centro criptológico nacional

Guía de Seguridad de las TIC
CCN-STIC 667

CONTENERIZACIÓN EN ARQUITECTURAS VIRTUALES

MARZO 2021

MINISTERIO DE DEFENSA

Edita:

2.5.4.13=Qualified Certificate: AAPP-SEP-M-SW-KPSC, ou=sello electrónico, serialNumber=S2800155J, o=CENTRO CRIPTOLOGICO NACIONAL, c=ES

NIPO: 083-21-068-2

Fecha de Edición: marzo de 2021
Sidertia Solutions S.L. ha participado en la realización y modificación del presente documento y sus anexos.

PRÓLOGO

En un mundo cada vez más complejo y globalizado, en el que las tecnologías de la información y la comunicación (TIC) desempeñan un papel de suma importancia, hemos de ser conscientes de que la gestión adecuada de la ciberseguridad constituye un reto colectivo al que necesariamente hemos de enfrentar. Resulta necesario garantizar la protección de la capacidad económica, tecnológica y política de nuestro país, máxime cuando la proliferación de ataques dirigidos y el robo de información sensible representan una realidad incontestable.

Por ello, resulta imprescindible estar al día de las amenazas y vulnerabilidades asociadas al uso de las nuevas tecnologías. El conocimiento de los riesgos que se ciernen sobre el ciberespacio ha de servir para implementar con garantías las medidas, tanto procedimentales como técnicas y organizativas, que permitan un entorno seguro y confiable.

La Ley 11/2002, de 6 de mayo, reguladora del Centro Nacional de Inteligencia (CNI), encomienda al Centro Nacional de Inteligencia el ejercicio de las funciones relativas a la seguridad de las tecnologías de la información y de protección de la información clasificada, a la vez que confiere a su Secretario de Estado Director la responsabilidad de dirigir el Centro Criptológico Nacional (CCN).

Partiendo del conocimiento y la experiencia del CNI sobre amenazas y vulnerabilidades en materia de riesgos emergentes, el Centro realiza, a través del Centro Criptológico Nacional, regulado por el Real Decreto 421/2004, de 12 de marzo, diversas actividades directamente relacionadas con la seguridad de las TIC, orientadas a la formación de personal experto, al empleo de tecnologías de seguridad adecuadas y a la aplicación de políticas y procedimientos de seguridad.

Precisamente, esta serie de documentos CCN-STIC es un claro reflejo de la labor que este organismo lleva a cabo en materia de implementación de seguridad, permitiendo la aplicación de políticas y procedimientos, pues las guías han sido elaboradas con un claro objetivo: mejorar el grado de ciberseguridad de las organizaciones, conscientes de la importancia que tiene el establecimiento de un marco de referencia en esta materia que sirva de apoyo para que el personal de la Administración lleve a cabo la difícil tarea de proporcionar seguridad a los sistemas de las TIC bajo su responsabilidad.

Con esta serie de documentos, el Centro Criptológico Nacional, en cumplimiento de sus cometidos y de lo reflejado en el Real Decreto 3/2010 por el que se regula el Esquema Nacional en el ámbito de la Administración electrónica, contribuye a mejorar la ciberseguridad española y mantener las infraestructuras y los sistemas de información de todas las administraciones públicas con unos niveles óptimos de seguridad. Todo ello, con el fin de generar confianza y garantías en el uso de estas tecnologías, protegiendo la confidencialidad de los datos y garantizando su autenticidad, integridad y disponibilidad.

marzo de 2021

Paz Esteban López
Secretaria de Estado
Directora del Centro Criptológico Nacional

ÍNDICE

1. SOBRE CCN-CERT, CERT GUBERNAMENTAL NACIONAL

El CCN-CERT es la Capacidad de Respuesta a incidentes de Seguridad de la Información del Centro Criptológico Nacional, CCN, adscrito al Centro Nacional de Inteligencia, CNI. Este servicio se creó en el año 2006 como CERT Gubernamental Nacional español y sus funciones quedan recogidas en la Ley 11/2002 reguladora del CNI, el RD 421/2004 de regulación del CCN y en el RD 3/2010, de 8 de enero, regulador del Esquema Nacional de Seguridad (ENS), modificado por el RD 951/2015 de 23 de octubre.

Su misión, por tanto, es contribuir a la mejora de la ciberseguridad española, siendo el centro de alerta y respuesta nacional que coopere y ayude a responder de forma rápida y eficiente a los ciberataques y a afrontar de forma activa las ciberamenazas, incluyendo la coordinación a nivel público estatal de las distintas Capacidades de Respuesta a Incidentes o Centros de Operaciones de Ciberseguridad existentes.

Todo ello, con el fin último de conseguir un ciberespacio más seguro y confiable, preservando la información clasificada (tal y como recoge el art. 4. F de la Ley 11/2002) y la información sensible, defendiendo el Patrimonio Tecnológico español, formando al personal experto, aplicando políticas y procedimientos de seguridad y empleando y desarrollando las tecnologías más adecuadas a este fin.

De acuerdo a esta normativa y la Ley 40/2015 de Régimen Jurídico del Sector Público es competencia del CCN-CERT la gestión de ciberincidentes que afecten a cualquier organismo o empresa pública. En el caso de operadores críticos del sector público la gestión de ciberincidentes se realizará por el CCN-CERT en coordinación con el CNPIC.

2. INTRODUCCIÓN

El actual uso de tecnologías emergentes que permiten la virtualización de los sistemas de las Tecnologías de la Información y la Comunicación (TIC), en adelante Sistemas, se ha hecho crítica con la extensión del empleo de dichas tecnologías a todos los ámbitos.

Las amenazas asociadas a un sistema, que pueden afectar a la confidencialidad, integridad y disponibilidad de la información manejada, o a la propia integridad y disponibilidad del sistema, son tenidas en cuenta a la hora de establecer los requisitos de seguridad mínimos, de tal manera, que las medidas de protección que se implementan tienen por objeto hacer frente a dichas amenazas reduciendo la superficie de exposición y minimizando el impacto de las mismas.

La seguridad se tratará desde la fase de diseño del sistema, donde se definirán las salvaguardas a implementar, permitiendo de esta manera que el análisis y gestión de riesgos de seguridad estén presentes en el proceso de desarrollo del sistema.

La Política STIC determina que todos los sistemas que requieran manejar información clasificada deberán ser previamente acreditados, de acuerdo con el procedimiento que, para tal fin, apruebe la Autoridad de Acreditación de Seguridad (AAS).

El procedimiento de acreditación de seguridad, entre otros requisitos, confirmará que las medidas de seguridad que satisfacen los requisitos STIC se encuentran implementadas en los Sistemas antes de manejar información clasificada.

3. OBJETO

El presente documento contiene una guía para la configuración segura de tecnologías de contenerización tomando como referente la tecnología de “Dockers” o “Podman” y como sistema base o anfitrión un sistema operativo Linux que posea configuraciones de seguridad correspondientes al grado de clasificación que se pretende implementar para los contenedores.

La configuración deberá realizarse en máquinas con el sistema operativo recién instalado, con la guía CCN-STIC que corresponda correctamente aplicada, si bien también se deben llevar a cabo periódicamente sobre cualquier máquina para comprobar el estado de seguridad de la misma.

Las recomendaciones de seguridad que se aplican a través de la presente guía se han diseñado para ser lo más restrictiva posible, minimizando la superficie de ataque y, por lo tanto, los riesgos que pudieran existir. En algunos casos y dependiendo de la funcionalidad requerida del cliente, podría ser necesario modificar la configuración, que aquí se recomienda, para permitir que el equipo proporcione los servicios adicionales.

No obstante, se tiene en consideración que los ámbitos de aplicación son muy variados y por lo tanto dependerán de su aplicación, las peculiaridades y funcionalidades de los servicios prestados por las diferentes organizaciones. Por lo tanto, las recomendaciones de seguridad se han generado definiendo unas pautas generales que permitan el cumplimiento de los mínimos establecidos en un entorno clasificado.

4. ALCANCE

La guía se ha elaborado para proporcionar información específica con objeto de asegurar diferentes tecnologías de contenerización sobre un cliente con el sistema operativo Linux, instalado en español en su última versión. Se incluyen, además, operaciones básicas de administración para la aplicación de las mismas, así como una serie de recomendaciones para su uso.

El escenario en el cual está basada la presente guía tiene las siguientes características técnicas:

a) Implementación de seguridad en un escenario de red clasificada, clientes y servidores Linux con tecnologías de contenerización.

Este documento incluye:

b) **Recomendaciones de seguridad para el equipo anfitrión.** Va a permitir implantar configuraciones de seguridad en servidores y clientes Linux que posean tecnologías de contenerización por medio de ejemplos prácticos.

c) **Recomendaciones de seguridad con ejemplos paso a paso.** Va a permitir implantar configuraciones de seguridad para distintas tecnologías de contenerización por medio de ejemplos prácticos.

d) **Recomendaciones de seguridad adicionales.** Completa descripción de aquellas características que, no encontrándose definidos por defecto, agregan seguridad adicional a una infraestructura que posea tecnologías de contenerización.

