

# Configurazione di Nginx Reverse Proxy per l'accesso senza VPN a Cisco Finesse (12.6 ES03)

## Sommario

---

### [Introduzione](#)

### [Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

### [Premesse](#)

[Modifiche in ES03](#)

[Note sull'aggiornamento per le configurazioni VPN senza ES01](#)

### [Autenticazione](#)

[Autenticazione non SSO](#)

[Autenticazione SSO](#)

[Autenticazione per connessioni Websocket](#)

### [Prevenzione degli attacchi di forza bruta](#)

[Registrazione](#)

[Installazione e configurazione di Fail2ban](#)

[Convalida URL risorse statiche](#)

[Memorizzazione nella cache delle intestazioni CORS](#)

### [Configurazione](#)

[Configura i componenti della soluzione per l'accesso VPN Less](#)

[Installare OpenResty come proxy inverso in DMZ](#)

[Installazione di OpenResty](#)

### [Configura Nginx](#)

[Configurazione della cache Nginx](#)

[Configura certificati SSL](#)

[Usa parametro Diffie-Hellman personalizzato](#)

[Verifica attivazione associazione OCSP - Controllo revoca certificato](#)

[Configurazione Nginx](#)

[Configura porta proxy inversa](#)

[Configura autenticazione TLS reciproca tra proxy inverso e componenti upstream](#)

[Cancella cache](#)

[Linee guida standard](#)

### [Configurare il file di mapping](#)

[Usa proxy inverso come file server di mapping](#)

[Protezione avanzata kernel CentOS 8](#)

[Protezione avanzata tabelle IP](#)

[Limita connessioni client](#)

[Blocca connessioni client](#)

[Blocca indirizzi IP distinti](#)

[Blocca un intervallo di indirizzi IP](#)

[Blocca tutti gli indirizzi IP in una subnet](#)

[SELinux](#)

[Verifica](#)

[Finesse](#)

[CUIC e Live Data](#)

[IDS](#)

[Prestazioni](#)

[Risoluzione dei problemi](#)


[SSO](#)

---


## Introduzione

Questo documento descrive come utilizzare un proxy inverso per accedere al desktop Cisco Finesse senza connettersi a una VPN basata sulle versioni 12.6 ES03 di Cisco Finesse, Cisco Unified Intelligence Center (CUIC) e Cisco Identity Service (IdS).

---

 Nota: l'installazione e la configurazione di Nginx non sono supportate da Cisco. Le richieste relative a questo argomento possono essere discusse sui [forum](#) della [community Cisco](#).

---

 Nota: per le implementazioni ES03 di VPN-Less, consultare il file Leggimi dei singoli componenti per pianificare gli aggiornamenti e controllare le restrizioni di compatibilità. [Leggimi di Cisco Finesse 12.6 ES03, CUIC/IdS 12.6 ES03](#)

---

## Prerequisiti

### Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Release Cisco Unified Contact Center Enterprise (UCCE)
- Cisco Finesse
- Amministrazione Linux
- Amministrazione della rete e amministrazione della rete Linux

### Componenti usati


Le informazioni fornite in questo documento si basano sulle seguenti versioni software e hardware:

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

Nota: le distribuzioni PCCE/UCCE 2k devono trovarsi nella versione 12.6 di CCE a causa dell'installazione co-residente di LD/CUIC

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

---

 Nota: la configurazione fornita in questo documento è stata configurata, rafforzata e testata con il proxy inverso Nginx (OpenResty) distribuito in CentOS 8.0, rispetto a un'implementazione UCCE di esempio di 2000 utenti. Le informazioni di riferimento sul profilo delle prestazioni sono disponibili in questo documento.

---

## Premesse

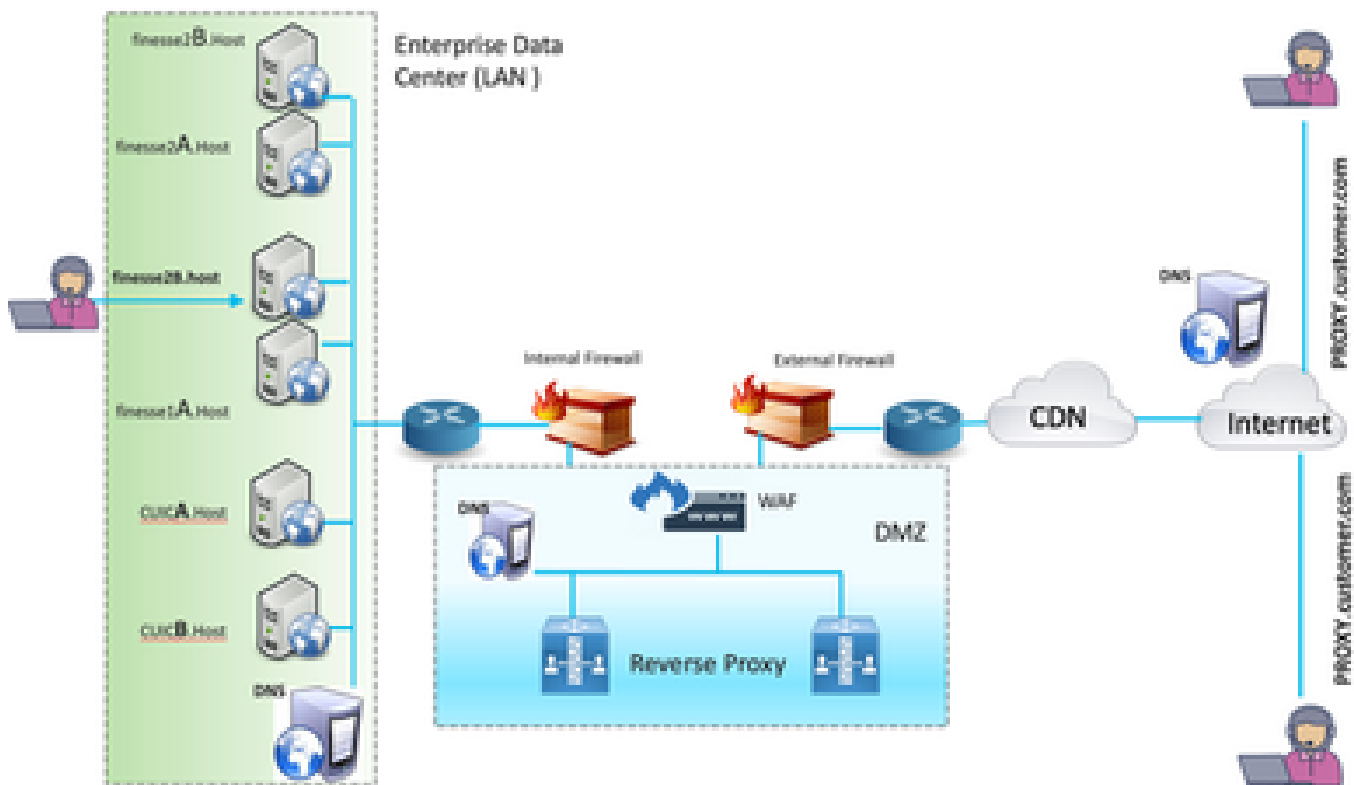
Questo modello di distribuzione è supportato per le soluzioni UCCE/PCCE e HCS for UCCE.

È supportata la distribuzione di un proxy inverso (disponibile dalla versione 12.6 di ES01) come opzione per accedere al desktop Cisco Finesse senza connettersi a una VPN. Questa funzionalità offre agli agenti la flessibilità necessaria per accedere al desktop Finesse da qualsiasi luogo tramite Internet.

Per abilitare questa funzione, è necessario distribuire una coppia di proxy inverso nella zona demilitarizzata (DMZ).

L'accesso ai supporti rimane invariato nelle distribuzioni proxy inversa. Per connettersi ai supporti, gli agenti possono utilizzare Cisco Jabber su una soluzione MRA (Mobile and Remote Access Solution) o la funzionalità di agente mobile di UCCE con una rete PSTN (Public Switched Telephone Network) o un endpoint mobile. Questo diagramma mostra l'aspetto della distribuzione di rete quando si accede a due cluster Finesse e a due nodi CUIC tramite una singola coppia di nodi proxy inverso ad alta disponibilità (HA, High Availability).

È supportato l'accesso simultaneo da agenti su Internet e da agenti che si connettono dalla LAN, come mostrato in questa immagine.



✎ Nota: per il supporto di questa distribuzione, vedere la guida alle funzioni per i criteri di selezione dei proxy di terze parti anziché Nginx.

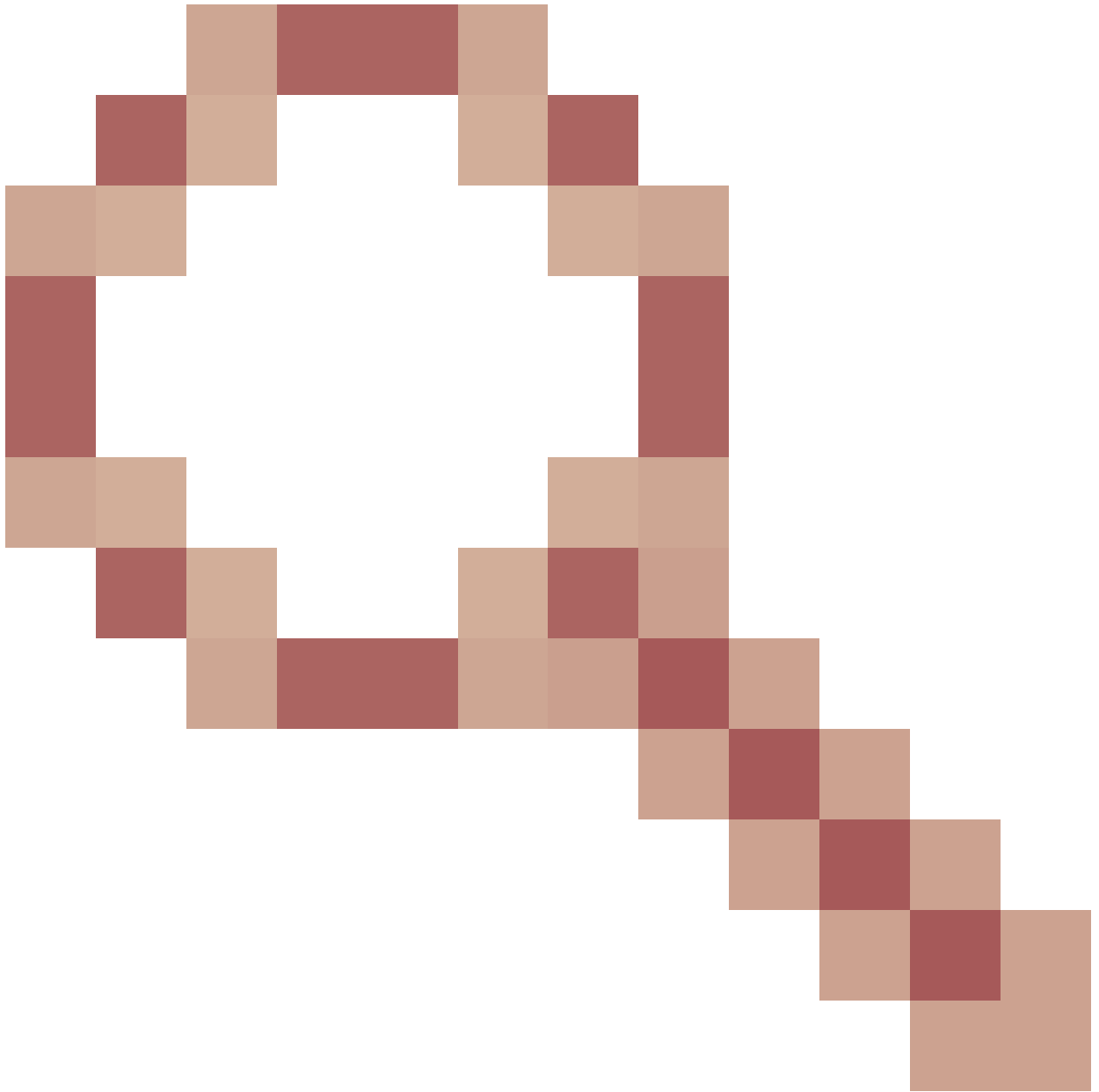
- [Guida alle funzionalità di UCCE 12.6](#) - Fornisce una panoramica delle funzionalità, la progettazione e [i dettagli di configurazione](#) per la funzionalità VPN-Less.
- [UCCE 12.6 Security Guide](#) - Fornisce le linee guida per la configurazione della sicurezza per la distribuzione di proxy inverso.

