Hacking for hoodies: MISP edition

- https://jeroenpinoy.website

#YearOfJeroen

About me

- Jeroen Pinoy
- Master Computer Science, +- 10 years of experience in software testing, system administration, incident response, CTI
- Currently MISP engineer at a large international organisation

@wachizungu@infosec.exchange

@wachizungu

A very short intro to MISP

Adversary details

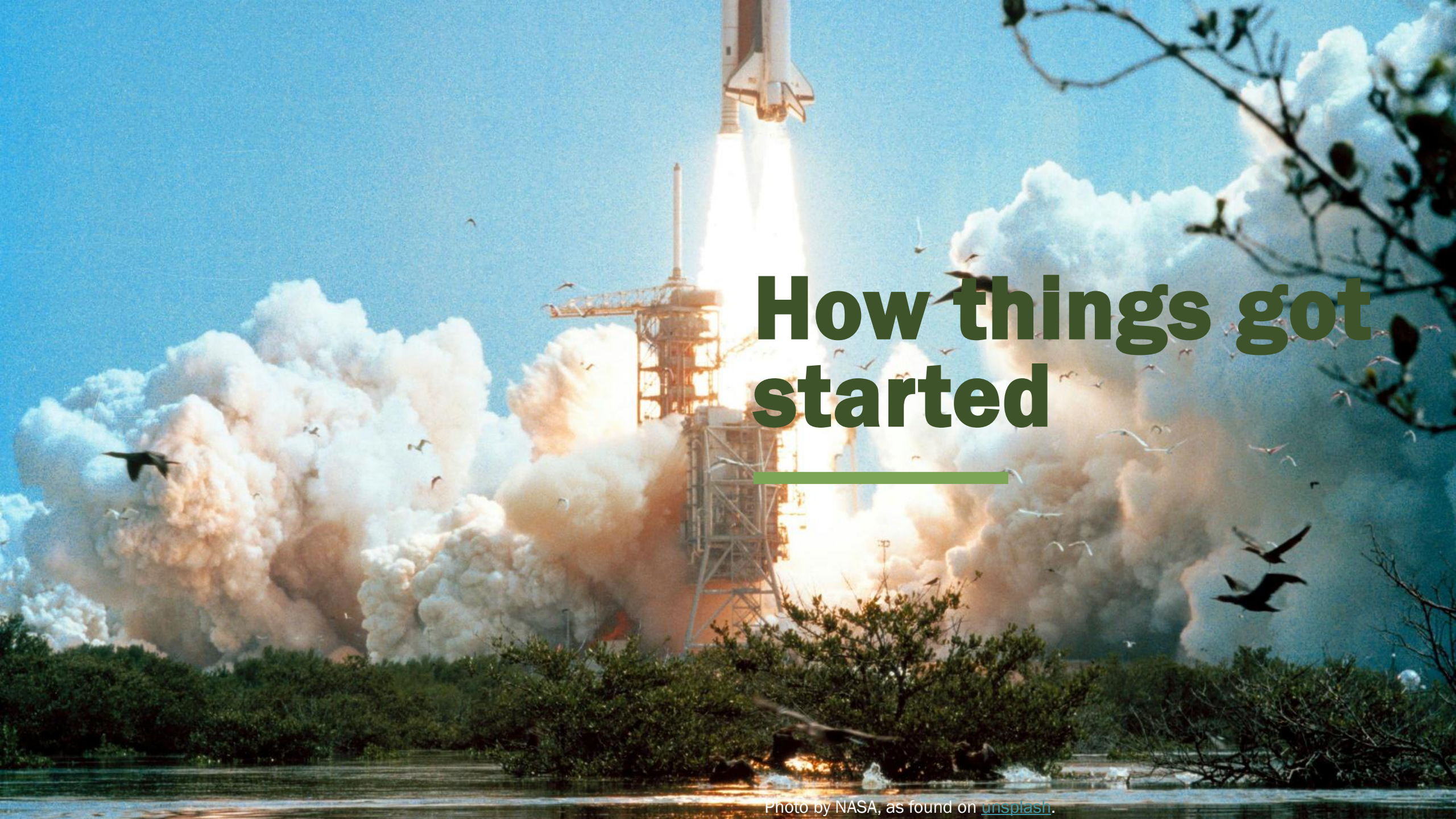
- Only interested in Häagen-Dazs, Magnum ice cream is not good enough
- Can jump 2m high
- Potential defense strategy: Keep Häagen-Dazs ice cream on shelves higher than 2m

Photo by Klara Kulikova, as found on unsplash.

