Personalizar a configuração de criptografia SSL do Expressway

Contents

Introdução

Este documento descreve as etapas para personalizar as strings de cifra pré-configuradas no Expressway.

Pré-requisitos

Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Cisco Expressway ou Cisco VCS.
- Protocolo TLS.

Componentes Utilizados

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

- Cisco Expressway versão X15.0.2.

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.

Informações de Apoio

A configuração padrão do Expressway inclui cadeias de cifras pré-configuradas, que, por razões de compatibilidade, permitem o suporte para algumas cifras que podem ser consideradas fracas sob algumas políticas de segurança da empresa. É possível personalizar as strings de cifra para ajustá-las de acordo com as políticas específicas de cada ambiente.

No Expressway, é possível configurar uma string de cifra independente para cada um destes protocolos:

- HTTPS
- LDAP
- Proxy reverso
- SIP
- SMTP
- provisionamento de TMS
- Descoberta de servidor UC
- XMP

As strings de cifra obedecem ao formato OpenSSL descrito na página de manual cifras OpenSSL. A versão X15.0.2 atual do Expressway é fornecida com a cadeia de caracteres padrão EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!eNULL:!aNULL:!aDH pré-configurada para todos os protocolos igualmente. Na página de administração da Web, em Manutenção > Segurança > Cifras, você pode modificar a string de cifra atribuída a cada protocolo, para adicionar ou remover cifras específicas ou grupos de cifras usando um algoritmo comum.

Inspecione a string de codificação

Usando o comando openssl ciphers -V "<cipher string>", você pode gerar uma lista com todas as cifras permitidas por uma determinada sequência, o que é útil para inspecionar visualmente as cifras. Este exemplo mostra a saída ao inspecionar a string de cifra padrão do Expressway:

<#root>

~ #

openssl ciphers -V "EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!eNULL:!aNULL:!aDH"

```
0x13,0x02 - TLS_AES_256_GCM_SHA384 TLSv1.3 Kx=any Au=any Enc=AESGCM(256) Mac=AEAD
0x13,0x03 - TLS_CHACHA20_POLY1305_SHA256 TLSv1.3 Kx=any Au=any Enc=CHACHA20/POLY1305(256) Mac=AEAD
0x13,0x01 - TLS_AES_128_GCM_SHA256 TLSv1.3 Kx=any Au=any Enc=AESGCM(128) Mac=AEAD
0xC0,0x2C - ECDHE-ECDSA-AES256-GCM-SHA384 TLSv1.2 Kx=ECDH Au=ECDSA Enc=AESGCM(256) Mac=AEAD
0xC0,0x30 - ECDHE-RSA-AES256-GCM-SHA384 TLSv1.2 Kx=ECDH Au=RSA Enc=AESGCM(256) Mac=AEAD
0xCC,0xA9 - ECDHE-ECDSA-CHACHA20-POLY1305 TLSv1.2 Kx=ECDH Au=ECDSA Enc=CHACHA20/POLY1305(256) Mac=AEAD
```

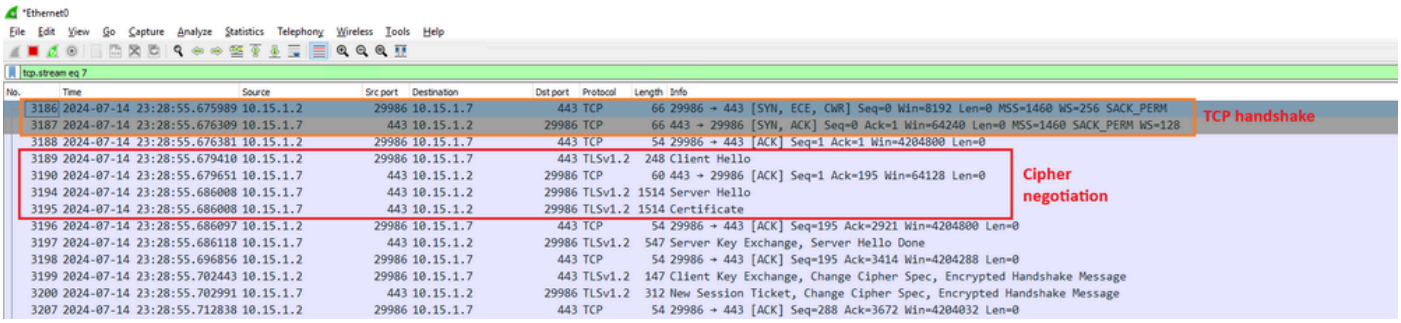
```
0xCC,0xA8 - ECDHE-RSA-CHACHA20-POLY1305 TLSv1.2 Kx=ECDH Au=RSA Enc=CHACHA20/POLY1305(256) Mac=AEAD
0xC0,0xAD - ECDHE-ECDSA-AES256-CCM TLSv1.2 Kx=ECDH Au=ECDSA Enc=AESCCM(256) Mac=AEAD
0xC0,0x2B - ECDHE-ECDSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH Au=ECDSA Enc=AESGCM(128) Mac=AEAD
0xC0,0x2F - ECDHE-RSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH Au=RSA Enc=AESGCM(128) Mac=AEAD
0xC0,0xAC - ECDHE-ECDSA-AES128-CCM TLSv1.2 Kx=ECDH Au=ECDSA Enc=AESCCM(128) Mac=AEAD
0xC0,0x24 - ECDHE-ECDSA-AES256-SHA384 TLSv1.2 Kx=ECDH Au=ECDSA Enc=AES(256) Mac=SHA384
0xC0,0x28 - ECDHE-RSA-AES256-SHA384 TLSv1.2 Kx=ECDH Au=RSA Enc=AES(256) Mac=SHA384
0xC0,0x23 - ECDHE-ECDSA-AES128-SHA256 TLSv1.2 Kx=ECDH Au=ECDSA Enc=AES(128) Mac=SHA256
0xC0,0x27 - ECDHE-RSA-AES128-SHA256 TLSv1.2 Kx=ECDH Au=RSA Enc=AES(128) Mac=SHA256
0xC0,0x09 - ECDHE-ECDSA-AES128-SHA TLSv1 Kx=ECDH Au=ECDSA Enc=AES(128) Mac=SHA1
0xC0,0x13 - ECDHE-RSA-AES128-SHA TLSv1 Kx=ECDH Au=RSA Enc=AES(128) Mac=SHA1
0x00,0xA3 - DHE-DSS-AES256-GCM-SHA384 TLSv1.2 Kx=DH Au=DSS Enc=AESGCM(256) Mac=AEAD
0x00,0x9F - DHE-RSA-AES256-GCM-SHA384 TLSv1.2 Kx=DH Au=RSA Enc=AESGCM(256) Mac=AEAD
0xCC,0xAA - DHE-RSA-CHACHA20-POLY1305 TLSv1.2 Kx=DH Au=RSA Enc=CHACHA20/POLY1305(256) Mac=AEAD
0xC0,0x9F - DHE-RSA-AES256-CCM TLSv1.2 Kx=DH Au=RSA Enc=AESCCM(256) Mac=AEAD
0x00,0xA2 - DHE-DSS-AES128-GCM-SHA256 TLSv1.2 Kx=DH Au=DSS Enc=AESGCM(128) Mac=AEAD
0x00,0x9E - DHE-RSA-AES128-GCM-SHA256 TLSv1.2 Kx=DH Au=RSA Enc=AESGCM(128) Mac=AEAD
0xC0,0x9E - DHE-RSA-AES128-CCM TLSv1.2 Kx=DH Au=RSA Enc=AESCCM(128) Mac=AEAD
0x00,0x6B - DHE-RSA-AES256-SHA256 TLSv1.2 Kx=DH Au=RSA Enc=AES(256) Mac=SHA256
0x00,0x6A - DHE-DSS-AES256-SHA256 TLSv1.2 Kx=DH Au=DSS Enc=AES(256) Mac=SHA256
0x00,0x67 - DHE-RSA-AES128-SHA256 TLSv1.2 Kx=DH Au=RSA Enc=AES(128) Mac=SHA256
0x00,0x40 - DHE-DSS-AES128-SHA256 TLSv1.2 Kx=DH Au=DSS Enc=AES(128) Mac=SHA256
0x00,0x33 - DHE-RSA-AES128-SHA SSLv3 Kx=DH Au=RSA Enc=AES(128) Mac=SHA1
0x00,0x32 - DHE-DSS-AES128-SHA SSLv3 Kx=DH Au=DSS Enc=AES(128) Mac=SHA1
0x00,0x9D - AES256-GCM-SHA384 TLSv1.2 Kx=RSA Au=RSA Enc=AESGCM(256) Mac=AEAD
0xC0,0x9D - AES256-CCM TLSv1.2 Kx=RSA Au=RSA Enc=AESCCM(256) Mac=AEAD
0x00,0x9C - AES128-GCM-SHA256 TLSv1.2 Kx=RSA Au=RSA Enc=AESGCM(128) Mac=AEAD
0xC0,0x9C - AES128-CCM TLSv1.2 Kx=RSA Au=RSA Enc=AESCCM(128) Mac=AEAD
0x00,0x3D - AES256-SHA256 TLSv1.2 Kx=RSA Au=RSA Enc=AES(256) Mac=SHA256
0x00,0x3C - AES128-SHA256 TLSv1.2 Kx=RSA Au=RSA Enc=AES(128) Mac=SHA256
0x00,0x2F - AES128-SHA SSLv3 Kx=RSA Au=RSA Enc=AES(128) Mac=SHA1
~ #
```

