Configuration des algorithmes de chiffrement, MAC et Kex sur les plates-formes Nexus

Table des matières

Introduction

Ce document décrit les étapes à suivre pour ajouter (ou supprimer) des algorithmes de chiffrement, MAC et Kex sur les plates-formes Nexus.

Conditions préalables

Exigences

Cisco vous recommande de comprendre les bases de Linux et de Bash.

Composants utilisés

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Nexus 3000 et 9000 NX-OS 7.0(3)I7(10)
- Nexus 3000 et 9000 NX-OS 9.3(13)
- Nexus 9000 NX-OS 10.2(7)
- Nexus 9000 NX-OS 10.3(5)
- Nexus 7000 NX-OS 8.4(8)
- Nexus 5600 NX-OS 7.3(14)N1(1)

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l'incidence possible des commandes.

Informations générales

Parfois, les analyses de sécurité peuvent détecter des méthodes de cryptage faibles utilisées par les périphériques Nexus. Si cela se produit, des modifications doivent être apportées au fichierdcos_sshd_config sur les commutateurs pour supprimer ces algorithmes non sécurisés.

Examen des algorithmes Kex, MAC et de chiffrement disponibles

Pour confirmer les algorithmes de chiffrement, MAC et Kex utilisés par une plate-forme et vérifier cela à partir d'un périphérique externe, vous pouvez utiliser les options suivantes :

Option 1. Utilisation de la ligne CMD à partir du PC

Ouvrez une ligne CMD sur un PC qui peut atteindre le périphérique Nexus et utilisez la commande ssh -vvv <hostname> .

<#root>

C:\Users\xxxxx>ssh -vvv <hostname>
--------- snipped ------------
debug2: peer server KEXINIT proposal
debug2:

KEX algorithms: diffie-hellman-group1-sha1,diffie-hellman-group14-sha1,diffie-hellman-group-exchange-sha

debug2: host key algorithms: ssh-rsa
debug2: ciphers ctos: aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,aes192-cbc,aes256-cbc
debug2:

ciphers stoc: aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,aes192-cbc,aes256-cbc <--- encryption algorith

debug2: MACs ctos: hmac-sha1
debug2:

MACs stoc: hmac-sha1 <--- mac algorithms

debug2: compression ctos: none,zlib@openssh.com
debug2:

compression stoc: none,zlib@openssh.com <--- compression algorithms

Option 2. Accédez au fichier "dcos_sshd_config" à l'aide de **Feature Bash-Shell**

Ceci s'applique à :

- N3K en cours 7. X, 9. X, 10. X
- Tous les codes N9K
- N7K exécutant 8.2 et versions ultérieures

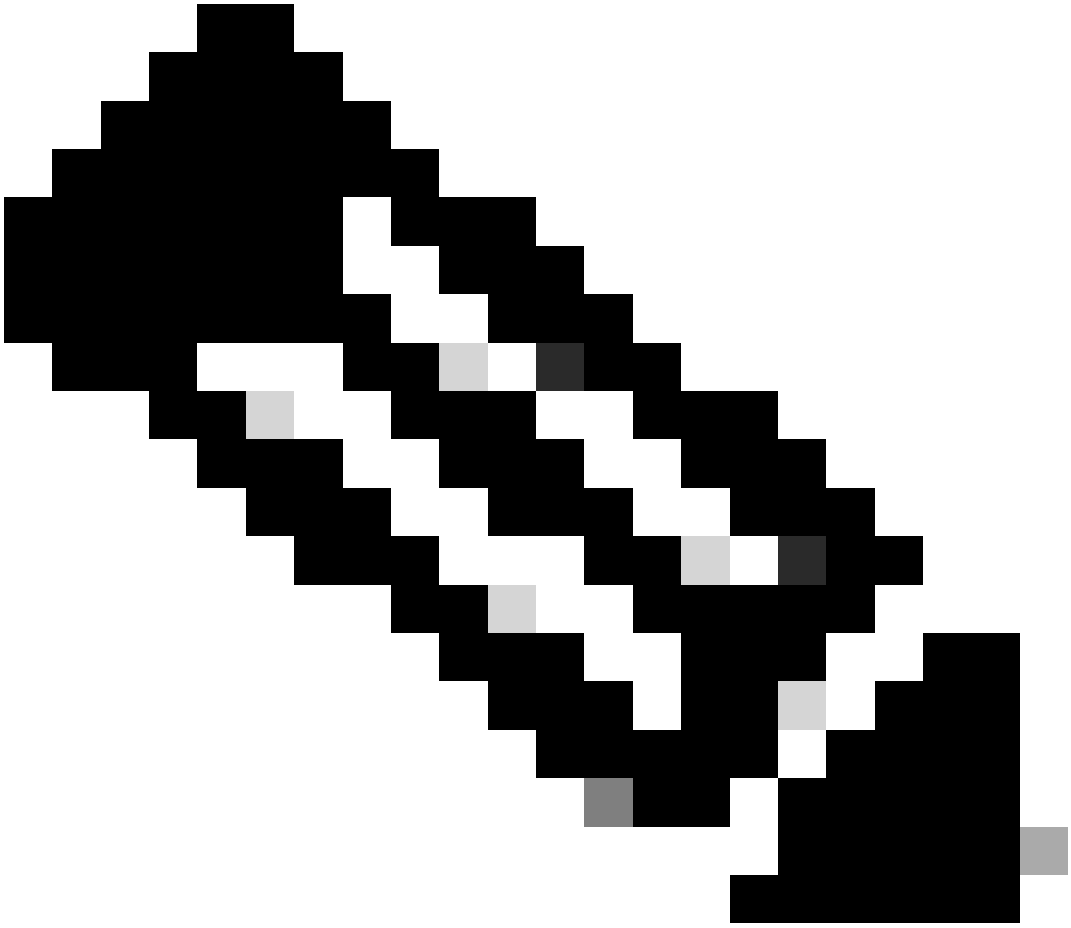
Étapes :

- Activez la fonctionnalité bash-shell et passez en mode bash :

```
switch(config)# feature bash-shell
switch(config)#
switch(config)# run bash
bash-4.3$
```

