

Análisis de logs en Bases de datos

Mariadb para análisis de logs.

Actualización de los repositorios del sistema.

```
sudo apt-get update
```

Actualización e instalación de MariaDB

```
sudo apt update
sudo apt install mariadb-server
sudo systemctl status mariadb
```

Una vez instalada Mariadb comprobamos con el estatus:

```
mariadb.service - MariaDB 10.6.12 database server
Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
Active: active (running) since Thu 2023-08-03 13:13:52 UTC; 11min ago
Docs: man:mariadb(8)
      https://mariadb.com/kb/en/library/systemd/
Process: 9717 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/run/mysqld (code=exited, status=0/SUCCESS)
Process: 9718 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
Process: 9720 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= || VAR=cd /usr/bin/..; /usr/bin/galera_recovery; [ $? -eq 0 ] && systemctl set-environment _WSREP_START_POSITION=$VAR || exit 1 (code=exited, status=0/SUCCESS)
Process: 9763 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
Process: 9765 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)
Main PID: 9749 (mariadb)
Status: "Taking your SQL requests now..."
Tasks: 8 (limit: 4556)
Memory: 61.0M
CPU: 235ms
CGroup: /system.slice/mariadb.service
└─9749 /usr/sbin/mariadb

ago 03 13:13:52 server2 mariadb[9749]: 2023-08-03 13:13:52 0 [Note] InnoDB: 10.6.12 started; log sequence number 41320; transaction id 14
ago 03 13:13:52 server2 mariadb[9749]: 2023-08-03 13:13:52 0 [Note] Plugin 'FEEDBACK' is disabled.
ago 03 13:13:52 server2 mariadb[9749]: 2023-08-03 13:13:52 0 [Note] InnoDB: Loading buffer pool(s) from /var/lib/mysql/lib_buffer_pool
ago 03 13:13:52 server2 mariadb[9749]: 2023-08-03 13:13:52 0 [Warning] You need to use --log-bin to make --expire-logs-days or --binlog-expire-logs-seconds work.
ago 03 13:13:52 server2 mariadb[9749]: 2023-08-03 13:13:52 0 [Note] Server socket created on IP: '127.0.0.1'.
ago 03 13:13:52 server2 mariadb[9749]: 2023-08-03 13:13:52 0 [Note] InnoDB: Buffer pool(s) load completed at 230803 13:13:52
ago 03 13:13:52 server2 mariadb[9749]: 2023-08-03 13:13:52 0 [Note] /usr/sbin/mariadb: ready for connections.
ago 03 13:13:52 server2 mariadb[9749]: Version: '10.6.12-MariaDB-0ubuntu22.04.1' socket: '/run/mysql/mysql.sock' port: 3306 Ubuntu 22.04
ago 03 13:13:52 server2 systemd[1]: Started MariaDB 10.6.12 database server.
ago 03 13:13:52 server2 /etc/mysql/debian-start[9778]: Checking for insecure root accounts.
```

Como no se tiene una configuración predeterminada de usuario ni password la contraseña esta vacía y el usuario por defecto es root. Ingresamos a la base de datos:

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

Creamos un usuario que pueda conectarse de manera remota a las bases de datos y le otorgamos los privilegios necesarios para acceder con los siguientes comandos.

```
CREATE USER 'remoto'@'%' IDENTIFIED BY '1234';  
grant all privileges on *.* to 'remoto'@'%' with grant option;
```

Realizamos la creación del usuario saliendo de root e ingresando con usuario “remoto” de manera local.

```
MariaDB [(none)]>exit  
mysql -u remoto -p  
exit
```

Ingresamos de forma exitosa:

```
root@server2:/home/server# mysql -u remoto -p  
Enter password:  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 35  
Server version: 10.6.12-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
MariaDB [(none)]> []
```

Habilitamos el acceso remoto de Mariadb, salimos del usuario creado e ingresamos al archivo de configuración desde el terminal y cambiamos el valor de bind address.

```
nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

```
# Instead of skip-networking the default is now to listen only on  
# localhost which is more compatible and is not less secure.  
bind-address            = 0.0.0.0
```

Cabe destacar que esta configuración solo es recomendable para entornos de prueba, en un entorno de producción tenemos que restringir lo mas posible que dirección o direcciones IP pueden conectarse.

Reiniciamos Mariadb para aplicar los cambios.

```
sudo systemctl restart mariadb
```

Tenemos la dirección ip del servidor “192.168.24.84” con la que realizaremos la prueba remota desde nuestro terminal.