Inspecione a negociação de cifra no handshake TLS com uma captura de pacote

Ao capturar uma negociação TLS em uma captura de pacote, você pode inspecionar os detalhes da negociação de cifra usando o Wireshark.

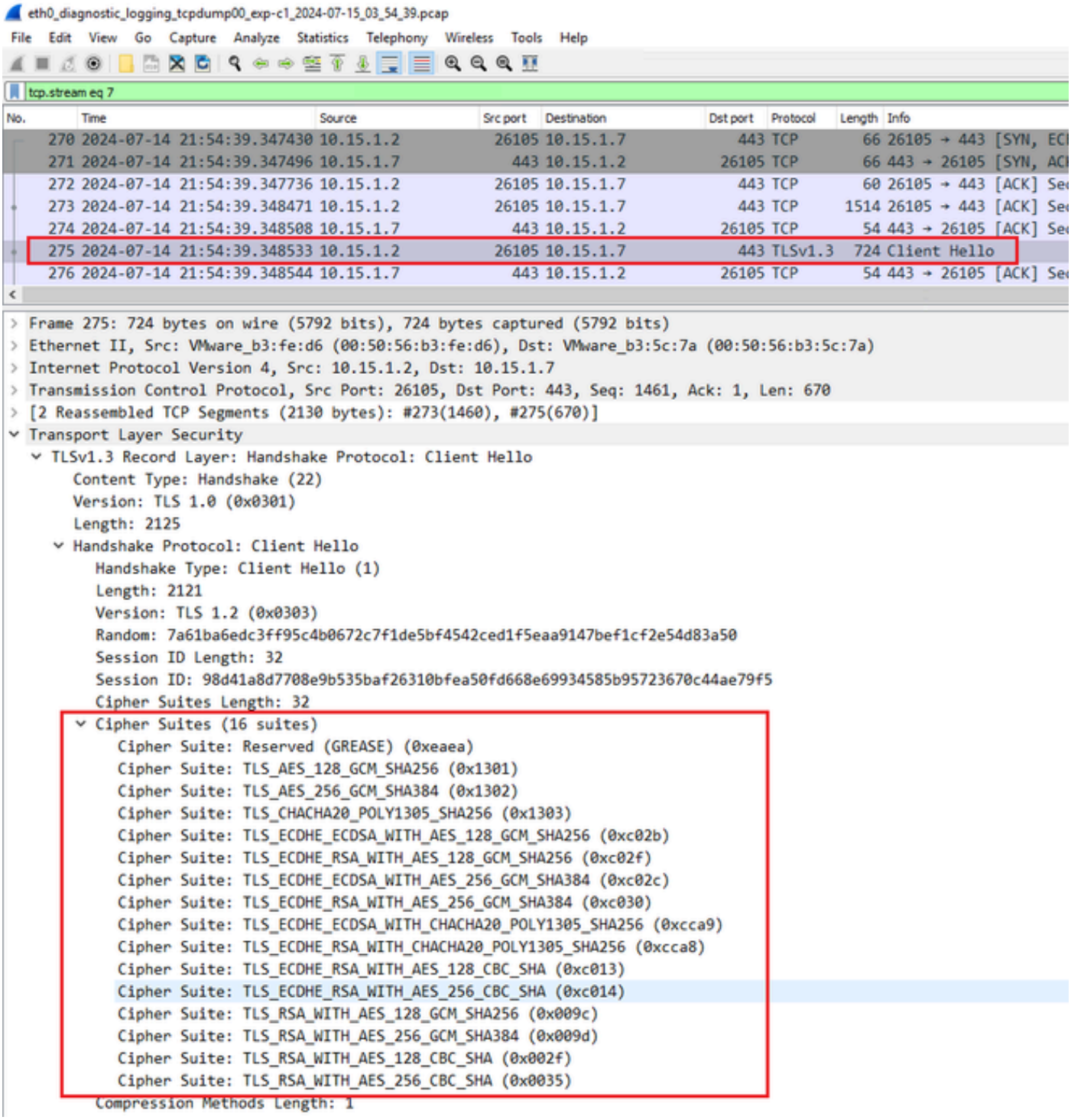
O processo de handshake TLS inclui um pacote ClientHello enviado pelo dispositivo cliente, fornecendo a lista das cifras que ele suporta de acordo com sua string de cifra configurada para o protocolo de conexão. O servidor analisa a lista, compara-a com sua própria lista de cifras permitidas (determinada por sua própria sequência de cifras) e escolhe uma cifra que ambos os sistemas suportam, para ser usada na sessão criptografada. Em seguida, ele responde com um pacote ServerHello indicando a cifra escolhida. Há diferenças importantes entre os diálogos de handshake TLS 1.2 e 1.3, no entanto, o mecanismo de negociação de cifra usa esse mesmo princípio em ambas as versões.

Este é um exemplo de uma negociação de cifra TLS 1.3 entre um navegador da Web e o Expressway na porta 443 como visto no Wireshark:



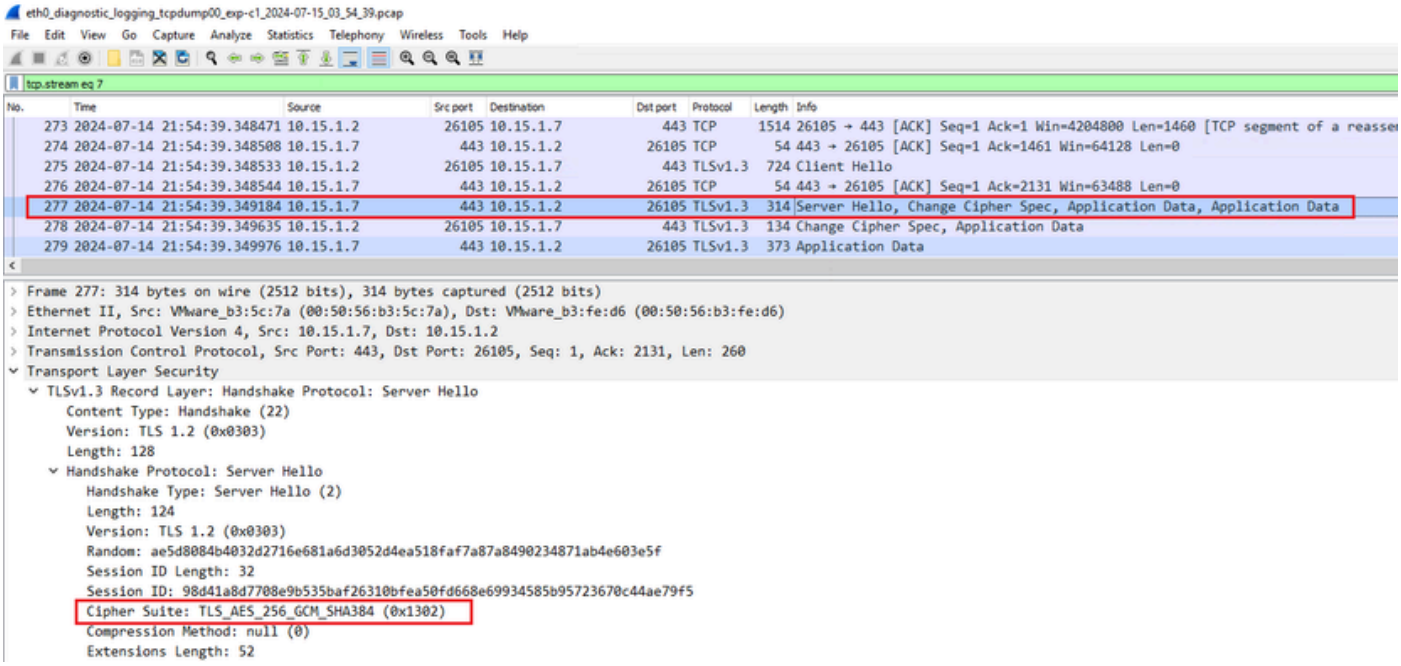
Exemplo de um handshake TLS no Wireshark

