

LAPORAN PENGUJIAN ENCRYPTION DATA IN TRANSIT

1.1 metode komunikasi mysql menggunakan SSL

1. Masuk ke dalam salah satu vm yng akan digunakan sebagai database server
2. Install mysql server di dalam VM menggunakan “sudo apt install mysql-server”
3. Lalu jalankan service dari mysql tersebut dengan mengetikkan “systemctl start mysql-server”
4. Lalu cek apakah service sudah berjalan menggunakan “sudo systemctl status mysql”

```
roni15@roni15:~$ sudo apt install mysql-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
mysql-server is already the newest version (8.0.41-0ubuntu0.24.04.1).
0 upgraded, 0 newly installed, 0 to remove and 125 not upgraded.
roni15@roni15:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Sat 2025-02-08 02:26:41 UTC; 1min 43s ago
   TriggeredBy: ● ssh.socket
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 2569 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 2571 (sshd)
     Tasks: 1 (limit: 4556)
    Memory: 3.1M (peak: 4.3M)
       CPU: 204ms
    CGroup: /system.slice/ssh.service
            └─2571 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Feb 08 02:26:41 roni15 systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Feb 08 02:26:41 roni15 sshd[2571]: Server listening on :: port 22.
Feb 08 02:26:41 roni15 systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
Feb 08 02:27:33 roni15 sshd[2579]: Accepted password for roni15 from 192.168.1.8 port 49917 ssh2
Feb 08 02:27:33 roni15 sshd[2579]: pam_unix(sshd:session): session opened for user roni15(uid=1000) by roni15(uid=0)
roni15@roni15:~$
```

5. Selanjutnya, kita akan menguba konf pada mysql dengan mengetikkan berikut

```
roni15@roni15:~$ sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

6. Selanjutnya, ketika sudah di dalam file tersebut, ubah pada bagian bind-address = 127.0.0.1, kita ubah ip dengan ip milik kita lalu simpan

```
[mysqld]
#
# * Basic Settings
#
user                = mysql
# pid-file           = /var/run/mysqld/mysqld.pid
# socket             = /var/run/mysqld/mysqld.sock
# port               = 3306
# datadir            = /var/lib/mysql

# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
# tmpdir             = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address         = 192.168.1.12
mysqlx-bind-address  = 127.0.0.1
#
```

7. Setelah disimpan, restart service mysql dengna “systemctl restart mysql”

```
roni15@roni15:~$ sudo systemctl restart mysql
roni15@roni15:~$ |
```

8. Lalu masuk ke dalam menggunakan “sudo mysql”

```
roni15@roni15:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.41-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> |
```

9. Jika sudah selanjutnya kita akan membuat user baru menggunakan perintah berikut

```
mysql> create user 'roni'@'192.168.1.12' identified by 'roni1505';
Query OK, 0 rows affected (0.04 sec)

mysql> |
```

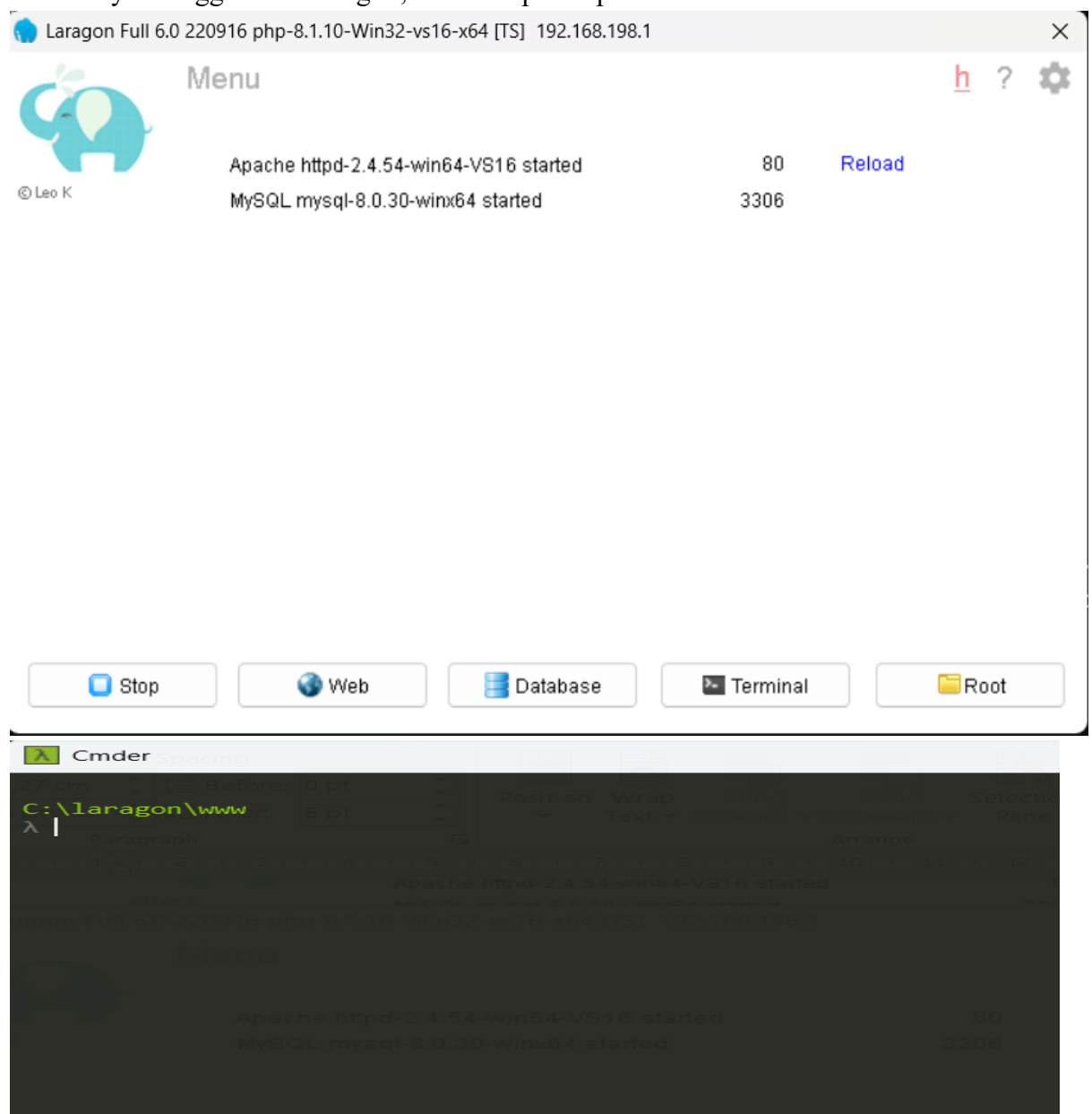
10. Selanjutnya berikan akses pada user dengan perintah berikut

