



Using Power On Auto Provisioning

This chapter contains the following sections:

- [Information About PowerOn Auto Provisioning, on page 1](#)
- [Guidelines and Limitations for POAP, on page 11](#)
- [Setting Up the Network Environment To Use POAP, on page 12](#)
- [Configuring a Switch Using POAP, on page 13](#)
- [Verifying the Device Configuration, on page 13](#)
- [Related Documents for POAP, on page 14](#)

Information About PowerOn Auto Provisioning

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

When a Cisco Nexus Series switch with the POAP feature boots and does not find the startup configuration, the switch enters POAP mode and checks for a USB device containing the configuration script file. If it finds one, it checks that device to see if it also contains the software image files and the switch configuration file.

If the switch does not find a USB device, or if the USB device does not contain the needed image files or switch configuration file, the switch also locates a DHCP server and bootstraps itself with its interface IP address, gateway, and DNS server IP addresses. The switch then obtains the IP address of a TFTP server or the URL of an HTTP server from which it downloads the necessary configuration files.



Note The DHCP information is used only during the POAP process if any configuration files are unavailable on the USB device.

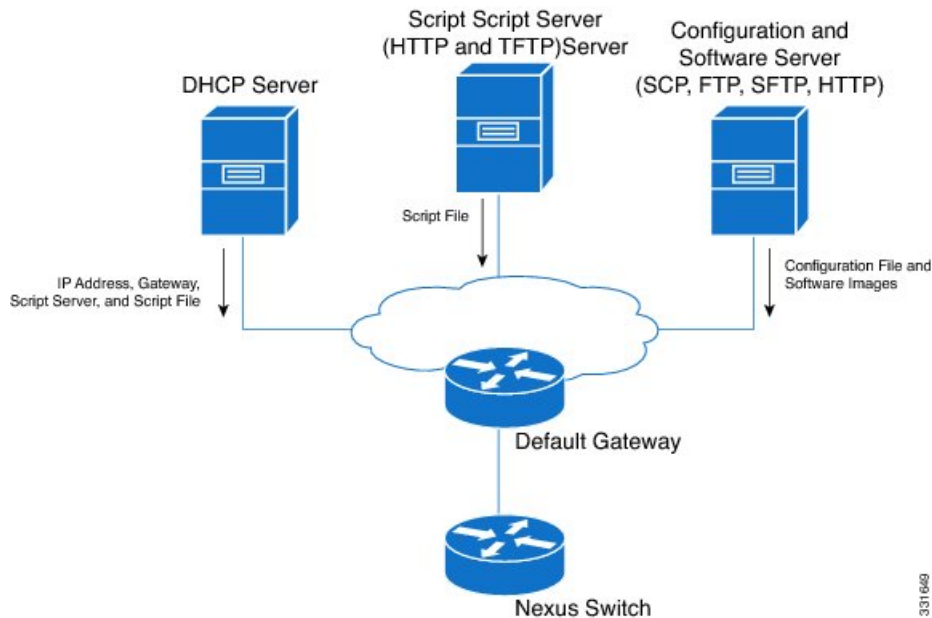
If the backup configuration file does not have the administrative username and the password, POAP causes a console lockout after completion. It is a mandatory step to add the username and the password in the configuration file.

Network Requirements for POAP

If a USB device containing the required installation files is not available, POAP requires the following network infrastructure:

- A DHCP server to bootstrap the interface IP address, gateway address, DNS server, and log server
- A TFTP or HTTP server containing the configuration script used to automate the software image installation and configuration process
- One or more servers containing the desired software images and configuration files

Figure 1: POAP Network Infrastructure



POAP Configuration Script

We provide sample configuration scripts that are developed using the Python programming language. You can customize these scripts to meet the requirements of your network environment.

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

For information about customizing this 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 all the 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 write erase and reload the device.

The following is a comprehensive list of POAP script options and can be specified to alter the POAP script behavior. When downloading from a server, 'hostname', 'username', and 'password' are required. For every mode except 'personality', the 'target_system_image' is also required. Required parameters are enforced by the script and the script will abort 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 as mentioned here will be used.

- **username**

The username to use when downloading files from the server. This is not required when using USB.

- **password**

The password to use when downloading files from the server. This is not required when using USB.

- **hostname**

The name or address of the server to download files from. This is not required when using USB.

