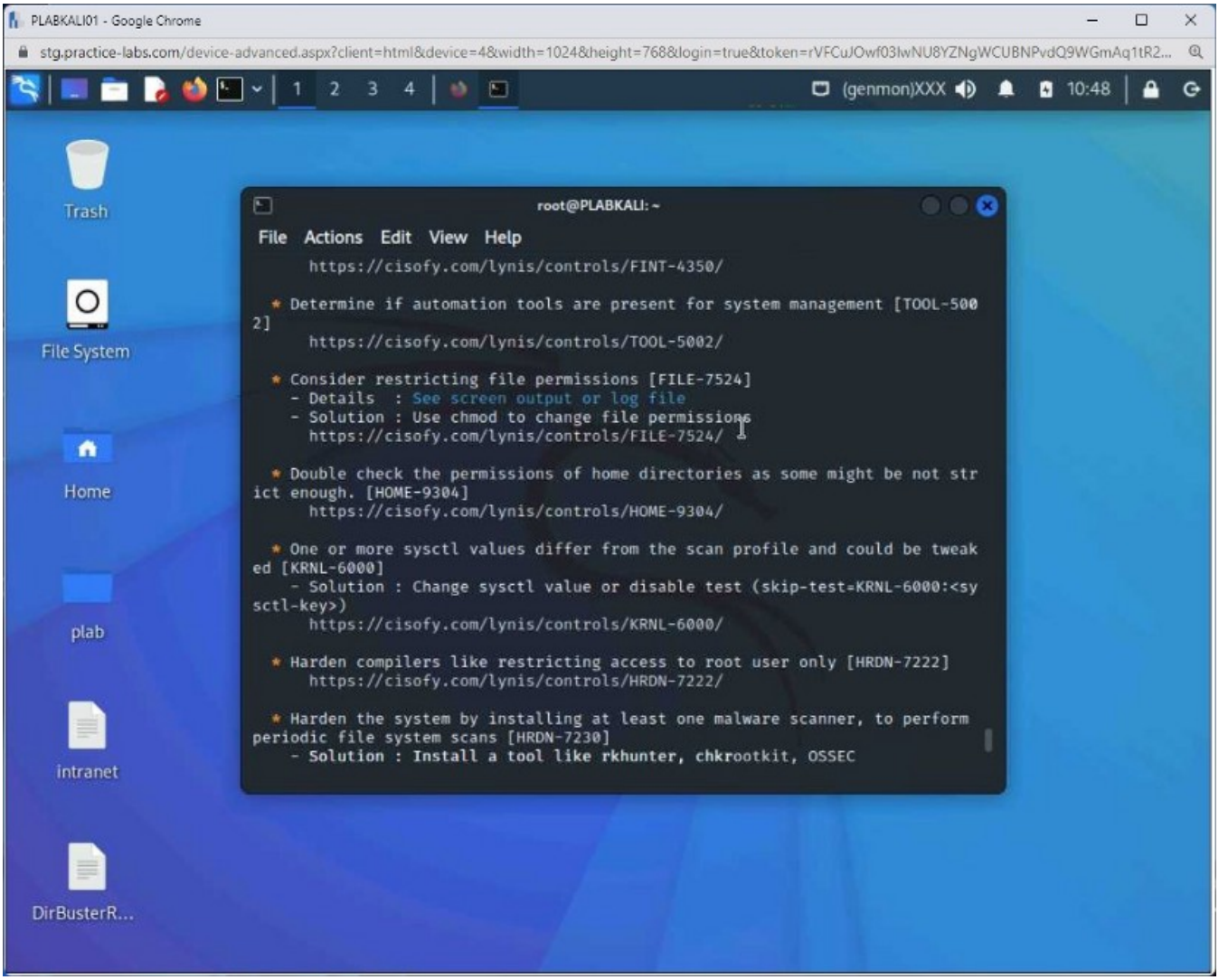


Step 4

You will need to scroll up to review the results. Notice that there are vulnerabilities that are located. Lynis also provides a suggestion to close the vulnerabilities.

