



# CISCO WLAN POLLER GUI

## User Guide

Document Version	v5.07
Application Version	WLAN Poller GUI v5.07
Date	June 2026
Classification	Customer Visible
Author	Wireless BU

# 1. Overview

The Cisco WLAN Poller GUI (WPGui) is a PySide6-based network automation tool designed for Cisco enterprise wireless environments. It enables network engineers to perform various operations like bulk CLI operations across Wireless LAN Controllers (WLC) and Access Points (AP) via SSH, collect structured outputs, validate firmware, and perform image upgrades with built-in safety checks.

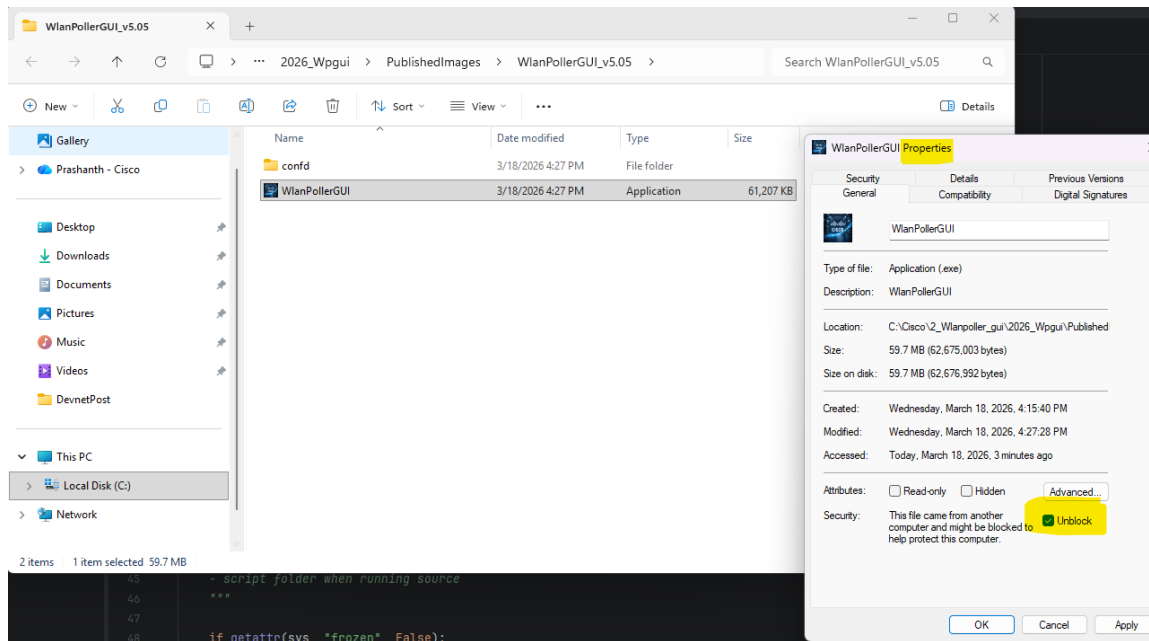
This document covers the advanced use cases available in v5.07 focusing on features introduced or significantly enhanced in this release. Each use case includes a detailed step-by-step procedure, expected output, and relevant safety alerts.

Wpgui Image Name for Windows & MAC Book:

System (Win/Mac)	Wpgui Image Name
------------------	------------------

System (Win/Mac)	Wpgui Image Name	Version
Windows	WpGui_Windows_Ver507.zip	V5.07
MAC Book X86	WpGui_MacX86_Ver_507.zip	V5.07
Mac Book Arm64 M1/M2/M3/M4+	WpGui_MacARM64_Ver507.zip	V5.07

Unzip the file & execute the wpgui application file. In Windows if you run into any errors to launch the application, please unblock & try again as shown below under properties.



## Features added in V5.07:

1. AP9136 not detected by wpgui bug fixed
2. Encrypted WLC and AP Password stored in the text file now
3. Iteration & Repetition feature
4. Cron Job Feature to run at periodic intervals
5. Multiple WLC support

## 2. Use Case Index

UC ID	Use Case Title	Applicable Mode
UC-01	AP Flash Checker — Flash Vulnerability Decoder	AP Only
UC-02	WLC & AP Mode — Dual-Stage Polling	WLC & AP
UC-03	WLC Only Mode — Controller-Only CLI Polling	WLC Only
UC-04	AP Only Mode — Direct SSH Polling	AP Only
UC-05	Device Reload via Confirm Injection (%reload%)	AP Only / WLC & AP
UC-06	Timed Delay Between Commands (sleep_N / pause_N)	AP Only / WLC & AP
UC-07	AP Image Download via archive download-sw	AP Only / WLC & AP
UC-08	Image-to-Model Mismatch Detection Alert	AP Only / WLC & AP
UC-09	Pre-Run Image Verification Warning (Step 4 Proceed Alert)	AP Only / WLC & AP
UC-10	Credential Encryption — XOR + Base64 Password Protection	AP Only / WLC & AP / WLC Only
UC-11	Log Search Analyzer - Parser	

## UC-01 AP Flash & Vulnerability Checker

---

### Description

The AP Flash Checker (Flash Vulnerability Decoder) is a built-in workflow in WPGui that connects to each AP via SSH and runs a set of diagnostic commands to read flash contents, decode the installed image, and identify vulnerable firmware versions. It is selected in Step 3 as a workflow option and requires no manual command entry — WPGui auto-generates the necessary commands based on the selected workflow.

### Use Case Attributes

<b>Operation Mode</b>	AP Only (or WLC & AP)
<b>Trigger</b>	Step 3 Select 'AP Flash Checker' from the workflow dropdown
<b>AP Commands</b>	Auto-generated: show flash:, show version, dir flash:
<b>WLC Connection</b>	Not required
<b>Output</b>	Per-AP .log file + vulnerability summary table in Step 7 Run Log

### Prerequisites

- AP list file (.txt or .csv) uploaded in Step 1. Format: IP Model Name (space or comma separated, one AP per line).
- AP SSH credentials (username, password, enable password) configured in Step 2.

### Step by Step Procedure

<b>1</b>	Step 1: Set Operation Type to 'AP Only'. Click Browse to upload the AP list file.
<b>2</b>	Step 2: Enter AP username, password, and enable password. Click Save.
<b>3</b>	Step 3: Select 'AP Flash Checker' from the workflow dropdown. No command entry is required — commands are auto-generated.
<b>4</b>	Step 6: Review the run preview. Click 'Confirm and Start WlanPoller'.
<b>5</b>	Step 7: Monitor the Run Log. Each AP will display flash contents and a vulnerability verdict: SAFE, VULNERABLE, or UNKNOWN.

