

ManageEngine Password Manager Pro – Security Specifications

Overview

Password Manager Pro (PMP) deals with administrative passwords, most of which having unlimited privileges. Any compromise on security will expose the organizations to serious risks. Keeping this in mind, PMP has been designed to offer maximum security right from application installation to user authentication, data transmission, storage and entire work flow.

Apart from the existing security measures as detailed below, we keep on striving to make the application more secure continuously. This document provides details about the security specifications of the product in brief.

Security at Various Levels

PMP protects the data at various levels, classified into the following seven categories:

1. Installation Master Key
2. Database Key
3. Authentication
4. Data Transmission
5. Data Storage
 - Dual Encryption
6. Data Access
7. FIPS-Compliant Mode
8. Provision to Prevent Execution of Malicious Scripts

Installation Master Key

PMP uses AES-256 encryption (the strongest known encryption and has been approved by the US Government) to secure the passwords and other sensitive information in the password database. The key used for encryption is auto-generated and is unique for every installation. By default, this encryption key is stored in a file named `pmp_key.key` under `<PMP_HOME>/conf` folder. For production instances, PMP does not allow the encryption key to be stored within its installation folder. This is done to ensure that the encryption key and the encrypted data, in both live and backed-up database, do not reside together.

We strongly recommend that you move and store this encryption key outside of the machine in which PMP is installed - in another machine or an external drive. You can supply the full path of the folder where you want to move the `pmp_key.key` file and manually move

the file to that location and delete any reference within PMP server installation folder. The path can be a mapped network drive or external USB (hard drive / thumb drive) device.

PMP will store the location of the pmp_key.key in a configuration file named manage_key.conf present under <PMP_HOME>/conf folder. You can also edit that file directly to change the key file location. After configuring the folder location, move the pmp_key.key file to that location and ensure the file or the key value is not stored anywhere within the PMP installation folder.

Database Key

1. Apart from the AES encryption, the PMP database is secured through a separate key, which is auto-generated and unique for every installation
2. The key for the database can be stored securely in the PMP itself
3. There is also option to store it at some other secure location accessible to the PMP server
4. The MySQL database accepts connections only from the host that it is running on and is not visible externally

Authentication

PMP allows the users to choose one of the following two types of authentication:

1. Local authentication - PMP's own authentication mechanism. It employs SHA1 algorithm to generate password, which ensures that each PMP login password is unique and irreversibly secured
2. Authentication by external identity stores - PMP can be integrated with external identity stores such as Active Directory/LDAP and can use the authentication service provided by them. When the authentication by an external identity store is enabled, local authentication can be disabled

Data Transmission

1. All data transmission between the PMP user interface and server are encrypted and take place through HTTPS.
2. For remote password reset actions, there is option to transmit passwords using SSH.

Data Storage

All sensitive data stored in PMP - passwords, files, digital keys, account names, IP addresses etc are encrypted.

Dual Encryption for Extra Security

Sensitive data are encrypted once in the application (AES 256-bit) and once in database

Data Access

1. All data access in PMP are subjected to the granular access control mechanism. Password ownership and sharing practices are well-defined and users get access only to authorized passwords
2. In the case of Application-to-Application passwords, PMP exposes a web API and the applications connect and interact with the PMP through HTTPS. The application's identity is verified by forcing it to issue a valid SSL certificate, matching the details already provided to PMP corresponding to that application.
3. All access to passwords (who accessed what passwords and when) and all operations done by the users on any resource are captured in the audit trails ensuring accountability for all users and actions

Preventing Execution of Malicious Code/Script

Provision to prevent the execution of malicious code/script in the application to combat cross-site scripting

FIPS-Compliant Mode

PMP can be set-up to run in FIPS 140-2 compliant mode (with SQL server back-end) where all encryption in PMP is done through FIPS 140-2 certified systems and libraries

Work Flow

PMP ensures security all along the application work flow. The following are few examples:

1. Enforcement of standard password policies and practices
2. Monitoring failed login attempts
3. Termination of inactive user sessions
4. Automatic passwords resets
5. Setting password age
6. Real-time notifications on password access

Disaster Recovery

The database backup generated by PMP follows all the above security aspects, which in turn makes disaster recovery secure. The backup copy will not have the Encryption Master Key since PMP does not allow encryption key and the encrypted data, in both live and backed-up database, reside together. Unless one presents the encryption key, sensitive data cannot be deciphered from the backup copy.