

Project Title

IMPLEMENTING SECURITY MEASURES WITHIN A THREAT INTELLIGENCE SHARING PLATFORM

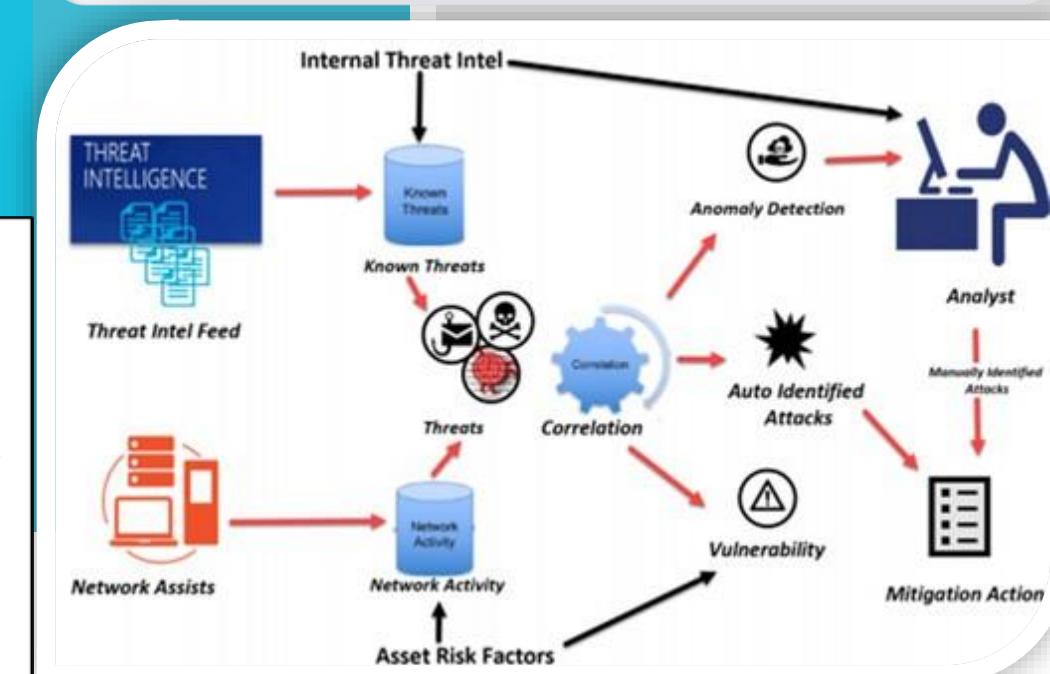
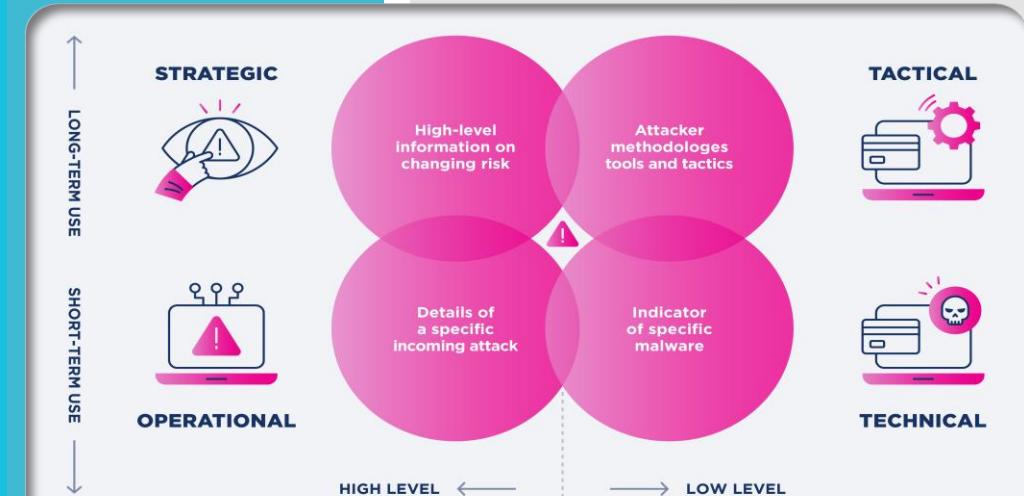
Risk Assessment,
Evaluation of Business
Objectives & Security
Needs

Choosing a Cost-
Effective,
Comprehensive,
Integrative Solution

Staff Training

Continuous
Evaluation and
Improvement

What is Threat intelligence?



Security Assessment of Existing Platform Architecture

Implementation Steps:

1. Conduct a Threat Modeling Exercise

Use **STRIDE** or **MITRE ATT&CK** to classify risks (Spoofing, Tampering, Denial of Service, etc.). Identify vulnerabilities in MISP's data ingestion, processing, and sharing mechanisms.

2. Perform Vulnerability Assessments

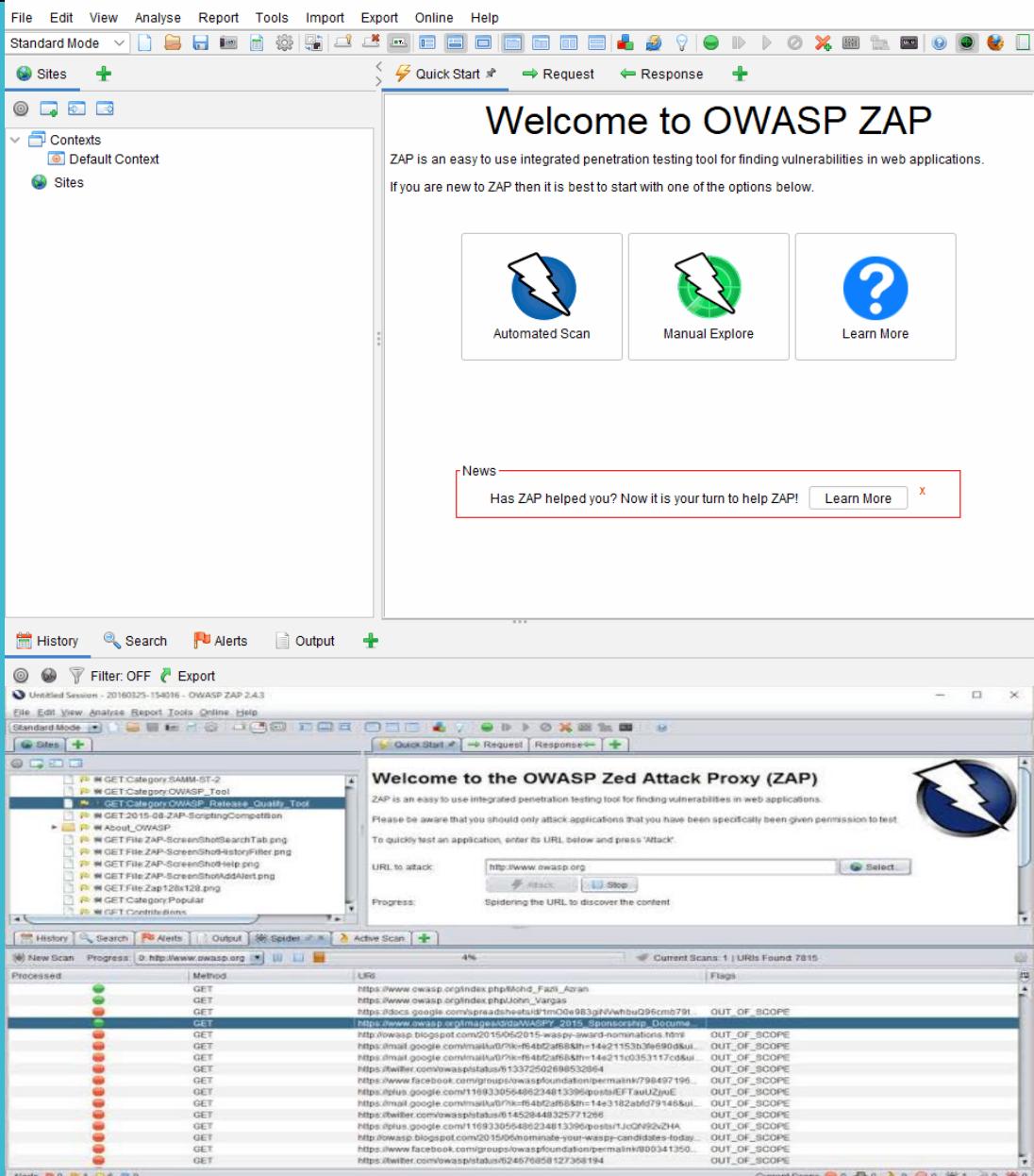
Run penetration testing using **OWASP ZAP**, **OpenVAS**, **Nessus**