```
mysql> grant all privileges on *.* to 'roni'@'192.168.1.12' with grant option;  
Query OK, 0 rows affected (0.02 sec)  
  
mysql> |
```

Penjelasan:

- GRANT ALL PRIVILEGES: Memberikan semua hak akses.
- ON *.* Akses untuk semua database dan tabel.
- TO widi@'192.168.1.5': Hak akses untuk pengguna 'widi' dari IP 192.168.1.5
- WITH GRANT OPTION: Memberikan hak untuk memberi akses kepada pengguna lain.

11. Disini saya menggunakan laragon, klik start pada aplikasi lalu klik terminal



12. Setelah berada di dalam terminal tersebut, kemudian masuk ke user yang tadi dibuat dengan menggunakan perintah berikut

```
C:\laragon\www
λ mysql -u roni -h 192.168.1.12 -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.41-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

13. Ketika sudah masuk ketik perintah \s untuk melihat apakah masuk menggunakan SSL

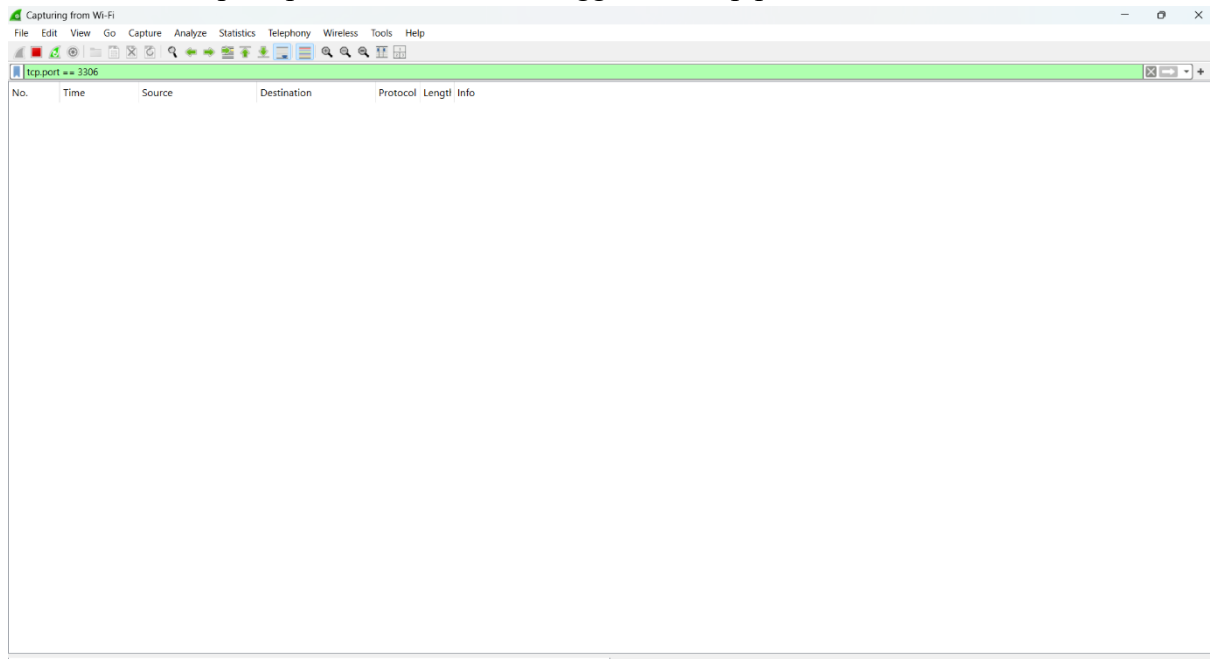
```
mysql> \s
-----
mysql Ver 8.0.30 for Win64 on x86_64 (MySQL Community Server - GPL)
Connection id:      8
Current database:
Current user:       roni@192.168.1.8
SSL:                Cipher in use is TLS_AES_256_GCM_SHA384
Using delimiter:    ;
Server version:     8.0.41-0ubuntu0.24.04.1 (Ubuntu)
Protocol version:   10
Connection:         192.168.1.12 via TCP/IP
Server characterset: utf8mb4
Db characterset:    utf8mb4
Client characterset: cp850
Conn. characterset: cp850
TCP port:           3306
Binary data as:     Hexadecimal
Uptime:             1 min 46 sec

Threads: 2  Questions: 5  Slow queries: 0  Opens: 119  Flush tables: 3  Open tables: 38  Queries per second avg: 0.047
-----
mysql>
```