5. DESCRIPCIÓN DEL USO DE ESTA GUÍA

Para entender esta guía de seguridad, es conveniente explicar el proceso de aplicación de seguridad que describe y los recursos que proporciona. Este proceso constará de los siguientes pasos:

a) Recomendaciones de seguridad sobre un escenario de servidores o clientes Linux con tecnologías de contenerización.

b) Adaptar y aplicar las recomendaciones de la presente guía en función del entorno que requiera su organización.

5.1 AVISOS IMPORTANTES A LOS USUARIOS DE ESTA GUÍA

Los contenidos y recomendaciones de esta guía son de aplicación a equipos que posean implementadas tecnologías basadas en “Docker” o “Podman” sobre Sistemas Operativos Linux en castellano, con el objetivo de reducir la superficie de exposición a ataques posibles con una instalación por defecto, manteniendo los principios de máxima seguridad, mínima exposición y servicios y mínimos privilegios que emanan de la CCN-STIC-301. En el caso de llevar a cabo la aplicación de esta guía sobre el Sistema Operativo con una configuración de idioma diferente al castellano, es posible que deba incorporar nuevos recursos y/o realizar ciertas modificaciones sobre los recursos que se adjuntan con este documento para permitir la correcta aplicación y uso del documento.

En un entorno de red clasificada donde se maneja información clasificada solo se admitirán versiones autorizadas de Sistemas Operativos Linux para la implementación de tecnologías de contenerización basadas en “Docker” o “Podman” con la opción de instalación de menor contenido.

En cuanto al sistema anfitrión de estas tecnologías, destacar que las imágenes MinimalCD contienen un mínimo de paquetes necesarios para una instalación funcional, sin comprometer la seguridad o la usabilidad de la red. Estas imágenes mínimas usan el instalador estándar de Linux con todas sus características regulares menos la selección de paquetes. Yum, Dnf, Zipper, etc. se puede usar después de completar la instalación y agregar o eliminar paquetes.

La guía ha sido desarrollada y probada en distintos entornos de uso de servicios Linux.

La guía de seguridad ha sido elaborada utilizando un laboratorio basado en una plataforma de virtualización tipo Hyper-V sobre Windows Server 2019 Datacenter con las siguientes características técnicas:

a) Servidor Dell PowerEdge™ T320:

   i. Intel Pentium Xeon CPU ES 2430 2.20GHz.

   ii. HDD 1 TB.

   iii. 64 GB de RAM.

   iv. Interfaz de Red 1 Gbit/s.

Esta guía de seguridad no funcionará con hardware que no cumpla con los requisitos de seguridad mínimos de la tecnología. Esto quiere decir que se requiere un kernel Linux versión 3.10 o superior, equipos con más de 2 GB de memoria RAM.

Las tecnologías de contenerización están disponibles en una amplia variedad de plataformas. Docker, por ejemplo, está disponible en distintas distribuciones Linux, MacOS, Windows 10, etc. A continuación, se muestra una tabla comparativa con la compatibilidad de Docker en los distintos sistemas y similar a la mayoría de las tecnologías de contenerización basadas en Docker.

Plataforma	x86_64 / amd64	ARM	ARM64 / AARCH64	IBM Power (ppc64le)	IBM Z (s390x)
Docker Desktop for Mac	☑				
Docker Desktop for Windows	☑				
CentOS	☑		☑		
Debian	☑	☑	☑		
Fedora	☑		☑		
Raspbian		☑	☑		
Ubuntu	☑	☑	☑	☑	☑

Nota: Si bien las instalaciones en sistemas derivados pueden funcionar, tanto "Docker" como "Podman" no prueban ni verifican la instalación en los mismos. Por ejemplo en sistemas como "Lubuntu", "Kubuntu", "Kali Linux", etc.

La guía ha sido desarrollada con el objetivo de dotar a las infraestructuras con la seguridad adecuada por medio de ejemplos de uso reales. Es posible que algunas de las funcionalidades esperadas se recomiende su desactivación y, por lo tanto, pueda ser necesario aplicar acciones adicionales para habilitar servicios, demonios o características deseadas.

Para garantizar la seguridad de los servidores o equipos anfitriones, deberán instalarse las actualizaciones recomendadas por el fabricante, disponibles a través del servicio del propio sistema operativo, así como renovar contenedores con el fin de aplicar a estos las últimas actualizaciones. Las actualizaciones del sistema operativo y de los servicios o componentes mantenidos por el sistema por lo general están disponibles de 24h a 72h dependiendo del mismo. Hay que tener presente que determinadas actualizaciones por su criticidad pueden ser liberadas en cualquier momento. Se deberá tener en cuenta la implementación de las actualizaciones tanto para el sistema operativo anfitrión como para los contenedores. Deberá tener en consideración que hay multitud de sistemas operativos basados en la misma rama de Linux y que sin embargo ofrecen diferentes tiempos de implementación de actualizaciones, por tanto, podrá haber actualizaciones críticas de Docker o Podman que será recomendable descargarse de las fuentes oficiales del proveedor de la solución. En líneas posteriores de la presente guía se tratarán las consideraciones oportunas.

Dependiendo de la naturaleza de estas actualizaciones, el lector podrá encontrarse con algunas diferencias respecto a lo descrito en esta guía. Esto viene motivado por los cambios que, en ocasiones, se realizan para las distintas actualizaciones de seguridad.

Docker, por ejemplo, posee tres tipos de canales de actualización: “stable”, “test” y “nightly”.

a) El canal **stable** ofrece los últimos lanzamientos para disponibilidad general.

   Los lanzamientos año-mes se realizan desde una rama de lanzamiento dividido de la rama maestra. La rama se crea con el formato <año>.<mes> , por ejemplo 18.09. El número indica el mes calendario donde el lanzamiento está generalmente disponible. Todos los lanzamientos de parches adicionales se realizan desde esa rama. Por ejemplo, una vez que se lanza v18.09.0 , todas las versiones de parche posteriores se crean a partir de la rama 18.09.

b) El canal **test** proporciona versiones preliminares que están listas para probar antes de la disponibilidad general (GA).

   En preparación para una nueva versión de año-mes, se crea una rama de la rama maestra con el formato YY.mm cuando los hitos deseados por Docker para el lanzamiento han alcanzado todas las características esperadas. Los lanzamientos previos, como las versiones beta y los candidatos de lanzamiento, se realizan desde sus respectivas ramas de lanzamiento. Las versiones de parche y las versiones preliminares correspondientes se realizan desde la rama de versión correspondiente.

c) El canal **Nightly** ofrece las últimas compilaciones de trabajo en progreso para la próxima versión principal.

   Las compilaciones Nightly ofrecen las últimas compilaciones de trabajo en progreso para la próxima versión principal. Se crean una vez al día desde la rama maestra con el formato de versión:

   i. 0.0.0-YYYYmmddHHMMSS-abcdefabcdef, donde el tiempo de confirmación en UTC y el sufijo final es el prefijo del hash de confirmación, por ejemplo 0.0.0-20180720214833-f61e0f7.

   Estas compilaciones permiten realizar pruebas desde el último código en la rama maestra.

Nota: No se realizan pruebas ni verificaciones que garanticen el correcto funcionamiento de las versiones nightly por parte de Docker.

Antes de aplicar esta guía en producción, deberá asegurarse el hecho de haber probado su configuración y comportamiento en un entorno aislado y controlado, en el cual se habrán aplicado las pruebas y posteriores cambios en la configuración que se ajusten a los criterios específicos de cada organización.

El espíritu de estas guías no está dirigido a remplazar políticas consolidadas y probadas de las organizaciones sino a servir como línea base de seguridad. Esta línea deberá ser adaptada a las necesidades propias de cada organización.

6. SEGURIDAD EN CONTENEDORES

Para asegurar de forma correcta cualquier sistema operativo y los componentes que en él se implementan, es recomendable seguir una serie de pautas de configuración desde el inicio. Por ello, se tendrán en cuenta configuraciones iniciales de instalación tales como “Content trust”, la creación personalizada de contenedores, la creación de usuarios con contraseña compleja entre otros.

Las contraseñas son las llaves del sistema. Deben ser lo más seguras posibles para evitar inicios de sesión no autorizados, que es el primer paso hacia problemas de seguridad mayores. El uso de contraseñas lo suficientemente fuertes como para amortizar un ataque es un paso decisivo y a la vez sencillo que ahorrará muchos problemas en el futuro.

