CHOWKIDAR

Automated Vulnerability Scan Report

CHOWKIDAR

Disclaimer:

This vulnerability report has been generated automatically based on data collected and analyzed by automated vulnerability assessment tools. While every effort has been made to accurately identify and categorize vulnerabilities, there may be limitations and false positives inherent in automated scanning.

This report is intended to provide an overview of potential vulnerabilities detected in the scanned systems or applications. It is not a substitute for manual security assessments or comprehensive penetration testing. Users of this report are advised to conduct thorough security reviews and validation tests to confirm the existence and severity of identified vulnerabilities.

The findings presented in this report are based on the automated tools' assessments at the time of scanning and may not reflect real-time security posture. It is recommended to regularly update and re-scan systems to identify new vulnerabilities and address any remediation efforts promptly.

The findings presented in this report are based on the automated tools' assessments at the time of scanning and may not reflect real-time security posture. It is recommended to regularly update and re-scan systems to identify new vulnerabilities and address any remediation efforts promptly.

Measurement Scales:

CRITICAL: Vulnerability is an otherwise high-severity issue with additional security implications that could lead to exceptional business impact. Findings are marked as critical severity to communicate an exigent need for immediate remediation. Examples include threats to human safety, permanent loss or compromise of business-critical data, and evidence of prior compromise.

HIGH: Vulnerability introduces significant technical risk to the system that is not contingent on other issues being present to exploit. Examples include creating a breach in the confidentiality or integrity of sensitive business data, customer information, or administrative and user accounts.

MEDIUM: Vulnerability does not in isolation lead directly to the exposure of sensitive business data. However, it can be leveraged in conjunction with another issue to expose business risk. Examples include insecurely storing user credentials, transmitting sensitive data unencrypted, and improper network segmentation.

LOW: Vulnerability may result in limited risk or require the presence of multiple additional vulnerabilities to become exploitable. Examples include overly verbose error messages, insecure TLS configurations, and detailed banner information disclosure.

INFO: Finding does not have a direct security impact but represents an opportunity to add an additional layer of security, is a deviation from best practices, or is a security-relevant observation that may lead to exploitable vulnerabilities in the future. Examples include vulnerable yet unused source code and missing HTTP security headers.

CHOWKIDAR

Report Summary:

The Report Summary provides a concise overview of key findings, insights, and conclusions from a larger report or analysis. It typically includes a brief introduction, the main objectives or scope of the report, key data points or results, important trends or patterns identified, and any significant recommendations or actions to be taken based on the findings. The purpose of a Report Summary is to quickly inform readers about the essential aspects of the report without having to go through the entire document.

SCOPE	http://your-web-url.com
AUDIT DATE	06 Sep 2024
VULNERABILITIES DISCOVERED	9

CRITICAL	HIGH	MEDIUM	LOW	INFO
1	1	5	1	1

CHOWKIDAR

Overview Table:

NO	VULNERABILITY NAME	SEVERITY	CVSS
1	Ports Running Services With Known Vulnerabilities	CRITICAL	9.4
2	Vulnerable To Slowloris DDoS Attack	HIGH	7.5
3	Missing Security Headers	MEDIUM	6.5
4	Vulnerable To Diffie-Hellman Key Exchange Attack	MEDIUM	5.9
5	Banner Grabbing	MEDIUM	5.3
6	Directory Listing Enabled	MEDIUM	5.3
7	Frameable response & Clickjacking	MEDIUM	4.3
8	Vulnerable To Poodle SSLv3 Attack	LOW	3.4
9	Endpoints Discovered	INFO	0.0

CHOWKIDAR

Detailed Vulnerability Report

CHOWKIDAR

Ports Running Services With Known Vulnerabilities

SEVERITY	CRITICAL
CVSS SCORE	9.4
CVSS STRING	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:L/A:H
CWE	CWE-923: Improper Restriction of Network Ports/Services

Vulnerability Description:

Ports running services with known vulnerabilities refers to the situation where network ports on a server or device are running software services with publicly disclosed security vulnerabilities. This type of vulnerability poses significant risks, as attackers can exploit known vulnerabilities in the services to gain unauthorized access, execute arbitrary code, or disrupt services.
Attackers can exploit these vulnerabilities in various ways, including:

- **Remote Code Execution:** Attackers may execute arbitrary code on the server running the vulnerable service, potentially compromising the server and the data it hosts.
- **Data Breach:** Services with known vulnerabilities may allow attackers to access and exfiltrate sensitive data stored on the server, leading to data breaches and privacy violations.
- **Denial of Service (DoS):** Attackers may exploit vulnerabilities to disrupt services and cause downtime, impacting the availability of applications and services.
- **Privilege Escalation:** Attackers may escalate their privileges on the server through vulnerabilities in running services, gaining higher-level access to the server and other connected systems.
- **Malware Distribution:** Attackers may use vulnerable services to distribute malware to users who interact with the server, spreading infections and causing further harm.
- **Man-in-the-Middle Attacks:** Vulnerable services may be exploited to intercept and manipulate data in transit, leading to man-in-the-middle attacks and data integrity issues.

Potential Risk Associated:

- **Unauthorized Access:** Attackers can exploit vulnerabilities in services to gain unauthorized access to the server or network, potentially compromising sensitive data and systems.
- **Data Breach:** Known vulnerabilities can be exploited to access, steal, or manipulate sensitive data, such as personally identifiable information (PII), financial data, or intellectual property.
- **Remote Code Execution:** Attackers may execute arbitrary code on the server running the vulnerable service, allowing them to take control of the server and perform malicious actions, such as deploying malware or conducting further attacks.
- **Denial of Service (DoS):** Vulnerable services can be exploited to launch DoS or distributed denial of service (DDoS) attacks, causing service disruptions and making the server or application unavailable to legitimate users.
- **Privilege Escalation:** Attackers can escalate their privileges on the server by exploiting vulnerabilities, gaining access to higher-level functions and other connected systems.
- **Malware Distribution:** Attackers may use vulnerable services as a means to distribute malware, such as ransomware or trojans, to users interacting with the server.
- **Data Integrity Issues:** Attackers may manipulate or alter data passing through the vulnerable services, compromising the integrity and reliability of the data.
- **Compliance Violations:** Failure to address known vulnerabilities in services may result in non-

compliance with regulations such as GDPR, HIPAA, or PCI DSS, leading to potential legal and financial penalties.

Evidence (POC):