Seperti yang bisa kita lihat, kita menggunakan SSL yaitu TLS_AES_256_GCM_SHA384

1.2 Pengujian penyadapan dengan menggunakan SSL

1. Buka wireshark pada pc, kemudia filter menggunakan tcp.port == 3306



Penjelasan:

- tcp: protocol yang berjalan, karena mysql berjalan di protocol tcp, maka kita buat tcp
 - port: filter berdasarkan port tertentu
 - 3306 : karena mysql berjalan di port 3306, maka kita filter berdasarkan port 3306
2. Selanjutnya kita mencoba login ulang

```
C:\laragon\www
λ mysql -u roni -h 192.168.1.12 -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.41-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

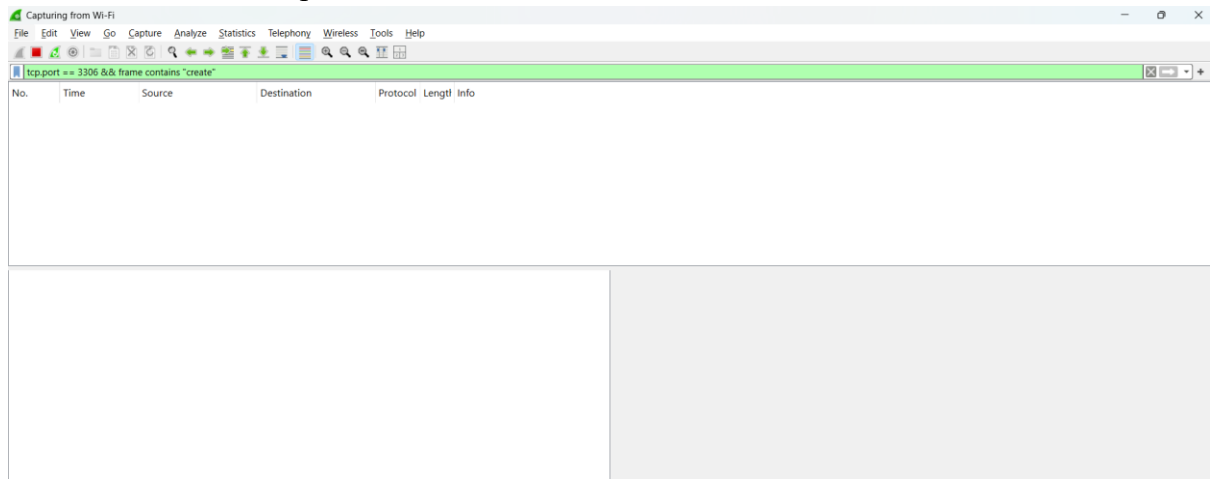
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

```
mysql> create database menggunakan_SSL;
Query OK, 1 row affected (0.01 sec)

mysql>
```

5. Setelah itu tambahkan perintah && frame contains create



Penjelasan:

Jika tidak menggunakan SSL pada saat login ke dalam database, maka, akan terlihat, query yang kita jalankan di dalam database tersebut ke dalam sebuah packet. Terlihat diatas bahwa, query create database sql tersebut tidak tampil di situ, yang artinya bahwa request query dari client dan server mysql tersebut telah terenkripsi dengan aman

- **frame:** Filter ini digunakan untuk mencari informasi di tingkat frame (bingkai) pada paket data. Setiap paket yang ditangkap oleh Wireshark disusun dalam format bingkai, dan filter ini membantu menganalisis metadata atau isi bingkai, seperti waktu, panjang paket, atau konten mentah.
- **contains:** Filter ini digunakan untuk mencari data yang mengandung string tertentu dalam payload paket. Ketika digunakan bersama frame, Wireshark akan mencari string tertentu (misalnya "create") di seluruh isi paket, termasuk bagian payload yang mengandung data mentah.
- **"create"** : Ini adalah string yang dicari dalam payload paket. Dalam konteks ini, Anda mencoba mendeteksi paket yang mengandung kata "create," yang kemungkinan besar menunjukkan perintah SQL untuk membuat database, tabel, atau entitas lainnya di server database.

2.1 Metode Komunikasi mysql tanp menggunakan SSL”

1. Masuk ke dalam terminal laragon lalu ketikan perintah berikut

