Configuración del proxy inverso Nginx para el acceso sin VPN a Cisco Finesse (12.6 ES03)

Contenido

Introducción

Este documento describe cómo utilizar un proxy inverso para acceder al escritorio de Cisco Finesse sin conectarse a una VPN basada en las versiones 12.6 ES03 de Cisco Finesse, Cisco Unified Intelligence Center (CUIC) y Cisco Identity Service (IdS).

Nota: Cisco no admite la instalación y configuración de Nginx. Las consultas sobre este tema se pueden discutir en los foros de la comunidad de Cisco.

Nota: Para las implementaciones ES03 de VPN-Less, vea el archivo léame de los componentes individuales para planificar las actualizaciones y verificar las restricciones de compatibilidad. Léame Cisco Finesse 12.6 ES03, CUIC/IdS 12.6 ES03 Léame

Prerequisites

Requirements

Cisco recomienda que tenga conocimiento sobre estos temas:

- Versión de Cisco Unified Contact Center Enterprise (UCCE)
- Cisco Finesse
- administración de Linux
- Administración de red y administración de red de Linux

Componentes Utilizados

La información que contiene este documento se basa en las siguientes versiones de software y hardware.

- Finesse - 12,6 ES03
- CUIC: 12,6 ES03
- IdS - 12.6 ES03
- UCCE/Hosted Collaboration Solution (HCS) para Contact Center (CC): 11.6 o posterior
- Packaged Contact Center Enterprise (PCCE): 12.5 o posterior

Nota: las implementaciones de PCCE/UCCE 2k tendrán que estar en la versión 12.6 de

CCE debido a la implementación de LD/CUIC co-residente

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

Nota: La configuración proporcionada en este documento se ha configurado, reforzado y probado con la carga de proxy inverso Nginx (OpenResty) implementado en CentOS 8.0, en comparación con una implementación de UCCE de 2000 usuarios de ejemplo. La información de referencia del perfil de rendimiento está disponible en este documento.

Antecedentes

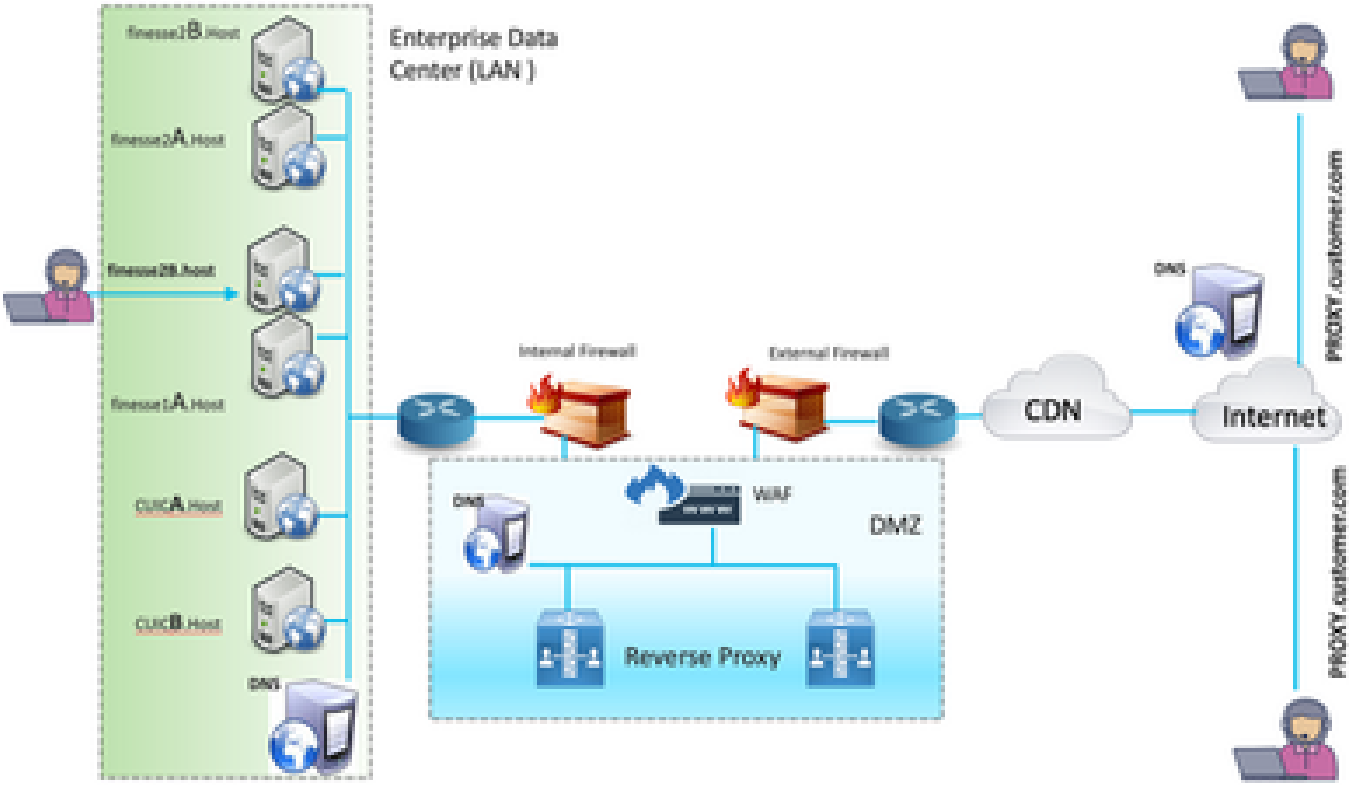
Este modelo de implementación es compatible con las soluciones UCCE/PCCE y HCS para UCCE.

Se admite la implementación de un proxy inverso (disponible desde 12.6 ES01) como opción para acceder al escritorio Cisco Finesse sin conectarse a una VPN. Esta función proporciona la flexibilidad necesaria para que los agentes accedan al escritorio Finesse desde cualquier lugar a través de Internet.

Para habilitar esta función, se debe implementar un par de proxy inverso en la zona desmilitarizada (DMZ).

El acceso a los medios permanece inalterado en las implementaciones de proxy inverso. Para conectarse a los medios, los agentes pueden utilizar Cisco Jabber a través de una solución de acceso remoto y móvil (MRA) o la función de agente móvil de UCCE con una red telefónica pública conmutada (PSTN) o un terminal móvil. Este diagrama muestra el aspecto que tendrá la implementación de red cuando acceda a dos clústeres de Finesse y dos nodos CUIC a través de un único par de nodos de proxy inverso de alta disponibilidad (HA).

Se admite el acceso simultáneo de agentes en Internet y agentes que se conectan desde la LAN, como se muestra en esta imagen.



Nota: Consulte la guía de funciones para conocer los criterios de selección de proxy de terceros en lugar de Nginx para admitir esta implementación.

- Guía de funciones de UCCE 12.6: proporciona una descripción general de las funciones, el diseño y los detalles de configuración para la función VPN-Less.
- Guía de seguridad de UCCE 12.6: proporciona directrices de configuración de seguridad para la implementación de proxy inverso.

Se recomienda revisar la sección VPN-Less de la guía de características y la guía de seguridad antes de leer este documento.

Cambios en ES03

- Características nuevas
  - Ahora se admiten funciones de supervisor Finesse mediante proxy inverso.
  - Los informes históricos y en tiempo real de CUIC se admiten ahora mediante gadgets de Finesse en un entorno con proxy.
  - Autenticación para todas las solicitudes/comunicaciones: requiere compatibilidad con Lua
    - Todas las solicitudes Finesse / CUIC / IM & Presence (IM&P) se autentican en el proxy antes de que se les permita entrar en el Data Center.
    - Las conexiones Websocket y Live Data SocketIO también están restringidas y solo se permiten desde clientes que han realizado con éxito una solicitud segura

a Finesse.
    - Detección y registro de ataques por fuerza bruta en el proxy, que se puede utilizar con Fail2Ban para bloquear direcciones IP malintencionadas.

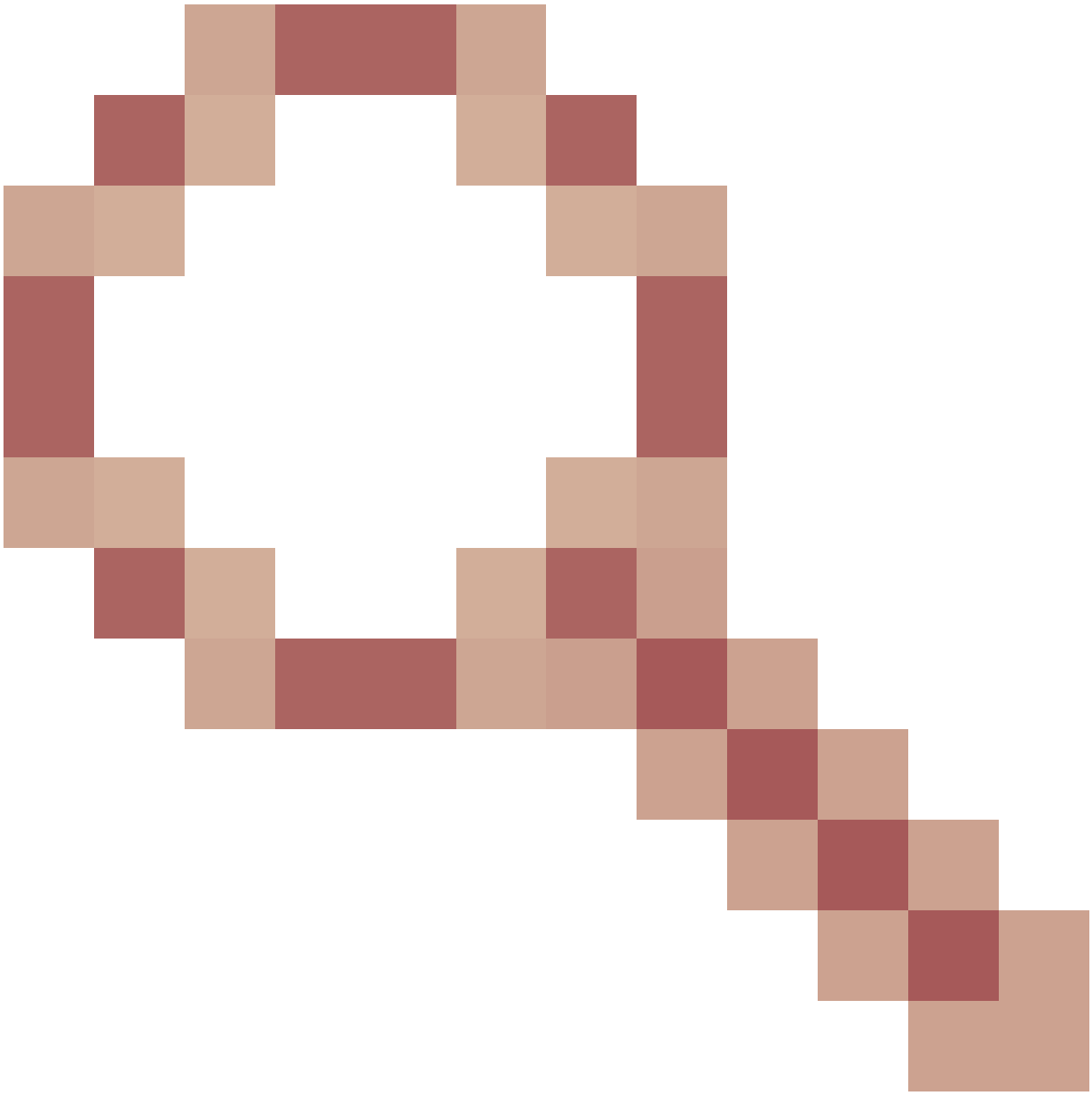
- Mejoras de seguridad para la configuración inversa de proxy; requiere compatibilidad con Lua
    - Autenticación mediante seguridad de capa de transporte mutua (TLS) entre los componentes de proxy inverso y ascendente (Finesse/IdS/CUIC/LiveData).
    - Configuración de SeLinux.
    - Habilitar la verificación de confianza mutua de Secure Sockets Layer (SSL) para las solicitudes de servidor proxy y de componentes.

- La seguridad mejorada de la configuración de proxy para evitar ataques de denegación de servicio (DoS) o de denegación de servicio distribuida (DDoS) requiere compatibilidad con Lua
    - Límites de velocidad de solicitud Nginx mejorados para varias partes del sistema.
    - Límites de velocidad para IpTables.
    - Verificación de solicitudes de recursos estáticos antes de solicitar el servidor de componentes ascendentes.
    - Páginas no autenticadas más claras y almacenables en caché que no llegan al servidor de componentes ascendente.

