

# Configurar o Nginx Reverse Proxy para acesso sem VPN ao Cisco Finesse (12.6 ES03)

## Contents

---

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Informações de Apoio](#)

[Alterações no ES03](#)

[Notas de atualização para configurações sem VPN baseadas em ES01](#)

[Autenticação](#)

[Autenticação não SSO](#)

[Autenticação SSO](#)

[Autenticação para Conexões Websocket](#)

[Prevenção de ataque de força bruta](#)

[Registro](#)

[Instalar e Configurar Fail2ban](#)

[Validar URLs de Recursos Estáticos](#)

[Armazenando Cabeçalhos CORS em Cache](#)

[Configurar](#)

[Configurar componentes da solução para VPN com menos acesso](#)

[Instalar o OpenResty como um proxy reverso na DMZ](#)

[Instalação do OpenResty](#)

[Configurar Nginx](#)

[Configurar o cache do Nginx](#)

[Configurar Certificados SSL](#)

[Usar parâmetro Diffie-Hellman personalizado](#)

[Verifique se o Grampeamento OCSP está Habilitado - Verificação de Revogação de Certificado](#)

[Configuração Do Nginx](#)

[Configurar porta de proxy reverso](#)

[Configurar a autenticação TLS mútua entre o proxy reverso e os componentes upstream](#)

[Limpar cache](#)

[Diretrizes padrão](#)

[Configurar o Arquivo de Mapeamento](#)

[Usar Proxy Reverso como o Servidor de Arquivos de Mapeamento](#)

[Endurecimento de kernel CentOS 8](#)

[Protegendo IPtables](#)

[Restringir Conexões de Cliente](#)

[Bloquear conexões de cliente](#)

[Bloquear endereços IP distintos](#)

[Bloquear um intervalo de endereços IP](#)

[Bloquear todos os endereços IP em uma sub-rede](#)

---

[SELinux](#)

[Verificar](#)

[Finesse](#)

[CUIC e Live Data](#)

[IDS](#)

[Desempenho](#)

[Troubleshooting](#)

[SSO](#)

---

## Introdução

Este documento descreve como usar um proxy reverso para acessar o desktop do Cisco Finesse sem se conectar a uma VPN baseada nas versões 12.6 ES03 do Cisco Finesse, Cisco Unified Intelligence Center (CUIC) e Cisco Identity Service (IdS).

---

 Observação: a instalação e a configuração do Nginx não são suportadas pela Cisco. As consultas sobre esse assunto podem ser discutidas nos [fóruns da comunidade Cisco](#).

---

 Observação: para as implantações ES03 de VPN-less, consulte o arquivo readme dos componentes individuais para planejar as atualizações e verificar as restrições de compatibilidade. [Leiamos do Cisco Finesse 12.6 ES03, CUIC / IdS 12.6 ES03 Readme](#)

---

## Pré-requisitos

### Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Versão do Cisco Unified Contact Center Enterprise (UCCE)
- Cisco Finesse
- administração do Linux
- Administração de rede e administração de rede Linux

### Componentes Utilizados

As informações neste documento são baseadas nestas versões de software e hardware:

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

Observação: as implantações de PCCE/UCCE 2k precisarão estar na versão 12.6 do CCE

devido à implantação co-residente de LD/CUIC

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

---

 Observação: a configuração fornecida neste documento foi configurada, reforçada e testada com carga com proxy reverso Nginx (OpenResty) implantado no CentOS 8.0, em comparação com uma implantação UCCE de usuário de exemplo 2000. As informações de referência do perfil de desempenho estão disponíveis neste documento.

---

## Informações de Apoio

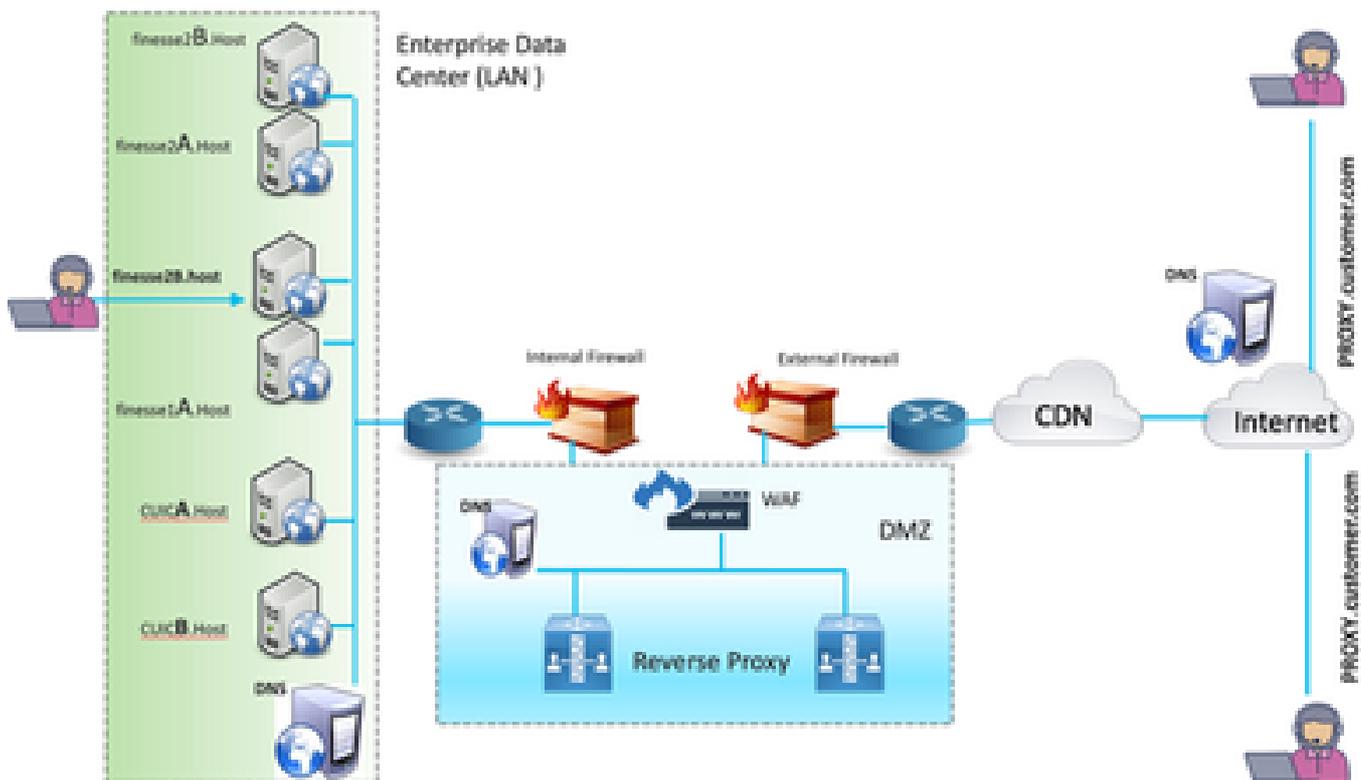
Esse modelo de implantação é compatível com as soluções UCCE/PCCE e HCS para UCCE.

A implantação de um proxy reverso é suportada (disponível a partir da 12.6 ES01) como uma opção para acessar o desktop Cisco Finesse sem se conectar a uma VPN. Esse recurso oferece a flexibilidade para que os agentes acessem o desktop Finesse de qualquer lugar pela Internet.

Para habilitar esse recurso, um par de proxy reverso deve ser implantado na zona desmilitarizada (DMZ).

O acesso à mídia permanece inalterado nas implantações de proxy reverso. Para se conectar à mídia, os agentes podem usar a solução Cisco Jabber over Mobile and Remote Access (MRA) ou o recurso do agente móvel do UCCE com uma rede telefônica pública comutada (PSTN) ou endpoint móvel. Este diagrama mostra como será a implantação de rede quando você acessar dois clusters Finesse e dois nós CUIC por meio de um único par de nós proxy reversos de alta disponibilidade (HA).

O acesso simultâneo de agentes na Internet e agentes que se conectam pela LAN é suportado como mostrado nesta imagem.



✎ Observação: consulte o guia de recursos para obter os critérios de seleção de proxy de terceiros no lugar do Nginx para oferecer suporte a essa implantação.