Primeiro, o navegador envia um pacote ClientHello com a lista de cifras suportadas:



Exemplo de um pacote ClientHello no Wireshark

O Expressway verifica sua sequência de cifra configurada para o protocolo HTTPS e encontra uma cifra que ela mesma e o cliente suportam. Neste exemplo, a cifra ECDHE-RSA-AES256-GCM-SHA384 é selecionada. O Expressway responde com seu pacote ServerHello indicando a cifra selecionada:



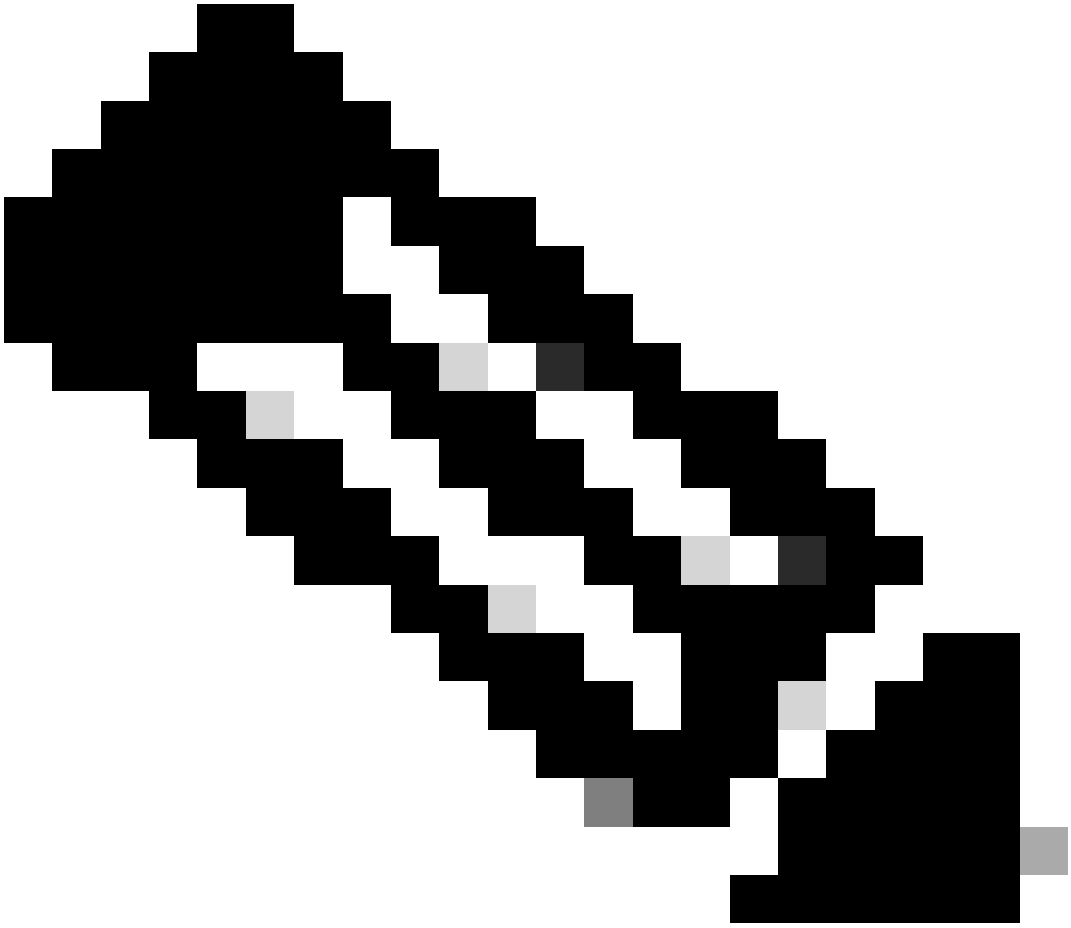
Exemplo de um pacote ServerHello no Wireshark

Configurar

O formato de string de cifra OpenSSL inclui vários caracteres especiais para executar operações na string, como remover uma cifra específica ou um grupo de cifras compartilhando um componente comum. Como o objetivo dessas personalizações é geralmente remover cifras, os caracteres usados nesses exemplos são:

- O caractere -, usado para remover cifras da lista. Algumas ou todas as cifras removidas podem ser permitidas novamente por opções que aparecem mais tarde na string.
- O caractere !, também usado para remover cifras da lista. Ao usá-la, as cifras removidas não podem ser permitidas novamente por nenhuma outra opção que apareça mais tarde na string.
- O caractere :, que atua como o separador entre os itens na lista.

Ambos podem ser usados para remover uma cifra da cadeia de caracteres, no entanto ! é preferível. Para obter uma lista completa de caracteres especiais, consulte a página OpenSSL Ciphers Manpage.



Observação: o site do OpenSSL afirma que, ao usar o caractere !, "as cifras excluídas nunca poderão reaparecer na lista, mesmo que sejam explicitamente declaradas". Isso não significa que as cifras são excluídas permanentemente do sistema, ele se refere ao escopo da interpretação da sequência de cifras.