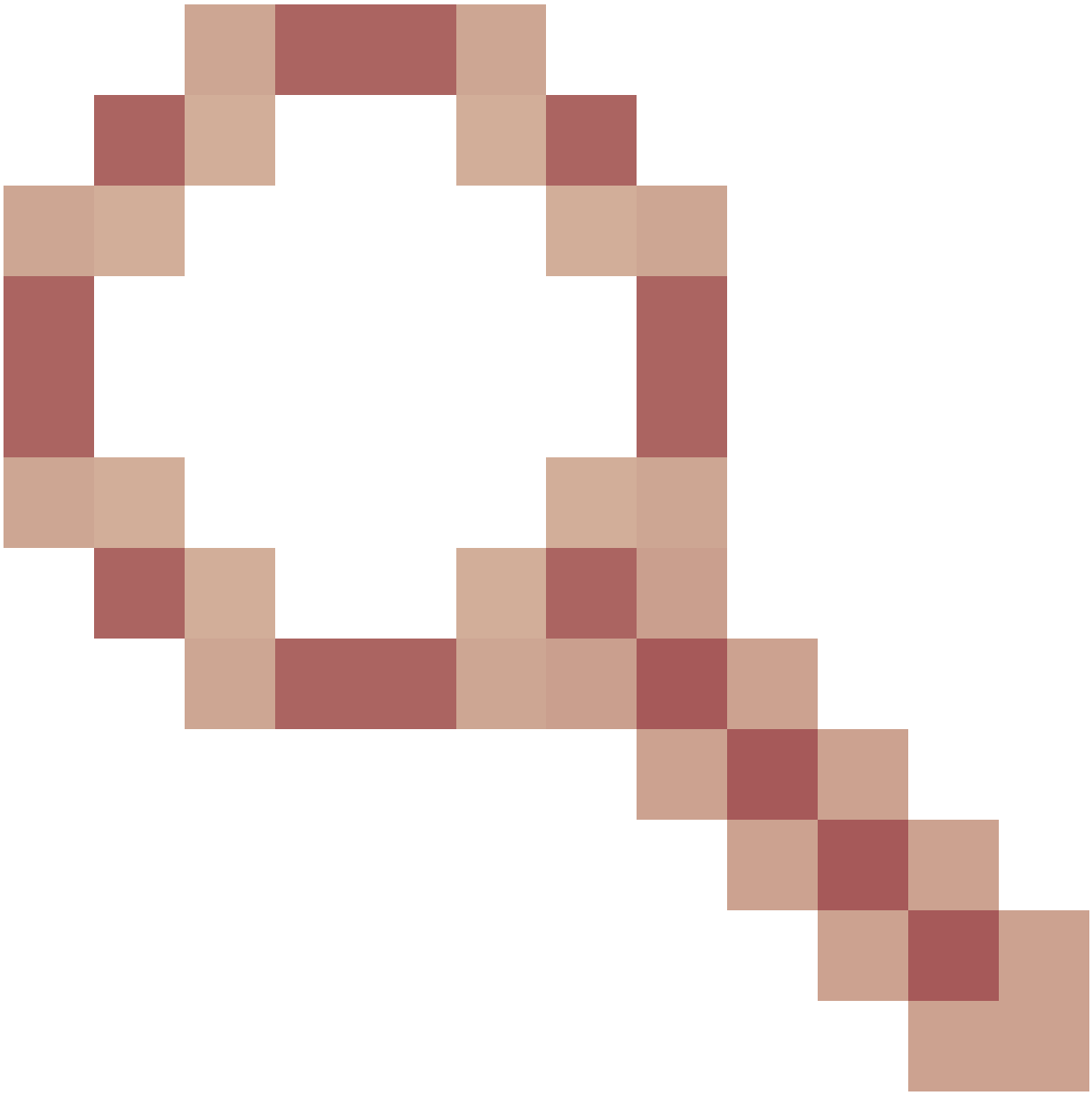
- Otras funciones diversas: requiere compatibilidad con Lua
    - Respuestas de detección automática de uso compartido de recursos entre orígenes (CORS) proporcionadas desde el proxy para facilitar la configuración automática y mejorar el rendimiento

- Corrección de defectos relacionados con VPN-Less

    - CSCwa26057

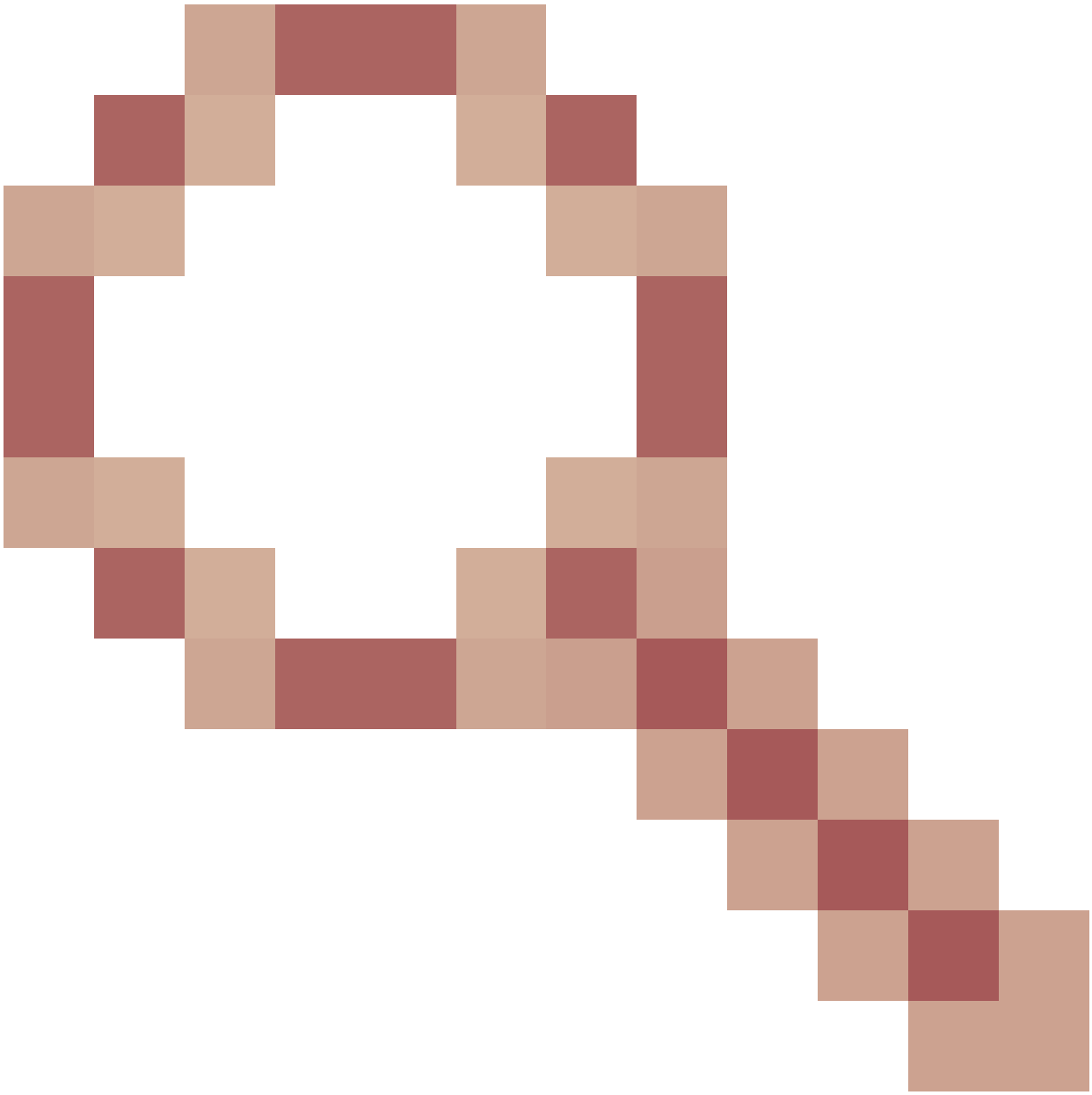


CSCwa26057

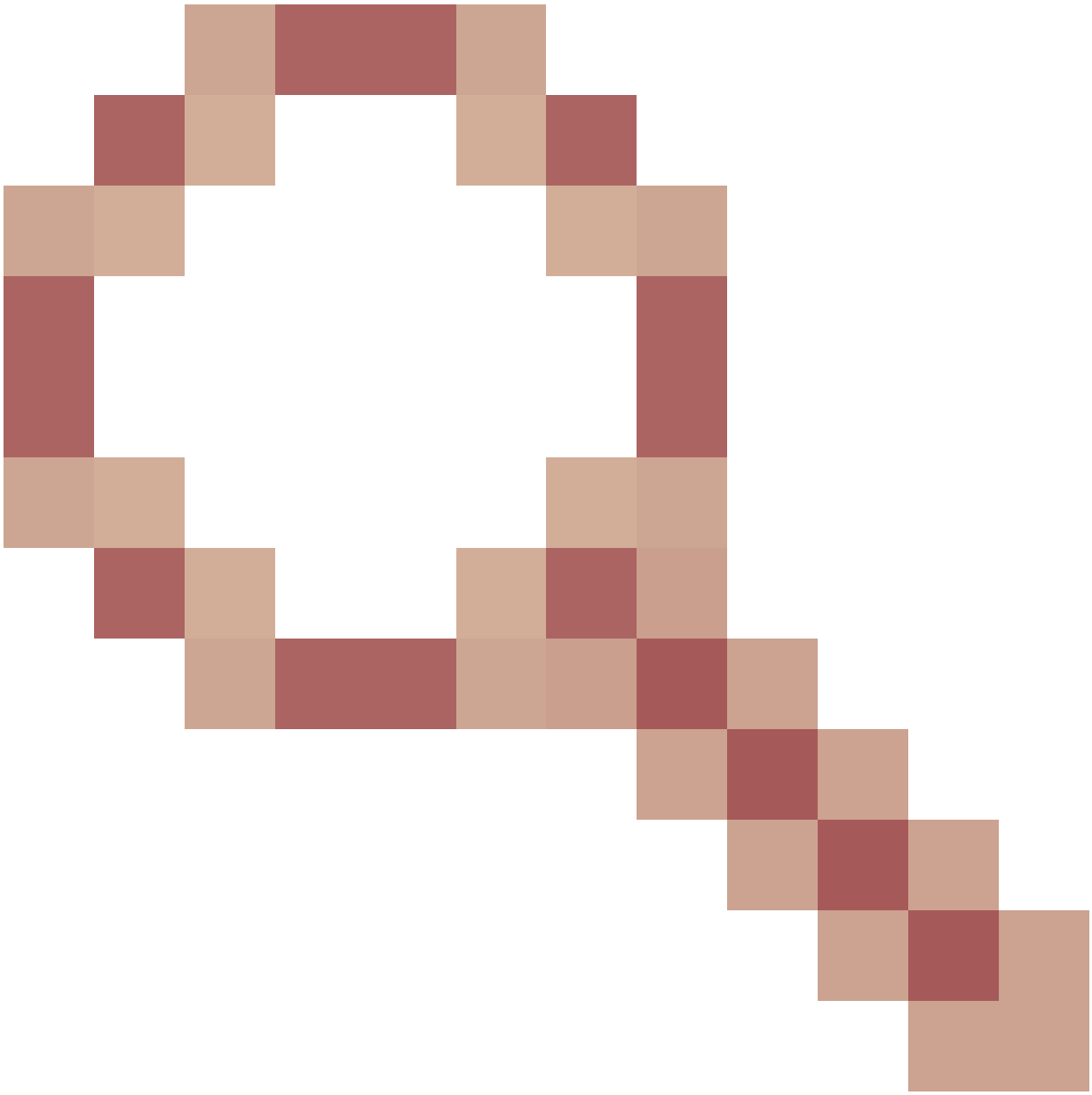


" />- Se ofrecen varios certificados al agente durante el inicio de sesión en el escritorio de Finesse

