IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

Web Patrol: Comprehensive Web security Scanner

[1]RumanaAnjum, [2]AffanBaig, [3]Afreen Suraiya,[4]Misbah Sultana,[5]Sara Farheen
[1]Assitant Professor,[2345]Student
[12345]DepartmentofCSE
[12345] VVIET, Mysuru, India

Abstract:It is tedious for a Vulnerability Management Analyst to perform binge-tool-scanning (running security scanningtools one after the other) sans automation. Unless you are skilled at automating stuff, it is a arduous task to performbinge-scan for each and every engagement. The final analysis of this program is to solve this drawback throughautomation; viz running multiple scanning tools to scrutinize vulnerabilities, efficiently analyzing errors and givesstreamlinedresults; allof these inone Scanner**.**

Index Terms - **VulnerabilityManagementAnalyst,Binge-toolscanning,Sans,Automation**.

I. Introduction

Intheprogressinglandscapeofcybersecurity,theneedforvigorouswebapplicationsecurityisparamount.Asorganizations strive to protect sensitive data and maintain the integrity of their digital assets, a versatile and comprehensivesolution becomes essential. Introducing "**Comprehensive Web Security Scanner" –** a cutting-edge project designed toelevatewebsecuritytonewheightsAComprehensiveWebSecurityScannerisavitaltoolintheever-evolvinglandscapeofcybersecurity,designedtoproactivelyidentifyandmitigatepotentialvulnerabilitieswithinwebapplicationsandwebsites. As organisations increasingly rely on digital platforms for their operations, the need to ensure the security of theseonlineassetsbecomeparamount.Thistoolservesasaproactivedefencemechanism,systematicallyscanningwebapplications for potential weaknesses, vulnerabilities, and security loopholes that couldbe exploited by malicious users.The primary goal is to identify and address these issues before they can be leveraged for unauthorized access,databreaches, or othercyberthreats.

A Comprehensive Web Security Scanner typically employs a range of techniques to assess the security posture of webapplications. This may include automated vulnerability assessments, penetration testing, and analysis of code for potentialflaws.Itsystematicallyexamines variouslayersofwebapplications,suchastheapplicationlayer,networklayer,and serverinfrastructure, to provide a thorough handholistic assessment.

II. Literature Survey

[1] Nscanner: Vulnerabilities Detection Tool for Web Application R.UtayaSurian,NorAzlinaAbdRahman,YogeswaranNathan.(2020).

The increasing vulnerabilities of web applications to security attacks, proposing the Nscanner system to detect SQLi, XSSand malware. It emphasizes the prevalence of social engineering attacks, particularly phishing. Regional analysis indicateshigher vulnerability in Asian countries. A literature review compares vulnerability scanners like Acunetix, Wapiti andMetasploit.Findingsfrom aparticipantawarenesssurveysuggestlimitedknowledgeofwebvulnerabilities,witharecommendation formalwaredetection inNscanner.

Drawback: Automated scanning tools may generate false positives (indicating a vulnerability that doesn't exist) or falsenegatives (missing actual vulnerabilities). Depending on the complexity of web applications and the diversity of attackvectors, the toolmay notalways provide accurate results.

[2] VulScan:AWeb-BasedVulnerabilityMulti-ScannerforWeb. RajabMohammedimam,IfeOlalekanEbo,AbdullahisaAhmed. (2023).

Theoutcomeofaddressingthesewebsecurityvulnerabilitiesthrougheffectivewebvulnerabilityscanningincludesenhanced protectionagainstcyber threats,reduced riskofunauthorized access, databreaches,and service disruptions.Focusing on issues such as SQL injection, XSS, CSRF, SSL stripping, clickjacking, and DoS/DDoS attacks. Implementingrobust security measures can safeguard sensitive information, maintain the integrity of web applications, and contribute to amoresecureonlineenvironmentforbusinessesandindividuals.Addressingwebsecurityvulnerabilitiesthroughvulnerabilityscanningcontributes to amore resilient, trustworthy, and secure online presence.

Drawback: Overhead and Performance Impact: Running comprehensive vulnerability scans can be resource-intensive andmay impact the performance of the second web applications. This could be a concern for production environments wheremaintainingoptimal performance is crucial.

[3] VulnerabilityScannerforWebApplicationswithFirewallTechniquesRathod ,S.KJagtap,J.RSatpute, A.P.Shikhare,K.A.Pujari.(2022).

The survey paper introduces an automated web vulnerability scanner focusing on SQL Injection and Cross-Site Scripting.Thesystemgeneratescomprehensivereports,includingendpointdetailsandrecommendedremediation,aidingwebdevelopers in addressing security weaknesses. It emphasizes the prevalence of web-based attacks due to vulnerabilities,showcasing the need for improved security. The system, implemented by the authors, offers automated scanning, detailedreporting, and potential integration withmachine learning for enhancedvulnerabilitydetection.

Drawback: False Positive and Negatives: Automated scanners, including those focusing on SQL Injection and Cross-SiteScripting, are prone to generating false positives (indicating vulnerabilities that do not exist) or false negatives (missingactual vulnerabilities). The accuracyofthe scanner depends onthe sophistication ofit’sdetection algorithms.

[4] WebApplicationthroughComprehensiveVulnerabilityAssessment PrasanthSatyaSaiKiranGandikota,SushaniS,DeekshithaValluri,GopiKrishnaYanala.(2023).

Thispaperprovidesacomprehensiveoverviewofwebapplicationvulnerabilityassessmentandpenetrationtesting,underscoring the importance of proactive security measures. The focus is on safeguarding sensitive data and preservingapplication integrity, The study aims to identify and categorize web application vulnerabilities, specifically followingOWASP guideline’s, tomitigate the risk ofuserdata breaches.

Drawback: Complexity of Security Threats: Modern web applications face a variety of security threats beyond SQLInjection and Cross-Site Scripting. The tool may not effectively address more complex or nuanced vulnerabilities thatrequire a deeperunderstanding ofapplication logic andbehavior.

[5] VulnerabilityScanners:AProactiveApproachToAssessWebApplicationSecurity.Sheetal Bairwa,BhawnaMewara andJyotiGajrani.(2014).

The provided text appears to be an expert from a research paper or article discussing vulnerability scanners and their role inassessing web application security. The authors discuss various vulnerability assessment techniques, such as static analysis,attack graph analysis and the usage of different vulnerability scanners like Nmap, Nessus, AcunetixWVS, Nikto, and BurpSuite.Theauthorsalsopresentacomparativestudyofthesescannersbasedonthevulnerabilitiestheydetect,includingSQL injection, improper error management, cross-site scripting (XSS) , rogue servers, denial of service, remote codeexecution, and format string identifier, among others. The conclusion suggests that no single tool is capable of detecting alltypesofvulnerabilities,andintegratingdifferenttoolsmightprovideamorecomprehensiveviewofthesecuritypostureofawebapplication or network.