Si consiglia di consultare la sezione VPN-Less della guida alle funzionalità e alla guida alla sicurezza prima di leggere questo documento.

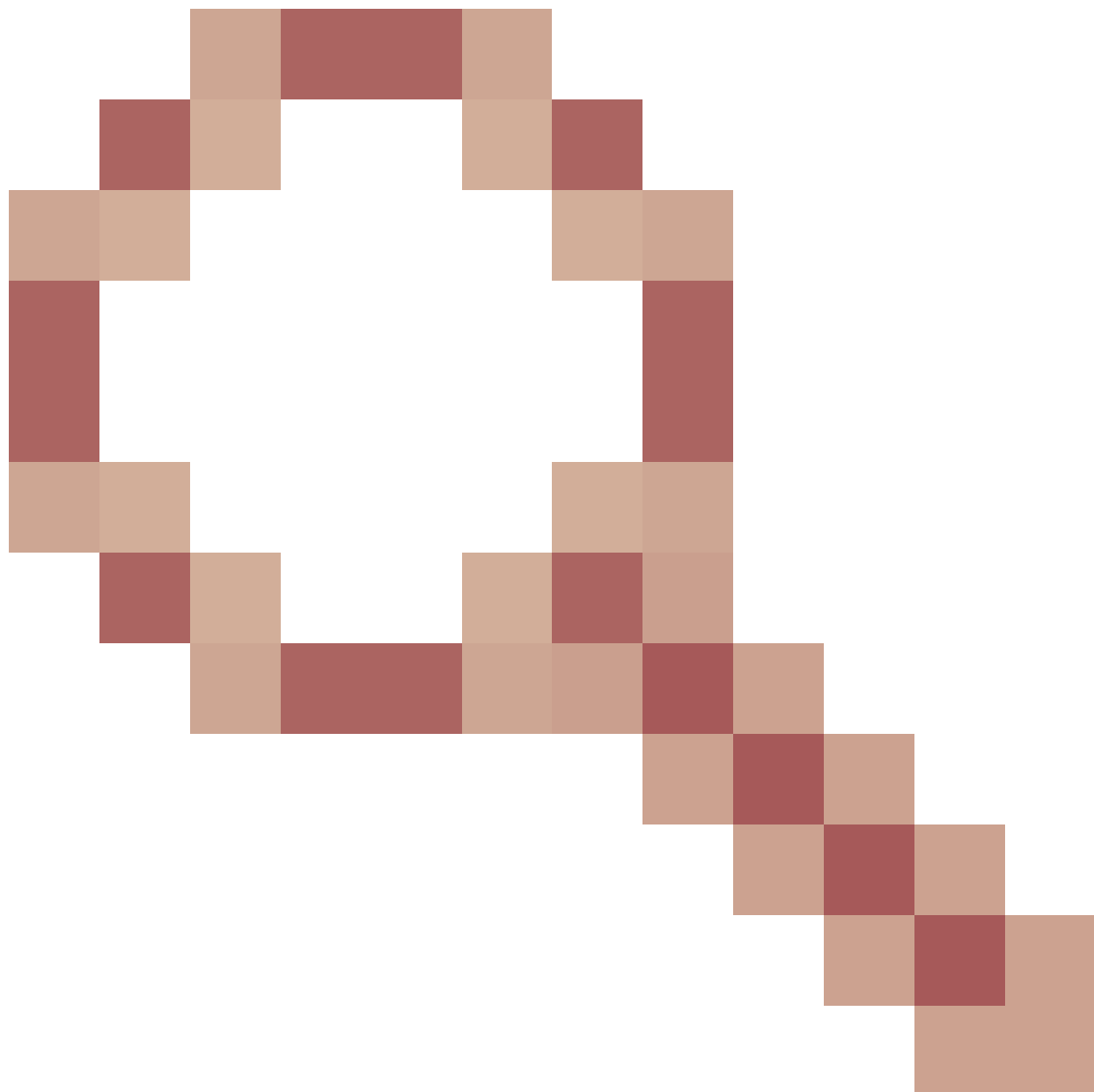
## Modifiche in ES03

- Nuove caratteristiche
  - Le funzionalità di Finesse Supervisor sono ora supportate tramite il proxy inverso.
  - I report cronologici e in tempo reale CUIC sono ora supportati tramite gadget Finesse in un ambiente proxy.
  - Autenticazione per tutte le richieste / comunicazioni - richiede supporto Lua
    - Tutte le richieste Finesse / CUIC / IM & Presence ( IM&P) vengono autenticate sul proxy prima di poter accedere al centro dati.
    - Anche le connessioni I/O Websocket e Live Data Socket sono limitate e consentite solo dai client che hanno inviato correttamente una richiesta protetta a Finesse.

- Rilevamento e registrazione di attacchi di forza bruta sul proxy, che può essere utilizzato con Fail2Ban per bloccare indirizzi IP dannosi.
- Miglioramenti della sicurezza per la configurazione del proxy inverso - richiede il supporto Lua
  - Autenticazione Mutual Transport Layer Security (TLS) tra i componenti reverse proxy e upstream (Finesse/IdS/CUIC/Livedata).
  - Impostazioni SeLinux.
  - Abilitare la verifica dell'attendibilità SSL (Secure Sockets Layer) reciproca per le richieste del server proxy e del server dei componenti.
- Maggiore sicurezza per la configurazione del proxy per prevenire attacchi DoS (Denial-of-Service)/DDoS (Distributed Denial-of-Service) - richiede il supporto Lua
  - Miglioramento dei limiti di tasso di richiesta Nginx per varie parti del sistema.
  - Limiti di velocità per le tabelle IP.
  - Verifica delle richieste di risorse statiche prima della richiesta del server del componente upstream.
  - Pagine non autenticate più leggere e inseribili nella cache che non toccano il server del componente upstream.
- Altre funzioni - richiede il supporto di Lua
  - Rilevamento automatico delle risposte CORS (Cross-Origin Resource Sharing) fornite dal proxy per facilitare la configurazione automatica e migliorare le prestazioni
- Correzioni dei difetti relative a VPN-Less
  - [CSCwa26057](#)

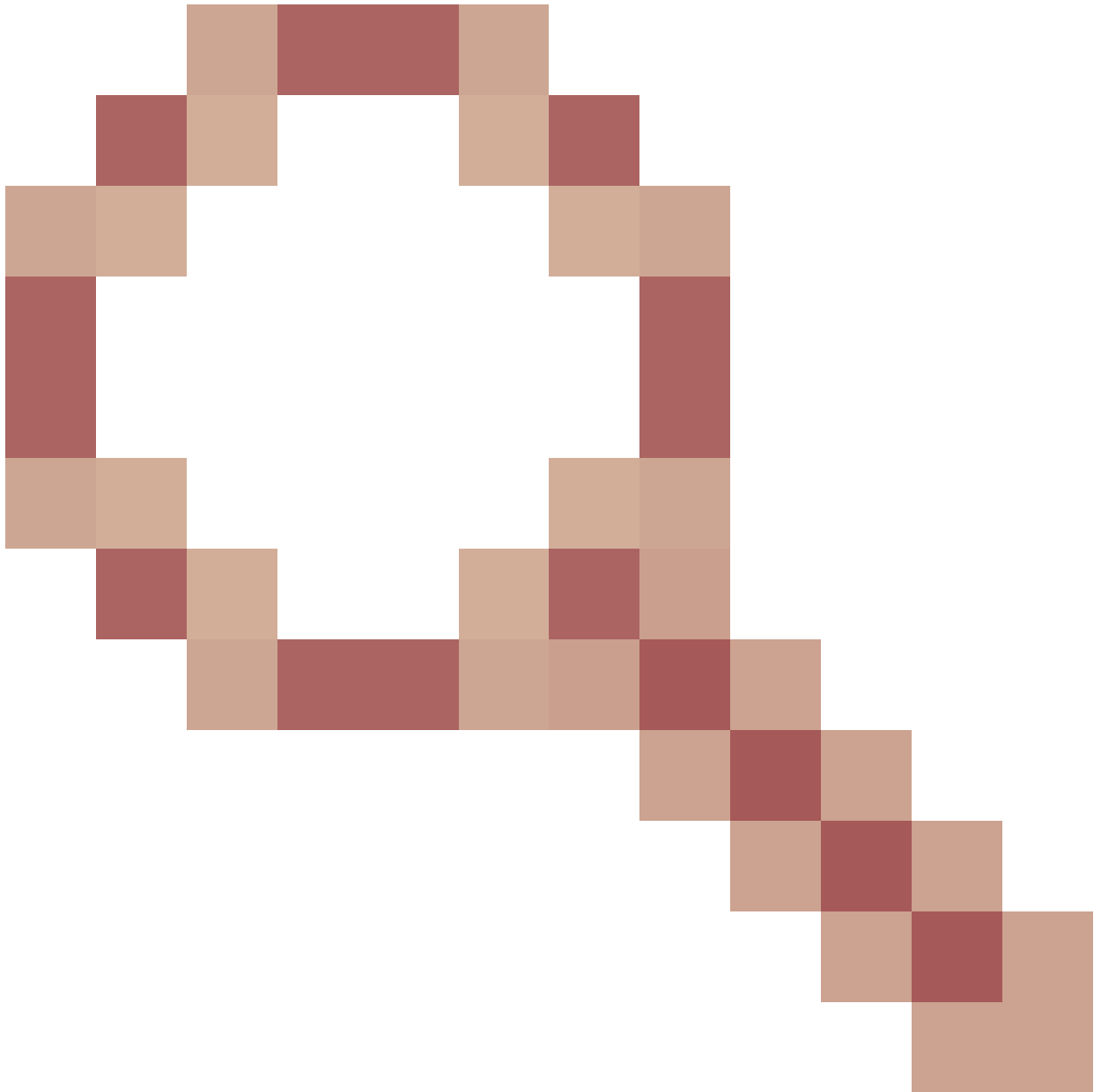


[CSCwa26057](#)



