Konfigurieren des Nginx Reverse Proxy für VPN-losen Zugriff auf Cisco Finesse 12.6 ES 02

Inhalt

Einleitung
Voraussetzungen
Anforderungen
Verwendete Komponenten
Hintergrundinformationen
Änderungen in ES02
Aktualisierungshinweise für ES01-basierte VPN-lose Konfigurationen
Authentifizierung
Nicht-SSO-Authentifizierung
SSO-Authentifizierung
Authentifizierung für Websocket-Verbindungen
Prävention von Brute-Force-Angriffen
Protokollieren
Validieren statischer Ressourcen-URLs
Cache der CORS-Header
Konfigurieren
Installieren Sie OpenResty als Reverse Proxy in der DMZ.
Installation von OpenResty
Nginx konfigurieren
Konfigurieren des Nginx-Cache
Konfigurieren von SSL-Zertifikaten
Benutzerdefinierter Diffie-Hellman-Parameter verwenden
Stellen Sie sicher, dass OCSP Stapling aktiviert ist - Prüfung auf Widerruf von Zertifikaten.
Nginx-Konfiguration
Reverse Proxy-Port konfigurieren
Konfigurieren der gegenseitigen TLS-Authentifizierung zwischen Reverse Proxy und Upstream-Komponenten
Cache löschen
Standardrichtlinien
Konfigurieren der Zuordnungsdatei
Reverse Proxy als Zuordnungsdateiserver verwenden
CentOS 8 Kernel-Härtung
IPtables-Härtung
Client-Verbindungen einschränken
Client-Verbindungen blockieren
Unterschiedliche IP-Adressen blockieren
Blockieren eines Bereichs von IP-Adressen
Alle IP-Adressen in einem Subnetz sperren
SELinux

Überprüfung
Finesse
CUIC- und Live-Daten
IDs
Leistung
Fehlerbehebung

Einleitung

In diesem Dokument wird beschrieben, wie Sie einen Reverse Proxy für den Zugriff auf den Cisco Finesse-Desktop verwenden, ohne eine Verbindung zu einem VPN herzustellen, das auf den Versionen 12.6 ES02 von Cisco Finesse, Cisco Unified Intelligence Center (CUIC) und Cisco Identity Service (IdS) basiert.

> **Anmerkung:** Installation und Konfiguration von Nginx werden von Cisco nicht unterstützt. Fragen zu diesem Thema können in den Cisco Community Foren diskutiert werden.

> **Anmerkung:** Bei ES02-Bereitstellungen von VPN-Less beachten Sie die Versionshinweise der einzelnen Komponenten, um die Upgrades zu planen und die Kompatibilitätsbeschränkungen zu überprüfen.

Voraussetzungen

Anforderungen

Cisco empfiehlt, dass Sie über Kenntnisse in folgenden Bereichen verfügen:

- Cisco Unified Contact Center Enterprise (UCCE)-Version
- Cisco Finesse
- Linux-Administration
- Netzwerkverwaltung und Linux-Netzwerkadministration

Verwendete Komponenten

Die Informationen in diesem Dokument basierend auf folgenden Software- und Hardware-Versionen:

- Finesse - 12,6 ES02
- CUIC - 12.6 ES02
- IDs - 12,6 ES02
- UCCE/Hosted Collaboration Solution (HCS) für Contact Center (CC) - 11.6 oder höher
- Packaged Contact Center Enterprise (PCCE) - 12.0 oder höher

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

Anmerkung: Die in diesem Dokument bereitgestellte Konfiguration wurde im Vergleich zu einer 2000 UCCE-Beispielbereitstellung mit dem auf CentOS 8.0 bereitgestellten Reverse Proxy von Nginx konfiguriert, gehärtet und geladen. Leistungs- und Skalierungszahlen finden Sie im Funktionsleitfaden.

Hintergrundinformationen

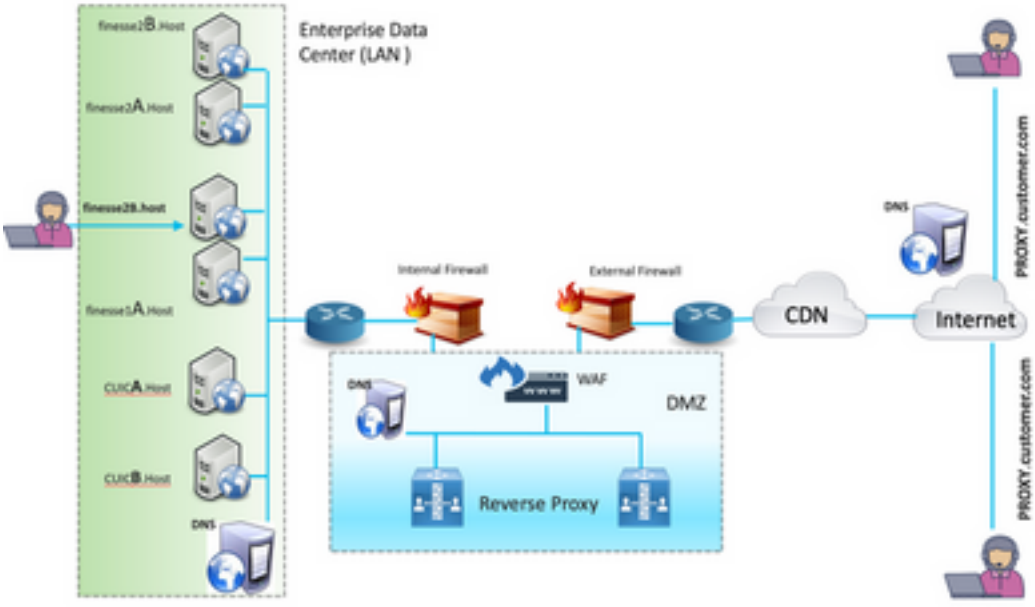
Dieses Bereitstellungsmodell wird für UCCE-/PCCE- und HCS-Lösungen für UCCE-Lösungen unterstützt.

Die Bereitstellung eines Reverse Proxy (ab 12.6 ES01 verfügbar) wird als Option für den Zugriff auf den Cisco Finesse Desktop ohne Verbindung zu einem VPN unterstützt. Diese Funktion bietet Agenten die Flexibilität, von überall aus über das Internet auf den Finesse Desktop zuzugreifen.

Um diese Funktion zu aktivieren, muss ein Reverse Proxy-Paar in der Demilitarized Zone (DMZ) bereitgestellt werden.

Der Medienzugriff bei Reverse Proxy-Bereitstellungen bleibt unverändert. Um eine Medienverbindung herzustellen, können Mitarbeiter Cisco Jabber over Mobile and Remote Access Solution (MRA) oder die Mobile Agent-Funktion von UCCE mit einem Public Switched Telefone Network (PSTN) oder einem mobilen Endgerät verwenden. Dieses Diagramm zeigt, wie die Netzwerkbereitstellung aussieht, wenn Sie über ein einzelnes HA-Paar (Reverse Proxy-Knoten) auf zwei Finesse-Cluster und zwei CUIC-Knoten zugreifen.

Der gleichzeitige Zugriff von Agenten im Internet und Agenten, die eine Verbindung über das LAN herstellen, wird unterstützt, wie in diesem Bild gezeigt.



Anmerkung: Jeder Reverse Proxy, der die erforderlichen Kriterien unterstützt, wie in diesem Funktionsleitfaden beschrieben, kann anstelle von Nginx verwendet werden, um diese Bereitstellung zu unterstützen.

- UCCE 12.6 Feature Guide: Dieser Leitfaden enthält eine Funktionsübersicht, ein Design sowie **Konfigurationsdetails** für die VPN-lose-Funktion.
- UCCE 12.6 Security Guide - Bietet Sicherheitskonfigurationsrichtlinien für die Reverse

Proxy-Bereitstellung.

Es wird empfohlen, sich vor dem Lesen dieses Dokuments den Abschnitt VPN-Less des Funktionsleitfadens und des Sicherheitsleitfadens anzusehen.