Drawback: Dependency on Regular Updates: The effectiveness of any vulnerability scanner relies on regular updates to itsvulnerability database. If the tool is not frequently updated to account for new vulnerabilities and attack techniques, it maybecome outdatedandless effective overtime.

[6] AStudyonWebApplicationSecurityandDetectingSecuritySandee pKumar,RenukaMahajan,NareshKumar. (2017).

This paper discusses the different aspects of web security and its weakness. The main elements of web security techniquessuch as the passwords, encryption, authentication and integrity are also discussed in this paper. The anatomy of webapplication attack and the attack techniques are also covered in details. This paper explores a number of methods forcombattingthisclassofthreatsandassesseswhytheyhavenotprovenmoresuccessful.Thispaperproposesabetterwayfor minimizing for minimizing these type of web vulnerabilities. It also provides the best security mechanisms for the saidattacks.

Drawback: Thepotentialintegrationwithmachinelearningforenhancedvulnerabilitydetectioncouldintroducecomplexities in terms of implementation, maintenance, and tuning. Machine learning models require continuous trainingand adaptation toevolving threats.

[7] EffectiveWebApplicationVulnerabilityTestingSystemProposedXSS_SQL_Scanning_Algorithm. (2020).ThinzarAung,ZinThuThuMyint

The research paper proposes an innovative approach, the XSS, SQL Scanning Algorithm, for detecting vulnerabilities inweb applications, with a focus on SQL injection and Cross-Site Scripting (XSS) attacks. The authors emphasize theincreasing use of web applications and the potential security risks associated with coding errors. The proposed algorithmintegrates crawling, payload forwarding, and response analysis, leveraging the Naïvepattern matching algorithm forefficient detection. The study compares the algorithms performance with the well-known Acunetix, making it a lightweightyetreliablesolution.Theresearchcontributestoenhancingwebsecuritybyprovidingasolution.Theresearchcontributesto enhancing web application security by providing a customizable, accurate, and the efficient the algorithm to coveradditional vulnerabilities beyondSQLinjection andXSS.

Drawback: Users of automated vulnerability scanners may develop a false sense of security, assuming that the toolidentifies all possible vulnerabilities. It's crucial to communicate that these tools are part of a border security strategy andnota comprehensive solution.

[8] DetectionofXSSVulnerabilitiesUsingSecurityTestingApproaches.Sanjuk ta Mohanty, ArupAbhimaAcharya.(2021).

This survey paper pioneers an advancedmethodology for detecting Cross-Sie Scripting (XSS) vulnerabilities in webapplications, introducing a novel combination of static taint analysis and evolutionary genetic algorithms (GA). In contrastto prior studies, which often overlooked false negatives in source code, the proposed approach integrates static analysis toidentifypotentialfalsenegativesandemploysGAtogeneratetargetedtestcases,effectivelyexposinggenuinevulnerabilities. This innovative fusion significantly enhances the precision and efficiency of XSS vulnerability detection.Thestudy'sbroadercontributionliesinemphasizingthecriticalneedtoaddressfalsenegativesinXSSdetection,strategically combining static and dynamic analysis techniques to overcome individual limitations. The introduced securityframework synthesizes insights from prior research, aiming to elevate the overall accuracy of vulnerability identification inthe realmofwebapplication security.

Drawback: Real-world Attacks: Automated scanners may not replicate the sophisticated techniques employed by real-world attackers. They may lack the creativity and adaptability needed to uncover vulnerabilities that go beyond knownpatterns.

[9] VulnerabilitiesandSecurityofWebApplications. DivyaniYadav,Deeksha Gupta,DhananjaySingh,DevendraKumar,UpasanaSharma.(2018).

This survey paper pioneers an advanced methodology for detecting Cross-Site Scripting(XSS) vulnerabilities in web applications, introducing a novel combination of static taint analysis and evolutionary geneticalgorithms (GA). In contrast to prior studies, which often overlooked false negatives in source code, the proposed approachintegratesstatictaintanalysistoidentifypotentialfalsenegativesandemploysGAtogeneratetargetedtestcases,effectively exposing genuinevulnerabilities.Thisinnovativefusion significantly enhancestheprecision andefficacy ofXSSvulnerabilitydetection.Thestudy'sbroadercontributionliesinemphasizingthecriticalneedtoaddressfalsenegativesinXSSdetection,strategicallycombiningstaticanddynamicanalysistechniquestoovercomeindividuallimitations. The introduced security framework synthesizes insights from prior research, aiming to elevate the overallaccuracyofvulnerability identification inthe realmofwebapplication security.

Drawback: The useofvulnerability scannersshouldalignwith ethicalstandards. Scanningsystemswithout properauthorization may raise ethical concerns, and organizations need to ensure they have the right to scan the targeted webapplications.

[10] OWASPTenDrivenSurvey onWebApplicationProtectionMethods. OmarCheikhrouhou, MoezKrichen,HabibHamam&AbdelouahidDerhab.(2021).

Web applications (WAs) are constantly evolving and deployed at broad scale. However, they are exposed to a variety ofattacks. The biggest challenge facing organizations is how to develop a WA that fulfills their requirements with respect tosensitive data exchange, E-commerce, and secure workflows. This paper identifies the most critical web vulnerabilities.Integration with IoT devices enhances the overall security ecosystem, extending protection beyond traditional boundaries.Automated rule creation facilitatesefficient handling of new attack scenarios, providing a proactive security approach.
Theframeworkoffersdetailedreporting,providinginsightsintopotentialvulnerabilitiesandaidingsecurityexpertsanddevelopers.

Drawback: Scalabilityconcernsmayariseastheframeworkevolves,requiringcarefulconsiderationforlarge-scaledeployment. Machine learning introduces the risk of false positives and negatives, decussating ongoing refinement andvalidation.

III. Proposed System

WebPatrol is an automates web security scanner designed to identify vulnerabilities in web applications, It streamlines the penetration testing process, providing efficient and accurate results. The system integrates multiple scanning tools, eliminating the need for manual execution and reducing human error. The frontend is a user-friendly web interface allows penetration testers to configure scans, view results, and generate reports. The backend has the Scanner Orchestrator, which coordinates the execution of various security scanners and manages tool selection, input parameters and result aggregation. The Scanner Modules where each module corresponds to a specific security tool and these modules perform target web application. The Databases stores scan results and vulnerability details along with historical data. The Reporting Engines generates comprehensive reports for stakeholders. Comprehensive web security scanner streamlines the penetration testing process through automation and efficient scanning. Pen testers input the target URL, authentication credentials and scan parameters, selecting desired modules. The system orchestrates selected scanners, analysing the web application for vulnerabilities. Results are aggregated, cross-references to identify false positives and prioritized based on severity. Comprehensive reports including an executive summary and detailed vulnerability description, aid in effective risk mitigation.