6.1 CONFIGURACIÓN DE CONTRASEÑAS

Muchas contraseñas utilizadas por usuarios son bastante fáciles de adivinar. Linux como sistema anfitrión de los contenedores proporcionan diferentes maneras de proveer autentificación al sistema, incluyendo contraseñas encriptadas con el comando crypt, las contraseñas shadow, Kerberos… Etc. En cualquier situación en la cual se elija una contraseña como parte de un esquema de autentificación, la seguridad de ese esquema estará por lo menos parcialmente a la merced de la complejidad de la contraseña elegida.

a) Una contraseña segura tiene que tener al menos estas características:

b) Tener una longitud mínima de 8 caracteres

c) Mayúsculas y minúsculas alternadas

d) Tantos signos de puntuación y números como sea posible

e) Evitar palabras o frases comunes que puedan figurar en cualquier diccionario

f) No tener relación evidente con datos personales del usuario: Nombre, fecha de nacimiento, etc.

Otro factor a tener en cuenta es la **caducidad de contraseñas**. Dentro de las tareas frecuentes que se realizan en Linux, se encuentra la de administrador de cuentas de usuario, tanto en su creación y edición, como en establecimiento o modificación de la caducidad y el vencimiento de las contraseñas de los usuarios, siendo política de seguridad modificar regularmente la misma.

Para esto, puede ser útil el comando “chage”, el cual es usado para modificar la información de caducidad de la contraseña de un usuario específica, permite ver la información de antigüedad de la cuenta de un usuario o cambiar el número de días entre los cambios de contraseña y la fecha de la última contraseña.

En esta guía se configurará de manera permanente una caducidad de contraseña para nuevos usuarios y modificará la política de seguridad de los usuarios ya existentes para que cumplan estos requisitos de seguridad establecidos. Las recomendaciones de configuración en cuanto a la caducidad de las contraseñas se configurarán en el fichero “/etc/login.defs” y serán las siguientes:

a) El periodo máximo durante el que se puede mantener una contraseña será de 60 días

b) La longitud mínima de la contraseña será de 8 caracteres.

c) El período mínimo durante el que se debe mantener una contraseña será de 15 días.

d) El período durante el que el sistema avisará de una futura caducidad de la contraseña será de 15 días.

6.2 CONFIGURACIÓN INICIAL

Por defecto el sistema operativo base de los contenedores, crea ciertas configuraciones para facilitar el acceso al usuario, habilitando la mayor parte de funcionalidades y aumentando la velocidad de instalación del mismo. Estas configuraciones en muchas ocasiones pueden ser motivo de posibles brechas de seguridad.

Para evitar brechas innecesarias, se configurarán ciertos parámetros de manera correcta:

a) **Creación de imágenes ajustadas.** La base de la seguridad en contenedores se centra en saber qué sistema operativo se está ejecutando, que contenido está incluido en el mismo y que permisos posee el acceso. Del mismo modo hay que revisar, protocolos y servicios que son publicados por el contenedor. Para lograr un mayor control de todo esto, la premisa siempre será la creación de contenedores propios y firmados.

b) **Creación de un usuario distinto a root.** Cuando se habla de root, se refiere a la cuenta superusuario en Linux, aquella que posee todos los privilegios y permisos para realizar acciones sobre el sistema. Para la realización de tareas cotidianas dentro de un contenedor, es necesario configurar usuarios adicionales, que normalmente no vienen configurados ni habilitados, con menor privilegio que sean los encargados de realizar estas tareas.

c) **Contraseña segura para root**. Para ciertas acciones que afectan al sistema de archivos, se requiere tener acceso root. Sin embargo, se debe tener un conocimiento sobre las acciones que se realizan, ya que una acción realizada de manera errónea podría ocasionar daños importantes en el sistema. Para evitar el uso de instrucciones con privilegios de superusuario la cuenta root tiene que estar dotada con una contraseña segura que evite que cualquier usuario malintencionado pueda comprometer de algún modo el sistema.

d) **Usuarios UID 0**. En el fichero /etc/passwd/ existe un campo UID por cada usuario, que corresponde al identificador de cada usuario. Algunas distribuciones de Linux por defecto crean varios usuarios con UID 0 que corresponde al identificador de superusuario. Si existen varios superusuarios en el sistema la probabilidad de vulnerar el mismo es mayor, por este motivo se deben limitar los usuarios con UID 0 únicamente a root, siendo el único usuario habilitado para tener control total sobre el sistema.

e) **Cuentas sin contraseñas**. En Linux existe la opción de configurar una cuenta de usuario sin contraseña, aunque ese usuario no pertenezca a los denominados “sudores” (administradores). En el sistema no debe de haber ningún usuario sin contraseña, esto supondría una vulnerabilidad, ya que cualquier usuario podría acceder a información sensible sin necesidad de estar autorizado para ello.

6.3 CONFIGURACIÓN A NIVEL DE HOST

La seguridad de los contenedores se asemeja a asegurar cualquier proceso en ejecución. Por tanto, se debe tener en consideración la seguridad de todas las capas antes de implementar cualquier solución de contenerización. Otro dato a tener en cuenta es la seguridad en todo el ciclo de vida de la aplicación y por ende del contenedor.

Los contenedores facilitan la tarea de promoción y migración de uno o varios servicios entre diferentes sistemas compatibles. Esta tarea, si bien facilita la versatilidad de los contenedores, también puede provocar problemas de seguridad si no se realiza entre sistemas seguros, por tanto, una de las principales medidas a adoptar para la implementación de un sistema de contenedores pasa por escoger correctamente y bastionar el equipo o equipos que contendrán esa tecnología. Para un sistema Linux, se deben tener en cuenta las siguientes pautas a nivel de host anfitrión.

a) Crear una partición separada para los contenedores
b) Usar un Kernel de Linux actualizado
c) No usar herramientas de desarrollo en producción
d) Realizar un correcto bastionado del sistema adaptándolo a las necesidades del contenedor
e) Borrar todos los servicios no esenciales en el sistema anfitrión
f) Mantener el demonio actualizado (solo en tecnologías basadas en Docker)
g) Permitir solo a los usuarios autorizados controlar el demonio (solo en tecnologías basadas en Docker)
h) Auditar el demonio (si lo hubiera) y los registros con audit:
    i. /var/lib/docker
    ii. /etc/docker
    iii. docker-registry.service
    iv. docker.service
    v. /var/run/docker.sock
    vi. /etc/sysconfig/docker
    vii. /etc/sysconfig/docker-network
    viii. etc/sysconfig/docker-registry
    ix. /etc/sysconfig/docker-storage
    x. /etc/default/docker
    xi. /etc/containers/policy.json
    xii. /usr/lib/tmpfiles.d/podman.conf
    xiii. /usr/share/containers/libpod.conf
    xiv. /etc/containers/registries.json
    xv. /etc/containers/storage.json
    xvi. /etc/containers/
    xvii. /usr/bin/podman

6.4 CONFIGURACIÓN DEL DEMONIO DOCKER

Docker al contrario que Podman posee un demonio activo en el sistema, el demonio Docker al tratarse de un programa que se ejecuta en segundo plano, fuera del control interactivo de los usuarios del sistema, es un punto significativo a tener en cuenta frente a ataques no deseados.

En este sentido, se deberán tener en cuenta ciertos drivers y configuraciones que permitan la ejecución del demonio de forma segura:

a) No usar drivers obsoletos de ejecución de “lxc”.

b) Restringir el tráfico de red entre contenedores.

c) Configurar el nivel de “logging” deseado.

d) No usar registros inseguros (sin TLS).

e) Configurar un registro espejo local.

f) No usar “aufs” como driver de almacenamiento.

g) No iniciar Docker para escuchar a una IP/Port o Unix socket diferente.

h) Configurar autenticación TLS para el demonio de Docker.

i) Configurar el “ulimit” por defecto de forma apropiada.

6.5 CONFIGURACIÓN DE FICHEROS Y PERMISOS DOCKER

En el proceso de diseño de una infraestructura de contenedores, es de suma importancia limitar y aislar recursos. El sistema debe poseer mecanismos que permitan el uso de grupos y roles, adecuando los permisos a las necesidades de cada usuario, limitando por tanto, usuarios como “root”.

