

配置Nginx反向代理，以便無VPN訪問Cisco Finesse (12.6 ES03)

目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[背景資訊](#)

[ES03中的更改](#)

[基於ES01的無VPN配置的升級說明](#)

[驗證](#)

[非SSO驗證](#)

[SSO驗證](#)

[Websocket連線的驗證](#)

[暴力攻擊預防](#)

[記錄](#)

[安裝和配置Fail2ban](#)

[驗證靜態資源URL](#)

[快取CORS標頭](#)

[設定](#)

[配置無VPN訪問的解決方案元件](#)

[在DMZ中安裝OpenResty作為反向代理](#)

[OpenResty安裝](#)

[配置Nginx](#)

[配置Nginx快取](#)

[配置SSL證書](#)

[使用自訂Diffie-Hellman引數](#)

[確保啟用OCSP裝訂-證書吊銷檢查](#)

[Nginx配置](#)

[配置反向代理埠](#)

[配置反向代理和上游元件之間的雙向TLS驗證](#)

[清除快取](#)

[標準準則](#)

[配置對映檔案](#)

[使用反向代理作為對映檔案伺服器](#)

[CentOS 8核心強化](#)

[IPtables強化](#)

[限制客戶端連線](#)

[阻止客戶端連線](#)

[阻止不同的IP地址](#)

[阻塞IP地址範圍](#)

[阻止子網中的所有IP地址](#)

[SELinux](#)

驗證

[Finse](#)

[CUIC和即時資料](#)

[IDS](#)

[效能](#)

疑難排解

[SSO](#)

簡介

本文檔介紹如何使用反向代理訪問Cisco Finesse案頭而不連線到基於Cisco Finesse、Cisco Unified Intelligence Center (CUIC)和Cisco Identity Service (Id)的12.6 ES03版本的VPN。



註：Cisco不支援Nginx安裝和配置。有關此主題的查詢可在[思科社群論壇](#)上討論。



注意：對於無VPN的ES03部署，請參閱各個元件的自述檔案以規劃升級並檢查相容性限制。
[Cisco Finesse 12.6 ES03 Readme](#)、[CUIC/IdS 12.6 ES03 Readme](#)

必要條件

需求

思科建議您瞭解以下主題：

- 思科整合客服中心企業版(UCCE)版本
- Cisco Finesse
- Linux管理
- 網路管理與Linux網路管理


採用元件

本文中的資訊係根據以下軟體和硬體版本：

- Finesse - 12.6 ES03
- CUIC - 12.6 ES03
- IdS - 12.6 ES03
- 適用於客服中心(CC)的UCCE/託管合作解決方案(HCS) - 11.6或更高版本
- 套裝客服中心企業版(PCCE) - 12.5或更高版本

注意：由於LD/CUIC共存部署，PCCE/UCCE 2k部署將需要在CCE 12.6版本上進行

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

 注意：針對一個2000使用者UCCE部署示例，本文檔中提供的配置已透過在CentOS 8.0上部署的Nginx反向代理(OpenResty)進行配置、強化以及負載測試。效能配置檔案參考資訊可在本文檔中找到。

背景資訊

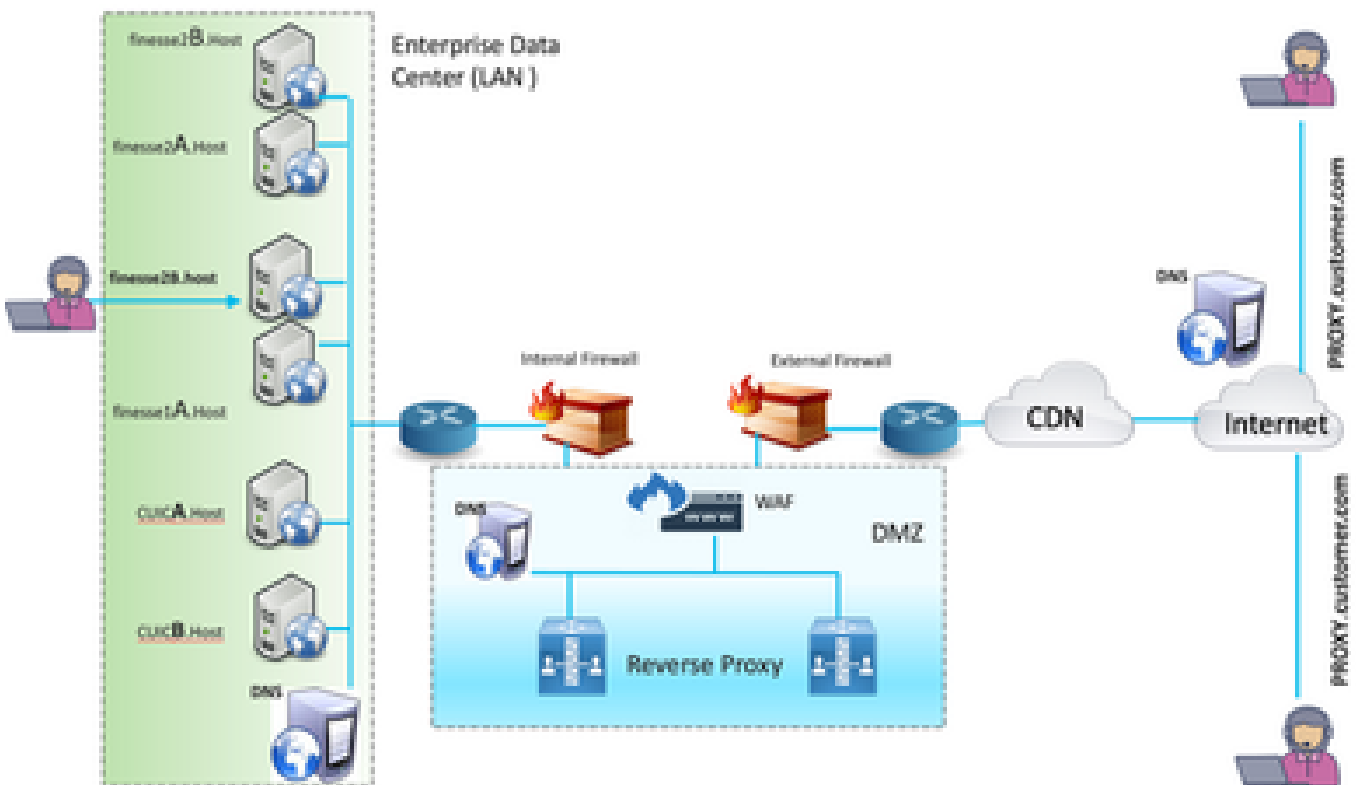
UCCE/PCCE和HCS的UCCE解決方案支援此部署模式。


支援部署反向代理（從12.6 ES01可用），作為無需連線到VPN即可訪問Cisco Finesse案頭的選項。此功能使座席能夠透過Internet從任何位置訪問Finesse案頭。

要啟用此功能，必須在隔離區(DMZ)中部署反向代理對。

在反向代理部署中，介質訪問保持不變。要連線到媒體，代理可以使用移動和遠端訪問思科 Jabber解決方案(MRA)或UCCE的移動代理功能，以及公共交換電話網路(PSTN)或移動終端。此圖顯示當您透過單一高可用性(HA)對反向代理節點訪問兩個Finesse集群和兩個CUIC節點時的網路部署外觀。