```
C:\laragon\www
λ mysql -u roni -h 192.168.1.12 -p --ssl-mode=disable
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.41-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Penjelasan:

Opsi `--ssl-mode=disable` pada MySQL digunakan untuk menonaktifkan koneksi SSL/TLS antara klien dan server, sehingga data dikirim tanpa enkripsi. Hal ini berguna dalam lingkungan lokal atau pengujian saat SSL tidak diperlukan atau terjadi masalah dengan konfigurasi SSL, seperti ketidakcocokan sertifikat atau autentikasi plugin seperti `caching_sha2_password`.

2. Ketika sudah masuk, ketikkan perintah “\s” melihat apakah pada saat masuk ke dalam mysql tersebut, mennggunakan ssl atau tidak

```
mysql> \s
-----
mysql Ver 8.0.30 for Win64 on x86_64 (MySQL Community Server - GPL)

Connection id:          11
Current database:
Current user:           roni@192.168.1.8
SSL:                    Not in use
Using delimiter:        ;
Server version:         8.0.41-0ubuntu0.24.04.1 (Ubuntu)
Protocol version:       10
Connection:             192.168.1.12 via TCP/IP
Server characterset:    utf8mb4
Db      characterset:    utf8mb4
Client characterset:    cp850
Conn.  characterset:    cp850
TCP port:               3306
Binary data as:         Hexadecimal
Uptime:                 13 min 10 sec

Threads: 2  Questions: 15  Slow queries: 0  Opens: 120  Flush tables: 3  Open tables: 39  Queries per se
cond avg: 0.018
-----

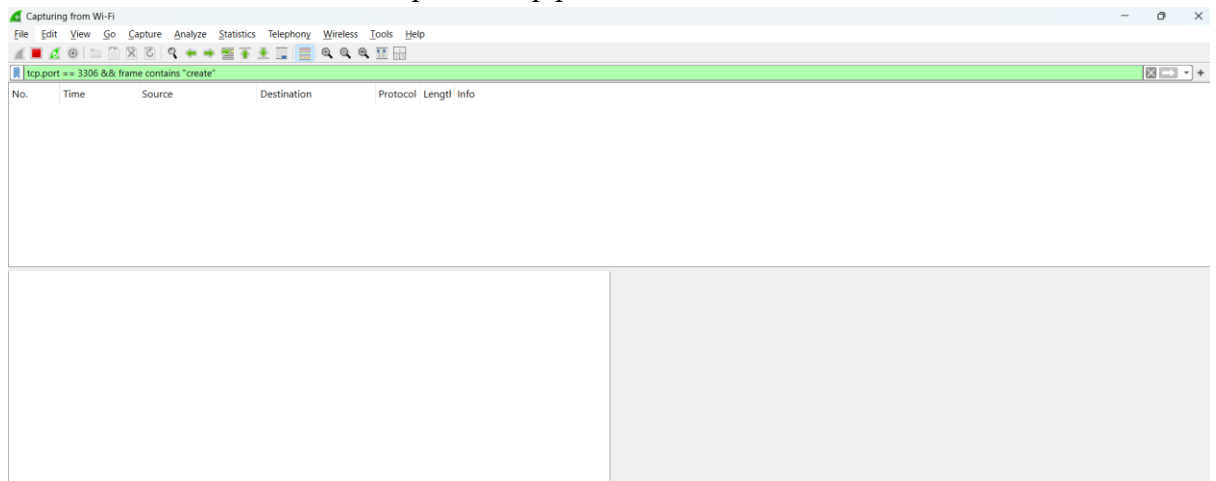
mysql>
```

Terlihat di atas, bahwasanya informasi yang di berikan tidak menampilkan bahwa pada saat kita login tadi kita menggunakan SSL, pada cara sebelum nya, kita tidak menggunakan perintah “`--ssl-mode=disabe`”, maka akan terlihat informasi bahwa

komunikasi dengan server telah di enkripsi dengan SSL. Maka dari itu, jika kita menggunakan “—ssl-mode=disable”, maka akan secara otomatis komunikasi dengan server tidak akan terenkripsi dengan baik.

