## MISP CORE DEVELOPMENT HANDS-ON EXERCISE

BUILDING A SMALL NIFTY FEATURE FOR THE MISP CORE

CIRCL / TEAM MISP PROJECT



**NSPA** 



MISP core development hands-on exercise

05

2022-08





### Some practical things first...

MISP core development hands-on exercise

└─Some practical things first...

 If you'd like to take a peak at the main files already implemented: https://github.com/iglocska/misp-dev-training-cheat-sheet
 Full implementation: https://github.com/MISP/MISP/tree/dev\_session/app

If you'd like to take a peak at the main files already implemented: https://github.com/iglocska/misp-dev-training-cheat-sheet

**Full implementation:** 

https://github.com/MISP/MISP/tree/dev\_session/app

08-05

### LET'S TRY TO DEVELOP A FEATURE TOGETHER

2022-08-05

### MISP core development hands-on exercise

Let's try to develop a feature together

Idea: Users should have the option to set alert filters for the publish alert e-mails
 By default receive all alerts as before
 If a filter is set, check if the alert is interesting for us or not

- Idea: Users should have the option to set alert filters for the publish alert e-mails
- By default receive all alerts as before
- If a filter is set, check if the alert is interesting for us or not

# How to ensure that the feature is useful for the community at large?

2022-08-05

MISP core development hands-on exercise

How to ensure that the feature is useful for the community at large?

How to ensure that the feature is useful for the community at large?

- Always try to think in reusable systems instead of fixing a single issue
- Much higher chance or getting a PR merged if it doesn't just cover your specific use-case
   Try to stay two steps ahead, see how your feature can be

- Always try to think in reusable systems instead of fixing a single issue
  - Much higher chance of getting a PR merged if it doesn't just cover your specific use-case
  - Try to stay two steps ahead, see how your feature can be reused for other tasks

### **USER SETTINGS - A LONG OVERDUE FEATURE**

2022-08-05 ₹

MISP core development hands-on exercise

└─User settings - a long overdue feature

Allow users to set preferences for certain views
 For high level users, all the technical details are sometimes
wasted
 Simply not being interested in certain types of data points
 Non-standard MISP denoments fisland only MISP

User pre-sets for certain settings

- Allow users to set preferences for certain views
- For high level users, all the technical details are sometimes wasted
- Simply not being interested in certain types of data points
- Non-standard MISP deployments (island only MISP instances, etc)
- User pre-sets for certain settings

### **OBJECTIVES OF THE FEATURE**

හ MISP core development hands-on exercise

└─Objectives of the feature

User should be able to do the following with filter rules:
 Sat
 Sat
 Farmore
 Index
 Filter rules should be flexible - we do not want to anticipate

Ensure that the system is easy to extend and reuse

- User should be able to do the following with filter rules:
  - set
  - get
  - remove
  - index
- Filter rules should be flexible we do not want to anticipate all possible settings in advance
- Ensure that the system is easy to extend and reuse

2022-08

### BEFORE WE START WITH ANYTHING...

MISP core development hands-on exercise

Before we start with anything...

BEFORE WE START WITH ANYTHING ...

- Update our MISP instance (git pull origin 2.4)
  Fork github.com/MISP/MISP (via the github interface)
- Add a new remote to our fork:
- via username/password auth: git remote ad https://github.com/inforce/a/MISD
- via ssh: git remote add my\_fork
- gitgithub.com:iglocska/MISPgit
- Generally a good idea to work on a new branch: git checkou
- Enable debug in MISP

- Update our MISP instance (git pull origin 2.4)
- Fork github.com/MISP/MISP (via the github interface)
- Add a new remote to our fork:
  - via username/password auth: git remote add my\_fork https://github.com/iglocska/MISP
  - via ssh: git remote add my\_fork gitgithub.com:iglocska/MISP.git
- Generally a good idea to work on a new branch: git checkout -b dev\_exercise
- Enable debug in MISP

05

8 8 0

### **IMPLEMENTATION**

မ္မ MISP core development hands-on exercise

8 8 0

2022

### -Implementation

- Storage:

### Storage:

- Single key/value table for all settings
- Each user should be able to set a single instance of a key
- Values could possible become complex, let's use JSON!
- Add timestamping for traceability
- Consider which fields we might want to look-up frequently for indexing

### THE DATABASE CHANGES WE NEED

MISP core development hands-on exercise

└─The database changes we need

- The table structure:
- id int(11) auto increment //primary ke
- key varchar(too) //add index!
  value text //ison
- user\_id int(1) //add indext
- timestamp int(11) //add indext
- He it to into the upgrade system (app/Model/AppModel.ph)
   Test our upgrade process! Check the output in the audit loss

### ■ The table structure:

- id int(11) auto increment //primary key
- key varchar(100) //add index!
- value text //json
- user\_id int(11) //add index!
- timestamp int(11) //add index!
- Tie it to into the upgrade system (app/Model/AppModel.php)
- Test our upgrade process! Check the output in the audit logs

05

2022-08

CHECKLIST

မ္မ MISP core development hands-on exercise

└─ Checklist

80

2022

- Outline of the changes needed:
- New Controller (UserSettingsControlle New Model (UserSetting php)
- New Views (setSetting, index)
- Add new controller actions to ACL

### Outline of the changes needed:

- New Controller (UserSettingsController.php)
- New Model (UserSetting.php)
- New Views (setSetting, index)
- Add new controller actions to ACL
- Update the e-mail alert system to use the functionality

### CREATE THE NEW MODEL SKELETON

မှု MISP core development hands-on exercise

8 8 0

2022-

#### CREATE THE NEW MODEL SKELETON

- location: /var/www/MISP/app/Model/UserSetting.php
   Create basic skeleton
- Add model relationships (hasMany/BelongsTo)
   Use the hooking functionality to deal with the JSON field
- (beforeSave(), beforeFind()) Add a function that can be used to check if a user should get
- an alert based on filters (checkPublishFilter()) Add a function to check if a user can access/modify a setting

- Iocation: /var/www/MISP/app/Model/UserSetting.php
- Create basic skeleton
- Add model relationships (hasMany/BelongsTo)
- Use the hooking functionality to deal with the JSON field (beforeSave(), beforeFind())
- Add a function that can be used to check if a user should get an alert based on filters (checkPublishFilter())
- Add a function to check if a user can access/modify a setting (checkAccess())

### **CREATE THE CONTROLLER SKELETON**

MISP core development hands-on exercise

Create the Controller skeleton

Iocation: /var/www/MISP/app/Model/UserSetting.ph Set pagination rule Define CRUD functions (exceptionally, we diverge here from

- location: /var/www/MISP/app/Model/UserSetting.php
- Create basic skeleton
- Set pagination rules
- Define CRUD functions (exceptionally, we diverge here from the norm)
  - setSetting()
  - getSetting()
  - index()
  - delete()

05

8 8 0

### START WITH AN API ONLY APPROACH AT FIRST

မ္မ MISP core development hands-on exercise

└─Start with an API only approach at first

setSetting():

- Accepted methods: ADD / POST
- Separate handling of API / UI
   POST should create/undate an ent
- GET should describe the API

### setSetting():

- Accepted methods: ADD / POST
- Separate handling of API / UI
- POST should create/update an entry
- GET should describe the API

80

### **GETSETTING / INDEX**

2022-08-0

### မ္မ MISP core development hands-on exercise

└─getSetting / index

GETSETTING / INDEX

- getSetting():
- Accepted methods: GET
   Retrieves a single setting based on either ID or setting
- Retrieves a single setting based on either ID or se and user\_id
- Encode the data depending on API/UI
  - Accepted methods: GEI
     List all settings
- Filter user scope on demand
- Filter available scopes based on

### getSetting():

- Accepted methods: GET
- Retrieves a single setting based on either ID or setting key and user\_id
- Encode the data depending on API/UI
- Accepted methods: GET
- List all settings
- Filter user scope on demand
- Filter available scopes based on role

DELETE

မှ MISP core development hands-on exercise

└\_ delete

2022-

- delete():
- Accepted methods: POST / DELETE
- Deletes a single entry based on ID or setti
   Encode the data depending on Phillips

