Red Hat Enterprise Linux 8

Endurecimiento de la seguridad

Seguridad de Red Hat Enterprise Linux 8

Last Updated: 2021-02-27

Red Hat Enterprise Linux 8 Endurecimiento de la seguridad

Seguridad de Red Hat Enterprise Linux 8

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

Legal Notice

Resumen

Este título ayuda a los usuarios y a los administradores a aprender los procesos y las prácticas para asegurar las estaciones de trabajo y los servidores contra la intrusión local y remota, la explotación y la actividad maliciosa. Centrado en Red Hat Enterprise Linux pero detallando conceptos y técnicas válidas para todos los sistemas Linux, esta guía detalla la planificación y las herramientas necesarias para crear un entorno informático seguro para el centro de datos, el lugar de trabajo y el hogar. Con los conocimientos administrativos, la vigilancia y las herramientas adecuadas, los sistemas que ejecutan Linux pueden ser totalmente funcionales y estar protegidos de los métodos de intrusión y explotación más comunes.

Table of Contents

HACER QUE EL CÓDIGO ABIERTO SEA MÁS INCLUSIVO

Red Hat se compromete a sustituir el lenguaje problemático en nuestro código, documentación y propiedades web. Estamos empezando con estos cuatro términos: maestro, esclavo, lista negra y lista blanca. Debido a la enormidad de este esfuerzo, estos cambios se implementarán gradualmente a lo largo de varias versiones próximas. Para más detalles, consulte el mensaje de nuestro CTO Chris Wright .

PROPORCIONAR COMENTARIOS SOBRE LA DOCUMENTACIÓN DE RED HAT

Agradecemos su opinión sobre nuestra documentación. Por favor, díganos cómo podemos mejorarla. Para ello:

- Para comentarios sencillos sobre pasajes concretos:
  1. Asegúrese de que está viendo la documentación en el formato *Multi-page HTML*. Además, asegúrese de ver el botón **Feedback** en la esquina superior derecha del documento.
  2. Utilice el cursor del ratón para resaltar la parte del texto que desea comentar.
  3. Haga clic en la ventana emergente **Add Feedback** que aparece debajo del texto resaltado.
  4. Siga las instrucciones mostradas.
- Para enviar comentarios más complejos, cree un ticket de Bugzilla:
  1. Vaya al sitio web de Bugzilla.
  2. Como componente, utilice **Documentation**.
  3. Rellene el campo **Description** con su sugerencia de mejora. Incluya un enlace a la(s) parte(s) pertinente(s) de la documentación.
  4. Haga clic en **Submit Bug**.

CAPÍTULO 1. VISIÓN GENERAL DEL ENDURECIMIENTO DE LA SEGURIDAD EN RHEL

Debido a la creciente dependencia de los potentes ordenadores conectados en red para ayudar a dirigir las empresas y controlar nuestra información personal, se han formado industrias enteras en torno a la práctica de la seguridad de las redes y los ordenadores. Las empresas han solicitado los conocimientos y habilidades de los expertos en seguridad para auditar adecuadamente los sistemas y adaptar las soluciones a los requisitos de funcionamiento de su organización. Dado que la mayoría de las organizaciones son cada vez más dinámicas, sus trabajadores acceden a los recursos informáticos críticos de la empresa de forma local y remota, por lo que la necesidad de contar con entornos informáticos seguros se ha acentuado.

Por desgracia, muchas organizaciones, así como los usuarios individuales, consideran la seguridad más bien como una idea tardía, un proceso que se pasa por alto en favor de una mayor potencia, productividad, comodidad, facilidad de uso y preocupaciones presupuestarias. La implementación adecuada de la seguridad a menudo se lleva a cabo postmortem, cuando ya se ha producido una intrusión no autorizada. Tomar las medidas correctas antes de conectar un sitio a una red no fiable, como Internet, es un medio eficaz para frustrar muchos intentos de intrusión.

1.1. ¿QUÉ ES LA SEGURIDAD INFORMÁTICA?

La seguridad informática es un término general que abarca un amplio ámbito de la informática y el procesamiento de la información. Las industrias que dependen de los sistemas y redes informáticos para llevar a cabo las transacciones comerciales diarias y acceder a información crítica consideran sus datos como una parte importante de sus activos generales. Varios términos y métricas han entrado en nuestro vocabulario empresarial diario, como el coste total de propiedad (TCO), el rendimiento de la inversión (ROI) y la calidad del servicio (QoS). Utilizando estas métricas, las industrias pueden calcular aspectos como la integridad de los datos y la alta disponibilidad (HA) como parte de sus costes de planificación y gestión de procesos. En algunos sectores, como el del comercio electrónico, la disponibilidad y fiabilidad de los datos puede significar la diferencia entre el éxito y el fracaso.

1.2. NORMALIZACIÓN DE LA SEGURIDAD

Las empresas de todos los sectores se basan en normas y reglas establecidas por organismos de normalización como la Asociación Médica Americana (AMA) o el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE). Los mismos conceptos son válidos para la seguridad de la información. Muchos consultores y proveedores de seguridad están de acuerdo con el modelo de seguridad estándar conocido como CIA, o *Confidentiality, Integrity, and Availability*. Este modelo de tres niveles es un componente generalmente aceptado para evaluar los riesgos de la información sensible y establecer una política de seguridad. A continuación se describe con más detalle el modelo CIA:

- Confidencialidad
- Integridad
- Disponibilidad

1.3. SOFTWARE CRIPTOGRÁFICO Y CERTIFICACIONES

Red Hat Enterprise Linux se somete a varias certificaciones de seguridad, como **FIPS 140-2** o **Common Criteria** (CC), para garantizar que se siguen las mejores prácticas del sector.

El artículo **RHEL 8 core crypto components** El artículo de la base de conocimientos proporciona una visión general de los componentes criptográficos del núcleo de Red Hat Enterprise Linux 8,

documentando cuáles son, cómo se seleccionan, cómo se integran en el sistema operativo, cómo soportan los módulos de seguridad de hardware y las tarjetas inteligentes, y cómo se aplican las certificaciones criptográficas a ellos.

1.4. CONTROLES DE SEGURIDAD

La seguridad informática suele dividirse en tres categorías principales distintas, comúnmente denominadas **controls**:

- Físico
- Técnica
- Administrativo

Estas tres grandes categorías definen los principales objetivos de una correcta implementación de la seguridad. Dentro de estos controles hay subcategorías que detallan más los controles y la forma de aplicarlos.

1.4.1. Controles físicos

El control físico es la aplicación de medidas de seguridad en una estructura definida que se utiliza para disuadir o impedir el acceso no autorizado a material sensible. Ejemplos de controles físicos son:

- Cámaras de vigilancia en circuito cerrado
- Sistemas de alarma térmica o de movimiento
- Guardias de seguridad
- Identificaciones con foto
- Puertas de acero con cerradura y cerrojo
- Biometría (incluye las huellas dactilares, la voz, el rostro, el iris, la escritura a mano y otros métodos automatizados utilizados para reconocer a las personas)

1.4.2. Controles técnicos

Los controles técnicos utilizan la tecnología como base para controlar el acceso y la utilización de datos sensibles en toda una estructura física y en una red. Los controles técnicos tienen un gran alcance y abarcan tecnologías como:

- Codificación
- Tarjetas inteligentes
- Autenticación en red
- Listas de control de acceso (ACL)
- Software de auditoría de la integridad de los archivos

1.4.3. Controles administrativos

Los controles administrativos definen los factores humanos de la seguridad. Implican a todos los niveles del personal de una organización y determinan qué usuarios tienen acceso a qué recursos e información por medios como:

- Formación y sensibilización
- Planes de preparación y recuperación en caso de catástrofe
- Estrategias de contratación y separación de personal
- Registro y contabilidad del personal

1.5. EVALUACIÓN DE LA VULNERABILIDAD

Con tiempo, recursos y motivación, un atacante puede entrar en casi cualquier sistema. Todos los procedimientos y tecnologías de seguridad disponibles en la actualidad no pueden garantizar que ningún sistema esté completamente a salvo de las intrusiones. Los routers ayudan a asegurar las puertas de entrada a Internet. Los cortafuegos ayudan a asegurar el borde de la red. Las redes privadas virtuales transmiten los datos de forma segura en un flujo cifrado. Los sistemas de detección de intrusos avisan de las actividades maliciosas. Sin embargo, el éxito de cada una de estas tecnologías depende de una serie de variables, entre ellas:

- La experiencia del personal responsable de configurar, supervisar y mantener las tecnologías.
- La capacidad de parchear y actualizar servicios y núcleos de forma rápida y eficaz.
- La capacidad de los responsables de mantener una vigilancia constante sobre la red.

Dado el estado dinámico de los sistemas de datos y las tecnologías, asegurar los recursos corporativos puede ser bastante complejo. Debido a esta complejidad, a menudo es difícil encontrar recursos expertos para todos sus sistemas. Aunque es posible tener personal con conocimientos en muchas áreas de la seguridad de la información a un alto nivel, es difícil retener al personal que es experto en más de unas pocas áreas temáticas. Esto se debe principalmente a que cada área temática de la seguridad de la información requiere una atención y un enfoque constantes. La seguridad de la información no se detiene.

Una evaluación de la vulnerabilidad es una auditoría interna de la seguridad de su red y sistema, cuyos resultados indican la confidencialidad, integridad y disponibilidad de su red. Normalmente, la evaluación de la vulnerabilidad comienza con una fase de reconocimiento, durante la cual se recopilan datos importantes sobre los sistemas y recursos objetivo. Esta fase conduce a la fase de preparación del sistema, en la que se comprueba esencialmente que el objetivo no presenta ninguna vulnerabilidad conocida. La fase de preparación culmina con la fase de informe, en la que los hallazgos se clasifican en categorías de riesgo alto, medio y bajo; y se discuten los métodos para mejorar la seguridad (o mitigar el riesgo de vulnerabilidad) del objetivo

Si tuviera que realizar una evaluación de la vulnerabilidad de su hogar, probablemente comprobaría cada puerta de su casa para ver si están cerradas y bloqueadas. También comprobaría todas las ventanas, asegurándose de que se cierran por completo y de que se cierran correctamente. Este mismo concepto se aplica a los sistemas, redes y datos electrónicos. Los usuarios malintencionados son los ladrones y vándalos de sus datos. Concéntrese en sus herramientas, su mentalidad y sus motivaciones, y así podrá reaccionar rápidamente ante sus acciones.

1.5.1. Definir la evaluación y las pruebas

Las evaluaciones de la vulnerabilidad pueden ser de dos tipos: *outside looking in* y *inside looking around*.

Cuando se realiza una evaluación de la vulnerabilidad desde fuera, se intenta comprometer los sistemas desde el exterior. Ser externo a su empresa le proporciona el punto de vista del cracker. Usted ve lo que un cracker ve

Cuando realizas una evaluación de vulnerabilidad desde dentro, estás en ventaja ya que eres interno y tu estatus se eleva a confianza. Este es el punto de vista que usted y sus compañeros de trabajo tienen una vez que se conectan a sus sistemas. Usted ve los servidores de impresión, los servidores de archivos, las bases de datos y otros recursos.

Hay diferencias notables entre los dos tipos de evaluaciones de vulnerabilidad. El hecho de ser interno en su empresa le da más privilegios que a una persona externa. En la mayoría de las organizaciones, la seguridad está configurada para mantener a los intrusos fuera. Se hace muy poco para asegurar las partes internas de la organización (como cortafuegos departamentales, controles de acceso a nivel de usuario y procedimientos de autenticación para los recursos internos). Normalmente, hay muchos más recursos cuando se mira en el interior, ya que la mayoría de los sistemas son internos de una empresa. Una vez que se encuentra fuera de la empresa, su situación no es de confianza. Los sistemas y recursos disponibles en el exterior suelen ser muy limitados.

Considere la diferencia entre las evaluaciones de vulnerabilidad y *penetration tests*. Piensa en una evaluación de vulnerabilidad como el primer paso para una prueba de penetración. La información obtenida en la evaluación se utiliza para las pruebas. Mientras que la evaluación se lleva a cabo para comprobar los agujeros y las posibles vulnerabilidades, las pruebas de penetración realmente intentan explotar los hallazgos.

La evaluación de la infraestructura de red es un proceso dinámico. La seguridad, tanto informática como física, es dinámica. La realización de una evaluación muestra una visión general, que puede dar lugar a falsos positivos y falsos negativos. Un falso positivo es un resultado en el que la herramienta encuentra vulnerabilidades que en realidad no existen. Un falso negativo es cuando omite vulnerabilidades reales.

Los administradores de seguridad son tan buenos como las herramientas que utilizan y los conocimientos que conservan. Tome cualquiera de las herramientas de evaluación disponibles en la actualidad, ejecútelas contra su sistema y es casi una garantía de que hay algunos falsos positivos. Ya sea por fallo del programa o por error del usuario, el resultado es el mismo. La herramienta puede encontrar falsos positivos o, peor aún, falsos negativos.

Ahora que la diferencia entre una evaluación de la vulnerabilidad y una prueba de penetración está definida, tome las conclusiones de la evaluación y revíselas cuidadosamente antes de realizar una prueba de penetración como parte de su nuevo enfoque de mejores prácticas.

AVISO

No intente explotar las vulnerabilidades en los sistemas de producción. Hacerlo puede tener efectos adversos en la productividad y eficiencia de sus sistemas y red.

La siguiente lista examina algunas de las ventajas de realizar evaluaciones de vulnerabilidad.

- Crea un enfoque proactivo en la seguridad de la información.
- Encuentra potenciales exploits antes de que los crackers los encuentren.
- Permite mantener los sistemas actualizados y con parches.

- Promueve el crecimiento y ayuda a desarrollar la experiencia del personal.
- Reduce las pérdidas financieras y la publicidad negativa.

1.5.2. Establecer una metodología para la evaluación de la vulnerabilidad

Para ayudar en la selección de herramientas para una evaluación de la vulnerabilidad, es útil establecer una metodología de evaluación de la vulnerabilidad. Desafortunadamente, no existe una metodología predefinida o aprobada por la industria en este momento; sin embargo, el sentido común y las mejores prácticas pueden actuar como una guía suficiente.

What is the target? Are we looking at one server, or are we looking at our entire network and everything within the network? Are we external or internal to the company? Las respuestas a estas preguntas son importantes, ya que ayudan a determinar no sólo qué herramientas seleccionar, sino también la forma de utilizarlas.

Para saber más sobre el establecimiento de metodologías, consulte el siguiente sitio web:

- https://www.owasp.org/

1.5.3. Herramientas de evaluación de la vulnerabilidad

Una evaluación puede comenzar utilizando algún tipo de herramienta de recopilación de información. Cuando se evalúa toda la red, primero hay que mapear la distribución para encontrar los hosts que están funcionando. Una vez localizado, examine cada host individualmente. Centrarse en estos hosts requiere otro conjunto de herramientas. Saber qué herramientas utilizar puede ser el paso más crucial para encontrar vulnerabilidades.

Las siguientes herramientas son sólo una pequeña muestra de las disponibles:

- **Nmap** es una popular herramienta que puede utilizarse para encontrar sistemas anfitriones y abrir puertos en esos sistemas. Para instalar **Nmap** desde el repositorio **AppStream**, introduzca el comando **yum install nmap** como usuario **root**. Consulte la página de manual **nmap(1)** para obtener más información.
- Las herramientas del conjunto **OpenSCAP**, como la utilidad de línea de comandos **oscap** y la utilidad gráfica **scap-workbench**, proporcionan una auditoría de cumplimiento totalmente automatizada. Para obtener más información, consulte la sección sobre la comprobación del cumplimiento de la seguridad y las vulnerabilidades del sistema.
- Advanced Intrusion Detection Environment (**AIDE**) es una utilidad que crea una base de datos de archivos en el sistema, y luego utiliza esa base de datos para asegurar la integridad de los archivos y detectar intrusiones en el sistema. Consulte Comprobación de la integridad con AIDE para obtener más información.

1.6. AMENAZAS A LA SEGURIDAD

1.6.1. Amenazas a la seguridad de la red

Las malas prácticas a la hora de configurar los siguientes aspectos de una red pueden aumentar el riesgo de un ataque.

Arquitecturas inseguras

Una red mal configurada es un punto de entrada principal para usuarios no autorizados. Dejar una red local abierta y basada en la confianza, vulnerable a la altamente insegura Internet, es como dejar una puerta entreabierta en un vecindario plagado de delincuentes

Redes de difusión

Los administradores de sistemas a menudo no se dan cuenta de la importancia del hardware de red en sus esquemas de seguridad. El hardware simple, como los concentradores y routers, se basa en el principio de difusión o no conmutación; es decir, cada vez que un nodo transmite datos a través de la red a un nodo receptor, el concentrador o router envía una difusión de los paquetes de datos hasta que el nodo receptor recibe y procesa los datos. Este método es el más vulnerable a la suplantación de direcciones del protocolo de resolución de direcciones (*ARP*) o del control de acceso a los medios (*MAC*) tanto por parte de intrusos externos como de usuarios no autorizados en hosts locales.

Servidores centralizados

Otro posible escollo de la red es el uso de la informática centralizada. Una medida común de reducción de costes para muchas empresas es consolidar todos los servicios en una sola máquina potente. Esto puede ser conveniente, ya que es más fácil de gestionar y cuesta mucho menos que las configuraciones de varios servidores. Sin embargo, un servidor centralizado introduce un único punto de fallo en la red. Si el servidor central se ve comprometido, puede hacer que la red sea completamente inútil o, peor aún, propensa a la manipulación o el robo de datos. En estas situaciones, un servidor central se convierte en una puerta abierta que permite el acceso a toda la red.

1.6.2. Amenazas a la seguridad de los servidores

La seguridad de los servidores es tan importante como la seguridad de la red, ya que los servidores suelen contener gran parte de la información vital de una organización. Si un servidor se ve comprometido, todo su contenido puede estar disponible para que el cracker lo robe o manipule a voluntad. Las siguientes secciones detallan algunos de los principales problemas.

Servicios no utilizados y puertos abiertos

Una instalación completa de Red Hat Enterprise Linux 8 contiene más de 1000 aplicaciones y paquetes de biblioteca. Sin embargo, la mayoría de los administradores de servidores no optan por instalar todos los paquetes de la distribución, sino que prefieren instalar una instalación base de paquetes, que incluya varias aplicaciones de servidor.

Un hecho común entre los administradores de sistemas es instalar el sistema operativo sin prestar atención a qué programas se están instalando realmente. Esto puede ser problemático porque se pueden instalar servicios innecesarios, configurados con los parámetros por defecto y posiblemente activados. Esto puede hacer que servicios no deseados, como Telnet, DHCP o DNS, se ejecuten en un servidor o estación de trabajo sin que el administrador se dé cuenta, lo que a su vez puede provocar tráfico no deseado en el servidor o incluso una posible vía de acceso al sistema para los crackers.

Servicios sin parches

La mayoría de las aplicaciones de servidor que se incluyen en una instalación por defecto son piezas de software sólidas y ampliamente probadas. Al haber estado en uso en entornos de producción durante muchos años, su código se ha perfeccionado a fondo y se han encontrado y corregido muchos de los errores.

Sin embargo, no existe el software perfecto y siempre se puede perfeccionar. Además, el software más nuevo no suele ser probado tan rigurosamente como cabría esperar, debido a su reciente llegada a los entornos de producción o porque puede no ser tan popular como otro software de servidor.

Los desarrolladores y administradores de sistemas suelen encontrar fallos explotables en las aplicaciones de los servidores y publican la información en sitios web de seguimiento de fallos y

relacionados con la seguridad, como la lista de correo Bugtraq (http://www.securityfocus.com) o el sitio web del Computer Emergency Response Team (CERT) (http://www.cert.org). Aunque estos mecanismos son una forma eficaz de alertar a la comunidad sobre las vulnerabilidades de seguridad, corresponde a los administradores de sistemas parchear sus sistemas con prontitud. Esto es especialmente cierto porque los crackers tienen acceso a estos mismos servicios de seguimiento de vulnerabilidades y utilizarán la información para crackear los sistemas no parcheados siempre que puedan. Una buena administración de sistemas requiere vigilancia, un seguimiento constante de los errores y un mantenimiento adecuado del sistema para garantizar un entorno informático más seguro.

Administración desatenta

Los administradores que no parchean sus sistemas son una de las mayores amenazas para la seguridad de los servidores. Esto se aplica tanto a los administradores inexpertos como a los administradores demasiado confiados o desmotivados.

Algunos administradores no parchean sus servidores y estaciones de trabajo, mientras que otros no vigilan los mensajes de registro del núcleo del sistema o el tráfico de red. Otro error común es cuando se dejan sin cambiar las contraseñas por defecto o las claves de los servicios. Por ejemplo, algunas bases de datos tienen contraseñas de administración por defecto porque los desarrolladores de la base de datos asumen que el administrador del sistema cambia estas contraseñas inmediatamente después de la instalación. Si el administrador de la base de datos no cambia esta contraseña, incluso un cracker inexperto puede utilizar una contraseña por defecto ampliamente conocida para obtener privilegios administrativos en la base de datos. Estos son sólo algunos ejemplos de cómo una administración desatenta puede llevar a servidores comprometidos.

Servicios intrínsecamente inseguros

Incluso la organización más vigilante puede ser víctima de vulnerabilidades si los servicios de red que elige son intrínsecamente inseguros. Por ejemplo, hay muchos servicios desarrollados bajo el supuesto de que se utilizan a través de redes de confianza; sin embargo, este supuesto falla tan pronto como el servicio está disponible en Internet

Una categoría de servicios de red inseguros son los que requieren nombres de usuario y contraseñas sin cifrar para la autenticación. Telnet y FTP son dos de estos servicios. Si el software de rastreo de paquetes está monitoreando el tráfico entre el usuario remoto y dicho servicio, los nombres de usuario y las contraseñas pueden ser fácilmente interceptados.

Intrínsecamente, estos servicios también pueden ser más fácilmente presa de lo que la industria de la seguridad denomina el ataque *man-in-the-middle*. En este tipo de ataque, un cracker redirige el tráfico de la red engañando a un servidor de nombres crackeado en la red para que apunte a su máquina en lugar del servidor previsto. Una vez que alguien abre una sesión remota al servidor, la máquina del atacante actúa como un conducto invisible, situándose tranquilamente entre el servicio remoto y el usuario desprevenido, capturando información. De este modo, un cracker puede recopilar contraseñas administrativas y datos en bruto sin que el servidor o el usuario se den cuenta.

Otra categoría de servicios inseguros son los sistemas de archivos de red y los servicios de información como NFS o NIS, desarrollados explícitamente para su uso en redes LAN pero que, desgraciadamente, se han extendido a las WAN (para usuarios remotos). NFS no tiene, por defecto, ningún mecanismo de autenticación o seguridad configurado para evitar que un cracker monte el recurso compartido NFS y acceda a cualquier cosa que contenga. NIS, además, tiene información vital que debe ser conocida por todos los ordenadores de una red, incluyendo contraseñas y permisos de archivos, dentro de una base de datos de texto plano ASCII o DBM (derivado de ASCII). Un cracker que consiga acceder a esta base de datos puede entonces acceder a todas las cuentas de usuario de una red, incluida la del administrador.

Por defecto, Red Hat Enterprise Linux 8 se publica con todos estos servicios desactivados. Sin embargo, dado que los administradores a menudo se ven obligados a utilizar estos servicios, una configuración cuidadosa es fundamental.

1.6.3. Amenazas para la seguridad de los puestos de trabajo y los ordenadores personales

Las estaciones de trabajo y los ordenadores domésticos pueden no ser tan propensos a los ataques como las redes o los servidores, pero como a menudo contienen datos sensibles, como la información de las tarjetas de crédito, son el objetivo de los crackers de sistemas. Las estaciones de trabajo también pueden ser cooptadas sin el conocimiento del usuario y utilizadas por los atacantes como máquinas "bot" en ataques coordinados. Por estas razones, conocer las vulnerabilidades de una estación de trabajo puede ahorrar a los usuarios el dolor de cabeza de reinstalar el sistema operativo, o peor aún, recuperarse del robo de datos.

Contraseñas incorrectas

Las contraseñas incorrectas son una de las formas más fáciles de que un atacante acceda a un sistema.

Aplicaciones cliente vulnerables

Aunque un administrador pueda tener un servidor totalmente seguro y parcheado, eso no significa que los usuarios remotos estén seguros al acceder a él. Por ejemplo, si el servidor ofrece servicios Telnet o FTP a través de una red pública, un atacante puede capturar los nombres de usuario y las contraseñas en texto plano mientras pasan por la red, y luego utilizar la información de la cuenta para acceder a la estación de trabajo del usuario remoto.

Incluso cuando se utilizan protocolos seguros, como SSH, un usuario remoto puede ser vulnerable a ciertos ataques si no mantiene sus aplicaciones cliente actualizadas. Por ejemplo, los clientes del protocolo SSH versión 1 son vulnerables a un ataque de reenvío de X desde servidores SSH maliciosos. Una vez conectado al servidor, el atacante puede capturar silenciosamente las pulsaciones de teclas y los clics del ratón realizados por el cliente a través de la red. Este problema se solucionó en la versión 2 del protocolo SSH, pero es responsabilidad del usuario estar al tanto de qué aplicaciones tienen esas vulnerabilidades y actualizarlas cuando sea necesario.

1.7. ATAQUES Y EXPLOITS COMUNES

Tabla 1.1, “Héroes comunes” detalla algunos de los exploits y puntos de entrada más comunes utilizados por los intrusos para acceder a los recursos de la red de la organización. La clave de estos exploits comunes son las explicaciones de cómo se realizan y cómo los administradores pueden salvaguardar adecuadamente su red contra estos ataques.

Tabla 1.1. Héroes comunes

Explotar	Descripción	Notas

Explotar	Descripción	Notas
Contraseñas nulas o por defecto	Dejar las contraseñas administrativas en blanco o utilizar una contraseña por defecto establecida por el proveedor del producto. Esto es más común en hardware como routers y firewalls, pero algunos servicios que se ejecutan en Linux pueden contener contraseñas de administrador por defecto también (aunque Red Hat Enterprise Linux 8 no se entrega con ellos).	Se asocia habitualmente con el hardware de red, como los routers, los cortafuegos, las VPN y los dispositivos de almacenamiento en red (NAS). Común en muchos sistemas operativos heredados, especialmente los que agrupan servicios (como UNIX y Windows) Los administradores a veces crean cuentas de usuario privilegiadas con prisas y dejan la contraseña nula, creando un punto de entrada perfecto para los usuarios maliciosos que descubren la cuenta.
Claves compartidas por defecto	Los servicios de seguridad a veces empaquetan claves de seguridad por defecto para fines de desarrollo o pruebas de evaluación. Si estas claves no se modifican y se colocan en un entorno de producción en Internet, los usuarios de **all** con las mismas claves por defecto tienen acceso a ese recurso de clave compartida, y a cualquier información sensible que contenga.	Más común en los puntos de acceso inalámbricos y en los dispositivos de servidor seguro preconfigurados.
Suplantación de IP	Una máquina remota actúa como un nodo en su red local, encuentra vulnerabilidades en sus servidores e instala un programa de puerta trasera o un troyano para obtener el control de sus recursos de red.	La suplantación de identidad es bastante difícil, ya que implica que el atacante prediga los números de secuencia TCP/IP para coordinar una conexión con los sistemas objetivo, pero existen varias herramientas que ayudan a los crackers a realizar dicha vulnerabilidad. Depende de que el sistema de destino ejecute servicios (como **rsh**, **telnet**, FTP y otros) que utilicen técnicas de autenticación *source-based*, que no se recomiendan en comparación con PKI u otras formas de autenticación cifrada utilizadas en **ssh** o SSL/TLS.

Explotar	Descripción	Notas
Espionaje	Recogida de datos que pasan entre dos nodos activos de una red mediante la escucha de la conexión entre los dos nodos.	Este tipo de ataque funciona sobre todo con protocolos de transmisión de texto plano como Telnet, FTP y transferencias HTTP. El atacante remoto debe tener acceso a un sistema comprometido en una LAN para poder realizar dicho ataque; normalmente el cracker ha utilizado un ataque activo (como la suplantación de IP o el man-in-the-middle) para comprometer un sistema en la LAN. Las medidas preventivas incluyen servicios con intercambio de claves criptográficas, contraseñas de un solo uso o autenticación encriptada para evitar el espionaje de contraseñas; también se aconseja un fuerte cifrado durante la transmisión.

Explotar	Descripción	Notas
Vulnerabilidades del servicio	Un atacante encuentra un fallo o una laguna en un servicio que se ejecuta a través de Internet; a través de esta vulnerabilidad, el atacante compromete todo el sistema y los datos que pueda contener, y posiblemente podría comprometer otros sistemas de la red.	Los servicios basados en HTTP, como los CGI, son vulnerables a la ejecución remota de comandos e incluso al acceso interactivo al shell. Incluso si el servicio HTTP se ejecuta como un usuario sin privilegios, como "nadie", se puede leer información como archivos de configuración y mapas de red, o el atacante puede iniciar un ataque de denegación de servicio que agote los recursos del sistema o lo deje indisponible para otros usuarios. Los servicios a veces pueden tener vulnerabilidades que pasan desapercibidas durante el desarrollo y las pruebas; estas vulnerabilidades (como *buffer overflows*, donde los atacantes bloquean un servicio utilizando valores arbitrarios que llenan el buffer de memoria de una aplicación, dando al atacante un prompt de comando interactivo desde el que puede ejecutar comandos arbitrarios) pueden dar un control administrativo completo a un atacante. Los administradores deben asegurarse de que los servicios no se ejecutan como usuario raíz, y deben estar atentos a los parches y actualizaciones de erratas de las aplicaciones de los proveedores u organizaciones de seguridad como CERT y CVE.

Explotar	Descripción	Notas
Vulnerabilidades de las aplicaciones	Los atacantes encuentran fallos en las aplicaciones de escritorio y de estaciones de trabajo (como los clientes de correo electrónico) y ejecutan código arbitrario, implantan troyanos para comprometerlos en el futuro o bloquean los sistemas. La explotación puede ser mayor si la estación de trabajo comprometida tiene privilegios administrativos en el resto de la red.	Los puestos de trabajo y los ordenadores de sobremesa son más propensos a ser explotados, ya que los trabajadores no tienen los conocimientos o la experiencia necesarios para prevenir o detectar un compromiso; es imperativo informar a las personas de los riesgos que corren cuando instalan software no autorizado o abren archivos adjuntos de correo electrónico no solicitados. Se pueden implementar salvaguardas para que el software del cliente de correo electrónico no abra o ejecute automáticamente los archivos adjuntos. Además, la actualización automática del software de las estaciones de trabajo mediante Red Hat Network; u otros servicios de gestión de sistemas pueden aliviar las cargas de las implantaciones de seguridad en varios puestos.

Explotar	Descripción	Notas
Ataques de denegación de servicio (DoS)	El atacante o grupo de atacantes se coordinan contra los recursos de red o servidores de una organización enviando paquetes no autorizados al host objetivo (ya sea servidor, router o estación de trabajo). Esto hace que el recurso deje de estar disponible para los usuarios legítimos.	El caso de DoS más reportado en los Estados Unidos ocurrió en el año 2000. Varios sitios comerciales y gubernamentales muy frecuentados quedaron inutilizados por un ataque coordinado de inundación de ping que utilizaba varios sistemas comprometidos con conexiones de gran ancho de banda que actuaban como *zombie*s, o nodos de difusión redirigidos. Los paquetes de origen suelen ser falsificados (así como retransmitidos), lo que dificulta la investigación del verdadero origen del ataque. Los avances en el filtrado de entrada (IETF rfc2267) mediante **iptables** y los sistemas de detección de intrusiones en la red, como **snort**, ayudan a los administradores a rastrear y prevenir los ataques DoS distribuidos.

CAPÍTULO 2. ASEGURAR RHEL DURANTE LA INSTALACIÓN

La seguridad comienza incluso antes de iniciar la instalación de Red Hat Enterprise Linux. Configurar su sistema de forma segura desde el principio facilita la implementación de ajustes de seguridad adicionales más adelante.

2.1. SEGURIDAD DE LA BIOS Y LA UEFI

La protección con contraseña de la BIOS (o su equivalente) y del gestor de arranque puede evitar que usuarios no autorizados que tengan acceso físico a los sistemas arranquen utilizando medios extraíbles u obtengan privilegios de root a través del modo de usuario único. Las medidas de seguridad que debe tomar para protegerse de estos ataques dependen tanto de la sensibilidad de la información en la estación de trabajo como de la ubicación de la máquina.

Por ejemplo, si una máquina se utiliza en una feria comercial y no contiene información sensible, entonces puede no ser crítico prevenir tales ataques. Sin embargo, si el portátil de un empleado con claves SSH privadas y sin cifrar para la red corporativa se deja sin vigilancia en esa misma feria, podría provocar una importante brecha de seguridad con ramificaciones para toda la empresa.

Sin embargo, si la estación de trabajo se encuentra en un lugar al que sólo tienen acceso personas autorizadas o de confianza, puede que no sea necesario asegurar la BIOS o el gestor de arranque.

2.1.1. Contraseñas de la BIOS

Las dos razones principales para proteger con contraseña la BIOS de un ordenador son[1]:

1. **Preventing changes to BIOS settings**
2. **Preventing system booting**

Dado que los métodos para configurar la contraseña de la BIOS varían entre los fabricantes de ordenadores, consulte el manual del ordenador para obtener instrucciones específicas.

Si olvida la contraseña de la BIOS, puede restablecerla con puentes en la placa base o desconectando la batería de la CMOS. Por esta razón, es una buena práctica bloquear la caja del ordenador si es posible. Sin embargo, consulte el manual del ordenador o de la placa base antes de intentar desconectar la batería del CMOS.

2.1.1.1. Seguridad de los sistemas no basados en BIOS

Otros sistemas y arquitecturas utilizan diferentes programas para realizar tareas de bajo nivel aproximadamente equivalentes a las de la BIOS en los sistemas x86. Por ejemplo, el shell *Unified Extensible Firmware Interface* (*UEFI*).