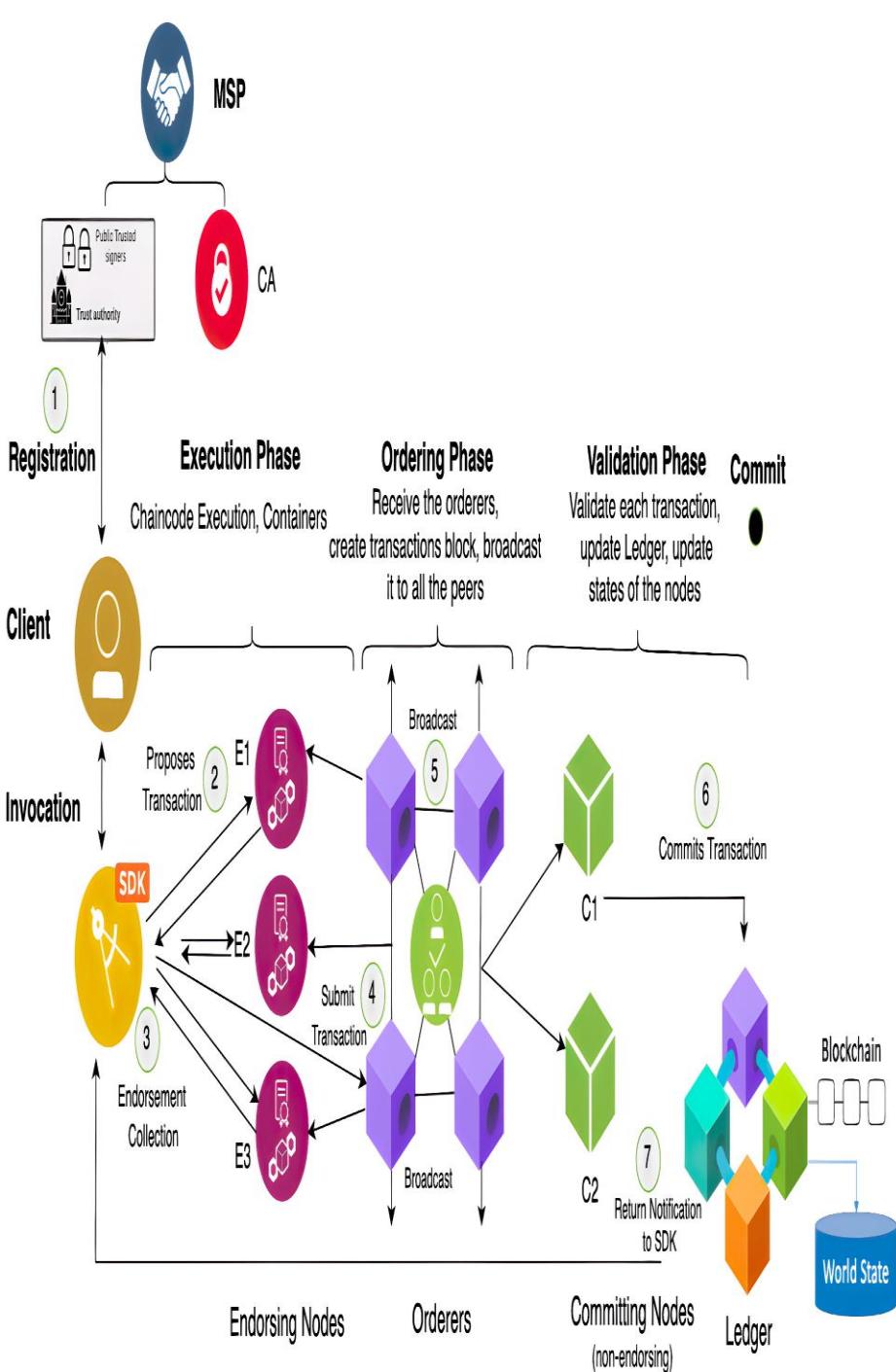
Test for **API security flaws**, **SQL injection**, **privilege escalation**

3. Review Encryption & Authentication Mechanisms

Audit **TLS configurations**, ensuring strong transport encryption (**TLS 1.3**)

Validate authentication methods (password policies, **OAuth 2.0**, **API keys**).





