Phase 3: Identify Vulnerabilities

Goal:
Identify the tools and techniques to be used to scan for vulnerabilities.

Procedure:
List the tools for use to perform vulnerability scanning and how they will be used. Include both Tenable Nessus and OpenVAS, as well as tools designed for specific technologies or platforms.

Tools and Techniques:

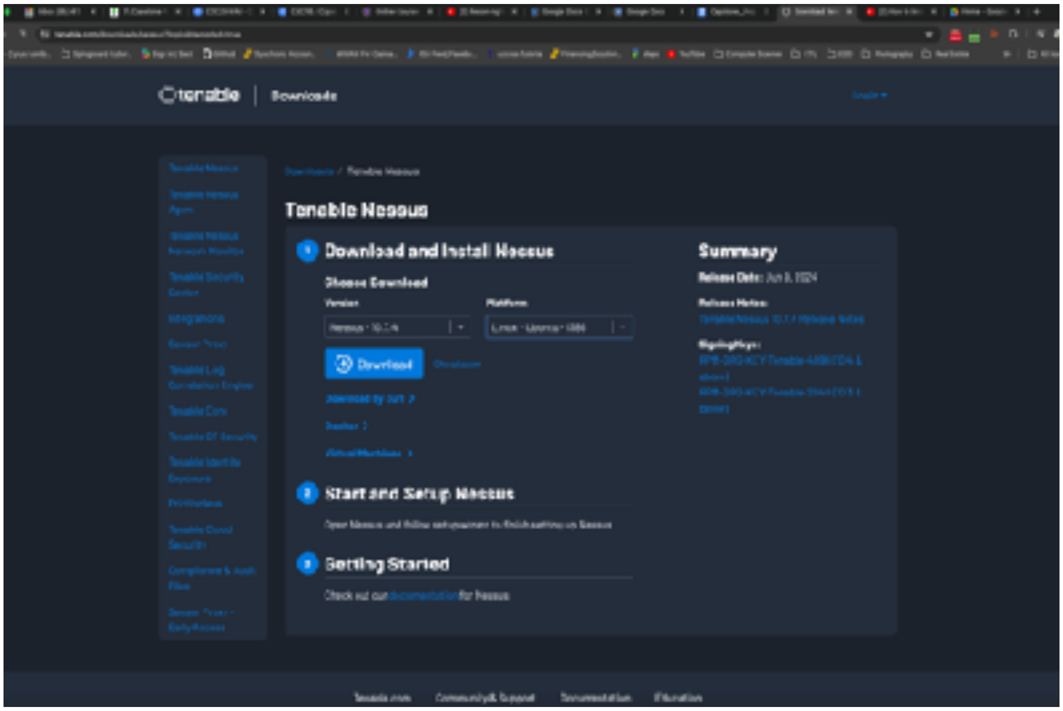
---

1. Tenable Nessus

Purpose:
Tenable Nessus is a widely used vulnerability scanner that can identify a broad range of vulnerabilities across different systems and applications.

How to Use:

- **Install Nessus**: Download and install Nessus on a dedicated scanning machine.



Tenable Nessus

1 Download and Install Nessus

Summary

2 Start and Setup Nessus

3 Getting Started

Check out our documentation for Nessus

https://www.tenable.com/downloads/nessus?loginAttempted=true

- **Configure Scan**:

  1. Launch Nesus and navigate to “New Scan”.
  2. Choose the appropriate scan template (e.g., “Basic Network Scan”).
  3. Enter the target IP range.
  4. Configure scan settings such as credentials for authenticated scans.

- **Run Scan**:

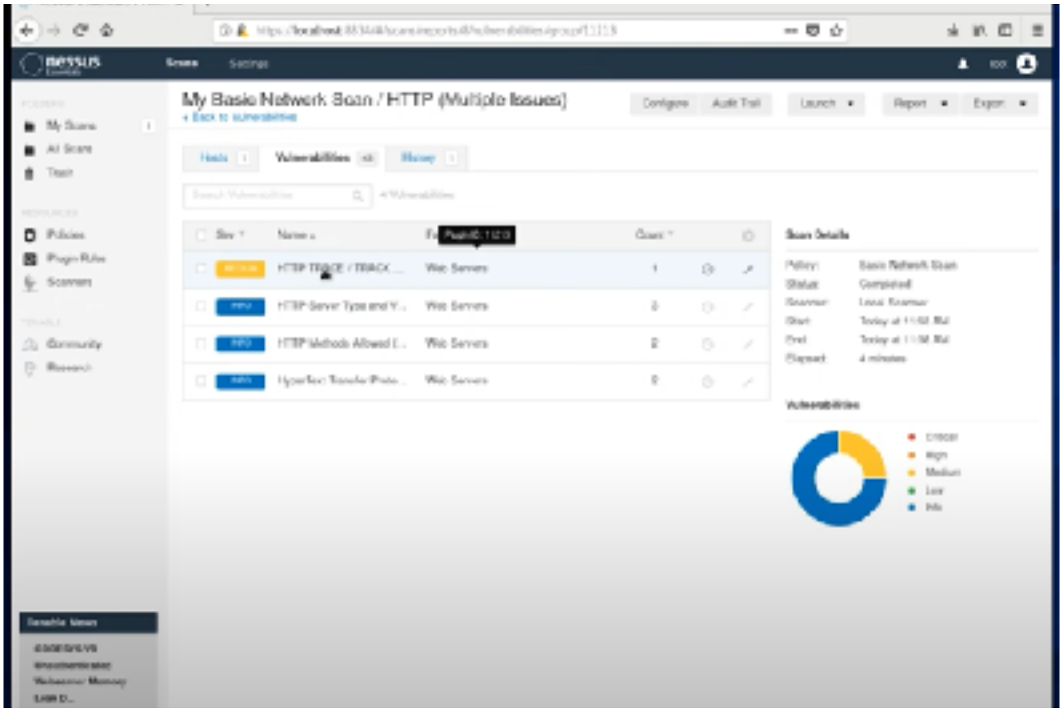
  5. Start the scan and monitor its progress.
  6. Review the scan results, focusing on critical and high-severity vulnerabilities.

Pros:

- Comprehensive vulnerability coverage.
- Regular updates with new vulnerability checks.
- User-friendly interface with detailed reporting.

Cons:

- Can be resource-intensive.
- Requires a license for full functionality.



My Basic Network Scan / HTTP (Multiple Issues)

Sev	Name	Family	Count
[illegible]	HTTP TRACE / TRACK ...	Web Servers	1
[illegible]	HTTP Server Type and V...	Web Servers	2
[illegible]	HTTP Methods Allowed (...	Web Servers	2
[illegible]	HyperText Transfer Proto...	Web Servers	[illegible]

Scan Details

Policy:	Basic Network Scan
Status:	Completed
Scanner:	Local Scanner
Start:	Today at [illegible]
End:	Today at [illegible]
Elapsed:	4 minutes