For example, hardening the SSH configuration. It has a setting **PermitRootLogin** set to **Yes**. Lynis audit provides suggestions to handle this vulnerability.

Note: *Take a few minutes and go through the audit report. If time permits, then you should use the following command to perform a full audit scan****: lynis audit system -c***



Exercise 4 — Vulnerability Assessment Reports

After the vulnerability assessments have been executed, their outcome should be in the form of reports. The outcome of the scan will contain one or more vulnerabilities that should be listed within a report, which should contain the remediation step for the vulnerabilities that have been located.

After the vulnerability assessment process is complete, it is time for you to write the vulnerability assessment report. The report should outline the following:

- Identified vulnerabilities during the scan
- Remediation steps to mitigate vulnerabilities
- CVD IDs for vulnerabilities
- The date on which vulnerabilities were discovered
- Score of each vulnerability

- Severity of each vulnerability

Most vulnerability assessment tools, such as Nessus and Nikto, provide you with the reports that list the vulnerabilities.

In this exercise, you will learn about vulnerability assessment reports.

Learning Outcomes

After completing this exercise, you will be able to**:**

- Create a Formatted Report With Nikto

After completing this exercise, you will have further knowledge of:

- The Vulnerability Assessment Report

Your Devices

You will be using the following devices in this lab. Please power these on now.

PLABDC01Domain Controller192.168.0.1/24PLABWIN10Domain MemberWorkstation192.168.0.3/24PLABKALI01Domain MemberWorkstation192.168.0.5/24

- PLABDC01

Windows Server 2019 — Domain Server192.168.0.1/24

- PLABWIN10

Windows 10 — Workstation192.168.0.3/24

- PLABKALI01

Kali 2022.1 — Linux Kali Workstation192.168.0.5/24

Task 1 — Create a Formatted Report With Nikto

Nikto can provide a report in an HTML format. You can run the command to execute the Nikto scanner, and then **-o** parameter needs to be defined with the file name, which will be the report.

In this task, you will learn to create a formatted report with Nikto.