### Supported Access Point Models

9117	9171
9124	9174
9130	9176
9136	9178
9162	9179
9163	9172
9164	9166
9167	

# WLAN POLLER GUI

- Step1 Operation Type
- Step2 Credentials
- Step3 Workflow
- Step4 CLI Cmd List
- Step5 AP Filters
- Step6 Preview
- Step7 Run/Results
- Parser

## Step1 - Select Operation Type

Choose Operation Type:

AP Only

Upload AP List File (Format: AP Ip, AP Name) C:/Users/lenovo/Downloads/Interim\_SrcCode/confd/ap\_ip\_list\_all.txt

Browse

Total: 4 | Valid: 4 | Invalid: 0 | Duplicates: 0

Enter Credentials

# WLAN POLLER GUI

- Step1 Operation Type
- Step2 Credentials
- Step3 Workflow
- Step4 CLI Cmd List
- Step5 AP Filters
- Step6 Preview
- Step7 Run/Results
- Parser

## Step3 - Choose WorkFlow

Choose a WorkFlow

AP Flash Checker

Back

Proceed

## Run Log (CLI Output)

```

2026-03-13 16:11:46 | 192.168.0.211 | Ap2802Den | show boot
2026-03-13 16:11:46 | 192.168.0.192 | Ap9105_Tattva | show flash | i cnssdaemon.log
2026-03-13 16:11:46 | 192.168.0.212 | Ap9105 | show flash | i cnssdaemon.log
2026-03-13 16:11:47 | 192.168.0.212 | Ap9105 | show boot
2026-03-13 16:11:47 | 192.168.0.192 | Ap9105_Tattva | show boot
2026-03-13 16:11:47 | 192.168.0.122 | CW9166 | show flash
    
```

100%

### AP Table

AP Name	AP Model	AP IP	Status
CW9166	CW9166D1-B	192.168.0.122	Success
Ap9105_Tattva	C9105AXI-B	192.168.0.192	Success

### Vulnerable APs & Recovery Table

AP Name	AP Model	AP IP	Recovery

==== RESULT SUMMARY =====

```

Operation Type Selected in Step1: AP Only
Total number of APs Processed: 4
Success APs: 4
    
```

## WLAN POLLER GUI

Step1 Operation Type

Step2 Credentials

Step3 Workflow

Step4 CLI Cmd List

Step5 AP Filters

Step6 Preview

Step7 Run/Results

Parser

### Run Log (CLI Output)

```

=====
Vulnerability scan complete. Found: 0 vulnerable AP(s)
=====

==== PARSE / FLASH CHECK SUMMARY =====
Total Vulnerable APs Detected: 0
    
```

100%

### AP Table

AP Name	AP Model	AP IP	Status
Ap9105_Tattva	C9105AXI-B	192.168.0.192	Success
Ap9105	C9105AXI-B	192.168.0.212	Success

### Vulnerable APs & Recovery Table

AP Name	AP Model	AP IP	Recovery

==== RESULT SUMMARY =====

```

Operation Type Selected in Step1: AP Only
Total number of APs Processed: 4
Success APs: 4
    
```

Save Run Log

Export Vulnerable Table to Excel

View Logs (Open Folder)

Close



## UC-02 WLC & AP Mode Dual-Stage Polling

---

### Description

WLC & AP mode executes a two-stage operation: it first runs a set of CLI commands on the Wireless LAN Controller over SSH, then automatically fetches the AP list from the WLC and polls each AP individually. This is the most complete operation mode, suitable for monitoring commands, full-site audits, configuration pushes, and image upgrades across all APs registered on the WLC.

### Use Case Attributes

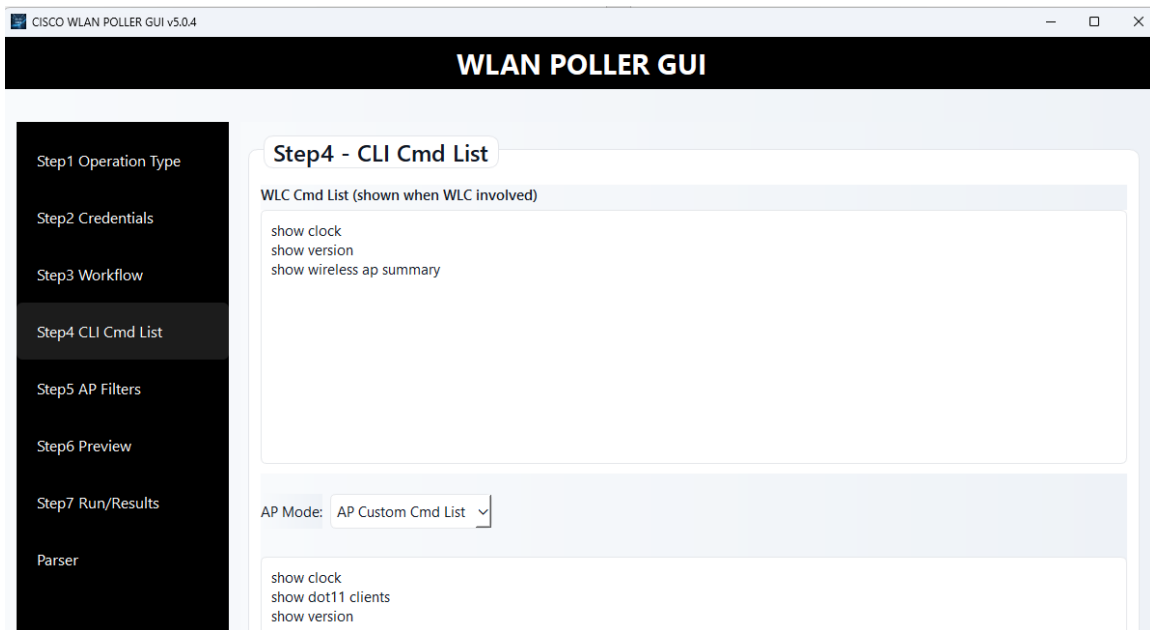
<b>Operation Mode</b>	WLC & AP
<b>Trigger</b>	Step 1 Select 'WLC & AP' from Operation Type dropdown
<b>WLC Commands</b>	Optional if provided, run before AP polling
<b>AP Commands</b>	Mandatory executed on each AP via SSH
<b>AP Filter</b>	Optional filter by Site Tag or AP Model Group
<b>Output</b>	Per-AP .log file + WLC _outputs.txt under data/YYYY/MM/DD/RUN-N/

### Prerequisites

- WLC IP, username, and password configured in Step 2.
- AP SSH credentials (username, password, enable password) configured in Step 2.
- WLC reachable on TCP port 22.
- WLC CLI account has privilege to run 'show ap summary'.