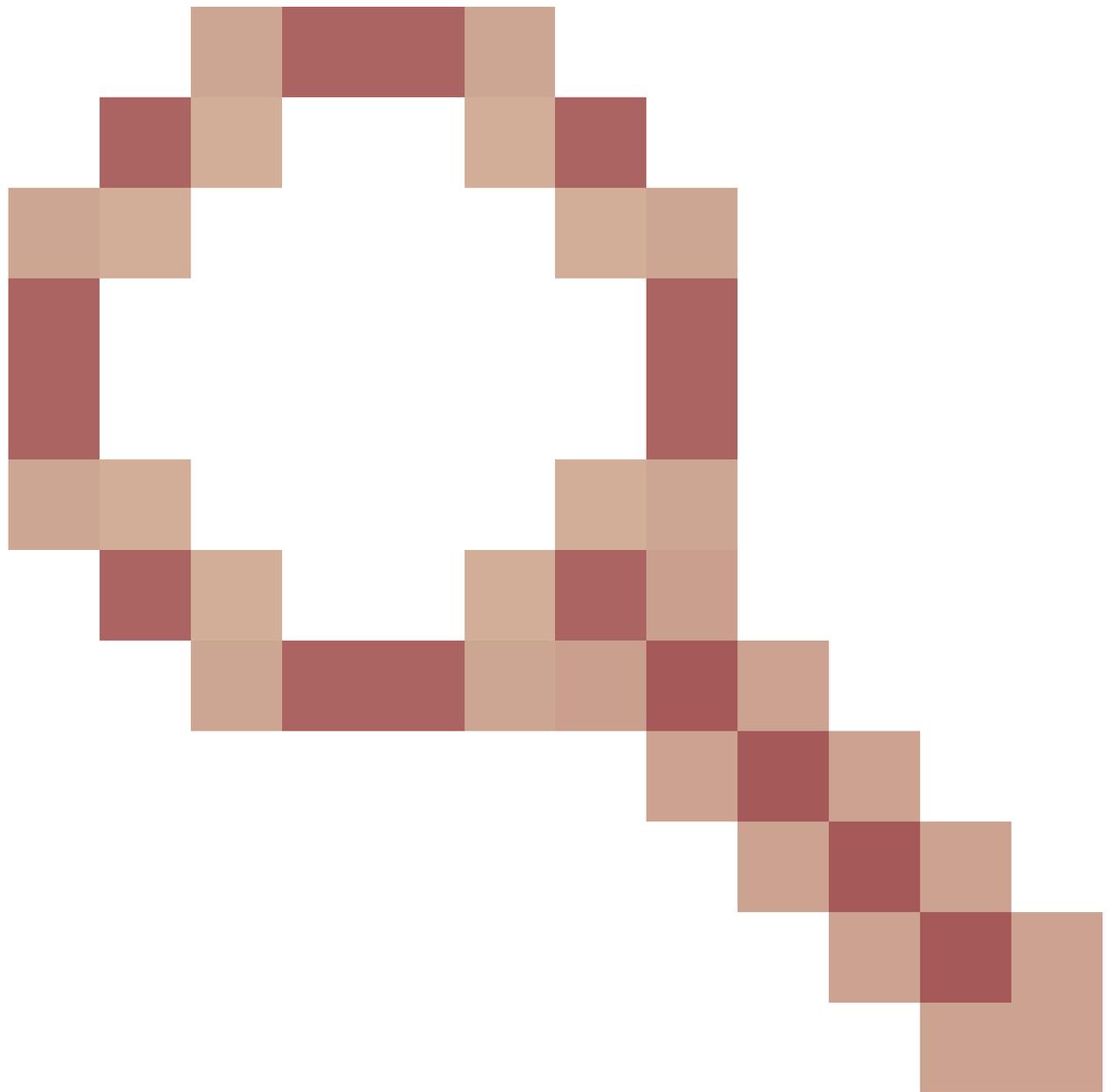
- [Guia de recursos do UCCE 12.6](#) - Fornece uma visão geral de recursos, design, bem como [detalhes de configuração](#) para o recurso sem VPN.
- [Guia de segurança UCCE 12.6](#) - Fornece diretrizes de configuração de segurança para a implantação de proxy reverso.

É recomendável rever a seção Sem VPN do guia de recursos e do guia de segurança antes de ler este documento.

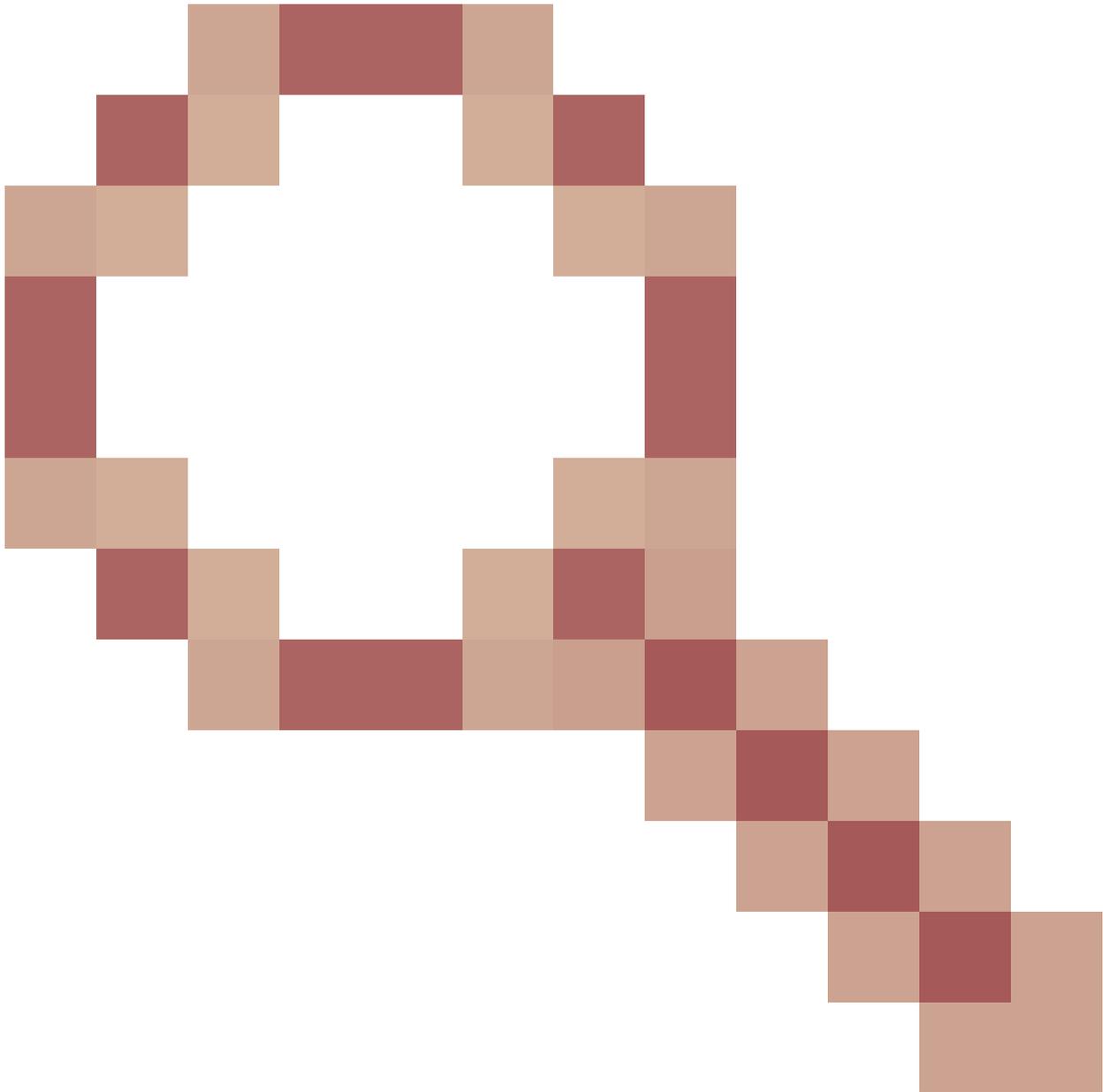
## Alterações no ES03

- Novos recursos
  - Os recursos de supervisor Finesse agora são suportados através de proxy reverso.
  - Agora, os relatórios CUIC RealTime e Historical são suportados por meio de gadgets do Finesse em um ambiente com proxy.
  - Autenticação para todas as solicitações/comunicações - requer suporte a Lua
    - Todas as solicitações Finesse / CUIC / IM & Presence (IM&P) são autenticadas no proxy antes de ter permissão para entrar no data center.
    - As conexões de E/S de soquete de dados ao vivo e WebSocket também são restritas e permitidas apenas de clientes que fizeram com êxito uma solicitação segura ao Finesse.

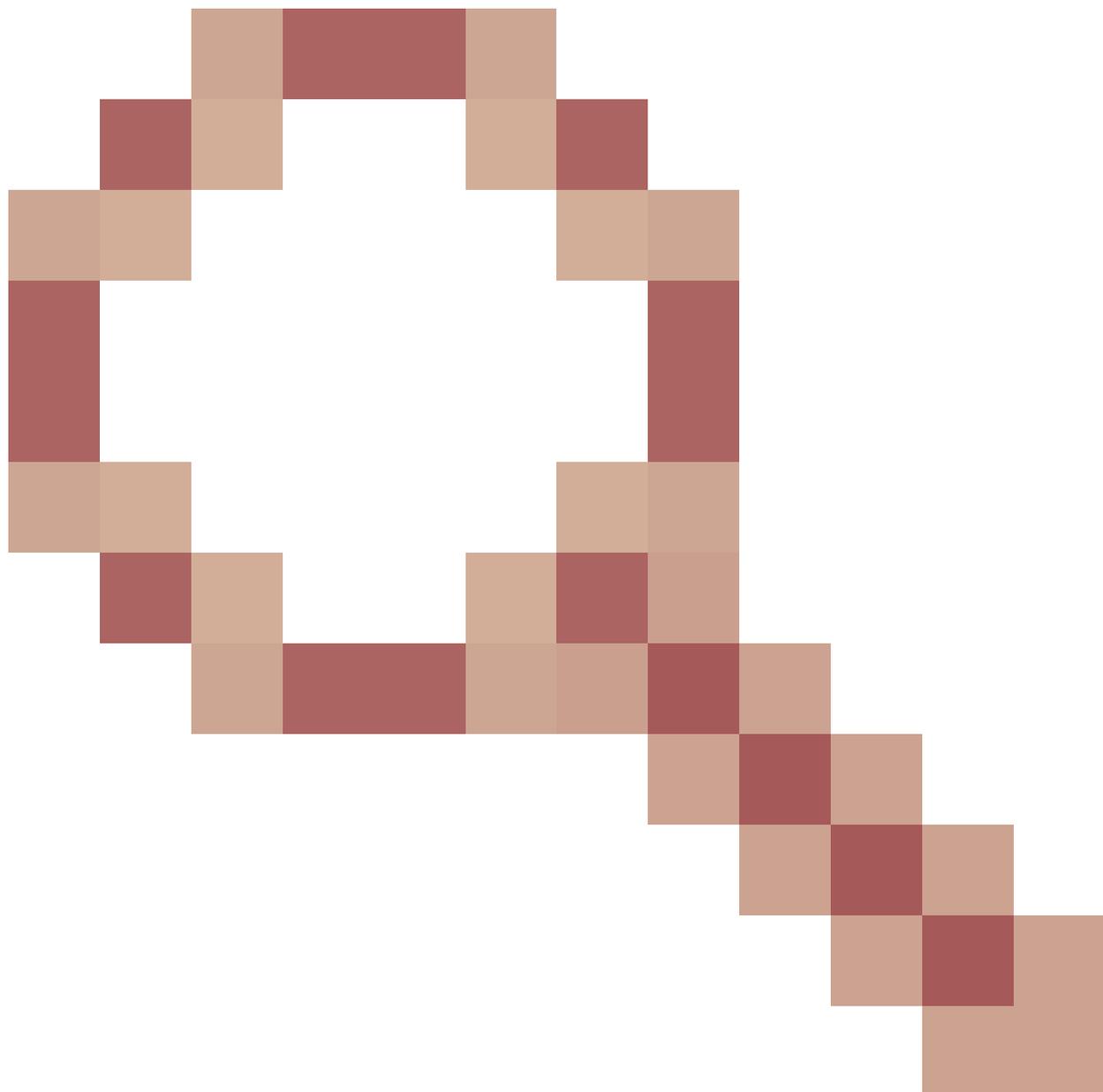
- Detecção de ataque de força bruta e registro no proxy, que pode ser usado com Fail2Ban para bloquear endereços IP mal-intencionados.
- Aprimoramentos de segurança para configuração de proxy reverso - requer suporte a Lua
  - Autenticação TLS (Transport Layer Security) mútua entre o proxy reverso e os componentes upstream (Finesse/IdS/CUIC/Livedata).
  - Configurações do SeLinux.
  - Habilite a verificação mútua de confiança SSL (Secure Sockets Layer) para solicitações de proxy e servidor de componentes.
- Segurança avançada para a configuração de proxy para evitar ataques de negação de serviço (DoS) / negação de serviço distribuído (DDoS) - requer suporte a Lua
  - Limites de taxa de solicitação Nginx aprimorados para várias partes do sistema.
  - Limites de taxa para IpTables.
  - Verificação das solicitações de recursos estáticos antes de solicitar o servidor de componentes upstream.
  - Páginas não autenticadas mais claras e armazenáveis em cache que não atingem o servidor de componentes upstream.
- Outros recursos diversos - requer suporte a Lua
  - Resposta de Compartilhamento de Recursos entre Origens (CORS) com detecção automática fornecida pelo proxy para auxiliar na configuração automática e melhorar o desempenho
- Correções de defeitos relacionadas a VPN-less
  - [CSCwa26057](#)