The following ports are open and running services that are vulnerable and outdated.

```
    22 ssh 4.7p1 Debian 8ubuntu1
 cpe:/a:openbsd:openssh:4.7p1:
    95499236-C9FE-56A6-9D7D-E943A24B633A    10.0 https://vulners.com/githubexploit/95499236-C9FE-56A6-9D7D-E943A24B633A *EXPLOIT*
    2C119FFA-ECE0-5E14-A4A4-354A2C38071A 10.0 https://vulners.com/githubexploit/2C119FFA-ECE0-5E14-A4A4-354A2C38071A *EXPLOIT*
    CVE-2023-38408  9.8   https://vulners.com/cve/CVE-2023-38408
    CVE-2016-1908    9.8   https://vulners.com/cve/CVE-2016-1908
    B8190CDB-3EB9-5631-9828-8064A1575B23    9.8   https://vulners.com/githubexploit/B8190CDB-3EB9-5631-9828-8064A1575B23 *EXPLOIT*
    8FC9C5AB-3968-5F3C-825E-E8DB5379A623    9.8   https://vulners.com/githubexploit/8FC9C5AB-3968-5F3C-825E-E8DB5379A623 *EXPLOIT*
    8AD01159-548E-546E-AA87-2DE89F3927EC    9.8   https://vulners.com/githubexploit/8AD01159-548E-546E-AA87-2DE89F3927EC *EXPLOIT*
    5E6968B4-DBD6-57FA-BF6E-D9B2219DB27A    9.8   https://vulners.com/githubexploit/5E6968B4-DBD6-57FA-BF6E-D9B2219DB27A *EXPLOIT*
    CVE-2015-5600    8.5   https://vulners.com/cve/CVE-2015-5600
    SSV:78173   7.8   https://vulners.com/seebug/SSV:78173 *EXPLOIT*
    SSV:69983   7.8   https://vulners.com/seebug/SSV:69983 *EXPLOIT*
    PACKETSTORM:98796   7.8   https://vulners.com/packetstorm/PACKETSTORM:98796 *EXPLOIT*
    PACKETSTORM:94556   7.8   https://vulners.com/packetstorm/PACKETSTORM:94556 *EXPLOIT*
    PACKETSTORM:140070 7.8   https://vulners.com/packetstorm/PACKETSTORM:140070   *EXPLOIT*
    PACKETSTORM:101052 7.8   https://vulners.com/packetstorm/PACKETSTORM:101052   *EXPLOIT*
    EXPLOITPACK:71D51B69AA2D3A74753D7A921EE79985   7.8   https://vulners.com/exploitpack/EXPLOITPACK:71D51B69AA2D3A74753D7A921EE79985 *EXPLOIT*
    EXPLOITPACK:67F6569F63A082199721C069C852BBD7   7.8   https://vulners.com/exploitpack/EXPLOITPACK:67F6569F63A082199721C069C852BBD7   *EXPLOIT*
    EXPLOITPACK:5BCA798C6BA71FAE29334297EC0B6A09   7.8   https://vulners.com/exploitpack/EXPLOITPACK:5BCA798C6BA71FAE29334297EC0B6A09   *EXPLOIT*
    EDB-ID:24450   7.8   https://vulners.com/exploitdb/EDB-ID:24450 *EXPLOIT*
    EDB-ID:15215   7.8   https://vulners.com/exploitdb/EDB-ID:15215 *EXPLOIT*
    CVE-2020-15778  7.8   https://vulners.com/cve/CVE-2020-15778
    CVE-2016-10012  7.8   https://vulners.com/cve/CVE-2016-10012
    CVE-2015-8325    7.8   https://vulners.com/cve/CVE-2015-8325
    1337DAY-ID-26494   7.8   https://vulners.com/zdt/1337DAY-ID-26494   *EXPLOIT*
    SSV:92579   7.5   https://vulners.com/seebug/SSV:92579 *EXPLOIT*
    SSV:61450   7.5   https://vulners.com/seebug/SSV:61450 *EXPLOIT*
    PACKETSTORM:173661 7.5   https://vulners.com/packetstorm/PACKETSTORM:173661   *EXPLOIT*
    F0979183-AE88-53B4-86CF-3AF0523F3807 7.5   https://vulners.com/githubexploit/F0979183-AE88-53B4-86CF-3AF0523F3807 *EXPLOIT*
    EDB-ID:40888   7.5   https://vulners.com/exploitdb/EDB-ID:40888 *EXPLOIT*
    CVE-2016-6515    7.5   https://vulners.com/cve/CVE-2016-6515
    CVE-2016-10708  7.5   https://vulners.com/cve/CVE-2016-10708
    CVE-2014-1692    7.5   https://vulners.com/cve/CVE-2014-1692
    CVE-2010-4478    7.5   https://vulners.com/cve/CVE-2010-4478
    1337DAY-ID-26576   7.5   https://vulners.com/zdt/1337DAY-ID-26576   *EXPLOIT*
    CVE-2016-10009  7.3   https://vulners.com/cve/CVE-2016-10009
    SSV:92582   7.2   https://vulners.com/seebug/SSV:92582 *EXPLOIT*
    CVE-2016-10010  7.0   https://vulners.com/cve/CVE-2016-10010
    SSV:92580   6.9   https://vulners.com/seebug/SSV:92580 *EXPLOIT*
    CVE-2015-6564    6.9   https://vulners.com/cve/CVE-2015-6564
    1337DAY-ID-26577   6.9   https://vulners.com/zdt/1337DAY-ID-26577   *EXPLOIT*
    EDB-ID:46516   6.8   https://vulners.com/exploitdb/EDB-ID:46516 *EXPLOIT*
    EDB-ID:46193   6.8   https://vulners.com/exploitdb/EDB-ID:46193 *EXPLOIT*
    CVE-2019-6110    6.8   https://vulners.com/cve/CVE-2019-6110
    CVE-2019-6109    6.8   https://vulners.com/cve/CVE-2019-6109
    C94132FD-1FA5-5342-B6EE-0DAF45EEFFE3 6.8   https://vulners.com/githubexploit/C94132FD-1FA5-5342-B6EE-0DAF45EEFFE3 *EXPLOIT*
    10213DBE-F683-58BB-B6D3-353173626207    6.8   https://vulners.com/githubexploit/10213DBE-F683-58BB-B6D3-353173626207 *EXPLOIT*
    CVE-2023-51385  6.5   https://vulners.com/cve/CVE-2023-51385
    CVE-2008-1657    6.5   https://vulners.com/cve/CVE-2008-1657
    EDB-ID:40858   6.4   https://vulners.com/exploitdb/EDB-ID:40858 *EXPLOIT*
    EDB-ID:40119   6.4   https://vulners.com/exploitdb/EDB-ID:40119 *EXPLOIT*
    EDB-ID:39569   6.4   https://vulners.com/exploitdb/EDB-ID:39569 *EXPLOIT*
    CVE-2016-3115    6.4   https://vulners.com/cve/CVE-2016-3115
    EDB-ID:40136   5.9   https://vulners.com/exploitdb/EDB-ID:40136 *EXPLOIT*
    EDB-ID:40113   5.9   https://vulners.com/exploitdb/EDB-ID:40113 *EXPLOIT*
    CVE-2023-48795  5.9   https://vulners.com/cve/CVE-2023-48795
    CVE-2019-6111    5.9   https://vulners.com/cve/CVE-2019-6111
    CVE-2016-6210    5.9   https://vulners.com/cve/CVE-2016-6210
    SSV:61911   5.8   https://vulners.com/seebug/SSV:61911 *EXPLOIT*
    EXPLOITPACK:98FE96309F9524B8C84C508837551A19   5.8   https://vulners.com/exploitpack/EXPLOITPACK:98FE96309F9524B8C84C508837551A19   *EXPLOIT*
```

*EXPLOITPACK:5330EA02EBDE345BFC9D6DDDD97F9E97 5.8 https://vulners.com/exploitpack/EXPLOITPACK:5330EA02EBDE345BFC9D6DDDD97F9E97 *EXPLOIT**

CVE-2014-2653 5.8 https://vulners.com/cve/CVE-2014-2653

*1337DAY-ID-32328 5.8 https://vulners.com/zdt/1337DAY-ID-32328 *EXPLOIT**

*1337DAY-ID-32009 5.8 https://vulners.com/zdt/1337DAY-ID-32009 *EXPLOIT**

*SSV:91041 5.5 https://vulners.com/seebug/SSV:91041 *EXPLOIT**

*PACKETSTORM:140019 5.5 https://vulners.com/packetstorm/PACKETSTORM:140019 *EXPLOIT**

*PACKETSTORM:136234 5.5 https://vulners.com/packetstorm/PACKETSTORM:136234 *EXPLOIT**

*EXPLOITPACK:F92411A645D85F05BDBD274FD222226F 5.5 https://vulners.com/exploitpack/EXPLOITPACK:F92411A645D85F05BDBD274FD222226F *EXPLOIT**

*EXPLOITPACK:9F2E746846C3C623A27A441281EAD138 5.5 https://vulners.com/exploitpack/EXPLOITPACK:9F2E746846C3C623A27A441281EAD138 *EXPLOIT**

*EXPLOITPACK:1902C998CBF9154396911926B4C3B330 5.5 https://vulners.com/exploitpack/EXPLOITPACK:1902C998CBF9154396911926B4C3B330 *EXPLOIT**

CVE-2016-10011 5.5 https://vulners.com/cve/CVE-2016-10011

*PACKETSTORM:181223 5.3 https://vulners.com/packetstorm/PACKETSTORM:181223 *EXPLOIT**

*MSF:AUXILIARY-SCANNER-SSH-SSH_ENUMUSERS- 5.3 https://vulners.com/metasploit/MSF:AUXILIARY-SCANNER-SSH-SSH_ENUMUSERS- *EXPLOIT**

*EDB-ID:45939 5.3 https://vulners.com/exploitdb/EDB-ID:45939 *EXPLOIT**

*EDB-ID:45233 5.3 https://vulners.com/exploitdb/EDB-ID:45233 *EXPLOIT**

CVE-2018-20685 5.3 https://vulners.com/cve/CVE-2018-20685

CVE-2018-15473 5.3 https://vulners.com/cve/CVE-2018-15473

CVE-2017-15906 5.3 https://vulners.com/cve/CVE-2017-15906

CVE-2016-20012 5.3 https://vulners.com/cve/CVE-2016-20012

*SSV:60656 5.0 https://vulners.com/seebug/SSV:60656 *EXPLOIT**

*SSH_ENUM 5.0 https://vulners.com/canvas/SSH_ENUM *EXPLOIT**

*PACKETSTORM:150621 5.0 https://vulners.com/packetstorm/PACKETSTORM:150621 *EXPLOIT**

*EXPLOITPACK:F957D7E8A0CC1E23C3C649B764E13FB0 5.0 https://vulners.com/exploitpack/EXPLOITPACK:F957D7E8A0CC1E23C3C649B764E13FB0 *EXPLOIT**

*EXPLOITPACK:EBDBC5685E3276D648B4D14B75563283 5.0 https://vulners.com/exploitpack/EXPLOITPACK:EBDBC5685E3276D648B4D14B75563283 *EXPLOIT**

CVE-2010-5107 5.0 https://vulners.com/cve/CVE-2010-5107

*1337DAY-ID-31730 5.0 https://vulners.com/zdt/1337DAY-ID-31730 *EXPLOIT**

CVE-2014-2532 4.9 https://vulners.com/cve/CVE-2014-2532

*EXPLOITPACK:802AF3229492E147A5F09C7F2B27C6DF 4.3 https://vulners.com/exploitpack/EXPLOITPACK:802AF3229492E147A5F09C7F2B27C6DF *EXPLOIT**

*EXPLOITPACK:5652DDAA7FE452E19AC0DC1CD97BA3EF 4.3 https://vulners.com/exploitpack/EXPLOITPACK:5652DDAA7FE452E19AC0DC1CD97BA3EF *EXPLOIT**

CVE-2015-5352 4.3 https://vulners.com/cve/CVE-2015-5352

*1337DAY-ID-25440 4.3 https://vulners.com/zdt/1337DAY-ID-25440 *EXPLOIT**

*1337DAY-ID-25438 4.3 https://vulners.com/zdt/1337DAY-ID-25438 *EXPLOIT**

CVE-2010-4755 4.0 https://vulners.com/cve/CVE-2010-4755

CVE-2021-36368 3.7 https://vulners.com/cve/CVE-2021-36368

CVE-2012-0814 3.5 https://vulners.com/cve/CVE-2012-0814

CVE-2011-5000 3.5 https://vulners.com/cve/CVE-2011-5000

*SSV:92581 2.1 https://vulners.com/seebug/SSV:92581 *EXPLOIT**

CVE-2011-4327 2.1 https://vulners.com/cve/CVE-2011-4327

CVE-2015-6563 1.9 https://vulners.com/cve/CVE-2015-6563

CVE-2008-3259 1.2 https://vulners.com/cve/CVE-2008-3259

*PACKETSTORM:151227 0.0 https://vulners.com/packetstorm/PACKETSTORM:151227 *EXPLOIT**

*PACKETSTORM:140261 0.0 https://vulners.com/packetstorm/PACKETSTORM:140261 *EXPLOIT**

*PACKETSTORM:138006 0.0 https://vulners.com/packetstorm/PACKETSTORM:138006 *EXPLOIT**

*PACKETSTORM:137942 0.0 https://vulners.com/packetstorm/PACKETSTORM:137942 *EXPLOIT**

