Configure Ciphers, MACs, Kex Algorithms in Nexus Platforms

Contents

Introduction

This document describes the steps to add (or) remove Ciphers, MACs, and Kex Algorithms in Nexus platforms.

Prerequisites

Requirements

Cisco recommends that you understand the basics of Linux and Bash.

Components Used

The information in this document is based on these hardware and software versions:

- Nexus 3000 and 9000 NX-OS 7.0(3)I7(10)
- Nexus 3000 and 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. If your network is live, ensure that you understand the potential impact of any command.

Background Information

Sometimes, security scans can find weak encryption methods used by Nexus devices. If this happens, changes to the dcos_sshd_config file on the switches are required to remove these insecure algorithms.

Review Available Ciphers, MACs, and Kex Algorithms

To confirm what Ciphers, MACs, and Kex Algorithms a platform uses and check this from an external device you can use these options:

Option 1. Using CMD Line from PC

Open a CMD line on a PC that can reach the Nexus device and use the command 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 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
```

Option 2. Access the "dcos_sshd_config" File by Using Feature Bash-Shell

This applies to:

- N3K running 7. X, 9. X, 10. X

- All N9K codes
- N7K running 8.2 and later

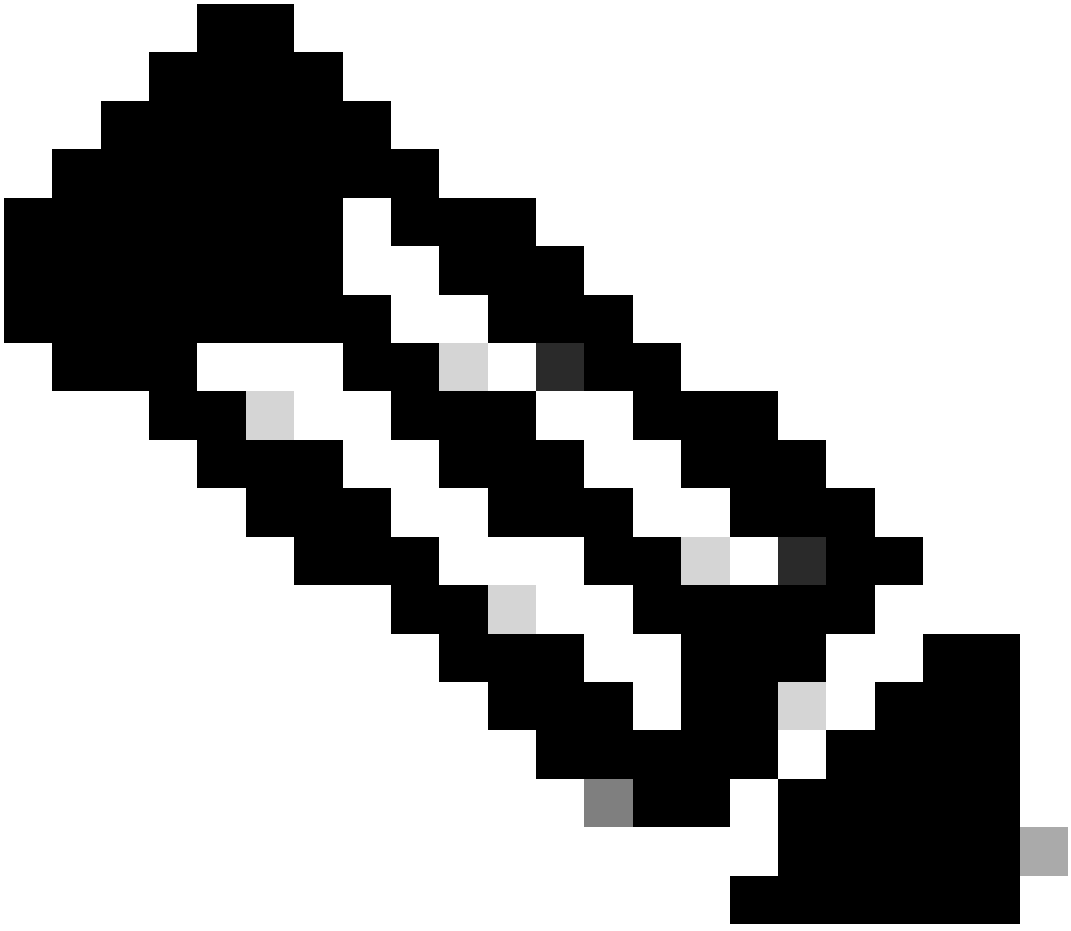
Steps:

1. Enable the bash-shell feature and get into bash mode:

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

2. Review the contents from the `dcos_sshd_config` file:

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



Note: You can use egrep to look at specific lines: cat /isan/etc/dcos_sshd_config | grep MAC

Option 3. Access the "dcos_sshd_config" File by Using a Dplug File

This applies to:

- N3Ks running 6. X that does not have access to the bash-shell
- All N5K and N6K codes
- N7Ks running 6. X and 7. X codes

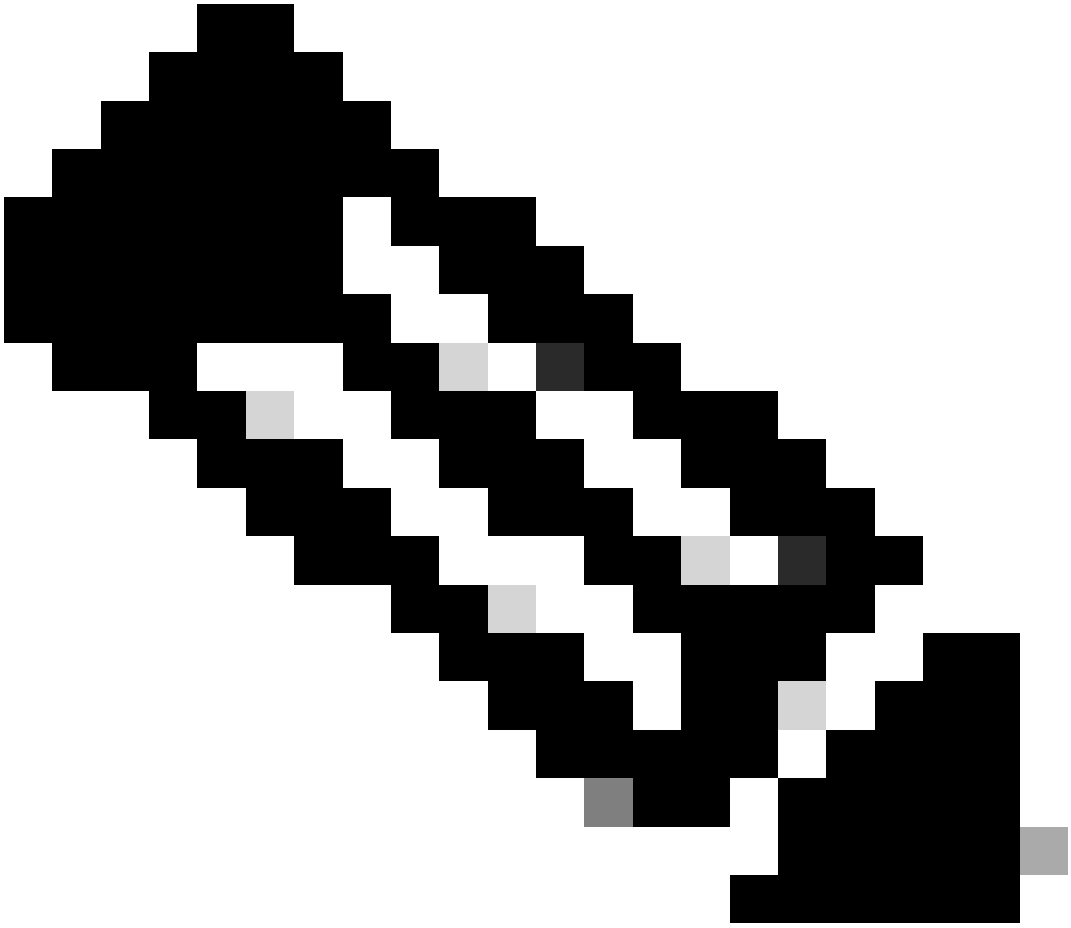
Steps:

1. Open a TAC case to obtain the dplug file that matches the NXOS version running on the switch.

2. Upload the dplug file to bootflash and create a copy of it.

<#root>

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



Note: A copy ("dp") of the original dplug file is created in bootflash, so that only the copy gets removed after the dplug is loaded and the original dplug file remains in bootflash for subsequent runs.

---

3. Load the copy of the dplug via the load command.

<#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. Review dcos_sshd_config file.

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

Solution

Step 1. Export the "dcos_sshd_config" File

1. Send a copy of the dcos_sshd_config file to bootflash:

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

2. Confirm the copy is on bootflash:

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

3. Export to a server:

```
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. Make the necessary changes to the file and import back to bootflash.

Step 2. Import the "dcos_sshd_config" File

1. Upload the modified dcos_sshd_config file to boot 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#
```

Step 3. Replace the Original "dcos_sshd_config" File with the Copy

Manual Process (Not Persistent Across Reboots) - All platforms

By replacing the existing dcos_sshd_config file under /isan/etc/ with a modified dcos_sshd_config file located in bootflash. This process is not persistent across reboots

1. Upload a modified ssh config file to bootflash:

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

2. While in bash or Linux(debug)# mode, overwrite the existing dcos_sshd_config file with the one in bootflash:

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

3. Confirm the changes were successful:

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

Automated Process - N7K

By using an EEM script that gets triggered when the log "VDC_MGR-2-VDC_ONLINE" comes up after a reload. If the EEM is triggered, a py script is run and replaces the existing dcos_sshd_config file under /isan/etc/ with a modified dcos_sshd_config file located in bootflash. This only applies to NX-OS versions that support "feature bash-shell".

1. Upload a modified ssh config file to bootflash:

<#root>

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

dcos_sshd_config_modified_7k

```
switch#
```

2. Create a py script that applies changes to the dcos_sshd_config file. Ensure to save the file with "py" extension.

<#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

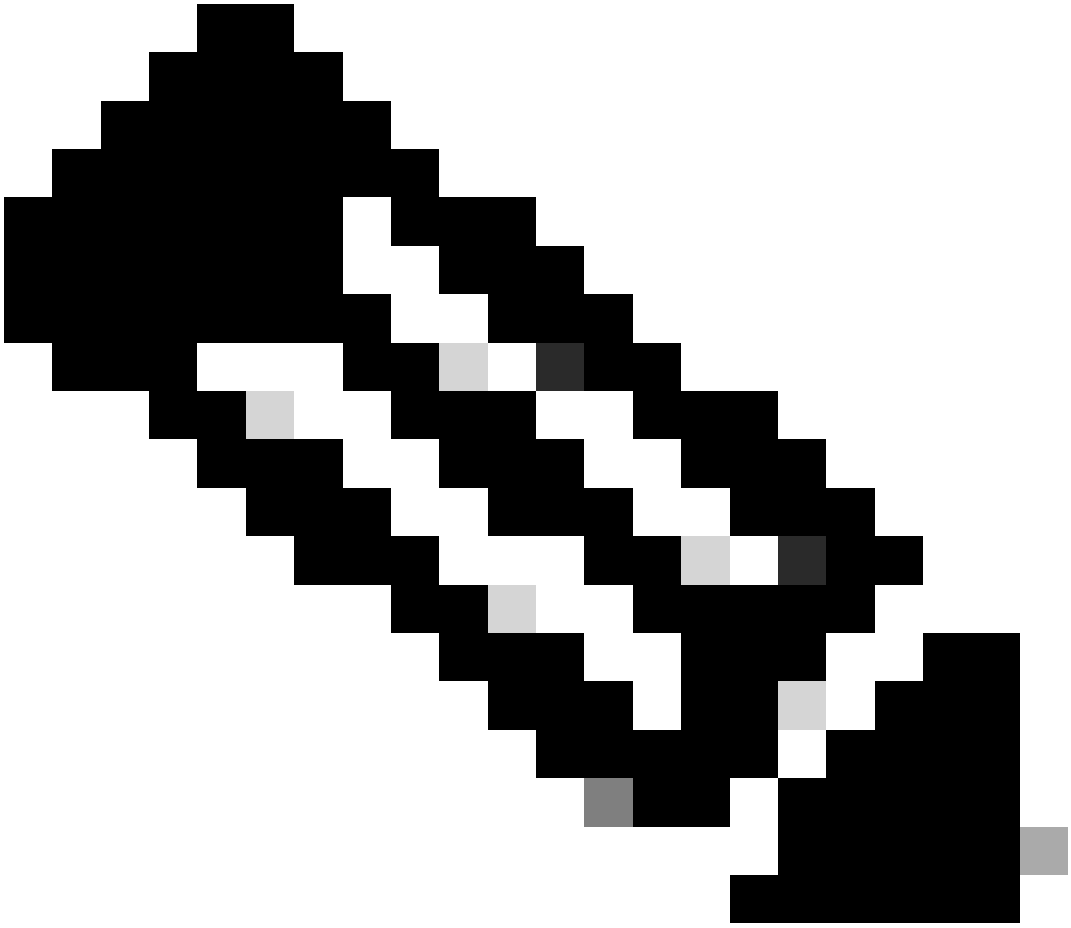
```
k /isan/etc/dcos_sshd_config\"")
```

3. Upload the Python script to bootflash.

<#root>

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

ssh_workaround_7k.py



Note: Python scripts are pretty much the same on all platforms, except for N7K which contains some additional lines to overcome Cisco bug ID CSCva14865.

4. Ensure the dcos_sshd_config file name from the script and bootflash (Step 1.) are the same:

<#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. Run the script once, so that the dcos_sshd_config file is changed.

<#root>

```
switch#
```

source ssh_workaround_7k.py

```
switch#
```

5. Configure an EEM script, so that the py script is run every time the switch is rebooted and comes back up.

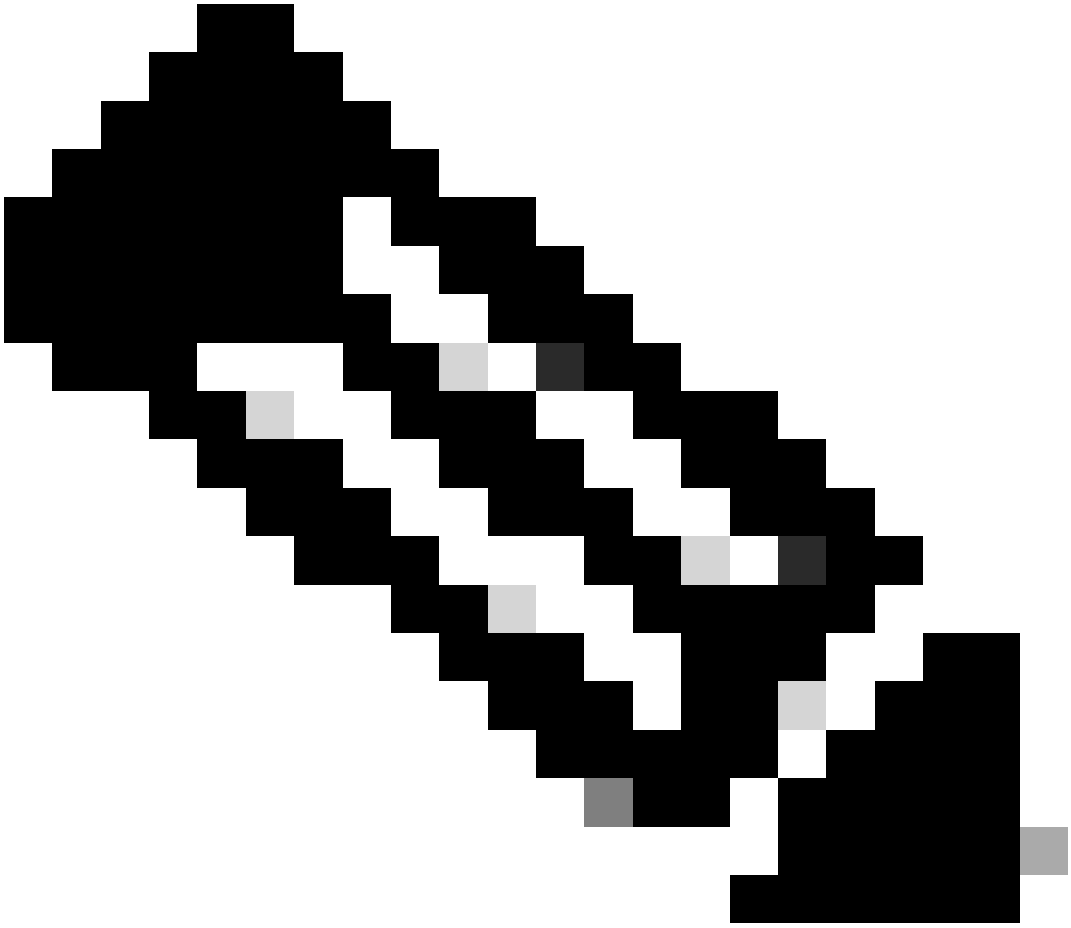
EEM N7K:

<#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"
```



Note: EEM syntax can vary on different NXOS releases (some versions require "action <id> cli" and others "action <id> cli command"), so ensure to check that the EEM commands are taken properly.

Automated Process - N9K, N3K

1. Upload a modified SSH config file to 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. Create a py script that applies changes to the dcos_sshd_config file. Ensure to save the file with the "py" extension.

<#root>

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

/bootflash/dcos_sshd_config_modified

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

3. Upload the python script to bootflash.

<#root>

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

ssh_workaround_9k.py

```
switch#
```

4. Ensure the dcos_sshd_config file name from the script and from bootflash (Step 1.) are the same:

<#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. Run the script once, so that the dcos_sshd_config file is changed.

<#root>

```
switch#
```

python bootflash:ssh_workaround_9k.py

5. Configure an EEM script, so that the py script is ran everytime the switch is rebooted and comes back up.

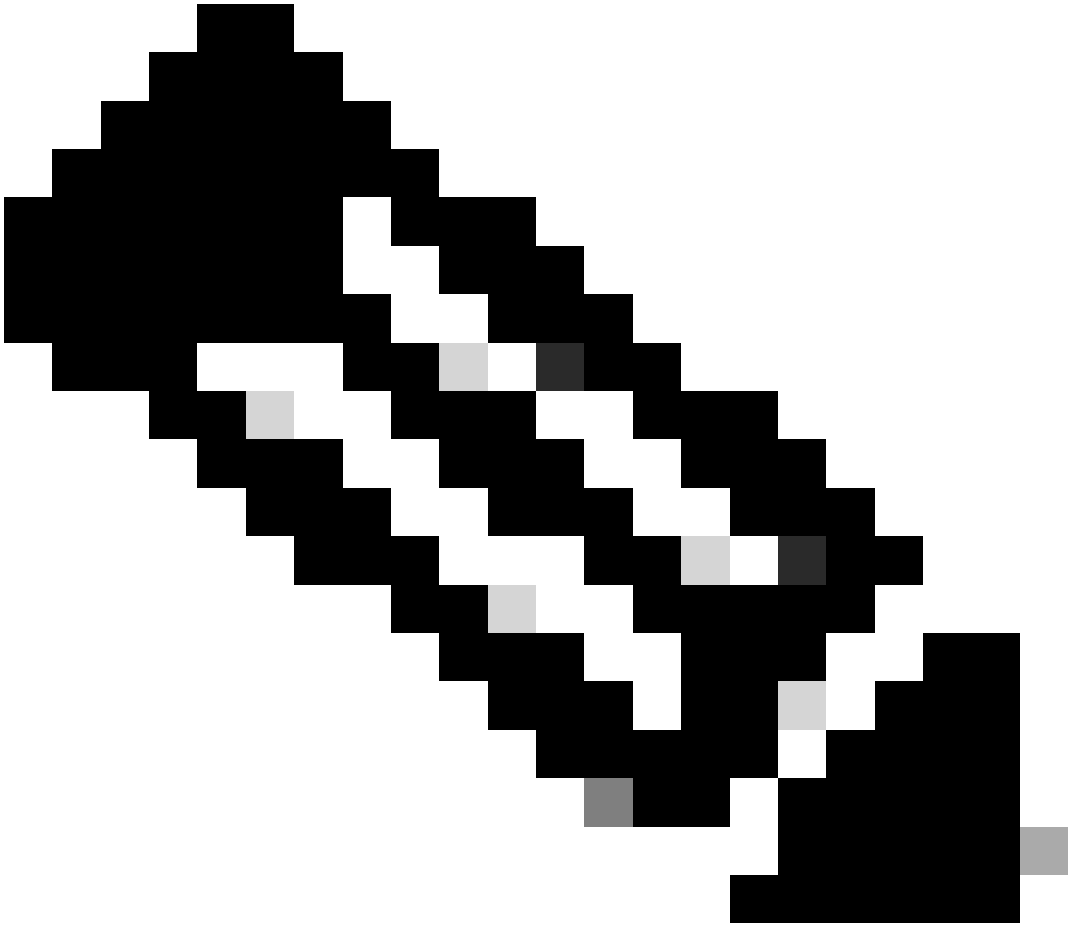
EEM N9K and N3K:

<#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
```



Note: EEM syntax can vary on different NXOS releases (some versions require "action <id> cli" and others "action <id> cli command"), so ensure to check that the EEM commands are taken properly.

Automated Process - N5K, N6K

A modified dplug file was created via Cisco bug ID CSCvr23488 to remove these Kex Algorithms:

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

The dpug files provided via Cisco bug ID CSCvr23488 are not the same as the ones that are used to access the Linux Shell. Open a TAC case to obtain the modified dplug from Cisco bug ID CSCvr23488.

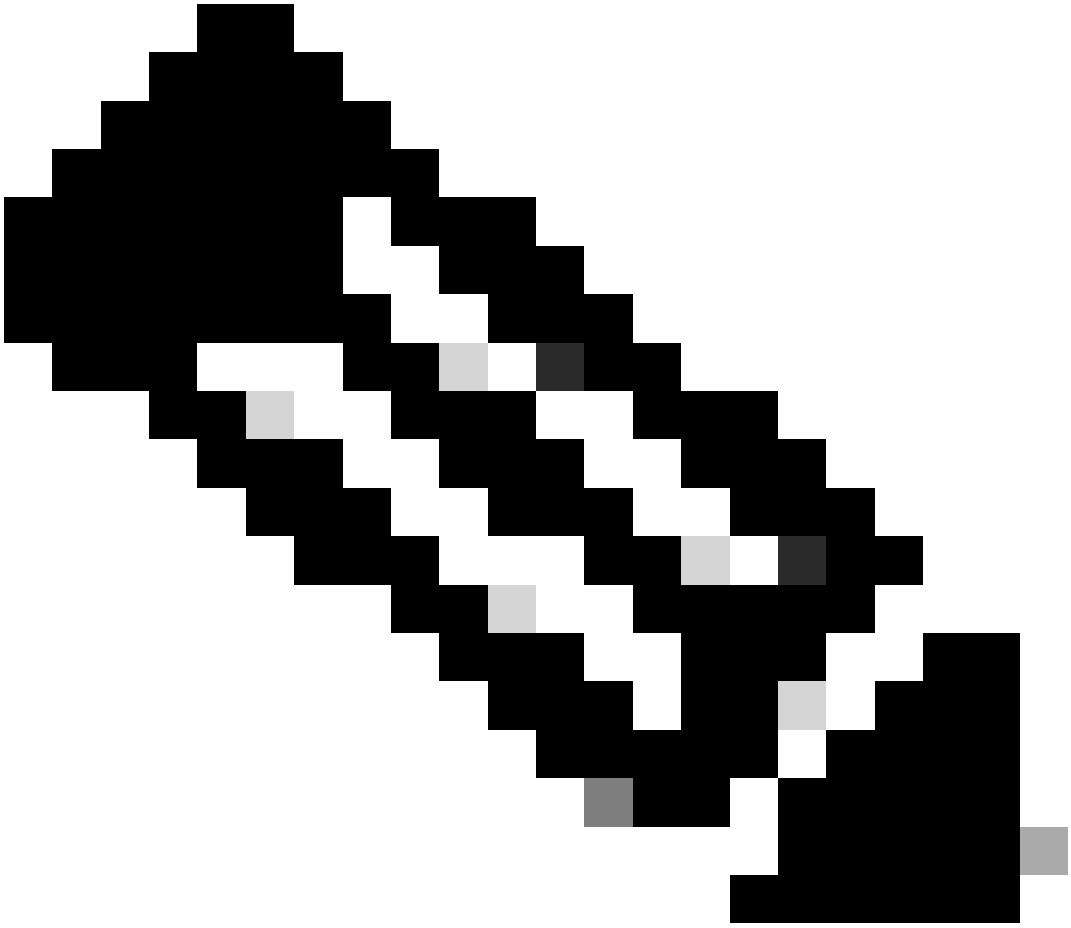
1. Verify the default `dcos_sshd_config` settings:

<#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. Create a copy of the modified dplug file.

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



Note: A copy ("dp") of the original dplug file is created in bootflash so that only the copy gets removed after the dplug is loaded and the original dplug file remains in bootflash for subsequent runs.

---

3. Apply the dplug file from Cisco bug ID CSCvr23488 manually:

```
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. Verify the new dcos_sshd_config settings:

<#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:
```

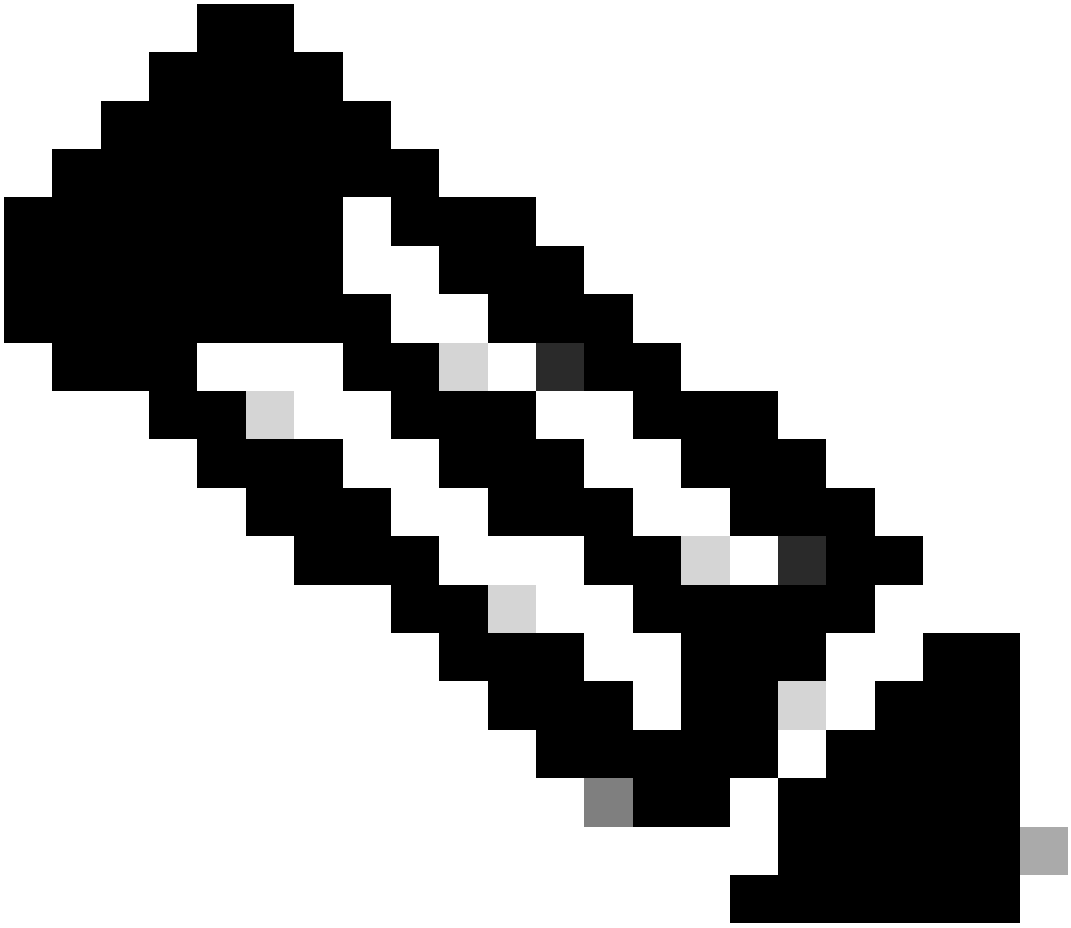
MACs stoc: hmac-sha1

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

compression stoc: none,zlib@openssh.com

5. Make this change persistent across reboots with an EEM script:

```
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"
```



Note:

- After the modified dplug is applied, the SSH feature must be reset on this platform.
- Ensure the dplug file is present in bootflash, and the EEM is configured with the proper dplug filename. The dplug filename can vary depending on the version of the switch, so make sure to modify the script as needed.
- Action 1 creates a copy of the original dplug file in bootflash to another called "dp", so the original dplug file is not deleted after being loaded.

Platform Considerations

N5K/N6K

- MAC (Message Authentication Code) cannot be changed on these platforms by modifying the dcos_sshd_config file. The only supported MAC is hmac-sha1.

N7K

- For MACs to be changed, an 8.4 code is required. See Cisco bug ID CSCwc26065for details.
- "Sudo su" is not available by default on 8.X. Reference Cisco bug ID: CSCva14865. If executed, this error is observed:

<#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$
```

To overcome this, type in:

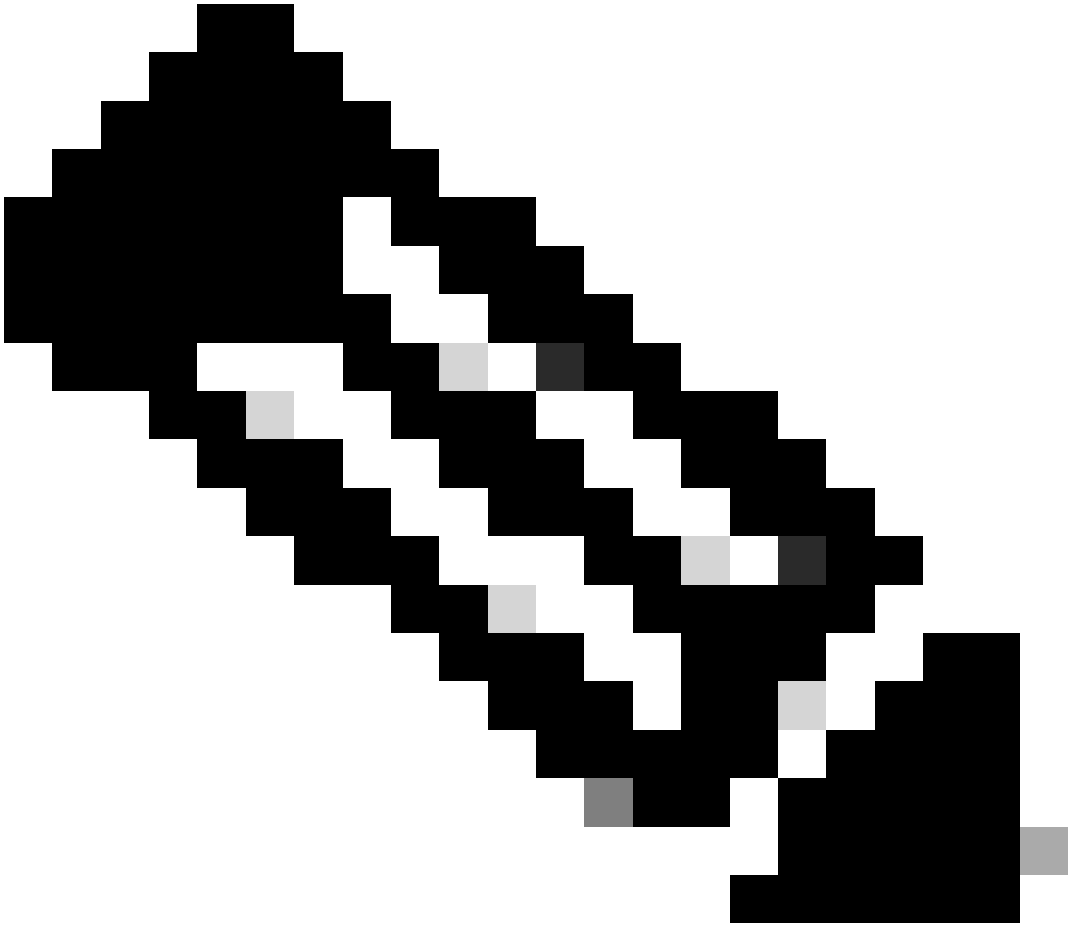
<#root>

```
    bash-4.3$
```

sudo usermod -s /bin/bash root

After this "sudo su" works:

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



Note: This change does not survive a reload.