- **mode**

Default is **serial_number**.

Use one of the following options:

1. **personality**

A method to restore the switch from a tarball.

2. **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`

3. **hostname**

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

4. **mac**

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

5. **raw**

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

6. **location**

The CDP neighbors are used to determine the configuration filename. The The format for using 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 the POAP interface is connected to. Example: `conf_remote-switch_Eth1_8.cfg`

- **required_space**

The required space in KB for that particular iteration of POAP. Default is 100000. For multi-step upgrades, specify the size of the last image in the upgrade path of the target image or images.

- **transfer_protocol**

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

- **config_path**

The path to the configuration file on the server. Example: /tftpboot. 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 is a required parameter for every mode except 'personality'. Default is "".

- **target_image_path**

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

- **target_kickstart_image**

The name of the kickstart image to download from the remote server. This is the kickstart image you get after POAP completes. Default is "".

- **destination_path**

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

- **destination_system_image**

The name for the destination system image file name. If not specified, the default will be the 'target_system_image' name.

- **destination_kickstart_image**

The name for the destination kickstart image file name. If not specified, the default will be the 'target_kickstart_image' name.

- **user_app_path**

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

- **disable_md5**

This is True if MD5 checking should be disabled. Default is False.

- **midway_system_image**

The name of the system 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. Default is "".

- **midway_kickstart_image**

The name of the kickstart 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. Default is "".

- **usb_slot**

The USB slot number to use for USB POAP. Default is 1.

- **source_config_file**

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

- **vrf**

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

- **destination_config**

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

- **split_config_first**

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

- **split_config_second**

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

- **timeout_config**

The timeout in seconds, for copying the configuration file. Default is 120. For non-legacy images, this option is not used and the POAP process will timeout. 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. Default is 2100. For non-legacy images, this option is not used and the POAP process will timeout. 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_kickstart**

The timeout in seconds, for copying the kickstart image. Default is 900. For non-legacy images, this option is not used and the POAP process will timeout. 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. Default is 900. For non-legacy images, this option is not used and the POAP process will timeout. 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. Default is 900. For non-legacy images, this option is not used and the POAP process will timeout. 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 to download the personality tarball from. Once the tarball is downloaded and the personality process is started, personality will download all files in the future from locations specified inside the tarball configuration. Default is /var/lib/tftpboot

- **source_tarball**

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

- **destination_tarball**

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

- **compact_image**

To copy the compact image set this value to "True" as shown below. The default value is "False".

```
"compact_image" : True
```

For more details on the compact image, see the [Compact Image for Cisco Nexus 3000](#) section in Cisco Nexus 3000 Series NX-OS Software Upgrade and Downgrade Guide.

POAP Upgrade Path

On a Cisco Nexus 3000 Series switch, moving from one software image to another requires intermediate upgrade steps that are handled as part of POAP. You need to know the upgrade path you require to avoid confusions while POAP loads multiple software images.

For the N3K-C3048TP-1GE-SUP platform, if you are using software versions older than Cisco NX-OS Release 5.0(3)U5(1), upgrade to Cisco NX-OS Release 5.0(3)U5(1) first, then upgrade to Cisco NX-OS Release 6.0(2)U6(2a), and finally upgrade to 6.0(2)U6(7) or a latest release.

If you like to skip this extra step of upgrading to Cisco NX-OS Release 6.0(2)U6(2a), you need to use the **midway_system_image** and **midway_kickstart_image** options.

The required software images for each step are:

- Cisco Release 5.0(3)U5(1) - n3000-uk9-kickstart.5.0.3.U5.1.bin, n3000-uk9.5.0.3.U5.1.bin
- Cisco Release 6.0(2)U6(2a) - n3000-uk9-kickstart.6.0.2.U6.2a.bin, n3000-uk9.6.0.2.U6.2a.bin
- Cisco Release 6.0(2)U6(7) - n3000-uk9-kickstart.6.0.2.U6.7.bin, n3000-uk9.6.0.2.U6.7.bin

For all other Cisco Nexus 3xxx platforms, if you are using software versions older than Cisco NX-OS Release 5.0(3)U5(1), upgrade to Cisco NX-OS Release 5.0(3)U5(1) first, then upgrade to 6.0(2)U6(7) or a latest release.