*1337DAY-ID-30937 0.0 https://vulners.com/zdt/1337DAY-ID-30937 *EXPLOIT**

53 domain 9.4.2

cpe:/a:isc:bind:9.4.2:

*SSV:2853 10.0 https://vulners.com/seebug/SSV:2853 *EXPLOIT**

CVE-2008-0122 10.0 https://vulners.com/cve/CVE-2008-0122

*95499236-C9FE-56A6-9D7D-E943A24B633A 10.0 https://vulners.com/githubexploit/95499236-C9FE-56A6-9D7D-E943A24B633A *EXPLOIT**

*2C119FFA-ECE0-5E14-A4A4-354A2C38071A 10.0 https://vulners.com/githubexploit/2C119FFA-ECE0-5E14-A4A4-354A2C38071A *EXPLOIT**

CVE-2021-25216 9.8 https://vulners.com/cve/CVE-2021-25216

CVE-2020-8616 8.6 https://vulners.com/cve/CVE-2020-8616

CVE-2016-1286 8.6 https://vulners.com/cve/CVE-2016-1286

*SSV:60184 8.5 https://vulners.com/seebug/SSV:60184 *EXPLOIT**

CVE-2012-1667 8.5 https://vulners.com/cve/CVE-2012-1667

*SSV:60292 7.8 https://vulners.com/seebug/SSV:60292 *EXPLOIT**

*PACKETSTORM:180552 7.8 https://vulners.com/packetstorm/PACKETSTORM:180552 *EXPLOIT**

*PACKETSTORM:138960 7.8 https://vulners.com/packetstorm/PACKETSTORM:138960 *EXPLOIT**

*PACKETSTORM:132926 7.8 https://vulners.com/packetstorm/PACKETSTORM:132926 *EXPLOIT**

*MSF:AUXILIARY-DOS-DNS-BIND_TKEY- 7.8 https://vulners.com/metasploit/MSF:AUXILIARY-DOS-DNS-BIND_TKEY- *EXPLOIT**

*EXPLOITPACK:BE4F638B632EA0754155A27ECC4B3D3F 7.8 https://vulners.com/exploitpack/EXPLOITPACK:BE4F638B632EA0754155A27ECC4B3D3F *EXPLOIT**

*EXPLOITPACK:46DEBFAC850194C04C54F93E0DFF5F4F 7.8 https://vulners.com/exploitpack/EXPLOITPACK:46DEBFAC850194C04C54F93E0DFF5F4F *EXPLOIT**

*EXPLOITPACK:09762DB0197BBAAAB6FC79F24F0D2A74 7.8 https://vulners.com/exploitpack/EXPLOITPACK:09762DB0197BBAAAB6FC79F24F0D2A74 *EXPLOIT**

*EDB-ID:42121 7.8 https://vulners.com/exploitdb/EDB-ID:42121 *EXPLOIT**

*EDB-ID:37723 7.8 https://vulners.com/exploitdb/EDB-ID:37723 *EXPLOIT**

*EDB-ID:37721 7.8 https://vulners.com/exploitdb/EDB-ID:37721 *EXPLOIT**

CVE-2017-3141 7.8 https://vulners.com/cve/CVE-2017-3141

CVE-2015-5722 7.8 https://vulners.com/cve/CVE-2015-5722
CVE-2015-5477 7.8 https://vulners.com/cve/CVE-2015-5477
CVE-2014-8500 7.8 https://vulners.com/cve/CVE-2014-8500
CVE-2012-5166 7.8 https://vulners.com/cve/CVE-2012-5166
CVE-2012-4244 7.8 https://vulners.com/cve/CVE-2012-4244
CVE-2012-3817 7.8 https://vulners.com/cve/CVE-2012-3817
CVE-2008-4163 7.8 https://vulners.com/cve/CVE-2008-4163
*1337DAY-ID-25325 7.8 https://vulners.com/zdt/1337DAY-ID-25325 *EXPLOIT**
*1337DAY-ID-23970 7.8 https://vulners.com/zdt/1337DAY-ID-23970 *EXPLOIT**
*1337DAY-ID-23960 7.8 https://vulners.com/zdt/1337DAY-ID-23960 *EXPLOIT**
*1337DAY-ID-23948 7.8 https://vulners.com/zdt/1337DAY-ID-23948 *EXPLOIT**
CVE-2010-0382 7.6 https://vulners.com/cve/CVE-2010-0382
*PACKETSTORM:180551 7.5 https://vulners.com/packetstorm/PACKETSTORM:180551 *EXPLOIT**
*PACKETSTORM:180550 7.5 https://vulners.com/packetstorm/PACKETSTORM:180550 *EXPLOIT**
*MSF:AUXILIARY-DOS-DNS-BIND_TSIG_BADTIME- 7.5 https://vulners.com/metasploit/MSF:AUXILIARY-DOS-DNS-BIND_TSIG_BADTIME- *EXPLOIT**
*MSF:AUXILIARY-DOS-DNS-BIND_TSIG- 7.5 https://vulners.com/metasploit/MSF:AUXILIARY-DOS-DNS-BIND_TSIG- *EXPLOIT**
*EDB-ID:40453 7.5 https://vulners.com/exploitdb/EDB-ID:40453 *EXPLOIT**
CVE-2023-50387 7.5 https://vulners.com/cve/CVE-2023-50387
CVE-2023-3341 7.5 https://vulners.com/cve/CVE-2023-3341
CVE-2021-25215 7.5 https://vulners.com/cve/CVE-2021-25215
CVE-2020-8617 7.5 https://vulners.com/cve/CVE-2020-8617
CVE-2017-3145 7.5 https://vulners.com/cve/CVE-2017-3145
CVE-2017-3143 7.5 https://vulners.com/cve/CVE-2017-3143
CVE-2016-9444 7.5 https://vulners.com/cve/CVE-2016-9444
CVE-2016-9131 7.5 https://vulners.com/cve/CVE-2016-9131
CVE-2016-8864 7.5 https://vulners.com/cve/CVE-2016-8864
CVE-2016-2848 7.5 https://vulners.com/cve/CVE-2016-2848
CVE-2016-2776 7.5 https://vulners.com/cve/CVE-2016-2776
CVE-2009-0265 7.5 https://vulners.com/cve/CVE-2009-0265
*BB688FBF-CEE2-5DD1-8561-8F76501DE2D4 7.5 https://vulners.com/githubexploit/BB688FBF-CEE2-5DD1-8561-8F76501DE2D4 *EXPLOIT**
*5EFDF373-FBD1-5C09-A612-00ADBFE574CF 7.5 https://vulners.com/githubexploit/5EFDF373-FBD1-5C09-A612-00ADBFE574CF *EXPLOIT**
*1337DAY-ID-34485 7.5 https://vulners.com/zdt/1337DAY-ID-34485 *EXPLOIT**
*EXPLOITPACK:D6DDF5E24DE171DAAD71FD95FC1B67F2 7.2 https://vulners.com/exploitpack/EXPLOITPACK:D6DDF5E24DE171DAAD71FD95FC1B67F2 *EXPLOIT**
CVE-2015-8461 7.1 https://vulners.com/cve/CVE-2015-8461
CVE-2015-5986 7.1 https://vulners.com/cve/CVE-2015-5986
CVE-2015-8705 7.0 https://vulners.com/cve/CVE-2015-8705
CVE-2016-1285 6.8 https://vulners.com/cve/CVE-2016-1285
CVE-2009-0025 6.8 https://vulners.com/cve/CVE-2009-0025
CVE-2020-8622 6.5 https://vulners.com/cve/CVE-2020-8622
CVE-2018-5741 6.5 https://vulners.com/cve/CVE-2018-5741
CVE-2016-6170 6.5 https://vulners.com/cve/CVE-2016-6170
CVE-2015-8704 6.5 https://vulners.com/cve/CVE-2015-8704
CVE-2010-3614 6.4 https://vulners.com/cve/CVE-2010-3614
CVE-2016-2775 5.9 https://vulners.com/cve/CVE-2016-2775
*SSV:4636 5.8 https://vulners.com/seebug/SSV:4636 *EXPLOIT**
CVE-2022-2795 5.3 https://vulners.com/cve/CVE-2022-2795
CVE-2021-25219 5.3 https://vulners.com/cve/CVE-2021-25219
CVE-2017-3142 5.3 https://vulners.com/cve/CVE-2017-3142
*SSV:30099 5.0 https://vulners.com/seebug/SSV:30099 *EXPLOIT**
*SSV:20595 5.0 https://vulners.com/seebug/SSV:20595 *EXPLOIT**
*PACKETSTORM:157836 5.0 https://vulners.com/packetstorm/PACKETSTORM:157836 *EXPLOIT**
*FBC03933-7A65-52F3-83F4-4B2253A490B6 5.0 https://vulners.com/githubexploit/FBC03933-7A65-52F3-83F4-4B2253A490B6 *EXPLOIT**
CVE-2015-8000 5.0 https://vulners.com/cve/CVE-2015-8000
CVE-2012-1033 5.0 https://vulners.com/cve/CVE-2012-1033
CVE-2011-4313 5.0 https://vulners.com/cve/CVE-2011-4313
CVE-2011-1910 5.0 https://vulners.com/cve/CVE-2011-1910
*SSV:11919 4.3 https://vulners.com/seebug/SSV:11919 *EXPLOIT**
CVE-2010-3762 4.3 https://vulners.com/cve/CVE-2010-3762
CVE-2010-0097 4.3 https://vulners.com/cve/CVE-2010-0097
CVE-2009-0696 4.3 https://vulners.com/cve/CVE-2009-0696
CVE-2010-0290 4.0 https://vulners.com/cve/CVE-2010-0290
*SSV:14986 2.6 https://vulners.com/seebug/SSV:14986 *EXPLOIT**
CVE-2009-4022 2.6 https://vulners.com/cve/CVE-2009-4022
*PACKETSTORM:142800 0.0 https://vulners.com/packetstorm/PACKETSTORM:142800 *EXPLOIT**
*1337DAY-ID-27896 0.0 https://vulners.com/zdt/1337DAY-ID-27896 *EXPLOIT**