- CSCwa24471

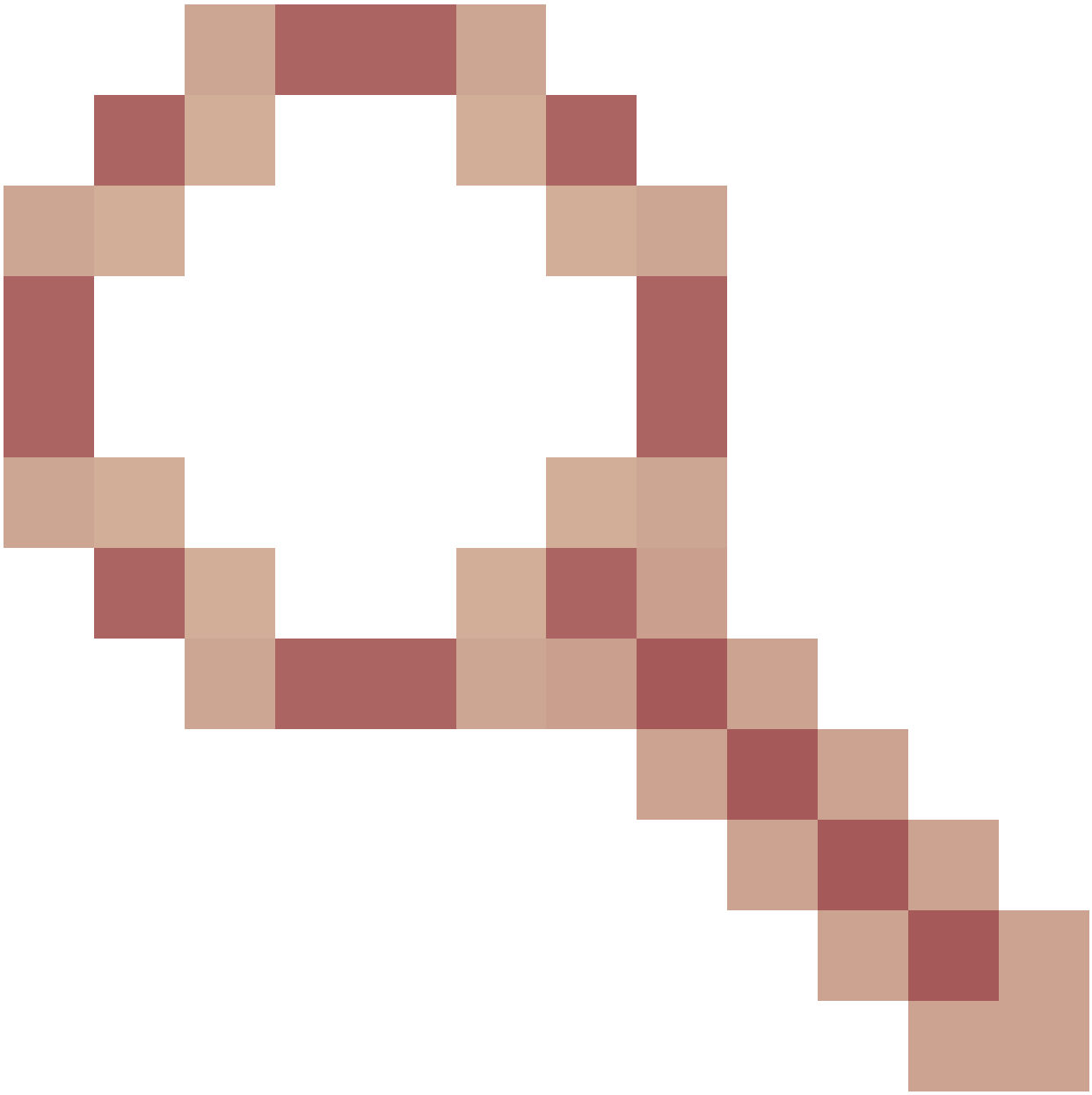


- La página de inicio de sesión de Finesse no muestra el nombre FQDN del agente SSO
  - CSCwa24519



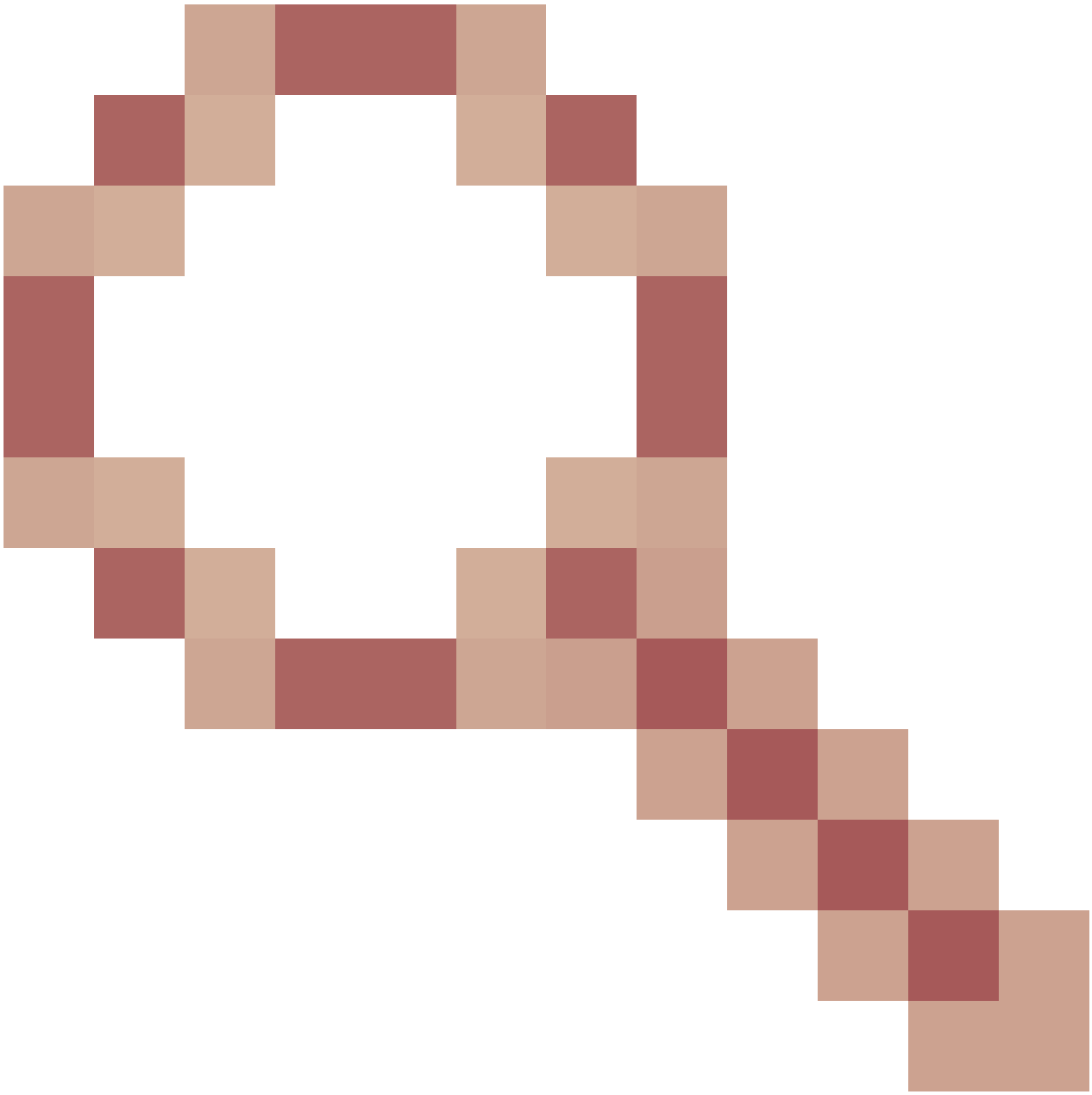
: el servicio Webproxy no se puede reiniciar si el nombre de host del proxy inverso no se puede resolver desde el componente

- CSCwa23252



: la confianza finura de proxy se rompe cuando la profundidad es más de uno para la cadena de certificados de CA

- CSCwa46459



vulnerabilidad de día cero de log4j expuesta en webservice

Notas de actualización para configuraciones basadas en ES01 sin VPN

- La configuración de ES03 requiere la instalación de Nginx con soporte Lua.
- Requisitos del certificado
  - Cisco Finesse, CUIC e IdS requerirán que se agregue el certificado de host Nginx / OpenResty al almacén de confianza de Tomcat y que se realice un reinicio antes de que la configuración de Nginx ES02 pueda conectarse correctamente al servidor ascendente.
  - Los certificados de servidor ascendente Cisco Finesse, CUIC e IdS deben configurarse en el servidor Nginx para utilizar la configuración basada en ES03.

Nota: se recomienda eliminar la configuración existente de Nginx basada en ES01 antes de instalar las configuraciones de Nginx ES03.

Nota: Los scripts de configuración ES03 también requieren la instalación de COP ES03 correspondiente en Cisco Finesse, CUIC e IdS.

Autenticación

Finesse 12.6 ES03 introduce la autenticación en el proxy. La autenticación es compatible con las implementaciones de inicio de sesión único (SSO) y sin SSO.

La autenticación se exige para todas las solicitudes y protocolos que se aceptan en el proxy antes de que se reenvíen a los servidores de componentes ascendentes, donde también tiene lugar la autenticación exigida por los servidores de componentes localmente. Toda la autenticación utiliza las credenciales de inicio de sesión comunes de Finesse para autenticar las solicitudes.

Las conexiones persistentes, como los websockets que se basan en protocolos de aplicación como el protocolo extensible de mensajería y presencia (XMPP) para la autenticación y la conexión posterior, se autentican en el proxy validando la dirección IP desde la que se ha realizado una autenticación de aplicación correcta antes de establecer la conexión de socket.

Autenticación no SSO

La autenticación no SSO no requiere ninguna configuración adicional y funcionará con los scripts de configuración Nginx una vez que se realicen los reemplazos de scripts requeridos. La autenticación se basa en el nombre de usuario y la contraseña utilizados para iniciar sesión en Finesse. El acceso a todos los terminales se validará con los servicios de autenticación Finesse.

La lista de usuarios válidos se almacena en caché en el proxy de forma local (actualiza la caché cada 15 minutos), que se utiliza para validar al usuario en una solicitud. Las credenciales de usuario se validan reenviando la solicitud al URI de Finesse configurado y, a partir de ese momento, el hash de credenciales se almacena en caché localmente (se almacena en caché durante 15 minutos) para autenticar las nuevas solicitudes de forma local. Si se realiza algún cambio en el nombre de usuario o la contraseña, solo se aplicará después de 15 minutos.

Autenticación SSO

La autenticación SSO requiere que el administrador configure la clave de cifrado del token de IdS en el servidor Nginx dentro del archivo de configuración. La clave de cifrado del token IdS se puede obtener del servidor IdS con el comando show ids secret CLI. La clave debe configurarse como parte de uno de los reemplazos #Must-change que el administrador debe realizar en los scripts para que la autenticación SSO funcione.

Consulte la guía del usuario de SSO para conocer las configuraciones SAML de IdS que se deben realizar para que la resolución de proxy funcione para los IdS.

Una vez configurada la autenticación SSO, se puede utilizar un par de tokens válidos para acceder a cualquiera de los terminales del sistema. La configuración de proxy valida las credenciales interceptando las solicitudes de recuperación de token hechas a los IdS o descifrando los tokens válidos y, a continuación, almacenándolos en caché localmente para

futuras validaciones.

Autenticación para conexiones de Websocket

Las conexiones de Websocket no se pueden autenticar con el encabezado de autorización estándar, ya que las implementaciones nativas de websocket en el explorador no admiten encabezados personalizados. Protocolos de autenticación de nivel de aplicación, donde la información de autenticación contenida en la carga útil no impide el establecimiento de la conexión de websocket y, por lo tanto, las entidades maliciosas pueden generar ataques de DOS o DDOS simplemente creando innumerables conexiones para saturar el sistema.

Con el fin de mitigar esta posibilidad, las configuraciones de proxy inverso nginx proporcionadas tienen verificaciones específicas para permitir que las conexiones websocket sean aceptadas SOLAMENTE desde aquellas direcciones IP que hayan realizado con éxito una solicitud REST autenticada antes del establecimiento de la conexión websocket. Esto significa que los clientes que intentan crear conexiones de websocket, antes de que se emita una solicitud REST, ahora obtendrán un error de error de autorización y no es un escenario de uso admitido.

Prevención de ataques de fuerza bruta

Los scripts de autenticación Finesse 12.6 ES02 evitan activamente los ataques de fuerza bruta que se pueden utilizar para adivinar la contraseña del usuario. Para ello, bloquea la dirección IP utilizada para acceder al servicio, después de un cierto número de intentos fallidos en poco tiempo. Estas solicitudes serán rechazadas por error de cliente 418. Se puede acceder a los detalles de las direcciones IP bloqueadas desde los archivos <nginx-install-directory>/logs/blocking.log y <nginx-install-directory>/logs/error.log.

El número de solicitudes fallidas, el intervalo de tiempo y la duración del bloqueo son configurables. Las configuraciones están presentes en el archivo <nginx-install-directory>/conf/conf.d/maps.conf.

```
## These two constants indicate five auth failures from a client can be allowed in thirty seconds.
## if  the threshold is crossed,client ip will be blocked.
map $host $auth_failure_threshold_for_lock {
    ## Must-change Replace below two parameters as per requirement
    default 5 ;
}

map $host $auth_failure_counting_window_secs {
    ## Must-change Replace below two parameters as per requirement
    default 30;
}

## This indicates duration of blocking a client to avoid brute force attack
map $host $ip_blocking_duration {
    ## Must-change Replace below parameter as per requirement
    default 1800;
}
```

Registro

Para encontrar las direcciones IP que están bloqueadas, ejecute los siguientes comandos desde el directorio <nginx-install-directory>/logs.

```
grep "will be blocked for" blocking.log
grep "IP is already blocked." error.log
```

```
2021/10/29 17:30:59 [emerg] 1181750#1181750: *19 [lua] block_unauthorized_users.lua:153:
_redirectAndSendError(): 10.68.218.190 will be blocked for 30 minutes for exceeding retry limit.,
client: 10.68.218.190, server: saproxy.cisco.com, request:
"GET /finesse/api/SystemInfo?nocache=1636456574482 HTTP/2.0", host: "saproxy.cisco.com:8445",
referrer: "https://saproxy.cisco.com:8445/desktop/container/?locale=en_US&"

2021/10/29 19:21:00 [error] 943068#943068: *43 [lua] block_unauthorized_users.lua:53: 10.70.235.30 ::
IP is already blocked..., client: 10.70.235.30, server: saproxy.cisco.com, request:
"GET /finesse/api/SystemInfo?nocache=1635591686497 HTTP/2.0", host: "saproxy.cisco.com:8445",
referrer: "https://saproxy.cisco.com:8445/desktop/container/?locale=en_US"
```

Se recomienda que los clientes se integren con Fail2ban o similar para agregar el ban a las reglas de la tabla IP/firewall.