Para obtener instrucciones sobre la protección con contraseña de los programas tipo BIOS, consulte las instrucciones del fabricante.

2.2. PARTICIÓN DEL DISCO

Red Hat recomienda crear particiones separadas para los directorios **/boot**, /, **/home**, **/tmp**, y **/var/tmp/**. Las razones para cada uno de ellos son diferentes, y nos ocuparemos de cada partición.

/boot

Esta partición es la primera que lee el sistema durante el arranque. El gestor de arranque y las

imágenes del kernel que se utilizan para arrancar su sistema en Red Hat Enterprise Linux 8 se almacenan en esta partición. Esta partición no debería estar encriptada. Si esta partición está incluida en / y esa partición está encriptada o no está disponible, entonces su sistema no podrá arrancar.

/home

Cuando los datos del usuario (**/home**) se almacenan en / en lugar de en una partición separada, la partición puede llenarse causando que el sistema operativo se vuelva inestable. Además, cuando actualice su sistema a la siguiente versión de Red Hat Enterprise Linux 8 es mucho más fácil cuando puede mantener sus datos en la partición **/home** ya que no se sobrescribirán durante la instalación. Si la partición raíz (/) se corrompe sus datos podrían perderse para siempre. Al usar una partición separada hay un poco más de protección contra la pérdida de datos. También puedes destinar esta partición a realizar copias de seguridad frecuentes.

/tmp y /var/tmp/

Los directorios **/tmp** y **/var/tmp/** se utilizan para almacenar datos que no necesitan ser almacenados durante un largo periodo de tiempo. Sin embargo, si una gran cantidad de datos inunda uno de estos directorios puede consumir todo su espacio de almacenamiento. Si esto ocurre y estos directorios se almacenan en /, su sistema podría volverse inestable y bloquearse. Por esta razón, mover estos directorios a sus propias particiones es una buena idea.

NOTA

Durante el proceso de instalación, tienes la opción de cifrar las particiones. Debes proporcionar una frase de contraseña. Esta frase de contraseña sirve como clave para desbloquear la clave de encriptación masiva, que se utiliza para asegurar los datos de la partición.

2.3. RESTRICCIÓN DE LA CONECTIVIDAD A LA RED DURANTE EL PROCESO DE INSTALACIÓN

Cuando se instala Red Hat Enterprise Linux 8, el medio de instalación representa una instantánea del sistema en un momento determinado. Debido a esto, puede que no esté actualizado con las últimas correcciones de seguridad y puede ser vulnerable a ciertos problemas que fueron corregidos sólo después de que el sistema proporcionado por el medio de instalación fuera lanzado.

Al instalar un sistema operativo potencialmente vulnerable, limite siempre la exposición sólo a la zona de red necesaria más cercana. La opción más segura es la zona "sin red", lo que significa dejar la máquina desconectada durante el proceso de instalación. En algunos casos, una conexión de LAN o intranet es suficiente, mientras que la conexión a Internet es la más arriesgada. Para seguir las mejores prácticas de seguridad, elija la zona más cercana a su repositorio mientras instala Red Hat Enterprise Linux 8 desde una red.

2.4. INSTALAR LA CANTIDAD MÍNIMA DE PAQUETES NECESARIOS

Es una buena práctica instalar sólo los paquetes que va a utilizar, ya que cada pieza de software en su ordenador podría contener una vulnerabilidad. Si está instalando desde el DVD, aproveche para seleccionar exactamente los paquetes que desea instalar durante la instalación. Si descubre que necesita otro paquete, siempre puede añadirlo al sistema más tarde.

2.5. PROCEDIMIENTOS POSTERIORES A LA INSTALACIÓN

Los siguientes pasos son los procedimientos relacionados con la seguridad que deberían realizarse inmediatamente después de la instalación de Red Hat Enterprise Linux 8.

- Actualice su sistema. Introduzca el siguiente comando como root:

```
# yum update
```

- Aunque el servicio de cortafuegos, **firewalld**, se habilita automáticamente con la instalación de Red Hat Enterprise Linux, hay escenarios en los que puede estar explícitamente deshabilitado, por ejemplo en la configuración de kickstart. En tal caso, se recomienda considerar la posibilidad de volver a habilitar el cortafuegos.
  Para iniciar **firewalld** introduzca los siguientes comandos como root:

```
# systemctl start firewalld
# systemctl enable firewalld
```

- Para mejorar la seguridad, desactive los servicios que no necesite. Por ejemplo, si no hay impresoras instaladas en su ordenador, desactive el servicio **cups** mediante el siguiente comando:

```
# systemctl disable cups
```

  Para revisar los servicios activos, introduzca el siguiente comando:

```
$ systemctl list-units | grep service
```

2.6. INSTALACIÓN DE UN SISTEMA RHEL 8 CON EL MODO FIPS ACTIVADO

Para habilitar las autocomprobaciones del módulo criptográfico exigidas por la Publicación 140-2 del Estándar Federal de Procesamiento de Información (FIPS), tiene que operar RHEL 8 en modo FIPS. Puede conseguirlo de la siguiente manera:

- Iniciar la instalación en modo FIPS.
- Pasar el sistema al modo FIPS después de la instalación.

Para evitar la regeneración de material de claves criptográficas y la reevaluación de la conformidad del sistema resultante asociada a la conversión de sistemas ya implantados, Red Hat recomienda iniciar la instalación en modo FIPS.

2.6.1. Norma Federal de Procesamiento de la Información (FIPS)

La publicación 140-2 de la Norma Federal de Procesamiento de la Información (FIPS) es una norma de seguridad informática desarrollada por el grupo de trabajo del Gobierno y la industria de Estados Unidos para validar la calidad de los módulos criptográficos. Consulte las publicaciones oficiales de FIPS en el Centro de Recursos de Seguridad Informática del NIST.

La norma FIPS 140-2 garantiza que las herramientas criptográficas implementen sus algoritmos correctamente. Uno de los mecanismos para ello es la autocomprobación en tiempo de ejecución. Consulte la norma FIPS 140-2 completa en FIPS PUB 140-2 para obtener más detalles y otras especificaciones de la norma FIPS.

Para conocer los requisitos de cumplimiento, consulte la página de normas gubernamentales de Red Hat.

2.6.2. Instalación del sistema con el modo FIPS activado

Para habilitar las autocomprobaciones del módulo criptográfico exigidas por la Publicación 140-2 del Estándar Federal de Procesamiento de Información (FIPS), habilite el modo FIPS durante la instalación del sistema.

IMPORTANTE

Red Hat recomienda instalar Red Hat Enterprise Linux 8 con el modo FIPS activado, en lugar de activar el modo FIPS más tarde. La habilitación del modo FIPS durante la instalación garantiza que el sistema genere todas las claves con algoritmos aprobados por FIPS y pruebas de supervisión continua en el lugar.

Procedimiento

- Añade la opción **fips=1** a la línea de comandos del kernel durante la instalación del sistema. Durante la etapa de selección del software, no instale ningún software de terceros.

Tras la instalación, el sistema se inicia automáticamente en modo FIPS.

Pasos de verificación

- Después de iniciar el sistema, compruebe que el modo FIPS está activado:

```
$ fips-mode-setup --check
FIPS mode is enabled.
```

Recursos adicionales

- Consulte la sección Editar las opciones de arranque en el documento Realizar una instalación avanzada de RHEL para obtener más información sobre las diferentes formas de editar las opciones de arranque.

2.6.3. Recursos adicionales

- Cambio del sistema al modo FIPS
- Activación del modo FIPS en un contenedor
- Lista de aplicaciones de RHEL 8 que utilizan criptografía no conforme con FIPS 140-2

[1] Dado que las BIOS de los sistemas difieren entre los fabricantes, es posible que algunas no admitan la protección con contraseña de ninguno de los dos tipos, mientras que otras pueden admitir un tipo pero no el otro.

CAPÍTULO 3. USO DE POLÍTICAS CRIPTOGRÁFICAS EN TODO EL SISTEMA

Crypto policies es un componente del sistema que configura los subsistemas criptográficos principales, cubriendo los protocolos TLS, IPSec, SSH, DNSSec y Kerberos. Proporciona un pequeño conjunto de políticas, que el administrador puede seleccionar.

3.1. POLÍTICAS CRIPTOGRÁFICAS PARA TODO EL SISTEMA

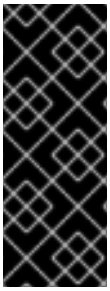
Una vez que se establece una política para todo el sistema, las aplicaciones en RHEL la siguen y se niegan a utilizar algoritmos y protocolos que no cumplan con la política, a menos que usted solicite explícitamente a la aplicación que lo haga. Es decir, la política se aplica al comportamiento por defecto de las aplicaciones cuando se ejecutan con la configuración proporcionada por el sistema, pero usted puede anularla si así lo requiere.

Red Hat Enterprise Linux 8 contiene los siguientes niveles de política:

DEFAULT	El nivel de política criptográfica por defecto en todo el sistema ofrece una configuración segura para los modelos de amenaza actuales. Permite los protocolos TLS 1.2 y 1.3, así como los protocolos IKEv2 y SSH2. Las claves RSA y los parámetros Diffie-Hellman se aceptan si tienen una longitud mínima de 2048 bits.
LEGACY	Esta política garantiza la máxima compatibilidad con Red Hat Enterprise Linux 5 y anteriores; es menos segura debido a una mayor superficie de ataque. Además de los algoritmos y protocolos de nivel **DEFAULT**, incluye soporte para los protocolos TLS 1.0 y 1.1. Se permiten los algoritmos DSA, 3DES y RC4, mientras que las claves RSA y los parámetros Diffie-Hellman se aceptan si tienen una longitud mínima de 1023 bits.
FUTURE	Un nivel de seguridad conservador que se cree que resistirá cualquier ataque futuro a corto plazo. Este nivel no permite el uso de SHA-1 en los algoritmos de firma. Las claves RSA y los parámetros Diffie-Hellman se aceptan si tienen una longitud mínima de 3072 bits.
FIPS	Un nivel de política que se ajusta a los requisitos de FIPS 140-2. Lo utiliza internamente la herramienta **fips-mode-setup**, que cambia el sistema RHEL al modo FIPS.

Red Hat ajusta continuamente todos los niveles de políticas para que todas las bibliotecas, excepto cuando se utiliza la política LEGACY, proporcionen valores predeterminados seguros. Aunque el perfil LEGACY no proporciona valores predeterminados seguros, no incluye ningún algoritmo que sea fácilmente explotable. Como tal, el conjunto de algoritmos habilitados o los tamaños de clave aceptables en cualquier política proporcionada pueden cambiar durante la vida de RHEL 8.

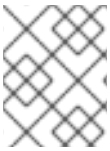
Estos cambios reflejan los nuevos estándares de seguridad y las nuevas investigaciones en materia de seguridad. Si debe garantizar la interoperabilidad con un sistema específico durante toda la vida útil de RHEL 8, debe optar por no aplicar políticas criptográficas a los componentes que interactúan con ese sistema.



IMPORTANTE

Debido a que una clave criptográfica utilizada por un certificado en la API del Portal del Cliente no cumple los requisitos de la política criptográfica de todo el sistema **FUTURE**, la utilidad **redhat-support-tool** no funciona con este nivel de política por el momento.

Para solucionar este problema, utilice la política de cifrado **DEFAULT** mientras se conecta a la API del Portal del Cliente.



NOTA

Los algoritmos y cifrados específicos descritos en los niveles de política como permitidos sólo están disponibles si una aplicación los soporta.

Herramienta para la gestión de las criptopolíticas

Para ver o cambiar la política criptográfica actual de todo el sistema, utilice la herramienta **update-crypto-policies**, por ejemplo:

```
$ update-crypto-policies --show
DEFAULT
# update-crypto-policies --set FUTURE
Setting system policy to FUTURE
```

Para asegurarse de que el cambio de la política criptográfica se aplica, reinicie el sistema.

Criptografía fuerte por defecto mediante la eliminación de suites de cifrado y protocolos inseguros

La siguiente lista contiene conjuntos de cifrado y protocolos eliminados de las bibliotecas criptográficas del núcleo en RHEL 8. No están presentes en las fuentes, o su soporte está deshabilitado durante la compilación, por lo que las aplicaciones no pueden utilizarlos.

- DES (desde RHEL 7)
- Todos los conjuntos de cifrado de grado de exportación (desde RHEL 7)
- MD5 en las firmas (desde RHEL 7)
- SSLv2 (desde RHEL 7)
- SSLv3 (desde RHEL 8)
- Todas las curvas ECC < 224 bits (desde RHEL 6)
- Todas las curvas ECC de campo binario (desde RHEL 6)

Suites de cifrado y protocolos desactivados en todos los niveles de política

Los siguientes conjuntos de cifrado y protocolos están deshabilitados en todos los niveles de la política de cifrado. Sólo se pueden habilitar mediante una configuración explícita de las aplicaciones individuales.

- DH con parámetros < 1024 bits
- RSA con tamaño de clave < 1024 bits
- Camelia

- ARIA
- SEED
- IDEA
- Suites de cifrado de sólo integridad
- Suites de cifrado en modo CBC de TLS con SHA-384 HMAC
- AES-CCM8
- Todas las curvas ECC incompatibles con TLS 1.3, incluida secp256k1
- IKEv1 (desde RHEL 8)

Suites de cifrado y protocolos habilitados en los niveles de criptopolíticas

La siguiente tabla muestra los conjuntos de cifrado y protocolos habilitados en los cuatro niveles de criptopolíticas.

	LEGACY	DEFAULT	FIPS	FUTURE
IKEv1	no	no	no	no
3DES	sí	no	no	no
RC4	sí	no	no	no
DH	mínimo. 1024 bits	mínimo. 2048 bits	mínimo. 2048 bits	mínimo. 3072 bits
RSA	mínimo. 1024 bits	mínimo. 2048 bits	mínimo. 2048 bits	mínimo. 3072 bits
DSA	sí	no	no	no
TLS v1.0	sí	no	no	no
TLS v1.1	sí	no	no	no
SHA-1 in digital signatures	sí	sí	no	no
CBC mode ciphers	sí	sí	sí	no
Symmetric ciphers with keys < 256 bits	sí	sí	sí	no
SHA-1 and SHA-224 signatures in certificates	sí	sí	sí	no

Recursos adicionales

- Para más detalles, consulte la página de manual **update-crypto-policies(8)**.

3.2. CAMBIO DE LA POLÍTICA CRIPTOGRÁFICA DE TODO EL SISTEMA AL MODO COMPATIBLE CON VERSIONES ANTERIORES

La política criptográfica por defecto en todo el sistema en Red Hat Enterprise Linux 8 no permite la comunicación utilizando protocolos antiguos e inseguros. Para los entornos que requieren ser compatibles con Red Hat Enterprise Linux 5 y en algunos casos también con versiones anteriores, está disponible el nivel de política menos seguro **LEGACY**.

AVISO

El cambio al nivel de política **LEGACY** resulta en un sistema y aplicaciones menos seguros.

Procedimiento

1. Para cambiar la política criptográfica de todo el sistema al nivel **LEGACY**, introduzca el siguiente comando como **root**:

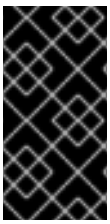
```
# update-crypto-policies --set LEGACY
Setting system policy to LEGACY
```

Recursos adicionales

- Para ver la lista de niveles de políticas criptográficas disponibles, consulte la página de manual **update-crypto-policies(8)**.

3.3. CAMBIO DEL SISTEMA AL MODO FIPS

Las políticas criptográficas de todo el sistema contienen un nivel de política que permite la autocomprobación de los módulos criptográficos de acuerdo con los requisitos de la Publicación 140-2 del Estándar Federal de Procesamiento de Información (FIPS). La herramienta **fips-mode-setup** que activa o desactiva el modo FIPS utiliza internamente el nivel de política criptográfica de todo el sistema **FIPS**.



IMPORTANTE

Red Hat recomienda instalar Red Hat Enterprise Linux 8 con el modo FIPS activado, en lugar de activar el modo FIPS más tarde. La habilitación del modo FIPS durante la instalación garantiza que el sistema genere todas las claves con algoritmos aprobados por FIPS y pruebas de supervisión continua en el lugar.

Procedimiento

1. Para cambiar el sistema al modo FIPS en RHEL 8:

```
# fips-mode-setup --enable
Setting system policy to FIPS
FIPS mode will be enabled.
Please reboot the system for the setting to take effect.
```

2. Reinicie su sistema para permitir que el kernel cambie al modo FIPS:

```
# reboot
```

Pasos de verificación

1. Tras el reinicio, puede comprobar el estado actual del modo FIPS:

```
# fips-mode-setup --check
FIPS mode is enabled.
```

Recursos adicionales

- La página de manual **fips-mode-setup(8)**.
- Lista de aplicaciones de RHEL 8 que utilizan criptografía y no cumplen con FIPS 140-2
- Para obtener más detalles sobre FIPS 140-2, consulte los requisitos de seguridad para los módulos criptográficos en el sitio web del Instituto Nacional de Normas y Tecnología (NIST).

3.4. ACTIVACIÓN DEL MODO FIPS EN UN CONTENEDOR

Permitir la autocomprobación de los módulos criptográficos de acuerdo con los requisitos de la Publicación 140-2 de la Norma Federal de Procesamiento de la Información (FIPS) en un contenedor:

Requisitos previos

- En primer lugar, el sistema anfitrión debe cambiarse al modo FIPS; consulte Cómo cambiar el sistema al modo FIPS.

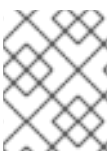
Procedimiento

1. Monte el archivo **/etc/system-fips** en el contenedor desde el host.
2. Establezca el nivel de política criptográfica FIPS en el contenedor:

```
$ update-crypto-policies --set FIPS
```

RHEL 8.2 introdujo un método alternativo para cambiar un contenedor al modo FIPS. Sólo requiere el uso del siguiente comando en el contenedor:

```
# mount --bind /usr/share/crypto-policies/back-ends/FIPS /etc/crypto-policies/back-ends
```



NOTA

En RHEL 8, el comando **fips-mode-setup** no funciona correctamente en un contenedor y no se puede utilizar para activar o comprobar el modo FIPS en este escenario.

3.5. LISTA DE APLICACIONES DE RHEL QUE UTILIZAN CRIPTOGRAFÍA QUE NO CUMPLE CON FIPS 140-2

Red Hat recomienda utilizar las librerías del conjunto de componentes criptográficos principales, ya que están garantizadas para pasar todas las certificaciones criptográficas relevantes, como FIPS 140-2, y también siguen las políticas criptográficas de todo el sistema RHEL.

Consulte el artículo sobre los componentes criptográficos del núcleo de RHEL 8 para obtener una visión general de los componentes criptográficos del núcleo de RHEL 8, la información sobre cómo se seleccionan, cómo se integran en el sistema operativo, cómo admiten los módulos de seguridad de hardware y las tarjetas inteligentes, y cómo se aplican las certificaciones criptográficas a ellos.

Además de la tabla siguiente, en algunas versiones de RHEL 8 Z-stream (por ejemplo, 8.1.1), se han actualizado los paquetes del navegador Firefox, que contienen una copia separada de la biblioteca de criptografía NSS. De este modo, Red Hat quiere evitar la interrupción que supone volver a basar un componente de tan bajo nivel en una versión de parche. Como resultado, estos paquetes de Firefox no utilizan un módulo validado por FIPS 140-2.

Tabla 3.1. Lista de aplicaciones de RHEL 8 que utilizan criptografía no conforme con FIPS 140-2

Aplicación	Detalles
FreeRADIUS	El protocolo RADIUS utiliza MD5
ghostscript	Criptografía propia (MD5, RC4, SHA-2, AES) para cifrar y descifrar documentos
ipxe	La pila criptográfica para TLS está compilada, pero no se utiliza
java-1.8.0-openjdk	Pila criptográfica completa[a]
libica	Software de respaldo para varios algoritmos como RSA y ECDH mediante instrucciones CPACF
Ovmf (firmware UEFI), Edk2, shim	Pila criptográfica completa (una copia incrustada de la biblioteca OpenSSL)
perl-Digest-HMAC	HMAC, HMAC-SHA1, HMAC-MD5
perl-Digest-SHA	SHA-1, SHA-224, ..
pidgin	DES, RC4
samba[b]	AES, DES, RC4
valgrind	AES, hashes[c]

Aplicación	Detalles

[a] En RHEL 8.1, java-1.8.0-openjdk requiere una configuración manual adicional para ser compatible con FIPS.

[b] A partir de RHEL 8.3, samba utiliza criptografía compatible con FIPS.

[c] Reimplementa en software las operaciones de descarga de hardware, como AES-NI.

3.6. EXCLUIR UNA APLICACIÓN DE SEGUIR LAS POLÍTICAS CRIPTOGRÁFICAS DE TODO EL SISTEMA

Puedes personalizar la configuración criptográfica utilizada por tu aplicación preferentemente configurando los conjuntos de cifrado y protocolos soportados directamente en la aplicación.

También puede eliminar un enlace simbólico relacionado con su aplicación del directorio **/etc/crypto-policies/back-ends** y sustituirlo por su configuración criptográfica personalizada. Esta configuración impide el uso de políticas criptográficas en todo el sistema para las aplicaciones que utilizan el back end excluido. Además, esta modificación no está soportada por Red Hat.

3.6.1. Ejemplos de exclusión de las políticas criptográficas de todo el sistema

wget

Para personalizar la configuración criptográfica utilizada por el descargador de red **wget**, utilice las opciones **--secure-protocol** y **--ciphers**. Por ejemplo:

```
$ wget --secure-protocol=TLSv1_1 --ciphers="SECURE128" https://example.com
```

Consulte la sección Opciones HTTPS (SSL/TLS) de la página de manual **wget(1)** para obtener más información.

rizo

Para especificar los cifrados utilizados por la herramienta **curl**, utilice la opción **--ciphers** y proporcione una lista de cifrados separada por dos puntos como valor. Por ejemplo:

```
$ curl https://example.com --ciphers '@SECLEVEL=0:DES-CBC3-SHA:RSA-DES-CBC3-SHA'
```

Consulte la página de manual **curl(1)** para obtener más información.

Firefox

Aunque no se puede optar por las políticas criptográficas de todo el sistema en el navegador web **Firefox**, se pueden restringir aún más los cifrados y las versiones de TLS compatibles en el Editor de Configuración de Firefox. Escriba **about:config** en la barra de direcciones y cambie el valor de la opción **security.tls.version.min** según sea necesario. Configurar **security.tls.version.min** a **1** permite TLS 1.0 como mínimo requerido, **security.tls.version.min 2** habilita TLS 1.1, y así sucesivamente.

OpenSSH

Para optar por las políticas criptográficas de todo el sistema para su servidor **OpenSSH**, descomente la línea con la variable **CRYPTO_POLICY=** en el archivo **/etc/sysconfig/sshd**. Después de este cambio, los valores que especifique en las secciones **Ciphers**, **MACs**, **KexAlgoritms**, y **GSSAPIKexAlgorithms**

en el archivo **/etc/ssh/sshd_config** no serán anulados. Consulte la página man **sshd_config(5)** para obtener más información.

Libreswan

Consulte la sección Configuración de conexiones IPsec que se excluyen de las políticas criptográficas de todo el sistema en el documento Protección de redes para obtener información detallada.

Recursos adicionales

- Para más detalles, consulte la página de manual **update-crypto-policies(8)**.

3.7. PERSONALIZACIÓN DE LAS POLÍTICAS CRIPTOGRÁFICAS DE TODO EL SISTEMA CON MODIFICADORES DE POLÍTICAS

Utilice este procedimiento para ajustar determinados algoritmos o protocolos de cualquier nivel de política criptográfica de todo el sistema o de una política personalizada completa.

NOTA

La personalización de las políticas criptográficas de todo el sistema está disponible desde RHEL 8.2.

Procedimiento

1. Acceda al directorio **/etc/crypto-policies/policies/modules/**:

```
# cd /etc/crypto-policies/policies/modules/
```

2. Cree módulos de política para sus ajustes, por ejemplo:

```
# touch MYCRYPTO1.pmod
# touch NO-AES128.pmod
```

IMPORTANTE

Utilice letras mayúsculas en los nombres de archivo de los módulos de política.

3. Abra los módulos de política en un editor de texto de su elección e inserte las opciones que modifican la política criptográfica de todo el sistema, por ejemplo:

```
# vi MYCRYPTO1.pmod
```

```
sha1_in_certs = 0
min_rsa_size = 3072
```

```
# vi NO-AES128.pmod
```

```
cifrado = -AES-128-GCM -AES-128-CCM -AES-128-CTR -AES-128-CBC
```

4. Guarde los cambios en los archivos del módulo.

5. Aplique sus ajustes de política al nivel de política criptográfica de todo el sistema **DEFAULT**:

```
# update-crypto-policies --set DEFAULT:MYCRYPTO1:NO-AES128
```

6. Para que la configuración criptográfica sea efectiva para los servicios y aplicaciones que ya se están ejecutando, reinicie el sistema:

```
# reboot
```

Recursos adicionales

- Para más detalles, consulte la sección **Custom Policies** en la página de manual **update-crypto-policies(8)** y la sección **Crypto Policy Definition Format** en la página de manual **crypto-policies(7)**.
- El artículo Cómo personalizar las políticas criptográficas en RHEL 8.2 proporciona ejemplos adicionales de cómo personalizar las políticas criptográficas de todo el sistema.

3.8. DESACTIVACIÓN DE SHA-1 MEDIANTE LA PERSONALIZACIÓN DE UNA POLÍTICA CRIPTOGRÁFICA PARA TODO EL SISTEMA

La función hash SHA-1 tiene un diseño inherentemente débil y el avance del criptoanálisis la ha hecho vulnerable a los ataques. Por defecto, RHEL 8 no utiliza SHA-1 pero algunas aplicaciones de terceros, por ejemplo las firmas públicas, siguen utilizando SHA-1. Para desactivar el uso de SHA-1 en los algoritmos de firma en su sistema, puede utilizar el módulo de políticas **NO-SHA1**.

NOTA

El módulo para desactivar SHA-1 está disponible desde RHEL 8.3. La personalización de las políticas criptográficas de todo el sistema está disponible desde RHEL 8.2.

Procedimiento

1. Aplique sus ajustes de política al nivel de política criptográfica de todo el sistema **DEFAULT**:

```
# update-crypto-policies --set DEFAULT:NO-SHA1
```

2. Para que la configuración criptográfica sea efectiva para los servicios y aplicaciones que ya se están ejecutando, reinicie el sistema:

```
# reboot
```

Recursos adicionales

- Para más detalles, consulte la sección **Custom Policies** en la página de manual **update-crypto-policies(8)** y la sección **Crypto Policy Definition Format** en la página de manual **crypto-policies(7)**.
- La entrada del blog Cómo personalizar las políticas criptográficas en RHEL 8.2 proporciona ejemplos adicionales de cómo personalizar las políticas criptográficas de todo el sistema.

3.9. CREACIÓN Y CONFIGURACIÓN DE UNA POLÍTICA CRIPTOGRÁFICA PERSONALIZADA PARA TODO EL SISTEMA

Los siguientes pasos demuestran la personalización de las políticas criptográficas de todo el sistema mediante un archivo de políticas completo.

NOTA

La personalización de las políticas criptográficas de todo el sistema está disponible desde RHEL 8.2.

Procedimiento

1. Cree un archivo de políticas para sus personalizaciones:

```
# cd /etc/crypto-policies/policies/
# touch MYPOLICY.pol
```

   También puede empezar copiando uno de los cuatro niveles de política predefinidos:

```
# cp /usr/share/crypto-policies/policies/DEFAULT.pol /etc/crypto-
policies/policies/MYPOLICY.pol
```

2. Edite el archivo con su política criptográfica personalizada en un editor de texto de su elección para que se ajuste a sus necesidades, por ejemplo:

```
# vi /etc/crypto-policies/policies/MYPOLICY.pol
```

3. Cambie la política criptográfica de todo el sistema a su nivel personalizado:

```
# update-crypto-policies --set MYPOLICY
```

4. Para que la configuración criptográfica sea efectiva para los servicios y aplicaciones que ya se están ejecutando, reinicie el sistema:

```
# reboot
```

Recursos adicionales

- Para más detalles, consulte la sección **Custom Policies** en la página de manual **update-crypto-policies(8)** y la sección **Crypto Policy Definition Format** en la página de manual **crypto-policies(7)**.
- El artículo Cómo personalizar las políticas criptográficas en RHEL 8.2 proporciona ejemplos adicionales de cómo personalizar las políticas criptográficas de todo el sistema.

3.10. INFORMACIÓN RELACIONADA

- Consulte los artículos de la base de conocimientos Políticas de cifrado en todo el sistema en RHEL 8 y Valores predeterminados de cifrado fuerte en RHEL 8 y eliminación de algoritmos de cifrado débiles en el Portal del cliente de Red Hat para obtener más información.

CAPÍTULO 4. CONFIGURACIÓN DE APLICACIONES PARA UTILIZAR HARDWARE CRIPTOGRÁFICO A TRAVÉS DE PKCS #11

Separar partes de su información secreta en dispositivos criptográficos dedicados, tales como tarjetas inteligentes y tokens criptográficos para la autenticación del usuario final y módulos de seguridad de hardware (HSM) para aplicaciones de servidor, proporciona una capa adicional de seguridad. En Red Hat Enterprise Linux 8, el soporte para el hardware criptográfico a través de la API PKCS #11 es consistente en las diferentes aplicaciones, y el aislamiento de los secretos en el hardware criptográfico no es una tarea complicada.

4.1. SOPORTE DE HARDWARE CRIPTOGRÁFICO A TRAVÉS DE PKCS #11

El PKCS #11 (Public-Key Cryptography Standard) define una interfaz de programación de aplicaciones (API) para los dispositivos criptográficos que contienen información criptográfica y realizan funciones criptográficas. Estos dispositivos se denominan tokens, y pueden implementarse en forma de hardware o software.

Un token PKCS #11 puede almacenar varios tipos de objetos, como un certificado, un objeto de datos y una clave pública, privada o secreta. Estos objetos son identificables de forma única a través del esquema PKCS #11 URI.

Un URI PKCS #11 es una forma estándar de identificar un objeto específico en un módulo PKCS #11 según los atributos del objeto. Esto permite configurar todas las bibliotecas y aplicaciones con la misma cadena de configuración en forma de URI.

Red Hat Enterprise Linux 8 proporciona por defecto el controlador OpenSC PKCS #11 para tarjetas inteligentes. Sin embargo, los tokens de hardware y los HSMs pueden tener sus propios módulos PKCS #11 que no tienen su contraparte en Red Hat Enterprise Linux. Puede registrar tales módulos PKCS #11 con la herramienta **p11-kit**, que actúa como una envoltura sobre los controladores de tarjetas inteligentes registrados en el sistema.

Para que su propio módulo PKCS #11 funcione en el sistema, añada un nuevo archivo de texto al directorio **/etc/pkcs11/modules/**

Puede añadir su propio módulo PKCS #11 en el sistema creando un nuevo archivo de texto en el directorio **/etc/pkcs11/modules/**. Por ejemplo, el archivo de configuración de OpenSC en **p11-kit** tiene el siguiente aspecto:

```
$ cat /usr/share/p11-kit/modules/opensc.module
module: opensc-pkcs11.so
```

Recursos adicionales

- Soporte consistente de PKCS #11 en Red Hat Enterprise Linux 8
- El esquema PKCS #11 URI
- Control de acceso a las tarjetas inteligentes

4.2. USO DE CLAVES SSH ALMACENADAS EN UNA TARJETA INTELIGENTE

Red Hat Enterprise Linux 8 le permite utilizar claves RSA y ECDSA almacenadas en una tarjeta inteligente en clientes OpenSSH. Use este procedimiento para habilitar la autenticación usando una tarjeta inteligente en lugar de usar una contraseña.

Requisitos previos

- En el lado del cliente, el paquete **opensc** está instalado y el servicio **pcscd** está funcionando.

Procedimiento

1. Enumerar todas las claves proporcionadas por el módulo PKCS #11 de OpenSC incluyendo sus URIs PKCS #11 y guardar el resultado en el archivo *keys.pub*:

```
$ ssh-keygen -D pkcs11: > keys.pub
$ ssh-keygen -D pkcs11:
ssh-rsa AAAAB3NzaC1yc2E...KKZMzcQZzx
pkcs11:id=%02;object=SIGN%20pubkey;token=SSH%20key;manufacturer=piv_II?module-
path=/usr/lib64/pkcs11/opensc-pkcs11.so
ecdsa-sha2-nistp256 AAA...J0hkYnnsM=
pkcs11:id=%01;object=PIV%20AUTH%20pubkey;token=SSH%20key;manufacturer=piv_II?
module-path=/usr/lib64/pkcs11/opensc-pkcs11.so
```

2. Para habilitar la autenticación mediante una tarjeta inteligente en un servidor remoto (*example.com*), transfiera la clave pública al servidor remoto. Utilice el comando **ssh-copy-id** con *keys.pub* creado en el paso anterior:

```
$ ssh-copy-id -f -i keys.pub username@example.com
```

3. Para conectarse a *example.com* utilizando la clave ECDSA de la salida del comando **ssh-keygen -D** en el paso 1, puede utilizar sólo un subconjunto de la URI, que hace referencia a su clave de forma exclusiva, por ejemplo:

```
$ ssh -i "pkcs11:id=%01?module-path=/usr/lib64/pkcs11/opensc-pkcs11.so" example.com
Enter PIN for 'SSH key':
[example.com] $
```

4. Puede utilizar la misma cadena URI en el archivo **~/.ssh/config** para que la configuración sea permanente:

```
$ cat ~/.ssh/config
IdentityFile "pkcs11:id=%01?module-path=/usr/lib64/pkcs11/opensc-pkcs11.so"
$ ssh example.com
Enter PIN for 'SSH key':
[example.com] $
```

Dado que OpenSSH utiliza el wrapper **p11-kit-proxy** y el módulo PKCS #11 de OpenSC está registrado en PKCS#11 Kit, puede simplificar los comandos anteriores:

```
$ ssh -i "pkcs11:id=%01" example.com
Enter PIN for 'SSH key':
[example.com] $
```

Si se omite la parte **id=** de un URI PKCS #11, OpenSSH carga todas las claves que están disponibles en el módulo proxy. Esto puede reducir la cantidad de escritura requerida:

```
$ ssh -i pkcs11: example.com
Enter PIN for 'SSH key':
[example.com] $
```

Recursos adicionales

- Fedora 28: Mejor soporte para tarjetas inteligentes en OpenSSH
- **p11-kit(8)** página de manual
- **ssh(1)** página de manual
- **ssh-keygen(1)** página de manual
- **opensc.conf(5)** página de manual
- **pcscd(8)** página de manual

4.3. USO DE HSM PARA PROTEGER LAS CLAVES PRIVADAS EN APACHE Y NGINX

Los servidores HTTP **Apache** y **Nginx** pueden trabajar con claves privadas almacenadas en módulos de seguridad de hardware (HSM), lo que ayuda a evitar la divulgación de las claves y los ataques de intermediario. Tenga en cuenta que esto suele requerir HSMs de alto rendimiento para los servidores ocupados.