[CSCwa26057](#)

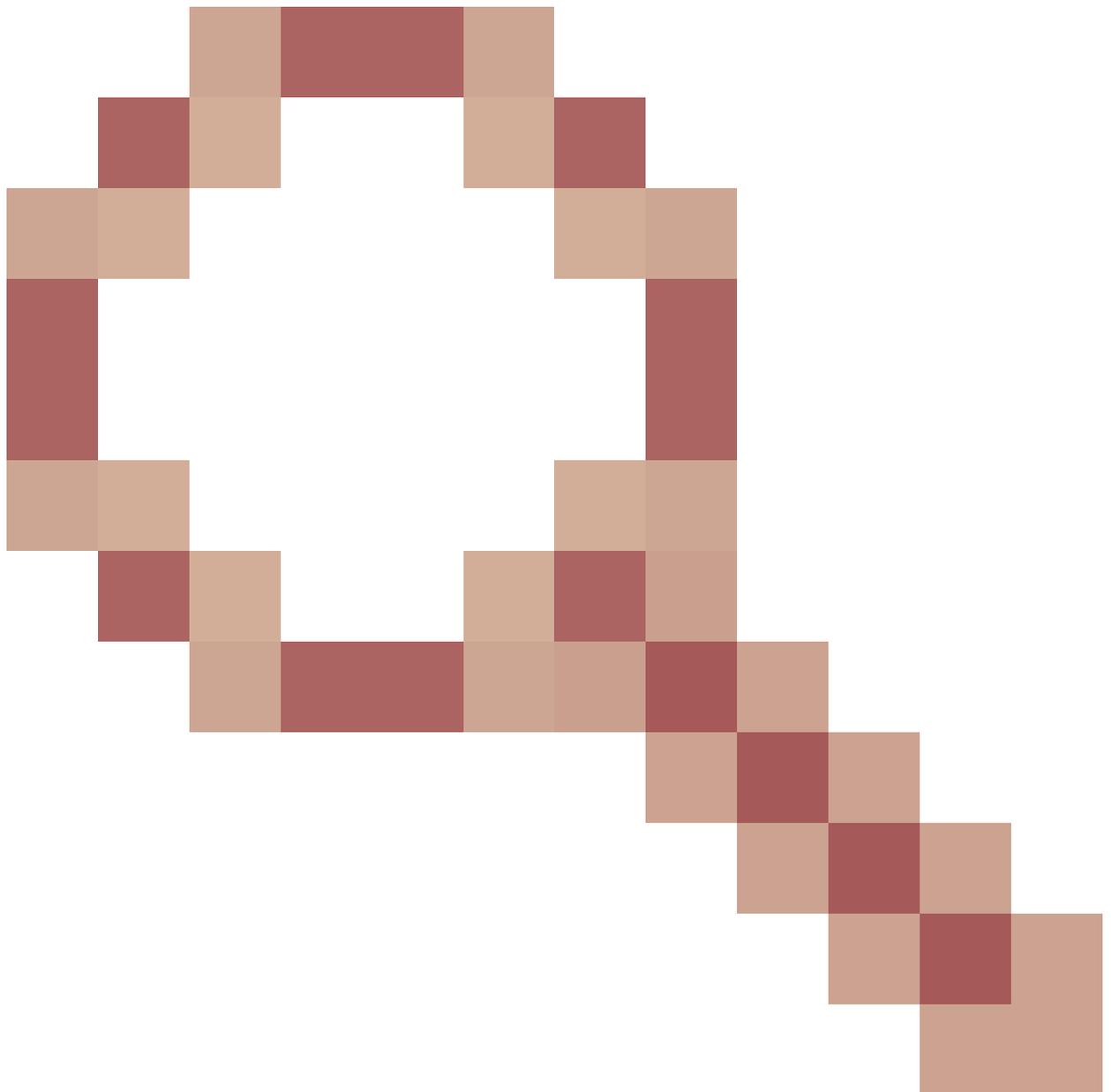


" />- Vários certificados oferecidos ao agente durante o login refinado no desktop  
◦ [CSCwa24471](#)



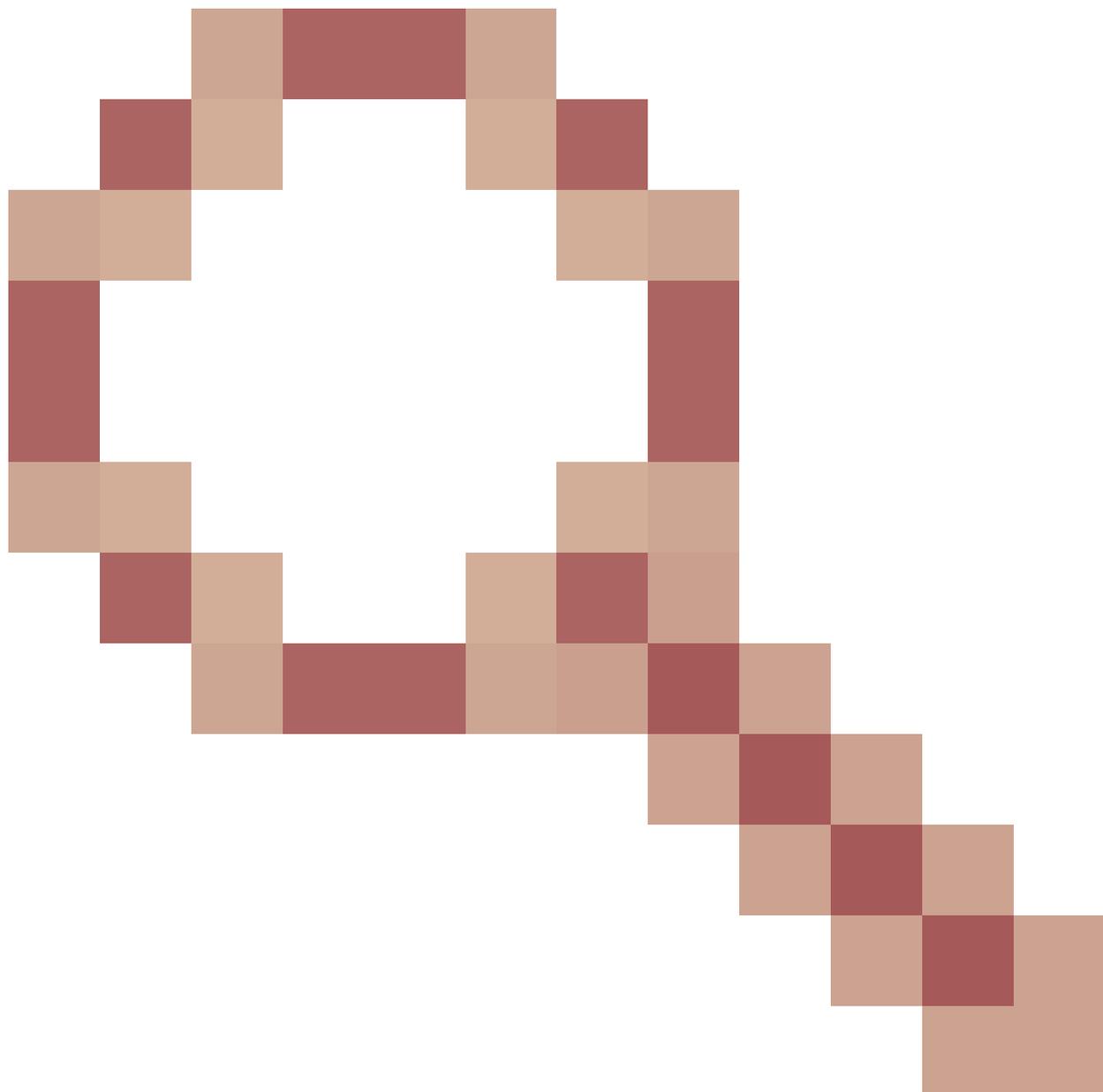
- A página de logon do Finesse não mostra o nome FQDN do Agente SSO

- [CSCwa24519](#)