Instalación y configuración de Fail2ban

Fail2ban analiza los archivos de registro y las IP de bloqueo que muestran signos maliciosos: demasiados fallos de contraseña, búsqueda de vulnerabilidades, etc. Generalmente, Fail2Ban se utiliza para actualizar las reglas del firewall para rechazar las direcciones IP durante un período de tiempo especificado, aunque también se puede configurar cualquier otra acción arbitraria (por ejemplo, enviar un correo electrónico). Para obtener más información, visite https://www.fail2ban.org/.

Fail2ban se puede configurar para monitorear el block.log para identificar las direcciones IP que son bloqueadas por Nginx al detectar ataques de fuerza bruta, y prohibirlas por una duración configurable. Los pasos para instalar y configurar fail2ban en un proxy inverso de CentOS son los siguientes:

1. Instale Fail2ban usando yum.

```
yum update && yum install epel-release
yum install fail2ban
```

2. Crear una cárcel local.

Las configuraciones de la cárcel permiten al administrador configurar diversas propiedades como los puertos a los que se va a prohibir el acceso por cualquier dirección IP bloqueada, el tiempo durante el cual la dirección IP permanece bloqueada, la configuración de filtro utilizada para identificar la dirección IP bloqueada del archivo de registro supervisado, etc. Los pasos para agregar una configuración personalizada para prohibir las direcciones IP que están bloqueadas para acceder a los servidores ascendentes son los siguientes:

2.1. Vaya al directorio de instalación Fail2ban (en este ejemplo /etc/fail2ban)

```
cd /etc/fail2ban
```

2.2. Hacer una copia de jail.conf en jail.local para mantener los cambios locales aislados.

```
cp jail.conf jail.local
```

2.3. Agregue estas configuraciones de carcel al final del archivo jail.local y sustituya los puertos de la plantilla por los puertos reales. Actualice las configuraciones de tiempo de prohibición según sea necesario.

```
# Jail configurations for HTTP connections.
[finesse-http-auth]
enabled = true
# The ports to be blocked. Add any additional ports.
port = http,https,<finesse-ports>,<cuic-ports>,<any-other-ports-to-be-blocked>
# Path to nginx blocking logs.
logpath = /usr/local/openresty/nginx/logs/blocking.log
# The filter configuration.
filter = finesseban
# Block the IP from accessing the port, once the IP is blocked by lua.
maxretry= 1
# Duration for retry set to 3 mins. Doesn't count as the maxretry is 1
findtime= 180
# Lock time is set to 3 mins. Change as per requirements.
bantime = 180
```

3. Configure un filtro.

Un filtro indica a Fail2ban qué buscar en los registros para identificar el host que se va a prohibir. Los pasos para crear un filtro son los siguientes:

3.1. Crear filtro.d/finesseban.conf.

```
touch filter.d/finesseban.conf
```

3.2. Añada estas líneas al filtro de archivos.d/finesseban.conf.

```
[Definition]
# The regex match that would cause blocking of the host.
failregex = <HOST> will be blocked for
```

4. Iniciar Fail2ban.

Ejecute este comando para iniciar fail2ban.

```
fail2ban-client start
```

Abra los archivos de registro fail2ban y compruebe que no haya errores. De forma predeterminada, los registros de fail2ban entran en el archivo /var/log/fail2ban.log.

Validar URL de recursos estáticos

Todos los terminales válidos a los que se puede acceder de una manera no autenticada se mantienen en seguimiento activo en los scripts ES03.

Las solicitudes a estas rutas no autenticadas se rechazan activamente, si se solicita un URI no válido, sin enviar estas solicitudes al servidor ascendente.

Almacenamiento en caché de encabezados CORS

Cuando la primera solicitud de opciones es exitosa, los encabezados de respuesta access-control-allow-header, access-control-allow-origin, access-control-allow-methods, access-control-expose-header y access-control-allow-credentials se almacenan en la memoria caché del proxy durante cinco minutos. Estos encabezados se almacenan en caché para cada servidor ascendente respectivo.

Configurar

Este documento describe la configuración de Nginx como el proxy inverso que se utilizará para habilitar el acceso sin VPN Finesse. Se proporcionan las versiones de SO, proxy y componente de la solución de UCCE utilizadas para verificar las instrucciones proporcionadas. Las instrucciones pertinentes deben adaptarse al sistema operativo/proxy de su elección.

- Nginx versión utilizada - OpenResty 1.19.9.1

- Sistema operativo utilizado para la configuración: CentOS 8.0

Nota: La configuración de Nginx descrita se puede descargar desde la página de descarga de software Finesse Release 12.6(1)ES3.

Configurar componentes de la solución para VPN Menos acceso

Después de configurar el proxy, configure los componentes de la solución (Finesse/CUIC/IdS) para VPN Menos acceso con el nombre de host planificado y la IP del proxy/los servicios utilizados para acceder a la solución con estos comandos.

```
utils system reverse-proxy allowed-hosts add
utils system reverse-proxy config-uri <uri> add
```

Los detalles de estos comandos se pueden encontrar en la Guía de funciones de UCCE 12.6 y se debe hacer referencia a ellos antes de utilizar este documento.

Instalación de OpenResty como proxy inverso en DMZ

En esta sección se detallan los pasos de instalación del proxy basado en OpenResty. El proxy inverso se configura normalmente como un dispositivo dedicado en la zona desmilitarizada (DMZ) de red, como se muestra en el diagrama de implementación mencionado anteriormente.

1. Instale el sistema operativo de su elección con las especificaciones de hardware necesarias. Los ajustes de los parámetros del núcleo e IPv4 pueden diferir en función del sistema operativo seleccionado, por lo que se recomienda a los usuarios que vuelvan a verificar estos aspectos si la versión del sistema operativo seleccionada es diferente.
2. Configure dos interfaces de red. Se necesitará una interfaz para el acceso público desde los clientes de Internet y otra para comunicarse con los servidores de la red interna.
3. Instale OpenResty.

Cualquier sabor de Nginx puede ser utilizado para este propósito, siempre y cuando se basen en Nginx 1.19+ y soporte Lua:

- Nginx Plus
- Nginx Open Source (el código abierto de Nginx tendrá que compilarse junto con los módulos Lua basados en OpenResty para que se pueda utilizar)
- OpenResty
- Extras de GetPageSpeed

Nota: La configuración proporcionada se ha probado con OpenResty 1.19 y se espera que funcione con otras distribuciones con solo actualizaciones menores, si las hubiera.

Instalación de OpenResty

1. Instale OpenResty. Consulte Paquetes Linux OpenResty. Como parte de la instalación de OpenResty, Nginx se instalará en esta ubicación y agregará la ruta de OpenResty a la variable PATH agregando el archivo ~/.bashrc.

   ```
   export PATH=/usr/local/openresty/bin:$PATH
   ```

2. Iniciar / detener Nginx.
   - Para iniciar Nginx, ingrese **openresty**.
   - Para detener Nginx, ingrese **openresty -s stop**.

Configurar Nginx

La configuración se explica para una instalación Nginx basada en OpenResty. Los directorios predeterminados para OpenResty son:

- <nginx-install-directory> = /usr/local/openresty/nginx
- <Openresty-install-directory> = /usr/local/openresty

1. Descargue y extraiga el archivo de la página de descarga del software Finesse Release 12.6(1)ES03 (12.6-ES03-reverse-proxy-config.zip) que contiene la configuración del proxy inverso para Nginx.
2. Copie nginx.conf, nginx/conf.d/, y nginx/html/ desde el directorio de configuración de proxy inverso extraído a <nginx-install-directory>/conf, <nginx-install-directory>/conf/conf.d/, y <nginx-install-directory>/html/ respectivamente.
3. Copie el directorio nginx/lua del directorio de configuración de proxy inverso extraído dentro del <nginx-install-directory>.
4. Copie el contenido de lualib en <Openresty-install-directory>/lualib/resty.
5. Configure la rotación del registro nginx copiando el archivo nginx/logrotate/saproxy en la carpeta <nginx-install-directory>/logrotate/. Modifique el contenido del archivo para que apunte a los directorios de registro correctos si no se utilizan los valores predeterminados de Nginx.
6. Nginx debe ejecutarse con una cuenta de servicio dedicada no privilegiada, que debe estar bloqueada y tener un shell no válido (o según sea aplicable para el SO elegido).
7. Busque la cadena "Must-change" en los archivos de las carpetas extraídas denominadas html y conf.d y reemplace los valores indicados con las entradas correspondientes.
8. Asegúrese de realizar todas las sustituciones obligatorias, que se describen con los comentarios Must-change en los archivos de configuración.
9. Asegúrese de que los directorios de caché configurados para CUIC y Finesse se crean en <nginx-install-directory>/cache junto con estos directorios temporales.
   - <nginx-install-directory>/cache/client_temp
   - <nginx-install-directory>/cache/proxy_temp

Nota: La configuración proporcionada es para una implementación de ejemplo de 2000 y debe ampliarse de forma adecuada para una implementación más grande.

Configuración de Nginx Cache

De forma predeterminada, las rutas de caché del proxy se almacenan en el sistema de archivos. Recomendamos cambiarlos a unidades en memoria creando una ubicación de caché en tmpfs como se muestra aquí.

1. Cree directorios para las diferentes rutas de caché de proxy bajo /home.

   A modo de ejemplo, estos directorios deben crearse para el Finesse principal. Se deben seguir los mismos pasos para los servidores CUIC y Finesse secundarios.

```
mkdir -p /home/primaryFinesse/rest
mkdir -p /home/primaryFinesse/desktop
mkdir -p /home/primaryFinesse/shindig
mkdir -p /home/primaryFinesse/openfire
mkdir -p /home/primaryCUIC/cuic
mkdir -p /home/primaryCUIC/cuicdoc
mkdir -p /home/client_temp
mkdir -p /home/proxy_temp
```

```
echo "tmpfs /home/primaryFinesse/rest tmpfs size=1510M,rw,auto,noexec,nodev,nosuid,gid=root,uid=root,mode=1700 0 0" >>
/etc/fstab echo "tmpfs /home/primaryFinesse/desktop tmpfs size=20M,rw,auto,noexec,nodev,nosuid,gid=root,uid=root,mode=1700 0 0"
>>
/etc/fstab echo "tmpfs /home/primaryFinesse/shindig tmpfs size=500M,rw,auto,noexec,nodev,nosuid,gid=root,uid=root,mode=1700 0 0"
>>
/etc/fstab echo "tmpfs /home/primaryFinesse/openfire tmpfs size=10M,rw,auto,noexec,nodev,nosuid,gid=root,uid=root,mode=1700 0 0"
>>
/etc/fstab echo "tmpfs /home/primaryCUIC/cuic tmpfs size=100M,rw,auto,noexec,nodev,nosuid,gid=root,uid=root,mode=1700 0 0" >>
/etc/fstab echo "tmpfs /home/primaryCUIC/cuicdoc tmpfs size=100M,rw,auto,noexec,nodev,nosuid,gid=root,uid=root,mode=1700 0 0"
>>
/etc/fstab echo "tmpfs /home/client_temp tmpfs size=2048M,rw,auto,noexec,nodev,nosuid,gid=root,uid=root,mode=1700 0 0" >>
/etc/fstab echo "tmpfs /home/proxy_temp tmpfs size=2048M,rw,auto,noexec,nodev,nosuid,gid=root,uid=root,mode=1700 0 0" >>
/etc/fstab
```

   Nota: Aumente las memorias caché de cliente y proxy_temp en 1 GB para cada nuevo clúster de Finesse agregado a la configuración.

2. Monte los nuevos puntos de montaje con el comando **mount -av**.
3. Valide que el sistema de archivos haya montado los nuevos puntos de montaje con el **df -h** comando.
4. Cambie las ubicaciones proxy_cache_path en los archivos de configuración de caché de Finesse y CUIC.

   Por ejemplo, para cambiar las trayectorias para el Finesse primario, vaya a <nginx-install-directory>conf/conf.d/finesse/caches y cambie la ubicación de caché existente

/usr/local/openresty/nginx/cache/finesse25/ a la ubicación del sistema de archivos recién creada /home/primaryFinesse.

