**Title:** Redis GENPASS

**Excerpt:** Redis stores its password in SHA256 hash format. To avoid brute force or dictionary attacks on passwords, it is recommended to generate Redis passwords using the ACL GENPASS command which generates a 64-byte alphanumeric string. This command allows you to generate a strong password by specifying the number of bits that determines the output string length.

**Permalink:** redis-genpass

**Category:** Redis

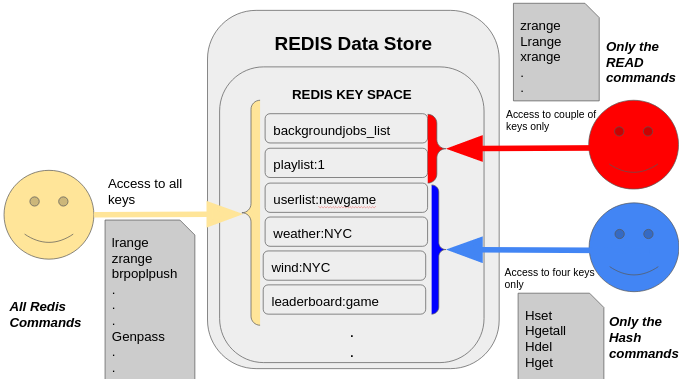
# 

# 

# 

# Redis ACL

Redis is a performance-optimized in-memory data store that uses a key-value pair format to store data. Redis allowed any new client connection to access every key and command within the data store until its version 6.0.0. From version 6.0.0, Redis introduced a new security layer with the access control list feature. With the ACL feature, Redis can limit the access levels to commands and keys for different client connections. When Redis ACL is in place, the client connection should authenticate with a username and password. Hence, each connection’s access level ranges from all keys and commands to a subset of those. The ‘default’ user has access to every command and key in the data store.



As shown in the above illustration, certain users have access to delimited keys and commands, while the default user has access to all the keys and operations in the store.

Usually, Redis supports the AUTHcommand with username and password arguments to authenticate as a specific user with certain restrictions applied.

## ACL Passwords

Redis ACL passwords are stored in SHA256 hashed format. Usually, these are more like shared secrets. These passwords are hexadecimal strings that contain pseudo-random numbers as shown in the following example.

| "ty6c75273d91t32df726fb545c8a4edc14rf0a95a6fd993950b10i814ad2r57u" |
| --- |

The ACL passwords are not entered by a human being. It is stored in some type of client software. Hence the password length is not a problem at all in this context.

In order to generate Redis ACL passwords you need to use a special command called *ACL GENPASS* that we will be discussing in the following section.

# The ACL GENPASS Command

The ACL GENPASS command is used to generate an alphanumeric string that has a size of 64-byte that contains 256 bits of pseudorandom data. To make the password stronger and hard to crack, this command uses the pseudorandom number generators like */dev/urandom* or */dev/random*. These are special types of files that can be used to create sequences of random number values where the generated random numbers have properties of a sequence of random numbers. These passwords are ten times harder to crack than a usual password.

## Syntax:

| ACL GENPASS [number\_of\_bits\_per\_output\_string] |
| --- |

***number\_of\_bits\_per\_output\_string***: This is an optional argument that specifies the number of bits needed to be contained in the output string.

This command will return a 64 bytes alphanumeric string which includes 256 bits of pseudorandom data values. Whenever you pass the optional argument where the number of bits ranges from 1 to 1024, the output string length will be changed accordingly. Usually, the output string length will be calculated by dividing the specified number of bits by 4. In addition, bits are always rounded to the next multiple of 4.

# Example 01 - Generate Default Length Password

Let’s assume we need to generate an ACL password for the default user who has access to every command and key in the Redis data store. We will use the ACL GENPASS command which is less complex and faster due to constant time complexity.



As expected the output is a 64 bytes alphanumeric string that has 256 bits of pseudorandom data. The output password length is 64 hex characters. That implies 256 bits are there.

# Example 02 - Generate Custom Length Password

The ACL GENPASS command gives full control to the user to define the number of bits needed to be in the final string. The optional parameter can be passed in the form of the number of bits to be in the generated password.

Let’s create an ACL password of the size 16 bits. Hence, the final alphanumeric string length should be 16 bits divided by four which equals 4 hex characters.

| acl genpass 16 |
| --- |



Let’s try to specify the number of bits to 6. The command will round this number to the next multiple of four. Hence, the number of bits becomes 8 under the hood.



As expected, the output string length is 2 hex characters that represent 8 bits of pseudorandom data.

As discussed in the above examples, the ACL GENPASS command can be used as a reliable password generator for Redis data stores.

# Conclusion

To summarize, Redis introduced the access control list feature from version 6.0.0. It enables a new security layer where certain client connections will be able to access limited commands and keys in the Redis data store. As mentioned, each client connection can be authenticated using the *AUTH* command with the username and password provided. These passwords are stored as SHA256 hashed values in the Redis storage. Hence, it is recommended to generate strong Redis passwords using the ACL GENPASS command. It generates a 64-byte alphanumeric string which is strong enough to avoid attacks.