2. Examinez le contenu du fichierdcos_sshd_config :

```
bash-4.3$ cat /isan/etc/dcos_sshd_config
```



Remarque : vous pouvez utiliser egrep pour consulter des lignes spécifiques : cat /isan/etc/dcos_sshd_config | grep MAC

Option 3. Accédez au fichier « dcos_sshd_config » à l'aide d'un **fichier Dplug**

Ceci s'applique à :

- N3K en cours d'exécution 6. X qui n'a pas accès à bash-shell

- Tous les codes N5K et N6K

- N7Ks en cours d'exécution 6. X et 7. codes X

Étapes :

1. Ouvrez un dossier TAC pour obtenir le fichier dplug qui correspond à la version de NXOS exécutée sur le commutateur.

2. Téléchargez le fichier dplug dans la mémoire Flash de démarrage et créez-en une copie.

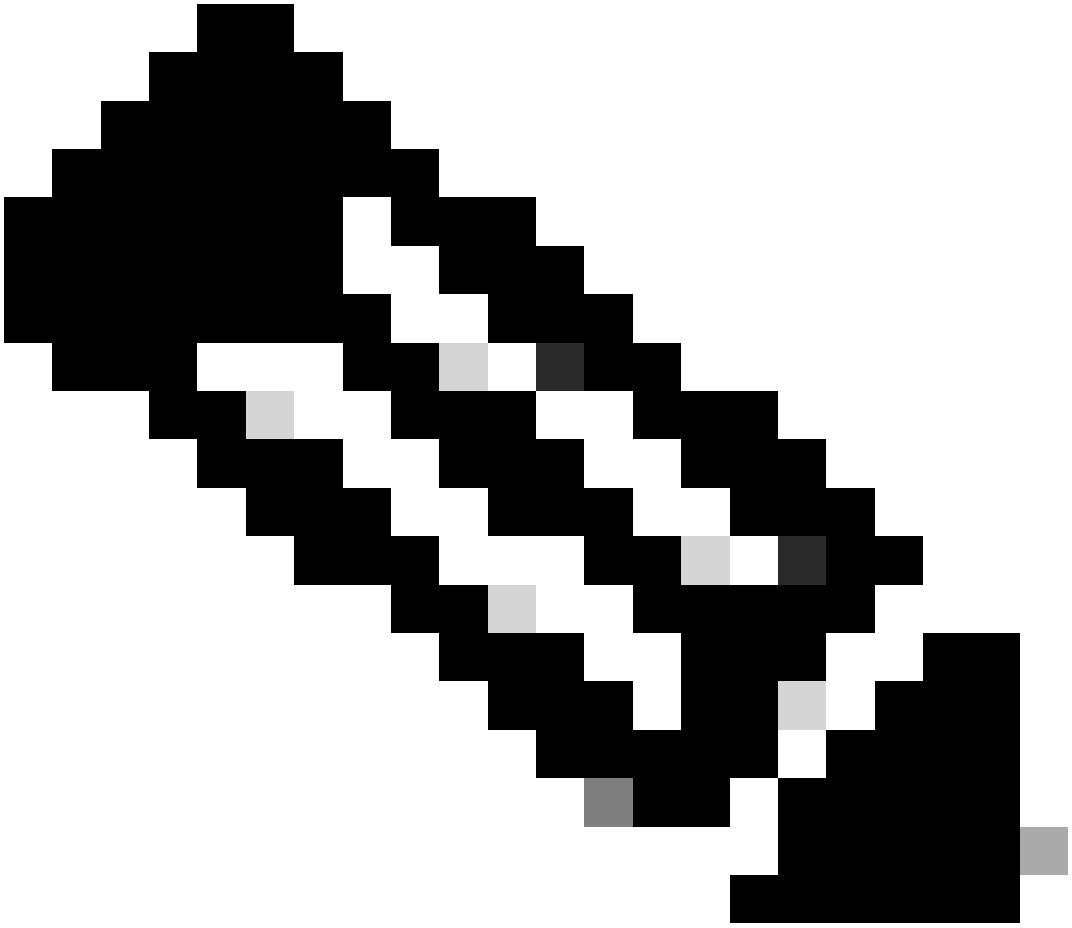
<#root>

switch# copy bootflash:

nuova-or-dplug-mzg.7.3.8.N1.1

bootflash:

dp



Remarque : une copie ("dp") du fichier dplug d'origine est créée dans bootflash, de sorte que seule la copie est supprimée après le chargement du fichier dplug et que le fichier dplug d'origine reste dans bootflash pour les exécutions suivantes.

3. Chargez la copie du dplug à l’aide de la load commande.

<#root>

n5k-1# load bootflash:dp
Loading plugin version 7.3(8)N1(1)
##
Warning: debug-plugin is for engineering internal use only!

For security reason, plugin image has been deleted.

```
#################################################################
Successfully loaded debug-plugin!!!
Linux(debug)#
Linux(debug)#
```

2. Examinez le fichierdcos_sshd_config.

```
Linux(debug)# cat /isan/etc/dcos_sshd_config
```

Solution

Étape 1. Exportez le fichier "dcos_sshd_config"

1. Envoyez une copie du fichierdcos_sshd_config à bootflash :

```
Linux(debug)# cd /isan/etc/
Linux(debug)# copy dcos_sshd_config /bootflash/dcos_sshd_config
Linux(debug)# exit
```

2. Vérifiez que la copie est sur bootflash :

```
switch(config)# dir bootflash: | i ssh
7372 Mar 24 02:24:13 2023 dcos_sshd_config
```

3. Exporter vers un serveur :

```
switch# copy bootflash: ftp:
Enter source filename: dcos_sshd_config
Enter vrf (If no input, current vrf 'default' is considered): management
Enter hostname for the ftp server: <hostname>
Enter username: <username>
Password:
***** Transfer of file Completed Successfully *****
Copy complete, now saving to disk (please wait)...
Copy complete.
```

4. Apportez les modifications nécessaires au fichier et réimportez-le dans bootflash.

Étape 2. Importez le fichier « dcos_sshd_config »

