

Configuration du proxy inverse Nginx pour un accès sans VPN à Cisco Finesse (12.6 ES03)

Table des matières

[Introduction](#)

[Conditions préalables](#)

[Exigences](#)

[Composants utilisés](#)

[Informations générales](#)

[Changements dans ES03](#)

[Notes de mise à niveau pour les configurations sans VPN basées sur ES01](#)

[Authentification](#)

[Authentification non SSO](#)

[Authentification SSO](#)

[Authentification pour les connexions WebSocket](#)

[Prévention des attaques par force brute](#)

[Journalisation](#)

[Installation et configuration de Fail2ban](#)

[Valider les URL de ressources statiques](#)

[Mise en cache des en-têtes CORS](#)

[Configurer](#)

[Configurer les composants de la solution pour un accès VPN moindre](#)

[Installer OpenResty en tant que proxy inverse dans DMZ](#)

[Installation de OpenResty](#)

[Configurer Nginx](#)

[Configuration du cache Nginx](#)

[Configurer les certificats SSL](#)

[Utiliser le paramètre Diffie-Hellman personnalisé](#)

[Vérifier que l'agrafage OCSP est activé - Contrôle de révocation de certificat](#)

[Configuration Nginx](#)

[Configurer le port proxy inverse](#)

[Configurer l'authentification TLS mutuelle entre le proxy inverse et les composants en amont](#)

[Effacer le cache](#)

[Lignes directrices types](#)

[Configuration du fichier de mappage](#)

[Utiliser le proxy inverse comme serveur de fichiers de mappage](#)

[Durcissement du noyau CentOS 8](#)

[IPtables Durcissement](#)

[Restreindre les connexions client](#)

[Bloquer les connexions client](#)

[Bloquer les adresses IP distinctes](#)

[Bloquer une plage d'adresses IP](#)

[Bloquer toutes les adresses IP dans un sous-réseau](#)

[SELinux](#)

[Vérier](#)

[Finesse](#)

[CUIC et données en direct](#)

[IDS](#)


[rendement](#)


[Dépannage](#)

[SSO](#)

Introduction

Ce document décrit comment utiliser un proxy inverse pour accéder au bureau Cisco Finesse sans se connecter à un VPN basé sur les versions 12.6 ES03 de Cisco Finesse, Cisco Unified Intelligence Center (CUIC) et Cisco Identity Service (IdS).

 Remarque : l'installation et la configuration de Nginx ne sont pas prises en charge par Cisco. Les questions relatives à ce sujet peuvent être abordées sur les [forums](#) de la [communauté Cisco](#).

 Remarque : pour les déploiements ES03 de VPN-Less, consultez le fichier readme des composants individuels afin de planifier les mises à niveau et vérifier les restrictions de compatibilité. [Lisez-moi Cisco Finesse 12.6 ES03, CUIC / IdS 12.6 ES03](#)

Conditions préalables

Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Version de Cisco Unified Contact Center Enterprise (UCCE)
- Cisco Finesse
- administration Linux
- Administration réseau et administration réseau Linux


Composants utilisés

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Finesse - 12,6 ES03
- CUIC - 12,6 ES03
- IdS - 12.6 ES03
- UCCE/HCS (Hosted Collaboration Solution) pour Contact Center (CC) - version 11.6 ou ultérieure
- Packaged Contact Center Enterprise (PCCE) - version 12.5 ou ultérieure

Remarque : les déploiements PCCE/UCCE 2k devront être sur la version 12.6 de CCE en raison du déploiement co-résident LD/CUIC

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l'incidence possible des commandes.

 Remarque : la configuration fournie dans ce document a été configurée, renforcée et testée en charge avec le proxy inverse Nginx (OpenResty) déployé sur CentOS 8.0, par rapport à un exemple de déploiement UCCE pour 2 000 utilisateurs. Les informations de référence du profil de performances sont disponibles dans ce document.

Informations générales

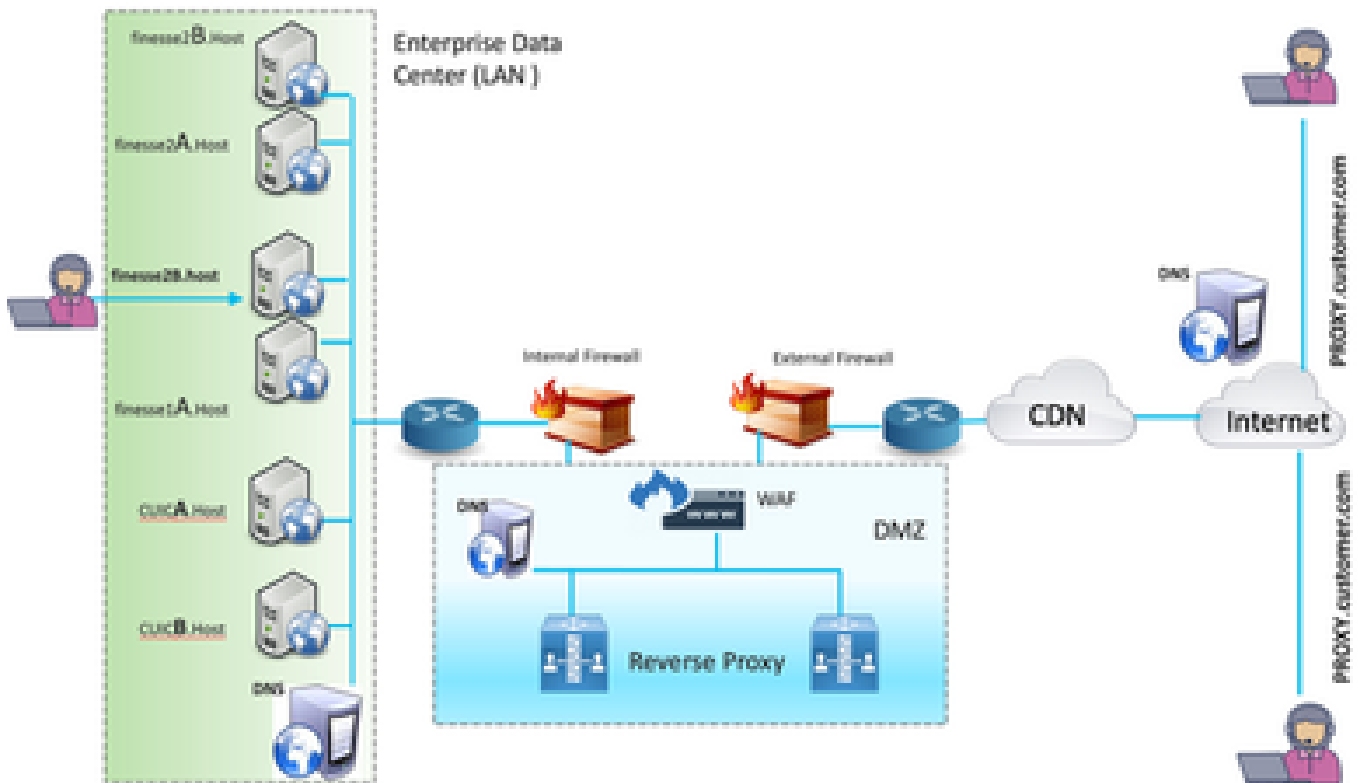
Ce modèle de déploiement est pris en charge pour les solutions UCCE/PCCE et HCS pour UCCE.

Le déploiement d'un proxy inverse est pris en charge (disponible à partir de la version 12.6 ES01) en tant qu'option permettant d'accéder au bureau Cisco Finesse sans se connecter à un VPN. Cette fonctionnalité offre aux agents la flexibilité d'accéder au bureau Finesse depuis n'importe quel endroit via Internet.

Pour activer cette fonctionnalité, une paire de proxies inverses doit être déployée dans la zone démilitarisée (DMZ).

L'accès au support reste inchangé dans les déploiements de proxy inverse. Pour se connecter aux supports, les agents peuvent utiliser la solution Cisco Jabber over Mobile and Remote Access (MRA) ou la fonctionnalité d'agent mobile d'UCCE avec un réseau téléphonique public commuté (RTPC) ou un terminal mobile. Ce schéma montre à quoi ressemblera le déploiement réseau lorsque vous accéderez à deux clusters Finesse et à deux noeuds CUIC via une seule paire de noeuds proxy inverse haute disponibilité (HA).

L'accès simultané des agents sur Internet et des agents qui se connectent à partir du LAN est pris en charge, comme illustré dans cette image.



Remarque : reportez-vous au guide des fonctionnalités pour connaître les critères de sélection de proxy tiers à la place de Nginx pour prendre en charge ce déploiement.

- [Guide des fonctionnalités UCCE 12.6](#) - Fournit une présentation des fonctionnalités, une conception et des [détails de configuration](#) pour la fonctionnalité VPN-Less.
- [Guide de sécurité UCCE 12.6](#) - Fournit des directives de configuration de sécurité pour le déploiement de proxy inverse.