##Must-change /usr/local/openresty/nginx/cache/finesse25 location would change depending on folder extraction proxy_cache_path /home/primaryFinesse/desktop levels=1:2 use_temp_path=on keys_zone=desktop_cache_fin25:10m max_size=15m inactive=3y use_temp_path=off; proxy_cache_path /home/primaryFinesse/shindig levels=1:2 use_temp_path=on keys_zone=shindig_cache_fin25:10m max_size=500m inactive=3y use_temp_path=off; proxy_cache_path /home/primaryFinesse/openfire levels=1:2 use_temp_path=on keys_zone=openfire_cache_fin25:10m max_size=10m inactive=3y use_temp_path=off; proxy_cache_path /home/primaryFinesse/rest levels=1:2 use_temp_path=on keys_zone=rest_cache_fin25:10m max_size=1500m inactive=40m use_temp_path=off;

5. Siga los mismos pasos para los servidores CUIC y secundarios de Finesse.

Nota: Asegúrese de que la suma de todos los tamaños de unidad tmpfs creados en todos los pasos anteriores se agrega al tamaño de memoria final para la implementación, ya que estas unidades son bloques de memoria configurados para que la aplicación parezca un disco y consuman tanto espacio de memoria.

Configurar certificados SSL

Utilizar certificados autofirmados: implementaciones de prueba

Los certificados autofirmados sólo se deben utilizar hasta que el proxy inverso esté listo para su lanzamiento en producción. En una implementación de producción, utilice únicamente un certificado firmado por la autoridad certificadora (CA).

1. Generar certificados Nginx para el contenido de la carpeta SSL. Antes de generar certificados, debe crear una carpeta denominada ssl en /usr/local/openresty/nginx. Debe generar dos certificados con la ayuda de estos comandos (uno para <reverseproxy_primary_fqdn> y otro para <reverseproxy_secondary_fqdn>).
   a. **sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /usr/local/openresty/nginx/ssl/nginx.key -out /usr/local/openresty/nginx/ssl/nginx.crt** (pase el nombre de host como: <reverseproxy_primary_fqdn>)
   b. **sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /usr/local/openresty/nginx/ssl/nginxnode2.key -out /usr/local/openresty/nginx/ssl/nginxnode2.crt** (pase el nombre de host como:<reverseproxy_secondary_fqdn>)
   c. Asegúrese de que la ruta del certificado sea /usr/local/openresty/nginx/ssl/nginx.crt y /usr/local/openresty/nginx/ssl/nginxnode2.crt, ya que éstos ya están configurados en los archivos de configuración de Finesse Nginx.
2. Cambie el permiso de la clave privada 400 (r--------).
3. Configure el firewall/iptables en el proxy inverso para habilitar la comunicación desde el firewall para que se corresponda con los puertos en los que se ha configurado el servidor Nginx para escuchar.
4. Agregue la dirección IP y el nombre de host de Finesse, IdS y CUIC bajo la entrada /etc/hosts en el servidor proxy inverso.

5. Consulte la guía de características de la solución para conocer las configuraciones que se deben realizar en los servidores componentes para configurar el host Nginx como proxy inverso.

Nota: La configuración proporcionada es para una implementación de ejemplo de 2000 y debe ampliarse de forma adecuada para una implementación más grande.

Utilizar certificado firmado por CA: implementaciones de producción

Se puede instalar un certificado firmado por CA en el proxy inverso con estos pasos:

1. Genere la solicitud de firma de certificado (CSR).

   Para generar la CSR y la clave privada,**openssl req -new -newkey rsa:4096 -keyout nginx.key -out nginx.csr**ingrese después de iniciar sesión en el proxy. Siga el mensaje y proporcione los detalles. Esto genera la CSR (nginx.csr en el ejemplo) y la clave privada RSA (nginx.key en el ejemplo) de la intensidad 4096 bits.

   Por ejemplo:

   [root@reverseproxyhost.companyname.com ssl]# openssl req -new -newkey rsa:4096 -keyout nginx.key -out nginx.csr Generating a RSA private key .....+++++ .....+++++ writing new private key to 'nginx.key' Enter PEM pass phrase:passphrase Verifying - Enter PEM pass phrase:passphrase ----- You are about to be asked to enter information that will be incorporated into your certificate request. What you are about to enter is what is called a Distinguished Name or a DN. There are quite a few fields but you can leave some blank For some fields there will be a default value, If you enter '.', the field will be left blank. ----- Country Name (2 letter code) [XX]:US State or Province Name (full name) []:CA Locality Name (eg, city) [Default City]:Orange County Organization Name (eg, company) [Default Company Ltd]:CompanyName Organizational Unit Name (eg, section) []:BusinessUnit Common Name (eg, your name or your server's hostname) []:reverseproxyhostname.companydomain.com Email Address []:john.doe@comapnydomain.com Please enter the following 'extra' attributes to be sent with your certificate request A challenge password []:challengePWD An optional company name []:CompanyName

   Anote la frase de contraseña PEM, ya que se utilizará para descifrar la clave privada durante la implementación.

2. Obtenga el certificado firmado de la CA.

   Envíe el CSR a la autoridad de certificación y obtenga el certificado firmado.

   Nota: si el certificado recibido de la CA no es una cadena de certificados que contenga todos los certificados respectivos, cree todos los certificados relevantes en un único archivo de cadena de certificados.

3. Implemente el certificado y la clave.

   Descifrar la clave generada anteriormente como parte del primer paso con**openssl rsa -in nginx.key -out nginx_decrypted.key**el comando. Coloque el certificado firmado por la CA y la clave descifrada

dentro de la carpeta /usr/local/openresty/nginx/ssl en el equipo proxy inverso. Actualice/agregue las configuraciones SSL relacionadas con el certificado en las configuraciones Nginx en el archivo de configuración /usr/local/openresty/nginx/conf/conf.d/ssl/ssl.conf.

```
ssl_certificate /usr/local/openresty/nginx/ssl/ca_signed_cert.crt; ssl_certificate_key /usr/local/openresty/nginx/ssl/nginx_decrypted.key;
```

4. Configure los permisos para los certificados.

   chmod 400 /usr/local/openresty/nginx/ssl/ca_signed_cert.crt Introduzca**chmod 400 /usr/local/openresty/nginx/ssl/nginx_decrypted.key**, de modo que el certificado tenga permiso de sólo lectura y esté restringido al propietario.

5. Recargue Nginx.

Usar parámetro Diffie-Hellman personalizado

Cree un parámetro Diffie-Hellman personalizado con estos comandos:

```
openssl dhparam -out /usr/local/openresty/nginx/ssl/dhparam.pem 2048 chmod 400 /usr/local/openresty/nginx/ssl/dhparam.pem
```

Modifique la configuración del servidor para utilizar los nuevos parámetros en el archivo /usr/local/openresty/nginx/conf/conf.d/ssl/ssl.conf:

```
ssl_dhparam /usr/local/openresty/nginx/ssl/dhparam.pem;
```

Asegúrese de que el grapado de OCSP esté habilitado: comprobación de revocación de certificados

Nota: para habilitar esto, el servidor debe utilizar un certificado firmado por CA y el servidor debe tener acceso a la CA que firmó el certificado.

Agregue o actualice esta configuración en la carpeta file/usr/local/openresty/nginx/conf/conf.d/ssl/ssl.conf:

```
ssl_stapling on; ssl_stapling_verify on;
```

Configuración De Nginx

El archivo de configuración predeterminado de Nginx (/usr/local/openresty/nginx/conf/nginx.conf) debe modificarse para que contenga estas entradas con el fin de aplicar la seguridad y proporcionar rendimiento. Este contenido se debe utilizar para modificar el archivo de configuración predeterminado creado por la instalación de Nginx.

```
# Increasing number of worker processes will not increase the processing the request. The number of wor
# in system CPU. Nginx provides "auto" option to automate this, which will spawn one worker for each CP
worker_processes  auto;

# Process id file location
pid /usr/local/openresty/nginx/logs/nginx.pid;

# Binds each worker process to a separate CPU
worker_cpu_affinity auto;

#Defines the scheduling priority for worker processes. This should be calculated by "nice" command. In
worker_priority 0;


error_log  /usr/local/openresty/nginx/logs/error.log info;

#user root root;

# current limit on the maximum number of open files by worker processes, keeping 10 times of worker_con

worker_rlimit_nofile 102400;

events {
    multi_accept on;


    # Sets the maximum number of simultaneous connections that can be opened by a worker process.
    # This should not be more the current limit on the maximum number of open files i.e. hard limit of
    # The appropriate setting depends on the size of the server and the nature of the traffic, and can
    worker_connections  10240;
    #debug_connection 10.78.95.21
}


http {

    include       mime.types;


    default_type  text/plain;


    ## Must-change Change with DNS resolver ip in deployment
    resolver 192.168.1.3;


    ## Must-change change lua package path to load lua libraries
    lua_package_path  "/usr/local/openresty/lualib/resty/?.lua;/usr/local/openresty/nginx/lua/?.lua;;"


    ## Must-change change proxy_temp folder as per cache directory configurations
    proxy_temp_path /usr/local/openresty/nginx/cache/proxy_temp 1 2 ;
    ## Must-change change client_temp folder as per cache directory configurations
    client_body_temp_path /usr/local/openresty/nginx/cache/client_temp 1 2 ;


    lua_shared_dict userlist 50m;
    lua_shared_dict credentialsstore 100m;
    lua_shared_dict userscount 100k;
    lua_shared_dict clientstorage 100m;
    lua_shared_dict blockingresources 100m;
```

```
lua_shared_dict tokencache_saproxy 10M;
lua_shared_dict tokencache_saproxy125 10M;
lua_shared_dict ipstore 10m;
lua_shared_dict desktopurllist 10m;
lua_shared_dict desktopurlcount 100k;
lua_shared_dict thirdpartygadgeturllist 10m;
lua_shared_dict thirdpartygadgeturlcount 100k;
lua_shared_dict corsheadersstore 100k;


init_worker_by_lua_block {
    local UsersListManager = require('users_list_manager')
    local UnauthenticatedDesktopResourcesManager = require("unauthenticated_desktopresources_manage
    local UnauthenticatedResourcesManager = require("unauthenticated_thirdpartyresources_manager")
    --  Must-change Replace saproxy.cisco.com with reverseproxy fqdn

    if  ngx.worker.id() == 0 then
        UsersListManager.getUserList("saproxy.cisco.com", "https://saproxy.cisco.com:8445/finesse/a
        UnauthenticatedDesktopResourcesManager.getDesktopResources("saproxy.cisco.com", "https://sa
        UnauthenticatedResourcesManager.getThirdPartyGadgetResources("saproxy.cisco.com", "https://
    end
}

include conf.d/*.conf;

sendfile        on;

tcp_nopush     on;

server_names_hash_bucket_size 512;
```

Configurar puerto proxy inverso

De forma predeterminada, la configuración de Nginx escucha las solicitudes del puerto 8445 para Finesse. A la vez, solo se puede habilitar un puerto desde un proxy inverso para admitir solicitudes Finesse, por ejemplo, 8445. Si el puerto 443 necesita ser soportado, edite el archivo <nginx-install-directory>conf/conf.d/finesse.conf para habilitar la escucha en 443 y deshabilitar la escucha en 8445.

Configurar la autenticación TLS mutua entre el proxy inverso y los componentes ascendentes

La autenticación del certificado SSL del cliente para las conexiones de hosts de proxy inverso se puede habilitar en los componentes de flujo ascendente de CCBU CUIC/Finesse/IdS/LiveData mediante la nueva opción CLI de CVOS, que es

utils system reverse-proxy client-auth enable/disable/status.

De forma predeterminada, esta opción está deshabilitada y el administrador debe habilitarla explícitamente mediante la ejecución de CLI en cada servidor ascendente de forma independiente. Una vez que se habilita esta opción, el servicio de proxy web de Cisco que se ejecuta en el host ascendente comenzará a autenticar los certificados de cliente en el protocolo

de enlace TLS para las conexiones que se originan en los hosts de proxy inverso de confianza agregados como parte de CLI utils system reverse-proxy allowed-hosts add <proxy-host>.