Step 1

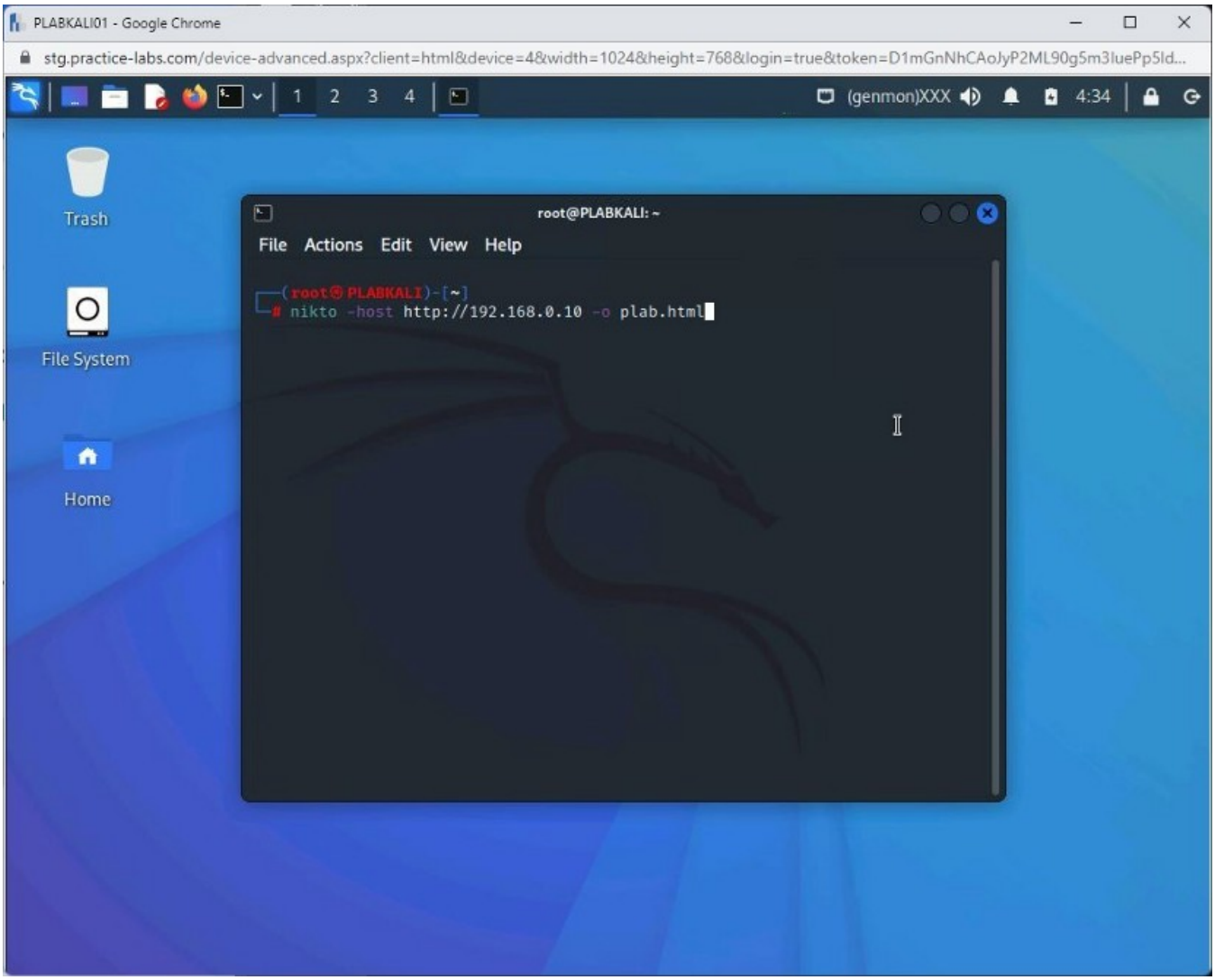
Connect to **PLABKALI01** and open a new terminal window.

To scan a website for vulnerabilities and save the output to an HTML file, type the following command:

Note: *Instead of the -host parameter, you can also use the -h parameter. Both parameters provide the same result.*

