Arquitectura de seguridad

Arquitectura de seguridad

Arquitectura de seguridad

Se describen los elementos de seguridad de cada componente de Pandora FMS, acordes a normativas como PCI/DSS, ISO 27001, ENS, LOPD y similares.

Además de una descripción específica de los mecanismos de seguridad de cada elemento de Pandora FMS, se incide en los posibles riesgos y la manera de mitigarlos, utilizando las herramientas de que dispone Pandora FMS u otros mecanismos posibles.

Implementación general de seguridad

- Los componentes de Pandora FMS tienen documentados sus puertos de entrada y salida, por lo que es posible asegurar mediante *firewalls* todos los accesos hacia y desde sus componentes.
- Tráfico seguro mediante cifrado y certificados: Pandora FMS, a todos los niveles (operación de usuario, comunicación entre componentes), soporta cifrado SSL/TLS y certificados en ambos extremos.
- Sistema de doble autenticación de acceso: Se puede implantar un sistema de doble autenticación. El primero, a nivel de acceso (HTTPS) integrado con cualquier sistema de token Opensource o comercial.
- Sistema de autenticación con terceros: A nivel de aplicación es gestionada por Pandora FMS, que se puede autenticar contra LDAP o Active Directory.
- SSO (Single Sign-On), con SAML.
- Políticas de seguridad en la gestión de usuarios: La gestión de usuarios está delimitada por políticas tanto a nivel de perfiles de usuario como a nivel de perfiles de visibilidad de operaciones, definido como el sistema de ACL Extendido de la versión Enterprise.
- Posibilidad de auditorías sobre las acciones de los elementos monitorizados: Pandora FMS, en su versión Enterprise, audita todas las acciones de los usuarios, incluyendo la información sobre los campos alterados o borrados. Además, incluye un sistema de validación por firma de estos registros.
- Cesión de datos de auditoría a gestores de logs externos: Los *logs* de auditoría están disponibles para su exportación mediante SQL y permiten integrarlos en una tercera fuente para mayor seguridad, en tiempo cercano al real.
- Separación física de los componentes: Que ofrecen una interfaz al usuario y los contenedores de información (*filesystem*). Tanto los datos almacenados en BBDD como los *filesystem* que almacenan la información de configuración de monitorización pueden estar en máquinas físicas separadas, en distintas redes, protegidas a través de sistemas perimetrales.
- Política activa de contraseñas: Que permiten forzar una política estricta de gestión de contraseñas de acceso a los usuarios de la aplicación (Consola web).
- Cifrado de datos sensibles: El sistema permite guardar de manera cifrada y segura los datos más sensibles, tales como credenciales de acceso.

Seguridad por componentes de la arquitectura

Servidor

- El servidor necesita permisos de root, pero puede ser instalado (con limitaciones) con permisos no root (solo en sistemas GNU/Linux).
- El servidor necesita acceso directo (lectura y escritura) a los ficheros de configuración remota de los agentes, que se propagan cuando los agentes contactan con el servidor, de manera periódica. Dichos ficheros están protegidos por el *filesystem*, con permisos estándar.
- El servidor como tal no escucha ningún puerto. Es el servidor de Tentacle quien escucha en un puerto, el servidor solo accede a los ficheros que este deja en disco.
- El servidor tiene su propio registro de eventos (*log*), muy detallado.
- El servidor se conecta con la base de datos (BBDD) principal mediante una conexión estándar MySQL/TCP.
- Parte del código es accesible (OpenSource) y el de la versión Enterprise se puede solicitar bajo condiciones específicas de contrato (solo para clientes).

Posibles vulnerabilidades y salvaguardas

- Acceso no autorizado a los archivos de configuración de los agentes. *Solución*:
  1. Implementar un contenedor asegurado externo para los archivos de configuración externos, vía NFS.
- Inserción de comandos en los agentes remotos a través de la manipulación de los ficheros de configuración almacenados en el contenedor de configuraciones. *Solución*:
  1. Deshabilitar configuración remota en los agentes especialmente sensibles después de la configuración y dejarlos funcionando sin poder alterar nada remotamente, para una seguridad absoluta.
  2. Monitorización remota -sin agentes- de los dispositivos más delicados.
- Vulnerabilidad ante ataques de falseo de información, simulando agentes que no tenemos en el sistema o suplantando su identidad. Para evitarlo *se pueden utilizar varios mecanismos*:
  1. Mecanismo de protección de contraseñas (que funciona por grupo).
  2. Limitando la autocreación de agentes, y crearlos de manera manual.
  3. Limitando la capacidad de autodetectar cambios en el agente y no tomar información nueva desde los XML a la ya existente.
- Captura maliciosa de la comunicación entre servidor y consola (captura de tráfico de red). *Solución*:
  1. Activar la comunicación TLS entre servidor y base de datos MySQL.