### Step by Step Procedure

1	Launch WPGui Application. On Step 1, set Operation Type to 'WLC & AP'.
2	Navigate to Step 2. Enter WLC IP, WLC username/password, AP username/password, and enable password. Click Save.
3	Navigate to Step 3. Select workflow: 'Custom CLI Commands'.
4	In Step 4. Enter WLC commands in the WLC Cmd List box. Enter AP commands in the AP Cmd List box. Click Save.
5	Navigate to Step 5. Optionally select a Site Tag or Model Group filter. Click Preview.
6	Review Step 6 Preview. Click 'Confirm and Start WlanPoller'.
7	Monitor Run Log and AP Table for per-AP status (Success / Fail).



# WLAN POLLER GUI

Step1 Operation Type

Step2 Credentials

Step3 Workflow

Step4 CLI Cmd List

Step5 AP Filters

Step6 Preview

Step7 Run/Results

Parser

## Run Log (CLI Output)

```
2026-03-13 15:15:49 | 192.168.0.122 | CW9166 | show version
[AP_UPDATE] ip=192.168.0.211 status=Success
[AP_UPDATE] ip=192.168.0.192 status=Success
[AP_UPDATE] ip=192.168.0.212 status=Success
[AP_UPDATE] ip=192.168.0.122 status=Success
[AP] Done. Success=4 Fail=0 Time=8s
```

100%

### AP Table

AP Name	AP Model	AP IP	Status
Ap2802Den	AIR-AP2802I-B-K9	192.168.0.211	Success
Ap9105_Tattva	C9105AXI-B	192.168.0.192	Success

### Vulnerable APs & Recovery Table

AP Name	AP Model	AP IP	Recovery
---------	----------	-------	----------

### ==== RESULT SUMMARY =====

Operation Type Selected in Step1: WLC & AP  
WLC IP address: 192.168.0.100  
Total number of APs Discovered: 4

Save Run Log

Export Vulnerable Table to Excel

View Logs (Open Folder)

Close

## UC-03 WLC Only Mode — Controller-Only CLI Polling

### Description

WLC Only mode connects exclusively to the Wireless LAN Controller over SSH and executes a defined set of CLI commands. No AP polling is performed. This mode is ideal for WLC-level diagnostics, configuration audits, bulk command execution on the controller, or any operation that does not require touching individual Access Points.

### Use Case Attributes

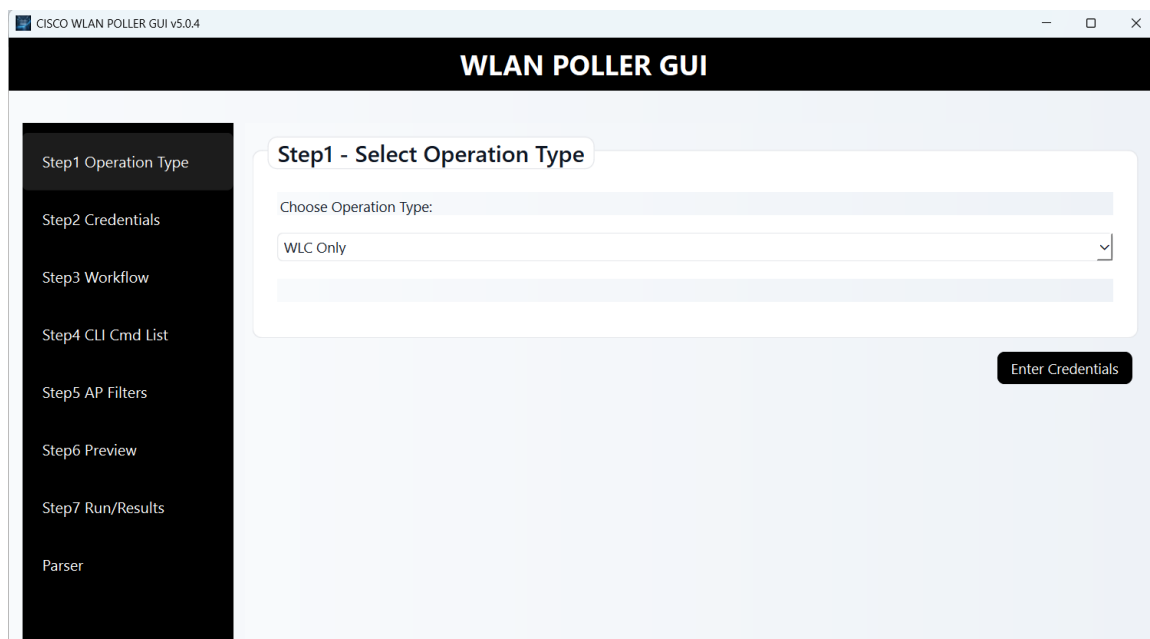
<b>Operation Mode</b>	WLC Only
<b>Trigger</b>	Step 1 — Select 'WLC Only' from the Operation Type dropdown
<b>WLC Commands</b>	Mandatory — entered in Step 4 WLC Cmd List box
<b>AP Connection</b>	Not required
<b>Output</b>	WLC_outputs.txt under data/YYYY/MM/DD/RUN-N/

### Prerequisites

- WLC IP, username, and password configured in Step 2.
- WLC reachable on TCP port 22.

### Step by Step Procedure

1	Step 1: Set Operation Type to 'WLC Only'.
2	Step 2: Enter WLC IP, WLC username, and WLC password. Click Save.
3	Step 3: Select workflow: 'Custom CLI Commands'.
4	Step 4: Enter WLC commands in the WLC Cmd List box. Click Save.
5	Step 6: Review the run preview. Click 'Confirm and Start WlanPoller'.
6	Step 7: Monitor the Run Log. WLC output is saved to data/YYYY/MM/DD/RUN-N/hostname_outputs.txt.