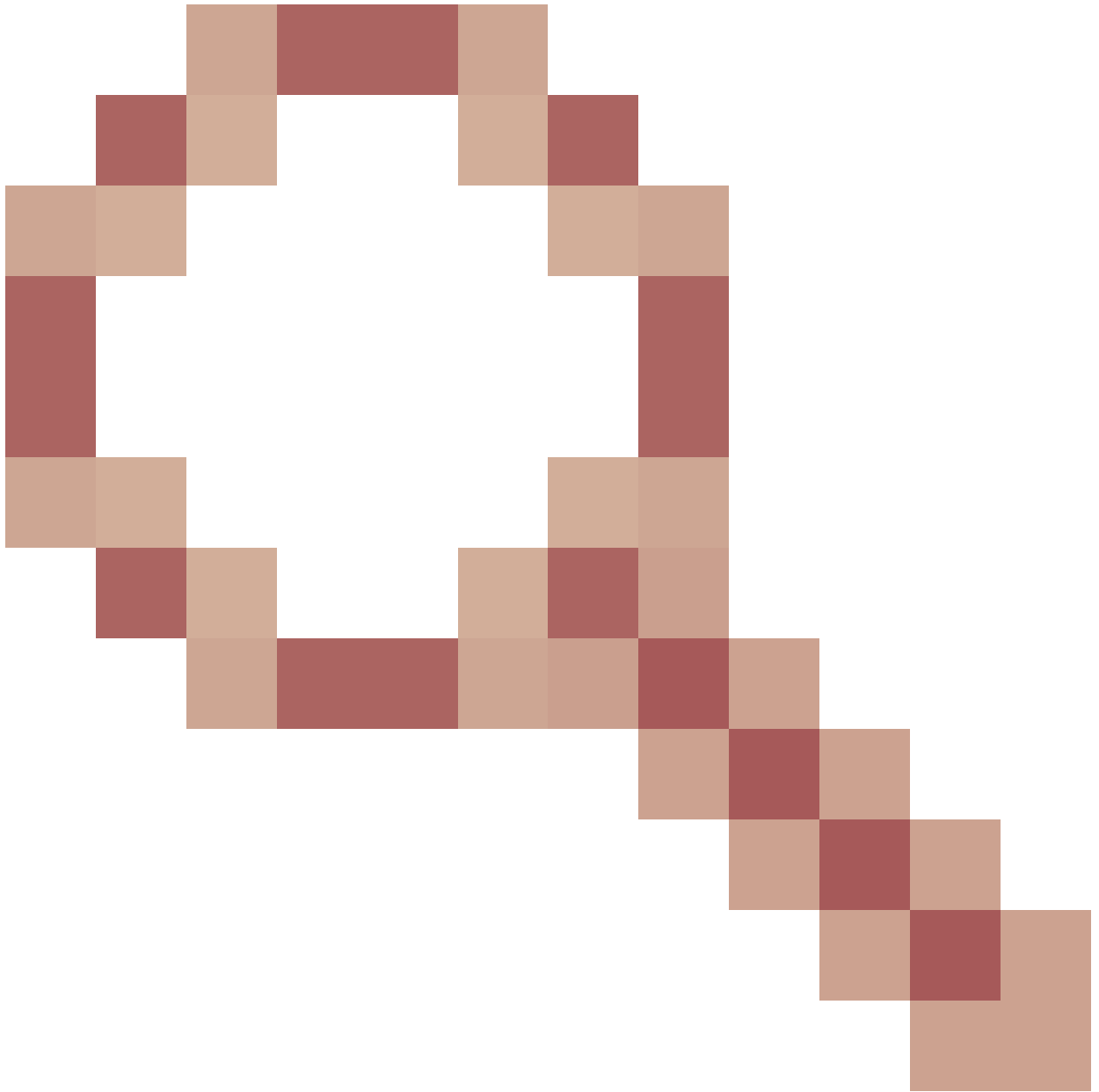
Il est recommandé de consulter la section VPN-Less du guide des fonctionnalités et du guide de sécurité avant de lire ce document.

Changements dans ES03

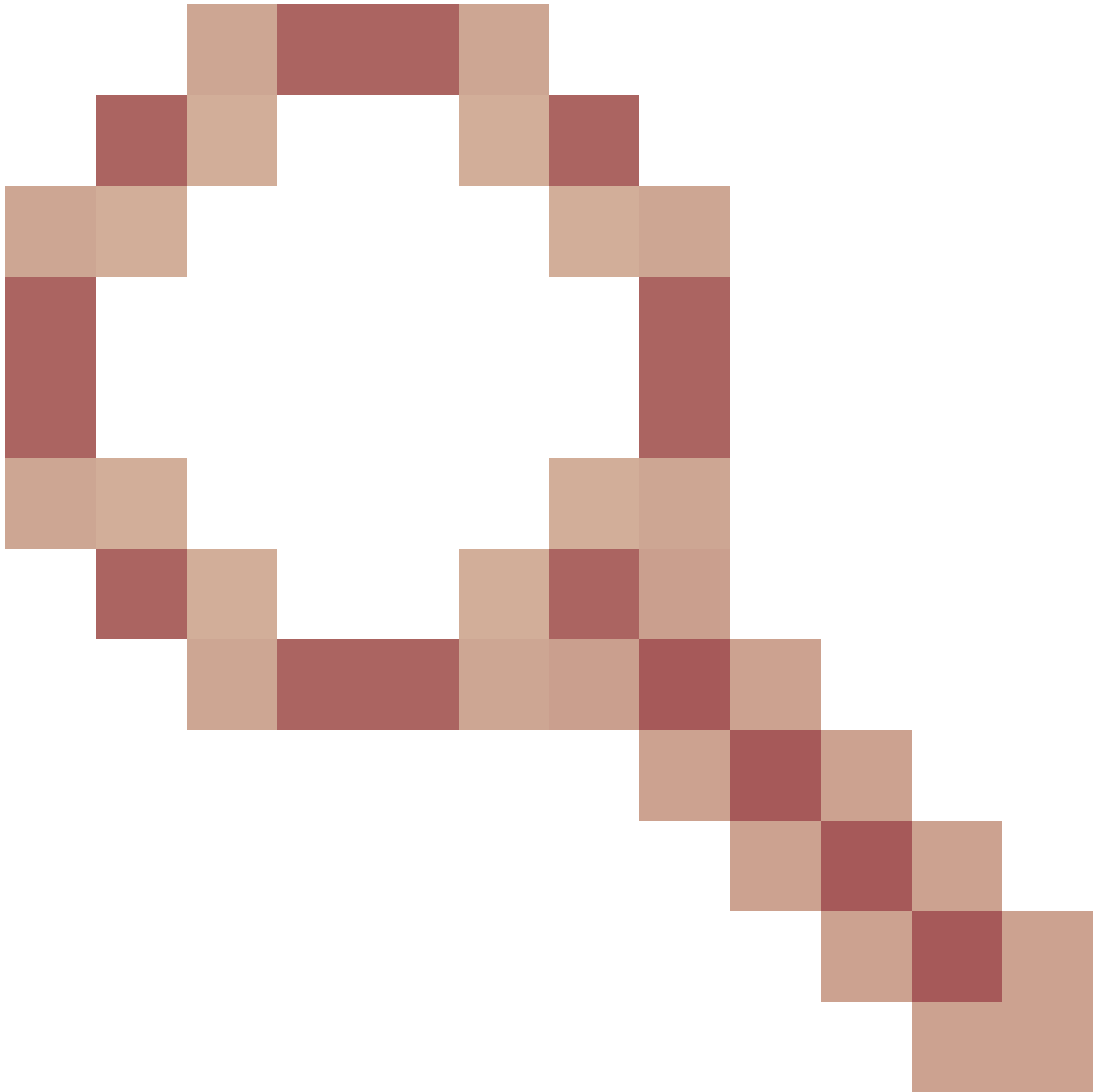
- Nouvelles fonctionnalités
 - Les fonctionnalités de supervision Finesse sont désormais prises en charge via le proxy inverse.
 - Les rapports CUIC RealTime et Historical sont désormais pris en charge via les gadgets Finesse dans un environnement proxy.
 - Authentification pour toutes les demandes / communications - nécessite la prise en charge de Lua
 - Toutes les demandes Finesse / CUIC / IM & Presence (IM&P) sont authentifiées au niveau du proxy avant d'être autorisées à entrer dans le data center.
 - Les connexions WebSocket et Live Data SocketIO sont également restreintes et

autorisées uniquement à partir des clients ayant effectué une demande sécurisée auprès de Finesse.

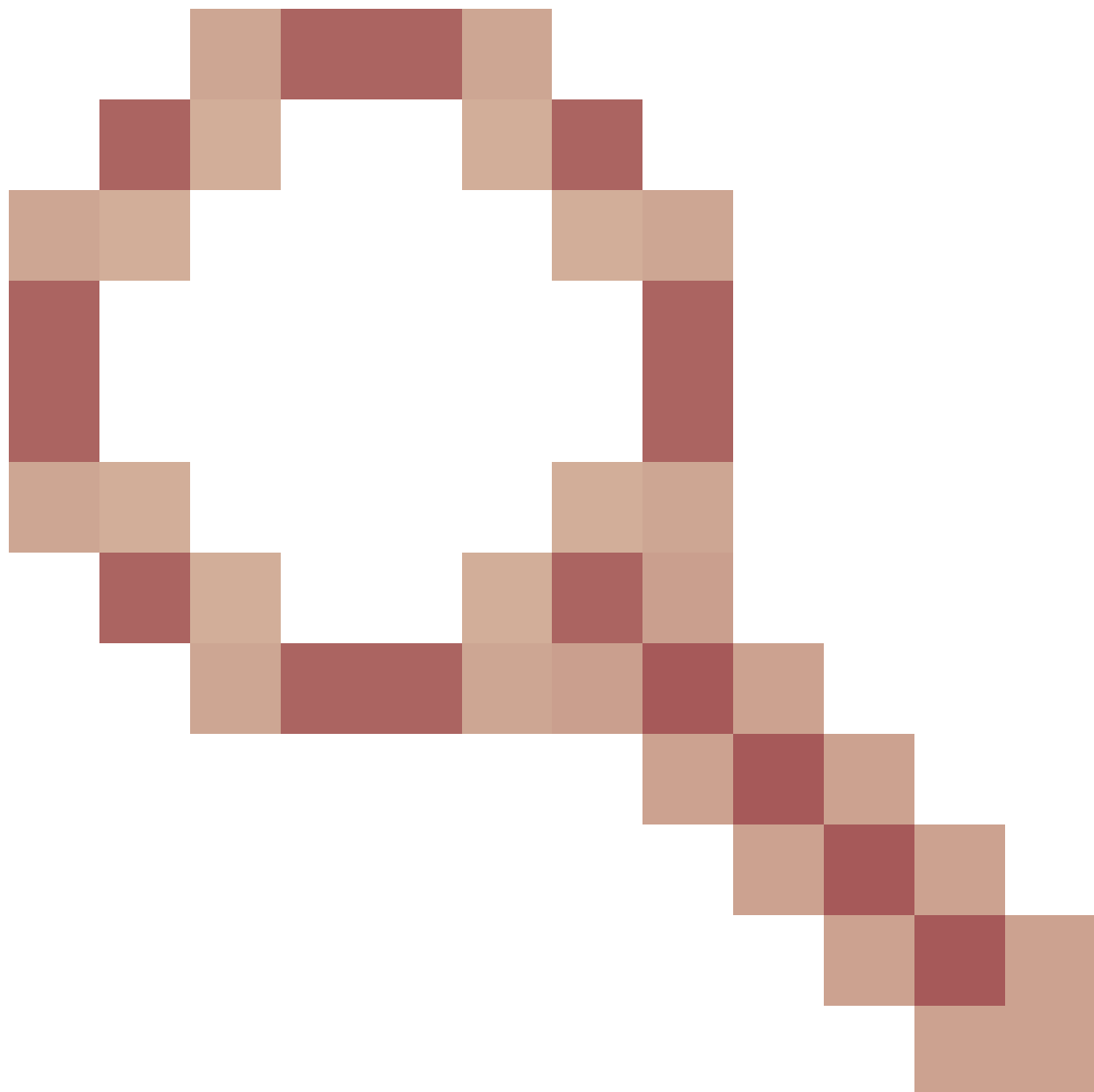
- Détection et journalisation des attaques en force au niveau du proxy, qui peut être utilisé avec Fail2Ban pour bloquer les adresses IP malveillantes.
- Améliorations de la sécurité pour la configuration de proxy inverse - nécessite la prise en charge de Lua
 - Authentification TLS (Mutual Transport Layer Security) entre le proxy inverse et les composants en amont (Finesse/IdS/CUIC/LiveData).
 - Paramètres SeLinux.
 - Activez la vérification de l'approbation SSL (Secure Sockets Layer) mutuelle pour les demandes de serveur proxy et de serveur de composants.
- Sécurité renforcée pour la configuration du proxy afin d'empêcher les attaques par déni de service (DoS) / par déni de service distribué (DDoS) - nécessite la prise en charge de Lua
 - Limites de débit de requête Nginx améliorées pour diverses parties du système.
 - Limites de taux pour les IpTables.
 - Vérification des demandes de ressources statiques avant de demander le serveur de composants en amont.
 - Pages non authentifiées plus légères et pouvant être mises en cache qui n'atteignent pas le serveur de composants en amont.
- Autres fonctionnalités diverses - nécessite la prise en charge de Lua
 - Détection automatique des réponses CORS (Cross-Origin Resource Sharing) fournies par le proxy pour faciliter la configuration automatique et améliorer les performances
- Corrections de défauts relatifs à VPN-Less
 - [CSCwa26057](#)