A continuación se muestra el bloque de configuración para el mismo en los archivos de configuración proxy, a saber, ssl.conf y ssl2.conf

#Must-change /usr/local/openresty/nginx/ssl/nginx.crt change this location accordingly proxy_ssl_certificate /usr/local/openresty/nginx/ssl/nginx.crt; #Must-change /usr/local/openresty/nginx/ssl/nginx.key change this location accordingly proxy_ssl_certificate_key /usr/local/openresty/nginx/ssl/nginx.key;

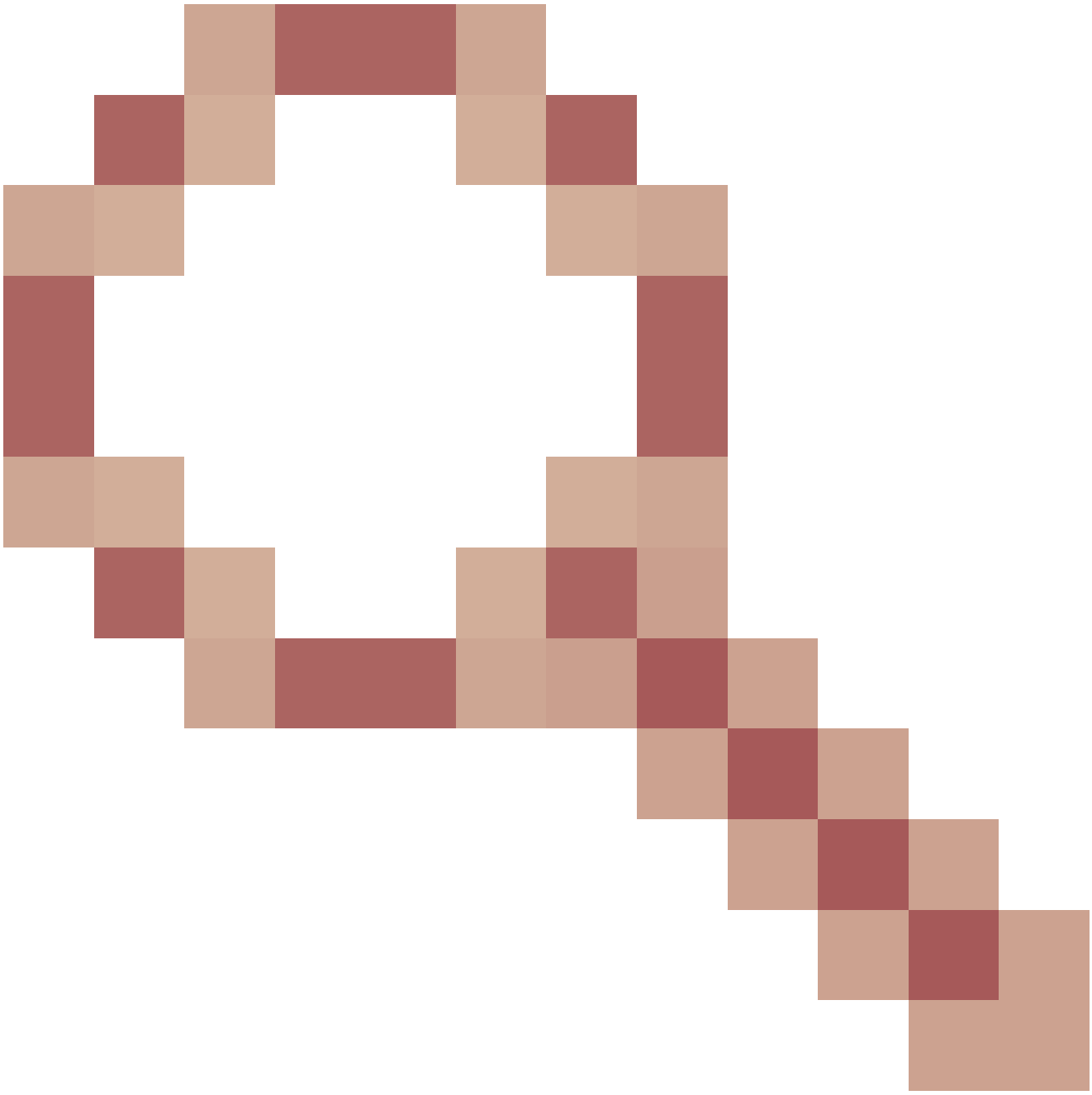
El certificado SSL utilizado para el tráfico saliente (de proxy a ascendente) puede ser el mismo que el certificado SSL configurado para el tráfico entrante (conector SSL para bloques de servidor de componentes). Si el certificado autofirmado se utiliza como proxy_ssl_certificate, debe cargarse en los componentes ascendentes (Finesse/IdS/CUIC/Livedata) del almacén de confianza de tomcat para que se autentique correctamente.

La validación ascendente de certificados de servidor mediante proxy inverso es opcional y está deshabilitada de forma predeterminada. Si desea lograr una autenticación mutua de TLS completa entre el proxy inverso y los hosts ascendentes, la siguiente configuración debe eliminarse de los archivos ssl.conf y ssl2.conf.

#Enforce upstream server certificate validation at proxy -> #this is not mandated as per CIS buit definitely adds to security. #It requires the administrator to upload all upstream server certificates to the proxy certificate store #Must-Change Uncomment below lines IF need to enforce upstream server certificate validation at proxy #proxy_ssl_verify on; #proxy_ssl_trusted_certificate /usr/local/openresty/nginx/ssl/finesse25.crt; proxy_ssl_trusted_certificate: This file should contain the all upstream certificate enteries concatenated together

Advertencias para configurar la autenticación TLS mutua:

- Una vez que esta función esté habilitada en los componentes CCBU, el certificado de cliente se solicitará a los clientes LAN también durante el intercambio de señales TLS. En caso de que se instalen certificados de cliente/personales en los exploradores de equipos cliente, puede optar por mostrar una ventana emergente al usuario final solicitándole que elija el certificado apropiado para la autenticación de cliente. Aunque no importa qué certificado elija el usuario final o pulse cancelar en las solicitudes emergentes, la autenticación de certificado de cliente no se aplica para los clientes LAN, pero habrá cambios en la experiencia. Consulte CDET CSCwa26057



para ver más detalles.

- El servicio webproxy de los componentes ascendentes no se activa si se agrega un host proxy a la lista de permitidos, lo que no se puede resolver mediante el servicio webproxy. Asegúrese de que los hosts de proxy inverso agregados a la lista de permitidos se pueden resolver desde el componente ascendente mediante la búsqueda de DNS.

Borrar caché

La memoria caché del proxy inverso se puede borrar con el

/clearCache.sh

comando.

Directrices estándar

En esta sección se describen brevemente las directrices estándar que deben seguirse al configurar Nginx como servidor proxy.

Estas directrices se derivan del Centro de seguridad de Internet. Para obtener más información sobre cada directriz, consulte la misma.

1. Siempre se recomienda utilizar la última versión estable de OpenResty y OpenSSL.
2. Se recomienda instalar Nginx en un disco de montaje independiente.
3. El ID de proceso de Nginx debe ser propiedad del usuario raíz (o, según corresponda, del sistema operativo elegido) y debe tener permiso 644 (rw-------) o más estricto.
4. Nginx debe bloquear las solicitudes de hosts desconocidos. Asegúrese de que cada bloque de servidor contenga la directiva nombre_servidor definida explícitamente. Para verificar, busque todos los bloques de servidor en el directorio nginx.conf y nginx/conf.d y verifique que todos los bloques de servidor contengan el nombre del servidor.
5. Nginx debe escuchar solamente en los puertos autorizados. Busque todos los bloques de servidor en el directorio nginx.conf y nginx/conf.d y verifique si hay directivas de escucha para verificar que sólo los puertos autorizados están abiertos para escucha.
6. Dado que Cisco Finesse no admite HTTP, se recomienda bloquear también el puerto HTTP del servidor proxy.
7. El protocolo Nginx SSL debe ser TLS 1.2. Se debe eliminar la compatibilidad con los protocolos SSL heredados. También debe inhabilitar los cifrados SSL débiles.
8. Se recomienda que los registros de acceso y errores de Nginx se envíen al servidor syslog remoto.
9. Se recomienda instalar el módulo mod_security que funciona como un firewall de aplicaciones web. Consulte el manual de ModSecurity para obtener más información. Tenga en cuenta que la carga de Nginx no ha sido verificada dentro del módulo mod_security en su lugar.

Configuración del archivo de asignación

La implementación de proxy inverso del escritorio Finesse requiere un archivo de asignación para configurar la lista de combinaciones de puerto/nombre de host visibles externamente y su asignación a los puertos y nombres de servidor reales que utilizan los servidores Finesse, IdS y CUIC. Este archivo de asignación que se configura en los servidores internos es la configuración clave que permite que los clientes conectados a través de Internet sean redirigidos a los hosts y puertos requeridos que se utilizan en Internet.

El archivo de asignación debe implementarse en un servidor web accesible para los servidores de componentes y su URI debe configurarse para que funcione la implementación. Se recomienda configurar el archivo de asignación mediante un servidor web dedicado disponible en la red. Si dicho servidor no está disponible, se puede utilizar el proxy inverso en su lugar, lo que requerirá que el proxy sea accesible desde dentro de la red y también presenta un riesgo de exponer la información a clientes externos que pueden hacer acceso no autorizado a la DMZ. En la siguiente sección se detalla cómo se puede lograr.

Consulte la guía de funciones para conocer los pasos exactos para configurar el URI del archivo de asignación en todos los servidores de componentes y para obtener más detalles sobre cómo

crear los datos del archivo de asignación.

Utilizar proxy inverso como servidor de archivos de asignación

Estos pasos sólo son necesarios si el proxy inverso también se utiliza como el host del archivo de asignación de proxy.

1. Configure el nombre de host del proxy inverso en el controlador de dominio utilizado por los hosts Finesse/CUIC e IdS de modo que se pueda resolver su dirección IP.
2. Cargue los certificados firmados Nginx generados en ambos nodos bajo tomcat-trust de cmplatform y reinicie el servidor.
3. Actualice los valores Must-change en <NGINX_HOME>/html/proxymap.txt.
4. Recargue las configuraciones de Nginx con el **nginx -s reload** comando.
5. Valide que se puede acceder al archivo de configuración desde otro host de red con el uso del **curl** comando.

Consolidación del núcleo de CentOS 8

Si el sistema operativo elegido es CentOS 8, se recomienda que el endurecimiento/ajuste del núcleo se realice con el uso de estas configuraciones de sistema para las instalaciones que utilizan un servidor dedicado para alojar el proxy.

```
## Configurations for kernel hardening - CentOS8. The file path is /etc/sysctl.conf
## Note that the commented configurations denote that CentOS 8's default value matches
## the recommended/tested value, and are not security related configurations.

# Avoid a smurf attack
net.ipv4.icmp_echo_ignore_broadcasts = 1

# Turn on protection for bad icmp error messages
net.ipv4.icmp_ignore_bogus_error_responses = 1

# Turn on syncookies for SYN flood attack protection
net.ipv4.tcp_syncookies = 1

# Turn on and log spoofed, source routed, and redirect packets
net.ipv4.conf.all.log_martians = 1
net.ipv4.conf.default.log_martians = 1

# Turn off routing
net.ipv4.ip_forward = 0
net.ipv4.conf.all.forwarding = 0
net.ipv6.conf.all.forwarding = 0

net.ipv4.conf.all.mc_forwarding = 0
net.ipv6.conf.all.mc_forwarding = 0

# Block routed packets
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.default.accept_source_route = 0
net.ipv6.conf.all.accept_source_route = 0
net.ipv6.conf.default.accept_source_route = 0
```

```
# Block ICMP redirects
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv6.conf.all.accept_redirects = 0
net.ipv6.conf.default.accept_redirects = 0
net.ipv4.conf.all.secure_redirects = 0
net.ipv4.conf.default.secure_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0

# Filter routing packets with inward-outward path mismatch(reverse path filtering)
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1

# Router solicitations & advertisements related.
net.ipv6.conf.default.router_solicitations = 0
net.ipv6.conf.default.accept_ra_rtr_pref = 0
net.ipv6.conf.default.accept_ra_pinfo = 0
net.ipv6.conf.default.accept_ra_defrtr = 0
net.ipv6.conf.default.autoconf = 0
net.ipv6.conf.default.dad_transmits = 0
net.ipv6.conf.default.max_addresses = 1
net.ipv6.conf.all.accept_ra = 0
net.ipv6.conf.default.accept_ra = 0

# Backlog - increased from default 1000 to 5000.
net.core.netdev_max_backlog = 5000

# Setting syn/syn-ack retries to zero, so that they don't stay in the queue.
net.ipv4.tcp_syn_retries = 0
net.ipv4.tcp_synack_retries = 0

# Max tcp listen backlog. Setting it to 511 to match nginx config
net.core.somaxconn = 511

# Reduce the duration of connections held in TIME_WAIT(seconds)
net.ipv4.tcp_fin_timeout = 6

# Maximum resources allotted
# fs.file-max = 2019273
# kernel.pid_max = 4194304
# net.ipv4.ip_local_port_range = 32768 60999

# TCP window size tuning
# net.ipv4.tcp_window_scaling = 1
# net.core.rmem_default = 212992
# net.core.rmem_max = 212992
# net.ipv4.tcp_rmem = 4096 87380 6291456
# net.ipv4.udp_rmem_min = 4096
# net.core.wmem_default = 212992
# net.core.wmem_max = 212992
# net.ipv4.tcp_wmem = 4096 16384 4194304
# net.ipv4.udp_wmem_min = 4096
# vm.lowmem_reserve_ratio = 256 256 32 0 0
# net.ipv4.tcp_mem = 236373 315167 472746

# Randomize virtual address space
kernel.randomize_va_space = 2

# Congestion control
# net.core.default_qdisc = fq_codel
# net.ipv4.tcp_congestion_control = cubic
```

```
# Disable SysReq
kernel.sysrq = 0

# Controls the maximum size of a message, in bytes
kernel.msgmnb = 65536

# Controls the default maximum size of a message queue
kernel.msgmax = 65536

# Controls the eagerness of the kernel to swap.
vm.swappiness = 1
```

Se recomienda reiniciar después de realizar los cambios recomendados.