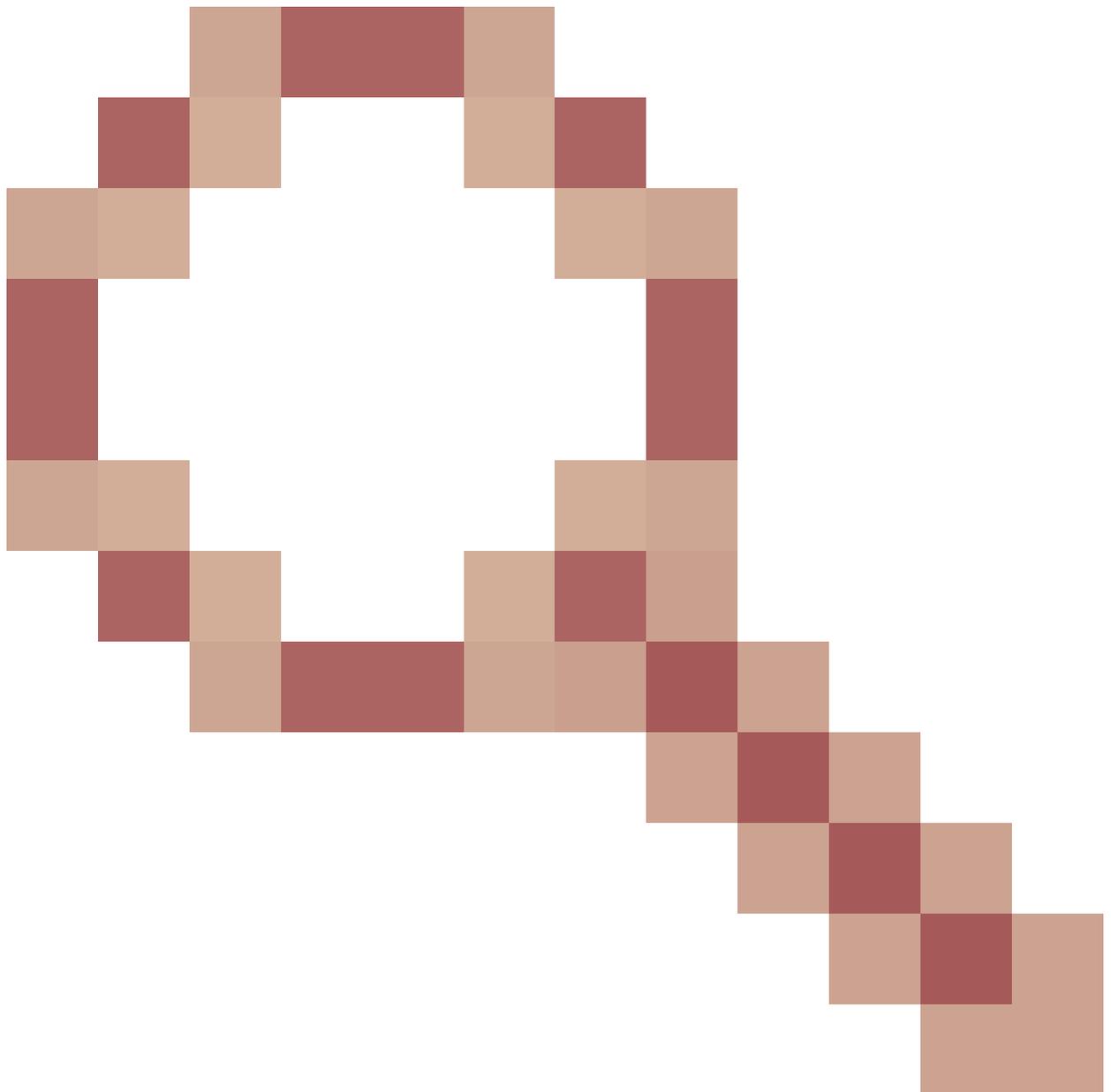
: O serviço Webproxy falhará ao reiniciar se o nome de host do proxy reverso não puder ser resolvido a partir do componente

- [CSCwa23252](#)



: A confiança de finesse do proxy é quebrada quando a profundidade é mais de um para a cadeia de certificados da autoridade de certificação

- [CSCwa46459](#)



vulnerabilidade de dia zero do log4j exposta no serviço da web

## Notas de atualização para configurações sem VPN baseadas em ES01

- A configuração ES03 requer a instalação do Nginx com suporte a Lua.
- Requisitos do certificado
  - O Cisco Finesse, o CUIIC e o IdS exigirão que o certificado de host Nginx / OpenResty seja adicionado ao armazenamento confiável do Tomcat e uma reinicialização seja feita, antes que a configuração do Nginx ES02 possa se conectar com êxito ao servidor upstream.
  - Os certificados de servidor upstream do Cisco Finesse, CUIIC e IdS precisam ser configurados no servidor Nginx para usar a configuração baseada em ES03.

---

 Observação: é recomendável remover a configuração Nginx baseada em ES01 existente antes de instalar as configurações Nginx ES03.

---

---

 Observação: os scripts de configuração ES03 também exigem a instalação COP ES03 correspondente no Cisco Finesse, CUIC e IdS.

---

## Autenticação

O Finesse 12.6 ES03 introduz a autenticação no proxy. A autenticação é suportada para implantações de Logon Único (SSO) e não-SSO.

A autenticação é imposta para todas as solicitações e protocolos que são aceitos no proxy antes de serem encaminhados para os servidores de componentes upstream, onde a autenticação imposta pelos servidores de componentes localmente também ocorre. Toda a autenticação usa as credenciais de login comuns do Finesse para autenticar as solicitações.

As conexões persistentes, como os websockets, que dependem de protocolos de aplicação como Extensible Messaging and Presence Protocol (XMPP) para autenticação e pós-conexão, são autenticadas no proxy validando o endereço IP do qual uma autenticação de aplicação bem-sucedida foi feita antes de estabelecer a conexão do soquete.

### Autenticação não SSO

A autenticação sem SSO não requer configurações extras e funcionará com scripts de configuração do Nginx prontos para uso assim que as substituições de script necessárias forem feitas. A autenticação se baseia no nome de usuário e senha usados para fazer login no Finesse. O acesso a todos os endpoints será validado com os serviços de autenticação Finesse.

A lista de usuários válidos é armazenada em cache localmente no proxy (atualiza o cache a cada 15 minutos), que é usado para validar o usuário em uma solicitação. As credenciais do usuário são validadas pelo encaminhamento da solicitação ao URI do Finesse configurado e, posteriormente, o hash da credencial é armazenado em cache localmente (15 minutos em cache) para autenticar novas solicitações localmente. Se houver qualquer alteração no nome de usuário ou na senha, ela entrará em vigor somente após 15 minutos.

### Autenticação SSO

A autenticação SSO requer que o administrador configure a chave de criptografia do token IdS no servidor Nginx dentro do arquivo de configuração. A chave de criptografia do token IdS pode ser obtida do servidor IdS com o comando CLI `show ids secret`. A chave deve ser configurada como parte de uma das `#Must-change` substituições que o administrador precisa executar nos scripts antes que a autenticação do SSO possa funcionar.

Consulte o guia do usuário do SSO para obter as configurações de IdS SAML a serem executadas para que a resolução de proxy funcione para IdS.

Uma vez configurada a autenticação SSO, um par válido de tokens pode ser usado para acessar qualquer um dos pontos finais no sistema. A configuração de proxy valida as credenciais interceptando as solicitações de recuperação de token feitas a IdS ou descriptografando tokens válidos e, em seguida, armazenando-os localmente em cache para outras validações.

## Autenticação para Conexões Websocket

As conexões de websocket não podem ser autenticadas com o cabeçalho de autorização padrão, pois os cabeçalhos personalizados não são suportados por implementações de websocket nativas no navegador. Protocolos de autenticação de nível de aplicação, em que as informações de autenticação contidas no payload não impedem o estabelecimento de conexões de websocket e, portanto, entidades mal-intencionadas podem processar ataques de DOS ou DDOS apenas criando inúmeras conexões para sobrecarregar o sistema.

Para reduzir essa possibilidade, as configurações de proxy reverso nginx fornecidas têm verificações específicas para permitir que conexões de websocket sejam aceitas SOMENTE a partir dos endereços IP que fizeram com sucesso uma solicitação REST autenticada antes do estabelecimento da conexão de websocket. Isso significa que os clientes que tentarem criar conexões de websocket, antes de uma solicitação REST ser emitida, agora receberão um erro de falha de autorização e não é um cenário de uso com suporte.

## Prevenção de ataque de força bruta

Os scripts de autenticação Finesse 12.6 ES02 evitam ativamente ataques de força bruta que podem ser usados para adivinhar a senha do usuário. Ele faz isso bloqueando o endereço IP usado para acessar o serviço, após um determinado número de tentativas falhas em um curto período de tempo. Essas solicitações serão rejeitadas pelo erro 418 do cliente. Os detalhes dos endereços IP bloqueados podem ser acessados nos arquivos <nginx-install-diretory>/logs/blocking.log e <nginx-install-diretory>/logs/error.log.

O número de solicitações com falha, o intervalo de tempo e a duração do bloqueio são configuráveis. As configurações estão presentes no arquivo <nginx-install-diretory>/conf/conf.d/maps.conf.