[CSCwa26057](#)

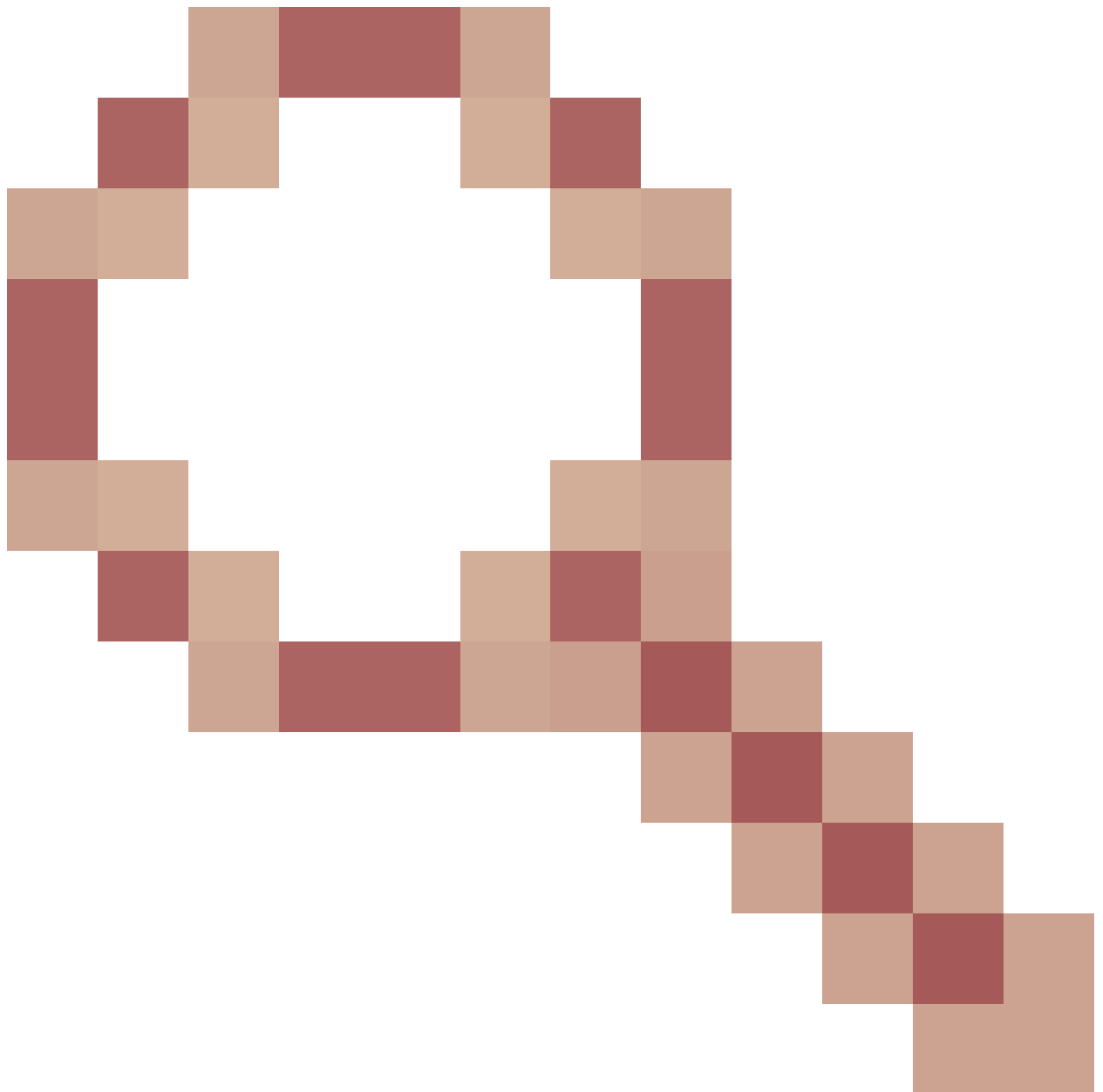


" />- Plusieurs certificats offerts à l'agent lors de la connexion au bureau finesse
◦ [CSCwa24471](#)



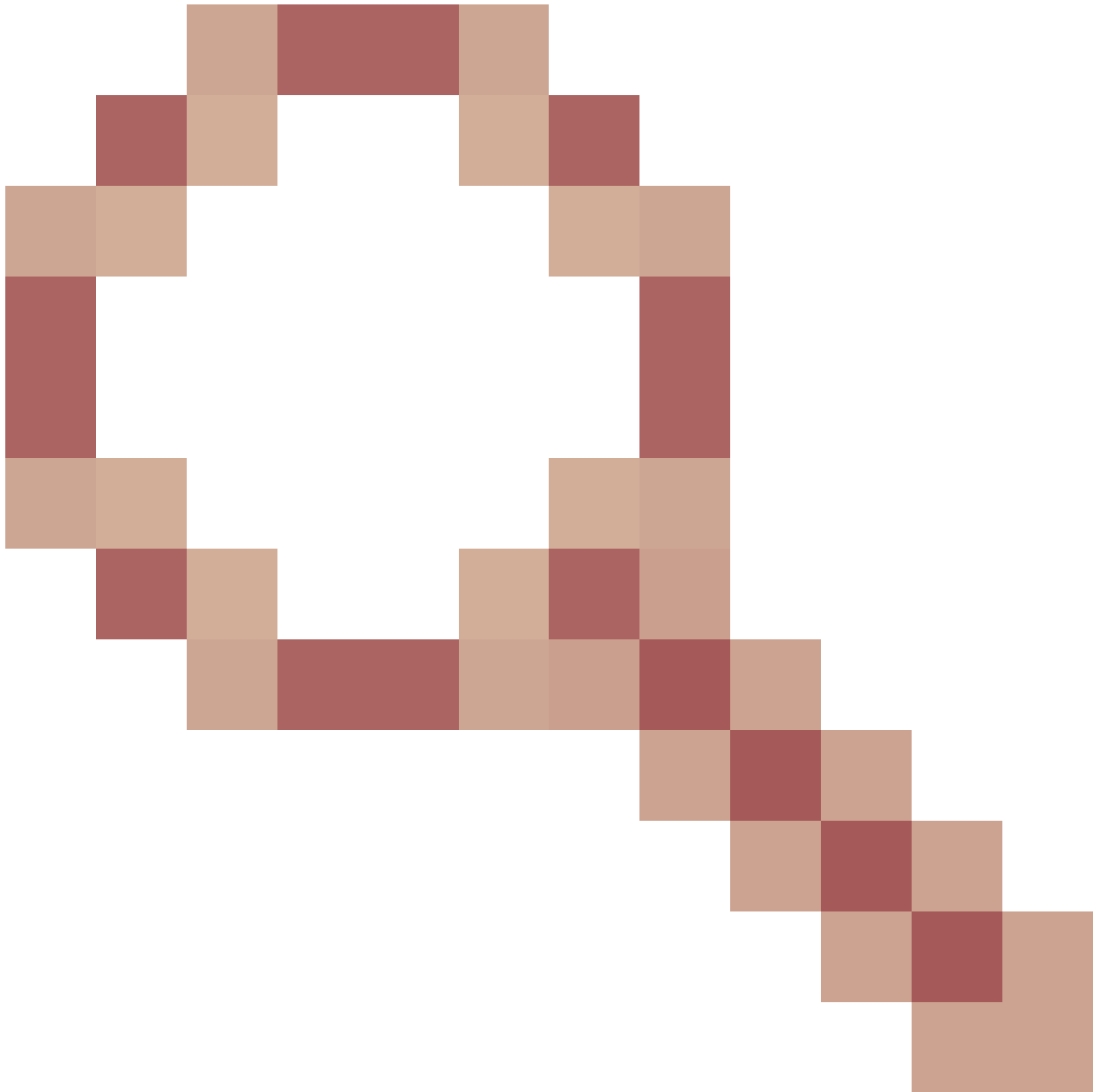
- La page de connexion Finesse n'affiche pas le nom de domaine complet de l'agent SSO

- [CSCwa24519](#)