支援從Internet上的代理和從LAN連線的代理進行併發訪問，如下圖所示。



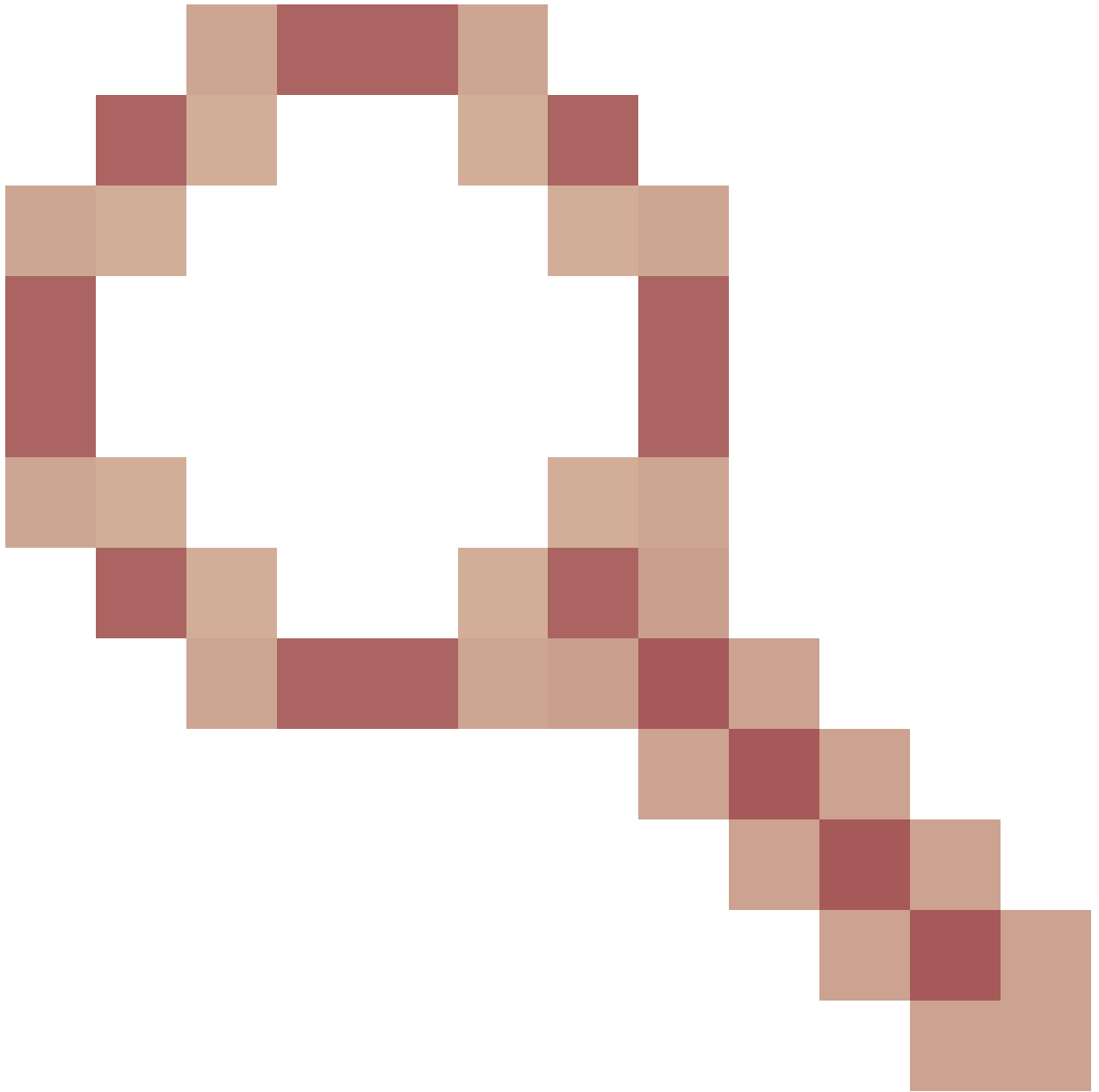
 注意：請參閱第三方代理選擇標準的功能指南以代替Nginx支援此部署。

- [UCCE 12.6功能指南](#) -提供無VPN功能的功能概述、設計以及[配置詳細資訊](#)。
- [UCCE 12.6安全指南](#) -提供反向代理部署的安全配置指南。

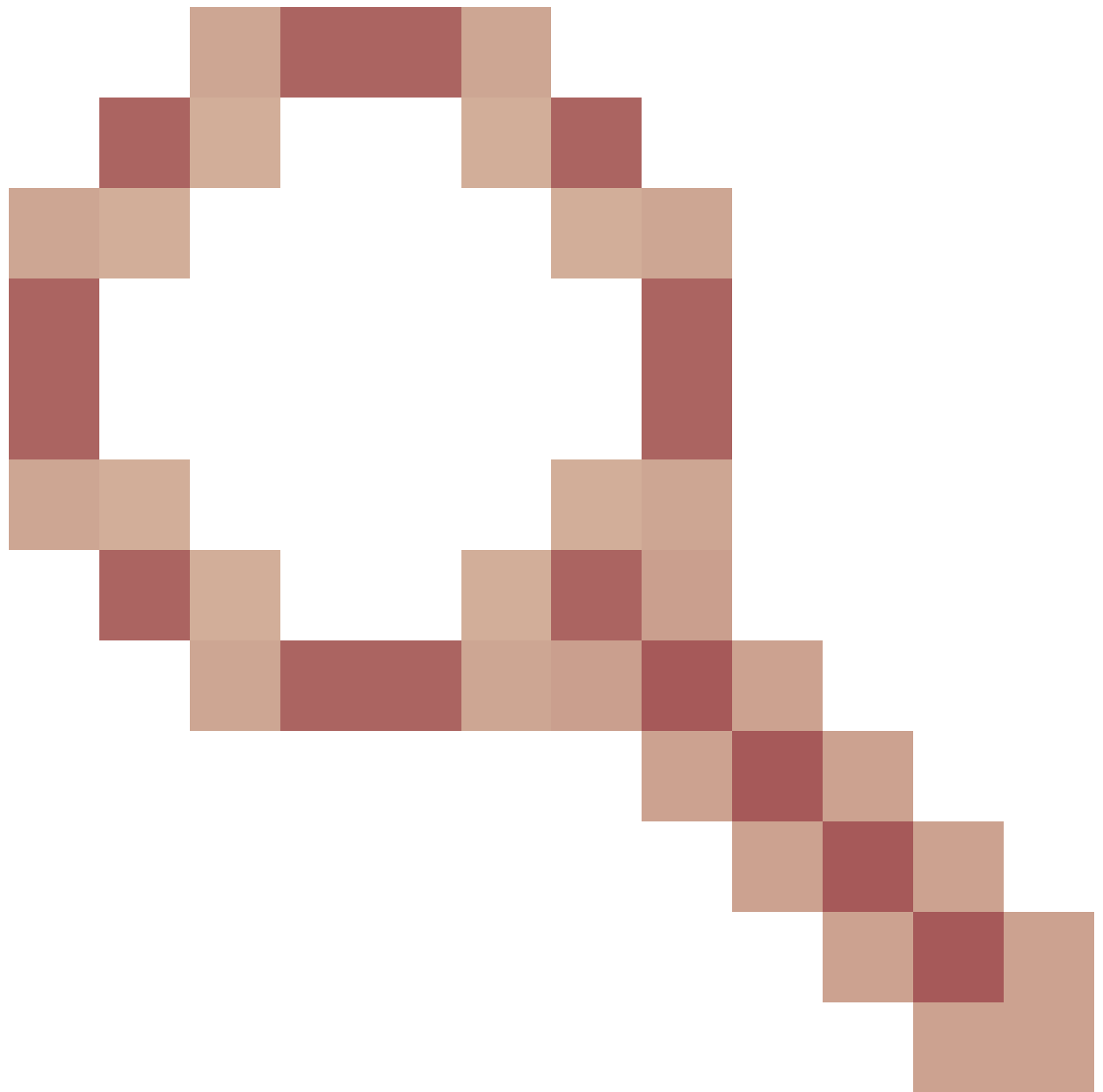
在閱讀本文檔之前，建議參閱功能指南和安全指南中的無VPN部分。

ES03中的更改

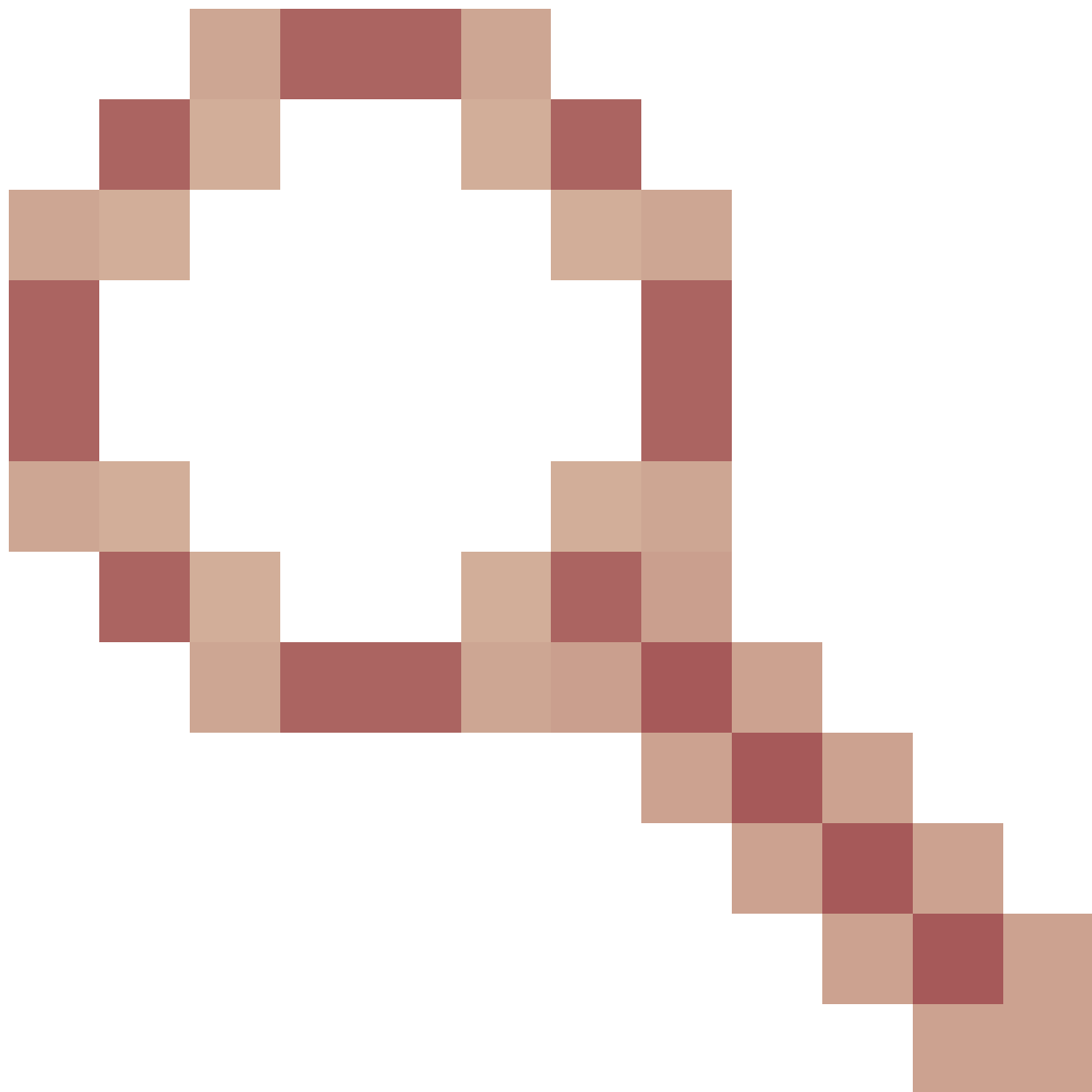
- 新功能
 - 現在透過反向代理支援Finesse Supervisor功能。
 - CUIC RealTime和歷史報告現在透過代理環境中的Finesse小工具支援。
 - 對所有請求/通訊進行身份驗證-需要Lua支援
 - 所有Finesse/CUIC/IM & Presence (IM&P)請求在允許進入資料中心之前均在Proxy上進行身份驗證。
 - Websocket和Live data socketIO連線也受到限制，並且僅允許來自已成功向Finesse發出安全請求的客戶端的連線。
 - 代理上的暴力攻擊感應和日誌記錄，可與Fail2Ban配合使用來阻止惡意IP地址。
- 反向代理配置的安全增強功能-需要Lua支援
 - 反向代理和上游元件(Finesse/IdS/CUIC/Livedata)之間的雙向傳輸層安全(TLS)身份驗證。
 - SeLinux設定。
 - 為代理伺服器和元件伺服器請求啟用相互安全套接字層(SSL)信任驗證。
- 增強代理配置的安全性，以防止拒絕服務(DoS)/分散式拒絕服務(DDoS)攻擊-需要Lua支援
 - 增強的系統各個部分的Nginx請求速率限制。
 - IpTables的速率限制。
 - 在請求上游元件伺服器之前驗證靜態資源請求。
 - 未命中上游元件伺服器的較輕且可快取的未驗證頁面。
- 其他其他功能-需要Lua支援
 - 從代理提供的自動感應跨來源資源共用(CORS)響應有助於自動配置和提高效能
- 與無VPN相關的缺陷修復
 - [CSCwa26057](#)