1. Téléchargez le fichier modifiédcos_sshd_config pour démarrer la mémoire flash.

```
switch# copy ftp: bootflash:
Enter source filename: dcos_sshd_config_modified.txt
Enter vrf (If no input, current vrf 'default' is considered): management
Enter hostname for the ftp server: <hostname>
Enter username: <username>
Password:
***** Transfer of file Completed Successfully *****
Copy complete, now saving to disk (please wait)...
Copy complete.
switch#
```

Étape 3. Remplacez le fichier d'origine « dcos_sshd_config » par le fichier de copie

Processus manuel (non persistant au redémarrage) - Toutes les plates-formes

En remplaçant le fichier existant dcos_sshd_config sous /isan/etc/ par un fichier modifiédcos_sshd_config situé dans le bootflash. Ce processus n'est pas persistant lors des redémarrages

- Téléchargez un fichier modifiéssh config vers bootflash :

```
switch# dir bootflash: | i ssh
7372 Mar 24 02:24:13 2023 dcos_sshd_config_modified
```

2. En mode bash ou Linux(debug)#, remplacez le fichier existantdcos_sshd_config par celui de la commande bootflash:

```
bash-4.3$ sudo su
bash-4.3# copy /bootflash/dcos_sshd_config_modified /isan/etc/dcos_sshd_config
```

3. Confirmez que les modifications ont réussi :

```
bash-4.3$ cat /isan/etc/dcos_sshd_config
```

Processus automatisé - N7K

En utilisant un script EEM qui se déclenche lorsque le journal "VDC_MGR-2-VDC_ONLINE" apparaît après un rechargement. Si l'EEM est déclenché, un script py est exécuté et remplace le fichier existant dcos_sshd_config sous /isan/etc/ par un fichier modifiédcos_sshd_config situé dans bootflash. Cela ne s'applique qu'aux versions de NX-OS qui prennent en charge « feature bash-shell ».

- Téléchargez un fichier de configuration ssh modifié vers bootflash:

```
<#root>
```

```
switch# dir bootflash: | i ssh
      7404    Mar 03 16:10:43 2023
```

dcos_sshd_config_modified_7k

```
switch#
```

2. Créez un script py qui applique les modifications au fichierdcos_sshd_config. Assurez-vous d'enregistrer le fichier avec l'extension "py".

```
<#root>
```

```
#!/usr/bin/env python
import os
os.system("sudo usermod -s /bin/bash root")
os.system("sudo su -c \"cp
```

/bootflash/dcos_sshd_config_modified_7

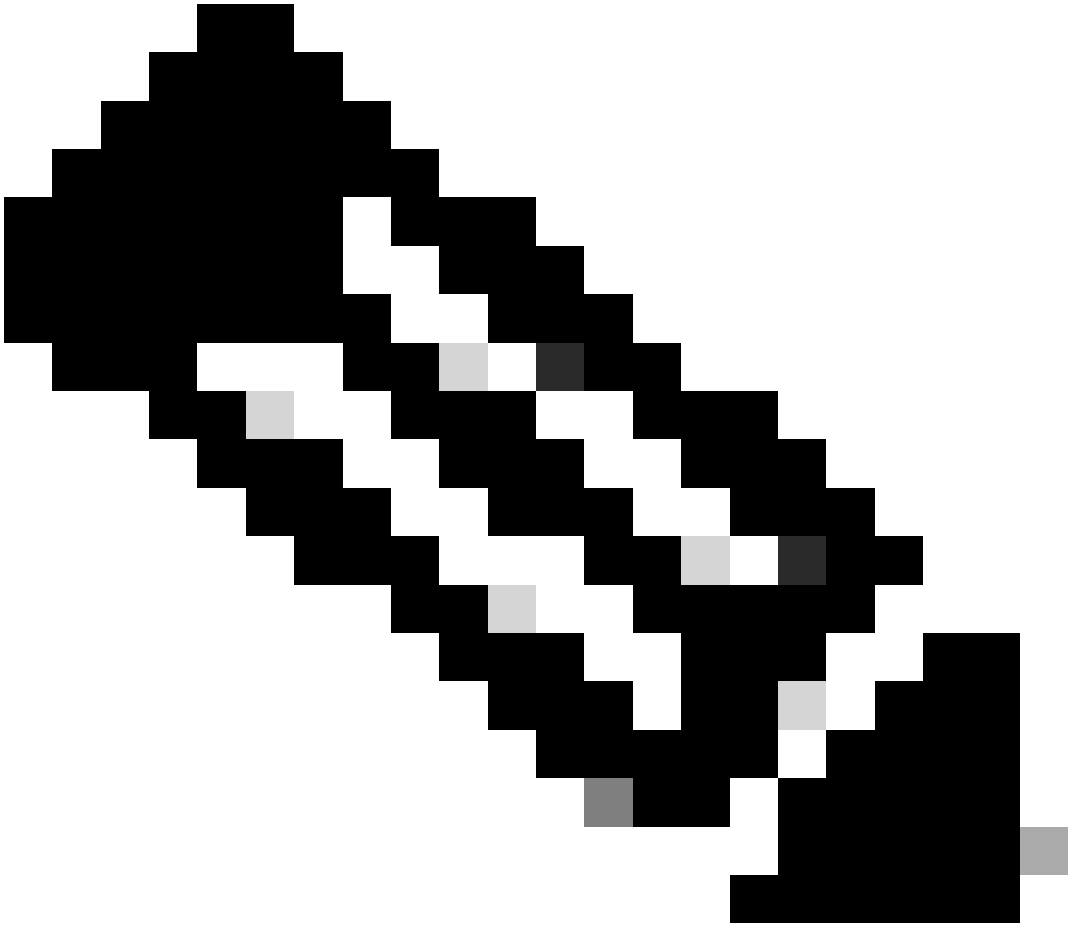
```
k /isan/etc/dcos_sshd_config\"")
```

3. Téléchargez le script Python vers bootflash.

```
<#root>
```

```
switch# dir bootflash:///scripts
      175    Mar 03 16:11:01 2023
```

ssh_workaround_7k.py



Remarque : les scripts Python sont à peu près identiques sur toutes les plates-formes, à l'exception de N7K qui contient quelques lignes supplémentaires pour surmonter le bogue Cisco ayant l'ID CSCva14865.

4. Assurez-vous que le nom de fichierdcos_sshd_config du script et le bootflash (Étape 1) sont identiques :

```
<#root>

switch# dir bootflash: | i ssh
      7404    Mar 03 16:10:43 2023

dcos_sshd_config_modified_7k


switch#
```

<#root>

switch# show file bootflash:///

scripts/ssh_workaround_7k.py

```
#!/usr/bin/env python
import os
os.system("sudo usermod -s /bin/bash root")
os.system("sudo su -c \"cp /
```

bootflash/dcos_sshd_config_modified_7k

```
 /isan/etc/dcos_sshd_config\"")
switch#
```

4. Exécutez le script une fois, afin que le fichier soitdcos_sshd_config modifié.

<#root>

switch#

source ssh_workaround_7k.py

```
switch#
```

5. Configurez un script EEM, de sorte que le script py soit exécuté chaque fois que le commutateur est redémarré et redémarre.

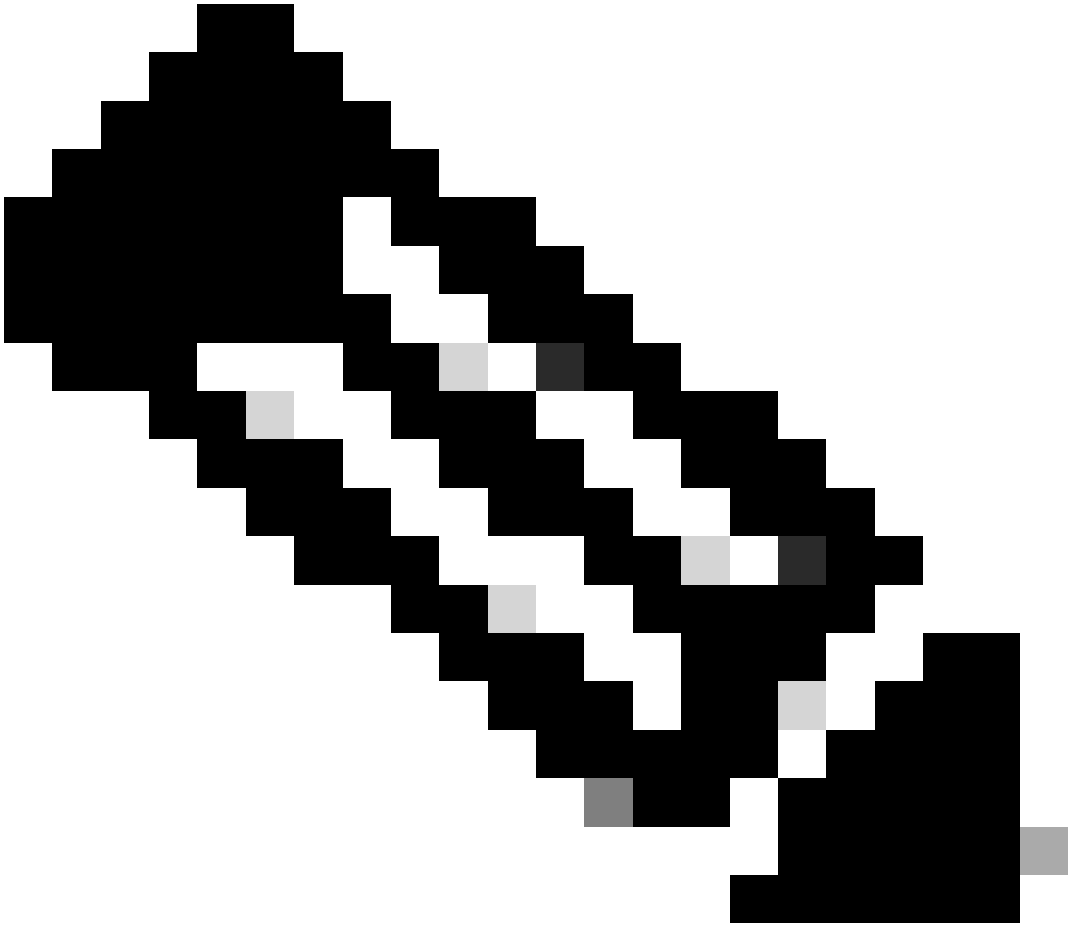
EEM N7K :

<#root>

event manager applet SSH_workaround
  event syslog pattern "vdc 1 has come online"
  action 1.0 cli command

"source ssh_workaround_7k.py"

```
  action 2 syslog priority alerts msg "SSH Workaround implemented"
```



Remarque : la syntaxe EEM peut varier selon les versions de NXOS (certaines versions nécessitent « action <id> cli » et d'autres « action <id> cli command »). Assurez-vous donc que les commandes EEM sont exécutées correctement.

Processus automatisé - N9K, N3K

- Téléchargez un fichier de configuration SSH modifié vers bootflash.

<#root>

switch# dir | i i ssh

7732 Jun 18 16:49:47 2024 dcos_sshd_config
7714 Jun 18 16:54:20 2024

dcos_sshd_config_modified

switch#

2. Créez un script py qui applique les modifications au fichierdcos_sshd_config. Assurez-vous d'enregistrer le fichier avec l'extension "py".

<#root>

#!/usr/bin/env python
import os
os.system("sudo su -c \"cp

/bootflash/dcos_sshd_config_modified

/isan/etc/dcos_sshd_config\"")

3. Téléchargez le script python vers bootflash.

<#root>

switch# dir | i i .py
127 Jun 18 17:21:39 2024

ssh_workaround_9k.py

switch#

4. Assurez-vous que le nom de fichierdcos_sshd_config du script et du bootflash (Étape 1) sont identiques :

<#root>

switch# dir | i i ssh
7732 Jun 18 16:49:47 2024 dcos_sshd_config
7714 Jun 18 16:54:20 2024

dcos_sshd_config_modified

127 Jun 18 17:21:39 2024 ssh_workaround_9k.py
switch#

<#root>

switch# sh file bootflash:ssh_workaround_9k.py

#!/usr/bin/env python
import os
os.system("sudo su -c \"cp

/bootflash/dcos_sshd_config_modified

 /isan/etc/dcos_sshd_config\"")
switch#

4. Exécutez le script une fois, afin que le fichier soitdcos_sshd_config modifié.

<#root>

switch#

python bootflash:ssh_workaround_9k.py

5. Configurez un script EEM, de sorte que le script py soit exécuté chaque fois que le commutateur est redémarré et redémarre.

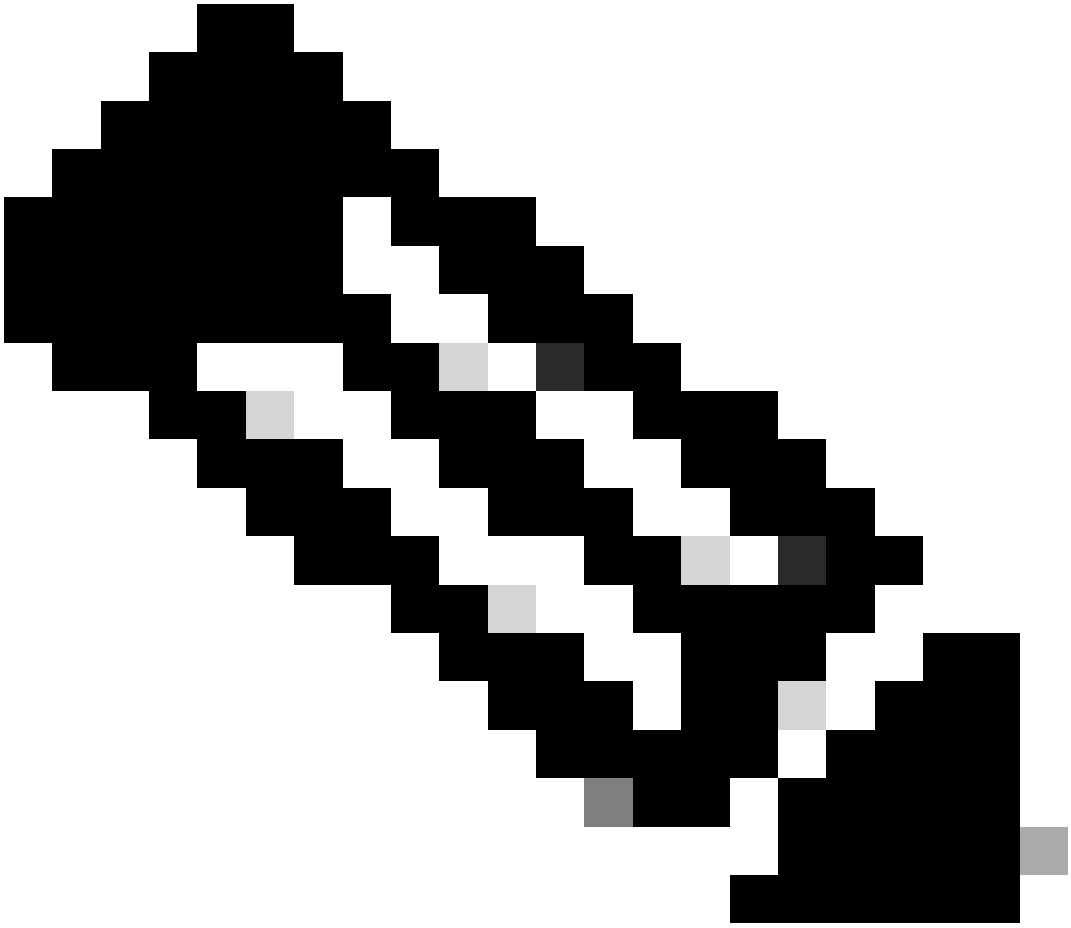
EEM N9K et N3K :

<#root>

event manager applet SSH_workaround
 event syslog pattern "vdc 1 has come online"
 action 1.0 cli

python bootflash:ssh_workaround_9k.py

  action 2 syslog priority alerts msg SSH Workaround implemented



Remarque : la syntaxe EEM peut varier selon les versions de NXOS (certaines versions nécessitent « action <id> cli » et d'autres « action <id> cli command »). Assurez-vous donc que les commandes EEM sont exécutées correctement.

Processus automatisé - N5K, N6K

Un fichier dplug modifié a été créé via l'ID de bogue Cisco CSCvr23488 pour supprimer ces algorithmes Kex :

- diffie-hellman-group-exchange-sha256
- diffie-hellman-group-exchange-sha1

- diffie-hellman-group1-sha1

Les fichiers de débogage fournis via l'ID de bogue Cisco CSCvr23488 ne sont pas les mêmes que ceux qui sont utilisés pour accéder à l'interpréteur de commandes Linux. Ouvrez un dossier TAC pour obtenir le fichier dplug modifié à partir de l'ID de bogue Cisco CSCvr23488.

- Vérifiez les paramètres par défautdcos_sshd_config :

<#root>

C:\Users\user>ssh -vvv admin@<hostname>
---- snipped ----
debug2: peer server KEXINIT proposal
debug2:

KEX algorithms: ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-

<--- kex algorithms
debug2:

host key algorithms: ssh-rsa

debug2: ciphers ctos: aes128-ctr,aes192-ctr,aes256-ctr
debug2:

ciphers stoc: aes128-ctr,aes192-ctr,aes256-ctr

<--- encryption algorithms
debug2: MACs ctos: hmac-sha1
debug2:

MACs stoc: hmac-sha1

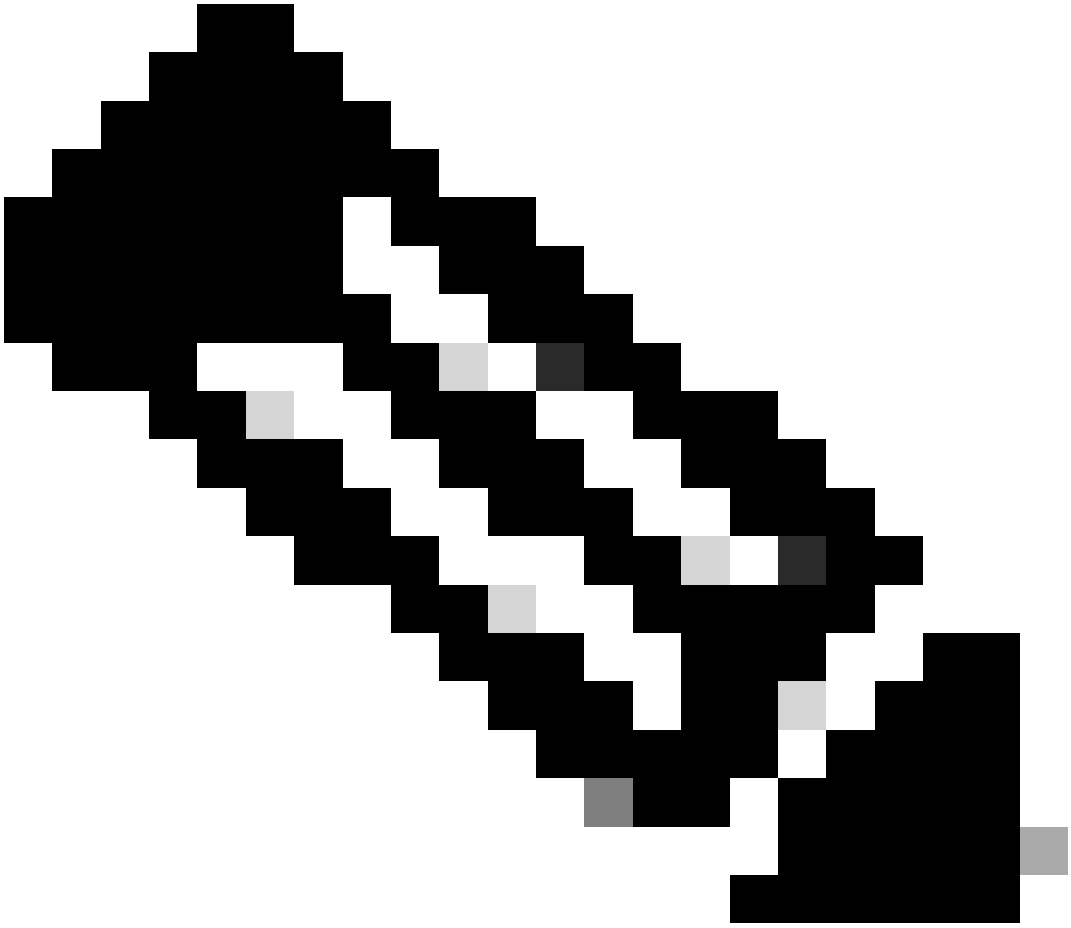
<--- mac algorithms
debug2: compression ctos: none,zlib@openssh.com
debug2:

compression stoc: none,zlib@openssh.com

<--- compression algorithms

2. Créez une copie du fichier dplug modifié.

switch# copy bootflash:nuova-or-dplug-mzg.7.3.14.N1.1_CSCvr23488.bin bootflash:dp



Remarque : une copie ("dp") du fichier dplug d'origine est créée dans bootflash de sorte que seule la copie est supprimée après le chargement de dplug et que le fichier dplug d'origine reste dans bootflash pour les exécutions suivantes.

---

3. Appliquez manuellement le fichier dplug à partir de l'ID de bogue Cisco CSCvr23488 :

```
switch# load bootflash:dp2
Loading plugin version 7.3(14)N1(1)
################################################################
  Warning: debug-plugin is for engineering internal use only!
  For security reason, plugin image has been deleted.
################################################################
Successfully loaded debug-plugin!!!
```

Workaround for CSCvr23488 implemented
switch#

4. Vérifiez les nouveaux dcos_sshd_config paramètres :

<#root>

C:\Users\user>ssh -vvv admin@<hostname>
    ---- snipped ----
debug2: peer server KEXINIT proposal
debug2:

KEX algorithms: diffie-hellman-group14-sha1,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521

debug2: host key algorithms: ssh-rsa
debug2: ciphers ctos: aes128-ctr,aes192-ctr,aes256-ctr
debug2:

ciphers stoc: aes128-ctr,aes192-ctr,aes256-ctr

debug2: MACs ctos: hmac-sha1
debug2:

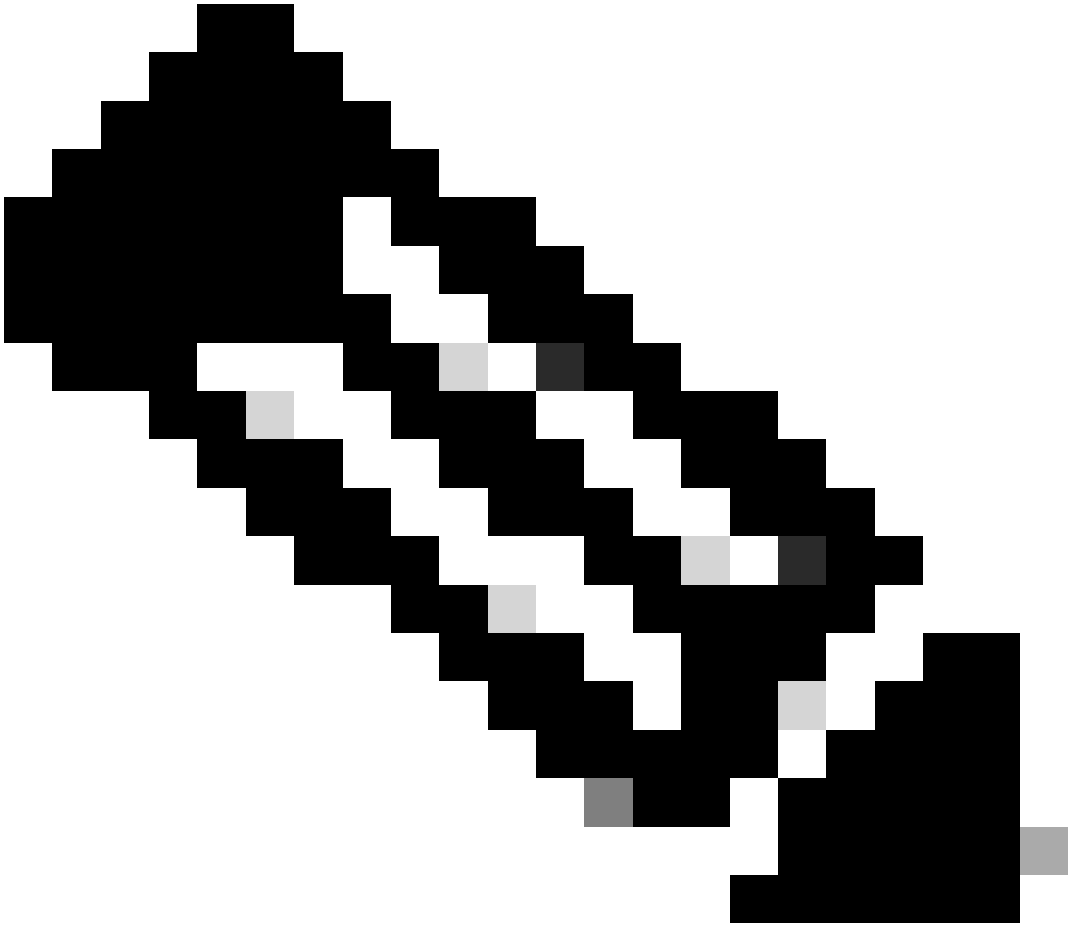
MACs stoc: hmac-sha1

debug2: compression ctos: none,zlib@openssh.com
debug2:

compression stoc: none,zlib@openssh.com

5. Rendez cette modification persistante au cours des redémarrages avec un script EEM :

event manager applet CSCvr23488_workaround
event syslog pattern "VDC_MGR-2-VDC_ONLINE"
action 1 cli command "copy bootflash:nuova-or-dplug-mzg.7.3.14.N1.1_CSCvr23488.bin bootflash:dp"
action 2 cli command "load bootflash:dp"
action 3 cli command "conf t ; no feature ssh ;feature ssh"
action 4 syslog priority alerts msg "CSCvr23488 Workaround implemented"



Remarque :

- Une fois le dplug modifié appliqué, la fonctionnalité SSH doit être réinitialisée sur cette plate-forme.
- Assurez-vous que le fichier dplug est présent dans bootflash et que l'EEM est configuré avec le nom de fichier dplug approprié. Le nom de fichier dplug peut varier en fonction de la version du commutateur. Veillez donc à modifier le script selon vos besoins.
- L'action 1 crée une copie du fichier dplug d'origine dans le bootflash vers un autre fichier appelé « dp », de sorte que le fichier dplug d'origine n'est pas supprimé après avoir été chargé.

Considérations relatives aux plates-formes

N5K/N6K

- Il est impossible de modifier le code d'authentification de message MAC sur ces plates-formes en modifiant le fichier dcos_sshd_config. Le seul MAC pris en charge est hmac-sha1.

N7K

- Pour modifier les adresses MAC, un code 8.4 est requis. Consultez l'ID de bogue Cisco CSCwc26065pour plus de détails.
- "Sudo su" n'est pas disponible par défaut sur 8.X. ID de bogue Cisco de référence : CSCva14865. Si elle est exécutée, cette erreur est observée :

<#root>

F241.06.24-N7706-1(config)# feature bash-shell
F241.06.24-N7706-1(config)# run bash
bash-4.3$ sudo su

Cannot execute /isanboot/bin/nobash: No such file or directory <---

bash-4.3$

Pour remédier à ce problème, tapez :

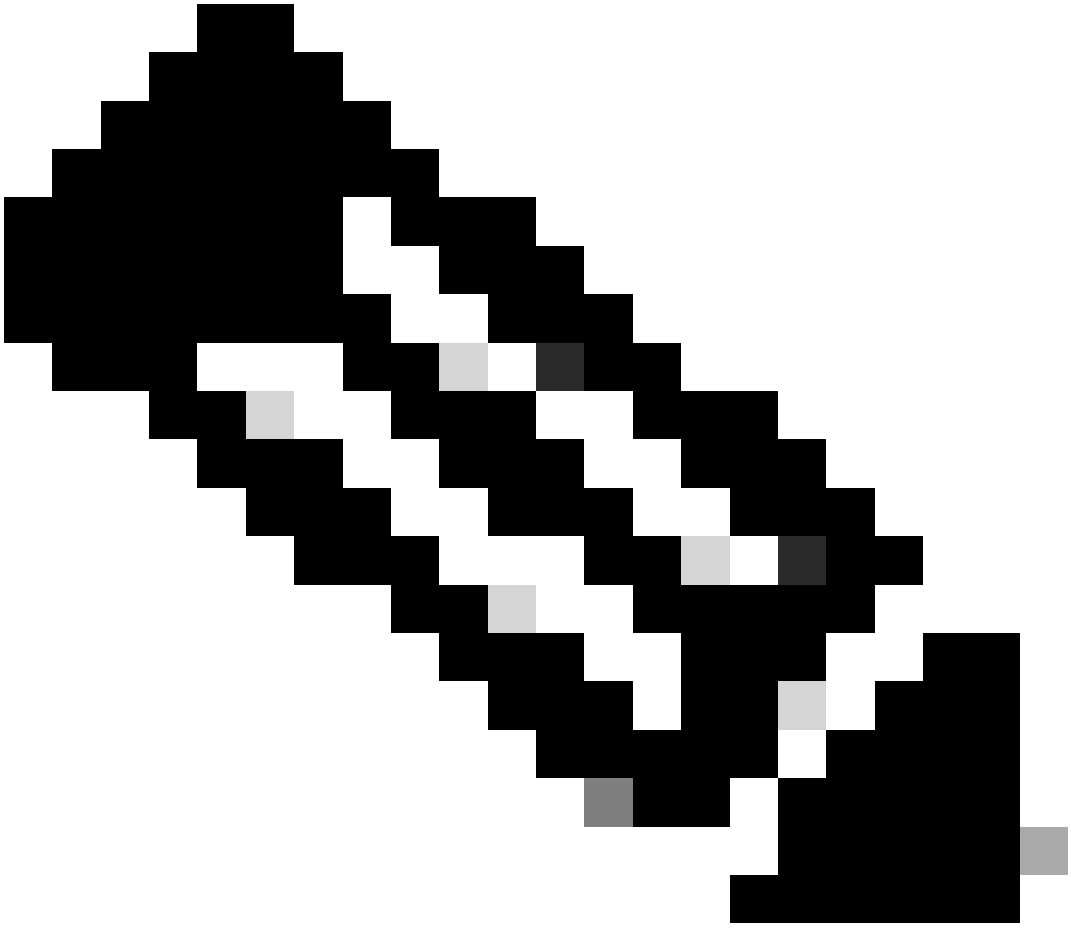
<#root>

bash-4.3$

sudo usermod -s /bin/bash root

Après ce "sudo su" fonctionne :

bash-4.3$ sudo su
bash-4.3#



Remarque : cette modification ne survit pas à un rechargement.