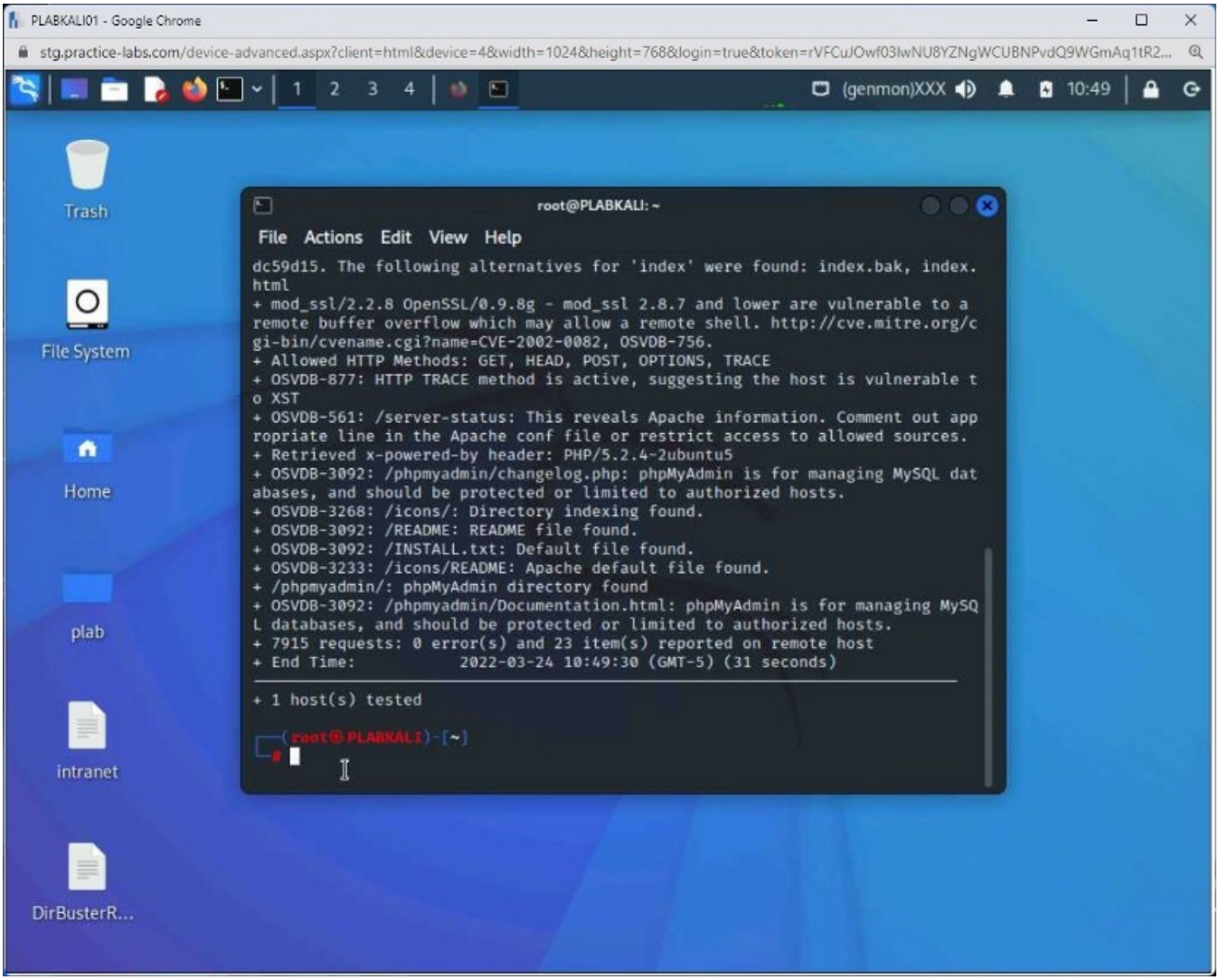
```
nikto -host http://192.168.0.10 -o plab.html
```

Press **Enter**.



Step 2

It takes a few minutes for the scan to complete.