CISCO WLAN POLLER GUI v5.0.4

## WLAN POLLER GUI

- Step1 Operation Type
- Step2 Credentials
- Step3 Workflow
- Step4 CLI Cmd List
- Step5 AP Filters
- Step6 Preview
- Step7 Run/Results
- Parser

### Step4 - CLI Cmd List

WLC Cmd List

```
show clock
show ap summary
```

BackSaveProceed

## WLAN POLLER GUI

### Run Log (CLI Output)

```
PollerWorker: engine created, starting operation
[WLC] Checking SSH port on 192.168.0.100...
[WLC] Connecting to 192.168.0.100 ...
[WLC] Running: show clock
[WLC] Running: show ap summary
[WLC] Done. Output: C:\Users\lenovo\Downloads\Interim_SrcCode\data\2026\03\13\RUN33\WLC_outputs.txt
```

**100%**

==== RESULT SUMMARY =====

```
Operation Type Selected in Step1: WLC Only
WLC IP address: 192.168.0.100
Time taken: 1s
writing outputs to the folder 'data' to the file named WLC_outputs.txt
```

# WLAN POLLER GUI

Step1 Operation Type

Step2 Credentials

Step3 Workflow

Step4 CLI Cmd List

Step5 AP Filters

Step6 Preview

Step7 Run/Results

Parser

## Step1 - Select Operation Type

Choose Operation Type:

AP Only

Upload AP List File (Format: AP Ip, AP Name) rs/lenovo/Downloads/Interim\_SrcCode/confd/ap\_ip\_list\_all.txt

Browse

Total: 4 | Valid: 4 | Invalid: 0 | Duplicates: 0

Enter Credentials

## UC-04 AP Only Mode — Direct SSH Polling

### Description

AP Only mode connects directly to a list of APs via SSH using credentials configured in Step 2. It does not connect to any WLC. The AP list is uploaded manually in Step 1 as a .txt or .csv file. This mode is the most flexible direct-SSH operation mode — it supports any AP reachable over TCP/22, regardless of whether a WLC is present.

### Use Case Attributes

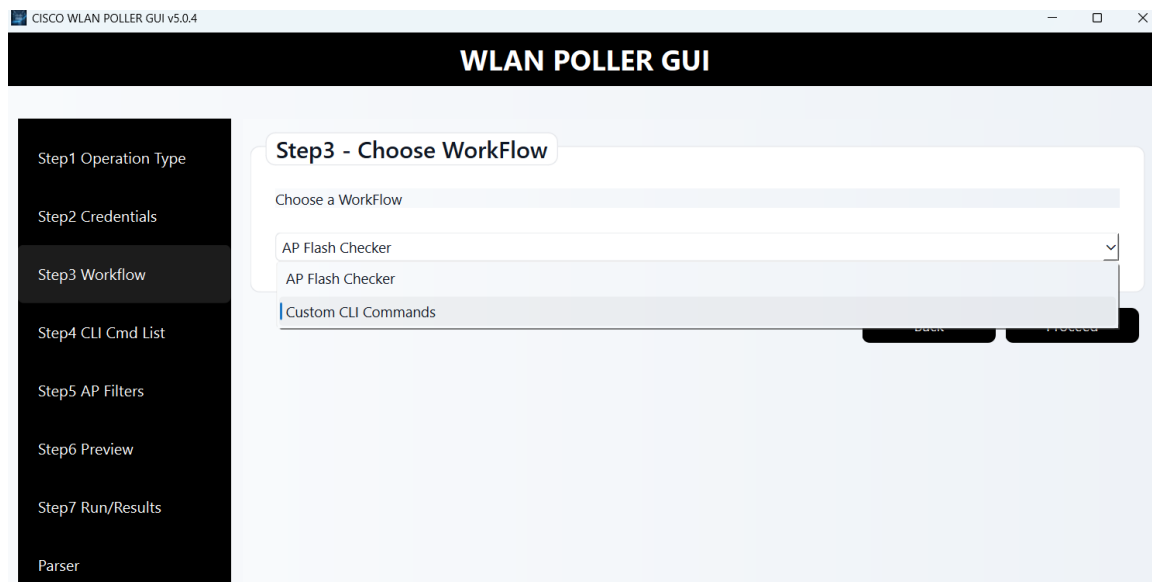
<b>Operation Mode</b>	AP Only
<b>Trigger</b>	Step 1 — Select 'AP Only' and upload AP list file
<b>AP List Format</b>	IP [Model] [Name] — space or comma separated, one per line
<b>AP Commands</b>	Mandatory — entered in Step 4 AP Cmd List box
<b>WLC Connection</b>	Not required
<b>Output</b>	Per-AP .log file under data/YYYY/MM/DD/RUN-N/

### Prerequisites

- AP list file (.txt or .csv) uploaded in Step 1.
- AP SSH credentials configured in Step 2.

### Step by Step Procedure

1	Step 1: Set Operation Type to 'AP Only'. Click Browse and upload the AP list file.
2	Step 2: Enter AP username, password, and enable password. Click Save.
3	Step 3: Select workflow: 'Custom CLI Commands'.
4	Step 4: Enter AP commands in the AP Cmd List box. Click Save.
5	Step 6: Review the run preview. Click 'Confirm and Start WlanPoller'.
6	Step 7: Monitor the Run Log and AP Table for per-AP status.



## UC-05 Device Reload via Confirm Injection (%reload%)

### Description

WPGui supports automatic confirmation injection for interactive CLI commands that display a [confirm] or (yes/no) prompt. Any command wrapped in % delimiters (e.g. %reload%) is treated as a confirm-required command. WPGui sends the command over the SSH channel, waits 3 seconds, then automatically injects a 'y' keystroke to confirm. This allows reload, erase, write erase, and similar disruptive commands to be scripted safely.