Apache Servidor HTTP

Para la comunicación segura en forma de protocolo HTTPS, el servidor HTTP **Apache** (**httpd**) utiliza la biblioteca OpenSSL. OpenSSL no soporta PKCS #11 de forma nativa. Para utilizar los HSM, tiene que instalar el paquete **openssl-pkcs11**, que proporciona acceso a los módulos PKCS #11 a través de la interfaz del motor. Puede utilizar un URI PKCS #11 en lugar de un nombre de archivo normal para especificar una clave de servidor y un certificado en el archivo de configuración **/etc/httpd/conf.d/ssl.conf**, por ejemplo:

```
SSLCertificateFile    "pkcs11:id=%01;token=softhsm;type=cert"
SSLCertificateKeyFile "pkcs11:id=%01;token=softhsm;type=private?pin-value=111111"
```

Instale el paquete **httpd-manual** para obtener la documentación completa del servidor HTTP **Apache**, incluida la configuración de TLS. Las directivas disponibles en el archivo de configuración **/etc/httpd/conf.d/ssl.conf** se describen en detalle en /usr/share/httpd/manual/mod/mod_ssl.html.

Nginx Servidor HTTP y proxy

Dado que **Nginx** también utiliza OpenSSL para las operaciones criptográficas, el soporte para PKCS #11 debe pasar por el motor de **openssl-pkcs11**. **Nginx** actualmente sólo soporta la carga de claves privadas desde un HSM, y un certificado debe ser proporcionado por separado como un archivo regular.

Modifique las opciones **ssl_certificate** y **ssl_certificate_key** en la sección **server** del archivo de configuración **/etc/nginx/nginx.conf**:

```
ssl_certificate     /path/to/cert.pem
ssl_certificate_key "engine:pkcs11:pkcs11:token=softhsm;id=%01;type=private?pin-value=111111";
```

Tenga en cuenta que el prefijo **engine:pkcs11:** es necesario para el URI PKCS #11 en el archivo de configuración **Nginx**. Esto se debe a que el otro prefijo **pkcs11** se refiere al nombre del motor.

4.4. CONFIGURAR LAS APLICACIONES PARA QUE SE AUTENTIFIQUEN MEDIANTE CERTIFICADOS DE TARJETAS INTELIGENTES

- El descargador de red **wget** le permite especificar URIs PKCS #11 en lugar de rutas a claves privadas almacenadas localmente, y así simplifica la creación de scripts para tareas que requieren claves privadas y certificados almacenados de forma segura. Por ejemplo:

  ```
  $ wget --private-key 'pkcs11:token=softhsm;id=;type=private?pin-value=111111' --certificate
  'pkcs11:token=softhsm;id=;type=cert' https://example.com/
  ```

  Consulte la página de manual **wget(1)** para obtener más información.

- La especificación de la URI PKCS #11 para su uso por la herramienta **curl** es análoga:

  ```
  $ curl --key 'pkcs11:token=softhsm;id=;type=private?pin-value=111111' --cert
  'pkcs11:token=softhsm;id=;type=cert' https://example.com/
  ```

  Consulte la página de manual **curl(1)** para obtener más información.

- El navegador web **Firefox** carga automáticamente el módulo **p11-kit-proxy**. Esto significa que se detectan automáticamente todas las tarjetas inteligentes compatibles con el sistema. Para utilizar la autenticación de cliente TLS, no se requiere ninguna configuración adicional y las claves de una tarjeta inteligente se utilizan automáticamente cuando un servidor las solicita.

Uso de URIs PKCS #11 en aplicaciones personalizadas

Si su aplicación utiliza la biblioteca **GnuTLS** o **NSS**, la compatibilidad con los URIs PKCS #11 está garantizada gracias a su soporte incorporado para PKCS #11. Además, las aplicaciones que dependen de la biblioteca **OpenSSL** pueden acceder a módulos de hardware criptográfico gracias al motor **openssl-pkcs11**.

En el caso de aplicaciones que requieran trabajar con claves privadas en tarjetas inteligentes y que no utilicen **NSS**, **GnuTLS**, o **OpenSSL**, utilice **p11-kit** para implementar el registro de módulos PKCS #11.

Consulte la página de manual **p11-kit(8)** para obtener más información.

4.5. INFORMACIÓN RELACIONADA

- **pkcs11.conf(5)** página de manual

CAPÍTULO 5. USO DE CERTIFICADOS DE SISTEMA COMPARTIDOS

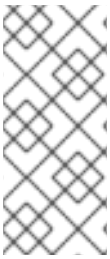
El almacenamiento compartido de certificados del sistema permite a NSS, GnuTLS, OpenSSL y Java compartir una fuente por defecto para recuperar anclas de certificados del sistema e información de la lista de bloques. Por defecto, el almacén de confianza contiene la lista de CA de Mozilla, incluyendo la confianza positiva y negativa. El sistema permite actualizar la lista de CA de Mozilla principal o elegir otra lista de certificados.

5.1. EL ALMACÉN DE CONFIANZA DE TODO EL SISTEMA

En Red Hat Enterprise Linux, el almacén de confianza consolidado de todo el sistema se encuentra en los directorios **/etc/pki/ca-trust/** y **/usr/share/pki/ca-trust-source/**. Las configuraciones de confianza en **/usr/share/pki/ca-trust-source/** son procesadas con menor prioridad que las configuraciones en **/etc/pki/ca-trust/**.

Los archivos de certificados se tratan en función del subdirectorio en el que se instalan en los siguientes directorios:

- para los anclajes de confianza
  - **/usr/share/pki/ca-trust-source/anchors/** o
  - **/etc/pki/ca-trust/source/anchors/**
- para los certificados de desconfianza
  - **/usr/share/pki/ca-trust-source/blacklist/** o
  - **/etc/pki/ca-trust/source/blacklist/**
- para certificados en el formato de archivo BEGIN TRUSTED ampliado
  - **/usr/share/pki/ca-trust-source/** o
  - **/etc/pki/ca-trust/source/**



NOTA

En un sistema criptográfico jerárquico, un ancla de confianza es una entidad autorizada que otras partes consideran digna de confianza. En la arquitectura X.509, un certificado raíz es un ancla de confianza de la que se deriva una cadena de confianza. Para permitir la validación de la cadena, la parte que confía debe tener acceso primero al ancla de confianza.

5.2. AÑADIR NUEVOS CERTIFICADOS

Para reconocer las aplicaciones de su sistema con una nueva fuente de confianza, añada el certificado correspondiente al almacén de todo el sistema y utilice el comando **update-ca-trust**.

Requisitos previos

- El paquete **ca-certificates** está presente en el sistema.

Procedimiento

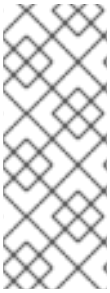
Procedimiento

1. Para añadir un certificado en los formatos de archivo PEM o DER simples a la lista de CAs de confianza del sistema, copie el archivo de certificado en el directorio **/usr/share/pki/ca-trust-source/anchors/** o **/etc/pki/ca-trust/source/anchors/**, por ejemplo:

```
# cp ~/certificate-trust-examples/Cert-trust-test-ca.pem /usr/share/pki/ca-trust-source/anchors/
```

2. Para actualizar la configuración del almacén de confianza de todo el sistema, utilice el comando **update-ca-trust**:

```
# update-ca-trust
```



NOTA

Aunque el navegador Firefox es capaz de utilizar un certificado añadido sin ejecutar **update-ca-trust**, Red Hat recomienda utilizar el comando **update-ca-trust** después de un cambio de CA. También tenga en cuenta que los navegadores, como Firefox, Epiphany o Chromium, guardan archivos en caché, y es posible que tenga que borrar la caché del navegador o reiniciar su navegador para cargar la configuración actual de los certificados del sistema.

5.3. GESTIÓN DE CERTIFICADOS DE SISTEMAS DE CONFIANZA

El comando **trust** proporciona una forma cómoda de gestionar los certificados en el almacén de confianza compartido de todo el sistema.

- Para listar, extraer, añadir, eliminar o cambiar las anclas de confianza, utilice el comando **trust**. Para ver la ayuda integrada de este comando, introdúzcalo sin argumentos o con la directiva **--help**:

```
$ trust
usage: trust command <args>...

Common trust commands are:
  list            List trust or certificates
  extract         Extract certificates and trust
  extract-compat  Extract trust compatibility bundles
  anchor          Add, remove, change trust anchors
  dump            Dump trust objects in internal format

See 'trust <command> --help' for more information
```

- Para listar todos los anclajes de confianza del sistema y los certificados, utilice el comando **trust list**:

```
$ trust list
pkcs11:id=%d2%87%b4%e3%df%37%27%93%55%f6%56%ea%81%e5%36%cc%8c%1e%3f%bd;type=cert
    type: certificate
    label: ACCVRAIZ1
    trust: anchor
    category: authority
```

```
pkcs11:id=%a6%b3%e1%2b%2b%49%b6%d7%73%a1%aa%94%f5%01%e7%73%65%4c%
ac%50;type=cert
    type: certificate
    label: ACEDICOM Root
    trust: anchor
    category: authority
...
```

- Para almacenar un ancla de confianza en el almacén de confianza de todo el sistema, utilice el subcomando **trust anchor** y especifique una ruta a un certificado. Sustituya *path.to/certificate.crt* por una ruta a su certificado y su nombre de archivo:

```
# trust anchor path.to/certificate.crt
```

- Para eliminar un certificado, utilice una ruta de acceso a un certificado o un ID de un certificado:

```
# trust anchor --remove path.to/certificate.crt
# trust anchor --remove "pkcs11:id=%AA%BB%CC%DD%EE;type=cert"
```

Recursos adicionales

- Todos los subcomandos de los comandos de **trust** ofrecen una ayuda integrada detallada, por ejemplo:

```
$ trust list --help
usage: trust list --filter=<what>

  --filter=<what>     filter of what to export
                      ca-anchors         certificate anchors
...
  --purpose=<usage>   limit to certificates usable for the purpose
                      server-auth        for authenticating servers
...
```

5.4. RECURSOS ADICIONALES

Para más información, consulte las siguientes páginas de manual:

- **update-ca-trust(8)**
- **trust(1)**

CAPÍTULO 6. ESCANEAR EL SISTEMA PARA COMPROBAR EL CUMPLIMIENTO DE LA CONFIGURACIÓN Y LAS VULNERABILIDADES

Una auditoría de cumplimiento es un proceso para determinar si un objeto dado sigue todas las reglas especificadas en una política de cumplimiento. La política de cumplimiento la definen los profesionales de la seguridad que especifican las configuraciones necesarias, a menudo en forma de lista de comprobación, que debe utilizar un entorno informático.

Las políticas de cumplimiento pueden variar sustancialmente entre organizaciones e incluso entre diferentes sistemas dentro de la misma organización. Las diferencias entre estas políticas se basan en el propósito de cada sistema y su importancia para la organización. Las configuraciones de software personalizadas y las características de despliegue también plantean la necesidad de contar con listas de comprobación de políticas personalizadas.

6.1. HERRAMIENTAS DE CUMPLIMIENTO DE LA CONFIGURACIÓN EN RHEL

Red Hat Enterprise Linux proporciona herramientas que le permiten realizar una auditoría de cumplimiento totalmente automatizada. Estas herramientas se basan en el estándar Security Content Automation Protocol (SCAP) y están diseñadas para la adaptación automatizada de las políticas de cumplimiento.

- **SCAP Workbench** - La utilidad gráfica **scap-workbench** está diseñada para realizar escaneos de configuración y vulnerabilidad en un solo sistema local o remoto. También puede utilizarla para generar informes de seguridad basados en estos escaneos y evaluaciones.

- **OpenSCAP** - La biblioteca **OpenSCAP**, con la utilidad de línea de comandos que la acompaña **oscap**, está diseñada para realizar escaneos de configuración y vulnerabilidad en un sistema local, para validar el contenido de cumplimiento de la configuración y para generar informes y guías basados en estos escaneos y evaluaciones.

- **SCAP Security Guide (SSG)** - El paquete **scap-security-guide** proporciona la última colección de políticas de seguridad para sistemas Linux. La guía consiste en un catálogo de consejos prácticos de endurecimiento, vinculados a los requisitos del gobierno cuando sea aplicable. El proyecto tiende un puente entre los requisitos políticos generalizados y las directrices de aplicación específicas.

- **Script Check Engine (SCE)** - SCE es una extensión del protocolo SCAP que permite a los administradores escribir su contenido de seguridad utilizando un lenguaje de scripting, como Bash, Python y Ruby. La extensión SCE se proporciona en el paquete **openscap-engine-sce**. El SCE en sí no forma parte del estándar SCAP.

Para realizar auditorías de cumplimiento automatizadas en múltiples sistemas de forma remota, puede utilizar la solución OpenSCAP para Red Hat Satellite.

Recursos adicionales

- **oscap(8)** - La página del manual de la utilidad de la línea de comandos **oscap** ofrece una lista completa de las opciones disponibles y explicaciones sobre su uso.

- Demostraciones de seguridad de Red Hat: Creación de contenido de políticas de seguridad personalizadas para automatizar el cumplimiento de la seguridad - Un laboratorio práctico para obtener una experiencia inicial en la automatización del cumplimiento de la seguridad utilizando

las herramientas que se incluyen en Red Hat Enterprise Linux para cumplir tanto con las políticas de seguridad estándar del sector como con las políticas de seguridad personalizadas. Si desea formación o acceso a estos ejercicios de laboratorio para su equipo, póngase en contacto con su equipo de cuentas de Red Hat para obtener más detalles.

- Demostraciones de seguridad de Red Hat: Defiéndase con las tecnologías de seguridad de RHEL - Un laboratorio práctico para aprender a implementar la seguridad en todos los niveles de su sistema RHEL, utilizando las principales tecnologías de seguridad disponibles en Red Hat Enterprise Linux, incluyendo OpenSCAP. Si desea formación o acceso a estos ejercicios de laboratorio para su equipo, póngase en contacto con su equipo de cuentas de Red Hat para obtener más detalles.

- **scap-workbench(8)** - La página del manual de la aplicación **SCAP Workbench** proporciona una información básica sobre la aplicación, así como algunos enlaces a posibles fuentes de contenido SCAP.

- **scap-security-guide(8)** - La página del manual del proyecto **scap-security-guide** proporciona más documentación sobre los distintos perfiles de seguridad SCAP disponibles. También se proporcionan ejemplos de cómo utilizar los puntos de referencia proporcionados utilizando la utilidad OpenSCAP.

- Para más detalles sobre el uso de OpenSCAP con Red Hat Satellite, consulte Gestión del cumplimiento de la seguridad en el Manual de administración de Red Hat Satellite.

6.2. EXPLORACIÓN DE LA VULNERABILIDAD

6.2.1. Avisos de seguridad de Red Hat Alimentación de OVAL

Las capacidades de auditoría de seguridad de Red Hat Enterprise Linux se basan en el estándar Security Content Automation Protocol (SCAP). SCAP es un marco de especificaciones polivalente que admite la configuración automatizada, la comprobación de vulnerabilidades y parches, las actividades de cumplimiento de controles técnicos y la medición de la seguridad.

Las especificaciones SCAP crean un ecosistema en el que el formato del contenido de seguridad es bien conocido y estandarizado, aunque la implementación del escáner o del editor de políticas no es obligatoria. Esto permite a las organizaciones construir su política de seguridad (contenido SCAP) una vez, sin importar cuántos proveedores de seguridad empleen.

El Lenguaje Abierto de Evaluación de Vulnerabilidades (OVAL) es el componente esencial y más antiguo de SCAP. A diferencia de otras herramientas y scripts personalizados, OVAL describe un estado requerido de los recursos de manera declarativa. El código de OVAL nunca se ejecuta directamente, sino que se utiliza una herramienta de interpretación de OVAL llamada escáner. La naturaleza declarativa de OVAL asegura que el estado del sistema evaluado no sea modificado accidentalmente.

Como todos los demás componentes de SCAP, OVAL se basa en XML. El estándar SCAP define varios formatos de documentos. Cada uno de ellos incluye un tipo de información diferente y sirve para un propósito distinto.

La Seguridad de Productos de Red Hat ayuda a los clientes a evaluar y gestionar los riesgos mediante el seguimiento y la investigación de todos los problemas de seguridad que afectan a los clientes de Red Hat. Proporciona parches y avisos de seguridad oportunos y concisos en el Portal del Cliente de Red Hat. Red Hat crea y soporta definiciones de parches OVAL, proporcionando versiones legibles por máquina de nuestros avisos de seguridad.

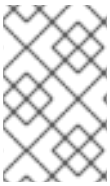
Debido a las diferencias entre plataformas, versiones y otros factores, las calificaciones cualitativas de gravedad de las vulnerabilidades de Red Hat Product Security no se alinean directamente con las

calificaciones de referencia del Sistema Común de Puntuación de Vulnerabilidades (CVSS) proporcionadas por terceros. Por lo tanto, le recomendamos que utilice las definiciones de RHSA OVAL en lugar de las proporcionadas por terceros.

Las definiciones de RHSA OVAL están disponibles individualmente y como un paquete completo, y se actualizan una hora después de que un nuevo aviso de seguridad esté disponible en el Portal del Cliente de Red Hat.

Cada definición de parche de OVAL se corresponde con un aviso de seguridad de Red Hat (RHSA). Debido a que un RHSA puede contener correcciones para múltiples vulnerabilidades, cada vulnerabilidad es listada por separado por su nombre de Vulnerabilidades y Exposiciones Comunes (CVE) y tiene un enlace a su entrada en nuestra base de datos pública de errores.

Las definiciones de RHSA OVAL están diseñadas para buscar versiones vulnerables de los paquetes RPM instalados en un sistema. Es posible ampliar estas definiciones para incluir más comprobaciones, por ejemplo, para averiguar si los paquetes se están utilizando en una configuración vulnerable. Estas definiciones están diseñadas para cubrir el software y las actualizaciones enviadas por Red Hat. Se requieren definiciones adicionales para detectar el estado de los parches del software de terceros.



NOTA

Para analizar los contenedores o las imágenes de contenedores en busca de vulnerabilidades de seguridad, consulte Análisis de contenedores e imágenes de contenedores en busca de vulnerabilidades.

Recursos adicionales

- Compatibilidad con Red Hat y OVAL
- Compatibilidad con Red Hat y CVE
- Notificaciones y avisos en el resumen de seguridad del producto
- Métricas de datos de seguridad
- Análisis de contenedores e imágenes de contenedores en busca de vulnerabilidades

6.2.2. Análisis del sistema en busca de vulnerabilidades

La utilidad de línea de comandos **oscap** permite escanear sistemas locales, validar el contenido de cumplimiento de la configuración y generar informes y guías basados en estos escaneos y evaluaciones. Esta utilidad sirve como front-end de la biblioteca OpenSCAP y agrupa sus funcionalidades en módulos (subcomandos) basados en el tipo de contenido SCAP que procesa.

Requisitos previos

- El repositorio **AppStream** está activado.

Procedimiento

1. Instale los paquetes **openscap-scanner** y **bzip2**:

```
# yum install openscap-scanner bzip2
```

2. Descargue las últimas definiciones de RHSA OVAL para su sistema:

```
# wget -O - https://www.redhat.com/security/data/oval/v2/RHEL8/rhel-8.oval.xml.bz2 | bzip2 -
-decompress > rhel-8.oval.xml
```

3. Analice el sistema en busca de vulnerabilidades y guarde los resultados en el archivo *vulnerability.html*:

```
# oscap oval eval --report vulnerability.html rhel-8.oval.xml
```

Pasos de verificación

1. Compruebe los resultados en un navegador de su elección, por ejemplo:

```
$ firefox vulnerability.html &
```

Recursos adicionales

- La página de manual **oscap(8)**.
- La lista de definiciones de Red Hat OVAL .

6.2.3. Análisis de sistemas remotos en busca de vulnerabilidades

También puede comprobar las vulnerabilidades de los sistemas remotos con el escáner OpenSCAP utilizando la herramienta **oscap-ssh** a través del protocolo SSH.

Requisitos previos

- El repositorio **AppStream** está activado.
- El paquete **openscap-scanner** está instalado en los sistemas remotos.
- El servidor SSH se está ejecutando en los sistemas remotos.

Procedimiento

1. Instale los paquetes **openscap-utils** y **bzip2**:

```
# yum install openscap-utils bzip2
```

2. Descargue las últimas definiciones de RHSA OVAL para su sistema:

```
# wget -O - https://www.redhat.com/security/data/oval/v2/RHEL8/rhel-8.oval.xml.bz2 | bzip2 -
-decompress > rhel-8.oval.xml
```

3. Analice un sistema remoto con el nombre de host *machine1*, SSH ejecutado en el puerto 22 y el nombre de usuario *joesec* en busca de vulnerabilidades y guarde los resultados en el archivo *remote-vulnerability.html*:

```
# oscap-ssh joesec@machine1 22 oval eval --report remote-vulnerability.html rhel-8.oval.xml
```

Recursos adicionales

- La página de manual **oscap-ssh(8)**.
- La lista de definiciones de Red Hat OVAL .

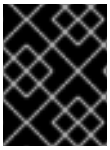
6.3. ESCANEO DEL CUMPLIMIENTO DE LA CONFIGURACIÓN

6.3.1. Cumplimiento de la configuración en RHEL 8

Puede utilizar el escaneo de cumplimiento de configuración para ajustarse a una línea de base definida por una organización específica. Por ejemplo, si trabaja con el gobierno de los Estados Unidos, puede que tenga que cumplir con el Perfil de Protección del Sistema Operativo (OSPP), y si es un procesador de pagos, puede que tenga que cumplir con el Estándar de Seguridad de Datos de la Industria de las Tarjetas de Pago (PCI-DSS). También puede realizar un análisis de cumplimiento de la configuración para reforzar la seguridad de su sistema.

Red Hat recomienda seguir el contenido del Protocolo de Automatización de Contenidos de Seguridad (SCAP) proporcionado en el paquete de la Guía de Seguridad SCAP porque está en línea con las mejores prácticas de Red Hat para los componentes afectados.

El paquete de la Guía de Seguridad SCAP proporciona contenido que se ajusta a los estándares SCAP 1.2 y SCAP 1.3. La utilidad **openscap scanner** es compatible con el contenido de SCAP 1.2 y SCAP 1.3 proporcionado en el paquete de la Guía de Seguridad SCAP.



IMPORTANTE

La realización de un escaneo de conformidad de la configuración no garantiza que el sistema sea conforme.

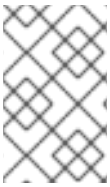
El conjunto de guías de seguridad SCAP proporciona perfiles para varias plataformas en forma de documentos de flujo de datos. Un flujo de datos es un archivo que contiene definiciones, puntos de referencia, perfiles y reglas individuales. Cada regla especifica la aplicabilidad y los requisitos de cumplimiento. RHEL 8 proporciona varios perfiles para el cumplimiento de las políticas de seguridad. Además del estándar de la industria, los flujos de datos de Red Hat también contienen información para remediar las reglas fallidas.

Estructura de los recursos de exploración de la conformidad

```
Data stream
    ├── xccdf
    |     ├── benchmark
    |           ├── profile
    |               ├──rule
    |                   ├── xccdf
    |                       ├── oval reference
    ├── oval                    ├── ocil reference
    ├── ocil                    ├── cpe reference
    └── cpe                     └── remediation
```

Un perfil es un conjunto de reglas basadas en una política de seguridad, como el Perfil de Protección del Sistema Operativo (OSPP) o el Estándar de Seguridad de Datos de la Industria de las Tarjetas de Pago (PCI-DSS). Esto permite auditar el sistema de forma automatizada para comprobar el cumplimiento de las normas de seguridad.

Puede modificar (adaptar) un perfil para personalizar ciertas reglas, por ejemplo, la longitud de la contraseña. Para obtener más información sobre la adaptación del perfil, consulte Personalizar un perfil de seguridad con SCAP Workbench.



NOTA

Para escanear contenedores o imágenes de contenedores en busca de cumplimiento de la configuración, consulte Escanear contenedores e imágenes de contenedores en busca de vulnerabilidades.

6.3.2. Posibles resultados de una exploración de OpenSCAP

Dependiendo de varias propiedades de su sistema y del flujo de datos y el perfil aplicado a una exploración de OpenSCAP, cada regla puede producir un resultado específico. Esta es una lista de posibles resultados con breves explicaciones de lo que significan.

Tabla 6.1. Posibles resultados de una exploración de OpenSCAP

Resultado	Explicación
Pasar	La exploración no encontró ningún conflicto con esta norma.
Falla	El escáner encontró un conflicto con esta norma.
No se ha comprobado	OpenSCAP no realiza una evaluación automática de esta regla. Compruebe manualmente si su sistema se ajusta a esta regla.
No se aplica	Esta regla no se aplica a la configuración actual.
No seleccionado	Esta regla no forma parte del perfil. OpenSCAP no evalúa esta regla y no muestra estas reglas en los resultados.
Error	El escaneo encontró un error. Para obtener información adicional, puede introducir el comando **oscap** con la opción**--verbose DEVEL**. Considere la posibilidad de abrir un informe de error.
Desconocido	El escaneo encontró una situación inesperada. Para obtener información adicional, puede introducir el comando **oscap** con la opción`**--verbose DEVEL**. Considere la posibilidad de abrir un informe de errores.

6.3.3. Visualización de perfiles para el cumplimiento de la configuración

Antes de decidirse a utilizar los perfiles para la exploración o la corrección, puede enumerarlos y comprobar sus descripciones detalladas mediante el subcomando **oscap info**.

Requisitos previos

- Los paquetes **openscap-scanner** y **scap-security-guide** están instalados.

Procedimiento

1. Lista de todos los archivos disponibles con perfiles de cumplimiento de seguridad proporcionados por el proyecto de la Guía de Seguridad SCAP:

```
$ ls /usr/share/xml/scap/ssg/content/
ssg-firefox-cpe-dictionary.xml  ssg-rhel6-ocil.xml
ssg-firefox-cpe-oval.xml        ssg-rhel6-oval.xml
...
ssg-rhel6-ds-1.2.xml          ssg-rhel8-oval.xml
ssg-rhel8-ds.xml              ssg-rhel8-xccdf.xml
...
```

2. Muestra información detallada sobre un flujo de datos seleccionado utilizando el subcomando **oscap info**. Los archivos XML que contienen flujos de datos se indican con la cadena **-ds** en sus nombres. En la sección **Profiles**, puede encontrar una lista de perfiles disponibles y sus identificaciones:

```
$ oscap info /usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
...
Profiles:
  Title: PCI-DSS v3.2.1 Control Baseline for Red Hat Enterprise Linux 8
    Id: xccdf_org.ssgproject.content_profile_pci-dss
  Title: OSPP - Protection Profile for General Purpose Operating Systems
    Id: xccdf_org.ssgproject.content_profile_ospp
...
```

3. Seleccionar un perfil del archivo de flujo de datos y mostrar detalles adicionales sobre el perfil seleccionado. Para ello, utilice **oscap info** con la opción **--profile** seguida de la última sección del ID mostrado en la salida del comando anterior. Por ejemplo, el ID del perfil PCI-DSS es: **xccdf_org.ssgproject.content_profile_pci-dss**, y el valor de la opción **--profile** es **pci-dss**:

```
$ oscap info --profile pci-dss /usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
...
Title: PCI-DSS v3.2.1 Control Baseline for Red Hat Enterprise Linux 8
Id: xccdf_org.ssgproject.content_profile_pci-dss

Description: Ensures PCI-DSS v3.2.1 security configuration settings are applied.
...
```

Recursos adicionales

- La página de manual **scap-security-guide(8)**.

6.3.4. Evaluar el cumplimiento de la configuración con una línea de base específica

Para determinar si su sistema se ajusta a una línea de base específica, siga estos pasos.

Requisitos previos

- Los paquetes **openscap-scanner** y **scap-security-guide** están instalados

- Usted conoce el ID del perfil dentro de la línea de base con el que el sistema debe cumplir. Para encontrar el ID, consulte Ver perfiles para el cumplimiento de la configuración.

Procedimiento

1. Evaluar la conformidad del sistema con el perfil seleccionado y guardar los resultados del escaneo en el archivo HTML report.html, por ejemplo:

```
$ sudo oscap xccdf eval --report report.html --profile ospp
/usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
```

2. Opcional: Escanee un sistema remoto con el nombre de host **machine1**, SSH ejecutándose en el puerto **22**, y el nombre de usuario **joesec** para comprobar la conformidad y guarde los resultados en el archivo **remote-report.html**:

```
$ oscap-ssh joesec@machine1 22 xccdf eval --report remote_report.html --profile ospp
/usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
```

Recursos adicionales

- **scap-security-guide(8)** página de manual
- La documentación de **SCAP Security Guide** instalada en el **file:///usr/share/doc/scap-security-guide/** directorio.
- El Manual de configuración segura de Red Hat Enterprise Linux 8 instalado con el paquete **scap-security-guide-doc**.

6.4. REMEDIAR EL SISTEMA PARA ALINEARLO CON UNA LÍNEA DE BASE ESPECÍFICA

Utilice este procedimiento para remediar el sistema RHEL 8 para alinearlo con una línea de base específica. Este ejemplo utiliza el perfil de protección para sistemas operativos de uso general (OSPP).

AVISO

Si no se utiliza con cuidado, la ejecución de la evaluación del sistema con la opción **Remediate** activada puede hacer que el sistema no funcione. Red Hat no proporciona ningún método automatizado para revertir los cambios realizados por las correcciones de seguridad. Las correcciones son compatibles con los sistemas RHEL en la configuración por defecto. Si su sistema ha sido alterado después de la instalación, la ejecución de la remediación podría hacer que no cumpla con el perfil de seguridad requerido.

Requisitos previos

- El paquete **scap-security-guide** está instalado en su sistema RHEL 8.

Procedimiento

1. Utilice el comando **oscap** con la opción **--remediate**:

```
$ sudo oscap xccdf eval --profile ospp --remediate /usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
```

2. Reinicie su sistema.

Paso de verificación

1. Evaluar la conformidad del sistema con el perfil OSPP, y guardar los resultados del escaneo en el archivo **ospp_report.html**:

```
$ oscap xccdf eval --report ospp_report.html --profile ospp
/usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
```

Recursos adicionales

- **scap-security-guide(8)** y **oscap(8)** páginas man

6.5. REMEDIAR EL SISTEMA PARA ALINEARLO CON UNA LÍNEA DE BASE ESPECÍFICA UTILIZANDO EL LIBRO DE JUGADAS DE SSG ANSIBLE

Utilice este procedimiento para remediar su sistema con una línea de base específica utilizando el archivo Ansible playbook del proyecto SCAP Security Guide. Este ejemplo utiliza el perfil de protección para sistemas operativos de uso general (OSPP).

AVISO

Si no se utiliza con cuidado, la ejecución de la evaluación del sistema con la opción **Remediate** activada puede hacer que el sistema no funcione. Red Hat no proporciona ningún método automatizado para revertir los cambios realizados por las correcciones de seguridad. Las correcciones son compatibles con los sistemas RHEL en la configuración por defecto. Si su sistema ha sido alterado después de la instalación, la ejecución de la remediación podría hacer que no cumpla con el perfil de seguridad requerido.

Requisitos previos

- El paquete **scap-security-guide** está instalado en su sistema RHEL 8.
- El paquete **ansible** está instalado. Consulte la Guía de instalación de Ansible para obtener más información.

Procedimiento

1. Remedie su sistema para alinearlo con OSPP usando Ansible:

```
# ansible-playbook -i localhost, -c local /usr/share/scap-security-guide/ansible/rhel8-playbook-ospp.yml
```

2. Reinicia el sistema.

Pasos de verificación

1. Evaluar la conformidad del sistema con el perfil OSPP, y guardar los resultados del escaneo en el archivo **ospp_report.html**:

```
# oscap xccdf eval --profile ospp --report ospp_report.html /usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
```

Recursos adicionales

- **scap-security-guide(8)** y **oscap(8)** páginas man
- Documentación de Ansible

6.6. CREACIÓN DE UN PLAYBOOK ANSIBLE DE REMEDIACIÓN PARA ALINEAR EL SISTEMA CON UNA LÍNEA DE BASE ESPECÍFICA

Utilice este procedimiento para crear un libro de jugadas de Ansible que contenga sólo las correcciones necesarias para alinear su sistema con una línea de base específica. Este ejemplo utiliza el perfil de protección para sistemas operativos de uso general (OSPP). Con este procedimiento, se crea un libro de jugadas más pequeño que no cubre los requisitos ya satisfechos. Siguiendo estos pasos, usted no modifica su sistema de ninguna manera, sólo prepara un archivo para su posterior aplicación.

Requisitos previos

- El paquete **scap-security-guide** está instalado en su sistema RHEL 8.

