

acl-genpass



- [Examples](#)

ACL users need a solid password in order to authenticate to the server without security risks. Such password does not need to be remembered by humans, but only by computers, so it can be very long and strong (unguessable by an external attacker). The `ACL GENPASS` command generates a password starting from `/dev/urandom` if available, otherwise (in systems without `/dev/urandom`) it uses a weaker system that is likely still better than picking a weak password by hand.

By default (if `/dev/urandom` is available) the password is strong and can be used for other uses in the context of an application, for instance in order to create unique session identifiers or other kind of unguessable and not colliding IDs. The password generation is also very cheap because we don't really ask `/dev/urandom` for bits at every execution. At startup Valkey creates a seed using `/dev/urandom`, then it will use SHA256 in counter mode, with HMAC-SHA256(seed,counter) as primitive, in order to create more random bytes as needed. This means that the application developer should be feel free to abuse `ACL GENPASS` to create as many secure pseudorandom strings as needed.

The command output is a hexadecimal representation of a binary string. By default it emits 256 bits (so 64 hex characters). The user can provide an argument in form of number of bits to emit from 1 to 1024 to change the output length. Note that the number of bits provided is always rounded to the next multiple of 4. So for instance asking for just 1 bit password will result in 4 bits to be emitted, in the form of a single hex character.

Examples

```
127.0.0.1:6379> ACL GENPASS
"dd721260bfe1b3d9601e7fbab36de6d04e2e67b0ef1c53de59d45950db0dd3cc"

127.0.0.1:6379> ACL GENPASS 32
"355ef3dd"

127.0.0.1:6379> ACL GENPASS 5
"90"
```

#1

2026-10-31 15:09:19 UTC

2026-10-31 15:09:19 UTC