Para realizar una buena configuración inicial se deberán revisar ciertos servicios y ficheros para que posean los permisos necesarios y no pertenezcan a usuarios no habilitados para tal fin. Para ello se deberán revisar los siguientes ficheros o servicios:

a) Verificar que el archivo docker.service pertenece al usuario root y grupo root.

b) Verificar que los permisos del archivo docker.service se encuentran configurados al menos como 644.

c) Verificar que el archivo docker-registry.service pertenece al usuario root y grupo root.

d) Verificar que los permisos del archivo docker-registry.service se encuentran configurados al menos como 644.

e) Verificar que el archivo docker.socket pertenece al usuario root y grupo root.

f) Verificar que los permisos del archivo docker.socket se encuentran configurados al menos como 644.

g) Verificar que el archivo de entorno Docker (/etc/sysconfig/docker o /etc/default/docker) pertenece al usuario root y grupo root.

h) Verificar que los permisos del archivo de entorno Docker (/etc/sysconfig/docker o /etc/default/docker) se encuentran configurados al menos como 644.

i) Si se usa systemd en el sistema se deben verificar los siguientes permisos:
   i. Verificar que el archivo /etc/sysconfig/docker-network pertenece al usuario root y grupo root.
   ii. Verificar que los permisos del archivo /etc/sysconfig/docker-network se encuentran configurados al menos como 644.
   iii. Verificar que el archivo /etc/sysconfig/docker-registry pertenece al usuario root y grupo root.
   iv. Verificar que los permisos del archivo /etc/sysconfig/docker-registry se encuentran configurados al menos como 644.
   v. Verificar que el archivo /etc/sysconfig/docker-storage pertenece al usuario root y grupo root.
   vi. Verificar que los permisos del archivo /etc/sysconfig/docker-storage se encuentran configurados al menos como 644.

j) Verificar que el directorio /etc/docker pertenece al usuario root y grupo root.

k) Verificar que los permisos del directorio /etc/docker se encuentran configurados al menos como 755.

l) Verificar que el certificado del "registry" pertenece al usuario root y grupo root.

m) Verificar que los permisos del certificado del registry se encuentran configurados al menos como 444.

n) Verificar que el certificado TLS CA pertenece al usuario root y grupo root.

o) Verificar que los permisos del certificado TLS CA se encuentran configurados al menos como 444.

p) Verificar que el certificado del servidor Docker pertenece al usuario root y grupo root.

q) Verificar que los permisos del certificado del servidor Docker se encuentran configurados al menos como 444.

r) Verificar que el archivo de clave del certificado del servidor Docker pertenece al usuario root y grupo root.

s) Verificar que los permisos del archivo de clave del certificado del servidor Docker se encuentran configurados al menos como 400.

t) Verificar que el archivo de socket de Docker pertenece al usuario root y grupo docker.

u) Verificar que los permisos del archivo de socket de Docker se encuentran configurados al menos como 600.

6.6 CONFIGURACIÓN DE IMÁGENES DE CONTENEDORES

Una imagen de un contenedor se asemeja a una "snapshot" de una máquina virtual o una plantilla de un sistema operativo siendo ésta más ligera que las descritas anteriormente. Desde una misma imagen se pueden ejecutar varios contenedores, por tanto, la imagen escogida para la realización de contenedores debe ser el punto de partida de la configuración de seguridad para los contenedores que se van a generar.

Existen infinidad de imágenes públicas con elementos básicos configurados como Java, Apache y diferentes sistemas operativos en las que se basan esas imágenes. La elección en la imagen base debe centrarse en garantizar la seguridad inicial, pasando por comprobar que se ha obtenido de fuentes oficiales y que no se ha modificado desde el inicio hasta la puesta en marcha en el servidor. Para esta tarea, siempre que se usen repositorios confiables de fuentes oficiales del proveedor, existen herramientas que pueden generar contenedores propios y personalizados como "docker build" o "buildah".

Para una correcta configuración y puesta en marcha de imágenes y contenedores, se debe tener en cuenta las siguientes configuraciones:

a) Los contenedores deben tener un usuario no común.

b) Usar imágenes de confianza para los contenedores de fuentes oficiales.

c) No instalar paquetes innecesarios en el contenedor.

d) No iniciar servicios innecesarios en el contenedor.

e) Regenerar las imágenes con parches de seguridad.

f) Actualizar periódicamente tanto la imagen inicial como los contenedores.

6.7 CONFIGURACIÓN A NIVEL DE EJECUCIÓN

La puesta en marcha de los contenedores motiva nuevas ramas de configuración como el nivel de ejecución de los mismos. En este nivel llamado "runtime daemon" se debe asegurar la ejecución de los procesos, la compartición de recursos con el anfitrión, los accesos administrativos y remotos, así como cualquier proceso que dependa de los contenedores actualmente en ejecución.

Para esta tarea se deben modificar ciertas configuraciones que limiten tanto la comunicación desde el anfitrión a los contenedores como los accesos a la configuración. Se describen a continuación ciertas configuraciones que deben tenerse en cuenta para la ejecución de los contenedores:

a) Verificar las configuraciones de seguridad de SELinux (RedHat, CentOS o Fedora).

b) Verificar que los contenedores estén ejecutando un solo proceso principal.

c) Restringir las "Linux Kernel Capabilities" dentro de los contenedores.

d) Configurar correctamente los parámetros del kernel en el anfitrión.

e) No usar contenedores con privilegios.

f) No montar directorios sensibles del anfitrión en los contenedores.

g) No ejecutar ssh dentro de los contenedores.

h) No configurar puertos privilegiados dentro de los contenedores.

i) Abrir solo puertos necesarios en el contenedor.

j) No usar el modo "host network" en un contenedor.

k) Limitar el uso de memoria por contenedor.

l) Configurar la prioridad de uso de CPU apropiadamente.

m) Montar el sistema de ficheros raíz (/) de un contenedor como solo lectura (ro).

n) Limitar el tráfico entrante al contenedor mediante una interfaz específica del anfitrión.

o) Configurar la política de reinicio “on-failure” de un contenedor.

p) No compartir PID de procesos del anfitrión con contenedores.

q) No compartir IPC del anfitrión con contenedores.

r) No exponer directamente dispositivos del anfitrión en contenedores.

s) Sobrescribir el “ulimit” por defecto en tiempo de ejecución si fuera necesario.

6.8 TAREAS ADICIONALES DE SEGURIDAD

De forma complementaria a las configuraciones definidas anteriormente, existen buenas prácticas y tareas adicionales que son de suma importancia para el mantenimiento de seguridad de una infraestructura de contenedores. Estas tareas se definen a continuación:

a) Se deberán realizar auditorías de seguridad tanto en el anfitrión como en los contenedores de forma regular.

b) Se recomienda monitorizar el uso, rendimiento y métricas de los contenedores para detectar posibles cambios en las funciones asignadas a los mismos.

c) Realizar un correcto backup de recuperación para datos del contenedor.

d) Usar un servicio centralizado de recolección de logs.

e) Evitar almacenar imágenes obsoletas o innecesarias.

f) Evitar almacenar contenedores obsoletos o innecesarios.

7. SEGURIDAD EN LA INTERCONEXIÓN

El escenario actual en la interconexión de redes hiperconvergentes requiere la automatización de políticas de seguridad para soportar el aprovisionamiento y la escalabilidad dinámica de las aplicaciones.

En un entorno de estas características es requisito una solución integral, basada en sistemas, a fin de cubrir las necesidades de seguridad para los entornos Cloud y de centros de datos de próxima generación.

La solución escogida debe abordar las necesidades de seguridad del centro de datos de próxima generación, mediante el uso de un enfoque centrado en aplicaciones y un modelo operativo común basado en políticas, y al mismo tiempo, garantizar el cumplimiento y reducir el riesgo de infracciones a la seguridad. Además, debe permitir una administración unificada que haga cumplir las políticas aplicadas en los centros de datos ya sean a través de medios físicos o virtuales logrando una detección y mitigación acelerada de las posibles amenazas.

Si se opta por una solución SDN para la interconexión de entornos hiperconvergentes hay que tener en cuenta ciertas premisas:

a) Modelo de política centrada en la aplicación: Un modelo de confianza cero, que no permita el tráfico a menos que una política lo permita explícitamente.

b) Administración unificada de política de seguridad de capas 4 a 7: Reduce la complejidad operativa y se aumenta la agilidad de TI sin comprometer la seguridad.

c) Segmentación basada en políticas: Segmentación detallada y flexible de los nodos físicos y virtuales basada en políticas de grupo, lo que reduce el alcance del cumplimiento y mitiga los riesgos de seguridad.

d) Cumplimiento automatizado: Evaluación de riesgos de TI en tiempo real y reducción del riesgo de un riesgo de seguridad en las organizaciones.

e) Seguridad integrada de capa 4 para el tráfico transversal: Un firewall independiente de capa 4, distribuido e integrado, para asegurar el tráfico transversal entre los componentes de la aplicación.

f) Alta visibilidad y detección rápida de ataques: Que permita una detección de manera rápida y anticipada de los ataques.

g) Respuesta automatizada ante incidentes: Una respuesta automatizada ante amenazas identificadas en la red.

8. CONFIGURACIÓN SEGURA DEL EQUIPO ANFITRIÓN

La seguridad de un contenedor puede verse comprometida si no se asegura correctamente el anfitrión que lo alberga. Para ello, es recomendable seguir una serie de pautas de configuración desde el momento inicial. Se definirán a continuación los mínimos recomendables para dotar al sistema de una configuración segura.