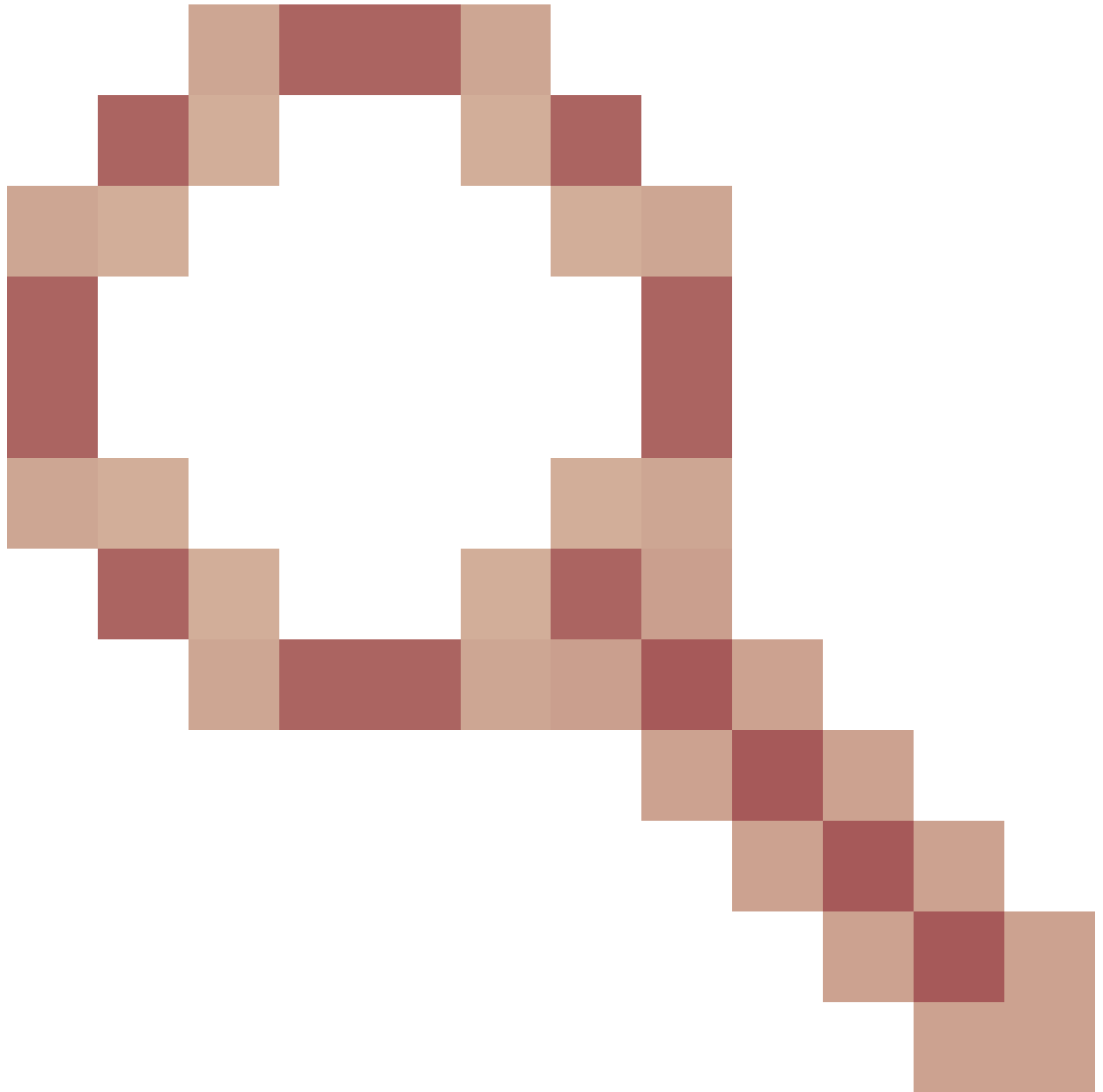
: le service Webproxy ne redémarre pas si le nom d'hôte du proxy inverse ne peut pas être résolu à partir du composant

- [CSCwa23252](#)



: la confiance de finesse du proxy est rompue lorsque la profondeur est supérieure à un pour la chaîne de certificats CA


- [CSCwa46459](#)




log4j vulnérabilité zero day exposée dans webservice

Notes de mise à niveau pour les configurations sans VPN basées sur ES01

- La configuration ES03 nécessite l'installation de Nginx avec prise en charge de Lua.
- Exigences du certificat
 - Cisco Finesse, CUIC et IdS nécessiteront l'ajout du certificat d'hôte Nginx / OpenResty au magasin de confiance Tomcat et un redémarrage effectué, avant que la configuration Nginx ES02 puisse se connecter avec succès au serveur en amont.
 - Les certificats des serveurs en amont Cisco Finesse, CUIC et IdS doivent être configurés sur le serveur Nginx pour utiliser la configuration basée sur ES03.

 Remarque : il est recommandé de supprimer la configuration Nginx basée sur ES01 existante avant d'installer les configurations Nginx ES03.

 Remarque : les scripts de configuration ES03 nécessitent également l'installation de la COP

Authentification

Finesse 12.6 ES03 introduit l'authentification au niveau du proxy. L'authentification est prise en charge pour les déploiements SSO (Single Sign On) et non SSO.

L'authentification est appliquée à toutes les requêtes et à tous les protocoles qui sont acceptés au niveau du proxy avant d'être transférés aux serveurs de composants en amont, où l'authentification appliquée par les serveurs de composants s'effectue également localement. Toutes les authentifications utilisent les identifiants de connexion Finesse communs pour authentifier les demandes.

Les connexions persistantes, telles que les websockets qui reposent sur des protocoles d'application tels que le protocole XMPP (Extensible Messaging and Presence Protocol) pour l'authentification et la post-connexion, sont authentifiées au niveau du proxy en validant l'adresse IP à partir de laquelle une authentification d'application réussie a été effectuée avant d'établir la connexion de socket.

Authentification non SSO

L'authentification non-SSO ne nécessite aucune configuration supplémentaire et fonctionnera avec les scripts de configuration Nginx prêts à l'emploi une fois que les remplacements de scripts requis auront été effectués. L'authentification repose sur le nom d'utilisateur et le mot de passe utilisés pour se connecter à Finesse. L'accès à tous les terminaux sera validé par les services d'authentification Finesse.

La liste des utilisateurs valides est mise en cache localement sur le proxy (met à jour le cache toutes les 15 minutes), qui est utilisé pour valider l'utilisateur dans une requête. Les informations d'identification de l'utilisateur sont validées en transférant la demande à l'URI Finesse configuré, puis le hachage des informations d'identification est mis en cache localement (mis en cache pendant 15 minutes) pour authentifier les nouvelles demandes localement. En cas de modification du nom d'utilisateur ou du mot de passe, celui-ci n'entrera en vigueur qu'au bout de 15 minutes.

Authentification SSO

L'authentification SSO nécessite que l'administrateur configure la clé de cryptage du jeton IdS sur le serveur Nginx dans le fichier de configuration. La clé de chiffrement de jeton IdS peut être obtenue à partir du serveur IdS avec la commande `show ids secret CLI`. Elles doivent être configurées dans le cadre de l'un des remplacements `#Must-change` que l'administrateur doit effectuer dans les scripts avant que l'authentification SSO puisse fonctionner.

Reportez-vous au guide de l'utilisateur SSO pour les configurations SAML IdS à effectuer pour que la résolution de proxy fonctionne pour IdS.

Une fois l'authentification SSO configurée, une paire valide de jetons peut être utilisée pour accéder à l'un des points d'extrémité du système. La configuration de proxy valide les informations

d'identification en interceptant les demandes de récupération de jeton envoyées aux ID ou en décryptant les jetons valides et en les mettant ensuite en cache localement pour d'autres validations.

Authentification pour les connexions WebSocket

Les connexions Websocket ne peuvent pas être authentifiées avec l'en-tête d'autorisation standard, car les en-têtes personnalisés ne sont pas pris en charge par les implémentations de websocket natives dans le navigateur. Les protocoles d'authentification au niveau de l'application, dans lesquels les informations d'authentification contenues dans la charge utile n'empêchent pas l'établissement de la connexion par interface Web, et donc les entités malveillantes peuvent provoquer des attaques DOS ou DDOS simplement en créant d'innombrables connexions pour submerger le système.

Afin d'atténuer cette possibilité, les configurations de proxy inverse nginx fournies ont des contrôles spécifiques pour permettre aux connexions de connexion Web d'être acceptées UNIQUEMENT à partir des adresses IP qui ont réussi à faire une requête REST authentifiée avant l'établissement de la connexion de connexion Web. Cela signifie que les clients qui tentent de créer des connexions de connexion Web, avant qu'une requête REST soit émise, recevront maintenant une erreur d'échec d'autorisation et n'est pas un scénario d'utilisation pris en charge.

Prévention des attaques par force brute

Les scripts d'authentification Finesse 12.6 ES02 empêchent activement les attaques en force qui peuvent être utilisées pour deviner le mot de passe utilisateur. Pour ce faire, il bloque l'adresse IP utilisée pour accéder au service, après un certain nombre de tentatives infructueuses en peu de temps. Ces demandes seront rejetées par une erreur du client 418. Les détails des adresses IP bloquées sont accessibles à partir des fichiers `<nginx-install-directory>/logs/blocking.log` et `<nginx-install-directory>/logs/error.log`.

Le nombre de demandes ayant échoué, l'intervalle de temps et la durée de blocage sont configurables. Les configurations sont présentes dans le fichier `<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;
```

```
}
```

Journalisation

Pour rechercher les adresses IP bloquées, exécutez les commandes suivantes à partir du répertoire `<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"
```

Il est recommandé aux clients d'intégrer Fail2ban ou similaire pour ajouter l'interdiction aux règles IPtable/pare-feu.

Installation et configuration de Fail2ban

Fail2ban analyse les fichiers journaux et interdit les adresses IP qui présentent des signes malveillants : trop d'échecs de mot de passe, recherche d'exploits, etc. En général, Fail2Ban est ensuite utilisé pour mettre à jour les règles de pare-feu afin de rejeter les adresses IP pour une durée spécifiée, bien que toute autre action arbitraire (par exemple l'envoi d'un e-mail) puisse également être configurée. Pour plus d'informations, visitez le site <https://www.fail2ban.org/>.

