PANDORAFMS

セキュリティアーキテクチャ

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)

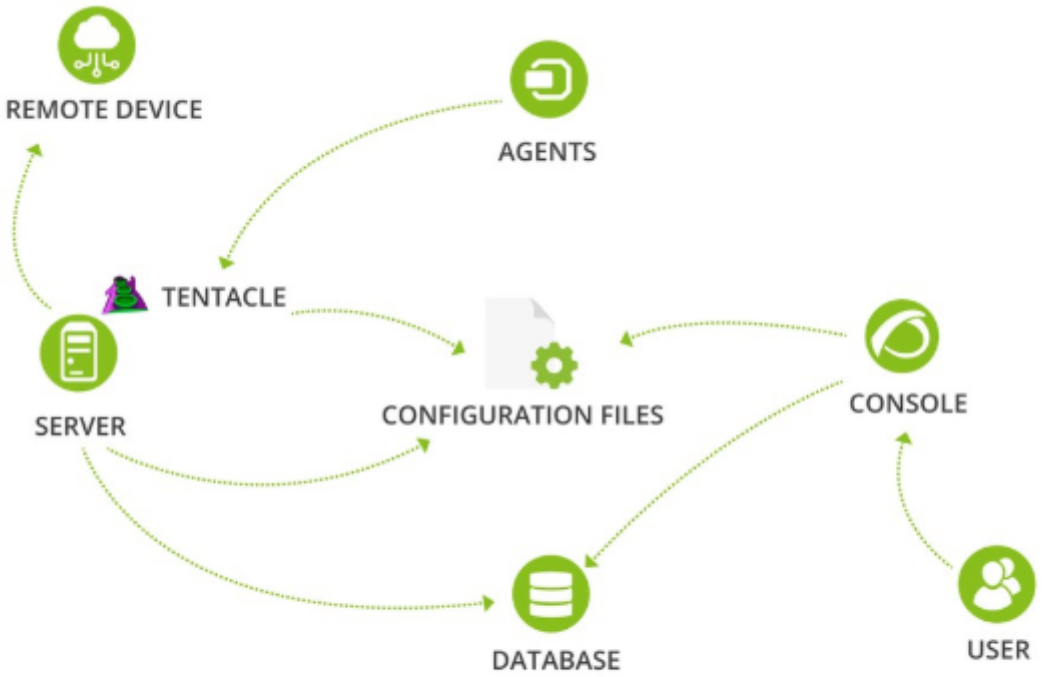
これらのポイントは、PCI / DSS、ISO 27001、National Security Scheme、LOPD などの国際標準に適用されます。これらは、それぞれの環境における安全な Pandora FMS 実装のガイドとして利用できます。

- 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 と許可された IP アドレスリストを提供します。
- 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。

ベースシステムの強化

システムの強化や保証は、企業のグローバルセキュリティ戦略の重要なポイントです。 メーカーと

して、標準の RHEL7 プラットフォームまたは同等の Centos7 に基づいて、すべての Pandora FMS コンポーネントの安全なインストールを実行するための一連の推奨事項を示します。これらの推奨事項は、GNU/Linux に基づく他の監視システムでも当てはまります。

アクセス認証

システムにアクセスするためには、特権のない必要に応じてアクセスを制限した目的に応じたユーザを作成します。 理想的には、各ユーザの認証は、トークンに基づく二段階認証システムと統合します。このガイドの範囲外ですが GNU/Linux に統合可能な Google 認証システム® などの 無料で安全な代替手段があります。 それらの使用を真剣に検討してください。

アプリケーション用に何らかのユーザを作成する必要がある場合は、リモートアクセス権限のないユーザにする必要があります(そのためには、シェルまたは同等の利用を無効にします)。

スーパーユーザアクセス

特定のユーザーが管理者権限を持つ必要がある場合は、sudo コマンドを使用します。

システムを最新の状態に保つ

ネットワークに接続するか、プロキシサーバを使用するように yum システムを設定するだけです。

このコマンドは、ライブラリー、設定などの変更で潜在的な問題を引き起こす可能性がありますが、特にシステムを本番環境として利用する場合は、システムのアップグレードを省略しないことが重要です。すでに利用中の本番システムの場合は、これらの重要なコンポーネント、つまり脆弱性のあるコンポーネントのみをアップグレードする必要がある場合があります。たとえば、mysql サーバ のみをアップグレードする場合は、アップグレードしたいパッケージ名を指定してコマンドを使用します。

```
yum update mysql-server
```