IV. Block Diagram

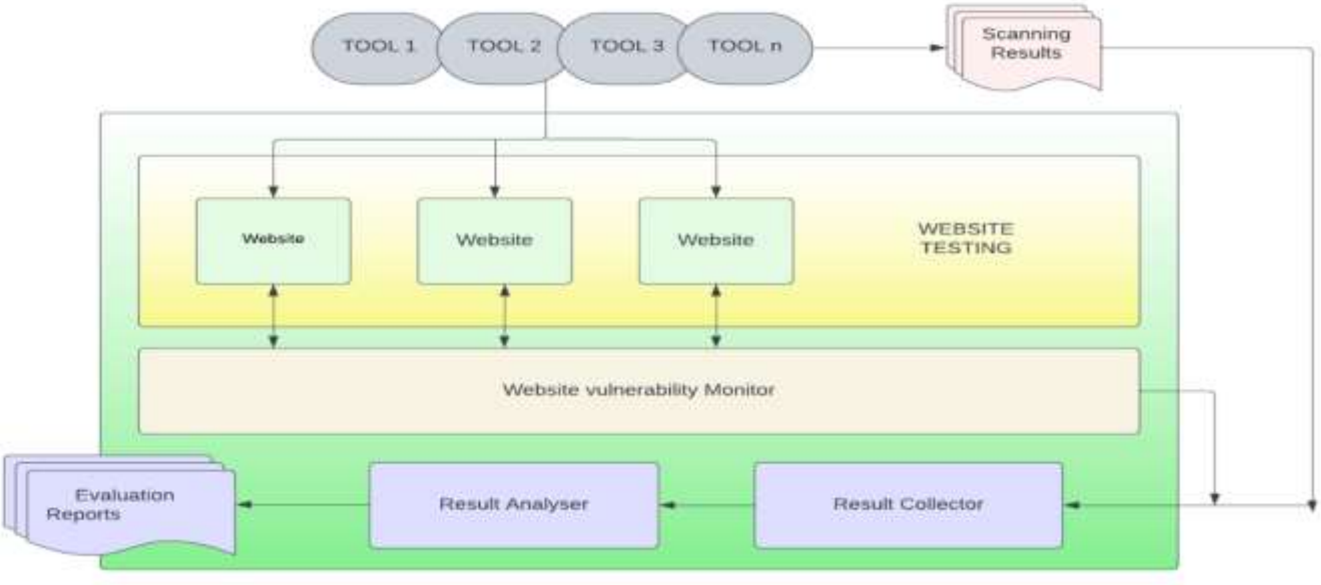
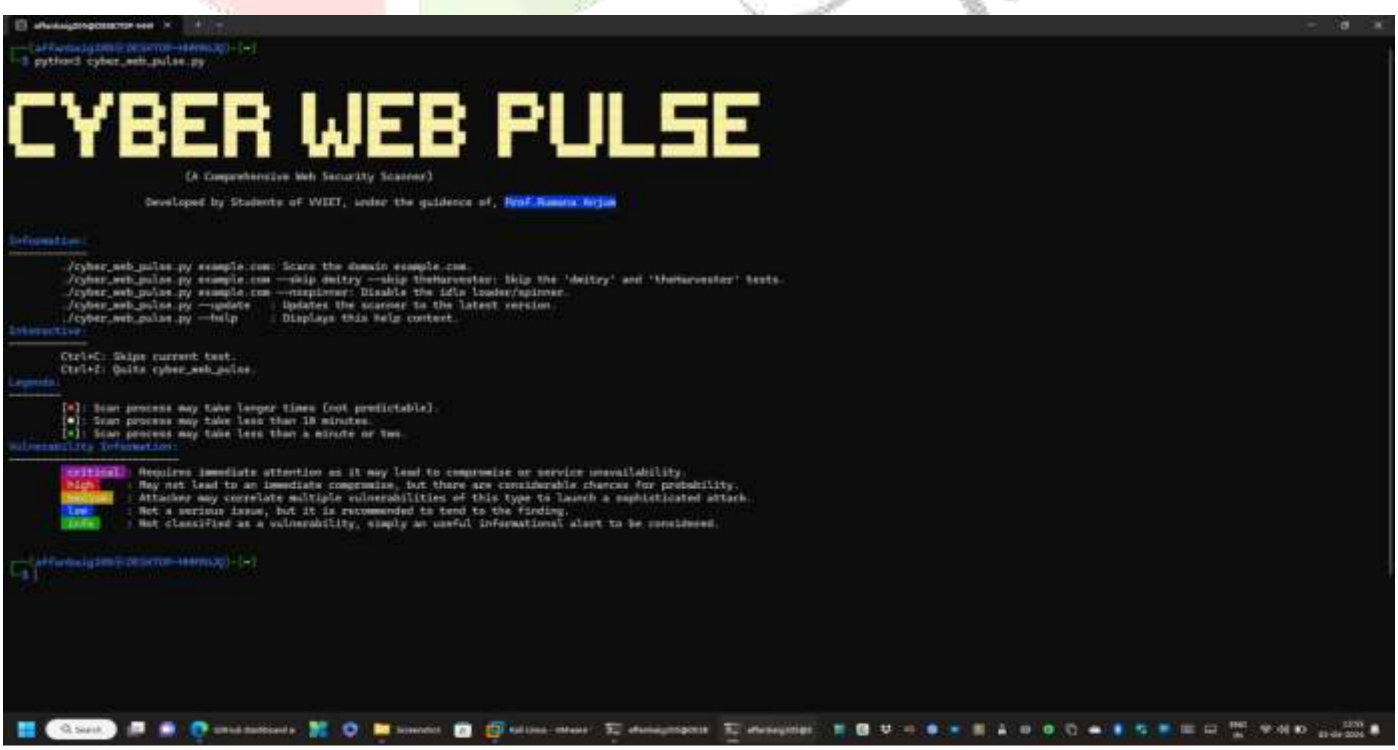