" />- Più certificati offerti all'agente durante l'accesso al desktop finesse

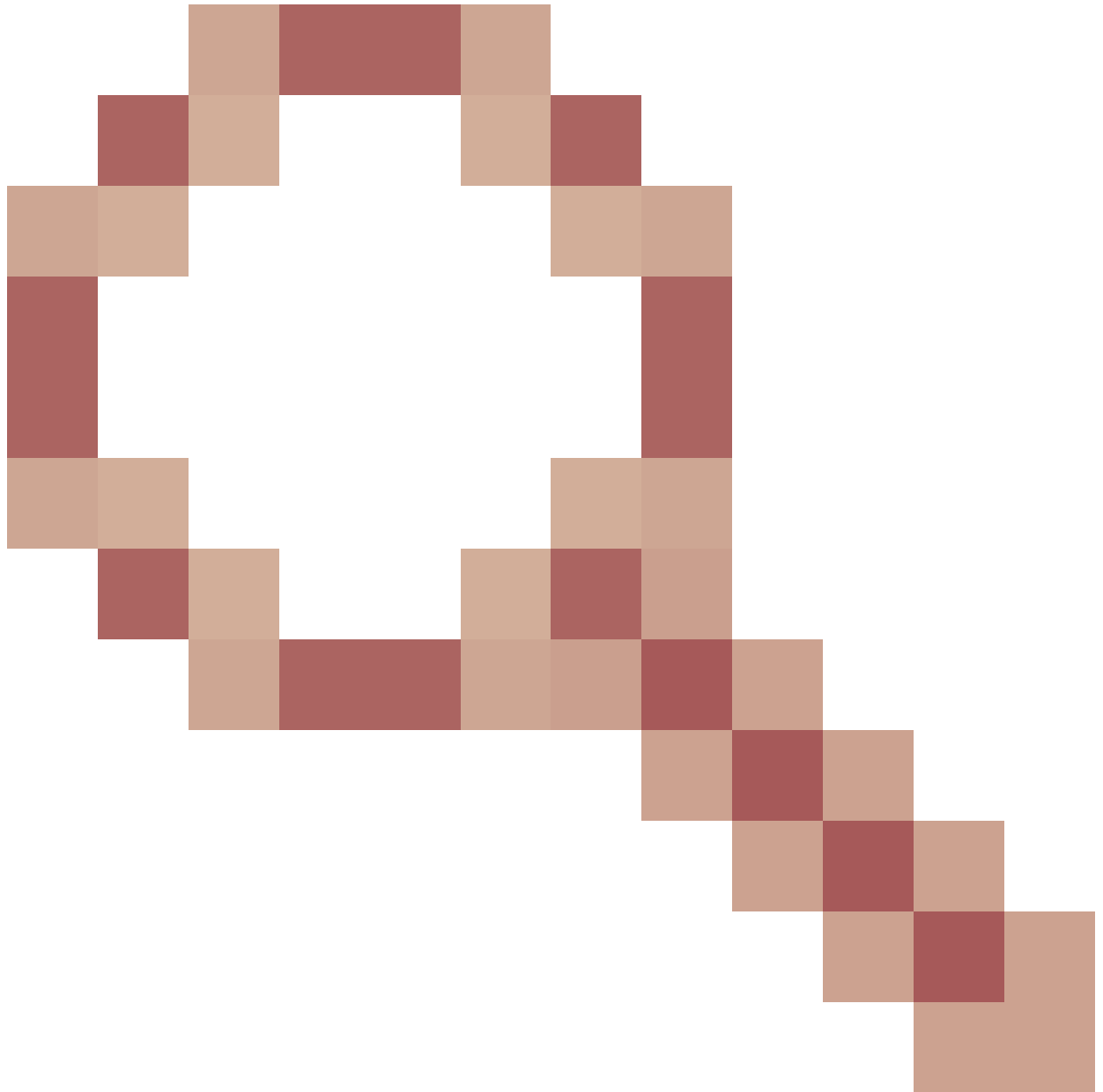
- [CSCwa2471](#)



- La pagina di accesso di Finesse non mostra il nome FQDN dell'agente SSO

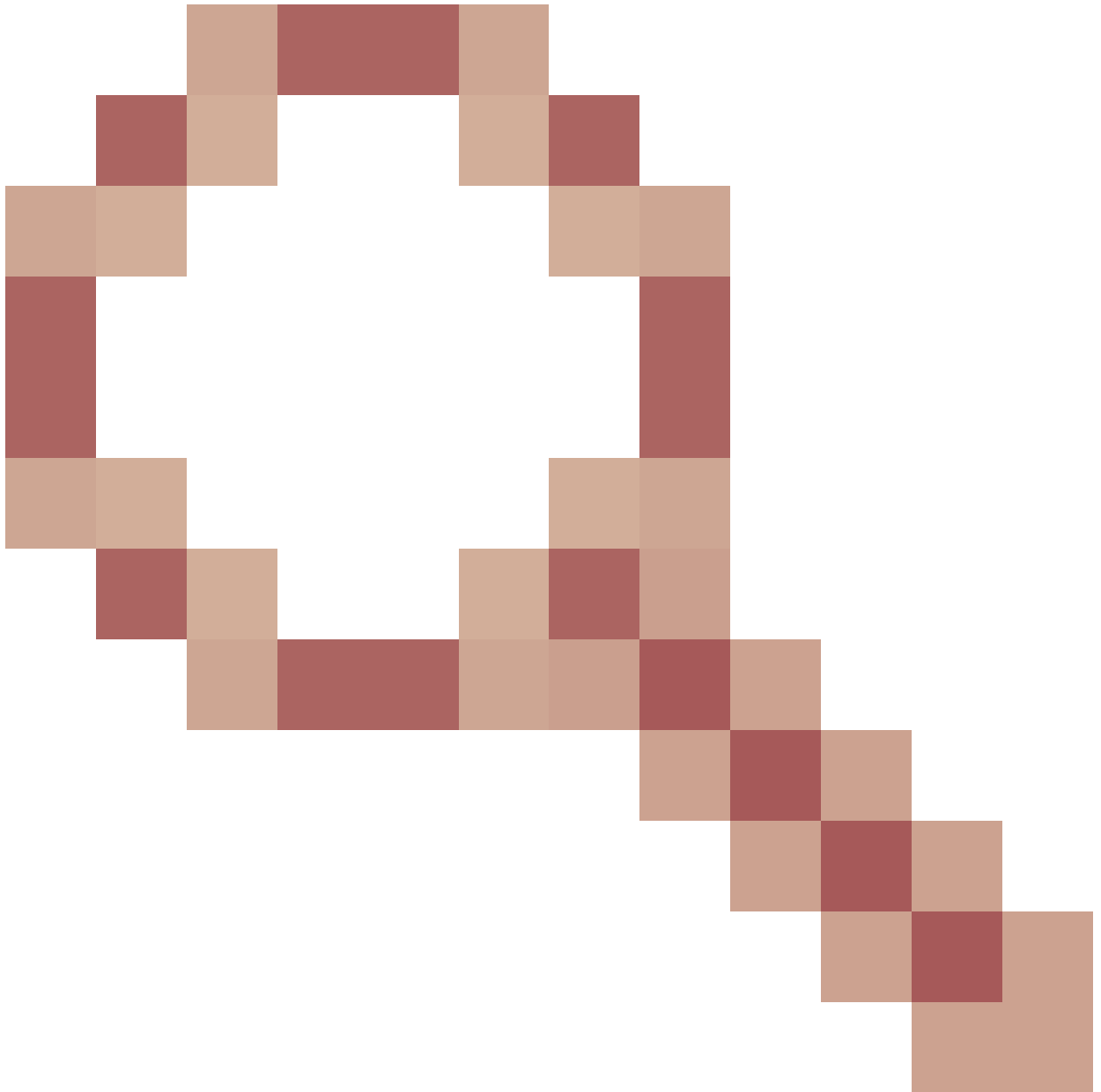
- [CSCwa24519](#)