2.2 Pengujian penyadapan dengan skenario tanpa menggunakan SSL:

1. Buka wireshark lalu masukkan perintah `tcp.port == 3306`



Penjelasan:

- tcp: protocol yang berjalan, karena mysql berjalan di protocol tcp, maka kita buat tcp
 - port: filter berdasarkan port tertentu
 - 3306 : karena mysql berjalan di port 3306, maka kita filter berdasarkan port 3306
2. Setelah aplikasi wireshark terbuka, dan melakukan mode listening packet, maka selanjutnya, kita mencoba login menggunakan user yang tadi tetapi dengan `sslmode=disable`,

```
C:\laragon\www
λ mysql -u roni -h 192.168.1.12 -p --ssl-mode=disable
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.41-0ubuntu0.24.04.1 (Ubuntu)

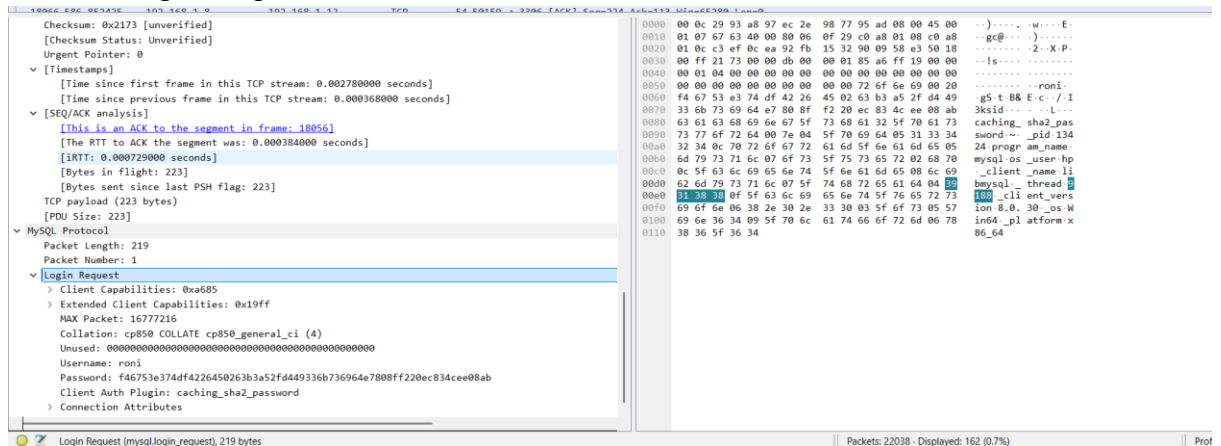
Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

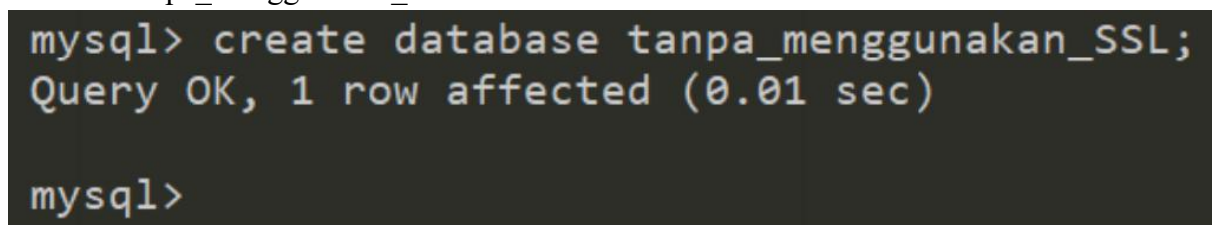
mysql>
```

3. Kita cari bagian request user di wireshark

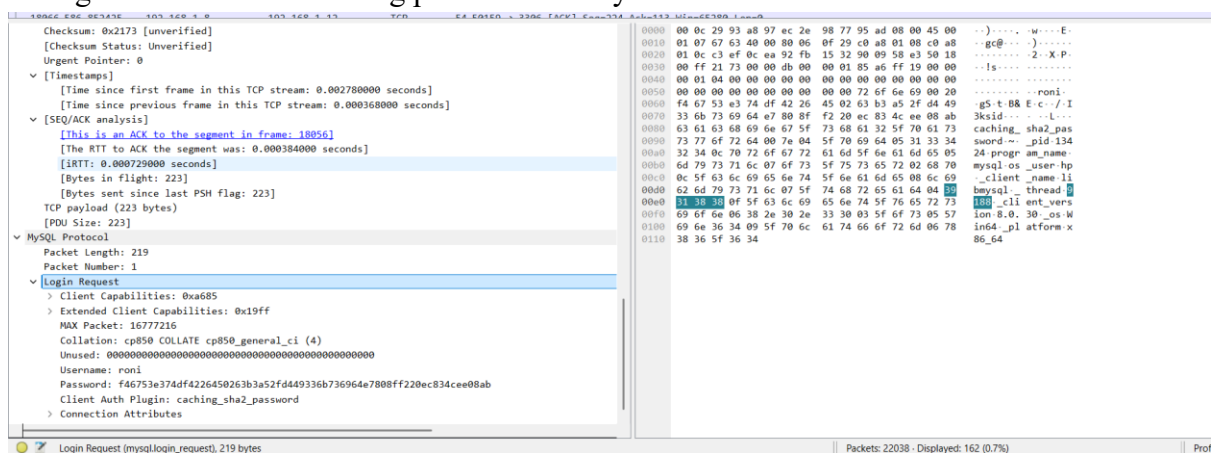


terlihat bahwa, pada kiri bawah bagian informasi login request. Usernamenya akan menampilkan informasi tentang kredensial dari username yang di pakai, terlihat bahwa username yang di pakai bernama widi, sedangkan password nya. Secara default terenkripsi. Tetapi, hanya enkripsi menggunakan algoritma sha2, yang artinya sangat tidak aman jika hanya menggunakan enkripsi itu saja. Maka dari itu kita di haruskan menggunakan SSL pada saat menggunakan database