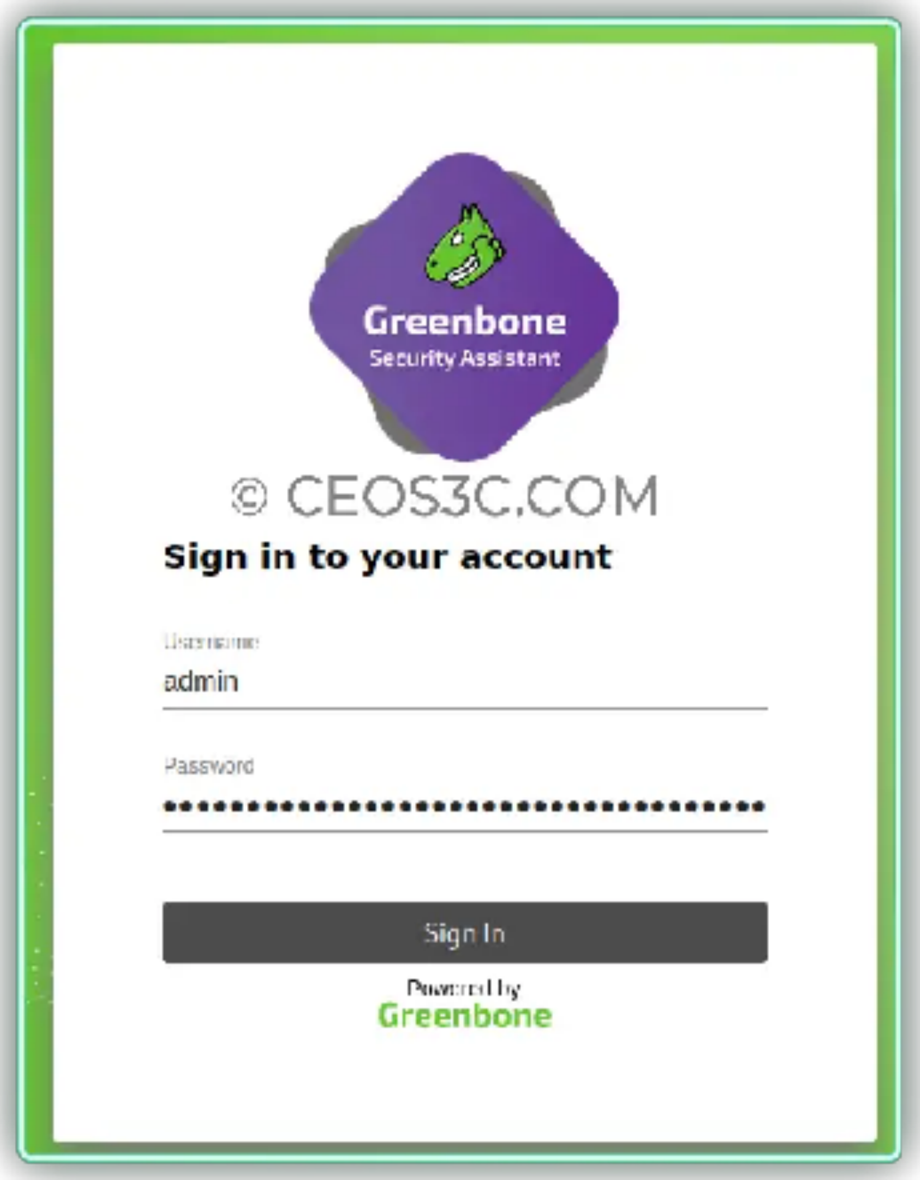
Nessus Basic Scan - Vulnerabilities

2. OpenVAS (Open Vulnerability Assessment Scanner)

Purpose:
OpenVAS is an open-source vulnerability scanner that provides comprehensive scans for known vulnerabilities.

How to Use:

- **Install OpenVAS**: Install on a dedicated server.



Sign in to your account

Username
admin

Password
••••••••••••••••••••••••••••••••••••

- **Configure Scan**:

7. Access the OpenVAS web interface.
8. Create a new scan task and configure the target IP range.
9. Set up scan configurations, including timing and performance options.

- **Run Scan**:

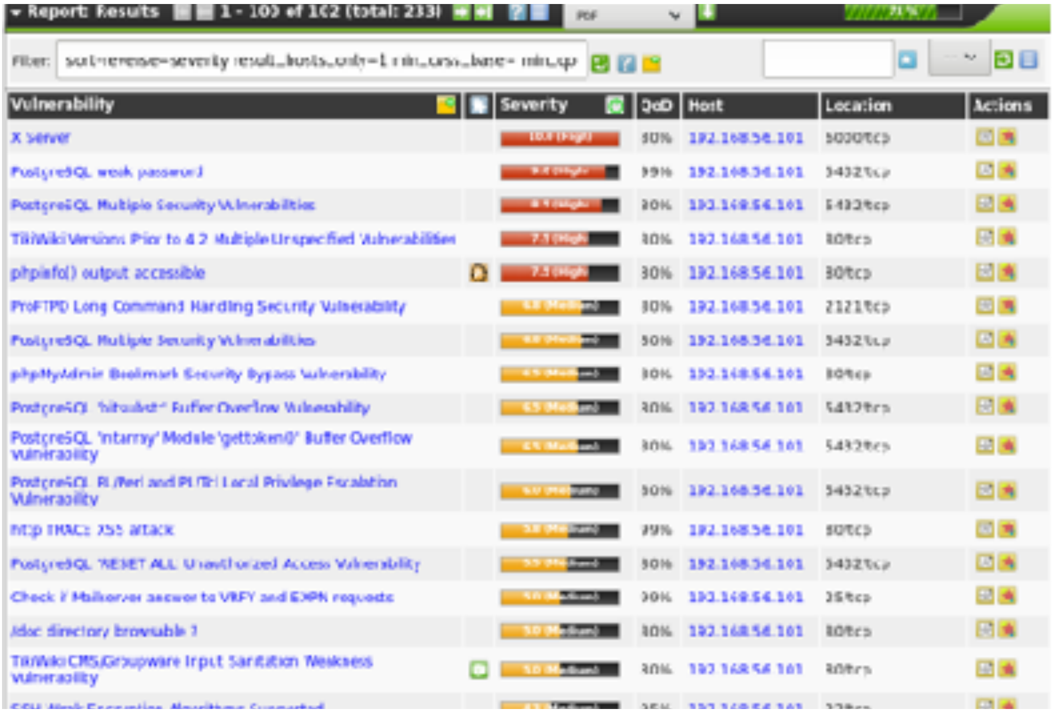
10. Start the scan and monitor its progress.
11. Analyze the scan results in the OpenVAS dashboard.