Fail2ban peut être configuré pour surveiller le `blocking.log` afin d'identifier les adresses IP qui sont bloquées par Nginx lors de la détection d'attaques bruteforce, et les interdire pour une durée configurable. Les étapes d'installation et de configuration de fail2ban sur un proxy inverse CentOS sont les suivantes :

1. Installez Fail2ban en utilisant yum.

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

2. Créer une prison locale.

Les configurations d'emplacement permettent à l'administrateur de configurer diverses propriétés, telles que les ports qui doivent être interdits d'accès par toute adresse IP bloquée, la durée pendant laquelle l'adresse IP reste bloquée, la configuration de filtre utilisée pour identifier l'adresse IP bloquée à partir du fichier journal surveillé, etc. Les étapes permettant d'ajouter une configuration personnalisée pour interdire l'accès aux serveurs en amont aux adresses IP bloquées sont les suivantes :

2.1. Accédez au répertoire d'installation de Fail2ban (dans cet exemple /etc/fail2ban)

```
cd /etc/fail2ban
```

2.2. Copier le fichier prison.conf dans le fichier prison.local pour isoler les modifications locales.

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

2.3. Ajoutez ces configurations de prison à la fin du fichier IN.local, et remplacez les ports dans le modèle par les ports réels. Mettez à jour les configurations d'heure de bande selon les besoins.

```
# 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. Configurez un filtre.

Un filtre indique à Fail2ban ce qu'il faut rechercher dans les journaux pour identifier l'hôte à bannir. La procédure de création d'un filtre est la suivante :

3.1. Créez filter.d/finesseban.conf.

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

3.2. Ajoutez ces lignes dans le fichier filter.d/finesseban.conf.

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

4. Démarrez Fail2ban.

Exécutez cette commande pour démarrer fail2ban.

```
fail2ban-client start
```

Ouvrez les fichiers journaux fail2ban et vérifiez qu'il n'y a aucune erreur. Par défaut, les journaux pour fail2ban vont dans le fichier /var/log/fail2ban.log.

Valider les URL de ressources statiques

Tous les points d'extrémité valides accessibles sans authentification sont activement suivis dans les scripts ES03.

Les requêtes vers ces chemins non authentifiés sont activement rejetées, si une URI non valide est demandée, sans envoyer ces requêtes au serveur en amont.


Mise en cache des en-têtes CORS

Lorsque la première demande d'options aboutit, les en-têtes de réponse access-control-allow-headers, access-control-allow-origin, access-control-allow-methods, access-control-expose-headers, et access-control-allow-credentials sont mis en cache au niveau du proxy pendant cinq minutes. Ces en-têtes sont mis en cache pour chaque serveur en amont respectif.

Configurer

Ce document décrit la configuration de Nginx comme le proxy inverse à utiliser pour activer l'accès Finesse VPN-Less. Le composant de la solution UCCE, le proxy et les versions du système d'exploitation utilisés pour vérifier les instructions fournies sont fournis. Les instructions pertinentes doivent être adaptées au système d'exploitation/proxy de votre choix.

- Version Nginx utilisée - OpenResty 1.19.9.1
- Système d'exploitation utilisé pour la configuration - CentOS 8.0

 Remarque : la configuration Nginx décrite peut être téléchargée à partir de la [page de téléchargement du logiciel Finesse version 12.6\(1\)ES3](#).

Configurer les composants de la solution pour un accès VPN moindre

Après avoir configuré le proxy, configurez les composants de la solution (Finesse/ CUIC / IdS) pour VPN Less access avec le nom d'hôte et l'IP planifiés du proxy/des services utilisés pour accéder à la solution avec ces commandes.

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

Les détails de ces commandes sont disponibles dans le [Guide des fonctionnalités UCCE 12.6](#) et doivent être consultés avant d'utiliser ce document.


Installer OpenResty en tant que proxy inverse dans DMZ

Cette section détaille les étapes d'installation du proxy basé sur OpenResty. Le proxy inverse est généralement configuré en tant que périphérique dédié dans la zone démilitarisée du réseau (DMZ), comme indiqué dans le schéma de déploiement mentionné précédemment.

1. Installez le système d'exploitation de votre choix avec la spécification matérielle requise. Les modifications des paramètres du noyau et IPv4 peuvent varier en fonction du système d'exploitation sélectionné. Il est conseillé aux utilisateurs de vérifier à nouveau ces aspects si la version du système d'exploitation choisie est différente.
2. Configurez deux interfaces réseau. Une interface sera nécessaire pour l'accès public à partir des clients Internet et une autre pour communiquer avec les serveurs du réseau interne.
3. Installez [OpenResty](#).

Toutes les saveurs de Nginx peuvent être utilisées à cette fin, à condition qu'elles soient basées sur Nginx 1.19+ et prennent en charge Lua :

- Nginx Plus
- Nginx Open Source (Nginx open source devra être compilé avec les modules Lua basés sur OpenResty pour être utilisé)
- OpenResty
- Extras GetPageSpeed

 Remarque : la configuration fournie a été testée avec OpenResty 1.19 et devrait fonctionner avec d'autres distributions avec seulement des mises à jour mineures, le cas échéant.

Installation de OpenResty

1. Installez OpenResty. Voir [Packages Linux OpenResty](#). Dans le cadre de l'installation d'OpenResty, Nginx sera installé à cet emplacement et ajoutera le chemin d'accès d'OpenResty à la variable PATH en ajoutant dans le fichier ~/.bashrc.

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


2. Démarrer / arrêter Nginx.

- Afin de démarrer Nginx, entrez `openresty`.
- Afin d'arrêter Nginx, entrez `openresty -s stop`.

Configurer Nginx

La configuration est expliquée pour une installation de Nginx basée sur OpenResty. Les répertoires par défaut pour OpenResty sont :

- <nginx-install-directory> = /usr/local/openresty/nginx
 - <Openresty-install-directory> = /usr/local/openresty
1. Téléchargez et extrayez le fichier de la [page de téléchargement du logiciel Finesse version 12.6\(1\)ES03](#) (12.6-ES03-reverse-proxy-config.zip) qui contient la configuration de proxy inverse pour Nginx.
 2. Copiez nginx.conf, nginx/conf.d/, et nginx/html/ du répertoire de configuration proxy inverse extrait vers <nginx-install-directory>/conf, <nginx-install-directory>/conf/conf.d/, et <nginx-install-directory>/html/ respectivement.
 3. Copiez le répertoire nginx/lua à partir du répertoire de configuration de proxy inverse extrait dans le répertoire <nginx-install-directory>.
 4. Copiez le contenu de lualib dans <Openresty-install-directory>/lualib/resty.
 5. Configurez la rotation du journal nginx en copiant le fichier nginx/logrotate/saproxy dans le dossier <nginx-install-directory>/logrotate/. Modifiez le contenu du fichier pour pointer vers les répertoires de journaux corrects si les valeurs par défaut de Nginx ne sont pas utilisées.
 6. Nginx doit être exécuté avec un compte de service non privilégié dédié, qui doit être verrouillé et avoir un shell non valide (ou le cas échéant pour le système d'exploitation choisi).
 7. Recherchez la chaîne "Must-change" dans les fichiers sous les dossiers extraits nommés html et conf.d et remplacez les valeurs indiquées par les entrées appropriées.
 8. Assurez-vous que tous les remplacements obligatoires sont effectués, qui sont décrits avec les commentaires Must-change dans les fichiers de configuration.
 9. Assurez-vous que les répertoires de cache configurés pour CUIIC et Finesse sont créés sous <nginx-install-directory>/cache avec ces répertoires temporaires.
 - <nginx-install-directory>/cache/client_temp
 - <nginx-install-directory>/cache/proxy_temp

 Remarque : la configuration fournie est destinée à un exemple de déploiement 2000 et doit être développée de manière appropriée pour un déploiement plus important.

Configuration du cache Nginx

Par défaut, les chemins du cache proxy sont stockés dans le système de fichiers. Nous vous recommandons de les remplacer par des lecteurs en mémoire en créant un emplacement de cache dans tmpfs, comme indiqué ici.

1. Créez des répertoires pour les différents chemins du cache proxy sous /home.

Par exemple, ces répertoires doivent être créés pour le Finesse principal. Les mêmes étapes doivent être suivies pour les serveurs Finesse et CUIC secondaires.

```
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
```




Remarque : augmentez les caches client et proxy_temp de 1 Go pour chaque nouveau cluster Finesse ajouté à la configuration.

2. Montez les nouveaux points de montage à l'aide de la commande `mount -av`.
3. Vérifiez que le système de fichiers a monté les nouveaux points de montage à l'aide de la `df -h` commande.
4. Modifiez les emplacements `proxy_cache_path` dans les fichiers de configuration du cache Finesse et CUIC.

Par exemple, pour modifier les chemins d'accès pour le principal Finesse, allez à `<nginx-install-directory>conf/conf.d/finesse/caches` et changez l'emplacement de cache existant `/usr/local/openresty/nginx/cache/finesse25/` pour l'emplacement du système de fichiers nouvellement créé `/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. Suivez les mêmes étapes pour les serveurs secondaire Finesse et CUIC.


 Remarque : assurez-vous que la somme de toutes les tailles de lecteur tmpfs créées dans toutes les étapes précédentes est ajoutée à la taille de mémoire finale pour le déploiement, car ces lecteurs sont des blocs de mémoire configurés pour ressembler à des disques pour l'application et consommer autant d'espace mémoire.

Configurer les certificats SSL

Utiliser des certificats auto-signés - Tester les déploiements

Les certificats auto-signés ne doivent être utilisés que jusqu'à ce que le proxy inverse soit prêt à être déployé en production. Sur un déploiement en production, utilisez uniquement un certificat signé par une autorité de certification (CA).

1. Générez des certificats Nginx pour le contenu du dossier SSL. Avant de générer des certificats, vous devez créer un dossier appelé ssl sous /usr/local/openresty/nginx. Vous devez générer deux certificats à l'aide de ces commandes (un pour <reverseproxy_primary_fqdn> et un autre pour <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` (passez le nom d'hôte en tant que : <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` (passez le nom d'hôte : <reverseproxy_secondary_fqdn>)
 - c. Assurez-vous que le chemin d'accès du certificat est /usr/local/openresty/nginx/ssl/nginx.crt et /usr/local/openresty/nginx/ssl/nginxnode2.crt, puisque ceux-ci sont déjà configurés dans les fichiers de configuration Finesse Nginx.
2. Modifiez l'autorisation de la clé privée 400 (r-----).
3. Configurez le pare-feu/[iptables](#) sur le proxy inverse pour permettre à la communication du pare-feu de correspondre aux ports sur lesquels le serveur Nginx a été configuré pour écouter.
4. Ajoutez l'adresse IP et le nom d'hôte de Finesse, IdS et CUIC sous l'entrée /etc/hosts sur le serveur proxy inverse.
5. Reportez-vous au guide des fonctionnalités de la solution pour connaître les configurations à effectuer sur les serveurs composants pour configurer l'hôte Nginx en tant que proxy inverse.

 Remarque : la configuration fournie est destinée à un exemple de déploiement 2000 et doit être développée de manière appropriée pour un déploiement plus important.

Utiliser le certificat signé par l'autorité de certification - Déploiements en production

Un certificat signé par une autorité de certification peut être installé sur le proxy inverse en procédant comme suit :

1. Générez la demande de signature de certificat (CSR).

Afin de générer le CSR et la clé privée, `openssl req -new -newkey rsa:4096 -keyout nginx.key -out nginx.csr` entrez après vous être connecté au proxy. Suivez l'invite et fournissez les détails. Ceci génère le CSR (nginx.csr dans l'exemple) et la clé privée RSA (nginx.key dans l'exemple) de 4096 bits de puissance.

Exemple :