システムのアップグレードは、定期的に実行する必要がある対応です。 システムパッケージインベントリを使用して、脆弱なバージョンをチェックし、緊急アップグレードを実行します。

アクセス監査

セキュリティログ /var/log/secure を有効化し、それらのログを監視する必要があります(これについては後で説明します)。

CentOS ではこれがデフォルトで有効になっています。そうでない場合は、/etc/rsyslog.conf または /etc/syslog.conf を確認してください。

監査システムのログを取得し、外部のログ管理システムで収集することをお勧めします。Pandora FMS はそれを簡単に行うことができ、必要に応じてアラートを発報したり、一元的に確認したりするのに役立ちます。

SSH サーバの保護

SSH サーバを使用すると、GNU/Linux システムにリモート接続してコマンドを実行できるため、これは重要なポイントであり、次の点に注意して保護する必要があります(そのためには、ファイル `/etc/ssh/sshd_config` を編集し、サービスを再起動します)。

- デフォルトポートの変更 (例: 31122)

```
#Port22     ->     Port 31122
```

- スーパーユーザである root ログイン の無効化

```
#PermitRootLogin yes        ->    PermitRootLogin no
```

- ポートフォワーディング の無効化

```
#AllowTcpForwarding yes        ->    AllowTcpForwarding no
```

- トンネリング の無効化

```
#PermitTunnel no        ->    PermitTunnel no
```

- リモート root アクセスのための SSH 鍵の削除。リモートアクセスに有効なユーザーは 1人だけであると想定します(例: artica)。 他にある場合は、それらもチェックする必要があります。 これを行うには、ファイル `/home/artica/.ssh/authorized_keys` の内容を調べて、どのマシンからのものかを確認します。 存在しないはずだと思われる場合は削除します。
- サーバがプライベートアクセスであり、資格情報を持たない人は切断する旨を説明する標準のリモートアクセス警告を設定します。

```
Banner /etc/issue.net
```

MySQL サーバの保護

待ち受けポート。 MySQL サーバが外部にサービスを提供する必要がある場合は、root の認証情報が安全であることを確認するだけです。 MySQL が内部にのみサービス提供する場合は、ローカルホストでのみ待ち受けるようにします。

```
netstat -an | grep 3306 | grep LIST
tcp        0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN
```

この場合、どこからでも接続を受け付けます。制限するには、ファイル `/etc/my.cnf` を編集し、セクション `[mysqld]` に次の行を追加します。

```

```

```
bind-address = 127.0.0.1
```

待ち受けを再度確認します。サービスを再起動した後、ローカルホストでのみ待ち受けています。

```
netstat -an | grep 3306 | grep LIST
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
```

MySQL パスワード

特権ユーザで MySQL コンソールに接続します。

```
mysql -h host -u root -p
```

パスワードが安全であり、パスワードの入力を求められていることを確認します。 そうでない場合は、次のコマンドで設定します。

```
mysqladmin password
```

このセキュリティ対策は、外部からの攻撃だけでなく、内部ユーザによる誤用からもデータベースを保護するために不可欠です。

Apache web サーバの保護

/etc/httpd/conf/httpd.conf ファイルに次の行を追加して、サーバ情報ヘッダの Apache と OS バージョンを非表示にします。

```
ServerTokens Prod
```

システムサービスの最小化

この手法は非常に網羅的であり、システムに不要なものをすべて排除するだけです。 このようにして、実際には必要のない、誤って設定されたアプリケーションで将来発生する可能性のある問題を回避します。 この方法へのアプローチを単純化するために、マシン上にポートを開いて待ち受けているアプリケーションのみを対象とします。このために、次のコマンドを実行します。

```
netstat -tulpn
```