[CSCwa26057](#)

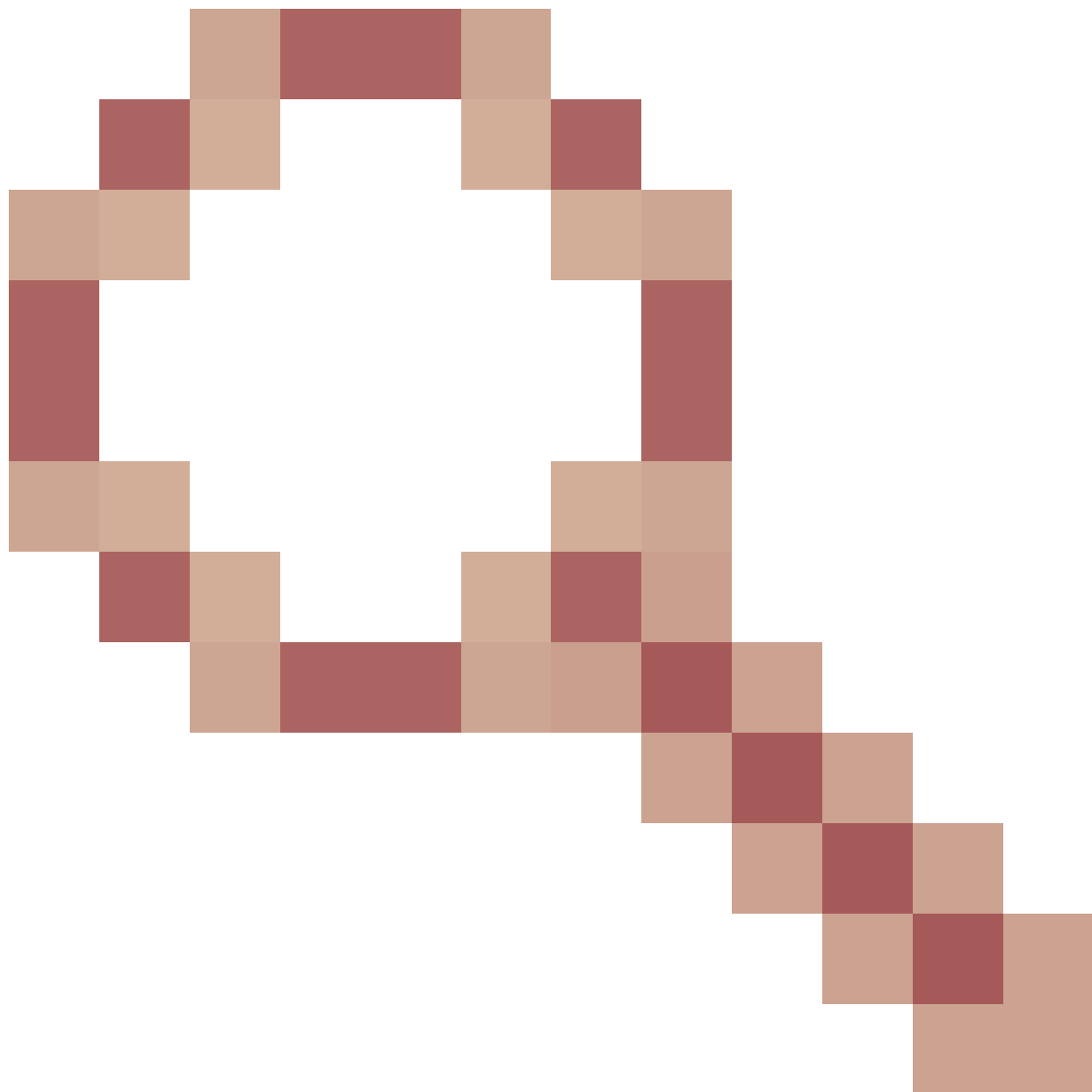


- 在 finesse 案頭登入期間，為座席提供的多個證書
[CSCwa24471](#)