2121 ftp 1.3.1

cpe:/a:proftpd:proftpd:1.3.1:

*SAINT:FD1752E124A72FD3A26EEB9B315E8382 10.0 https://vulners.com/saint/SAINT:FD1752E124A72FD3A26EEB9B315E8382 *EXPLOIT**
*SAINT:950EB68D408A40399926A4CCAD3CC62E 10.0 https://vulners.com/saint/SAINT:950EB68D408A40399926A4CCAD3CC62E *EXPLOIT**
*SAINT:63FB77B9136D48259E4F0D4CDA35E957 10.0 https://vulners.com/saint/SAINT:63FB77B9136D48259E4F0D4CDA35E957 *EXPLOIT**
*SAINT:1B08F4664C428B180EEC9617B41D9A2C 10.0 https://vulners.com/saint/SAINT:1B08F4664C428B180EEC9617B41D9A2C *EXPLOIT**
*PROFTPD_MOD_COPY 10.0 https://vulners.com/canvas/PROFTPD_MOD_COPY *EXPLOIT**
*PACKETSTORM:162777 10.0 https://vulners.com/packetstorm/PACKETSTORM:162777 *EXPLOIT**

*PACKETSTORM:132218 10.0 https://vulners.com/packetstorm/PACKETSTORM:132218 *EXPLOIT**
*PACKETSTORM:131567 10.0 https://vulners.com/packetstorm/PACKETSTORM:131567 *EXPLOIT**
*PACKETSTORM:131555 10.0 https://vulners.com/packetstorm/PACKETSTORM:131555 *EXPLOIT**
*PACKETSTORM:131505 10.0 https://vulners.com/packetstorm/PACKETSTORM:131505 *EXPLOIT**
*MSF:EXPLOIT-UNIX-FTP-PROFTPD_MODCOPY_EXEC- 10.0 https://vulners.com/metasploit/MSF:EXPLOIT-UNIX-FTP-PROFTPD_MODCOPY_EXEC- *EXPLOIT**
*EDB-ID:49908 10.0 https://vulners.com/exploitdb/EDB-ID:49908 *EXPLOIT**
*EDB-ID:37262 10.0 https://vulners.com/exploitdb/EDB-ID:37262 *EXPLOIT**
*95499236-C9FE-56A6-9D7D-E943A24B633A 10.0 https://vulners.com/githubexploit/95499236-C9FE-56A6-9D7D-E943A24B633A *EXPLOIT**
*2C119FFA-ECE0-5E14-A4A4-354A2C38071A 10.0 https://vulners.com/githubexploit/2C119FFA-ECE0-5E14-A4A4-354A2C38071A *EXPLOIT**
*1337DAY-ID-36298 10.0 https://vulners.com/zdt/1337DAY-ID-36298 *EXPLOIT**
*1337DAY-ID-23720 10.0 https://vulners.com/zdt/1337DAY-ID-23720 *EXPLOIT**
*1337DAY-ID-23544 10.0 https://vulners.com/zdt/1337DAY-ID-23544 *EXPLOIT**
CVE-2019-12815 9.8 https://vulners.com/cve/CVE-2019-12815
*SSV:26016 9.0 https://vulners.com/seebug/SSV:26016 *EXPLOIT**
*SSV:24282 9.0 https://vulners.com/seebug/SSV:24282 *EXPLOIT**
CVE-2011-4130 9.0 https://vulners.com/cve/CVE-2011-4130
*SSV:96525 7.5 https://vulners.com/seebug/SSV:96525 *EXPLOIT**
CVE-2023-51713 7.5 https://vulners.com/cve/CVE-2023-51713
CVE-2021-46854 7.5 https://vulners.com/cve/CVE-2021-46854
CVE-2020-9272 7.5 https://vulners.com/cve/CVE-2020-9272
CVE-2019-19272 7.5 https://vulners.com/cve/CVE-2019-19272
CVE-2019-19271 7.5 https://vulners.com/cve/CVE-2019-19271
CVE-2019-19270 7.5 https://vulners.com/cve/CVE-2019-19270
CVE-2019-18217 7.5 https://vulners.com/cve/CVE-2019-18217
CVE-2016-3125 7.5 https://vulners.com/cve/CVE-2016-3125
*739FE495-4675-5A2A-BB93-EEF94AC07632 7.5 https://vulners.com/githubexploit/739FE495-4675-5A2A-BB93-EEF94AC07632 *EXPLOIT**
*SSV:20226 7.1 https://vulners.com/seebug/SSV:20226 *EXPLOIT**
*PACKETSTORM:95517 7.1 https://vulners.com/packetstorm/PACKETSTORM:95517 *EXPLOIT**
CVE-2010-3867 7.1 https://vulners.com/cve/CVE-2010-3867
*SSV:12447 6.8 https://vulners.com/seebug/SSV:12447 *EXPLOIT**
*SSV:11950 6.8 https://vulners.com/seebug/SSV:11950 *EXPLOIT**
*EDB-ID:33128 6.8 https://vulners.com/exploitdb/EDB-ID:33128 *EXPLOIT**
CVE-2010-4652 6.8 https://vulners.com/cve/CVE-2010-4652
CVE-2009-0543 6.8 https://vulners.com/cve/CVE-2009-0543
CVE-2023-48795 5.9 https://vulners.com/cve/CVE-2023-48795
*SSV:12523 5.8 https://vulners.com/seebug/SSV:12523 *EXPLOIT**
CVE-2009-3639 5.8 https://vulners.com/cve/CVE-2009-3639
CVE-2017-7418 5.5 https://vulners.com/cve/CVE-2017-7418
CVE-2011-1137 5.0 https://vulners.com/cve/CVE-2011-1137
CVE-2019-19269 4.9 https://vulners.com/cve/CVE-2019-19269
CVE-2008-7265 4.0 https://vulners.com/cve/CVE-2008-7265
CVE-2012-6095 1.2 https://vulners.com/cve/CVE-2012-6095

3306 mysql 5.0.51a-3ubuntu5

cpe:/a:mysql:mysql:5.0.51a-3ubuntu5:
*SSV:15006 6.8 https://vulners.com/seebug/SSV:15006 *EXPLOIT**
CVE-2009-4028 6.8 https://vulners.com/cve/CVE-2009-4028
*SSV:3280 4.6 https://vulners.com/seebug/SSV:3280 *EXPLOIT**
CVE-2008-2079 4.6 https://vulners.com/cve/CVE-2008-2079
CVE-2010-3682 4.0 https://vulners.com/cve/CVE-2010-3682
CVE-2010-3677 4.0 https://vulners.com/cve/CVE-2010-3677

5432 postgresql 8.3.0 - 8.3.7

cpe:/a:postgresql:postgresql:8.3:
*SSV:60718 10.0 https://vulners.com/seebug/SSV:60718 *EXPLOIT**
CVE-2013-1903 10.0 https://vulners.com/cve/CVE-2013-1903
CVE-2013-1902 10.0 https://vulners.com/cve/CVE-2013-1902
CVE-2019-10211 9.8 https://vulners.com/cve/CVE-2019-10211
CVE-2015-3166 9.8 https://vulners.com/cve/CVE-2015-3166
CVE-2015-0244 9.8 https://vulners.com/cve/CVE-2015-0244
CVE-2018-1115 9.1 https://vulners.com/cve/CVE-2018-1115
CVE-2022-1552 8.8 https://vulners.com/cve/CVE-2022-1552
CVE-2021-32027 8.8 https://vulners.com/cve/CVE-2021-32027
CVE-2020-25695 8.8 https://vulners.com/cve/CVE-2020-25695
CVE-2019-10164 8.8 https://vulners.com/cve/CVE-2019-10164
CVE-2019-10127 8.8 https://vulners.com/cve/CVE-2019-10127
CVE-2015-0243 8.8 https://vulners.com/cve/CVE-2015-0243
CVE-2015-0242 8.8 https://vulners.com/cve/CVE-2015-0242
CVE-2015-0241 8.8 https://vulners.com/cve/CVE-2015-0241
*SSV:30015 8.5 https://vulners.com/seebug/SSV:30015 *EXPLOIT**
*SSV:19652 8.5 https://vulners.com/seebug/SSV:19652 *EXPLOIT**
CVE-2010-1447 8.5 https://vulners.com/cve/CVE-2010-1447
CVE-2010-1169 8.5 https://vulners.com/cve/CVE-2010-1169
CVE-2016-5423 8.3 https://vulners.com/cve/CVE-2016-5423
CVE-2021-23214 8.1 https://vulners.com/cve/CVE-2021-23214
CVE-2020-25694 8.1 https://vulners.com/cve/CVE-2020-25694