Key Tasks and Implementation Plan

1. Conduct a Threat Modeling Exercise

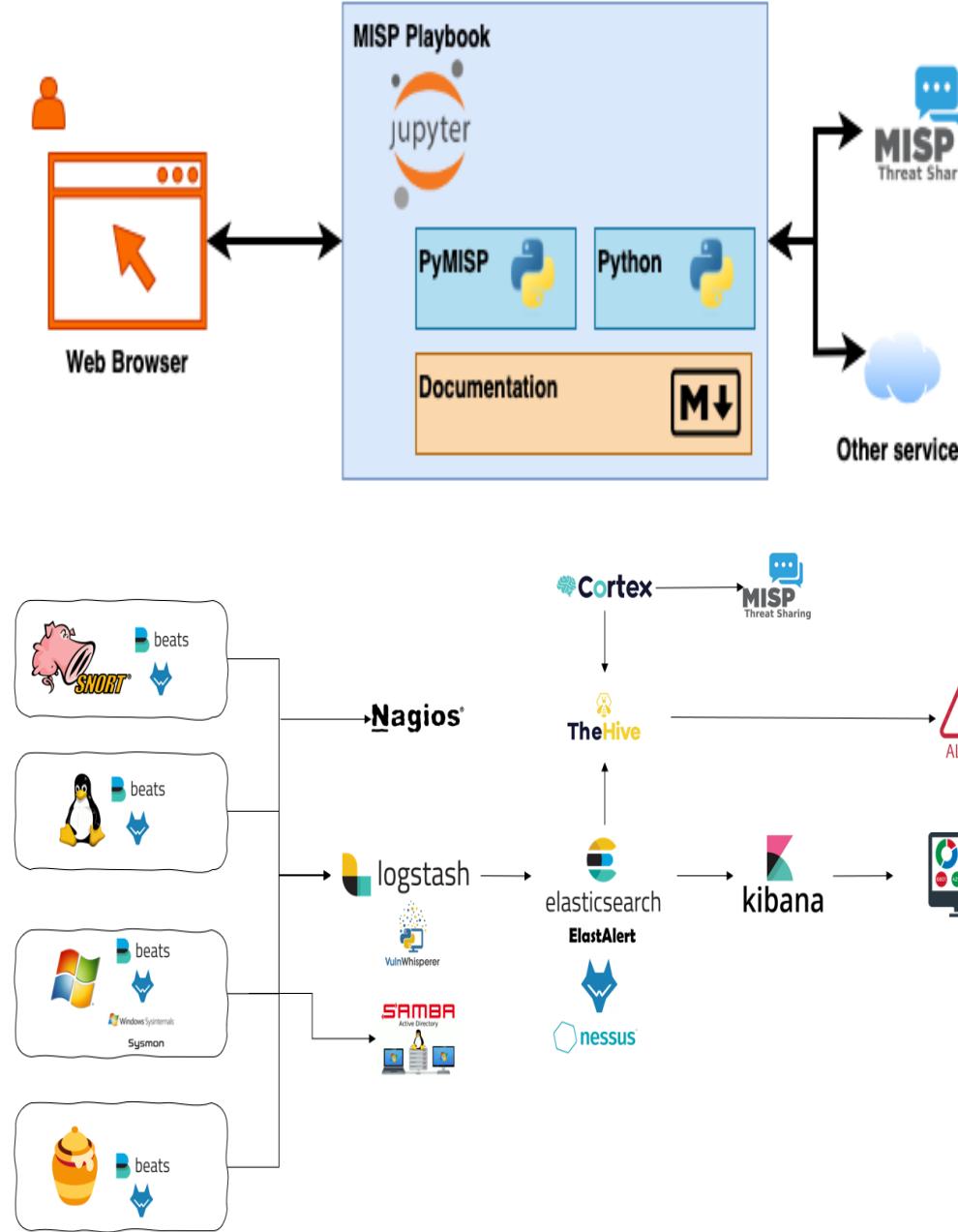
Use the **STRIDE framework** to model threats:

- ❖ **Spoofing** (fake identities)
- ❖ **Tampering** (data manipulation)
- ❖ **Repudiation** (lack of logs)
- ❖ **Information Disclosure** (sensitive data exposure)
- ❖ **Denial of Service**
- ❖ **Elevation of Privilege**

Focus areas in MISP:

- ❖ User authentication and role-based access
- ❖ API endpoints and integrations
- ❖ Event sharing (data disclosure)
- ❖ Background workers (possible DoS target)

Tool: Use **Microsoft Threat Modeling Tool** or **OWASP Threat Dragon**



2. Perform Vulnerability Assessments

- Run **OpenVAS**, **Nikto**, or **Nmap** scans on the server hosting MISP.
- Conduct **authenticated vulnerability scans** on:
 - Web UI (Apache/PHP)
 - API endpoints (access control)
 - Database (MySQL/MariaDB)

Use **OWASP ZAP** to test MISP's web interface for:

- XSS
- CSRF
- Insecure headers
- Auth bypass attempts

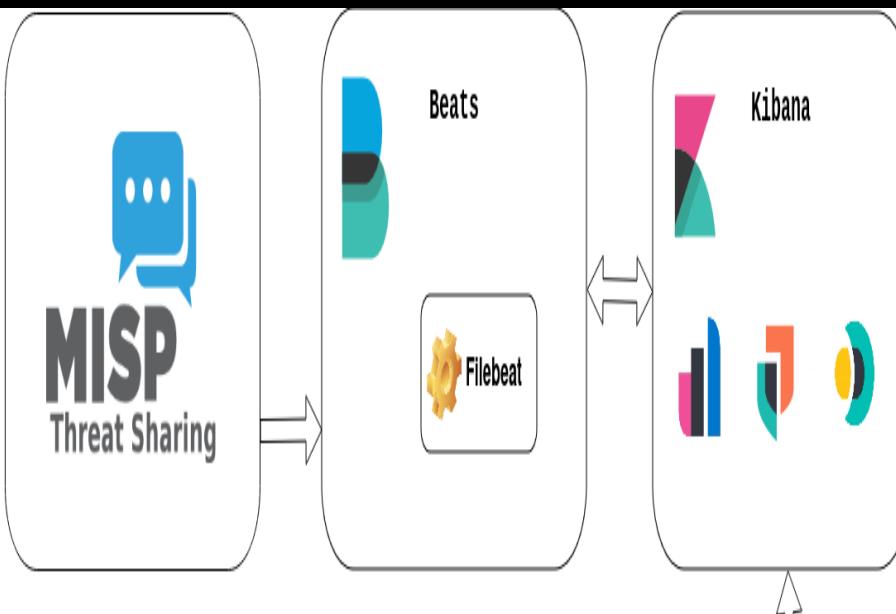
3. Review Current Security Controls

Check:

- Encryption**
 - Is HTTPS/TLS enforced for all web/API traffic?
 - Are PGP/GPG keys used to sign/share events?
- Authentication**
 - Are default users disabled?
 - Is MFA enforced (via SSO/SAML)?
- Data Privacy**
 - Are event distribution settings correctly scoped (Org Only, Sharing Groups)?
 - Are audit logs active?



Practical Step: Perform a Basic Vulnerability Scan with Nmap on MISP Server



Example

bash

```
# Scan open ports and service versions  
nmap -sV -p- <misp-server-ip>
```

Then check for vulnerabilities:

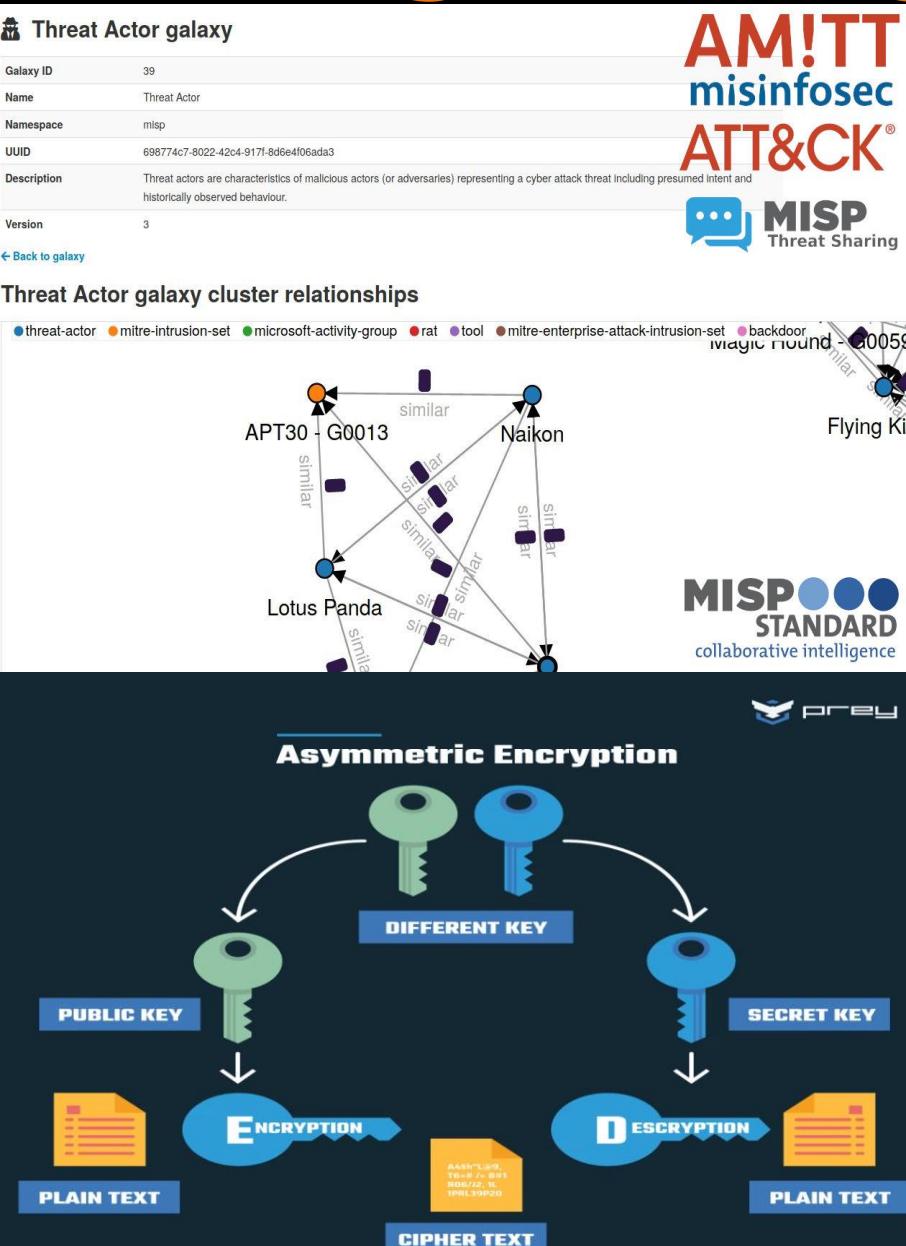
```
nmap --script vuln -p 80,443 <misp-server-ip>
```

This identifies potential issues like outdated Apache modules, SSL weaknesses, or misconfigurations.

Output might show:

PORT	State Service	Version
22/TCP	Open SSH	Open SSH 8.2
80/TCP	Open HTTP	Apache HTTPD 2.4.41
443/TCP	Open SSL/HTTP	Apache HTTPD 2.4.41 (OPEN SSL)
3306/TCP	Open MYSQL	MYSQL 5.7.31

Strengthening Data Encryption in MISP



Step 1: Secure Data in Transit (TLS/HTTPS)

Goal: Prevent unauthorized access or interception of data exchanged between users, MISP, and integrated systems.

Actions:

1. Enable HTTPS (TLS 1.2 or 1.3)

- ❖ Use a valid SSL/TLS certificate (e.g., Let's Encrypt or enterprise CA).
- ❖ Configure Apache/Nginx (depending on your MISP setup) for HTTPS:

```
sudo a2enmod ssl
```

```
sudo a2ensite default-ssl
```

```
sudo systemctl restart apache2
```

2. Configure TLS securely

- Disable weak ciphers and SSL versions:

```
SSLProtocol all -SSLv3 -TLSv1 -TLSv1.1
```

```
SSLCipherSuite HIGH:!aNULL:!MD5
```

Data Encryption and Protection



Strengthening Data Security with CSRC Encryption

Enhanced Confidentiality

Robust Authentication



3. Force HTTPS

- ❖ Enforce HTTPS redirect in web server config or via HTTP headers (HSTS).

4. Secure API Connections

- ❖ MISP's REST API should only be accessed via HTTPS.
- ❖ Validate certificates in all integrations (e.g., PyMISP, other MISP peers).

Step 2: Encrypt Data at Rest

Goal: Prevent access to stored data if the server is compromised.

Actions:

1. Use Full Disk Encryption (FDE):

Enable LUKS (Linux Unified Key Setup) on MISP server volumes
This encrypts the entire disk and requires passphrase/key at boot.

Encrypt Database Storage:

- Use MySQL with InnoDB tablespace encryption:

ALTER INSTANCE ROTATE INNODB MASTER KEY;

Enable “innodb_encrypt-tables and innodb_encrypt_log”.

Encrypt MISP Attachments (Optional):

Store attachments (malware samples, documents) using gpg or AES-256.

Protect file directories using file-level encryption like enCryptfs

Strengthening Data Encryption and Protection

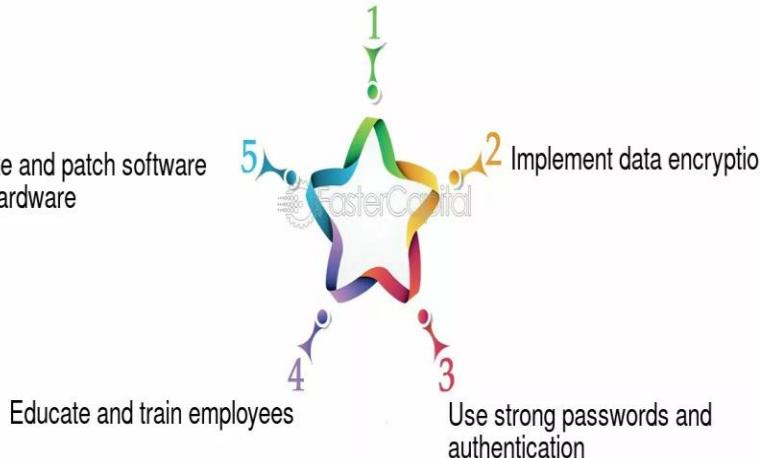
Encrypting data is a crucial aspect of modern payment systems, as it ensures the protection of sensitive information during transmission and storage.



Strengthening Data Security Measures

Conduct a data security audit

Update and patch software and hardware



Backup Encryption:

- Use gpg or openssl to encrypt backups before storage or transfer..
- Automate with cron jobs:

```
mysqldump ... | openssl enc -aes-256-cbc -e -out backup.sql.enc
```

Step 3: Implement Secure Key Management

Goal: Protect encryption keys from unauthorized access.

Actions:

1. Use a Key Management System (KMS):

Options:

- a) HashiCorp Vault
- b) AWS KMS / Azure Key Vault / GCP KMS (if using cloud)

Store MySQL master keys, TLS certs, and backup keys securely.

2. Best Practices:

- a) Rotate keys periodically (automate if possible).
- b) Use separate keys for different functions (e.g., database vs. file encryption).
- c) Restrict key access to only required services/users (principle of least privilege).