次のように、リスニングポートごとに結果が返されます。ただし、環境によって異なります。

```
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN
996/master
tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN
```

```
75171/httpd
tcp        0      0 0.0.0.0:31122           0.0.0.0:*               LISTEN
872/sshd
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
75097/mysqld
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
75171/httpd
tcp        0      0 0.0.0.0:6099            0.0.0.0:*               LISTEN
7721/Xvfb
tcp6       0      0 :::4444                 :::*                    LISTEN
7726/java
tcp6       0      0 :::34599                :::*                    LISTEN
7726/java
tcp6       0      0 :::6099                 :::*                    LISTEN
7721/Xvfb
```

IPv6 を無効にした場合、tcp6 の行は、sysctl を使用してシステムに変更を加えた後、再起動せずに残されたサービスでない限り、表示されないはずです。

各ポートを調査し、その背後にあるアプリケーションを知る必要があります。 この場合、443、80 は http サービスからのもののようですが、どのシステムプロセスが各ポートを使用しているかを確実に分析します。 これを行うには、lsof コマンドを使用します。これは、デフォルトではインストールされないため、yum とともにインストールする必要があります。

localhost (127.0.0.1) でリッスンするサービスは、すべての IP アドレス (0.0.0.0) および一部のサービス(オープンでリッスンしている場合)をリッスンするサービスよりも安全です。 、ローカルホストのみをリッスンするようにセキュリティで保護する必要があります。 この画面例では、たとえば MySQL(3306)に対して実行されています。

たとえば、メインの postfix プロセスが実行されていることがわかります。 このサービスは必要ないため、システムからアンインストールします。

```
yum remove postfix
```

疑わしい各プロセスの PID を調査することにより(前述のステップを参照)、そのポートにあるプロセスを確認します。

```
ps aux | grep 7726 root 7726 0.1 8.5 3258724 248608 ? Sl Mar09 60:01
/usr/bin/java -jar /usr/lib/pwr/selenium-server-standalone-2.53.1.jar -host
185.247.117.28 -port 4444 -firefoxProfileTemplate /opt/firefox_profile root
79041 0.0 0.0 112716 960 pts/4 S+ 11:54 0:00 grep --color=auto 7726
```

また、そのサービスを使用していない場合は、削除できます。

プロセスの“ 調査 ”のためのこの処理は、時間をかけて徹底的かつ反復的に行う必要があります。Pandora FMS プロセスインベントリシステムを使用して、時間の経過とともに新しいプロセスが開始されていないことを確認する必要があります。 サーバの待ち受けポートは、セキュリティの観

点から非常に重要なものであり、建物の正面にある窓のようなものです。 私たちはそれが閉じていて安全であると信じているかもしれませんが、窓は常に権限のある人や、やる気のある侵入者の入り口となります。

