

# MISP CLI

AUTOMATE ALL THE THINGS

CIRCL / TEAM MISP PROJECT



MISP PROJECT

2024-04-15

MISP CLI

MISP CLI

AUTOMATE ALL THE THINGS

CIRCL / TEAM MISP PROJECT



MISP PROJECT

- The MISP API is great for remotely executing administrative tasks
- But sometimes we want to simplify the process / avoid having to deal with authentication
- MISP also has an extensive CLI sub-system for this reason

### └─ MISP CLI functionalities

- The MISP API is great for remotely executing administrative tasks
- But sometimes we want to simplify the process / avoid having to deal with authentication
- MISP also has an extensive CLI sub-system for this reason

- Automating recurring tasks
- Recovery from loss of access
- Updates / initialisation
- Background worker management

## └ Types of objectives for the scripts

- Automating recurring tasks
- Recovery from loss of access
- Updates / initialisation
- Background worker management

■ <https://path.to.your.misp/events/automation>

## Administering the background workers via the API.

You can start/stop and view the background workers via the API.

**Add worker:** `http://localhost:5001/servers/startWorker/[queue_name]`

**Stop worker:** `http://localhost:5001/servers/stopWorker/[worker_pid]`

**Get worker info:** `http://localhost:5001/servers/getWorkers`

## Administering MISP via the CLI

Certain administrative tasks are exposed to the API, these help with maintaining and configuring MISP in an automated way / via external tools.:

**Get Setting:** `MISP/app/Console/cake Admin getSetting [setting]`

**Set Setting:** `MISP/app/Console/cake Admin setSetting [setting] [value]`

**Get Authkey:** `MISP/app/Console/cake Admin getAuthkey [email]`

**Set Baseurl:** `MISP/app/Console/cake Admin setBaseurl [baseurl]`

**Change Password:** `MISP/app/Console/cake Admin setPassword [email] [new_password] [--override_password_change]`

**Clear Bruteforce Entries:** `MISP/app/Console/cake Admin clearBruteForce [user_email]`

**Run Database Update:** `MISP/app/Console/cake Admin updateDatabase`

**Update All JSON Structures:** `MISP/app/Console/cake Admin updateJSON`

**Update Galaxy Definitions:** `MISP/app/Console/cake Admin updateGalaxies`

**Update Taxonomy Definitions:** `MISP/app/Console/cake Admin updateTaxonomies`

**Update Object Templates:** `MISP/app/Console/cake Admin updateObjectTemplates`

**Update Warninglists:** `MISP/app/Console/cake Admin updateWarningLists`

2024-04-15

└─ CLI documentation

■ <https://path.to.your.misp/events/automation>

### Administering the background workers via the API.

You can start/stop and view the background workers via the API.

**Add worker:** `http://localhost:5001/servers/startWorker/[queue_name]`

**Stop worker:** `http://localhost:5001/servers/stopWorker/[worker_pid]`

**Get worker info:** `http://localhost:5001/servers/getWorkers`

### Administering MISP via the CLI

Certain administrative tasks are exposed to the API, these help with maintaining and configuring MISP in an automated way / via external tools.:

**Get Setting:** `MISP/app/Console/cake Admin getSetting [setting]`

**Set Setting:** `MISP/app/Console/cake Admin setSetting [setting] [value]`

**Get Authkey:** `MISP/app/Console/cake Admin getAuthkey [email]`

**Set Baseurl:** `MISP/app/Console/cake Admin setBaseurl [baseurl]`

**Change Password:** `MISP/app/Console/cake Admin setPassword [email] [new_password] [--override_password_change]`

**Clear Bruteforce Entries:** `MISP/app/Console/cake Admin clearBruteForce [user_email]`

**Run Database Update:** `MISP/app/Console/cake Admin updateDatabase`

**Update All JSON Structures:** `MISP/app/Console/cake Admin updateJSON`

**Update Galaxy Definitions:** `MISP/app/Console/cake Admin updateGalaxies`

**Update Taxonomy Definitions:** `MISP/app/Console/cake Admin updateTaxonomies`

**Update Object Templates:** `MISP/app/Console/cake Admin updateObjectTemplates`

**Update Warninglists:** `MISP/app/Console/cake Admin updateWarningLists`

```
/var/www/MISP/app/Console/cake [Shell] [Command]  
[parameters]
```

## ■ Example:

- ▶ `/var/www/MISP/app/Console/cake Password "andras.iklody@gmail.com" "Nutella"`
- ▶ Change password to "Nutella" for my user
- ▶ Some shells are single use and don't need a command parameter

■ Also used by the background processing

■ Automation is meant to be used via cron jobs

## Usage

```
/var/www/MISP/app/Console/cake [Shell] [Command]  
[parameters]  
■ Example:  
  ▶ /var/www/MISP/app/Console/cake Password  
    "andras.iklody@gmail.com" "Nutella"  
  ▶ Change password to "Nutella" for my user  
  ▶ Some shells are single use and don't need a command  
    parameter  
■ Also used by the background processing  
■ Automation is meant to be used via cron jobs
```

- Edit crontab of www-data user
- `crontab -u www-data -e`
- `0 3,9,15,21 * * *`  
`/var/www/MISP/app/Console/cake Server pull 1`  
`30 full`
- Pull server ID #30 as user #1 every 6 hours
- `@hourly /var/www/MISP/app/Console/cake Server`  
`cacheFeed 1 csv full`
- Cache all csv feeds as user #1 every hour

### Automation via crontab

- Edit crontab of www-data user
- `crontab -u www-data -e`
- `0 3,9,15,21 * * *`  
`/var/www/MISP/app/Console/cake Server pull 1`  
`30 full`
- Pull server ID #30 as user #1 every 6 hours
- `@hourly /var/www/MISP/app/Console/cake Server`  
`cacheFeed 1 csv full`
- Cache all csv feeds as user #1 every hour