Procedimiento

1. Escanee el sistema y guarde los resultados:

```
# oscap xccdf eval --profile ospp --results ospp-results.xml /usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
```

2. Genere un playbook de Ansible basado en el archivo generado en el paso anterior:

```
# oscap xccdf generate fix --fix-type ansible --output ospp-remediations.yml ospp-results.xml
```

3. El archivo **ospp-remediations.yml** contiene las correcciones de Ansible para las reglas que fallaron durante el análisis realizado en el paso 1. Después de revisar este archivo generado, puede aplicarlo con el comando **ansible-playbook ospp-remediations.yml**.

Pasos de verificación

1. En un editor de texto de su elección, revise que el archivo **ospp-remediations.yml** contenga las reglas que fallaron en el análisis realizado en el paso 1.

Recursos adicionales

- **scap-security-guide(8)** y **oscap(8)** páginas man
- Documentación de Ansible

6.7. CREACIÓN DE UN SCRIPT BASH DE REMEDIACIÓN PARA UNA APLICACIÓN POSTERIOR

Utilice este procedimiento para crear un script Bash que contenga correcciones que alineen su sistema con un perfil de seguridad como PCI-DSS. Utilizando los siguientes pasos, no realiza ninguna modificación en su sistema, sólo prepara un archivo para su posterior aplicación.

Requisitos previos

- El paquete **scap-security-guide** está instalado en su sistema RHEL 8.

Procedimiento

1. Utilice el comando **oscap** para analizar el sistema y guardar los resultados en un archivo XML. En el siguiente ejemplo, **oscap** evalúa el sistema según el perfil **pci-dss**:

```
# oscap xccdf eval --profile pci-dss --results pci-dss-results.xml
/usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
```

2. Generar un script Bash basado en el archivo de resultados generado en el paso anterior:

```
# oscap xccdf generate fix --profile pci-dss --fix-type bash --output pci-dss-remediations.sh
pci-dss-results.xml
```

3. El archivo **pci-dss-remediations.sh** contiene remedios para las reglas que fallaron durante el análisis realizado en el paso 1. Después de revisar este archivo generado, puede aplicarlo con el comando **./pci-dss-remediations.sh** cuando se encuentre en el mismo directorio que este archivo.

Pasos de verificación

1. En un editor de texto de su elección, revise que el archivo **pci-dss-remediations.sh** contenga las reglas que fallaron en el análisis realizado en el paso 1.

Recursos adicionales

- **scap-security-guide(8)**, **oscap(8)**, y **bash(1)** páginas de manual

6.8. ESCANEAR EL SISTEMA CON UN PERFIL PERSONALIZADO UTILIZANDO SCAP WORKBENCH

SCAP Workbench, que está contenida en el paquete **scap-workbench**, es una utilidad gráfica que permite a los usuarios realizar escaneos de configuración y de vulnerabilidad en un solo sistema local o en uno remoto, realizar la corrección del sistema y generar informes basados en las evaluaciones de los

escaneos. Tenga en cuenta que **SCAP Workbench** tiene una funcionalidad limitada en comparación con la utilidad de línea de comandos **oscap**. **SCAP Workbench** procesa el contenido de seguridad en forma de archivos de flujo de datos.

6.8.1. Uso de SCAP Workbench para escanear y remediar el sistema

Para evaluar su sistema con respecto a la política de seguridad seleccionada, utilice el siguiente procedimiento.

Requisitos previos

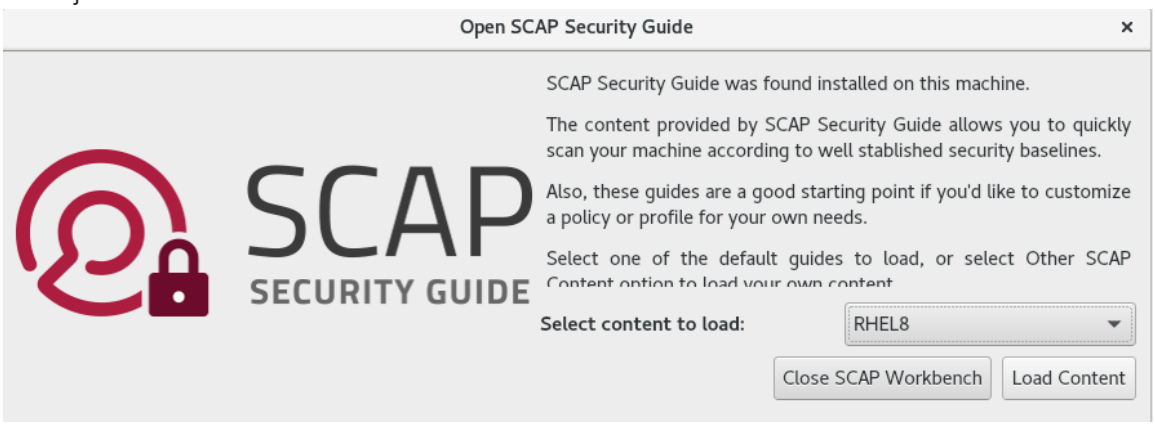
- El paquete **scap-workbench** está instalado en su sistema.

Procedimiento

1. Para ejecutar **SCAP Workbench** desde el entorno de escritorio **GNOME Classic**, pulse la tecla **Super** para entrar en **Activities Overview**, escriba **scap-workbench** y pulse **Enter**. Como alternativa, utilice:

```
$ scap-workbench &
```

2. Seleccione una política de seguridad utilizando las siguientes opciones:
   - **Load Content** botón en la ventana de inicio
   - **Open content from SCAP Security Guide**
   - **Open Other Content** en el menú **File**, y busque el respectivo archivo XCCDF, SCAP RPM o de flujo de datos.

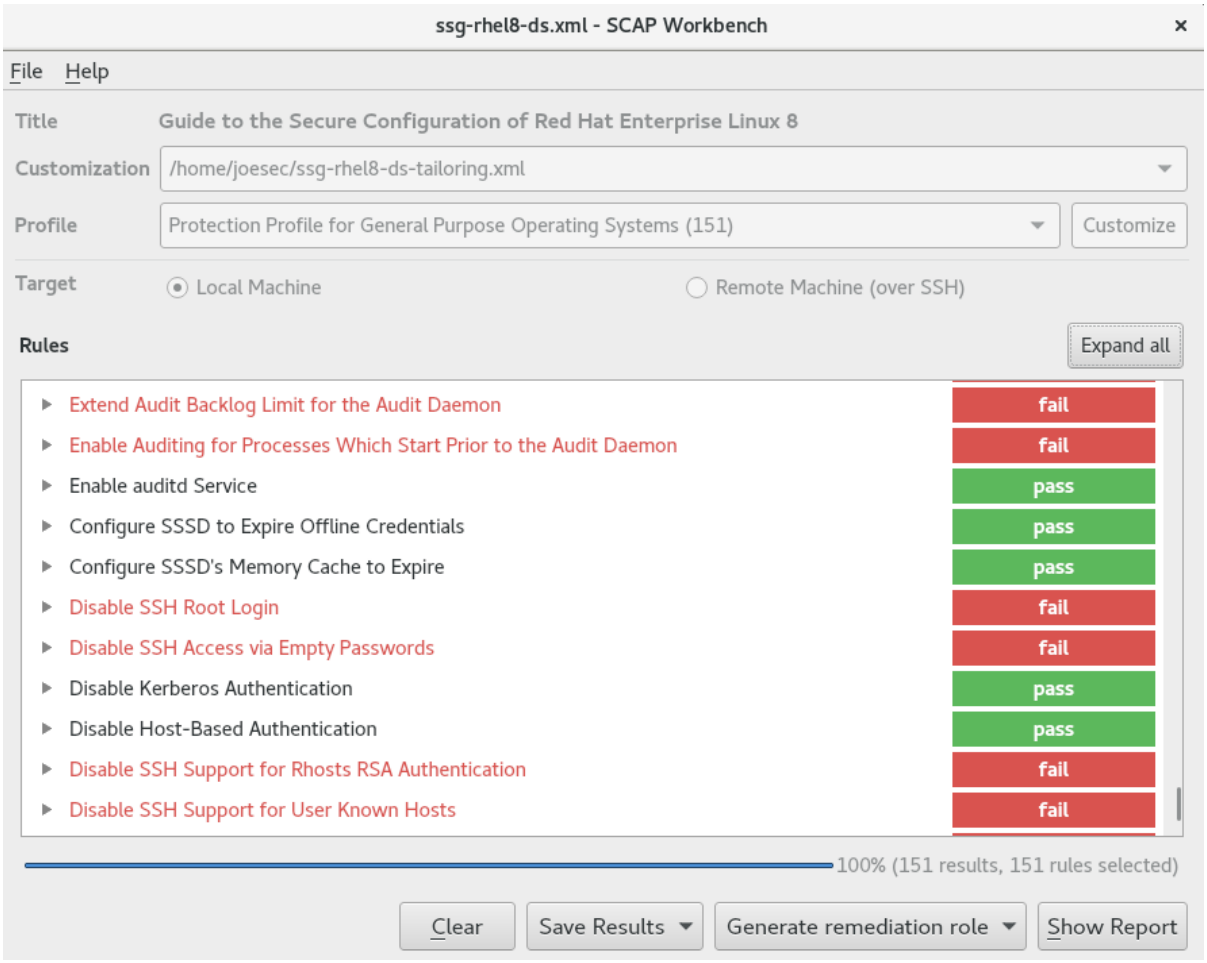


3. Puede permitir la corrección automática de la configuración del sistema seleccionando la casilla **Remediar**. Con esta opción activada, **SCAP Workbench** intenta cambiar la configuración del sistema de acuerdo con las reglas de seguridad aplicadas por la política. Este proceso debería corregir las comprobaciones relacionadas que fallan durante el análisis del sistema.

AVISO

Si no se utiliza con cuidado, la ejecución de la evaluación del sistema con la opción **Remediate** activada puede hacer que el sistema no funcione. Red Hat no proporciona ningún método automatizado para revertir los cambios realizados por las correcciones de seguridad. Las correcciones son compatibles con los sistemas RHEL en la configuración por defecto. Si su sistema ha sido alterado después de la instalación, la ejecución de la remediación podría hacer que no cumpla con el perfil de seguridad requerido.

4. Analice su sistema con el perfil seleccionado haciendo clic en el botón **Analizar**.



5. Para almacenar los resultados del escaneo en forma de archivo XCCDF, ARF o HTML, haga clic en el cuadro combinado **Guardar resultados**. Elija la opción **HTML Report** para generar el informe de escaneo en formato legible para el ser humano. Los formatos XCCDF y ARF (flujo de datos) son adecuados para su posterior procesamiento automático. Puede elegir repetidamente las tres opciones.

6. Para exportar las correcciones basadas en resultados a un archivo, utilice el menú emergente **Generar rol de corrección**.

6.8.2. Personalización de un perfil de seguridad con SCAP Workbench

Puede personalizar un perfil de seguridad cambiando los parámetros de ciertas reglas (por ejemplo, la longitud mínima de la contraseña), eliminando las reglas que cubra de manera diferente y seleccionando reglas adicionales, para implementar políticas internas. No se pueden definir nuevas reglas al personalizar un perfil.

El siguiente procedimiento demuestra el uso de **SCAP Workbench** para personalizar (adaptar) un perfil. También puede guardar el perfil adaptado para utilizarlo con la utilidad de línea de comandos **oscap**.

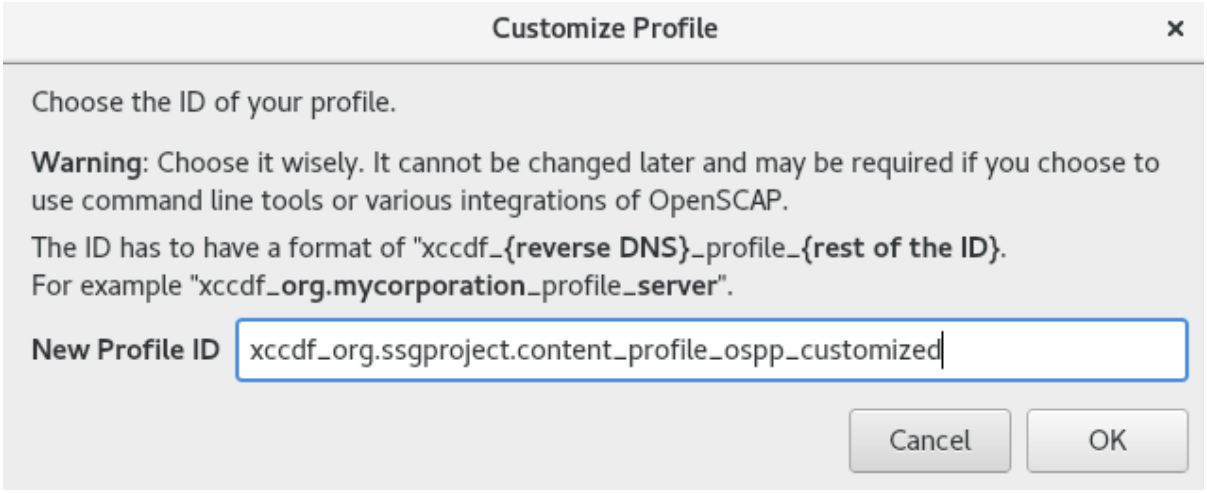
Requisitos previos

- El paquete **scap-workbench** está instalado en su sistema.

Procedimiento

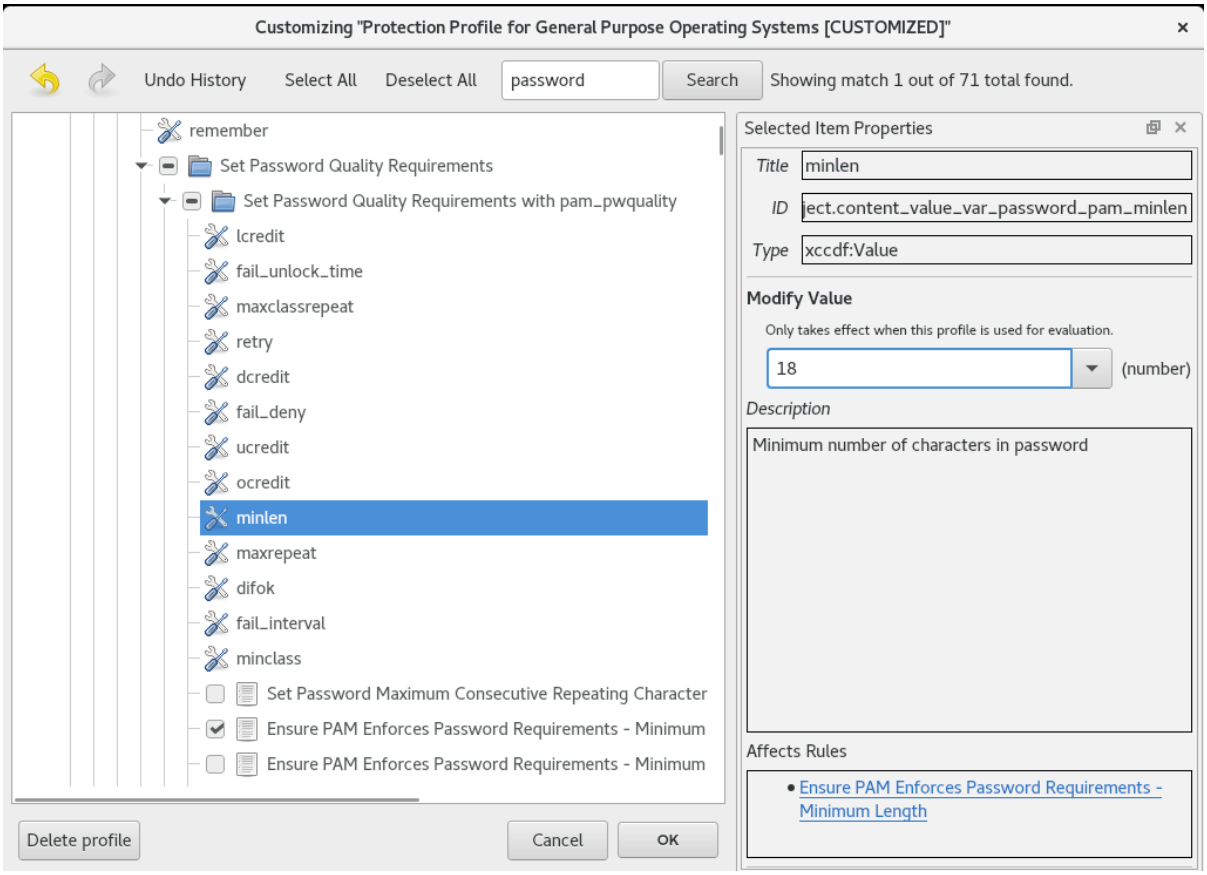
1. Ejecute **SCAP Workbench**, y seleccione el perfil a personalizar mediante **Open content from SCAP Security Guide** o **Open Other Content** en el menú **File**.

2. Para ajustar el perfil de seguridad seleccionado según sus necesidades, haga clic en el botón **Personalizar**.
   Esto abre la nueva ventana de personalización que le permite modificar el perfil actualmente seleccionado sin cambiar el archivo de flujo de datos original. Elija un nuevo ID de perfil.



3. Encuentre una regla para modificar utilizando la estructura de árbol con reglas organizadas en grupos lógicos o el campo de **búsqueda**.

4. Incluya o excluya reglas mediante casillas de verificación en la estructura de árbol, o modifique los valores de las reglas cuando corresponda.



5. Confirme los cambios haciendo clic en el botón **OK**.

6. Para almacenar los cambios de forma permanente, utilice una de las siguientes opciones:

   - Guarde un archivo de personalización por separado utilizando **Save Customization Only** en el menú **File**.

   - Guarde todo el contenido de seguridad a la vez en **Save All** en el menú **File**.
     Si selecciona la opción **Into a directory**, **SCAP Workbench** guarda tanto el archivo de flujo de datos como el archivo de personalización en la ubicación especificada. Puede utilizarlo como solución de copia de seguridad.

     Seleccionando la opción **As RPM**, puede ordenar a **SCAP Workbench** que cree un paquete RPM que contenga el archivo de flujo de datos y el archivo de personalización. Esto es útil para distribuir el contenido de seguridad a sistemas que no pueden ser escaneados remotamente, y para entregar el contenido para su posterior procesamiento.

NOTA

Dado que **SCAP Workbench** no admite correcciones basadas en resultados para perfiles adaptados, utilice las correcciones exportadas con la utilidad de línea de comandos **oscap**.

6.8.3. Información relacionada

- **scap-workbench(8)** página de manual
- Manual del usuario de SCAP Workbench
- Despliegue de políticas SCAP personalizadas con Satellite 6. x: un artículo de la Base de conocimientos sobre la adaptación de scripts

6.9. IMPLANTACIÓN DE SISTEMAS QUE CUMPLEN CON UN PERFIL DE SEGURIDAD INMEDIATAMENTE DESPUÉS DE UNA INSTALACIÓN

Puede utilizar el paquete OpenSCAP para desplegar sistemas RHEL que cumplan con un perfil de seguridad, como OSPP o PCI-DSS, inmediatamente después del proceso de instalación. Utilizando este método de despliegue, puede aplicar reglas específicas que no se pueden aplicar más tarde utilizando scripts de corrección, por ejemplo, una regla para la fuerza de la contraseña y la partición.

6.9.1. Implantación de sistemas RHEL compatibles con la línea de base mediante la instalación gráfica

Utilice este procedimiento para desplegar un sistema RHEL que esté alineado con una línea de base específica. Este ejemplo utiliza el perfil de protección para el sistema operativo de uso general (OSPP).

Requisitos previos

- Ha iniciado el programa de instalación **graphical**. Tenga en cuenta que el **OSCAP Anaconda Add-on** no admite la instalación de sólo texto.
- Ha accedido a la ventana **Installation Summary**.

Procedimiento

1. En la ventana **Installation Summary**, haga clic en **Software Selection**. Se abre la ventana **Software Selection**.
2. En el panel **Base Environment**, seleccione el entorno **Server**. Sólo puede seleccionar un entorno base.

AVISO

No utilice el entorno base de **Server with GUI** si desea desplegar un sistema compatible. Los perfiles de seguridad proporcionados como parte de **SCAP Security Guide** pueden no ser compatibles con el conjunto de paquetes extendidos de **Server with GUI**. Para más información, consulte, por ejemplo, BZ#1648162, BZ#1787156 o BZ#1816199.

3. Haga clic en **Done** para aplicar la configuración y volver a la ventana **Installation Summary**.
4. Haga clic en **Security Policy**. Se abre la ventana **Security Policy**.
5. Para habilitar las políticas de seguridad en el sistema, cambie el interruptor **Apply security policy** a **ON**.
6. Seleccione **Protection Profile for General Purpose Operating Systems** en el panel de perfiles.
7. Haga clic en **Select Profile** para confirmar la selección.

8. Confirme los cambios en el panel **Changes that were done or need to be done** que aparece en la parte inferior de la ventana. Complete los cambios manuales restantes.

9. Dado que OSPP tiene estrictos requisitos de partición que deben cumplirse, cree particiones separadas para **/boot**, **/home**, **/var**, **/var/log**, **/var/tmp** y **/var/log/audit**.

10. Completa el proceso de instalación gráfica.

NOTA

El programa de instalación gráfica crea automáticamente un archivo Kickstart correspondiente después de una instalación exitosa. Puede utilizar el archivo **/root/anaconda-ks.cfg** para instalar automáticamente sistemas compatibles con OSPP.

Pasos de verificación

1. Para comprobar el estado actual del sistema una vez finalizada la instalación, reinicie el sistema e inicie un nuevo análisis:

```
# oscap xccdf eval --profile ospp --report eval_postinstall_report.html
/usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
```

Recursos adicionales

- Para más detalles sobre la partición, consulte Configuración de la partición manual .

6.9.2. Implantación de sistemas RHEL compatibles con la línea de base mediante Kickstart

Utilice este procedimiento para desplegar sistemas RHEL que estén alineados con una línea de base específica. Este ejemplo utiliza el perfil de protección para el sistema operativo de uso general (OSPP).

Requisitos previos

- El paquete **scap-security-guide** está instalado en su sistema RHEL 8.

Procedimiento

1. Abra el archivo **/usr/share/scap-security-guide/kickstart/ssg-rhel8-ospp-ks.cfg** Kickstart en un editor de su elección.

2. Actualice el esquema de particiones para que se ajuste a sus requisitos de configuración. Para el cumplimiento de OSPP, las particiones separadas para **/boot**, **/home**, **/var**, **/var/log**, **/var/tmp**, y **/var/log/audit** deben ser preservadas, y sólo puede cambiar el tamaño de las particiones.

AVISO

Dado que el plugin **OSCAP Anaconda Addon** no admite la instalación de sólo texto, no utilice la opción **text** en su archivo Kickstart. Para más información, consulte RHBZ#1674001.

3. Inicie una instalación Kickstart como se describe en Realización de una instalación automatizada mediante Kickstart.

IMPORTANTE

Las contraseñas en forma de hash no pueden ser verificadas para los requisitos de OSPP.

Pasos de verificación

1. Para comprobar el estado actual del sistema una vez finalizada la instalación, reinicie el sistema e inicie un nuevo análisis:

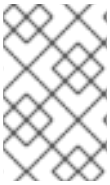
```
# oscap xccdf eval --profile ospp --report eval_postinstall_report.html
/usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
```

Recursos adicionales

- Para más detalles, consulte la página del proyecto OSCAP Anaconda Addon.

6.10. ESCANEO DE VULNERABILIDADES EN CONTENEDORES E IMÁGENES DE CONTENEDORES

Utilice este procedimiento para encontrar vulnerabilidades de seguridad en un contenedor o una imagen de contenedor.



NOTA

El comando **oscap-podman** está disponible a partir de RHEL 8.2. Para RHEL 8.1 y 8.0, utilice la solución descrita en el artículo de la base de conocimientos Using OpenSCAP for scanning containers in RHEL 8.

Requisitos previos

- El paquete **openscap-utils** está instalado.

Procedimiento

1. Descargue las últimas definiciones de RHSA OVAL para su sistema:

```
# wget -O - https://www.redhat.com/security/data/oval/v2/RHEL8/rhel-8.oval.xml.bz2 | bzip2 -
-decompress > rhel-8.oval.xml
```

2. Obtiene el ID de un contenedor o de una imagen de contenedor, por ejemplo:

```
# podman images
REPOSITORY                          TAG     IMAGE ID      CREATED      SIZE
registry.access.redhat.com/ubi8/ubi   latest  096cae65a207  7 weeks ago  239 MB
```

3. Analice el contenedor o la imagen del contenedor en busca de vulnerabilidades y guarde los resultados en el archivo *vulnerability.html*:

```
# oscap-podman 096cae65a207 oval eval --report vulnerability.html rhel-8.oval.xml
```

Tenga en cuenta que el comando **oscap-podman** requiere privilegios de root, y el ID de un contenedor es el primer argumento.

Pasos de verificación

1. Compruebe los resultados en un navegador de su elección, por ejemplo:

```
$ firefox vulnerability.html &
```

Recursos adicionales

- Para más información, consulte las páginas de manual **oscap-podman(8)** y **oscap(8)**.

6.11. EVALUACIÓN DEL CUMPLIMIENTO DE LA SEGURIDAD DE UN CONTENEDOR O UNA IMAGEN DE CONTENEDOR CON UNA LÍNEA DE BASE ESPECÍFICA

Siga estos pasos para evaluar el cumplimiento de su contenedor o de una imagen de contenedor con una línea base de seguridad específica, como el Perfil de Protección del Sistema Operativo (OSPP) o el Estándar de Seguridad de Datos de la Industria de las Tarjetas de Pago (PCI-DSS).

NOTA

El comando **oscap-podman** está disponible a partir de RHEL 8.2. Para RHEL 8.1 y 8.0, utilice la solución descrita en el artículo de la base de conocimientos Using OpenSCAP for scanning containers in RHEL 8.

Requisitos previos

- Los paquetes **openscap-utils** y **scap-security-guide** están instalados.

Procedimiento

1. Obtiene el ID de un contenedor o de una imagen de contenedor, por ejemplo:

```
# podman images
REPOSITORY                          TAG     IMAGE ID      CREATED      SIZE
registry.access.redhat.com/ubi8/ubi   latest  096cae65a207  7 weeks ago  239 MB
```

2. Evaluar la conformidad de la imagen del contenedor con el perfil OSPP y guardar los resultados del escaneo en el archivo HTML *report.html*

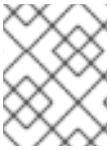
```
# oscap-podman 096cae65a207 xccdf eval --report report.html --profile ospp
/usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
```

Sustituya *096cae65a207* por el ID de su imagen de contenedor y el valor *ospp* por *pci-dss* si evalúa el cumplimiento de la seguridad con la línea de base PCI-DSS. Tenga en cuenta que el comando **oscap-podman** requiere privilegios de root.

Pasos de verificación

1. Compruebe los resultados en un navegador de su elección, por ejemplo:

```
$ firefox report.html &
```



NOTA

Las reglas marcadas como *notapplicable* son reglas que no se aplican a los sistemas en contenedores. Estas reglas sólo se aplican a los sistemas bare-metal y virtualizados.

Recursos adicionales

- Para más información, consulte las páginas de manual **oscap-podman(8)** y **scap-security-guide(8)**.
- La documentación de **SCAP Security Guide** instalada en el **file:///usr/share/doc/scap-security-guide/** directorio.

6.12. VERSIONES SOPORTADAS DE LA GUÍA DE SEGURIDAD SCAP EN RHEL

Las versiones oficialmente soportadas de la Guía de Seguridad SCAP son las versiones proporcionadas en la versión menor de RHEL relacionada o en la actualización por lotes de RHEL relacionada.

Tabla 6.2. Versiones soportadas de la Guía de Seguridad SCAP en RHEL

Versión de Red Hat Enterprise Linux	Versión de la Guía de Seguridad SCAP
RHEL 6.6	scap-security-guide-0.1.18-3.el6
RHEL 6.9	scap-security-guide-0.1.28-3.el6
RHEL 6.10	scap-security-guide-0.1.28-4.el6
RHEL 7.2 AUS	scap-security-guide-0.1.25-3.el7
RHEL 7.3 AUS	scap-security-guide-0.1.30-5.el7_3
RHEL 7.4 AUS, E4S	scap-security-guide-0.1.33-6.el7_4
RHEL 7.5 (actualización por lotes)	scap-security-guide-0.1.36-10.el7_5

Versión de Red Hat Enterprise Linux	Versión de la Guía de Seguridad SCAP
RHEL 7.6 EUS	scap-security-guide-0.1.40-13.el7_6
RHEL 7.7 EUS	scap-security-guide-0.1.43-13.el7
RHEL 7.8 (actualización por lotes)	scap-security-guide-0.1.46-11.el7
RHEL 7.9	scap-security-guide-0.1.52-2.el7_9
RHEL 8.0 SAP	scap-security-guide-0.1.42-11.el8
RHEL 8.1 EUS	scap-security-guide-0.1.47-8.el8_1
RHEL 8.2 (actualización por lotes)	scap-security-guide-0.1.48-10.el8_2
RHEL 8.3	scap-security-guide-0.1.50-16.el8_3

6.13. PERFILES DE LA GUÍA DE SEGURIDAD SCAP SOPORTADOS EN RHEL 8

Utilice sólo el contenido SCAP proporcionado en la versión menor particular de RHEL. Esto se debe a que los componentes que participan en el endurecimiento a veces se actualizan con nuevas capacidades. El contenido SCAP cambia para reflejar estas actualizaciones, pero no siempre es compatible con versiones anteriores.

En las siguientes tablas, puede encontrar los perfiles proporcionados en cada versión menor de RHEL, junto con la versión de la política con la que se alinea el perfil.

Tabla 6.3. Perfiles de la Guía de Seguridad SCAP soportados en RHEL 8.3

Nombre del perfil	Identificación del perfil	Versión política
Prueba de referencia de CIS Red Hat Enterprise Linux 8	**xccdf_org.ssgproject.content_profile_cis**	1.0.0
Información no clasificada en sistemas de información y organizaciones no federales (NIST 800-171)	**xccdf_org.ssgproject.content_profile_cui**	r1
Centro Australiano de Ciberseguridad (ACSC) Ocho esenciales	**xccdf_org.ssgproject.content_profile_e8**	no versionado
Ley de Portabilidad y Responsabilidad del Seguro Médico (HIPAA)	**xccdf_org.ssgproject.content_profile_hipaa**	no versionado

Nombre del perfil	Identificación del perfil	Versión política
Perfil de protección para sistemas operativos de uso general	**xccdf_org.ssgproject.content_profile_ospp**	4.2.1
Línea base de control PCI-DSS v3.2.1 para Red Hat Enterprise Linux 8	**xccdf_org.ssgproject.content_profile_pci-dss**	3.2.1
DRAFT] El Manual de Implementación Técnica de Seguridad de la Agencia de Sistemas de Información de Defensa (DISA STIG) para Red Hat Enterprise Linux 8	**xccdf_org.ssgproject.content_profile_stig**	borrador

Tabla 6.4. Perfiles de la Guía de Seguridad SCAP soportados en RHEL 8.2

Nombre del perfil	Identificación del perfil	Versión política
Centro Australiano de Ciberseguridad (ACSC) Ocho esenciales	**xccdf_org.ssgproject.content_profile_e8**	no versionado
Perfil de protección para sistemas operativos de uso general	**xccdf_org.ssgproject.content_profile_ospp**	4.2.1
Línea base de control PCI-DSS v3.2.1 para Red Hat Enterprise Linux 8	**xccdf_org.ssgproject.content_profile_pci-dss**	3.2.1
[BORRADOR] STIG de DISA para Red Hat Enterprise Linux 8	**xccdf_org.ssgproject.content_profile_stig**	borrador

Tabla 6.5. Perfiles de la Guía de Seguridad SCAP soportados en RHEL 8.1

Nombre del perfil	Identificación del perfil	Versión política
Perfil de protección para sistemas operativos de uso general	**xccdf_org.ssgproject.content_profile_ospp**	4.2.1
Línea base de control PCI-DSS v3.2.1 para Red Hat Enterprise Linux 8	**xccdf_org.ssgproject.content_profile_pci-dss**	3.2.1

Tabla 6.6. Perfiles de la Guía de Seguridad SCAP soportados en RHEL 8.0

Nombre del perfil	Identificación del perfil	Versión política
OSPP - Perfil de protección para sistemas operativos de propósito general	**xccdf_org.ssgproject.content_profile_ospp**	borrador
Línea base de control PCI-DSS v3.2.1 para Red Hat Enterprise Linux 8	**xccdf_org.ssgproject.content_profile_pci-dss**	3.2.1

Recursos adicionales

- Para obtener información sobre los perfiles en RHEL 7, consulte la Guía de seguridad de SCAP Perfiles admitidos en RHEL 7

6.14. INFORMACIÓN RELACIONADA

- La página del proyecto OpenSCAP - La página principal del proyecto OpenSCAP proporciona información detallada sobre la utilidad **oscap** y otros componentes y proyectos relacionados con SCAP.
- Lapágina del proyecto SCAP Workbench - La página de inicio del proyecto SCAP Workbench ofrece información detallada sobre la aplicación **scap-workbench**.
- La página del proyecto SCAP Security Guide (SSG) - La página de inicio del proyecto SSG que proporciona el último contenido de seguridad para Red Hat Enterprise Linux.
- Demostraciones de seguridad de Red Hat: Creación de contenido de políticas de seguridad personalizadas para automatizar el cumplimiento de la seguridad - Un laboratorio práctico para obtener una experiencia inicial en la automatización del cumplimiento de la seguridad utilizando las herramientas que se incluyen en Red Hat Enterprise Linux para cumplir tanto con las políticas de seguridad estándar del sector como con las políticas de seguridad personalizadas. Si desea formación o acceso a estos ejercicios de laboratorio para su equipo, póngase en contacto con su equipo de cuentas de Red Hat para obtener más detalles.
- Demostraciones de seguridad de Red Hat: Defiéndase con las tecnologías de seguridad de RHEL - Un laboratorio práctico para aprender a implementar la seguridad en todos los niveles de su sistema RHEL, utilizando las principales tecnologías de seguridad disponibles en Red Hat Enterprise Linux, incluyendo OpenSCAP. Si desea formación o acceso a estos ejercicios de laboratorio para su equipo, póngase en contacto con su equipo de cuentas de Red Hat para obtener más detalles.
- Página SCAP del Instituto Nacional de Normas y Tecnología (NIST) - Esta página representa una amplia colección de materiales relacionados con SCAP, incluyendo publicaciones, especificaciones y el Programa de Validación de SCAP.
- Base de datos nacional de vulnerabilidad (NVD) - Esta página representa el mayor repositorio de contenido SCAP y otros datos de gestión de vulnerabilidad basados en las normas SCAP.
- Repositorio de contenido de Red Hat OVAL - Este es un repositorio que contiene definiciones de OVAL para vulnerabilidades de los sistemas Red Hat Enterprise Linux. Esta es la fuente recomendada de contenido de vulnerabilidades.