8.1 PARTICIONADO Y SISTEMA DE ARCHIVOS

Se debe establecer la cantidad y tamaño de las particiones, así como el sistema de archivos a utilizar. Aunque estos factores dependen en gran medida del uso que se vaya a hacer del sistema, se debe seguir una serie de recomendaciones para ayudar a su correcta elección.

Para realizar una correcta implementación del sistema de archivos hay que tener en cuenta los tipos de archivos más comunes que existen en Linux.

Sistema de archivos	Sistema operativo	Descripción
FAT	Heredado	Sistema de archivos heredado que se ha adoptado universalmente.FAT12,FAT16 y FAT32.
Ext2	Linux	El segundo Filesystem: Sigla de "Extended Graphics Array" utilizado por muchas distribuciones Linux.
Ext3	Linux	El tercero Filesystem: Se añadió registro diario (journaling), utilizado por muchas distribuciones Linux.
Ext4	Linux	El cuarto Filesystem: utilizado por muchas distribuciones Linux. “Extiende los límites de almacenamiento.”
JFS	Linux	Journaled File System: fue introducido por IBM y aún se admite, pero ha sido sustituido por Ext4.
XFS	Linux/Red-HAT/CentOS	Sistema de archivos de 64 Bits, actualmente opción por defecto en Red Hat/CentOS Linux.

Sistema de archivos	Sistema operativo	Descripción
ReiserFS	Linux/SUSE	Se trataba de un formato de archivo que estaba en uso en varias distribuciones, pero ha sido reemplazado por Ext3.
Btrfs	Linux/SUSE	SUSE ofrece este sistema por defecto, recomendándolo para particiones críticas del sistema.

Nota: Se optará por elegir XFS como sistema de archivos recomendado.

A continuación, Se muestra una organización de las particiones como ejemplo, siendo viables alternativas en función de los usos del sistema.

PARTICIÓN	TAMAÑO
/	120 GiB xfs
/boot	500 MiB xfs
/boot/efi	Default MiB vfat
/var	50 GiB xfs
/tmp	25 GiB xfs
/var/log	35 GiB xfs
/home	200 GiB xfs
/var/log/audit	15 GiB xfs
/var/www	ELIMINAR
swap	½ Memoria RAM Equipo
/var/lib/docker	50 GiB xfs [Dependiente de la funcionalidad para implementaciones basadas en Docker]
/var/lib/containers	50 GiB xfs [Dependiente de la funcionalidad para implementaciones basadas en Podman]

Nota: una vez instalado el sistema se recomienda cifrar las particiones aumentando la seguridad de la misma e impidiendo que personal no autorizado pueda acceder a datos críticos.
Se deberá crear una partición independiente para contenedores desde la instalación.

8.1.1 PROTECCIÓN DE LAS PARTICIONES

Para proteger el uso indebido de las diferentes particiones y los ficheros alojados en ellas, se deberá analizar cuál es el uso principal de cada partición, y así determinar las opciones que se utilizarán para montarlas. Estas opciones se reflejarán en el fichero /etc/fstab.

Las opciones que se configuren en el fichero /etc/fstab se aplicarán de forma automática en el inicio de montaje de cada partición.

Las particiones se pueden montar de distintas formas para que limiten determinados permisos:

a) **Noauto**: La partición no se montará automáticamente.

b) **Noexec**: La partición no admitirá la ejecución de ficheros desde la misma.

c) **Nodev**: La partición no admitirá la instalación de dispositivos.

d) **Permisos (ro), (rw):** La partición se configurará con permisos read-only (ro, solo lectura), read-write (rw, lectura y escritura).

Nota: Hay que tener en cuenta que la opción default monta la partición con las opciones rw, suid, dev, exec, auto, nouser, async.

A continuación, se listan las particiones más importantes y las recomendaciones seguras de montaje:

a) **/boot**. Contiene información sobre el arranque del sistema. Se montará con las opciones noauto, noexec, nodev, nosuid, ro.

b) **/boot/efi**. Se montará con los siguientes parámetros umask=0077, shortname=winnt <dump> 0 <pass> 0.

c) **/usr y /opt**. Contienen ficheros ejecutables del sistema. Se montará con las opciones nodev, ro.

d) **/var**. Contiene archivos muy variables del sistema (logs, BBDD, contenido web, etc.). Se montará con las opciones defaults, nosuid.

e) **/var/log**. Contiene los logs del sistema. Se montará con las opciones nodev, noexec, nosuid, rw.

f) **/var/log/audit**. Contiene información y logs de la herramienta Audit. Se montará con las opciones nodev, noexec, nosuid, rw.

g) **/var/www**. Contiene principalmente el contenido Web. Se montará con las opciones nodev, noexec, nosuid, rw.

h) **/home y /tmp**. Contienen los archivos del usuario y los temporales del sistema. Se montará con las opciones nodev, noexec, nosuid, rw y especialmente en /home se le asignará cuota de disco.

i) **/media/XXX**. Contiene montaje de particiones de dispositivos extraíbles. Se montará con las opciones noauto, nodev, nosuid, rw.

j) **/.** Contiene la partición raíz del sistema. Se montará con la opción de lectura y escritura (rw).

k) **Swap.** Se trata de la memoria de intercambio, memoria que usará el sistema cuando necesite espacio en memoria RAM. Se montará con la opción defaults, <dump> 0 <pass> 0.

Excepciones. Para realizar ciertas acciones en determinados momentos se modificará la forma de montaje.

a) **/boot.** Si se tuviera que actualizar el kernel, será necesario montar la partición temporalmente en lectura y escritura (rw).

b) **/usr y /opt.** Si se desea instalar una nueva aplicación o actualizar una ya existente, será necesario montar la partición correspondiente manualmente en modo de lectura y escritura(rw).

c) **/var/lib/containers** o **/var/lib/docker**. Al crear una partición separada para contenedores, se garantiza el poder aislar los permisos de montaje de la misma sin afectar a otras partes del sistema.

Nota: <dump> - Utilizado por el programa dump («volcado») para decidir cuándo hacer una copia de seguridad. Dump comprueba la entrada en el archivo fstab y el número de la misma le indica si un sistema de archivos debe ser respaldado o no. Las entradas posibles son 0 y 1. Si es 0, dump ignorará el sistema de archivos, mientras que, si el valor es 1, dump realizará una copia de seguridad. La mayoría de los usuarios no tendrán dump instalado, por lo que deben poner el valor 0 para la entrada <dump>.
<pass> -Utilizado por fsck para decidir el orden en el que los sistemas de archivos serán comprobados. Las entradas posibles son 0, 1 y 2. El sistema de archivos raíz («root») debe tener la más alta prioridad: 1 - todos los demás sistemas de archivos que desea comprobar deben tener un 2-. La utilidad fsck no comprobará los sistemas de archivos que vengan ajustados con un valor 0 en <pass>.

Una vez realizadas todas las modificaciones debe quedar el fichero /etc/fstab de esta manera:

Particiones	Sistema de archivos	Permisos	
/root	xfs	defaults	1 1
/boot	xfs	noauto,noexec,nodev,nosuid,ro	1 2
/boot/efi	xfs	umask=0077,shortname=winnt	0 0
/usr	xfs	nodev,ro	1 2
/opt	xfs	nodev,ro	1 2
/home	xfs	nodev,noexec,nosuid,rw	1 2
/tmp	xfs	nodev,noexec,nosuid,rw	1 2
/var	xfs	defaults,nosuid	1 2
/var/log	xfs	nodev,noexec,nosuid,rw	1 2
/var/log/audit	xfs	nodev,noexec,nosuid,rw	1 2
/var/www	xfs	nodev,noexec,nosuid,rw	1 2
/swap	swap	defaults	0 0
/var/lib/containers	xfs	Dependerá del tipo de contenedor y su función.	
/var/lib/docker	xfs	Dependerá del tipo de contenedor y su función.	

8.2 PROTECCIÓN DEL SISTEMA

En la actualidad, al menos en algún momento, Importantes organizaciones, tanto públicas como privadas, han sufrido ataques a sus sistemas informáticos. Estos ciberataques no sólo afectan a ciertas compañías, sino que pueden llegar a afectar a la seguridad nacional.

Por esta y por muchas razones es prioritario proteger cualquier sistema que tenga cierta criticidad. Esta parte se centrará en la protección de las partes más vulnerables del sistema, evitando dejar configuraciones por defecto y permisos innecesarios.

8.2.1 CONFIGURACIÓN SEGURA DE RED

Siempre que sea posible deberá configurarse la red de forma estática, asignando direcciones IP de forma manual a cada sistema en lugar de utilizar los protocolos DHCP o BOOTP.