Consolidación de tablas IP

IPtables es una aplicación que permite al administrador del sistema configurar las tablas, cadenas y reglas IPv4 e IPv6 proporcionadas por el firewall del núcleo Linux.

Estas reglas de tablas IP se configuran para proteger la aplicación proxy de ataques de fuerza bruta mediante la restricción del acceso en el firewall del kernel de Linux.

Los comentarios de la configuración indican qué servicio se está limitando a la velocidad mediante las reglas.

Nota: si los administradores utilizan un puerto diferente o amplían el acceso a varios servidores utilizando los mismos puertos, se debe realizar el dimensionamiento adecuado para estos puertos en función de estos números.

```
## Configuration for iptables service
## The file path is /etc/sysconfig/iptables
## Make a note for must-change values to be replaced.
## Restart of the iptable service is required after applying following rules

*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]

# Ensure loopback traffic is configured
-A INPUT -i lo -j ACCEPT
-A OUTPUT -o lo -j ACCEPT
-A INPUT -s 127.0.0.0/8 -j DROP

# Ensure ping openeded only for the particular source and blocked for rest
# Must-Change: Replace the x.x.x.x with valid ip address
-A INPUT -p ICMP --icmp-type 8 -s x.x.x.x -j ACCEPT

# Ensure outbound and established connections are configured
-A INPUT -p tcp -m state --state RELATED,ESTABLISHED -j ACCEPT
-A OUTPUT -p tcp -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
```

```
# Block ssh for external interface
# Must-Change: Replace the ens224 with valid ethernet interface
-A INPUT -p tcp -i ens224 --dport 22 -j DROP
# Open inbound ssh(tcp port 22) connections
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT


# Configuration for finesse 8445 port
-A INPUT -p tcp -m tcp --dport 8445 --tcp-flags SYN SYN -m connlimit --connlimit-above 10 --connlimit-ma
-A INPUT -p tcp -m tcp --dport 8445 --tcp-flags SYN SYN -m connlimit --connlimit-above 10 --connlimit-ma
 -A INPUT -p tcp -m tcp --dport 8445 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 6/sec --hashlimi
-A INPUT -p tcp -m tcp --dport 8445 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG -
-A INPUT -p tcp -m tcp --dport 8445 --tcp-flags SYN SYN -j DROP

# Configuration for IdS 8553 port
-A INPUT -p tcp -m tcp --dport 8553 --tcp-flags SYN SYN -m connlimit --connlimit-above 6 --connlimit-ma
-A INPUT -p tcp -m tcp --dport 8553 --tcp-flags SYN SYN -m connlimit --connlimit-above 6 --connlimit-ma
-A INPUT -p tcp -m tcp --dport 8553 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 2/sec --hashlimit
-A INPUT -p tcp -m tcp --dport 8553 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG -
-A INPUT -p tcp -m tcp --dport 8553 --tcp-flags SYN SYN -j DROP

# Configuration for IdP 443 port
-A INPUT -p tcp -m tcp --dport 443 --tcp-flags SYN SYN -m connlimit --connlimit-above 8 --connlimit-mas
-A INPUT -p tcp -m tcp --dport 443 --tcp-flags SYN SYN -m connlimit --connlimit-above 8 --connlimit-mas
-A INPUT -p tcp -m tcp --dport 443 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 4/sec --hashlimit-
-A INPUT -p tcp -m tcp --dport 443 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG --
-A INPUT -p tcp -m tcp --dport 443 --tcp-flags SYN SYN -j DROP


# Must-Change: A2A file transfer has not been considered for below IMNP configuration.
# For A2A for support, these configuration must be recalculated to cater different file transfer scenar

# Configuration for IMNP 5280 port
-A INPUT -p tcp -m tcp --dport 5280 --tcp-flags SYN SYN -m connlimit --connlimit-above 30 --connlimit-ma
-A INPUT -p tcp -m tcp --dport 5280 --tcp-flags SYN SYN -m connlimit --connlimit-above 30 --connlimit-ma
-A INPUT -p tcp -m tcp --dport 5280 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 20/sec --hashlimi
-A INPUT -p tcp -m tcp --dport 5280 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG -
-A INPUT -p tcp -m tcp --dport 5280 --tcp-flags SYN SYN -j DROP

# Configuration for IMNP 15280 port
-A INPUT -p tcp -m tcp --dport 15280 --tcp-flags SYN SYN -m connlimit --connlimit-above 30 --connlimit-m
-A INPUT -p tcp -m tcp --dport 15280 --tcp-flags SYN SYN -m connlimit --connlimit-above 30 --connlimit-m
-A INPUT -p tcp -m tcp --dport 15280 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 20/sec --hashlim
-A INPUT -p tcp -m tcp --dport 15280 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG
-A INPUT -p tcp -m tcp --dport 15280 --tcp-flags SYN SYN -j DROP


# Configuration for IMNP 25280 port
-A INPUT -p tcp -m tcp --dport 25280 --tcp-flags SYN SYN -m connlimit --connlimit-above 30 --connlimit-m
-A INPUT -p tcp -m tcp --dport 25280 --tcp-flags SYN SYN -m connlimit --connlimit-above 30 --connlimit-m
-A INPUT -p tcp -m tcp --dport 25280 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 20/sec --hashlim
-A INPUT -p tcp -m tcp --dport 25280 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG
-A INPUT -p tcp -m tcp --dport 25280 --tcp-flags SYN SYN -j DROP


# Configuration for CUIC 8444 port
-A INPUT -p tcp -m tcp --dport 8444 --tcp-flags SYN SYN -m connlimit --connlimit-above 6 --connlimit-ma
-A INPUT -p tcp -m tcp --dport 8444 --tcp-flags SYN SYN -m connlimit --connlimit-above 6 --connlimit-ma
-A INPUT -p tcp -m tcp --dport 8444 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 2/sec --hashlimit
-A INPUT -p tcp -m tcp --dport 8444 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG -
-A INPUT -p tcp -m tcp --dport 8444 --tcp-flags SYN SYN -j DROP
```

```
# Configuration for CUIC 8447 port
-A INPUT -p tcp -m tcp --dport 8447 --tcp-flags SYN SYN -m connlimit --connlimit-above 6 --connlimit-ma
-A INPUT -p tcp -m tcp --dport 8447 --tcp-flags SYN SYN -m connlimit --connlimit-above 6 --connlimit-ma
-A INPUT -p tcp -m tcp --dport 8447 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 2/sec --hashlimit
-A INPUT -p tcp -m tcp --dport 8447 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG -
-A INPUT -p tcp -m tcp --dport 8447 --tcp-flags SYN SYN -j DROP

# Configuration for LiveData 12005  port
-A INPUT -p tcp -m tcp --dport 12005 --tcp-flags SYN SYN -m connlimit --connlimit-above 10 --connlimit-
-A INPUT -p tcp -m tcp --dport 12005 --tcp-flags SYN SYN -m connlimit --connlimit-above 10 --connlimit-
-A INPUT -p tcp -m tcp --dport 12005 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 6/sec --hashlimi
-A INPUT -p tcp -m tcp --dport 12005 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG
-A INPUT -p tcp -m tcp --dport 12005 --tcp-flags SYN SYN -j DROP

# Configuration for LiveData 12008  port
-A INPUT -p tcp -m tcp --dport 12008 --tcp-flags SYN SYN -m connlimit --connlimit-above 10 --connlimit-
-A INPUT -p tcp -m tcp --dport 12008 --tcp-flags SYN SYN -m connlimit --connlimit-above 10 --connlimit-
-A INPUT -p tcp -m tcp --dport 12008 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 6/sec --hashlimi
-A INPUT -p tcp -m tcp --dport 12008 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG
-A INPUT -p tcp -m tcp --dport 12008 --tcp-flags SYN SYN -j DROP

# Block all other ports
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited

COMMIT
```

Estas reglas se pueden aplicar directamente editando el archivo /etc/sysconfig/iptables manualmente o, alternativamente, guardar la configuración en un archivo como iptables.conf y ejecutar cat iptables.conf >>/etc/sysconfig/iptables para aplicar las reglas.

Se requiere reiniciar el servicio IPtables después de aplicar las reglas. Ingrese **systemctl restart iptables** para reiniciar el servicio IPtables.

Restringir conexiones de cliente

Además de la configuración anterior de tablas IP, se recomienda que las instalaciones que conocen el rango de direcciones para los clientes que utilizan el proxy utilicen este conocimiento para asegurar las reglas de acceso al proxy. Esto puede proporcionar enormes ganancias cuando se trata de proteger el proxy de botnets de redes maliciosas que a menudo se crean en el rango de direcciones IP de países que tienen reglas más laxas con respecto a la seguridad en línea. Por lo tanto, se recomienda restringir los rangos de direcciones IP a rangos de país/estado o basados en ISP si está seguro de los patrones de acceso.

Bloquear conexiones de cliente

También resulta útil saber cómo bloquear un intervalo específico de direcciones cuando se identifica un ataque que se va a realizar desde una dirección IP o un intervalo de direcciones IP. En tales casos, las solicitudes de esas direcciones IP pueden bloquearse con reglas iptable.

Bloquear direcciones IP distintas

Para bloquear varias direcciones IP distintas, agregue una línea al archivo de configuración IPTables para cada dirección IP.

Por ejemplo, para bloquear las direcciones 192.0.2.3 y 192.0.2.4, introduzca:

<#root>

```
iptables -A INPUT -s
```

192.0.2.3

```
 -j DROP iptables -A INPUT -s
```

192.0.2.4

```
 - j DROP.
```

Bloquear un intervalo de direcciones IP

Bloquee varias direcciones IP en un rango y agregue una sola línea al archivo de configuración IPTables con el rango IP.

Por ejemplo, para bloquear direcciones de 192.0.2.3 a 192.0.2.35, introduzca:

```
iptables -A INPUT -m iprange --src-range 192.0.2.3-192.0.2.35 -j DROP.
```

Bloquear todas las direcciones IP de una subred

Bloquee todas las direcciones IP de una subred completa agregando una sola línea al archivo de configuración IPTables con el uso de la notación de ruteo entre dominios sin clase para el rango de direcciones IP. Por ejemplo, para bloquear todas las direcciones de clase C, introduzca:

```
iptables -A INPUT -s 192.0.0.0/16 -j DROP.
```

SELinux

SELinux es un marco de seguridad de plataforma integrado como una mejora en el sistema operativo Linux. El procedimiento para instalar y agregar políticas de SELinux para ejecutar OpenResty como proxy inverso se proporciona a continuación.

1. Detenga el proceso con el **openresty -s stop** comando.
2. Configure e inicie /stop nginx server con el **systemctl** comando para que durante el arranque el proceso de OpenResty se inicie automáticamente. Introduzca estos

comandos como usuario raíz.

a. Diríjase a /usr/lib/systemd/system.
b. Abra un archivo llamado openresty.service.
c. Actualice el contenido del archivo según la ubicación PIDFile.

```
[Unit]
Description=The OpenResty Application Platform
After=syslog.target network-online.target remote-fs.target nss-lookup.target
Wants=network-online.target

[Service]
Type=forking
PIDFile=/usr/local/openresty/nginx/logs/nginx.pid
ExecStartPre=/usr/local/openresty/nginx/sbin/nginx -t
ExecStart=/usr/local/openresty/nginx/sbin/nginx
ExecReload=/bin/kill -s HUP $MAINPID
ExecStop=/bin/kill -s QUIT $MAINPID
PrivateTmp=true

[Install]
WantedBy=multi-user.target
```

d. Como usuario raíz, introduzca **sudo systemctl enable openresty**.
e. Inicie o detenga el servicio OpenResty con el **systemctl start openresty / systemctl stop openresty** comando y asegúrese de que el proceso se inicia o se detiene como usuario raíz.