3. Integrate Vault with MISP (optional advanced):

- a) Use Vault CLI/API to load secrets dynamically at runtime.
- b) Example: Pull DB passphrase on system boot and mount encrypted volum

% time	seconds	usecs/call	calls	errors	syscall	00:42	0.004517	52156	2	1	wal4
45.42	0.000798	49	16		pselect6	17.97	0.017397	138	126	4	access
23.96	0.000421	23	18		read	3.50	0.003388	42	80	13	openat
12.01	0.000211	211	1		clone	1.01	0.000975	243	4	4	stat
5.24	0.00092	13	7		write	0.73	0.000705		73		getdents64
4.15	0.000073	4	16		ioctl	0.65	0.000630				close
2.96	0.000052	3	15		rt_sigprocmask	0.60	0.000580		28		write
2.05	0.000036	0	43		rt_sigaction	0.42	0.000410		76		mmap
1.42	0.000025	25	1		pipe	0.40	0.000391		68	2	ioctl
0.91	0.000016	1	10		close	0.35	0.000341		100		fstat
0.80	0.000014	7	2		setsockopt	0.23	0.000276		6		rt_sigaction
0.74	0.000013	6									pread64
0.23	0.000004	4									unlink
0.06	0.000001	1									poll
0.06	0.000001	1									select
0.00	0.000000	0									clone
0.00	0.000000	0									mprotect
0.00	0.000000	0									socket
0.00	0.000000	0									brk
0.00	0.000000	0									munmap
0.00	0.000000	0									2 connect
0.00	0.000000	0									rt_sigprocmask
0.00	0.000000	0									select
0.00	0.000000	0									fcntl
0.00	0.000000	0									lseek
0.00	0.000000	0									geteuid
0.00	0.000000	0									getpid
0.00	0.000000	0									uname
0.00	0.000000	0									pipe
0.00	0.000000	0									getcwd
0.00	0.000000	0									getegid
0.00	0.000000	0									getuid
0.00	0.000000	0									arch_prctl
100.00	0.001757		170	5	total		0.01	0.00009	4	2	1 arch_prctl

AUDITD



Step 4: Monitor and Audit

Goal: Detect misconfigurations and unauthorized access attempts.

Actions:

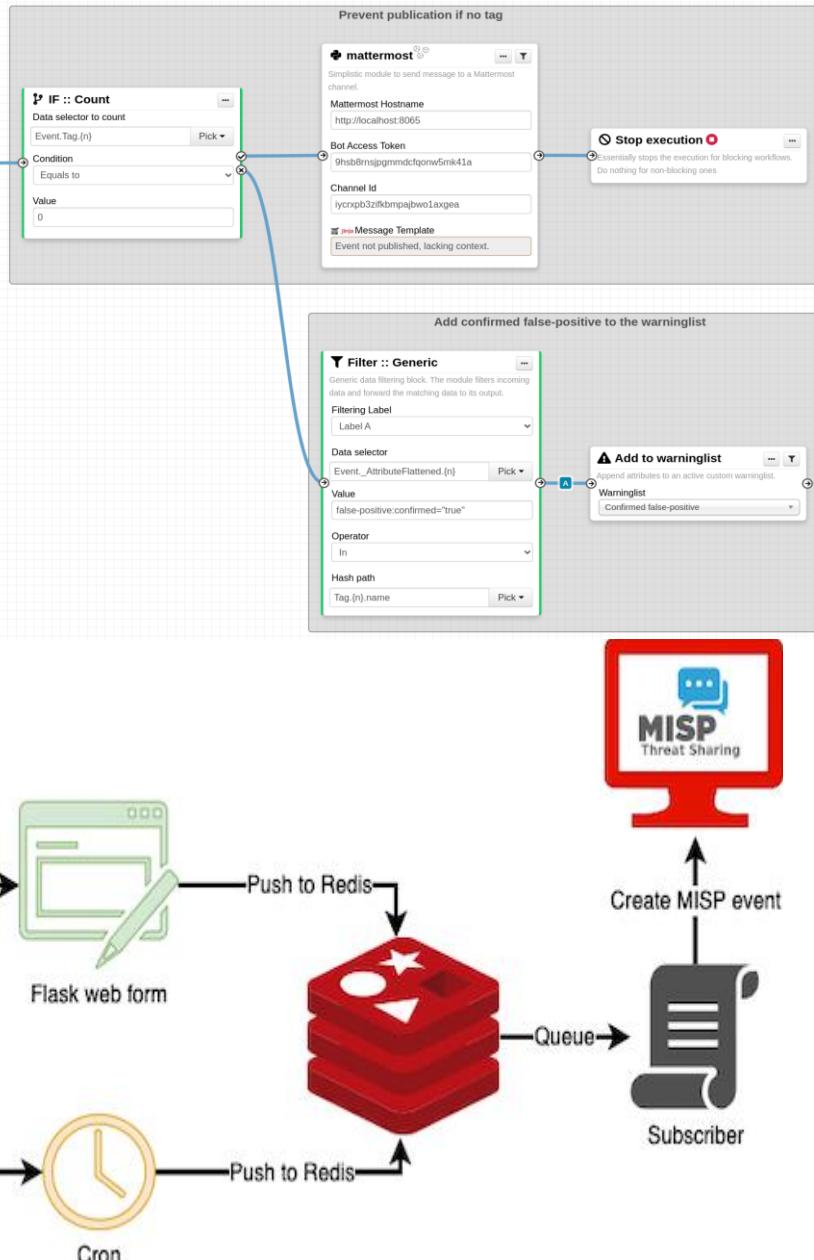
- Enable logging for:
 1. TLS handshakes
 2. Access to encrypted volumes
 3. Key usage and rotation
- Use tools like **Auditd**, **Filebeat**, or **OSSEC** for file and key access auditing.
- Regularly review **MISP logs** in /var/www/MISP/app/tmp/logs

Step 5: Documentation & Maintenance

1. Maintain a **data encryption policy** with:
 - Encryption algorithms used
 - Key lifecycle and access control rules
 - TLS certificate renewal schedule
2. Train admins on secure handling of encrypted assets and key material.



Enhancing Authentication and Authorization in MISP



Step 1: Implement Multi-Factor Authentication (MFA)

- Objective:** Add a second layer of security beyond username/password

Actions:

1. Enable TOTP-based MFA in MISP (Time-based One-Time Password)

MISP supports Google Authenticator or any TOTP-compatible app.

- ❖ Go to Administration → List Users
- ❖ Click on the target user → Enable Two-Factor Authentication
- ❖ Scan QR code in authenticator app (Google Authenticator, Authy, etc.)

2. Enforce MFA Platform-wide

- In config.php (`/var/www/MTSP/app/config/config.php`) set:

```
Configure::write('Security.require_mfa', true);
```

Enhancing Online Security: The Role of Multi-Factor Authentication

[Read More](#)



IT: CYBERSECURITY

SINGLE SIGN-ON VS. MULTI-FACTOR AUTHENTICATION



3. Log & Monitor MFA Events

- All login attempts (success/failure) are logged in:

`/var/www/MISP/app/tmp/logs/*`

Step 2: Configure Role-Based Access Control (RBAC)

- Objective: Assign permissions based on user responsibilities.