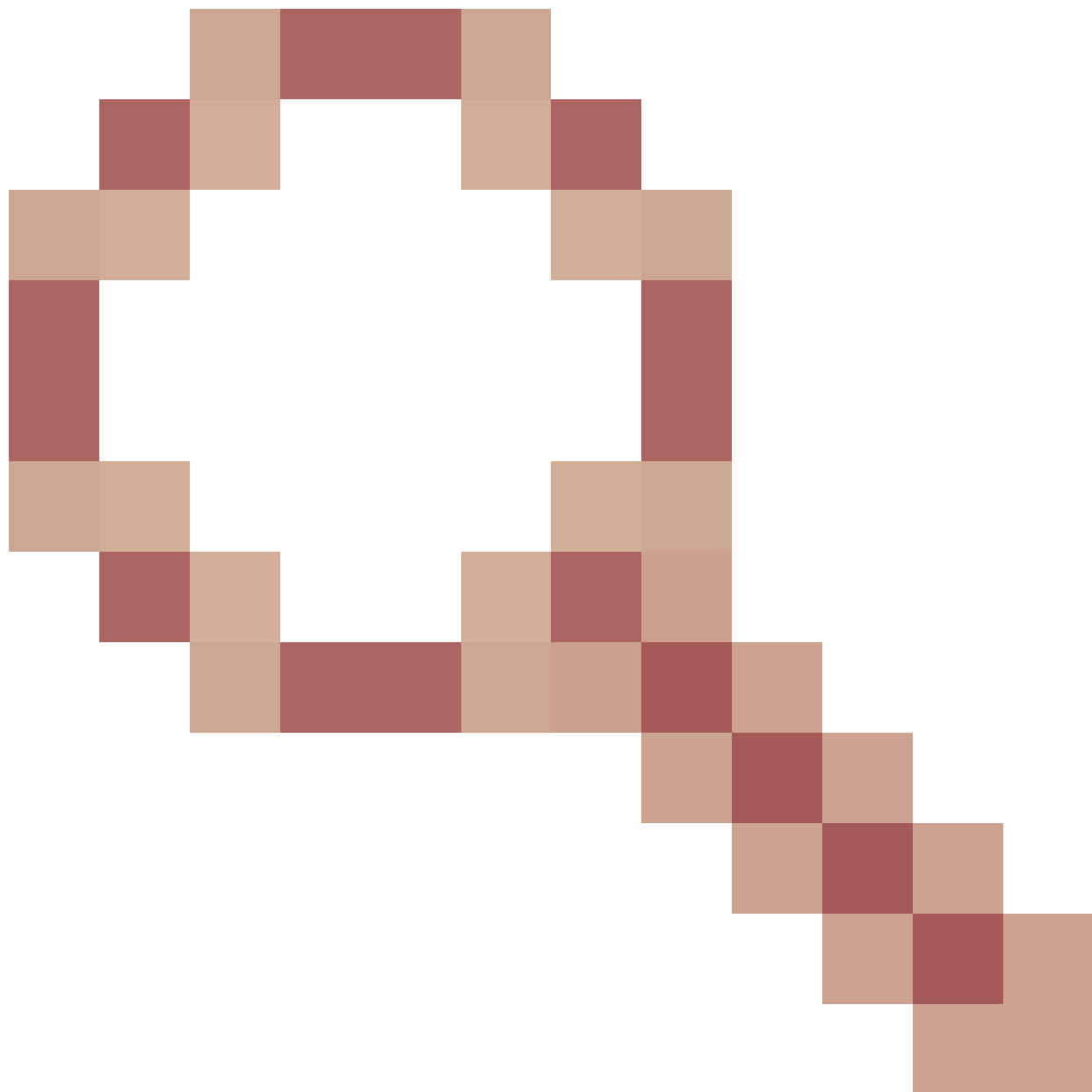


- Finesse登入頁不顯示SSO代理FQDN名稱

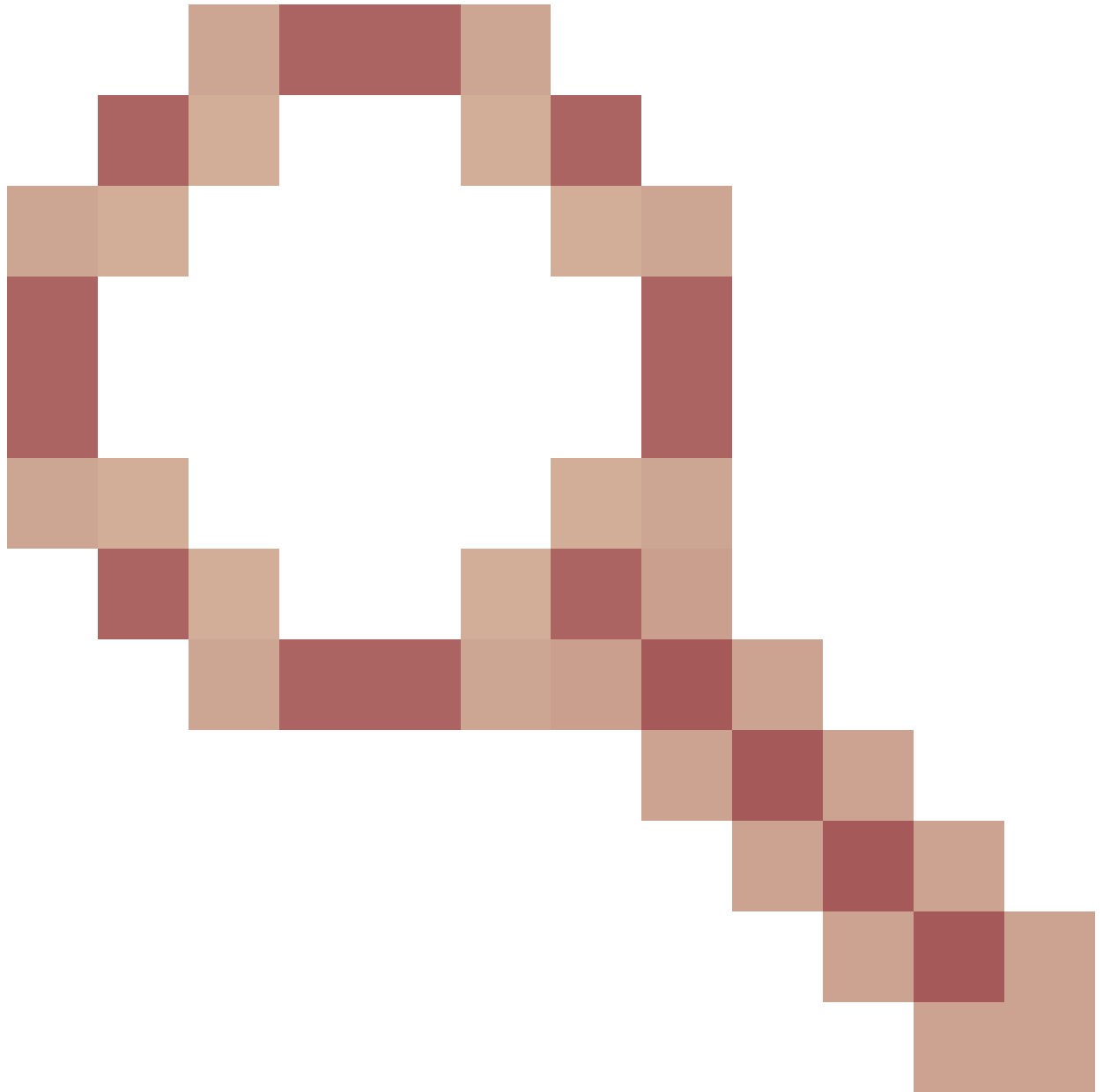
- [CSCwa24519](#)



：如果無法從元件解析反向代理主機名，則Web代理服務無法重新啟動
◦ [CSCwa23252](#)




：當CA證書鏈的深度大於一個時，代理Finesse信任被破壞
◦ [CSCwa46459](#)



log4j zero day漏洞在webservice中暴露

基於ES01的無VPN配置的升級說明

- ES03配置需要安裝Nginx並支援Lua。
- 憑證需求
 - 在Nginx ES02配置能夠成功連線到上游伺服器之前，Cisco Finesse、CUIC和Id需要將Nginx/OpenResty主機證書增加到Tomcat信任庫並重新啟動。
 - 需要在Nginx伺服器中配置Cisco Finesse、CUIC和IdS上游伺服器證書，以使用基於ES03的配置。

 注意：建議在安裝ES03 Nginx配置之前刪除基於ES01的現有Nginx配置。

 注意：ES03配置指令碼還需要在Cisco Finesse、CUIC和Id中安裝相應的ES03 COP。

驗證

Finesse 12.6 ES03在代理上引入身份驗證。單一登入(SSO)和非SSO部署支援驗證。

在將請求和協定轉發到上游元件伺服器之前，對代理處接受的所有請求和協定強制執行身份驗證，而本地由元件伺服器實施的身份驗證也在上游元件伺服器中進行。所有身份驗證都使用通用的Finesse登入憑證對請求進行身份驗證。

持久連線，例如依賴於可擴展消息傳送和線上狀態協定(XMPP)等應用協定進行身份驗證和後續連線的網路插座，透過在建立套接字連線之前驗證從中成功進行應用身份驗證的IP地址而在代理處進行身份驗證。

非SSO驗證

非SSO身份驗證不需要任何額外配置，並且在完成所需的指令碼替換後，將使用開箱即用的Nginx配置指令碼。身份驗證依賴於用於登入Finesse的使用者名稱和密碼。對所有端點的訪問都將透過Finesse身份驗證服務進行驗證。

有效使用者清單在本機代理處快取（每15分鐘更新一次快取），用於驗證請求中的使用者。透過將請求轉發到配置的Finesse URI來驗證使用者憑證，然後憑證雜湊被本地快取（快取了15分鐘）以驗證本地的新請求。如果使用者名稱或密碼有任何變更，則僅會在15分鐘後生效。

SSO驗證

SSO身份驗證要求管理員在配置檔案中的Nginx伺服器上配置IdS令牌加密金鑰。使用show ids secret CLI命令可以從IdS伺服器獲取IdS令牌加密金鑰。這些金鑰必須配置為管理員必須在指令碼中執行的一#Must-change替代的一部分，SSO身份驗證才能正常工作。

請參閱SSO使用手冊，瞭解為代理解析執行IdS而要執行的IdS SAML配置。

配置SSO身份驗證後，可以使用一對有效的令牌訪問系統中的任何終端。代理配置透過擷取對IdS發出的令牌檢索請求或解密有效的令牌並隨後本地快取它們以進行進一步驗證，來驗證憑證。

Websocket連線的驗證

無法使用標準授權標頭驗證Websocket連線，因為瀏覽器中的原生Websocket實作不支援自訂標頭。應用層級驗證協定，其中負載中包含的驗證資訊不會阻止Web套接字連線的建立，因此惡意實體僅透過建立大量連線來淹沒系統，即可呈現DOS或DDOS攻擊。

為了降低這種可能性，所提供的nginx反向代理配置具有特定的檢查，只允許從在建立Websocket連線之前成功發出經過驗證的REST請求的IP地址接受Websocket連線。這表示嘗試建立Web通訊端連線的使用者端在發出REST要求之前，現在會收到授權失敗的錯誤，而且不支援使用案例。

暴力攻擊預防

Finesse 12.6 ES02身份驗證指令碼主動阻止可用於猜測使用者密碼的暴力攻擊。在短時間內嘗試一定數量的失敗嘗試後，它會阻止用於訪問服務的IP地址。這些請求將因418客戶端錯誤而被拒絕。可以從檔案<nginx-install-directory>/logs/blocking.log和<nginx-install-directory>/logs/error.log訪問被阻止的IP地址的詳細資訊。

失敗請求數、時間間隔和阻止持續時間是可配置的。配置存在於<nginx-install-directory>/conf/conf.d/maps.conf檔案中。

```
## 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;
}
```

記錄

要查詢被阻止的IP地址，請從目錄<nginx-install-directory>/logs運行以下命令。

```
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"
```

建議客戶與Fail2ban或類似產品整合，以便將該禁令增加到IPtable/防火牆規則中。

安裝和配置Fail2ban

Fail2ban掃描日誌檔案並禁止顯示惡意標識的IP -密碼失敗過多、尋求利用漏洞等。通常，Fail2Ban用於更新防火牆規則，以在指定時間內拒絕IP地址，但也可以配置任何其它操作（例如傳送電子郵件）。有關詳細資訊，請訪問<https://www.fail2ban.org/>。

Fail2ban可以配置為監控blocking.log，以辨識在檢測到暴力攻擊時被Nginx阻止的IP地址，並在可配置的持續時間內禁止這些地址。在CentOS reverseproxy上安裝及設定fail2ban的步驟如下：