1. Instalar Selinux
   - De forma predeterminada, sólo algunos paquetes de SELinux se instalarán en CentOs.
   - El paquete policycoreutils-devel y sus dependencias deben ser instalados para generar la política SELinux.
   - Ingrese este comando para instalar policy-coreutils-devel

```
yum install policycoreutils-devel
```

   - Asegúrese de que después de instalar el paquete, el **sepolicy** comando funcione.

```
usage: sepolicy [-h] [-P POLICY]

                {booleans,communicate,generate,gui,interface,manpage,network,transition}
                ...

SELinux Policy Inspection Tool
```

2. Crear un nuevo usuario y mapa de Linux con el usuario de SElinux

   a. Ingrese **semanage login -l** para ver el mapeo entre usuarios de Linux y usuarios de

SELinux.

```
[root@loadproxy-cisco-com ~]# semanage login -l

Login Name           SELinux User         MLS/MCS Range        Service

__default__          unconfined_u         s0-s0:c0.c1023       *                *
root                 unconfined_u         s0-s0:c0.c1023       *
```

b. Como root, cree un nuevo usuario de Linux (nginx user) que esté asignado al usuario SELinux user_u.

```
useradd -Z user_u nginxuser
[root@loadproxy-cisco-com ~]# passwd nginxuser
Changing password for user nginxuser.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

c. Para ver la correspondencia entre nginxuser y user_u, ingrese este comando como root:

```
[root@loadproxy-cisco-com ~]# semanage login -l

Login Name           SELinux User         MLS/MCS Range        Service

__default__          unconfined_u         s0-s0:c0.c1023       *
nginxuser            user_u               s0                   *
root                 unconfined_u         s0-s0:c0.c1023       *
```

d. SELinux __default_ login mapeado de forma predeterminada al usuario SELinux unconfined_u. Es necesario hacer que user_u esté confinado de forma predeterminada con este comando:

```
semanage login -m -s user_u -r s0 __default__
```

Para verificar si el comando funcionó correctamente, ingrese **semanage login -l**. Debe producir este resultado:

```
Login Name           SELinux User         MLS/MCS Range        Service

__default__          user_u               s0                   *
nginxuser            user_u               s0                   *
root                 unconfined_u         s0-s0:c0.c1023       *
```

e. Modifique nginx.conf y realice cambios de propiedad para nginxuser.

   i. Ingrese **chown -R nginxuser:nginxuser** * en el directorio <Openresty-install-directory>.

   ii. Modifique el archivo nginx.conf para incluir nginxuser como usuario para ejecutar los procesos de trabajo.

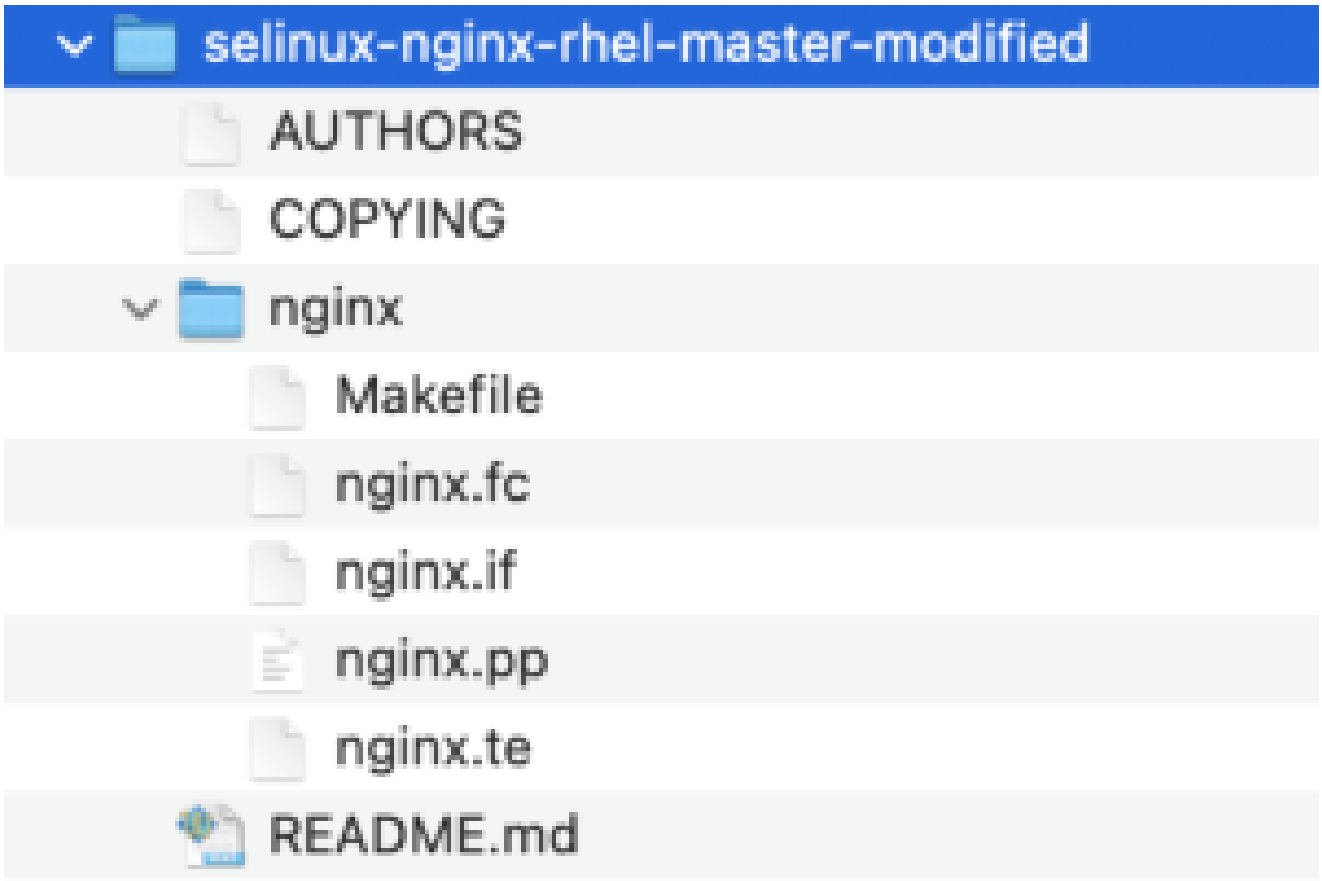
```
........

user nginxuser nginxuser;

..........
```

Escribir la política de SELinux para Nginx

1. En lugar de generar una nueva política personalizada predeterminada para Nginx con el **sepolicy generate --init /usr/bin/nginx** comando, se prefiere comenzar con una política existente.
2. El archivo nginx.fc (archivo de contextos de archivo) y los archivos nginx.te (archivo de aplicación de tipos) descargados de la URL proporcionada se han modificado para ajustarse al uso de proxy inverso.
3. Esta versión modificada se puede utilizar como referencia, ya que se ha corregido para el caso práctico concreto.
4. Descargue el archivo selinux-nginx-rhel-master-modified.tar de la página de descarga del software de archivos.



5. Extraiga el archivo .tar y navegue hasta el directorio nginx dentro de él.
6. Abra el archivo .fc y verifique las rutas de acceso de archivo requeridas del instalador nginx, la memoria caché y el archivo pid.
7. Compile la configuración con el **make** comando.
8. Se generará el archivo nginx.pp.
9. Cargue la directiva con el **semodule** comando.

```
semodule -i nginx.pp
```

10. Vaya a /root y cree un archivo vacío llamado touch /.autorelabel.
11. Reinicie el sistema.
12. Ingrese este comando para verificar que la política se haya cargado correctamente.

```
semodule --list-modules=full
```

```
[root@loadproxy-cisco-com ~]# semodule --list-modules=full
400 nginx                   pp
200 container               pp
200 flatpak                 pp
100 abrt                    pp
100 accountsd               pp
100 acct                    pp
100 afs                     pp
100 aiccu                   pp
100 aide                    pp
100 ajaxterm                pp
100 alsa                    pp
```

13. Nginx debe ejecutarse sin ninguna violación. (Las infracciones estarán disponibles en /var/log/messages y /var/log/audit/audit.log).
14. Ingrese este comando para verificar el estado de Nginx.

```
ps -aefZ | grep nginx
```

```
[root@loadproxy-cisco-com ~]# ps -aefZ |grep nginx
system_u:system_r:nginx_t:s0    root        1686       1  0 16:14 ?        00:00:00 nginx: master process /usr/bin/nginx
system_u:system_r:nginx_t:s0    nginxus+    1687    1686  0 16:14 ?        00:00:00 nginx: worker process
system_u:system_r:nginx_t:s0    nginxus+    1688    1686  0 16:14 ?        00:00:00 nginx: worker process
system_u:system_r:nginx_t:s0    nginxus+    1689    1686  0 16:14 ?        00:00:00 nginx: worker process
system_u:system_r:nginx_t:s0    nginxus+    1690    1686  0 16:14 ?        00:00:00 nginx: worker process
system_u:system_r:nginx_t:s0    nginxus+    1691    1686  0 16:14 ?        00:00:00 nginx: worker process
system_u:system_r:nginx_t:s0    nginxus+    1692    1686  0 16:14 ?        00:00:00 nginx: worker process
system_u:system_r:nginx_t:s0    nginxus+    1693    1686  0 16:14 ?        00:00:00 nginx: worker process
system_u:system_r:nginx_t:s0    nginxus+    1694    1686  0 16:14 ?        00:00:00 nginx: worker process
system_u:system_r:nginx_t:s0    nginxus+    1695    1686  0 16:14 ?        00:00:00 nginx: cache manager process
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 root 2543 2252  0 16:17 pts/0 00:00:00 grep --color=auto nginx
```

15. Ahora el escritorio agente/supervisor de Finesse debe estar accesible.

Verificación

Utilize esta sección para confirmar que su configuración funcione correctamente.

Finesse

1. Solicite https://<reverseproxy:port>/finesse/api/SystemInfo. a la DMZ y compruebe si son accesibles.
2. Los valores de <host> en <primaryNode> y <secondaryNode> son hostnames de proxy inverso válidos. No deben ser hostnames de Finesse.

CUIC y datos en directo

1. Si los nombres de host de Finesse aparecen en la respuesta en lugar de los nombres de host de proxy inverso, valide las configuraciones de asignación de proxy y los hosts permitidos se agregan correctamente en los servidores de Finesse como se describe en la sección "Rellenar datos de traducción de red" de "Acceso VPN-Less a Finesse Desktop" en

la Guía de Funciones de Finesse 12.6 UCCE.
2. Si los gadgets de LiveData se cargan correctamente en Finesse Desktop, las configuraciones de proxy de CUIC y LiveData son las adecuadas.
3. Para validar la configuración de CUIC y LiveData, realice solicitudes HTTP a estas URL desde la DMZ y vea si son accesibles.
    - https://<reverseproxy:cuic_port>/cuic/rest/about
    - https://<reverseproxy:ldweb_port>/livedata/security
    - https://<reverseproxy:ldsocketio_port>/security

IDS

Para validar la configuración de IdS, realice estos pasos:

1. Inicie sesión en la interfaz IdSAdmin en https://<ids_LAN_host:ids_port>:8553/idsadmin desde la LAN, ya que la interfaz de administración no se expone a través del proxy inverso.
2. Elija Settings > IdS Trust.
3. Valide que el nodo editor del clúster de proxy aparezca en la página Descargar metadatos SP y haga clic en Siguiente.
4. Valide que el proxy IDP se muestre correctamente si está configurado en la página Cargar metadatos de IDP y haga clic en Siguiente.
5. El inicio del SSO de prueba a través de todos los nodos de clúster de proxy desde la página Test SSO y la validación de todos se realizan correctamente. Esto requiere conectividad del equipo cliente a los nodos de proxy inverso.

Rendimiento

El análisis de datos de la captura de rendimiento equivalente superior, realizado con la herramienta nmon, está disponible en la página de descarga de software de Finesse versión 12.6(1) ES03 (load_result.zip). Los datos representan el estado del proxy para las operaciones de escritorio y supervisor, en una implementación de 2000 UCCE de muestra mediante inicios de sesión de SSO e informes CUIC LD configurados en el diseño predeterminado para 2000 usuarios durante un período de ocho horas. Se puede utilizar para derivar los requisitos informáticos, de disco y de red de una instalación mediante Nginx en hardware comparable.

Troubleshoot

SSO

1. Las redirecciones de escritorio no se realizan mediante proxy
    1. Verifique que los nombres de host estén configurados en casos correctos según los nombres de host vm reales en varias configuraciones como proxymap.txt, server_filter file etc.
    2. Asegúrese de que IdS se agrega con el nombre de host en mayúsculas correcto en el inventario de CCE, ya que la misma información se envía a los componentes cuando se registra para SSO desde CCE web admin.

2. Los inicios de sesión de SSO no se producen
    1. Asegúrese de que la confianza IdS-IDP esté establecida para el host proxy.

SELinux

1. Si Nginx no se inicia de forma predeterminada o el escritorio del agente Finesse no está accesible, establezca SELinux en el modo permisivo con este comando:

```
setenforce 0
```

2. Intente reiniciar el Nginx con el **systemctl restart nginx** comando.
3. Las infracciones estarán disponibles en /var/log/messages y en /var/log/audit/audit.log.
4. Es necesario volver a generar el archivo .te con reglas de permiso para abordar esas violaciones mediante cualquiera de estos comandos:

```
cat /var/log/audit/audit.log | audit2allow -m nginx1 > nginx1.te. # this will create nginx1.te fil
                or
ausearch -c 'nginx' --raw | audit2allow -M  my-nginx  # this will create my-nginx.te file
```

5. Actualice el archivo original nginx.te presente en el directorio selinux-nginx-rhel-master-modified/nginx con las reglas de permiso recién generadas.
6. Compile el mismo con el **make** comando.
7. El archivo nginx.pp se regenerará.
8. Cargue la directiva mediante un comando de módulo.

```
semodule -i nginx.pp
```

9. Haga que SELinux aplique el modo con este comando:

```
setenforce
```

10. Reinicie el sistema.
11. Repita este procedimiento hasta que se solucionen las infracciones necesarias.

Acerca de esta traducción
Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).