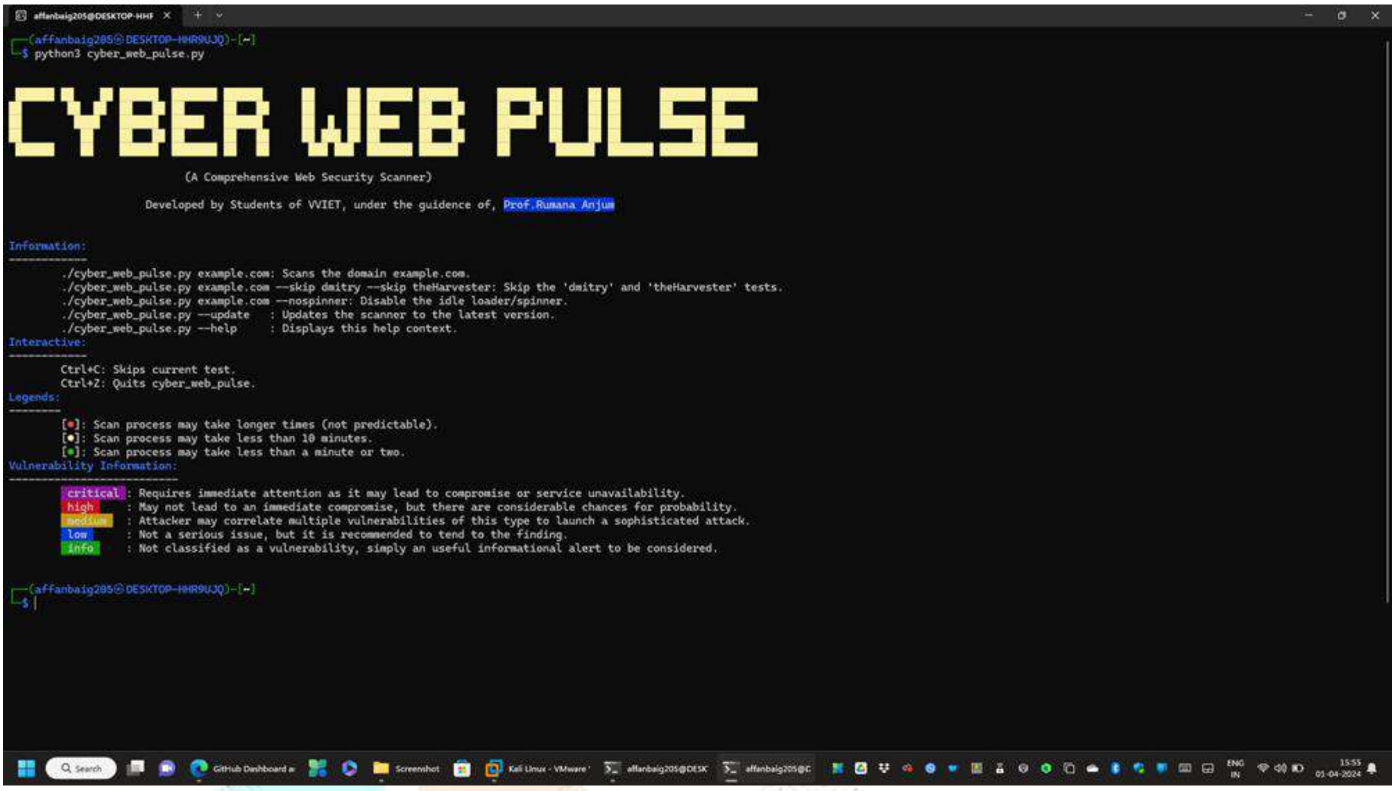
Fig 1. Block Diagram

V. Results



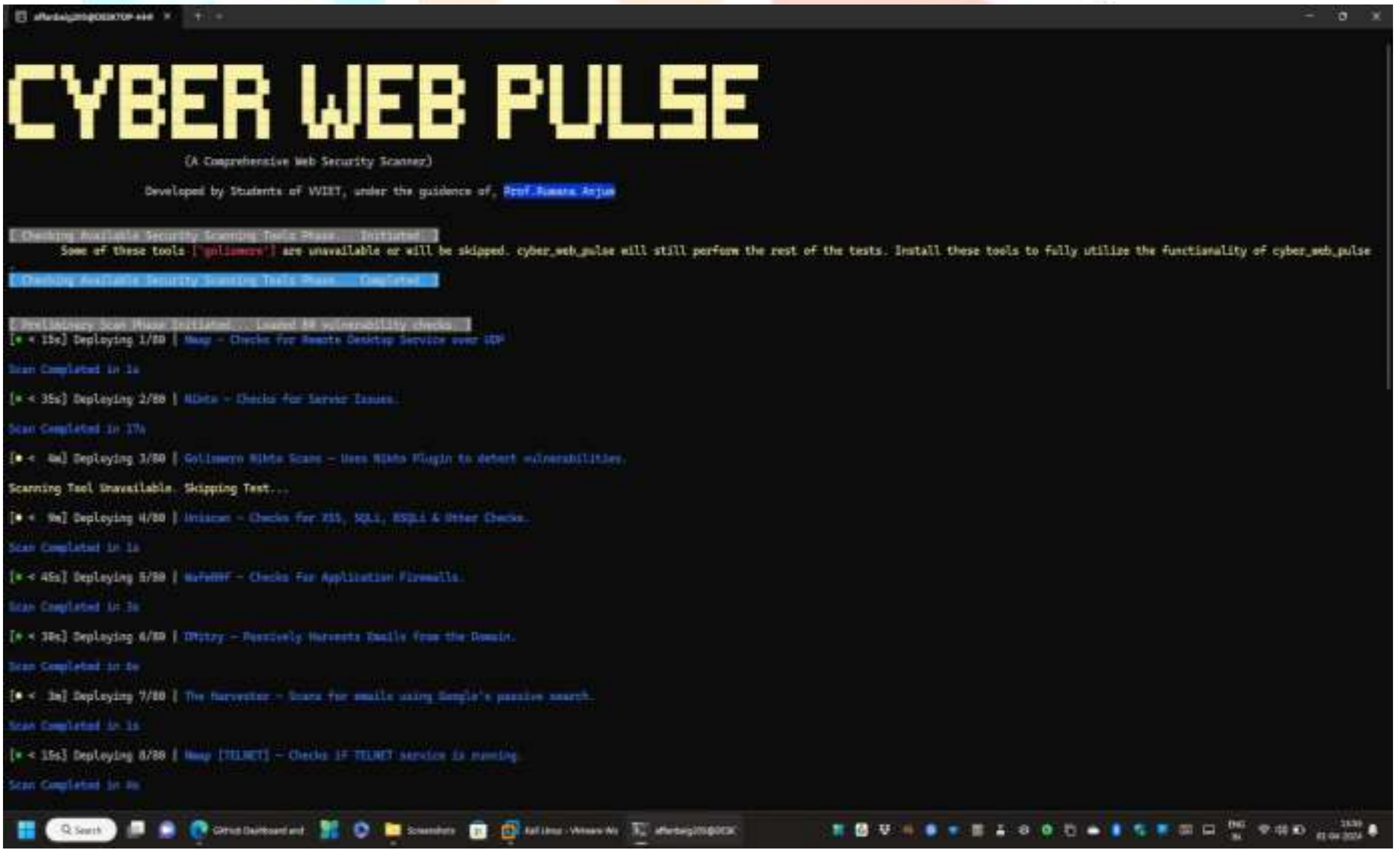
```
┌──([illegible])-[~]
└─$ python3 cyber_web_pulse.py

CYBER WEB PULSE
                    (A Comprehensive Web Security Scanner)

          Developed by Students of VVIET, under the guidence of, Prof.Ramana Arjun

Information:
------------
        ./cyber_web_pulse.py example.com: Scans the domain example.com.
        ./cyber_web_pulse.py example.com --skip dmitry --skip theHarvester: Skip the 'dmitry' and 'theHarvester' tests.
        ./cyber_web_pulse.py example.com --nospinner: Disable the idle loader/spinner.
        ./cyber_web_pulse.py --update    : Updates the scanner to the latest version.
        ./cyber_web_pulse.py --help      : Displays this help context.
Interactive:
------------
        Ctrl+C: Skips current test.
        Ctrl+Z: Quits cyber_web_pulse.
Legends:
--------
        [●]: Scan process may take longer times (not predictable).
        [●]: Scan process may take less than 10 minutes.
        [●]: Scan process may take less than a minute or two.
Vulnerability Information:
--------------------------
        critical : Requires immediate attention as it may lead to compromise or service unavailability.
        high     : May not lead to an immediate compromise, but there are considerable chances for probability.
        medium   : Attacker may correlate multiple vulnerabilities of this type to launch a sophisticated attack.
        low      : Not a serious issue, but it is recommended to tend to the finding.
        info     : Not classified as a vulnerability, simply an useful informational alert to be considered.

┌──([illegible])-[~]
└─$
```