```
[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.companymain.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
```

Notez la phrase de passe PEM, car elle sera utilisée pour déchiffrer la clé privée pendant le déploiement.

2. Obtenez le certificat signé auprès de l'autorité de certification.

Envoyez le CSR à l'autorité de certification et obtenez le certificat signé.

Remarque : si le certificat reçu de l'autorité de certification n'est pas une chaîne de certificats contenant tous les certificats respectifs, composez tous les certificats pertinents dans un fichier de chaîne de certificats unique.

3. Déployez le certificat et la clé.

Déchiffrez la clé générée précédemment dans le cadre de la première étape avec `openssl rsa -in nginx.key -out nginx_decrypted.key` la commande. Placez le certificat signé par l'autorité de certification et la clé déchiffrée dans le dossier `/usr/local/openresty/nginx/ssl` de l'ordinateur proxy inverse. Mettez à jour/ajoutez des configurations SSL associées au certificat dans les configurations Nginx du fichier de configuration

```
/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. Configurez les autorisations pour les certificats.

```
chmod 400 /usr/local/openresty/nginx/ssl/ca_signed_cert.crt
```

Entrez `and chmod 400 /usr/local/openresty/nginx/ssl/nginx_decrypted.key`, afin que le certificat dispose d'une autorisation en lecture seule et soit limité au propriétaire.

5. Rechargez Nginx.

Utiliser le paramètre Diffie-Hellman personnalisé

Créez un paramètre Diffie-Hellman personnalisé à l'aide des commandes suivantes :

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

Modifiez la configuration du serveur pour utiliser les nouveaux paramètres du fichier `/usr/local/openresty/nginx/conf/conf.d/ssl/ssl.conf`:

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

Vérifier que l'agrafage OCSP est activé - Contrôle de révocation de certificat

Remarque : pour activer cette fonction, le serveur doit utiliser un certificat signé par une autorité de certification et avoir accès à l'autorité de certification qui a signé le certificat.

Ajoutez/mettez à jour cette configuration dans le répertoire `file/usr/local/openresty/nginx/conf/conf.d/ssl/ssl.conf`:

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

Configuration Nginx

Le fichier de configuration Nginx par défaut (`/usr/local/openresty/nginx/conf/nginx.conf`) doit être modifié pour contenir ces entrées afin d'appliquer la sécurité et d'assurer les performances. Ce contenu doit être utilisé pour modifier le fichier de configuration par défaut créé par l'installation 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 UnauthenticatedDesktopResourceManager = require("unauthenticated_desktopresources_manager")
    local UnauthenticatedResourceManager = 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")
        UnauthenticatedDesktopResourceManager.getDesktopResources("saproxy.cisco.com", "https://sa")
        UnauthenticatedResourceManager.getThirdPartyGadgetResources("saproxy.cisco.com", "https://sa")
    end
}

include conf.d/*.conf;

sendfile        on;

tcp_nopush      on;

server_names_hash_bucket_size 512;

```

Configurer le port proxy inverse

Par défaut, la configuration Nginx écoute les requêtes Finesse sur le port 8445. À la fois, un seul port peut être activé à partir d'un proxy inverse pour prendre en charge les requêtes Finesse, par exemple 8445. Si le port 443 doit être pris en charge, modifiez le fichier <nginx-install-directory>conf/conf.d/finesse.conf afin d'activer l'écoute sur 443 et de désactiver l'écoute sur 8445.

Configurer l'authentification TLS mutuelle entre le proxy inverse et les composants en amont

L'authentification de certificat SSL client pour les connexions à partir d'hôtes proxy inverses peut être activée sur les composants en amont CCBU CUIC/Finesse/IdS/Livedata via la nouvelle option CLI CVOS qui est

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

Par défaut, cette option est désactivée et doit être explicitement activée par l'administrateur en exécutant l'interface de ligne de commande sur chaque serveur en amont indépendamment. Une fois cette option activée, le service proxy Web Cisco exécuté sur l'hôte en amont commence à authentifier les certificats clients dans la connexion TLS pour les connexions provenant d'hôtes proxy inverse approuvés ajoutés dans le cadre de l'interface de ligne de commande `utils system reverse-proxy allowed-hosts add <proxy-host>`.

Vous trouverez ci-dessous le bloc de configuration pour le même dans les fichiers de configuration proxy, à savoir `ssl.conf` et `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;
```

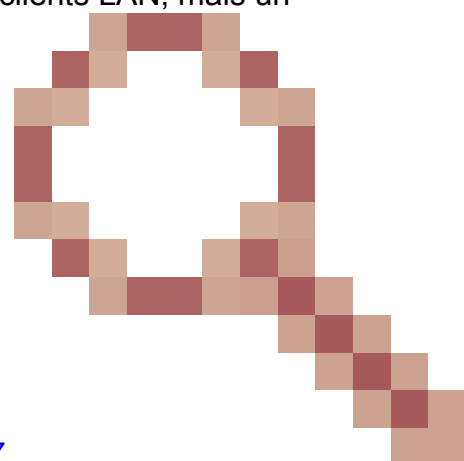
Le certificat SSL utilisé pour le trafic sortant (du proxy vers l'amont) peut être identique au certificat SSL configuré pour le trafic entrant (connecteur SSL pour les blocs de serveurs de composants). Si le certificat auto-signé est utilisé comme `proxy_ssl_certificate`, il doit être téléchargé vers les composants en amont (Finesse/IdS/CUIC/Livedata) pour créer un magasin de confiance afin qu'il soit authentifié avec succès.

La validation du certificat du serveur en amont par proxy inverse est facultative et désactivée par défaut. Si vous souhaitez obtenir une authentification mutuelle TLS complète entre le proxy inverse et les hôtes en amont, la configuration ci-dessous doit être sans commentaire à partir des fichiers `ssl.conf` et `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
```

Avertissements relatifs à la configuration de l'authentification TLS mutuelle :

- Une fois cette fonctionnalité activée sur les composants CCBU, le certificat client sera demandé aux clients LAN également pendant la connexion TLS. Si des certificats client/personnels sont installés sur des ordinateurs clients, les navigateurs peuvent choisir d'afficher une fenêtre contextuelle demandant à l'utilisateur final de choisir le certificat approprié pour l'authentification client. Bien que peu importe le certificat choisi par l'utilisateur final ou que vous appuyiez sur la touche Annuler pour les requêtes contextuelles, l'authentification par certificat client n'est pas appliquée aux clients LAN, mais un



changement d'expérience s'opère. Voir CDET [CSCwa26057](#) pour plus de détails.

- Le service webproxy des composants en amont ne s'active pas si un hôte proxy est ajouté à la liste d'accès autorisée, ce qui n'est pas résoluble par le service webproxy. Assurez-vous que les hôtes proxy inverses ajoutés à la liste autorisée peuvent être résolus à partir du composant en amont via la recherche DNS.

Effacer le cache

La