Sharing / collaboration

- Other victim reports cat can reach 2.20m. This led to significant loss of ice cream (and pizza toppings), despite implementing proposed strategy

Example use case



How things got started

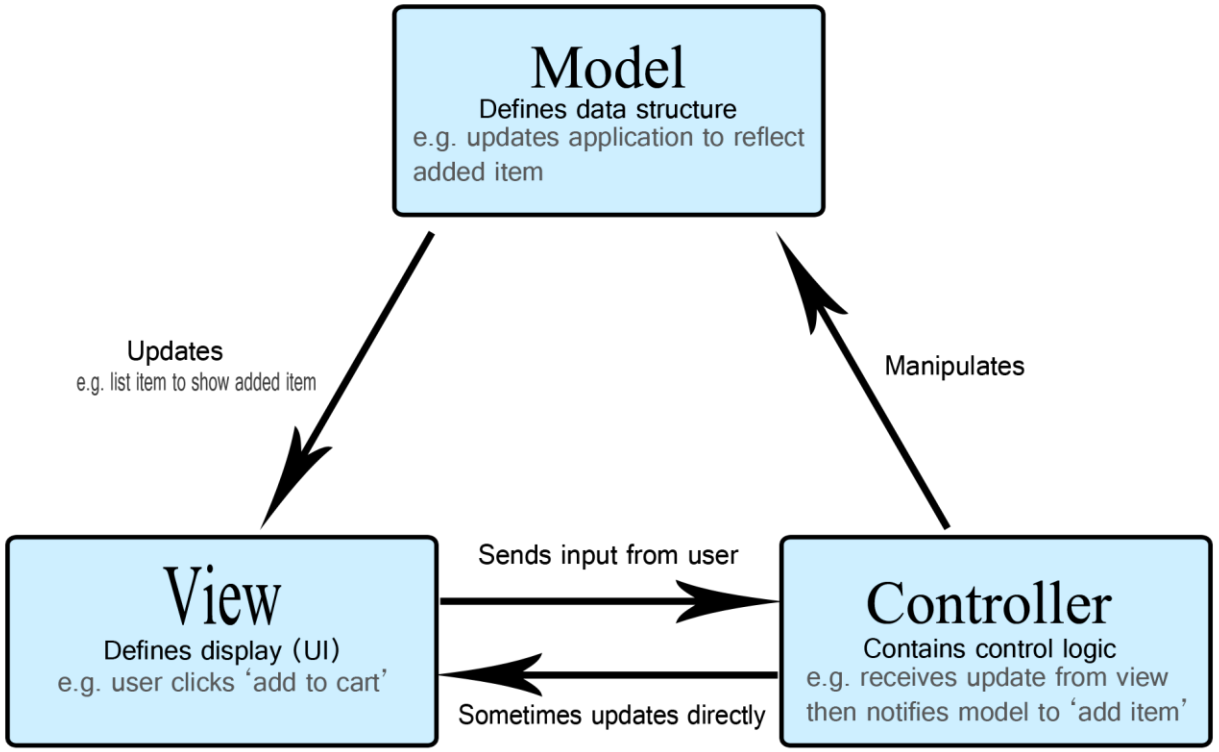
Photo by NASA, as found on unsplash.

Getting familiar with Cerebrate

1. Can it help us?
2. How is it built?

https://github.com/cerebrate-project/cerebrate

Model-View-Controller (MVC) CakePHP application



As found at https://developer.mozilla.org/en-US/docs/Glossary/MVC , CC-BY SA 3.0

Cerebrate security findings (broken access control)

- Vulnerability allowing group admins to add any local organisation to their org group (no CVE assigned)
- Vulnerability allowing group admins to inject users into organisations not managed by them (no CVE assigned)
- Vulnerability allowing privilege escalation from org admin to site admin in some scenarios (no CVE assigned)
- Vulnerability allowing privilege escalation from org admin to community admin in same organization (no CVE assigned)
- Vulnerability allowing users to view user settings of any other user (gcve-1-2025-0003)

Hoodie?

- “Personal opinion” -> get to the point to “deserve to wear a hoodie”
  - For this and future projects
  - Trigger for memory and sort of trophy
    - Can also be for development or other contributions

Approach

Photo by Alexandre Dulaunoy, as found on flickr.

Using my (beautiful) eyes

Manual code reviewing and exploratory testing

+ Some LLM usage because why not

Tools

JETBRAINS

- IDE, with step debugger, “find all usages”…
  - PHPStorm (with Xdebug), PyCharm, …

- LLMs (Kagi Search/Assistant)

- GitHub

- Burp Suite Community Edition

Picture by Alexandre Dulaunoy, cropped version of picture found on flickr.

Specific focus areas

- Prioritize critical components
- Look for specific vulnerability types
- Pivot on recently disclosed vulns
- Look at config options for inspiration