CVE-2016-7048 8.1 https://vulners.com/cve/CVE-2016-7048
CVE-2022-2625 8.0 https://vulners.com/cve/CVE-2022-2625
CVE-2019-10128 7.8 https://vulners.com/cve/CVE-2019-10128
*SSV:19754 7.5 https://vulners.com/seebug/SSV:19754 *EXPLOIT**
CVE-2020-25696 7.5 https://vulners.com/cve/CVE-2020-25696
CVE-2017-7484 7.5 https://vulners.com/cve/CVE-2017-7484
CVE-2016-0773 7.5 https://vulners.com/cve/CVE-2016-0773
CVE-2016-0768 7.5 https://vulners.com/cve/CVE-2016-0768
CVE-2015-3167 7.5 https://vulners.com/cve/CVE-2015-3167
*EDB-ID:45184 7.3 https://vulners.com/exploitdb/EDB-ID:45184 *EXPLOIT**
CVE-2020-14350 7.3 https://vulners.com/cve/CVE-2020-14350
CVE-2020-10733 7.3 https://vulners.com/cve/CVE-2020-10733
CVE-2017-14798 7.3 https://vulners.com/cve/CVE-2017-14798
CVE-2023-2454 7.2 https://vulners.com/cve/CVE-2023-2454
CVE-2020-14349 7.1 https://vulners.com/cve/CVE-2020-14349
CVE-2016-5424 7.1 https://vulners.com/cve/CVE-2016-5424
CVE-2019-10210 7.0 https://vulners.com/cve/CVE-2019-10210
*PACKETSTORM:148884 6.9 https://vulners.com/packetstorm/PACKETSTORM:148884 *EXPLOIT**
*EXPLOITPACK:6F8D33BC4F1C65AE0911D23B5E6EB665 6.9 https://vulners.com/exploitpack/EXPLOITPACK:6F8D33BC4F1C65AE0911D23B5E6EB665 *EXPLOIT**
*1337DAY-ID-30875 6.9 https://vulners.com/zdt/1337DAY-ID-30875 *EXPLOIT**
*SSV:30152 6.8 https://vulners.com/seebug/SSV:30152 *EXPLOIT**
CVE-2013-0255 6.8 https://vulners.com/cve/CVE-2013-0255
CVE-2012-0868 6.8 https://vulners.com/cve/CVE-2012-0868
CVE-2009-3231 6.8 https://vulners.com/cve/CVE-2009-3231
*SSV:62083 6.5 https://vulners.com/seebug/SSV:62083 *EXPLOIT**
*SSV:62016 6.5 https://vulners.com/seebug/SSV:62016 *EXPLOIT**
*SSV:61543 6.5 https://vulners.com/seebug/SSV:61543 *EXPLOIT**
*SSV:19018 6.5 https://vulners.com/seebug/SSV:19018 *EXPLOIT**
CVE-2021-3677 6.5 https://vulners.com/cve/CVE-2021-3677
CVE-2021-32029 6.5 https://vulners.com/cve/CVE-2021-32029
CVE-2021-32028 6.5 https://vulners.com/cve/CVE-2021-32028
CVE-2014-0065 6.5 https://vulners.com/cve/CVE-2014-0065
CVE-2014-0064 6.5 https://vulners.com/cve/CVE-2014-0064
CVE-2014-0063 6.5 https://vulners.com/cve/CVE-2014-0063
CVE-2014-0061 6.5 https://vulners.com/cve/CVE-2014-0061
CVE-2012-0866 6.5 https://vulners.com/cve/CVE-2012-0866
CVE-2010-4015 6.5 https://vulners.com/cve/CVE-2010-4015
CVE-2010-0442 6.5 https://vulners.com/cve/CVE-2010-0442
CVE-2015-5288 6.4 https://vulners.com/cve/CVE-2015-5288
CVE-2010-3433 6.0 https://vulners.com/cve/CVE-2010-3433
CVE-2010-1170 6.0 https://vulners.com/cve/CVE-2010-1170
CVE-2021-23222 5.9 https://vulners.com/cve/CVE-2021-23222
*SSV:19669 5.5 https://vulners.com/seebug/SSV:19669 *EXPLOIT**
CVE-2010-1975 5.5 https://vulners.com/cve/CVE-2010-1975
CVE-2023-2455 5.4 https://vulners.com/cve/CVE-2023-2455
*SSV:61546 4.9 https://vulners.com/seebug/SSV:61546 *EXPLOIT**
*SSV:60334 4.9 https://vulners.com/seebug/SSV:60334 *EXPLOIT**
CVE-2014-0062 4.9 https://vulners.com/cve/CVE-2014-0062
CVE-2012-3488 4.9 https://vulners.com/cve/CVE-2012-3488
*SSV:61544 4.6 https://vulners.com/seebug/SSV:61544 *EXPLOIT**
CVE-2014-0067 4.6 https://vulners.com/cve/CVE-2014-0067
CVE-2021-3393 4.3 https://vulners.com/cve/CVE-2021-3393
CVE-2021-20229 4.3 https://vulners.com/cve/CVE-2021-20229
CVE-2015-3165 4.3 https://vulners.com/cve/CVE-2015-3165
CVE-2014-8161 4.3 https://vulners.com/cve/CVE-2014-8161
CVE-2012-2143 4.3 https://vulners.com/cve/CVE-2012-2143
*SSV:61547 4.0 https://vulners.com/seebug/SSV:61547 *EXPLOIT**
*SSV:61545 4.0 https://vulners.com/seebug/SSV:61545 *EXPLOIT**
*SSV:60186 4.0 https://vulners.com/seebug/SSV:60186 *EXPLOIT**
CVE-2014-0066 4.0 https://vulners.com/cve/CVE-2014-0066
CVE-2014-0060 4.0 https://vulners.com/cve/CVE-2014-0060
CVE-2012-2655 4.0 https://vulners.com/cve/CVE-2012-2655
CVE-2009-3229 4.0 https://vulners.com/cve/CVE-2009-3229
CVE-2022-41862 3.7 https://vulners.com/cve/CVE-2022-41862
*SSV:19322 3.5 https://vulners.com/seebug/SSV:19322 *EXPLOIT**
*PACKETSTORM:127092 3.5 https://vulners.com/packetstorm/PACKETSTORM:127092 *EXPLOIT**
CVE-2010-0733 3.5 https://vulners.com/cve/CVE-2010-0733

Suggesting Fixes:

To address the issue of open ports running vulnerable and outdated services, consider the following suggested fixes:

- **Close Unnecessary Ports:** Identify and close any open ports that are not needed for the normal operation of your services. Only keep essential ports open to minimize the attack surface.
- **Update and Patch Services:** Ensure that the services running on open ports are up to date

with the latest versions and security patches. This helps address known vulnerabilities and protect against exploitation.

- **Configure Firewalls:** Use firewalls to control access to open ports. Configure firewall rules to restrict traffic to only trusted sources and required services.
- **Monitor Network Traffic:** Continuously monitor network traffic for signs of suspicious activity on open ports. Implement intrusion detection or prevention systems (IDS/IPS) to identify and block potential threats.
- **Limit Access to Services:** Implement strict access controls on services running on open ports. Use IP whitelisting, VPNs, or other access control mechanisms to restrict who can connect to the services.
- **Implement Security Best Practices:** Secure services running on open ports by following security best practices such as enforcing strong authentication, encryption, and secure configurations.

CHOWKIDAR

Vulnerable To Slowloris DDoS Attack

SEVERITY	HIGH
CVSS SCORE	7.5
CVSS STRING	AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H
CWE	CWE-400: Uncontrolled Resource Consumption

Vulnerability Description:

A vulnerability to Slowloris Denial of Service (DoS) attacks occurs when a server or application does not adequately handle partially open connections or slow HTTP requests. Slowloris is a low-bandwidth, application-layer attack designed to consume server resources by making multiple HTTP requests and keeping them open for as long as possible, while sending data very slowly.

Attackers exploit this vulnerability by establishing numerous slow connections to the target server and maintaining them for an extended period. By doing so, Slowloris gradually exhausts the server's available connections, sockets, or memory, preventing legitimate users from establishing new connections.

Potential Risk Associated:

- **Service Disruption:** The server's capacity to handle new connections is reduced, potentially leading to service interruptions or outages.
- **Resource Consumption:** Slowloris consumes server resources such as CPU, memory, and sockets, impacting the server's performance and availability.
- **Denial of Service:** Legitimate users may experience slow response times or be unable to access the server altogether due to resource exhaustion caused by Slowloris.

Evidence (POC):

Ports and Services Vulnerable To Slowloris DDoS Attack are

```
80 http
VULNERABLE:
Slowloris DOS attack
  State: LIKELY VULNERABLE
  IDs:  CVE:CVE-2007-6750
    Slowloris tries to keep many connections to the target web server open and hold
    them open as long as possible.  It accomplishes this by opening connections to
    the target web server and sending a partial request. By doing so, it starves
    the http server's resources causing Denial Of Service.

  Disclosure date: 2009-09-17
  References:
    https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
    http://ha.ckers.org/slowloris/
```

Suggesting Fixes:

To mitigate the risks associated with Slowloris Denial of Service (DoS) attacks, consider implementing the following suggested fixes:

- **Connection Timeouts:** Configure the server to enforce timeouts for connections that remain idle for too long. This can prevent attackers from keeping connections open indefinitely.
- **Rate Limiting:** Implement rate limiting on incoming connections to restrict the number of simultaneous connections or requests from a single IP address. This can help prevent attackers from overwhelming the server with slow connections.
- **Connection Management:** Limit the number of connections each client can establish and control the maximum number of connections per IP address. This can prevent any single client from monopolizing server resources.
- **Application-Layer Firewalls:** Use an application-layer firewall to monitor and filter incoming HTTP requests. These firewalls can detect and block suspicious or malicious requests that exhibit Slowloris-like behavior.
- **Reverse Proxies and Load Balancers:** Deploy a reverse proxy or load balancer to distribute traffic across multiple servers. This can help absorb and mitigate the impact of Slowloris attacks by spreading the load and filtering out malicious requests.
- **Request Limitations:** Set limits on the number of headers, cookies, or query parameters a single request can contain. This can help protect against overly complex or malformed requests used in Slowloris attacks.
- **Web Server Configuration:** Optimize web server configurations to handle slow clients more efficiently, such as adjusting worker processes, thread pooling, and other performance settings.
- **IP Blocking and Blacklisting:** Implement IP blocking or blacklisting for known malicious IP addresses exhibiting Slowloris attack patterns.

CHOWKIDAR

Missing Security Headers

SEVERITY	MEDIUM
CVSS SCORE	6.5
CVSS STRING	AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N
CWE	CWE-525: Missing HTTP Security Headers

Vulnerability Description:

Missing security headers refers to the absence of HTTP headers that enhance the security of web applications and protect against common attacks. These headers provide instructions to web browsers on how to handle certain aspects of web pages, such as content type, origin policy, caching behavior, and scripts execution. Common security headers that may be missing include:

- **Content Security Policy (CSP):** Helps prevent cross-site scripting (XSS) and data injection attacks by specifying which sources and types of content are allowed to be loaded and executed on the page.
- **HTTP Strict Transport Security (HSTS):** Forces browsers to connect to the server using a secure HTTPS connection and protects against downgrade attacks.
- **X-Content-Type-Options:** Protects against MIME type sniffing, which can lead to the execution of malicious content.
- **X-Frame-Options:** Prevents clickjacking attacks by controlling whether the page can be embedded within an iframe.
- **X-XSS-Protection:** Provides basic XSS filtering capabilities and instructs the browser to block or sanitize the page if an XSS attack is detected.
- **Referrer-Policy:** Controls the information sent in the Referer header, protecting user privacy by limiting data leakage.