1. 使用yum安裝Fail2ban。

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

2. 建立本地監獄。

管理員可以透過監獄配置各種屬性，例如任何被阻止的IP地址禁止訪問的埠、IP地址被阻止的持續時間、用於從受監控的日誌檔案中標識被阻止的IP地址的過濾器配置等。增加自定義配置以禁止被阻止訪問上游伺服器的IP地址的步驟如下：

2.1. 轉至Fail2ban安裝目錄（在本例中為/etc/fail2ban）

```
cd /etc/fail2ban
```

2.2. 將jail.conf 副本複製到just.local中，以隔離對本地所做的更改。

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

2.3. 將這些Jail配置增加到檔案jail.local的末尾，並用實際埠替換模板中的埠。根據需要更新禁止時間配置。

```
# 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. 設定過濾條件。

過濾器告訴Fail2ban在日誌中查詢什麼以辨識要禁止的主機。建立過濾器的步驟如下：

3.1. 建立filter.d/finesseban.conf。

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

3.2. 將這些行增加到檔案filter.d/finesseban.conf中。

[Definition]

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

4. 啟動Fail2ban。

運行此命令以啟動fail2ban。

```
fail2ban-client start
```

打開fail2ban日誌檔案並驗證沒有任何錯誤。預設情況下，fail2ban的日誌會進入/var/log/fail2ban.log檔案。

驗證靜態資源URL

可以未經驗證的方式訪問的所有有效終端都會在ES03指令碼中主動跟蹤。

如果請求的URI無效，則對這些未經身份驗證的路徑的請求將被主動拒絕，而不會將這些請求傳送到上游伺服器。

快取CORS標頭

當第一個選項請求成功時，會在代理處快取響應標頭access-control-allow-headers、access-control-allow-origin、access-control-allow-methods、access-control-expose-headers和access-control-allow-credentials，並等待5分鐘。這些標頭會快取至每個對應的上游伺服器。

設定

本文檔介紹將Nginx配置為用於啟用Finesse VPN無訪問的反向代理的配置。提供了用於驗證所提供說明的UCCE解決方案元件、代理和作業系統版本。相關指示必須根據您選擇的作業系統/代理進行

調整。

- 使用的Nginx版本- OpenResty 1.19.9.1
- 用於配置的作業系統- CentOS 8.0

 注意：所述的Nginx配置可以從[Finesse版本12.6\(1\)ES3軟體下載頁面](#)下載。

配置無VPN訪問的解決方案元件

配置代理後，請務必使用計畫的主機名和IP配置解決方案元件(Finesse/CUIC/IdS)，以便透過VPN無訪問方式訪問解決方案。

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

有關這些命令的詳細資訊，請參閱[UCCE 12.6功能指南](#)，在使用本文檔之前應參閱該指南。


在DMZ中安裝OpenResty作為反向代理

本節詳細介紹基於OpenResty的代理安裝步驟。反向代理通常配置為網路隔離區(DMZ)中的專用裝置，如前面提到的部署圖所示。

1. 安裝您所選擇的作業系統，並符合所需的硬體規格。核心和IPv4引數調整可能因選擇的作業系統而異，如果選擇的作業系統版本不同，建議使用者重新驗證這些方面。
2. 配置兩個網路介面。從Internet客戶端進行公共訪問需要一個介面，與內部網路中的伺服器通訊需要一個介面。
3. 安裝[OpenResty](#)。

任何形式的Nginx都可以用於此目的，只要它們基於Nginx 1.19+並支援Lua：

- Nginx Plus
- Nginx開放源 (Nginx開放源需要與基於OpenResty的Lua模組一起編譯才能使用)
- OpenResty
- GetPageSpeed額外專案

 注意：提供的配置已經過OpenResty 1.19測試，預計只能透過少量更新 (如果有) 使用其他分發。

OpenResty安裝

1. 安裝OpenResty。請參閱[OpenResty Linux程式包](#)。作為OpenResty安裝的一部分，Nginx將安裝在此位置，並透過增加~/.bashrc檔案將OpenResty路徑增加到PATH變數。

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

2. 啟動/停止Nginx。

- 要啟動Nginx，請輸入openresty。
- 若要停止Nginx，請輸入openresty -s stop。

配置Nginx

此配置說明基於OpenResty的Nginx安裝。OpenResty的預設目錄為：

- <nginx-install-directory> = /usr/local/openresty/nginx
 - <Openresty-install-directory> = /usr/local/openresty
1. 從包含Nginx的反向Proxy配置的[Finesse版本12.6\(1\)ES03軟體下載頁](#)(12.6-ES03-reverse-proxy-config.zip)下載並解壓縮檔案。
 2. 將nginx.conf、nginx/conf.d/和nginx/html/從提取的反向代理配置目錄分別複製到<nginx-install-directory>/conf、<nginx-install-directory>/conf/conf.d/和<nginx-install-directory>/html/。
 3. 從<nginx-install-directory>中的已解壓反向代理配置目錄複製nginx/lua目錄。
 4. 將lualib的內容複製到<Openresty-install-directory>/lualib/resty。
 5. 透過將nginx/logrotate/saproxy檔案複製到<nginx-install-directory>/logrotate/資料夾來配置nginx日誌旋轉。如果未使用Nginx預設值，請修改檔案內容以指向正確的記錄目錄。
 6. Nginx必須使用專用非特權服務帳戶運行，該帳戶必須鎖定並具有無效的shell（或適用於所選作業系統）。
 7. 在名為html和conf.d的解壓縮資料夾下的檔案中查詢「Must-change」字串，並用相應的條目替換指示的值。
 8. 確保所有必需的替換都已完成，在配置檔案中，這些替換以Must-change 註釋進行了說明。
 9. 確保為CUIC和Finesse配置的快取目錄與這些臨時目錄一起建立在<nginx-install-directory>/cache下。
 - <nginx-install-directory>/cache/client_temp
 - <nginx-install-directory>/cache/proxy_temp



注意：提供的配置用於2000部署示例，必須針對較大的部署進行適當擴展。

配置Nginx快取

依預設，代理快取路徑會儲存在檔案系統中。我們建議在tmpfs中建立快取位置，將其變更為記憶體中的磁碟機，如此處所示。

1. 在/home下為不同的代理快取路徑建立目錄。

例如，必須為主Finesse建立這些目錄。輔助Finesse和CUIC伺服器應採取相同的步驟。

```
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
```



注意：為增加到配置的每個新Finesse集群增加1 GB的客戶端和proxy_temp快取。

2. 使用`mount -av`命令裝載新的裝載點。
3. 使用`df -h`命令驗證檔案系統是否已裝載新的裝載點。
4. 更改Finesse和CUIC快取配置檔案中的`proxy_cache_path`位置。

例如，要更改Finesse主目錄的路徑，請轉到`<nginx-install-directory>conf/conf.d/finesse/cache`，並將現有快取位置`/usr/local/openresty/nginx/cache/finesse25/`更改為新建立的檔案系統位置`/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. 對Finesse輔助伺服器 and CUIC伺服器執行相同的步驟。



注意：確保在前面所有步驟中建立的所有tmpfs驅動器大小的總和已增加到最終的部署記憶體大小中，因為這些驅動器是配置為類似於應用程式的磁碟並消耗最多記憶體空間的記憶體塊。

使用自簽名證書-測試部署

只有在反向代理準備好部署到生產環境之前，才應使用自簽名證書。在生產部署中，僅使用證書頒發機構(CA)簽名證書。

1. 為SSL資料夾內容生成Nginx證書。生成證書之前，需要在/usr/local/openresty/nginx下建立一個名為ssl的資料夾。您需要透過這些指令的協助產生兩個憑證(一個用於<reverseproxy_primary_fqdn>，另一個用於<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` (將主機名傳遞為：`<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` (將主機名傳遞為：`<reverseproxy_secondary_fqdn>`)
 - c. 確保證書路徑為/usr/local/openresty/nginx/ssl/nginx.crt和/usr/local/openresty/nginx/ssl/nginxnode2.crt，因為這些路徑已在Finesse Nginx配置檔案中配置。
2. 更改私鑰400 (r-----)的許可權。
3. 在反向代理上配置防火牆/[iptables](#)，使來自防火牆的通訊與Nginx伺服器已配置為監聽的埠相對應。
4. 在反向代理伺服器的/etc/hosts條目下增加Finesse、IdS和CUIC的IP地址和主機名。
5. 有關要在元件伺服器上執行的配置，請參閱《解決方案功能指南》，以將Nginx主機配置為反向代理。



注意：提供的配置用於2000部署示例，必須針對較大的部署進行適當擴展。

使用CA簽名的證書-生產部署

透過以下步驟，可以在反向代理上安裝CA簽署的憑證：

1. 產生憑證簽署請求(CSR)。

若要產生CSR和私密金鑰，請在您登入Proxy後輸入`openssl req -new -newkey rsa:4096 -keyout nginx.key -out nginx.csr`。按照提示進行操作，並提供詳細資訊。這會產生強度4096位元的CSR (範例中的nginx.csr) 和RSA私密金鑰 (範例中的nginx.key)。

舉例來說：

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

attributes to be sent with your certificate request A challenge password []:challengePWD An optional company name []:CompanyName

記下PEM密碼，因為此密碼將用於解密部署期間的私鑰。

2. 從CA取得簽署的憑證。

將CSR傳送到證書頒發機構並獲取簽名證書。

注意：如果從CA收到的證書不是包含所有相應證書的證書鏈，請將所有相關證書合成一個證書鏈檔案。

3. 部署證書和金鑰。

使用 `openssl rsa -in nginx.key -out nginx_decrypted.key` 命令解密之前生成的金鑰，這是第一個步驟的一部分。將CA簽名的證書和解密的金鑰放在反向代理電腦中的資料夾 `/usr/local/openresty/nginx/ssl` 中。在配置檔案 `/usr/local/openresty/nginx/conf/conf.d/ssl/ssl.conf` 的Nginx配置中更新/增加與證書相關的SSL配置。

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

4. 配置證書的許可權。

`chmod 400 /usr/local/openresty/nginx/ssl/ca_signed_cert.crt` 輸入 `chmod 400 /usr/local/openresty/nginx/ssl/nginx_decrypted.key`，使憑證具有唯讀許可權，且僅限於擁有者。

5. 重新載入Nginx。

使用自訂Diffie-Hellman引數

使用以下命令建立自定義Diffie-Hellman引數：

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

更改伺服器配置，以使用檔案 `/usr/local/openresty/nginx/conf/conf.d/ssl/ssl.conf` 中的新引數：

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

確保啟用OCSP裝訂-證書吊銷檢查

注意：要啟用此功能，伺服器應使用由CA簽名的證書，並且伺服器應有權訪問由證書簽名的CA。

在 `file/usr/local/openresty/nginx/conf/conf.d/ssl/ssl.conf` 中增加/更新此配置：

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

Nginx配置

必須修改預設Nginx配置檔案(/usr/local/openresty/nginx/conf/nginx.conf)以包含這些條目來實施安全和提供效能。此內容應該用於修改Nginx安裝建立的預設配置檔案。