1. Looking at critical components

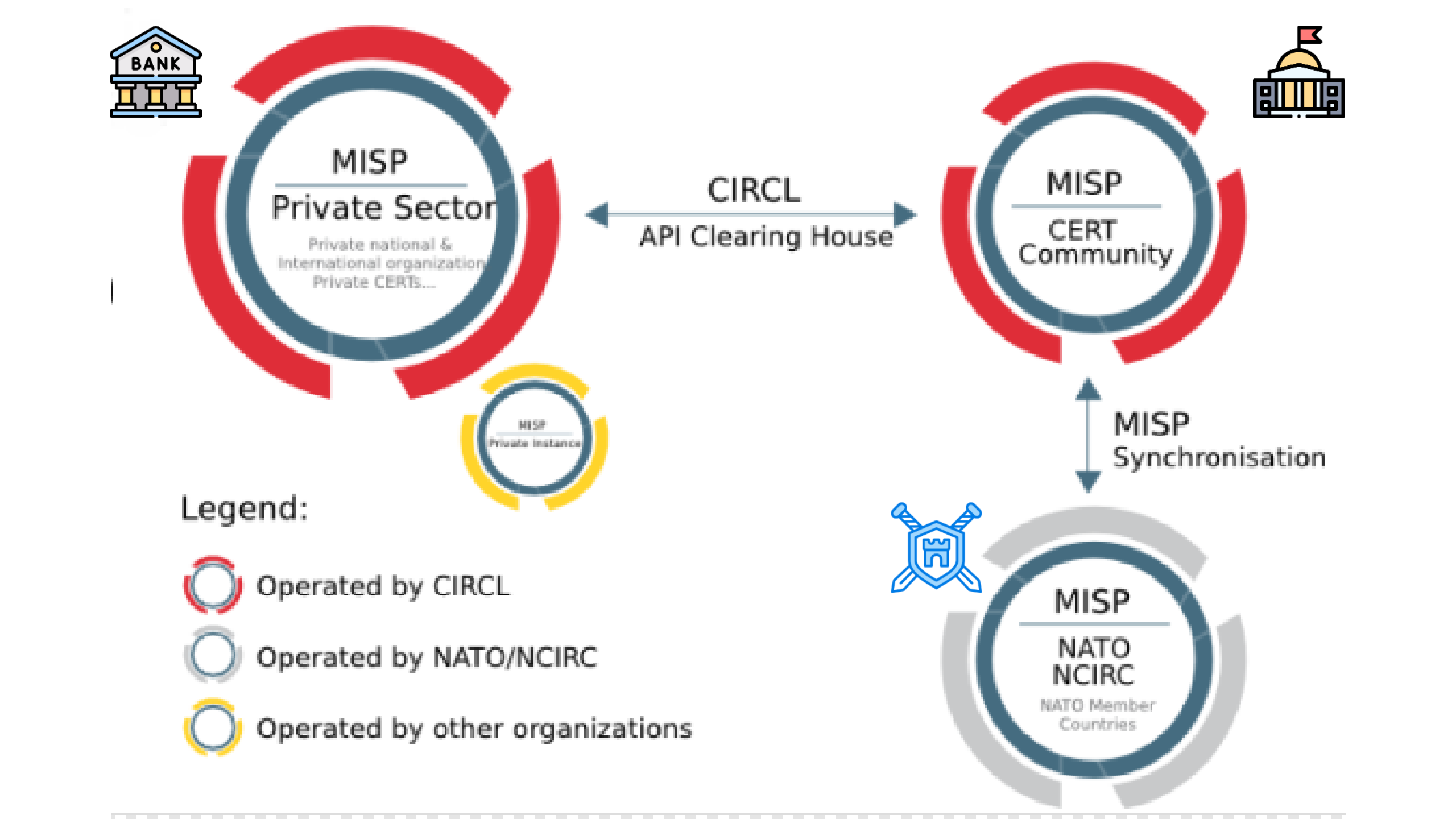
Photo by Alexandre Dulaunoy, as found on flickr.

Critical components: Users & Authkeys

Photo by Alexandre Dulaunoy, as found on flickr.

MISP user & role types

- Site admin
- Sync users
- Organisation admin
- Plebs (publisher, user, read-only user)...


BANK
MISP
Private Sector
Private national &
International organization
Private CERTs...
MISP
Private Instance
CIRCL
API Clearing House
MISP
CERT
Community
MISP
Synchronisation
Legend:
Operated by CIRCL
Operated by NATO/NCIRC
Operated by other organizations
MISP
NATO
NCIRC
NATO Member
Countries

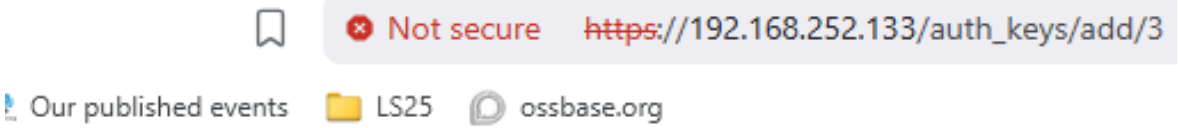
Users index

Click here to reset the API keys of all sync and org admin users in one shot. This will also a

« previous | next »

All | Enabled | Disabled | Inactive

	ID	Org	Role	Email
☐	2	Org1	Org Admin	orgadmin_org1@test.test



Dashboard Galaxies Input Filters Global Actions Sync Actions Administr

Add auth key

Auth keys are used for API access. A user can have more than one authkey, so if you would like

User

user_org2@test.test

Comment

Allowed IPs

Expiration (keep empty for indefinite)

YYYY-MM-DD

Read only (it will unset all permissions. This should not be used for sync users)

Submit

AuthKey added.