Potential Risk Associated:

The impact of having missing or misconfigured security headers can be detrimental to the security and privacy of a web application and its users:

- **Cross-Site Scripting (XSS) Attacks:** Without a properly configured Content Security Policy (CSP) header, the application is more vulnerable to XSS attacks. Attackers can inject malicious scripts into web pages, potentially leading to data theft, session hijacking, or unauthorized actions on behalf of the user.
- **Clickjacking Attacks:** The absence of the X-Frame-Options header allows attackers to embed the web application within an iframe and trick users into clicking on hidden or deceptive content. This can result in unauthorized actions being performed, such as changing user settings or executing transactions.
- **Man-in-the-Middle Attacks:** Missing the HTTP Strict Transport Security (HSTS) header can expose the application to man-in-the-middle attacks, where attackers intercept and manipulate data transmitted between the user and the server. This can lead to data breaches, privacy violations, and tampering with sensitive information.
- **MIME Type Sniffing:** Without the X-Content-Type-Options header set to "nosniff," the browser may attempt to determine the content type of a response based on its content rather than the declared content type. This can lead to unintended content execution or the display of malicious

content, posing security risks.

- **Information Leakage:** The absence of the Referrer-Policy header can result in unintentional exposure of user data, such as the URL of the referring page. This can lead to privacy concerns and potential data leakage.
- **Data Integrity Risks:** Without the X-XSS-Protection header, the application lacks basic protection against XSS attacks. This can impact data integrity and user trust in the application's ability to secure their information.

Evidence (POC):

During the scan, the following security headers were missing from the response headers:

- **X-Frame-Options**
- **X-Content-Type-Options**
- **Content-Security-Policy**
- **Referrer-Policy**
- **Permissions-Policy**
- **Cross-Origin-Embedder-Policy**
- **Cross-Origin-Resource-Policy**
- **Cross-Origin-Opener-Policy**

Suggesting Fixes:

Content Security Policy (CSP)

Configuring Content Security Policy involves adding the Content-Security-Policy HTTP header to a web page and giving it values to control w2at resources the user agent is allowed to load for that page. For example, a page that uploads and displays images could allow images from anywhere, but restrict a form action to a specific endpoint. A properly designed Content Security Policy helps protect a page against a cross-site scripting attack.

```
Content-Security-Policy: default-src 'self'
Content-Security-Policy: default-src 'self' example.com *.example.com
Content-Security-Policy: default-src 'self'; img-src *; media-src example.net; script-src scripts.example.com
```

Strict-Transport-Security (HSTS)

If a website accepts a connection through HTTP and redirects to HTTPS, visitors may initially communicate with the non-encrypted version of the site before being redirected, if, for example, the visitor types http://www.foo.com/ or even just foo.com. This creates an opportunity for a man-in-the-middle attack. The redirect could be exploited to direct visitors to a malicious site instead of the secure version of the original site.
The HTTP Strict Transport Security header informs the browser that it should never load a site using HTTP and should automatically convert all attempts to access the site using HTTP to HTTPS requests instead.

```
Strict-Transport-Security: max-age=<expire-time>
Strict-Transport-Security: max-age=<expire-time>; includeSubDomains
Strict-Transport-Security: max-age=<expire-time>; includeSubDomains; preload
```

X-Content-Type-Options

The X-Content-Type-Options response HTTP header is a marker used by the server to indicate that the MIME types advertised in the Content-Type headers should be followed and not be changed. The header allows you to avoid MIME type sniffing by saying that the MIME types are deliberately configured.

```
X-Content-Type-Options: nosniff
```

X-Frame-Options

The X-Frame-Options HTTP response header can be used to indicate whether a browser should be allowed to render a page in a <frame>, <iframe>, <embed> or <object>. Sites can use this to avoid click-jacking attacks, by ensuring that their content is not embedded into other sites.

X-Frame-Options: DENY

X-Frame-Options: SAMEORIGIN

Referrer-Policy

The Referrer-Policy HTTP header controls how much referrer information (sent with the Referer header) should be included with requests. Aside from the HTTP header, you can set this policy in HTML.

Referrer-Policy: no-referrer

Referrer-Policy: no-referrer-when-downgrade

Referrer-Policy: origin

Referrer-Policy: origin-when-cross-origin

Referrer-Policy: same-origin

Referrer-Policy: strict-origin

Referrer-Policy: strict-origin-when-cross-origin

Referrer-Policy: unsafe-url

Permissions Policy

Permissions Policy provides mechanisms for web developers to explicitly declare what functionality can and cannot be used on a website. You define a set of "policies" that restrict what APIs the site's code can access or modify the browser's default behavior for certain features. This allows you to enforce best practices, even as the codebase evolves — as well as more safely compose third-party content.

Permissions Policy is similar to Content Security Policy but controls features instead of security behavior.

Permissions-Policy: <directive>=<allowlist>

Permissions-Policy: geolocation=()

Permissions-Policy: geolocation=(self "https://a.example.com" "https://b.example.com")

Permissions-Policy: picture-in-picture=(), geolocation=(self https://example.com), camera=;*

Cross-Origin-Embedder-Policy (COEP)

The HTTP Cross-Origin-Embedder-Policy (COEP) response header configures embedding cross-origin resources into the document.

Cross-Origin-Embedder-Policy: unsafe-none | require-corp | credentialless

Cross-Origin Resource Policy (CORP)

Cross-Origin Resource Policy is a policy set by the Cross-Origin-Resource-Policy HTTP header that lets websites and applications opt in to protection against certain requests from other origins (such as those issued with elements like <script> and <img>), to mitigate speculative side-channel attacks, like Spectre, as well as Cross-Site Script Inclusion attacks.

Cross-Origin-Resource-Policy: same-site | same-origin | cross-origin

Cross-Origin-Opener-Policy (COOP)

The HTTP Cross-Origin-Opener-Policy (COOP) response header allows you to ensure a top-level document does not share a browsing context group with cross-origin documents. COOP will process-isolate your document and potential attackers can't access your global object if they were to open it in a popup, preventing a set of cross-origin attacks dubbed XS-Leaks.

Cross-Origin-Opener-Policy: unsafe-none

Cross-Origin-Opener-Policy: same-origin-allow-popups

Cross-Origin-Opener-Policy: same-origin

CHOWKIDAR

Vulnerable To Diffie-Hellman Key Exchange Attack

SEVERITY	MEDIUM
CVSS SCORE	5.9
CVSS STRING	AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:N/A:N
CWE	CWE-324: Use of a Key Exchange Without Entity Authentication

Vulnerability Description:

Vulnerability to a Diffie-Hellman key exchange attack occurs when the Diffie-Hellman key exchange process is implemented using weak or insecure parameters, making it susceptible to cryptographic attacks such as the Logjam attack. Diffie-Hellman key exchange is a method used to establish a shared secret between two parties over an insecure communication channel.

- **Man-in-the-Middle (MitM) Attacks:** An attacker can intercept and manipulate the key exchange process, inserting themselves between the two parties to establish two separate secure channels. This allows the attacker to decrypt, modify, or inject data as it passes through them.
- **Weak Parameters:** The use of weak or commonly used parameters (e.g., small or non-random prime numbers) in the Diffie-Hellman key exchange process can allow attackers to crack the key exchange and compromise the shared secret.
- **Insufficient Key Sizes:** Using insufficiently large key sizes can make the Diffie-Hellman key exchange vulnerable to brute-force or other cryptographic attacks.

Potential Risk Associated:

- **Data Interception:** Attackers can eavesdrop on encrypted communications, intercepting sensitive information such as authentication credentials, personal data, or financial information.
- **Data Manipulation:** Attackers may manipulate the data being transmitted between parties, leading to data corruption or unauthorized changes.
- **Loss of Confidentiality:** By breaking the key exchange process, attackers can gain access to the shared secret and decrypt communications, resulting in the loss of confidentiality.
- **Loss of Integrity:** Attackers can modify data in transit, compromising the integrity of the communication and potentially leading to further attacks.

