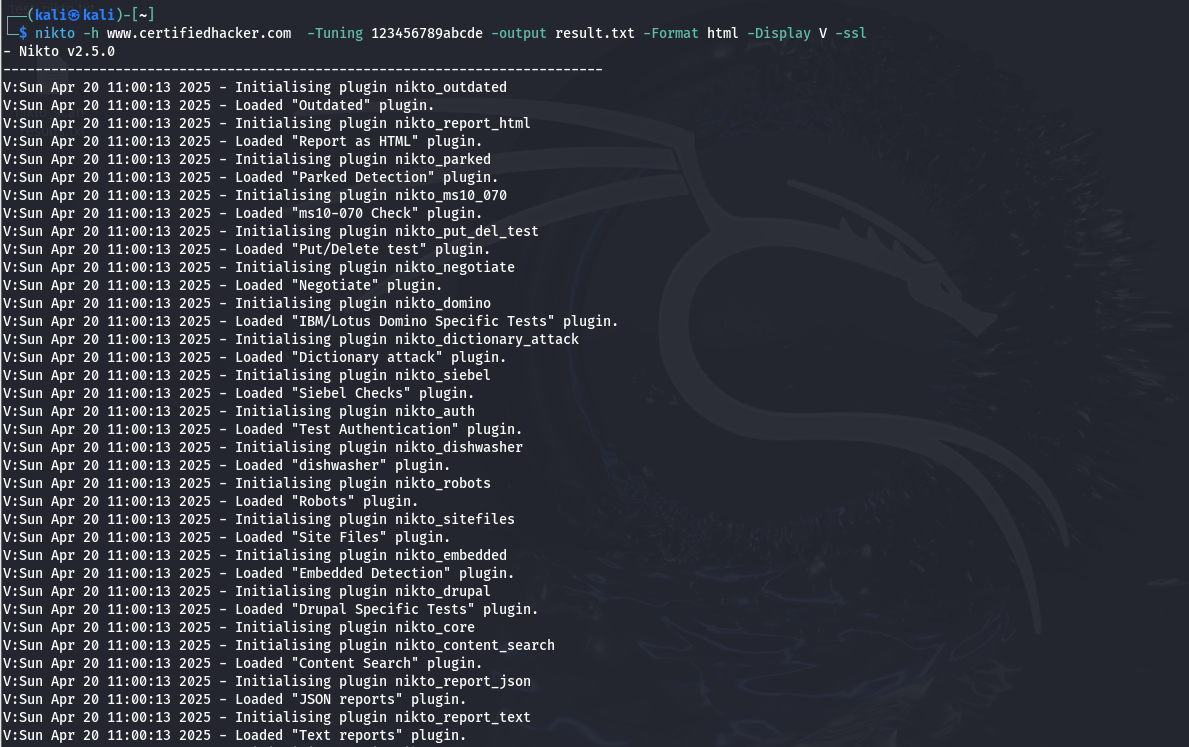
**1) Question:1**You are tasked with assessing the security of a web application for a prominent e-commerce company. The first step involves conducting a comprehensive scan of the target web application using Nikto. Once the scan is completed, you need to analyse the Nikto scan results to identify potential vulnerabilities, misconfigurations, or security issues. Finally, based on your findings, you must provide recommendations on how to address the identified vulnerabilities and security concerns in the web application.1:Describe your approach to performing a comprehensive scan of the target web application using Nikto. What specific settings, parameters, and options would you configure in Nikto to ensure a thorough assessment of the web application's security vulnerabilities? 2: Upon completion of the Nikto scan, you are presented with a detailed report outlining potential vulnerabilities, misconfigurations, and security concerns within the web application. How would you analyse these findings to prioritise the identified issues effectively? What criteria would you consider when determining the severity and potential impact of each vulnerability?3: Now that you have analysed the Nikto scan results and identified the web application's vulnerabilities, it's time to provide recommendations for remediation. How would you develop actionable recommendations tailored to address the specific vulnerabilities and security concerns identified in the assessment? Consider factors such as the criticality of the vulnerabilities, the feasibility of implementing fixes, and the potential impact on the organisation's overall security posture.You are tasked with assessing the security of a web application for a prominent e-commerce company. The first step involves conducting a comprehensive scan of the target web application using Nikto. Once the scan is completed, you need to analyse the Nikto scan results to identify potential vulnerabilities, misconfigurations, or security issues. Finally, based on your findings, you must provide recommendations on how to address the identified vulnerabilities and security concerns in the web application.

**Ans:**

Login kali Linux than app> vulnerabilities analyze>nikto

Here target is <http://www.certifiedhacker.com>, type nikto -h www.certifiedhacker.com -Tuning 123456789abcde -output result.txt -Format html -Display V -ssl and hit enter



**2**.Once the scan is complete, I’d follow a structured approach to analyze and prioritize vulnerabilities:

Analysis Strategy:

1.Categorize the Findings

* Group vulnerabilities by type:
* Server misconfigurations
* Outdated components
* Missing headers
* Exposed files or directories
* Potential injection points

**2.Determine Severity Level** Evaluate each issue based on:

* **CVSS Score** (if available)
* Exploitability (Can it be exploited remotely? Is authentication required?)
* Impact (Could it lead to data loss, RCE, privilege escalation?)
* Visibility (Is it accessible to unauthenticated users?)
* Whether it has **known exploits** in public databases (e.g., Exploit-DB)

3.Business Context Evaluation

* Is the vulnerability present on a critical page (e.g., login, checkout)?
* Does it affect sensitive data (PII, payment data)?
* Is the vulnerable asset internet-facing?

* Prioritization Criteria

|  |  |
| --- | --- |
| **Criteria** | **Example** |
| Critical | Admin panel exposed, RCE, SQL injection |
| High | Missing authentication on sensitive files |
| Medium | Missing headers (CSP, X-Frame-Options) |
| Low | Outdated version with no known critical CVEs |
| Informational | HTTP server banner disclosure |

3. Developing Actionable Recommendations for Remediation

* Remediation Strategy

To ensure recommendations are practical and effective, I would:

1. Tailor Fixes to the Environment

* Identify whether fixes involve config changes, code changes, or third-party updates.

1. Prioritize Based on Impact and Feasibility

* Critical/high issues addressed immediately
* Medium/low issues scheduled for regular update cycles
* Quick wins (e.g., enabling headers) implemented ASAP

1. Communicate with Developers/Admins

* Provide clear, concise, and technical recommendations
* Suggest testing before deployment to avoid breaking functionality

Sample Recommendation Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| **Issue** | **Recommendation** | **Justification** | **Feasibility** |
| Missing X-Frame-Options | Add header X-Frame-Options: DENY in server config | Prevent clickjacking attacks | Easy |
| Outdated Apache version | Upgrade to latest stable Apache | Patches known vulnerabilities | Medium |
| Directory browsing enabled | Disable directory listing in .htaccess or httpd.conf | Prevent leakage of file structure | Easy |
| Admin panel exposed | Restrict access via IP or authentication | Reduces risk of brute force or RCE | High |
| Detected test.php | Remove unused/test files from server | Eliminates unnecessary attack surface | Easy |

**Final Security Posture Enhancements**

* Introduce **Web Application Firewall (WAF)**
* Schedule **regular vulnerability scans**
* Perform **manual code review** or **penetration testing**
* Train developers in **secure coding practices**

**Summary**

By leveraging Nikto with comprehensive tuning options, categorizing and analyzing results effectively, and providing context-aware remediation plans, we ensure not only the identification of security flaws but also the practical implementation of solutions that improve the web application's overall security posture.