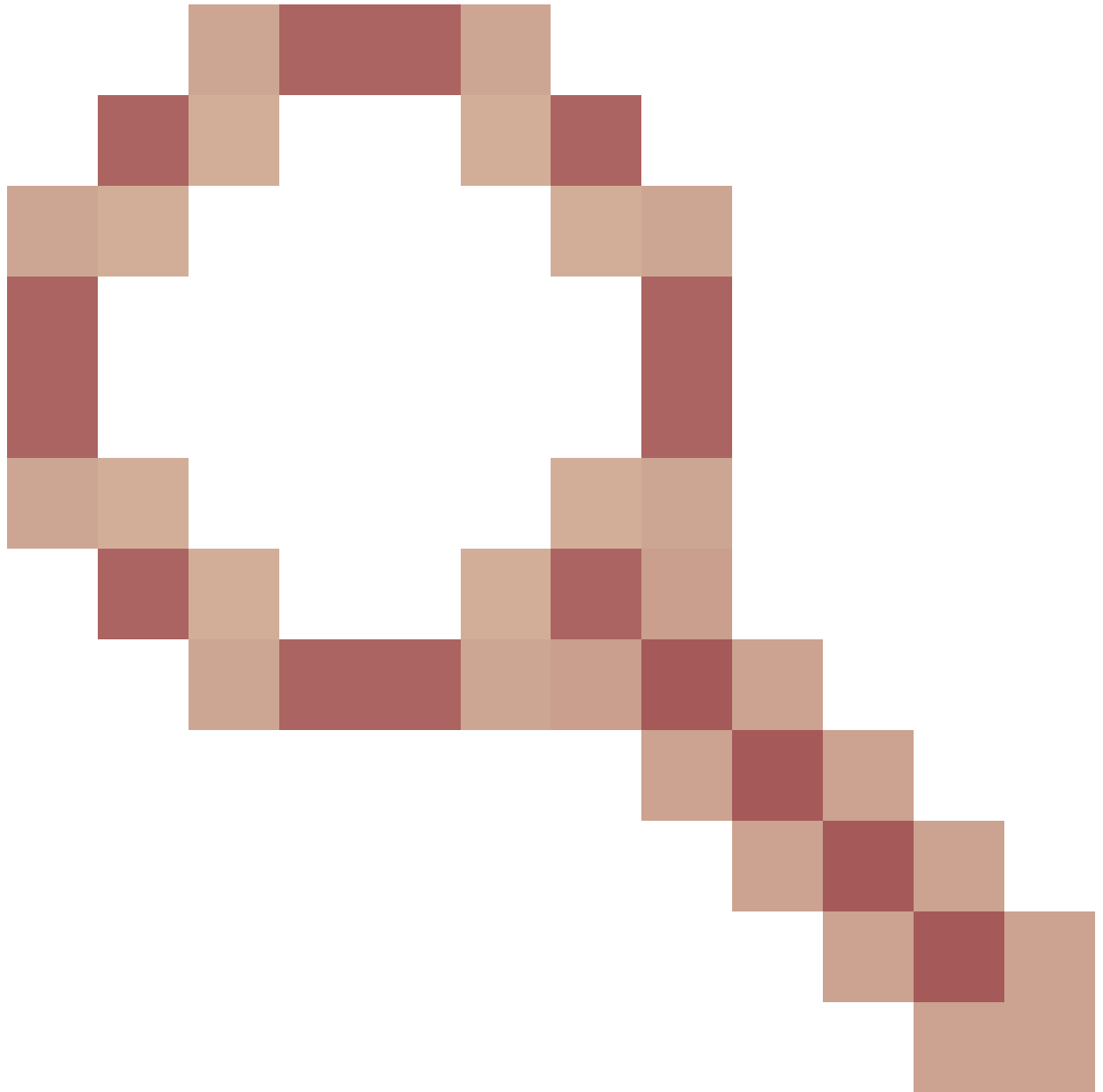
: il servizio Webproxy non viene riavviato se il nome host del proxy inverso non è risolvibile dal componente

- [CSCwa23252](#)



: il trust finesse proxy viene interrotto quando la profondità è maggiore di uno per la catena di certificati CA

- [CSCwa46459](#)




vulnerabilità di log4j pari a zero giorni esposta in webservice


## Note sull'aggiornamento per le configurazioni VPN senza ES01

- La configurazione ES03 richiede l'installazione di Nginx con supporto Lua.
- Requisiti dei certificati
  - Cisco Finesse, CUIC e IdS richiederanno l'aggiunta del certificato host Nginx / OpenResty all'archivio di attendibilità Tomcat e il riavvio, prima che la configurazione di Nginx ES02 possa connettersi correttamente al server upstream.
  - I certificati dei server upstream Cisco Finesse, CUIC e IdS devono essere configurati nel server Nginx per utilizzare la configurazione basata su ES03.

---

 Nota: si consiglia di rimuovere la configurazione esistente di Nginx basata su ES01 prima di installare le configurazioni ES03 Nginx.

---

 Nota: gli script di configurazione ES03 richiedono anche l'installazione di ES03 COP

---

## Autenticazione

Finesse 12.6 ES03 introduce l'autenticazione al proxy. L'autenticazione è supportata per le distribuzioni Single Sign-On (SSO) e non SSO.

L'autenticazione viene applicata per tutte le richieste e i protocolli accettati nel proxy prima di essere inoltrati ai server componenti upstream, dove viene eseguita anche l'autenticazione applicata localmente dai server componenti. Tutte le autenticazioni utilizzano le credenziali di accesso comuni di Finesse per autenticare le richieste.

Le connessioni permanenti, come i websockets che si basano su protocolli applicativi come Extensible Messaging and Presence Protocol (XMPP) per l'autenticazione e la post-connessione, vengono autenticate sul proxy convalidando l'indirizzo IP da cui è stata effettuata l'autenticazione dell'applicazione prima di stabilire la connessione socket.

### Autenticazione non SSO

L'autenticazione non SSO non richiede configurazioni aggiuntive e funziona con gli script di configurazione Nginx predefiniti una volta effettuate le sostituzioni di script richieste.

L'autenticazione si basa sul nome utente e sulla password utilizzati per accedere a Finesse. L'accesso a tutti gli endpoint verrà convalidato con i servizi di autenticazione Finesse.

L'elenco degli utenti validi viene memorizzato nella cache locale del proxy (aggiorna la cache ogni 15 minuti), che viene utilizzato per convalidare l'utente in una richiesta. Le credenziali utente vengono convalidate inoltrando la richiesta all'URI Finesse configurato e quindi l'hash delle credenziali viene memorizzato nella cache locale (15 minuti) per autenticare le nuove richieste localmente. In caso di modifica del nome utente o della password, la modifica sarà effettiva solo dopo 15 minuti.

### Autenticazione SSO

L'autenticazione SSO richiede che l'amministratore configuri la chiave di crittografia del token IdS nel server Nginx all'interno del file di configurazione. La chiave di crittografia del token IdS può essere ottenuta dal server IdS con il comando `show ids secret CLI`. La chiave deve essere configurata come parte di una delle sostituzioni di `#Must-change` che l'amministratore deve eseguire negli script prima che l'autenticazione SSO possa funzionare.

Fare riferimento alla guida per l'utente SSO per le configurazioni SAML IdS da eseguire affinché la risoluzione proxy funzioni per IdS.

Una volta configurata l'autenticazione SSO, è possibile utilizzare una coppia di token valida per accedere a qualsiasi endpoint nel sistema. La configurazione proxy convalida le credenziali intercettando le richieste di recupero del token effettuate agli IdS o decrittografando i token validi e quindi memorizzandoli nella cache locale per ulteriori convalide.

## Autenticazione per connessioni Websocket

Le connessioni Websocket non possono essere autenticate con l'intestazione di autorizzazione standard, poiché le intestazioni personalizzate non sono supportate dalle implementazioni websocket native nel browser. Protocolli di autenticazione a livello di applicazione, in cui le informazioni di autenticazione contenute nel payload non impediscono la connessione al socket Web e, di conseguenza, entità dannose possono eseguire attacchi DOS o DDOS semplicemente creando innumerevoli connessioni per sopraffare il sistema.

Per ridurre questa possibilità, le configurazioni di proxy inverso fornite da Ingnix prevedono controlli specifici per consentire l'accettazione delle connessioni al socket Web SOLO dagli indirizzi IP che hanno effettuato una richiesta REST autenticata prima di stabilire la connessione al socket Web. Ciò significa che i client che tentano di creare connessioni a socket Web prima di inviare una richiesta REST riceveranno ora un errore di autorizzazione non riuscita e non è uno scenario di utilizzo supportato.

## Prevenzione degli attacchi di forza bruta

Gli script di autenticazione Finesse 12.6 ES02 prevengono attivamente gli attacchi di forza bruta che possono essere utilizzati per indovinare la password dell'utente. A tale scopo, blocca l'indirizzo IP utilizzato per accedere al servizio, dopo un certo numero di tentativi non riusciti in breve tempo. Queste richieste verranno rifiutate da un errore client 418. I dettagli degli indirizzi IP bloccati sono accessibili dai file <nginx-install-directory>/logs/blocking.log e <nginx-install-directory>/logs/error.log.

È possibile configurare il numero di richieste non riuscite, l'intervallo di tempo e la durata del blocco. Le configurazioni sono presenti nel file <nginx-install-directory>/conf/conf.d/maps.conf.

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

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

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

## Registrazione

Per trovare gli indirizzi IP bloccati, eseguire i comandi seguenti dalla directory <nginx-install-directory>/logs.

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

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

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

Per aggiungere il blocco alle regole IPtable/firewall, si consiglia l'integrazione con Fail2ban o simili.

## Installazione e configurazione di Fail2ban

Fail2ban analizza i file di log e gli IP di blocco che mostrano i segnali dannosi - troppi errori di password, ricerca di exploit, ecc. In genere, Fail2Ban viene quindi utilizzato per aggiornare le regole del firewall per rifiutare gli indirizzi IP per un determinato periodo di tempo, anche se è possibile configurare qualsiasi altra azione arbitraria (ad esempio l'invio di un messaggio di posta elettronica). Per ulteriori informazioni, visitare il sito <https://www.fail2ban.org/>.

Fail2ban può essere configurato per monitorare il blocking.log per identificare gli indirizzi IP bloccati da Nginx al rilevamento di attacchi bruteforce e per bloccarli per una durata configurabile. Per installare e configurare fail2ban su un proxy inverso CentOS, procedere come segue:

### 1. Installare Fail2ban utilizzando yum.

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

### 2. Creare una prigione locale.

Le configurazioni delle carceri consentono all'amministratore di configurare varie proprietà, come le porte a cui non è consentito l'accesso da parte di qualsiasi indirizzo IP bloccato, la durata del blocco dell'indirizzo IP, la configurazione del filtro utilizzata per identificare l'indirizzo IP bloccato dal file di registro monitorato, ecc. Per aggiungere una configurazione personalizzata che impedisca l'accesso ai server upstream agli indirizzi IP bloccati, procedere come segue:

2.1. Andare alla directory di installazione Fail2ban (in questo esempio /etc/fail2ban)

```
cd /etc/fail2ban
```

2.2. Fare una copia di jail.conf in jail.local per mantenere le modifiche locali isolate.

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

2.3. Aggiungere queste configurazioni di carceri alla fine del file jail.local e sostituire le porte nel modello con quelle effettive. Aggiornare le configurazioni dei tempi di interdizione in base alle esigenze.

```
# 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. Configurare un filtro.

Un filtro indica a Fail2ban gli elementi da cercare nei log per identificare l'host da escludere. Per creare un filtro, procedere come segue:

3.1. Creare filter.d/finesseban.conf.

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

3.2. Aggiungere queste righe nel file filter.d/finesseban.conf.

[Definition]

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

#### 4. Avviare Fail2ban.

Eseguire questo comando per avviare fail2ban.