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

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

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

## Registro

Para localizar os endereços IP bloqueados, execute os seguintes comandos no diretório <nginx-install-diretorio>/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"
```

Recomenda-se que os clientes se integrem com Fail2ban ou similar para adicionar o banimento às regras de tabela IP/firewall.

## Instalar e Configurar Fail2ban

O Fail2ban examina arquivos de log e IPs proibidos que mostram sinais maliciosos - muitas falhas de senha, busca por explorações, etc. Geralmente, o Fail2Ban é usado para atualizar as regras de firewall para rejeitar os endereços IP por um período de tempo especificado, embora qualquer outra ação arbitrária (por exemplo, enviar um e-mail) também possa ser configurada. Para obter mais informações, visite <https://www.fail2ban.org/>.

Fail2ban pode ser configurado para monitorar o blocking.log para identificar os endereços IP que são bloqueados pelo Nginx na detecção de ataques bruteforce, e proibi-los por uma duração configurável. As etapas para instalar e configurar fail2ban em um proxy reverso CentOS são as seguintes:

1. Instale o Fail2ban usando o yum.

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

2. Crie uma prisão local.

As configurações de prisão permitem que o administrador configure várias propriedades, como as

portas que devem ser proibidas de serem acessadas por qualquer endereço IP bloqueado, a duração pela qual o endereço IP permanece bloqueado, a configuração de filtro usada para identificar o endereço IP bloqueado no arquivo de registro monitorado etc. As etapas para adicionar uma configuração personalizada para proibir endereços IP que são bloqueados para acessar os servidores upstream são as seguintes:

### 2.1. Vá para o diretório de instalação Fail2ban (neste exemplo /etc/fail2ban)

```
cd /etc/fail2ban
```

### 2.2. Faça uma cópia de jail.conf em jail.local para manter as mudanças locais isoladas.

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

### 2.3. Adicione essas configurações de cadeia ao final do arquivo chain.local e substitua as portas do modelo pelas reais. Atualize as configurações de tempo de proibição conforme necessário.

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

## 3. Configure um filtro.

Um filtro informa ao Fail2ban o que procurar nos logs para identificar o host a ser banido. As etapas para criar um filtro são as seguintes:

### 3.1. Crie filter.d/finesseban.conf.

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

### 3.2. Adicione essas linhas ao filtro de arquivo.d/finesseban.conf.

[Definition]

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

### 4. Inicie Fail2ban.

Execute este comando para iniciar fail2ban.

```
fail2ban-client start
```

Abra os arquivos de log fail2ban e verifique se não há erros. Por padrão, os logs de fail2ban vão para o arquivo /var/log/fail2ban.log.

## Validar URLs de Recursos Estáticos

Todos os endpoints válidos que podem ser acessados de forma não autenticada são rastreados ativamente nos scripts ES03.

As solicitações para esses caminhos não autenticados são ativamente rejeitadas, se um URI inválido for solicitado, sem enviar essas solicitações ao servidor upstream.

## Armazenando Cabeçalhos CORS em Cache

Quando a primeira solicitação de opções é bem-sucedida, os cabeçalhos de resposta access-control-allow-headers, access-control-allow-, access-control-allow-methods, access-control-expose-headers e access-control-allow-credentials são armazenados em cache no proxy por cinco minutos. Esses cabeçalhos são armazenados em cache para cada servidor upstream respectivo.

## Configurar

Este documento descreve a configuração do Nginx como o proxy reverso a ser usado para ativar o acesso sem VPN Finesse. O componente da solução UCCE, o proxy e as versões do sistema operacional usados para verificar as instruções fornecidas são fornecidos. As instruções relevantes devem ser adaptadas ao sistema operacional ou ao proxy de sua escolha.

- Versão do Nginx usada - OpenResty 1.19.9.1
- SO usado para configuração - CentOS 8.0

---

 Observação: a configuração Nginx descrita pode ser baixada da [página de download do](#)

---

## Configurar componentes da solução para VPN com menos acesso

Após a configuração do proxy, configure os componentes da solução (Finesse/ CUIC / IdS) para VPN Menos acesso com o nome do host e IP planejados do proxy/serviços usados para acessar a solução com estes comandos.

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

Os detalhes desses comandos podem ser encontrados no [Guia de recursos do UCCE 12.6](#) e devem ser consultados antes de você usar este documento.

## Instalar o OpenResty como um proxy reverso na DMZ

Esta seção detalha as etapas de instalação do proxy com base no OpenResty. O proxy reverso é normalmente configurado como um dispositivo dedicado na zona desmilitarizada (DMZ) da rede, como mostrado no diagrama de implantação mencionado anteriormente.

1. Instale o SO de sua escolha com a especificação de hardware necessária. Os ajustes de parâmetros de kernel e IPv4 podem variar dependendo do sistema operacional selecionado, e os usuários são aconselhados a verificar novamente esses aspectos se a versão do sistema operacional escolhida for diferente.
2. Configure duas interfaces de rede. Uma interface será necessária para acesso público dos clientes da Internet e outra para se comunicar com os servidores na rede interna.
3. Instale o [OpenResty](#).

Quaisquer sabores de Nginx podem ser usados para este fim, desde que sejam baseados em Nginx 1.19+ e suportem Lua:

- Nginx Plus
- Código aberto Nginx (código aberto Nginx precisará ser compilado junto com módulos Lua baseados em OpenResty para que seja usado)
- OpenResty
- Extras do GetPageSpeed

---

 **Observação:** a configuração fornecida foi testada com o OpenResty 1.19 e espera-se que funcione com outras distribuições com apenas pequenas atualizações, se houver.

---

## Instalação do OpenResty

1. Instale o OpenResty. Consulte [Pacotes OpenResty Linux](#). Como parte da instalação do

OpenResty, o Nginx será instalado nesse local e adicionará o caminho do OpenResty à variável PATH adicionando o arquivo ~/.bashrc.

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

## 2. Iniciar/parar Nginx.

- Para iniciar o Nginx, insira `openresty`.
- Para parar Nginx, insira `openresty -s stop`.

## Configurar Nginx

A configuração é explicada para uma instalação Nginx baseada em OpenResty. Os diretórios padrão do OpenResty são:

- <nginx-install-diretory> = /usr/local/openresty/nginx
  - <Diretório\_Instalação\_Abertura> = /usr/local/openresty
1. Faça o download e extraia o arquivo da [página de download do software Finesse Release 12.6\(1\)ES03](#) (12.6-ES03-reverse-proxy-config.zip) que contém a configuração de proxy reverso para Nginx.
  2. Copie nginx.conf, nginx/conf.d/, e nginx/html/ do diretório de configuração de proxy reverso extraído para <nginx-install-diretory>/conf, <nginx-install-diretory>/conf/conf.d/, e <nginx-install-diretory>/html/, respectivamente.
  3. Copie o diretório nginx/lua do diretório de configuração de proxy reverso extraído dentro do <nginx-install-diretory>.
  4. Copie o conteúdo de lualib para <Openresty-install-diretory>/lualib/resty.
  5. Configure a rotação de log nginx copiando o arquivo nginx/logrotate/saproxy para a pasta <nginx-install-diretory>/logrotate/. Modifique o conteúdo do arquivo para apontar para os diretórios de log corretos se os padrões do Nginx não forem usados.
  6. O Nginx deve ser executado com uma conta de serviço dedicada não privilegiada, que deve ser bloqueada e ter um shell inválido (ou conforme aplicável para o SO escolhido).
  7. Localize a string "Must-change" nos arquivos das pastas extraídas chamadas html e conf.d e substitua os valores indicados pelas entradas apropriadas.
  8. Certifique-se de que todas as substituições obrigatórias sejam feitas, que são descritas com os comentários Must-change nos arquivos de configuração.
  9. Certifique-se de que os diretórios de cache configurados para CUIIC e Finesse sejam criados em <nginx-install-diretory>/cache junto com esses diretórios temporários.
    - <nginx-install-diretory>/cache/client\_temp
    - <nginx-install-diretory>/cache/proxy\_temp

---

 Observação: a configuração fornecida destina-se a uma implantação de exemplo do 2000 e deve ser expandida adequadamente para uma implantação maior.

---

## Configurar o cache do Nginx

Por padrão, os caminhos de cache do proxy são armazenados no sistema de arquivos.

Recomendamos alterá-los para unidades na memória, criando um local de cache em tmpfs, como mostrado aqui.

1. Crie diretórios para os diferentes caminhos de cache de proxy em /home.

Por exemplo, esses diretórios devem ser criados para o Finesse primário. As mesmas etapas devem ser seguidas para os servidores secundários 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
```



Observação: aumente os caches client e proxy\_temp em 1 GB para cada novo cluster Finesse adicionado à configuração.

---

2. Monte os novos pontos de montagem com o comando `mount -av`.
3. Valide se o sistema de arquivos montou os novos pontos de montagem com o `df -h` comando.
4. Altere as localizações proxy\_cache\_path nos arquivos de configuração de cache Finesse e CUIC.

Por exemplo, para alterar os caminhos para o principal Finesse, vá para `<nginx-install-directory>conf/conf.d/finesse/caches` e altere o local do cache existente `/usr/local/openresty/nginx/cache/finesse25/` para o local do sistema de arquivos recém-criado `/home/primaryFinesse`.

```
##Must-change /usr/local/openresty/nginx/cache/finesse25 location would change depending on folder extraction proxy_cache_path
```

```
/home/primaryFinesse/desktop levels=1:2 use_temp_path=on keys_zone=desktop_cache_fin25:10m max_size=15m inactive=3y
use_temp_path=off; proxy_cache_path /home/primaryFinesse/shindig levels=1:2 use_temp_path=on
keys_zone=shindig_cache_fin25:10m max_size=500m inactive=3y use_temp_path=off; proxy_cache_path
/home/primaryFinesse/openfire levels=1:2 use_temp_path=on keys_zone=openfire_cache_fin25:10m max_size=10m inactive=3y
use_temp_path=off; proxy_cache_path /home/primaryFinesse/rest levels=1:2 use_temp_path=on keys_zone=rest_cache_fin25:10m
max_size=1500m inactive=40m use_temp_path=off;
```

## 5. Siga as mesmas etapas para os servidores Finesse secundário e CUIC.

 Observação: certifique-se de que a soma de todos os tamanhos de unidade tmpfs criados em todas as etapas anteriores seja adicionada ao tamanho de memória final para a implantação, uma vez que essas unidades são blocos de memória configurados para parecer discos para o aplicativo e consomem tanto espaço de memória.

## Configurar Certificados SSL

Usar certificados com assinatura automática - Testar implantações

Certificados autoassinados só devem ser usados até que o proxy reverso esteja pronto para ser implantado na produção. Em uma implantação de produção, use apenas um certificado assinado por uma CA (Autoridade de Certificação).