Tentacle

- Tentacle es un servicio oficial de Internet, documentado como tal por IANA. Esto significa que puede ser protegido fácilmente con cualquier herramienta de seguridad perimetral.
- No necesita root ni privilegios especiales.
- Tiene cuatro niveles de seguridad: Sin cifrado (por defecto), SSL/TLS Básico, SSL/TLS con certificado en ambos extremos, y SSL/TLS con certificado y validación de CA.
- Específicamente diseñado para no dar pistas a posibles intrusos en los mensajes de error y con períodos de expiración de tiempo (*timeouts*) específicos para desalentar ataques de fuerza bruta.
- Tiene un *log* de auditoría propio.
- El 100% del código es accesible (bajo licencia Opensource GPL2).

Posibles vulnerabilidades y salvaguardas

- Ataques al *filesystem*. Debe acceder al contenedor de configuraciones. *Solución*:
  1. Se protege de la misma manera que el servidor, mediante un sistema NFS externo asegurado.
- Ataques DoS por sobrecarga. *Soluciones*:
  1. Montar una solución de HA sobre el servicio TCP que ofrece para balanceo, o un clúster activo/activo. Vale cualquier solución hardware o software disponible al tratarse de un servicio TCP estándar.

Consola web

- No necesita root, se instala con un usuario sin privilegios.
- Debe tener acceso al repositorio de configuraciones de agentes (*filesystem*).
- Escucha en puertos estándar HTTP o HTTPS.
- Registra todas las peticiones vía registro de peticiones HTTP.
- Ofrece una API pública vía HTTP/HTTPS, asegurada con credenciales y lista de direcciones IP permitidas de antemano.
- Existe una auditoría específica de aplicación, que registra la actividad de cada usuario sobre cada objeto del sistema.
- Se puede restringir el acceso de cada usuario a cualquier sección de la aplicación, y se pueden crear incluso administradores con permisos restringidos.
- La aplicación incorpora un sistema de doble autenticación.
- La aplicación incorpora un sistema de autenticación delegada (LDAP, AD).
- Se puede montar un sistema de solo lectura. Sin acceso a las configuraciones de dispositivos.
- La información confidencial (contraseñas) se puede guardar cifrada en la BBDD.
- La aplicación se conecta con la BBDD principal mediante una conexión estándar MySQL/TCP.
- Parte del código es accesible (OpenSource) y el de la versión Enterprise se puede solicitar bajo condiciones específicas de contrato (solo para clientes).
- Existe una implementación fuerte de políticas de seguridad respecto a las contraseñas (longitud, cambio forzado, histórico, tipo de caracteres validos, etc.).

Posibles vulnerabilidades y salvaguardas

- Ataques al *filesystem*. Debe acceder al contenedor de configuraciones. *Solución*:
  1. Se protege de la misma manera que el servidor, mediante un sistema NFS externo asegurado.
- Ataques de fuerza bruta o diccionario contra la autenticación de usuario. *Soluciones*:
  1. Implementar una política de contraseñas complejas.
  2. Implementar un mecanismo de doble autenticación.
- Captura de tráfico (*eavesdropping*) del tráfico a la consola. *Solución*:
  1. Implementar SSL/TLS.
- Captura de tráfico (*eavesdropping*) del tráfico a la base de datos. *Solución*:
  1. Implementar SSL/TLS.
- Ataques de tipo SQL injection para obtener información confidencial de la BBDD de la aplicación. *Solución*:
  1. Implementar almacenamiento cifrado de datos.
- Mal uso (intencionado o no intencionado) por parte de los usuarios de la aplicación. *Soluciones*:
  1. Activar el *log* de auditoría y mostrarle a los usuarios que existe y su precisión.
  2. Activar el sistema de ACL extendido para restringir al máximo las funciones de cada usuario.
  3. Exportar el *log* de auditoría a un sistema externo de forma regular.

- Ejecución de código malicioso en herramientas locales de la consola, reemplazando ficheros binarios. *Solución*:
    1. Fortalecimiento (*hardening*) del servidor que contiene la aplicación.