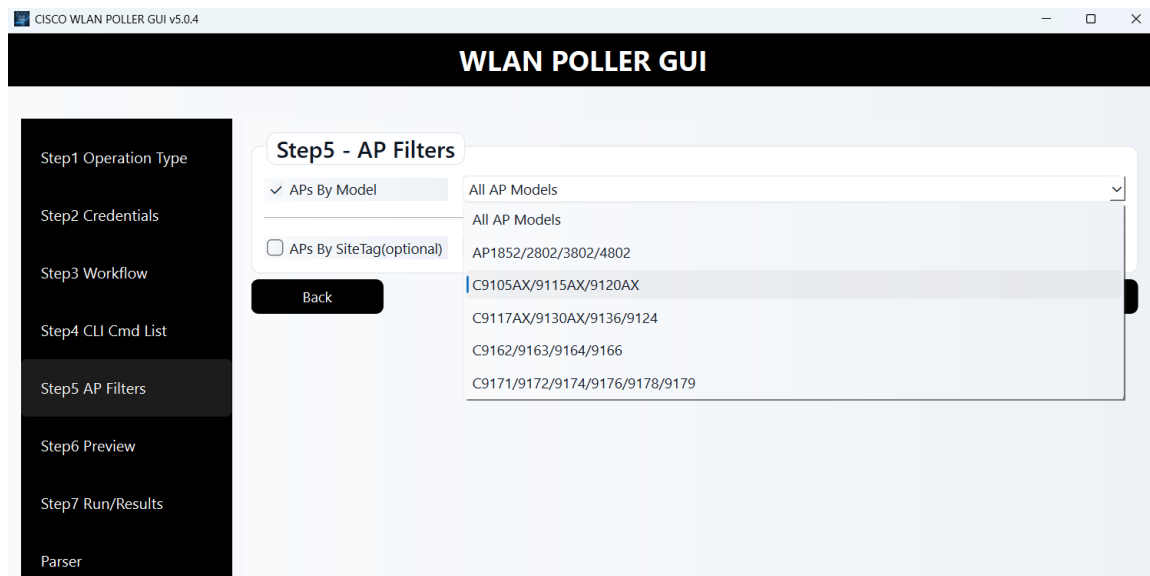
### Use Case Attributes

<b>Syntax</b>	%command% (e.g. %reload%)
<b>Behaviour</b>	Sends command, waits 3s, injects 'y' to confirm
<b>Applicable Modes</b>	AP Only, WLC & AP
<b>Commands supported</b>	reload, write erase, erase flash:, format flash:, and any other [confirm] or (yes/no) prompt commands

### Step by Step Procedure

1	Step 4: Enter %reload% (or any other confirm-required command wrapped in % delimiters) in the AP Cmd List box.
2	Step 6: Review the run preview. Verify the command list is correct.
3	Click 'Confirm and Start WlanPoller'. WPGui will automatically inject 'y' to confirm the reload prompt.
4	Step 7: Monitor the Run Log. Connection will drop after reload — this is expected. The AP will reconnect on next run.

**WARNING:** Use with caution. Confirm injection cannot be cancelled once sent. Verify AP list and command before starting.



CISCO WLAN POLLER GUI v5.0.4

## WLAN POLLER GUI

Step1 Operation Type

Step2 Credentials

Step3 Workflow

Step4 CLI Cmd List

Step5 AP Filters

Step6 Preview

Step7 Run/Results

Parser

### Step4 - CLI Cmd List

**AP Cmd List:**

AP Mode: AP Custom Cmd List

```
show ver
sleep_5
%reload%
```

Back
Save
Proceed

### Run Log (CLI Output)

```
2026-03-13 16:52:08 | 192.168.0.212 | Ap9105 | show ver
[SLEEP] Waiting 5s before next command...
2026-03-13 16:52:16 | 192.168.0.212 | Ap9105 | reload
[AP_UPDATE] ip=192.168.0.212 status=Success
[AP] Done. Success=1 Fail=0 Time=27s
```

100%

CISCO WLAN POLLER GUI v5.0.4

## WLAN POLLER GUI

Step1 Operation Type

Step2 Credentials

Step3 Workflow

Step4 CLI Cmd List

Step5 AP Filters

Step6 Preview

Step7 Run/Results

Parser

### Step4 - CLI Cmd List

**AP Cmd List:**

AP Mode: AP Custom Cmd List

```
show clock
Sleep_6
Pause(5)
```

Back
Save
Proceed

## UC-06 Timed Delay Between Commands (sleep\_N / pause\_N)

### Description

WPGui supports inserting a timed wait between CLI commands using the `sleep_N` or `pause_N` special command syntax. When WPGui encounters this token in the command list, it pauses execution for N seconds before sending the next command. This is useful for device operations that require a stabilisation period — for example, waiting after a reload command before polling again, or delaying between configuration pushes.

### Use Case Attributes

<b>Syntax</b>	<code>sleep_N</code> or <code>pause_N</code> where N is an integer number of seconds
<b>Example</b>	<code>sleep_30</code> — pauses for 30 seconds
<b>Maximum Value</b>	3600 seconds (1 hour). Values above 3600 are rejected with a log warning.
<b>Applicable Modes</b>	AP Only, WLC & AP
<b>Behaviour if N &gt; 3600</b>	Command is skipped with a warning logged. No delay is inserted.

### Step by Step Procedure

1	Step 4: Add <code>sleep_N</code> or <code>pause_N</code> as a line in the AP Cmd List between two commands.
2	Step 6: Review the run preview. The sleep command will be listed in the AP section.
3	Click 'Confirm and Start WlanPoller'. WPGui will log <code>[SLEEP] Waiting Ns...</code> at the appropriate point.
4	Step 7: Monitor the Run Log for the <code>[SLEEP]</code> message confirming the delay is active.

## WLAN POLLER GUI

### Run Log (CLI Output)

```
[SLEEP] Waiting 6s before next command...
[SLEEP] Waiting 6s before next command...
2026-03-13 16:58:02 | 192.168.0.122 | CW9166 | show clock
[SLEEP] Waiting 6s before next command...
[SLEEP] Waiting 5s before next command...
[SLEEP] Waiting 5s before next command...
[SLEEP] Waiting 5s before next command...
```

100%

#### AP Table

AP Name	AP Model	AP IP	Status
Ap2802Den	AIR-AP2802I-B-K9	192.168.0.211	Success
Ap9105	C9105AXI-B	192.168.0.212	Success

## UC-07 AP Image Download via archive download-sw

### Description

WPGui supports AP firmware image download using TFTP, SFTP, or SCP protocols. When an AP CLI command contains sftp:// or scp://, WPGui routes the command through a specialised interactive handler that monitors the AP's output channel in real-time, detects transfer progress symbols (!!!! or ....), automatically injects SFTP/SCP credentials if prompted, and detects completion or failure conditions. This handler replaces the standard send\_command path for image transfer commands only.

