セキュリティアーキテクチャ

Pandora FMS ドキュメント一覧に戻る

セキュリティアーキテクチャ

このドキュメントの目的は、各 Pandora FMS コンポーネントのセキュリティ要素を説明し、管理者がそれらを理解し、PCI / DSS、ISO 27001、ENS、LOPD などの規制に従って、より安全なアーキテクチャを実装し使用する方法を知ることです。 さらに、ここでは、Pandora FMS で利用可能なツールや他の取りうるメカニズムを使用して、起こりうるリスクおよびそれらを最小化する方法の説明を提供します。

セキュリティの実装 (一般)

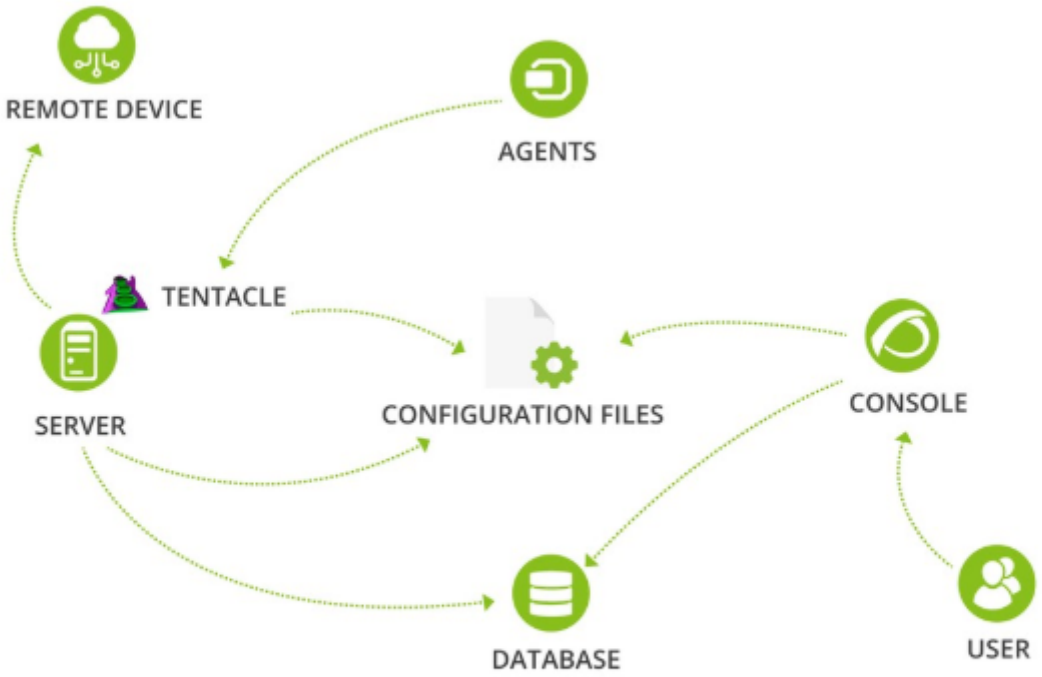
これらのポイントは、PCI / DSS、ISO 27001、National Security Scheme、LOPD などの国際標準に適用されます。これらは、それぞれの環境における安全な Pandora FMS 実装のガイドとして利用できます。

- Pandora FMS コンポーネントの入出力は文書化されているため、**ファイアウォールを使用してコンポーネントとの間のすべてのアクセスを保護する**ことができます。
- **暗号化と証明書による安全なトラフィック**、Pandora FMS は、すべてのレベル(ユーザ操作、コンポーネント間の通信)でSSL / TLS 暗号化と証明書をサポートします。
- **デュアルアクセス認証システム**：二段階認証システムを実装できます。 1つ目は、オープンソースまたは商用トークンシステムと統合されたアクセスレベル(HTTP)に配置されます。
- **サードパーティシステムとの認証**、Pandora FMS によって管理されるアプリケーションレベルで、LDAP または Active Directory による認証ができます。
- SAML による **SSO** (シングルサインオン)。
- **ユーザ管理におけるセキュリティポリシー**：ユーザ管理は、Enterprise 版の拡張 ACL システムとして定義される、ユーザプロファイルと運用可視性プロファイルの両方のポリシーによって定義されます。
- **監視対象要素の操作に対する監査が可能**、Pandora FMS Enterprise 版は、情報または変更、または削除されたフィールドを含むすべてのユーザ操作を監査します。 さらに、これらのレコードに署名できる検証システムが含まれています。