1. Gerar certificados Nginx para conteúdo de pasta SSL. Antes de gerar certificados, você precisa criar uma pasta chamada ssl em /usr/local/openresty/nginx. Você precisa gerar dois certificados com a ajuda desses comandos (um para <reverseproxy\_primary\_fqdn> e outro para <reverseproxy\_secondary\_fqdn>).
  - a. `sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /usr/local/openresty/nginx/ssl/nginx.key -out /usr/local/openresty/nginx/ssl/nginx.crt` (passe o nome do host como: <reverseproxy\_primary\_fqdn>)
  - b. `sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /usr/local/openresty/nginx/ssl/nginxnode2.key -out /usr/local/openresty/nginx/ssl/nginxnode2.crt` (passe o nome do host como:<reverseproxy\_secondary\_fqdn>)
  - c. Certifique-se de que o caminho do certificado seja /usr/local/openresty/nginx/ssl/nginx.crt e /usr/local/openresty/nginx/ssl/nginxnode2.crt, já que eles já estão configurados nos arquivos de configuração do Finesse Nginx.
2. Altere a permissão da chave privada 400 (r-----).
3. Configure o firewall/[iptables](#) no proxy reverso para permitir a comunicação do firewall para corresponder às portas nas quais o servidor Nginx foi configurado para escutar.
4. Adicione o endereço IP e o nome de host de Finesse, IdS e CUIC na entrada /etc/hosts no servidor proxy reverso.
5. Consulte o guia de recursos da solução para obter as configurações a serem executadas nos servidores de componentes para configurar o host Nginx como um proxy reverso.

 Observação: a configuração fornecida destina-se a uma implantação de exemplo do 2000 e



deve ser expandida adequadamente para uma implantação maior.

Usar Certificado Assinado pela CA - Implantações de Produção

Um certificado assinado por uma autoridade de certificação pode ser instalado no proxy reverso com estas etapas:

### 1. Gere a CSR (solicitação de assinatura de certificado).

Para gerar o CSR e a chave privada, `openssl req -new -newkey rsa:4096 -keyout nginx.key -out nginx.csr` insira depois de fazer login no proxy. Siga o prompt e forneça os detalhes. Isso gera o CSR (nginx.csr no exemplo) e a chave privada RSA (nginx.key no exemplo) de força de 4096 bits.

Por exemplo:

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

Anote a senha PEM, pois ela será usada para descriptografar a chave privada durante a implantação.

### 2. Obtenha o certificado assinado da CA.

Envie o CSR à autoridade de certificação e obtenha o certificado assinado.

Observação: se o certificado recebido da CA não for uma cadeia de certificados que contenha todos os respectivos certificados, compõe todos os certificados relevantes em um único arquivo de cadeia de certificados.

### 3. Implante o certificado e a chave.

Descriptografe a chave gerada anteriormente como parte da primeira etapa com `openssl rsa -in nginx.key -out nginx_decrypted.key` comando. Coloque o certificado assinado pela autoridade de certificação e a chave descriptografada dentro da pasta `/usr/local/openresty/nginx/ssl` na máquina proxy inversa. Atualize/adicione as configurações SSL relacionadas ao certificado nas configurações do Nginx no arquivo de configuração `/usr/local/openresty/nginx/conf/conf.d/ssl/ssl.conf`.

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

#### 4. Configure permissões para os certificados.

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

and para que o certificado tenha permissão somente leitura e seja restrito ao proprietário.

#### 5. Recarregue o Nginx.

Usar parâmetro Diffie-Hellman personalizado

Crie um parâmetro Diffie-Hellman personalizado com estes comandos:

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

Altere a configuração do servidor para usar os novos parâmetros no arquivo  
`/usr/local/openresty/nginx/conf/conf.d/ssl/ssl.conf`:

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

### Verifique se o Grampeamento OCSP está Habilitado - Verificação de Revogação de Certificado

Observação: para habilitar isso, o servidor deve usar um certificado assinado pela CA e deve ter acesso à CA que assinou o certificado.

Adicione/atualize essa configuração no diretório  
`file/usr/local/openresty/nginx/conf/conf.d/ssl/ssl.conf`:

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

### Configuração Do Nginx

O arquivo de configuração Nginx padrão (`/usr/local/openresty/nginx/conf/nginx.conf`) deve ser modificado para conter essas entradas para garantir a segurança e o desempenho. Este conteúdo deve ser usado para modificar o arquivo de configuração padrão que é criado pela instalação do Nginx.

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

```
# Process id file location
```

```

pid /usr/local/openresty/nginx/logs/nginx.pid;

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

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

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

#user root root;

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

events {
    multi_accept on;

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

http {

    include mime.types;

    default_type text/plain;

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

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

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

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

```

```

lua_shared_dict thirdpartygadgeturlcount 100k;
lua_shared_dict corsheadersstore 100k;

init_worker_by_lua_block {
    local UsersListManager = require('users_list_manager')
    local UnauthenticatedDesktopResourceManager = require("unauthenticated_desktopresources_manager")
    local UnauthenticatedResourceManager = require("unauthenticated_thirdpartyresources_manager")
    -- Must-change Replace saproxy.cisco.com with reverseproxy fqdn

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

include conf.d/*.conf;

sendfile          on;

tcp_nopush       on;

server_names_hash_bucket_size 512;

```

## Configurar porta de proxy reverso

Por padrão, a configuração do Nginx ouve as solicitações do Finesse na porta 8445. De cada vez, apenas uma porta pode ser habilitada de um proxy reverso para suportar solicitações Finesse, por exemplo, 8445. Se a porta 443 precisar de suporte, edite o arquivo <nginx-install-directory>conf/conf.d/finesse.conf para ativar a escuta no 443 e desativar a escuta no 8445.

## Configurar a autenticação TLS mútua entre o proxy reverso e os componentes upstream

A autenticação de certificado SSL do cliente para conexões de hosts de proxy reverso pode ser habilitada nos componentes de upstream do CCBU CUIK/Finesse/IdS/Livedata através da nova opção CLI do CVOS, que é

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

Por padrão, isso está desabilitado e deve ser explicitamente habilitado pelo administrador, executando a CLI em cada servidor upstream independentemente. Quando esta opção estiver habilitada, o Cisco Web proxy Service em execução no host upstream iniciará a autenticação de certificados de clientes no handshake TLS para conexões originárias de hosts de proxy reverso confiáveis adicionados como parte de utilitários CLI sistema proxy reverso hosts permitidos adicionar <host proxy>.

Abaixo está o bloco de configuração para o mesmo nos arquivos de configuração de proxy, 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;
```

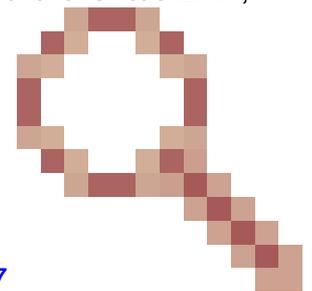
O certificado SSL usado para tráfego de saída (proxy para upstream) pode ser o mesmo certificado SSL configurado para tráfego de entrada (conector SSL para blocos de servidor de componentes). Se o certificado autoassinado for usado como `proxy_ssl_certificate`, ele terá que ser carregado para o repositório de confiança upstream (Finesse/IdS/CUIC/Livedata) para ser autenticado com êxito.

A validação do certificado do servidor upstream pelo proxy reverso é opcional e desabilitada por padrão. Se você deseja obter a autenticação mútua TLS completa entre o proxy reverso e os hosts de upstream, a configuração abaixo precisa ter os comentários removidos dos arquivos `ssl.conf` e `ssl2.conf`.

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

Avisos para configurar a autenticação TLS mútua:

- Quando esse recurso estiver habilitado nos componentes do CCBU, o certificado do cliente será solicitado aos clientes da LAN também durante o handshake TLS. Caso algum certificado pessoal/de cliente esteja instalado em navegadores de máquinas clientes, você pode optar por exibir um pop-up para o usuário final solicitando a escolha do certificado apropriado para a autenticação do cliente. Embora não importa qual certificado o usuário final escolhe ou pressione cancelar nas solicitações pop-up, as solicitações serão bem-sucedidas, pois a autenticação de certificado do cliente não é imposta para clientes LAN,



mas a mudança na experiência estará lá. Consulte CDET [CSCwa26057](#) para obter mais detalhes.

- O serviço de proxy da Web dos componentes upstream não será ativado se um host de proxy for adicionado à lista de permissões, que não pode ser resolvida pelo serviço de proxy da Web. Certifique-se de que os hosts de proxy reverso adicionados à lista de permitidos possam ser resolvidos a partir do componente upstream através de pesquisa DNS.

## Limpar cache

O cache de proxy reverso pode ser limpo com o

`/clearCache.sh`  
comando.

## Diretrizes padrão

Esta seção descreve brevemente as diretrizes padrão que precisam ser seguidas quando você configura o Nginx como um servidor proxy.

Essas diretrizes são derivadas do [Centro de Segurança da Internet](#). Para obter mais detalhes sobre cada diretriz, consulte a mesma.

1. É sempre recomendável usar a versão estável mais recente do OpenResty e do OpenSSL.
2. É aconselhável instalar o Nginx em uma montagem de disco separada.
3. A ID do processo Nginx deve ser de propriedade do usuário raiz (ou, conforme aplicável, do SO escolhido) e deve ter permissão 644 (rw-----) ou mais estrita.
4. O Nginx deve bloquear solicitações para hosts desconhecidos. Certifique-se de que cada bloco de servidor contenha a diretiva `server_name` explicitamente definida. Para verificar, pesquise todos os blocos de servidor no diretório `nginx.conf` e `nginx/conf.d` e verifique se todos os blocos de servidor contêm o `server_name`.
5. O Nginx deve escutar apenas nas portas autorizadas. Pesquise todos os blocos de servidor no diretório `nginx.conf` e `nginx/conf.d` e verifique as diretivas de escuta para verificar se apenas as portas autorizadas estão abertas para escuta.
6. Como o Cisco Finesse não suporta HTTP, é recomendável bloquear a porta HTTP do servidor proxy também.
7. O protocolo Nginx SSL deve ser TLS 1.2. O suporte a protocolos SSL herdados deve ser removido. Ele também deve desativar as cifras SSL fracas.
8. É aconselhável que os logs de erro e de acesso do Nginx sejam enviados ao servidor syslog remoto.
9. Recomenda-se instalar o módulo `mod_security` que funciona como um firewall de aplicação Web. Consulte o [manual ModSecurity](#) para obter mais informações. Observe que a carga Nginx não foi verificada no módulo `mod_security` no lugar.

## Configurar o Arquivo de Mapeamento

A implantação de proxy reverso da área de trabalho do Finesse requer um arquivo de mapeamento para configurar a lista de combinações de nome de host/porta visíveis externamente e seu mapeamento para os nomes de servidor e portas reais que são usados pelos servidores Finesse, IdS e CUIC. Esse arquivo de mapeamento, que é configurado em servidores internos, é a configuração-chave que permite que os clientes conectados pela Internet sejam redirecionados para os hosts e portas necessários que são usados na Internet.

O arquivo de mapeamento deve ser implantado em um servidor Web acessível aos servidores de componentes e seu URI precisa ser configurado para que a implantação funcione. Recomenda-se que o arquivo de mapeamento seja configurado usando um servidor Web dedicado disponível na rede. Se esse servidor não estiver disponível, o proxy reverso poderá ser usado, o que exigirá que o proxy esteja acessível de dentro da rede e também apresentará um risco de exposição das informações a clientes externos que podem fazer acesso não autorizado à DMZ. A próxima seção

detalha como isso pode ser feito.

Consulte o guia de recursos para obter as etapas exatas para configurar o URI do arquivo de mapeamento em todos os servidores de componentes e para obter mais detalhes sobre como criar os dados do arquivo de mapeamento.

## Usar Proxy Reverso como o Servidor de Arquivos de Mapeamento

Essas etapas são necessárias apenas se o proxy reverso também for usado como host do arquivo de mapeamento de proxy.

1. Configure o nome de host do proxy reverso no controlador de domínio usado pelos hosts Finesse/CUIC e IdS para que seu endereço IP possa ser resolvido.
2. Carregue os certificados assinados do Nginx gerados em ambos os nós sob tomcat-trust de cmplatform e reinicie o servidor.
3. Atualize os valores Must-change em <NGINX\_HOME>/html/proxymap.txt.
4. Recarregue as configurações do Nginx com o `nginx -s reload` comando.
5. Valide se o arquivo de configuração pode ser acessado de outro host da rede com o uso do `curl` comando.

## Endurecimento de kernel CentOS 8

Se o sistema operacional escolhido for o CentOS 8, recomenda-se que o endurecimento/ajuste do kernel seja feito com o uso dessas configurações `sysctl` para instalações que usam um servidor dedicado para hospedar o 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.udprmem_min = 4096
# net.core.wmem_default = 212992
# net.core.wmem_max = 212992
# net.ipv4.tcp_wmem = 4096 16384 4194304
# net.ipv4.udpwmem_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
```