### Use Case Attributes

<b>Supported Protocols</b>	TFTP, SFTP, SCP
<b>Interactive Handler Trigger</b>	sftp:// or scp:// detected in the AP command string
<b>Credential Injection</b>	SFTP/SCP username and password injected automatically from confd/config.ini [FTP] section
<b>Progress Display</b>	!!!! and .... symbols from AP output shown in Run Log
<b>Completion Detection</b>	archive done / archive download completed / image transfer complete / error / failed / AP prompt returned
<b>Timeout</b>	3600 seconds per command

### Interactive Handler Behaviour

<b>username: prompt detected</b>	Injects ftp_user from [FTP] section
<b>password: prompt detected</b>	Injects ftp_pasw from [FTP] section
<b>[confirm] prompt detected</b>	Injects newline (Enter)
<b>(yes/no) prompt detected</b>	Injects 'yes'
<b>error / failed / no such file</b>	Failure — loop exits, last line logged as result
<b>permission denied / connection refused / timed out</b>	Failure — loop exits, connection error logged

### Step by Step Procedure

<b>1</b>	Step 3: Select workflow 'Custom CLI Commands'.
<b>2</b>	Step 4: Enter the archive download-sw command with the appropriate protocol and image path.
<b>3</b>	Step 6: Confirm the pre-run image verification dialog (UC-10). Click 'Confirm and Start WlanPoller'.
<b>4</b>	Step 7: Monitor the Run Log. WPGui loops on AP output and logs the [IMAGE DOWNLOAD] Result line on completion.

### Expected Run Log Output — Success

```
2026-03-12 17:46:05 | 192.168.0.212 | Ap9105 | archive download-sw /no-reload sftp://...
!!!!!!!!!!!!!!!!!!!!!!
.....
!!!!!!!!!!!!!!!!!!!!!!
[IMAGE DOWNLOAD] Result: archive done
```

### Expected Run Log Output — Failure

```
2026-03-12 17:46:05 | 192.168.0.212 | Ap9105 | archive download-sw /no-reload sftp://...
.
[IMAGE DOWNLOAD] Result: Image transfer failed
```

# WLAN POLLER GUI

Step1 Operation Type

Step2 Credentials

Step3 Workflow

Step4 CLI Cmd List

Step5 AP Filters

Step6 Preview

Step7 Run/Results

Parser

## Step4 - CLI Cmd List

WLC Cmd List (shown when WLC involved)

AP Mode: AP Image Download

```
archive download-sw /no-reload sftp://192.168.0.24/C:/Users/sftpuser/ap3g3-k9w8-tar.17_15_5_36.tar
```

### AP Image Download Settings

Protocol: SFTP

SFTP Username: sftpuser

SFTP Password: .....

Back

Save

Proceed

# WLAN POLLER GUI

## Run Log (CLI Output)

```
.  
[IMAGE DOWNLOAD] Result: Archive done.  
[AP_UPDATE] ip=192.168.0.211 status=Success  
[AP] Done. Success=1 Fail=0 Time=105s
```

100%

### AP Table

AP Name	AP Model	AP IP	Status
Ap2802Den	AIR-AP2802I-B-K9	192.168.0.211	Success

### Vulnerable APs & Recovery Table

AP Name	AP Model	AP IP	Recovery
---------	----------	-------	----------

==== RESULT SUMMARY =====

```
Operation Type Selected in Step1: AP Only  
Total number of APs Processed: 1  
Success APs: 1
```

Save Run Log

Export Vulnerable Table to Excel

View Logs (Open Folder)

Close

```
Directory /tmp/nthevents not found.  
status 'upgrade.sh: Script called with args:[ACTIVATE]'  
do ACTIVATE, part2 is active part  
status 'upgrade.sh: mount ubifs /dev/ubivol/part1 as /bootpart, status=0'  
status 'upgrade.sh: Verifying image signature in part1'  
status 'upgrade.sh: status 'Successfully verified image in part1.'  
status 'upgrade.sh: activate part1, set BOOT to part1'  
status 'upgrade.sh: AP primary version after reload: 17.15.5.36'  
status 'upgrade.sh: AP backup version after reload: 17.12.6.28'  
Successfully setup AP image.  
Archive done.
```

Ap2802Den#

## UC-08 Image-to-Model Mismatch Detection Alert

### Description

WPGui validates the firmware image filename in the archive download-sw command against the AP model in the AP list. If the image prefix does not match the expected prefix for the AP model, an [IMAGE MISMATCH] warning is emitted in the run log and written to the per-AP output file before the transfer is attempted. The transfer still proceeds — it is a warning, not a blocker.

### AP Model to Image Prefix Mapping

AP1852	ap1g3 — e.g. ap1g3-k9w8-tar.152-4.JA1.tar
AP2702	ap3g2 — e.g. ap3g2-k9w8-tar.153-3.JJ8.tar
AP2802 / AP3802 / AP4802	ap3g3 — e.g. ap3g3-k9w8-tar.17_15_5_36.tar
AP9105 / AP9115 / AP9120	ap1g7 — e.g. ap1g7-k9w8-tar.17_12_6_28.tar
AP9117 / AP9130 / AP9136 / AP9162 / AP9179	ap1g8 — e.g. ap1g8-k9w8-tar.17_14_1.tar

### Example Mismatch Run Log Output

```
[IMAGE MISMATCH] Model C9105AXI-B (AP9105) expects image prefix 'ap1g7'
but command uses 'ap3g3'. File: ap3g3-k9w8-tar.17_15_5_36.tar

# Transfer still attempted - result will typically be:
[IMAGE DOWNLOAD] Result: upgrade.sh: ERROR: Image type mismatch. Expected:ap1g7 Got:ap3g3
```

**NOTE:** The mismatch detection fires before the transfer command is sent. Even though the transfer proceeds, the AP itself will also report a mismatch error during extraction. This alert serves as an early warning so the operator can cancel and correct the image filename before wide-scale deployment.