```
mysql -h 192.168.24.84 -P 3306 -u remoto -p
```

```
(rchavez@rchavez)~  
$ mysql -h 192.168.24.84 -P 3306 -u remoto -p  
Enter password:  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 31  
Server version: 10.6.12-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MariaDB [(none)]> show databases;  
+-----+  
| Database | Alerts |  
+-----+  
| information_schema | |  
| mysql | |  
| performance_schema | |  
| sys | |  
+-----+  
4 rows in set (0,001 sec)
```

Para que se registren los logs de MariaDB se deben realizar cambios en el archivo de configuración quitando el símbolo de comentario en las siguientes líneas.

```
nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

```
# Both location gets rotated by the cronjob.  
# Be aware that this log type is a performance killer.  
# Recommend only changing this at runtime for short testing periods if needed!  
general_log_file      = /var/log/mysql/mysql.log  
general_log           = 1
```

Los logs se registran en el directorio `/var/lib/mysql/server_audit.log`. Luego debemos instalar el plugging de auditoria en el servidor mariadb.

```
mysql -u remoto -p
```

```
INSTALL SONAME 'server_audit';  
show plugins;  
SHOW GLOBAL VARIABLES LIKE "server_Audit%";  
SET GLOBAL server_audit_events= 'CONNECT,QUERY_DML,TABLE';  
SET GLOBAL server_audit_logging =ON;  
SHOW GLOBAL VARIABLES LIKE "server_Audit%";  
SET GLOBAL server_audit_excl_users = remoto;  
SHOW GRANTS FOR 'remoto';
```

```
nano /etc/mysql/mariadb.conf.d/50-server.cnf
plugin_load=server_audit=server_audit.so
sudo systemctl restart mariadb
```

Instalación de rsyslog en nuestro servidor MariaDB.

```
sudo apt-get install -y rsyslog
```

Después debemos enviar los registros al servidor de Wazuh. En el archivo de configuración de rsyslog, deberá definir qué enviar al servidor de Wazuh, por UDP o TCP, la dirección IP del servidor de Wazuh y el puerto.

```
sudo nano /etc/rsyslog.conf
```

```
# provides UDP syslog reception
#module(load="imudp")
#input(type="imudp" port="514")
*. * @192.168.24.92:514
```

Para aplicar los cambios se debe reiniciar el servicio Rsyslog.

```
systemctl restart rsyslog
```

Se debe habilitar el complemento de auditoría

```
nano /etc/mysql/my.cnf
```

```
#Complemento de auditoría
plugin_load_add = server_audit
#habilitación de plugin de auditoría
server_audit = FORCE_PLUS_PERMANENT
#archivo de salida
server_audit_logging=ON
server_audit_file_path=/var/log/mysql/mariadb-audit.log

# registro
server_audit_events='CONNECT,QUERY,TABLE,QUERY_DDL,QUERY_DML,QUERY_DCL,QUERY_DML_NO_SELECT'

# Syslog de auditoría
server_audit_syslog_ident='mariadb-audit'
```