Agentes

- Puede ejecutarse sin permisos de superusuario (con funcionalidades limitadas).
- La gestión remota del agente se puede desactivar (en local y en remoto), de manera que se puede minimizar el impacto de una intrusión en el sistema central.
- El agente no escucha puertos de red, es él el que se conecta al servidor de Pandora FMS.
- Existe un registro de cada ejecución.
- Los ficheros de configuración están protegidos por defecto, mediante permisos del filesystem. Solo un usuario con permisos de superadministrador puede modificarlos.
- El 100% del código es accesible (bajo licencia Opensource GPL2).

Posibles vulnerabilidades y salvaguardas

- Intrusión en el sistema central que permita distribuir ejecución de comandos maliciosos a los agentes. *Soluciones*:
    1. Limitar qué usuarios pueden realizar esas modificaciones de políticas o configuraciones (via ACL ordinario de la consola o ACL extendido).
    2. Activar el modo de solo lectura (*readonly*) de los agentes (no permiten modificaciones de su configuración remotamente), para aquellos sistemas especialmente sensibles.
- Debilidad en el *filesystem* que permita modificar ficheros. *Solución*:
    1. Correcta configuración de permisos.
- Ejecución de *plugins* o comandos maliciosos. *Solución*:
    1. Limitar qué usuarios pueden subir ejecutables (vía ACL ordinario de la consola o ACL extendido).
    2. Realizar una auditoría de plugins *nuevos*.

Base de datos

- Es un producto estándar (MySQL).

Posibles vulnerabilidades y salvaguardas

- *Eavesdropping* (captura de tráfico de red). *Solución*:
    1. Implementación de una conexión segura TLS. MySQL la soporta.
- Permisos incorrectos. *Solución*:
    1. Configuración correcta de permisos de acceso.
- Vulnerabilidades conocidas de MySQL: Se debe establecer un plan de actualización del servidor MySQL en el que se pueda tener lo más actualizado posible el mismo, y de esta forma evitar posibles vulnerabilidades por tener versiones antiguas.

Aseguramiento del sistema base

El *hardening* de sistemas es un punto clave en la estrategia de seguridad global de una compañía.

Como fabricantes se emite una serie de recomendaciones para realizar una instalación segura de todos los componentes de Pandora FMS, basados en una plataforma estándar RHEL8 o Ubuntu server.

Estas mismas recomendaciones son válidas para cualquier otro sistema de monitorización basado en GNU/Linux.

Credenciales de acceso

Para acceder al sistema, se crearán usuarios nominativos de acceso, sin privilegios y con acceso restringido a las necesidades que tengan.

Idealmente se debería integrar la autenticación de cada usuario con un sistema de autenticación doble, basado en *token*. Existen alternativas gratuitas y seguras como Google Authenticator®integrables en GNU/Linux, aunque fuera del alcance de esta guía. Considere seriamente su uso.

Si es necesario crear otros usuarios para aplicaciones, deben ser usuarios sin acceso remoto (para ello, desactivar su *Shell* o método equivalente).

Acceso de superusuario

En el caso de que ciertos usuarios tengan que tener permisos de administrador se utilizará el comando sudo.

Sistema operativo actualizado

Bastará con estar conectado a internet o configurar el sistema dnf o apt para que utilice un servidor *proxy*.

Este comando puede ocasionar potenciales problemas de cambio de librerías, configuraciones, etcétera. Es importante actualizar el sistema operativo antes de poner el sistema en producción. Si está revisando un sistema de producción ya activo, quizás solamente se necesite actualizar los componentes críticos, por ejemplo aquellos que tengan una vulnerabilidad.

Por ejemplo para actualizar solamente MySQL en un sistema RHEL: `dnf update mysql-server`.

La actualización del sistema operativo es un proceso que debería ser periódico. Mediante el inventariado de paquetes del sistema, puede consultar versiones vulnerables y ejecutar actualizaciones de emergencia.

Auditoría de acceso

Es necesario tener activo el log de seguridad `/var/log/secure` y monitorizar esos *logs* con la monitorización.

Por defecto esto viene activado, si no es así, revisar el fichero `/etc/rsyslog.conf` o `/etc/syslog.conf` .

Se recomienda que se lleve los *logs* del sistema de auditoría y los recoja con un sistema externo de gestión de *logs*, Pandora FMS puede hacerlo fácilmente y le será útil para establecer alertas o revisarlos de manera centralizada en caso de necesidad.

Servidor SSH

El servidor SSH permite la conexión remota a sistemas GNU/Linux para la ejecución de comandos, por lo que se trata de un punto crítico y debe asegurarse prestando atención a los siguientes puntos (para ello edite el fichero `/etc/ssh/sshd_config` y posteriormente, reinicie el servicio).