- MITRE CVE - Esta es una base de datos de vulnerabilidades de seguridad conocidas públicamente proporcionada por la corporación MITRE. Para RHEL, se recomienda utilizar el contenido de OVAL CVE proporcionado por Red Hat.

- MITRE OVAL - Este es un proyecto relacionado con OVAL proporcionado por la corporación MITRE. Entre otra información relacionada con OVAL, estas páginas contienen el lenguaje de OVAL y un repositorio de contenido de OVAL con miles de definiciones de OVAL. Tenga en cuenta que para escanear RHEL, se recomienda utilizar el contenido de OVAL CVE proporcionado por Red Hat.

- Gestión del cumplimiento de la seguridad en Red Hat Satellite - Este conjunto de guías describe, entre otros temas, cómo mantener la seguridad del sistema en múltiples sistemas utilizando OpenSCAP.

CAPÍTULO 7. COMPROBACIÓN DE LA INTEGRIDAD CON AIDE

Advanced Intrusion Detection Environment (**AIDE**) es una utilidad que crea una base de datos de archivos en el sistema, y luego utiliza esa base de datos para asegurar la integridad de los archivos y detectar intrusiones en el sistema.

7.1. INSTALACIÓN DE AIDE

Los siguientes pasos son necesarios para instalar **AIDE** e iniciar su base de datos.

Requisitos previos

- El repositorio **AppStream** está activado.

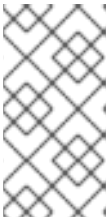
Procedimiento

1. Para instalar el paquete *aide*:

```
# yum install aide
```

2. Para generar una base de datos inicial:

```
# aide --init
```



NOTA

En la configuración por defecto, el comando **aide --init** comprueba sólo un conjunto de directorios y archivos definidos en el archivo **/etc/aide.conf**. Para incluir directorios o archivos adicionales en la base de datos **AIDE**, y para cambiar sus parámetros observados, edite **/etc/aide.conf** en consecuencia.

3. Para empezar a utilizar la base de datos, elimine la subcadena **.new** del nombre inicial del archivo de la base de datos:

```
# mv /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz
```

4. Para cambiar la ubicación de la base de datos **AIDE**, edite el archivo **/etc/aide.conf** y modifique el valor **DBDIR**. Para mayor seguridad, almacene la base de datos, la configuración y el archivo binario **/usr/sbin/aide** en una ubicación segura, como un soporte de sólo lectura.

7.2. REALIZACIÓN DE COMPROBACIONES DE INTEGRIDAD CON AIDE

Requisitos previos

- **AIDE** está correctamente instalado y su base de datos está inicializada. Véase Sección 7.1, “Instalación de AIDE”

Procedimiento

1. Para iniciar un control manual:

```
# aide --check
Start timestamp: 2018-07-11 12:41:20 +0200 (AIDE 0.16)
AIDE found differences between database and filesystem!!
...
[trimmed for clarity]
```

2. Como mínimo, **AIDE** debería estar configurado para ejecutar un escaneo semanal. Como máximo, **AIDE** debería ejecutarse diariamente. Por ejemplo, para programar una ejecución diaria de **AIDE** a las *04:05* a.m. utilizando el comando **cron**, agregue la siguiente línea al archivo **/etc/crontab**:

```
05 4 * * * root /usr/sbin/aide --check
```

7.3. ACTUALIZACIÓN DE UNA BASE DE DATOS AIDE

Después de verificar los cambios de su sistema, como las actualizaciones de los paquetes o los ajustes de los archivos de configuración, se recomienda actualizar su base de datos de referencia **AIDE**.

Requisitos previos

- **AIDE** está correctamente instalado y su base de datos está inicializada. Véase Sección 7.1, “Instalación de AIDE”

Procedimiento

1. Actualice su base de datos de referencia **AIDE**:

```
# aide --update
```

   El comando **aide --update** crea el archivo de base de datos **/var/lib/aide/aide.db.new.gz**.

2. Para empezar a utilizar la base de datos actualizada para las comprobaciones de integridad, elimine la subcadena **.new** del nombre del archivo.

7.4. INFORMACIÓN RELACIONADA

Para obtener más información sobre **AIDE**, consulte la página de manual **aide(1)**.

CAPÍTULO 8. CIFRADO DE DISPOSITIVOS DE BLOQUE MEDIANTE LUKS

El cifrado de discos protege los datos de un dispositivo de bloque cifrándolos. Para acceder al contenido descifrado del dispositivo, el usuario debe proporcionar una frase de paso o una clave como autenticación. Esto es especialmente importante cuando se trata de ordenadores móviles y medios extraíbles: ayuda a proteger el contenido del dispositivo aunque se haya retirado físicamente del sistema. El formato LUKS es una implementación por defecto del cifrado de dispositivos en bloque en RHEL.

8.1. CIFRADO DE DISCO LUKS

El sistema Linux Unified Key Setup-on-disk-format (LUKS) permite cifrar dispositivos de bloque y proporciona un conjunto de herramientas que simplifica la gestión de los dispositivos cifrados. LUKS permite que varias claves de usuario descifren una clave maestra, que se utiliza para el cifrado masivo de la partición.

RHEL utiliza LUKS para realizar el cifrado del dispositivo de bloque. Por defecto, la opción de cifrar el dispositivo de bloque está desmarcada durante la instalación. Si selecciona la opción de cifrar el disco, el sistema le pedirá una frase de contraseña cada vez que arranque el ordenador. Esta frase de contraseña "desbloquea" la clave de cifrado masivo que descifra su partición. Si eliges modificar la tabla de particiones por defecto, puedes elegir qué particiones quieres cifrar. Esto se establece en la configuración de la tabla de particiones.

Qué hace LUKS

- LUKS encripta dispositivos de bloques enteros y, por tanto, es muy adecuado para proteger el contenido de dispositivos móviles, como medios de almacenamiento extraíbles o unidades de disco de ordenadores portátiles.
- El contenido subyacente del dispositivo de bloque cifrado es arbitrario, lo que lo hace útil para cifrar dispositivos de intercambio. También puede ser útil con ciertas bases de datos que utilizan dispositivos de bloque con un formato especial para el almacenamiento de datos.
- LUKS utiliza el subsistema del kernel de mapeo de dispositivos existente.
- LUKS proporciona un refuerzo de la frase de contraseña que protege contra los ataques de diccionario.
- Los dispositivos LUKS contienen varias ranuras para claves, lo que permite a los usuarios añadir claves de seguridad o frases de contraseña.

Qué hace LUKS*not*

- Las soluciones de cifrado de discos como LUKS protegen los datos sólo cuando el sistema está apagado. Una vez que el sistema está encendido y LUKS ha descifrado el disco, los archivos de ese disco están disponibles para cualquiera que normalmente tendría acceso a ellos.
- LUKS no es adecuado para escenarios que requieran que muchos usuarios tengan claves de acceso distintas para el mismo dispositivo. El formato LUKS1 proporciona ocho ranuras para llaves, LUKS2 hasta 32 ranuras para llaves.
- LUKS no es adecuado para aplicaciones que requieran encriptación a nivel de archivo.

Cifras

El cifrado por defecto utilizado para LUKS es **aes-xts-plain64**. El tamaño de la clave por defecto para LUKS es de 512 bits. El tamaño de la clave por defecto para LUKS con **Anaconda** (modo XTS) es de 512 bits. Los cifrados disponibles son:

- AES - Estándar de cifrado avanzado - FIPS PUB 197
- Twofish (un cifrado en bloque de 128 bits)
- Serpent

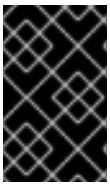
Recursos adicionales

- Página de inicio del proyecto LUKS
- Especificación del formato LUKS en disco

8.2. VERSIONES DE LUKS EN RHEL 8

En RHEL 8, el formato por defecto para el cifrado LUKS es LUKS2. El formato heredado LUKS1 sigue siendo totalmente compatible y se proporciona como un formato compatible con las versiones anteriores de RHEL.

El formato LUKS2 está diseñado para permitir futuras actualizaciones de varias partes sin necesidad de modificar las estructuras binarias. LUKS2 utiliza internamente el formato de texto JSON para los metadatos, proporciona redundancia de metadatos, detecta la corrupción de metadatos y permite la reparación automática a partir de una copia de metadatos.



IMPORTANTE

No utilice LUKS2 en sistemas que necesiten ser compatibles con sistemas heredados que sólo soporten LUKS1. Tenga en cuenta que RHEL 7 admite el formato LUKS2 desde la versión 7.6.

AVISO

LUKS2 y LUKS1 utilizan diferentes comandos para cifrar el disco. Utilizar el comando incorrecto para una versión de LUKS podría causar la pérdida de datos.

Versión LUKS	Comando de encriptación
LUKS2	**cryptsetup reencrypt**
LUKS1	**cryptsetup-reencrypt**

Reencriptación en línea

El formato LUKS2 permite volver a encriptar los dispositivos encriptados mientras éstos están en uso. Por ejemplo, no es necesario desmontar el sistema de archivos del dispositivo para realizar las siguientes tareas:

- Cambiar la tecla de volumen
- Cambiar el algoritmo de encriptación

Cuando se encripta un dispositivo no encriptado, hay que desmontar el sistema de archivos. Puedes volver a montar el sistema de archivos tras una breve inicialización del cifrado.

El formato LUKS1 no admite la recodificación en línea.

Conversión

El formato LUKS2 se inspira en LUKS1. En determinadas situaciones, se puede convertir LUKS1 en LUKS2. La conversión no es posible específicamente en los siguientes escenarios:

- Un dispositivo LUKS1 está marcado como utilizado por una solución de descifrado basada en políticas (PBD - Clevis). La herramienta **cryptsetup** se niega a convertir el dispositivo cuando se detectan algunos metadatos de **luksmeta**.
- Un dispositivo está activo. El dispositivo debe estar en estado inactivo antes de que sea posible cualquier conversión.

8.3. OPCIONES DE PROTECCIÓN DE DATOS DURANTE LA RECODIFICACIÓN DE LUKS2

LUKS2 ofrece varias opciones que priorizan el rendimiento o la protección de los datos durante el proceso de recodificación:

checksum

Este es el modo por defecto. Equilibra la protección de los datos y el rendimiento.
Este modo almacena sumas de comprobación individuales de los sectores en el área de recodificación, por lo que el proceso de recuperación puede detectar los sectores que LUKS2 ya ha recodificado. El modo requiere que la escritura del sector del dispositivo de bloque sea atómica.

journal

Es el modo más seguro pero también el más lento. Este modo registra el área de recodificación en el área binaria, por lo que LUKS2 escribe los datos dos veces.

none

Este modo da prioridad al rendimiento y no proporciona ninguna protección de datos. Protege los datos sólo contra la terminación segura del proceso, como la señal **SIGTERM** o la pulsación por parte del usuario de **Ctrl**+**C**. Cualquier fallo inesperado del sistema o de la aplicación puede provocar la corrupción de los datos.

Puede seleccionar el modo mediante la opción **--resilience** de **cryptsetup**.

Si un proceso de recodificación de LUKS2 termina inesperadamente por la fuerza, LUKS2 puede realizar la recuperación de una de las siguientes maneras:

- Automáticamente, durante la siguiente acción de apertura del dispositivo LUKS2. Esta acción se desencadena mediante el comando **cryptsetup open** o al adjuntar el dispositivo con **systemd-cryptsetup**.

- Manualmente, utilizando el comando **cryptsetup repair** en el dispositivo LUKS2.

8.4. CIFRADO DE DATOS EXISTENTES EN UN DISPOSITIVO DE BLOQUES MEDIANTE LUKS2

Este procedimiento encripta los datos existentes en un dispositivo aún no encriptado utilizando el formato LUKS2. Se almacena una nueva cabecera LUKS en el cabezal del dispositivo.

Requisitos previos

- El dispositivo de bloque contiene un sistema de archivos.
- Has hecho una copia de seguridad de tus datos.

AVISO

Podrías perder tus datos durante el proceso de encriptación: debido a un fallo de hardware, del núcleo o humano. Asegúrate de tener una copia de seguridad fiable antes de empezar a encriptar los datos.

Procedimiento

1. Desmonte todos los sistemas de archivos del dispositivo que vaya a cifrar. Por ejemplo:

```
# umount /dev/sdb1
```

2. Deje espacio libre para almacenar una cabecera LUKS. Elija una de las siguientes opciones que se adapte a su escenario:
   - En el caso de la encriptación de un volumen lógico, se puede ampliar el volumen lógico sin cambiar el tamaño del sistema de archivos. Por ejemplo:

```
# lvextend -L 32M vg00/lv00
```

   - Amplíe la partición utilizando herramientas de gestión de particiones, como **parted**.
   - Reduzca el sistema de archivos del dispositivo. Puede utilizar la utilidad **resize2fs** para los sistemas de archivos ext2, ext3 o ext4. Ten en cuenta que no puedes reducir el sistema de archivos XFS.

3. Inicializar la encriptación. Por ejemplo:

```
# cryptsetup reencrypt \
        --encrypt \
        --init-only \
        --reduce-device-size 32M \
        /dev/sdb1 sdb1_encrypted
```

   El comando le pide una frase de contraseña y comienza el proceso de encriptación.

4. Monta el dispositivo:

```
# mount /dev/mapper/sdb1_encrypted /mnt/sdb1_encrypted
```

5. Inicie la codificación en línea:

```
# cryptsetup reencrypt --resume-only /dev/sdb1
```

Recursos adicionales

- Para más detalles, consulte las páginas de manual **cryptsetup(8)**, **lvextend(8)**, **resize2fs(8)**, y **parted(8)**.

8.5. CIFRADO DE DATOS EXISTENTES EN UN DISPOSITIVO DE BLOQUE MEDIANTE LUKS2 CON UNA CABECERA SEPARADA

Este procedimiento encripta los datos existentes en un dispositivo de bloque sin crear espacio libre para almacenar una cabecera LUKS. La cabecera se almacena en una ubicación independiente, lo que también sirve como capa adicional de seguridad. El procedimiento utiliza el formato de cifrado LUKS2.

Requisitos previos

- El dispositivo de bloque contiene un sistema de archivos.
- Has hecho una copia de seguridad de tus datos.

AVISO

Podrías perder tus datos durante el proceso de encriptación: debido a un fallo de hardware, del núcleo o humano. Asegúrate de tener una copia de seguridad fiable antes de empezar a encriptar los datos.

Procedimiento

1. Desmontar todos los sistemas de archivos del dispositivo. Por ejemplo:

```
# umount /dev/sdb1
```

2. Inicializar la encriptación:

```
# cryptsetup reencrypt \
        --encrypt \
        --init-only \
        --header /path/to/header \
        /dev/sdb1 sdb1_encrypted
```

Sustituya */path/to/header* por una ruta al archivo con una cabecera LUKS separada. La cabecera LUKS separada tiene que ser accesible para que el dispositivo encriptado pueda ser desbloqueado posteriormente.

El comando le pide una frase de contraseña y comienza el proceso de encriptación.

3. Monta el dispositivo:

```
# mount /dev/mapper/sdb1_encrypted /mnt/sdb1_encrypted
```

4. Inicie la codificación en línea:

```
# cryptsetup reencrypt --resume-only --header /path/to/header /dev/sdb1
```

Recursos adicionales

- Para más detalles, consulte la página de manual **cryptsetup(8)**.

8.6. CIFRADO DE UN DISPOSITIVO DE BLOQUE EN BLANCO MEDIANTE LUKS2

Este procedimiento proporciona información sobre el cifrado de un dispositivo de bloque en blanco utilizando el formato LUKS2.

Requisitos previos

- Un dispositivo de bloque en blanco.

Procedimiento

1. Configurar una partición como una partición LUKS cifrada:

```
# cryptsetup luksFormat /dev/sdb1
```

2. Abra una partición LUKS encriptada:

```
# cryptsetup open /dev/sdb1 sdb1_encrypted
```

Esto desbloquea la partición y la asigna a un nuevo dispositivo utilizando el mapeador de dispositivos. Esto alerta al kernel de que **device** es un dispositivo encriptado y debe ser direccionado a través de LUKS usando el ***/dev/mapper/device_mapped_name*** para no sobrescribir los datos encriptados.

3. Para escribir datos encriptados en la partición, se debe acceder a ella a través del nombre mapeado del dispositivo. Para ello, debes crear un sistema de archivos. Por ejemplo:

```
# mkfs -t ext4 /dev/mapper/sdb1_encrypted
```

4. Monta el dispositivo:

```
# montaje /dev/mapper/sdb1_encrypted
```

Recursos adicionales

Recursos adicionales

- Para más detalles, consulte la página de manual **cryptsetup(8)**.

8.7. CREACIÓN DE UN VOLUMEN ENCRIPTADO LUKS UTILIZANDO EL ROL DE ALMACENAMIENTO

Puede utilizar el rol **storage** para crear y configurar un volumen encriptado con LUKS ejecutando un playbook de Ansible.

Requisitos previos

- Tiene instalado Red Hat Ansible Engine en el sistema desde el que desea ejecutar el libro de jugadas.

NOTA

No es necesario tener Red Hat Ansible Automation Platform instalado en los sistemas en los que se desea crear el volumen.

- Tiene el paquete **rhel-system-roles** instalado en el controlador Ansible.
- Dispone de un archivo de inventario en el que se detallan los sistemas en los que desea desplegar un volumen encriptado LUKS mediante el rol de sistema de almacenamiento.

Procedimiento

1. Cree un nuevo ***playbook.yml*** archivo con el siguiente contenido:

```
- hosts: all
  vars:
    storage_volumes:
      - name: barefs
        type: disk
        disks:
         - sdb
        fs_type: xfs
        fs_label: label-name
        mount_point: /mnt/data
        encryption: true
        encryption_password: your-password
  roles:
   - rhel-system-roles.storage
```

2. Opcional. Verificar la sintaxis del libro de jugadas:

```
# ansible-playbook --syntax-check playbook.yml
```

3. Ejecute el libro de jugadas en su archivo de inventario:

```
# ansible-playbook -i inventory.file /path/to/file/playbook.yml
```

Recursos adicionales

- Para más información sobre LUKS, véase 17. Cifrado de dispositivos de bloque mediante LU KS..
- Para más detalles sobre los parámetros utilizados en el rol de sistema **storage**, consulte el archivo **/usr/share/ansible/roles/rhel-system-roles.storage/README.md**.

CAPÍTULO 9. CONFIGURACIÓN DEL DESBLOQUEO AUTOMÁTICO DE VOLÚMENES ENCRIPTADOS MEDIANTE EL DESCIFRADO BASADO EN POLÍTICAS

El descifrado basado en políticas (PBD) es un conjunto de tecnologías que permiten desbloquear los volúmenes raíz y secundarios cifrados de los discos duros de las máquinas físicas y virtuales. PBD utiliza diversos métodos de desbloqueo, como las contraseñas de los usuarios, un dispositivo Trusted Platform Module (TPM), un dispositivo PKCS #11 conectado a un sistema, por ejemplo, una tarjeta inteligente, o un servidor de red especial.

La PBD permite combinar diferentes métodos de desbloqueo en una política, lo que hace posible desbloquear el mismo volumen de diferentes maneras. La implementación actual de la PBD en Red Hat Enterprise Linux consiste en el marco de trabajo Clevis y en complementos llamados *pins*. Cada pin proporciona una capacidad de desbloqueo independiente. Actualmente, los siguientes pines están disponibles:

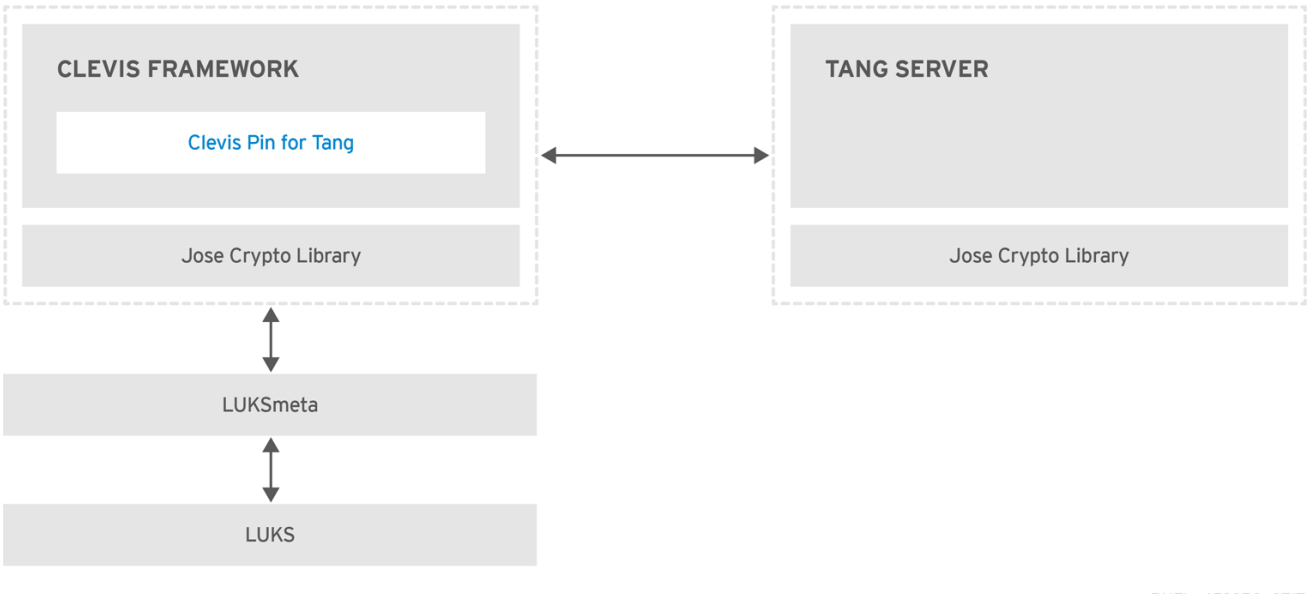
- **tang** - permite desbloquear volúmenes mediante un servidor de red
- **tpm2** - permite desbloquear volúmenes mediante una política TPM2

El Network Bound Disc Encryption (NBDE) es una subcategoría de PBD que permite vincular volúmenes cifrados a un servidor de red especial. La implementación actual del NBDE incluye un pasador de horquilla para el servidor Tang y el propio servidor Tang.

9.1. ENCRIPTACIÓN DE DISCOS EN RED

En Red Hat Enterprise Linux, NBDE se implementa a través de los siguientes componentes y tecnologías:

Figura 9.1. Esquema NBDE cuando se utiliza un volumen encriptado por LUKS1. El paquete luksmeta no se utiliza para los volúmenes LUKS2.



Tang es un servidor para vincular los datos a la presencia en la red. Hace que un sistema que contiene sus datos esté disponible cuando el sistema está vinculado a una determinada red segura. Tang no tiene estado y no requiere TLS ni autenticación. A diferencia de las soluciones basadas en escrow, en las que

el servidor almacena todas las claves de cifrado y tiene conocimiento de todas las claves utilizadas, Tang nunca interactúa con ninguna clave del cliente, por lo que nunca obtiene ninguna información de identificación del cliente.

Clevis es un marco enchufable para el descifrado automatizado. En NBDE, Clevis proporciona el desbloqueo automatizado de volúmenes LUKS. El paquete **clevis** proporciona el lado del cliente de la función.

Un *Clevis pin* es un plug-in en el marco de Clevis. Uno de estos pines es un plug-in que implementa las interacciones con el servidor NBDE

Clevis y Tang son componentes genéricos de cliente y servidor que proporcionan encriptación ligada a la red. En Red Hat Enterprise Linux, se utilizan junto con LUKS para cifrar y descifrar los volúmenes de almacenamiento raíz y no raíz para lograr el cifrado de disco ligado a la red.

Tanto los componentes del lado del cliente como los del lado del servidor utilizan la biblioteca *José* para realizar operaciones de cifrado y descifrado.

Cuando se inicia el aprovisionamiento de NBDE, la clavija de Clevis para el servidor Tang obtiene una lista de las claves asimétricas anunciadas del servidor Tang. Como alternativa, dado que las claves son asimétricas, se puede distribuir una lista de las claves públicas de Tang fuera de banda para que los clientes puedan operar sin acceso al servidor Tang. Este modo se denomina *offline provisioning*.

El pasador de horquilla para Tang utiliza una de las claves públicas para generar una clave de cifrado única y criptográficamente fuerte. Una vez que los datos se encriptan utilizando esta clave, ésta se descarta. El cliente Clevis debe almacenar el estado producido por esta operación de aprovisionamiento en una ubicación conveniente. Este proceso de encriptación de datos es el *provisioning step*.

La versión 2 de LUKS (LUKS2) es el formato por defecto en Red Hat Enterprise Linux 8, por lo tanto, el estado de aprovisionamiento para NBDE se almacena como un token en una cabecera LUKS2. El aprovechamiento del estado de aprovisionamiento para NBDE por el paquete **luksmeta** sólo se utiliza para los volúmenes cifrados con LUKS1. El pasador de horquilla para Tang soporta tanto LUKS1 como LUKS2 sin necesidad de especificación.

Cuando el cliente está listo para acceder a sus datos, carga los metadatos producidos en el paso de aprovisionamiento y responde para recuperar la clave de cifrado. Este proceso es el *recovery step*.

En NBDE, Clevis vincula un volumen LUKS mediante un pin para que pueda ser desbloqueado automáticamente. Una vez completado con éxito el proceso de vinculación, el disco puede ser desbloqueado utilizando el desbloqueador Dracut proporcionado.

9.2. INSTALACIÓN DE UN CLIENTE DE ENCRIPTACIÓN - CLEVIS

Utilice este procedimiento para desplegar y empezar a utilizar el marco enchufable de Clevis en su sistema.

Procedimiento

1. Para instalar Clevis y sus pines en un sistema con un volumen encriptado:

```
# yum install clevis
```

2. Para descifrar los datos, utilice un comando **clevis decrypt** y proporcione un texto cifrado en el formato JSON Web Encryption (JWE), por ejemplo:

```
$ clevis decrypt < secret.jwe
```

Recursos adicionales

- Para una referencia rápida, consulte la ayuda de la CLI incorporada:

```
$ clevis
Usage: clevis COMMAND [OPTIONS]

clevis decrypt           Decrypts using the policy defined at encryption time
clevis encrypt sss       Encrypts using a Shamir's Secret Sharing policy
clevis encrypt tang      Encrypts using a Tang binding server policy
clevis encrypt tpm2      Encrypts using a TPM2.0 chip binding policy
clevis luks bind         Binds a LUKS device using the specified policy
clevis luks list         Lists pins bound to a LUKSv1 or LUKSv2 device
clevis luks pass         Returns the LUKS passphrase used for binding a particular slot.
clevis luks regen        Regenerate LUKS metadata
clevis luks report       Report any key rotation on the server side
clevis luks unbind       Unbinds a pin bound to a LUKS volume
clevis luks unlock       Unlocks a LUKS volume
```

- Para más información, consulte la página de manual **clevis(1)**.

9.3. DESPLIEGUE DE UN SERVIDOR TANG CON SELINUX EN MODO REFORZADO

Utilice este procedimiento para desplegar un servidor Tang que se ejecuta en un puerto personalizado como un servicio confinado en el modo de aplicación de SELinux.

Requisitos previos

- El paquete **policycoreutils-python-utils** y sus dependencias están instalados.

Procedimiento

1. Para instalar el paquete **tang** y sus dependencias, introduzca el siguiente comando como **root**:

```
# yum install tang
```

2. Elija un puerto desocupado, por ejemplo, *7500/tcp*, y permita que el servicio **tangd** se vincule a ese puerto:

```
# semanage port -a -t tangd_port_t -p tcp 7500
```

   Tenga en cuenta que un puerto sólo puede ser utilizado por un servicio a la vez, por lo que un intento de utilizar un puerto ya ocupado implica el mensaje de error **ValueError: Port already defined**.

3. Abre el puerto en el firewall:

```
# firewall-cmd --add-port=7500/tcp
# firewall-cmd --runtime-to-permanent
```

4. Habilite el servicio **tangd**:

```
# systemctl enable tangd.socket
```

5. Crear un archivo de anulación:

```
# systemctl edit tangd.socket
```

6. En la siguiente pantalla del editor, que abre un archivo **override.conf** vacío situado en el directorio **/etc/systemd/system/tangd.socket.d/**, cambie el puerto por defecto del servidor Tang de 80 al número elegido anteriormente añadiendo las siguientes líneas:

```
[Socket]
ListenStream=
ListenStream=7500
```

Guarde el archivo y salga del editor.

7. Recarga la configuración modificada:

```
# systemctl daemon-reload
```

8. Comprueba que tu configuración funciona:

```
# systemctl show tangd.socket -p Listen
Listen=[::]:7500 (Stream)
```

9. Inicie el servicio **tangd**:

```
# systemctl start tangd.socket
```

Dado que **tangd** utiliza el mecanismo de activación de sockets **systemd**, el servidor se inicia en cuanto llega la primera conexión. En el primer arranque se genera automáticamente un nuevo juego de claves criptográficas. Para realizar operaciones criptográficas como la generación manual de claves, utilice la utilidad **jose**.

Recursos adicionales

- **tang(8)** página de manual
- **semanage(8)** página de manual
- **firewall-cmd(1)** página de manual
- **systemd.unit(5)** y **systemd.socket(5)** páginas man
- **jose(1)** página de manual

9.4. ROTACIÓN DE LAS CLAVES DEL SERVIDOR TANG Y ACTUALIZACIÓN DE LOS ENLACES EN LOS CLIENTES

Siga los siguientes pasos para rotar las claves del servidor Tang y actualizar los enlaces existentes en los clientes. El intervalo exacto con el que debe rotarlas depende de su aplicación, del tamaño de las claves y de la política institucional.

Requisitos previos

- Se está ejecutando un servidor Tang.
- Los paquetes **clevis** y **clevis-luks** están instalados en sus clientes.
- Tenga en cuenta que **clevis luks list**, **clevis luks report**, y **clevis luks regen** se han introducido en RHEL 8.2.

Procedimiento

1. Para rotar las claves, genere nuevas claves utilizando el comando **/usr/libexec/tangd-keygen** en el directorio de la base de datos de claves **/var/db/tang** en el servidor Tang:

```
# ls /var/db/tang
UV6dqXSwe1bRKG3KbJmdiR020hY.jwk y9hxLTQSiSB5jSEGWnjhY8fDTJU.jwk
# /usr/libexec/tangd-keygen /var/db/tang
# ls /var/db/tang
UV6dqXSwe1bRKG3KbJmdiR020hY.jwk y9hxLTQSiSB5jSEGWnjhY8fDTJU.jwk
3ZWS6-cDrCG61UPJS2BMmPU4I54.jwk zyLuX6hijUy_PSeUEFDi7hi38.jwk
```

2. Compruebe que su servidor Tang anuncia la clave de firma del nuevo par de claves, por ejemplo:

```
# tang-show-keys 7500
3ZWS6-cDrCG61UPJS2BMmPU4I54
```

3. Cambie el nombre de las claves antiguas para que tengan un **.** inicial para ocultarlas de la publicidad. Tenga en cuenta que los nombres de archivo en el siguiente ejemplo difieren de los nombres de archivo únicos en el directorio de la base de datos de claves de su servidor Tang:

```
# cd /var/db/tang
# ls -l
-rw-r--r--. 1 root tang 354 Sep 23 16:08 3ZWS6-cDrCG61UPJS2BMmPU4I54.jwk
-rw-r--r--. 1 root tang 349 Sep 23 16:08 I-zyLuX6hijUy_PSeUEFDi7hi38.jwk
-rw-r--r--. 1 root root 349 Feb  7 14:55 UV6dqXSwe1bRKG3KbJmdiR020hY.jwk
-rw-r--r--. 1 root root 354 Feb  7 14:55 y9hxLTQSiSB5jSEGWnjhY8fDTJU.jwk
# mv UV6dqXSwe1bRKG3KbJmdiR020hY.jwk .UV6dqXSwe1bRKG3KbJmdiR020hY.jwk
# mv y9hxLTQSiSB5jSEGWnjhY8fDTJU.jwk .y9hxLTQSiSB5jSEGWnjhY8fDTJU.jwk
```

   Tang recoge inmediatamente todos los cambios. No es necesario reiniciar. En este momento, los nuevos enlaces de los clientes recogen las nuevas claves y los antiguos clientes pueden seguir utilizando las antiguas claves.

4. En sus clientes NBDE, utilice el comando **clevis luks report** para comprobar si las claves anunciadas por el servidor Tang siguen siendo las mismas. Puede identificar las ranuras con el enlace correspondiente utilizando el comando **clevis luks list**, por ejemplo:

```
# clevis luks list -d /dev/sda2
1: tang '{"url":"http://tang.srv"}'
# clevis luks report -d /dev/sda2 -s 1
```

```
...
Report detected that some keys were rotated.
Do you want to regenerate luks metadata with "clevis luks regen -d /dev/sda2 -s 1"? [ynYN]
```

5. Para regenerar los metadatos LUKS para las nuevas claves, pulse **y** en el prompt del comando anterior, o utilice el comando **clevis luks regen**:

```
# clevis luks regen -d /dev/sda2 -s 1
```

6. Cuando esté seguro de que todos los clientes antiguos utilizan las nuevas claves, puede eliminar las antiguas claves del servidor Tang, por ejemplo:

```
# cd /var/db/tang
# rm .*.jwk
```

AVISO

La eliminación de las claves antiguas mientras los clientes aún las utilizan puede provocar la pérdida de datos. Si elimina accidentalmente dichas claves, utilice el comando **clevis luks regen** en los clientes y proporcione su contraseña LUKS manualmente.