Desativar uma cifra específica

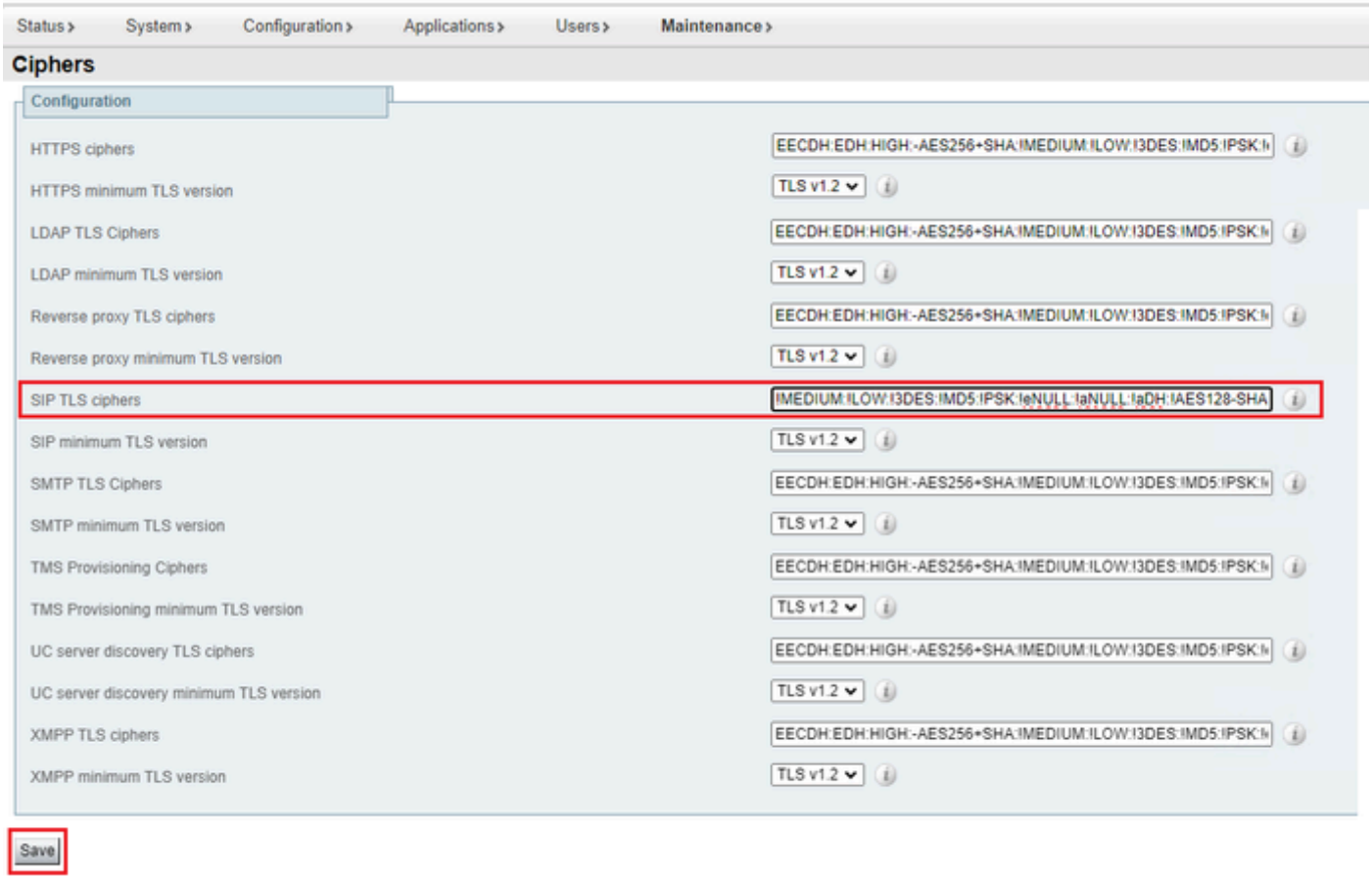
Para desabilitar uma cifra específica, anexe à cadeia de caracteres padrão o : separador, o sinal ! ou - e o nome da cifra a ser desabilitado. O nome da cifra deve obedecer ao formato de nomeação OpenSSL, disponível na página de manual de cifras OpenSSL. Por exemplo, se você precisar desativar a cifra AES128-SHA para conexões SIP, configure uma sequência de cifra como esta:

<#root>

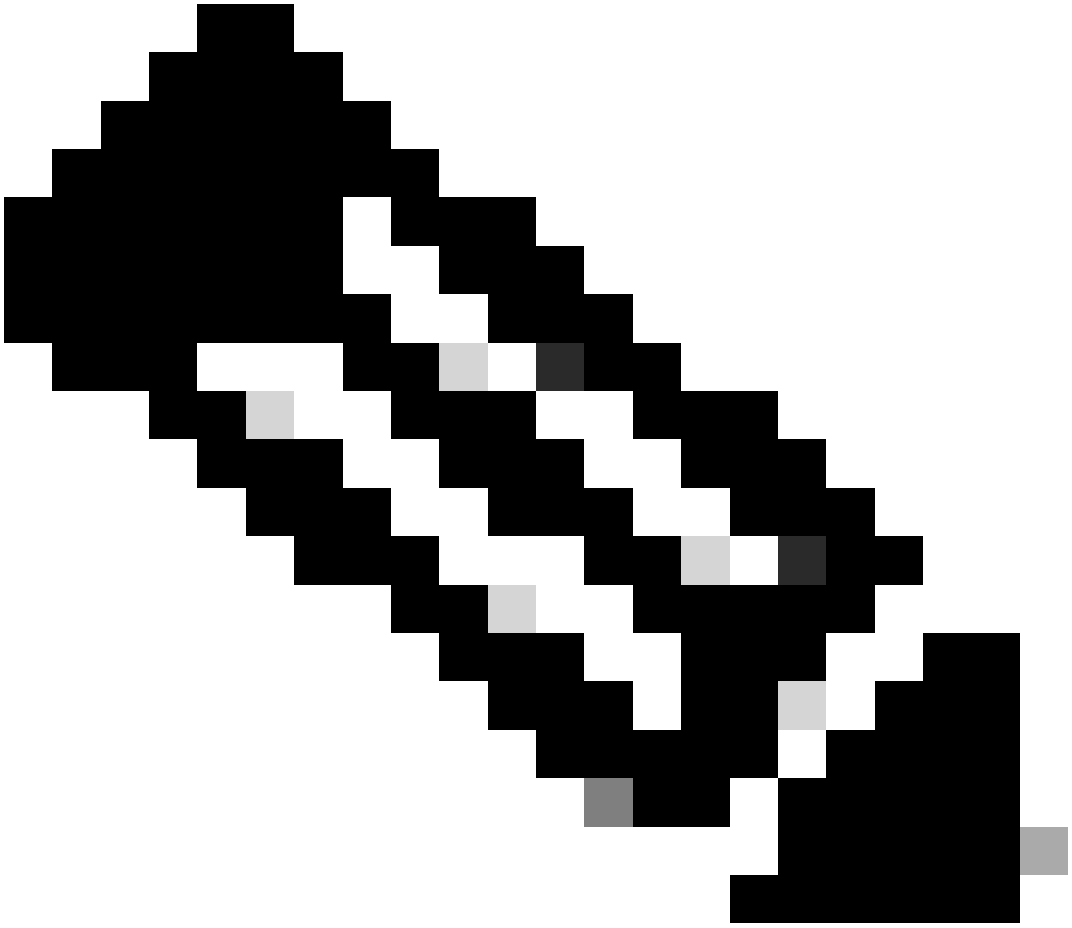
EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!eNULL:!aNULL:!aDH

:!AES128-SHA

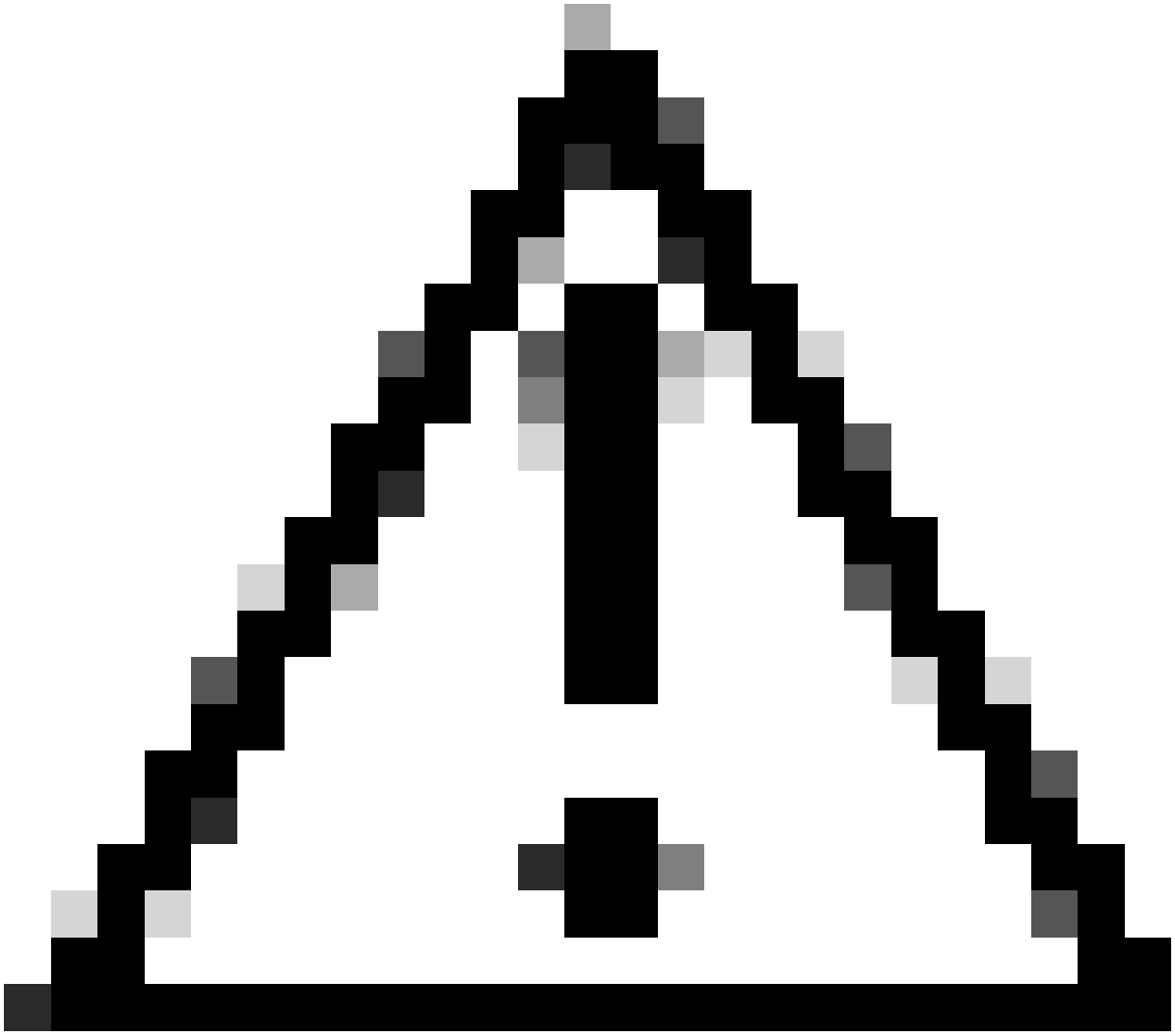
Em seguida, navegue até a página de administração da Web do Expressway, navegue até Manutenção > Segurança > Cifras, atribua a sequência de caracteres personalizada aos protocolos necessários e clique em Salvar. Para que a nova configuração seja aplicada, é necessário reiniciar o sistema. Neste exemplo, a string personalizada é atribuída ao protocolo SIP em cifras SIP TLS:



Página de configurações de codificação no portal de administração da Web do Expressway



Observação: no caso de um cluster do Expressway, faça as alterações somente no servidor primário. A nova configuração é replicada para o restante dos membros do cluster.



Cuidado: use a sequência de reinicialização de cluster recomendada fornecida no Guia de implantação de criação e manutenção de cluster do Cisco Expressway. Comece reiniciando o servidor primário, espere que ele esteja acessível através da interface da Web e, em seguida, faça o mesmo com cada peer na ordem de acordo com a lista configurada em System > Clustering.

Desativar um grupo de cifras usando um algoritmo comum

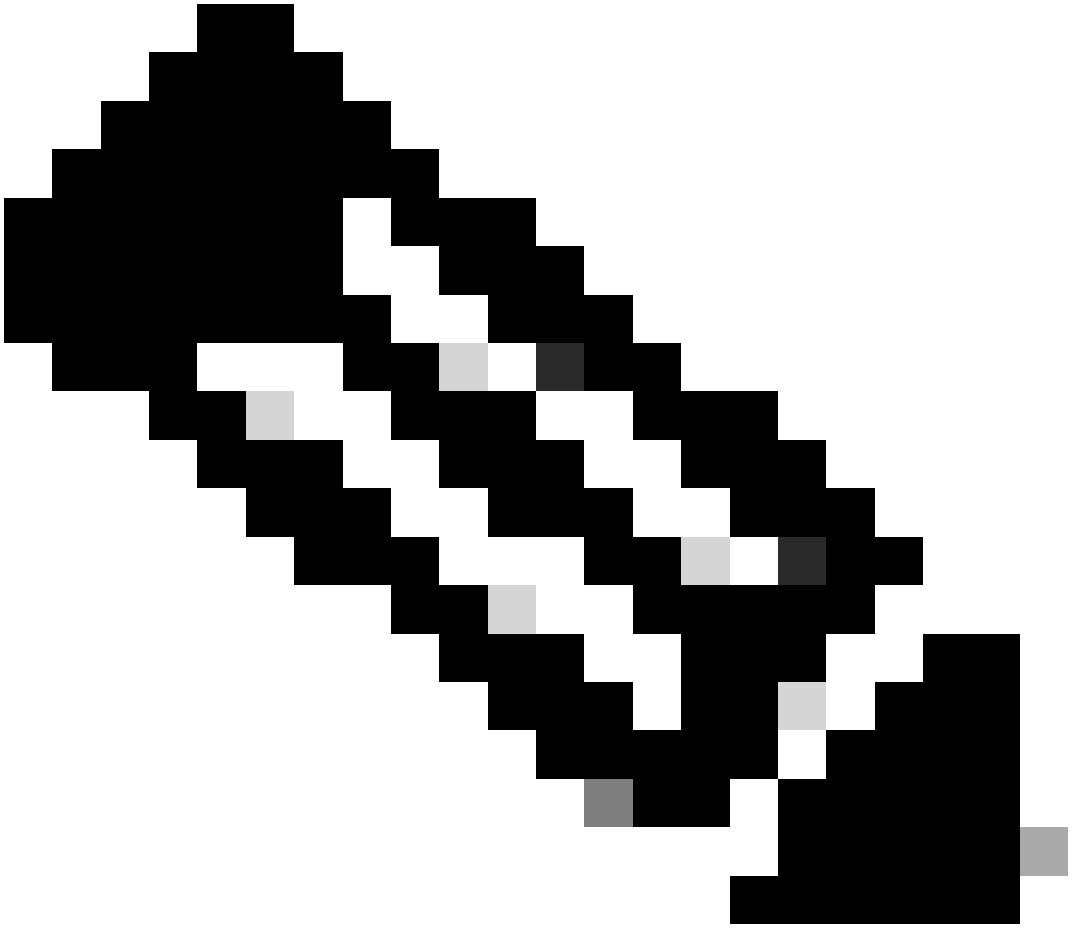
Para desabilitar um grupo de cifras usando um algoritmo comum, anexe à cadeia de caracteres padrão o : separador, o sinal ! ou - e o nome do algoritmo a ser desabilitado. Os nomes de algoritmos suportados estão disponíveis na página de manual OpenSSL Ciphers. Por exemplo, se você precisar desativar todas as cifras que usam o algoritmo DHE, configure uma sequência de cifras como esta:

<#root>

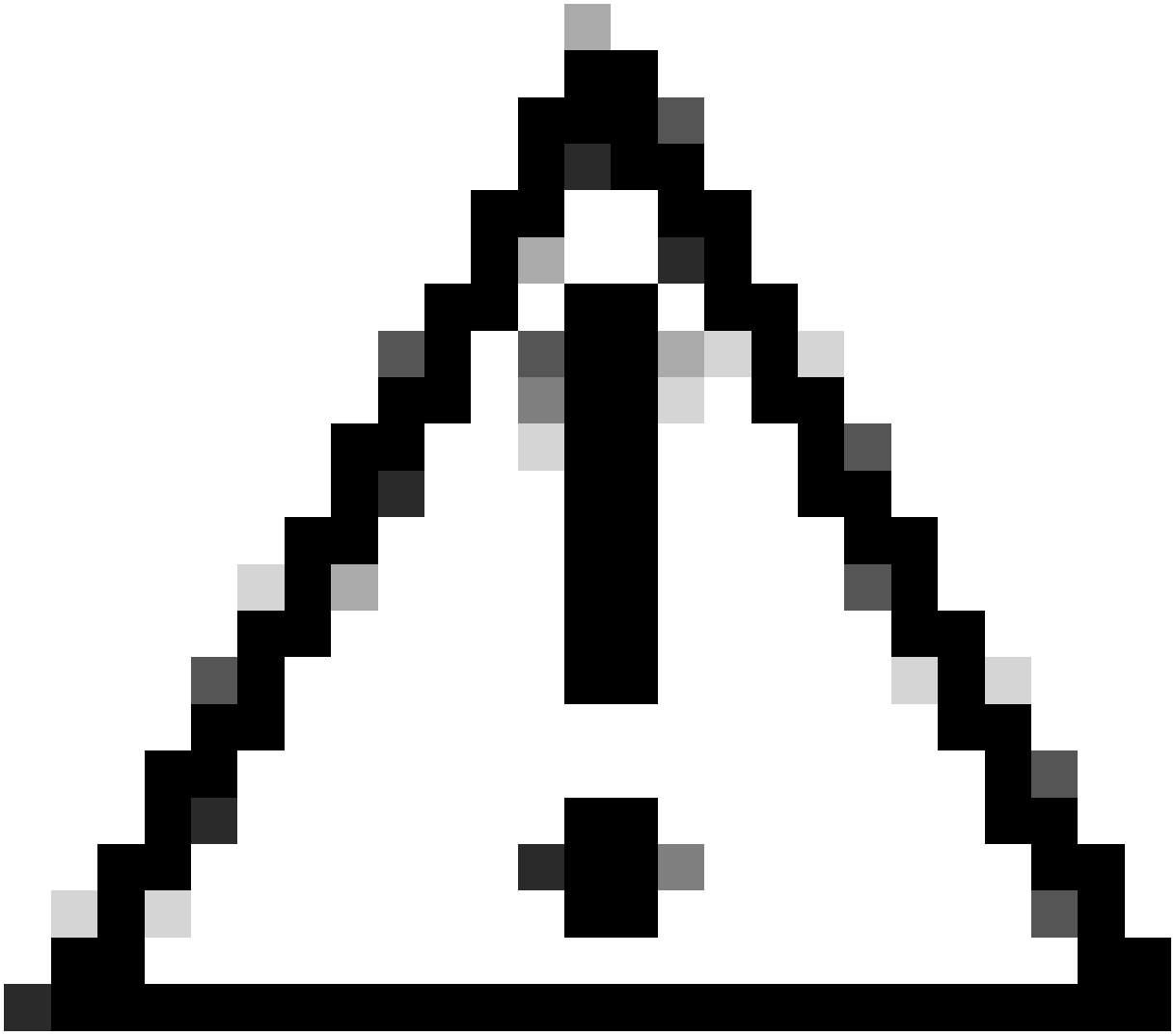
```
EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!eNULL:!aNULL:!aDH
```