- Modificar puerto por defecto (por ejemplo al 31122)

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

- Deshabilitar el inicio de sesión al superusuario root login

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

- Deshabilitar el reenvío de puertos port forwarding

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

- Deshabilitar tunneling

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

- Eliminar llaves SSH para acceso remoto de root. Asumiendo que solamente hay un usuario válido para acceso remoto (por ejemplo, pfms). Si hubiere otros, también habría que comprobarlos. Para ello, se revisa el contenido del fichero `/home/pfms/.ssh/authorized_keys` y comprobar de cuáles máquinas pertenecen. *Borrarlo si se cree que no debería haber ninguno*.
- Establecer un aviso de acceso remoto estándar que explique que el servidor es de acceso privado y que cualquier persona sin credenciales debe desconectar.

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

Servidor MySQL

Si el MySQL solamente da servicio a un elemento interno se debe verificar de que únicamente

escucha en *localhost*, utilice netstat:

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

En este caso de ejmplo se está escuchando para todo el mundo, edite el fichero `/etc/my.cnf` y en la sección `[mysqld]` añada la siguiente línea:

```
bind-address = 127.0.0.1
```

Tras reiniciar el servicio vuelva a comprobar el puerto de escucha.

Contraseña MySQL

Conectar a la consola de MySQL con un usuario con privilegios:

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

Verificar que la contraseña es compleja y que ha solicitado contraseña. De no ser así, se establece con el comando:

```
mysqladmin password
```

Esta medida de seguridad es imprescindible para proteger bases de datos no solamente ante ataques externos sino ante usos incorrectos por parte de usuarios internos.

Servidor web Apache

Añada la siguiente línea al fichero `/etc/httpd/conf/httpd.conf` para ocultar la versión del servidor web (Apache, Nginx) y del sistema operativo (RHEL, Ubuntu server) en las cabeceras de información del servidor.

```
ServerTokens Prod
```

Motor de aplicaciones PHP

Para *securizar* bien el motor de aplicaciones sobre el que funciona Pandora FMS puede ser necesario, en algunos entornos especialmente sensibles con la seguridad, *securizar* el acceso a la aplicación para que las *cookies* de sesión solo sean transmitidas con SSL.

Para ello, en el fichero `php.ini` incluir los siguientes *tokens* de configuración:

```
session.cookie_httponly = 1
```

```
session.cookie_secure = 1
```

Esto hará que la aplicación no funcione cuando se usa sobre HTTP (sin cifrado).

Minimizar servicios en el sistema

Esta técnica, que puede ser muy exhaustiva, consiste en eliminar todo lo que no sea necesario en el sistema. Así se evitan posibles problemas en un futuro con aplicaciones mal configuradas que realmente no necesitan. Para simplificar la aproximación a esta práctica, vamos a considerar únicamente aquellas aplicaciones que tienen un puerto abierto en la máquina, para ello, ejecutar `netstat -tulpn`.

Se debería investigar cada puerto y conocer la aplicación que hay detrás. Para ello se puede usar el comando lsof, que habrá que instalar con dnf o apt porque no viene por defecto instalado.

Aquellos servicios que escuchan en localhost (`127.0.0.1`) son más seguros que los que escuchan a todas las direcciones IP (`0.0.0.0`) y algunos de ellos, si están escuchando en puerto abierto, se debería intentar corregirlos para que escuchen solamente a localhost.

Mediante el sistema de inventario de procesos de Pandora FMS, se deben verificar que no se inicien nuevos procesos a lo largo del tiempo.