4. Setelah itu sekarang kita ujin dengan membuat database dengan menggunakan “creat database tanpa_menggunakan_SSL”



5. Lalu gunakan kembali filtering paket sebelumnya



Penjelasan:

Terlihat, bahwasanya pada saat kita melakukan pembuatan sebuah database pada mysql tanpa menggunakan SSL, maka akan terlihat, bahwa ada packet yang tertangkap dengan info Request Query/ pada saat dibuka packet tersebut, maka akan terlihat pada kiri bawah bahwasanya request “create database tanpa_menggunakan_SSL;” di tampilkan

di request yang di tangkap oleh wireshark, maka dari itu, dengan tidak menggunakan SSL pada saat berkomunikasi dengan mysql, merupakan cara yang tidak di rekomendasikan. Karena, para attacker akan dapat menangkap perintah query apa saja yang di kirimkan oleh kita ke sebuah server mysql.

- frame: Filter ini digunakan untuk mencari informasi di tingkat frame (bingkai) pada paket data. Setiap paket yang ditangkap oleh Wireshark disusun dalam format bingkai, dan filter ini membantu menganalisis metadata atau isi bingkai, seperti waktu, panjang paket, atau konten mentah.
- contains: Filter ini digunakan untuk mencari data yang mengandung string tertentu dalam payload paket. Ketika digunakan bersama frame, Wireshark akan mencari string tertentu (misalnya "create") di seluruh isi paket, termasuk bagian payload yang mengandung data mentah.
- "create" : Ini adalah string yang dicari dalam payload paket. Dalam konteks ini, Anda mencoba mendeteksi paket yang mengandung kata "create," yang kemungkinan besar menunjukkan perintah SQL untuk membuat database, tabel, atau entitas lainnya di server database.

3.1 Metode komunikasi mysql tanpa menggunakan SSH Tunnel:

1. Buka terminal laragon lalu ketikkan perintah berikut

```
C:\laragon\www
λ ssh -L 3336:192.168.1.12:3306 roni15@192.168.1.12
roni15@192.168.1.12's password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-52-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Sat Feb  8 03:14:43 AM UTC 2025

System load:          0.1
Usage of /:           28.1% of 23.45GB
Memory usage:        17%
Swap usage:           0%
Processes:            223
Users logged in:      1
IPv4 address for ens33: 192.168.1.12
IPv6 address for ens33: 2001:448a:1020:3cc6:20c:29ff:fe93:a897

Expanded Security Maintenance for Applications is not enabled.

131 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

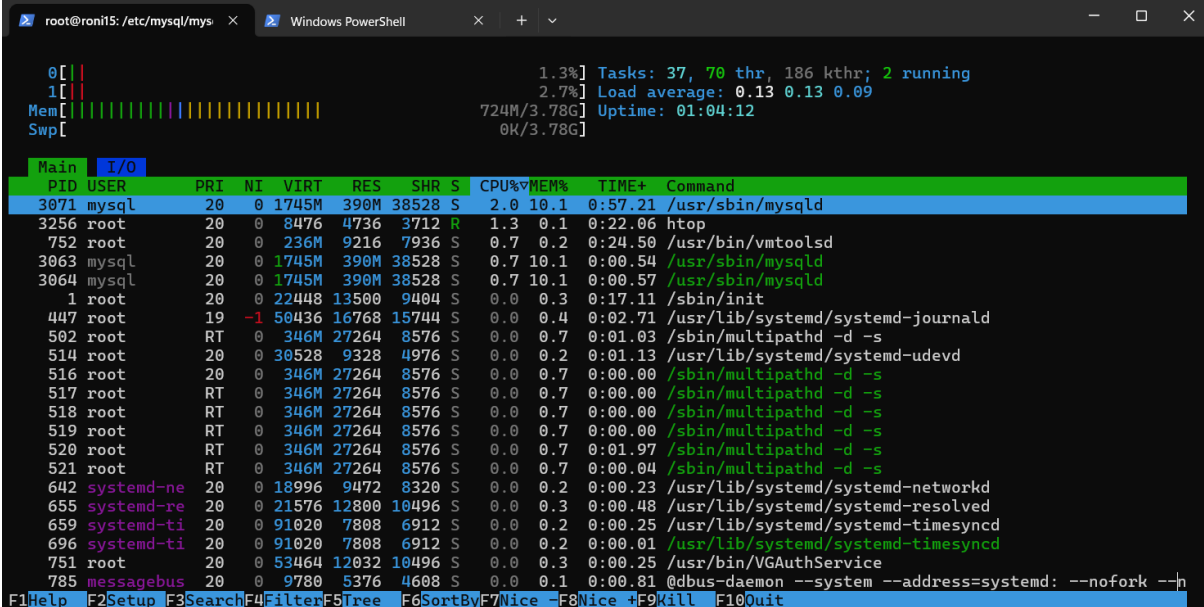
Last login: Sat Feb  8 02:27:35 2025 from 192.168.1.8
roni15@roni15:~$
```