### Upgrading ...

```
status 'upgrade.sh: Script called with args:[NO_UPGRADE]'
```

```
do NO_UPGRADE, part2 is active part
```

```
status 'upgrade.sh: Script called with args:[-c PREDOWNLOAD]'
```

```
do PREDOWNLOAD, part2 is active part
```

```
status 'upgrade.sh: Creating before-upgrade.log'
```

```
status 'upgrade.sh: Start doing upgrade arg1=PREDOWNLOAD arg2=,from_cli arg3= ...'
```

```
status 'upgrade.sh: Using image /tmp/cli_part.tar on ax-bcm32 ...'
```

```
status 'upgrade.sh: ERROR: Image type mismatch. Expected:ap1g8 Got:ap3g3'
```

```
status 'upgrade.sh: Cleanup for do_upgrade...'
```

```
status 'upgrade.sh: /tmp/upgrade_in_progress cleaned'
```

```
status 'upgrade.sh: Cleanup tmp files ...'
```

## UC-9 Pre-Run Image Verification Warning (Step 4 Proceed Alert)

---

### Description

When the operator clicks Proceed on Step 4 and the AP command list contains an image download command (archive download-sw, sftp://, or scp://), or the AP Mode dropdown is set to 'AP Image Download', WPGui displays a modal warning dialog requiring explicit Yes confirmation before advancing to the next step. This is a last-chance checkpoint before the run configuration is locked and the image download is queued.

### Trigger Conditions

- AP Cmd List contains archive download-sw in any command line, OR
- AP Cmd List contains sftp:// or scp:// in any command line, OR
- AP Mode dropdown is set to 'AP Image Download'.

### Dialog Content

```
Title: Verify Image Before Proceeding

Message: Please double-check before continuing:
- AP Model in your AP list file
- Image filename in the command matches that model
Wrong image will cause transfer failure on the AP.
Are you sure you want to proceed?

Buttons: [ Yes ] [ No (default) ]
```

**NOTE:** The dialog defaults to 'No'. The operator must actively click 'Yes' to proceed. Clicking 'No' returns to Step 4, allowing the operator to review and correct the command or image filename.

### Step by Step Procedure

1	Step 4: Enter an AP command containing archive download-sw or sftp://. Click Proceed.
2	Read the dialog carefully. Verify the AP model and image filename match.
3	Click 'Yes' to proceed to Step 5 / Step 6, or 'No' to return to Step 4 and correct the configuration.

# WLAN POLLER GUI

Step1 Operation Type

Step2 Credentials

Step3 Workflow

Step4 CLI Cmd List

Step5 AP Filters

Step6 Preview

Step7 Run/Results

Parser

## Step4 - CLI Cmd List

### AP Cmd List:

AP Mode: AP Image Download

```
archive download-sw /no-reload sftp://192.168.0.24/C:/Users/sftpuser/ap3g3-k9w8-tar.17_15_5_36.tar
```

Verify Image Before Pro...

⚠ Please double-check before continuing:

- AP Model in your AP list file
- Image filename matches that model

Wrong image will cause transfer failure.

Proceed?

Yes No

### AP Image Download Settings

Protocol: SFTP

SFTP Username: SFTP username

SFTP Password: SFTP password

Back

Save

Proceed

## UC-10 Credential Encryption — XOR + Base64 Password Protection

### Description

WPGui automatically encrypts all credential fields (WLC password, AP password, AP enable password, and SFTP password) before writing them to `confd/config.ini`. Encryption is applied transparently by the `IniStore` class — the operator always enters and sees plain-text credentials in the GUI. The encrypted values are stored on disk using a two-step process: XOR byte-level obfuscation against an internal key, followed by Base64 encoding. On every read, values are automatically decrypted in memory before being passed to the SSH engine. No plain-text passwords are ever written to disk.

### Encryption Mechanism

<b>Algorithm</b>	XOR byte-level obfuscation + Base64 encoding
<b>Internal Key</b>	WlanPollerKey (fixed, embedded in <code>PollerEngine.py</code> )
<b>Prefix</b>	ENC:: — all encrypted values are prefixed to distinguish them from plain text
<b>Scope</b>	Any <code>config.ini</code> field whose key contains <code>pasw</code> , <code>password</code> , or <code>enable</code>
<b>Storage Location</b>	<code>confd/config.ini</code> under <code>[WLC]</code> , <code>[AP]</code> , and <code>[FTP]</code> sections
<b>Decryption</b>	Automatic on every <code>IniStore.get()</code> call — engine always receives plain text

### Encryption Flow

1	Operator enters credentials in Step 2 (WLC/AP passwords) or Step 4 (SFTP password). Values are displayed as masked fields in the GUI.
2	Operator clicks Save. GUI calls <code>IniStore.bulk_set()</code> which detects password fields by key name ( <code>pasw</code> / <code>password</code> / <code>enable</code> ).
3	<code>IniStore.encrypt_value()</code> runs: raw UTF-8 bytes are XOR'd against the repeating <code>WlanPollerKey</code> byte sequence, then Base64-encoded. The result is prefixed with <code>ENC::</code> and written to <code>config.ini</code> .
4	On the next application launch, <code>IniStore</code> reads <code>config.ini</code> . When <code>get()</code> is called for a password field, it detects the <code>ENC::</code> prefix, Base64-decodes the value, reverses the XOR, and returns the original plain-text string.
5	The decrypted value is passed directly to <code>ConnectHandler</code> (Netmiko) for SSH authentication. It is never written back to disk in plain text.

### config.ini Encrypted Format

```
[WLC]
wlc_ip = 192.168.1.1
wlc_user = admin
wlc_pasw = ENC::HxsYBgMRCxQHDg== <- encrypted

[AP]
ap_user = Cisco
ap_pasw = ENC::GRYFDxcSAhUB <- encrypted
ap_enable = ENC::GRYFDxcSAhUB <- encrypted

[FTP]
ftp_user = sftpuser
ftp_pasw = ENC::BxILEA8TCxB <- encrypted
```

### Prerequisites

- `confd/` folder must exist and be writable. WPGui creates it automatically on first launch.
- No manual action is required by the operator — encryption and decryption are fully automatic.
- If `config.ini` is copied to another machine, credentials will decrypt correctly because the XOR key is embedded in `PollerEngine.py`.

**NOTE:** XOR + Base64 is an obfuscation mechanism, not cryptographic-grade encryption. It prevents plain-text passwords from being visible in the config file and in log output, but it is not a substitute for OS-level file permissions. Ensure confd/config.ini is protected by appropriate file system access controls on the host machine.