Auth key created

Please make sure that you note down the auth key below,

MISP will use the first and the last 4 characters for identific

```
jztj81oUGk0W61Hs1M1mb7ScKQ8fZ3Fvmb7IcIAD
```

I have noted down my key, take me back now

Check for organisation ID is missing!

```
private function __canCreateAuthKeyForUser($user_id)
{
    if (!$user_id)
        return true;
    if ($this->_isAdmin()) {
        if ($this->_isSiteAdmin()) {
            return true;   // site admin is OK for all
        } else {
            // org admin only for non-admin users and themselves
            $user = $this->AuthKey->User->find('first', [
                'recursive' => -1,
                'condition[illegible] => [
                    [illegible]r.[illegible] => [illegible]user_id,
                    'User.[illegible]d' => fal[illegible]
                ],
                'fields' => ['User.id', 'User.org_id', 'User.disabled'],
                'contain' => [
                    'Role' => [
                        [illegible]s' => [
                        [illegible]ol[illegible]erm[illegible]dm[illegible]le.[illegible]m_a[illegible]n', 'Rol[illegible]er[illegible]uth[illegible]
                    [illegible]]
                    ]
                ]
            ]);
            i[illegible]r[illegible]e'[illegible]si[illegible]] [illegible]
              [illegible]er[illegible]ole[illegible]'p[illegible]ad[illegible] $u[illegible]['User']['id'] !== $this->Auth->user('id')) ||
              [illegible]er[illegible]ole[illegible]'p[illegible]a[illegible]
                // no create/edit for [illegible]e ad[illegible] or other org admin
                return false;
            } else {
                // ok for themselves or users
                return true;
            }
        }
    } else {
        // user for themselves
        return (int)$user_id === (int)$this->Auth->user('id');
    }
}
```

Impact

- Can see data of other organisations
- Can potentially become “sync user” -> privilege escalation
- Can potentially become part of the host organization -> privilege escalation

Critical components: Logs

Photo by Alexandre Dulaunoy, as found on flickr.

Application Logs

« previous next »

Emails | Authentication issues

Id	IP	Email	Org	Created
872	192.168.252.1	user_admin@test.test	ADMIN	2025-04-23 16:1
830	192.168.252.1	user_admin@test.test	ADMIN	2025-04-23 16:0
829	192.168.252.1	user_admin@test.test	ADMIN	2025-04-23 15:5

Dashboard Galaxies Input Filters Global Actions Sync Actions Logs API

« previous next »

Emails Authentication issues

Id	IP	Email	Org	Created	Model	Model ID	Action	Title	Change
872	192.168.252.1	user_admin@test.test	ADMIN	2025-04-23 16:16:04	User	4	login	User (4): user_admin@test.test	{"user_agent":"Mozilla
871	192.168.252.1	admin@admin.test	ADMIN	2025-04-23 16:15:54	User	1	logout	User (1): admin@admin.test	

Plugin.Enrichment_vmware_nsx_analysis_api_token () => (dadadada)

									Request body: _meth
869	192.168.252.1	admin@admin.test	ADMIN	2025-04-23 16:15:26	User	1	request	Paranoid log entry	HTTP method: GET Target: /admin/users/
868	192.168.252.1	admin@admin.test	ADMIN	2025-04-23 16:15:20	User	1	request	Paranoid log entry	HTTP method: GET Target: /users/view/1
867	192.168.252.1	admin@admin.test	ADMIN	2025-04-23 16:15:19	User	1	request	Paranoid log entry	HTTP method: PUT Target: /users/edit Request body: _method=PUT&data%

Impact

- Potential lateral movement
- Potential privilege escalation
- Extra powerful attack chain combined with the previous vulnerability

2. Looking for specific vulnerability types

Photo by Alexandre Dulaunoy, as found on flickr.


2021
A01:2021-Broken Access Control
A02:2021-Cryptographic Failures
A03:2021-Injection
(New)
A04:2021-Insecure Design
A05:2021-Security Misconfiguration
A06:2021-Vulnerable and Outdated Components
A07:2021-Identification and Authentication Failures
(New)
A08:2021-Software and Data Integrity Failures
A09:2021-Security Logging and Monitoring Failures*
(New)
A10:2021-Server-Side Request Forgery (SSRF)*
* From the Survey

Claude 3.7 Sonnet with extended thinking | Web Access

I am an ethical hacker looking for vulnerabilities in software, to report to the developers. I am focusing on doing manual code reviews of a CakePHP application. To help me in my work, I am listing code patterns to watch out for to detect specific types of vulnerabilities.