Penjelasan:

- ssh untuk koneksi ssh ke server
- -L untuk local port forwarding fungsinya untuk mengarahkan koneksi dari port lokal ke port server tujuan melalui koneksi ssh
- 3336 adalah port lokal di vm kita dan MySQL akan terhubung ke port ini
- 3306 adalah port di server yang digunakan oleh MySQL

4.1 Pengukuran Performa

1. Setelah melakukan percobaan selanjutnya kita mengukur performa server dengan “htop” pada vm



The screenshot shows the htop interface. At the top, system statistics are displayed: Tasks: 37, 70 thr, 186 kthr; 2 running; Load average: 0.13 0.13 0.09; Uptime: 01:04:12; Memory: 724M/3.78G; Swap: 0K/3.78G. Below this is a table of running processes. The process with PID 3071 (mysql) is highlighted in blue, showing it is using 2.0% CPU and 10.1% memory. The command for this process is /usr/sbin/mysqld.

PID	USER	PRI	NI	VRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
3071	mysql	20	0	1745M	390M	38528	S	2.0	10.1	0:57.21	/usr/sbin/mysqld
3256	root	20	0	8476	4736	3712	R	1.3	0.1	0:22.06	htop
752	root	20	0	236M	9216	7936	S	0.7	0.2	0:24.50	/usr/bin/vmtoolsd
3063	mysql	20	0	1745M	390M	38528	S	0.7	10.1	0:00.54	/usr/sbin/mysqld
3064	mysql	20	0	1745M	390M	38528	S	0.7	10.1	0:00.57	/usr/sbin/mysqld
1	root	20	0	22448	13500	9404	S	0.0	0.3	0:17.11	/sbin/init
447	root	19	-1	50436	16768	15744	S	0.0	0.4	0:02.71	/usr/lib/systemd/systemd-journald
502	root	RT	0	346M	27264	8576	S	0.0	0.7	0:01.03	/sbin/multipathd -d -s
514	root	20	0	30528	9328	4976	S	0.0	0.2	0:01.13	/usr/lib/systemd/systemd-udevd
516	root	20	0	346M	27264	8576	S	0.0	0.7	0:00.00	/sbin/multipathd -d -s
517	root	RT	0	346M	27264	8576	S	0.0	0.7	0:00.00	/sbin/multipathd -d -s
518	root	RT	0	346M	27264	8576	S	0.0	0.7	0:00.00	/sbin/multipathd -d -s
519	root	RT	0	346M	27264	8576	S	0.0	0.7	0:00.00	/sbin/multipathd -d -s
520	root	RT	0	346M	27264	8576	S	0.0	0.7	0:01.97	/sbin/multipathd -d -s
521	root	RT	0	346M	27264	8576	S	0.0	0.7	0:00.04	/sbin/multipathd -d -s
642	systemd-ne	20	0	18996	9472	8320	S	0.0	0.2	0:00.23	/usr/lib/systemd/systemd-networkd
655	systemd-re	20	0	21576	12800	10496	S	0.0	0.3	0:00.48	/usr/lib/systemd/systemd-resolved
659	systemd-ti	20	0	91020	7808	6912	S	0.0	0.2	0:00.25	/usr/lib/systemd/systemd-timesyncd
696	systemd-ti	20	0	91020	7808	6912	S	0.0	0.2	0:00.01	/usr/lib/systemd/systemd-timesyncd
751	root	20	0	53464	12032	10496	S	0.0	0.3	0:00.25	/usr/bin/VGAuthService
785	messagebus	20	0	9780	5376	4608	S	0.0	0.1	0:00.81	@dbus-daemon --system --address=systemd: --nofork --

mysql> create table users (
-> id INT auto_increment primary key,
-> username varchar(20) not null,
-> password varchar(20) not null
->);
Query OK, 0 rows affected (0.08 sec)

mysql> |

Saya menggunakan htop untuk pengukuran performanya, tampak di gambar atas cpunya hanya sampai 2.0% karena kita sedang melakukan perintah didatabasenya sehingga cpunya tidak terlalu berkerja, untuk mengetes cpu agar bekerja lebih banyak saya menggunakan mysqlslap

2. Selanjutnya ketikkan perinah berikut di termnial laragon

```
roni15@roni15:~$ mysqlslap --user=roni --host=192.168.1.12 --password=1505 --concurrency=10 --iteration  
s=200 --create-schema=test --number-int-cols=10 --number-char-cols=10 --auto-generate-sql --verbose  
mysqlslap: [Warning] Using a password on the command line interface can be insecure.  
mysqlslap: Error when connecting to server: Access denied for user 'roni'@'192.168.1.12' (using password  
: YES)  
roni15@roni15:~$
```