Fig 2: Welcome Page



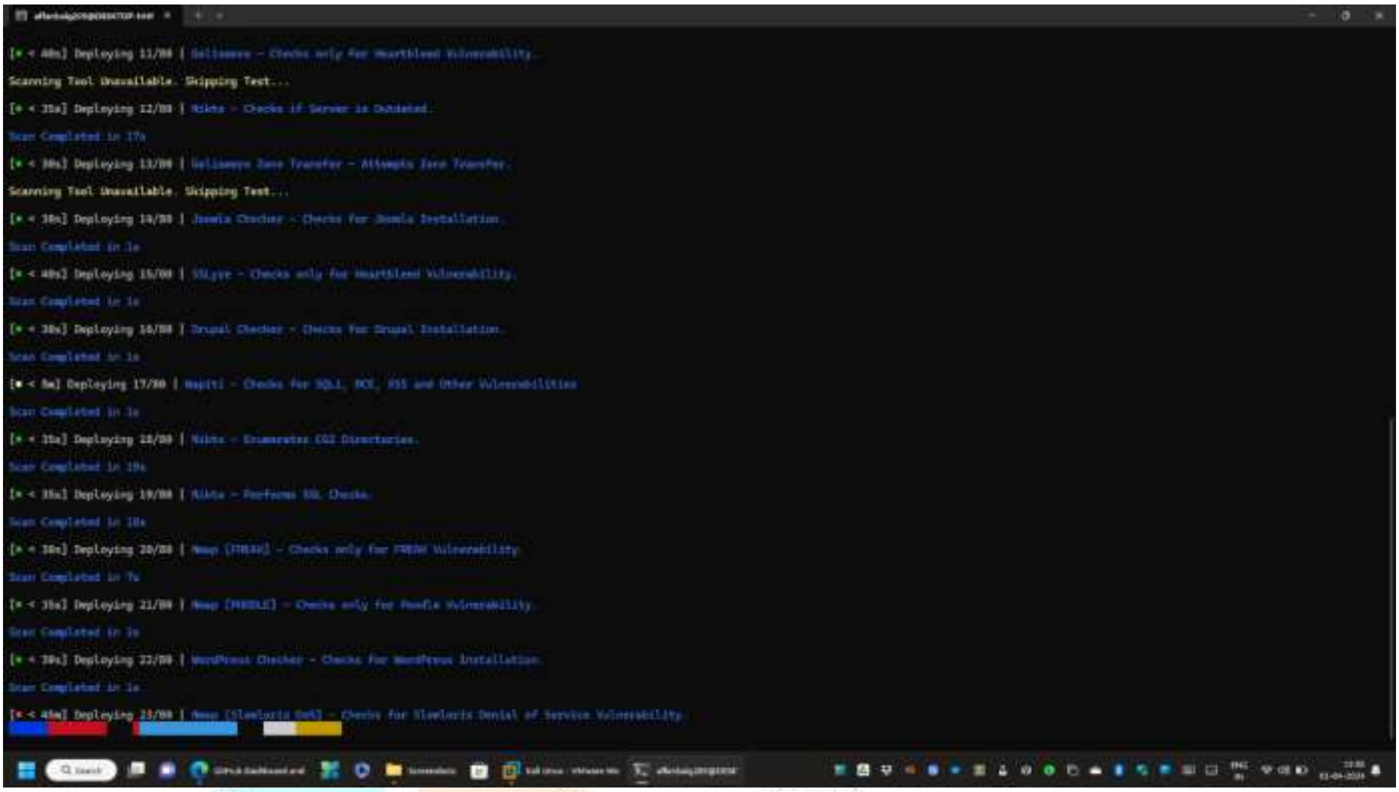
```
┌──(affanbaig205㉿DESKTOP-HHR9UJQ)-[~]
└─$ python3 cyber_web_pulse.py

CYBER WEB PULSE

                    (A Comprehensive Web Security Scanner)

              Developed by Students of VVIET, under the guidence of, Prof.Rumana Anjum

Information:
------------
        ./cyber_web_pulse.py example.com: Scans the domain example.com.
        ./cyber_web_pulse.py example.com --skip dmitry --skip theHarvester: Skip the 'dmitry' and 'theHarvester' tests.
        ./cyber_web_pulse.py example.com --nospinner: Disable the idle loader/spinner.
        ./cyber_web_pulse.py --update   : Updates the scanner to the latest version.
        ./cyber_web_pulse.py --help     : Displays this help context.
Interactive:
------------
        Ctrl+C: Skips current test.
        Ctrl+Z: Quits cyber_web_pulse.
Legends:
--------
        [●]: Scan process may take longer times (not predictable).
        [●]: Scan process may take less than 10 minutes.
        [●]: Scan process may take less than a minute or two.
Vulnerability Information:
--------------------------
        critical : Requires immediate attention as it may lead to compromise or service unavailability.
        high     : May not lead to an immediate compromise, but there are considerable chances for probability.
        medium   : Attacker may correlate multiple vulnerabilities of this type to launch a sophisticated attack.
        low      : Not a serious issue, but it is recommended to tend to the finding.
        info     : Not classified as a vulnerability, simply an useful informational alert to be considered.

┌──(affanbaig205㉿DESKTOP-HHR9UJQ)-[~]
└─$ |
```