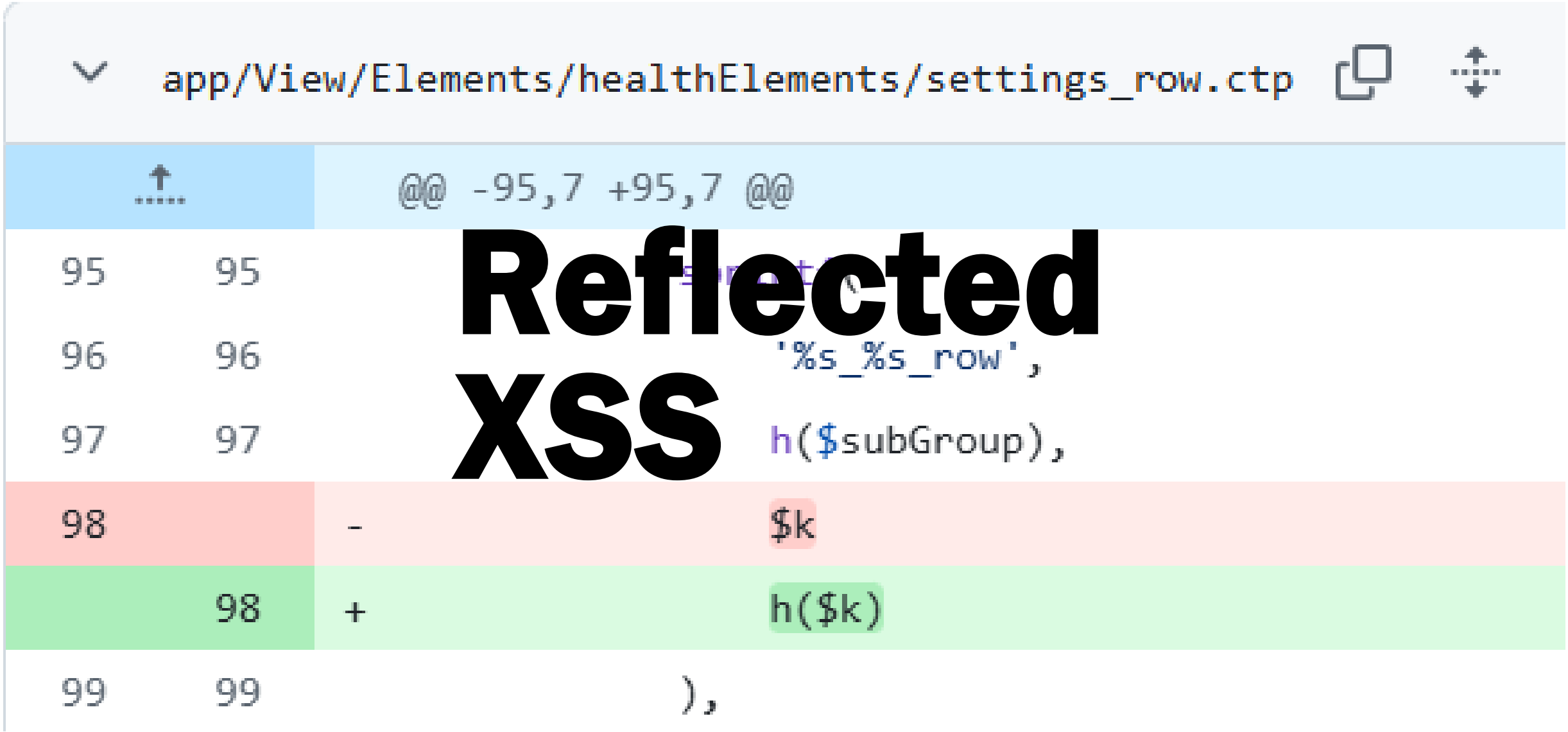
Which PHP function usage or code patterns should I look for, if I'm hunting for the types of vulnerabilities in the below list? Split your output per vulnerability type.
* XSS
* Remote code execution / system command execution
* SQL injection
* SSRF
* Broken access control
* HTML injection or other types of injection in PDF generation functions

▶ Thinking

Vulnerability Detection Guide for CakePHP Code Reviews

Code patterns - XSS

- echo
- print, printf, sprintf
- <?=



Reflected XSS

Code patterns – Remote Code Execution

- include, include_once, require, require_once
- exec, shell_exec, system

Code patterns – SQL injection

- query
- SELECT

Code patterns – SSRF

- file_get_contents, curl_exec
- fopen, fsockopen

Broken Access Control: The MISP ACLComponent

- Lists access control rules per endpoint
- Convenient list of all API endpoints / attack surface

- https://github.com/MISP/MISP/blob/2.5/app/Controller/Component/ACLComponent.php

app/Controller/Component/ACLComponent.php

```
@@ -251,11 +251,11 @@ class ACLComponent extends Component
251 251                 'restore' => array('perm_add'),
252 252                 'index' => array('*'),
253 253                 'getProxyMISPElements' => array('*'),
254     -               'extractAllFromReport' => array('*'),
255     -               'extractFromReport' => array('*'),
    254 +               'extractAllFromReport' => array('perm_add'),
    255 +               'extractFromReport' => array('perm_add'),
256 256                 'replaceSuggestionInReport' => array('*'),
257     -               'importReportFromUrl' => array('*'),
258     -               'sendToLLM' => ['*'],
    257 +               'importReportFromUrl' => array('perm_add'),
    258 +               'sendToLLM' => ['perm_add'],
259 259                 'configPermissionRole' => 'perm_add',
260 260                 'defaultAsPDF' => ['*'],
261 261                 'addTag' => ['perm_tagger'],
```

app/Controller/EventReportsController.php

```
@@ -483,6 +483,10 @@ public function importReportFromUrl($event_id)
483 483         if (!$this->request->is('ajax') && !$this->_isRest()) {
484 484             throw new MethodNotAllowedException(__('This function can only be reached via AJAX and via the API.'));
485 485         }
    486 +
    487 +       // throws exception if the user can't modify it
    488 +       $this->__canModifyReport($event_id);
```

Check for modification permission is missing!

