Nexusプラットフォームでの暗号、MAC、Kexアルゴリズムの設定

内容

はじめに

このドキュメントでは、Nexusプラットフォームで暗号、MAC、およびKexアルゴリズムを追加（または）削除する手順について説明します。

前提条件

要件

LinuxとBashの基本を理解しておくことをお勧めします。

使用するコンポーネント

このドキュメントの情報は、次のハードウェアとソフトウェアのバージョンに基づいています。

- Nexus 3000および9000 NX-OS 7.0(3)I7(10)
- Nexus 3000および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)

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

セキュリティスキャンで、Nexusデバイスで使用される脆弱な暗号化方式を検出できる場合があります。この場合、これらの安全でないアルゴリズムを削除するには、スイッチ上の dcos_sshd_configファイルに対する変更が必要になります。

使用可能な暗号、MAC、およびKexアルゴリズムの確認

プラットフォームで使用されている暗号、MAC、およびKexアルゴリズムを確認し、これを外部デバイスから確認するには、次のオプションを使用できます。

オプション 1PCからのCMD行の使用

Nexusデバイスに到達できるPCでCMD行を開き、コマンド ssh -vvv <hostname>を使用します。

<#root>

```
C:\Users\xxxxx>ssh -vvv




--------- 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 algor

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
```

オプション 2Feature Bash-Shellを使用して「dcos_sshd_config」ファイルにアクセスする

これは次の製品に適用されます。

- N3K実行中7.X、9。X、10。X
- すべてのN9Kコード
- 8.2以降を実行するN7K

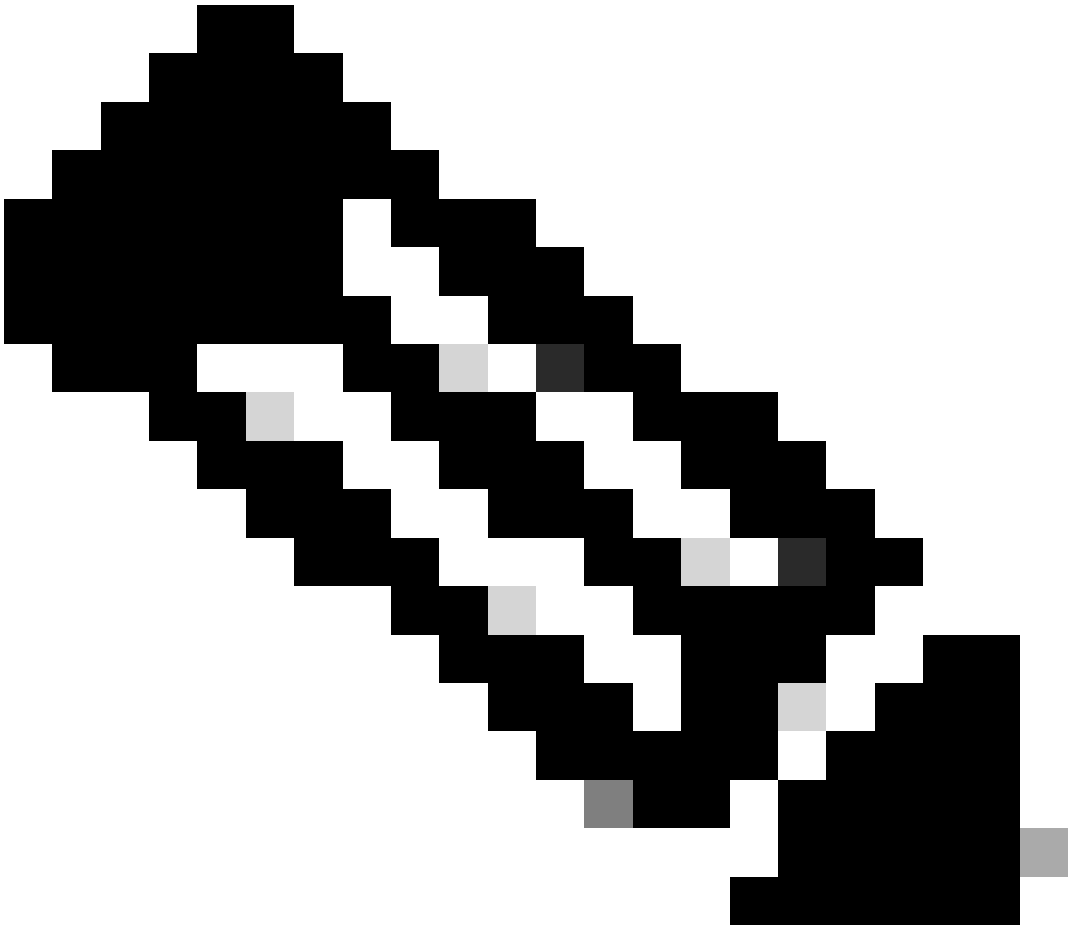
手順：

1. bashシェル機能を有効にして、bashモードに入ります。

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

2. dcos_sshd_configファイルの内容を確認します。

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



注：特定の行を確認するには、egrepを使用できます。cat /isan/etc/dcos_sshd_config | grep MAC

オプション 3Dplugファイルを使用した「dcos_sshd_config」ファイルへのアクセス

これは次の製品に適用されます。

- 6を実行するN3Kbashシェルにアクセスできないx
- すべてのN5KおよびN6Kコード
- 6を実行するN7K。xと7。xコード

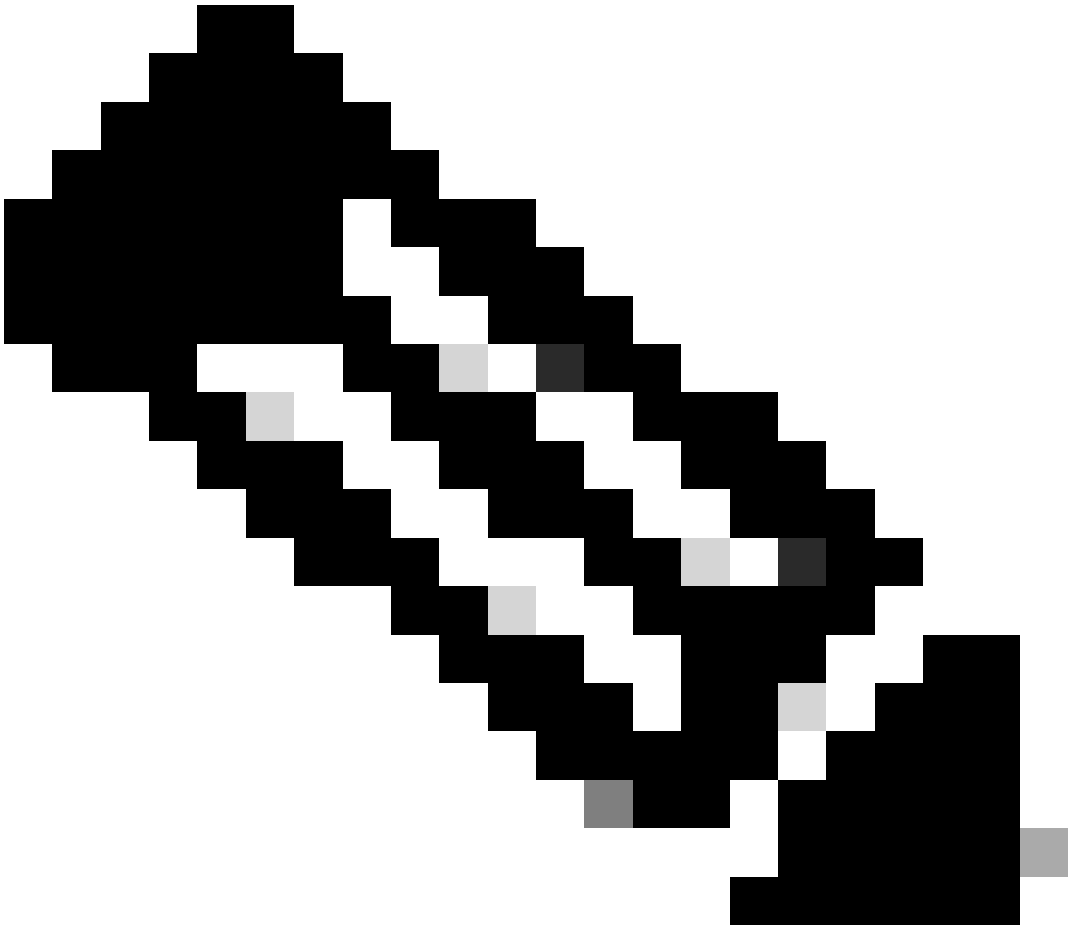
手順：

1. TACケースを開き、スイッチで実行されているNXOSのバージョンと一致するdplugファイルを取得します。

2. dplugファイルをブートフラッシュにアップロードし、そのコピーを作成します。

<#root>

```
switch# copy bootflash:
nuova-or-dplug-mzg.7.3.8.N1.1
 bootflash:
dp
```



注：元のdplugファイルのコピー(「dp」)がブートフラッシュに作成されます。そのため

、dplugがロードされた後はコピーだけが削除され、元のdplugファイルは以降の実行でブートフラッシュに残ります。

3. loadコマンドを使用して、dplugのコピーをロードします。

<#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. dcos_sshd_configファイルを確認します。

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

解決方法

ステップ 1：「dcos_sshd_config」ファイルのエクスポート

1. dcos_sshd_configファイルのコピーをbootflash：に送信します。

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

2. コピーがブートフラッシュにあることを確認します。

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

3. サーバーにエクスポートする：

```
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. ファイルに必要な変更を加え、ブートフラッシュにインポートして戻します。

ステップ 2：「dcos_sshd_config」ファイルのインポート

1. 変更したdcos_sshd_configファイルをブートフラッシュにアップロードします。

```
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#
```

ステップ 3：元の「dcos_sshd_config」ファイルをコピーで置き換える

手動プロセス（リブート後も保持されない）：すべてのプラットフォーム

/isan/etc/の下にある既存のdcos_sshd_configファイルを、ブートフラッシュにある修正済みの dcos_sshd_configファイルに置き換えます。 このプロセスは、リブート後も保持されません

1. 変更したssh configファイルをbootflash：に

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

2. bashまたはLinux(debug)#モードで、既存のdcos_sshd_configファイルをbootflash：にあるファイル

で上書きします。

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

3. 変更が正常に行われたことを確認します。

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

自動プロセス：N7K

リロード後にログ「VDC_MGR-2-VDC_ONLINE」が起動したときにトリガーされるEEMスクリプトを使用するEEMがトリガーされると、pyスクリプトが実行され、/isan/etc/の下にある既存のdcos_sshd_configファイルが、ブートフラッシュ内にある修正されたdcos_sshd_configファイルに置き換えられます。 これは、「機能bash-shell」をサポートするNX-OSバージョンにのみ適用されます。

1. 変更したsshコンフィギュレーションファイルをbootflash：に

<#root>

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

dcos_sshd_config_modified_7k

```
switch#
```

2. dcos_sshd_configファイルに変更を適用するpyスクリプトを作成します。ファイルは「py」拡張子を付けて保存してください。

<#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

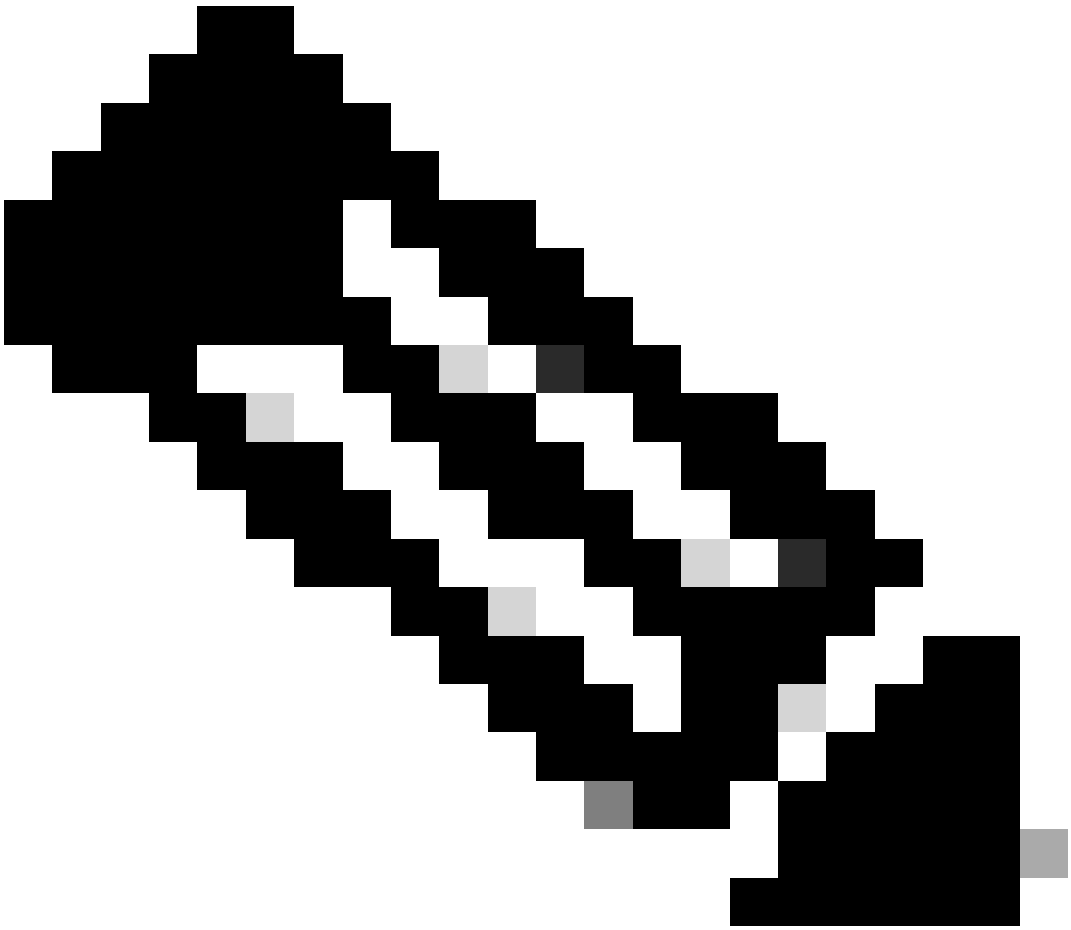
```
k /isan/etc/dcos_sshd_config\"")
```

3. Pythonスクリプトをブートフラッシュにアップロードします。

<#root>

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

ssh_workaround_7k.py



注:Pythonスクリプトはすべてのプラットフォームでほぼ同じですが、N7KにはCisco Bug ID CSCva14865を解決するためのいくつかの追加行が含まれることが異なります。

---

4. スクリプト（ステップ1）とブートフラッシュ（ステップ1）で確認したdcos_sshd_configファイル名が同じであることを確認します。

<#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.dcos_sshd_configファイルが変更されるように、スクリプトを1回実行します。

<#root>

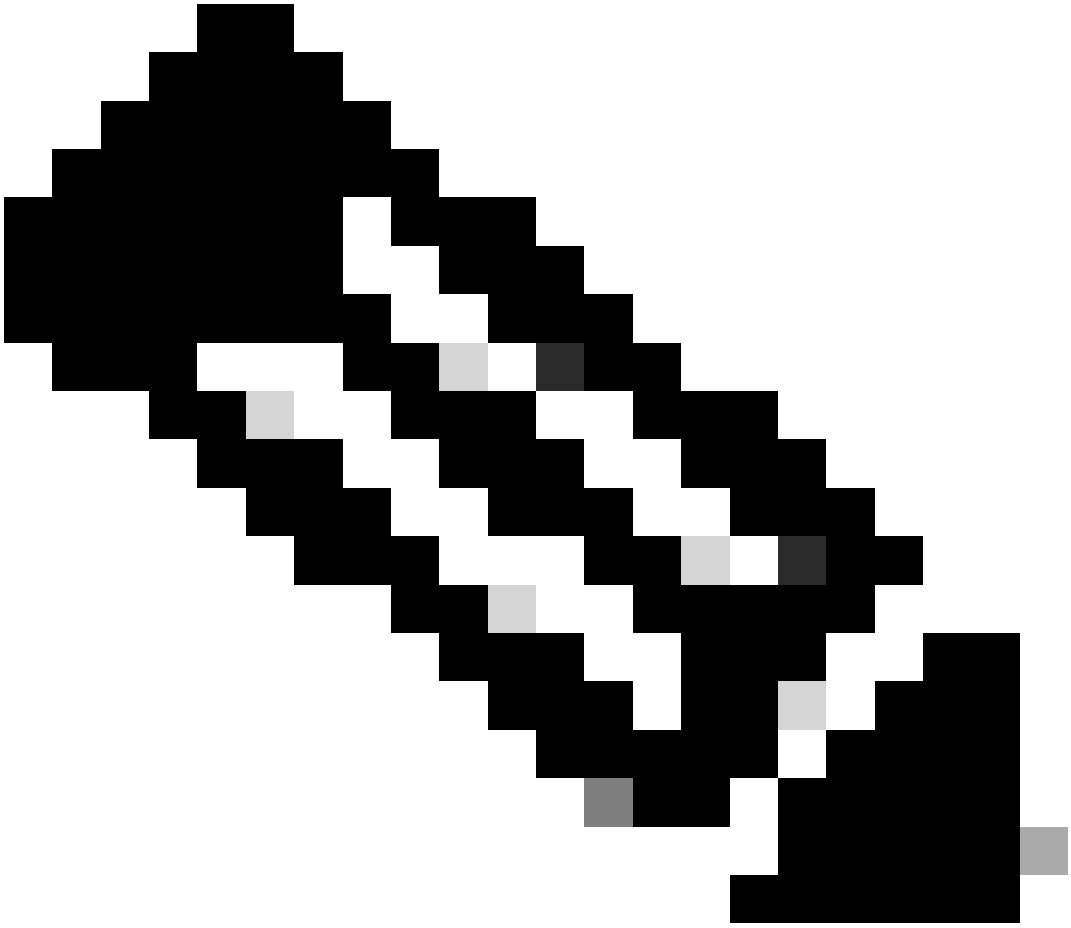
```
switch#
source ssh_workaround_7k.py


switch#
```

5. EEMスクリプトを設定し、スイッチがリブートされて再起動するたびにpyスクリプトが実行されるようにします。

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"
```



注:EEM構文はNXOSのリリースによって異なる可能性があるため（バージョンによっては「action <id> cli」が必要な場合と、他の「action <id> cli command」が必要な場合があります）、EEMコマンドが正しく実行されていることを確認してください。

自動プロセス：N9K、N3K

1. 変更したSSH設定ファイルをブートフラッシュにアップロードします。

<#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. dcos_sshd_configファイルに変更を適用するpyスクリプトを作成します。ファイルは「py」拡張子を付けて保存してください。

<#root>

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

/bootflash/dcos_sshd_config_modified

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

3. Pythonスクリプトをブートフラッシュにアップロードします。

<#root>

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

ssh_workaround_9k.py

```
switch#
```

4. スクリプト（ステップ1）とブートフラッシュ（ステップ1）で確認したdcos_sshd_configファイル名が同じであることを確認します。

<#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.dcos_sshd_configファイルが変更されるように、スクリプトを1回実行します。

<#root>

```
switch#
```

python bootflash:ssh_workaround_9k.py

5. EEMスクリプトを設定し、スイッチがリブートされて再起動するたびにpyスクリプトが実行されるようにします。

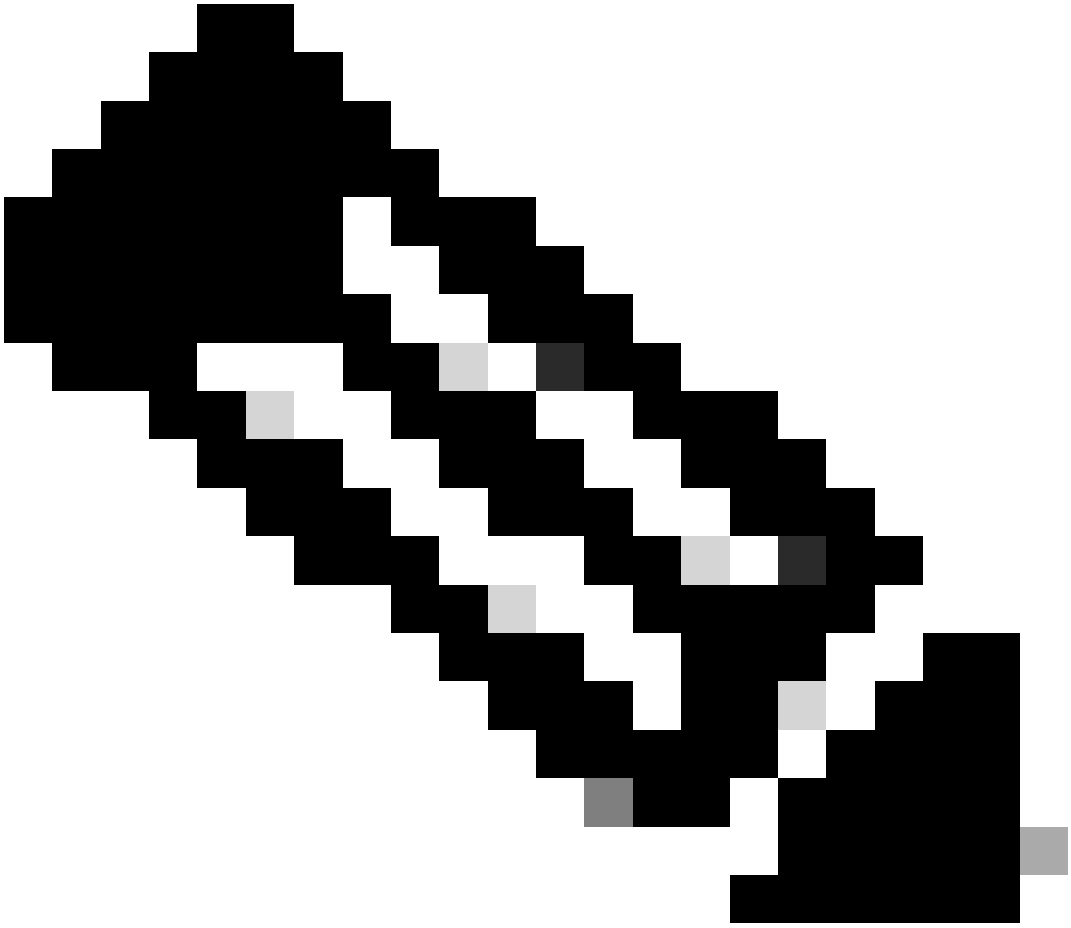
EEM N9KおよびN3K:

<#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
```



注:EEM構文はNXOSのリリースによって異なる可能性があるため（バージョンによっては「action <id> cli」が必要な場合と、他の「action <id> cli command」が必要な場合があります）、EEMコマンドが正しく実行されていることを確認してください。

自動プロセス：N5K、N6K

修正されたdplugファイルは、Cisco Bug ID CSCvr23488で次のKexアルゴリズムを削除するために作成されました。

- diffie-hellman-group-exchange-sha256
- diffie-hellman-group-exchange-sha1
- diffie-hellman-group1-sha1

Cisco Bug ID CSCvr23488によって提供されるdpugファイルは、Linuxシェルへのアクセスに使用されるファイルと同じではありません。TACケースをオープンし、修正されたdplugをCisco Bug ID CSCvr23488から取得します。

1. デフォルトのdcos_sshd_config設定を確認します。

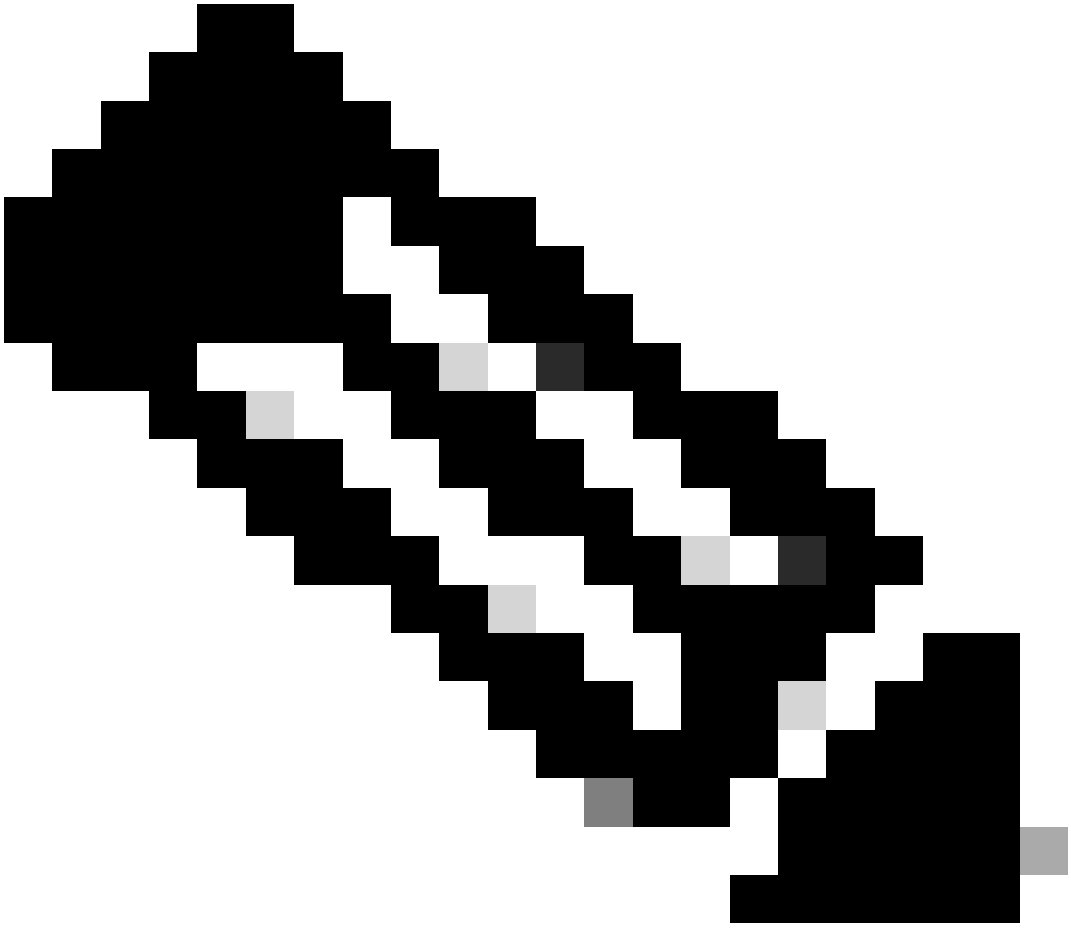
<#root>

```
C:\Users\user>ssh -vvv admin@
```

```
      ---- 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. 変更したdplugファイルのコピーを作成します。

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



注：元のdplugファイルのコピー(「dp」)がブートフラッシュに作成されるので、dplugがロードされた後はコピーだけが削除され、元のdplugファイルは以降の実行でブートフラッシュに残ります。

3. Cisco Bug ID CSCvr23488のdplugファイルを手動で適用します。

```
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#
```

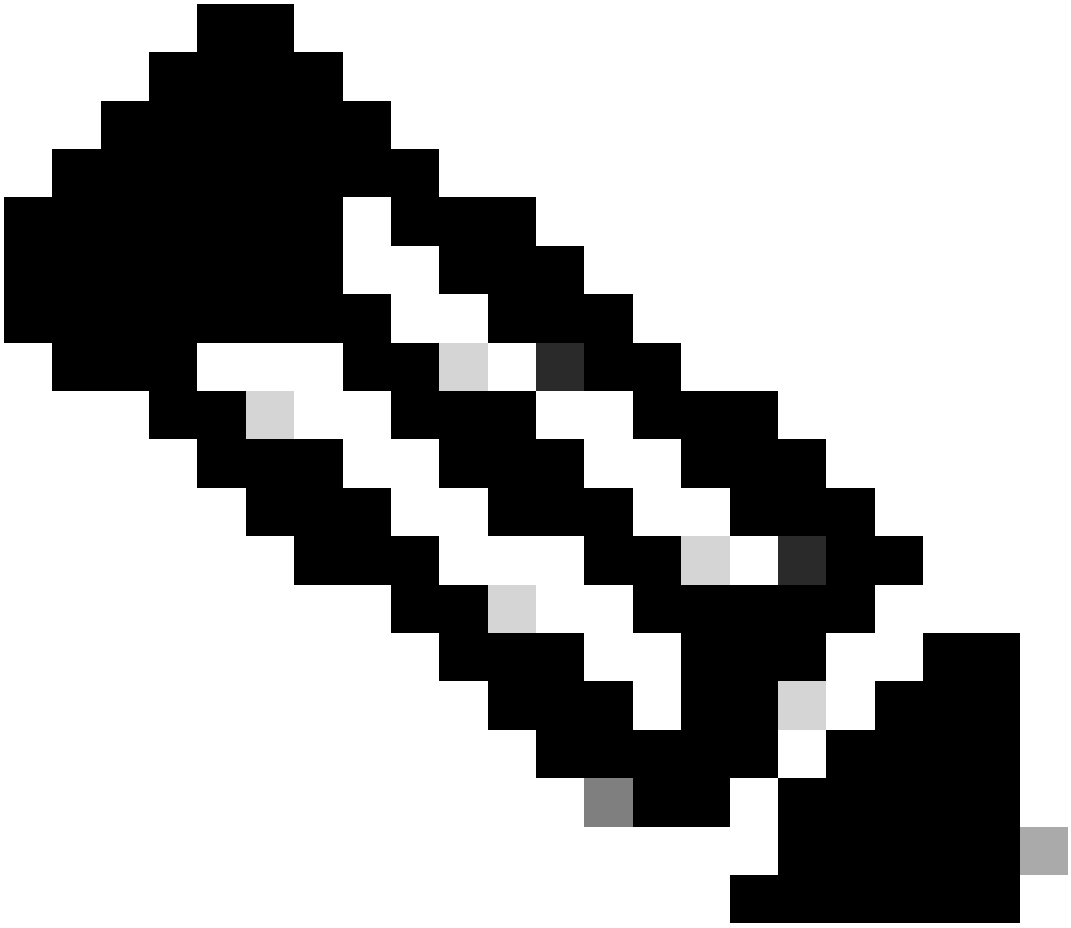
4. 新しいdcos_sshd_config設定を確認します。

<#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. EEMスクリプトを使用して、この変更をリブート後も保持されるようにします。

```
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"
```



注：

- 修正されたdplugが適用された後、このプラットフォームでSSH機能をリセットする必要があります。
- ブートフラッシュにdplugファイルが存在し、EEMが適切なdplugファイル名で設定されていることを確認します。dplugファイル名はスイッチのバージョンによって異なる場合があるため、必要に応じてスクリプトを変更してください。
- アクション1は、ブートフラッシュにある元のdplugファイルのコピーを「dp」と呼ばれる別のファイルに作成します。そのため、元のdplugファイルはロード後に削除されません。

プラットフォームの考慮事項

N5K/N6K

- これらのプラットフォームでは、dcos_sshd_configファイルを変更してもMAC(メッセージ

認証コード)は変更できません。サポートされているMACはhmac-sha1だけです。

N7K

- MACを変更するには、8.4コードが必要です。 詳細については、Cisco Bug ID CSCwc26065を参照してください。
- 「Sudo su」は、デフォルトでは8.Xでは使用できません。Cisco Bug ID CSCva14865を参照してください。実行すると、次のエラーが発生します。

<#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$
```

これを解決するには、次のように入力します。

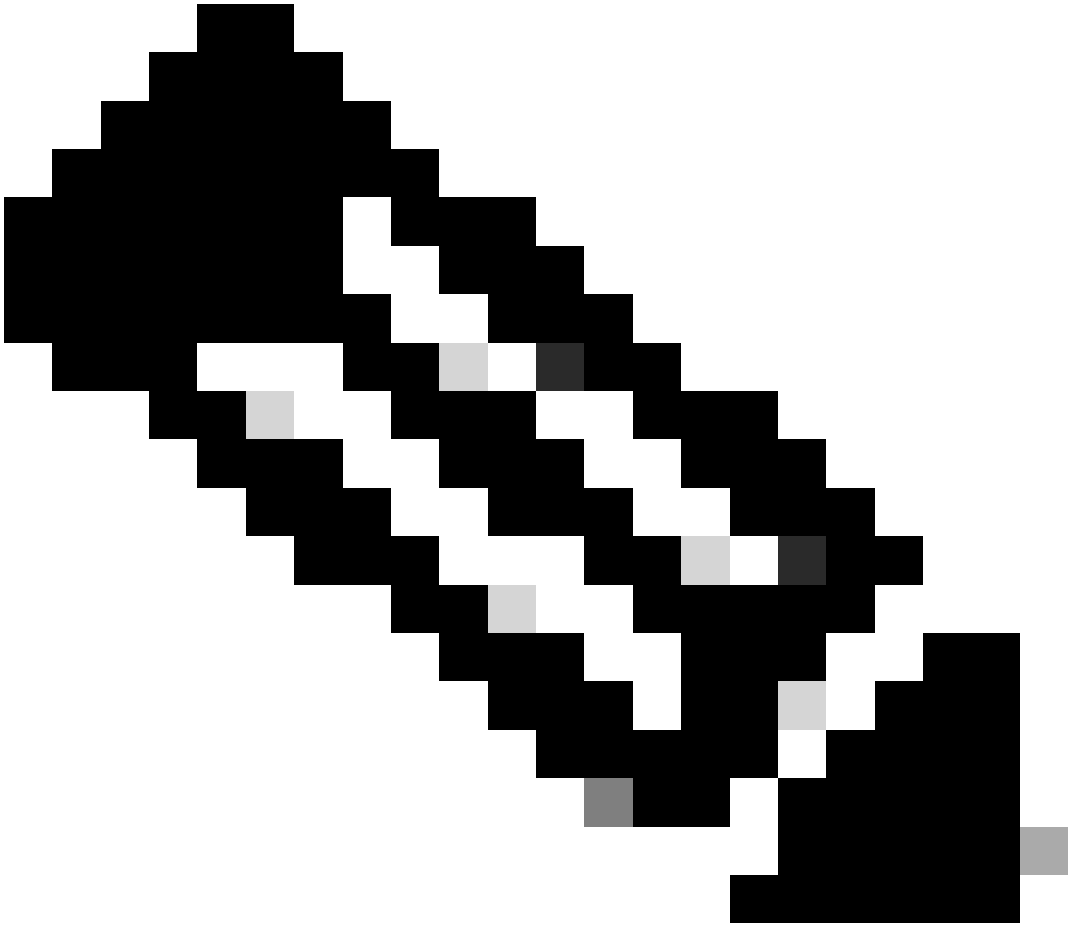
<#root>

```
    bash-4.3$
```

sudo usermod -s /bin/bash root

この「sudo su」が動作した後は、次のようになります。

```
    bash-4.3$ sudo su
    bash-4.3#
```



注：この変更はリロード後も有効です。

- VDCごとに個別のdcos_sshd_configファイルがあります。異なるVDCでSSHパラメータを変更する必要がある場合は、対応するdcos_sshd_configファイルを変更してください。

<#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

- dcos_sshd_configファイルの変更は、どのNexusプラットフォームでもリブート後は保持されません。変更を永続的に行う必要がある場合は、スイッチが起動するたびにEEMを使用してファイルを変更できます。
- 10.4(2)以降では、N9Kの機能拡張によってこの点が変更されています。 10.5(1)でも使用できます。 以前のソフトウェアトレインの新しいバージョンには追加されていません。
- 詳細については、Cisco Bug ID CSCwd82985を参照してください。

10.5(1)を実行するスイッチからのCLIの例：

```
switch(config)# ssh ?
cipher-mode Set Cipher-mode for ssh
ciphers Ciphers to encrypt the connection <<<<<<<<<
idle-timeout SSH Client session idle timeout value
kexalgos Key exchange methods that are used to generate per-connection keys <<<<<<<<<
key Generate SSH Key
keytypes Public key algorithms that the server can use to authenticate itself to the client
login-attempts Set maximum login attempts from ssh
login-gracetime Set login gracetime for ssh connection
macs Message authentication codes used to detect traffic modification <<<<<<<<<
port Set port number for ssh
rekey Renegotiate ssh key

switch(config)# ssh ciphers ?
WORD Algorithm name to be configured (Max Size 128)
aes256-gcm <Deprecated> enable aes256-gcm
all Control known weak SSH algorithms in current version of NX-OS in addition to the base set of strong

switch(config)# ssh macs ?
WORD Algorithm name to be configured (Max Size 128)
all Control known weak SSH algorithms in current version of NX-OS in addition to the base set of strong

switch(config)# ssh kexalgos ?
WORD Algorithm name to be configured (Max Size 128)
all Control known weak SSH algorithms in current version of NX-OS in addition to the base set of strong
```

10.3(6)を実行するスイッチからのCLIの例：

```
switch(config)# ssh kexalgos ?
all Enable algorithms supported in current version of SSH
ecdh-sha2-nistp384 Enable ecdh-sha2-nistp384

switch(config)# ssh ciphers ?
aes256-gcm Enable aes256-gcm
all Enable algorithms supported in current version of SSH
```

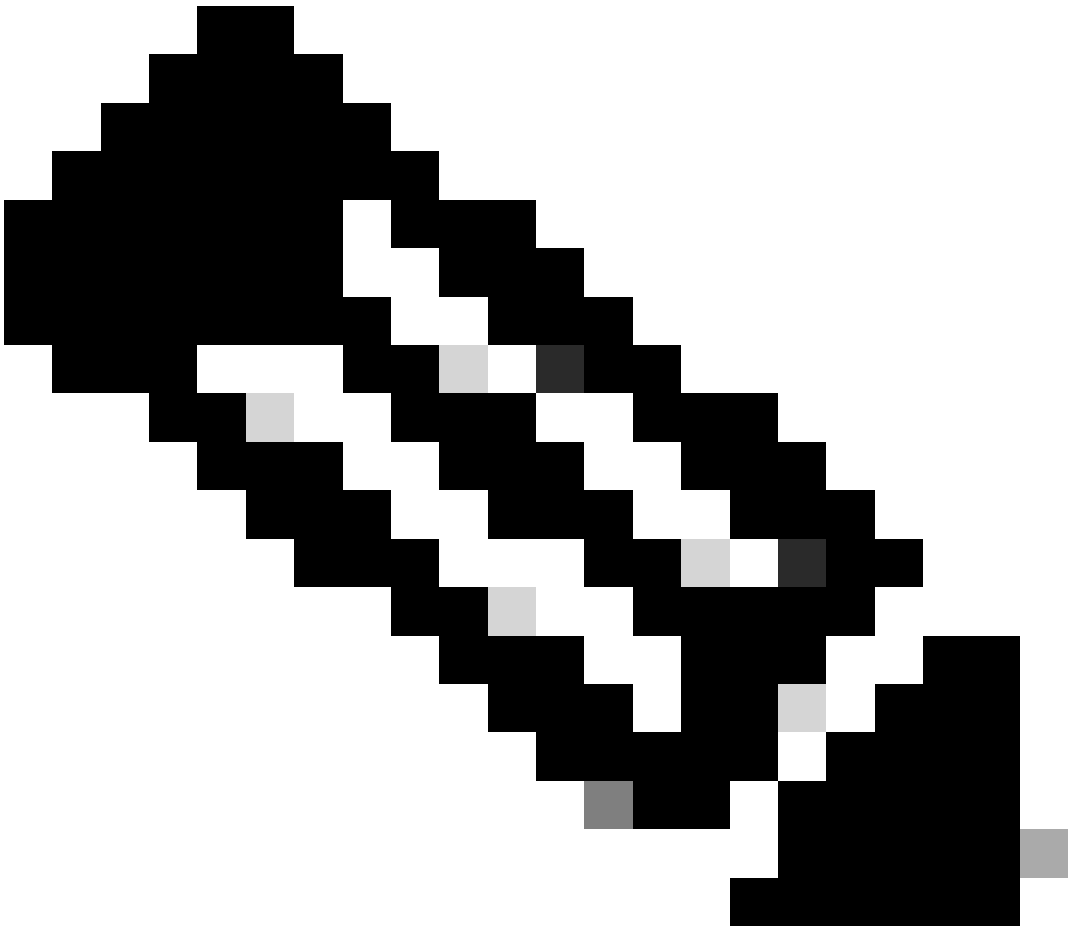
```
switch(config)# ssh macs ?
all Enable algorithms supported in current version of SSH
```

N7K、N9K、N3K

必要に応じて追加できる追加の暗号、MAC、およびKexAlgorithmsがあります。

<#root>

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



注：これらのコマンドは、リリース8.3(1)以降のNexus 7000で使用できます。Nexus 3000/9000プラットフォームでは、リリース7.0(3)I7(8)以降でコマンドが使用可能になります。(すべての9.3(x)リリースにこのコマンドがあります。『Cisco Nexus 9000 Series NX-OS Security Configuration Guide, Release 9.3(x)』を参照してください。

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。