```
# 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 CPU
worker_processes auto;

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

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

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

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

#user root root;

# current limit on the maximum number of open files by worker processes, keeping 10 times of worker_con
worker_rlimit_nofile 102400;

events {
    multi_accept on;

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

http {

    include mime.types;

    default_type text/plain;

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

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

```

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

lua_shared_dict userlist 50m;
lua_shared_dict credentialsstore 100m;
lua_shared_dict userscount 100k;
lua_shared_dict clientstorage 100m;
lua_shared_dict blockingresources 100m;
lua_shared_dict tokencache_saproxy 10M;
lua_shared_dict tokencache_saproxy125 10M;
lua_shared_dict ipstore 10m;
lua_shared_dict desktopurllist 10m;
lua_shared_dict desktopurlcount 100k;
lua_shared_dict thirdpartygadgeturllist 10m;
lua_shared_dict thirdpartygadgeturlcount 100k;
lua_shared_dict corsheadersstore 100k;

init_worker_by_lua_block {
    local UsersListManager = require('users_list_manager')
    local UnauthenticatedDesktopResourcesManager = require("unauthenticated_desktopresources_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/a")
        UnauthenticatedDesktopResourcesManager.getDesktopResources("saproxy.cisco.com", "https://sa")
        UnauthenticatedResourcesManager.getThirdPartyGadgetResources("saproxy.cisco.com", "https://")
    end
}

include conf.d/*.conf;

sendfile        on;

tcp_nopush     on;

server_names_hash_bucket_size 512;

```

配置反向代理埠

預設情況下，Nginx配置在埠8445上偵聽Finesse請求。一次只能從反向代理啟用一個埠來支援Finesse請求，例如8445。如果需要支援埠443，請編輯<nginx-install-directory>conf/conf.d/finesse.conf檔案，以便在443上啟用偵聽功能，並在8445上停用偵聽功能。

配置反向代理和上游元件之間的雙向TLS驗證

可在CCBU上游元件CUIC/Finesse/IdS/Livedata上透過新的CVOS CLI選項(即

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

預設情況下，這是停用的，管理員必須在每個上游伺服器上單獨執行CLI來明確啟用此功能。啟用此選項後，在上遊主機上運行的Cisco Web代理服務將對源自CLI實用程式中增加的受信任反向代理主機的連線的TLS握手客戶端證書進行身份驗證。utils system reverse-proxy allowed-hosts add <proxy-host>。

以下是代理配置檔案ssl.conf和ssl2.conf中的配置塊

```
#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;
```

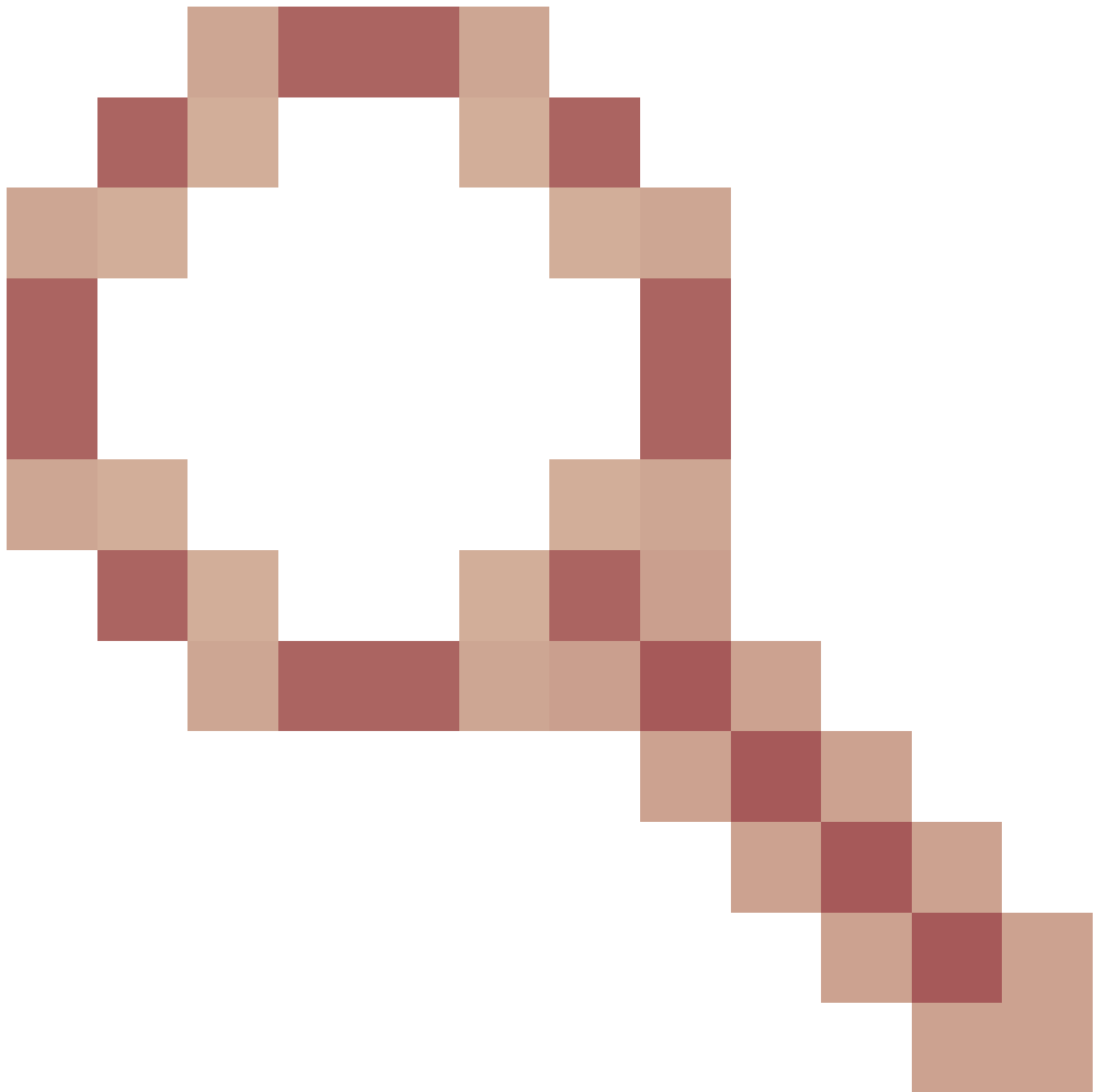
用於出站流量的SSL證書（代理到上游）可以與為入站流量配置的ssl證書（用於元件伺服器塊的SSL聯結器）相同。如果將自簽名證書用作proxy_ssl_certificate，則必須將其上載到上游元件（Finesse/IdS/CUIC/Livedata）tomcat trust store才能成功對其進行身份驗證。

使用反向代理驗證上游伺服器證書是可選的，預設情況下已停用。如果您希望在反向代理和上遊主機之間實現完整的TLS相互身份驗證，則需要從ssl.conf和ssl2.conf檔案中取消註釋以下配置。

```
#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
```

配置雙向TLS身份驗證的警告：

- 在CCBU元件上啟用此功能後，在TLS握手期間，也將向LAN客戶端請求客戶端證書。如果客戶端電腦上安裝了任何客戶端/個人證書，瀏覽器可能會選擇向終端使用者顯示彈出窗口，要求為客戶端身份驗證選擇適當的證書。儘管終端使用者選擇哪個證書或在彈出請求上按取消並不重要，但會成功，因為LAN客戶端不會強制進行客戶端證書身份驗證，但體驗會有所變化。請參閱CDET [CSCwa26057](#)