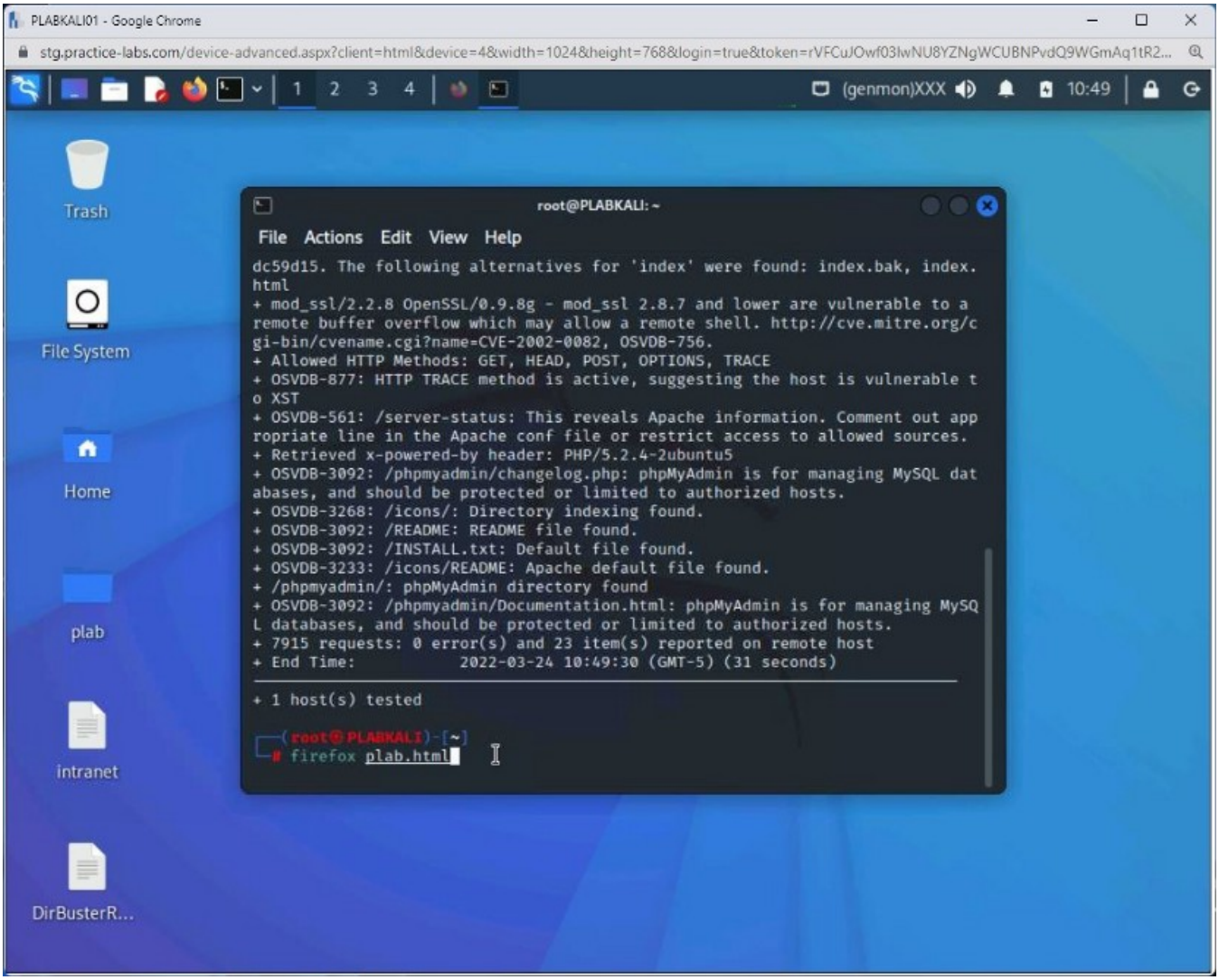
Step 3

Let the vulnerability scanning process complete.

