Security architecture
in Pandora FMS
The purpose of this document is to describe the security elements of each Pandora FMS component, so that the administrator knows them and knows how to use them to implement a more secure architecture, in accordance with regulations such as PCI / DSS, ISO 27001, ENS, LOPD or similar. In addition, this file provides a specific description of the security mechanisms of each Pandora FMS element, possible risks as well as the way to minimize them, using the tools available in Pandora FMS or other possible mechanisms.
SECURITY IMPLEMENTATION (GENERAL)

These points apply to international standards such as PCI / DSS, ISO 27001, National Security Scheme, LOPD, etc. They work as a guide for a safe Pandora FMS implementation in your environment.

1. Pandora FMS components have their input and output ports documented, so it is possible to **secure all accesses to and from their components by means of Firewalls**.

2. **Safe traffic through encryption and certificates**: Pandora FMS supports SSL / TLS encryption and certificates at both ends and at all levels (user operation, communication between components).

3. **Dual access authentication system**: A double authentication system can be implemented. The first one is placed at access level (HTTP) integrated with any open source or commercial token system.

4. **Authentication with third parties system**: located at the application level, it is managed by Pandora FMS, which can be authenticated against LDAP or Active Directory.

5. **SSO** (Single Sign-On), with SAML.

6. **Security policies in user management**: User management is defined by policies both at user profile level and at operational visibility profile level, defined as the Enterprise version Extended ACL system.

7. **Possibility of audits on the actions** of the monitored elements: Pandora FMS Enterprise version audits all user actions, including information on altered or deleted fields. In addition, it includes a validation system by signing these records.
8. **Audit data transfer to external log managers:** Audit logs are available for export through SQL and allow them to be integrated into a 3rd source for higher security, almost in real time.

9. **Physical separation** of components that offer an interface to the user and the information containers (filesystem). Both the data stored in the database and the filesystems that store monitoring configuration information can be in separate physical machines, in different networks, protected through perimeter systems.

10. **Active password policy** that enforces a strict password management policy for users to access the application (console).

11. **Sensitive data encryption.** The system allows the most sensitive data such as login credentials to be stored in an encrypted and secure manner.
SECURITY BY ARCHITECTURE COMPONENTS

Pandora FMS architecture, in a very simple way, can be summarized as follows:
SERVER:

1. The server needs root permissions, but it can be installed (with certain limits) with non-root permissions (Linux systems only).

2. The server needs direct access (read and write) to the agent remote configuration files of the agents, which are spread when the agents contact the server periodically. These files are protected by the filesystem, with standard permissions.

3. The server itself does not listen to any port. It is the Tentacle server who listens on a port, the server only accesses the files left by the Tentacle server on disk.

4. The server has its own very detailed log.

5. The server connects to the main database using a standard MySQL / TCP connection.

6. Part of the code is accessible (OpenSource) and that of the enterprise version can be requested under specific contract conditions (for customers only).
Possible vulnerabilities and safeguards

1. Unauthorized access to agent configuration files. Solution:
   • Implement an external secured container for external configuration files through NFS.

2. Command injection on remote agents through the manipulation of configuration files stored in the configuration container. Solution:
   • Disable remote configuration on highly sensitive agents after configuration and leave them running without being able to carry out any changes remotely, for total security.
   • Remote monitoring - without agents - of the most delicate devices.

3. Vulnerable against false information attacks, such as simulating agents that are not in the system or impersonating their identity. To avoid this, several mechanisms can be used:
   • Password protection system (which works by group).
   • Limiting agent self-creation, and creating them instead manually.
   • Limiting the ability to auto detect changes in the agent and not take new information from the XML, apart from the existing one.

4. Malicious capture of communication between server and console (network traffic capture). Solution:
   • Activate TLS communication between server and MySQL database.
1. Tentacle is an official internet service, documented as such by IANA. This means that it can be easily protected with any perimeter security tool.

2. It does not need root or special privileges.

3. It has four security levels: No encryption (default), SSL / TLS Basic, SSL / TLS with certificate at both ends and SSL / TLS with certificate and CA validation.

4. Specifically designed not to give clues to possible intruders in error messages and with specific timeouts to prevent brute force attacks.

5. It has its own audit log.

6. 100% of the code is accessible (under opensource GPL2 licence).

Possible weak spots and safeguards

1. Attacks on the filesystem. The configuration container must be accessed. Solutions:
   • It is protected in the same way as the server, by means of a secured external NFS system.

2. DoS attacks due to overload. Solutions:
   • Set up an HA solution on the TCP service it offers for balancing, or an active / active cluster. Any hardware or software solution available is valid because it is a standard TCP service.
CONSOLE

1. It does not need root, it is installed with a user without privileges.

2. It must have access to the agent configuration repository (filesystem).

3. It listens on standard HTTP or HTTPS ports.

4. It registers all requests via HTTP request log.

5. It offers a public API via HTTP / HTTPS, secured with credentials.

6. There is an application specific audit, which records the activity of each user on each system object.

7. Each user access to any section of the application can be restricted, and even administrators with restricted permissions can be created.