- **外部ログマネージャーへの監査データ転送**：監査ログは、SQLを介してエクスポートでき、セキュリティを高めるためにほぼリアルタイムで外部ソースに統合できます。
- ユーザインタフェースと情報コンテナ(ファイルシステム)を提供するコンポーネントの**物理的な分離**。 データベースに保存されているデータと監視構成情報を保存しているファイルシステムの両方を、境界ネットワークによって保護された、異なるネットワークの別々の物理マシンに置くことができます。
- ユーザがアプリケーション(コンソール)にアクセスするための厳密なパスワード管理ポリシーを要求する**アクティブパスワードポリシー**
- **機密データの暗号化**。 このシステムでは、ログイン資格情報などの最も機密性の高いデータを暗号化された安全な方法で保存できます。

コンポーネントごとのセキュリティ

Pandora FMS アーキテクチャは、非常に簡略化すると、次のように要約できます。



サーバ

- サーバは root 権限が必要ですが、(いくつかの制限はありますが)非 root 権限でのインストールが可能です(Linux システムのみ)
- サーバは、エージェントのリモート設定ファイルへの直接アクセス(読み取りおよび書き込み)が必要です。これらのファイルは、エージェントが定期的にサーバに接続する際に展開されます。 これらのファイルは、標準の権限でのファイルシステムによって保護されています。

- サーバ自体は任意のポートでの待ち受けをしません。待ち受けをするのは Tentacle サーバです。サーバは、Tentacle サーバがディスクに書き込んだファイルにのみアクセスします。
- サーバ自身は詳細のログを持ちます。
- サーバは、通常の MySQL / TCP 接続を用いてメインデータベースに接続します。
- コードの一部はアクセス可能(オープンソース)であり、Enterprise 版のコードは特定の契約条件の下で要求できます(お客様のみ)。

考えられる脆弱性と保護手段

- エージェント設定ファイルへの不正アクセス。解決策:

#NFS を使用して、外部構成ファイル用の外部保護コンテナを実装します。

- 設定コンテナに格納されている設定ファイルの操作によるリモートエージェントへのコマンドインジェクション。 解決策：
  1. 設定後に機密性の高いエージェントのリモート設定を無効にし、完全なセキュリティを確保するために、リモートから設定変更できないようにします。
  2. 最もデリケートなデバイスのであれば、エージェントを用いないリモート監視をします。
- システムに存在しないエージェントをシミュレートしたり、ID を偽装するなど、偽情報攻撃に対して脆弱です。 これを回避するには、いくつかのメカニズムを使用できます。
  1. (グループごとに機能する)パスワード保護システム。
  2. エージェントの自動作成を制限し、代わりに手動で作成します。
  3. すでに設定があるものを除き XML から新しい情報を取得しないようにしたり、エージェントの変更を自動検出する機能を制限します。
- サーバとコンソール間の通信の悪意のあるキャプチャ(ネットワークトラフィックキャプチャ)。解決策：
  1. サーバと MySQL データベース間の TLS 通信を有効にします。

Tentacle

- Tentacle は公式のインターネットサービスであり、IANA によって文書化されています。 これは、あらゆる境界セキュリティツールで簡単に保護できることを意味します。
- root や特別な権限は必要ありません。
- 4つのセキュリティレベルがあります：暗号化なし(デフォルト)、SSL / TLS Basic、両端に証明書がある SSL / TLS、および証明書と CA 検証がある SSL / TLS。
- 特にブルートフォース攻撃を防ぐために、特定のタイムアウトを使用して、エラーメッセージで侵入者の手がかりを与えないような特別な設計がされています。
- 独自の監査ログがあります。
- コードは 100% 公開されています。(GPL2 ライセンスによるオープンソース)

考えられる脆弱性と保護手段

- ファイルシステムへの攻撃。 設定コンテナにアクセスする必要があります。解決策：
  1. セキュアな外部 NFS システムにより、サーバと同じ方法で保護できます。
- DoS 攻撃による過負荷。解決策：
  1. バランシングのための TCP サービス、またはアクティブ/アクティブクラスターで HA ソ