However, if you have a mix of N3K-C3048TP-1GE-SUP platform and other devices, you need to use the default upgrade path (no options needed) for the N3K-C3048TP-1GE-SUP platform, and a different script with the **midway** options set for your other devices. You can use the Vendor Class ID to distinguish between the type of devices.

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.

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. For more details, see the options under [Using the POAP Script and POAP Script Options, on page 2](#).

POAP Process

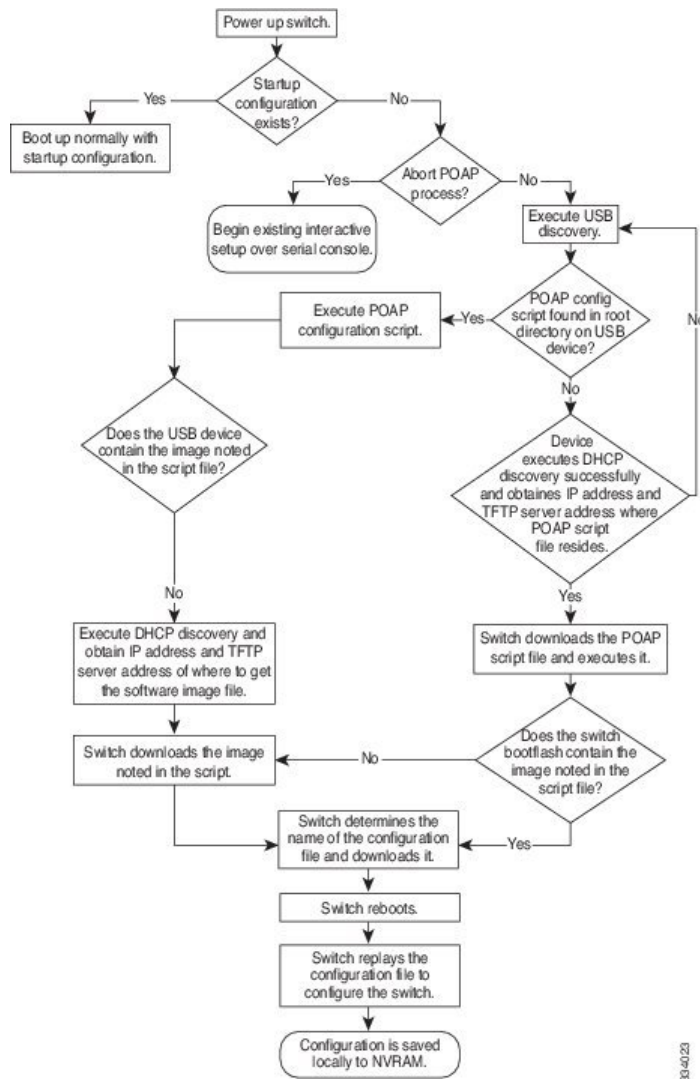
The POAP process has the following phases:

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 2: POAP Process



Power-Up Phase

When you power-up a switch for the first time, it loads the software image installed at manufacturing and tries to find a configuration file to apply after the switch boots. When no configuration file is found, POAP mode starts.

During startup, a prompt appears asking if you want to abort POAP and continue with 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 Layer 2 mode, which ensures that the device does not participate in any Layer 2 forwarding.

USB Discovery Phase

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

If the configuration script file is found on a USB device, POAP begins running the configuration script. If the configuration 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 all of the active interfaces (including the mgmt interface) soliciting DHCP offers from the DHCP server or servers. The DHCP client on the Cisco Nexus switch uses the switch serial number or its MAC address 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 file name, 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 DHCP negotiation.

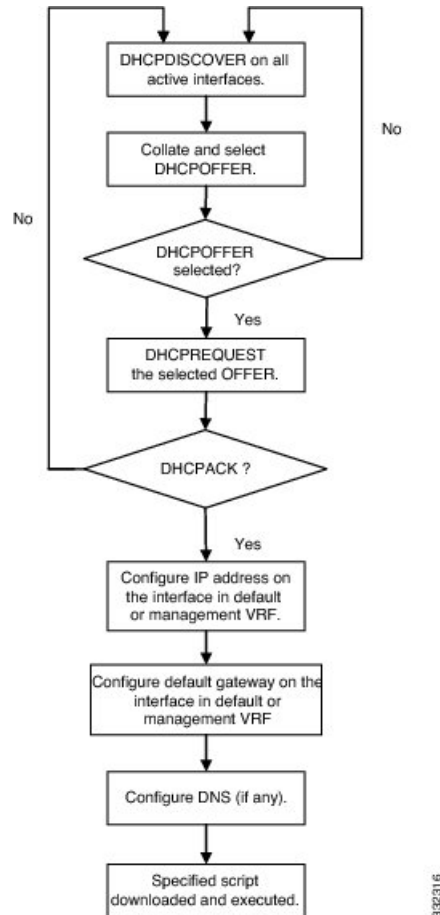
The DHCP discover message also mandates some of the options and solicits these options from the DHCP server after receiving the DHCP OFFER from the DHCP server.