Reiniciamos para aplicar los cambios

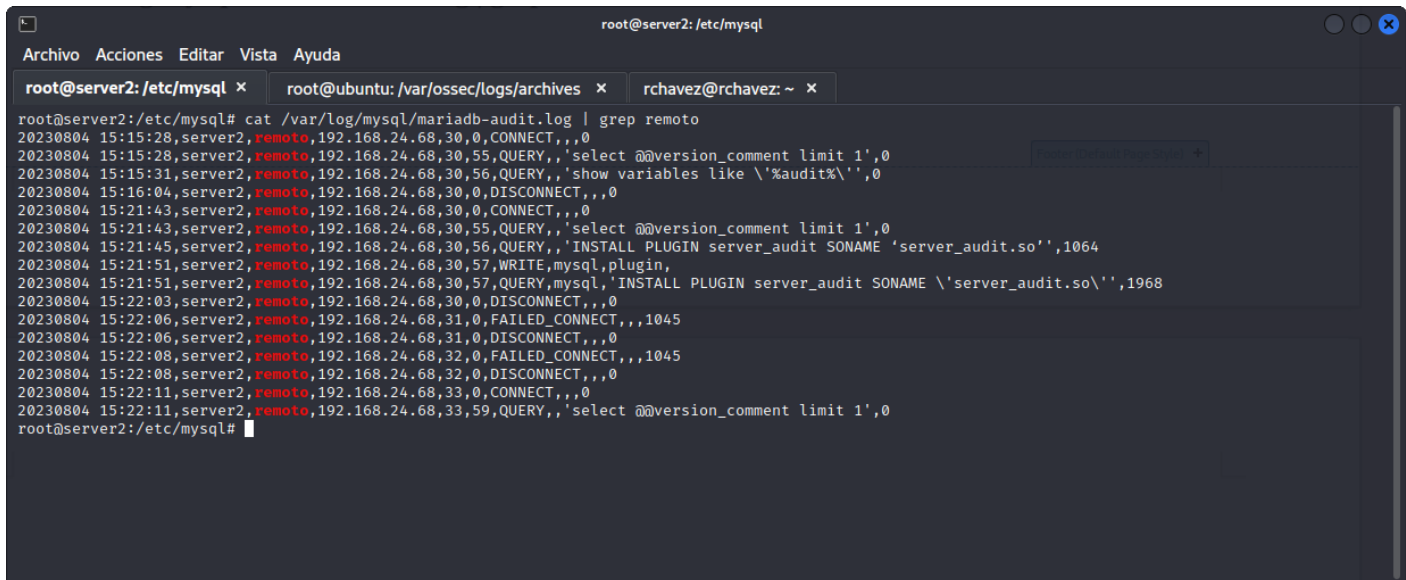
```
systemctl restart mariadb
```

Para realizar una prueba usamos una contraseña incorrecta y correcta en repetidas ocasiones desde un terminal remoto:

```
mysql -h 192.168.24.84 -P 3306 -u remoto -p
```

Verificamos el archivo de auditoria.

```
cat /var/log/mysql/mariadb-audit.log | grep remoto
```



```
root@server2: /etc/mysql
Archivo Acciones Editar Vista Ayuda
root@server2: /etc/mysql x root@ubuntu: /var/ossec/logs/archives x rchavez@rchavez: ~ x
root@server2:/etc/mysql# cat /var/log/mysql/mariadb-audit.log | grep remoto
20230804 15:15:28,server2,remoto,192.168.24.68,30,0,CONNECT,,,0
20230804 15:15:28,server2,remoto,192.168.24.68,30,55,QUERY,, 'select @@version_comment limit 1',0
20230804 15:15:31,server2,remoto,192.168.24.68,30,56,QUERY,, 'show variables like '%audit%',0
20230804 15:16:04,server2,remoto,192.168.24.68,30,0,DISCONNECT,,,0
20230804 15:21:43,server2,remoto,192.168.24.68,30,0,CONNECT,,,0
20230804 15:21:43,server2,remoto,192.168.24.68,30,55,QUERY,, 'select @@version_comment limit 1',0
20230804 15:21:45,server2,remoto,192.168.24.68,30,56,QUERY,, 'INSTALL PLUGIN server_audit SONAME 'server_audit.so'',1064
20230804 15:21:51,server2,remoto,192.168.24.68,30,57,WRITE,mysql,plugin,
20230804 15:21:51,server2,remoto,192.168.24.68,30,57,QUERY,mysql,'INSTALL PLUGIN server_audit SONAME \'server_audit.so'',1968
20230804 15:22:03,server2,remoto,192.168.24.68,30,0,DISCONNECT,,,0
20230804 15:22:06,server2,remoto,192.168.24.68,31,0,FAILED_CONNECT,,,1045
20230804 15:22:06,server2,remoto,192.168.24.68,31,0,DISCONNECT,,,0
20230804 15:22:08,server2,remoto,192.168.24.68,32,0,FAILED_CONNECT,,,1045
20230804 15:22:08,server2,remoto,192.168.24.68,32,0,DISCONNECT,,,0
20230804 15:22:11,server2,remoto,192.168.24.68,33,0,CONNECT,,,0
20230804 15:22:11,server2,remoto,192.168.24.68,33,59,QUERY,, 'select @@version_comment limit 1',0
root@server2:/etc/mysql#
```