Pros:

- Free and open source.
- Regular updates and a large database of vulnerability checks.

Cons:

- Can produce false positives.
- Less user-friendly compared to commercial tools.



Report: Results 1 - 100 of 102 (total: 233)

Filter: sort-reverse=severity result_hosts_only=1 min_cvss_base= min_qo

Vulnerability		Severity	QoD	Host	Location	Actions
X Server		10.0 (High)	80%	192.168.56.101	6000/tcp	
PostgreSQL weak password		9.0 (High)	99%	192.168.56.101	5432/tcp	
PostgreSQL Multiple Security Vulnerabilities		8.5 (High)	80%	192.168.56.101	5432/tcp	
TikiWiki Versions Prior to 4.2 Multiple Unspecified Vulnerabilities		7.5 (High)	80%	192.168.56.101	80/tcp	
phpinfo() output accessible		7.5 (High)	80%	192.168.56.101	80/tcp	
ProFTPD Long Command Handling Security Vulnerability		6.8 (Medium)	80%	192.168.56.101	2121/tcp	
PostgreSQL Multiple Security Vulnerabilities		6.8 (Medium)	80%	192.168.56.101	5432/tcp	
phpMyAdmin BookMark Security Bypass Vulnerability		6.5 (Medium)	80%	192.168.56.101	80/tcp	
PostgreSQL 'bitsubstr' Buffer Overflow Vulnerability		6.5 (Medium)	80%	192.168.56.101	5432/tcp	
PostgreSQL 'intarray' Module 'gettoken()' Buffer Overflow Vulnerability		6.5 (Medium)	80%	192.168.56.101	5432/tcp	
PostgreSQL PL/Perl and PL/Tcl Local Privilege Escalation Vulnerability		6.0 (Medium)	80%	192.168.56.101	5432/tcp	
http TRACE XSS attack		5.8 (Medium)	99%	192.168.56.101	80/tcp	
PostgreSQL 'RESET ALL' Unauthorized Access Vulnerability		5.5 (Medium)	80%	192.168.56.101	5432/tcp	
Check if Mailserver answer to VRFY and EXPN requests		5.0 (Medium)	99%	192.168.56.101	25/tcp	
/doc directory browsable ?		5.0 (Medium)	80%	192.168.56.101	80/tcp	
TikiWiki CMS/Groupware Input Sanitation Weakness Vulnerability		5.0 (Medium)	80%	192.168.56.101	80/tcp	
SSH Weak Encryption Algorithms Supported		4.3 (Medium)	95%	192.168.56.101	22/tcp	

OpenVAS scan

3. Burp Suite

Purpose:
Burp Suite is a powerful web application security testing tool that can identify vulnerabilities in web applications.

How to Use:

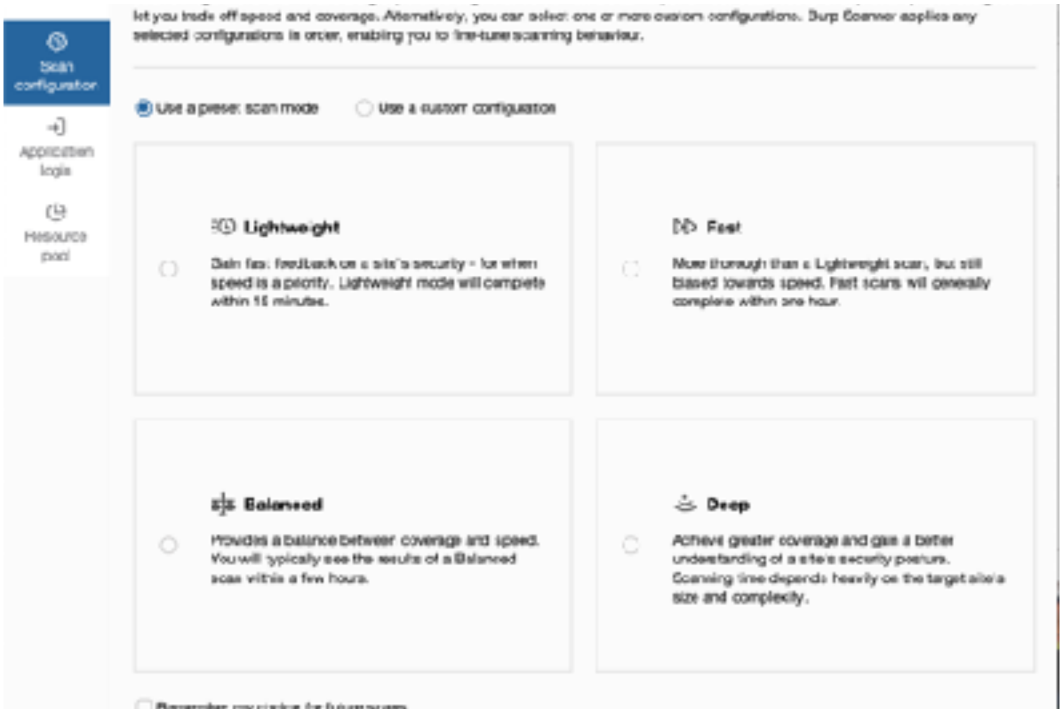
- **Install Burp Suite**: Install on a dedicated testing machine.

- **Configure Scan:**

- Launch Burp Suite and set up a new project.
- Configure the target web application URL.
- Set up scan configurations, such as scope and scan speed:

  1. Select **Scan Configuration**. From here, you can fine-tune Burp's scanner to suit different use cases and target sites.

  2. Select **Use a preset scan mode** or **Use a custom configuration**. The **Lightweight** scan mode is intended to give a high-level overview of a target as quickly as possible. Scans using this mode run for a maximum of 15 minutes.



let you trade off speed and coverage. Alternatively, you can select one or more custom configurations. Burp Scanner applies any selected configurations in order, enabling you to fine-tune scanning behaviour.

● Use a preset scan mode ○ Use a custom configuration

○ **Lightweight**
Gain fast feedback on a site's security - for when speed is a priority. Lightweight mode will complete within 15 minutes.

○ **Fast**
More thorough than a Lightweight scan, but still biased towards speed. Fast scans will generally complete within one hour.

○ **Balanced**
Provides a balance between coverage and speed. You will typically see the results of a Balanced scan within a few hours.

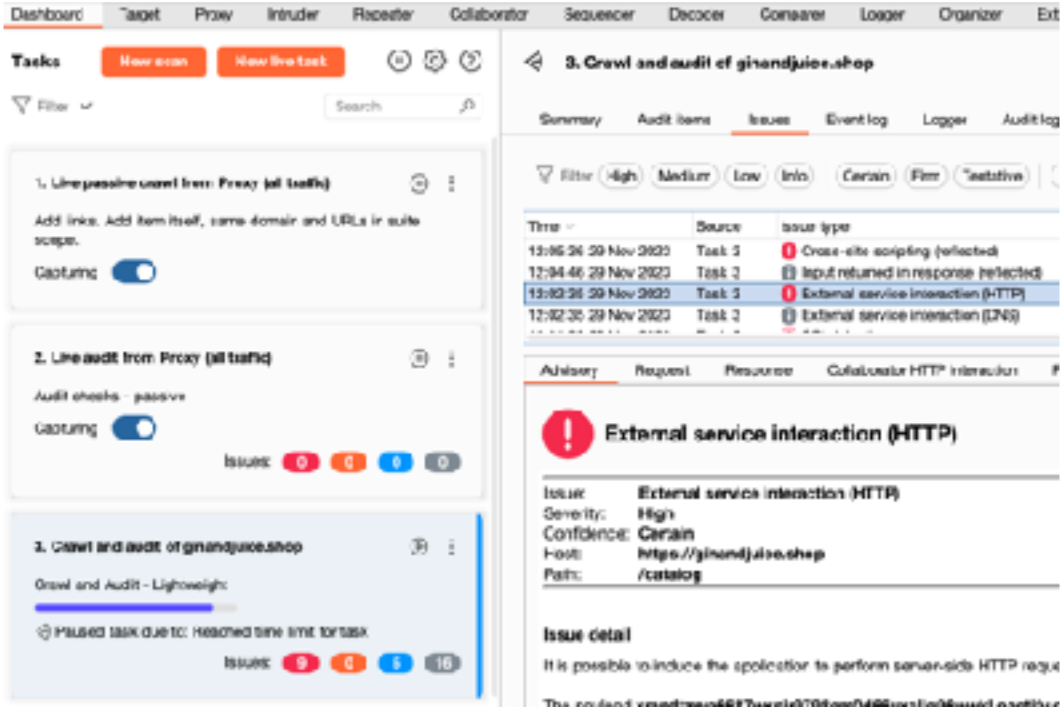
○ **Deep**
Achieve greater coverage and gain a better understanding of a site's security posture. Scanning time depends heavily on the target site's size and complexity.