Actions:

1. Define Roles in MISP
2. Go to Administration → Roles
3. Predefined roles:
 - I. Org Admin
 - II. Site Admin
 - III. User
4. You can also create custom roles (e.g., Read-Only Analyst, Data Contributor, Threat Analyst)

1. Assign Specific Permissions to Roles

For each role, set:

- Event publishing rights
- Tagging rights
- Access to specific object types
- Sync permissions with other MISP instances

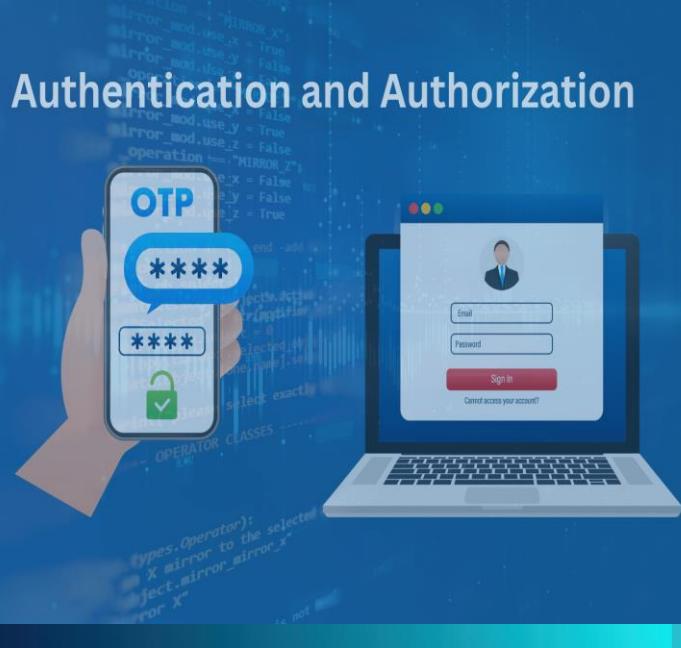
2. Assign Users to Roles

- Go to Administration → Users
- Edit the user → Choose appropriate role from dropdown

3. Use Orgs for Isolation

Group users into **organizations** to segregate data and access across teams or partners.

Authentication and Authorization



Step 3: Integrate with Single Sign-On (SSO)

Objective: Streamline secure user access through identity federation.

🔧 MISP Supports SSO via:

- SAML 2.0
- LDAP/AD
- HTTP Header-based Auth (e.g., behind a proxy like Keycloak or Shibboleth)

🔧 Actions (SAML / Keycloak Example):

1. Use a Reverse Proxy (Apache/Nginx)

Place MISP behind Keycloak, which handles SSO.

Use mod_auth_opendc for Apache.

2. Configure HTTP Header-based Login in MISP

In config.php, enable header-based login:

```
Configure::write('Security.auth_enforced', true);
```

```
Configure::write('Security.auth_user_header', 'HTTP_REMOTE_USER');
```

All users

User type == Guest

Manage view

3. Set User Creation via Header

- You can auto-create users if they don't exist in MISP:

```
Configure::write('Security.advanced_authkeys', true);
```

4. SSO Providers Supported:

Keycloak (OpenID Connect)

Okta / Azure AD / ADFS (SAML 2.0)

LDAP via Apache's mod_ldap.

Step 4: Review and Monitor Access

- Periodically review
- Role assignments
- MFA adoption logs
- SSO login audits
- Remove inactive users or expired sessions

Enable session expiration and timeouts:

```
Configure::write('Session.timeout', 30); // in minutes
```

Guest Sign-in Time	Signed-in User	Signed-in Application Name	City	State	Country	Device Browser	Device OS with Version
5/9/2023 21:42:28 PM	ross@0365droid.onmicrosoft.com	Microsoft Account Controls..	Chennai	Tamil Nadu	IN	Chrome 112.0	Windows 10
5/9/2023 21:41:51 PM	ross@0365droid.onmicrosoft.com	Office365 Shell WCSS-Client	Chennai	Tamil Nadu	IN	Chrome 112.0	Windows 10
5/9/2023 21:41:51 PM	ross@0365droid.onmicrosoft.com	Office365 Shell WCSS-Client	Chennai	Tamil Nadu	IN	Chrome 112.0	Windows 10
5/9/2023 21:41:50 PM	ross@0365droid.onmicrosoft.com	Office365 Shell WCSS-Client	Chennai	Tamil Nadu	IN	Chrome 112.0	Windows 10
4/18/2023 12:54:02 PM	andrea.powell@0365droid.onmicrosoft.com	Microsoft Teams Web Client	Chennai	Tamil Nadu	IN	Chrome 112.0	Windows 10
4/18/2023 12:54:02 PM	andrea.powell@0365droid.onmicrosoft.com	Microsoft Teams Web Client	Chennai	Tamil Nadu	IN	Chrome 112.0	Windows 10
4/14/2023 4:20:06 AM	andrea.powell@0365droid.onmicrosoft.com	SharePoint Online Web Client	Chennai	Tamil Nadu	IN	Chrome 111.0	Windows 10
4/14/2023 3:37:54 AM	andrea.powell@0365droid.onmicrosoft.com	Microsoft Teams Web Client	Chennai	Tamil Nadu	IN	Chrome 111.0	Windows 10
4/6/2023 10:56:41 AM	andrea.powell@0365droid.onmicrosoft.com	Microsoft Teams	Chennai	Tamil Nadu	IN	Edge 18.19044	Windows 10
4/6/2023 10:41:43 AM	andrea.powell@0365droid.onmicrosoft.com	Microsoft Teams Web Client	Chennai	Tamil Nadu	IN	Chrome 111.0	Windows 10

Improving Access Control for Threat Intelligence Data in MISP

Add Role

Name	Permission
	Read Only
<input type="checkbox"/> Perm Admin	<input type="checkbox"/> Perm Sync
<input type="checkbox"/> Perm Audit	<input type="checkbox"/> Perm Regexp Access
<input type="checkbox"/> Perm Tagger	<input type="checkbox"/> Perm Template
<input type="checkbox"/> Perm Sharing Group	<input type="checkbox"/> Perm Delegate

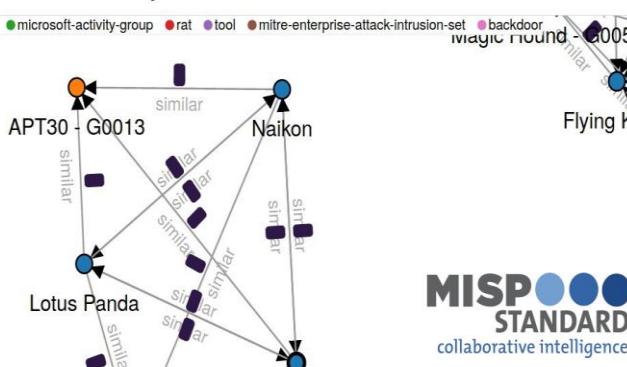
Add

Threat Actor galaxy

Galaxy ID	39
Name	Threat Actor
Namespace	misp
UUID	698774c7-8022-42c4-917f-8d6e4f06ada3
Description	Threat actors are characteristics of malicious actors (or adversaries) representing a cyber attack threat including presumed intent and historically observed behaviour.
Version	3

[← Back to galaxy](#)

Threat Actor galaxy cluster relationships



Step 1: Review & Update Access Control Lists (ACLs)

MISP uses **roles and organizations** to enforce access control.