Por ultimo verificamos el archivo syslog para confirmar que la información este correlacionada.

```
root@server2:/etc/mysql# cat /var/lo
local/ lock/ log/
root@server2:/etc/mysql# cat /var/log/syslog | grep remoto
Aug 3 15:10:25 server2 mariadb[10699]: 2023-08-03 15:10:25 31 [Warning] Access denied for user 'remoto'@'192.168.24.68' (using password: YES)
Aug 3 15:10:29 server2 mariadb[10699]: 2023-08-03 15:10:29 32 [Warning] Access denied for user 'remoto'@'192.168.24.68' (using password: YES)
Aug 3 15:24:24 server2 mariadb[10699]: 2023-08-03 15:24:24 34 [Warning] Access denied for user 'remoto'@'192.168.24.68' (using password: YES)
Aug 3 15:25:01 server2 mariadb[10699]: 2023-08-03 15:25:01 35 [Warning] Access denied for user 'remoto'@'192.168.24.68' (using password: YES)
Aug 3 15:25:07 server2 mariadb[10699]: 2023-08-03 15:25:07 36 [Warning] Access denied for user 'remoto'@'192.168.24.68' (using password: YES)
Aug 3 16:04:07 server2 mariadb[10699]: 230803 16:04:07 server_audit: server_audit_excl_users set to 'remoto'.
Aug 3 16:13:22 server2 mariadb[10699]: 2023-08-03 16:13:22 39 [Warning] Access denied for user 'remoto'@'localhost' (using password: YES)
Aug 4 14:38:02 server2 mariadb[22622]: 2023-08-04 14:38:02 30 [Warning] Access denied for user 'remoto'@'192.168.24.68' (using password: YES)
Aug 4 15:22:06 server2 mariadb[23017]: 2023-08-04 15:22:06 31 [Warning] Access denied for user 'remoto'@'192.168.24.68' (using password: YES)
Aug 4 15:22:08 server2 mariadb[23017]: 2023-08-04 15:22:08 32 [Warning] Access denied for user 'remoto'@'192.168.24.68' (using password: YES)
```

En el servidor configuramos la conexión remota con esta acción Wazuh también obtendrá registros de nuestro servidor MariaDB, para eso, necesitamos editar el archivo de configuración.

```
sudo nano /var/ossec/etc/ossec.conf
```

```
<!-- Conexión por puerto 514 a mariadb -->
<remote>
  <connection>syslog</connection>
  <port>514</port>
```

```
<protocol>udp</protocol>
<allowed-ips>192.168.24.0/24</allowed-ips>
</remote>
```

```
systemctl restart wazuh-manager.service
```

Para el archivo de logs en Wazuh, en el agente.

```
<!-- Registro de logs mariadb -->
<localfile>
  <log_format>syslog</log_format>
  <location>/var/log/mysql/mariadb-audit.log</location>
</localfile>
```

```
nano /var/ossec/etc/rules/local_rules.xml
```

```
<!--
                                REGLAS PERSONALIZADAS MARIADB
-->
<group name="mariadb-syslog,">
    <!-- Acceso fallido -->

    <rule id="100002" level="4">
    < if_sid>88100</if_sid>
    < match>FAILED_CONNECT</match>
    < description>Acceso fallido a base de datos mariadb.</description>
    </rule>
</group>
```

```
systemctl restart wazuh-manager
```

Con todas las configuraciones realizadas se pueden ver las alertas en el dashboard de Wazuh manager.

| Table | JSON | Rule |
|------------------|------|---|
| @timestamp | | 2023-08-21T15:21:49.364Z |
| _id | | 2Z-wGloB26m-iovAM9Yy |
| agent.id | | 008 |
| agent.ip | | 192.168.24.84 |
| agent.name | | basedatos1 |
| data.dstuser | | remoto |
| data.scrip | | 192.168.24.68 |
| data.srcuser | | server2 |
| decoder.name | | mariadb-syslog |
| full_log | | 20230821 15:21:48,server2,remoto,192.168.24.68,75,0,FAILED_CONNECT,,,1045 |
| id | | 1692631309.382381 |
| input.type | | log |
| location | | /var/log/mysql/mariadb-audit.log |
| manager.name | | ubuntu |
| rule.description | | Acceso fallido a base de datos mariadb. |
| rule.firetimes | | 2 |
| rule.groups | | mariadb-syslog |
| rule.id | | 100002 |
| rule.level | | 4 |
| rule.mail | | false |
| timestamp | | 2023-08-21T15:21:49.364+0000 |

Elaboración de reglas en Wazuh manager para la auditoria de bases de datos.