Änderungen in ES02

- Neue Funktionen Die Supervisor-Funktionen von Finesse werden nun über Reverse Proxy unterstützt.CUIC RealTime- und Verlaufsberichte werden jetzt über Finesse-Gadgets in einer proxisierten Umgebung unterstützt.Authentifizierung für alle Anfragen/Mitteilungen Alle Finesse-/CUIC-/IM & Presence-Anfragen (IM&P) werden am Edge authentifiziert, bevor sie in das Rechenzentrum eindringen dürfen.WebSocket- und Live Data SocketIO-Verbindungen sind ebenfalls beschränkt und nur von Clients erlaubt, die erfolgreich eine gesicherte Anfrage an Finesse gestellt haben.Brute Force-Erkennung und -Protokollierung von Angriffen, die mit Fail2Ban verwendet werden können, um ungültige Benutzer zu blockieren.

- Sicherheitsverbesserungen für die Konfiguration von Reverse Proxy - Lua erforderlich Gegenseitige TLS-Authentifizierung (Transport Layer Security) zwischen Reverse Proxy und Upstream-Komponenten (Finesse/IdS/CUIC/Livedata).SeLinux-Einstellungen.Aktivieren Sie die gegenseitige Überprüfung der SSL-Vertrauenswürdigkeit (Secure Sockets Layer) für Proxy- und Komponentenserveranforderungen.

- Verbesserte Sicherheit für die Proxy-Konfiguration zur Verhinderung von Denial-of-Service (DoS)-/Distributed Denial-of-Service (DDoS)-Angriffen Erweiterte Nginx-Anforderungsratenbeschränkungen für verschiedene Teile des Systems.Übertragungsratenbeschränkungen für IP-Tabellen.Überprüfung statischer Ressourcenanforderungen, bevor der Upstream angefordert wird.Leichtere und zwischenspeicherbare, nicht authentifizierte Seiten, die nicht auf die Upstream-Ebene treffen.Zwischengespeicherte Cross-Origin Resource Sharing (CORS)-Antworten vom Proxy unterstützen die automatische Konfiguration und die Leistungssteigerung.

Aktualisierungshinweise für ES01-basierte VPN-lose Konfigurationen

- Die ES02-Konfiguration erfordert die Installation von Nginx mit Lua.
- ES02-Konfigurationsskripte erfordern auch die entsprechende ES02 COP-Installation in Cisco Finesse, CUIC und IdS
- Zertifikatanforderungen Cisco Finesse, CUIC und IdS erfordern, dass das Nginx/OpenResty-Hostzertifikat dem Tomcat Trust Store hinzugefügt und ein Neustart durchgeführt wird, bevor die Nginx ES02-Konfiguration erfolgreich eine Verbindung zum Upstream-Server herstellen kann.Die Upstream-Serverzertifikate Cisco Finesse, CUIC und IdS müssen im Nginx-Server konfiguriert werden, um die ES02-basierte Konfiguration zu verwenden.

Anmerkung: Es wird empfohlen, die vorhandene ES01-basierte Nginx-Konfiguration zu entfernen, bevor Sie die ES02 Nginx-Konfigurationen installieren.

Authentifizierung

Finesse 12.6 ES02 führt die Authentifizierung am Netzwerk-Edge ein. Die Authentifizierung wird für Single Sign On (SSO)- und Nicht-SSO-Bereitstellungen unterstützt.

Die Authentifizierung wird für alle Anforderungen und Protokolle erzwungen, die vom Proxy akzeptiert werden, bevor sie an die Upstream-Komponentenserver weitergeleitet werden, wo auch die von den Komponentenservern lokal erzwungene reguläre Authentifizierung erfolgt. Bei der Authentifizierung werden die Anforderungen mit den allgemeinen Anmeldeinformationen von Finesse authentifiziert.

Persistente Verbindungen, z. B. Websockets, die Anwendungsprotokolle wie Extensible Messaging and Presence Protocol (XMPP) für die Authentifizierung und die Nachverbindung verwenden, werden am Proxy authentifiziert, indem die IP-Adresse der zulässigen Liste hinzugefügt wird, aus der eine erfolgreiche Anwendungsauthentifizierung vor dem Herstellen der Socket-Verbindung erstellt wurde.

Nicht-SSO-Authentifizierung

Für die Nicht-SSO-Authentifizierung sind keine zusätzlichen Konfigurationen erforderlich. Sobald die erforderlichen Skriptersetzungen vorgenommen wurden, können Sie sofort Nginx-Konfigurationsskripts verwenden. Die Authentifizierung stützt sich auf den Benutzernamen und das Kennwort für die Anmeldung bei Finesse. Der Zugriff auf alle Endpunkte wird mit den Authentifizierungsservices von Finesse validiert.

Alle gültigen Benutzer werden im Proxy zwischengespeichert (der Cache wird alle 15 Minuten aktualisiert), sodass die Anforderung nicht von einem ungültigen Benutzer auf den Finesse-Server zugreifen kann. Nach erfolgreicher Authentifizierung werden die Details des Benutzers 15 Minuten lang im Proxy zwischengespeichert, wodurch weitere Authentifizierungsanfragen nicht zu Finesse gelangen. Wenn der Benutzername oder das Kennwort geändert werden, wird dies erst nach 15 Minuten wirksam.

SSO-Authentifizierung

Für die SSO-Authentifizierung muss der Administrator den IDS-Token-Verschlüsselungsschlüssel auf dem Nginx-Server in der Konfigurationsdatei konfigurieren. Der IDs-Token-Verschlüsselungsschlüssel kann vom IDs-Server mit dem **show ids secret** CLI-Befehl. Der Schlüssel muss als Teil eines Austauschs für #MustChange konfiguriert werden, den der Administrator in den Skripten ausführen muss, bevor die SSO-Authentifizierung funktionieren kann.

Im SSO-Benutzerhandbuch finden Sie Informationen zu den durchzuführenden IDS-SAML-Konfigurationen für die Proxy-Auflösung, um IDs zu erstellen.

Nach der Konfiguration der SSO-Authentifizierung kann ein gültiges Paar Token für den Zugriff auf alle Endpunkte im System verwendet werden.

Authentifizierung für Websocket-Verbindungen

Websocket-Verbindungen können nicht mit dem standardmäßigen Autorisierungs-Header authentifiziert werden, da benutzerdefinierte Header von nativen Websocket-Implementierungen im Browser nicht unterstützt werden. Authentifizierungsprotokolle auf Anwendungsebene, bei denen die in der Payload enthaltenen Authentifizierungsinformationen die Einrichtung von Websocket-Verbindungen nicht verhindern. Daher können böswillige Instanzen DOS- oder DDOS-

Angriffe rendern, indem sie unzählige Verbindungen erstellen, um das System zu überlasten.

Um diese Möglichkeit abzuschwächen, verfügen die bereitgestellten Reverse Proxy-Konfigurationen des Nginx über spezielle Prüfungen, um zu ermöglichen, dass Websockverbindungen NUR von den IP-Adressen akzeptiert werden können, die vor der Einrichtung der Websockverbindung erfolgreich eine authentifizierte REST-Anfrage gestellt haben. Dies bedeutet, dass Clients, die versuchen, Websocket-Verbindungen zu erstellen, bevor eine REST-Anfrage ausgegeben wird, jetzt einen Fehler bei der Autorisierung erhalten, der nicht unterstützt wird.