Impact

- Injection of data into other orgs events, even with a read only user!
  - Event reports and some attribute type data
  - For example on a threat intel provider instance!
  - Injection of XSS payload

3. Use previously disclosed vulns as inspiration

Photo by Alexandre Dulaunoy, as found on flickr.

MISP Threat Sharing

HOME FEATURES DATA MODELS DOCUMENTATION COMMUNITIES DOWNLOAD

Another auditlog endpoint also missing ACL

- CVE-2023-4863 < MISP 2.4.176 - An issue was discovered in MISP before 2.4.176. app/Model/AppModel.php mishandles filters.
- CVE-2023-48659 < MISP 2.4.176 - An issue was discovered in MISP before 2.4.176. app/Controller/[illegible].php mishandles parameters.
- CVE-2023-48658 < MISP 2.4.176 - An issue was discovered in MISP before 2.4.176. app/Model/AppModel.php lacks a checkParam function for alphanumerics, underscore, dash, period, and space.
- CVE-2023-49926 < MISP 2.4.179 - app/Lib/Tools/EventTimelineTool.php in MISP before 2.4.179 allows XSS in the event timeline widget.
- CVE-2023-50918 < MISP 2.4.182 - app/Controller/AuditLogsController.php in MISP before 2.4.182 mishandles ACLs for new audit log features (not enabled by default).

app/Controller/AuditLogsController.php

```
@@ -222,8 +222,14 @@ public function eventIndex($eventId, $org = null)
222 222
223 223         public function fullChange($id)
224 224         {
    225 +           $acl = $this->__applyAuditACL($this->Auth->user());
225 226           $log = $this->AuditLog->find('first', [
226     -             'conditions' => ['id' => $id],
    227 +             'conditions' => [
    228 +                 'AND' => [
    229 +                     $acl,
    230 +                     'id' => $id
    231 +                 ]
    232 +             ],
227 233               'recursive' => -1,
228 234               'fields' => ['change', 'action'],
229 235           ]);
```