☐ Remember my choice for future scans

Burp Suite- Scan Configuration

- **Run Scan**:

  1. Start the scan and monitor its progress in the dashboard.
  2. Review the results, focusing on critical web application vulnerabilities.



1. Live passive crawl from Proxy (all traffic)

Add links. Add item itself, same domain and URLs in suite scope.

Capturing

2. Live audit from Proxy (all traffic)

Audit checks - passive

Capturing

Issues: 0 0 0 0

3. Crawl and audit of ginandjuice.shop

Crawl and Audit - Lightweight

Paused task due to: Reached time limit for task

Issues: 9 0 5 16

3. Crawl and audit of ginandjuice.shop

Time	Source	Issue type
13:06:36 29 Nov 2023	Task 3	Cross-site scripting (reflected)
12:04:46 29 Nov 2023	Task 2	Input returned in response (reflected)
13:03:36 29 Nov 2023	Task 3	External service interaction (HTTP)
12:02:35 29 Nov 2023	Task 2	External service interaction (DNS)

External service interaction (HTTP)

Issue:	External service interaction (HTTP)
Severity:	High
Confidence:	Certain
Host:	https://ginandjuice.shop
Path:	/catalog

Issue detail

It is possible to induce the application to perform server-side HTTP reque

Burp Suite- Viewing Identified Issues

Pros:

Comprehensive web application testing capabilities.
Includes manual testing tools alongside automated scanning.
Regular updates and extensive support.

Cons:

Steeper Learning curve.
Can be expensive for the professional edition.

4. Nikto

Purpose:
Nikto is an open-source web server scanner designed to identify vulnerabilities, outdated software, and configuration issues on web servers.

How to Use:

- **Install Nikto**: Install on dedicated testing VMs using Kali Linux Terminal using the command:

```
  -$ sudo apt install nikto
```