Prävention von Brute-Force-Angriffen

Finesse 12.6 ES02-Authentifizierungsskripts verhindern aktiv Brute-Force-Angriffe, die zum Erraten des Benutzerkennworts verwendet werden können. Dies geschieht, indem die IP-Adresse für den Zugriff auf den Dienst blockiert wird, nachdem in kurzer Zeit eine bestimmte Anzahl von Fehlversuchen aufgetreten ist. Diese Anfragen werden durch einen **Client-Fehler** von **418** abgelehnt. Auf Details zu den blockierten IP-Adressen können Sie von den Dateien **<nginx-install-directory>/logs/blocking.log** und **<nginx-install-directory>/logs/error.log** aus zugreifen**.**

Die Anzahl der fehlgeschlagenen Anfragen, das Zeitintervall und die Blockierungsdauer können konfiguriert werden. Konfigurationen sind in der Datei **<nginx-install-directory>/conf/conf.d/maps.conf** vorhanden.

```
## 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;
}
```

Protokollieren

```
grep -r "IP is already blocked." error.log
========================================

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"

tail -f blocking.log
====================

2021/10/29 17:30:59 [error] 939738#939738: *1857 [lua] content_by_lua(rest_cache:189 2:
```

```
[10.70.235.30] will be blocked for [ 30 minutes ] for exceeding retry limit., client:
10.70.235.30, server: saproxy.cisco.com, request: "GET /finesse/api/SystemInfo HTTP/1.1", host:
"saproxy.cisco.com:8445"
```

Es wird empfohlen, dass Kunden in Fail2Ban oder ähnlich integrieren, um die Verbot zur IPtable / Firewall-Regeln hinzuzufügen.

Validieren statischer Ressourcen-URLs

Alle gültigen Endpunkte, auf die nicht authentifiziert zugegriffen werden kann, werden in den ES02-Skripten aktiv nachverfolgt.

Anforderungen an diese nicht authentifizierten Pfade werden bei Anforderung eines ungültigen URI aktiv abgelehnt, ohne dass diese Anforderungen an den Upstream-Server gesendet werden.

Cache der CORS-Header

Wenn die erste Optionsanfrage erfolgreich ist, werden die **Zugriffskontroll-Zugriffskontroll-Header, die Zugriffskontroll-Zugriffskontrollmethoden, die Zugriffskontroll-Expose-Header** und die **Zugriffskontroll-Zulassen-Anmeldeinformationen** fünf Minuten lang im Proxy zwischengespeichert. Diese Header werden für jeden jeweiligen Upstream-Server zwischengespeichert.

Konfigurieren

In diesem Dokument wird die Konfiguration von Nginx als umgekehrter Proxy beschrieben, der zur Aktivierung von Finesse VPN-Less-Zugriff verwendet wird. Die UCCE-Lösungskomponente, die Proxy- und Betriebssystemversionen, mit denen die bereitgestellten Anweisungen verifiziert werden, werden bereitgestellt. Die entsprechenden Anweisungen müssen an das Betriebssystem/Proxy Ihrer Wahl angepasst werden.

- Verwendete Nginx-Version - OpenResty 1.19.9.1
- Betriebssystem für Konfiguration - CentOS 8.0

> **Anmerkung:** Die beschriebene Nginx-Konfiguration kann von der Finesse Release 12.6(1)ES2 Software-Download-Seite heruntergeladen werden.

Installieren Sie OpenResty als Reverse Proxy in der DMZ.

In diesem Abschnitt werden die Schritte zur Proxy-Installation auf Basis von OpenResty beschrieben. Der umgekehrte Proxy wird in der Regel als dediziertes Gerät in der demilitarisierten Zone (DMZ) des Netzwerks konfiguriert, wie im zuvor erwähnten Bereitstellungsdiagramm gezeigt.

1. Installieren Sie das **Betriebssystem Ihrer Wahl** mit der erforderlichen Hardwarespezifikation. Die Parametereinstellungen für Kernel und IPv4 können je nach ausgewähltem Betriebssystem unterschiedlich sein. Den Benutzern wird empfohlen, diese Aspekte erneut zu überprüfen, wenn sich die gewählte Betriebssystemversion von der anderen Version unterscheidet.
2. Konfigurieren Sie zwei Netzwerkschnittstellen. Für den öffentlichen Zugriff von den Internet-Clients wird eine Schnittstelle benötigt, für die Kommunikation mit den Servern im internen Netzwerk eine andere.

3. Installieren Sie OpenResty.

Zu diesem Zweck können alle Nginx-Aromen verwendet werden, sofern sie auf Nginx 1.19+ basieren und Lua unterstützen:

- Nginx Plus
- Nginx Open Source (Nginx Open Source muss zusammen mit OpenResty-basierten Lua-Modulen kompiliert werden, damit sie verwendet werden kann)
- ÖffnenResty
- GetPageSpeed-Extras

Anmerkung: Die bereitgestellte Konfiguration wurde mit OpenResty 1.19 getestet und sollte mit anderen Distributionen, falls vorhanden, nur mit kleineren Updates funktionieren.

Installation von OpenResty

1. Installieren Sie OpenResty. Siehe OpenResty Linux Packages. Im Rahmen der OpenResty-Installation wird Nginx an diesem Speicherort installiert und der ***PATH***-Variable den OpenResty-Pfad hinzugefügt, indem die ***~/.bashrc***-Datei hinzugefügt wird.
   ```
   export PATH=/usr/local/openresty/bin:$PATH
   ```
2. Starten/Beenden von Nginx Um Nginx zu starten, geben Sie **openresty**.Um Nginx zu stoppen, geben Sie **openresty -s stop**.

Nginx konfigurieren

Die Konfiguration wird für eine OpenResty-basierte Nginx-Installation erklärt. Die Standardverzeichnisse für OpenResty sind:

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

1. Laden Sie die Datei von der Finesse Release 12.6(1)ES2 Software-Download-Seite herunter (12.6-ES02-reverse-proxy-config.zip), die die umgekehrte Proxy-Konfiguration für Nginx enthält, und extrahieren Sie sie.
2. Kopieren Sie nginx.conf, nginx/conf.d/ und nginx/html/aus dem extrahierten Verzeichnis für die Konfiguration des Reverse Proxy in das Verzeichnis **<nginx-install-directory>/conf**, **<nginx-install-directory>/conf/conf.d/** und **<nginx-install-directory>/html/**bzw.
3. Kopieren Sie das Verzeichnis **nginx/lua** aus dem extrahierten Verzeichnis für die Reverse Proxy-Konfiguration im **<nginx-install-directory>.**
4. Kopieren Sie den Inhalt von **lualib** in **<Openresty-install-directory>/lualib/resty**.
5. Konfigurieren Sie die Nginx-Protokollrotation, indem Sie die **Datei "nginx/logrotate/saproxy"** in den **<nginx-install-directory>/logrotate/**Ordner kopieren. Ändern Sie den Dateiinhalt so, dass er auf die richtigen Protokollverzeichnisse zeigt, wenn die Nginx-Standardwerte nicht verwendet werden.
6. Nginx muss mit einem dedizierten, nicht privilegierten Dienstkonto ausgeführt werden, das gesperrt werden muss und über eine ungültige Shell (oder für das ausgewählte Betriebssystem) verfügen muss.
7. Suchen Sie die Zeichenfolge **"Must-change"** in den Dateien unter den extrahierten Ordnern **html** und **conf.d** und ersetzen Sie die angegebenen Werte durch entsprechende Einträge.

8. Stellen Sie sicher, dass alle obligatorischen Ersetzungen vorgenommen werden, die mit den Kommentaren **zum Ändern** der Konfiguration in den Konfigurationsdateien beschrieben werden.
9. Stellen Sie sicher, dass die für CUIC und Finesse konfigurierten Cache-Verzeichnisse unter **<nginx-install-directory>/cache** zusammen mit diesen temporären Verzeichnissen erstellt werden. <nginx-install-directory>/cache/client_temp<nginx-install directory>/cache/proxy_temp

Anmerkung: Die bereitgestellte Konfiguration ist für eine Beispielbereitstellung für das Jahr 2000 vorgesehen und muss für eine größere Bereitstellung entsprechend erweitert werden.

Konfigurieren des Nginx-Cache

Standardmäßig werden die Proxycache-Pfade im Dateisystem gespeichert. Wir empfehlen, diese in In-Memory-Laufwerke zu ändern, indem Sie einen Cache-Speicherort in den tmpfs erstellen, wie hier gezeigt.