追加設定

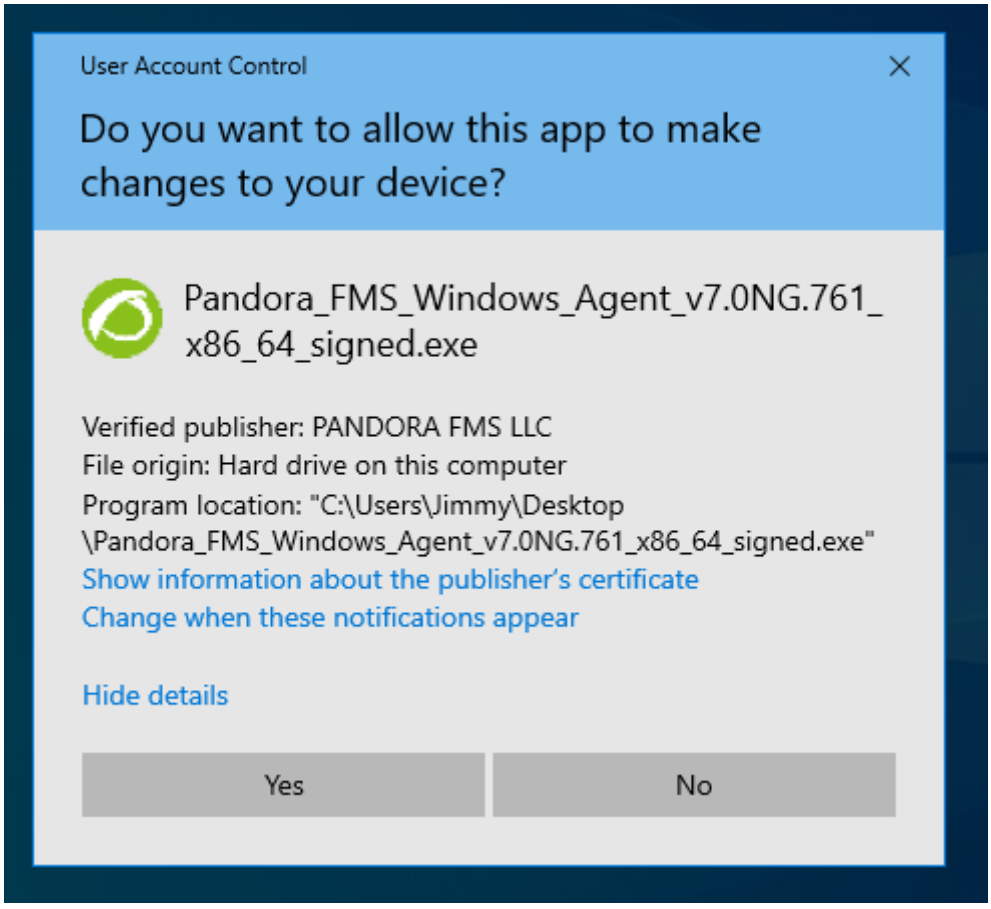
NTP 時刻同期

システムの時刻同期を設定することをお勧めします。

```
yum install ntpdate
echo "ntpdate 0.us.pool.ntp.org"> /etc/cron.daily/ntp
chmod 755 /etc/cron.daily/ntp
```

ローカル監視

システムには、Pandora FMS ソフトウエアエージェントをインストールして起動します。MS Windows® オペレーティングシステムの場合、バージョン 761 以降、インストール実行ファイルは署名されています。



標準チェックに加えて、次のアクティブチェックをお勧めします。

- セキュリティプラグイン
- 完全なシステムインベントリ(特にユーザーとインストール済みパッケージ)。
- システムおよびセキュリティログの収集

```
module_plugin grep_log_module /var/log/messages Syslog \.\*
module_plugin grep_log_module /var/log/secure Secure \.\*
```

ソフトウェアエージェントをインストールしたら、少なくとも次の情報をエージェントタブで手動で定義する必要があります。

- 説明
- IP アドレス(複数ある場合は全て設定します)
- グループ
- 部門、責任者および物理的な場所(カスタムフィールド)

GNU/Linux におけるセキュリティ監視

公式プラグインを使用すると、実行のたびに、ほぼリアルタイムでエージェントのセキュリティをプロアクティブに監視でき、関連するイベントの警告を発することができます。

このプラグインは、最新の GNU/Linux コンピュータでのみ実行することを目的としています。 64ビットと 32ビットで実行できるように準備されています。

これには、32および 64bit 版の John the ripper 1.8 + Contrib パッチのカスタムビルドが含まれています。プラグインの主な概要は、モノリシックであり、強化ポイントを検出し、管理者に何も尋ねることなくディストリビューション間の違いを解決しようとすることです。そのため、バージョン、ディストリビューション、またはアーキテクチャに関係なく、どのシステムでも同じように展開することができます。

このプラグインは以下をチェックします。

- 最も一般的な500個のパスワードを含む辞書(あらかじめ提供)を使用したユーザパスワードの監査チェック。 通常、これには数秒しかかかりません。 数百人のユーザがいる場合は、プラグインの実行を 2〜6 時間ごとにのみ実行するようにカスタマイズする必要があります。 組織の一般的なパスワードを “basic_security/password-list” ファイルに追加するだけで、パスワード辞書をカスタマイズできます。
- SSH がデフォルトのポートで実行されていないことの確認。
- FTB がデフォルトのポートで実行されていないことの確認。
- SSH が root アクセスを許可していないことの確認。
- MySQL が root パスワード無しで実行されているかどうかの確認。
- その他チェック。

Pandora FMS ドキュメント一覧に戻る