Recursos adicionales

- **tang-show-keys(1)**, **clevis-luks-list(1)**, **clevis-luks-report(1)**, y **clevis-luks-regen(1)** páginas de manual

9.5. CONFIGURACIÓN DEL DESBLOQUEO AUTOMÁTICO MEDIANTE UNA LLAVE TANG EN LA CONSOLA WEB

Configure el desbloqueo automático de un dispositivo de almacenamiento cifrado con LUKS utilizando una clave proporcionada por un servidor Tang.

Requisitos previos

- Se ha instalado la consola web de RHEL 8.
  Para más detalles, véase Instalación de la consola web .
- El paquete **cockpit-storaged** está instalado en su sistema.
- El servicio **cockpit.socket** se ejecuta en el puerto 9090.
- Los paquetes **clevis**, **tang**, y **clevis-dracut** están instalados.
- Se está ejecutando un servidor Tang.

Procedimiento

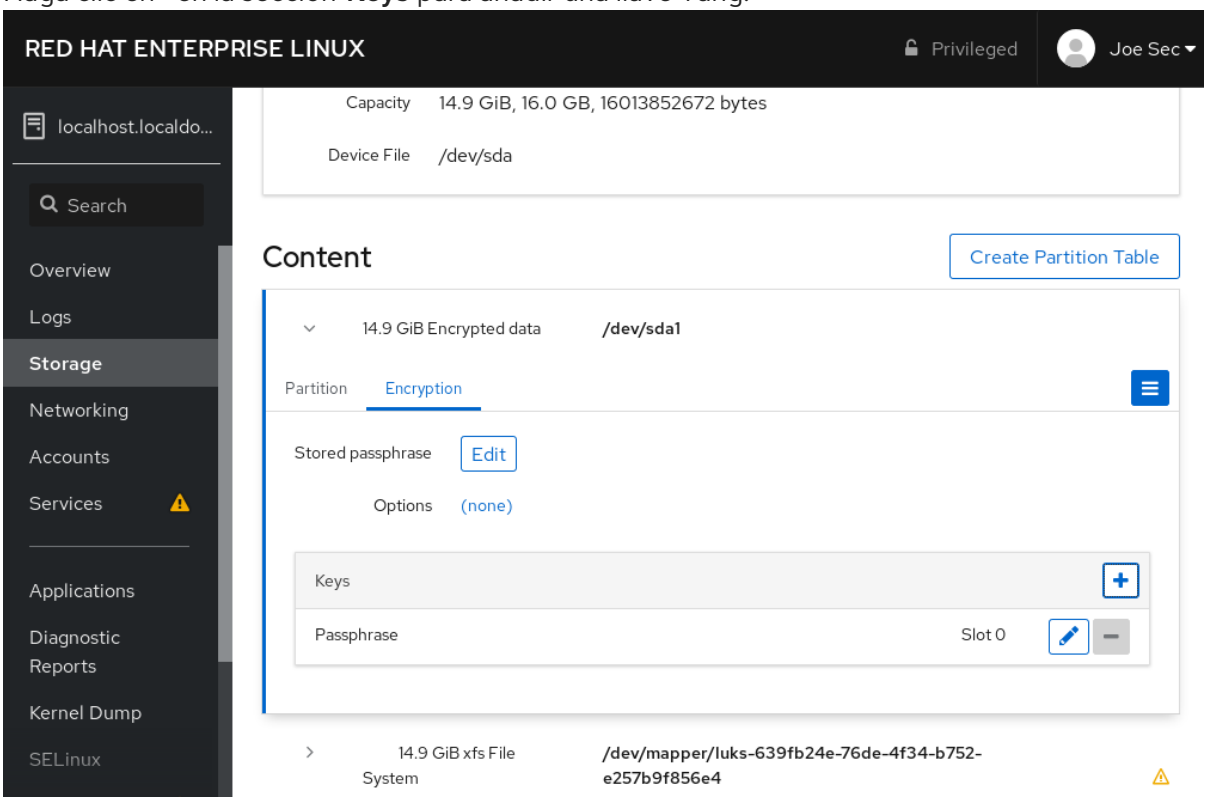
1. Abra la consola web de RHEL introduciendo la siguiente dirección en un navegador web:

> https://*localhost*:9090

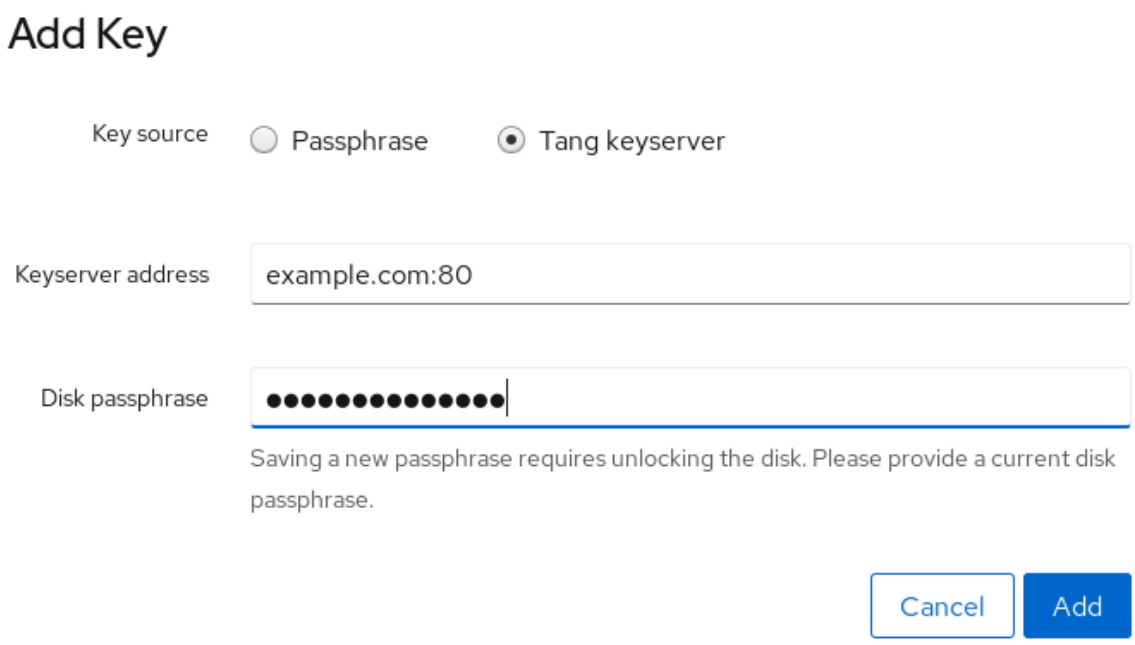
Sustituya la parte *localhost* por el nombre del servidor remoto o la dirección IP cuando se conecte a un sistema remoto.

2. Proporcione sus credenciales y haga clic en **Almacenamiento**. Seleccione un dispositivo cifrado y haga clic en **Cifrado** en la parte **Content**:

3. Haga clic en en la sección **Keys** para añadir una llave Tang:



4. Proporcione la dirección de su servidor Tang y una contraseña que desbloquee el dispositivo cifrado con LUKS. Haz clic en **Añadir** para confirmar:



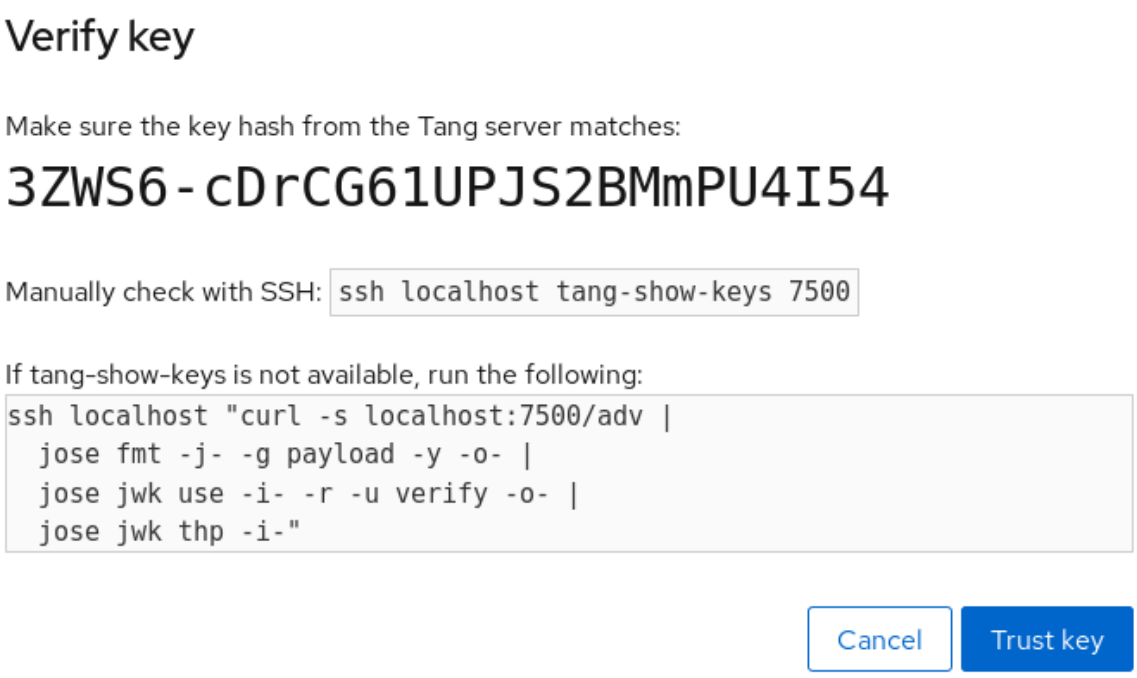
5. La siguiente ventana de diálogo proporciona un comando para verificar que el hash de la clave coincide. RHEL 8.2

```
# tang-show-keys 7500
3ZWS6-cDrCG61UPJS2BMmPU4I54
```

En RHEL 8.1 y anteriores, obtenga el hash de la clave utilizando el siguiente comando:

```
# curl -s localhost:7500/adv | jose fmt -j- -g payload -y -o- | jose jwk use -i- -r -u verify -o- |
jose jwk thp -i-
3ZWS6-cDrCG61UPJS2BMmPU4I54
```

6. Haga clic en **Confiar en la clave** cuando los hashes de la clave en la consola web y en la salida de los comandos enumerados anteriormente sean iguales:

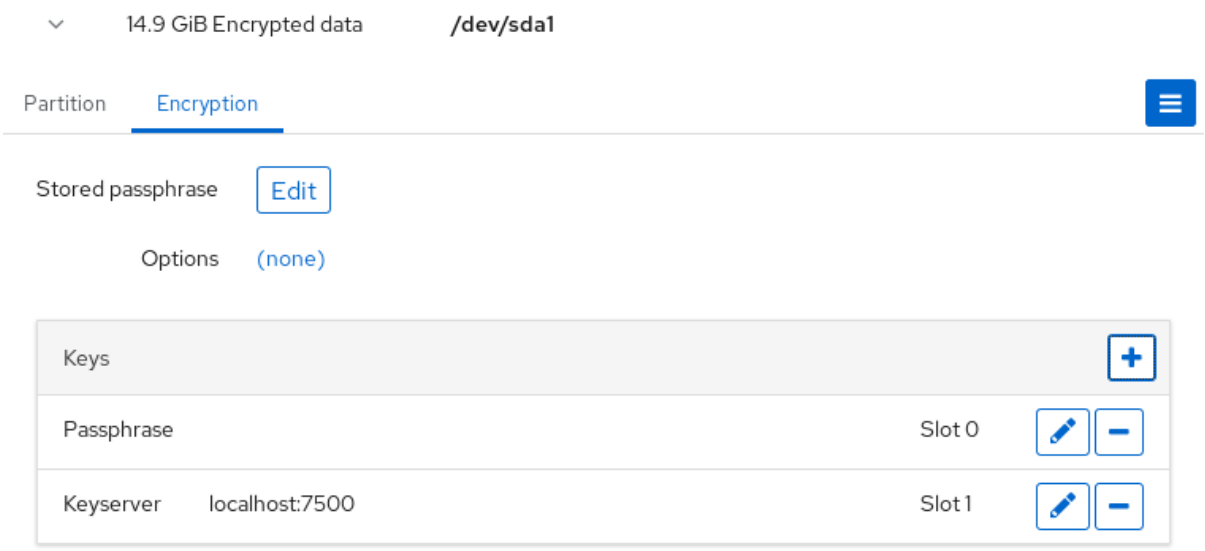


7. Para permitir que el sistema de arranque temprano procese la unión de discos, haga clic en **Terminal** en la parte inferior de la barra de navegación izquierda e introduzca los siguientes comandos:

```
# yum install clevis-dracut
# dracut -fv --regenerate-all
```

Pasos de verificación

1. Compruebe que la clave Tang recién añadida aparece ahora en la sección **Keys** con el tipo **Keyserver**:



2. Comprueba que las fijaciones están disponibles para el arranque temprano, por ejemplo:

```
# lsinitrd | grep clevis
clevis
clevis-pin-sss
clevis-pin-tang
clevis-pin-tpm2
-rwxr-xr-x   1 root     root         1600 Feb 11 16:30 usr/bin/clevis
-rwxr-xr-x   1 root     root         1654 Feb 11 16:30 usr/bin/clevis-decrypt
...
-rwxr-xr-x   2 root     root           45 Feb 11 16:30 usr/lib/dracut/hooks/initqueue/settled/60-
clevis-hook.sh
-rwxr-xr-x   1 root     root         2257 Feb 11 16:30 usr/libexec/clevis-luks-askpass
```

Recursos adicionales

- Para más detalles sobre la instalación y el inicio de sesión en la consola web de RHEL, consulte el capítulo Introducción al uso de la consola web de RHEL .

9.6. DESPLIEGUE DE UN CLIENTE DE ENCRIPTACIÓN PARA UN SISTEMA NBDE CON TANG

El siguiente procedimiento contiene los pasos para configurar el desbloqueo automático de un volumen cifrado con un servidor de red Tang.

Requisitos previos

- El marco de la horquilla está instalado.
- Hay un servidor Tang disponible.

Procedimiento

1. Para vincular un cliente de encriptación Clevis a un servidor Tang, utilice el subcomando **clevis encrypt tang**:

```
$ clevis encrypt tang '{"url":"http://tang.srv:port"}' < input-plain.txt > secret.jwe
The advertisement contains the following signing keys:

_OsIk0T-E2l6qjfdDiwVmidoZjA

Do you wish to trust these keys? [ynYN] y
```

Cambie la URL de *http://tang.srv:port* en el ejemplo anterior para que coincida con la URL del servidor donde está instalado **tang**. El archivo de salida *secret.jwe* contiene su texto cifrado en el formato JSON Web Encryption. Este texto cifrado se lee desde el archivo de entrada *input-plain.txt*.

Alternativamente, si su configuración requiere una comunicación no interactiva con un servidor Tang sin acceso SSH, puede descargar un anuncio y guardarlo en un archivo:

```
$ curl -sfg http://tang.srv:port/adv -o adv.jws
```

Utilice el anuncio en el archivo *adv.jws* para cualquier tarea siguiente, como la encriptación de archivos o mensajes:

```
$ echo 'hello' | clevis encrypt tang '{"url":"http://tang.srv:port","adv":"adv.jws"}'
```

2. Para descifrar los datos, utilice el comando **clevis decrypt** y proporcione el texto cifrado (JWE):

```
$ clevis decrypt < secret.jwe > output-plain.txt
```

Recursos adicionales

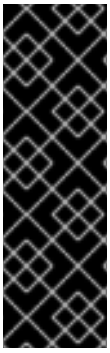
- Para obtener una referencia rápida, consulte la página man **clevis-encrypt-tang(1)** o utilice la ayuda incorporada de la CLI:

```
$ clevis
$ clevis decrypt
$ clevis encrypt tang
Usage: clevis encrypt tang CONFIG < PLAINTEXT > JWE
...
```

- Para más información, consulte las siguientes páginas de manual:
  - **clevis(1)**
  - **clevis-luks-unlockers(7)**

9.7. EXTRACCIÓN MANUAL DE UN PASADOR DE HORQUILLA DE UN VOLUMEN CIFRADO CON LUKS

Utilice el siguiente procedimiento para eliminar manualmente los metadatos creados por el comando **clevis luks bind** y también para borrar una ranura de llave que contenga una frase de contraseña añadida por Clevis.



IMPORTANTE

La forma recomendada de eliminar un pasador de horquilla de un volumen cifrado con LUKS es a través del comando **clevis luks unbind**. El procedimiento de eliminación mediante **clevis luks unbind** consta de un solo paso y funciona tanto para volúmenes LUKS1 como LUKS2. El siguiente comando de ejemplo elimina los metadatos creados por el paso de vinculación y borra la ranura de la llave *1* en el dispositivo */dev/sda2*:

```
# clevis luks unbind -d /dev/sda2 -s 1
```

Requisitos previos

- Un volumen encriptado por LUKS con una encuadernación de horquilla.

Procedimiento

1. Compruebe con qué versión de LUKS está encriptado el volumen, por ejemplo */dev/sda2*, e identifique una ranura y un token que esté vinculado a Clevis:

```
# cryptsetup luksDump /dev/sda2
LUKS header information
Version:        2
...
Keyslots:
  0: luks2
...
1: luks2
      Key:        512 bits
      Priority:   normal
      Cipher:     aes-xts-plain64
...
      Tokens:
        0: clevis
              Keyslot:  1
...
```

En el ejemplo anterior, la ficha de la horquilla se identifica con *0* y la ranura de la llave asociada es *1*.

2. En el caso de la encriptación LUKS2, retire el token:

```
# cryptsetup token remove --token-id 0 /dev/sda2
```

3. Si su dispositivo está encriptado por LUKS1, lo que se indica con la cadena **Version: 1** en la salida del comando **cryptsetup luksDump**, realice este paso adicional con el comando **luksmeta wipe**:

```
# luksmeta wipe -d /dev/sda2 -s 1
```

4. Limpie la ranura de la llave que contiene la frase de contraseña de la Clevis:

```
# cryptsetup luksKillSlot /dev/sda2 1
```

Recursos adicionales

Recursos adicionales

- Para más información, consulte las páginas de manual **clevis-luks-unbind(1)**, **cryptsetup(8)**, y **luksmeta(8)**.

9.8. IMPLEMENTACIÓN DE UN CLIENTE DE CIFRADO CON UNA POLÍTICA TPM 2.0

El siguiente procedimiento contiene los pasos para configurar el desbloqueo automático de un volumen cifrado con una política de Trusted Platform Module 2.0 (TPM 2.0).

Requisitos previos

- El marco de trabajo de Clevis está instalado. Ver Instalación de un cliente de encriptación - Clevis
- Un sistema con arquitectura Intel de 64 bits o AMD de 64 bits

Procedimiento

1. Para desplegar un cliente que cifre utilizando un chip TPM 2.0, utilice el subcomando **clevis encrypt tpm2** con el único argumento en forma de objeto de configuración JSON:

   ```
   $ clevis encrypt tpm2 '{}' < input-plain.txt > secret.jwe
   ```

   Para elegir una jerarquía, un hash y unos algoritmos de clave diferentes, especifique las propiedades de configuración, por ejemplo:

   ```
   $ clevis encrypt tpm2 '{"hash":"sha1","key":"rsa"}' < input-plain.txt > secret.jwe
   ```

2. Para descifrar los datos, proporcione el texto cifrado en el formato JSON Web Encryption (JWE):

   ```
   $ clevis decrypt < secret.jwe > output-plain.txt
   ```

El pin también admite el sellado de datos a un estado de los Registros de Configuración de la Plataforma (PCR). De este modo, los datos solo pueden desprecintarse si los valores de los hashes de los PCR coinciden con la política utilizada al sellarlos.

Por ejemplo, para sellar los datos al PCR con índice 0 y 1 para el banco SHA-1:

```
$ clevis encrypt tpm2 '{"pcr_bank":"sha1","pcr_ids":"0,1"}' < input-plain.txt > secret.jwe
```

Recursos adicionales

- Para más información y la lista de posibles propiedades de configuración, consulte la página man **clevis-encrypt-tpm2(1)**.

9.9. CONFIGURACIÓN DE LA INSCRIPCIÓN MANUAL DE VOLÚMENES CIFRADOS CON LUKS

Siga los siguientes pasos para configurar el desbloqueo de volúmenes cifrados con LUKS con NBDE.

Requisito previo

- Un servidor Tang está funcionando y está disponible.

Procedimiento

1. Para desbloquear automáticamente un volumen cifrado con LUKS, instale el subpaquete **clevis-luks**:

```
# yum install clevis-luks
```

2. Identifique el volumen cifrado por LUKS para PBD. En el siguiente ejemplo, el dispositivo de bloque se denomina */dev/sda2*:

```
# lsblk
NAME                                          MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINT
sda                                             8:0    0    12G  0 disk
├─sda1                                          8:1    0     1G  0 part  /boot
└─sda2                                          8:2    0    11G  0 part
  └─luks-40e20552-2ade-4954-9d56-565aa7994fb6 253:0    0    11G  0 crypt
    ├─rhel-root                               253:0    0   9.8G  0 lvm   /
    └─rhel-swap                               253:1    0   1.2G  0 lvm   [SWAP]
```

3. Vincule el volumen a un servidor Tang utilizando el comando **clevis luks bind**:

```
# clevis luks bind -d /dev/sda2 tang '{"url":"http://tang.srv"}'
The advertisement contains the following signing keys:

_OsIk0T-E2l6qjfdDiwVmidoZjA

Do you wish to trust these keys? [ynYN] y
You are about to initialize a LUKS device for metadata storage.
Attempting to initialize it may result in data loss if data was
already written into the LUKS header gap in a different format.
A backup is advised before initialization is performed.

Do you wish to initialize /dev/sda2? [yn] y
Enter existing LUKS password:
```

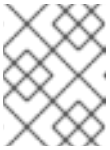
Este comando realiza cuatro pasos:

a. Crea una nueva clave con la misma entropía que la clave maestra LUKS.

b. Cifra la nueva clave con la horquilla.

c. Almacena el objeto Clevis JWE en el token de la cabecera LUKS2 o utiliza LUKSMeta si se utiliza la cabecera LUKS1 no predeterminada.

d. Habilita la nueva clave para su uso con LUKS.



NOTA

El procedimiento de vinculación supone que hay al menos una ranura de contraseña LUKS libre. El comando **clevis luks bind** toma una de las ranuras.

4. El volumen se puede desbloquear ahora con su contraseña existente, así como con la política de la horquilla.

5. Para permitir que el sistema de arranque temprano procese la unión de discos, introduzca los siguientes comandos en un sistema ya instalado:

```
# yum install clevis-dracut
# dracut -fv --regenerate-all
```

Pasos de verificación

1. Para comprobar que el objeto JWE de la horquilla se ha colocado correctamente en una cabecera LUKS, utilice el comando **clevis luks list**:

```
# clevis luks list -d /dev/sda2
1: tang '{"url":"http://tang.srv:port"}'
```

IMPORTANTE

Para utilizar NBDE para clientes con configuración IP estática (sin DHCP), pase su configuración de red a la herramienta dracut manualmente, por ejemplo:

```
# dracut -fv --regenerate-all --kernel-cmdline
"ip=192.0.2.10::192.0.2.1:255.255.255.0::ens3:none:192.0.2.45"
```

Como alternativa, cree un archivo .conf en el directorio **/etc/dracut.conf.d/** con la información de red estática. Por ejemplo:

```
# cat /etc/dracut.conf.d/static_ip.conf
kernel_cmdline="ip=192.0.2.10::192.0.2.1:255.255.255.0::ens3:none:192.0.2.45"
```

Regenerar la imagen inicial del disco RAM:

```
# dracut -fv --regenerate-all
```

Recursos adicionales

Para más información, consulte las siguientes páginas de manual:

- **clevis-luks-bind(1)**
- **dracut.cmdline(7)**

9.10. CONFIGURACIÓN DE LA INSCRIPCIÓN AUTOMATIZADA DE VOLÚMENES CIFRADOS CON LUKS MEDIANTE KICKSTART

Siga los pasos de este procedimiento para configurar un proceso de instalación automatizado que utilice Clevis para la inscripción de volúmenes cifrados con LUKS.

Procedimiento

1. Indique a Kickstart que particione el disco de forma que se habilite el cifrado LUKS para todos los puntos de montaje, excepto **/boot**, con una contraseña temporal. La contraseña es temporal para este paso del proceso de inscripción.

```
part /boot --fstype="xfs" --ondisk=vda --size=256
part / --fstype="xfs" --ondisk=vda --grow --encrypted --passphrase=temppass
```

Tenga en cuenta que los sistemas de reclamación OSPP requieren una configuración más compleja, por ejemplo:

```
part /boot --fstype="xfs" --ondisk=vda --size=256
part / --fstype="xfs" --ondisk=vda --size=2048 --encrypted --passphrase=temppass
part /var --fstype="xfs" --ondisk=vda --size=1024 --encrypted --passphrase=temppass
part /tmp --fstype="xfs" --ondisk=vda --size=1024 --encrypted --passphrase=temppass
part /home --fstype="xfs" --ondisk=vda --size=2048 --grow --encrypted --
passphrase=temppass
part /var/log --fstype="xfs" --ondisk=vda --size=1024 --encrypted --passphrase=temppass
part /var/log/audit --fstype="xfs" --ondisk=vda --size=1024 --encrypted --
passphrase=temppass
```

2. Instale los paquetes de la horquilla relacionados con ella, enumerándolos en la sección **%packages**:

```
%packages
clevis-dracut
%end
```

3. Llame a **clevis luks bind** para realizar la vinculación en la sección **%post**. Después, elimine la contraseña temporal:

```
%post
curl -sfg http://tang.srv/adv -o adv.jws
clevis luks bind -f -k- -d /dev/vda2 \
tang '{"url":"http://tang.srv","adv":"adv.jws"}' \ <<< "temppass"
cryptsetup luksRemoveKey /dev/vda2 <<< "temppass"
%end
```

En el ejemplo anterior, observe que descargamos el anuncio del servidor Tang como parte de nuestra configuración de vinculación, lo que permite que la vinculación sea completamente no interactiva.

AVISO

El comando **cryptsetup luksRemoveKey** impide cualquier administración posterior de un dispositivo LUKS2 en el que se aplique. Puede recuperar una llave maestra eliminada utilizando el comando **dmsetup** sólo para dispositivos LUKS1.

Puede utilizar un procedimiento análogo cuando utilice una política TPM 2.0 en lugar de un servidor Tang.

Recursos adicionales

- **clevis(1)**, **clevis-luks-bind(1)**, **cryptsetup(8)**, y **dmsetup(8)** páginas de manual
- Instalación de Red Hat Enterprise Linux 8 mediante Kickstart

9.11. CONFIGURACIÓN DEL DESBLOQUEO AUTOMÁTICO DE UN DISPOSITIVO DE ALMACENAMIENTO EXTRAÍBLE CIFRADO CON LUKS

Utilice este procedimiento para configurar un proceso de desbloqueo automático de un dispositivo de almacenamiento USB cifrado con LUKS.

Procedimiento

1. Para desbloquear automáticamente un dispositivo de almacenamiento extraíble cifrado con LUKS, como una unidad USB, instale el paquete **clevis-udisks2**:

```
# yum install clevis-udisks2
```

2. Reinicie el sistema y, a continuación, realice el paso de vinculación mediante el comando **clevis luks bind** tal y como se describe en Configuración de la vinculación manual de volúmenes cifrados con LUKS, por ejemplo:

```
# clevis luks bind -d /dev/sdb1 tang '{"url":"http://tang.srv"}'
```

3. El dispositivo extraíble encriptado por LUKS puede ahora ser desbloqueado automáticamente en su sesión de escritorio GNOME. El dispositivo vinculado a una política de Clevis también puede desbloquearse mediante el comando **clevis luks unlock**:

```
# clevis luks unlock -d /dev/sdb1
```

Puede utilizar un procedimiento análogo cuando utilice una política TPM 2.0 en lugar de un servidor Tang.

Recursos adicionales

Para más información, consulte la siguiente página de manual:

- **clevis-luks-unlockers(7)**

9.12. IMPLANTACIÓN DE SISTEMAS NBDE DE ALTA DISPONIBILIDAD

Tang ofrece dos métodos para construir un despliegue de alta disponibilidad:

Redundancia de clientes (recomendada)

Los clientes deben estar configurados con la capacidad de vincularse a múltiples servidores Tang. En esta configuración, cada servidor Tang tiene sus propias claves y los clientes pueden descifrar contactando con un subconjunto de estos servidores. Clevis ya soporta este flujo de trabajo a través de su plug-in **sss**. Red Hat recomienda este método para un despliegue de alta disponibilidad.

Compartir las llaves

Por motivos de redundancia, se puede desplegar más de una instancia de Tang. Para configurar una segunda instancia o cualquier instancia posterior, instale los paquetes **tang** y copie el directorio de claves al nuevo host utilizando **rsync** sobre **SSH**. Tenga en cuenta que Red Hat no recomienda este método porque compartir claves aumenta el riesgo de compromiso de las mismas y requiere una infraestructura de automatización adicional.

9.12.1. NBDE de alta disponibilidad utilizando el secreto compartido de Shamir

La compartición de secretos de Shamir (SSS) es un esquema criptográfico que divide un secreto en varias partes únicas. Para reconstruir el secreto, se requiere un número de partes. El número se denomina umbral y el SSS también se conoce como esquema de umbralización.

Clevis proporciona una implementación de SSS. Crea una clave y la divide en un número de piezas. Cada trozo es encriptado usando otra clavija incluyendo incluso SSS recursivamente. Además, define el umbral **t**. Si un despliegue de NBDE descifra al menos **t** piezas, entonces recupera la clave de cifrado y el proceso de descifrado tiene éxito. Cuando Clevis detecta un número de piezas inferior al especificado en el umbral, imprime un mensaje de error.

9.12.1.1. Ejemplo 1: Redundancia con dos servidores Tang

El siguiente comando descifra un dispositivo cifrado con LUKS cuando al menos uno de los dos servidores Tang está disponible:

```
# clevis luks bind -d /dev/sda1 sss '{"t":1,"pins":{"tang":[{"url":"http://tang1.srv"},
{"url":"http://tang2.srv"}]}}'
```

El comando anterior utilizaba el siguiente esquema de configuración:

```
{
   "t":1,
   "pins":{
      "tang":[
         {
            "url":"http://tang1.srv"
         },
         {
            "url":"http://tang2.srv"
         }
      ]
   }
}
```

En esta configuración, el umbral de SSS **t** se establece en **1** y el comando **clevis luks bind** reconstruye con éxito el secreto si al menos uno de los dos servidores de la lista **tang** está disponible.

9.12.1.2. Ejemplo 2: Secreto compartido en un servidor Tang y un dispositivo TPM

El siguiente comando descifra con éxito un dispositivo cifrado con LUKS cuando tanto el servidor **tang** como el dispositivo **tpm2** están disponibles:

```
# clevis luks bind -d /dev/sda1 sss '{"t":2,"pins":{"tang":[{"url":"http://tang1.srv"}], "tpm2":
{"pcr_ids":"0,1"}}}'
```

El esquema de configuración con el umbral SSS 't' fijado en '2' es ahora:

```
{
    "t":2,
    "pins":{
        "tang":[
            {
                "url":"http://tang1.srv"
            }
        ],
        "tpm2":{
            "pcr_ids":"0,1"
        }
    }
}
```

Recursos adicionales

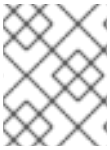
- Para más información sobre la configuración recomendada de alta disponibilidad de NBDE, consulte las siguientes páginas de manual:
  - **tang(8)**, sección **High Availability**
  - **clevis(1)**, sección **Shamir's Secret Sharing**
  - **clevis-encrypt-sss(1)**

9.13. DESPLIEGUE DE MÁQUINAS VIRTUALES EN UNA RED NBDE

El comando **clevis luks bind** no cambia la clave maestra de LUKS. Esto implica que si creas una imagen cifrada con LUKS para usarla en una máquina virtual o en un entorno de nube, todas las instancias que ejecuten esta imagen compartirán una clave maestra. Esto es extremadamente inseguro y debe evitarse en todo momento.

Esto no es una limitación de Clevis sino un principio de diseño de LUKS. Si desea tener volúmenes raíz encriptados en una nube, necesita asegurarse de que realiza el proceso de instalación (normalmente usando Kickstart) para cada instancia de Red Hat Enterprise Linux en una nube también. Las imágenes no pueden ser compartidas sin compartir también una llave maestra LUKS.

Si pretende implementar el desbloqueo automatizado en un entorno virtualizado, Red Hat le recomienda encarecidamente que utilice sistemas como lorax o virt-install junto con un archivo Kickstart (véase Configuración de la inscripción automatizada de volúmenes encriptados por LUKS utilizando Kickstart) u otra herramienta de aprovisionamiento automatizado para garantizar que cada VM encriptada tenga una clave maestra única.



NOTA

El desbloqueo automático con una política TPM 2.0 no es compatible con una máquina virtual.

Recursos adicionales

Para más información, consulte la siguiente página de manual:

- **clevis-luks-bind(1)**

9.14. CREACIÓN DE IMÁGENES DE MÁQUINAS VIRTUALES AUTOMÁTICAMENTE INSCRIBIBLES PARA ENTORNOS DE NUBE MEDIANTE NBDE

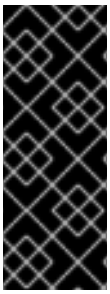
El despliegue de imágenes encriptadas automáticamente en un entorno de nube puede proporcionar un conjunto único de desafíos. Al igual que en otros entornos de virtualización, se recomienda reducir el número de instancias iniciadas a partir de una sola imagen para evitar compartir la clave maestra LUKS.