1. Erstellen Sie Verzeichnisse für die verschiedenen Proxycache-Pfade unter /home. Beispielsweise müssen diese Verzeichnisse für die primäre Finesse erstellt werden. Für die sekundären Finesse- und CUIC-Server sollten dieselben Schritte ausgeführt werden.
```
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
```
Anmerkung: Erhöhen Sie die Cache-Speicher für Client und proxy_temp für jedes neue Finesse-Cluster, das der Konfiguration hinzugefügt wurde, um 1 GB.
2. Montieren Sie die neuen Bereitstellungspunkte mit dem Befehl **mount -av**.
3. Validieren Sie, dass das Dateisystem die neuen Mount-Punkte mit dem **df -h** aus.
4. Ändern Sie die proxy_cache_path-Speicherorte in den Konfigurationsdateien für Finesse und CUIC-Cache. Um beispielsweise die Pfade für das Finesse Primär zu ändern, gehen Sie zu **<nginx-install-directory>conf/conf.d/finesse/caches** und ändern Sie den vorhandenen Cache-Standort **/usr/local/openresty/nginx/cache/finesse25/** in den neu erstellten Dateisystemstandort **/home/primaryFinesse.**
```
##Must-change /usr/local/openresty/nginx/cache/
location would change depending on folder extraction ##
```

```
Nginx config file to cache the desktop/shindig and notification service related static
files.
proxy_cache_path /home/primaryFinesse/desktop levels=1:2 use_temp_path=on
keys_zone=desktop_cache_primary: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_primary: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_primary: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:10m max_size=1500m inactive=40m
use_temp_path=off;
```

5. Befolgen Sie die gleichen Schritte für die sekundären und CUIC-Server von Finesse.

Anmerkung: Stellen Sie sicher, dass die Summe aller im vorherigen Schritt erstellten tmpfs-Festplattengrößen der endgültigen Speichergröße für die Bereitstellung hinzugefügt wird, da es sich bei diesen Festplatten um Speicherblöcke handelt, die so konfiguriert sind, dass sie wie Festplatten für die Anwendung aussehen und so viel Speicherplatz belegen.

Konfigurieren von SSL-Zertifikaten

Eigene Zertifikate verwenden - Testbereitstellungen

Selbstsignierte Zertifikate sollten nur verwendet werden, bis der umgekehrte Proxy zur Produktion bereit ist. Verwenden Sie bei einer Produktionsumgebung nur ein Zertifizierungsstellen-signiertes Zertifikat.

1. Generieren Sie Nginx-Zertifikate für SSL-Ordnerinhalte. Bevor Sie Zertifikate generieren, müssen Sie einen Ordner namens **ssl** unter **/usr/local/openresty/nginx** erstellen. Da Sie zwei Hostnamen (denselben Proxyserver) für den Zugriff auf Finesse node1 und Finesse node2 verwenden, müssen Sie mithilfe dieser Befehle zwei Zertifikate generieren (eines für **<reverseproxy_primary_fqdn>** und eines für **<reverseproxy_Sekundär_fqdn>**). **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** (Übergeben Sie den Hostnamen als: <reverseproxy_primary_fqdn>)**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** (Übergeben Sie den Hostnamen als :<reverseproxy_sekundär_fqdn>)Stellen Sie sicher, dass der Zertifikatspfad **etc/nginx/ssl/nginx.crt** und **/usr/local/openresty/nginx/ssl/nginxnode2.crt** lautet, da diese bereits in den Finesse Nginx-Konfigurationsdateien konfiguriert sind.
2. Ändern Sie die Berechtigung für den privaten Schlüssel **400 (r—)**.
3. Konfigurieren Sie Firewall/iptables auf dem Reverse Proxy, um die Kommunikation von der Firewall so zu ermöglichen, dass sie den Ports entspricht, auf denen der Nginx-Server für die Überwachung konfiguriert wurde.
4. Fügen Sie die IP-Adresse und den Hostnamen von Finesse, IdS und CUIC unter dem Eintrag **/etc/hosts** auf dem umgekehrten Proxy-Server hinzu.
5. Im Leitfaden mit den Lösungsfunktionen finden Sie die Konfigurationen, die auf den Komponentenservern durchgeführt werden müssen, um den Nginx-Host als Reverse Proxy zu konfigurieren.

Anmerkung: Die bereitgestellte Konfiguration ist für eine Beispielbereitstellung für das Jahr 2000 vorgesehen und muss für eine größere Bereitstellung entsprechend erweitert werden.

Von CA signiertes Zertifikat verwenden - Produktionsbereitstellungen

Ein CA-signiertes Zertifikat kann auf dem Reverse Proxy mit den folgenden Schritten installiert werden:

1. Erstellen Sie die Zertifikatssignierungsanfrage (Certificate Signing Request, CSR). Um CSR und privaten Schlüssel zu generieren, geben Sie **openssl req -new -newkey rsa:4096 -keyout nginx.key -out nginx.csr** nachdem Sie sich beim Proxy angemeldet haben. Folgen Sie der Eingabeaufforderung, und geben Sie die Details an. Dies generiert den CSR (nginx.csr im Beispiel) und den privaten RSA-Schlüssel (nginx.key im Beispiel) für Stärke 4096 Bit.Beispiele:

```
[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.comEmail 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
```

   Notieren Sie sich die PEM-Kennzeichenfolge, da diese während der Bereitstellung zum Entschlüsseln des privaten Schlüssels verwendet wird.
2. Sie erhalten das signierte Zertifikat von der Zertifizierungsstelle. Senden Sie die CSR-Anfrage an die Zertifizierungsstelle, und erhalten Sie das signierte Zertifikat.
3. Stellen Sie das Zertifikat und den Schlüssel bereit. Entschlüsseln Sie den zuvor generierten Schlüssel im ersten Schritt mit dem **openssl rsa -in nginx.key -out nginx_decrypted.key** aus. Platzieren Sie das signierte Zertifikat der Zertifizierungsstelle und den entschlüsselten Schlüssel in einem Ordner (/usr/local/openresty/nginx/ssl im Beispiel) auf dem Reverse Proxy-System. Aktualisieren/Hinzufügen von SSL-Konfigurationen für das Zertifikat in den Nginx-Konfigurationen

```
server {
    server_name <proxy-name>;
    listen 8445 ssl reuseport http2;
    listen [::]:8445 ssl reuseport http2;
    .....
    ssl_certificate /usr/local/openresty/nginx/ssl/ca_signed_cert.crt;
    ssl_certificate_key /usr/local/openresty/nginx/ssl/nginx_decrypted.key;
    .....
```

```
}
```

4. Konfigurieren Sie Berechtigungen für die Zertifikate. Eingabe **chmod 400 /usr/local/openresty/nginx/ssl/ca_signed_cert.crt** und **chmod 400 /usr/local/openresty/nginx/ssl/nginx_decrypted.key**, sodass das Zertifikat über eine Schreibschutzberechtigung verfügt und auf den Besitzer beschränkt ist.
5. Nginx erneut laden.

Benutzerdefinierter Diffie-Hellman-Parameter verwenden

Erstellen Sie mit den folgenden Befehlen einen benutzerdefinierten Diffie-Hellman-Parameter:

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

Ändern Sie die Serverkonfiguration, um die neuen Parameter zu verwenden:

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

Stellen Sie sicher, dass OCSP Stapling aktiviert ist - Prüfung auf Widerruf von Zertifikaten.

> **Anmerkung:** Um dies zu aktivieren, sollte der Server ein Zertifikat mit Zertifizierungsstellenzahl verwenden und der Server sollte Zugriff auf die Zertifizierungsstelle haben, die das Zertifikat signiert hat.