Es recomendable deshabilitar ciertos protocolos que pueden afectar a la vulnerabilidad del sistema y que están orientados a usuarios sin conocimientos de administración de redes, uno de estos protocolos es el protocolo **Zeroconf**.

Zeroconf o APIPA (Automatic Private Ip Address), es un protocolo que se encarga de la asignación automática por parte del sistema operativo de una ip tipo 169.254.X.X con máscara 255.255.0.0. De este modo, dos equipos sin configuración de red, podrían comunicarse entre sí por medio de este protocolo.

Del mismo modo, el protocolo de enrutamiento IPV6 está diseñado para resolver muchos de los problemas que se producen en la versión actual del conjunto de protocolo de Internet (conocido como IPv4) en relación con el agotamiento de direcciones, la configuración automática, la extensibilidad, etc. Este protocolo debe de ser desactivado en caso de que no sea necesaria su utilización para el buen funcionamiento de la red.

Al igual que el protocolo “RPC” (Remote Procedure Call) para IPV6 debe ser deshabilitado si no se contempla administración remota del sistema por medio de redes IPV6.

Para prevenir ataques a las posibles vulnerabilidades en la implementación de algunos protocolos de la pila de red de Linux (dccp, sctp, rds, tipc) se añaden archivos “.conf” al directorio “/etc/modprobe.d” para que se ejecute la shell “/bin/false” en lugar de cargar el módulo del protocolo indicado.

Por otra parte, las tecnologías de contenerización poseen sus propias redes como por ejemplo Docker que por defecto crea una interfaz llamada “docker0” en la que se puede elegir entre varias opciones:

a) **None**. sin red.

b) **Bridge**. Crea una red propia para el stack usando la interfaz docker0 y proporcionando el aislamiento necesario.

c) **Host.** Utiliza directamente la red del anfitrión. Algo poco recomendable y que puede llevar a problemas.

d) **container:<nombre|id>.** Reúsa la red de otro contenedor.

e) **<nombre-red|id-red>**. Se conecta a una red creada por el usuario.

En este sentido la última opción puede ser la opción más recomendable si se considera una red segura donde se pueden configurar todas las limitaciones y restricciones necesarias, aislando los contenedores por medio de dispositivos firewalls o similares.

Por su parte, Podman principalmente utiliza dos formas diferentes de conexiones dependiendo de si el contenedor se define como “rootless” o “rootfull”.

Un contenedor “rootless” pertenecerá al usuario que lo ha creado, así como sus imágenes, mientras que un contenedor “rootfull” pertenecerá al usuario “root” y se convertirá en un contenedor de sistema, siendo accesible solo por root y los usuarios “sudoers” autorizados para tal fin.

Cuando se utiliza Podman con usuarios comunes (rootless), la configuración de la red es automática. Técnicamente, el contenedor en sí no tiene una dirección IP ya que sin privilegios de root, no se puede lograr la asociación de dispositivos de red. Además, hacer ping desde un contenedor "rootless" no es posible puesto que carece de la capacidad de seguridad CAP_NET_RAW que requiere el comando.

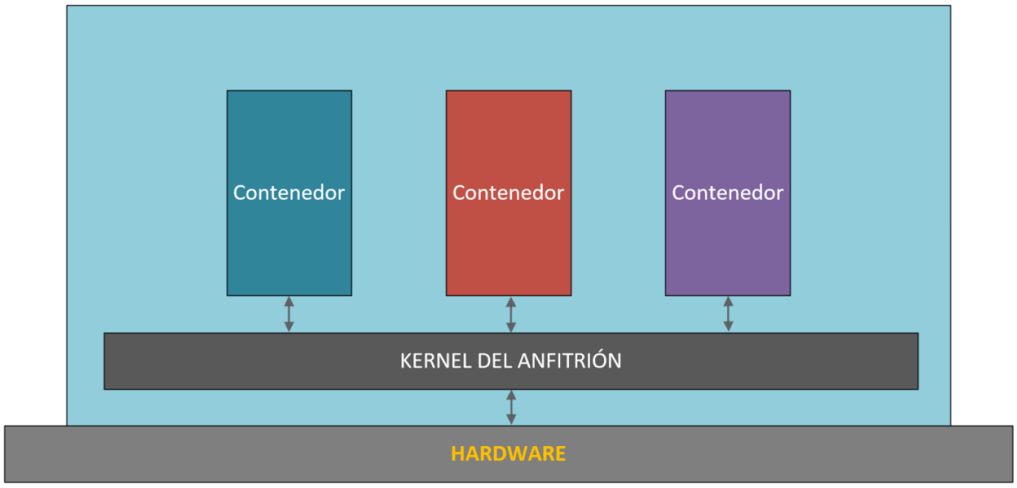
Otra de las configuraciones recomendadas pasa por aislar la comunicación entre contenedores por medio de políticas de firewall o iptables además de por los propios mecanismos que permita la tecnología.

8.2.2 PARÁMETROS DEL KERNEL

El kernel Linux permite modificar una gran cantidad de parámetros sin necesidad de volverlo a compilar. Estos parámetros afectan al funcionamiento del sistema anfitrión en mayor o menor medida así que conviene tener conocimiento de cómo modificarlos. El comando "sysctl" suele ser la forma más común de hacerlo. Los valores se almacenan en el directorio "/proc/sys".

Hay que tener en cuenta que cuando se modifican los parámetros del kernel vía sysctl los cambios surten efecto al instante, pero estos cambios se perderán en el momento que el equipo se reinicie, por eso conviene guardar los cambios en el fichero de configuración de sysctl "/etc/sysctl.conf".

Docker y Podman utilizan las funciones de aislamiento de recursos del kernel de Linux como "cgroups" y "namespaces" del kernel para permitir que se ejecuten contenedores dentro de una sola instancia de Linux. La seguridad del kernel de un contenedor se encuentra por tanto supeditada a las configuraciones del kernel del anfitrión, a sus actualizaciones, etc.



En esta guía se verá como configurar el sistema con ciertos parámetros que afectan a la seguridad ya sea directa o indirectamente.

a) **No responder a peticiones icmp.** Los mensajes ICMP pueden ser utilizados por atacantes remotos, ya sea para identificar ciertas máquinas activas o para intentan explotar las debilidades del protocolo ICMP. Este se ha diseñado para comunicaciones unidireccionales que no requieren autentificación, lo cual habilita a los atacantes a desencadenar ataques DoS o ataques que brindan acceso a los paquetes entrantes y salientes a individuos desautorizados como pueden ser ataques por flujo de ping, por flujo ICMP_ECHO y ataques "smurf".

b) **No responder a peticiones broadcast.** Cuando una máquina envía un paquete a la dirección de broadcast (por ejemplo, 192.168.1.255), éste es entregado a todas las máquinas existentes en la red local. A continuación, todas las máquinas deben enviar un mensaje ECHO del protocolo ICMP. Esto puede provocar una congestión de la red, a la vez que permite determinar que sistemas están activos en la red.

c) **Deshabilitar source routing.** El source routing (o encaminamiento en origen) es una funcionalidad propia del protocolo IP que permite enviar dentro del mismo paquete de datos la información necesaria para su enrutamiento, es decir, la dirección IP de cada uno de los dispositivos de red intermedios q deben cruzarse hasta llegar al destino final. Esto permite al emisor de un paquete dictar la ruta por la que deberá transmitirse a lo largo de la red. Esta característica presenta un grave riesgo de seguridad. De hecho, la mayoría de los routers ignoran ya por defecto esta opción.

d) **Protegerse ante ataques tcp syn.** El "ataque SYN" (también denominado "inundación TCP/SYN") consiste en saturar el tráfico de la red aprovechando el mecanismo de negociación de tres vías del protocolo TCP comenzando varias veces el proceso de establecimiento de conexión a una máquina, sin llegar a completarlo.

e) **Deshabilitar la redirección icmp.** Si un host envía un paquete por una ruta no válida, los routers utilizarán los mensajes de redireccionamiento ICMP para informarle a los hosts en el link de datos que está disponible una ruta mejor para el destino en particular. Dicho mensaje origina que el host modifique sus tablas de enrutamiento. Sin embargo, si un atacante tiene la capacidad de enviar este tipo de mensajes de redirección, podría modificar las tablas de enrutamiento a voluntad, pudiendo conseguir que todo su tráfico saliente se enrutara a otra máquina controlada por el atacante. Por lo tanto, y a pesar de las ventajas que supone en sí mismo este tipo de redirección, podría ser interesante ignorarla a fin de evitar una posible vía de ataque.

f) **Deshabilitar la redirección ip.** La redirección IP se refiere a la capacidad que dispone un sistema de varias interfaces de conexión a distintas subredes, de recibir por una los paquetes destinados a cualquier otra. Este comportamiento es correcto en equipos o dispositivos que actúen como routers o cortafuegos, pero no en un equipo ordinario.