## WLAN POLLER GUI

- Step1 Operation Type
- Step2 Credentials**
- Step3 Workflow
- Step4 CLI Cmd List
- Step5 AP Filters
- Step6 Preview
- Step7 Run/Results

### Step2 - WLC / AP Details

WLC IP Address	192.168.0.100
WLC Username	admin
WLC Password	••••••••
AP Username	admin
AP Password	••••••••
Enable Password	••••••••

Back Save Proceed

Saved

OK

Credentials saved to confd/config.ini

```
[WLC]
wlc_ip = 192.168.0.100
wlc_user = admin
wlc_pasw = ENC::FAUSDT9eXl8=

[AP]
ap_user = admin
ap_pasw = ENC::FgUGAjVeXl8=
ap_enable = ENC::FgUGAjVeXl8=

[FTP]
ftp_addr = 192.168.0.24
ftp_path = C:/Users/sftpuser
ftp_user = sftpuser
ftp_pasw = ENC::GhUyGiIAAgs1EzgWSGVf
ftp_proto = SFTP
scp_port = 22

[GENERAL]
last_ap_list_file = ap_ip_list.txt
```



## UC-11 — Log Search Analyzer

---

### Description

The Parser is a built-in analysis feature in WPGui that allows users to scan previously generated AP or WLC output logs for specific keywords or patterns. It helps engineers quickly locate relevant information such as errors, configuration parameters, or operational messages without manually opening each log file.

The Parser works by searching the log files generated during earlier WlanPoller runs and identifying files that contain the specified pattern. It is accessed from the Parser section and requires only a search keyword or regular expression to begin analysis.

### Use Case Attributes

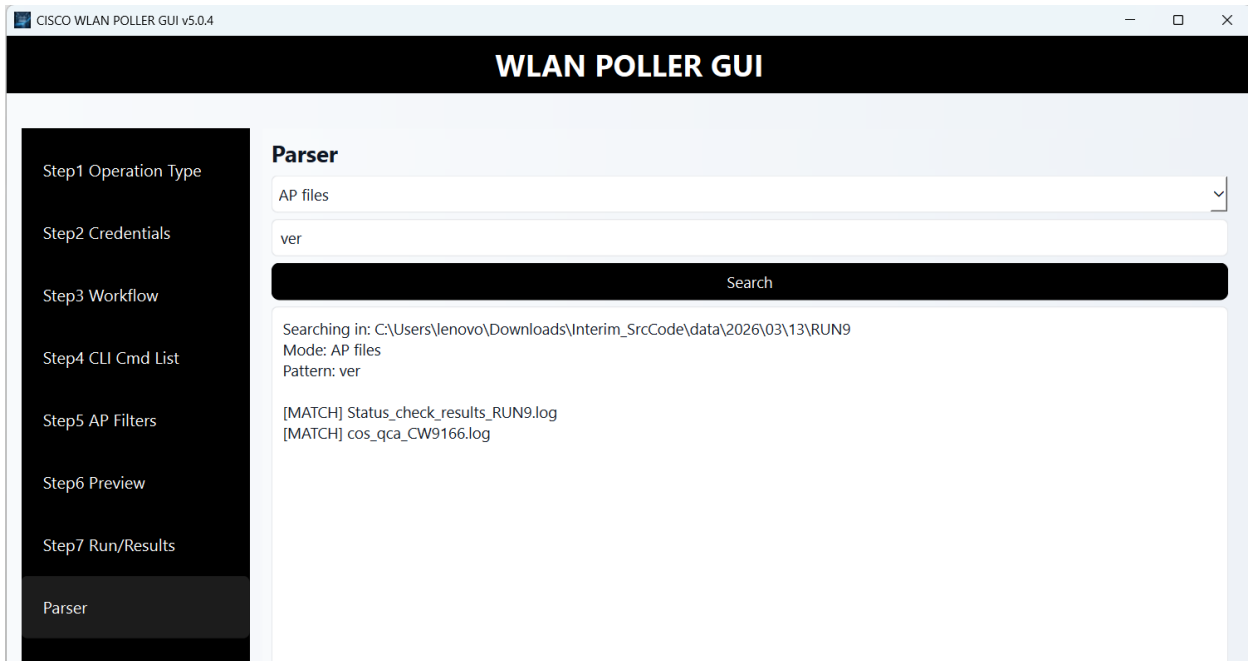
<b>Operation Mode</b>	Parser
<b>Trigger</b>	Open the Parser page and enter a search pattern
<b>Input</b>	Keyword or Regular Expression
<b>Search Scope</b>	AP log files or WLC log files from previous runs
<b>Output</b>	List of log files containing the matching pattern displayed in the Parser output window

### Prerequisites

- At least one WlanPoller run must have been completed to generate log files.
- Log files must exist in the data directory created during polling runs.
- A valid search keyword or regex pattern must be provided.

### Step By Step Procedure

<b>1</b>	Open the Parser section from the WPGui sidebar.
<b>2</b>	Select the search mode depending on the type of logs to analyze (AP logs or WLC logs).
<b>3</b>	Enter a search keyword or regular expression pattern in the search field.
<b>4</b>	Click Search to start the parsing operation.
<b>5</b>	View the results in the Parser Output window, which lists all log files where the pattern was found.



## Appendix A — Special Command Reference

Syntax	Example	Behaviour
sleep_N	sleep_30	Delay N seconds before next command. Max 3600s.
pause_N	pause_60	Identical to sleep_N.
%cmd%	%reload%	Send cmd + auto-inject 'y' confirmation after 3s.
sftp:// in cmd	archive download-sw ... sftp://...	Routed to interactive handler. Credentials auto-injected.
scp:// in cmd	copy scp://... flash:	Routed to interactive handler. Credentials auto-injected.
tftp:// in cmd	archive download-sw ... tftp://...	Normal CLI path. No credential injection. 3600s timeout.
http:// in cmd	archive download-sw ... http://...	Normal CLI path. No credential injection. 3600s timeout.

## Appendix B — Watchdog Timeout Reference

Condition	Timeout	Notes
Normal CLI run (no image download commands)	30 seconds	Watchdog fires warning if no log/progress seen.
Run contains archive download-sw, sftp://, or scp://	3600 seconds	Extended to allow for large image transfers.
_ap_send_command normal commands	180 seconds	Standard per-command read timeout.
_ap_send_command with archive download-sw	3600 seconds	Extended timeout for TFTP/HTTP image downloads.
run_command_interactive safety timeout	3600 seconds	Absolute fallback if no completion string detected.

Support Channel: You can reach out to wlanpollercisco.com mailer alias for any queries/questions related to this tool.

Cisco Systems, Inc. | WLAN Poller GUI v5.07 | User Guide | June 2026