Fügen Sie diese zur Konfiguration hinzu:

```
server {
    .....
    ssl_stapling on;
    ssl_stapling_verify on;
    .....
}
```

Nginx-Konfiguration

Die Standard-Nginx-Konfigurationsdatei (**/usr/local/openresty/nginx/conf/nginx.conf**) muss so geändert werden, dass sie diese Einträge enthält, um Sicherheit und Leistung zu gewährleisten. Dieser Inhalt sollte zum Ändern der Standardkonfigurationsdatei verwendet werden, die von der Nginx-Installation erstellt wird.

```
# Increasing number of worker processes will not increase the processing the request. The number
of worker process will be same as number of cores
# in system CPU. Nginx provides "auto" option to automate this, which will spawn one worker for
each CPU core.
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 our proxy set up the value is 0
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_connections

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 maximum number of open files for the user (ulimit -Hn)
    # The appropriate setting depends on the size of the server and the nature of the traffic,
and can be discovered through testing.
    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_manager")
        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/api/Users")
            UnauthenticatedDesktopResourcesManager.getDesktopResources("saproxy.cisco.com",
"https://saproxy.cisco.com:8445/desktop/api/urls?type=desktop")
            UnauthenticatedResourcesManager.getThirdPartyGadgetResources("saproxy.cisco.com",
"https://saproxy.cisco.com:8445/desktop/api/urls?type=3rdParty")         end     }     include
conf.d/*.conf;     sendfile        on;     tcp_nopush     on;     server_names_hash_bucket_size
512;
```

Reverse Proxy-Port konfigurieren

Standardmäßig überwacht die Nginx-Konfiguration den Port 8445 für Finesse-Anforderungen. Es kann jeweils nur ein Port von einem Reverse Proxy aktiviert werden, um Finesse-Anfragen zu unterstützen, z. B. 8445. Wenn Port 443 unterstützt werden soll, bearbeiten Sie die Datei **<nginx-install-directory>conf/conf.d/finesse.conf**, um das Abhören auf 443 zu aktivieren und das Abhören auf 8445 zu deaktivieren.

Konfigurieren der gegenseitigen TLS-Authentifizierung zwischen Reverse Proxy und Upstream-Komponenten

In ES02 sind standardmäßig alle Upstream-Komponenten so konfiguriert, dass alle Hosts, die als Teil des CLI-**UTILS-Systems** für "allowed-**proxy-allowed-hosts** hinzugefügt werden, validiert werden**, fügen Sie den** Befehl **<proxy-host>** hinzu. Damit eine erfolgreiche Kommunikation zwischen Reverse Proxy-Hosts und Upstream-Komponenten (Finesse/IdS/CUIC/Livedata) möglich ist, sollten Reverse Proxy-Zertifikate in den tomcat Trust Store aller Upstream-Komponenten hochgeladen werden. Starten Sie den Knoten neu, sobald Zertifikate hochgeladen wurden.

Die Validierung von Upstream-Serverzertifikaten durch den umgekehrten Proxy ist optional und standardmäßig deaktiviert. Wenn Sie eine vollständige gegenseitige TLS-Authentifizierung zwischen Reverse Proxy und Upstream-Hosts erreichen möchten, muss diese Konfiguration in den Dateien **ssl.conf** und **ssl2.conf** nicht kommentiert werden.

```
#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;
```
Die Datei **proxy_ssl_trust_certificate** sollte alle Upstream-Zertifikateinträge enthalten, die miteinander verknüpft sind.

Cache löschen

Der umgekehrte Proxy-Cache kann mit dem aus.

Standardrichtlinien

Dieser Abschnitt beschreibt kurz die Standardrichtlinien, die befolgt werden müssen, wenn Sie Nginx als Proxyserver einrichten.

Diese Richtlinien stammen vom Center for Internet Security. Weitere Einzelheiten zu den einzelnen Leitfäden finden Sie in derselben.

1. Es wird immer empfohlen, die neueste stabile Version von OpenResty und OpenSSL zu verwenden.
2. Es wird empfohlen, Nginx in einer separaten Datenträgerbereitstellung zu installieren.
3. Die Nginx-Prozess-ID muss dem Root-Benutzer gehören (oder für ein bestimmtes Betriebssystem zutreffend) und über die Berechtigung **644 (rw—)** oder eine striktere verfügen.
4. Nginx muss Anfragen für unbekannte Hosts blockieren. Stellen Sie sicher, dass jeder Serverblock die explizit definierte server_name-Direktive enthält. Durchsuchen Sie zum Überprüfen alle Serverblöcke im Verzeichnis **nginx.conf** und **nginx/conf.d**, und überprüfen Sie, ob alle Serverblöcke den Servernamen enthalten.
5. Nginx darf nur auf autorisierten Ports überwachen. Durchsuchen Sie alle Serverblöcke im Verzeichnis **nginx.conf** und **nginx/conf.d**, und überprüfen Sie, ob die Richtlinien überwacht werden, um sicherzustellen, dass nur die autorisierten Ports für das Abhören geöffnet sind.
6. Da Cisco Finesse HTTP nicht unterstützt, wird empfohlen, auch den HTTP-Port des Proxyservers zu blockieren.
7. Das Nginx SSL-Protokoll muss TLS 1.2 sein. Die Unterstützung für ältere SSL-Protokolle muss entfernt werden. Außerdem müssen schwache SSL-Chiffren deaktiviert werden.
8. Es wird empfohlen, Nginx-Fehler- und Zugriffsprotokolle an den Remote-Syslog-Server zu senden.
9. Es wird empfohlen, das **mod_security-**Modul zu installieren, das als Web-Anwendungs-Firewall funktioniert. Weitere Informationen finden Sie im ModSecurity-Handbuch. Beachten Sie, dass die Nginx-Last nicht im **mod_security-**Modul überprüft wurde.

Konfigurieren der Zuordnungsdatei

Für die Reverse Proxy-Bereitstellung des Finesse-Desktops ist eine Zuordnungsdatei erforderlich, um die Liste der extern sichtbaren Hostnamen/Port-Kombinationen und deren Zuordnung zu den tatsächlichen Servernamen und Ports zu konfigurieren, die von den Finesse-, IdS- und CUIC-Servern verwendet werden. Diese auf internen Servern konfigurierte Zuordnungsdatei ist die Schlüsselkonfiguration, mit der die über das Internet verbundenen Clients an die erforderlichen Hosts und Ports weitergeleitet werden können, die im Internet verwendet werden.

Die Zuordnungsdatei muss auf einem Webserver bereitgestellt werden, auf den die

Komponentenserver zugreifen können. Der URI muss konfiguriert werden, damit die Bereitstellung funktioniert. Es wird empfohlen, die Zuordnungsdatei mithilfe eines dedizierten, im Netzwerk verfügbaren Webservers zu konfigurieren. Wenn ein solcher Server nicht verfügbar ist, kann stattdessen der umgekehrte Proxy verwendet werden. Dies erfordert, dass der Proxy von innerhalb des Netzwerks aus erreichbar ist. Außerdem besteht die Gefahr, dass die Informationen externen Clients zugänglich gemacht werden, die nicht autorisierten Zugriff auf die DMZ gewähren können. Im nächsten Abschnitt wird beschrieben, wie dies möglich ist.

Im Funktionsleitfaden finden Sie genaue Schritte zum Konfigurieren des Mapping-Datei-URI auf allen Komponentenservern sowie weitere Informationen zum Erstellen der Mapping-Dateidaten.

Reverse Proxy als Zuordnungsdateiserver verwenden

Diese Schritte sind nur erforderlich, wenn der umgekehrte Proxy auch als Host für die Proxyzuordnungsdatei verwendet wird.

1. Konfigurieren Sie den umgekehrten Proxy-Hostnamen im Domänen-Controller, der von den Finesse/CUIC- und IDS-Hosts verwendet wird, sodass dessen IP-Adresse aufgelöst werden kann.
2. Laden Sie die generierten signierten Nginx-Zertifikate auf beide Knoten unter "tomcat-trust" von cmplatform hoch und starten Sie den Server neu.
3. Aktualisieren Sie die Werte unter **<NGINX_HOME>/html/proxymap.txt**.
4. Laden Sie Nginx-Konfigurationen mit dem **nginx -s reload** aus.
5. Überprüfen Sie, ob der Zugriff auf die Konfigurationsdatei von einem anderen Netzwerk-Host aus mithilfe der **curl** aus.

CentOS 8 Kernel-Härtung

Wenn das gewählte Betriebssystem CentOS 8 ist, wird empfohlen, dass Kernel-Härtung/Tuning mit der Verwendung dieser Systemkonfigurationen für Installationen durchgeführt wird, die einen dedizierten Server zum Hosten des Proxys verwenden.

```
## 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
```

Nach Durchführung der empfohlenen Änderungen wird ein Neustart empfohlen.