g) **Ignorar los mensajes de error mal formados.** El protocolo ICMP, dispone de mensajes de error para notificar alguna situación anormal en la red. Sin embargo, esta característica se puede utilizar para atacar a los equipos, ya que se les puede inducir a pensar que la red está en un estado distinto al real. En muchas ocasiones, un mensaje de error mal formado indica que se está cometiendo un ataque.

h) **Protección frente a ip spoofing.** Esta protección impide que el sistema sea utilizado para el envío de paquetes IP cuya dirección de destino sea inválida, lo que puede ser indicativo de que se está cometiendo un ataque con el fin de saturar los recursos de comunicación suplantando una dirección ip válida.

i) **Logging de actividades sospechosas.** Mediante esta protección se consigue que el sistema anote en sus registros (logs) la ocurrencia de paquetes con dirección IP inválida (conocido como ataque “IP spoofing”), paquetes que indiquen cambios de rutas (por ejemplo, por haberse activado en el origen el “source routing”) y la ocurrencia de otros paquetes anormales o excepcionales.

j) **Protección frente a buffer overflow.** ASLR (Address Space Layout Randomization) es una técnica de seguridad implicada en la protección de los ataques de desbordamiento de la pila.

k) **Bloqueo IPv6.** La mayoría de las distribuciones Linux es que IPv6 venga configurado por defecto. Sin embargo, no son muchos los usuarios que, aun teniendo esta configuración, hagan uso de alguna aplicación o servicio sobre IPv6, al menos conscientemente. Sus equipos pueden cambiar a trabajar en modo IPv6 en cualquier momento, y a veces en el menos inesperado, haciendo que sea víctima de algún ataque de red que afecte a IPv6, como son el de envenenamiento de vecinos - algo similar al ARP-Spoofing pero con paquetes ICMPv6 spoofeados que realizan aplicaciones como insane6, parasite6 o Scapy, a un ataque de Rogue DHCPv6 que configure un servicio de DNS o una puerta de enlace maliciosa o a un ataque de Man in the middle por medio del protocolo SLAAC.

l) **Activar la protección "DEFRAGGING".** Esta protección se debería aplicar en equipos que actúen como Gateway y que se dediquen a enmascarar tráfico interno (conocido como "IP-masquerading"). A través de este parámetro se le permitiría dividir los paquetes que lo atraviesan, a fin de evitar un consumo excesivo de recursos. Añadiendo la siguiente línea:

   i. i.net.ipv4.always_defrag = 1

8.3 LIMITACIÓN DE RECURSOS DE USUARIO

Con el fin de limitar la utilización de recursos por parte de los usuarios del sistema y las acciones que los programas que ejecuta pueden llevar a cabo, es necesario aplicar ciertas configuraciones.

8.3.1 BLOQUEAR LA GENERACIÓN DE VOLCADOS DE MEMORIA

Para prevenir la creación de volcados de memoria (core dumps) de programas que abortan su ejecución (ya que esta información puede revelar datos confidenciales, y únicamente tiene valor para desarrolladores), se limitara "soft – core" y "hard – core" a 0.

8.3.2 LÍMITE DE LOS RECURSOS DISPONIBLES PARA CADA USUARIO

Se debe limitar la cantidad de **procesos** que un usuario puede tener simultáneamente, tanto en los contenedores, como en el sistema. Del mismo modo se debe limitar la cantidad de **memoria** residente de la que hace uso un usuario. A demás de los límites anteriores, se debe limitar las **conexiones** simultaneas al sistema que cada usuario puede realizar. Todos estos parámetros se configuran en el siguiente fichero de configuración "/etc/security/limits.conf".

Por último, hay que limitar la cantidad de hilos concurrentes que se ejecutan en el sistema, evitando que cualquier programa que se ejecute aumente hasta provocar una denegación de servicio, esta configuración se realizará en "/etc/sysctl.conf".

8.3.3 BLOQUEAR EL USO DE ATAJOS CRÍTICOS

Para prevenir reinicios del sistema no deseados al utilizar la combinación de teclas Ctrl-Alt-Supr, se debe deshabilitar en los sistemas Core. Para distribuciones de GNU/Linux que utilizan Systemd como sistema de gestión de tareas y servicios durante el inicio, el comportamiento de teclas Ctrl-Alt-Supr se determina por un enlace simbólico denominado "/usr/lib/systemd/system/ctrl-alt-del.target" que apunta hacia el archivo "reboot.target", localizado dentro del mismo directorio.

8.3.4 ESTABLECIMIENTO DE CUOTAS DE DISCO

El uso de cuotas de disco permite limitar la cantidad de espacio en disco que utiliza un usuario. La diferencia respecto a los sistemas de archivos extendidos (extended file system o ext) es que XFS requiere habilitar las cuotas a través del parámetro de kernel" rootflags" en tiempo de arranque (boot). Se debe entonces añadir el parámetro de kernel en la configuración de grub. La variable que contiene los parámetros es "GRUB_CMDLINE_LINUX ".

Una vez activada la característica, se debe de asignar parámetros de cuotas a la partición que se requiera limitar por usuarios, comúnmente se suele asignar cuotas a la partición "/home", puesto que en ella suelen estar los archivos personales de cada usuario. En un entorno de contenedores, es recomendable aplicar cuotas a la partición o particiones que albergaran los contenedores.

8.4 LIMITE DE ACCESO AL SISTEMA

Esta guía se basa en la asunción de que no puede haber ningún sistema perfecto, libre de bugs o errores. Dado que cada entorno cuenta con millones de líneas de código e interacciones software/hardware. Un error crítico en cualquiera de estas interacciones puede ser suficiente para que un software malicioso pueda tomar el control de un sistema.

Por esto mismo se debe limitar al máximo los accesos al sistema, así como los permisos, evitando en la medida de lo posible los automatismos y las posibles formas de intrusión. Reduciendo las consecuencias e incluso previniendo los problemas legales.

8.4.1 CONTROL DE INFORMACIÓN DIVULGADA POR EL SISTEMA

Ciertos ficheros del sistema contienen información que se muestra a los usuarios que intentan acceder al sistema. Esta información deberá revisarse para comprobar que no se está divulgando información confidencial. Así mismo, se sustituirá esa información por avisos legales previniendo las consecuencias del acceso no autorizado al sistema.

8.4.2 CONFIGURACIÓN SEGURA DE SSH

Para evitar el uso de versiones inseguras del protocolo SSH se comprobará que la configuración del cliente SSH fuerza la versión 2 del protocolo modificando en el fichero de configuración /etc/ssh/ssh_config la línea correspondiente al protocolo. Por parte del servidor de SSH se requerirán más directrices que se configurarán en el fichero de configuración /etc/ssh/sshd_config:

a) Se forzará el uso de la versión 2 del protocolo.

b) Se denegará el uso de aplicaciones gráficas de modo remoto por medio de X11

c) Se configurarán los usuarios no administradores como acceso denegado.

d) Se limitará el tiempo total para hacer login en 120 segundos.

e) Se establecerá el tiempo mínimo de inactividad antes de la desconexión.

f) SSH puede emular el comportamiento del comando rsh obsoleto al permitir a los usuarios habilitar el acceso inseguro a sus cuentas a través de archivos .rhosts. por lo que se procederá a eliminar este comportamiento.

g) La autenticación criptográfica basada en host de SSH es más segura que la autenticación .rhosts. Sin embargo, no se recomienda que los hosts confíen unilateralmente entre sí, incluso dentro de una organización. Por lo que se procederá a eliminar esta característica.

h) Se denegarán los inicios de sesión root por medio de SSH.

i) Se denegarán los accesos por medio de usuarios sin contraseña.

j) Se configurará correctamente un banner que disuada a los posibles atacantes.

k) Se garantizará que los usuarios no puedan usar variables de entorno al demonio SSH.

l) Se configurará el uso únicamente del protocolo SSH con los algoritmos de cifrado permitidos.

El acceso a los contenedores debe restringirse de tal manera que se limite cualquier acceso remoto que no sea local. En primera instancia en sistemas clasificados el acceso remoto por SSH solo se podrá mantener para accesos locales entre contenedor y anfitrión. Se deberán aplicar todas las configuraciones de seguridad recomendadas como si de un equipo físico se tratara.

8.4.3 MÓDULOS PAM DE AUTENTICACIÓN

Los administradores de sistemas de una organización deben decidir cuánto acceso administrativo se les otorga a los usuarios dentro de la organización a sus máquinas. A través de un módulo PAM llamado “pam_console.so”, se permiten algunas actividades normalmente reservadas para superusuarios, tales como el reinicio o el montaje de medios removibles, al primer usuario que se conecte en la consola física. Sin embargo, otras tareas importantes de administración de sistemas, tales como la modificación de las configuraciones de la red, configurar un nuevo ratón o montar dispositivos de red, son imposibles sin privilegios administrativos. En consecuencia, los administradores deben decidir cuánto acceso administrativo deberían recibir los usuarios en su sistema.