Por lo tanto, la mejor práctica es crear imágenes personalizadas que no se compartan en ningún repositorio público y que proporcionen una base para el despliegue de una cantidad limitada de instancias. El número exacto de instancias a crear debe ser definido por las políticas de seguridad del despliegue y basado en la tolerancia al riesgo asociado al vector de ataque de la llave maestra LUKS.

Para construir despliegues automatizados con LUKS, se deben utilizar sistemas como Lorax o virt-install junto con un archivo Kickstart para asegurar la unicidad de la llave maestra durante el proceso de construcción de la imagen.

Los entornos de nube permiten dos opciones de despliegue del servidor Tang que consideramos aquí. En primer lugar, el servidor Tang puede desplegarse dentro del propio entorno de la nube. En segundo lugar, el servidor Tang puede desplegarse fuera de la nube en una infraestructura independiente con un enlace VPN entre las dos infraestructuras.

El despliegue de Tang de forma nativa en la nube permite un fácil despliegue. Sin embargo, dado que comparte infraestructura con la capa de persistencia de datos de texto cifrado de otros sistemas, puede ser posible que tanto la clave privada del servidor Tang como los metadatos de Clevis se almacenen en el mismo disco físico. El acceso a este disco físico permite comprometer completamente los datos del texto cifrado.



IMPORTANTE

Por esta razón, Red Hat recomienda encarecidamente mantener una separación física entre la ubicación donde se almacenan los datos y el sistema donde se ejecuta Tang. Esta separación entre la nube y el servidor Tang garantiza que la clave privada del servidor Tang no pueda combinarse accidentalmente con los metadatos de Clevis. También proporciona un control local del servidor Tang si la infraestructura de la nube está en peligro.

9.15. INTRODUCCIÓN A LAS FUNCIONES DEL SISTEMA DE HORQUILLA Y TANG

RHEL System Roles es una colección de roles y módulos de Ansible que proporcionan una interfaz de configuración consistente para gestionar remotamente múltiples sistemas RHEL.

RHEL 8.3 introdujo los roles de Ansible para el despliegue automatizado de soluciones de descifrado basado en políticas (PBD) utilizando Clevis y Tang. El paquete **rhel-system-roles** contiene estos roles de sistema, los ejemplos relacionados y también la documentación de referencia.

El rol de sistema **nbde_client** le permite desplegar múltiples clientes Clevis de forma automatizada. Tenga en cuenta que el rol **nbde_client** sólo admite enlaces Tang, y no puede utilizarlo para enlaces TPM2 por el momento.

El rol **nbde_client** requiere volúmenes que ya están encriptados usando LUKS. Esta función permite vincular un volumen cifrado con LUKS a uno o más servidores vinculados a la red (NBDE) - servidores Tang. Puede conservar el cifrado del volumen existente con una frase de contraseña o eliminarla. Una

vez eliminada la frase de contraseña, puede desbloquear el volumen sólo con NBDE. Esto es útil cuando un volumen está inicialmente encriptado utilizando una clave o contraseña temporal que debe eliminar después de aprovisionar el sistema.

Si proporciona tanto una frase de contraseña como un archivo de claves, el rol utiliza lo que ha proporcionado primero. Si no encuentra ninguno de ellos válido, intenta recuperar una frase de contraseña de un enlace existente.

PBD define una vinculación como una asignación de un dispositivo a una ranura. Esto significa que se pueden tener varios enlaces para el mismo dispositivo. La ranura por defecto es la ranura 1.

El rol **nbde_client** proporciona también la variable **state**. Utilice el valor **present** para crear un nuevo enlace o actualizar uno existente. Al contrario que el comando **clevis luks bind**, puede utilizar **state: present** también para sobrescribir un enlace existente en su ranura de dispositivo. El valor **absent** elimina un enlace especificado.

Utilizando el rol **nbde_server**, puede desplegar y gestionar un servidor Tang como parte de una solución de encriptación de disco automatizada. Este rol soporta las siguientes características:

- Llaves Tang giratorias
- Despliegue y copia de seguridad de las llaves Tang

Recursos adicionales

- Para una referencia detallada sobre las variables de rol Network-Bound Disk Encryption (NBDE), instale el paquete **rhel-system-roles**, y vea los archivos **README.md** y **README.html** en los directorios **/usr/share/doc/rhel-system-roles/nbde_client/** y **/usr/share/doc/rhel-system-roles/nbde_server/**.
- Para ver ejemplos de playbooks de system-roles, instale el paquete **rhel-system-roles**, y vea los directorios **/usr/share/ansible/roles/rhel-system-roles.nbde_server/examples/**.
- Para obtener más información sobre las funciones del sistema RHEL, consulte Introducción a las funciones del sistema RHEL

9.16. USO DEL ROL DE SISTEMA NBDE_SERVER PARA CONFIGURAR MÚLTIPLES SERVIDORES TANG

Siga los pasos para preparar y aplicar un playbook de Ansible que contenga la configuración de su servidor Tang.

Requisitos previos

- Su suscripción a Red Hat Ansible Engine está conectada al sistema. Consulte el artículo Cómo descargar e instalar Red Hat Ansible Engine para obtener más información.

Procedimiento

1. Habilitar el repositorio RHEL Ansible, por ejemplo:

```
# subscription-manager repos --enable ansible-2-for-rhel-8-x86_64-rpms
```

2. Instale el motor Ansible:

```
# yum install ansible
```

3. Instalar los roles del sistema RHEL:

```
# yum install rhel-system-roles
```

4. Prepare su libro de jugadas con la configuración de los servidores Tang. Puede empezar desde cero o utilizar uno de los libros de juego de ejemplo del directorio **/usr/share/ansible/roles/rhel-system-roles.nbde_server/examples/**.

```
# cp /usr/share/ansible/roles/rhel-system-roles.nbde_server/examples/simple_deploy.yml
./my-tang-playbook.yml
```

5. Edite el libro de jugadas en un editor de texto de su elección, por ejemplo:

```
# vi my-tang-playbook.yml
```

6. Añada los parámetros necesarios. El siguiente ejemplo de libro de jugadas asegura el despliegue de su servidor Tang y una rotación de llaves:

```
---
- hosts: all

  vars:
    nbde_server_rotate_keys: yes

  roles:
    - linux-system-roles.nbde_server
```

7. Aplicar el libro de jugadas terminado:

```
# ansible-playbook -i host1,host2,host3 my-tang-playbook.yml
```

Recursos adicionales

- Para más información, instale el paquete **rhel-system-roles** y consulte los directorios **/usr/share/doc/rhel-system-roles/nbde_server/** y **usr/share/ansible/roles/rhel-system-roles.nbde_server/**.

9.17. USO DE LA FUNCIÓN DEL SISTEMA NBDE_CLIENT PARA CONFIGURAR VARIOS CLIENTES CLEVIS

Siga los pasos para preparar y aplicar un libro de jugadas de Ansible que contenga su configuración de Clevis-client.

NOTA

El rol de sistema **nbde_client** sólo admite enlaces Tang. Esto significa que, por el momento, no se puede utilizar para los enlaces TPM2.

Requisitos previos

- Su suscripción a Red Hat Ansible Engine está conectada al sistema. Consulte el artículo Cómo descargar e instalar Red Hat Ansible Engine para obtener más información.
- Sus volúmenes ya están encriptados por LUKS.

Procedimiento

1. Habilitar el repositorio RHEL Ansible, por ejemplo:

```
# subscription-manager repos --enable ansible-2-for-rhel-8-x86_64-rpms
```

2. Instale el motor Ansible:

```
# yum install ansible
```

3. Instalar los roles del sistema RHEL:

```
# yum install rhel-system-roles
```

4. Prepare su libro de jugadas con la configuración de los clientes de Clevis. Puede empezar desde cero o utilizar uno de los libros de juego de ejemplo del directorio **/usr/share/ansible/roles/rhel-system-roles.nbde_client/examples/**.

```
# cp /usr/share/ansible/roles/rhel-system-roles.nbde_client/examples/high_availability.yml
./my-clevis-playbook.yml
```

5. Edite el libro de jugadas en un editor de texto de su elección, por ejemplo:

```
# vi my-clevis-playbook.yml
```

6. Añada los parámetros necesarios. El siguiente ejemplo de libro de jugadas configura los clientes Clevis para el desbloqueo automático de dos volúmenes cifrados con LUKS cuando al menos uno de los dos servidores Tang está disponible:

```
---
- hosts: all

  vars:
    nbde_client_bindings:
      - device: /dev/rhel/root
        encryption_key_src: /etc/luks/keyfile
        servers:
          - http://server1.example.com
          - http://server2.example.com
      - device: /dev/rhel/swap
        encryption_key_src: /etc/luks/keyfile
        servers:
          - http://server1.example.com
          - http://server2.example.com

  roles:
    - linux-system-roles.nbde_client
```

7. Aplicar el libro de jugadas terminado:

```
# ansible-playbook -i host1,host2,host3 my-clevis-playbook.yml
```

Recursos adicionales

- Para obtener detalles sobre los parámetros e información adicional sobre la función **nbde_client**, instale el paquete **rhel-system-roles** y consulte los directorios **/usr/share/doc/rhel-system-roles/nbde_client/** y **/usr/share/ansible/roles/rhel-system-roles.nbde_client/**.

9.18. RECURSOS ADICIONALES

- Las páginas de manual **tang(8)**, **clevis(1)**, **jose(1)**, y **clevis-luks-unlockers(7)**.
- El artículo de la base de conocimientos Cómo configurar el cifrado de discos en red con varios dispositivos LUKS (desbloqueo de la horquilla).

CAPÍTULO 10. AUDITORÍA DEL SISTEMA

La auditoría no proporciona seguridad adicional a su sistema; más bien, puede utilizarse para descubrir violaciones de las políticas de seguridad utilizadas en su sistema. Estas violaciones pueden evitarse además con medidas de seguridad adicionales como SELinux.

10.1. AUDITORÍA LINUX

El sistema de Auditoría de Linux proporciona una manera de rastrear la información relevante para la seguridad en su sistema. Basado en reglas preconfiguradas, Audit genera entradas de registro para registrar tanta información como sea posible sobre los eventos que están ocurriendo en su sistema. Esta información es crucial para que los entornos de misión crítica puedan determinar quién ha violado la política de seguridad y las acciones que ha realizado.

La siguiente lista resume parte de la información que Audit es capaz de registrar en sus archivos de registro:

- Fecha y hora, tipo y resultado de un evento.
- Etiquetas de sensibilidad de sujetos y objetos.
- Asociación de un evento con la identidad del usuario que lo ha provocado.
- Todas las modificaciones de la configuración de Auditoría y los intentos de acceso a los archivos de registro de Auditoría.
- Todos los usos de los mecanismos de autenticación, como SSH, Kerberos y otros.
- Cambios en cualquier base de datos de confianza, como **/etc/passwd**.
- Intentos de importar o exportar información hacia o desde el sistema.
- Incluya o excluya eventos en función de la identidad del usuario, las etiquetas de sujetos y objetos, y otros atributos.

El uso del sistema Audit también es un requisito para una serie de certificaciones relacionadas con la seguridad. Audit está diseñado para cumplir o superar los requisitos de las siguientes certificaciones o guías de cumplimiento:

- Perfil de protección de acceso controlado (CAPP)
- Perfil de protección de seguridad etiquetado (LSPP)
- Control de acceso basado en conjuntos de reglas (RSBAC)
- Manual operativo del Programa Nacional de Seguridad Industrial (NISPOM)
- Ley Federal de Gestión de la Seguridad de la Información (FISMA)
- Industria de las tarjetas de pago
- Guías técnicas de aplicación de la seguridad (STIG)

La auditoría también lo ha sido:

- Evaluado por National Information Assurance Partnership (NIAP) y Best Security Industries (BSI).

- Certificado para LSPP/CAPP/RSBAC/EAL4 en Red Hat Enterprise Linux 5.
- Certificado para el Perfil de Protección del Sistema Operativo / Nivel de Garantía de Evaluación 4 (OSPP/EAL4) en Red Hat Enterprise Linux 6.

Casos de uso

Vigilancia del acceso a los archivos

La auditoría puede rastrear si se ha accedido a un archivo o a un directorio, si se ha modificado, si se ha ejecutado o si se han cambiado los atributos del archivo. Esto es útil, por ejemplo, para detectar el acceso a archivos importantes y tener un rastro de Auditoría disponible en caso de que uno de estos archivos se corrompa.

Supervisión de las llamadas del sistema

La auditoría puede configurarse para generar una entrada de registro cada vez que se utiliza una llamada del sistema en particular. Esto puede utilizarse, por ejemplo, para rastrear los cambios en la hora del sistema mediante la supervisión de las llamadas al sistema **settimeofday**, **clock_adjtime**, y otras relacionadas con la hora.

Grabación de los comandos ejecutados por un usuario

La auditoría puede rastrear si un archivo ha sido ejecutado, por lo que se pueden definir reglas para registrar cada ejecución de un comando en particular. Por ejemplo, se puede definir una regla para cada ejecutable en el directorio **/bin**. Las entradas de registro resultantes pueden buscarse por ID de usuario para generar un registro de auditoría de los comandos ejecutados por usuario.

Grabación de la ejecución de los nombres de ruta del sistema

Además de vigilar el acceso a los archivos que traduce una ruta a un inodo en la invocación de la regla, Auditoría puede ahora vigilar la ejecución de una ruta incluso si no existe en el momento de la invocación de la regla, o si el archivo es reemplazado después de la invocación de la regla. Esto permite que las reglas sigan funcionando después de actualizar el ejecutable de un programa o incluso antes de instalarlo.

Registro de eventos de seguridad

El módulo de autenticación **pam_faillock** es capaz de registrar los intentos fallidos de inicio de sesión. La auditoría puede configurarse para registrar también los intentos de inicio de sesión fallidos y proporciona información adicional sobre el usuario que intentó iniciar sesión.

Búsqueda de eventos

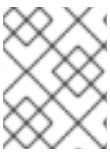
Audit proporciona la utilidad **ausearch**, que se puede utilizar para filtrar las entradas del registro y proporcionar una pista de auditoría completa basada en varias condiciones.

Ejecución de informes de síntesis

La utilidad **aureport** puede utilizarse para generar, entre otras cosas, informes diarios de los eventos registrados. Un administrador del sistema puede entonces analizar estos informes e investigar más a fondo las actividades sospechosas.

Supervisión del acceso a la red

Las utilidades **iptables** y **ebtables** pueden ser configuradas para desencadenar eventos de Auditoría, permitiendo a los administradores del sistema monitorear el acceso a la red.



NOTA

El rendimiento del sistema puede verse afectado en función de la cantidad de información que recoja la Auditoría.

10.2. ARQUITECTURA DEL SISTEMA DE AUDITORÍA

El sistema de auditoría consta de dos partes principales: las aplicaciones y utilidades del espacio de usuario y el procesamiento de las llamadas al sistema del lado del núcleo. El componente del núcleo recibe las llamadas al sistema de las aplicaciones del espacio del usuario y las filtra a través de uno de los siguientes filtros: **user**, **task**, **fstype**, o **exit**.

Una vez que una llamada al sistema pasa el filtro **exclude**, es enviada a través de uno de los filtros mencionados, el cual, basado en la configuración de la regla de Auditoría, la envía al demonio de Auditoría para su posterior procesamiento.

El demonio de Auditoría del espacio de usuario recoge la información del kernel y crea entradas en un archivo de registro. Otras utilidades de espacio de usuario de Auditoría interactúan con el demonio de Auditoría, el componente de Auditoría del kernel o los archivos de registro de Auditoría:

- **auditctl**
- Las restantes utilidades de Auditoría toman el contenido de los archivos de registro de Auditoría como entrada y generan una salida basada en los requerimientos del usuario. Por ejemplo, la utilidad **aureport** genera un informe de todos los eventos registrados.

En RHEL 8, la funcionalidad del demonio despachador de Auditoría (**audisp**) está integrada en el demonio de Auditoría (**auditd**). Los archivos de configuración de los plugins para la interacción de los programas analíticos en tiempo real con los eventos de Audit se encuentran por defecto en el directorio **/etc/audit/plugins.d/**.

10.3. CONFIGURACIÓN DE AUDITD PARA UN ENTORNO SEGURO

La configuración por defecto de **auditd** debería ser adecuada para la mayoría de los entornos. Sin embargo, si su entorno tiene que cumplir con políticas de seguridad estrictas, se sugieren los siguientes ajustes para la configuración del demonio de auditoría en el archivo **/etc/audit/auditd.conf**:

archivo_de_registro

El directorio que contiene los archivos de registro de Auditoría (normalmente **/var/log/audit/**) debería residir en un punto de montaje separado. Esto evita que otros procesos consuman espacio en este directorio y proporciona una detección precisa del espacio restante para el demonio de Auditoría.

archivo_de_registro_máximo

Especifica el tamaño máximo de un solo archivo de registro de Auditoría, debe establecerse para hacer uso completo del espacio disponible en la partición que contiene los archivos de registro de Auditoría.

max_log_file_action

Decide qué acción se lleva a cabo una vez que se alcanza el límite establecido en **max_log_file**, debería establecerse en **keep_logs** para evitar que se sobrescriban los archivos de registro de auditoría.

espacio_izquierdo

Especifica la cantidad de espacio libre que queda en el disco para que se active una acción establecida en el parámetro **space_left_action**. Debe establecerse un número que dé al administrador tiempo suficiente para responder y liberar espacio en el disco. El valor de **space_left** depende de la velocidad a la que se generan los archivos de registro de auditoría.

acción_espacio_izquierda

Se recomienda establecer el parámetro **space_left_action** en **email** o **exec** con un método de notificación apropiado.

espacio_administrador_izquierdo

Especifica la cantidad mínima absoluta de espacio libre para la cual se desencadena una acción establecida en el parámetro **admin_space_left_action**, debe establecerse en un valor que deje suficiente espacio para registrar las acciones realizadas por el administrador.

admin_space_left_action

Se debe establecer en **single** para poner el sistema en modo monopuesto y permitir al administrador liberar algo de espacio en el disco.

disk_full_action

Especifica una acción que se desencadena cuando no hay espacio libre disponible en la partición que contiene los archivos de registro de Auditoría, debe establecerse en **halt** o **single**. Esto asegura que el sistema se apague o funcione en modo monopuesto cuando Audit no pueda registrar más eventos.

acción_error_disco

Especifica una acción que se desencadena en caso de que se detecte un error en la partición que contiene los archivos de registro de auditoría, debe establecerse en **syslog**, **single**, o **halt**, dependiendo de sus políticas locales de seguridad en relación con el manejo de los fallos de hardware.

descarga

Debe establecerse en **incremental_async**. Funciona en combinación con el parámetro **freq**, que determina cuántos registros pueden enviarse al disco antes de forzar una sincronización con el disco duro. El parámetro **freq** debe ajustarse a **100**. Estos parámetros aseguran que los datos de los eventos de Auditoría estén sincronizados con los archivos de registro en el disco, manteniendo un buen rendimiento para las ráfagas de actividad.

El resto de las opciones de configuración deben establecerse de acuerdo con su política de seguridad local.

10.4. INICIO Y CONTROL DE LA AUDITORÍA

Una vez configurado **auditd**, inicie el servicio para recoger la información de auditoría y almacenarla en los archivos de registro. Utiliza el siguiente comando como usuario root para iniciar **auditd**:

```
# service auditd start
```

Para configurar **auditd** para que se inicie en el momento del arranque:

```
# systemctl enable auditd
```

Se pueden realizar otras acciones en **auditd** utilizando el comando **service auditd *action*** donde *action* puede ser uno de los siguientes:

stop

Para **auditd**.

restart

Reinicia **auditd**.

reload o force-reload

Recarga la configuración de **auditd** desde el archivo **/etc/audit/auditd.conf**.

rotate

Rota los archivos de registro en el directorio **/var/log/audit/**.

resume

Reanuda el registro de eventos de Auditoría después de haber sido suspendido previamente, por ejemplo, cuando no hay suficiente espacio libre en la partición del disco que contiene los archivos de registro de Auditoría.

condrestart o **try-restart**

Reinicia **auditd** sólo si ya está en marcha.

status

Muestra el estado de funcionamiento de **auditd**.

NOTA

El comando **service** es la única manera de interactuar correctamente con el demonio **auditd**. Es necesario utilizar el comando **service** para que el valor de **auid** se registre correctamente. Puede utilizar el comando **systemctl** sólo para dos acciones: **enable** y **status**.

10.5. COMPRENSIÓN DE LOS ARCHIVOS DE REGISTRO DE AUDITORÍA

Por defecto, el sistema de Auditoría almacena las entradas de registro en el archivo **/var/log/audit/audit.log**; si se activa la rotación de registros, los archivos **audit.log** rotados se almacenan en el mismo directorio.

Añada la siguiente regla de auditoría para registrar cada intento de lectura o modificación del archivo **/etc/ssh/sshd_config**:

```
# auditctl -w /etc/ssh/sshd_config -p warx -k sshd_config
```

Si el demonio **auditd** se está ejecutando, por ejemplo, el siguiente comando crea un nuevo evento en el archivo de registro de auditoría:

```
$ cat /etc/ssh/sshd_config
```

Este evento en el archivo **audit.log** tiene el siguiente aspecto:

```
type=SYSCALL msg=audit(1364481363.243:24287): arch=c000003e syscall=2 success=no exit=-13
a0=7fffd19c5592 a1=0 a2=7fffd19c4b50 a3=a items=1 ppid=2686 pid=3538 auid=1000 uid=1000
gid=1000 euid=1000 suid=1000 fsuid=1000 egid=1000 sgid=1000 fsgid=1000 tty=pts0 ses=1
comm="cat" exe="/bin/cat" subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
key="sshd_config"
type=CWD msg=audit(1364481363.243:24287):  cwd="/home/shadowman"
type=PATH msg=audit(1364481363.243:24287): item=0 name="/etc/ssh/sshd_config" inode=409248
dev=fd:00 mode=0100600 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:etc_t:s0
nametype=NORMAL cap_fp=none cap_fi=none cap_fe=0 cap_fver=0
type=PROCTITLE msg=audit(1364481363.243:24287) :
proctitle=636174002F6574632F7373682F737368645F636F6E666967
```

El evento anterior consta de cuatro registros, que comparten el mismo sello de tiempo y número de serie. Los registros siempre comienzan con la palabra clave **type=**. Cada registro consta de varios ***name=value*** pares separados por un espacio en blanco o una coma. A continuación se presenta un análisis detallado del suceso anterior:

Primer disco

type=SYSCALL

El campo **type** contiene el tipo de registro. En este ejemplo, el valor **SYSCALL** especifica que este registro fue activado por una llamada del sistema al kernel.

msg=audit(1364481363.243:24287):

El campo **msg** registra:

- un sello de tiempo y un ID único del registro en la forma **audit(*time_stamp*:*ID*)**. Varios registros pueden compartir la misma marca de tiempo e ID si se generaron como parte del mismo evento de auditoría. El sello de tiempo utiliza el formato de tiempo Unix - segundos desde las 00:00:00 UTC del 1 de enero de 1970.

- varios pares de eventos específicos ***name=value*** proporcionados por el kernel o las aplicaciones del espacio de usuario.

arch=c000003e

El campo **arch** contiene información sobre la arquitectura de la CPU del sistema. El valor, **c000003e**, está codificado en notación hexadecimal. Cuando busque registros de auditoría con el comando **ausearch**, utilice la opción **-i** o **--interpret** para convertir automáticamente los valores hexadecimales en sus equivalentes legibles para el ser humano. El valor **c000003e** se interpreta como **x86_64**.

syscall=2

El campo **syscall** registra el tipo de llamada al sistema que se envió al kernel. El valor, **2**, puede ser comparado con su equivalente legible por humanos en el archivo **/usr/include/asm/unistd_64.h**. En este caso, **2** es la llamada al sistema **open**. Tenga en cuenta que la utilidad **ausyscall** le permite convertir los números de las llamadas al sistema en sus equivalentes legibles para el ser humano. Utilice el comando **ausyscall --dump** para mostrar un listado de todas las llamadas al sistema junto con sus números. Para más información, consulte la página de manual **ausyscall**(8).

success=no

El campo **success** registra si la llamada al sistema registrada en ese evento concreto tuvo éxito o fracasó. En este caso, la llamada no tuvo éxito.

exit=-13

El campo **exit** contiene un valor que especifica el código de salida devuelto por la llamada al sistema. Este valor varía según la llamada al sistema. Puede interpretar el valor a su equivalente legible para humanos con el siguiente comando:

```
# ausearch --interpret --exit -13
```

Tenga en cuenta que el ejemplo anterior asume que su registro de auditoría contiene un evento que falló con el código de salida **-13**.

a0=7fffd19c5592, a1=0, a2=7fffd19c5592, a3=a

Los campos **a0** a **a3** registran los cuatro primeros argumentos, codificados en notación hexadecimal, de la llamada al sistema en este evento. Estos argumentos dependen de la llamada al sistema que se utilice; pueden ser interpretados por la utilidad **ausearch**.

items=1

El campo **items** contiene el número de registros auxiliares PATH que siguen al registro syscall.

ppid=2686

El campo **ppid** registra el ID del proceso padre (PPID). En este caso, **2686** era el PPID del proceso padre como **bash**.

pid=3538

El campo **pid** registra el ID del proceso (PID). En este caso, **3538** era el PID del proceso **cat**.

auid=1000

El campo **auid** registra el ID de usuario de la auditoría, es decir, el loginuid. Este ID se asigna a un usuario al iniciar la sesión y es heredado por todos los procesos, incluso cuando la identidad del usuario cambia, por ejemplo, al cambiar de cuenta de usuario con el comando **su - john**.

uid=1000

El campo **uid** registra el ID del usuario que inició el proceso analizado. El ID de usuario puede ser interpretado en nombres de usuario con el siguiente comando **ausearch -i --uid *UID***.

gid=1000

El campo **gid** registra el ID del grupo del usuario que inició el proceso analizado.

euid=1000

El campo **euid** registra el ID de usuario efectivo del usuario que inició el proceso analizado.

suid=1000

En el campo **suid** se registra el ID del usuario que inició el proceso analizado.

fsuid=1000

El campo **fsuid** registra el ID de usuario del sistema de archivos del usuario que inició el proceso analizado.

egid=1000

El campo **egid** registra el ID de grupo efectivo del usuario que inició el proceso analizado.

sgid=1000

En el campo **sgid** se registra el ID del grupo del usuario que inició el proceso analizado.

fsgid=1000

El campo **fsgid** registra el ID del grupo del sistema de archivos del usuario que inició el proceso analizado.

tty=pts0

El campo **tty** registra el terminal desde el que se invocó el proceso analizado.

ses=1

El campo **ses** registra el ID de la sesión desde la que se invocó el proceso analizado.

comm="cat"

El campo **comm** registra el nombre de la línea de comandos que se utilizó para invocar el proceso analizado. En este caso, se utilizó el comando **cat** para activar este evento de Auditoría.

exe="/bin/cat"

El campo **exe** registra la ruta del ejecutable que se utilizó para invocar el proceso analizado.

subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

El campo **subj** registra el contexto SELinux con el que el proceso analizado fue etiquetado en el momento de la ejecución.

key="sshd_config"

El campo **key** registra la cadena definida por el administrador asociada a la regla que generó este evento en el registro de auditoría.

Segundo disco

type=CWD

En el segundo registro, el valor del campo **type** es **CWD**
El propósito de este registro es registrar la ubicación del proceso actual en caso de que una ruta relativa termine siendo capturada en el registro PATH asociado. De esta manera se puede reconstruir la ruta absoluta.

msg=audit(1364481363.243:24287)

El campo **msg** contiene la misma marca de tiempo y valor de identificación que el valor del primer registro. El sello de tiempo utiliza el formato de tiempo Unix - segundos desde las 00:00:00 UTC del 1 de enero de 1970.

cwd="/home/*user_name*"

El campo **cwd** contiene la ruta del directorio en el que se invocó la llamada al sistema.

Tercer disco

type=PATH

En el tercer registro, el valor del campo **type** es **PATH**. Un evento de Auditoría contiene un registro de tipo **PATH** para cada ruta que se pasa a la llamada del sistema como un argumento. En este evento de auditoría, sólo se utilizó una ruta (**/etc/ssh/sshd_config**) como argumento.

msg=audit(1364481363.243:24287):

El campo **msg** contiene el mismo sello de tiempo y valor de identificación que el valor del primer y segundo registro.

item=0

El campo **item** indica de qué artículo, del total de artículos referenciados en el registro de tipo **SYSCALL**, se trata el registro actual. Este número está basado en cero; un valor de **0** significa que es el primer elemento.

name="/etc/ssh/sshd_config"

El campo **name** registra la ruta del archivo o directorio que se pasó a la llamada del sistema como argumento. En este caso, fue el archivo **/etc/ssh/sshd_config**.

inode=409248

El campo **inode** contiene el número de inodo asociado al archivo o directorio registrado en este evento. El siguiente comando muestra el archivo o directorio que está asociado con el número de inodo **409248**:

```
# find / -inum 409248 -print
/etc/ssh/sshd_config
```

dev=fd:00

El campo **dev** especifica el ID menor y mayor del dispositivo que contiene el archivo o directorio registrado en este evento. En este caso, el valor representa el dispositivo **/dev/fd/0**.

mode=0100600

El campo **mode** registra los permisos de los archivos o directorios, codificados en notación numérica tal y como los devuelve el comando **stat** en el campo **st_mode**. Consulte la página de manual **stat(2)** para obtener más información. En este caso, **0100600** puede interpretarse como **-rw-------**, lo que significa que sólo el usuario root tiene permisos de lectura y escritura en el archivo **/etc/ssh/sshd_config**.

ouid=0

El campo **ouid** registra el ID de usuario del propietario del objeto.

ogid=0

El campo **ogid** registra el ID del grupo del propietario del objeto.

rdev=00:00

El campo **rdev** contiene un identificador de dispositivo grabado sólo para archivos especiales. En este caso, no se utiliza ya que el archivo grabado es un archivo normal.

obj=system_u:object_r:etc_t:s0

El campo **obj** registra el contexto SELinux con el que el archivo o directorio registrado fue etiquetado en el momento de la ejecución.

nametype=NORMAL

El campo **nametype** registra la intención de la operación de cada registro de ruta en el contexto de una determinada syscall.

cap_fp=none

El campo **cap_fp** registra datos relacionados con la configuración de una capacidad permitida basada en el sistema de archivos del objeto de archivo o directorio.

cap_fi=none

El campo **cap_fi** registra datos relacionados con la configuración de una capacidad heredada basada en el sistema de archivos del objeto de archivo o directorio.

cap_fe=0

El campo **cap_fe** registra la configuración del bit efectivo de la capacidad basada en el sistema de archivos del objeto de archivo o directorio.

cap_fver=0

El campo **cap_fver** registra la versión de la capacidad basada en el sistema de archivos del objeto de archivo o directorio.

Cuarto disco

type=PROCTITLE

El campo **type** contiene el tipo de registro. En este ejemplo, el valor **PROCTITLE** especifica que este registro da la línea de comandos completa que desencadenó este evento de Auditoría, desencadenado por una llamada del sistema al kernel.

proctitle=636174002F6574632F7373682F737368645F636F6E666967

El campo **proctitle** registra la línea de comandos completa del comando que se utilizó para invocar el proceso analizado. El campo está codificado en notación hexadecimal para no permitir que el usuario influya en el analizador del registro de Auditoría. El texto se decodifica al comando que desencadenó este evento de Auditoría. Al buscar registros de Auditoría con el comando **ausearch**, utilice la opción **-i** o **--interpret** para convertir automáticamente los valores hexadecimales en sus equivalentes legibles para el ser humano. El valor **636174002F6574632F7373682F737368645F636F6E666967** se interpreta como **cat /etc/ssh/sshd_config**.

10.6. USO DE AUDITCTL PARA DEFINIR Y EJECUTAR REGLAS DE AUDITORÍA

El sistema de auditoría funciona con un conjunto de reglas que definen lo que se captura en los archivos de registro. Las reglas de auditoría pueden establecerse en la línea de comandos mediante la utilidad **auditctl** o en el directorio **/etc/audit/rules.d/**.

El comando **auditctl** le permite controlar la funcionalidad básica del sistema de Auditoría y definir las reglas que deciden qué eventos de Auditoría se registran.

Ejemplos de reglas del sistema de archivos

1. Para definir una regla que registre todos los accesos de escritura y todos los cambios de atributos del archivo **/etc/passwd**:

```
# auditctl -w /etc/passwd -p wa -k passwd_changes
```

2. Para definir una regla que registre todos los accesos de escritura y todos los cambios de atributos de todos los archivos del directorio **/etc/selinux/**:

```
# auditctl -w /etc/selinux/ -p wa -k selinux_changes
```

Ejemplos de reglas de llamada al sistema

1. Para definir una regla que cree una entrada de registro cada vez que las llamadas al sistema **adjtimex** o **settimeofday** sean utilizadas por un programa, y el sistema utilice la arquitectura de 64 bits:

```
# auditctl -a always,exit -F arch=b64 -S adjtimex -S settimeofday -k time_change
```

2. Para definir una regla que cree una entrada en el registro cada vez que un usuario del sistema cuyo ID es igual o superior a 1000 elimine o cambie el nombre de un archivo:

```
# auditctl -a always,exit -S unlink -S unlinkat -S rename -S renameat -F auid>=1000 -F auid!=4294967295 -k delete
```

Tenga en cuenta que la opción **-F auid!=4294967295** se utiliza para excluir a los usuarios cuyo UID de inicio de sesión no está establecido.

Reglas de los archivos ejecutables

Para definir una regla que registre toda la ejecución del programa **/bin/id**, ejecute el siguiente comando:

```
# auditctl -a always,exit -F exe=/bin/id -F arch=b64 -S execve -k execution_bin_id
```

Recursos adicionales

- Consulte la página man **audictl(8)** para obtener más información, consejos de rendimiento y ejemplos adicionales de uso.

10.7. DEFINICIÓN DE REGLAS DE AUDITORÍA PERSISTENTES

Para definir reglas de Auditoría que sean persistentes a través de los reinicios, debe incluirlas directamente en el archivo **/etc/audit/rules.d/audit.rules** o utilizar el programa **augenrules** que lee las reglas ubicadas en el directorio **/etc/audit/rules.d/**.