### delete():

- Accepted methods: POST / DELETE
- Deletes a single entry based on ID or setting key
- Encode the data depending on API/UI

### ADD THE ACL FUNCTIONALITIES

L

05

8 8 0

2022

### MISP core development hands-on exercise

Add the ACL functionalities

- Tie functions into checkAccess():
- Check if user is allowed to execute actions and thro exceptions if not
   Add it to cotSotting() / outSotting() / doise()
  - Consider that:
  - Site admins have full reign
  - Org admins can manage their ow
     Evenuone else can self-manage

### ■ Tie functions into checkAccess():

- Check if user is allowed to execute actions and throw exceptions if not
- Add it to: setSetting() / getSetting() / delete()

### Consider that:

- Site admins have full reign
- Org admins can manage their own users
- Everyone else can self-manage

### **TEST THE FUNCTIONALITIES**

MISP core development hands-on exercise

05

8 8 0

2022

#### └─ Test the functionalities

- Use the REST client
- Expectations
- GET on /setSetting and /delete des
- POST (setsetting with "key": "publish\_niter", "value": "Event.tags": "Ksofacy%" should return newly added or
- modified filter

  F GET on /index should list our entries GET on /getSetting
- Get on yindex should use our entries, det on ygetsett should show an individual entry
- DELETE on /delete should delete the entry

### Use the REST client

### Expectations

- GET on /setSetting and /delete describing our endpoints
- POST /setSetting with "key": "publish\_filter", "value": "Event.tags":"%sofacy%" should return newly added or modified filter
- GET on /index should list our entries, GET on /getSetting should show an individual entry
- DELETE on /delete should delete the entry

### START ADDING THE UI COMPONENTS

2022-08-05

### MISP core development hands-on exercise

└─Start adding the UI components

- We now have a rudimentary CRUD, let's add some simple UI views
- setSetting as a simple form
- Index should use the parametrised generators (IndexTa
- Add both views to the menu systems (side-menu, globa menu)
- Don't forget about sanitisation and translations

- We now have a rudimentary CRUD, let's add some simple UI views
  - setSetting as a simple form
  - index should use the parametrised generators (IndexTable)
  - Add both views to the menu systems (side-menu, global menu)
  - Don't forget about sanitisation and translations!

### Add the checkPublishFilter() function to the emailing

 $_{\odot}^{N}$  MISP core development hands-on exercise

-Add the checkPublishFilter() function to the e-mailing

ADD THE CHECKPUBLISHFILTER() FUNCTION TO THE E-MAILING

Trace the code path of the e-mail sending to understand the process
 Decide on the best place to inject our check
 Don't break the flow of the process!
 What do we have access to at this point? What format are they in?

- Trace the code path of the e-mail sending to understand the process
- Decide on the best place to inject our check
- Don't break the flow of the process!
- What do we have access to at this point? What format are they in?

80

2022-

### TEST IF OUR CODE WORKS CORRECTLY

MISP core development hands-on exercise

└─Test if our code works correctly

Do we see any notices / errors?
 Is our code easily accessible?
 Consider other rolest Can users/org admins do things we
 don't want them to do?
 Is our code-base breaking the default behaviour?
 Is our code-strict working as expected?

### Do we see any notices / errors?

- Is our code easily accessible?
- Consider other roles! Can users/org admins do things we don't want them to do?
- Is our code-base breaking the default behaviour?
- Is our update script working as expected?

08-05

# PUSH OUR CODE TO OUR FORK AND CREATE A PULL REQUEST

2022-08-05

MISP core development hands-on exercise

Push our code to our fork and create a pull request

PUSH OUR CODE TO OUR FORK AND CREATE A PULL REQUEST

- git status to check what changed / got added
   git add /path/to/file to add files we want to commit
   git commit (format: is "rew/fbx/chg: [topic] My description"
   git push my\_fork
   E Create pull request from the github interface
- Wait for Travis to run, update the code if needed

- sit status to check what changed / got added
- git add /path/to/file to add files we want to commit
- git commit (format: is "new/fix/chg: [topic] My description"
- git push my\_fork
- Create pull request from the github interface
- Wait for Travis to run, update the code if needed