- Option 66 (TFTP server name) or Option 150 (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.
- IP address
- Default gateway
- Option 67 (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, an offer is randomly chosen. 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 there is a failure 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. The POAP process is reinitiated until it succeeds or you manually abort the POAP process.

Figure 3: DHCP Discovery Phase



Script Execution Phase

Once the device has bootstrapped itself using the information in the DHCP acknowledgement, the switch downloads the script file from the TFTP server or the HTTP 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 currently running 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 one 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 clears the existing configuration if any, and applies (replays) the configuration on the upgraded software image. Afterward, the switch copies the running configuration to the startup configuration.

Guidelines and Limitations for POAP

- The Cisco Nexus switch software image must support POAP for this feature to function.
- POAP can be triggered even when the startup-config is present using the **boot poap enable** command.
- If a LACP Layer 3 port-channel is configured on an uplink device connected to the Cisco Nexus device that is being bootstrapped using POAP, the port-channel is not active because all the member links are in a suspended state. Therefore, the Cisco Nexus device that is being bootstrapped using POAP cannot reach the DHCP server or any other infrastructure device needed for POAP. To work around this issue, configure a static L3 port-channel on the uplink device connected to the Cisco Nexus device that is being bootstrapped using POAP.
- If you use POAP to bootstrap a Cisco Nexus device that is a part of a vPC pair using static port-channels on the VPC links, the Cisco Nexus device activates all of its links upon POAP startup. 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 connected to the Cisco Nexus device, and the traffic would be lost.

To work around this issue, you can configure 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 Series 7000 device through a LACP port-channel, the Cisco Nexus 7000 Series device 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 7000 Series device to not suspend its member ports 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.
- 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.
- The Scheduler configuration cannot be replayed using POAP. The reason that the Scheduler configuration cannot be replayed is that it is associated with the user (for example "admin") that was logged in when the Scheduler configuration was created. Because the configuration replay using POAP is not associated with any specific user, the scheduler configuration cannot be replayed and fails.

Instead of configuring the Scheduler, configure the Embedded Event Manager (EEM). An EEM configuration can be downloaded and replayed using POAP.

- You can bypass password and basic POAP configuration by using the **skip** option at the POAP prompt.

When you use the **skip** option, no password will be configured for the **admin** user. The **copy running-config startup-config** command will be blocked until a valid password is set for the **admin** user.

- The certificates (for example SSL) or configuration that are needed to be applied to the switch should be present in the configuration file.
- The syntax of the poap_script.py file should be validated using any python validation tool before using the file for POAP. Otherwise, if the poap_script.py file is edited and has a syntax error, the POAP process will exit without giving an error.
- 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 3000 switches.

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.

Setting Up the Network Environment To Use POAP

Procedure

-
- 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 configuration 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. For more information, see the [Setting Up the Network Environment To Use POAP, on page 12](#) section immediately preceding this section.

Procedure

- 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.
- Step 4** (Optional) If you want to bypass password and basic POAP configuration, enter **skip**. POAP is aborted and password configuration is skipped.
-

What to do next

Verify the configuration.

Verifying the Device Configuration

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

Command	Purpose
show running-config	Displays the running configuration.
show startup-config	Displays the startup configuration.

For detailed information about the fields in the output from these commands, see the Cisco Nexus command reference for your device.

Related Documents for POAP

Related Topic	Document Title
Configuration Script	<i>Cisco Nexus 3000 Series NX-OS Python API Reference Guide</i>
DHCP Options and BOOTP Vendor Extensions	RFC2132— http://tools.ietf.org/html/rfc2132
TFTP Server Address Option for DHCPv4	RFC5859— http://tools.ietf.org/html/rfc5859