Fig 3. Project Update



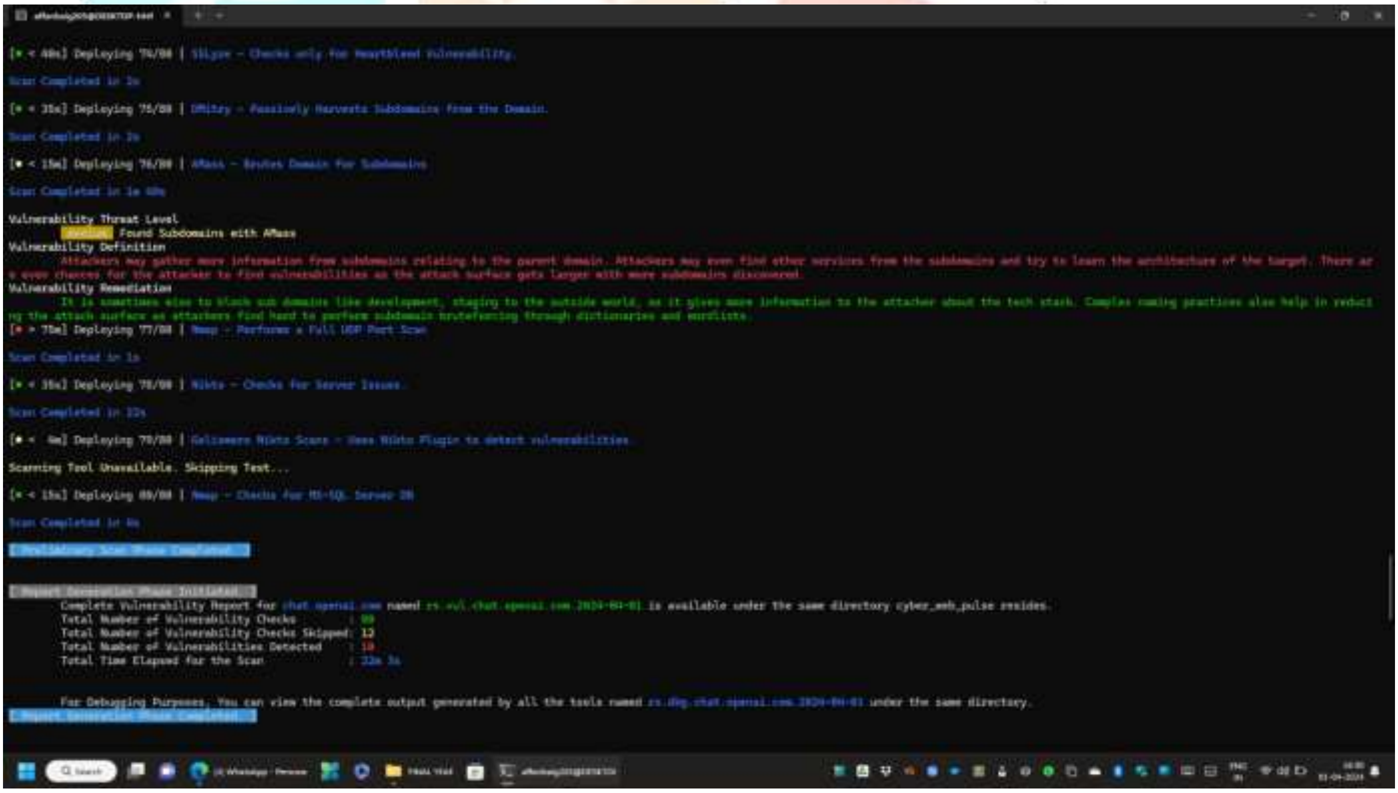
```
CYBER WEB PULSE

                    (A Comprehensive Web Security Scanner)

              Developed by Students of VVIET, under the guidence of, Prof.Rumana Anjum

[ Checking Available Security Scanning Tools Phase... Initiated. ]
        Some of these tools ['golismero'] are unavailable or will be skipped. cyber_web_pulse will still perform the rest of the tests. Install these tools to fully utilize the functionality of cyber_web_pulse
.
[ Checking Available Security Scanning Tools Phase... Completed. ]

[ Preliminary Scan Phase Initiated... Loaded 80 vulnerability checks. ]
[● < 15s] Deploying 1/80 | Nmap - Checks for Remote Desktop Service over UDP

Scan Completed in 1s

[● < 35s] Deploying 2/80 | Nikto - Checks for Server Issues.

Scan Completed in 17s

[● <  4m] Deploying 3/80 | Golismero Nikto Scans - Uses Nikto Plugin to detect vulnerabilities.

Scanning Tool Unavailable. Skipping Test...

[● <  9m] Deploying 4/80 | Uniscan - Checks for XSS, SQLi, BSQLi & Other Checks.

Scan Completed in 1s

[● < 45s] Deploying 5/80 | WafW00f - Checks for Application Firewalls.

Scan Completed in 3s

[● < 30s] Deploying 6/80 | DMitry - Passively Harvests Emails from the Domain.

Scan Completed in 6s

[● <  3m] Deploying 7/80 | The Harvester - Scans for emails using Google's passive search.

Scan Completed in 1s

[● < 15s] Deploying 8/80 | Nmap [TELNET] - Checks if TELNET service is running.

Scan Completed in 4s
```

Fig 4. Scanning of Website for Vulnerabilities



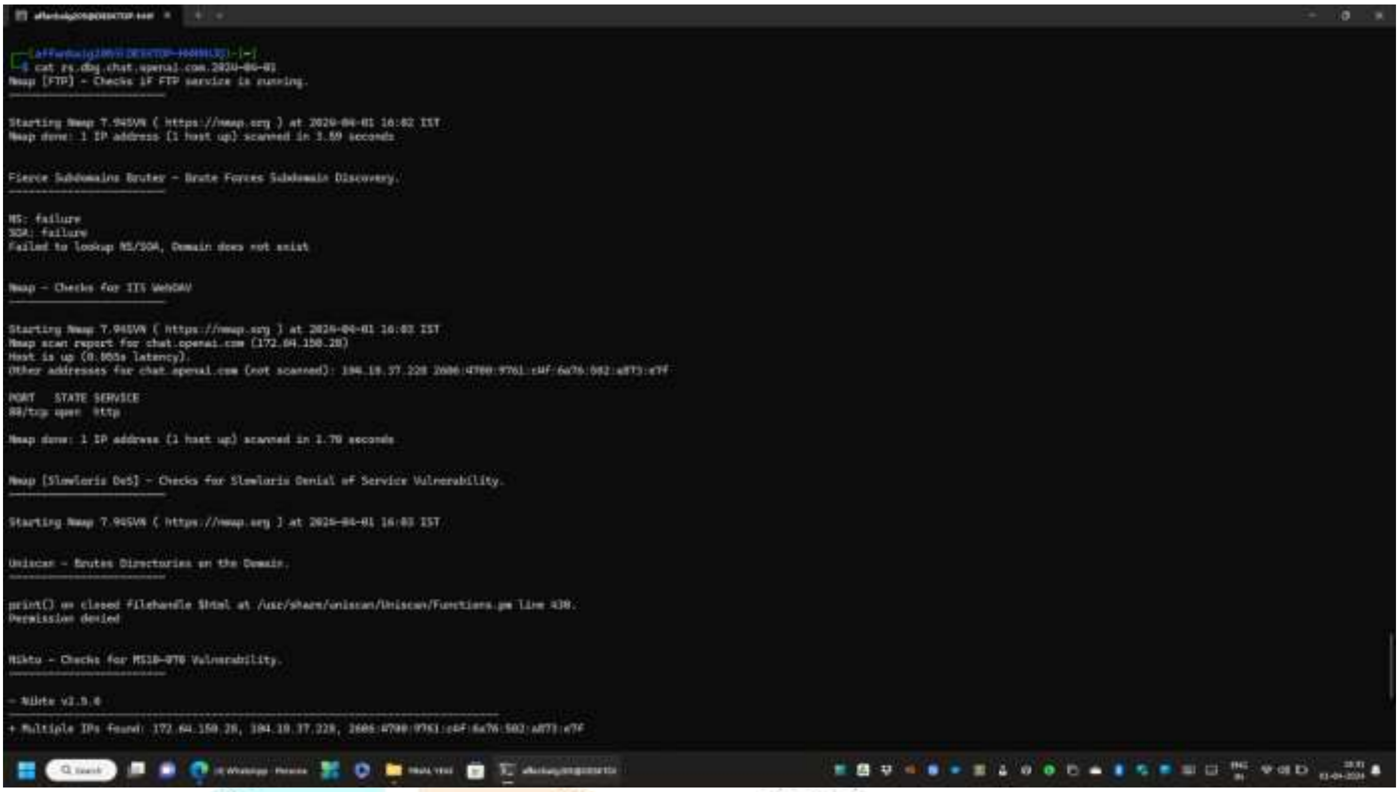
```
[● < 40s] Deploying 11/80 | Golismero - Checks only for Heartbleed Vulnerability.
Scanning Tool Unavailable. Skipping Test...
[● < 35s] Deploying 12/80 | Nikto - Checks if Server is Outdated.
Scan Completed in 17s
[● < 30s] Deploying 13/80 | Golismero Zone Transfer - Attempts Zone Transfer.
Scanning Tool Unavailable. Skipping Test...
[● < 30s] Deploying 14/80 | Joomla Checker - Checks for Joomla Installation.
Scan Completed in 1s
[● < 40s] Deploying 15/80 | SSLyze - Checks only for Heartbleed Vulnerability.
Scan Completed in 1s
[● < 30s] Deploying 16/80 | Drupal Checker - Checks for Drupal Installation.
Scan Completed in 1s
[● < 5m] Deploying 17/80 | Wapiti - Checks for SQLi, RCE, XSS and Other Vulnerabilities
Scan Completed in 1s
[● < 35s] Deploying 18/80 | Nikto - Enumerates CGI Directories.
Scan Completed in 19s
[● < 35s] Deploying 19/80 | Nikto - Performs SSL Checks.
Scan Completed in 18s
[● < 30s] Deploying 20/80 | Nmap [FREAK] - Checks only for FREAK Vulnerability.
Scan Completed in 7s
[● < 35s] Deploying 21/80 | Nmap [POODLE] - Checks only for Poodle Vulnerability.
Scan Completed in 1s
[● < 30s] Deploying 22/80 | WordPress Checker - Checks for WordPress Installation.
Scan Completed in 1s
[● < 45m] Deploying 23/80 | Nmap [Slowloris DoS] - Checks for Slowloris Denial of Service Vulnerability.
```

