Visualise all the things

Building dashboard widgets for MISP

CIRCL / Team MISP Project

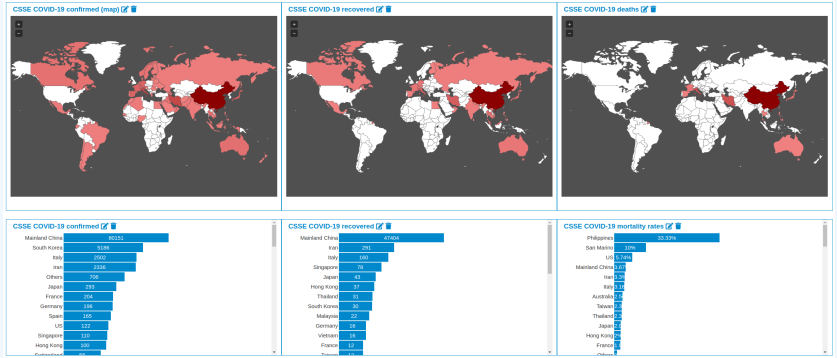
http://www.misp-project.org/
Twitter: @MISPProject

13th ENISA-EC3 Workshop

Visualise all the things
Building dashboard widgets for MISP
CIRCL / Team MISP Project
http://www.misp-project.org/
Twitter: @MISPProject
13th ENISA-EC3 Workshop
MISP
Threat Sharing

Dashboard in MISP

- User configurable simple dashboard interface
- **Visualise**, **aggregate** and **track** data important to you
- Brand new feature, still undergoing reworks



2024-09-11

Visualise all the things

└Dashboard in MISP

Dashboard in MISP
User configurable simple dashboard interface
Visualise, aggregate and track data important to you
Brand new feature, still undergoing reworks

The internals of awidget

- **Backend** for the widget, full access to all MISP internals
- **Load, convert, format** to be represented via view widgets
- **Widget metadata** - size, name, description, behaviours
- Only main function required to be implemented: **handler()**
- Optional: **checkPermissions() for ACL**
- Accepts **user configuration** for which a template can be provided
- Located in /var/www/MISP/app/Lib/Dashboard/
- Custom widgets can be placed in /var/www/MISP/app/Lib/Dashboard/Custom/

The internals of awidget
Backend for the widget, full access to all MISP internals
Load, convert, format to be represented via view widgets
Widget metadata - size, name, description, behaviours
Only main function required to be implemented: handler()
Optional: checkPermissions() for ACL
Accepts user configuration for which a template can be provided
Located in /var/www/MISP/app/Lib/Dashboard/
Custom widgets can be placed in /var/www/MISP/app/Lib/Dashboard/Custom/

The view layer of a widget

- View files are included by default and reusable
- Currently we have a small but growing list of views
  - BarChart
  - SimpleList
  - WorldMap
- Converts the data passed by the Widget logic to HTML
- Located in /var/www/MISP/view/Elements/dashboard/Widgets/

The view layer of a widget
- View files are included by default and reusable
- Currently we have a small but growing list of views
  - BarChart
  - SimpleList
  - WorldMap
- Converts the data passed by the Widget logic to HTML
- Located in /var/www/MISP/view/Elements/dashboard/Widgets/

Widget behaviours

- Widgets can additionally be tied to certain **behaviours**:
  - Caching
    - Executions of the widget logic are cached
    - **Separate caches for each organisation in addition to site admins**
    - Cache duration is controlled by the widget logic
  - Refresh
    - Widgets can be set to refresh after x seconds
  - Both of these should be used with special care in regards to the use of **system resources**

Widget behaviours
Widgets can additionally be tied to certain behaviours:
Caching
Executions of the widget logic are cached
Separate caches for each organisation in addition to site admins
Cache duration is controlled by the widget logic
Refresh
Widgets can be set to refresh after x seconds
Both of these should be used with special care in regards to the use of system resources

Exercise module: simple Whoami

- Let's start with a skeleton
- Create /var/www/MISP/app/Lib/Dashboard/Custom/WhoamiWidget.php
- MISP will parse anything ending with Widget.php in this directory

Exercise module: simple Whoami
- Let's start with a skeleton
- Create /var/www/MISP/app/Lib/Dashboard/Custom/WhoamiWidget.php
- MISP will parse anything ending with Widget.php in this directory

Exercise module: simple Whoami

```php
<?php
class MispWhoamiWidget
{
  public $title = 'Whoami';
  public $render = 'SimpleList';
  public $width = 2;
  public $height = 2;
  public $params = array();
  public $description = 'Shows information about the
      currently logged in user.';
  public $cacheLifetime = false;
  public $autoRefreshDelay = 3;

  public function handler($user, $options = array())
  {
    $data = array();
    return $data;
  }
}
```


Exercise module: simple Whoami

Meta information

- **$title**: The name of the widget
- **$description**: A description of the widget
- **$render**: The view element to use in rendering the widget
- **$width & $height**: Default relative dimensions
- **$params**: Configuration array with explanations for each key
- **$cacheLifetime**: The lifetime of the caches in seconds (false disables it)
- **$autoRefreshDelay**: The time in seconds between each refresh (false disables it)

Meta information
- **$title**: The name of the widget
- **$description**: A description of the widget
- **$render**: The view element to use in rendering the widget
- **$width & $height**: Default relative dimensions
- **$params**: Configuration array with explanations for each key
- **$cacheLifetime**: The lifetime of the caches in seconds (false disables it)
- **$autoRefreshDelay**: The time in seconds between each refresh (false disables it)

The handler

```php
public function handler($user, $options = array())
{
  $this->Log = ClassRegistry::init('Log');
  $entries = $this->Log->find('all', array(
    'recursive' => -1,
    'conditions' => array(
      'action' => 'login', 'user_id' => $user['id']
    ),
    'order' => 'id desc',
    'limit' => 5,
    'fields' => array('created', 'ip')
  ));
  foreach ($entries as &$entry) {
    $entry = $entry['Log']['created'] . ' --- ' .
    (
      empty($entry['Log']['ip']) ?
      'IP not logged' :
       $entry['Log']['ip']
    );
  }
  return array(
    array('title' => 'Email', 'value' => $user['email']),
    array(
      'title' => 'Role', 'value' => $user['Role']['name']
    ),
    array(
      'title' => 'Organisation',
      'value' => $user['Organisation']['name']
    ),
    array(
      'title' => 'IP', 'value' => $_SERVER['REMOTE_ADDR']
    ),
    array('title' => 'Last logins', 'value' => $entries)
  );
}
```


Result

Whoami

Email: admin@admin.test
Role: admin
Organisation: ORGNAME
IP: ::1
Last logins:
2020-03-05 06:50:46 --- ::1
2020-03-04 21:35:15 --- IP not logged
2020-03-04 09:34:44 --- IP not logged
2020-03-03 16:58:35 --- IP not logged
2020-03-03 06:49:10 --- IP not logged

Result