- Il y a un fichier séparé dcos_sshd_config pour chaque VDC, si les paramètres SSH doivent être modifiés sur un VDC différent, assurez-vous de modifier le fichier correspondantdcos_sshd_config.

<#root>

N7K# run bash
bash-4.3$ cd /isan/etc/
bash-4.3$ ls -la | grep ssh
-rw-rw-r-- 1 root root 7564 Mar 27 13:48

dcos_sshd_config

```
<--- VDC 1
-rw-rw-r-- 1 root root 7555 Mar 27 13:48
```

dcos_sshd_config.2

```
<--- VDC 2
-rw-rw-r-- 1 root root 7555 Mar 27 13:48
```

dcos_sshd_config.3

```
<--- VDC 3
```

N9K

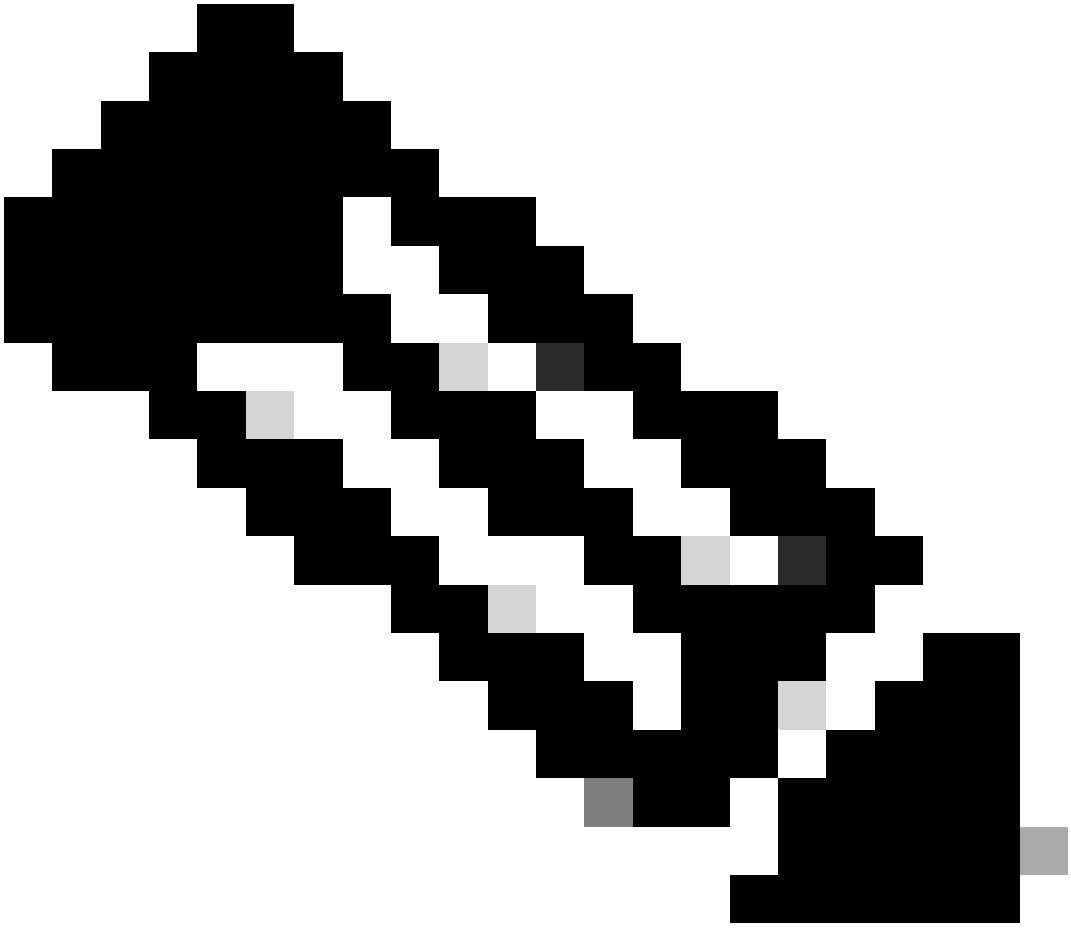
- Les modifications apportées au fichierdcos_sshd_config ne sont pas conservées lors des redémarrages sur les plates-formes Nexus. Si les modifications doivent être persistantes, un EEM peut être utilisé pour modifier le fichier à chaque démarrage du commutateur. L'amélioration apportée à N9K modifie cette configuration à partir de 10.4. Consultez l'ID de bogue Cisco CSCwd82985pour plus de détails.

N7K, N9K, N3K

Des algorithmes de chiffrement, MAC et KexAlgorithms supplémentaires peuvent être ajoutés si nécessaire :

<#root>

switch(config)# ssh kexalgos [all | key-exchangealgorithm-name]
switch(config)# ssh macs [all | mac-name]
switch(config)# ssh ciphers [all | cipher-name]



Remarque : ces commandes sont disponibles sur le Nexus 7000 avec les versions 8.3(1) et ultérieures. Pour la plate-forme Nexus 3000/9000, la commande devient disponible avec la version 7.0(3)I7(8) et ultérieure. (Toutes les versions 9.3(x) ont également cette commande. Voir Guide de configuration de la sécurité NX-OS de la gamme Cisco Nexus 9000, version 9.3(x))

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.