Configuring login security and password policies

You can configure various security policies for user authentication and session management on the Security Policies landing page.

The following security policies are included:

- Password format policies that defend your system against brute force attacks, in which an attacker tries thousands of randomly generated credentials or popular passwords from a password dictionary to gain access to your application.
- CAPTCHA policies that augment passwords to guard against brute force attacks by automated processes.
- Session lockout policies that guard against brute force attacks by locking operator IDs with too many failed login attempts.
- Policies for auditing logins.
- Multifactor authentication that augments identity verification by requiring a second one-time password that is sent to the operator from a separate device or account.
- Policies for automatically disabling operators who have not been active for a specified number of days.

The first four policies are most frequently used in production systems, and it is highly recommended that you configure and enable these policies.

The Security Policies landing page is available to operators who have the pzViewAuthPoliciesLP privilege in their access roles. This privilege is part of the standard PegaRULES:SecurityAdministrator role.

Basic credentials, SAML, OpenID Connect, and anonymous authentication service types use the PRAuth servlet at run time. You can enable and disable security policies for these services from the Security policies tab of the authentication service rule form. Starting in 8.2, the default user name and password authentication is explicitly defined as an authentication service and uses PRAuth.

Authentication service types that do not use PRAuth use the servlets that are listed in Application URL patterns for various authentication service types. You enable and disable security policies for these services from the Security Policies landing page. Prior to 8.2, the default user name and password authentication uses PRServlet.

For all types of authentication, you define the security policy details, such as minimum password length and failed login attempts before lockout, on the Security Policies landing page.

You can skip ahead to the topic you're interested in:

- Enabling security policies for PRAuth-based authentication
- Enabling security policies for PRServlet-based authentication
- Password format policies
- CAPTCHA policies
- Lockout policies
- Audit policies
- Multifactor authentication policies
- Operator disablement policies
- Review history of changes to the security policies

Enabling security policies for PRAuth-based authentication

To enable a security policy for PRAuth-based authentication services, define the policy on the Security Policies landing page, and select the policy on the Security policies tab of the authentication service rule form. For more information, see Authentication services and security policies.

Enabling security policies for PRServlet-based authentication

1. In the header of Dev Studio, click Configure > System > Settings > Security Policies.
2. Select Enable frequently required policies to enable and configure password format policies, CAPTCHA, lockouts for failed logins, and logging levels for auditing logins.
3. Enable and configure two-factor authentication.
4. Enable and configure automatic disablement of inactive operators.
5. Click Submit.
Policy changes take effect immediately after you click Submit, and affect all operators. If you change password-related policies, all users are prompted to change their passwords when they next log in.

**Password format policies**

Set the minimum number of characters that an operator ID must have. For example, if your company is ThisCo, and every operator ID ends with "@thisCo.com", you can set the minimum length to 12 (the length of the mandatory string plus one character). With this policy enabled, users cannot create an operator ID that is shorter than the required length.

- Minimum operator identifier (ID) length: Set a value between 3 and 64. The default is 8.

Good system security includes requiring passwords to be complex enough so that they are not easy to guess. A password that includes letters, numbers, and special characters, and that also must be changed at regular intervals, protects the operator's account and the whole system. Set values for the following policies:

- Minimum operator password length: Set a value between 3 and 64. The default is 8.
- Minimum numeric [0-9] characters required in operator password: Set the number of numerical characters that the password must contain, between 0 and 64. The default is 1.
- Minimum alphabetical [a-z, A-Z] characters required in operator password: Set the number of alphabetical characters (a-z and A-Z) that the password must contain, between 0 and 64. The default is 1.
- Minimum lowercase [a-z] characters required in operator password: Set the minimum number of lowercase characters that the password can contain.
- Minimum uppercase [A-Z] characters required in operator password: Set the minimum number of uppercase characters that the password can contain.
- Minimum special characters required in operator password: Set the number of special characters (~!@#$%^&*()_+-=\[\]{}|:";'<>.,/) that the password must contain. The default is 1.

Make sure that the sum of the values for the preceding minimum policies is not greater than 64.

- Maximum unique historical operator passwords: Set the number of passwords that the system remembers for each operator, starting with the most recent, between 0 and 128. The default is 5.
- Maximum operator password age in days: Set the maximum number of days before the operator must change the password, between 1 and 128. The default is 30. If the value is set to 0, the password never expires.
- Minimum operator password age in days: Set the minimum number of days before the operator must change the password, between 1 and 128. The default is 0. If the value is set to 0, the password never expires.
- Minimum number of different characters between current and new operator passwords: Set the minimum number of characters that should be different between the current password and the new password when the password is changed.