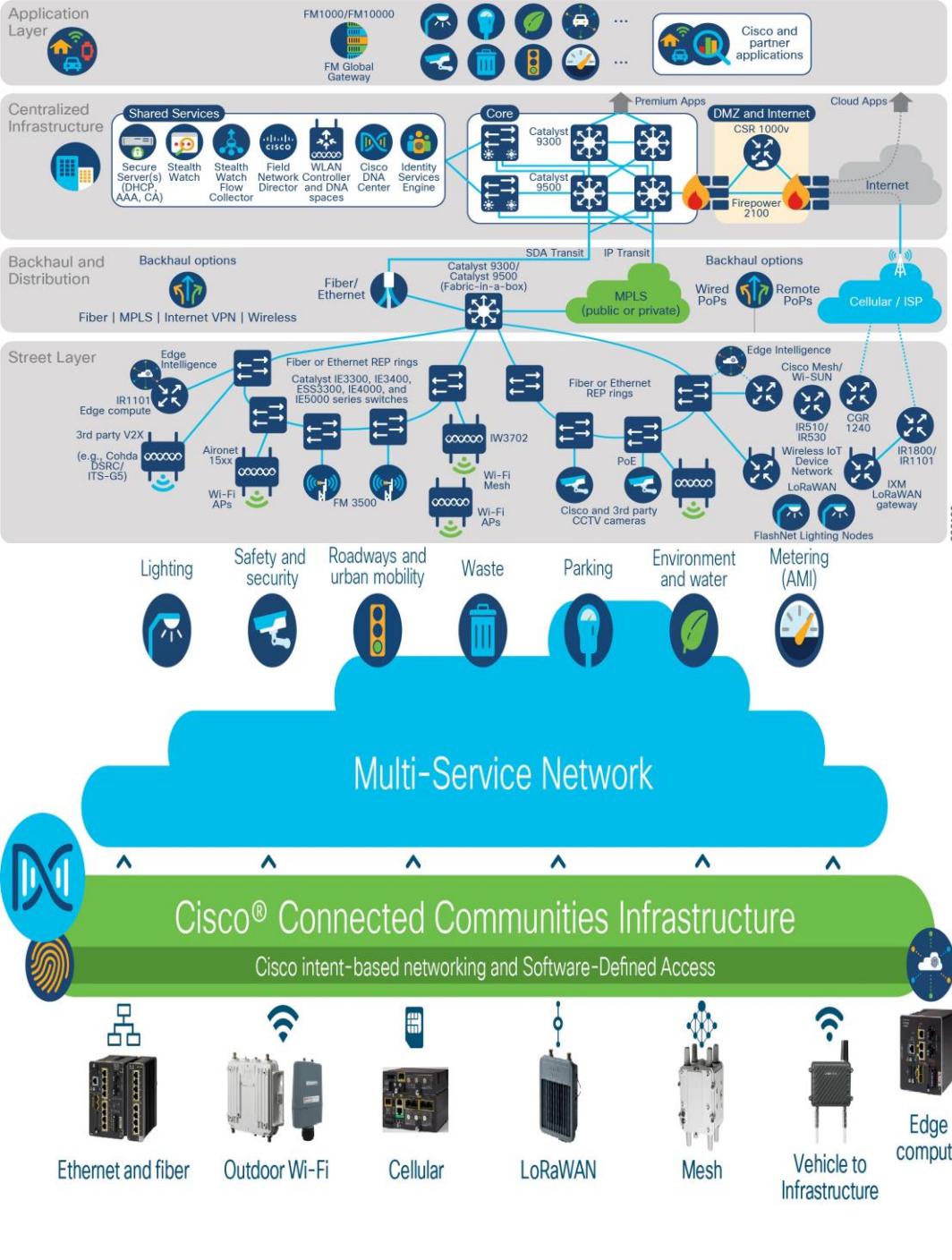
Users are grouped under organizations, and their access depends on:

- ❖ Role permissions
- ❖ Event distribution settings
- ❖ Sharing groups

Actions:

1. Review User Roles

- ❖ Go to: Administration → Roles
- ❖ Check role settings for:
 - Perm_add
 - Perm_modify
 - Perm_publish
 - Perm_sync
 - Perm_site_admin



2. Audit Go to: Administration → Users

- Review user role and org alignment
- Disable or delete unused accounts
- Existing Users

3. Limit Event Access via Distribution Settings

MISP event Distribution levels:

- 0 – Your Organization Only
- 1 – This Community Only
- 2 – Connected Communities
- 3 – All Communities
- 4 – Sharing Group

Use distribution = 0 or 4 (with custom Sharing groups) for restricted intelligence.

Step 2: Implement Granular Read/Write Controls via Roles and Tags

MISP supports fine-grained control using roles, tag filters, and object-level distribution.

Create Fine-Grained Roles

- Go to: Administration → Roles → Add Role
- Customize permissions:
 - ❖ Read-only (disable perm_add, perm_modify)
 - ❖ Analysts with limited object creation rights.
 - ❖ Admin with full event visibility.

View Event

Emotet Cobalt Strike Infection on Windows machine

Event ID	379
UUID	aab601cf-5679-4dcb-84c9-657fedf7715
Creator org	THM-MISP
Creator user	Analyst@THM.thm
Tags	Missing taxonomies: priority-level, tlp, workflow
Date	2022-03-07
Threat Level	2 High
Analysis	Completed
Distribution	This community only
Info	Emotet Cobalt Strike Infection on Windows machine
Published	No
#Attributes	79 (8 Objects)
First recorded change	2022-03-10 20:05:55
Last change	2022-03-13 17:59:02
Modification map	
Sightings	0 (0 - restricted to own organization only)

- Photos - Galaxy - Event graph - Event timeline - Correlation graph - ATTACK matrix - Event reports - Attributes - Discussion

#379: Emotet Cobalt...

Galleries

+ previous next + view all

Home Event Actions Dashboard Getaway Input Filters Global Actions API

Events

My Events Org Events

Published	Creator org	Clusters	Tags	Mins	#Cont	Date	Last modified at	Info	Distribution	Actions
✓	Threat Actor	0	green	0	1	2018-10-28	2018-09-23 13:50:09	CSINT - Operation SAIN (Ninety)	All	•
✓	CthulhuSPL.be	8	green	0	1	2018-10-02	2018-02-05 08:30:37	CSINT Shellshock scanning IPs from OpenDNS	All	•
✓	CthulhuSPL.be	8	green	0	1	2018-10-20	2017-06-22 22:03:38	CSINT OrisAPK - A while of a fake blog post by PWIC	All	•
✓	CthulhuSPL.be	8	green	0	1	2018-09-01	2015-21-23 15:53:40	CSINT Watching Attackers Through Vtunstat blog post by Brandon Dixon (Stipule)	All	•
✓	CthulhuSPL.be	8	green	0	1	2018-10-23	2014-10-29 01:41:08	Expansion on CSINT Operation Pawn Storm: The Red in SEDNT from Trend Micro	All	•
✓	CthulhuSPL.be	8	green	0	1	2018-10-11	2014-10-14 11:53:20	CSINT Shellshock exploitation from Red Sky Weekly blog post	All	•
✓	CthulhuSPL.be	8	green	0	1	2018-10-09	2014-10-13 10:17:38	CSINT Democracy in Hong Kong Under Attack blog post from Voluntary (Steven Adler)	All	•
✓	CthulhuSPL.be	8	green	0	1	2018-10-09	2014-10-10 11:16:03	CSINT Evolution of the Nuclear Exploit Kit by Cisco Talos group	All	•
✓	CthulhuSPL.be	8	green	0	1	2018-10-03	2014-10-06 08:12:57	CSINT New Indicators of Compromise for APT Group Nitro Uncovered blog post by Paul Alton Networks	All	•

Page 1 of 1, showing 8 records out of 8 total, starting at record 1, ending at 8

+ previous next +

- Use Tag-Based Access Filtering
- Tag events (e.g., tlp:red , classified , internal)
- Create roles or shaping groups that exclude/limit certain tags

Example:

Prevent users from exporting tlp:red data

Leverage Object-Level Distribution

You can restrict individual attributes/objects in an event:

- I. Set object distribution different from the parent event
- II. Great for cases where one part of an event is more sensitive

Step 3: Enforce Trusted Sharing Policies

MISP allows verified sharing through organizations and Sharing Groups.