```
/clearCache.sh
```

commande permet d'effacer le cache proxy inverse.

Lignes directrices types

Cette section décrit brièvement les directives standard qui doivent être suivies lorsque vous configurez Nginx en tant que serveur proxy.

Ces directives sont dérivées du [Centre pour la sécurité Internet](#). Pour plus de détails sur chaque recommandation, reportez-vous à la même.

1. Il est toujours recommandé d'utiliser la dernière version stable d'OpenResty et d'OpenSSL.
2. Il est conseillé d'installer Nginx sur un support de disque séparé.
3. L'ID de processus Nginx doit appartenir à l'utilisateur racine (ou, le cas échéant, au système d'exploitation choisi) et doit avoir une autorisation 644 (rw-----) ou plus stricte.
4. Nginx doit bloquer les requêtes pour les hôtes inconnus. Vérifiez que chaque bloc de serveur contient la directive `server_name` explicitement définie. Afin de vérifier, recherchez tous les blocs de serveur dans le répertoire `nginx.conf` et `nginx/conf.d` et vérifiez que tous les blocs de serveur contiennent le `nom_serveur`.
5. Nginx doit écouter uniquement sur les ports autorisés. Recherchez tous les blocs de serveur dans le répertoire `nginx.conf` et `nginx/conf.d` et vérifiez s'il y a écoute des directives afin de vérifier que seuls les ports autorisés sont ouverts pour l'écoute.
6. Étant donné que Cisco Finesse ne prend pas en charge le protocole HTTP, est-il recommandé de bloquer également le port HTTP du serveur proxy ?
7. Le protocole SSL de Nginx doit être TLS 1.2. La prise en charge des protocoles SSL hérités doit être supprimée. Il doit également désactiver les chiffrements SSL faibles.
8. Il est conseillé d'envoyer les journaux d'accès et d'erreur Nginx au serveur syslog distant.
9. Il est conseillé d'installer le module `mod_security` qui fonctionne comme un pare-feu d'application web. Consultez le [manuel ModSecurity](#) pour plus d'informations. Notez que la charge Nginx n'a pas été vérifiée dans le module `mod_security` en place.

Configuration du fichier de mappage

Le déploiement du proxy inverse du bureau Finesse nécessite un fichier de mappage pour configurer la liste des combinaisons nom d'hôte/port visibles en externe et leur mappage aux noms de serveurs et aux ports réels utilisés par les serveurs Finesse, IdS et CUIC. Ce fichier de mappage, qui est configuré sur les serveurs internes, est la configuration clé qui permet aux clients connectés via Internet d'être redirigés vers les hôtes et les ports requis qui sont utilisés sur Internet.

Le fichier de mappage doit être déployé sur un serveur Web accessible aux serveurs composants et son URI doit être configuré pour que le déploiement fonctionne. Il est recommandé de configurer le fichier de mappage à l'aide d'un serveur Web dédié disponible sur le réseau. Si un tel

serveur n'est pas disponible, le proxy inverse peut être utilisé à la place, ce qui nécessite que le proxy soit accessible à partir du réseau et présente également un risque d'exposition des informations aux clients externes qui peuvent effectuer un accès non autorisé dans la DMZ. La section suivante décrit en détail comment cela peut être réalisé.

Reportez-vous au guide des fonctionnalités pour connaître les étapes exactes de configuration de l'URI du fichier de mappage sur tous les serveurs de composants et pour plus de détails sur la création des données du fichier de mappage.

Utiliser le proxy inverse comme serveur de fichiers de mappage

Ces étapes ne sont requises que si le proxy inverse est également utilisé comme hôte du fichier de mappage proxy.

1. Configurez le nom d'hôte proxy inverse dans le contrôleur de domaine utilisé par les hôtes Finesse/CUIC et IdS de sorte que son adresse IP puisse être résolue.
2. Téléchargez les certificats Nginx signés générés sur les deux noeuds sous tomcat-trust de cmplatform et redémarrez le serveur.
3. Mettez à jour les valeurs Must-change dans <NGINX_HOME>/html/proxymap.txt.
4. Rechargez les configurations Nginx à l'aide de la `nginx -s reload` commande.
5. Validez que le fichier de configuration est accessible à partir d'un autre hôte réseau à l'aide de la `curl` commande.

Durcissement du noyau CentOS 8

Si le système d'exploitation choisi est CentOS 8, il est recommandé d'effectuer le durcissement/réglage du noyau en utilisant ces configurations sysctl pour les installations qui utilisent un serveur dédié pour héberger le 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
```


Un redémarrage est recommandé après avoir effectué les modifications recommandées.

IPtables Durcissement

IPtables est une application qui permet à un administrateur système de configurer les tables, chaînes et règles IPv4 et IPv6 fournies par le pare-feu du noyau Linux.

Ces règles IPtables sont configurées pour sécuriser l'application proxy contre les attaques en force en limitant l'accès dans le pare-feu du noyau Linux.

Les commentaires de la configuration indiquent quel service est limité en débit à l'aide des règles.

 Remarque : si les administrateurs utilisent un port différent ou étendent l'accès à plusieurs serveurs utilisant les mêmes ports, le dimensionnement approprié doit être effectué pour ces ports en fonction de ces numéros.

```
## 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 opened 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 --hashlimit
-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-ma
-A INPUT -p tcp -m tcp --dport 443 --tcp-flags SYN SYN -m connlimit --connlimit-above 8 --connlimit-ma
-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-
-A INPUT -p tcp -m tcp --dport 15280 --tcp-flags SYN SYN -m connlimit --connlimit-above 30 --connlimit-
-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-
-A INPUT -p tcp -m tcp --dport 25280 --tcp-flags SYN SYN -m connlimit --connlimit-above 30 --connlimit-
-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-max 6 --connlimit-logger LOG
-A INPUT -p tcp -m tcp --dport 8444 --tcp-flags SYN SYN -m connlimit --connlimit-above 6 --connlimit-max 6 --connlimit-logger LOG
-A INPUT -p tcp -m tcp --dport 8444 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 2/sec --hashlimit-mode srcip --hashlimit-logger LOG
-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-max 6 --connlimit-logger LOG
-A INPUT -p tcp -m tcp --dport 8447 --tcp-flags SYN SYN -m connlimit --connlimit-above 6 --connlimit-max 6 --connlimit-logger LOG
-A INPUT -p tcp -m tcp --dport 8447 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 2/sec --hashlimit-mode srcip --hashlimit-logger LOG
-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-max 10 --connlimit-logger LOG
-A INPUT -p tcp -m tcp --dport 12005 --tcp-flags SYN SYN -m connlimit --connlimit-above 10 --connlimit-max 10 --connlimit-logger LOG
-A INPUT -p tcp -m tcp --dport 12005 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 6/sec --hashlimit-mode srcip --hashlimit-logger LOG
-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-max 10 --connlimit-logger LOG
-A INPUT -p tcp -m tcp --dport 12008 --tcp-flags SYN SYN -m connlimit --connlimit-above 10 --connlimit-max 10 --connlimit-logger LOG
-A INPUT -p tcp -m tcp --dport 12008 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 6/sec --hashlimit-mode srcip --hashlimit-logger LOG
-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

```

Ces règles peuvent être appliquées directement en éditant le fichier `/etc/sysconfig/iptables` manuellement ou en enregistrant la configuration dans un fichier tel que `iptables.conf` et en exécutant `cat iptables.conf >>/etc/sysconfig/iptables` pour appliquer les règles.

Un redémarrage du service IPtables est nécessaire après l'application des règles. Entrez `systemctl restart iptables` afin de redémarrer le service IPtables.

Restreindre les connexions client

Outre la configuration précédente des tables IP, il est recommandé aux installations qui connaissent la plage d'adresses des clients qui utilisent le proxy d'utiliser ces connaissances pour sécuriser les règles d'accès proxy. Cela peut fournir des retours énormes quand il s'agit de sécuriser le proxy à partir de botnets de réseaux malveillants qui sont souvent créés dans la gamme d'adresses IP de pays qui ont des règles plus laxistes en ce qui concerne la sécurité en ligne. Par conséquent, il est vivement recommandé de limiter les plages d'adresses IP au pays/état ou aux plages d'adresses IP basées sur le FAI si vous êtes sûr des modèles d'accès.

Bloquer les connexions client

Il est également utile de savoir comment bloquer une plage spécifique d'adresses lorsqu'une attaque est identifiée comme étant effectuée à partir d'une adresse IP ou d'une plage d'adresses IP. Dans de tels cas, les requêtes de ces adresses IP peuvent être bloquées avec des règles iptable.

Bloquer les adresses IP distinctes

Afin de bloquer plusieurs adresses IP distinctes, ajoutez une ligne au fichier de configuration IPTables pour chaque adresse IP.

Par exemple, pour bloquer les adresses 192.0.2.3 et 192.0.2.4, entrez :

```
<#root>
```

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

```
192.0.2.3
```

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

```
192.0.2.4
```

```
- j DROP.
```

Bloquer une plage d'adresses IP

Bloquez plusieurs adresses IP dans une plage et ajoutez une seule ligne au fichier de configuration IPTables avec la plage IP.

Par exemple, pour bloquer les adresses comprises entre 192.0.2.3 et 192.0.2.35, saisissez :

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

Bloquer toutes les adresses IP dans un sous-réseau

Bloquez toutes les adresses IP dans un sous-réseau entier en ajoutant une seule ligne au fichier de configuration IPTables avec l'utilisation de la notation de routage interdomaine sans classe pour la plage d'adresses IP. Par exemple, pour bloquer toutes les adresses de classe C, saisissez :

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


SELinux

SELinux est un cadre de sécurité de plate-forme intégré en tant qu'amélioration dans le système d'exploitation Linux. La procédure d'installation et d'ajout de politiques SELinux pour exécuter OpenResty comme proxy inverse est fournie ci-dessous.

1. Arrêtez le processus à l'aide de la `openresty -s stop` commande.
2. Configurez et démarrez /stop nginx server à l'aide de la `systemctl` commande afin que le processus OpenResty démarre automatiquement au démarrage. Entrez ces commandes en tant qu'utilisateur racine.
 - a. Accédez à `/usr/lib/systemd/system`.
 - b. Ouvrez le fichier `openresty.service`.
 - c. Mettez à jour le contenu du fichier selon l'emplacement 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. En tant qu'utilisateur root, entrez `sudo systemctl enable openresty`.
- e. Démarrez / Arrêtez le service OpenResty avec la `systemctl start openresty / systemctl stop openresty` commande et assurez-vous que le processus démarre / s'arrête en tant qu'utilisateur racine.

1. Installation de Selinux

- Par défaut, seuls quelques paquets SELinux seront installés dans CentOS.
- Le paquet `policy-coreutils-devel` et ses dépendances doivent être installés afin de générer la politique SELinux.
- Entrez cette commande afin d'installer `policy coreutils-devel`

```
yum install policycoreutils-devel
```

- Assurez-vous qu'après avoir installé le package, la `sepolicy` commande fonctionne.