El primer paso para las reglas personalizadas esta en el servidor de wazuh manager, ingresamos al archivo de configuración y añadimos el directorio de reglas personalizadas.

```
nano /var/ossec/etc/ossec.conf
```

Dentro del archivo de configuración en <ossec_config> y <ruleset>

```
<ruleset>
<!-- Código anterior ... -->

<!-- Directorio de reglas locales de wazuh -->
  <rule_dir>/var/ossec/etc/rules/local_rules.xml</rule_dir>
</ruleset>
```

Decodificador para MariaDB

La creación de un decoder para Mariadb es necesaria para la interpretación de Wazuh manager para posteriormente crear alertas generadas por actividades realizadas dentro de las bases de datos. Los tipos de logs de mariadb generados por el plugin de auditoria son de la siguiente manera:

- Para el caso de conexión fallida:

```
20240321 13:54:26,server2,remoto,192.168.24.68,45,0,FAILED_CONNECT,,,1045
```

- Para eliminación de base de datos y tablas

```
20240317 16:10:02,server2,remoto,192.168.24.68,31,65,QUERY,mysql,'drop database dbprueba',0
```

```
20240316 19:26:25,server2,remoto,192.168.24.68,71,140,QUERY,uno,'drop table tablapru',0
```

Ejemplo de decodificador, en el archivo de decodificadores locales de Wazuh:

```
<decoder name="mariadb-syslog">
  <prematch> ^(\d+ \d\d:\d\d:\d\d),</prematch>
  <regex offset="after_prematch"> ^\w+,(\w+),(\S+),\d+,\d+,(w+),</regex>
  <order>dstuser,srcip,action</order>
</decoder>
```

El nombre del decodificador es muy importante ya que tiene que pertenecer al grupo de decodificadores por defecto de Wazuh para que no entre en conflicto con los decodificadores que interpreta Wazuh manager si es que estos están dentro de los programas monitoreados por defecto. Se realiza la prueba de este decodificador donde pre-match es la coincidencia de elementos al principio de cada log, y la parte de regex es la que nos importa para extraer los datos necesarios para nosotros, que se extraen en el apartado order. Para verificar su funcionamiento tenemos una herramienta llamada decoders test en Wazuh manager que nos indica si los datos que necesitamos se pudieron extraer de forma correcta.

```
▶ Test

**Messages:
  WARNING: (7003): '9208dc78' token expires
  INFO: (7202): Session initialized with token '32e61ca8'

**Phase 1: Completed pre-decoding.
  full event: '20230817 16:10:02,server2,remoto,192.168.24.68,31,65,QUERY,mysql,'drop database cinco',0'

**Phase 2: Completed decoding.
  name: 'mariadb-syslog'
  dstuser: 'remoto'
  scrip: '192.168.24.68'
  srcuser: 'server2'
```

```
Decoders Test

20230821 13:54:26,server2,remoto,192.168.24.68,45,0,FAILED_CONNECT,,,1045

▶ Test

**Messages:
  WARNING: (7003): '9208dc78' token expires
  INFO: (7202): Session initialized with token '86a5f1de'

**Phase 1: Completed pre-decoding.
  full event: '20230821 13:54:26,server2,remoto,192.168.24.68,45,0,FAILED_CONNECT,,,1045'

**Phase 2: Completed decoding.
  name: 'mariadb-syslog'
  dstuser: 'remoto'
  scrip: '192.168.24.68'
  srcuser: 'server2'
```

Podemos observar los tipos de logs generados por el plugin de auditoria de mariadb enviados a Wazuh manager para que sea decodificado por nuestro decodificador y posteriormente reconoce los datos requeridos como el nombre del decodificador, usuarios e IP proveniente.

Para la siguiente fase debemos crear reglas personalizadas a partir de estos decodificadores.

Reglas personalizadas para MariaDB.

Debemos crear una regla personalizada para que este detecte cuando una base de datos ha sido borrada. En este caso la regla se activa cuando Wazuh manager encuentra la palabras drop database, ingresamos al archivo de configuración.

```
nano /var/ossec/etc/rules/local_rules.xml
```

```
<!--  
  
                REGLAS PERSONALIZADAS MARIADB  
  
-->  
<group name="mariadb-syslog,">  
    <!-- Acceso fallido -->  
    []<rule id="100002" level="4">  
        [] <if_sid>88100</if_sid>  
        [] <match>FAILED_CONNECT</match>  
        [] <description>Acceso fallido a base de datos mariadb.</description>  
    []</rule>  
  
        <!-- Eliminación de tabla, base de datos, usuario -->  
        <rule id="100003" level="4">  
            <if_sid>88100</if_sid>  
            <match>DROP TABLE</match>  
            <description>Tabla eliminada en Mariadb.</description>  
        </rule>[]  
    []<rule id="100004" level="6">  
        <if_sid>88100</if_sid>  
        <match>DROP DATABASE</match>  
        <description>Base de datos eliminada en Mariadb</description>  
    []</rule>  
    []<rule id="100005" level="8">  
        <if_sid>88100</if_sid>  
        <match>DROP USER</match>  
        <description>Alerta inusual usuario eliminado en Mariadb</description>  
    []</rule>  
  
        <!-- Creación de usuario o cambio de atributos -->  
        []<rule id="100006" level="3">  
            <if_sid>88100</if_sid>  
            <match>CREATE USER</match>  
            <description>Usuario creado en Mariadb</description>  
        []</rule>  
    []<rule id="100007" level="8">  
        <if_sid>88100</if_sid>  
        <match>GRANT ALL PRIVILEGES</match>  
        <description>Alerta inusual se han cambiado los atributos de un usuario en Mariadb</description>  
    []</rule>[]
```

```
</group>
```

