Eagle Eye Application Note - AN095

Utilizing the Eagle Eye Networks CLI Tool

2025-10-06 Revision 1.0

Target Audience

This document is intended for technical professionals responsible for deploying, maintaining, and integrating the Eagle Eye Networks Cloud VMS at scale. Our target audience includes System Integrators, IT Administrators, and software Developers who need to manage large device fleets and complex system configurations. This application note is designed to empower users to move beyond the limitations of the standard web interface by leveraging the Command Line Interface (CLI) for powerful, fast automation and precise troubleshooting. This guide is the essential reference for those who need to maximize operational efficiency through scripting, bulk management, and advanced diagnostics.

Introduction

The Eagle Eye Networks Command Line Interface (CLI) is engineered for our most technical users who demand speed, precision, and efficiency at scale. This tool is the workhorse for System Integrators, IT Administrators, and Developers, enabling them to bypass the typical web interface limitations. It allows an authorized user to instantly manage an entire account by scripting complex operations, from bulk configuring hundreds of camera settings and automating video retrieval to executing fleet-wide health diagnostics and troubleshooting. The CLI transforms complex, repetitive tasks into seamless automation, ensuring maximum operational efficiency across large deployments. Please note that this powerful feature is exclusively available to accounts utilizing the Enterprise Edition.

Eagle Eye Network's CLI Tool Allows:

- **Bulk System Edits and Management**: Perform mass configuration updates (e.g., changing resolution on 50 cameras), bulk user management, and automated firmware updates across all devices.
- **Bulk Downloads and Data Retrieval**: Script video downloads based on specific events or time ranges, and retrieve comprehensive system logs for debugging purposes.
- **Bulk Troubleshooting and Health Diagnostics**: Instantly check the health/status (CPU,

memory, uptime) of an entire fleet of bridges and cameras, and output the results into filterable formats, such as CSV.

First Time Setup

Follow the steps in this section to set up the CLI tool initially.

Step 1: Extract the Executable File

The process for extracting the (een) executable depends on your operating system.

For Linux/macOS:

There are multiple ways to extract the file from the compressed (.zip) format:

1. **Context Menu (GUI Method 1):** Right-click the compressed file and select **Extract Here** (to extract to the same directory) or **Extract to** (to select a new directory).
2. **Double-Click (GUI Method 2):** Double-click the compressed file to open a window, and select the **Extract** button to choose the desired directory.
3. **Terminal (CLI Method):** Open the terminal in the directory where the compressed file is stored and run:

```
None
unzip een.zip
```

For Windows:

1. Right-click on the compressed file.
2. Select **Extract All** from the context menu.
3. A window will prompt you to insert or confirm the directory path where you want the file extracted.
4. Click the **Extract** button to complete the process.

Step 2: Execute the Extracted File

Once the executable file (een) is extracted, you must map its directory path in your terminal. All subsequent commands must be prefixed with ./ to run the local executable.

1. Navigate to the directory containing the extracted (een) file.
2. Run the following command to check if the executable is working correctly:

```
None
./een --version
```

If successful, the output should display the current version number, confirming your setup is complete.

Essential Tips

CSV Outputs: Open all CSV file outputs in Google Sheets or Libre Office, as Excel has known formatting issues.

For help on setting up EEN's CLI command line tool, and further information, please visit our developer portal

- https://developer.eagleeyenetworks.com/docs/introduction

Syntax

This section contains an overview of the syntax used in the Eagle Eye Networks' CLI tool. It follows a simple structure:

```
None
./een <object> <command> <arguments>
```

- **NOTE:** You must remove "**<**' & "**>**' after inserting your value in the designated placeholders when entering your CLI commands.

- **Syntax:** Every command must begin with the executable file: `./een`
- **Help:** Use the built-in help function to explore specific commands:
  - For general help: `./een help`
  - For help on a specific object (e.g., cameras): `./een camera --help`
- **No Prompt:** To run commands that typically require a confirmation (e.g., a deletion), you can use the `--no-prompt` option. Be careful, as this will overwrite or execute without confirmation.
- **Verbosity:** For detailed output, use the `--verbose` flag along with your command.

- **Time Format:** When specifying times (like for downloads), ensure you use a supported format, such as `YYYYMMDDHHMMSS.000` or `YYYY-MM-DDTHH:MM:SS`

Common CLI Commands

For additional commands, refer to our developer portal

- https://developer.eagleeyenetworks.com/docs/introduction

TROUBLESHOOTING		
Goal	**Example Command**	**Description**
Ping a Camera	`./een camera ping <camera_id>`	Sends a test packet to the camera to confirm basic network connectivity and latency.
Run a Bridge Health Check	`./een bridge telemetry <bridge_id>`	Fetches performance data, including CPU utilization, memory usage, and storage health, for a specific bridge.
List All Users and Sites	`./een user list --format tree`	Displays a hierarchical view of all users and the sites to which they have access.
Check CLI Tool Version	`./een --version`	The fastest way to confirm you are running the latest version of the Command Line Interface.
List All Devices (Bridges and Cameras)	`./een device list`	A comprehensive command to view all connected devices and their current online/offline status.
Get Detailed Camera Status	`./een camera status <camera_id> --verbose`	Provides an extensive output on the current operational status, including bitrate and firmware.
List Active Alerts	`./een event list --status active --limit 50`	Displays the 50 most recent events or alerts that are currently unresolved.