```
fail2ban-client start
```

Aprire i file di log fail2ban e verificare che non siano presenti errori. Per impostazione predefinita, i log per fail2ban vengono inseriti nel file `/var/log/fail2ban.log`.

### Convalida URL risorse statiche

Tutti gli endpoint validi a cui è possibile accedere in modo non autenticato vengono attivamente registrati negli script ES03.

Le richieste a questi percorsi non autenticati vengono attivamente rifiutate, se viene richiesto un URI non valido, senza inviare tali richieste al server upstream.

### Memorizzazione nella cache delle intestazioni CORS


Quando la prima richiesta di opzioni ha esito positivo, le intestazioni di risposta `access-control-allow-headers`, `access-control-allow-origin`, `access-control-allow-method`, `access-control-expose-headers` e `access-control-allow-credentials` vengono memorizzate nella cache del proxy per cinque minuti. Queste intestazioni vengono memorizzate nella cache per ogni server upstream corrispondente.

## Configurazione

Questo documento descrive la configurazione di Nginx come proxy inverso da utilizzare per abilitare l'accesso Finesse VPN-Less. Versioni del componente della soluzione UCCE, del proxy e del sistema operativo utilizzate per verificare le istruzioni fornite. Le istruzioni pertinenti devono essere adattate al sistema operativo/proxy prescelto.

- Versione Nginx utilizzata - OpenResty 1.19.9.1
- Sistema operativo utilizzato per la configurazione - CentOS 8.0

---

 Nota: la configurazione Nginx descritta può essere scaricata dalla [pagina di download del software Finesse release 12.6\(1\)ES3](#).

---

Configura i componenti della soluzione per l'accesso VPN Less



Dopo aver configurato il proxy, configurare i componenti della soluzione (Finesse/ CUIC / IdS) per l'accesso VPN Less con il nome host e l'IP pianificati del proxy/servizi utilizzati per accedere alla soluzione con questi comandi.

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

Per ulteriori informazioni su questi comandi, consultare la [UCCE 12.6 Feature Guide](#) (Guida alle funzionalità di UCCE 12.6) prima di consultare il presente documento.

## Installare OpenResty come proxy inverso in DMZ


In questa sezione vengono illustrati in dettaglio i passaggi per l'installazione del proxy basato su OpenResty. Il proxy inverso è in genere configurato come dispositivo dedicato nella zona demilitarizzata di rete (DMZ, Network Demilitarized Zone), come illustrato nel diagramma di distribuzione menzionato in precedenza.

1. Installare il sistema operativo desiderato con le specifiche hardware richieste. Le modifiche ai parametri del kernel e IPv4 possono variare a seconda del sistema operativo selezionato e si consiglia agli utenti di verificare nuovamente questi aspetti se la versione del sistema operativo scelta è diversa.
2. Configurare due interfacce di rete. Per l'accesso pubblico da client Internet sarà necessaria un'interfaccia e un'altra per comunicare con i server della rete interna.
3. Installare [OpenResty](#).

Qualsiasi sapore di Nginx può essere utilizzato per questo scopo, purché siano basati su Nginx 1.19+ e supportino Lua:

- Nginx Plus
- Nginx Open Source (Nginx open source dovrà essere compilato insieme ai moduli Lua basati su OpenResty per poter essere utilizzato)
- ApriRiposo
- Funzionalità aggiuntive GetPageSpeed

---

 Nota: la configurazione fornita è stata testata con OpenResty 1.19 e dovrebbe funzionare con altre distribuzioni con solo aggiornamenti minori, se presenti.

---

## Installazione di OpenResty

1. Installare OpenResty. Vedere [Pacchetti Linux OpenResty](#). Nell'ambito dell'installazione di OpenResty, Nginx verrà installato in questa posizione e il percorso di OpenResty verrà aggiunto alla variabile PATH aggiungendo nel file ~/.bashrc.

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

## 2. Avviare/arrestare Nginx.


- Per avviare Nginx, immettere `openresty`.
- Per arrestare Nginx, immettere `openresty -s stop`.

## Configura Nginx

La configurazione viene spiegata per un'installazione Nginx basata su OpenResty. Le directory predefinite per OpenResty sono:

- `<nginx-install-directory>` = `/usr/local/openresty/nginx`
  - `<directory-installazione-Openresty>` = `/usr/local/openresty`
1. Scaricare ed estrarre il file dalla [pagina di download del software Finesse release 12.6\(1\)ES03](#) (12.6-ES03-reverse-proxy-config.zip) che contiene la configurazione del proxy inverso per Nginx.
  2. Copiare `nginx.conf`, `nginx/conf.d/`, e `nginx/html/` dalla directory di configurazione del proxy inverso estratto in `<nginx-install-directory>/conf`, `<nginx-install-directory>/conf/conf.d/`, e `<nginx-install-directory>/html/` rispettivamente.
  3. Copiare la directory `nginx/lua` dalla directory di configurazione del proxy inverso estratta all'interno della `<directory-installazione-nginx>`.
  4. Copiare il contenuto di `lua` in `<Openresty-install-directory>/lua/lib/resty`.
  5. Configurare la rotazione del log di Nginx copiando il file `nginx/logrotate/saproxy` nella cartella `<nginx-install-directory>/logrotate/`. Modificare il contenuto del file in modo che faccia riferimento alle directory di log corrette se non vengono utilizzate le impostazioni predefinite di Nginx.
  6. Nginx deve essere eseguito con un account di servizio dedicato non privilegiato, che deve essere bloccato e avere una shell non valida (o applicabile per il sistema operativo scelto).
  7. Trovare la stringa "Must-change" nei file delle cartelle estratte denominate `html` e `conf.d` e sostituire i valori indicati con le voci appropriate.
  8. Accertarsi che vengano effettuate tutte le sostituzioni obbligatorie, descritte con i commenti Must-change nei file di configurazione.
  9. Assicursi che le directory della cache configurate per CUI e Finesse vengano create in `<nginx-install-directory>/cache` insieme a queste directory temporanee.
    - `<directory-installazione-nginx>/cache/client_temp`
    - `<directory-installazione-nginx>/cache/proxy_temp`

---

 Nota: la configurazione fornita è per un'implementazione di esempio 2000 e deve essere espansa in modo appropriato per un'implementazione più ampia.

---

## Configurazione della cache Nginx

Per impostazione predefinita, i percorsi della cache proxy vengono memorizzati nel file system. È consigliabile sostituirli con unità in memoria creando un percorso di cache in `tmpfs`, come mostrato di seguito.

## 1. Creare directory per i diversi percorsi della cache proxy in /home.

Ad esempio, queste directory devono essere create per Finesse primario. Gli stessi passaggi devono essere seguiti per i server secondari Finesse e CUIC.

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

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



**Nota:** aumentare il client e le cache proxy\_temp di 1 GB per ogni nuovo cluster Finesse aggiunto alla configurazione.

---

2. Montare i nuovi punti di montaggio con il comando **mount -av**.
3. Verificare che il file system abbia montato i nuovi punti di montaggio con il **df -h** comando.
4. Modificate le posizioni proxy\_cache\_path nei file di configurazione della cache Finesse e CUIC.


Ad esempio, per modificare i percorsi per il file primario Finesse, passare a <nginx-install-directory>conf/conf.d/finesse/caches e modificare il percorso della cache esistente /usr/local/openresty/nginx/cache/finesse25/ nel nuovo percorso del file system creato /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. Seguire la stessa procedura per i server Finesse secondario e CUIC.

---

 Nota: assicurarsi che la somma di tutte le dimensioni delle unità tmpfs create in tutti i passi precedenti venga aggiunta al dimensionamento finale della memoria per la distribuzione, poiché queste unità sono blocchi di memoria configurati per assomigliare ai dischi dell'applicazione e consumano lo stesso spazio di memoria.

---

## Configura certificati SSL

Usa certificati autofirmati - Distribuzioni di test

I certificati autofirmati devono essere utilizzati solo fino a quando il proxy inverso non è pronto per l'implementazione in produzione. In una distribuzione di produzione, utilizzare solo un certificato firmato da un'Autorità di certificazione (CA).

1. Genera certificati Nginx per il contenuto della cartella SSL. Prima di generare i certificati, è necessario creare una cartella denominata `ssl` in `/usr/local/openresty/index`. Con questi comandi è necessario generare due certificati (uno per `<reverseproxy_primary_fqdn>` e l'altro per `<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` (passare il nome host come: `<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` (passare il nome host come `:<reverseproxy_secondary_fqdn>`)
  - c. Verificare che il percorso del certificato sia `/usr/local/openresty/nginx/ssl/nginx.crt` e `/usr/local/openresty/nginx/ssl/nginxnode2.crt`, poiché questi sono già configurati nei file di configurazione di Finesse Nginx.
2. Modificare l'autorizzazione della chiave privata 400 (r—).
3. Configurare il firewall/[iptables](#) sul proxy inverso per consentire la comunicazione dal firewall in modo che corrisponda alle porte su cui il server Nginx è stato configurato per l'ascolto.
4. Aggiungere l'indirizzo IP e il nome host di Finesse, IdS e CUIC nella voce `/etc/hosts` sul server proxy inverso.
5. Fare riferimento alla guida alle funzionalità della soluzione per le configurazioni da eseguire sui server componenti per configurare l'host Nginx come proxy inverso.

---

 Nota: la configurazione fornita è per un'implementazione di esempio 2000 e deve essere espansa in modo appropriato per un'implementazione più ampia.

---

Usa certificato firmato CA - Distribuzioni di produzione

È possibile installare un certificato firmato dall'autorità di certificazione sul proxy inverso eseguendo la procedura seguente:

### 1. Generare la richiesta di firma del certificato (CSR).

Per generare la CSR e la chiave privata, `openssl req -new -newkey rsa:4096 -keyout nginx.key -out nginx.csr` metterla dopo aver effettuato l'accesso al proxy. Seguire il prompt e fornire i dettagli. In questo modo vengono generati il CSR (nginx.csr nell'esempio) e la chiave privata RSA (nginx.key nell'esempio) con una forza di 4096 bit.

Ad esempio:

```
[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
```

Prendere nota della passphrase PEM, in quanto verrà utilizzata per decrittografare la chiave privata durante la distribuzione.

### 2. Ottenere il certificato firmato dalla CA.

Inviare il CSR all'autorità di certificazione e ottenere il certificato firmato.

Nota: se il certificato ricevuto dall'autorità di certificazione non è una catena di certificati contenente tutti i rispettivi certificati, comporre tutti i certificati pertinenti in un unico file della catena di certificati.

### 3. Distribuire il certificato e la chiave.

Decrittografare la chiave generata in precedenza come parte del primo passaggio con `openssl rsa -in nginx.key -out nginx_decrypted.key` il comando. Posizionare il certificato firmato dalla CA e la chiave decrittografata nella cartella `/usr/local/openresty/nginx/ssl` nel computer proxy inverso. Aggiornare/aggiungere le configurazioni SSL relative al certificato nelle configurazioni Nginx nel file di configurazione `/usr/local/openresty/nginx/conf/conf.d/ssl/ssl.conf`.

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

### 4. Configurare le autorizzazioni per i certificati.

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

Immettere and 

```
chmod 400
```

`/usr/local/openresty/nginx/ssl/nginx_decrypted.key`, in modo che il certificato disponga di autorizzazioni di sola lettura e sia limitato al proprietario.

## 5. Riavviare Nginx.

Usa parametro Diffie-Hellman personalizzato

Creare un parametro Diffie-Hellman personalizzato con i seguenti comandi:

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

Modificare la configurazione del server per utilizzare i nuovi parametri nel file `/usr/local/openresty/nginx/conf/conf.d/ssl/ssl.conf`:

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

## Verifica attivazione associazione OCSP - Controllo revoca certificato

Nota: per abilitare questa funzionalità, è necessario che il server utilizzi un certificato firmato dalla CA e che abbia accesso alla CA che ha firmato il certificato.

Aggiungere/aggiornare questa configurazione nella cartella `file/usr/local/openresty/nginx/conf/conf.d/ssl/ssl.conf`:

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

## Configurazione Nginx

Il file di configurazione Nginx predefinito (`/usr/local/openresty/nginx/conf/nginx.conf`) deve essere modificato in modo da contenere queste voci per applicare la protezione e fornire prestazioni. Questo contenuto deve essere utilizzato per modificare il file di configurazione predefinito creato dall'installazione di Nginx.