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

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

SELinux Policy Inspection Tool

2. Créer un nouvel utilisateur Linux et mapper avec un utilisateur SELinux

- a. Entrez `semanage login -l` afin d'afficher le mappage entre les utilisateurs Linux et les utilisateurs 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. En tant que root, créez un nouvel utilisateur Linux (nginx user) qui est mappé à l'utilisateur 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. Afin d'afficher le mappage entre `nginxuser` et `user_u`, entrez cette commande en tant que 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 par défaut mappé à l'utilisateur SELinux `unrestricted_u`. Il est nécessaire de faire en sorte que `user_u` soit confiné par défaut avec cette commande :

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

Afin de vérifier si la commande fonctionnait correctement, entrez `semanage login -l`. Il devrait produire ce résultat :

Login Name	SELinux User	MLS/MCS Range	Service
__default__	user_u	s0	*
nginxuser	user_u	s0	*
root	unconfined_u	s0-s0:c0.c1023	*

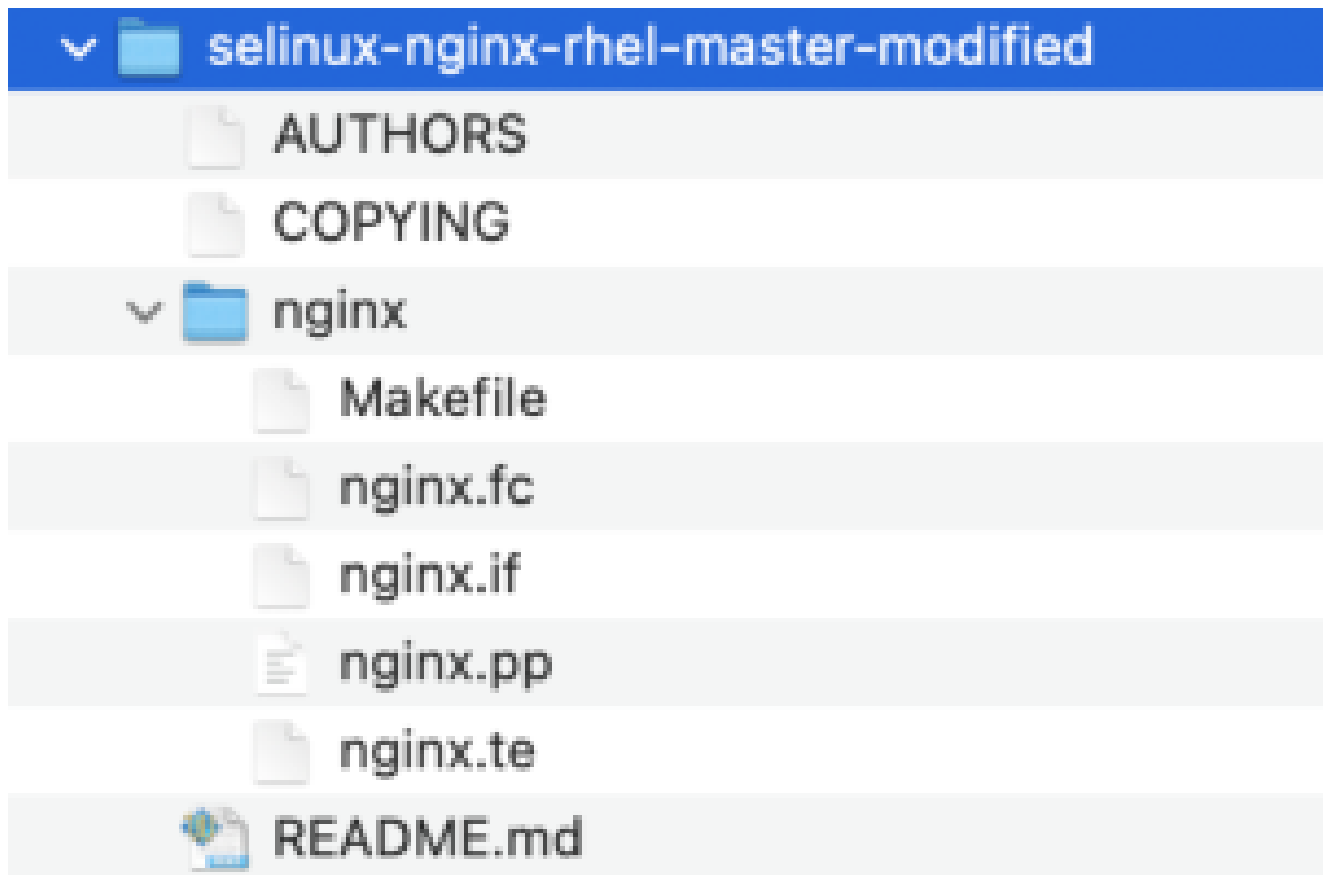
e. Modifiez le fichier `nginx.conf` et modifiez la propriété de `nginxuser`.

- i. Entrez `chown -R nginxuser:nginxuser *` dans le répertoire `<Openresty-install-directory>`.
- ii. Modifiez le fichier `nginx.conf` pour inclure `nginxuser` en tant qu'utilisateur pour l'exécution des processus de travail.

```
.....  
user nginxuser nginxuser;  
.....
```

Écrire la politique SELinux pour Nginx

1. Au lieu de générer une nouvelle stratégie personnalisée par défaut pour Nginx avec la `sepolicy generate --init /usr/bin/nginx` commande, il est préférable de commencer par une stratégie existante.
2. Les fichiers `nginx.fc` (File Contexts file) et `nginx.te` (Type Enforcement file) téléchargés à partir de l'URL fournie ont été modifiés pour s'adapter à l'utilisation du proxy inverse.
3. Cette version modifiée peut être utilisée comme référence puisqu'elle a été corrigée pour le cas d'utilisation particulier.
4. Téléchargez le fichier `selinux-nginx-rhel-master-modified.tar` depuis la [page de téléchargement du logiciel de fichier](#).



5. Extrayez le fichier .tar et naviguez jusqu'au répertoire nginx qu'il contient.
6. Ouvrez le fichier .fc et vérifiez les chemins d'accès requis du programme d'installation de nginx, du cache et du fichier pid.
7. Compilez la configuration à l'aide de la `make` commande.
8. Le fichier nginx.pp sera généré.
9. Chargez la stratégie à l'aide de la `semodule` commande.

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

10. Accédez à /root et créez un fichier vide appelé touch /.autorelabel.
11. Redémarrez le système.
12. Entrez cette commande afin de vérifier que la stratégie a bien été chargée.

```
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 doit s'exécuter sans violation. (Les violations seront disponibles dans /var/log/messages et /var/log/audit/audit.log).
14. Entrez cette commande afin de vérifier l'état 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. Le bureau de l'agent/superviseur Finesse doit maintenant être accessible.

Vérifier

Utilisez cette section pour confirmer que votre configuration fonctionne correctement.

Finesse

1. Demandez <https://<inverseproxy:port>/finesse/api/SystemInfo>. à la DMZ et vérifiez qu'ils sont accessibles.
2. Les valeurs de contrôle <host> dans <primaryNode> et <secondaryNode> sont des noms d'hôte de proxy inverse valides. Il ne doit pas s'agir de noms d'hôte Finesse.

CUIC et données en direct

1. Si des noms d'hôte Finesse sont affichés dans la réponse au lieu de noms d'hôte proxy inverses, les configurations de validation de mappage de proxy et les hôtes autorisés sont correctement ajoutés dans les serveurs Finesse, comme décrit dans la section « Renseigner les données de traduction réseau » de « Accès sans VPN à Finesse Desktop » dans le

[Guide des fonctionnalités UCCE de Finesse 12.6.](#)

2. Si les gadgets LiveData se chargent correctement dans Finesse Desktop, les configurations CUIC et LiveData proxy sont correctes.
3. Afin de valider la configuration CUIC et LiveData, envoyez des requêtes HTTP à ces URL depuis la DMZ et vérifiez si elles sont accessibles.
 - https://<inverseproxy:cuic_port>/cuic/rest/ à propos de
 - https://<inverseproxy:ldweb_port>/livedata/security
 - https://<inverseproxy:ldsocketio_port>/security

IDS

Pour valider la configuration des ID, procédez comme suit :

1. Connectez-vous à l'interface IdSAdmin à l'adresse https://<ids_LAN_host:ids_port>:8553/idsadmin à partir du réseau local, car l'interface d'administration n'est pas exposée sur le proxy inverse.
2. Choisissez Settings > IdS Trust.
3. Vérifiez que le noeud éditeur de cluster proxy est répertorié sur la page Télécharger les métadonnées SP et cliquez sur Suivant.
4. Vérifiez que le proxy IDP est correctement affiché s'il est configuré sur la page Upload IDP metadata et cliquez sur Next.
5. Lancez le test SSO via tous les noeuds de cluster proxy à partir de la page Test SSO et vérifiez que tous les tests ont réussi. Cela nécessite la connectivité de l'ordinateur client pour inverser les noeuds proxy.

rendement

L'analyse des données de capture des performances équivalentes les plus élevées, réalisée avec l'outil nmon, est disponible sur la [page de téléchargement du logiciel Finesse version 12.6\(1\) ES03](#) (load_result.zip). Les données représentent l'état du proxy pour les opérations de bureau et de superviseur, sur un exemple de déploiement UCCE 2000 utilisant des connexions SSO et des rapports CUIC LD configurés dans la disposition par défaut pour 2000 utilisateurs pendant une période de huit heures. Il peut être utilisé pour déterminer les besoins en matière de calcul, de disque et de réseau pour une installation utilisant Nginx sur un matériel comparable.

Dépannage

SSO

1. Les redirections du bureau ne passent pas par le proxy
 1. Vérifiez que les noms d'hôte sont configurés dans les cas corrects en fonction des noms d'hôte réels de la machine virtuelle dans diverses configurations telles que proxymap.txt, fichier server_filter, etc.
 2. Assurez-vous que l'IDs est ajouté avec le nom d'hôte en cas correct dans l'inventaire CCE, car les mêmes informations sont transmises aux composants lors de

l'enregistrement pour l'authentification unique à partir de l'administrateur Web CCE.

2. Les connexions SSO n'ont pas lieu

1. Assurez-vous que l'approbation IdS-IDP est établie pour l'hôte proxy.

SELinux

1. Si Nginx n'est pas démarré par défaut ou si le bureau de l'agent Finesse n'est pas accessible, définissez SELinux en mode permissive avec cette commande :

```
setenforce 0
```

2. Essayez de redémarrer Nginx avec la `systemctl restart nginx` commande.

3. Les violations seront disponibles dans `/var/log/messages` et `/var/log/audit/audit.log`.

4. Il est nécessaire de régénérer le fichier `.te` avec des règles d'autorisation pour traiter ces violations par l'une de ces commandes :

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

5. Mettez à jour le fichier `nginx.te` original présent dans le répertoire `selinux-nginx-rhel-master-modified/nginx` avec les règles d'autorisation nouvellement générées.

6. Compilez-les avec la `make` commande.

7. Le fichier `nginx.pp` sera régénéré.

8. Chargez la commande `policy` by `semodule`.

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

9. Faites de SELinux le mode application avec cette commande :

```
setenforce
```

10. Redémarrez le système.

11. Répétez cette procédure jusqu'à ce que les violations requises soient corrigées.

À propos de cette traduction

Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.