Penjelasan:

Mysqlslap adalah tool benchmarking bawaan MySQL yang digunakan untuk mengukur performa server MySQL. Tool ini membantu kita untuk melakukan pengujian beban dengan mensimulasikan banyak koneksi klien yang menjalankan berbagai query ke server MySQL

- -user=widi untuk menentukan nama pengguna MySQL yang digunakan untuk login
- -host=192.168.1.16 menentukan alamat host tempat server MySQL berjalan

- -password=123 menentukan password untuk user ari
 - -concurrency=10 menentukan jumlah koneksi klien MySQL yang dijalankan secara bersamaan, dalam hal ini 10 klien akan mengirim query ke server secara bersamaan
 - -iterations=200 perintah ini akan menjalankan 200 pengujian (batch) dengan masing masing batch terdiri dari 10 koneksi klien
 - -create-schema=test menentukan skema yang digunakan selama pengujian, dalam hal ini database bernama test akan digunakan pastikan database test sudah ada di server
 - -number-int-cols=10 menentukan jumlah kolom bertipe integer yang akan dibuat secara otomatis dalam tabel uji, dalam hal ini tabel akan memiliki 10 kolom bertipe integer
 - -number-char-cols=10 menentukan jumlah kolom bertipe char/varchar yang akan dibuat secara otomatis dalam tabel uji dalam hal ini tabel akan memiliki 10 kolom bertipe karakter
 - -auto-generate-sql menginstruksikan mysqlslap untuk secara otomatis menghasilkan query SQL untuk pengujian tool ini akan membuat tabel, memasukkan data tabel, dan menjalankan query SELECT, INSERT, UPDATE, dan DELETE secara otomatis
 - -verbose menampilkan informasi detail selama pengujian berlangsung seperti statistik hasil pengujian
3. Selah menggunakan mysqlslap cpu terliha naik sampai 4.0%, secara teknis tercatat sampai 7.0% namun saya hanya mempunyai bukti sampai 4.0%

```

root@roni15: /etc/mysql/mys Windows PowerShell
0.0% Tasks: 37, 70 thr, 187 kthr; 1 running
1.0% Load average: 0.08 0.12 0.09
Mem[ 725M/3.78G] Uptime: 01:04:36
Swp[ 0K/3.78G]

Main I/O
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
3071 mysql 20 0 1745M 390M 38528 S 4.0 10.1 0:57.67 /usr/sbin/mysqld
3279 root 20 0 8476 4736 3712 R 1.3 0.1 0:00.22 htop
752 root 20 0 236M 9216 7936 S 0.7 0.2 0:24.58 /usr/bin/vmtoolsd
1 root 20 0 22448 13500 9404 S 0.0 0.3 0:17.11 /sbin/init
447 root 19 -1 50436 16768 15744 S 0.0 0.4 0:02.71 /usr/lib/systemd/systemd-journald
502 root RT 0 346M 27264 8576 S 0.0 0.7 0:01.03 /sbin/multipathd -d -s
514 root 20 0 30528 9328 4976 S 0.0 0.2 0:01.13 /usr/lib/systemd/systemd-udevd
516 root 20 0 346M 27264 8576 S 0.0 0.7 0:00.00 /sbin/multipathd -d -s
517 root RT 0 346M 27264 8576 S 0.0 0.7 0:00.00 /sbin/multipathd -d -s
518 root RT 0 346M 27264 8576 S 0.0 0.7 0:00.00 /sbin/multipathd -d -s
519 root RT 0 346M 27264 8576 S 0.0 0.7 0:00.00 /sbin/multipathd -d -s
520 root RT 0 346M 27264 8576 S 0.0 0.7 0:01.99 /sbin/multipathd -d -s
521 root RT 0 346M 27264 8576 S 0.0 0.7 0:00.04 /sbin/multipathd -d -s
642 systemd-ne 20 0 18996 9472 8320 S 0.0 0.2 0:00.23 /usr/lib/systemd/systemd-networkd
655 systemd-re 20 0 21576 12800 10496 S 0.0 0.3 0:00.48 /usr/lib/systemd/systemd-resolved
659 systemd-ti 20 0 91020 7808 6912 S 0.0 0.2 0:00.25 /usr/lib/systemd/systemd-timesyncd
696 systemd-ti 20 0 91020 7808 6912 S 0.0 0.2 0:00.01 /usr/lib/systemd/systemd-timesyncd
751 root 20 0 53464 12032 10496 S 0.0 0.3 0:00.25 /usr/bin/VGAuthService
785 messagebus 20 0 9780 5376 4608 S 0.0 0.1 0:00.81 @dbus-daemon --system --address=systemd: --nofork --n
808 polkitd 20 0 300M 7936 7168 S 0.0 0.2 0:00.39 /usr/lib/polkit-1/polkitd --no-debug
828 root 20 0 236M 9216 7936 S 0.0 0.2 0:00.00 /usr/bin/vmtoolsd
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice F8Nice F9Kill F10Quit

```