8. The application incorporates a dual authentication system.

9. The application incorporates a delegated authentication system (LDAP, AD).

10. A read-only system can be built. With no access to device configurations.

11. Confidential information (passwords) can be stored encrypted in the database.

12. The application connects to the main database using a standard MySQL / TCP connection.

13. Part of the code is accessible (OpenSource) and that of the enterprise version can be requested under specific contract conditions (for customers only).

14. There is a strong implementation of security policies regarding passwords (length, forced change, history, type of valid characters, etc.)
Possible vulnerabilities and safeguards

1. Attacks on the filesystem. The configuration container must be accessed.
   Solutions:
   • It is protected in a similar fashion as the server, by means of a secured external NFS system.

2. Brute force or dictionary attacks against user authentication. Solution:
   • Implement a hard password policy (point 14).
   • Implement a double authentication system (point 8).

3. Traffic capture (eavesdropping) of traffic to the console. Solution:
   • Implement SSL / TLS.

4. Traffic capture (eavesdropping) of traffic to the database. Solution:
   • Implement SSL / TLS.

5. SQL injection attacks to obtain confidential information from the application database. Solution:
   • Implement encrypted data storage.

6. Application user misuse (intentional or unintended). Solution:
   • Activate the audit log and show the users that it exists and its accuracy.
   • Activate the extended ACL system to restrict the functions of each user as much as possible.
   • Export the audit log to an external system on a regular basis.

7. Execution of malicious code in local console tools, replacing binary files. Solution:
   • Enforcing server security (hardening) of the server that contains the application.
AGENTS

1. It can be run without superuser permissions (with limited features).

2. Remote agent management can be disabled (locally and remotely), so that the impact of a break-in on the central system can be minimized.

3. The agent does not listen to network ports, it is the one who connects to Pandora FMS server.

4. There is a record of each execution.

5. Configuration files are protected by default through file system permissions. Only a user with super administrator permissions can modify them.

6. 100% of the code is accessible (under opensource GPL2 licence).

Possible vulnerabilities and safeguards

1. Intrusion into the central system that allows distributing malicious command execution to agents. Solutions:
   - Limit which users can make these policy or configuration modifications (via ordinary console ACL or extended ACL).
   - Activate the “readonly” mode of the agents (they do not allow configuration remote modifications) for those especially sensitive systems.

2. Vulnerability in the filesystem that allows modifying files. Solution:
   - Correct permission settings.

3. Execution of plugins or malicious commands. Solution:
   - Limit which users can upload executables (via ordinary console ACL or extended ACL).
   - Perform an audit of new plugins.
DATABASE

It is a standard product (MySQL)

Possible vulnerabilities and safeguards

1. Eavesdropping (network traffic capture). Solution:
   • Implementation of a secure TLS connection. MySQL supports it.

2. Incorrect permissions. Solution:
   • Correct configuration of access permissions.

3. Known MySQL weak spots. It is advisable to establish an update plan for the MySQL server in which you can have it as updated as possible and therefore get rid of any vulnerabilities that old versions may have.
• **AUTHENTICATION FOR MALICIOUS DATA INPUT MANAGEMENT**

Pandora FMS includes a specific system for protection against XSS and SQL injection attacks by means of codifying data in the database, which guarantees all user logins to be validated, as well as unsafe logouts. It does not trust any data input that might be misinterpreted by the application.

• **STORE CRYPTOGRAPHIC VERIFICATION**

Key access is safely managed, by codifying data with a custom password chosen by the client. The cryptage uses OpenSSL algorithms to ensure data security. In addition, a random number generator is used when randomization is required. But there is also a custom kernel, for cryptographic modules to fail safely and errors to be correctly managed.

• **DATA PROTECTION VERIFICATION**

This system grants confidentiality, integrity, and data availability. On that note, it uses an extensive AC multiprofile system, it has a fine grained audit system available and it contains several systems to grant data availability, including redundancy. Remember that Pandora FMS offers a dual authentication system that limits improper access risks.
• SECURITY VERIFICATION IN COMMUNICATIONS

Pandora FMS enables a safe communication system based on SSL/TLS in all its architecture components.

• VERIFYING CONFIGURATION REQUIREMENTS

Pandora FMS ISO distribution features a default hardening, in addition to the updated version of all its components, as well as safe configuration by omission in the application.

• BUSINESS LOGIC VERIFICATION

It does not apply, OWASP is only available for ASVS Level 2 or higher.

• MALICIOUS CONTROL VERIFICATION

It does not apply, OWASP is only available for ASVS Level 3 or higher.

• VERIFY HTTP SECURITY SETTINGS

The application server (Apache/PHP) is an standard within the whole industry and there are some procedures to ensure safe implementation.

• REST AND SOAP WEB SERVICE VERIFICATION

The application server (Apache/PHP) is an standard within the whole industry and there are some procedures to ensure safe implementation. Pandora FMS API Rest systems contain different security measures discussed in this section.
c/ Gran Vía 62
Madrid, España.
(+34) 91 559 72 22
info@pandorafms.com

www.pandorafms.com