IPtables-Härtung

IPtables ist eine Anwendung, mit der Systemadministratoren die von der Linux-Kernel-Firewall bereitgestellten IPv4- und IPv6-Tabellen, -Ketten und -Regeln konfigurieren können.

Diese IPtables-Regeln werden konfiguriert, um die Proxy-Anwendung vor Brute-Force-Angriffen zu schützen, indem der Zugriff in der Linux-Kernel-Firewall eingeschränkt wird.

Die Kommentare in der Konfiguration geben an, welcher Service mithilfe der Regeln mit einer Ratenbeschränkung betrieben wird.

> **Anmerkung:** Wenn Administratoren einen anderen Port verwenden oder den Zugriff auf mehrere Server über die gleichen Ports erweitern, muss die entsprechende Dimensionierung für diese Ports entsprechend diesen Zahlen vorgenommen werden.

```
## 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-mask 32 --connlimit-saddr -m limit --limit 1/min --limit-burst 1
-j LOG --log-prefix " Connections to 8445 exceeded connlimit " -A INPUT -p tcp -m tcp --dport
8445 --tcp-flags SYN SYN -m connlimit --connlimit-above 10 --connlimit-mask 32 --connlimit-saddr
-j DROP -A INPUT -p tcp -m tcp --dport 8445 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto
6/sec --hashlimit-burst 8 --hashlimit-mode srcip,dstport --hashlimit-name TCP_8445_DOS -j ACCEPT
-A INPUT -p tcp -m tcp --dport 8445 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -
```

```
j LOG --log-prefix " Exceeded 8445 hashlimit " -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-mask 32 --connlimit-saddr -m limit --
limit 1/min --limit-burst 1 -j LOG --log-prefix " IdS connection limit exceeded" -A INPUT -p tcp
-m tcp --dport 8553 --tcp-flags SYN SYN -m connlimit --connlimit-above 6 --connlimit-mask 32 --
connlimit-saddr -j DROP -A INPUT -p tcp -m tcp --dport 8553 --tcp-flags SYN SYN -m hashlimit --
hashlimit-upto 2/sec --hashlimit-burst 4 --hashlimit-mode srcip,dstport --hashlimit-name
TCP_8553_DOS -j ACCEPT -A INPUT -p tcp -m tcp --dport 8553 --tcp-flags SYN SYN -m limit --limit
1/min --limit-burst 1 -j LOG --log-prefix " Exceeded 8553 hashlimit " -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-mask 32 --connlimit-
saddr -m limit --limit 1/min --limit-burst 1 -j LOG --log-prefix " IdP connection limit
exceeded" -A INPUT -p tcp -m tcp --dport 443 --tcp-flags SYN SYN -m connlimit --connlimit-above
8 --connlimit-mask 32 --connlimit-saddr -j DROP -A INPUT -p tcp -m tcp --dport 443 --tcp-flags
SYN SYN -m hashlimit --hashlimit-upto 4/sec --hashlimit-burst 6 --hashlimit-mode srcip,dstport -
-hashlimit-name TCP_443_DOS -j ACCEPT -A INPUT -p tcp -m tcp --dport 443 --tcp-flags SYN SYN -m
limit --limit 1/min --limit-burst 1 -j LOG --log-prefix " Exceeded 443 hashlimit " -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 scenarios. # Configuration for IMNP 5280 port -A
INPUT -p tcp -m tcp --dport 5280 --tcp-flags SYN SYN -m connlimit --connlimit-above 30 --
connlimit-mask 32 --connlimit-saddr -m limit --limit 1/min --limit-burst 1 -j LOG --log-prefix "
IMNP connection limit exceeded" -A INPUT -p tcp -m tcp --dport 5280 --tcp-flags SYN SYN -m
connlimit --connlimit-above 30 --connlimit-mask 32 --connlimit-saddr -j DROP -A INPUT -p tcp -m
tcp --dport 5280 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 20/sec --hashlimit-burst 25 -
-hashlimit-mode srcip,dstport --hashlimit-name TCP_5280_DOS -j ACCEPT -A INPUT -p tcp -m tcp --
dport 5280 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG --log-prefix "
Exceeded 5280 hashlimit " -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-mask 32 --connlimit-saddr -m limit --limit 1/min --
limit-burst 1 -j LOG --log-prefix " IMNP connection limit exceeded" -A INPUT -p tcp -m tcp --
dport 15280 --tcp-flags SYN SYN -m connlimit --connlimit-above 30 --connlimit-mask 32 --
connlimit-saddr -j DROP -A INPUT -p tcp -m tcp --dport 15280 --tcp-flags SYN SYN -m hashlimit --
hashlimit-upto 20/sec --hashlimit-burst 25 --hashlimit-mode srcip,dstport --hashlimit-name
TCP_15280_DOS -j ACCEPT -A INPUT -p tcp -m tcp --dport 15280 --tcp-flags SYN SYN -m limit --
limit 1/min --limit-burst 1 -j LOG --log-prefix " Exceeded 15280 hashlimit " -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-mask
32 --connlimit-saddr -m limit --limit 1/min --limit-burst 1 -j LOG --log-prefix " IMNP
connection limit exceeded" -A INPUT -p tcp -m tcp --dport 25280 --tcp-flags SYN SYN -m connlimit
--connlimit-above 30 --connlimit-mask 32 --connlimit-saddr -j DROP -A INPUT -p tcp -m tcp --
dport 25280 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 20/sec --hashlimit-burst 25 --
hashlimit-mode srcip,dstport --hashlimit-name TCP_25280_DOS -j ACCEPT -A INPUT -p tcp -m tcp --
dport 25280 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG --log-prefix "
Exceeded 25280 hashlimit " -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-mask 32 --connlimit-saddr -m limit --limit 1/min --
limit-burst 1 -j LOG --log-prefix " CUIC connection limit exceeded" -A INPUT -p tcp -m tcp --
dport 8444 --tcp-flags SYN SYN -m connlimit --connlimit-above 6 --connlimit-mask 32 --connlimit-
saddr -j DROP -A INPUT -p tcp -m tcp --dport 8444 --tcp-flags SYN SYN -m hashlimit --hashlimit-
upto 2/sec --hashlimit-burst 4 --hashlimit-mode srcip,dstport --hashlimit-name TCP_8444_DOS -j
ACCEPT -A INPUT -p tcp -m tcp --dport 8444 --tcp-flags SYN SYN -m limit --limit 1/min --limit-
burst 1 -j LOG --log-prefix " Exceeded 8444 hashlimit " -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-mask 32 --connlimit-saddr -m
limit --limit 1/min --limit-burst 1 -j LOG --log-prefix " CUIC connection limit exceeded" -A
INPUT -p tcp -m tcp --dport 8447 --tcp-flags SYN SYN -m connlimit --connlimit-above 6 --
connlimit-mask 32 --connlimit-saddr -j DROP -A INPUT -p tcp -m tcp --dport 8447 --tcp-flags SYN
SYN -m hashlimit --hashlimit-upto 2/sec --hashlimit-burst 4 --hashlimit-mode srcip,dstport --
hashlimit-name TCP_8447_DOS -j ACCEPT -A INPUT -p tcp -m tcp --dport 8447 --tcp-flags SYN SYN -m
limit --limit 1/min --limit-burst 1 -j LOG --log-prefix " Exceeded 8447 hashlimit " -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-mask 32 --connlimit-saddr -m limit --limit 1/min --limit-burst 1 -j LOG --log-prefix "
```

```
LD connection limit exceeded" -A INPUT -p tcp -m tcp --dport 12005 --tcp-flags SYN SYN -m
connlimit --connlimit-above 10 --connlimit-mask 32 --connlimit-saddr -j DROP -A INPUT -p tcp -m
tcp --dport 12005 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 6/sec --hashlimit-burst 8 --
hashlimit-mode srcip,dstport --hashlimit-name TCP_12005_DOS -j ACCEPT -A INPUT -p tcp -m tcp --
dport 12005 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG --log-prefix "
Exceeded 12005 hashlimit " -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-mask 32 --connlimit-saddr -m limit --limit 1/min --
limit-burst 1 -j LOG --log-prefix " LD connection limit exceeded" -A INPUT -p tcp -m tcp --dport
12008 --tcp-flags SYN SYN -m connlimit --connlimit-above 10 --connlimit-mask 32 --connlimit-
saddr -j DROP -A INPUT -p tcp -m tcp --dport 12008 --tcp-flags SYN SYN -m hashlimit --hashlimit-
upto 6/sec --hashlimit-burst 8 --hashlimit-mode srcip,dstport --hashlimit-name TCP_12008_DOS -j
ACCEPT -A INPUT -p tcp -m tcp --dport 12008 --tcp-flags SYN SYN -m limit --limit 1/min --limit-
burst 1 -j LOG --log-prefix " Exceeded 12008 hashlimit " -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
```

Diese Regeln können direkt angewendet werden, indem Sie die **/etc/sysconfig/iptables** manuell bearbeiten oder die Konfiguration in einer Datei wie **iptables.conf** speichern und **die Regeln** mithilfe von cat **iptables.conf >>/etc/sysconfig/iptables** ausführen.