**CAPTCHA policies**

A CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) creates a challenge that is easy enough for a human user to meet, but which is likely to defeat standard automated software. In its most typical form, a CAPTCHA presents an image of numbers and letters, and prompts the user to enter the characters in the image into a field. If the entered characters match the characters in the image, the system examines the operator ID and password that the user supplied.

Set the following policies:

- CAPTCHA implementation: If set to Default, the system presents the CAPTCHA implementation that is included with Pega Platform™. If set to Custom, the system presents the custom CAPTCHA implementation enabled for this system. An application can use third-party CAPTCHA solutions on the application login screen. However, a certain amount of development work is required to prepare the custom ruleset to deliver the third-party resource.
- Enable CAPTCHA Reverse Turing test module: If enabled, the system presents the CAPTCHA on authentication failure, with a probability set by the Probability that CAPTCHA will be presented field. If disabled, no CAPTCHA is displayed, even on login failure. (This setting does not apply to PRAuth-based authentication services.)
- Probability that CAPTCHA will be presented: If the CAPTCHA Reverse Turing Test is enabled, the percentage set here is the likelihood that the CAPTCHA is displayed.
- Enable presentation of CAPTCHA upon initial login: If enabled, the CAPTCHA is displayed the first time the user logs in to a new system or from a new computer.
Lockout policies

Pega Platform offers two types of lockout policies. In the first type, after a specified number of failed login attempts, the system imposes a penalty. The penalty is a time delay before the next login attempt is possible, with the delays increasing after each subsequent failure. For this type of lockout policy, set Enable authentication lockout policy to Enabled, and also set the following values:

- Failed login attempts before employing authentication lockout penalty: Set the number of failed attempts, between 0 and 128, after which the user experiences a login delay for each subsequent attempt.
- Initial authentication lockout penalty in seconds: This is the length of the first login delay. Set the initial delay between 0 and 128 seconds.
- Track login failures duration in minutes: After this period of inactivity, the failed login counter is reset to 0.

For example, enable the lockout penalty, set the number of failed login attempts that triggers the penalty to 5, and set the initial penalty to 10. When an automated tool (or a human) tries to log in with a user name and a random password, and fails, the system notes the failure. When the automated tool tries and fails again, the system increments the failure count. If the automated tool tries five times, the system notes that it has reached the penalty threshold and imposes the initial delay of 10 seconds before the login screen becomes available again. When the automated tool tries and fails again, the delay increases to 20 seconds (the initial delay of 10 seconds plus the penalty). When the automated tool tries and fails again, the delay increases to 30 seconds (the initial delay, plus the first penalty and the second penalty), and continues in a similar manner.

Activating account lockout

In the second type of lockout policy, to prevent brute force attacks, you can enforce account lockouts after repeated failed attempts by an operator. When an account is locked, Pega Platform does not allow additional login attempts until the account is unlocked. The account can be unlocked manually or automatically, based on your preferences. To configure your account lockout policy, set Enable authentication lockout policy to Disabled, and also set the following values:

- Failed login attempts before password lockout: Set to the maximum number of failed login attempts. When the number of failed attempts exceeds the number set in this policy, the account is locked.
- Password lockout duration in minutes: Set to the time period for which you want the account to remain locked:
  - Set the policy to a non-zero value if you want the account to be unlocked automatically after the specified time is over.
  - Set the policy to zero if you want the account to be unlocked manually.

While an account is locked, users who try to log in by using a browser authentication challenge or session time-out login page are not notified of the account lockout.

Unlocking an account

You can unlock a locked account in two ways: auto and manual.

Auto: Use this method to unlock the account automatically after a certain time period without resetting the operator's password. You can specify the time period for which the account needs to be locked in the Password lockout duration policy. Until the specified time period expires after the lockout, additional attempts are not allowed. After the lockout period expires, users can log in with valid credentials.

Manual: Use this method to unlock the account manually, which requires you to reset the password. To activate this method, set the Password lockout duration policy to zero. To unlock an account manually, do the following steps:

1. In the header of Dev Studio, click Configure > Org & Security > Authentication > Operator Access.
2. In the Locked operators section, click Unlock & Reset next to the selected operator. The account is unlocked and the password is reset and displayed on the screen.
3. Notify the operator of the new password.

Instead of notifying operators every time their account is unlocked, you can automate the notification process. To do so, customize the pyUnlockOperatorExtension activity. When you customize this activity, the password is automatically sent to the operator when you click Unlock & Reset. In addition, you can customize the message that users see when their password is reset. To do so, customize the pyPasswordResetDetails HTML rule.
Audit policies

The audit log records login attempts. For each logged event, the log captures the following information:

- **Create Date/Time** – The time of the attempt
- **Remote IP Address** – The IP address of the system from which the attempt was made
- **Remote Host** – The name of the computer involved
- **User Name** – The operator ID used in the login attempt
- **Message** – The status message returned during authentication of the login
- **Browser (User-Agent)** – HTTP Request Header "User-Agent"
- **Referrer** – HTTP Referrer of the "User-Agent"
- **Via** – Records proxies through which the request was sent

1. Select a level for the audit log level policy:
   - **None** – No log entry is added.
   - **Basic** – Record failed login attempts only.
   - **Advanced** – Record failed and successful login attempts.
2. Click Display Audit Log to review the log in its own window. The log is displayed as a Report Definition report.