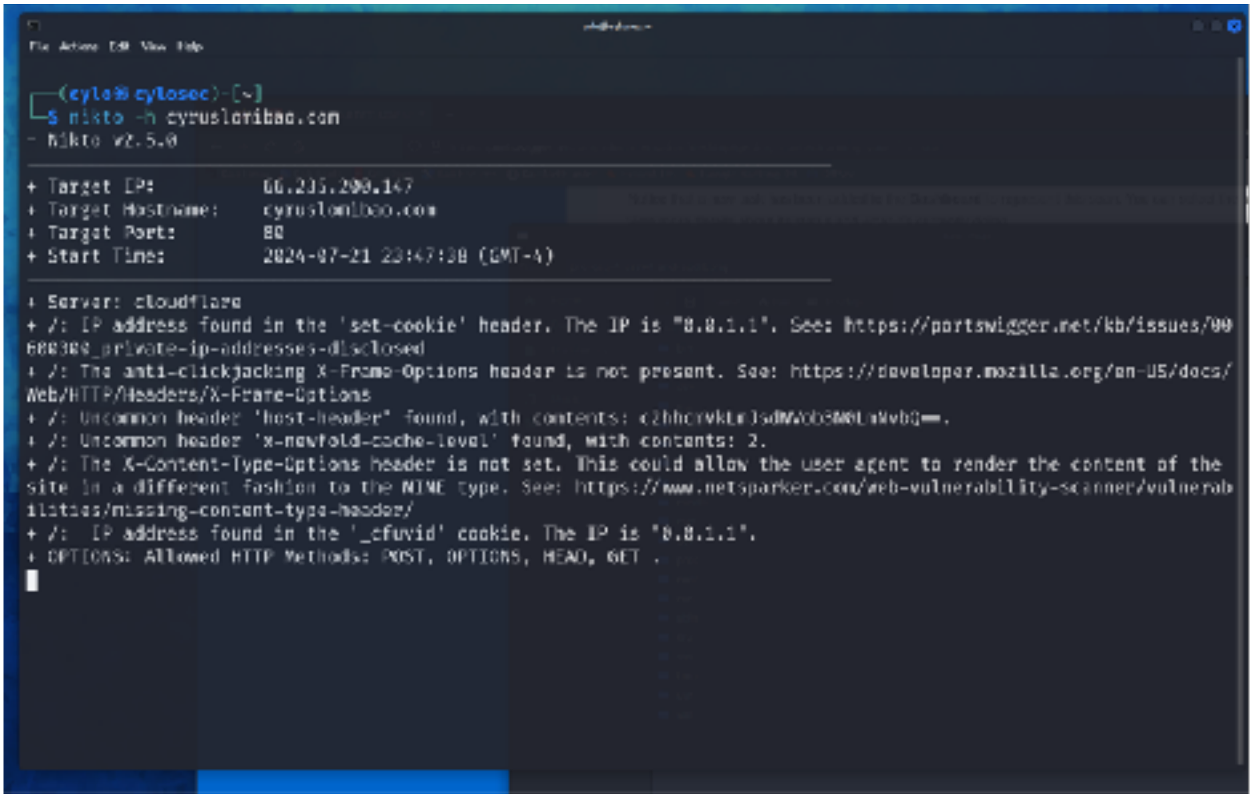
- Basic usage:

```
  $ nikto -h

   Options:
       -ask+               Whether to ask about submitting updates
                               yes   Ask about each (default)
                               no    Don't ask, don't send
                               auto  Don't ask, just send
       -Cgidirs+           Scan these CGI dirs: "none", "all", or values like "/cgi/ /
cgi-a/"
       -config+            Use this config file
       -Display+           Turn on/off display outputs:
                               1     Show redirects
                               2     Show cookies received
                               3     Show all 200/OK responses
                               4     Show URLs which require authentication
                               D     Debug output
                               E     Display all HTTP errors
                               P     Print progress to STDOUT
                               S     Scrub output of IPs and hostnames
                               V     Verbose output
       -dbcheck           Check database and other key files for syntax errors
       -followredirects   Follow 3xx redirects to new location
       -evasion+          Encoding technique:
                               1     Random URI encoding (non-UTF8)
                               2     Directory self-reference (/./)
                               3     Premature URL ending
                               4     Prepend long random string
                               5     Fake parameter
                               6     TAB as request spacer
                               7     Change the case of the URL
                               8     Use Windows directory separator (\)
                               A     Use a carriage return (0x0d) as a request spacer
                               B     Use binary value 0x0b as a request spacer
        -Format+           Save file (-o) format:
                               csv   Comma-separated-value
                               htm   HTML Format
                               msf+  Log to Metasploit
                               nbe   Nessus NBE format
                               txt   Plain text
                               xml   XML Format
                               (if not specified the format will be taken from the
file extension passed to -output)
       -Help              Extended help information
       -host+             Target host
       -IgnoreCode        Ignore Codes--treat as negative responses
       -id+               Host authentication to use, format is id:pass or
id:pass:realm
       -key+              Client certificate key file
       -list-plugins      List all available plugins, perform no testing
       -maxtime+          Maximum testing time per host
       -mutate+           Guess additional file names:
                               1     Test all files with all root directories
                               2     Guess for password file names
                               3     Enumerate user names via Apache (/~user type
requests)
                               4     Enumerate user names via cgiwrap (/cgi-bin/
cgiwrap/~user type requests)
                               5     Attempt to brute force sub-domain names, assume
that the host name is the parent domain
                               6     Attempt to guess directory names from the
supplied dictionary file
       -mutate-options    Provide information for mutates
       -nointeractive     Disables interactive features
       -nolookup          Disables DNS lookups
       -noslash           Strip trailing slash from URL (e.g., '/admin/' to '/admin')
       -nossl             Disables the use of SSL
       -no404             Disables nikto attempting to guess a 404 page
       -output+           Write output to this file ('.' for auto-name)
       -Pause+            Pause between tests (seconds, integer or float)
       -Plugins+          List of plugins to run (default: ALL)
       -port+             Port to use (default 80)
       -RSAcert+          Client certificate file
       -root+             Prepend root value to all requests, format is /directory
       -Save              Save positive responses to this directory ('.' for auto-
name)
       -ssl               Force ssl mode on port
       -Tuning+           Scan tuning:
                               1     Interesting File / Seen in logs
                               2     Misconfiguration / Default File
                               3     Information Disclosure
                               4     Injection (XSS/Script/HTML)
                               5     Remote File Retrieval - Inside Web Root
                               6     Denial of Service
                               7     Remote File Retrieval - Server Wide
                               8     Command Execution / Remote Shell
                               9     SQL Injection
                               0     File Upload
                               a     Authentication Bypass
                               b     Software Identification
                               c     Remote Source Inclusion
                               x     Reverse Tuning Options (i.e., include all except
specified)
       -timeout+          Timeout for requests (default 10 seconds)
       -Userdbs           Load only user databases, not the standard databases
                               all   Disable standard dbs and load only user dbs
                               tests Disable only db_tests and load udb_tests
       -until             Run until the specified time or duration
       -update            Update databases and plugins from CIRT.net
       -useproxy          Use the proxy defined in nikto.conf
       -usecookies        Use cookies from responses in future requests
       -Version           Print plugin and database versions
       -vhost+            Virtual host (for Host header)
              + requires a value
```