Para realizar la prueba ingresamos de forma remota

```
mysql -h 192.168.24.84 -P 3306 -u remoto -p
```

Se crea una base de datos de prueba y se elimina.

```
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+-----+
4 rows in set (0,001 sec)

MariaDB [(none)]> CREATE DATABASE prueba;
Query OK, 1 row affected (0,001 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql              |
| performance_schema |
| prueba             |
| sys                |
+-----+
```

```
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+-----+
4 rows in set (0,001 sec)

MariaDB [(none)]> CREATE DATABASE prueba;
Query OK, 1 row affected (0,001 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql              |
| performance_schema |
| prueba             |
| sys                |
+-----+
```

Posteriormente observamos la alerta en Wazuh manager.

| Time | Technique(s) | Tactic(s) | Description | Level | Rule ID |
|-----------------------------|--------------|-----------|---------------------------------|-------|---------|
| Aug 21, 2023 @ 11:26:19.688 | | | Base de datos eliminada Mariadb | 8 | 100004 |

| Table | JSON | Rule |
|------------------|------|--|
| @timestamp | | 2023-08-21T15:26:19.688Z |
| _id | | 6Z-DQloB26m-lovAUdbt |
| agent.id | | 008 |
| agent.ip | | 192.168.24.84 |
| agent.name | | basedatos1 |
| data.dstuser | | remoto |
| data.scrip | | 192.168.24.68 |
| data.srcuser | | server2 |
| decoder.name | | mariadb-syslog |
| full_log | | 20230821 15:26:19.server2,remoto,192.168.24.68,76,252,QUERY_mysql,'drop database onceymedia',0 |
| id | | 1692631579.985084 |
| input.type | | log |
| location | | /var/log/mysql/mariadb-audit.log |
| manager.name | | ubuntu |
| rule.description | | Base de datos eliminada Mariadb |
| rule.firedtimes | | 1 |
| rule.groups | | mariadb-syslog |

Posteriormente podemos realizar una regla mas en caso de que se elimine una tabla dentro de una base de datos:

```

35 <rule id="100004" level="7">
36   <if_sid>88100</if_sid>
37   <match>drop table </match>
38   <description>Tabla eliminada en Mariadb.</description>
39 </rule>

```

Revisando las alertas generadas en el dashboard de Wazuh manager podemos observar una alerta de nivel 7 con la ubicación del archivo de log para obtener mas información.

| Time | Technique(s) | Tactic(s) | Description | Level | Rule ID |
|-----------------------------|--------------|-----------|----------------------------|-------|---------|
| Aug 16, 2023 @ 16:33:53.137 | | | MySQL drop table detected. | 7 | 100004 |

| Table | JSON | Rule |
|------------------|------|---|
| @timestamp | | 2023-08-16T20:33:53.137Z |
| _id | | RZ80AloB26m-lovAFJKC |
| agent.id | | 008 |
| agent.ip | | 192.168.24.84 |
| agent.name | | basedatos1 |
| full_log | | 230816 20:33:52 72 Query drop table miercoles |
| id | | 1692218033.1286955 |
| input.type | | log |
| location | | /var/log/mysql/mysql.log |
| manager.name | | ubuntu |
| rule.description | | MySQL drop table detected. |
| rule.firedtimes | | 1 |
| rule.gdpr | | IV_35.7.d |
| rule.groups | | mariadb, mysql_log |
| rule.id | | 100004 |
| rule.level | | 7 |
| rule.mail | | false |
| rule.pci_dss | | 102.5 |
| timestamp | | 2023-08-16T20:33:53.137+0000 |

Revision #4

Created 9 abril 2024 11:33:53 by Ricardo Chavez

Updated 10 abril 2024 12:16:47 by Ricardo Chavez