En el siguiente apartado se definen las acciones recomendadas para el módulo "pam_faillock.so" pudiendo ser válido cualquier módulo que, siendo oficial del sistema operativo Linux, realice la misma función:

a) Se contabilizarán los intentos fallidos de acceso o cambio de privilegios mediante su.

b) Se bloquearán aquellas cuentas que superen 5 intentos fallidos.

c) Para evitar que la cuenta root se bloquee intencionadamente se fijará manualmente un máximo de intentos fallidos.

d) Se recordarán las 7 últimas contraseñas utilizadas por cada usuario y no permitirá su repetición.

e) Se limitará el acceso de usuarios wheel a cuentas administrativas

8.4.4 LÍMITE DE SERVICIOS DEL SISTEMA

Como se comentó en otras partes de la guía se necesita minimizar la superficie de ataque, eliminando elementos innecesarios. Por ello no se deben mantener servicios activos que no son necesarios para el correcto funcionamiento del sistema.

Dependiendo de la distribución escogida como base de los contenedores la forma de controlar los servicios del sistema varía. En la mayoría de las distribuciones actuales, se pasa del uso del comando "service" y del control de servicios a través de "/etc/init.d" a la gestión a través del service manager systemctl.

Se pueden listar fácilmente todos los servicios del sistema que corren al inicio mediante el comando systemctl list-unit-files. Procediendo a deshabilitar los que no sean necesarios.

Si la distribución permite su instalación, es posible usar el asistente "ntsysv" que muestra de manera más gráfica una lista de servicios disponibles los cuales se pueden seleccionar y definir para que arranquen automáticamente junto con el sistema.

8.5 ELEMENTOS INNECESARIOS DEL SISTEMA

En este punto es necesario tratar siempre de deshabilitar todos aquellos elementos del sistema que no sean necesarios, minimizando la superficie de posibles ataques al mismo.

8.5.1 PAQUETES INNECESARIOS

Una de las características del software libre es su carácter colaborativo. De esta manera existen cientos de miles de librerías disponibles, que permiten a los desarrolladores crear una aplicación sin tener que empezar de cero. Disponiendo de componentes de diferentes tamaños con un objetivo o funcionalidad específica y que permiten hacer la aplicación más robusta.

De esta característica se nutren las distribuciones Linux. Para que esas aplicaciones se ejecuten correctamente, se necesita que estén instalados el resto de paquetes. De esta forma, cuando se instala una aplicación, también se instalan aquellos paquetes necesarios para su funcionamiento.

Estos paquetes necesarios son los que se conocen como dependencias. Sin embargo, hay que tener en cuenta el momento de en la desinstalación de una aplicación, puesto que al desinstalar el paquete padre (aplicación principal) no siempre se desinstalan las dependencias. Al contrario, esas dependencias quedan instaladas en el equipo ocupando un espacio innecesario. Estos paquetes son los que se conoce como paquetes huérfanos.

Los contenedores, poseen una distribución de Linux en sí misma, por tanto la revisión de paquetes y dependencias innecesarias se realizará dependiendo de la distribución escogida. No obstante, es igual de importante la eliminación de software innecesario incluido en el contenedor.

8.5.2 USUARIOS INNECESARIOS

Como se ha comentado anteriormente, por defecto el sistema operativo anfitrión o del contenedor, se crean configuraciones para facilitar el uso de los mismos, una de esas configuraciones, son los usuarios predefinidos como ftp, games, etc. Estos usuarios tienen permisos y configuraciones para ciertas partes del sistema operativo. El tener usuarios predefinidos en el S.O puede ser motivo de posibles brechas de seguridad.

Por esto, los usuarios del sistema anfitrión o del sistema del contendor tienen que ser los mínimos necesarios e indispensables, eliminando el resto y restringiendo ciertos permisos a los que por necesidad deban mantenerse.

9. CONSIDERACIONES EN INFRAESTRUCTURAS HIPERCONVERGENTES

Una arquitectura hiperconvergente está compuesta como mínimo de una capa de hipervisor, una red definida por software (SDN) y un almacenamiento definido por software (SDS). Todo ello permite unificar los elementos físicos en capas lógicas, con una gestión centralizada de todo el conjunto. Cada nodo del clúster constituye un bloque funcional que incorpora todos los componentes necesarios: proceso, red y almacenamiento.

Es por ello que, las tecnologías de hiperconvergencia proporcionan la agilidad y escalabilidad necesaria para llevar a cabo la implementación de soluciones basadas en una nube sin tener que renunciar a la propiedad del hardware y la seguridad física del centro de datos propio. Cuando se habla de hiperconvergencia, se habla también de nubes privadas, ya que estos dos conceptos están estrechamente ligados.

Es por ello también que la implementación de sistemas clasificados en un entorno hiperconvergente de tipo nube privada esta admitido siempre y cuando se cumplan con las medidas de seguridad adecuadas al nivel de clasificación de dicho sistema y diseñadas para la hiperconvergencia.

Pasar de virtualización a hiperconvergencia no es un proceso rápido ni sencillo. Requiere de mucha planificación e inversión en hardware específico. Pero, sobre todo, requiere adaptar las medidas de seguridad tradicionales al nuevo modelo de seguridad hiperconvergente.

Las infraestructuras basadas en hiperconvergencia (HCI) están definidas por software en su totalidad, en donde se aíslan todas las operaciones relacionadas con el hardware del sistema y se unifican a nivel de hipervisor en un único bloque.

En cada bloque se integran todos los recursos necesarios de procesamiento, almacenamiento y comunicaciones para que sea autónomo o trabajo en un conjunto de clúster. Todas las funciones esenciales de una infraestructura de servidores se ejecutan en una capa de software estrechamente integrada, en lugar de utilizar un hardware diferente para cada tipo de función.

Para la implantación de una infraestructura hiperconvergente hay que tener en cuenta ciertas pautas:

a) **Infraestructura escalable**. Una de las mayores ventajas de una infraestructura hiperconvergente pasa por la escalabilidad de la misma sin que conlleve extensos tiempos de inactividad. Esta ventaja se logra planificando la infraestructura desde el diseño inicial, ya que, dotar a la infraestructura de una adecuada seguridad sin planificar su futura expansión puede provocar que en el futuro la tarea resulte impracticable.

b) **Compatibilidad con arquitecturas mixtas.** La estructura deseable para una infraestructura hiperconvergente siempre será la adquisición de nuevos dispositivos que logren la implementación de una nueva infraestructura, no obstante, esto no es siempre del todo posible y puede darse el caso de tratar de interconectar redes clasificadas cumpliendo con las medidas de seguridad establecidas según normativa y no ser posible por incompatibilidades entre ellas.

c) **Tratamiento de los datos.** Una infraestructura hiperconvergente debe permitir el traslado de los datos a centros de datos o zonas de almacenamiento ya implementados. Para el correcto cumplimiento de la norma, la información debe ser almacenada por un tiempo predeterminado, esto puede provocar que se deba recurrir a infraestructuras mixtas donde se deba almacenar la información desde la infraestructura hiperconvergente hacia un centro de datos común. Asimismo, el administrador de la infraestructura debe tener la capacidad de migrar y administrar el proceso sin precisar de servicios externos.

d) **Compatibilidad con diversos hipervisores.** Esta característica dependerá en gran medida de las necesidades de la organización. En la medida de lo posible, lo recomendable sería escoger tecnologías hiperconvergentes que tengan soporte con la mayoría de los hipervisores existentes en la actualidad, de esta manera, se logra tener una alternativa viable en caso de necesitar una migración por diversos motivos a una tecnología diferente.

e) **Protección del dato.** La infraestructura hiperconvergente debe estar dotada de medidas de recuperación frente a catástrofes de cualquier índole. Las soluciones de infraestructura hiperconvergentes, como mínimo, deben proporcionar las siguientes protecciones:

   i. Duplicación de datos (RAID). Protege el dato frente a errores humanos o de hardware.

   ii. Replicación local. Proporciona protección de fallos de hardware realizando copias de los datos en múltiples sistemas

   iii. Replicación remota. Proporciona protección frente a cortes de servicio, dotando al sistema de una continuidad frente a errores de hardware.

   iv. Capacidad completa de recuperación ante desastres. La infraestructura debe encontrarse capacitada para una recuperación completa ante desastres.

f) **Almacenamiento efectivo.** En escenarios donde el almacenamiento se encuentra compartido dentro de una infraestructura hiperconvergente, se debe asegurar que la protección de datos está integrada en el sistema sin necesidad de capas de software adicionales. El objetivo es conseguir un almacenamiento seguro, que no disminuya el rendimiento, que reduzca la complejidad de administración y que permita una continuidad absoluta en la información manejada por el sistema.