以獲取更多詳細資訊。

- 如果將代理主機增加到無法由Web代理服務解析的允許清單，則上游元件的Web代理服務將無法啟動。確保透過DNS查詢從上游元件解析增加到允許清單的反向代理主機。

清除快取

反向代理快取可以透過

```
/clearCache.sh
```

命令來清除。

標準準則

本節簡要介紹將Nginx設定為代理伺服器時需要遵循的標準準則。

這些指南源自[Internet安全中心](#)。有關每個指南的更多詳情，請參閱同一指南。

1. 建議始終使用最新穩定的OpenResty和OpenSSL版本。
2. 建議將Nginx安裝在單獨的磁碟裝載中。
3. Nginx進程ID必須由root使用者擁有（或適用於所選作業系統），並且必須具有644 (rw-----)或更嚴格的許可權。
4. Nginx必須阻止未知主機的請求。確定每個伺服器區塊都包含明確定義的server_name指令。要進行驗證，請搜尋nginx.conf和nginx/conf.d目錄中的所有伺服器塊，並驗證所有伺服器塊都包含server_name。
5. Nginx只能偵聽授權埠。搜尋nginx.conf和nginx/conf.d目錄中的所有伺服器區塊，並檢查是否有listen指令，以便驗證只有已授權的連線埠可以開啟供偵聽。
6. 由於Cisco Finesse不支援HTTP，因此建議同時阻止代理伺服器HTTP埠。
7. Nginx SSL協定必須是TLS 1.2。必須移除對舊版SSL通訊協定的支援。還必須停用弱的SSL密碼。
8. 建議將Nginx錯誤和訪問日誌傳送到遠端系統日誌伺服器。
9. 建議安裝用作Web應用程式防火牆的mod_security模組。有關詳細資訊，請參閱[ModSecurity手冊](#)。請注意，Nginx載入尚未在mod_security模組內進行驗證。

配置對映檔案

Finesse案頭的反向代理部署需要一個對映檔案來配置外部可見主機名/埠組合清單以及它們與Finesse、Id和CUIC伺服器使用的實際伺服器名稱和埠的對映。在內部伺服器上配置的此對映檔案是允許透過Internet連線的客戶端重定向到Internet上使用的所需主機和埠的主要配置。

對映檔案必須部署在元件伺服器可訪問的Web伺服器上，並且需要配置其URI以使部署正常工作。建議使用網路中可用的專用Web伺服器配置對映檔案。如果不能使用此類伺服器，可以使用反向代理，這將要求代理從網路內部訪問，並且還會使資訊暴露給外部客戶端，這些外部客戶端可能會未經授權訪問DMZ。下一節將詳細介紹如何實現此目標。

有關在所有元件伺服器上配置對映檔案URI的確切步驟以及如何建立對映檔案資料的詳細資訊，請參閱功能指南。

使用反向代理作為對映檔案伺服器

僅當反向代理也用作代理對映檔案主機時，才需要執行這些步驟。

1. 在Finesse/CUIC和IdS主機使用的域控制器中配置反向代理主機名，以便可以解析其IP地址。
2. 將生成的Nginx簽名證書上傳到cmplatform的tomcat-trust下的兩個節點上，然後重新啟動伺服器。
3. 更新<NGINX_HOME>/html/proxymap.txt中的Must-change值。
4. 使用nginx -s reload命令重新載入Nginx配置。
5. 使用curl命令驗證配置檔案是否可從其他網路主機訪問。

CentOS 8核心強化

如果選擇的作業系統是CentOS 8，建議在使用專用伺服器代管Proxy的安裝中，使用這些sysctl組態來完成核心強化/調整。


```
## 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
```


進行建議的變更後，建議重新開機。


IPtables強化

IPtables是一種應用程式，允許系統管理員配置Linux核心防火牆提供的IPv4和IPv6表、鏈和規則。

這些IPtables規則配置為透過限制Linux核心防火牆的訪問來保護代理應用程式免受暴力攻擊。

配置中的註釋指示使用規則對哪些服務進行速率限制。

 注意：如果管理員使用不同的埠，或者使用相同的埠擴展對多台伺服器的訪問，則必須根據這

 些數字相應地確定這些埠的大小。

```
## 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 --hashlimi
-A INPUT -p tcp -m tcp --dport 8445 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG -
-A INPUT -p tcp -m tcp --dport 8445 --tcp-flags SYN SYN -j DROP
```

```
# Configuration for IdS 8553 port
```

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

```
# Configuration for IdP 443 port
```

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

```
# Must-Change: A2A file transfer has not been considered for below IMNP configuration.
```

```
# For A2A for support, these configuration must be recalculated to cater different file transfer scenar
```

```
# Configuration for IMNP 5280 port
```

```
-A INPUT -p tcp -m tcp --dport 5280 --tcp-flags SYN SYN -m connlimit --connlimit-above 30 --connlimit-m
```

```
-A INPUT -p tcp -m tcp --dport 5280 --tcp-flags SYN SYN -m connlimit --connlimit-above 30 --connlimit-m
-A INPUT -p tcp -m tcp --dport 5280 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 20/sec --hashlimi
-A INPUT -p tcp -m tcp --dport 5280 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG -
-A INPUT -p tcp -m tcp --dport 5280 --tcp-flags SYN SYN -j DROP
```

```
# Configuration for IMNP 15280 port
```

```
-A INPUT -p tcp -m tcp --dport 15280 --tcp-flags SYN SYN -m connlimit --connlimit-above 30 --connlimit-m
-A INPUT -p tcp -m tcp --dport 15280 --tcp-flags SYN SYN -m connlimit --connlimit-above 30 --connlimit-m
-A INPUT -p tcp -m tcp --dport 15280 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 20/sec --hashlimi
-A INPUT -p tcp -m tcp --dport 15280 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG
-A INPUT -p tcp -m tcp --dport 15280 --tcp-flags SYN SYN -j DROP
```

```
# Configuration for IMNP 25280 port
```

```
-A INPUT -p tcp -m tcp --dport 25280 --tcp-flags SYN SYN -m connlimit --connlimit-above 30 --connlimit-m
-A INPUT -p tcp -m tcp --dport 25280 --tcp-flags SYN SYN -m connlimit --connlimit-above 30 --connlimit-m
-A INPUT -p tcp -m tcp --dport 25280 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 20/sec --hashlimi
-A INPUT -p tcp -m tcp --dport 25280 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG
-A INPUT -p tcp -m tcp --dport 25280 --tcp-flags SYN SYN -j DROP
```

```
# Configuration for CUIC 8444 port
```

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

```
# Configuration for CUIC 8447 port
```

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

```
# Configuration for LiveData 12005 port
```

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

```
# Configuration for LiveData 12008 port
```

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

```
# Block all other ports
```

```
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
```

```
COMMIT
```

可以透過手動編輯/etc/sysconfig/iptables直接應用這些規則，或者將配置儲存到檔案(例如iptables.conf)中並執行cat iptables.conf >>/etc/sysconfig/iptables來應用這些規則。

應用規則後，需要重新啟動IPtables服務。輸入systemctl restart iptables以重新啟動IPtables服務。

限制客戶端連線

除了以前的IPtables配置之外，建議為使用Proxy的客戶端安裝知道地址範圍的客戶端，以便利用此知識保護Proxy訪問規則。當涉及從惡意網路的殭屍網路保護代理時，這可以帶來巨大的回報，這些惡意網路通常在各國的IP地址範圍內建立，而這些國家在線上安全方面的規則較為寬鬆。因此，如果確定訪問模式，強烈建議將IP地址範圍限制為國家/州或基於ISP的IP範圍。

阻止客戶端連線

當確定攻擊來自IP地址或IP地址範圍時，知道如何阻止特定地址範圍也很有用。在這些情況下，可以用iptables規則阻止來自這些IP地址的請求。

阻止不同的IP地址

要阻止多個不同的IP地址，請在每個IP地址的IPTables配置檔案中增加一行。

例如，要阻止地址192.0.2.3和192.0.2.4，請輸入：

```
<#root>
```

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

```
192.0.2.3
```

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

```
192.0.2.4
```

```
- j DROP.
```

阻塞IP地址範圍

阻止一個範圍中的多個IP地址，並使用IP範圍在IPTables配置檔案中增加一行。

例如，要阻止從192.0.2.3到192.0.2.35的地址，請輸入：

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

阻止子網中的所有IP地址

使用無類域間路由表示法將IP地址範圍增加到IPTables配置檔案中，以阻止整個子網中的所有IP地址。例如，要阻止所有C類地址，請輸入：

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

SELinux

SELinux是一種平台安全架構，它整合在Linux作業系統中，作為增強功能。安裝並增加SELinux策略以運行OpenResty的過程（接下來提供反向代理）。

1. 使用`openresty -s stop`命令停止進程。
2. 使用`systemctl`命令配置並啟動/stop nginx伺服器，以便在啟動期間自動啟動OpenResty進程。以root使用者身份輸入這些命令。
 - a. 轉到`/usr/lib/systemd/system`。
 - b. 打開名為`openresty.service`的檔案。
 - c. 根據PIDFile位置更新檔案的內容。

```
[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. 以root使用者的身份輸入`sudo systemctl enable openresty`。
- e. 使用`systemctl start openresty / systemctl stop openresty`命令啟動/停止OpenResty服務，並確保進程以root使用者身份啟動/停止。

1. 安裝Selinux

- 依預設，只有部分SELinux套裝軟體會安裝在CentOs中。
- 需要安裝`policycoreutils-devel`軟體套件及其依賴項，才能生成SELinux策略。
- 輸入以下命令可安裝`policycoreutils-devel`

```
yum install policycoreutils-devel
```

- 確保在安裝軟體套件之後，`sepolicy`命令起作用。

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

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

```
SELinux Policy Inspection Tool
```

2. 建立新的Linux使用者並使用SELinux使用者進行對映

- a. 輸入 `semanage login -l`以檢視Linux使用者和SELinux使用者之間的對映。

