If the validate_password component is enabled, it exposes several system variables that enable configuration of password checking:

```
mysql> SHOW VARIABLES LIKE 'validate_password.%';
+--------------------------------------+--------+
| Variable_name                        | Value  |
+--------------------------------------+--------+
| validate_password.check_user_name    | ON     |
| validate_password.dictionary_file    |        |
| validate_password.length             | 8      |
| validate_password.mixed_case_count   | 1      |
| validate_password.number_count       | 1      |
| validate_password.policy             | MEDIUM |
| validate_password.special_char_count | 1      |
+--------------------------------------+--------+
```

To change how passwords are checked, you can set these system variables at server startup or at runtime. The following list describes the meaning of each variable.

- validate_password.check_user_name

Command-Line Format	--validate-password.check-user-name[={OFF|ON}]
System Variable	validate_password.check_user_name
Scope	Global
Dynamic	Yes
SET_VAR Hint Applies	No
Type	Boolean
Default Value	ON

Whether validate_password compares passwords to the user name part of the effective user account for the current session and rejects them if they match. This variable is unavailable unless validate_password is installed.

By default, validate_password.check_user_name is enabled. This variable controls user name matching independent of the value of validate_password.policy.

When validate_password.check_user_name is enabled, it has these effects:

- Checking occurs in all contexts for which validate_password is invoked, which includes use of statements such as ALTER USER or SET PASSWORD to change the current user's password, and invocation of functions such as VALIDATE_PASSWORD_STRENGTH().

- The user names used for comparison are taken from the values of the USER() and CURRENT_USER() functions for the current session. An implication is that a user who has sufficient privileges to set another user's password can set the password to that user's name, and cannot set that user's password to the name of the user executing the statement. For example, 'root'@'localhost' can set the password for 'jeffrey'@'localhost' to 'jeffrey', but cannot set the password to 'root.

- Only the user name part of the USER() and CURRENT_USER() function values is used, not the host name part. If a user name is empty, no comparison occurs.

- If a password is the same as the user name or its reverse, a match occurs and the password is rejected.