```
# Increasing number of worker processes will not increase the processing the request. The number of worker processes should be set according to the number of CPU 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 general, the priority should be 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_con

worker_rlimit_nofile 102400;

events {
    multi_accept on;

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

http {

    include mime.types;

    default_type text/plain;

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

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

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

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

    init_worker_by_lua_block {
        local UsersListManager = require('users_list_manager')
        local UnauthenticatedDesktopResourcesManager = require("unauthenticated_desktopresources_manage")
        local UnauthenticatedResourcesManager = require("unauthenticated_thirdpartyresources_manager")
    }
}

```

```

-- Must-change Replace saproxy.cisco.com with reverseproxy fqdn

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

include conf.d/*.conf;

sendfile        on;

tcp_nopush     on;

server_names_hash_bucket_size 512;

```

## Configura porta proxy inversa

Per impostazione predefinita, la configurazione Nginx resta in ascolto sulla porta 8445 per le richieste Finesse. È possibile abilitare una sola porta alla volta da un proxy inverso per supportare le richieste Finesse, ad esempio 8445. Se la porta 443 deve essere supportata, modificare il file <nginx-install-directory>conf/conf.d/finesse.conf per abilitare l'ascolto su 443 e disabilitare l'ascolto su 8445.

## Configura autenticazione TLS reciproca tra proxy inverso e componenti upstream

L'autenticazione del certificato SSL del client per le connessioni dagli host proxy inverso può essere abilitata sui componenti upstream CCBU CUIC/Finesse/IdS/Livedata tramite la nuova opzione CVOS CLI che è

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

Per impostazione predefinita, questa opzione è disabilitata e deve essere abilitata in modo esplicito dall'amministratore tramite l'esecuzione della CLI su ciascun server upstream in modo indipendente. Se questa opzione è abilitata, il servizio proxy Web Cisco in esecuzione sull'host a monte inizierà l'autenticazione dei certificati client nell'handshake TLS per le connessioni originate da host di proxy inverso trusted aggiunti come parte dell'utilità CLI system reverse proxy allowed-hosts add <proxy-host>.

Di seguito è riportato il blocco di configurazione per lo stesso nei file di configurazione proxy, ossia ssl.conf e ssl2.conf

```

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

```

Il certificato SSL utilizzato per il traffico in uscita (da proxy a upstream) può essere uguale al



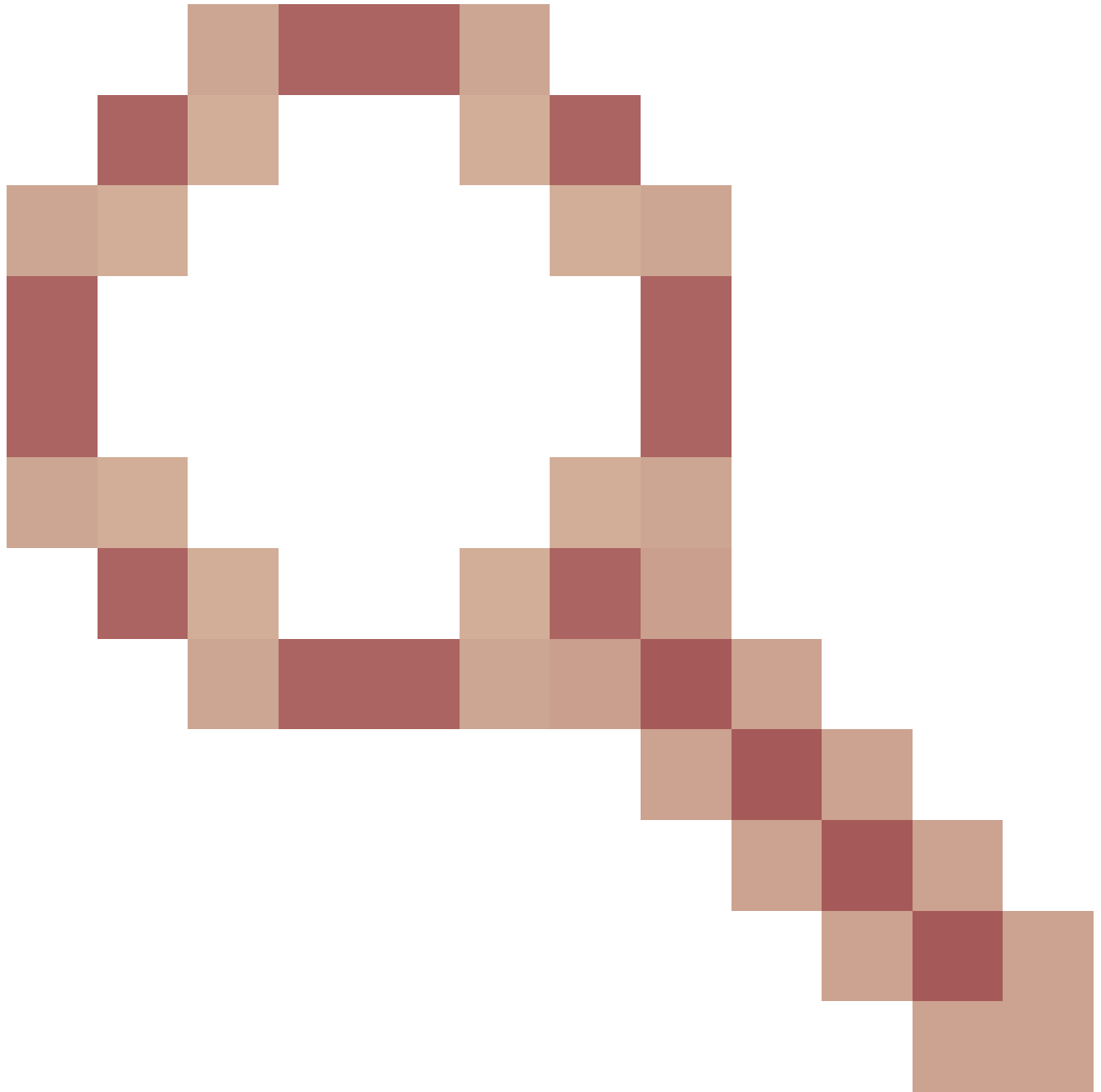
certificato SSL configurato per il traffico in entrata (connettore SSL per blocchi di server componenti). Se come proxy\_ssl\_certificate viene utilizzato un certificato autofirmato che deve essere caricato nei componenti upstream (Finesse/IdS/CUIC/Livedata) per poter autenticare correttamente l'archivio di attendibilità.

La convalida del certificato del server upstream tramite proxy inverso è facoltativa e disabilitata per impostazione predefinita. Se si desidera ottenere l'autenticazione reciproca TLS completa tra host reverse proxy e upstream, è necessario rimuovere il commento dalla configurazione sottostante dai file ssl.conf e ssl2.conf.

```
#Enforce upstream server certificate validation at proxy -> #this is not mandated as per CIS but 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
```

Avvertenze per la configurazione dell'autenticazione TLS reciproca:

- Una volta abilitata questa funzionalità sui componenti CCBU, il certificato client verrà richiesto ai client LAN anche durante l'handshake TLS. Se sul computer client sono installati certificati client/personali, i browser possono scegliere di visualizzare un popup per l'utente finale in cui viene richiesto di scegliere il certificato appropriato per l'autenticazione del client. Anche se non importa quale certificato l'utente finale sceglie o preme Annulla sulle richieste popup avrà successo, in quanto l'autenticazione del certificato del client non è applicata per i client LAN, ma ci sarà un cambiamento nell'esperienza. Fare riferimento a CDET [CSCwa26057](#)



per ulteriori dettagli.

- Il servizio webproxy dei componenti upstream non viene attivato se un host proxy viene aggiunto a allowed-list non risolvibile dal servizio webproxy. Verificare che gli host proxy inverso aggiunti all'elenco degli host consentiti siano risolvibili dal componente a monte tramite la ricerca DNS.

## Cancella cache

La cache del proxy inverso può essere cancellata con il

`/clearCache.sh`  
comando.

## Linee guida standard

Questa sezione descrive brevemente le linee guida standard da seguire quando si configura Nginx come server proxy.

Queste linee guida derivano da [Center for Internet Security](#). Per ulteriori dettagli su ciascuna linea guida, fare riferimento alla stessa.

1. Si consiglia sempre di utilizzare la versione stabile più recente di OpenResty e OpenSSL.
2. Si consiglia di installare Nginx in un montaggio su disco separato.
3. L'ID di processo Nginx deve essere di proprietà dell'utente root (o come applicabile per il sistema operativo scelto) e deve avere l'autorizzazione 644 (rw—) o una versione più restrittiva.
4. Nginx deve bloccare le richieste per gli host sconosciuti. Assicurarsi che ogni blocco server contenga la direttiva `server_name` definita in modo esplicito. Per procedere alla verifica, eseguire una ricerca in tutti i blocchi server nelle directory `nginx.conf` e `nginx/conf.d` e verificare che tutti i blocchi server contengano il `nome_server`.
5. Nginx deve eseguire l'ascolto solo sulle porte autorizzate. Cercare in tutti i blocchi server nelle directory `nginx.conf` e `nginx/conf.d` e verificare che le direttive di ascolto siano in ascolto per verificare che solo le porte autorizzate siano aperte per l'ascolto.
6. Poiché Cisco Finesse non supporta il protocollo HTTP, si consiglia di bloccare anche la porta HTTP del server proxy.
7. Il protocollo SSL Nginx deve essere TLS 1.2. È necessario rimuovere il supporto per i protocolli SSL legacy. Deve inoltre disabilitare le cifrature SSL deboli.
8. Si consiglia di inviare al server syslog remoto i log degli errori e degli accessi Nginx.
9. Si consiglia di installare il modulo `mod_security` che funziona come un firewall dell'applicazione Web. Per ulteriori informazioni, consultare il [manuale ModSecurity](#). Si noti che il caricamento Nginx non è stato verificato all'interno del modulo `mod_security` in uso.

## Configurare il file di mapping

La distribuzione reverse proxy del desktop Finesse richiede un file di mapping per configurare l'elenco delle combinazioni di nome host/porta visibili esternamente e il relativo mapping ai nomi e alle porte effettivi del server utilizzati dai server Finesse, IdS e CUIC. Questo file di mappatura configurato sui server interni è la configurazione chiave che consente ai client connessi tramite Internet di essere reindirizzati agli host e alle porte necessari utilizzati su Internet.

Il file di mapping deve essere distribuito in un server Web accessibile ai server componenti e il relativo URI deve essere configurato affinché la distribuzione funzioni. Si consiglia di configurare il file di mappatura utilizzando un server Web dedicato disponibile in rete. Se tale server non è disponibile, è possibile utilizzare il proxy inverso, che richiede che il proxy sia accessibile dall'interno della rete e presenta anche un rischio di esposizione delle informazioni a client esterni che possono effettuare l'accesso non autorizzato alla DMZ. Nella sezione successiva viene illustrato in dettaglio come eseguire questa operazione.

Fare riferimento alla guida alle funzionalità per i passaggi esatti per configurare l'URI del file di mapping su tutti i server componenti e per ulteriori dettagli su come creare i dati del file di mapping.

## Usa proxy inverso come file server di mapping

Questi passaggi sono necessari solo se il proxy inverso viene utilizzato anche come host dei file di mappatura proxy.

1. Configurare il nome host del proxy inverso nel controller di dominio utilizzato dagli host Finesse/CUIC e IdS in modo che sia possibile risolvere il relativo indirizzo IP.
2. Caricare i certificati firmati Nginx generati su entrambi i nodi in tomcat-trust di cmplatform e riavviare il server.
3. Aggiornare i valori Must-change in <NGINX\_HOME>/html/proxymap.txt.
4. Ricaricare le configurazioni Nginx con il `nginx -s reload` comando.
5. Verificare che il file di configurazione sia accessibile da un altro host di rete utilizzando il `curl` comando.

## Protezione avanzata kernel CentOS 8

Se il sistema operativo scelto è CentOS 8, si consiglia di utilizzare queste configurazioni sysctl per le installazioni che utilizzano un server dedicato per ospitare il proxy.

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