Evidence (POC):

Ports and Services Vulnerable To Diffie-Hellman Key Exchange Attack are

```
25 smtp
VULNERABLE:
Anonymous Diffie-Hellman Key Exchange MitM Vulnerability
  State: VULNERABLE
    Transport Layer Security (TLS) services that use anonymous
    Diffie-Hellman key exchange only provide protection against passive
    eavesdropping, and are vulnerable to active man-in-the-middle attacks
    which could completely compromise the confidentiality and integrity
    of any data exchanged over the resulting session.
  Check results:
    ANONYMOUS DH GROUP 1
```

Cipher Suite: TLS_DH_anon_WITH_RC4_128_MD5
Modulus Type: Safe prime
Modulus Source: postfix builtin
Modulus Length: 1024
Generator Length: 8
Public Key Length: 1024
References:
https://www.ietf.org/rfc/rfc2246.txt

Transport Layer Security (TLS) Protocol DHE_EXPORT Ciphers Downgrade MitM (Logjam)
State: VULNERABLE
IDs: BID:74733 CVE:CVE-2015-4000
The Transport Layer Security (TLS) protocol contains a flaw that is triggered when handling Diffie-Hellman key exchanges defined with the DHE_EXPORT cipher. This may allow a man-in-the-middle attacker to downgrade the security of a TLS session to 512-bit export-grade cryptography, which is significantly weaker, allowing the attacker to more easily break the encryption and monitor or tamper with the encrypted stream.
Disclosure date: 2015-5-19
Check results:
EXPORT-GRADE DH GROUP 1
Cipher Suite: TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
Modulus Type: Safe prime
Modulus Source: Unknown/Custom-generated
Modulus Length: 512
Generator Length: 8
Public Key Length: 512
References:
https://www.securityfocus.com/bid/74733
https://weakdh.org
https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-4000

Diffie-Hellman Key Exchange Insufficient Group Strength
State: VULNERABLE
Transport Layer Security (TLS) services that use Diffie-Hellman groups of insufficient strength, especially those using one of a few commonly shared groups, may be susceptible to passive eavesdropping attacks.
Check results:
WEAK DH GROUP 1
Cipher Suite: TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
Modulus Type: Safe prime
Modulus Source: postfix builtin
Modulus Length: 1024
Generator Length: 8
Public Key Length: 1024
References:
https://weakdh.org

5432 postgresql
VULNERABLE:
Diffie-Hellman Key Exchange Insufficient Group Strength
State: VULNERABLE
Transport Layer Security (TLS) services that use Diffie-Hellman groups of insufficient strength, especially those using one of a few commonly shared groups, may be susceptible to passive eavesdropping attacks.
Check results:
WEAK DH GROUP 1
Cipher Suite: TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
Modulus Type: Safe prime
Modulus Source: Unknown/Custom-generated
Modulus Length: 1024
Generator Length: 8
Public Key Length: 1024
References:
https://weakdh.org

Suggesting Fixes:

To mitigate these risks, one should use secure and updated Diffie-Hellman key exchange implementations, avoid using commonly used or weak parameters, and opt for sufficiently large key sizes. Additionally, using other cryptographic protocols, such as Elliptic Curve Diffie-Hellman (ECDH), can provide stronger security and protection against such attacks.

CHOWKIDAR

Banner Grabbing

SEVERITY	MEDIUM
CVSS SCORE	5.3
CVSS STRING	AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
CWE	CWE-213: Exposed Information through Server Headers

Vulnerability Description:

When an HTTP response includes a server header that discloses the server software and its version, it provides attackers with valuable information about the underlying infrastructure. This disclosure can help attackers tailor their attacks to target known vulnerabilities associated with the specific server software version. Such information may include the type of server (e.g., Apache, Nginx, or IIS) and its exact version number.

Attackers can use this information for:

- **Targeted Attacks:** Knowing the server software and its version allows attackers to research known vulnerabilities and exploits specific to that version. This can lead to targeted attacks such as buffer overflow, code execution, or denial-of-service attacks.
- **Reconnaissance:** Server header information can provide attackers with insights into the technology stack of the application, enabling them to map out the network infrastructure and identify potential attack vectors.
- **Fingerprinting:** Attackers can use server version information for fingerprinting purposes, gathering details about the server environment to inform their attack strategies.

Potential Risk Associated:

The impact of banner grabbing can be significant, especially when attackers obtain sensitive information about a server's configuration and software versions:

- **Targeted Attacks:** Banner grabbing provides attackers with details about the server's software, including type and version. Attackers can use this information to tailor their attacks to exploit known vulnerabilities specific to the server software version.
- **Reconnaissance and Mapping:** Attackers can gather information about the server and its technology stack, aiding in reconnaissance and mapping the network infrastructure. This information can be used to identify potential attack vectors and weaknesses in the network.
- **Exploitation of Known Vulnerabilities:** By knowing the server's software and version, attackers can look up known vulnerabilities in vulnerability databases and attempt to exploit them. This can lead to unauthorized access, data breaches, and other forms of exploitation.
- **Increased Risk of Other Attacks:** Knowing the server's software and version can provide attackers with insights into potential weaknesses that could be exploited through other attack methods, such as cross-site scripting (XSS), cross-site request forgery (CSRF), or SQL injection.
- **Information Leakage:** Banner grabbing may reveal other sensitive information, such as the server's operating system, configuration settings, or internal network details. This information can aid attackers in crafting more precise and effective attacks.

Evidence (POC):

The headers revealing the server version was discovered

- **X-Powered-By: PHP/5.2.4-2ubuntu5.10**
- **Server: Apache/2.2.8 (Ubuntu) DAV/2**

Suggesting Fixes:

To address the issue of a header revealing the server version, consider the following suggested fixes:

- **Remove or Obfuscate the Server Header:** Configure the server to remove or modify the server header in HTTP responses. This can help prevent attackers from easily identifying the server software and its version.
- **Server Configuration:** Most web servers allow you to customize or disable the server header. Refer to your server's documentation (e.g., Apache, Nginx, IIS) to configure the server appropriately.
- **Implement a Web Application Firewall (WAF):** A WAF can help filter and block suspicious requests and responses, potentially providing an additional layer of protection and masking certain server information.

CHOWKIDAR

Directory Listing Enabled

SEVERITY	MEDIUM
CVSS SCORE	5.3
CVSS STRING	AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
CWE	CWE-548: Exposure of Information Through Directory Listing

Vulnerability Description:

Web servers can be configured to automatically list the contents of directories that do not have an index page present. This can aid an attacker by enabling them to quickly identify the resources at a given path, and proceed directly to analyzing and attacking those resources. It particularly increases the exposure of sensitive files within the directory that are not intended to be accessible to users, such as temporary files and crash dumps.

Directory listings themselves do not necessarily constitute a security vulnerability. Any sensitive resources within the web root should in any case be properly access-controlled, and should not be accessible by an unauthorized party who happens to know or guess the URL. Even when directory listings are disabled, an attacker may guess the location of sensitive files using automated tools.

Potential Risk Associated:

- **Sensitive Files:** Directory listings can expose files that contain sensitive information, such as configuration files, backup files, or private documents. These files might include credentials, internal communications, or other confidential data.
- **Discovery of Hidden Resources:** Attackers can discover hidden resources that are not linked or indexed, such as admin panels, test files, or deprecated scripts.
- **Identification of Vulnerable Files:** Attackers can identify and exploit files with known vulnerabilities (e.g., outdated scripts with security flaws).
- **Website Structure Insight:** Attackers can gain a comprehensive understanding of the website's directory structure, making it easier to identify important files and directories.

Evidence (POC):

Discovering endpoints can occur through techniques such as:

- **Web Crawling:** Scanning the application's publicly accessible URLs and directories to discover available endpoints.
- **Brute Forcing:** Systematically trying different URL patterns or parameter combinations to uncover hidden endpoints.
- **Reverse Engineering:** Analyzing client-side code, such as JavaScript files, or network traffic to identify and map application endpoints.
  - **http://your-web-url.com/classes/**
  - **http://your-web-url.com/documentation/**
  - **http://your-web-url.com/images/**
  - **http://your-web-url.com/includes/**
  - **http://your-web-url.com/passwords/**

Suggesting Fixes:

There is not usually any good reason to provide directory listings, and disabling them may place additional hurdles in the path of an attacker. This can normally be achieved in two ways:

- Configure your web server to prevent directory listings for all paths beneath the web root;
- Place into each directory a default file (such as index.htm) that the web server will display instead of returning a directory listing.
- Disable directory listing in the web server configuration (e.g., using .htaccess for Apache or nginx.conf for NGINX).
- Implement proper access controls to restrict unauthorized access to directories.

For **Apache** web server, disabling directory listing can be achieved by adding the following line to the *.htaccess* file or the server configuration:

```
Options -Indexes
```

For **NGINX**, the following directive can be added to the server block:

```
autoindex off;
```

CHOWKIDAR

Frameable response & Clickjacking

SEVERITY	MEDIUM
CVSS SCORE	4.3
CVSS STRING	AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:L/A:N
CWE	CWE-1021: Improper Restriction of Rendered UI Layers or Frames

Vulnerability Description:

Clickjacking is an interface-based attack in which a user is tricked into clicking on actionable content on a hidden website by clicking on some other content in a decoy website. Consider the following example: A web user accesses a decoy website (perhaps this is a link provided by an email) and clicks on a button to win a prize. Unknowingly, they have been deceived by an attacker into pressing an alternative hidden button and this results in the payment of an account on another site. This is an example of a clickjacking attack. The technique depends upon the incorporation of an invisible, actionable web page (or multiple pages) containing a button or hidden link, say, within an iframe. The iframe is overlaid on top of the user's anticipated decoy web page content. This attack differs from a CSRF attack in that the user is required to perform an action such as a button click whereas a CSRF attack depends upon forging an entire request without the user's knowledge or input.

Potential Risk Associated:

- **Unauthorized Actions:** Users can be tricked into performing unintended actions such as changing settings, transferring funds, or deleting data.
- **Account Compromise:** Clickjacking can be used to steal sensitive information, such as login credentials or personal data, leading to account compromise.
- **Sensitive Data Exposure:** Attackers can exploit clickjacking to gain access to sensitive information or functionality within an application.

Evidence (POC):

Save the following code after replacing the URL with the scope url as *filename.html*

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Clickjacking</title>
</head>
<body>

<iframe src="https://affected_URL" width="1000px;" height="700px;"></iframe>

</body>
</html>
```

Suggesting Fixes:

X-Frame-Options

The X-Frame-Options HTTP response header can be used to indicate whether a browser should be allowed to render a page in a <frame>, <iframe>, <embed> or <object>. Sites can use this to avoid click-jacking attacks, by ensuring that their content is not embedded into other sites.