Uma reinicialização é recomendada depois que você faz as alterações recomendadas.

## Protegendo IPtables

O IPtables é um aplicativo que permite ao administrador do sistema configurar as tabelas, cadeias e regras IPv4 e IPv6 fornecidas pelo firewall do kernel do Linux.

Essas regras de tabelas IP são configuradas para proteger o aplicativo proxy de ataques de força bruta, restringindo o acesso no firewall do kernel do Linux.

Os comentários na configuração indicam qual serviço está sendo limitado por taxa usando as regras.

---

 Observação: se os administradores usarem uma porta diferente ou expandirem o acesso a vários servidores usando as mesmas portas, o dimensionamento apropriado dessas portas deverá ser feito de acordo com esses números.

---

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

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

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

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

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

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

# Configuration for IMNP 15280 port
-A INPUT -p tcp -m tcp --dport 15280 --tcp-flags SYN SYN -m connlimit --connlimit-above 30 --connlimit-
-A INPUT -p tcp -m tcp --dport 15280 --tcp-flags SYN SYN -m connlimit --connlimit-above 30 --connlimit-
-A INPUT -p tcp -m tcp --dport 15280 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 20/sec --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-
-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 --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-max 6 -j LOG --log-prefix "connlimit-above 6"
-A INPUT -p tcp -m tcp --dport 8444 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 2/sec --hashlimit-mode srcip -j LOG --log-prefix "hashlimit-upto 2/sec"
-A INPUT -p tcp -m tcp --dport 8444 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG --log-prefix "limit 1/min"
-A INPUT -p tcp -m tcp --dport 8444 --tcp-flags SYN SYN -j DROP

# Configuration for CUIC 8447 port
-A INPUT -p tcp -m tcp --dport 8447 --tcp-flags SYN SYN -m connlimit --connlimit-above 6 --connlimit-max 6 -j LOG --log-prefix "connlimit-above 6"
-A INPUT -p tcp -m tcp --dport 8447 --tcp-flags SYN SYN -m connlimit --connlimit-above 6 --connlimit-max 6 -j LOG --log-prefix "connlimit-above 6"
-A INPUT -p tcp -m tcp --dport 8447 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 2/sec --hashlimit-mode srcip -j LOG --log-prefix "hashlimit-upto 2/sec"
-A INPUT -p tcp -m tcp --dport 8447 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG --log-prefix "limit 1/min"
-A INPUT -p tcp -m tcp --dport 8447 --tcp-flags SYN SYN -j DROP

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

# Configuration for LiveData 12008 port
-A INPUT -p tcp -m tcp --dport 12008 --tcp-flags SYN SYN -m connlimit --connlimit-above 10 --connlimit-max 10 -j LOG --log-prefix "connlimit-above 10"
-A INPUT -p tcp -m tcp --dport 12008 --tcp-flags SYN SYN -m connlimit --connlimit-above 10 --connlimit-max 10 -j LOG --log-prefix "connlimit-above 10"
-A INPUT -p tcp -m tcp --dport 12008 --tcp-flags SYN SYN -m hashlimit --hashlimit-upto 6/sec --hashlimit-mode srcip -j LOG --log-prefix "hashlimit-upto 6/sec"
-A INPUT -p tcp -m tcp --dport 12008 --tcp-flags SYN SYN -m limit --limit 1/min --limit-burst 1 -j LOG --log-prefix "limit 1/min"
-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

```

Essas regras podem ser aplicadas diretamente editando o `/etc/sysconfig/iptables` manualmente ou, como alternativa, salve a configuração em um arquivo como `iptables.conf` e execute `cat iptables.conf >>/etc/sysconfig/iptables` para aplicar as regras.

É necessário reiniciar o serviço IPtables após a aplicação das regras. Insira `systemctl restart iptables` para reiniciar o serviço IPtables.

## Restringir Conexões de Cliente

Além da configuração anterior das tabelas IP, as instalações que conhecem o intervalo de endereços para clientes que usam o proxy são recomendadas para usar esse conhecimento para proteger as regras de acesso ao proxy. Isso pode fornecer grandes retornos quando se trata de proteger o proxy de botnets de redes maliciosas que são muitas vezes criadas no intervalo de endereços IP de países que têm regras mais frouxas com relação à segurança on-line. É altamente recomendável, portanto, restringir os intervalos de endereços IP a país/estado ou intervalos de IP baseados em ISP se você tiver certeza dos padrões de acesso.

## Bloquear conexões de cliente

Também é útil saber como bloquear um intervalo específico de endereços quando um ataque é identificado como sendo feito a partir de um endereço IP ou de um intervalo de endereços IP.

Nesses casos, as solicitações desses endereços IP podem ser bloqueadas com regras iptable.

### Bloquear endereços IP distintos

Para bloquear vários endereços IP distintos, adicione uma linha ao arquivo de configuração IPTables para cada endereço IP.

Por exemplo, para bloquear os endereços 192.0.2.3 e 192.0.2.4, insira:

```
<#root>
```

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

### Bloquear um intervalo de endereços IP

Bloqueie vários endereços IP em um intervalo e adicione uma única linha ao arquivo de configuração IPTables com o intervalo IP.

Por exemplo, para bloquear endereços de 192.0.2.3 a 192.0.2.35, insira:

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

### Bloquear todos os endereços IP em uma sub-rede

Bloqueie todos os endereços IP em uma sub-rede inteira adicionando uma única linha ao arquivo de configuração IPTables com o uso da notação de roteamento entre domínios classless para o intervalo de endereços IP. Por exemplo, para bloquear todos os endereços de classe C, insira:

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

## SELinux

O SELinux é uma estrutura de segurança de plataforma integrada como um aprimoramento no sistema operacional Linux. O procedimento para instalar e adicionar políticas SELinux para executar OpenResty como o proxy reverso é fornecido em seguida.

1. Pare o processo com o `openresty -s stop` comando.
2. Configure e inicie o servidor `/stop nginx` com o `systemctl` comando para que durante a inicialização o processo OpenResty inicie automaticamente. Digite esses comandos como usuário `root`.
  - a. Vá para `/usr/lib/systemd/system`.
  - b. Abra um arquivo chamado `openresty.service`.
  - c. Atualize o conteúdo do arquivo de acordo com a localização do PIDFile.

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

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

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

- d. Como usuário `root`, insira `sudo systemctl enable openresty`.
- e. Inicie/interrompa o serviço OpenResty com o `systemctl start openresty / systemctl stop openresty` comando e verifique se o processo é iniciado/interrompido como usuário `raiz`.

## 1. Instalar o SELinux

- Por padrão, somente alguns pacotes SELinux serão instalados no CentOS.
- O pacote `policycoreutils-devel` e suas dependências devem ser instalados para gerar a política SELinux.
- Insira este comando para instalar `policycoreutils-devel`

```
yum install policycoreutils-devel
```

- Certifique-se de que, depois de instalar o pacote, o `sepolicy` comando funcione.

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

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