SYSTEM EDITS		
Goal	**Example Command**	**Description**
Rename a Bridge	`./een bridge edit <bridge_id> --name "New Bridge Name"`	Updates the friendly name of a specific bridge device in the system.
Add a New User	`./een user create --email user@example.com --role "Viewer"`	Creates a new user account with a specified email and user role.
Logout of Current Session	`./een auth logout`	Safely logs out the current user session from the CLI tool.
Delete an Unused Bridge	`./een bridge delete <bridge_id> --no-prompt`	Permanently removes a specified bridge device from the account without requiring confirmation.
Enable Archiving for a Camera	`./een camera edit <camera_id> --archiving-enabled true`	Turns on the long-term video archiving feature for a specific camera.
Disable an Event Type	`./een event-type edit <type_id> --enabled false`	Disables a specific type of alert (e.g., motion detection) across the system.
Set Camera Stream Resolution	`./een camera edit <camera_id> --resolution "1920x1080"`	Changes the streaming and recording quality for a specific camera.
Change a User's Role	`./een user edit --email john@example.com --role "Administrator"`	Elevates or demotes the access level of a specified user.

DOWNLOADS		
Goal	**Example Command**	**Description**
Download Specific Bridge Logs	`./een bridge logs download <bridge_id> --start 2024-09-01 --end 2024-09-30`	Retrieves system logs for a bridge within a specified date range for deeper analysis.
Download an Event Snapshot	`./een event download --event-id <event_id> --output-file /tmp/alert.jpg`	Downloads the still image associated with a specific triggered event/alert.
Download Video as MP4	`./een media download --camera <camera_id> --start 2024-09-29T11:00:00Z --end 2024-09-29T11:01:00Z --format mp4`	Forces the video clip download into the common MP4 file format.
List All Archives for a Camera	`./een media archives list --camera <camera_id>`	Shows all previously created, downloadable archives for a single camera.
Download a Specific Archive	`./een media archives download <archive_id> --output-dir /videos/backup`	Initiates the download of a specific, pre-created archive file.
Download Camera Configuration	`./een camera config download <camera_id>`	Retrieves the current configuration file for a camera, useful for backups or replication.
List Recorded Media Segments	`./een media list --camera <camera_id> --date 2024-10-01`	Lists all recorded video segments available for a specific camera on a specific day.

Setting Up Custom CLI Scripts

Creating and running a custom script involves structuring your commands, saving them in a script file, and granting the file permission to execute.

Step 1: Prepare the Environment and Commands

Ensure your environment is set up and authenticate the executable before running the script.

1. **Authenticate (Manual):** Before executing any script, you must manually authenticate the CLI session in your terminal by running the login command:
   Bash

```
None
./een auth login
```

2. **Test Commands:** Write and test each command line individually to ensure the syntax is correct and the desired output is achieved. Every EEN command must start with `./een`.
Bash

```
None
# Example command to test

./een camera status --verbose --format csv
```

Step 2: Create the Script File

You will use a simple text file (e.g., a `.sh` file for Unix/Linux or a `.bat` file for Windows) to contain your sequence of EEN commands.

1. **Open Editor:** Use any text editor (like Nano, VS Code, or Notepad).
2. **Add Commands:** Paste your sequence of commands into the file. Remember to use the `--no-prompt` flag if the script must automatically confirm potentially destructive actions (like overwrites).

Example Script (Save as daily_reports.sh):
Bash

```
None

#!/bin/bash

# 1. Check bridge status and save data to a unique CSV file

./een bridge list --verbose --format csv >
/tmp/bridge_status_$(date +%Y%m%d).csv

# 2. Get the current status for all cameras

./een camera status

# 3. Pull all archives created in the last 24 hours

./een media archives list --since 24h
```

Step 3: Grant Execution Permissions (Linux/macOS Only)

On Unix-like systems, you must grant the file permission to be run as a program. This step can be skipped on Windows.

1. **Change Directory (if necessary):** Navigate to the folder where you saved your script file.
   Bash

```
None
cd /path/to/your/script/folder
```

2. **Grant Permission:** Use the chmod command.
   Bash

None
```
chmod +x daily_reports.sh
```

Step 4: Execute and Automate

Once permissions are set, you can run the custom script manually or schedule it for future automation.

1. **Run Manually:** Execute the script using the executable prefix (`./`).
   Bash

None
```
./daily_reports.sh
```

2. **Automate (Schedule):** Integrate this executable script with your system's scheduler (e.g., **`cron`** on Linux/macOS or **Task Scheduler** on Windows) to run unattended at specific intervals.

Crucial Note for Automations: When scheduling, ensure the user account running the script has a valid, active EEN CLI session token (obtained from the initial `auth login` step) to avoid authentication errors