```
X-Frame-Options: DENY
X-Frame-Options: SAMEORIGIN
```

CHOWKIDAR

Vulnerable To Poodle SSLv3 Attack

SEVERITY	LOW
CVSS SCORE	3.4
CVSS STRING	AV:N/AC:H/PR:N/UI:R/S:C/C:L/I:N/A:N
CWE	CWE-319: Cleartext Transmission of Sensitive Information

Vulnerability Description:

The vulnerability to the POODLE (Padding Oracle On Downgraded Legacy Encryption) SSLv3 attack occurs when a server or client supports the outdated SSLv3 encryption protocol. The POODLE attack exploits a design flaw in the SSLv3 protocol's padding mechanism, allowing attackers to decrypt encrypted communications and access sensitive information.

In the POODLE attack, an attacker intercepts encrypted data between a client and server and manipulates the data to create errors in the decryption process. By observing how the server responds to these errors, the attacker can use a padding oracle attack to gradually recover the plaintext data from the encrypted messages.

Key points about the POODLE SSLv3 vulnerability:

- **Legacy Protocol:** SSLv3 is an outdated and insecure protocol that should no longer be used due to its susceptibility to attacks like POODLE.
- **Decryption of Encrypted Data:** Attackers can exploit the POODLE vulnerability to decrypt encrypted communications and gain access to sensitive information such as usernames, passwords, session cookies, and other confidential data.
- **Man-in-the-Middle Attack:** The POODLE attack is a type of man-in-the-middle attack that involves intercepting and manipulating data between a client and server.
- **Downgrade Attack:** Attackers can force a downgrade of the encryption protocol from a more secure version (e.g., TLS 1.2) to SSLv3 to exploit the vulnerability.

Potential Risk Associated:

The impact of vulnerability to the POODLE (Padding Oracle On Downgraded Legacy Encryption) SSLv3 attack can be significant:

- **Decryption of Encrypted Data:** Attackers can exploit the POODLE vulnerability to decrypt encrypted communications, exposing sensitive data such as usernames, passwords, session cookies, financial information, or other confidential data.
- **Man-in-the-Middle Attacks:** The POODLE attack involves intercepting and manipulating data between a client and server, allowing attackers to perform man-in-the-middle attacks. This can lead to data tampering, interception, or redirection of traffic.
- **Session Hijacking:** By decrypting session cookies and other session-related information, attackers can hijack user sessions, gaining unauthorized access to accounts and performing actions on behalf of the user.
- **Loss of Confidentiality:** Successful exploitation of the POODLE attack compromises the confidentiality of encrypted communications, leading to data breaches and exposure of sensitive information.
- **Compliance Violations:** Using the outdated SSLv3 protocol may violate data protection regulations or industry standards that require the use of up-to-date encryption protocols. This

can result in legal consequences, fines, or penalties for non-compliance.

Evidence (POC):

Ports and Services Vulnerable To Poodle SSLv3 Attack are

```
    25 smtp
VULNERABLE:
SSL POODLE information leak
  State: VULNERABLE
  IDs:  CVE:CVE-2014-3566  BID:70574
        The SSL protocol 3.0, as used in OpenSSL through 1.0.1i and other
        products, uses nondeterministic CBC padding, which makes it easier
        for man-in-the-middle attackers to obtain cleartext data via a
        padding-oracle attack, aka the "POODLE" issue.
  Disclosure date: 2014-10-14
  Check results:
    TLS_RSA_WITH_AES_128_CBC_SHA
  References:
    https://www.imperialviolet.org/2014/10/14/poodle.html
    https://www.securityfocus.com/bid/70574
    https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-3566
    https://www.openssl.org/~bodo/ssl-poodle.pdf


    5432 postgresql
VULNERABLE:
SSL POODLE information leak
  State: VULNERABLE
  IDs:  CVE:CVE-2014-3566  BID:70574
        The SSL protocol 3.0, as used in OpenSSL through 1.0.1i and other
        products, uses nondeterministic CBC padding, which makes it easier
        for man-in-the-middle attackers to obtain cleartext data via a
        padding-oracle attack, aka the "POODLE" issue.
  Disclosure date: 2014-10-14
  Check results:
    TLS_RSA_WITH_AES_128_CBC_SHA
  References:
    https://www.imperialviolet.org/2014/10/14/poodle.html
    https://www.securityfocus.com/bid/70574
    https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-3566
    https://www.openssl.org/~bodo/ssl-poodle.pdf
```

Suggesting Fixes:

To mitigate the risks associated with vulnerability to the POODLE (Padding Oracle On Downgraded Legacy Encryption) SSLv3 attack, consider the following suggested fixes:

- **Disable SSLv3 Support:** Remove support for the SSLv3 encryption protocol in servers and clients. This ensures that connections use more secure and modern encryption protocols such as TLS 1.2 or later.
- **Use Secure Protocols:** Enable and prioritize the use of modern and secure encryption protocols, such as TLS 1.2 or TLS 1.3, which are not susceptible to the POODLE attack.
- **Update Software and Libraries:** Ensure that server software, libraries, and clients are up to date with the latest security patches. This helps protect against known vulnerabilities and ensures the use of secure protocols.
- **Configure Cipher Suites:** Configure secure cipher suites for SSL/TLS connections. Use strong encryption algorithms and avoid weak or deprecated cipher suites.

CHOWKIDAR

Endpoints Discovered

SEVERITY	INFO
CVSS SCORE	0.0
CVSS STRING	AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:N
CWE	CWE-200: Exposure of Sensitive Information to an Unauthorized Actor

Vulnerability Description:

Endpoints discovered refer to the process where attackers identify various API or web application endpoints that are publicly accessible or exposed. These endpoints can reveal valuable information about the application's structure, functionality, and available services. Attackers can leverage this information to gain insights into how the application works, what data it handles, and potential vulnerabilities in the exposed endpoints. Once endpoints are discovered, attackers can use this information to probe for vulnerabilities, exploit security weaknesses, or manipulate the application's data and behavior.

Potential Risk Associated:

The impact of discovering endpoints in an application can vary depending on the level of exposure and the sensitivity of the endpoints:

- **Unauthorized Access:** Attackers who discover endpoints may attempt to access them directly, potentially bypassing authentication or authorization mechanisms. This can lead to unauthorized access to sensitive data, functionalities, or resources.
- **Reconnaissance:** Knowledge of endpoints can provide attackers with valuable information about the structure and functionality of an application. This can aid in planning and executing targeted attacks.
- **Exploitation of Vulnerabilities:** If the endpoints have vulnerabilities, attackers may exploit them to gain access to the application or its data. This can lead to data breaches, account takeovers, or other forms of exploitation.
- **Manipulation of Data:** Attackers may use discovered endpoints to manipulate data within the application. This can include altering records, injecting malicious data, or disrupting normal operations.

Evidence (POC):

Discovering endpoints can occur through techniques such as:

- **Web Crawling:** Scanning the application's publicly accessible URLs and directories to discover available endpoints.
- **Brute Forcing:** Systematically trying different URL patterns or parameter combinations to uncover hidden endpoints.
- **Reverse Engineering:** Analyzing client-side code, such as JavaScript files, or network traffic to identify and map application endpoints.
    - **http://your-web-url.com/.buildpath**
    - **http://your-web-url.com/.project**
    - **http://your-web-url.com/classes/**

- **http://your-web-url.com/config.inc**
- **http://your-web-url.com/documentation/**
- **http://your-web-url.com/favicon.ico**
- **http://your-web-url.com/footer.php**
- **http://your-web-url.com/footer**
- **http://your-web-url.com/header**
- **http://your-web-url.com/header.php**
- **http://your-web-url.com/home.php**
- **http://your-web-url.com/home**
- **http://your-web-url.com/images/**
- **http://your-web-url.com/inc**
- **http://your-web-url.com/includes/**
- **http://your-web-url.com/installation**
- **http://your-web-url.com/installation.php**
- **http://your-web-url.com/installation/**
- **http://your-web-url.com/login/cpanel.svg**
- **http://your-web-url.com/login/administrator/**
- **http://your-web-url.com/login/**
- **http://your-web-url.com/login/cpanel.gz**
- **http://your-web-url.com/login**
- **http://your-web-url.com/login.php**
- **http://your-web-url.com/login/cpanel.zip**
- **http://your-web-url.com/login/cpanel.js**
- **http://your-web-url.com/login/cpanel.aspx**
- **http://your-web-url.com/login/cpanel.html**
- **http://your-web-url.com/login/cpanel.css**
- **http://your-web-url.com/login/cpanel.json**
- **http://your-web-url.com/login/super**
- **http://your-web-url.com/login/login**
- **http://your-web-url.com/login/cpanel/**
- **http://your-web-url.com/login/cpanel.pdf**
- **http://your-web-url.com/login/admin/**
- **http://your-web-url.com/login/cpanel.php**
- **http://your-web-url.com/login/index**
- **http://your-web-url.com/login/cpanel.java**
- **http://your-web-url.com/login/oauth/**
- **http://your-web-url.com/login/admin/admin.asp**
- **http://your-web-url.com/passwords/**
- **http://your-web-url.com/phpMyAdmin**
- **http://your-web-url.com/phpinfo.php**
- **http://your-web-url.com/phpinfo**
- **http://your-web-url.com/phpMyAdmin/index.php**
- **http://your-web-url.com/phpMyAdmin/**
- **http://your-web-url.com/phpMyAdmin/phpMyAdmin/index.php**
- **http://your-web-url.com/phpMyAdmin/scripts/setup.php**
- **http://your-web-url.com/phpMyAdmin.php**
- **http://your-web-url.com/register.php**
- **http://your-web-url.com/register**
- **http://your-web-url.com/robots.txt**

Suggesting Fixes:

To address the risks associated with the discovery of endpoints, organizations should implement the following suggested fixes:

- **Proper Authentication and Authorization:** Secure endpoints by requiring authentication and authorization checks for all requests. Use role-based access control (RBAC) to limit access to endpoints based on user roles and permissions.
- **Rate Limiting and Throttling:** Implement rate limiting and request throttling to prevent attackers from exploiting endpoints through brute-force attacks or denial-of-service (DoS) attacks.
- **Input Validation and Output Encoding:** Validate all input data to ensure it meets expected criteria and avoid injection attacks. Encode output data to protect against cross-site scripting (XSS) and other injection attacks.
- **Secure API Design:** Follow best practices for secure API design, such as using HTTPS for all communication, avoiding unnecessary data exposure, and using secure tokens for

authentication.

- **Minimize Endpoint Exposure:** Limit the number of publicly accessible endpoints to only those necessary for the application's functionality. Hide internal or administrative endpoints behind secure firewalls or VPNs.