Fig 5. Scanning of Website for Vulnerabilities



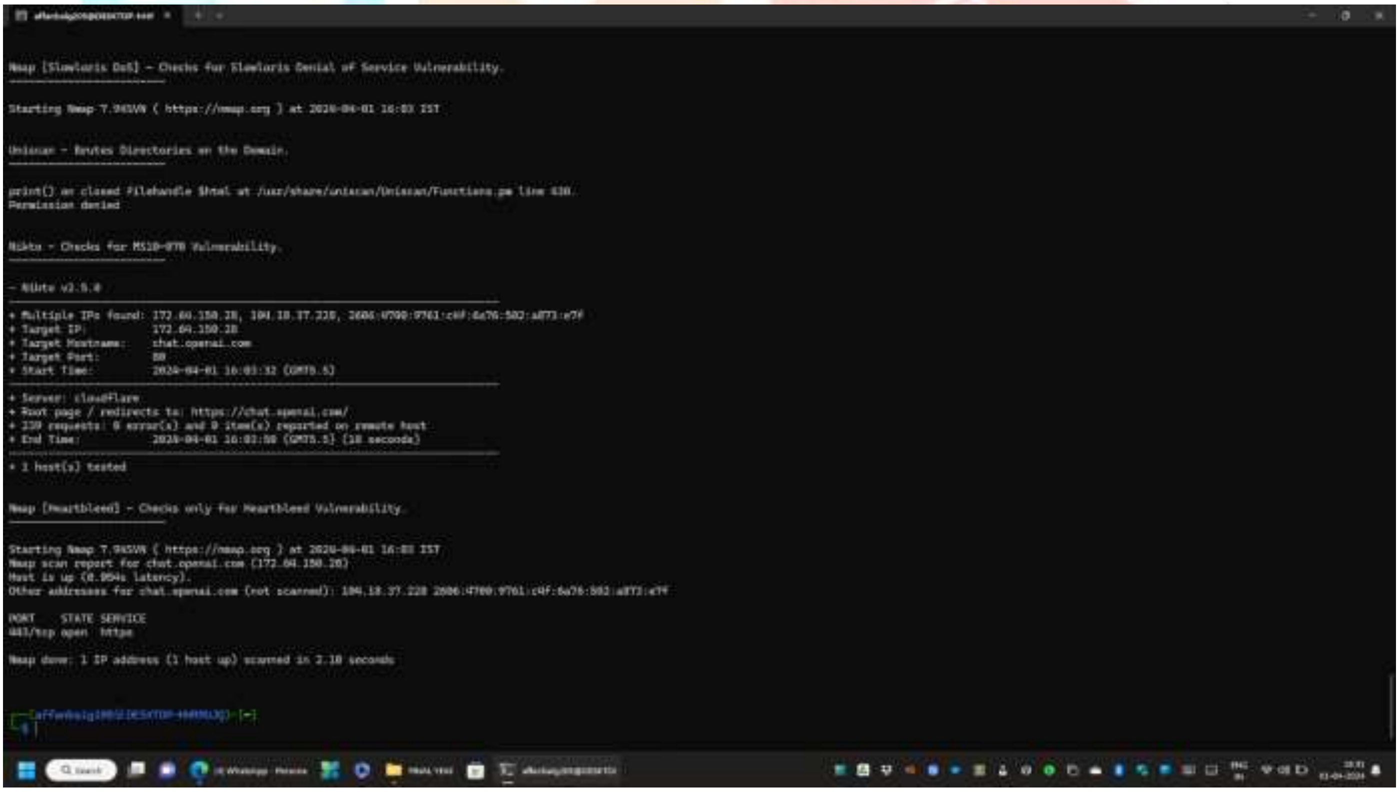
```
[● < 40s] Deploying 74/80 | SSLyze - Checks only for Heartbleed Vulnerability.
Scan Completed in 2s
[● < 35s] Deploying 75/80 | DMitry - Passively Harvests Subdomains from the Domain.
Scan Completed in 2s
[● < 15m] Deploying 76/80 | AMass - Brutes Domain for Subdomains
Scan Completed in 1m 48s
Vulnerability Threat Level
        medium  Found Subdomains with AMass
Vulnerability Definition
        Attackers may gather more information from subdomains relating to the parent domain. Attackers may even find other services from the subdomains and try to learn the architecture of the target. There ar
e even chances for the attacker to find vulnerabilities as the attack surface gets larger with more subdomains discovered.
Vulnerability Remediation
        It is sometimes wise to block sub domains like development, staging to the outside world, as it gives more information to the attacker about the tech stack. Complex naming practices also help in reduci
ng the attack surface as attackers find hard to perform subdomain bruteforcing through dictionaries and wordlists.
[● > 75m] Deploying 77/80 | Nmap - Performs a Full UDP Port Scan
Scan Completed in 1s
[● < 35s] Deploying 78/80 | Nikto - Checks for Server Issues.
Scan Completed in 22s
[● <  4m] Deploying 79/80 | Golismero Nikto Scans - Uses Nikto Plugin to detect vulnerabilities.
Scanning Tool Unavailable. Skipping Test...
[● < 15s] Deploying 80/80 | Nmap - Checks for MS-SQL Server DB
Scan Completed in 8s
[ Preliminary Scan Phase Completed. ]


[ Report Generation Phase Initiated. ]
        Complete Vulnerability Report for chat.openai.com named rs.vul.chat.openai.com.2024-04-01 is available under the same directory cyber_web_pulse resides.
        Total Number of Vulnerability Checks          : 80
        Total Number of Vulnerability Checks Skipped: 12
        Total Number of Vulnerabilities Detected      : 10
        Total Time Elapsed for the Scan               : 32m 3s


        For Debugging Purposes, You can view the complete output generated by all the tools named rs.dbg.chat.openai.com.2024-04-01 under the same directory.
[ Report Generation Phase Completed. ]
```