リューションをセットアップします。 標準の TCP サービスであるため、いずれのハードウェアまたはソフトウェアソリューションも利用できます。

コンソール

- root は必要ありません。権限のないユーザとともにインストールされます。
- エージェント設定リポジトリー(ファイルシステム)へのアクセス権が必要です。
- 標準の HTTP または HTTPS ポートで待ち受けます。
- HTTP アクセスログを介してすべての要求を記録します。
- 資格情報で保護された HTTP / HTTPS 経由の公開 API を提供します。
- 各システムオブジェクトの各ユーザの操作を記録するアプリケーション固有の監査があります。
- アプリケーションの任意のセクションへの各ユーザのアクセスを制限できます。また、管理者においてもアクセスが制限されたユーザを作成することもできます。
- このアプリケーションには、二段階認証システムが組み込まれています。
- このアプリケーションには、外部認証システム(LDAP AD)が組み込まれています。
- 読み取り専用システムを構築できます。デバイス設定にアクセスできません。
- 機密情報(パスワード)を暗号化してデータベースに保存できます。
- アプリケーションは、標準の MySQL / TCP 接続を使用してメインデータベースに接続します。
- コードの一部はアクセス可能(OpenSource)であり Enterprise 版のコードは特定の契約条件の下で要求できます(お客様のみ)。
- パスワードに関するセキュリティポリシーの強力な実装があります(長さ、強制変更、履歴、有効な文字のタイプなど)。

考えられる脆弱性と保護手段

- ファイルシステムへの攻撃。 設定コンテナーにアクセスする必要があります。 解決策：
  1. 保護された外部 NFS システムにより、サーバと同様の方法で保護できます。
- ユーザ認証に対するブルートフォース攻撃または辞書攻撃。 解決策：
  1. 厳格なパスワードポリシーを実装します(ポイント14)。
  2. 二段階認証システムを実装します(ポイント8)。
- コンソールへのトラフィックのキャプチャ(盗聴)。 解決策：
  1. SSL/TLS を実装します。
- データベースへのトラフィックのキャプチャ(盗聴)。 解決策：
  1. SSL/TLS を実装します。
- アプリケーションデータベースから機密情報を取得する SQL インジェクション攻撃。 解決策：
  1. 暗号化データストレージの実装。
- アプリケーションユーザーの誤用(意図的または意図しない)。 解決策：
  1. 監査ログを有効化し、ユーザにその存在とその正確性を示します。
  2. 拡張 ACL システムを有効化して、各ユーザの機能をできるだけ制限します。

3. 定期的に監査ログを外部システムにエクスポートします。

- ローカルコンソールツールでの悪意のあるコードの実行、バイナリファイルの置き換え。解決策：
  1. アプリケーションを含むサーバのセキュリティ強化。

エージェント

- 管理者権限なしで実行できます(機能が制限されます)。
- リモートエージェント管理を無効にして(ローカルおよびリモート)、中央システムへの侵入の影響を最小限に抑えることができます。
- エージェントはネットワークのポートを待ち受けずPandora FMS サーバに接続します。
- 各処理実行の記録があります。
- 設定ファイルは、ファイルシステムのパーミッションによってデフォルトで保護されています。管理者権限を持つユーザのみがそれらを変更できます。
- コードは 100% 参照可能です(GPL2 ライセンスのオープンソース)。

考えられる脆弱性と保護手段

- 悪意のあるコマンドの実行をエージェントに配信できる中央システムへの侵入。 解決策：
  1. これらのポリシーまたは設定を変更できるユーザを制限します(通常のコンソール ACL または拡張 ACL を使用)。
  2. 特に機密性の高いシステムに対して、エージェントの“ 読み取り専用 ”モードを有効化します(設定のリモート変更は許可されません)。
- ファイルの変更を可能にするファイルシステムの脆弱性。 解決策：
  1. パーミッション設定を正しくします。
- プラグインまたは悪意のあるコマンドの実行。 解決策：
  1. (通常のコンソール ACL または拡張 ACL を介して)実行可能ファイルをアップロードできるユーザを制限します。
  2. 新しいプラグインの監査をします。

データベース

- 標準製品(MySQL)です。

考えられる脆弱性と保護手段

- 盗聴(ネットワークトラフィックキャプチャ。 解決策：
  1. 安全なTLS接続の実装 MySQL はそれをサポートしています。
- 不正なパーミッション。 解決策：
  1. アクセスパーミッション設定の修正。
- 既知の MySQL の弱点 MySQL サーバの更新計画を確立して、可能な限り更新することをお勧めします。これにより古いバージョンに存在する可能性のある脆弱性を取り除くことができます。