```
[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. 以root身份，建立對映至SELinux `user_u`使用者的新Linux使用者(nginx使用者)。

```
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. 要檢視nginxuser和user_u之間的對映，請以root身份輸入以下命令：

```
[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. 依預設，SELinux `__default__` login對映至SELinux `unconfined_u`使用者。需要使用以下命令使預設情況下的user_u受到限制：

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

要檢查命令是否工作正常，請輸入`semanage login -l`。它應該會產生以下輸出：

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

e. 修改nginx.conf並更改nginxuser的所有權。

- i. 在<Openresty-install-directory>目錄中輸入`chown -R nginxuser:nginxuser*`。
- ii. 修改nginx.conf檔案，以包含nginxuser作為正在運行的工作進程使用者。

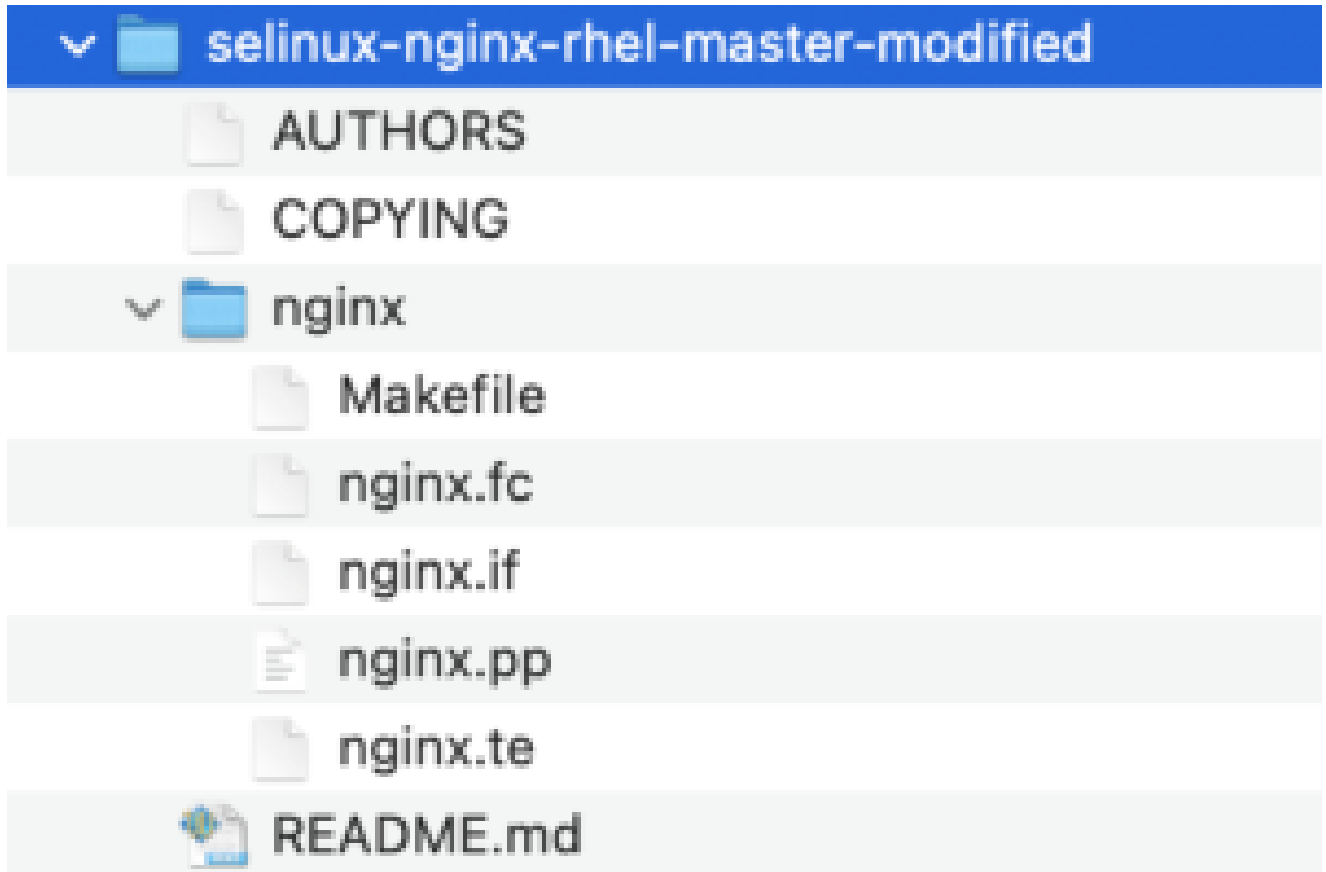
```

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

```

為Nginx編寫SELinux策略

1. 相對於使用`sepolicy generate --init /usr/bin/nginx`命令為Nginx生成新的預設自定義策略，我們更傾向於從現有策略開始。
2. 從提供的URL下載的nginx.fc檔案（檔案上下文檔案）和nginx.te（型別實施檔案）檔案已修改為適合反向代理使用。
3. 此修改後的版本可用作參考，因為它已針對特定使用案例進行了修復。
4. 從[檔案軟體下載頁](#)下載檔案selinux-nginx-rhel-master-modified.tar。



5. 解壓縮.tar檔案，並導航到該檔案中的nginx目錄。
6. 打開.fc檔案並驗證nginx安裝程式、快取和pid檔案所需的檔案路徑。
7. 用make命令編譯配置。
8. 生成nginx.pp檔案。
9. 使用semodule命令載入策略。

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

10. 轉到/root並建立一個名為touch /.autorelabel的空檔案。
11. 重新啟動系統。
12. 輸入此命令以驗證策略是否已成功載入。

```
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應運行而無任何違規。(違規將位於/var/log/messages和/var/log/audit/audit.log中)。
14. 輸入以下命令以檢查Nginx的狀態。

```
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. 現在，應該可以訪問Finesse代理/Supervisor案頭。

驗證

使用本節內容，確認您的組態是否正常運作。

Finse

1. 從DMZ請求https://<reverseproxy : port>/finesse/api/SystemInfo.並檢查它們是否可訪問。
2. 檢查<primaryNode>和<secondaryNode>中的<host>值都是有效的反向代理主機名。它不應是Finesse主機名。

CUIC和即時資料

1. 如果在響應中看到Finesse主機名而不是反向代理主機名，則按照《[Finesse 12.6 UCCE功能指南](#)》「無VPN訪問Finesse案頭」的「填充網路轉換資料」部分所述，驗證代理對映配置和允許的主機正確增加到Finesse伺服器。
2. 如果LiveData小工具在Finesse Desktop中正確載入，則CUIC和LiveData代理配置正確。
3. 為了驗證CUIC和LiveData配置，請從DMZ向這些URL發出HTTP請求，並檢視它們是否可以訪問。

- https://<reverseproxy : cuic_port>/cuic/rest/about
- https://<reverseproxy : ldweb_port>/livedata/security
- https://<reverseproxy : ldsocketio_port>/security

IDS

要驗證IdS配置，請執行以下步驟：

1. 從LAN登入到IdSAdmin介面，地址為https://<ids_LAN_host : ids_port> : 8553/idsadmin，因為管理介面未通過反向代理公開。
2. 選擇Settings > IdS Trust。
3. 驗證代理群集發佈伺服器節點是否列在「下載SP後設資料」頁上，然後按一下下一步。
4. 驗證是否已在「上傳IDP後設資料」頁上配置的情況下正確顯示IDP代理，然後按一下下一步。
5. 從「測試SSO」頁面透過所有代理叢集節點起始測試SSO，並驗證所有節點是否成功。這需要客戶端電腦連線以反向代理節點。

效能

使用nmon工具完成的頂級等效效能捕獲的資料分析可從[Finesse版本12.6\(1\) ES03軟體下載頁](#) (load_result.zip)獲得。資料表示使用SSO登入和CUIC LD報告的示例2000 UCCE部署上用於案頭和Supervisor操作的Proxy的狀態，該配置在預設佈局中為2000個使用者配置了8小時。它可用於推導在類似硬體上使用Nginx進行安裝的計算、磁碟和網路要求。

疑難排解

SSO

1. 案頭重定向不透過proxy
 1. 根據不同配置（如proxymap.txt、server_filter檔案等）中的實際vm主機名，檢查主機名配置是否正確。
 2. 確保在CCE清單中增加ld和正確的主機名，因為從CCE Web管理員註冊SSO時，相同資訊會推送到元件。
2. 未進行SSO登入
 1. 確保已為代理主機建立IdS-IDP信任。

SELinux

1. 如果預設情況下未啟動Nginx或者無法訪問Finesse座席案頭，請使用此命令將SELinux設定為permissive模式：

```
setenforce 0
```

2. 嘗試使用systemctl restart nginx命令重新啟動Nginx。
3. 違規將位於/var/log/messages和/var/log/audit/audit.log中。

4. 需要重新產生.te檔案，並具備允許規則，以便透過下列任一指令處理這些違規：

```
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. 使用新生成的允許規則更新selinux-nginx-rhel-master-modified/nginx目錄中的原始nginx.te檔案。
6. 使用make命令進行編譯。
7. 將重新生成nginx.pp檔案。
8. 透過semodule命令載入策略。

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

9. 使用以下命令將SELinux設定為enforce模式：

```
setenforce
```

10. 重新啟動系統。
11. 重複此程式，直到修正所需的違規為止。

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。