Corrigir erro de algoritmos criptográficos do AnyConnect com FIPS ativado

Contents

Introduction

Este documento descreve por que os usuários podem não ser capazes de se conectar com o uso de um cliente habilitado para FIPS (Federal Information Processing Standard) a um ASA (Adaptive Security Appliance), que tem uma política que suporta algoritmos de criptografia habilitados para FIPS.

Informações de Apoio

Durante a configuração de uma conexão Internet Key Exchange Version 2 (IKEv2), o iniciador nunca sabe quais propostas são aceitáveis pelo par, então o iniciador deve adivinhar qual grupo Diffie-Hellman (DH) usar quando a primeira mensagem IKE for enviada. O grupo DH usado para esta estimativa é geralmente o primeiro grupo DH na lista de grupos DH configurados. Em seguida, o iniciador calcula os dados de chave para os grupos adivinhados, mas também envia uma lista completa de todos os grupos para o peer, o que permite que o peer selecione um grupo DH diferente se o grupo estimado estiver errado.

No caso de um cliente, não há lista configurada pelo usuário de políticas IKE. Em vez disso, há uma lista pré-configurada de políticas suportadas pelo cliente. Por isso, a fim de reduzir a carga computacional no cliente quando você calcula os dados chave para a primeira mensagem com um grupo que é possivelmente o errado, a lista de grupos DH foi ordenada do mais fraco ao mais forte. Assim, o cliente escolhe o DH menos intensivo computacionalmente e, portanto, o grupo menos intensivo de recursos para a suposição inicial, mas então muda para o grupo escolhido pelo headend em mensagens subsequentes.

> **Note**: Esse comportamento é diferente dos clientes do AnyConnect versão 3.0 que solicitaram os grupos DH do mais forte para o mais fraco.

No entanto, no headend, o primeiro grupo DH na lista enviada pelo cliente que corresponde a um grupo DH configurado no gateway é o grupo selecionado. Portanto, se o ASA também tiver grupos DH mais fracos configurados, ele usará o grupo DH mais fraco suportado pelo cliente e configurado no headend, apesar da disponibilidade de um grupo DH mais seguro em ambas as extremidades.

Este comportamento foi corrigido no cliente através da ID de bug Cisco CSCub92935. Todas as versões de cliente com a correção deste bug revertem a ordem na qual os grupos DH são listados

quando são enviados ao headend. No entanto, para evitar um problema de compatibilidade com versões anteriores com gateways não Suite B, o grupo DH mais fraco (um para o modo não FIPS e dois para o modo FIPS) permanece no topo da lista.

> **Note**: Após a primeira entrada na lista (grupo 1 ou 2), os grupos são listados na ordem do mais forte para o mais fraco. Isso coloca os grupos de curva elíptica em primeiro lugar (21, 20, 19), seguido pelos grupos Modular Exponencial (MODP) (24, 14, 5, 2).

> **Tip**: Se o gateway estiver configurado com vários grupos DH na mesma política e o grupo 1 (ou 2 no modo FIPS) estiver incluído, o ASA aceitará o grupo mais fraco. A correção é incluir apenas o grupo DH 1 em uma política configurada no gateway. Quando vários grupos são configurados em uma política, mas o grupo 1 não é incluído, o mais forte é selecionado. Por exemplo:
>
> - No ASA versão 9.0 (conjunto B) com política IKEv2 definida como 1 2 5 14 24 19 20 21, o **grupo 1 é selecionado** conforme esperado.
>
> - No ASA versão 9.0 (conjunto B) com política IKEv2 definida como 2 5 14 24 19 20 21, o **grupo 21 é selecionado** conforme esperado.
>
> - Com o cliente no modo FIPS no ASA versão 9.0 (conjunto B) com a política IKEv2 definida como 1 2 5 14 24 19 20 21, o **grupo 2 é selecionado** conforme esperado.
>
> - Com o cliente testado no modo FIPS no ASA versão 9.0 (conjunto B) com a política IKEv2 definida como 5 14 24 19 20 21, o **grupo 21 é selecionado** como esperado.
>
> - No ASA versão 8.4.4 (não-suite B) com política IKEv2 definida como 1 2 5 14, o **grupo 1 é selecionado** conforme esperado.
>
> - No ASA versão 8.4.4 (não-suite B) com política IKEv2 definida como 2 5 14, o **grupo 14 é selecionado** conforme esperado.

Problema

O ASA é configurado com estas políticas IKEv2:

```
crypto ikev2 policy 1
encryption aes-gcm-256
integrity null
group 20
prf sha384 sha
lifetime seconds 86400
crypto ikev2 policy 10
encryption aes-192
integrity sha
group 5 2
prf sha
lifetime seconds 86400
crypto ikev2 policy 20
encryption aes
integrity sha
group 5 2
```

```
prf sha
lifetime seconds 86400
```

Nesta configuração, a política 1 é claramente configurada para suportar todos os algoritmos criptográficos ativados por FIPS. No entanto, quando um usuário tenta se conectar de um cliente habilitado para FIPS, a conexão falha com a mensagem de erro:

```
The cryptographic algorithms required by the secure gateway do not match those supported by
AnyConnect.
Please contact your network administrator.
```

No entanto, se o administrador alterar a política1 de modo que use o grupo DH 2 em vez de 20, a conexão funcionará.