```
# Avoid a smurf attack
```

```
net.ipv4.icmp_echo_ignore_broadcasts = 1
```

```
# Turn on protection for bad icmp error messages
```

```
net.ipv4.icmp_ignore_bogus_error_responses = 1
```

```
# Turn on syncookies for SYN flood attack protection
```

```
net.ipv4.tcp_syncookies = 1
```

```
# Turn on and log spoofed, source routed, and redirect packets
```

```
net.ipv4.conf.all.log_martians = 1
```

```
net.ipv4.conf.default.log_martians = 1
```

```
# Turn off routing
```

```
net.ipv4.ip_forward = 0
```

```
net.ipv4.conf.all.forwarding = 0
```

```
net.ipv6.conf.all.forwarding = 0
```

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

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

```
# Block routed packets
```

```
net.ipv4.conf.all.accept_source_route = 0
```

```
net.ipv4.conf.default.accept_source_route = 0
```

```
net.ipv6.conf.all.accept_source_route = 0
```

```
net.ipv6.conf.default.accept_source_route = 0
```

```
# Block ICMP redirects
```

```
net.ipv4.conf.all.accept_redirects = 0
```

```
net.ipv4.conf.default.accept_redirects = 0
```

```
net.ipv6.conf.all.accept_redirects = 0
```

```
net.ipv6.conf.default.accept_redirects = 0
net.ipv4.conf.all.secure_redirects = 0
net.ipv4.conf.default.secure_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0

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

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

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

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

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

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

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

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

# Randomize virtual address space
kernel.randomize_va_space = 2

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

# Disable SysReq
kernel.sysrq = 0
```

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

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

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

Dopo aver apportato le modifiche consigliate, si consiglia di riavviare il sistema.


## Protezione avanzata tabelle IP

IPtables è un'applicazione che consente a un amministratore di sistema di configurare le tabelle, le catene e le regole IPv4 e IPv6 fornite dal firewall del kernel Linux.

Queste regole IPtables sono configurate per proteggere l'applicazione proxy dagli attacchi di forza bruta limitando l'accesso nel firewall del kernel Linux.

I commenti nella configurazione indicano quale servizio è soggetto a limitazioni di velocità utilizzando le regole.

---

 Nota: se gli amministratori utilizzano una porta diversa o espandono l'accesso a più server utilizzando le stesse porte, è necessario eseguire il ridimensionamento appropriato di queste porte in base a questi numeri.

---

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

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

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

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

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

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

```
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
```

```
# Configuration for finesse 8445 port
```

```
-A INPUT -p tcp -m tcp --dport 8445 --tcp-flags SYN SYN -m connlimit --connlimit-above 10 --connlimit-ma
-A INPUT -p tcp -m tcp --dport 8445 --tcp-flags SYN SYN -m connlimit --connlimit-above 10 --connlimit-ma
-A INPUT -p tcp -m tcp --dport 8445 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 6/sec --hashlimit
-A INPUT -p tcp -m tcp --dport 8445 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG -
-A INPUT -p tcp -m tcp --dport 8445 --tcp-flags SYN SYN -j DROP
```

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

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

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

```
-A INPUT -p tcp -m tcp --dport 443 --tcp-flags SYN SYN -m connlimit --connlimit-above 8 --connlimit-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-
-A INPUT -p tcp -m tcp --dport 15280 --tcp-flags SYN SYN -m connlimit --connlimit-above 30 --connlimit-
-A INPUT -p tcp -m tcp --dport 15280 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 20/sec --hashlim
-A INPUT -p tcp -m tcp --dport 15280 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG
-A INPUT -p tcp -m tcp --dport 15280 --tcp-flags SYN SYN -j DROP
```

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

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

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

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

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

```
-A INPUT -p tcp -m tcp --dport 8447 --tcp-flags SYN SYN -m connlimit --connlimit-above 6 --connlimit-ma
-A INPUT -p tcp -m tcp --dport 8447 --tcp-flags SYN SYN -m connlimit --connlimit-above 6 --connlimit-ma
-A INPUT -p tcp -m tcp --dport 8447 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 2/sec --hashlimit
```

```

-A INPUT -p tcp -m tcp --dport 8447 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG -
-A INPUT -p tcp -m tcp --dport 8447 --tcp-flags SYN SYN -j DROP

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

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

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

COMMIT

```

Queste regole possono essere applicate direttamente modificando manualmente `/etc/sysconfig/iptables` oppure salvando la configurazione in un file come `iptables.conf` ed eseguendo `cat iptables.conf >>/etc/sysconfig/iptables` per applicarle.

Dopo l'applicazione delle regole, è necessario riavviare il servizio IPTables. Immettere `systemctl restart iptables` per riavviare il servizio IPTables.

## Limita connessioni client

Oltre alla precedente configurazione delle tabelle IP, le installazioni che conoscono l'intervallo di indirizzi per i client che utilizzano il proxy sono consigliate di utilizzare questa conoscenza per proteggere le regole di accesso al proxy. Ciò può fornire enormi vantaggi quando si tratta di proteggere il proxy da botnet di reti maligne che sono spesso creati nella gamma di indirizzi IP di paesi che hanno regole più lassiste per quanto riguarda la sicurezza online. Si consiglia pertanto di limitare gli intervalli di indirizzi IP agli intervalli di indirizzi IP basati su paese/stato o ISP, se si è certi dei modelli di accesso.

## Blocca connessioni client

È inoltre utile sapere come bloccare un intervallo specifico di indirizzi quando si identifica un attacco proveniente da un indirizzo IP o da un intervallo di indirizzi IP. In questi casi, le richieste provenienti da questi indirizzi IP possono essere bloccate con regole modificabili.

### Blocca indirizzi IP distinti

Per bloccare più indirizzi IP distinti, aggiungere una riga al file di configurazione IPTables per ogni indirizzo IP.



Ad esempio, per bloccare gli indirizzi 192.0.2.3 e 192.0.2.4, immettere:

```
<#root>
```

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

Blocca un intervallo di indirizzi IP

Bloccare più indirizzi IP in un intervallo e aggiungere una singola riga al file di configurazione IPTables con l'intervallo IP.

Ad esempio, per bloccare gli indirizzi da 192.0.2.3 a 192.0.2.35, immettere:

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

Blocca tutti gli indirizzi IP in una subnet

Bloccare tutti gli indirizzi IP di un'intera subnet aggiungendo una singola riga al file di configurazione IPTables con la notazione di routing tra domini senza classe per l'intervallo di indirizzi IP. Ad esempio, per bloccare tutti gli indirizzi di classe C, immettere:

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

## SELinux

SELinux è una struttura di sicurezza della piattaforma integrata come miglioramento nel sistema operativo Linux. La procedura per installare e aggiungere i criteri SELinux per eseguire OpenResty come proxy inverso è fornita di seguito.

1. Arrestare il processo con il `openresty -s stop` comando.
2. Configurare e avviare `/stop nginx server` con il `systemctl` comando in modo che durante l'avvio il processo OpenResty venga avviato automaticamente. Immettere questi comandi come utente root.
  - a. Andare su `/usr/lib/systemd/system`.

- b. Aprire il file denominato `openresty.service`.
- c. Aggiornare il contenuto del file in base alla posizione `PIDFile`.

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

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

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

- d. Come utente `root`, immettere `sudo systemctl enable openresty`.
- e. Avviare/arrestare il servizio `OpenResty` con il `systemctl start openresty / systemctl stop openresty` comando e verificare che il processo venga avviato/arrestato come utente `root`.

## 1. Installa SELinux

- Per impostazione predefinita, solo alcuni pacchetti SELinux verranno installati in CentOS.
- Per generare la policy SELinux è necessario installare il pacchetto `policycoreutils-devel` e le relative dipendenze.
- Immettere questo comando per installare `policycoreutils-devel`

```
yum install policycoreutils-devel
```

- Verificare che, dopo l'installazione del pacchetto, il `sepolicy` comando funzioni.

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

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

```
SELinux Policy Inspection Tool
```

## 2. Creazione di un nuovo utente Linux e mappatura con l'utente SELinux

- a. Immettere `semanage login -l` per visualizzare il mapping tra gli utenti Linux e gli utenti SELinux.

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

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

- b. Come utente root, creare un nuovo utente Linux (utente Linux) mappato all'utente user\_u SELinux.