:!DHE

Navegue para a página Expressway web admin, navegue para Manutenção > Segurança > Cifras, atribua a string personalizada aos protocolos necessários e clique em Salvar. Para que a nova configuração seja aplicada, é necessário reiniciar o sistema.



Observação: no caso de um cluster do Expressway, faça as alterações somente no servidor primário. A nova configuração é replicada para o restante dos membros do cluster.



Cuidado: use a sequência de reinicialização de cluster recomendada fornecida no Guia de implantação de criação e manutenção de cluster do Cisco Expressway. Comece reiniciando o servidor primário, espere que ele esteja acessível através da interface da Web e, em seguida, faça o mesmo com cada peer na ordem de acordo com a lista configurada em System > Clustering.

Verificar

Inspecione a lista de cifras permitidas pela sequência de cifras

Você pode inspecionar a sequência de cifra personalizada usando o comando openssl ciphers -V "<cipher string>". Revise a saída para confirmar se as cifras indesejadas não estão mais listadas após as alterações. Neste exemplo, a cadeia de caracteres EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!eNULL:!aNULL:!aDH:!DHE é inspecionada. A saída do comando confirma que a string não permite nenhuma das cifras que usam o algoritmo DHE:

<#root>

```
    ~ # openssl ciphers -V "EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!eNULL:!aNULL:!aDH
:!DHE
"
0x13,0x02 - TLS_AES_256_GCM_SHA384 TLSv1.3 Kx=any Au=any Enc=AESGCM(256) Mac=AEAD
0x13,0x03 - TLS_CHACHA20_POLY1305_SHA256 TLSv1.3 Kx=any Au=any Enc=CHACHA20/POLY1305(256) Mac=AEAD
0x13,0x01 - TLS_AES_128_GCM_SHA256 TLSv1.3 Kx=any Au=any Enc=AESGCM(128) Mac=AEAD
0xC0,0x2C - ECDHE-ECDSA-AES256-GCM-SHA384 TLSv1.2 Kx=ECDH Au=ECDSA Enc=AESGCM(256) Mac=AEAD
0xC0,0x30 - ECDHE-RSA-AES256-GCM-SHA384 TLSv1.2 Kx=ECDH Au=RSA Enc=AESGCM(256) Mac=AEAD
0xCC,0xA9 - ECDHE-ECDSA-CHACHA20-POLY1305 TLSv1.2 Kx=ECDH Au=ECDSA Enc=CHACHA20/POLY1305(256) Mac=AEAD
0xCC,0xA8 - ECDHE-RSA-CHACHA20-POLY1305 TLSv1.2 Kx=ECDH Au=RSA Enc=CHACHA20/POLY1305(256) Mac=AEAD
0xC0,0xAD - ECDHE-ECDSA-AES256-CCM TLSv1.2 Kx=ECDH Au=ECDSA Enc=AESCCM(256) Mac=AEAD
0xC0,0x2B - ECDHE-ECDSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH Au=ECDSA Enc=AESGCM(128) Mac=AEAD
0xC0,0x2F - ECDHE-RSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH Au=RSA Enc=AESGCM(128) Mac=AEAD
0xC0,0xAC - ECDHE-ECDSA-AES128-CCM TLSv1.2 Kx=ECDH Au=ECDSA Enc=AESCCM(128) Mac=AEAD
0xC0,0x24 - ECDHE-ECDSA-AES256-SHA384 TLSv1.2 Kx=ECDH Au=ECDSA Enc=AES(256) Mac=SHA384
0xC0,0x28 - ECDHE-RSA-AES256-SHA384 TLSv1.2 Kx=ECDH Au=RSA Enc=AES(256) Mac=SHA384
0xC0,0x23 - ECDHE-ECDSA-AES128-SHA256 TLSv1.2 Kx=ECDH Au=ECDSA Enc=AES(128) Mac=SHA256
0xC0,0x27 - ECDHE-RSA-AES128-SHA256 TLSv1.2 Kx=ECDH Au=RSA Enc=AES(128) Mac=SHA256
0xC0,0x09 - ECDHE-ECDSA-AES128-SHA TLSv1 Kx=ECDH Au=ECDSA Enc=AES(128) Mac=SHA1
0xC0,0x13 - ECDHE-RSA-AES128-SHA TLSv1 Kx=ECDH Au=RSA Enc=AES(128) Mac=SHA1
0x00,0x9D - AES256-GCM-SHA384 TLSv1.2 Kx=RSA Au=RSA Enc=AESGCM(256) Mac=AEAD
0xC0,0x9D - AES256-CCM TLSv1.2 Kx=RSA Au=RSA Enc=AESCCM(256) Mac=AEAD
0x00,0x9C - AES128-GCM-SHA256 TLSv1.2 Kx=RSA Au=RSA Enc=AESGCM(128) Mac=AEAD
0xC0,0x9C - AES128-CCM TLSv1.2 Kx=RSA Au=RSA Enc=AESCCM(128) Mac=AEAD
0x00,0x3D - AES256-SHA256 TLSv1.2 Kx=RSA Au=RSA Enc=AES(256) Mac=SHA256
0x00,0x3C - AES128-SHA256 TLSv1.2 Kx=RSA Au=RSA Enc=AES(128) Mac=SHA256
0x00,0x2F - AES128-SHA SSLv3 Kx=RSA Au=RSA Enc=AES(128) Mac=SHA1
~ #
```