```
SELinux Policy Inspection Tool
```

## 2. Crie um novo usuário do Linux e mapeie com o usuário do SELinux

- a. Digite `semanage login -l` para ver o mapeamento entre usuários do Linux e usuários do SELinux.

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

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

- b. Como root, crie um novo usuário do Linux (nginx user) que seja mapeado para o usuário `user_u` do 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. Para visualizar o mapeamento entre `nginxuser` e `user_u`, insira este comando como root:

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

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

- d. O login do SELinux `__default__` por padrão é mapeado para o usuário do SELinux `unconfined_u`. É necessário fazer com que `user_u` seja confinado por padrão com este comando:

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

Para verificar se o comando funcionou corretamente, insira `semanage login -l`. Ele deve produzir esta saída:

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

e. Modifique nginx.conf e execute a alteração de propriedade para nginxuser.

- i. Digite `chown -R nginxuser:nginxuser *` no diretório <Openresty-install-diretory>.
- ii. Modifique o arquivo nginx.conf para incluir nginxuser como o usuário para executar processos de trabalho.

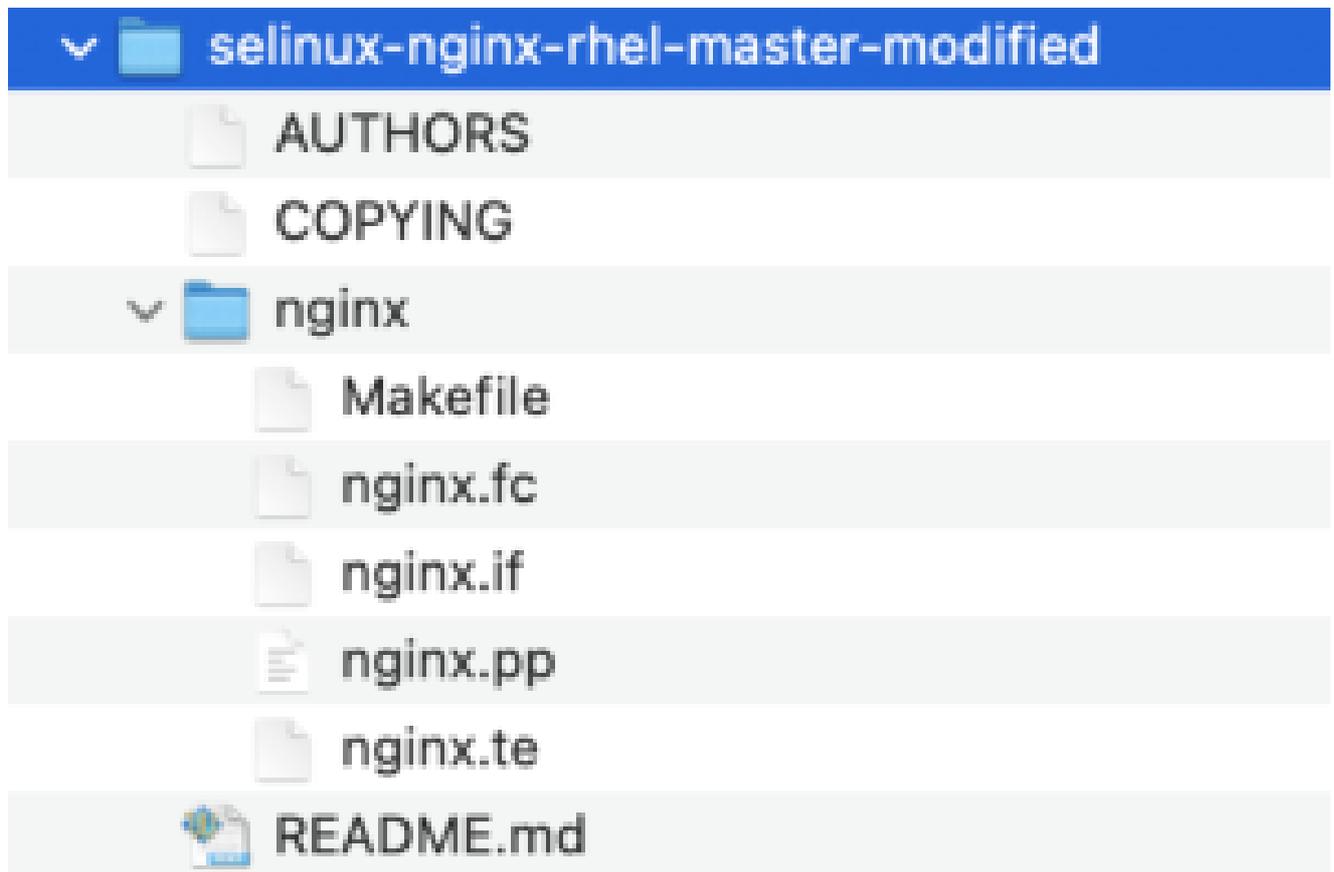
```

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

```

Escreva a política do SELinux para Nginx

1. Em vez de gerar uma nova política personalizada padrão para o Nginx com o `sepolicy generate --init /usr/bin/nginx` comando, é preferível começar com uma política existente.
2. O download dos arquivos nginx.fc (arquivo de contextos de arquivo) e nginx.te (arquivo de imposição de tipo) do URL fornecido foi modificado para se ajustar ao uso de proxy reverso.
3. Essa versão modificada pode ser usada como referência, já que foi corrigida para o caso de uso específico.
4. Descarregue o arquivo `selinux-nginx-rhel-master-modified.tar` da [página de download de software de arquivo](#).



5. Extraia o arquivo .tar e navegue até o diretório nginx dentro dele.
6. Abra o arquivo .fc e verifique os caminhos de arquivo necessários do instalador do nginx, cache e arquivo pid.
7. Compile a configuração com o `make` comando.
8. O arquivo `nginx.pp` será gerado.
9. Carregue a política com o `semodule` comando.

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

10. Vá para `/root` e crie um arquivo vazio chamado `touch /.autorelabel`.
11. Reinicialize o sistema.
12. Insira este comando para verificar se a política foi carregada com êxito.

```
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 als                  pp
```

13. O Nginx deve ser executado sem nenhuma violação. (As violações estarão disponíveis em /var/log/messages e /var/log/audit/audit.log).
14. Insira este comando para verificar o status do 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. Agora o desktop do agente/supervisor Finesse deve estar acessível.

## Verificar

Use esta seção para confirmar se a sua configuração funciona corretamente.

### Finesse

1. Solicite <https://<reverseproxy:port>/finesse/api/SystemInfo>. da DMZ e verifique se eles estão acessíveis.
2. Verifique os valores de <host> em <primaryNode> e <secondaryNode> são nomes de host de proxy reverso válidos. Não devem ser nomes de host Finesse.

### CUIC e Live Data

1. Se os nomes de host Finesse forem vistos na resposta em vez de nomes de host de proxy reverso, valide as configurações de mapeamento de proxy e os hosts permitidos são adicionados corretamente nos servidores Finesse, conforme descrito na seção "Preencha os dados de conversão de rede" do "Acesso sem VPN ao Finesse Desktop" no [Guia de](#)

## [recursos UCCE do Finesse 12.6.](#)

2. Se os gadgets LiveData forem carregados corretamente no Finesse Desktop, as configurações de proxy do CUIC e LiveData serão apropriadas.
3. Para validar a configuração do CUIC e do LiveData, faça solicitações HTTP a esses URLs da DMZ e veja se eles estão acessíveis.
  - [https://<reverseproxy:cuic\\_port>/cuic/rest/about](https://<reverseproxy:cuic_port>/cuic/rest/about)
  - [https://<reverseproxy:ldweb\\_port>/livedata/security](https://<reverseproxy:ldweb_port>/livedata/security)
  - [https://<reverseproxy:ldsocketio\\_port>/security](https://<reverseproxy:ldsocketio_port>/security)

## IDS

Para validar a configuração de IdS, execute estas etapas:

1. Faça login na interface IdSAdmin em [https://<ids\\_LAN\\_host:ids\\_port>:8553/idsadmin](https://<ids_LAN_host:ids_port>:8553/idsadmin) da LAN, pois a interface admin não é exposta pelo proxy reverso.
2. Escolha Settings > IdS Trust.
3. Valide se o nó do editor do cluster de proxy está listado na página Baixar metadados da controladora de armazenamento e clique em Avançar.
4. Valide se o proxy IDP é exibido corretamente se configurado na página Carregar metadados IDP e clique em Avançar.
5. Inicie o SSO de teste por meio de todos os nós de cluster de proxy na página Testar SSO e valide se todos foram bem-sucedidos. Isso requer conectividade com a máquina cliente para nós de proxy reverso.

## Desempenho

A análise de dados da captura de desempenho superior equivalente, feita com a ferramenta nmon, está disponível na [página de download do software Finesse Release 12.6\(1\) ES03](#) (load\_result.zip). Os dados representam o estado do proxy para operações de desktop e supervisor, em um exemplo de implantação de 2000 UCCE usando logons SSO e relatórios CUIC LD, conforme configurado no layout padrão para 2000 usuários por um período de oito horas. Ele pode ser usado para derivar os requisitos de computação, disco e rede para uma instalação usando o Nginx em hardware comparável.

## Troubleshooting

### SSO

1. O desktop redireciona não indo através de proxy
  1. Verifique se os nomes de host estão configurados em casos corretos de acordo com os nomes de host reais da vm em várias configurações, como proxymap.txt, arquivo server\_filter etc.
  2. Certifique-se de que o IdS seja adicionado com o nome de host em cascata correto no inventário do CCE, pois as mesmas informações são enviadas aos componentes quando registrados para o SSO do administrador da Web do CCE.

## 2. Logons SSO não estão ocorrendo

1. Verifique se a confiança de IdS-IDP foi estabelecida para o host proxy.

## SELinux

1. Se o Nginx não for iniciado por padrão ou o desktop do agente Finesse não estiver acessível, configure o SELinux para o modo permissivo com este comando:

```
setenforce 0
```

2. Tente reiniciar o Nginx com o `systemctl restart nginx` comando.
3. As violações estarão disponíveis em `/var/log/messages` e `/var/log/audit/audit.log`.
4. É necessário gerar novamente o arquivo `.te` com regras de permissão para lidar com essas violações por qualquer um destes comandos:

```
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. Atualize o arquivo `nginx.te` original presente no diretório `selinux-nginx-rhel-master-modified/nginx` com as regras de permissão recém-geradas.
6. Faça o mesmo com o `make` comando.
7. O arquivo `nginx.pp` será gerado novamente.
8. Carregue a política pelo comando `semodule`.

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

9. Faça com que o SELinux imponha o modo com este comando:

```
setenforce
```

10. Reinicialize o sistema.

11. Repita este procedimento até que as violações obrigatórias sejam corrigidas.

## Sobre esta tradução

A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês ([link fornecido](#)) seja sempre consultado.