Solução

Com base nos sintomas, a primeira conclusão seria que o cliente só dá suporte ao grupo DH 2 quando o FIPS está habilitado e nenhum dos outros funciona. Na verdade, isso está incorreto. Se você habilitar essa depuração no ASA, poderá ver as propostas enviadas pelo cliente:

```
debug crypto ikev2 proto 127
```

Durante uma tentativa de conexão, a primeira mensagem de depuração é:

```
IKEv2-PROTO-2: Received Packet [From 192.168.30.5:51896/To 192.168.30.2:500/
VRF i0:f0]
Initiator SPI : 74572B8D1BEC5873 - Responder SPI : 0000000000000000 Message id: 0
IKEv2 IKE_SA_INIT Exchange REQUESTIKEv2-PROTO-3: Next payload: SA, version:
2.0 Exchange type: IKE_SA_INIT, flags: INITIATOR Message id: 0, length: 747
Payload contents:
SA Next payload: KE, reserved: 0x0, length: 316
last proposal: 0x2, reserved: 0x0, length: 140
Proposal: 1, Protocol id: IKE, SPI size: 0, #trans: 15 last transform: 0x3,
reserved: 0x0: length: 12
type: 1, reserved: 0x0, id: AES-GCM
last transform: 0x3, reserved: 0x0: length: 12
type: 1, reserved: 0x0, id: AES-GCM
last transform: 0x3, reserved: 0x0: length: 12
type: 1, reserved: 0x0, id: AES-GCM
last transform: 0x3, reserved: 0x0: length: 8
type: 2, reserved: 0x0, id: SHA512
last transform: 0x3, reserved: 0x0: length: 8
type: 2, reserved: 0x0, id: SHA384
last transform: 0x3, reserved: 0x0: length: 8
type: 2, reserved: 0x0, id: SHA256
last transform: 0x3, reserved: 0x0: length: 8
type: 2, reserved: 0x0, id: SHA1
last transform: 0x3, reserved: 0x0: length: 8
type: 3, reserved: 0x0, id: None
last transform: 0x3, reserved: 0x0: length: 8
type: 4, reserved: 0x0, id: DH_GROUP_1024_MODP/Group 2
last transform: 0x3, reserved: 0x0: length: 8
type: 4, reserved: 0x0, id: DH_GROUP_521_ECP/Group 21
last transform: 0x3, reserved: 0x0: length: 8
type: 4, reserved: 0x0, id: DH_GROUP_384_ECP/Group 20
last transform: 0x3, reserved: 0x0: length: 8
type: 4, reserved: 0x0, id: DH_GROUP_256_ECP/Group 19
last transform: 0x3, reserved: 0x0: length: 8
```

```
type: 4, reserved: 0x0, id: DH_GROUP_2048_MODP_256_PRIME/Group 24
last transform: 0x3, reserved: 0x0: length: 8
type: 4, reserved: 0x0, id: DH_GROUP_2048_MODP/Group 14
last transform: 0x0, reserved: 0x0: length: 8
type: 4, reserved: 0x0, id: DH_GROUP_1536_MODP/Group 5
last proposal: 0x0, reserved: 0x0, length: 172
Proposal: 2, Protocol id: IKE, SPI size: 0, #trans: 19 last transform: 0x3,
reserved: 0x0: length: 12
type: 1, reserved: 0x0, id: AES-CBC
last transform: 0x3, reserved: 0x0: length: 12
type: 1, reserved: 0x0, id: AES-CBC
last transform: 0x3, reserved: 0x0: length: 12
type: 1, reserved: 0x0, id: AES-CBC
last transform: 0x3, reserved: 0x0: length: 8
type: 1, reserved: 0x0, id: 3DES
last transform: 0x3, reserved: 0x0: length: 8
type: 2, reserved: 0x0, id: SHA512
last transform: 0x3, reserved: 0x0: length: 8
type: 2, reserved: 0x0, id: SHA384
last transform: 0x3, reserved: 0x0: length: 8
type: 2, reserved: 0x0, id: SHA256
last transform: 0x3, reserved: 0x0: length: 8
type: 2, reserved: 0x0, id: SHA1
last transform: 0x3, reserved: 0x0: length: 8
type: 3, reserved: 0x0, id: SHA512
last transform: 0x3, reserved: 0x0: length: 8
type: 3, reserved: 0x0, id: SHA384
last transform: 0x3, reserved: 0x0: length: 8
type: 3, reserved: 0x0, id: SHA256
last transform: 0x3, reserved: 0x0: length: 8
type: 3, reserved: 0x0, id: SHA96
last transform: 0x3, reserved: 0x0: length: 8
type: 4, reserved: 0x0, id: DH_GROUP_1024_MODP/Group 2
last transform: 0x3, reserved: 0x0: length: 8
type: 4, reserved: 0x0, id: DH_GROUP_521_ECP/Group 21
last transform: 0x3, reserved: 0x0: length: 8
type: 4, reserved: 0x0, id: DH_GROUP_384_ECP/Group 20
last transform: 0x3, reserved: 0x0: length: 8
type: 4, reserved: 0x0, id: DH_GROUP_256_ECP/Group 19
last transform: 0x3, reserved: 0x0: length: 8
type: 4, reserved: 0x0, id: DH_GROUP_2048_MODP_256_PRIME/Group 24
last transform: 0x3, reserved: 0x0: length: 8
type: 4, reserved: 0x0, id: DH_GROUP_2048_MODP/Group 14
last transform: 0x0, reserved: 0x0: length: 8
type: 4, reserved: 0x0, id: DH_GROUP_1536_MODP/Group 5
KE Next payload: N, reserved: 0x0, length: 136
DH group: 2, Reserved: 0x0

fc c9 90 2b 15 35 31 34 0e 75 88 c0 f9 2a 1e 0a
a5 6b e3 8e e1 73 b9 d1 56 1e 60 9f 82 71 6c 4e
5c 1c a4 bd b5 23 a2 bc 82 f2 11 17 61 28 33 3f
02 c9 e7 cb f7 84 a6 22 4a 64 eb fa d7 84 a1 d9
ad c7 5d 77 cd 2a 65 79 95 9a d4 5c 22 8c 62 ae
0e fc c8 fd bd c8 4d 66 0d c3 69 d3 c4 cb e8 33
72 1a f1 cc 31 5f 08 75 65 6b 77 3b 23 c3 b8 74
02 fa 15 6e e4 7a b2 73 17 8f 08 02 20 7e b8 d7
N Next payload: VID, reserved: 0x0, length: 24

87 4d 63 76 cc 10 30 0e 4c 95 40 24 d3 b3 3b f3
44 be 0f e5
```