Nach Anwendung der Regeln ist ein Neustart des IPtables-Dienstes erforderlich. Eingabe **systemctl restart iptables** um den IPtables-Dienst neu zu starten.

Client-Verbindungen einschränken

Zusätzlich zur vorherigen Konfiguration von IPtables werden Installationen, die den Adressbereich für Clients, die den Proxy verwenden, kennen, empfohlen, dieses Wissen zu verwenden, um die Proxyzugriffsregeln zu sichern. Dies kann zu enormen Kosten führen, wenn es darum geht, den Proxy vor Botnets schädlicher Netzwerke zu schützen, die häufig in Ländern mit weniger strengen Regeln für die Online-Sicherheit erstellt werden, die IP-Adressen umfassen. Es wird daher dringend empfohlen, die IP-Adressbereiche auf Länder-/Bundesland- oder ISP-basierte IP-Bereiche zu beschränken, wenn Sie sich über die Zugriffsmuster im Klaren sind.

Client-Verbindungen blockieren

Außerdem ist es hilfreich zu wissen, wie ein bestimmter Adressbereich blockiert wird, wenn ein Angriff anhand einer IP-Adresse oder eines Bereichs von IP-Adressen erkannt wird. In solchen Fällen können die Anforderungen dieser IP-Adressen mit **unzulässigen** Regeln blockiert werden.

Unterschiedliche IP-Adressen blockieren

Um mehrere unterschiedliche IP-Adressen zu blockieren, fügen Sie der Konfigurationsdatei **IPTables** für jede IP-Adresse eine Zeile hinzu.

Um beispielsweise die Adressen **192.0.2.3** und **192.0.2.4** zu blockieren, geben Sie Folgendes ein:

```
iptables -A INPUT -s 192.0.2.3 -j DROP iptables -A INPUT -s 192.0.2.4 - j DROP.
```

Blockieren eines Bereichs von IP-Adressen

Blockieren Sie mehrere IP-Adressen in einem Bereich, und fügen Sie der **IPTables-**Konfigurationsdatei mit dem IP-Bereich eine einzelne Leitung hinzu.

Um beispielsweise Adressen von 192.0.2.3 bis 192.0.2.35 zu blockieren, geben Sie Folgendes ein:

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

Alle IP-Adressen in einem Subnetz sperren

Blockieren Sie alle IP-Adressen in einem gesamten Subnetz, indem Sie der **IPTables-**Konfigurationsdatei eine einzelne Leitung hinzufügen, wobei für den IP-Adressbereich die klassenlose Routing-Notation zwischen den Domänen verwendet wird. Um z. B. alle Klasse-**C-Adressen** zu blockieren, geben Sie Folgendes ein:

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

SELinux

SELinux ist ein Plattform-Sicherheits-Framework, das als Erweiterung des Linux-Betriebssystems integriert ist. Das Verfahren zum Installieren und Hinzufügen von SELinux-Richtlinien, um OpenResty als umgekehrten Proxy auszuführen, wird als Nächstes bereitgestellt.

1. Stoppen Sie den Vorgang mit dem **openresty -s stop** aus.
2. Konfigurieren und Starten des /stop nginx-Servers mit dem **systemctl** , sodass der OpenResty-Prozess beim Hochfahren automatisch gestartet wird. Geben Sie diese Befehle als root ein. Wechseln Sie zu **/usr/lib/systemd/system**.Öffnen Sie die Datei **openresty.service**.Aktualisieren Sie den Inhalt der Datei entsprechend dem **PIDFile-**Speicherort.
   ```
   [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
   ```
   Als root-Benutzer geben Sie **sudo systemctl enable openresty**.Starten / Beenden des OpenResty-Dienstes mit dem **systemctl start openresty / systemctl stop openresty** -Befehl ein, und stellen Sie sicher, dass der Prozess als Root-Benutzer gestartet/beendet wird.

1. **Selinux installieren** Standardmäßig werden nur einige SELinux-Pakete in CentOs installiert.Das **policycoreutils-devel-**Paket und seine Abhängigkeiten müssen installiert werden, um die SELinux-Richtlinie zu generieren.Geben Sie diesen Befehl ein, um **policycoreutils-devel** zu installieren.
   ```
   yum install policycoreutils-devel
   ```
   Stellen Sie sicher, dass nach der Installation des Pakets **sepolicy** funktioniert.
   ```
   usage: sepolicy [-h] [-P POLICY]

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

```
                ...

SELinux Policy Inspection Tool
```

2. **Erstellen eines neuen Linux-Benutzers und Zuordnen zu einem SElinux-Benutzer**Eingabe **semanage login -l** um die Zuordnung zwischen Linux-Benutzern und SELinux-Benutzern anzuzeigen.

```
[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       *
```

Erstellen Sie als root einen neuen Linux-Benutzer (**nginx**-Benutzer), der dem SELinux **user_u-**Benutzer zugeordnet ist.

```
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.
```

Um die Zuordnung zwischen **nginxuser** und **user_u** anzuzeigen, geben Sie den folgenden Befehl als root ein:

```
[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       *
```

SELinux **__default_**login ist standardmäßig dem **Benutzer** SELinux **unrestricted_u** zugeordnet. Es ist erforderlich, **user_u** standardmäßig mit dem folgenden Befehl zu begrenzen:

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

Um zu überprüfen, ob der Befehl ordnungsgemäß funktioniert hat, geben Sie **semanage login -l**. Diese Ausgabe sollte erstellt
werden:

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

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

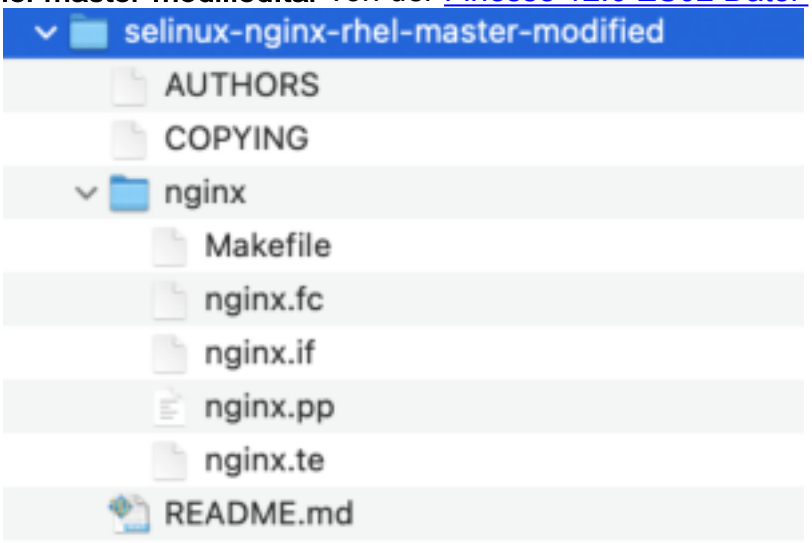
Ändern Sie nginx.conf, und führen Sie die Besitzänderung für nginxuser durch.Eingabe **chown -R nginxuser:nginxuser** * im Verzeichnis **<Openresty-install-directory>**.Ändern Sie die Datei **nginx.conf**, um nginxuser als Benutzer für die Ausführung von Workerprozessen einzuschließen.

