Vulnerability Scan Configuration

- Command Line Arguments
- INI Options

Vulnerability scanning can be configured using either command line arguments, the INI file, or a combination of both.

Command Line Arguments

- `--read-stdin`: Read WordPress base paths from stdin. If not specified, paths will automatically be read from stdin when input is not from a TTY.
- `-s`, `--path-separator`: Separator used to delimit paths when reading from stdin. Defaults to the null byte.
- `-w`, `--wordpress-path`: Path to the root of a WordPress installation to scan for core vulnerabilities.
- `-p`, `--plugin-directory`: Path to a directory containing WordPress plugins to scan for vulnerabilities.
- `-t`, `--theme-directory`: Path to a directory containing WordPress themes to scan for vulnerabilities.
- `-C`, `--relative-content-path`: Alternate path of the wp-content directory relative to the WordPress root.
- `-P`, `--relative-plugins-path`: Alternate path of the wp-content/plugins directory relative to the WordPress root.
- `-M`, `--relative-mu-plugins-path`: Alternate path of the wp-content/mu-plugins directory relative to the WordPress root.
- `--output`: Write results to stdout. This is the default behavior when --output-path is not specified.
- `--output-path`: Path to which to write results.
- `--output-columns`: An ordered, comma-delimited list of columns to include in the output. Available columns: `software_type`, `slug`, `version`, `id`, `title`, `link`, `description`, `cve`, `cvss_vector`, `cvss_score`, `cvss_rating`, `cwe_id`, `cwe_name`, `cwe_description`, `patched`,

`remediation`, `published`, `updated`, `scanned_path`. Compatible formats: csv, tsv, null-delimited, line-delimited.

- `-m`, `--output-format`: Output format used for result data.
- `--output-headers`: Include column headers in output. Compatible formats: csv, tsv, null-delimited, line-delimited.
- `-e`, `--exclude-vulnerability`: Vulnerability UUIDs or CVE IDs to exclude from scan results.
- `-i`, `--include-vulnerability`: Vulnerabilitiy UUIDs or CVE IDs to include in scan results.
- `-I`, `--informational`: Include informational vulnerability records in results.
- `-f`, `--feed`: The feed to use for vulnerability information. The production feed provides all available information fields. The scanner feed contains only the minimum fields necessary to conduct a scan and may be a better choice when detailed vulnerability information is not needed.
- `--require-path`: When enabled, an error will be issued if at least one path to scan is not specified. This is the default behavior when running in a terminal.

INI Options

```
[VULN_SCAN]
# Read WordPress paths from stdin
read_stdin = [on|off]
# Separator to use when reading paths from stdin, defaults to null byte
path_separator = <separator>
# Alternate relative path for wp-content
relative_content_path = <path>
# Alternate relative path for wp-content/plugins
relative_plugins_path = <path>
# Alternate relative path for wp-content/mu-plugins
relative_mu_plugins_path = <path>
# Controls whether or not output is written to stdout
output = [on|off]
output_path = <path to which to write results>
# Comma-delimited list of columns to include in output (`software_type`, `slug`, `version`, `id`, `title`, `link`, `description`, `cve`, `cvss_vector`, `cvss_score`, `cvss_rating`, `cwe_id`, `cwe_name`, `cwe_description`, `patched`, `remediation`, `published`, `updated`, `scanned_path`)
output_columns = <columns>
output_format = [human|csv|tsv|null-delimited|line-delimited]
# Whether to include headers in output
output_headers = [on|off]
# Comma-delimited list of vulnerability UUIDs or CVE IDs
exclude_vulnerability = <ids>
```

```
include_vulnerability = <ids>
# Toggle informational vulnerabilities
informations = [on|off]
feed = [production|scanner]
allow_nested = [on|off]
allow_io_errors = [on|off]
```

□□□ #2
□ □□□ □□□ 2026-08-29 22:03:38 UTC
□ □□□ □□□ 2026-08-29 22:03:38 UTC