The default time period covered is one month. You can adjust the range by clicking the link to the right of Filters, located above the report. You can edit, save, summarize, print, and export the report from the Actions menu.

Multifactor authentication policies

Peg Platform provides support for two-factor authentication in custom authentication services, by sending a one-time password (OTP) to an operator through email. You can also use the activities that implement two-factor authentication in case flows to authenticate users in special circumstances. Use the pxSendOTP activity to generate and store one-time passwords and the pxVerifyOTP activity to verify passwords against user entries. Settings on the Security Policies landing page control the behavior of the authentication process.

For Custom Authentication activities you can use the REST OTP Generation API and the REST OTP Verification API.

To enable two-factor authentication, you must select a valid email or SMS account that is used to send the one-time passwords to operators.

- Email account from which one-time password needs to be sent: If necessary, click the Add icon to edit the selected email account or create a new one.
- SMS account from which one-time password needs to be sent: If necessary, click the Add icon to edit the selected SMS account or create a new one.

You can override the email or SMS account setting when you call the pxSendOTP activity.

The remaining settings in this section control the authentication process. Set the allowed number of failed login attempts before the one-time password becomes invalid. Specify how long a current one-time password is valid or the time range when the operator has to respond with a valid one-time password. Finally, decide how long the current one-time password confirmation is valid before another authentication with one-time password is required for further transactions in the session.

- **Maximum one-time password failure attempts**: Set a value between 1 and 3. The default is 3. Setting a lower value helps prevent brute force attacks.
- **Maximum age of one-time password token in seconds**: The default is 30 in Pega 7 and 180 in version 8.1. The maximum age of the one-time password token must be less than the shortlived requestor timeout period, which is defined in minutes in the prconfig setting timeout/requestor/shortlived, and which defaults to 1 minute. If you set the maximum age to be greater than one minute, you must increase the timeout/requestor/shortlived setting.
- **Validity of one-time password confirmation in minutes**: The default is 60.

One-time passwords generation and storing

Use the pxSendOTP activity to generate and store one-time passwords. The pxSendOTP activity generates an OTP as a fixed eight digit number, sends it to pyEmailSettings.pyEmailAccount, and saves the OTP in memory until it expires.

This activity returns a ReferenceID value that can be used to identify the specific OTP generated for a specific user. The pxVerifyOTP activity uses the ReferenceID to later verify the user-entered OTP.
Input parameters for the pxSendOTP activity must be supplied in instance class Embed-OTPInputs supplied as parameter OTPInputs in the activity.

The obligatory parameters include:

- pyContext - Enter a business context why the OTP has to be generated.
- pySendMode - Enter how the OTP is shared with the user. This should have the value EMAIL.
- pyEmailSettings.pyEmailAccount - Enter the email account that is used for email outbound communication.
- pyEmailSettings.pyToAddress - Enter the email account that receives the OTP.
- pyEmailSettings.pySubject - Enter the subject of the email message.
- pyInputClass - Enter your application level work class so that you are able to refer to a correspondence rule.

**One-time password verification**

The pxVerifyOTP activity takes a user-entered OTP as an input and verifies that the OTP that was previously stored has not expired, and matches the OTP that the user has entered.

Input parameters for the pxVerifyOTP activity must be supplied in instance class Embed-OTPInputs supplied as parameter OTPInputs in the activity.

The obligatory parameters include:

- pyOTPValue - Enter the OTP value the user entered.
- pyReferenceID - Enter the reference ID returned by the pxSendOTP activity it sends the OTP.

**Operator disablement policies**

As a system administrator, you can manually disable and reenable operator IDs, as described in the Unlocking an account section. In addition, you can define a policy that automatically disables operator IDs that have been inactive for a specified number of days. To enable this policy, you must exclude certain operator IDs from this policy so that access to a system is never denied to administrator IDs that are needed to reenable access.

- Number of days of inactivity: Set a value between 1 and 90. The default is 90.
- Exclusion list of operator IDs: Shows a list of operators who are excluded from the policy. Click Add Operator to exclude operators from the policy.

**Review history of changes to the security policies**

Click View History on the Security Policies tab to see a report of changes to security policies. The report lists the changes, who made the changes, and when the changes took place. You can edit, save, print, and export the report from the Actions menu.