Configuración adicional

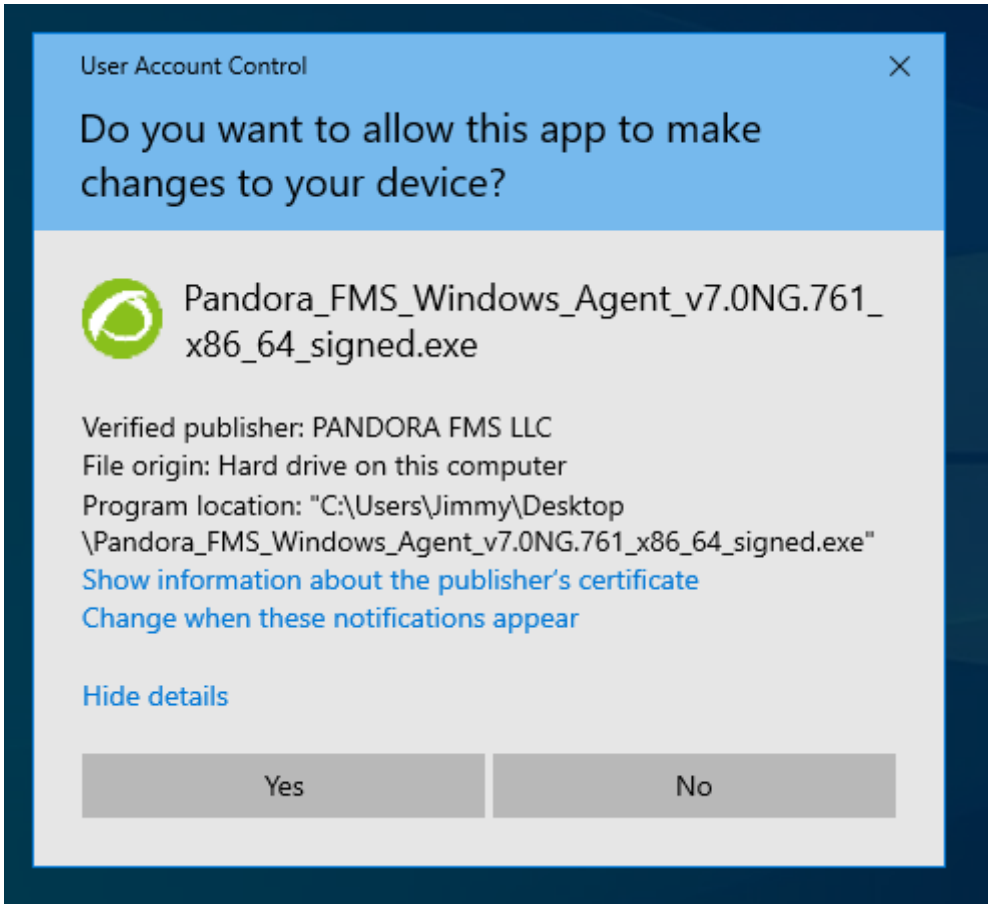
Sincronización de tiempos NTP

Se recomienda configurar la sincronización horaria del sistema, en un sistema RHEL:

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

Monitorización local

El sistema debería tener un Agente Software Pandora FMS instalado y lanzado contra nuestro servidor. Para el sistema operativo MS Windows®, a partir de la versión 761, los ejecutables de instalación están firmados.



Se recomiendan los siguientes chequeos activos además de los chequeos estándar:

- Complemento (*plugin*) de seguridad activo.
- Inventario completo del sistema (especialmente usuarios y paquetes instalados).
- Recogida de *logs* del sistema y seguridad:

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

Una vez instalado el Agente Software, habrá que definir manualmente al menos la siguiente información en la ficha de agente:

- Descripción.
- Dirección IP (si tiene varias, poner todas).
- Grupo.
- Departamento, responsable y ubicación física (*custom fields*).

Monitorización de la seguridad en GNU/Linux

El plugin oficial permite monitorizar de forma proactiva la seguridad en el agente, en cada ejecución, casi en tiempo real, ofreciendo algunos chequeos que pueden alertarnos de algunos sucesos relevantes de manera proactiva.

Este plugin está pensado para funcionar solamente en equipos GNU/Linux modernos. Está

preparado para funcionar en 64 y 32 bits.

Contiene una compilación personalizada de John the ripper 1.8 + parches Contrib con binarios estáticos de 32 y 64. El concepto principal del *plugin* es ser monolítico, detectar lo que puede ser reforzado y tratar de resolver las diferencias entre distros sin preguntar nada al administrador, por lo que el despliegue podría ser el mismo para cualquier sistema, ignorando versiones, distro o arquitectura.

Este *plugin* comprobará:

- Comprobación de auditoría de contraseñas de usuario, usando el diccionario (proporcionado) con las 500 contraseñas más comunes. Esto no suele llevar más de unos segundos. Si tiene cientos de usuarios, probablemente necesite personalizar la ejecución del plugin para que se ejecute sólo cada 2-6 horas. Puede personalizar el diccionario de contraseñas simplemente añadiendo la contraseña típica de su organización en el archivo “basic_security/password-list”.
- Compruebe que SSH no se ejecute el puerto por defecto.
- Compruebe que el FTP no se ejecuta en el puerto por defecto.
- Compruebe que SSH no permita el acceso de root.
- Verifique si hay un MySQL corriendo sin la contraseña de root definida.
- Otros chequeos.

Volver al Índice de Documentación Pandora FMS