Tenga en cuenta que el archivo **/etc/audit/audit.rules** se genera cada vez que se inicia el servicio **auditd**. Los archivos de **/etc/audit/rules.d/** utilizan la misma sintaxis de la línea de comandos de **auditctl** para especificar las reglas. Las líneas vacías y el texto que sigue a un signo de almohadilla (#) se ignoran.

Además, puede utilizar el comando **auditctl** para leer las reglas de un archivo específico utilizando la opción **-R**, por ejemplo:

```
# auditctl -R /usr/share/audit/sample-rules/30-stig.rules
```

10.8. USO DE ARCHIVOS DE REGLAS PRECONFIGURADOS

En el directorio **/usr/share/audit/sample-rules**, el paquete **audit** proporciona un conjunto de archivos de reglas preconfiguradas según varias normas de certificación:

30-nispom.rules

Configuración de la regla de auditoría que cumple con los requisitos especificados en el capítulo de Seguridad del Sistema de Información del Manual Operativo del Programa Nacional de Seguridad Industrial.

30-ospp-v42*.rules

Configuración de la regla de auditoría que cumple con los requisitos definidos en el perfil OSPP (Protection Profile for General Purpose Operating Systems) versión 4.2.

30-pci-dss-v31.rules

Configuración de la regla de auditoría que cumple los requisitos establecidos por la norma de seguridad de datos del sector de las tarjetas de pago (PCI DSS) v3.1.

30-reglas de la estigmatización

Configuración de la regla de auditoría que cumple con los requisitos establecidos por las Guías Técnicas de Implementación de Seguridad (STIG).

Para utilizar estos archivos de configuración, cópielos en el directorio **/usr/share/audit/sample-rules** y utilice el comando **augenrules --load**, por ejemplo:

```
# cd /usr/share/audit/sample-rules/
# cp 10-base-config.rules 30-stig.rules 31-privileged.rules 99-finalize.rules /etc/audit/rules.d/
# augenrules --load
```

Recursos adicionales

- Puede ordenar las reglas de auditoría utilizando un esquema de numeración. Consulte el archivo **/usr/share/audit/sample-rules/README-rules** para obtener más información.
- Consulte la página de manual **audit.rules(7)** para obtener más información, solucionar problemas y obtener ejemplos adicionales de uso.

10.9. USO DE AUGENRULES PARA DEFINIR REGLAS PERSISTENTES

El script **augenrules** lee las reglas ubicadas en el directorio **/etc/audit/rules.d/** y las compila en un archivo **audit.rules**. Este script procesa todos los archivos que terminan en **.rules** en un orden específico basado en su orden natural de clasificación. Los archivos de este directorio están organizados en grupos con los siguientes significados:

- 10 - Configuración del kernel y auditctl
- 20 - Reglas que podrían coincidir con las reglas generales pero que usted quiere que coincidan de otra manera
- 30 - Normas principales
- 40 - Normas opcionales
- 50 - Reglas específicas del servidor
- 70 - Normas locales del sistema

- 90 - Finalizar (inmutable)

Las normas no están pensadas para ser utilizadas todas a la vez. Son piezas de una política que deben ser pensadas y copiadas individualmente en **/etc/audit/rules.d/**. Por ejemplo, para establecer un sistema en la configuración STIG, copie las reglas **10-base-config**, **30-stig**, **31-privileged**, y **99-finalize**.

Una vez que tenga las reglas en el directorio **/etc/audit/rules.d/**, cárguelas ejecutando el script **augenrules** con la directiva **--load**:

```
# augenrules --load
/sbin/augenrules: No change
No rules
enabled 1
failure 1
pid 742
rate_limit 0
...
```

Recursos adicionales

- Para más información sobre las reglas de auditoría y el script **augenrules**, consulte las páginas de manual **audit.rules(8)** y **augenrules(8)**.

10.10. DESACTIVACIÓN DE AUGENRULES

Siga los siguientes pasos para desactivar la utilidad **augenrules**. Esto cambia la Auditoría para utilizar las reglas definidas en el archivo **/etc/audit/audit.rules**.

Procedimiento

1. Copie el archivo **/usr/lib/systemd/system/auditd.service** en el directorio **/etc/systemd/system/**:

```
# cp -f /usr/lib/systemd/system/auditd.service /etc/systemd/system/
```

2. Edite el archivo **/etc/systemd/system/auditd.service** en un editor de texto de su elección, por ejemplo:

```
# vi /etc/systemd/system/auditd.service
```

3. Comente la línea que contiene **augenrules**, y descomente la línea que contiene el comando **auditctl -R**:

```
#ExecStartPost=-/sbin/augenrules --load
ExecStartPost=-/sbin/auditctl -R /etc/audit/audit.rules
```

4. Recarga el demonio **systemd** para obtener los cambios en el archivo **auditd.service**:

```
# systemctl daemon-reload
```

5. Reinicie el servicio **auditd**:

```
# service auditd restart
```

-

Recursos adicionales

- Las páginas de manual **augenrules(8)** y **audit.rules(8)**.
- El reinicio del servicio Auditd anula los cambios realizados en el artículo /etc/audit/audit.rules.

10.11. INFORMACIÓN RELACIONADA

Para más información sobre el sistema de auditoría, consulte las siguientes fuentes.

Fuentes en línea

- El artículo de la base de conocimientos de referencia del sistema de auditoría de RHEL .
- Las opciones de ejecución de Auditd en un contenedor Artículo de la base de conocimientos.
- La página del Proyecto de Documentación de Auditoría de Linux .

Documentación instalada

La documentación proporcionada por el paquete **audit** se encuentra en el directorio **/usr/share/doc/audit/**.

Páginas del manual

- **audispd.conf(5)**
- **auditd.conf(5)**
- **ausearch-expression(5)**
- **audit.rules(7)**
- **audispd(8)**
- **auditctl(8)**
- **auditd(8)**
- **aulast(8)**
- **aulastlog(8)**
- **aureport(8)**
- **ausearch(8)**
- **ausyscall(8)**
- **autrace(8)**
- **auvirt(8)**

CAPÍTULO 11. BLOQUEO Y AUTORIZACIÓN DE APLICACIONES MEDIANTE FAPOLICYD

El establecimiento y la aplicación de una política que permita o deniegue la ejecución de aplicaciones basándose en un conjunto de reglas previene eficazmente la ejecución de software desconocido y potencialmente malicioso.

11.1. INTRODUCCIÓN A LA FAPOLICÍA

El marco de software **fapolicyd** controla la ejecución de las aplicaciones en función de una política definida por el usuario. Esta es una de las formas más eficaces de evitar la ejecución de aplicaciones no fiables y posiblemente maliciosas en el sistema.

El marco **fapolicyd** proporciona los siguientes componentes:

- **fapolicyd** servicio
- **fapolicyd** utilidades de línea de comandos
- **fapolicyd** Plugin YUM
- **fapolicyd** lenguaje de la regla

El administrador puede definir las reglas de ejecución de **allow** y **deny** para cualquier aplicación con la posibilidad de auditar en base a una ruta, hash, tipo MIME o confianza.

El marco **fapolicyd** introduce el concepto de confianza. Una aplicación es de confianza cuando está correctamente instalada por el gestor de paquetes del sistema y, por tanto, está registrada en la base de datos RPM del sistema. El demonio **fapolicyd** utiliza la base de datos RPM como una lista de binarios y scripts de confianza. El plugin **fapolicyd** YUM registra cualquier actualización del sistema que sea manejada por el gestor de paquetes YUM. El plugin notifica al demonio **fapolicyd** sobre los cambios en esta base de datos.

Una instalación mediante la utilidad **rpm** requiere una actualización manual de la base de datos, y otras formas de añadir aplicaciones requieren la creación de reglas personalizadas y el reinicio del servicio **fapolicyd**.

La configuración del servicio **fapolicyd** se encuentra en el directorio **/etc/fapolicyd/** con la siguiente estructura:

- El archivo **fapolicyd.rules** contiene las reglas de ejecución **allow** y **deny**.
- El archivo **fapolicyd.conf** contiene las opciones de configuración del demonio. Este archivo es útil principalmente para ajustar el rendimiento.

Recursos adicionales

- Consulte las páginas de manual **fapolicyd(8)**, **fapolicyd.rules(5)**, y **fapolicyd.conf(5)** para obtener más información.

11.2. DESPLIEGUE DE FAPOLICYD

Para desplegar el marco **fapolicyd** en RHEL:

Procedimiento

1. Instale el paquete **fapolicyd**:

```
# yum install fapolicyd
```

2. Habilite e inicie el servicio **fapolicyd**:

```
# systemctl enable --now fapolicyd
```

Pasos de verificación

1. Compruebe que el servicio **fapolicyd** está funcionando correctamente:

```
# systemctl status fapolicyd
● fapolicyd.service - File Access Policy Daemon
   Loaded: loaded (/usr/lib/systemd/system/fapolicyd.service; enabled; vendor p>
   Active: active (running) since Tue 2019-10-15 18:02:35 CEST; 55s ago
  Process: 8818 ExecStart=/usr/sbin/fapolicyd (code=exited, status=0/SUCCESS)
 Main PID: 8819 (fapolicyd)
    Tasks: 4 (limit: 11500)
   Memory: 78.2M
   CGroup: /system.slice/fapolicyd.service
           └─8819 /usr/sbin/fapolicyd

Oct 15 18:02:35 localhost.localdomain systemd[1]: Starting File Access Policy D>
Oct 15 18:02:35 localhost.localdomain fapolicyd[8819]: Initialization of the da>
Oct 15 18:02:35 localhost.localdomain fapolicyd[8819]: Reading RPMDB into memory
Oct 15 18:02:35 localhost.localdomain systemd[1]: Started File Access Policy Da>
Oct 15 18:02:36 localhost.localdomain fapolicyd[8819]: Creating database
```

2. Inicie sesión como un usuario sin privilegios de root y compruebe que **fapolicyd** funciona, por ejemplo:

```
$ cp /bin/ls /tmp
$ /tmp/ls
bash: /tmp/ls: Operation not permitted
```

11.3. MARCAR LOS ARCHIVOS COMO DE CONFIANZA UTILIZANDO UNA FUENTE DE CONFIANZA ADICIONAL

Puede utilizar este procedimiento para utilizar una fuente de confianza adicional para **fapolicyd**. Antes de RHEL 8.3, **fapolicyd** sólo confiaba en los archivos contenidos en la base de datos RPM. El marco **fapolicyd** ahora soporta también el uso del archivo de texto plano **/etc/fapolicyd/fapolicyd.trust** como fuente de confianza. Puede modificar **fapolicyd.trust** directamente con un editor de texto o a través de los comandos CLI de **fapolicyd**.

NOTA

Prefiero marcar los archivos como de confianza usando **fapolicyd.trust** en lugar de escribir reglas personalizadas de **fapolicyd**.

Requisitos previos

- El marco **fapolicyd** se despliega en su sistema.

Procedimiento

1. Copie su binario personalizado en el directorio requerido, por ejemplo:

```
$ cp /bin/ls /tmp
$ /tmp/ls
bash: /tmp/ls: Operation not permitted
```

2. Marca tu binario personalizado como de confianza:

```
# fapolicyd-cli --file add /tmp/ls
```

   Tenga en cuenta que el comando anterior añade la línea correspondiente a **/etc/fapolicyd/fapolicyd.trust**.

3. Reiniciar **fapolicyd**:

```
# systemctl restart fapolicyd
```

Pasos de verificación

1. Comprueba que tu binario personalizado puede ejecutarse ahora, por ejemplo:

```
$ /tmp/ls
ls
```

Recursos adicionales

- Consulte la página de manual **fapolicyd.trust(5)** para obtener más información.

11.4. AÑADIR REGLAS PERSONALIZADAS DE PERMISO Y DENEGACIÓN PARA FAPOLICYD

El conjunto de reglas por defecto del paquete **fapolicyd** no afecta a las funciones del sistema. Para escenarios personalizados, como el almacenamiento de binarios y scripts en un directorio no estándar o la adición de aplicaciones sin los instaladores **yum** o **rpm**, debe modificar las reglas existentes o añadir otras nuevas. Los siguientes pasos demuestran la adición de una nueva regla para permitir un binario personalizado.

Requisitos previos

- El marco **fapolicyd** se despliega en su sistema.

Procedimiento

1. Copie su binario personalizado en el directorio requerido, por ejemplo:

```
$ cp /bin/ls /tmp
$ /tmp/ls
bash: /tmp/ls: Operation not permitted
```

2. Detenga el servicio **fapolicyd**:

```
# systemctl stop fapolicyd
```

3. Utilice el modo de depuración para identificar la regla correspondiente. Como la salida del comando **fapolicyd --debug** es verbosa y sólo puede detenerla pulsando **Ctrl**+**C** o matando el proceso correspondiente, redirija la salida de error a un archivo:

```
# fapolicyd --debug 2> fapolicy.output &
[1] 51341
```

Como alternativa, puede ejecutar el modo de depuración **fapolicyd** en otro terminal.

4. Repite el comando que no fue permitido:

```
$ /tmp/ls
bash: /tmp/ls: Operation not permitted
```

5. Detener el modo de depuración reanudándolo en primer plano y pulsando **Ctrl**+**C**:

```
# fg
fapolicyd --debug
^Cshutting down...
Inter-thread max queue depth 1
Allowed accesses: 2
Denied accesses: 1
[...]
```

Alternativamente, matar el proceso de **fapolicyd** modo de depuración:

```
# kill 51341
```

6. Encuentre una regla que deniegue la ejecución de su aplicación:

```
# cat fapolicy.output
[...]
rule:9 dec=deny_audit perm=execute auid=1000 pid=51362 exe=/usr/bin/bash : file=/tmp/ls
ftype=application/x-executable
[...]
```

7. Agregue una nueva regla **allow** *before* la regla que negó la ejecución de su binario personalizado en el archivo **/etc/fapolicyd/fapolicyd.rules**. La salida del comando anterior indicó que la regla es la regla número *9* en este ejemplo:

```
allow perm=execute exe=/usr/bin/bash trust=1 : path=/tmp/ls ftype=application/x-executable
trust=0
```

Como alternativa, puede permitir la ejecución de todos los binarios en el directorio **/tmp** añadiendo la siguiente regla en el archivo **/etc/fapolicyd/fapolicyd.rules**:

```
allow perm=execute exe=/usr/bin/bash trust=1 : dir=/tmp/ all trust=0
```

8. Para evitar cambios en el contenido de su binario personalizado, defina la regla requerida utilizando una suma de comprobación SHA-256:

```
$ sha256sum /tmp/ls
780b75c90b2d41ea41679fcb358c892b1251b68d1927c80fbc0d9d148b25e836  ls
```

Cambia la regla por la siguiente definición:

```
allow perm=execute exe=/usr/bin/bash trust=1 :
sha256hash=780b75c90b2d41ea41679fcb358c892b1251b68d1927c80fbc0d9d148b25e836
```

9. Inicie el servicio **fapolicyd**:

```
# systemctl start fapolicyd
```

Pasos de verificación

1. Comprueba que tu binario personalizado puede ejecutarse ahora, por ejemplo:

```
$ /tmp/ls
ls
```

Recursos adicionales

- Consulte la página de manual **fapolicyd.trust(5)** para obtener más información.

11.5. SOLUCIÓN DE PROBLEMAS RELACIONADOS CON FAPOLICYD

En la siguiente sección se ofrecen consejos para la resolución de problemas básicos del marco de trabajo de las aplicaciones de **fapolicyd** y una guía para añadir aplicaciones mediante el comando **rpm**.

Instalación de aplicaciones con rpm

- Si se instala una aplicación mediante el comando **rpm**, hay que realizar una actualización manual de la base de datos RPM de **fapolicyd**:

  1. Instale su *application*:

```
# rpm -i application.rpm
```

  2. Actualiza la base de datos:

```
# fapolicyd-cli --update
```

  Si se omite este paso, el sistema puede congelarse y debe reiniciarse.

Estado del servicio

- Si **fapolicyd** no funciona correctamente, compruebe el estado del servicio:

```
# systemctl status fapolicyd
```

Modo de depuración

- El modo de depuración proporciona información detallada sobre las reglas coincidentes, el estado de la base de datos, etc. Para cambiar **fapolicyd** al modo de depuración:

  1. Detenga el servicio **fapolicyd**:

     ```
     # systemctl stop fapolicyd
     ```

  2. Utilice el modo de depuración para identificar la regla correspondiente:

     ```
     # fapolicyd --debug
     ```

     Dado que la salida del comando **fapolicyd --debug** es verbosa, puede redirigir la salida de errores a un archivo:

     ```
     # fapolicyd --debug 2> fapolicy.output
     ```

Eliminación de la base de datos**fapolicyd**

- Para resolver los problemas relacionados con la base de datos **fapolicyd**, intente eliminar el archivo de la base de datos:

  ```
  # systemctl stop fapolicyd
  # fapolicyd-cli --delete-db
  ```

AVISO

No elimine el directorio **/var/lib/fapolicyd/**. El marco **fapolicyd** restaura automáticamente sólo el archivo de la base de datos en este directorio.

Volcado de la base de datos**fapolicyd**

- El **fapolicyd** contiene entradas de todas las fuentes de confianza habilitadas. Puede comprobar las entradas después de volcar la base de datos:

  ```
  # fapolicyd-cli dump-db
  ```

Tubo de aplicación

- En raras ocasiones, la eliminación del archivo de tuberías **fapolicyd** puede resolver un bloqueo:

  ```
  # rm -f /var/run/fapolicyd/fapolicyd.fifo
  ```

Recursos adicionales

- Consulte la página de manual **fapolicyd-cli(1)** para obtener más información.

11.6. RECURSOS ADICIONALES

- Para obtener más información, consulte las páginas de manual relacionadas con **fapolicyd** mediante el comando **man -k fapolicyd**.
- La presentación del FOSDEM 2020 fapolicyd ofrece varios ejemplos de cómo añadir reglas personalizadas **fapolicyd**.

CAPÍTULO 12. PROTECCIÓN DE LOS SISTEMAS CONTRA LOS DISPOSITIVOS USB INTRUSIVOS

Los dispositivos USB pueden estar cargados con spyware, malware o troyanos, que pueden robar sus datos o dañar su sistema. Como administrador de Red Hat Enterprise Linux, puede prevenir tales ataques USB con **USBGuard**.

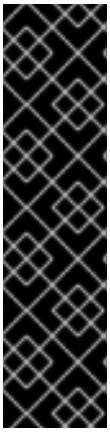
12.1. USBGUARD

Con el marco de software USBGuard, puede proteger sus sistemas contra dispositivos USB intrusivos utilizando listas básicas de dispositivos permitidos y prohibidos basadas en la función de autorización de dispositivos USB en el kernel.

El marco de trabajo de USBGuard proporciona los siguientes componentes:

- El componente de servicio del sistema con una interfaz de comunicación entre procesos (IPC) para la interacción dinámica y la aplicación de políticas
- La interfaz de línea de comandos para interactuar con un servicio del sistema **usbguard** en ejecución
- El lenguaje de reglas para escribir políticas de autorización de dispositivos USB
- La API de C para interactuar con el componente de servicio del sistema implementado en una biblioteca compartida

El archivo de configuración del servicio del sistema **usbguard** (**/etc/usbguard/usbguard-daemon.conf**) incluye las opciones para autorizar a los usuarios y grupos a utilizar la interfaz IPC.



IMPORTANTE

El servicio del sistema proporciona la interfaz IPC pública de USBGuard. En Red Hat Enterprise Linux, el acceso a esta interfaz está limitado por defecto sólo al usuario root.

Considere la posibilidad de configurar la opción **IPCAccessControlFiles** (recomendada) o las opciones **IPCAllowedUsers** y **IPCAllowedGroups** para limitar el acceso a la interfaz IPC.

Asegúrese de no dejar la lista de control de acceso (ACL) sin configurar, ya que esto expone la interfaz IPC a todos los usuarios locales y les permite manipular el estado de autorización de los dispositivos USB y modificar la política de USBGuard.

12.2. INSTALACIÓN DE USBGUARD

Utilice este procedimiento para instalar e iniciar el marco **USBGuard**.

Procedimiento

1. Instale el paquete **usbguard**:

```
# yum install usbguard
```

2. Crear un conjunto de reglas iniciales:

```
# usbguard generate-policy > /etc/usbguard/rules.conf
```

3. Inicie el demonio **usbguard** y asegúrese de que se inicie automáticamente al arrancar:

```
# systemctl enable --now usbguard
```

Pasos de verificación

1. Compruebe que el servicio **usbguard** está en funcionamiento:

```
# systemctl status usbguard
● usbguard.service - USBGuard daemon
   Loaded: loaded (/usr/lib/systemd/system/usbguard.service; enabled; vendor preset:
disabled)
   Active: active (running) since Thu 2019-11-07 09:44:07 CET; 3min 16s ago
     Docs: man:usbguard-daemon(8)
 Main PID: 6122 (usbguard-daemon)
    Tasks: 3 (limit: 11493)
   Memory: 1.2M
   CGroup: /system.slice/usbguard.service
           └─6122 /usr/sbin/usbguard-daemon -f -s -c /etc/usbguard/usbguard-daemon.conf

Nov 07 09:44:06 localhost.localdomain systemd[1]: Starting USBGuard daemon...
Nov 07 09:44:07 localhost.localdomain systemd[1]: Started USBGuard daemon.
```

2. Lista de dispositivos USB reconocidos por **USBGuard**:

```
# usbguard list-devices
4: allow id 1d6b:0002 serial "0000:02:00.0" name "xHCI Host Controller" hash...
```

Recursos adicionales

- **usbguard(1)** y **usbguard-daemon.conf(5)** páginas man

12.3. BLOQUEO Y AUTORIZACIÓN DE UN DISPOSITIVO USB MEDIANTE CLI

Este procedimiento describe cómo autorizar y bloquear un dispositivo USB utilizando el comando **usbguard**.

Requisitos previos

- El servicio **usbguard** está instalado y funcionando.

Procedimiento

1. Lista de dispositivos USB reconocidos por **USBGuard**:

```
# usbguard list-devices
1: allow id 1d6b:0002 serial "0000:00:06.7" name "EHCI Host Controller" hash
"JDOb0BiktYs2ct3mSQKopnOOV2h9MGYADwhT+oUtF2s=" parent-hash
"4PHGcaDKWtPjKDwYpIRG722cB9SlGz9l9lea93+Gt9c=" via-port "usb1" with-interface
09:00:00
```

```
...
6: block id 1b1c:1ab1 serial "000024937962" name "Voyager" hash
"CrXgiaWIf2bZAU+5WkzOE7y0rdSO82XMzubn7HDb95Q=" parent-hash
"JDOb0BiktYs2ct3mSQKopnOOV2h9MGYADwhT+oUtF2s=" via-port "1-3" with-interface
08:06:50
```

2. Autoriza al dispositivo *6* a interactuar con el sistema:

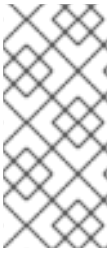
```
# usbguard allow-device 6
```

3. Desautorizar y eliminar el dispositivo *6*:

```
# usbguard reject-device 6
```

4. Desautorizar y retener el dispositivo *6*:

```
# usbguard block-device 6
```



NOTA

USBGuard utiliza los términos *block* y *reject* con los siguientes significados:

- *block*: no interactúe con este dispositivo por ahora.
- *reject*: ignora este dispositivo como si no existiera.

Recursos adicionales

- Lista todas las opciones del comando **usbguard**:

```
$ usbguard --help
```

- **usbguard(1)** página de manual

12.4. BLOQUEAR Y AUTORIZAR PERMANENTEMENTE UN DISPOSITIVO USB

Puede bloquear y autorizar permanentemente un dispositivo USB utilizando la opción **-p**. Esto añade una regla específica del dispositivo a la política actual.

Requisitos previos

- El servicio **usbguard** está instalado y funcionando.

Procedimiento

1. Configure SELinux para permitir que el demonio **usbguard** escriba reglas.
   a. Muestra los booleanos de **semanage** relevantes para **usbguard**.

```
# semanage boolean -l | grep usbguard
usbguard_daemon_write_conf     (off  ,  off)  Allow usbguard to daemon write conf
usbguard_daemon_write_rules    (on   ,   on)  Allow usbguard to daemon write rules
```

b. Opcional: Si el booleano **usbguard_daemon_write_rules** está desactivado, actívelo.

```
# semanage boolean -m --on usbguard_daemon_write_rules
```

2. Lista de dispositivos USB reconocidos por **USBGuard**:

```
# usbguard list-devices
1: allow id 1d6b:0002 serial "0000:00:06.7" name "EHCI Host Controller" hash
"JDOb0BiktYs2ct3mSQKopnOOV2h9MGYADwhT+oUtF2s=" parent-hash
"4PHGcaDKWtPjKDwYpIRG722cB9SlGz9l9lea93+Gt9c=" via-port "usb1" with-interface
09:00:00
...
6: block id 1b1c:1ab1 serial "000024937962" name "Voyager" hash
"CrXgiaWIf2bZAU+5WkzOE7y0rdSO82XMzubn7HDb95Q=" parent-hash
"JDOb0BiktYs2ct3mSQKopnOOV2h9MGYADwhT+oUtF2s=" via-port "1-3" with-interface
08:06:50
```

3. Autorizar permanentemente el dispositivo *6* para interactuar con el sistema:

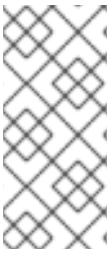
```
# usbguard allow-device 6 -p
```

4. Desautorizar permanentemente y eliminar el dispositivo *6*:

```
# usbguard reject-device 6 -p
```

5. Desautorizar permanentemente y retener el dispositivo *6*:

```
# usbguard block-device 6 -p
```



NOTA

USBGuard utiliza los términos *block* y *reject* con el siguiente significado:

- *block*: no interactúe con este dispositivo por ahora.
- *reject*: ignora este dispositivo como si no existiera.

Verificación

1. Compruebe que las reglas de **USBGuard** incluyen los cambios realizados.

```
# usbguard list-rules
```

Recursos adicionales

- Lista todas las opciones del comando **usbguard**:

```
$ usbguard --help
```

- **usbguard(1)** página de manual

12.5. CREACIÓN DE UNA POLÍTICA PERSONALIZADA PARA DISPOSITIVOS USB

El siguiente procedimiento contiene los pasos para crear un conjunto de reglas para dispositivos USB que refleje los requisitos de su escenario.

Requisitos previos

- El servicio **usbguard** está instalado y funcionando.
- El archivo **/etc/usbguard/rules.conf** contiene un conjunto de reglas inicial generado por el comando **usbguard generate-policy**.

Procedimiento

1. Cree una política que autorice los dispositivos USB actualmente conectados y almacene las reglas generadas en el archivo **rules.conf**:

```
# usbguard generate-policy --no-hashes > ./rules.conf
```

   La opción **--no-hashes** no genera atributos hash para los dispositivos. Evita los atributos hash en tus ajustes de configuración porque podrían no ser persistentes.

2. Edite el archivo **rules.conf** con un editor de texto de su elección, por ejemplo:

```
# vi ./rules.conf
```

3. Añada, elimine o edite las reglas según sea necesario. Por ejemplo, la siguiente regla sólo permite que los dispositivos con una única interfaz de almacenamiento masivo interactúen con el sistema:

```
allow with-interface equals { 08:*:* }
```

   Consulte la página de manual **usbguard-rules.conf(5)** para obtener una descripción detallada del lenguaje de reglas y más ejemplos.

4. Instale la política actualizada:

```
# install -m 0600 -o root -g root rules.conf /etc/usbguard/rules.conf
```

5. Reinicie el demonio **usbguard** para aplicar los cambios:

```
# systemctl restart usbguard
```

Pasos de verificación

1. Compruebe que sus reglas personalizadas están en la política activa, por ejemplo:

```
# usbguard list-rules
...
4: allow with-interface 08:*:*
```

```
...
```

Recursos adicionales

- **usbguard-rules.conf(5)** página de manual

12.6. CREACIÓN DE UNA POLÍTICA PERSONALIZADA ESTRUCTURADA PARA LOS DISPOSITIVOS USB

Puede organizar su política personalizada **USBGuard** en varios archivos **.conf** dentro del directorio **/etc/usbguard/rules.d/**. El **usbguard-daemon** combina entonces el archivo principal **rules.conf** con los archivos **.conf** dentro del directorio en orden alfabético.

Requisitos previos

- El servicio **usbguard** está instalado y funcionando.

Procedimiento

1. Cree una política que autorice los dispositivos USB actualmente conectados, y almacene las reglas generadas en un nuevo archivo **.conf**, por ejemplo, ***policy.conf***.

```
# usbguard generate-policy --no-hashes > ./policy.conf
```

   La opción **--no-hashes** no genera atributos hash para los dispositivos. Evita los atributos hash en tus ajustes de configuración porque podrían no ser persistentes.

2. Visualice el ***policy.conf*** archivo con un editor de texto de su elección, por ejemplo:

```
# vi ./policy.conf
...
allow id 04f2:0833 serial "" name "USB Keyboard" via-port "7-2" with-interface { 03:01:01
03:00:00 } with-connect-type "unknown"
...
```

3. Mover las líneas seleccionadas a un archivo separado **.conf**.

NOTA

Los dos dígitos al principio del nombre del archivo especifican el orden en que el demonio lee los archivos de configuración.

   Por ejemplo, copie las reglas de sus teclados en un nuevo archivo **.conf**.

```
# grep "USB Keyboard" ./policy.conf > ./10keyboards.conf
```

4. Instale la nueva política en el directorio **/etc/usbguard/rules.d/**.

```
# install -m 0600 -o root -g root 10keyboards.conf /etc/usbguard/rules.d/10keyboards.conf
```

5. Mueve el resto de las líneas a un archivo principal **rules.conf**.

```
# grep -v "USB Keyboard" ./policy.conf > ./rules.conf
```

6. Instale las reglas restantes.

```
# install -m 0600 -o root -g root rules.conf /etc/usbguard/rules.conf
```

7. Reinicie el demonio **usbguard** para aplicar los cambios.

```
# systemctl restart usbguard
```

Pasos de verificación

1. Muestra todas las reglas activas de **USBGuard**.

```
# usbguard list-rules
...
15: allow id 04f2:0833 serial "" name "USB Keyboard" hash
"kxM/iddRe/WSCocgiuQlVs6Dn0VEza7KiHoDeTz0fyg=" parent-hash
"2i6ZBJfTl5BakXF7Gba84/Cp1gslnNc1DM6vWQpie3s=" via-port "7-2" with-interface {
03:01:01 03:00:00 } with-connect-type "unknown"
...
```

2. Muestra el contenido del archivo **rules.conf** y todos los archivos **.conf** en el directorio **/etc/usbguard/rules.d/**.

```
# cat /etc/usbguard/rules.conf /etc/usbguard/rules.d/*.conf
```

3. Compruebe que las reglas activas contienen todas las reglas de los archivos y están en el orden correcto.

Recursos adicionales

- **usbguard-rules.conf(5)** página de manual

12.7. AUTORIZACIÓN DE USUARIOS Y GRUPOS PARA UTILIZAR LA INTERFAZ USBGUARD IPC

Utilice este procedimiento para autorizar a un usuario específico o a un grupo a utilizar la interfaz IPC pública de USBGuard. Por defecto, sólo el usuario root puede utilizar esta interfaz.

Requisitos previos

- El servicio **usbguard** está instalado y funcionando.
- El archivo **/etc/usbguard/rules.conf** contiene un conjunto de reglas inicial generado por el comando **usbguard generate-policy**.

Procedimiento

1. Edite el archivo **/etc/usbguard/usbguard-daemon.conf** con un editor de texto de su elección:

```
# vi /etc/usbguard/usbguard-daemon.conf
```

2. Por ejemplo, añada una línea con una regla que permita a todos los usuarios del grupo **wheel** utilizar la interfaz IPC y guarde el archivo:

```
IPCAllowGroups=rueda
```

3. También puedes añadir usuarios o grupos con el comando **usbguard**. Por ejemplo, el siguiente comando permite que el usuario *joesec* tenga acceso completo a las secciones **Devices** y **Exceptions**. Además, *joesec* puede listar la política actual y escuchar las señales de la política.

```
# usbguard add-user joesec --devices ALL --policy list,listen --exceptions ALL
```

Para eliminar los permisos concedidos al usuario *joesec*, utilice el comando **usbguard remove-user joesec**.

4. Reinicie el demonio **usbguard** para aplicar los cambios:

```
# systemctl restart usbguard
```

Recursos adicionales

- **usbguard(1)** y **usbguard-rules.conf(5)** páginas man

12.8. REGISTRO DE EVENTOS DE AUTORIZACIÓN DE USBGUARD EN EL REGISTRO DE AUDITORÍA DE LINUX

Siga los siguientes pasos para integrar el registro de eventos de autorización de USBguard al registro de auditoría estándar de Linux. Por defecto, el demonio **usbguard** registra los eventos en el archivo **/var/log/usbguard/usbguard-audit.log**.

Requisitos previos

- El servicio **usbguard** está instalado y funcionando.
- El servicio **auditd** está funcionando.

Procedimiento

1. Edite el archivo **usbguard-daemon.conf** con un editor de texto de su elección:

```
# vi /etc/usbguard/usbguard-daemon.conf
```

2. Cambie la opción **AuditBackend** de **FileAudit** a **LinuxAudit**:

```
AuditBackend=LinuxAudit
```

3. Reinicie el demonio **usbguard** para aplicar el cambio de configuración:

```
# systemctl restart usbguard
```

Pasos de verificación

1. Consultar el registro del demonio **audit** para un evento de autorización USB, por ejemplo:

```
# ausearch -ts recent -m USER_DEVICE
```

Recursos adicionales

- **usbguard-daemon.conf(5)** página de manual

12.9. RECURSOS ADICIONALES

- **usbguard(1)**, **usbguard-rules.conf(5)**, **usbguard-daemon(8)**, y **usbguard-daemon.conf(5)** páginas de manual
- Página de USBGuard