Testar uma conexão TLS negociando uma cifra desativada

Você pode usar o comando openssl s_client para verificar se uma tentativa de conexão usando uma cifra desativada foi rejeitada. Use a opção -connect para especificar o endereço e a porta do Expressway, e use a opção -cipher para especificar a cifra única a ser negociada pelo cliente durante o handshake TLS:

openssl s_client -connect <endereço>:<porta> -cipher <cifra> -no_tls1_3

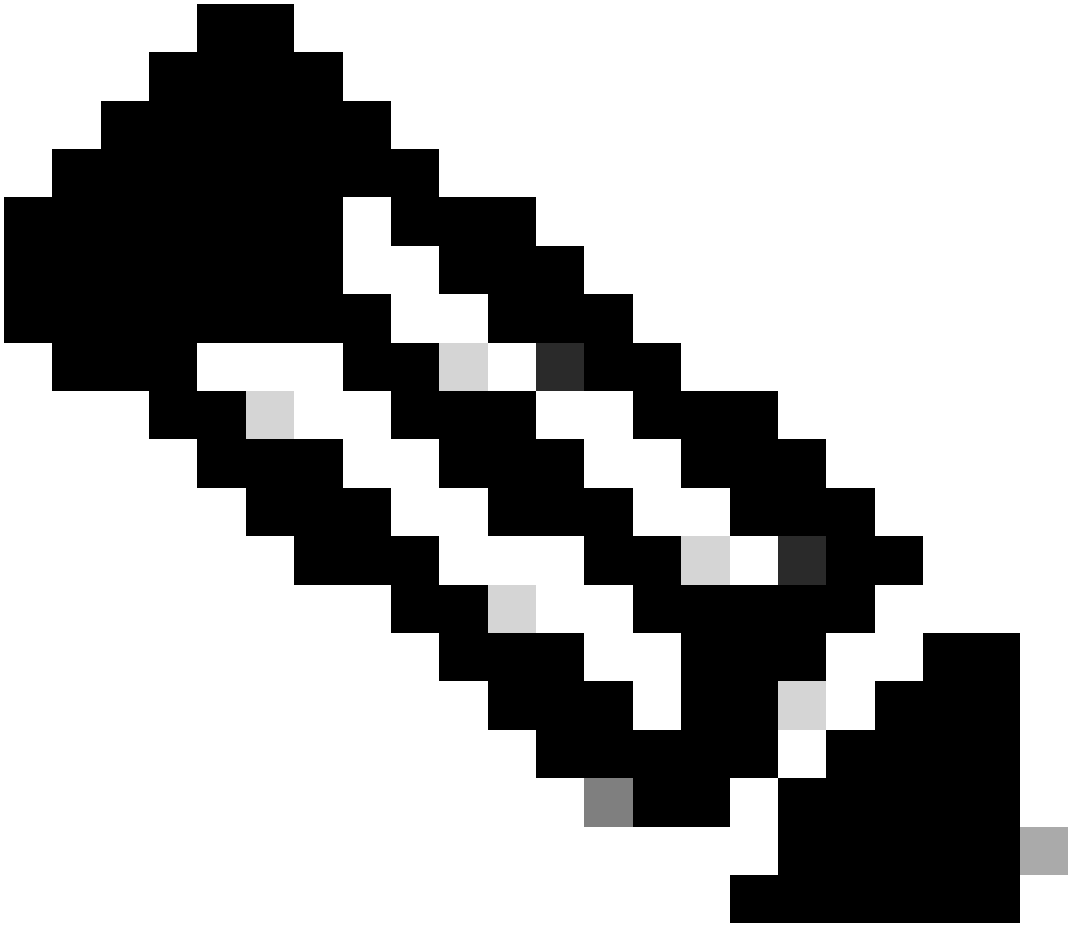
Neste exemplo, uma conexão TLS para o Expressway é tentada em um PC com Windows com o openssl instalado. O PC, como cliente, negocia somente a cifra DHE-RSA-AES256-CCM indesejada, que usa o algoritmo DHE:

<#root>

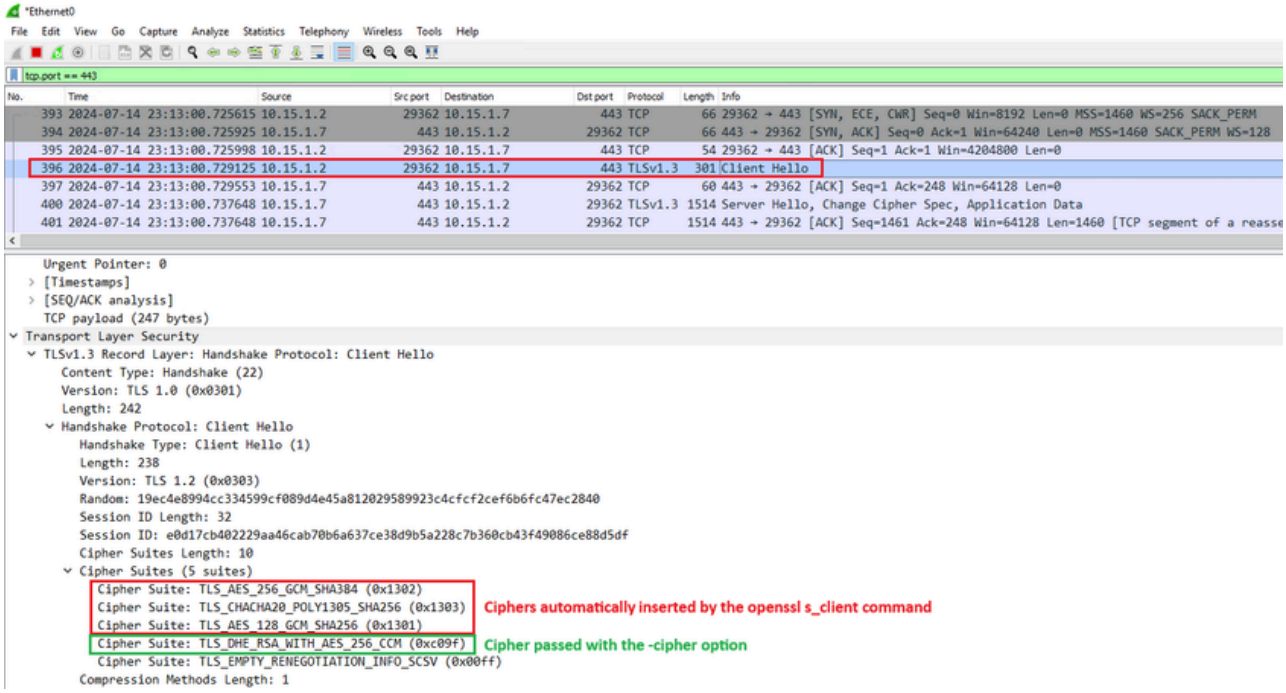
```
C:\Users\Administrator>

openssl s_client -connect exp.example.com:443 -cipher DHE-RSA-AES256-CCM -no_tls1_3


Connecting to 10.15.1.7
CONNECTED(00000154)
D0130000:error:0A000410:SSL routines:ssl3_read_bytes:
```

```
ssl/tls alert handshake failure

:..\ssl\record\rec_layer_s3.c:865:

SSL alert number 40


---
no peer certificate available
---
No client certificate CA names sent
---
SSL handshake has read 7 bytes and written 118 bytes
Verification: OK
---
New, (NONE), Cipher is (NONE)
Secure Renegotiation IS NOT supported
No ALPN negotiated
SSL-Session:
Protocol : TLSv1.2
Cipher : 0000
Session-ID:
Session-ID-ctx:
Master-Key:
PSK identity: None
PSK identity hint: None
SRP username: None
Start Time: 1721019437
Timeout : 7200 (sec)
Verify return code: 0 (ok)
Extended master secret: no
---

C:\Users\Administrator>
```

A saída do comando mostra que a tentativa de conexão falha com uma mensagem de erro "ssl/tls alert handshake failure:..\ssl\record\rec_layer_s3.c:865:SSL alert number 40", porque o Expressway está configurado para usar a string EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!PSK:!eNULL:!aNULL:!aDH:!DHE cipher string para conexões HTTPS, que desabilita cifras que usam o algoritmo DHE.