- **Configure Scan**:

    1. Launch Nikto and input the target web server URL.



```
┌──(cylo㉿cylosec)-[~]
└─$ nikto -h cyruslomibao.com
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          66.235.200.147
+ Target Hostname:    cyruslomibao.com
+ Target Port:        80
+ Start Time:         2024-07-21 23:47:38 (GMT-4)
---------------------------------------------------------------------------
+ Server: cloudflare
+ /: IP address found in the 'set-cookie' header. The IP is "0.0.1.1". See: https://portswigger.net/kb/issues/00
600300_private-ip-addresses-disclosed
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/
Web/HTTP/Headers/X-Frame-Options
+ /: Uncommon header 'host-header' found, with contents: c2hhcmVkLmJsdWVob3N0LmNvbQ==.
+ /: Uncommon header 'x-newfold-cache-level' found, with contents: 2.
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the
site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerab
ilities/missing-content-type-header/
+ /:  IP address found in the '_cfuvid' cookie. The IP is "0.0.1.1".
+ OPTIONS: Allowed HTTP Methods: POST, OPTIONS, HEAD, GET .
```

    2. Configure additional options, such as output format and scan depth.
    You can create a text file and list multiple domains for nikto to scan:

```
$ nano domain_list.txt // create a list of domains for nikto to scan
$ nikto -h domain_list.txt // runs a scan on all domains listed
$ nikto -h domain_list.txt -o domain_scan.csv -Format csv // outputs the
information into a csv file for later parsing and analysis
```

- **Run Scan:**

    1. Run scans using multiple options and monitor its progress.
    2. Output scans and document results.
    3. Analyze the results, focusing on vulnerabilities and misconfigurations, and research vulnerabilities with resources given by nikto.

Pros:

Free and open source.
Quick and easy to use.

Provides detailed information on web server vulnerabilities.

Cons:

- Limited to web servers.
- Can be noisy and easily detectable by web application firewalls.

Selection Reasoning:

- **Tenable Nessus**: Selected for its comprehensive vulnerability coverage and detailed reporting.
- **OpenVAS**: Chosen for its open-source nature and extensive vulnerability database.
- **Burp Suite**: Included for its powerful web application testing capabilities.
- **Cisco Security Scanner**: Essential for identifying vulnerabilities specific to Cisco devices during the transition to Fortinet.
- **Nikto**: Useful for quick identification of web server vulnerabilities and misconfigurations.

Potential Drawbacks and Limitations:

- **Resource Intensive**: Comprehensive scans can be time-consuming and require significant resources.
- **False Positives**: Tools like OpenVAS and Nikto may produce false positives, requiring additional validation.
- **Detection**: Some tools, particularly Nikto, can be easily detected by security systems, potentially alerting the target.

By carefully selecting and utilizing these tools, you can effectively identify vulnerabilities across different systems and applications, providing valuable insights for the subsequent phases of the penetration test.