```
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. Per visualizzare il mapping tra nginxuser e user\_u, immettere questo comando come root:

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

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

- d. SELinux \_\_default\_\_ accesso mappato per impostazione predefinita all'utente SELinux unconfined\_u. Per impostazione predefinita, user\_u deve essere limitato da questo comando:

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

Per verificare che il comando funzioni correttamente, immettere `semanage login -l`. Dovrebbe produrre questo output:

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

e. Modificare nginx.conf ed eseguire la modifica della proprietà per nginxuser.

- i. Immettere `chown -R nginxuser:nginxuser *` nella directory <Openresty-install-directory>.
- ii. Modificare il file nginx.conf per includere nginxuser come utente per l'esecuzione dei processi di lavoro.

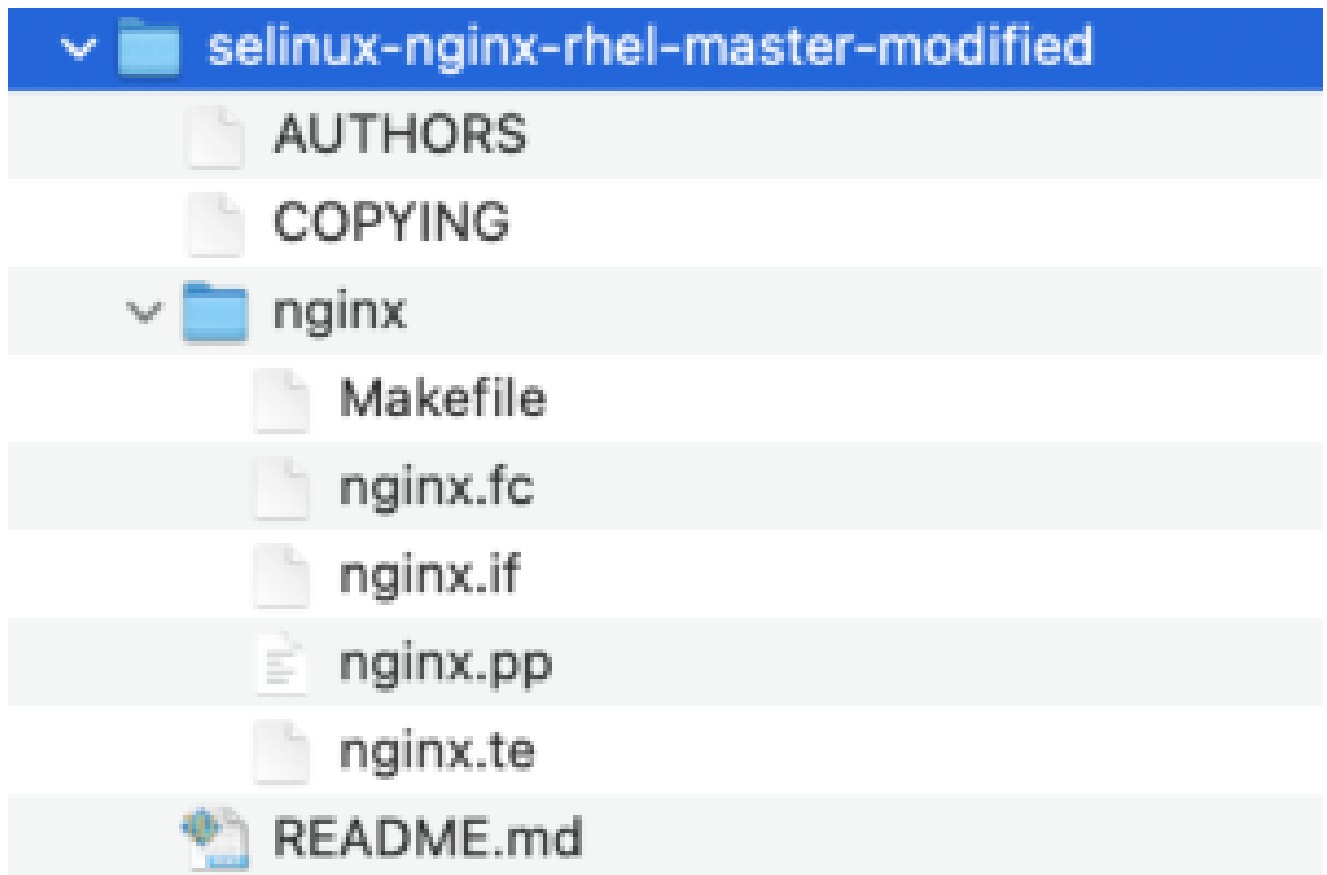
```

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

```

Scrivi la policy SELinux per Nginx

1. Anziché generare un nuovo criterio personalizzato predefinito per Nginx con il `sepolicy generate --init /usr/bin/nginx` comando, è preferibile iniziare con un criterio esistente.
2. I file `nginx.fc` (file dei contesti dei file) e `nginx.te` (file di imposizione del tipo) scaricati dall'URL fornito sono stati modificati per adattarli all'utilizzo del proxy inverso.
3. Questa versione modificata può essere utilizzata come riferimento poiché è stata corretta per lo Use Case specifico.
4. Scaricare il file `selinux-ninx-rhel-master-modified.tar` dalla [pagina di download del file software](#).



5. Estrarre il file .tar e passare alla directory nginx al suo interno.
6. Aprire il file .fc e verificare i percorsi di file richiesti del programma di installazione di nginx, della cache e del file pid.
7. Compilare la configurazione con il `make` comando.
8. Il file `nginx.pp` verrà generato.
9. Caricare il criterio con il `semodule` comando.

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

10. Passare a `/root` e creare un file vuoto denominato `touch /.autorelabel`.
11. Riavviare il sistema.
12. Immettere questo comando per verificare che il criterio sia stato caricato correttamente.

```
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 deve essere eseguito senza alcuna violazione. (Le violazioni saranno disponibili in /var/log/messages e /var/log/audit/audit.log).
14. Immettere questo comando per controllare lo stato di Nginx.

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

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

15. A questo punto dovrebbe essere possibile accedere al desktop dell'agente/supervisore Finesse.

## Verifica

Fare riferimento a questa sezione per verificare che la configurazione funzioni correttamente.

## Finesse

1. Richiedere <https://<reverseproxy:port>/finesse/api/SystemInfo>. dalla DMZ e verificare se sono raggiungibili.
2. Verificare che i valori <host> in <primaryNode> e <secondaryNode> siano nomi host proxy inverso validi. Non devono essere nomi host Finesse.

## CUIC e Live Data

1. Se nella risposta vengono visualizzati i nomi host Finesse anziché i nomi host proxy inverso, convalidare le configurazioni di mapping dei proxy e gli host consentiti vengono aggiunti correttamente nei server Finesse come descritto nella sezione "Populate Network"

Translation Data" di "VPN-Less Access to Finesse Desktop" nella [Guida alle funzionalità UCCE di Finesse 12.6](#).

2. Se i gadget LiveData vengono caricati correttamente in Finesse Desktop, le configurazioni dei proxy CUIC e LiveData sono corrette.
3. Per convalidare la configurazione di CUIC e LiveData, effettuare le richieste HTTP a questi URL dalla DMZ e verificare se sono raggiungibili.
  - [https://<reverseproxy:cuic\\_port>/cuic/rest/informazioni](https://<reverseproxy:cuic_port>/cuic/rest/informazioni)
  - [https://<reverseproxy:ldweb\\_port>/livedata/security](https://<reverseproxy:ldweb_port>/livedata/security)
  - [https://<reverseproxy:ldsocketio\\_port>/security](https://<reverseproxy:ldsocketio_port>/security)

## IDS

Per convalidare la configurazione IdS, attenersi alla seguente procedura:

1. Accedere all'interfaccia IdSAdmin all'indirizzo [https://<ids\\_LAN\\_host:ids\\_port>:8553/idsadmin](https://<ids_LAN_host:ids_port>:8553/idsadmin) dalla LAN, in quanto l'interfaccia di amministrazione non è esposta tramite il proxy inverso.
2. Scegliete Impostazioni > Attendibilità IdS.
3. Verificare che il nodo del server di pubblicazione del cluster proxy sia elencato nella pagina Scarica metadati SP e fare clic su Avanti.
4. Verificare che il proxy IDP sia visualizzato correttamente se configurato nella pagina Carica metadati IDP e fare clic su Avanti.
5. Avviare il test dell'SSO tramite tutti i nodi del cluster proxy dalla pagina Test dell'SSO e verificare che tutte le operazioni abbiano esito positivo. Per invertire i nodi proxy è necessaria la connettività del computer client.

## Prestazioni

L'analisi dei dati relativi all'acquisizione di prestazioni di livello superiore equivalente, eseguita con lo strumento nmon, è disponibile nella [pagina di download del software Finesse release 12.6\(1\) ES03](#) (load\_result.zip). I dati rappresentano lo stato del proxy per le operazioni di desktop e supervisor su un'implementazione UCCE di esempio di 2000 utilizzando login SSO e rapporti LD CUIC come configurato nel layout predefinito per 2000 utenti per un periodo di otto ore. Può essere utilizzato per determinare i requisiti di computer, disco e rete per un'installazione che utilizza Nginx su hardware paragonabile.

## Risoluzione dei problemi

### SSO

1. Reindirizzamenti del desktop non disponibili tramite proxy
  1. Verificare che i nomi host siano configurati in casi corretti in base ai nomi host effettivi della macchina virtuale in diverse configurazioni, ad esempio proxymap.txt, file server\_filter e così via.
  2. Verificare che l'IDS sia stato aggiunto con il nome host corretto in maiuscole/minuscole

nell'inventario CCE, in quanto le stesse informazioni vengono inviate ai componenti quando registrati per l'SSO dall'amministratore Web CCE.

## 2. Accessi SSO non eseguiti

1. Verificare che l'attendibilità IdS-IDP sia stata stabilita per l'host proxy.

## SELinux

1. Se Nginx non viene avviato per impostazione predefinita o se il desktop dell'agente Finesse non è accessibile, impostare SELinux sulla modalità permissiva con questo comando:

```
setenforce 0
```

2. Provare a riavviare Nginx con il `systemctl restart nginx` comando.
3. Le violazioni saranno disponibili in `/var/log/messages` e `/var/log/audit/audit.log`.
4. È necessario rigenerare il file `.te` con regole di autorizzazione per la risoluzione di tali violazioni mediante uno qualsiasi dei seguenti comandi:

```
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. Aggiornare il file `nginx.te` originale presente nella directory `selinux-nginx-rhel-master-modified/nginx` con le nuove regole di autorizzazione generate.
6. Compilare lo stesso con il `make` comando.
7. Il file `nginx.pp` verrà rigenerato.
8. Caricare il criterio tramite il comando `semodule`.

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

9. Impostare SELinux per applicare la modalità con questo comando:

```
setenforce
```

10. Riavviare il sistema.
11. Ripetere questa procedura finché non vengono corrette le violazioni richieste.



## Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l'accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).