Observação: para que os testes com o comando openssl s_client funcionem conforme explicado, a opção -no_tls1_3 precisa ser passada para o comando. Se não for incluído, o cliente insere automaticamente cifras TLS 1.3 no pacote ClientHello:



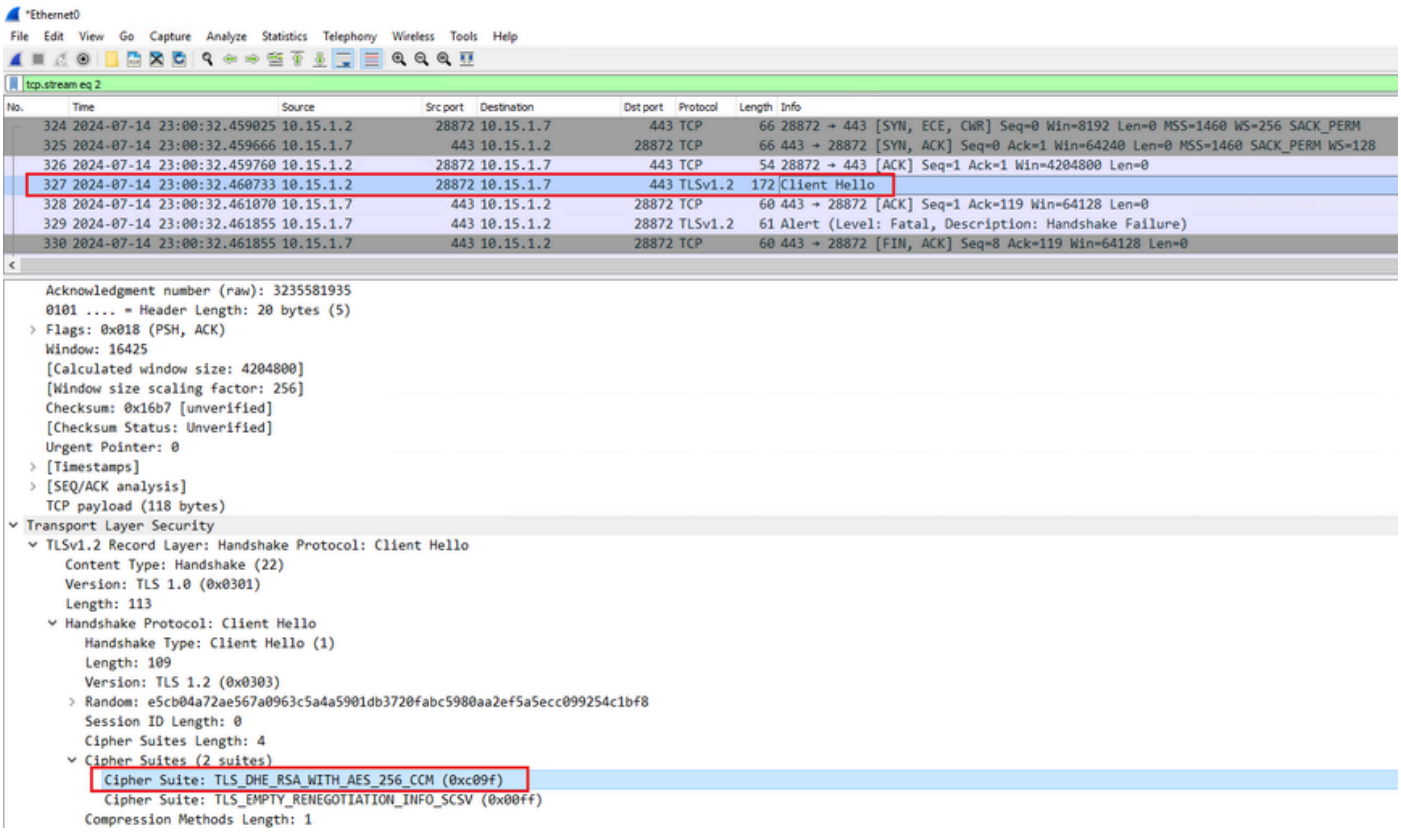
Pacote ClientHello com cifras adicionadas automaticamente

Se o Expressway de destino suportar essas cifras, uma delas pode ser escolhida em vez da cifra específica que você precisa testar. A conexão é bem-sucedida, o que pode levá-lo a acreditar que uma conexão foi possível usando a cifra desativada passada para o comando com a opção -cipher.

Inspecionar uma captura de pacote de um handshake TLS usando uma cifra desativada

Você pode coletar uma captura de pacote, do dispositivo de teste ou do Expressway, enquanto executa um teste de conexão usando uma das cifras desativadas. Você pode então inspecioná-lo com o Wireshark para analisar mais detalhadamente os eventos de handshake.

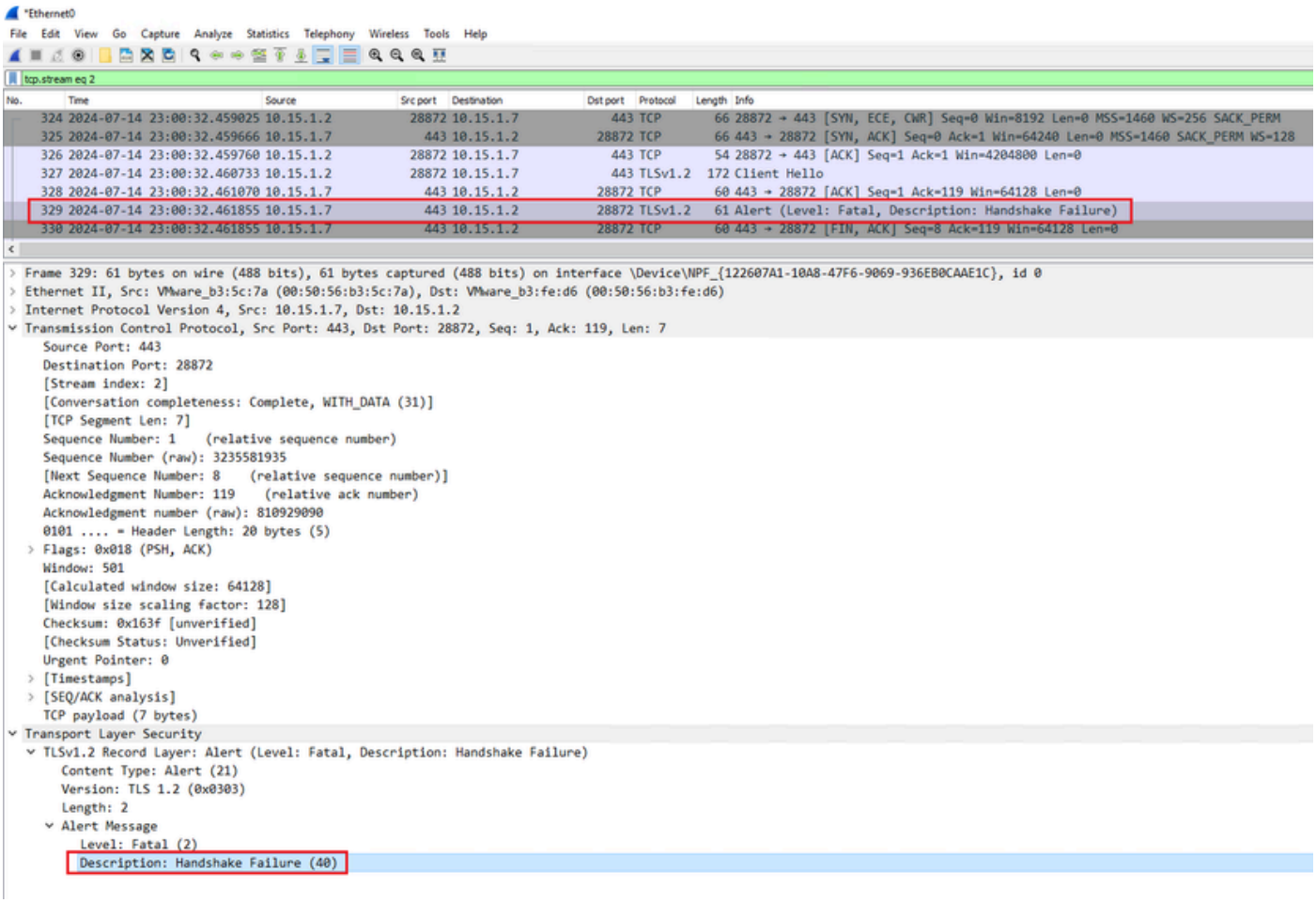
Localize o ClientHello enviado pelo dispositivo de teste. Confirme se ele negocia apenas a cifra de teste indesejada, neste exemplo, uma cifra usando o algoritmo DHE:



Exemplo de um pacote ClientHello no Wireshark

:

Confirme se o Expressway responde com um pacote de alerta TLS fatal, recusando a conexão. Neste exemplo, como o Expressway não suporta cifras DHE por sua string de cifra configurada para o protocolo HTTPS, ele responde com um pacote de alerta TLS fatal contendo o código de falha 40.



Um pacote de alerta fatal TLS no Wireshark

Informações Relacionadas

- [Página do OpenSSL Ciphers Manager]
- [Guia do administrador do Cisco Expressway (X15.0) - Capítulo: Gerenciando a segurança - Configurando a versão TLS mínima e pacotes de codificação]

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.