Then, type the following command:

```
firefox plab.html
```

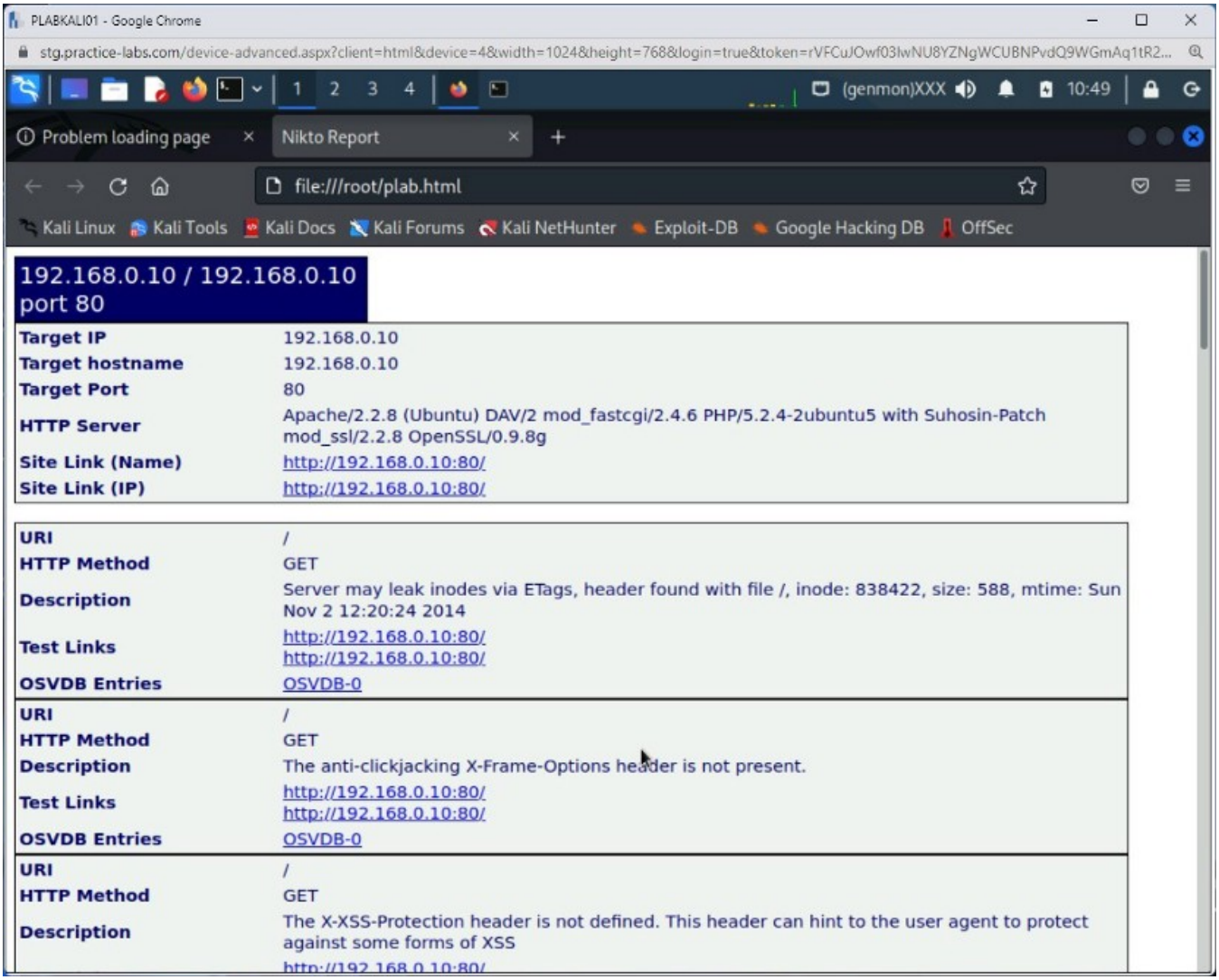
Press **Enter**.



Step 4

A new **Firefox** window opens.

Notice that the vulnerabilities are listed on the HTML webpage.


PLABKALI01 - Google Chrome
stg.practice-labs.com/device-advanced.aspx?client=html&device=4&width=1024&height=768&login=true&token=rVFCuJOwf03IwNU8YZNgWCUBNPvdQ9WGmAq1tR2...
1 2 3 4
(genmon)XXX
10:49
Problem loading page
Nikto Report
file:///root/plab.html
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
192.168.0.10 / 192.168.0.10
port 80
Target IP 192.168.0.10
Target hostname 192.168.0.10
Target Port 80
HTTP Server Apache/2.2.8 (Ubuntu) DAV/2 mod_fastcgi/2.4.6 PHP/5.2.4-2ubuntu5 with Suhosin-Patch mod_ssl/2.2.8 OpenSSL/0.9.8g
Site Link (Name) http://192.168.0.10:80/
Site Link (IP) http://192.168.0.10:80/
URI /
HTTP Method GET
Description Server may leak inodes via ETags, header found with file /, inode: 838422, size: 588, mtime: Sun Nov 2 12:20:24 2014
Test Links http://192.168.0.10:80/ http://192.168.0.10:80/
OSVDB Entries OSVDB-0
URI /
HTTP Method GET
Description The anti-clickjacking X-Frame-Options header is not present.
Test Links http://192.168.0.10:80/ http://192.168.0.10:80/
OSVDB Entries OSVDB-0
URI /
HTTP Method GET
Description The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS