Using PowerOn Auto Provisioning

This chapter contains the following sections:

About PowerOn Auto Provisioning

PowerOn Auto Provisioning (POAP) automates the process of upgrading software images and installing configuration files on devices that are being deployed in the network for the first time.

When a device with the POAP feature boots and does not find the startup configuration, the device enters POAP mode, locates a DHCP server, and bootstraps itself with its interface IP address, gateway, and DNS server IP addresses. The device also obtains the IP address of a TFTP server and downloads a configuration script that enables the switch to download and install the appropriate software image and configuration file.

Note The DHCP information is used only during the POAP process.

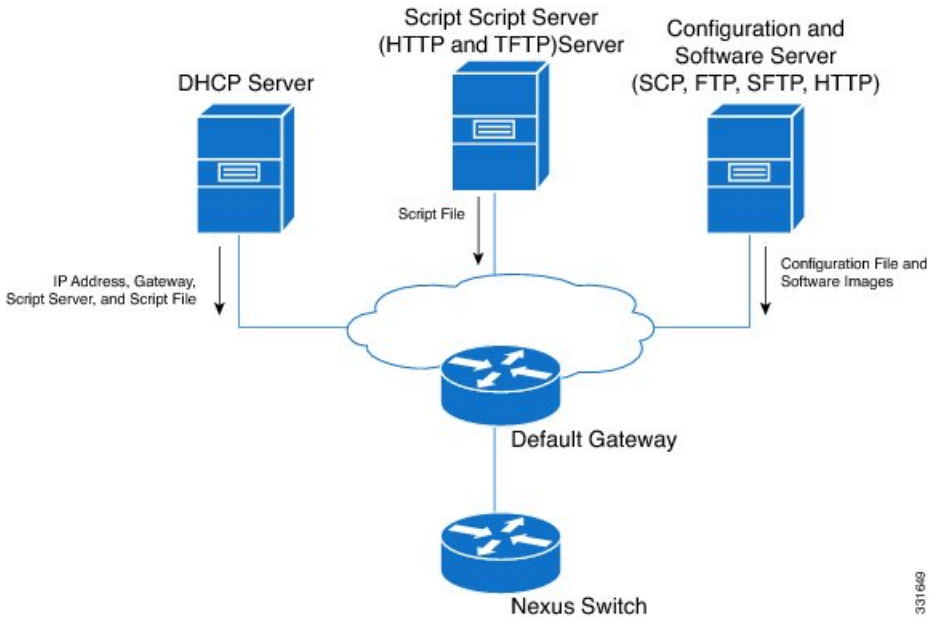
Network Requirements for POAP

POAP requires the following network infrastructure:

- A DHCP server to bootstrap the interface IP address, gateway address, and Domain Name System (DNS) server.
- A TFTP server that contains the configuration script used to automate the software image installation and configuration process.
- One or more servers that contains the desired software images and configuration files.

- If you use USB, then no DHCP server or TFTP server are required for POAP.

Figure 1: POAP Network Infrastructure



Secure Download of POAP Script

Beginning with Cisco NX-OS Release 10.2(3)F, you have the option of securely downloading the POAP script. When a device with the POAP feature boots and does not find the startup configuration, the device enters POAP mode, locates a DHCP server, and bootstraps itself with its interface IP address, gateway, and DNS server IP addresses. The device also obtains the IP address of an HTTPS server and downloads POAP script securely. The script enables the switch to download and install the appropriate software image and configuration file.

To download the POAP script securely, you need to select specific POAP options. Until Cisco NX-OS Release 10.2(3)F, POAP used options 66 and 67 for IPv4, and options 77 and 15 for IPv6 to extract the booting script information. However, the transfer of the script uses http, and is not very secure. Beginning with Cisco NX-OS Release 10.2(3)F, option 43 specifies the secure POAP related provisioning script information for IPv4 and option 17 specifies the same for IPv6. Additionally, these options allow the POAP to reach the file server in a secure manner. The POAP options 66, 67, 77, and 15 continue to be supported in Cisco NX-OS Release10.2(3)F. Furthermore, if you are using option 43 or 17, you can use the earlier options as fallback options, if required.

Note The maximum character length is 512 bytes for both option 43 and option 17.

The sub-options available for option 43 and option 17 are discussed in the following sections:

- Option 43 - IPv4
- Option 17 - IPv6

IPv4

Option 43 has the following sub-options for IPv4:

- option space poap length width 2;
- option poap.version code 1 = unsigned integer 8;

Note This sub-option is mandatory.

- option poap.ca_list code 50 = text;
- option poap.url code 2 = text;

Note This sub-option is mandatory.

- option poap.debug code 51 = unsigned integer 8;
- option poap.ntp code 3 = ip-address;

Note This sub-option is only supported for IPv4 (Option 43).

- option poap.flag code 52 = unsigned integer 8;

Note Flag is used to skip server certificate validation in the client.

Sample configuration for IPv4 is as follows:

```
host dhclient-n9kv {
hardware ethernet 00:50:56:85:c5:30;
fixed-address 3.3.3.1;
default-lease-time 3600;
option broadcast-address 192.168.1.255;
#option log-servers 1.1.1.1;
max-lease-time 3600;
option subnet-mask 255.255.255.0;
option routers 10.77.143.1;
#option domain-name-servers 1.1.1.1;
          vendor-option-space poap;
option poap.version 1;
option poap.ca_list "https://<ip>/poap/ca_file1.pem, https://<ip>/poap/ca_file2.pem";
option poap.url "https://<url>/poap.py";
option poap.debug 1;
option poap.ntp 10.1.1.39;
option poap.flag 0;
  }
```

IPv6

Option 17 has the following sub-options for IPv6:

- option space poap_v6 length width 2;
- option poap_v6.version code 1 = unsigned integer 8;

Note This sub-option is mandatory.

- option poap_v6.ca_list code 50 = text;
- option poap_v6.url code 3 = text;

Note This sub-option is mandatory.

- option poap_v6.debug code 51 = unsigned integer 8;
- option vsio.poap_v6 code 9 = encapsulate poap_v6;

Sample configuration for IPv6 is as follows:

```
 option dhcp6.next-hop-rt-prefix code 242 = { ip6-address, unsigned integer 16,
unsigned integer 16, unsigned integer 32, unsigned integer 8, unsigned integer 8, ip6-address
 };
 option dhcp6.bootfile-url code 59 = string;

 default-lease-time 3600;
 max-lease-time 3600;
 log-facility local7;
 subnet6 2003::/64 {

   # This statement configures actual values to be sent
 # RTPREFIX option code = 243, RTPREFIX length = 22
 # Ignore value 22. It is something related to option-size RT_PREFIX option length.
 # lifetime = 9000 seconds
 # route ETH1_IPV6_GW/64
 # metric 1
 option dhcp6.next-hop-rt-prefix 2003::2222 243 22 9000 0 1 ::;
 #ipv6 ::/0 2003::2222
 #Another example - support not there in NXOS - CSCvs05271:
 #option dhcp6.next-hop-rt-prefix 2003::2222 243 22 9000 112 1 2003::1:2:3:4:5:0;
 #ipv6 2003::1:2:3:4:5:0/112 2003::2222

   # Additional options
 #option dhcp6.name-servers fec0:0:0:1::1;
 #option dhcp6.domain-search "domain.example";


 range6 2003::b:1111 2003::b:9999;
 option dhcp6.bootfile-url "tftp://2003::1111/poap_github_v6.py";
 vendor-option-space poap_v6;
 option poap_v6.version 1;
 option poap_v6.ca_list "https://<ip>/new_ca.pem,https://<ip>/another_ca.pem";
 option poap_v6.url "https://<ip>/poap_github_v4.py";
```

```
option poap_v6.debug 1;
  }
```

Network Requirements for Secure POAP

Secure POAP requires the following network infrastructure:

- A DHCP server to bootstrap the interface IP address, gateway address, and Domain Name System (DNS) server.
- An HTTPS server that contains the POAP script used for software image installation and configuration process.

Note

- If the HTTPS server runs on a non-SUDI device, a physical USB drive with the CA certificates of the file-server is required.
- In case of secure download of POAP script, the TFTP server is replaced with the HTTPS server. Hence, when you read the content related to the TFTP server in this chapter, remember to read the TFTP server as the HTTPS server.

- One or more servers that contain the desired software images and configuration files.

Deployment Scenarios

Cisco devices have a unique identifier known as the Secure Unique Device Identifier (SUDI). The hardware SUDI can be used for authentication, as it can be used for asymmetric key operations such as encryption, decryption, signing, and verifying that allow passage of the data to be operated upon. All non-Cisco devices are classified as non-SUDI devices. For a non-SUDI device, the root-CA bundle is required to authenticate the file server. However, the file server can be hosted on either a SUDI or a non-SUDI device.

Based on all these capabilities, you can use one of the following deployment scenarios to download the POAP script in a secure way:

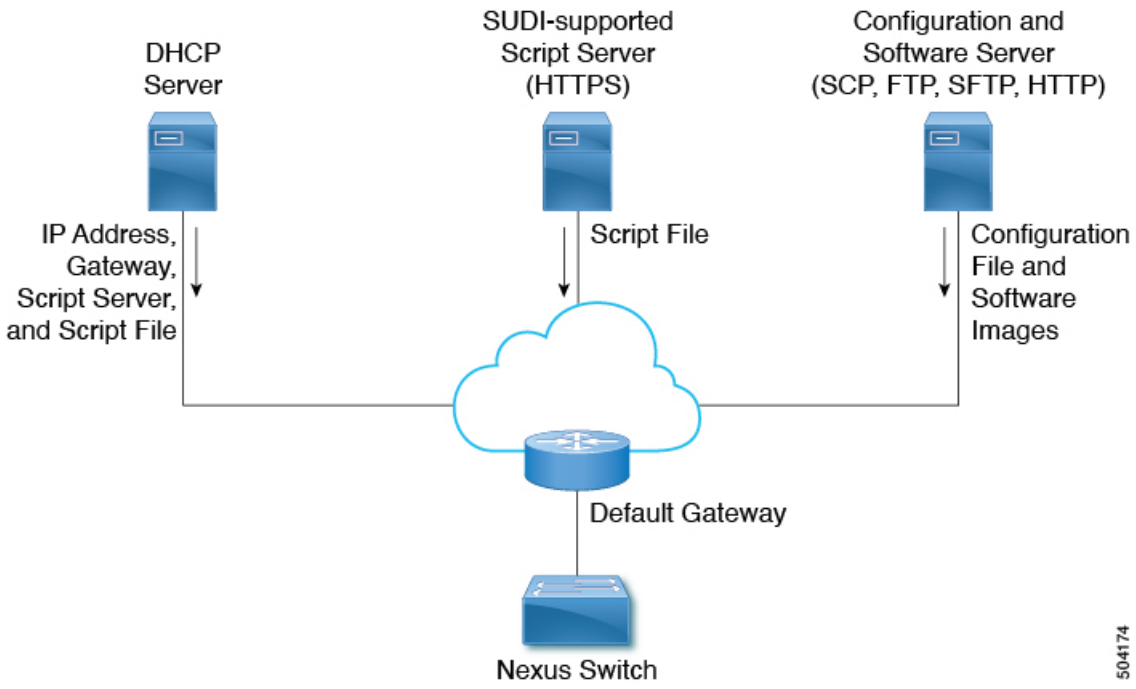
- SUDI Supported Device as File Server
- Non-SUDI Supported Device as a File Server

SUDI Supported Device as File Server

The SUDI supported devices are Cisco devices. Unlike the earlier implementation, the DHCP server now provides a https location rather than http/tftp. In this scenario, only the DHCP server and the SUDI supported script server (HTTPS server) are required, other than one or more servers that contain the required software images and configuration files.

Note SUDI only supports TLSv1.2 or below. Also, the SUDI solution only considers secure download using https, but not sftp.

Figure 2: SUDI Supported Device as File Server Infrastructure



The workflow for SUDI supported devices is as follows:

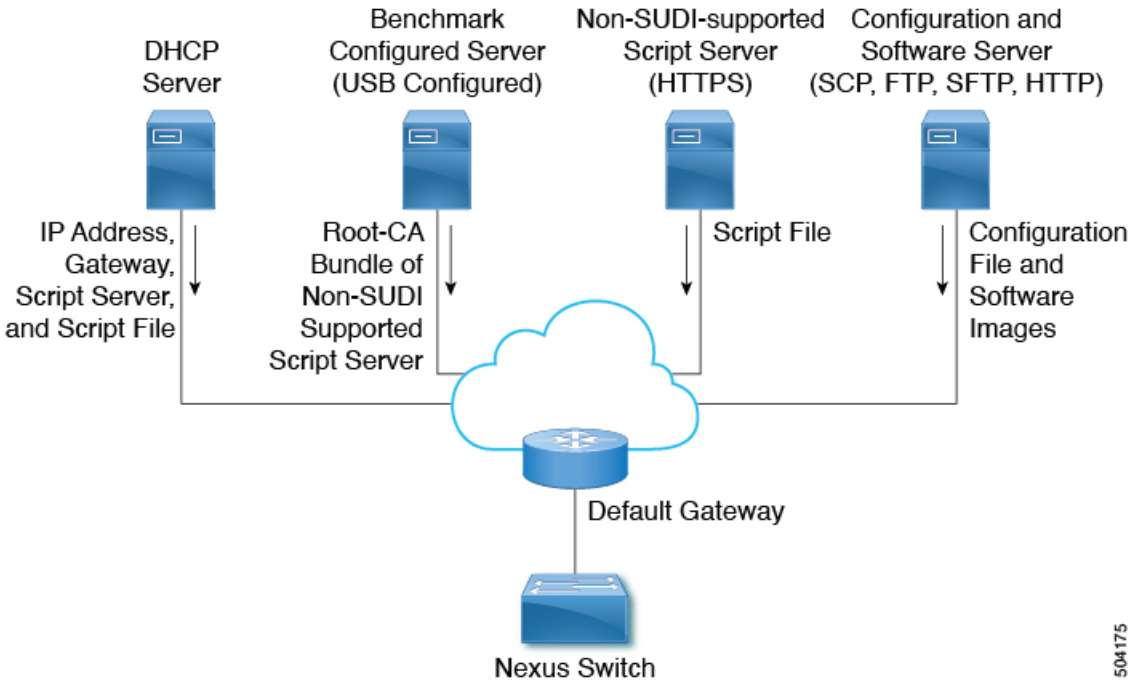
- Booting device is SUDI capable and has the needed trust store to verify a SUDI certificate
- Booting device sends out DHCP discover
- DHCP server responds to booting device with https server details
- Device establishes the secure channel using standard SSL APIs
- Authentication is done by verifying SUDI on both sides
- Downloads **poap.py**

Non-SUDI Supported Device as a File Server

In this scenario, the Root-CA bundle must be installed in the booting device. The Root-CA bundle is required for authentication. Here, the DHCP server, intermediate device, and non-SUDI supported script server (HTTPS server) are required, other than one or more servers that contain the required software images and configuration files.

The DHCP offer has the details of intermediate server that has the Root-CA bundle available. The intermediate device should support SUDI. The booting device uses the intermediate device to download the Root-CA bundle, install it, and then communicate with the file server. The intermediate devices should be provisioned first.

Figure 3: Non-SUDI Supported Device as File Server Infrastructure



The workflow for non-SUDI supported devices is as follows:

- Booting device is SUDI capable and has the needed trust store to verify a SUDI certificate
- Intermediate device that hosts a server with Root-CA bundle is also SUDI capable
- Booting device sends out DHCP discover
- DHCP server responds to booting device with https server details and Root-CA server details
- Booting device reaches to intermediate device, gets the CA bundle, adds it to the trust store
- Booting device reaches the file server to download **poap.py**

Benchmark Configured Device

The root CA certificate chain file for the Non-SUDI-supported script server must be placed in `/bootflash/poap/sudi_fs` on the Benchmark Configured Server.

Note To change port on the Benchmark Configured Device, use the **file-server** <*port-number*> command. Avoid using standard ports such as port 80 (http) and port 443 (https).

The **file-server** <*port-number*> command only serves content over the management interface.

Secure POAP on a Device Shipped with Old Image

Support for secure POAP will be available only for devices that are shipped with image that has secure POAP feature.

If the device does not have the secure POAP feature, then use the legacy DHCP options to move the device to a later version of the image that supports secure POAP. Then these devices can be reloaded and use the secure POAP feature.

Troubleshooting Secure POAP

Perform the following steps to collect debugging information regarding secure POAP:

1. Set the debug option for IPv4 in option 43 to 1 and for IPv6 in option 17.

   The debug option enables additional logs.

2. Allow the switch to run one cycle of POAP.
3. Abort POAP.
4. When the system boots up, run the **show tech-support poap** command.

   This command displays POAP status and configuration.

Disabling POAP

POAP is enabled when there is no configuration in the system. It runs as a part of bootup. However, you can bypass POAP enablement during initial setup. If you want to disable POAP permanently (even when there is no configuration in the system), you can use the 'system no poap' command. This command ensures that POAP is not started during the next boot (even if there is no configuration). To enable POAP, use the 'system poap' command or the 'write erase poap' command. The 'write erase poap' command erases the POAP flag and enables POAP.

- Example: Disabling POAP

```
switch# system no poap
switch# sh boot
Current Boot Variables:
 sup-1
NXOS variable = bootflash:/nxos.9.2.1.125.bin
Boot POAP Disabled

POAP permanently disabled using 'system no poap'


Boot Variables on next reload:

sup-1
NXOS variable = bootflash:/nxos.9.2.1.125.bin
Boot POAP Disabled

POAP permanently disabled using 'system no poap'


switch# sh system poap
System-wide POAP is disabled  using exec command 'system no poap'
POAP will be bypassed on write-erase reload.
(Perpetual POAP cannot be enabled when system-wide POAP is disabled)
```

- Example: Enabling POAP

```
switch# system poap

switch# sh system poap

System-wide POAP is enabled
```

- Example: Erase POAP

```
switch# write erase poap
This command will erase the system wide POAP disable flag only if it is set.
Do you wish to proceed anyway? (y/n)  [n] y
System wide POAP disable flag erased.

switch# sh system poap
System-wide POAP is enabled
```

POAP Configuration Script

We provide a sample configuration script that is developed using the Python programming language. We recommend using the provided script and modifying it to meet the requirements of your network environment.

The POAP script can be found at https://github.com/datacenter/nexus9000/blob/master/nx-os/poap/poap.py.

To modify the script using Python, see the *Cisco NX-OS Python API Reference Guide* for your platform.

Using the POAP Script and POAP Script Options

Before using the POAP script, perform the following actions:

1. Edit the options dictionary at the top of the script to ensure that all relevant options for your setup are included in the script. Do not change the defaults (in the default options function) directly.
2. Update the MD5 checksum of the POAP script as shown using shell commands.

   ```
   f=poap_nexus_script.py ; cat $f | sed '/^#md5sum/d' > $f.md5 ; sed -i
   "s/^#md5sum=.*/#md5sum=\"$(md5sum $f.md5 | sed 's/ .*//')\"/" $f
   ```

3. If the device has a startup configuration, perform a write erase and reload the device.

The following POAP script options can be specified to alter the POAP script behavior. When you download files from a server, the hostname, username, and password options are required. For every mode except personality, the target_system_image is also required. Required parameters are enforced by the script, and the script aborts if the required parameters are not present. Every option except hostname, username, and password has a default option. If you do not specify the option in the options dictionary, the default is used.

- **username**

  The username to use when downloading files from the server.

- **password**

  The password to use when downloading files from the server.

- **hostname**

  The name or address of the server from which to download files.

- **mode**

  The default is **serial_number**.

  Use one of the following options:

  - **personality**

    A method to restore the switch from a tarball.

  - **serial_number**

    The serial number of the switch to determine the configuration filename. The format for the serial number in the configuration file is conf.*serialnumber*. Example: conf.FOC123456

  - **hostname**

    The hostname as received in the DHCP options to determine the configuration filename. The format for the hostname in the configuration file is conf_*hostname*.cfg. Example: conf_3164-RS.cfg

  - **mac**

    The interface MAC address to determine the configuration filename. The format for the hostname in the configuration file is conf_*macaddress*.cfg. Example: conf_7426CC5C9180.cfg

  - **raw**

    The configuration filename is used exactly as provided in the options. The filename is not altered in any way.

  - **location**

    The CDP neighbors are used to determine the configuration filename. The format for the location in the configuration file is conf_*host_intf*.cfg, where *host* is the host connected to the device over the POAP interface, and *intf* is the remote interface to which the POAP interface is connected. Example: conf_remote-switch_Eth1_8.cfg

- **required_space**

  The required space in KB for that particular iteration of POAP. The default is 100,000. For multi-step upgrades, specify the size of the last image in the upgrade path of the target image.

- **transfer_protocol**

  Any transfer protocol such as http, https, ftp, scp, sftp, or tftp that is supported by VSH. The default is scp.

- **config_path**

  The path to the configuration file on the server. Example: /tftpboot. The default is /var/lib/tftpboot.

- **target_system_image**

  The name of the image to download from the remote server. This is the image you get after POAP completes. This option is a required parameter for every mode except personality. The default is "".

- **target_image_path**

  The path to the image on the server. Example: /tftpboot. The default is /var/lib/tftpboot.

- **destination_path**

The path to which to download images and MD5 sums. The default is /bootflash.

- **destination_system_image**

  The name for the destination image filename. If not specified, the default will be the target_system_image name.

- **user_app_path**

  The path on the server where the user scripts, agents, and user data are located. The default is /var/lib/tftpboot.

- **disable_md5**

  This is True if MD5 checking should be disabled. The default is False.

- **midway_system_image**

  The name of the image to use for the midway system upgrade. By default, the POAP script finds the name of any required midway images in the upgrade path and uses them. Set this option if you prefer to pick a different midway image for a two-step upgrade. The default is "".

- **source_config_file**

  The name of the configuration file when raw mode is used. The default is poap.cfg.

- **vrf**

  The VRF to use for downloads and so on. The VRF is automatically set by the POAP process. The default is the POAP_VRF environment variable.

- **destination_config**

  The name to use for the downloaded configuration. The default is poap_replay.cfg.

- **split_config_first**

  The name to use for the first configuration portion if the configuration needs to be split. It is applicable only when the configuration requires a reload to take effect. The default is poap_1.cfg.

- **split_config_second**

  The name to use for the second configuration portion if the configuration is split. The default is poap_2.cfg.

- **timeout_config**

  The timeout in seconds for copying the configuration file. The default is 120. For non-legacy images, this option is not used, and the POAP process times out. For legacy images, FTP uses this timeout for the login process and not for the copy process, while scp and other protocols use this timeout for the copy process.

- **timeout_copy_system**

  The timeout in seconds for copying the system image. The default is 2100. For non-legacy images, this option is not used, and the POAP process times out. For legacy images, FTP uses this timeout for the login process and not for the copy process, while scp and other protocols use this timeout for the copy process.

- **timeout_copy_personality**

  The timeout in seconds for copying the personality tarball. The default is 900. For non-legacy images, this option is not used, and the POAP process times out. For legacy images, FTP uses this timeout for

the login process and not for the copy process, while scp and other protocols use this timeout for the copy process.

- **timeout_copy_user**

  The timeout in seconds for copying any user scripts and agents. The default is 900. For non-legacy images, this option is not used, and the POAP process times out. For legacy images, FTP uses this timeout for the login process and not for the copy process, while scp and other protocols use this timeout for the copy process.

- **personality_path**

  The remote path from which to download the personality tarball. Once the tarball is downloaded and the personality process is started, the personality will download all files in the future from locations specified inside the tarball configuration. The default is /var/lib/tftpboot.

- **source_tarball**

  The name of the personality tarball to download. The default is personality.tar.

- **destination_tarball**

  The name for the downloaded personality tarball after it is downloaded. The default is personality.tar.

Setting up the DHCP Server without DNS for POAP

Beginning with Cisco NX-OS Release 7.0(3)I6(1), the tftp-server-name can be used without the DNS option. To enable POAP functionality without DNS on earlier releases, a custom option of 150 must be used to specify the tftp-server-address.

To use the tftp-server-address option, specify the following at the start of your dhcpd.conf file.

```
option tftp-server-address code 150 = ip-address;
```

For example:

```
host MyDevice {
    option dhcp-client-identifier "\000SAL12345678";
    fixed-address 2.1.1.10;
    option routers 2.1.1.1;
    option host-name "MyDevice";
    option bootfile-name "poap_nexus_script.py";
    option tftp-server-address 2.1.1.1;
}
```

The below example shows Configuring DHCPv6 for POAP over IPv6:

```
default-lease-time 3600;
max-lease-time 3600;
log-facility local7;
subnet6 2003::/64 {

        # This statement configures actual values to be sent
        # RTPREFIX option code = 243, RTPREFIX length = 22
        # Ignore value 22. It is something related to option-size RT_PREFIX option length.
        # lifetime = 9000 seconds
        # route ETH1_IPV6_GW/64
        # metric 1
        option dhcp6.next-hop-rt-prefix 2003::2222 243 22 9000 0 1 ::;
        #ipv6 ::/0 2003::2222
```

```
#Another example - support not there in NXOS - CSCvs05271:
#option dhcp6.next-hop-rt-prefix 2003::2222 243 22 9000 112 1 2003::1:2:3:4:5:0;
#ipv6 2003::1:2:3:4:5:0/112 2003::2222

# Additional options
#option dhcp6.name-servers fec0:0:0:1::1;
#option dhcp6.domain-search "domain.example";

range6 2003::b:1111 2003::b:9999;
#range6 2003::c:2222 2003::c:2222;
option dhcp6.bootfile-url "tftp://2003::1111/poap_github_v6.py";
```

Downloading and Using User Data, Agents, and Scripts as part of POAP

Under the options dictionary, you can find the **download_scripts_and_agents** function. If you choose to download user scripts and data, uncomment the first **poap_log** line and then use a series of **download_user_app** function calls to download each application. Since older Cisco NX-OS versions do not support recursive copy of directories, such directories must be put into a tarball (TAR archive) and then unpacked once on the switch. The parameters for the **download_scripts_and_agents** function are as follows:

- **source_path** - The path to where the file or tarball is located. This is a required parameter. Example: /var/lib/tftpboot.
- **source_file** - The name of the file to download. This is a required parameter. Example: agents.tar, script.py, and so on.
- **dest_path** - The location to download the file on the switch. Any directories that do not exist earlier will be created. This is an optional parameter. The default is /bootflash.
- **dest_file** - The name to give the downloaded file. This is an optional parameter. The default is unchanged source_file.
- **unpack** - Indicates whether a tarball exists for unpacking. Unpacking is done with **tar -xf** *tarfile* **-C /bootflash**. This is an optional parameter. The default is False.
- **delete_after_unpack** - Indicates whether to delete the downloaded tarball after unpack is successful. There is no effect if unpack is False. The default is False.

Using the download functionality, you can download all the agents and files needed to run POAP. To start the agents, you should have the configuration present in the running configuration downloaded by POAP. Then the agents, scheduler, and cron entry, along with EEM, can be used.

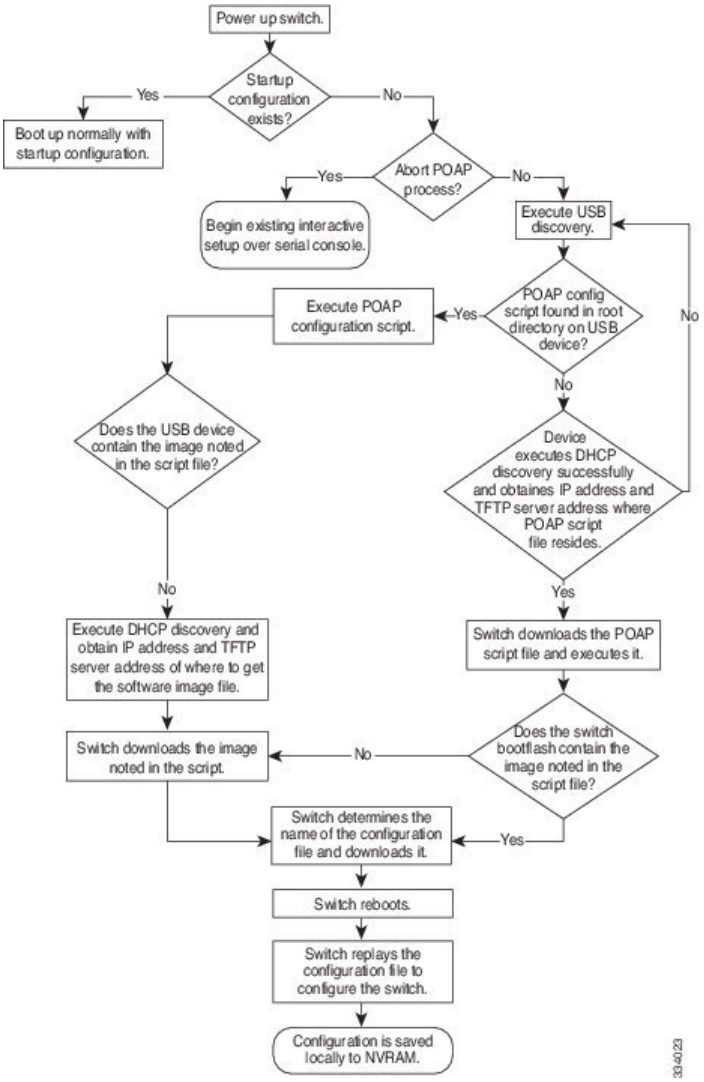
POAP Process

The POAP process has the following phases:

1. Power up
2. USB discovery
3. DHCP discovery
4. Script execution
5. Post-installation reload

Within these phases, other process and decision points occur. The following illustration shows a flow diagram of the POAP process.

Figure 4: POAP Process



Power-Up Phase

When you powerup the device for the first time, it loads the software image that is installed at manufacturing and tries to find a configuration file from which to boot. When a configuration file is not found, POAP mode starts.

During startup, a prompt appears asking if you want to abort POAP and continue with a normal setup. You can choose to exit or continue with POAP.

Note No user intervention is required for POAP to continue. The prompt that asks if you want to abort POAP remains available until the POAP process is complete.

If you exit POAP mode, you enter the normal interactive setup script. If you continue in POAP mode, all the front-panel interfaces are set up in the default configuration.

USB Discovery Phase

When POAP starts, the process searches the root directory of all accessible USB devices for the POAP script file (the Python script file, poap_script.py), configuration files, and system and kickstart images.

If the script file is found on a USB device, POAP begins running the script. If the script file is not found on the USB device, POAP executes DHCP discovery. (When failures occur, the POAP process alternates between USB discovery and DHCP discovery, until POAP succeeds or you manually abort the POAP process.)

If the software image and switch configuration files specified in the configuration script are present, POAP uses those files to install the software and configure the switch. If the software image and switch configuration files are not on the USB device, POAP does some cleanup and starts DHCP phase from the beginning.

DHCP Discovery Phase

The switch sends out DHCP discover messages on the front-panel interfaces or the MGMT interface that solicit DHCP offers from the DHCP server or servers. (See the following figure.) The DHCP client on the Cisco Nexus switch uses the switch serial number in the client-identifier option to identify itself to the DHCP server. The DHCP server can use this identifier to send information, such as the IP address and script filename, back to the DHCP client.

POAP requires a minimum DHCP lease period of 3600 seconds (1 hour). POAP checks the DHCP lease period. If the DHCP lease period is set to less than 3600 seconds (1 hour), POAP does not complete the DHCP negotiation.

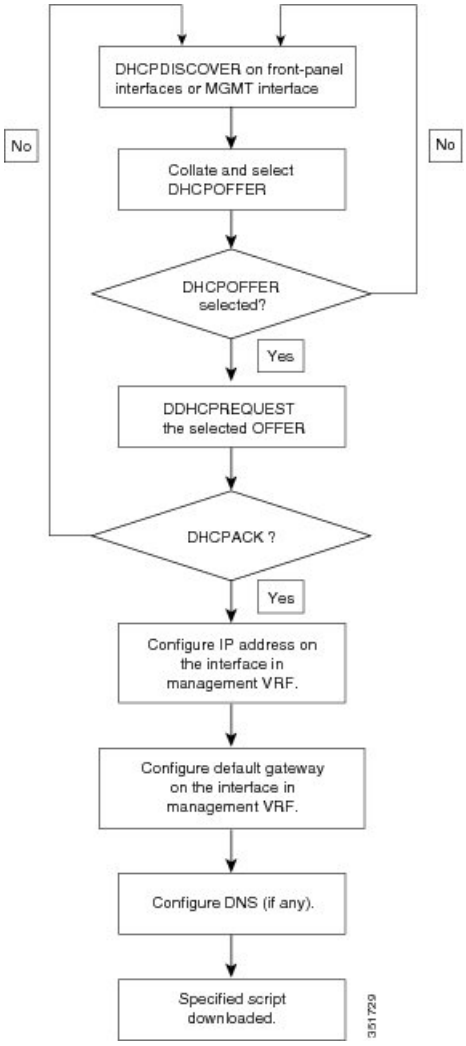
The DHCP discover message also solicits the following options from the DHCP server:

- TFTP server name or TFTP server address—The DHCP server relays the TFTP server name or TFTP server address to the DHCP client. The DHCP client uses this information to contact the TFTP server to obtain the script file.
- Bootfile name—The DHCP server relays the bootfile name to the DHCP client. The bootfile name includes the complete path to the bootfile on the TFTP server. The DHCP client uses this information to download the script file.

When multiple DHCP offers that meet the requirement are received, the one arriving first is honored and the POAP process moves to next stage. The device completes the DHCP negotiation (request and acknowledgment) with the selected DHCP server, and the DHCP server assigns an IP address to the switch. If a failure occurs in any of the subsequent steps in the POAP process, the IP address is released back to the DHCP server.

If no DHCP offers meet the requirements, the switch does not complete the DHCP negotiation (request and acknowledgment) and an IP address is not assigned.

Figure 5: DHCP Discovery Process



POAP Dynamic Breakout

Beginning with Cisco NX-OS Release 7.0(3)I4(1), POAP dynamically breaks out ports in an effort to detect a DHCP server behind one of the broken-out ports. Previously, the DHCP server used for POAP had to be directly connected to a normal cable because breakout cables were not supported.

POAP determines which breakout map (for example, 10gx4, 50gx2, 25gx4, or 10gx2) will bring up the link connected to the DHCP server. If breakout is not supported on any of the ports, POAP skips the dynamic breakout process. After the breakout loop completes, POAP proceeds with the DHCP discovery phase as normal.

Note For more information on dynamic breakout, see the interfaces configuration guide for your device.

Script Execution Phase

After the device bootstraps itself using the information in the DHCP acknowledgement, the script file is downloaded from the TFTP server.

The switch runs the configuration script, which downloads and installs the software image and downloads a switch-specific configuration file.

However, the configuration file is not applied to the switch at this point, because the software image that currently runs on the switch might not support all of the commands in the configuration file. After the switch reboots, it begins running the new software image, if an image was installed. At that point, the configuration is applied to the switch.

Note If the switch loses connectivity, the script stops, and the switch reloads its original software images and bootup variables.

Post-Installation Reload Phase

The switch restarts and applies (replays) the configuration on the upgraded software image. Afterward, the switch copies the running configuration to the startup configuration.

POAPv3

PowerOn Auto Provisioning version 3 (POAPv3) is introduced in Cisco NX-OS Release 9.3(5). With this feature you can install license, RPM, and certificate through POAP.

Perform the following steps to install license or RPM or certificate through POAP.

1. Create a folder on the POAP server with serial number of the box as the name.
2. Create .yaml or .yml file with files to be installed. Make sure the file name is in <serial-number>.yaml or <serial-number>.yml format.
3. Create MD5 checksum for the .yaml or .yml file.
4. Make sure the format of the .yaml file should be similar to the below format:

```
Version : 1

Target-image : nxos.9.3.4.bin

Description : Yaml for box XYZ12345 poap provisioning. N9k Leaf mode box

License : [license1.lic, XYZ12345/license2.lic, folder1/license3.lic]

RPM :

  - rpm1.rpm

  - patches/reload/rpm2-reload.rpm

  - rpm3.rpm

Certificate : [ssh1.pub, XYZ12345/ssh2key.pub]
```

```
Trustpoint :

    CA1 :

        cert_1.p12 : password1 (priv_key_passphrase)

        XYZ12345/CA1/cert_2.pfx : password2

    CA2 :

        CA2/XYZ12345/cert_3.p12 : password3
```

5. Note that the yaml keywords must match the format shown in above example.
6. Place all files in appropriate path.
7. Update the POAP script with install_path variable as the path where folder with the serial number as name is placed.

The following list provides the guidelines and limitations related to POAPv3:

- YAML is a human friendly data serialization standard for all programming languages. YAML stands for YAML Ain't Markup Language, and this file format technology is used in documents. These documents are saved in plain text format and are appended with the . yml extension. YAML is the file format and .yml is the file extension.
- YAML is a superset of JSON and the YAML parser understands JSON. YAML file formats are used for configuration management because it is easy to read and comments are useful.
- The Target_image mentioned in yaml should be kept only in the target_system_image path mentioned within POAP script. Relative path is not supported for the Target_image in yaml file.
- Both .yaml and .yml extensions are supported. You have an option to choose to use any of these extensions. If you don’t choose any option, the <serial>.yaml extension will be tried first and if it fails the <serial>.yml is considered.
- The MD5 files of yaml/yml is required similar to the configuration file. But if the disable_md5 is ‘True’ then the MD5 files of yaml/yml are not required.
- Although 'install_path' is set in the POAP script file if no yaml file for device is found, then POAP workflow will proceed with the legacy path, i.e., without any installation of RPMs, licenses and certificates.
- Install reset is highly preferred over write erase if PoAP with RPM installation is done in scenarios apart from Day-0.
- ISSU is the new default for moving to new image via PoAP. Note that you need to use "use_nxos_boot": True, if legacy boot nxos <> is required.
- The Filetype checks for .pfx,.p12 in trustpoints; .lic in license; and .rpm in rpms and aborts the current POAP if the checks/fileformats are not honoured.
- In case of .rpm, you need to provide the original file name in the yaml file.

  For example: if you renamed customCliGoApp-1.0-1.7.5.x86_64.rpm to custom.rpm then PoAP will bail out indicating the name mismatch.

  To get the original name of rpm:

```
bash-4.3$ rpm -qp --qf '%{NAME}-%{VERSION}-%{RELEASE}.%{ARCH}.rpm' custom.rpm
customCliGoApp-1.0-1.7.5.x86_64.rpm
bash-4.3$
```

- Once ISSU via POAP begins, abort of PoAP will be blocked. If ISSU fails for some reason, then abort capability will be re-enabled.

Guidelines and Limitations for POAP

POAP configuration guidelines and limitations are as follows:

- The bootflash:poap_retry_debugs.log is a file populated by POAP-PNP for internal purposes only. This file has no relevance in case of any POAP failures.
- Due to limitations in Syslog, securePOAP pem file name characters length is limited to 230 characters, though secure POAP supports 256 characters length for a pem file name.
- The switch software image must support POAP for this feature to function.
- POAP does not support provisioning of the switch after it has been configured and is operational. Only auto-provisioning of a switch with no startup configuration is supported.
- The **https_ignore_certificate** option should be turned on to use the **ignore-certificate** keyword with https protocol in POAP. This would enable you to successfully perform HTTPS transfer in the POAP script and without this option https as protocol cannot work with POAP.
- For those who uses HTTP/HTTPS servers for Day 0 provisioning, provisioning instructions will be given based on the MAC information and other related details in the HTTP header. POAP uses these details from HTTP GET headers so that the correct provisioning script is identified and used. This was available for other vendors (and other Cisco OSs).These additional information will be available in HTTP get headers from Cisco NX-OS Release 10.2(1) for Cisco Nexus 9000. This feature will be available by default for POAP and non-POAP HTTP get operations.
- When you use copy http/https GET commands, the following fields are shared as part of the HTTP header:

```
Host: IP address
User-Agent: cisco-nxos
X-Vendor-SystemMAC: System MAC
X-Vendor-ModelName: Switch-Model
X-Vendor-Serial: Serial_Num
X-Vendor-HardwareVersion: Hardwareversion
X-Vendor-SoftwareVersion: sw_version
X-Vendor-Architecture: Architecture
```

- If you use POAP to bootstrap a Cisco Nexus device that is a part of a virtual port channel (vPC) pair using static port channels on the vPC links, the Cisco Nexus device activates all of its links when POAP starts up. The dually connected device at the end of the vPC links might start sending some or all of its traffic to the port-channel member links that are connected to the Cisco Nexus device, which causes traffic to get lost.

  To work around this issue, you can configure Link Aggregation Control Protocol (LACP) on the vPC links so that the links do not incorrectly start forwarding traffic to the Cisco Nexus device that is being bootstrapped using POAP.
- If you use POAP to bootstrap a Cisco Nexus device that is connected downstream to a Cisco Nexus 9000 Series switch through a LACP port channel, the Cisco Nexus 9000 Series switch defaults to suspend its member port if it cannot bundle it as a part of a port channel. To work around this issue, configure the

Cisco Nexus 9000 Series switch to not suspend its member ports by using the **no lacp suspend-individual** command from interface configuration mode.

- Important POAP updates are logged in the syslog and are available from the serial console.
- Critical POAP errors are logged to the bootflash. The filename format is *date-time*_poap_*PID*_[init,1,2].log, where *date-time* is in the YYYYMMDD_hhmmss format and *PID* is the process ID.
- You can bypass the password and the basic POAP configuration by using the **skip** option at the POAP prompt. When you use the **skip** option, no password is configured for the admin user. The **copy running-config startup-config** command is blocked until a valid password is set for the admin user.
- If the **boot poap enable** command (perpetual POAP) is enabled on the switch, on a reload, a POAP boot is triggered even if there is a startup configuration present. If you do not want to use POAP in this scenario, remove the boot poap enable configuration by using the **no boot poap enable** command.
- Script logs are saved in the bootflash directory. The filename format is *date-time*_poap_*PID*_script.log, where *date-time* is in the YYYYMMDD_hhmmss format and *PID* is the process ID.

  You can configure the format of the script log file. Script file log formats are specified in the script. The template of the script log file has a default format; however, you can choose a different format for the script execution log file.
- The POAP feature does not require a license and is enabled by default. However for the POAP feature to function, appropriate licenses must be installed on the devices in the network before the deployment of the network.
- USB support for POAP enables checking a USB device containing the configuration script file in POAP mode. This feature is supported on the Nexus 9300-EX, -FX, -FX2, -FX3, and Nexus 9200-X, -FX2 switches.
- POAP DHCP transaction may fail if the device receives high traffic rate. This issue happens when POAP uses a front panel. To avoid this issue, make sure POAP uses a management port.
- Beginning with NX-OS 7.0(3)I7(4), RFC 3004 (User Class Option for DHCP) is supported. This enables POAP to support user-class option 77 for DHCPv4 and user-class option 15 for DHCPv6. The text displayed for the user class option for both DHCPv4 and DHCPv6 is "Cisco-POAP".
  - With RFC 3004 (User Class Option for DHCP) support, POAP over IPv6 is supported on Nexus 9000 switches.
    - Beginning with NX-OS 9.2(2), POAP over IPv6 is supported on Nexus 9504 and Nexus 9508 switches with –R line cards.

  The POAP over IPv6 feature enables the POAP process to use IPv6 when IPv4 fails. The feature is designed to cycle between IPv4 and IPv6 protocols when a connection failure occurs.
- For secure POAP, ensure that DHCP snooping is enabled.
- To support POAP, set firewall rules to block unintended or malicious DHCP servers.
- To maintain system security and make POAP more secure, configure the following:
  - Enable DHCP snooping.
  - Set firewall rules to block unintended or malicious DHCP servers.

- POAP is supported on both MGMT ports and in-band ports.
- Beginning with Cisco NX-OS Release 9.3(9), for RFC 2132 (refer to section 9.6 for message type option 53 for DHCP), when your offer does not have the option 53 for DHCPV4 as the first packet, the next POAP detects the offer and does the auto provisioning.
- Beginning with Cisco NX-OS Release 10.2(3)F, the Hardware SUDI for POAP/HTTPS feature provides an option to securely download the POAP script.
- To collect the debugging information on POAP, use the **show tech-support poap** command, post abort of POAP.
- Beginning with Cisco NX-OS Release 10.3(1)F, POAP is supported on Cisco Nexus X9836DM-A line card of the Cisco Nexus 9808 platform switches.

Setting Up the Network Environment to Use POAP

Step 1 Modify the basic configuration script provided by Cisco or create your own script. For information, see the *Python Scripting and API Configuration Guide*.

Step 2 Every time you make a change to the configuration script, ensure that you recalculate the MD5 checksum by running **# f=poap_nexus_script.py ; cat $f | sed '/^#md5sum/d' > $f.md5 ; sed -i "s/^#md5sum=.*/#md5sum=\"$(md5sum $f.md5 | sed 's/ .*//')\"/" $f** using a bash shell. For more information, see the *Python API Reference Guide*.

Step 3 (Optional) Put the POAP script and any other desired software image and switch configuration files on a USB device accessible to the switch.

Step 4 Deploy a DHCP server and configure it with the interface, gateway, and TFTP server IP addresses and a bootfile with the path and name of the configuration script file. (This information is provided to the switch when it first boots.) You do not need to deploy a DHCP server if all software image and switch configuration files are on the USB device.

Step 5 Deploy a TFTP or HTTP server to host the configuration script. In order to trigger the HTTP request to the server, prefix HTTP:// to the TFTP server name. HTTPS is not supported.

Step 6 Add the URL portion into the TFTP script name to show correct path to the file name.

Step 7 Deploy one or more servers to host the software images and configuration files.

Configuring a Switch Using POAP

Before you begin

Make sure that the network environment is set up to use POAP.

Step 1 Install the switch in the network.

Step 2 Power on the switch.

If no configuration file is found, the switch boots in POAP mode and displays a prompt that asks if you want to abort POAP and continue with a normal setup.

No entry is required to continue to boot in POAP mode.

Step 3 (Optional) If you want to exit POAP mode and enter the normal interactive setup script, enter **y** (yes).

The switch boots, and the POAP process begins.

What to do next

Verify the configuration.

Creating md5 Files

Every time you make a change to the configuration script, ensure that you recalculate the MD5 checksum by running # f=poap_fabric.py ; cat $f | sed '/^#md5sum/d' > $f.md5 ; sed -i "s/^#md5sum=.*/#md5sum=\"$(md5sum $f.md5 | sed 's/ .*//')\"/" $f using a bash shell.

This procedure replaces md5sum in `poap_fabric.py` with a new value if there was any change in that file.

Note Steps 1-4 and 7-8 are needed only if you are using the BASH shell. If you have access to any other Linux server, these steps are not required.

Before you begin

Access to the BASH shell.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature bash-shell** **Example:** `switch(config)# feature bash-shell`	Enable BASH shell feature.
Step 3	**exit** **Example:** `switch(config)# exit`	Exit configuration mode.

	Command or Action	Purpose
Step 4	**run bash** **Example:** `switch# run bash`	Open Linux BASH.
Step 5	**md5sum /bootflash/nxos.***release_number***.bin** > **/bootflash/nxos.***release_number***.bin.md5** **Example:** `bash-4.2$ md5sum /bootflash/nxos.7.0.3.I6.1.bin > /bootflash/nxos.7.0.3.I6.1.bin.md5`	Creates md5sum for the `.bin` file.
Step 6	**md5sum /bootflash/poap.cfg > /bootflash/poap.cfg.md5** **Example:** `bash-4.2$ md5sum /bootflash/poap.cfg > /bootflash/poap.cfg.md5`	Creates md5sum for the `.cfg` file.
Step 7	**exit** **Example:** `switch(config)# exit`	Exit the BASH shell.

Step 8	**dir | i .md5**
	Example:
	`switch# dir | i .md5`
	`65 Jun 09 12:38:48 2017 nxos.7.0.3.I6.1.bin.md5`
	`54 Jun 09 12:39:36 2017 poap.cfg.md5`
	`67299 Jun 09 12:48:58 2017 poap.py.md5`
		Display the .md5 files.

Step 9	**copy bootflash:poap.cfg.md5 scp://***ip_address***/**
	Example:
	`copy bootflash:poap.cfg.md5 scp://10.1.100.3/`
	`Enter vrf (If no input, current vrf 'default' is considered): management`
	`Enter username: root`
	`root@10.1.100.3's password:`
	`poap.cfg.md5 100%`
	`54 0.1KB/s 00:00`
	`Copy complete.`
		Uploads the files to the Configuration and Software Server.

Verifying the Device Configuration

To verify the configuration, use one of the following commands:

Command	Purpose
show running-config [**[exclude]** *command*] **[sanitized]**	Displays the contents of the currently running configuration or a subset of that configuration, use the **show running-config** command in the appropriate mode. : • **exclude**: (Optional) Excludes a specific configuration from the display. Use the **exclude** keyword followed by a *command* argument to exclude a specific configuration from the display. • *command*: (Optional) Displays only a single command or a subset of commands available under a specified command mode. • **sanitized**: (Optional) Displays a sanitized configuration for safe distribution and analysis. Beginning with Cisco NX-OS Release 10.3(2)F, **sanitized** keyword is supported on Cisco Nexus 9000 series switches.
show startup-config	Displays the startup configuration.
show time-stamp running-config last-changed	Displays the timestamp when the running configuration was last changed.

The following example shows sample output of **show running-config** command with the **sanitized** keyword. The sanitized configuration is used to share a configuration without exposing some configuration details.

This option masks the sensitive words in running configuration output with <removed> keyword.

```
switch# show running-config sanitized

!Command: show running-config sanitized
!Running configuration last done at: Wed Oct 12 09:14:54 2022
!Time: Wed Oct 12 13:52:55 2022

version 10.3(2) Bios:version 07.69

username admin password 5 <removed> role network-admin

copp profile strict
snmp-server user admin network-admin auth md5 <removed> priv aes-128 <removed> localizedV2key
rmon event 1 log trap <removed> description FATAL(1) owner PMON@FATAL
rmon event 2 log trap <removed> description CRITICAL(2) owner PMON@CRITICAL
rmon event 3 log trap <removed> description ERROR(3) owner PMON@ERROR
rmon event 4 log trap <removed> description WARNING(4) owner PMON@WARNING
rmon event 5 log trap <removed> description INFORMATION(5) owner PMON@INFO
--More--
```

Troubleshooting for POAP

The following is a list of known issues and suggestions while using POAP:

- Issue: POAP script execution fails immediately with no syslogs or output except for a "Script execution failed" statement.

Suggestion: Use the **python** *script-name* command on the server and make sure there are no syntax errors. The options dictionary is a Python dictionary so each entry must be comma separated and have the key or option and the value separated by a colon.

- Issue: A TypeError exception occurs at various places depending on the incorrectly used option.

  Suggestion: Some options use integers (for example, timeouts and other numeric values). Check the options dictionary for numeric values that are enclosed in quotes. Refer to the options list for the correct usage.

- Issue: POAP over USB is not finding the files that are present.

  Suggestion: Some devices have two USB slots. If you are using USB slot 2, you need to specify that as an option.

- Issue: Any issue with POAP.

  Suggestion: Abort POAP, and when the system boots up, run the **show tech-support poap** command, which displays POAP status and configuration.

Managing the POAP Personality

POAP Personality

The POAP personality feature, which is introduced in Cisco NX-OS Release 7.0(3)I4(1), enables user data, Cisco NX-OS and third-party patches, and configuration files to be backed up and restored. In previous releases, POAP can restore only the configuration.

The POAP personality is defined by tracked files on the switch. The configuration and package list in the personality file are ASCII files.

Binary versions are recorded in the personality file, but the actual binary files are not included. Because binary files are typically large, they are accessed from a specified repository.

The personality file is a .tar file, which would typically be extracted into a temporary folder. Here is an example:

```
switch# dir bootflash: 042516182843personality # timestamp name
46985 Dec 06 23:12:56 2015 running-config  Same as "show running-configuration" command.
20512 Dec 06 23:12:56 2015 host-package-list  Package/Patches list
58056 Dec 06 23:12:56 2015 data.tar  User Data
25    Dec 06 23:12:56 2015 IMAGEFILE  Tracked image metadata
```

Backing Up the POAP Personality

You can create a backup of the POAP personality either locally on the switch or remotely on the server. The personality backup taken from the switch should be restored only on a switch of the same model.

Note If you are using the Cisco scheduler feature for backups, you can configure it to also back up the POAP personality, as shown in the following example. For more information on the scheduler, see the *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*.

```
switch(config)# scheduler schedule name weeklybkup
switch(config-schedule)# time weekly mon:07:00
switch(config-schedule)# job name personalitybkup
switch(config-schedule)# exit
switch(config)# scheduler job name personalitybkup
switch(config-job)# personality backup bootflash:/personality-file ; copy
bootflash:/personality-file tftp://10.1.1.1/ vrf management
```

SUMMARY STEPS

1. **personality backup** [**bootflash:***uri* | **scp:***uri*]

DETAILED STEPS

	Command or Action	Purpose

Step 1	Required: **personality backup** [**bootflash:***uri* | **scp:***uri*]
	Example:
	`switch# personality backup bootflash:personality1.tar`
	Example:
	`switch# personality backup scp://root@2.1.1.1/var/lib/tftpboot/backup.tar`
		Creates a backup of the POAP personality.

Configuring the POAP Personality

You can specify whether the POAP personality should be derived from the running state of the system or the committed (startup) state.

SUMMARY STEPS

1. **configure terminal**
2. **personality**
3. **track** [**running-state** | **startup-state** | **data** *local-directories-or-files*]
4. **binary-location** *source-uri-folder*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Required: **configure terminal** **Example:** `switch# configure terminal switch(config)#`	Enters global configuration mode.

	Command or Action	Purpose

Step 2	Required: **personality**
	Example:
	`switch# personality`
	`switch(config-personality)#`
		Enters personality configuration mode.

Step 3	Required: **track** [**running-state** | **startup-state** | **data** *local-directories-or-files*]
	Example:
	`switch(config-personality)# track data bootflash:myfile1`
	Example:
	`switch(config-personality)# track data bootflash:user_scripts/*.py`
	Example:
	`switch(config-personality)# track data bootflash:basedir/*/backup_data`
		Specifies how the POAP personality is derived. The following options are available:
		• **running-state**—Captures the following information: the running configuration (as shown in the **show running-config** command), active Cisco NX-OS patches and third-party packages in the host system, and the image name (as shown in the **show version** command). This is the default option.
		• **startup-state**—Captures the following information: the startup configuration (as shown in the **show startup-config** command), committed Cisco NX-OS patches and third-party packages in the host system, and the image name (as shown in the **show version** command).
		• **data** *local-directories-or-files*—Specifies a directory or file to be backed up. You can enter this command multiple times to back up multiple directories and files. UNIX-style wildcard characters are supported. In the example, one folder and two directories are specified.
		Note Do not use this command to backup binary files in the bootflash and do not point to the entire bootflash.
		Note Guest Shell packages are not tracked.
		Note Signed RPMs (which require a key) are not supported. The POAP personality feature does not work with signed RPMs.

Step 4	Required: **binary-location** *source-uri-folder* **Example:** `switch(config-personality)# binary-location scp://remote-dir1/nxos_patches/`	Specifies the local or remote directory from which to pick up binary files when the POAP personality is restored. You can enter this command multiple times (in order of priority) to specify multiple locations.

Restoring the POAP Personality

During the POAP script execution phase, the personality module in the script restores the POAP personality, provided that the currently booted switch image is Cisco NX-OS Release 7.0(3)I4(1) or later. If necessary, upgrade the switch to the correct software image.

Note A personality restore is done with the same software image used for the personality backup. Upgrading to a newer image is not supported through the POAP personality feature. To upgrade to a newer image, use the regular POAP script.

Note If the personality script fails to execute for any reason (such as not enough space in the bootflash or a script execution failure), the POAP process returns to the DHCP discovery phase.

The restore process performs the following actions:

1. Untars and unzips the personality file in the bootflash.
2. Validates the personality file.
3. Reads the configuration and package list files from the personality file to make a list of the binaries to be downloaded.
4. If the current image or patches are not the same as specified in the personality file, downloads the binaries to the bootflash (if not present) and reboots with the correct image and then applies the packages or patches.
5. Unzips or untars the user data files relative to "/".
6. Copies the configuration file in the POAP personality to the startup configuration.
7. Reboots the switch.

POAP Personality Sample Script

The following sample POAP script (poap.py) includes the personality feature:

```
#md5sum="b00a7fffb305d13a1e02cd0d342afca3"
# The above is the (embedded) md5sum of this file taken without this line, # can be # created
 this way:
# f=poap.py ; cat $f | sed '/^#md5sum/d' > $f.md5 ; sed -i "s/^#md5sum=.*/#md5sum=$(md5sum
 $f.md5 | sed 's/ .*//')/" $f # This way this script's integrity can be checked in case you
 do not trust # tftp's ip checksum. This integrity check is done by /isan/bin/poap.bin).
# The integrity of the files downloaded later (images, config) is checked # by downloading
 the corresponding file with the .md5 extension and is # done by this script itself.

from poap.personality import POAPPersonality import os

# Location to download system image files, checksums, etc.
download_path = "/var/lib/tftpboot"
# The path to the personality tarball used for restoration personality_tarball =
"/var/lib/tftpboot/foo.tar"
# The protocol to use to download images/config protocol = "scp"
# The username to download images, the personality tarball, and the # patches and RPMs
during restoration username = "root"
# The password for the above username
password = "passwd754"
# The hostname or IP address of the file server server = "2.1.1.1"

# The VRF to use for downloading and restoration vrf = "default"
if os.environ.has_key('POAP_VRF'):
    vrf = os.environ['POAP_VRF']
```

```
# Initialize housekeeping stuff (logs, temp dirs, etc.) p = POAPPersonality(download_path,
 personality_tarball, protocol, username, password, server, vrf)

p.get_personality()
p.apply_personality()

sys.exit(0)
```