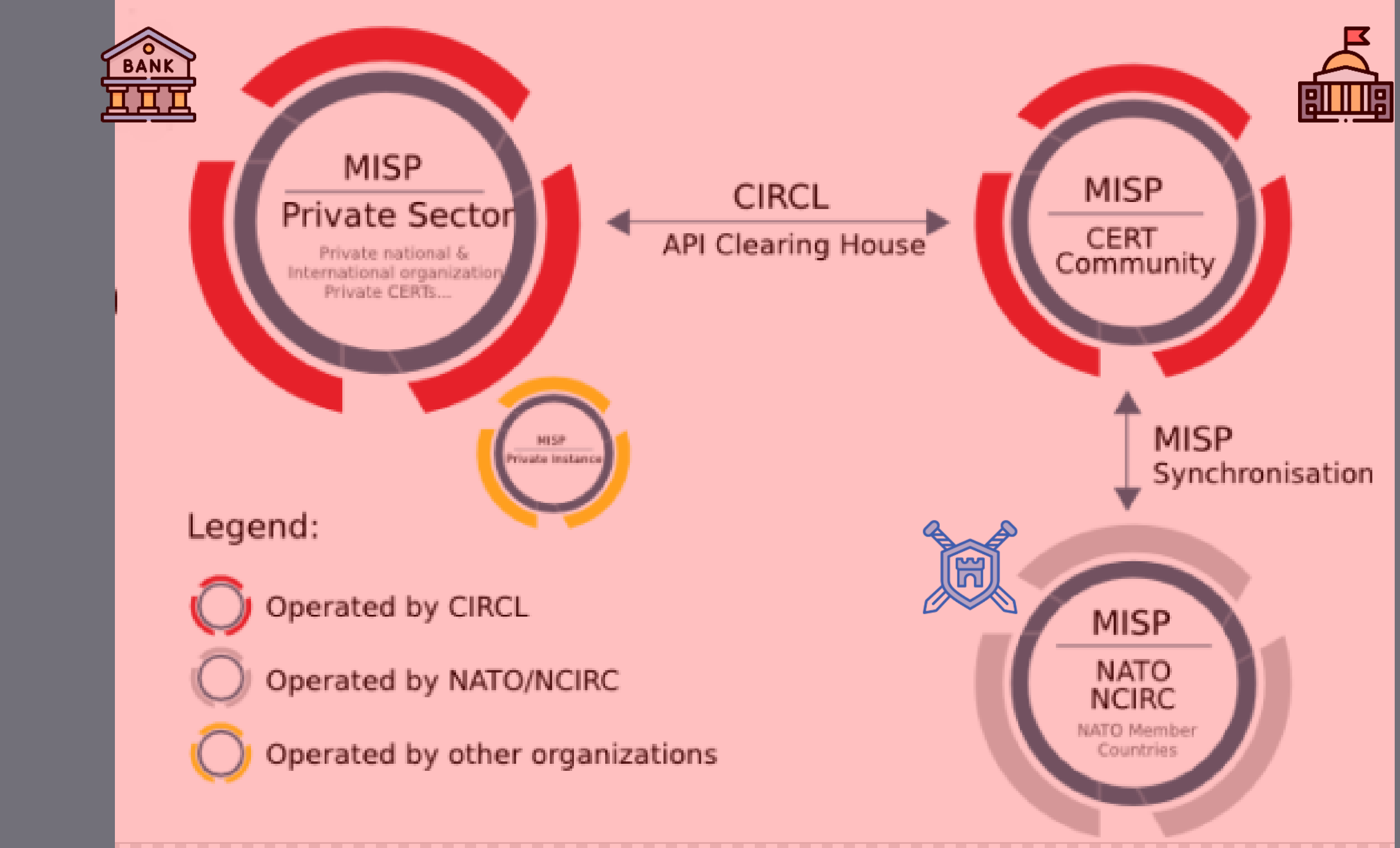
Impact

- Data leaked (events, users, ...)
- Potential lateral movement

Top 10 Web Hacking Techniques

Every year, numerous security researchers choose to share their findings with the community through conference presentations, blog posts, whitepapers, videos, and even simple disclosures. This is great, but the sheer volume and diversity means understated discoveries from aspiring researchers can be overlooked. Even flashy vulnerabilities eventually get eclipsed and forgotten, as people chase after the next shiny logo. While well-established risks are tracked by the OWASP Top Ten and Testing Guide, new threats are easily lost.

Since 2006, Jeremiah Grossman and Matt Johansen have annually collaborated with the infosec community to pick the top 10 web hacking techniques of each year. This has been invaluable in drawing deserved attention to the most exciting and innovative research to have come out of the community.


BANK
MISP
Private Sector
Private national &
International organization
Private CERTs...
MISP
Private Instance
CIRCL
API Clearing House
MISP
CERT
Community
MISP
Synchronisation
Legend:
Operated by CIRCL
Operated by NATO/NCIRC
Operated by other organizations
MISP
NATO
NCIRC
NATO Member
Countries

Placeholder – gif if exploited vulnerability

- On view of event report – XSS is triggered which
  - Turns off csp_enforce setting (if triggered by site admin) and reloads the view with a specific query param
  - On reload, creates an authkey and sends its to attacker controlled endpoint

4. Reviewing config options

Photo by Alexandre Dulaunoy, as found on flickr.

Server Settings & Maintenance

Overview | MISP (48 ⚠) | Encryption (8) | Proxy (5) | Security (10) | Plugin (634) | SimpleBackgroundJobs

Priority	Setting	Value	Description
Critical	Security.rest_client_enable_arbitrary_urls	false	**[CLI only]** Enable this setting if yo executed by the MISP server, so ir
Critical	Security.disable_local_feed_access	false	**[CLI only]** Disabling this setting wi feed sources to be network based the system that the apache user h
Recommended	Security.log_each_individual_auth_fail	true	By default API authentication failur allows administrators to more easi API keys. On the other hand, this r interesting, so if you fall into the la
Recommended	Security.rest_client_baseurl	https://192.168.252.133	If left empty, the baseurl of your M override the baseurl with a url thro

5. Just review all the code (commits) and changelogs

Photo by Alexandre Dulaunoy, as found on flickr.

test 2

ID	44
UUID	f2532c29-c8d7-4767-99b6-06e6b10f0a54
Event	#16: test html injection 2
Distribution	Inherit event
Last update	2025-04-23 20:49:27

Edit | Split Screen | Markdown | Raw | Save | Menu | Help

Download | Download PDF (via misp-module)

Impact

- None in a way, since the functionality was broken at the time of discovery ☺
- For the “fixed” functionality
  - SSRF
  - Local file disclosure (patched?)
- Recommendation: only use it on internal systems with very limited access

Consolidated findings

Photo by Markus Winkler, as found on unsplash.

Findings – currently 34 MISP “security” commits related to things I reported

Severity	Amount
Critical	? 2 minimum
High	?
Medium	?
Low	?