Fig 6. Final Result



```
┌──(affanbaig1005㉿DESKTOP-H4MMLJQ)-[~]
└─$ cat rs.dbg.chat.openai.com.2024-04-01
Nmap [FTP] - Checks if FTP service is running.
------------------------------

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-01 16:02 IST
Nmap done: 1 IP address (1 host up) scanned in 3.59 seconds


Fierce Subdomains Bruter - Brute Forces Subdomain Discovery.
------------------------------

NS: failure
SOA: failure
Failed to lookup NS/SOA, Domain does not exist


Nmap - Checks for IIS WebDAV
------------------------------

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-01 16:03 IST
Nmap scan report for chat.openai.com (172.64.150.28)
Host is up (0.055s latency).
Other addresses for chat.openai.com (not scanned): 104.18.37.228 2606:4700:9761:c4f:6a76:502:a8f3:e7f

PORT   STATE SERVICE
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 1.70 seconds


Nmap [Slowloris DoS] - Checks for Slowloris Denial of Service Vulnerability.
------------------------------

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-01 16:03 IST


Uniscan - Brutes Directories on the Domain.
------------------------------

print() on closed filehandle $html at /usr/share/uniscan/Uniscan/Functions.pm line 438.
Permission denied


Nikto - Checks for MS10-070 Vulnerability.
------------------------------

- Nikto v2.5.0
---------------------------------------------------------------------------
+ Multiple IPs found: 172.64.150.28, 104.18.37.228, 2606:4700:9761:c4f:6a76:502:a8f3:e7f
```

Fig 7. Report of Website under Scan



```
Nmap [Slowloris DoS] - Checks for Slowloris Denial of Service Vulnerability.
------------------------------

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-01 16:03 IST


Uniscan - Brutes Directories on the Domain.
------------------------------

print() on closed filehandle $html at /usr/share/uniscan/Uniscan/Functions.pm line 438.
Permission denied


Nikto - Checks for MS10-070 Vulnerability.
------------------------------

- Nikto v2.5.0
---------------------------------------------------------------------------
+ Multiple IPs found: 172.64.150.28, 104.18.37.228, 2606:4700:9761:c4f:6a76:502:a8f3:e7f
+ Target IP:          172.64.150.28
+ Target Hostname:    chat.openai.com
+ Target Port:        80
+ Start Time:         2024-04-01 16:03:32 (GMT5.5)
---------------------------------------------------------------------------
+ Server: cloudflare
+ Root page / redirects to: https://chat.openai.com/
+ 239 requests: 0 error(s) and 0 item(s) reported on remote host
+ End Time:           2024-04-01 16:03:50 (GMT5.5) (18 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested


Nmap [Heartbleed] - Checks only for Heartbleed Vulnerability.
------------------------------

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-01 16:03 IST
Nmap scan report for chat.openai.com (172.64.150.28)
Host is up (0.054s latency).
Other addresses for chat.openai.com (not scanned): 104.18.37.228 2606:4700:9761:c4f:6a76:502:a8f3:e7f

PORT    STATE SERVICE
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 2.18 seconds


┌──(affanbaig1005㉿DESKTOP-H4MMLJQ)-[~]
└─$ 
```

Fig 8. Report of the Website under Scan

VI. Conclusion

Project offer a powerful approach to enhance web application security. It’s a robust method to boost web application. Security having diverse tools and techniques for a thorough and effective vulnerability identification, with a streamlined and accurate process for addressing vulnerabilities efficiently. Valuable for organizations dedicated to safeguarding their web applications and data.

VII. Acknowledgement

We wish to thank everyone who anonymously participated in the user studies.

VIII. References

[1]. R.UtayaSurian, Nor Azlina Abd Rahman, Yogeswaran Nathan. (2020). Nscanner: Vulnerabilities Detection Tool for Web Application
[2]. Rajab Mohammed imam, Ife Olalekan Ebo, Abdullahi isa Ahmed. (2023). VulScan: A Web-Based Vulnerability Multi-Scanner for Web.
[3]. Rathod S.K, Japtap J. R., Satpute A.P., Shikhare K.A, Pujari A.S, & Pandit A. (2022). An Automatic VulnerabilityScanner for Web Applications with Firewall Techniques.
[4]. Prashanth Satya Sai Kiran Gandikota, Sushani S, DeekshithaValluri, Gopi Krishna Yanala. (2023). Web Application through Comprehensive Vulnerability Assessment.
[5]. Sheetal Bairwa, Bhawna Mewara and Jyoti Gajrani. (2014). Vulnerability Scanners: A Proactive Approach To Assess Web Application Security.
[6]. Sandeep Kumar, Renuka Mahajan, Naresh Kumar, Sunil Kumar Khatri. (2017). A study on Web Application Security and Detecting Security.
[7]. Thinazar Aung, Zin Thu ThuMyint. (2020). Effective Web Application Vulnerability Testing System Using ProposedXSS_SQL_Scanning_Algorithm.

[8]. Sanjukta Mohanty, Arup Abhinna Acharya. (2021). Detection of XSS Vulnerabilities Using Security Testing Approaches.
[9]. Divyani Yadav, Deeksha Gupta, Dhananjay Singh, Devendra Kumar, Upansana Sharma. (2018). Vulnerabilities and Security of Web Applications.
[10]. Omar Cheikhrouhou, Moez Krichen, Habib Hamam &AbdelouahidDerhab. (2021). OWASP Ten Driven Survey on Web Application Protection Methods.

.