- There is a separate dcos_sshd_config file for each VDC, in case SSH parameters need to be modified on a different VDC, ensure to modify the corresponding dcos_sshd_config file.

<#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

- Changes on the dcos_sshd_config file is not persistent across reboots on any Nexus platform. If changes need to be persistent, an EEM can be used to modify the file every time the switch boots up.
- Enhancement on N9K changes this starting 10.4(2). It is also available in 10.5(1). It was not added to newer version of previous software trains.
- See Cisco bug ID CSCwd82985for details.

Example CLI from a switch running 10.5(1):

```
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
```

Example CLI from a switch running 10.3(6):

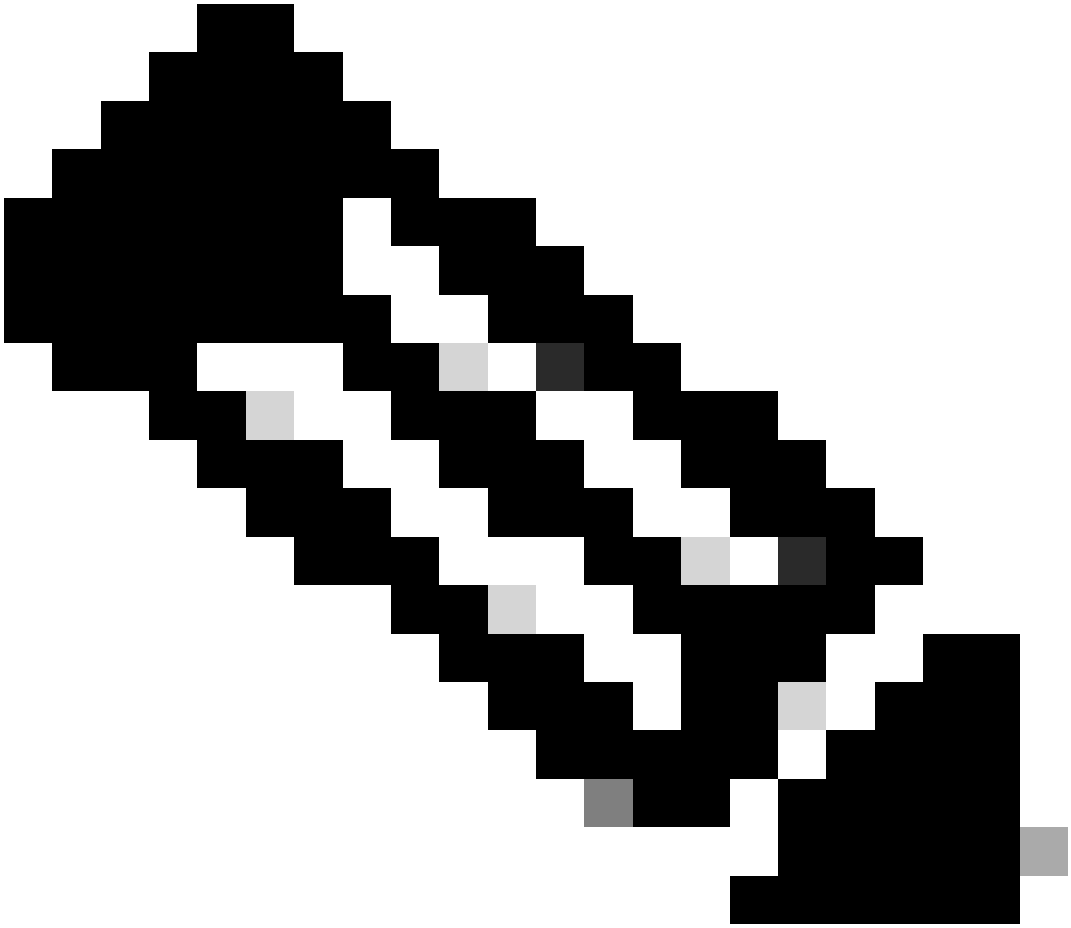
```
switch(config)# ssh kexalgos ?
all Enable algorithms supported in current version of SSH
ecdh-sha2-nistp384 Enable ecdh-sha2-nistp384

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

There are additional Ciphers, MACs, and KexAlgorithms that can be added if required:

<#root>

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



Note: These commands are available on the Nexus 7000 with releases 8.3(1) and later. For the Nexus 3000/9000 platform, the command becomes available with release 7.0(3)I7(8) and later. (All 9.3(x) releases have this command as well. See Cisco Nexus 9000 Series NX-OS Security Configuration Guide, Release 9.3(x))