Findings – other CIRCL tool “security” commits related to things I reported

Tool	Commits
Cerebrate	5
[illegible]kup	7
	2
	1
Lacus	0

LACUS

Recommendations for MISP users - 1

- Update frequently
- Limit site admin usage
- Monitor site admin activity
  - Site admin user creation
  - Auth key creation
- Limit site admin specific pages to allowlisted IP ranges

Recommendations for MISP users - 2

- Update frequently
- Review authkey creations (in scope of the vulnerability I mentioned)
- Install MISP-modules on separate server or in container
- If you can, provide your feedback to the MISP project team. If you notice some bug or potential improvement, don’t be afraid to reach out

Picture by Alexandre Dulaunoy, cropped version of picture found on flickr.

Recommendations for MISP project team

- If you are interested, join the club!
- Continue fostering community engagement
- Look into optimizing automated testing flows

Personal reflections

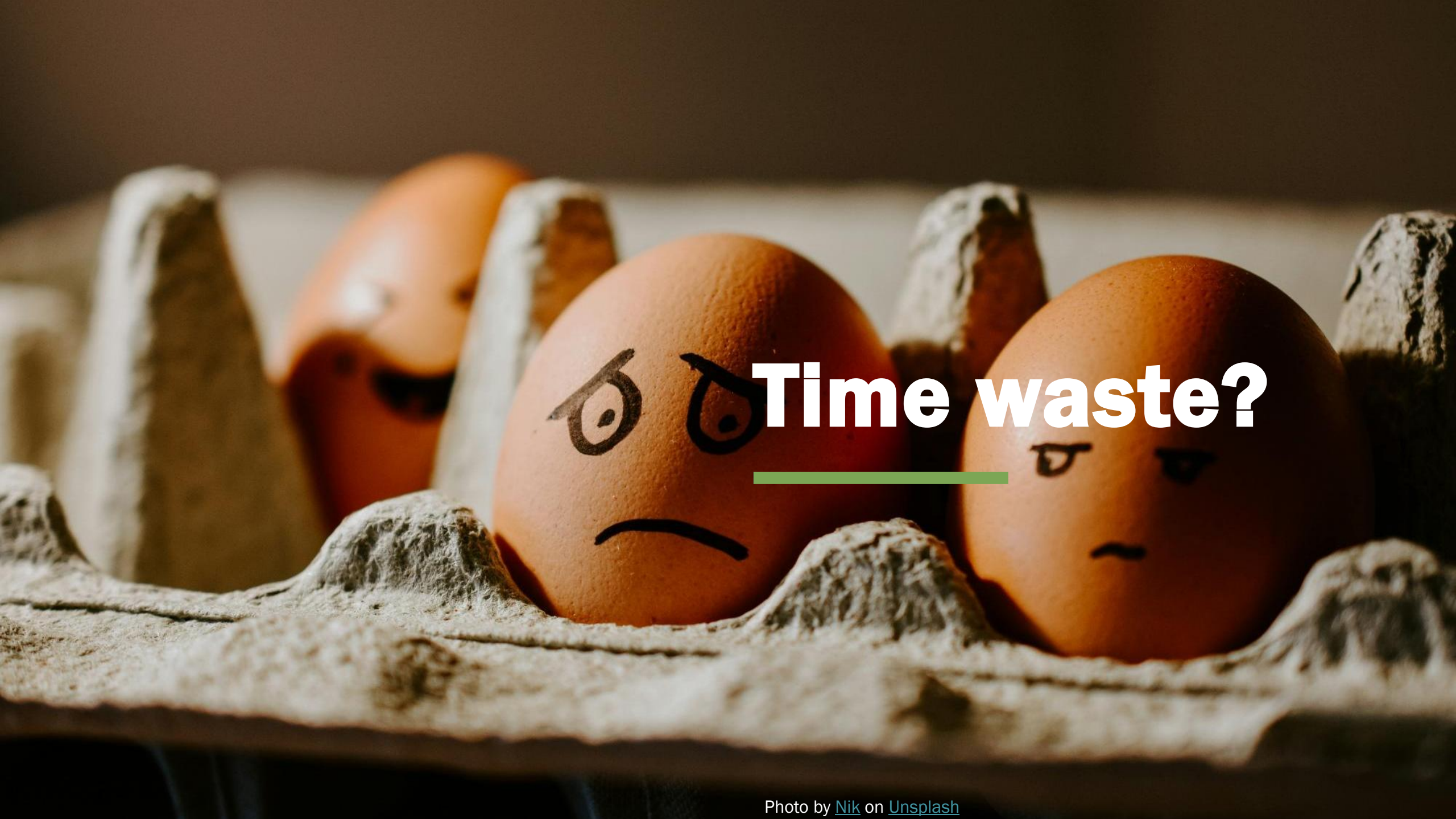


Photo by Nik on Unsplash

Photo by Alexandre Dulaunoy, cropped version of photo found on flickr.

Personal reflections

- A word on LLM usage
  - Beneficial in some areas
  - Makes you go down rabbit holes
  - Got better -> some recent usable findings
- Even reviewing commits can help / be fun
- What's next???
  - Similar work -> continue on this path
  - Sync review
  - Work on documentation, automated testing

Thank you.
Is there anyone who CAN have a question?

@wachizungu@infosec.exchange

@wachizungu

https://jeroenpinoy.website