```
........

user nginxuser nginxuser;

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

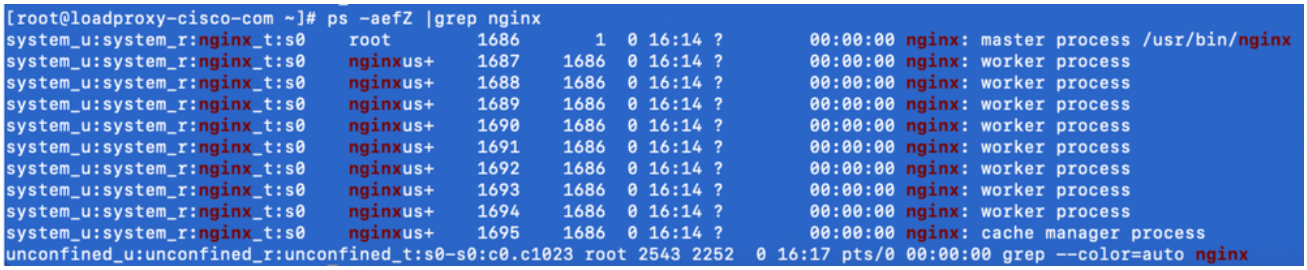
Schreiben der SELinux-Richtlinie für Nginx

1. Anstatt eine neue benutzerdefinierte Standardrichtlinie für Nginx mit dem **sepolicy generate --init /usr/bin/nginx** verwenden, sollte mit einer bestehenden Richtlinie begonnen werden.
2. Die Dateien **nginx.fc** (Datei Dateikontexte) und **nginx.te** (Datei zur Typdurchsetzung), die von

der angegebenen URL heruntergeladen werden, wurden so geändert, dass sie der Verwendung des umgekehrten Proxys entsprechen.
3. Diese geänderte Version kann als Referenz verwendet werden, da sie für den jeweiligen Anwendungsfall festgelegt wurde.
4. Laden Sie die Datei **selinux-nginx-rhel-master-modified.tar** von der Finesse 12.6 ES02 Datei-Software-Download-Seite herunter.



5. Extrahieren Sie die .tar-Datei, und navigieren Sie zum Verzeichnis **nginx** darin.
6. Öffnen Sie die **.fc**-Datei, und überprüfen Sie die erforderlichen Dateipfade für die Datei nginx installer, cache und pid.
7. Kompilieren Sie die Konfiguration mit dem **make** aus.
8. Die Datei **nginx.pp** wird generiert.
9. Laden Sie die Richtlinie mit dem **semodule** aus.
   ```
   semodule -i nginx.pp
   ```
10. Wechseln Sie zu **/root**, und erstellen Sie eine leere Datei mit dem Namen **touch /.autorelabel**.
11. Starten Sie das System neu.
12. Geben Sie diesen Befehl ein, um zu überprüfen, ob die Richtlinie erfolgreich geladen wurde.
    ```
    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 sollte ohne Verletzung ausgeführt werden. (Verstöße sind in **/var/log/messages** und **/var/log/audit/audit.log** verfügbar).
14. Geben Sie diesen Befehl ein, um den Status von Nginx zu überprüfen.
    ```
    ps -aefZ | grep nginx
    ```



15. Jetzt sollte der Zugriff auf den Finesse Agent-/Supervisor-Desktop möglich sein.

Überprüfung

In diesem Abschnitt überprüfen Sie, ob Ihre Konfiguration ordnungsgemäß funktioniert.

Finesse

1. https://<reseproxy:port>/finesse/api/SystemInfo anfordern. aus der DMZ und überprüfen Sie, ob sie erreichbar sind.
2. Die **<host>**-Werte in **<primaryNode>** und **<sekundärNode>** sind gültige umgekehrte Proxy-Hostnamen. Es sollte keine Finesse-Hostnamen sein.

CUIC- und Live-Daten

1. Wenn in der Antwort statt umgekehrter Proxy-Hostnamen die Finesse-Hostnamen aufgeführt sind, validieren Sie die Proxy-Zuordnungskonfigurationen, und zulässige Hosts werden ordnungsgemäß in Finesse-Servern hinzugefügt. Dies wird im Abschnitt "Auffüllen von Netzwerkübersetzungsdaten" von "VPN-Less Access to Finesse Desktop" im Finesse 12.6 UCCE Feature Guide beschrieben.
2. Wenn LiveData-Gadgets im Finesse Desktop ordnungsgemäß geladen werden, sind die CUIC- und LiveData-Proxy-Konfigurationen korrekt.
3. Um die CUIC- und LiveData-Konfiguration zu validieren, müssen Sie HTTP-Anfragen an diese URLs von der DMZ senden und überprüfen, ob diese erreichbar sind. https://<reseproxy:cuic_port>/cuic/rest/abouthttps://<reseproxy:ldweb_port>/livedata/security https://<Reverseproxy:ldsocketio_port>/security

IDs

So validieren Sie die IDs-Konfiguration:

1. Melden Sie sich bei der IDSAdmin-Schnittstelle unter **https://<ids_LAN_host:ids_port>:8553/idsadmin** vom LAN an, da die Admin-Schnittstelle nicht über den Reverse Proxy verfügbar ist.
2. Wählen Sie **Einstellungen > IDs Trust**.
3. Überprüfen Sie, ob der Proxycluster-Herausgeberknoten auf der Seite SP-Metadaten herunterladen aufgeführt ist, und klicken Sie auf **Weiter**.
4. Überprüfen Sie, ob der IDP-Proxy korrekt angezeigt wird, wenn er auf der Seite "IDP-Metadaten hochladen" konfiguriert ist, und klicken Sie auf **Weiter**.
5. Initiieren Sie die Test-SSO über alle Proxycluster-Knoten auf der Seite Test SSO, und validieren Sie, ob alle erfolgreich sind. Dies erfordert die Verbindung des Client-Computers,

um Proxyknoten rückgängig zu machen.

Leistung

Die Datenanalyse der mit dem NMON-Tool erstellten Erfassung der Spitzenleistung ist auf der Download-Seite der Finesse Release 12.6(1)ES2 Software verfügbar (**load_result.zip**). Die Daten stellen den Status des Proxys für Desktop- und Supervisor-Vorgänge in einer Beispielbereitstellung für 2000 UCCE dar, bei der SSO-Anmeldungen und CUIC LD-Berichte verwendet werden, wie im Standardlayout für 2.000 Benutzer für einen Zeitraum von acht Houts konfiguriert. Sie kann verwendet werden, um die Computing-, Festplatten- und Netzwerkanforderungen für eine Installation mithilfe von Nginx auf vergleichbarer Hardware abzuleiten.

Fehlerbehebung

SELinux

1. Wenn Nginx nicht standardmäßig gestartet wird oder der Finesse Agent-Desktop nicht zugänglich ist, stellen Sie mit dem folgenden Befehl den **Permissive**-Modus für SELinux ein:
   ```
   setenforce 0
   ```
2. Versuchen Sie, den Nginx mit dem **systemctl restart nginx** aus.
3. Verstöße sind in **/var/log/messages** und **/var/log/audit/audit.log** verfügbar.
4. Es ist erforderlich, die .te-Datei neu zu generieren, wobei Regeln zur Behandlung dieser Verletzungen durch einen der folgenden Befehle verwendet werden können:
   ```
   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. Aktualisieren Sie die ursprüngliche **nginx.te**-Datei im **selinux-nginx-rhel-master-modifizierten/nginx-**Verzeichnis mit neu generierten Zulassungsregeln.
6. Gleiches mit dem **make** aus.
7. Die Datei nginx.pp wird neu generiert.
8. Laden Sie die Richtlinie über den semodul-Befehl.
   ```
   semodule -i nginx.pp
   ```
9. Wählen Sie mit diesem Befehl den **Erzwingungsmodus für** SELinux aus:
   ```
   setenforce
   ```
10. Starten Sie das System neu.
11. Wiederholen Sie dieses Verfahren, bis die erforderlichen Verstöße behoben sind.