Portanto, apesar do fato de que o cliente enviou os grupos 2,21,20,19,24,14 e 5 (esses grupos compatíveis com FIPS), o headend ainda só conecta o grupo 2 habilitado na política 1 na configuração anterior. Esse problema fica evidente mais abaixo nas depurações:

```
IKEv2 received all requested SPIs from CTM to respond to a tunnel request.
IKEv2-PROTO-5: (64): SM Trace-> SA: I_SPI=74572B8D1BEC5873 R_SPI=E4160C492A824B5F
(R) MsgID = 00000006 CurState: R_VERIFY_AUTH Event: EV_OK_RECD_IPSEC_RESP
IKEv2-PROTO-2: (64): Processing IKE_AUTH message
IKEv2-PROTO-1: Tunnel Rejected: Selected IKEv2 encryption algorithm (AES-CBC-192)
is not strong enough to secure proposed IPsec encryption algorithm (AES-GCM-256).
IKEv2-PROTO-1: (64): Failed to find a matching policy
IKEv2-PROTO-1: (64): Received Policies:
ESP: Proposal 1: AES-GCM-256 AES-GCM-192 AES-GCM-128 None Don't use ESN

ESP: Proposal 2: AES-CBC-256 AES-CBC-192 AES-CBC-128 3DES SHA512 SHA384 SHA256 SHA96
Don't use ESN

IKEv2-PROTO-1: (64): Failed to find a matching policy
IKEv2-PROTO-1: (64): Expected Policies:
ESP: Proposal 0: AES-GCM-256 SHA384 Don't use ESN

IKEv2-PROTO-5: (64): Failed to verify the proposed policies
IKEv2-PROTO-1: (64): Failed to find a matching policy
```

A conexão falha devido a uma combinação de fatores:

1. Com o FIPS habilitado, o cliente envia apenas políticas específicas e elas devem ser correspondentes. Entre essas políticas, ele propõe apenas a criptografia AES (Advanced Encryption Standard) com um tamanho de chave maior ou igual a 256.

2. O ASA é configurado com várias políticas IKEv2, duas das quais têm o grupo 2 habilitado. Conforme descrito anteriormente, nesse cenário, a política que tem o grupo 2 habilitado é usada para a conexão. No entanto, o algoritmo de criptografia em ambas as políticas usa um tamanho de chave de 192, que é muito baixo para um cliente habilitado para FIPS.

Portanto, neste caso, o ASA e o cliente se comportam de acordo com a configuração. Há três maneiras de resolver esse problema para clientes habilitados para FIPS:

1. Configure apenas uma política com as propostas exatas desejadas.

2. Se forem necessárias várias propostas, não configure uma com o grupo 2; caso contrário, essa opção será sempre selecionada.

3. Se o grupo 2 precisar ser ativado, verifique se ele tem o algoritmo de criptografia correto configurado (Aes-256 ou aes-gcm-256).