Actions:

1. Use Sharing Groups for Trust Control

- I. Go to: Sync Actions → List Sharing Groups → Add Sharing Group
- II. Add only trusted orgs
- III. Set required conditions:

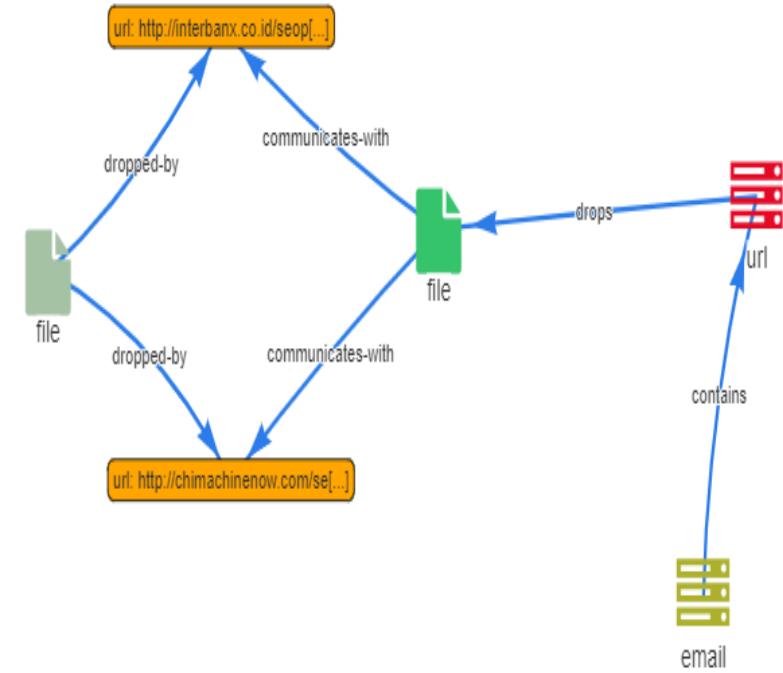
Must be in the same trust level

Must accept your terms (e.g., ISAC agreements)

2. Restrict Attribute Distribution

Even within shared events, sensitive attributes can be scoped using:

- Distribution = o (Org only)
- Distribution = 4 ith a selective Sharing Group



3. Configure Feed Access Control

If you're exposing MISP feeds externally:

- Use authentication tokens or IP whitelisting.
- Ensure feeds only show distribution ≤ 2 unless controlled.

Step 4: Monitoring & Audit

Log Access Events

Check MISP logs: /var/www/MISP/app/tmp/logs/

Log types:

1. User access
2. Exploit requests
3. Failed logins
4. Distribution violation (if audit enabled)

Feeds

Generate feed lookup caches or fetch feed data (enabled feeds only)

[Load default feed metadata](#) [Cache all feeds](#) [Cache freetext/CSV feeds](#) [Cache MISP feeds](#) [Fetch and store all feed data](#)

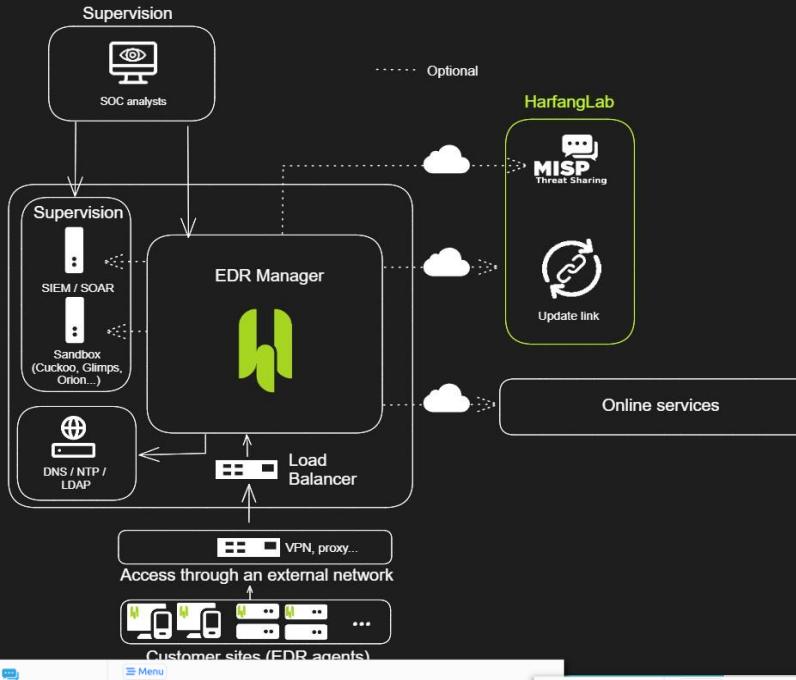
« previous 1 2 next » last

Default feeds		Custom feeds		All feeds	Enabled feeds				
<input type="checkbox"/>	Id	Enabled	Caching	Name	Format	Provider	Org	Source	URL
<input type="checkbox"/>	1	✓	✓	CIRCL	misp	CIRCL	network	https://www.circl.lu/doc/misp/feed-osint/	
<input type="checkbox"/>	2	✓	✓	Botvrij.eu	misp	Botvrij.eu	network	https://www.botvrij.eu/data/feed-osint	
<input type="checkbox"/>	9	✓	✓	Tor exit nodes	csv	TOR Node List from dan.me.uk	network	https://www.dan.me.uk/torlist/?exit	

Enable Auditing

- ❖ Periodically export user role and sharing group audits
- ❖ Review access to tlp:red or org-only tagges items.

Practical Implementation Step: Create a MISP Export Filter Module



This example shows how to mask email addresses in MISP attributes before sharing.

Step-by-Step: Create a Custom MISP Export Module

1. Create a new Python module

- Location:

```
/var/www/MISP/mispmodules/misp_modules/export_mod/anon_export.py
```

2. Example Code: Email Masking Export Module

The screenshot shows the MISP Modules interface. On the left, a sidebar lists 'History', 'History Session', 'History Tree', 'Config', and 'MISP Modules'. The main area displays a query result for 'typosquatting-finder.circl.lu'. It shows an 'Input Attribute' of 'domain' and a 'Module' of 'circl_passivedns'. The results pane shows a single entry: 'circl_passivedns' with a status of 'Errors'. Below this, a detailed view of the 'circl_passivedns' module is shown, listing 'rrtype', 'rname', and 'rdtype' fields with their respective values and query counts.

The screenshot shows a code editor window with Python code for an 'anon_export.py' module. The code defines a function 'mask_email' that takes an email address as input and returns a masked version where the characters between the first and last '@' symbols are replaced by asterisks. The code includes comments explaining the purpose and usage of the function.

```
import re

def mask_email(email):
    """
    Masks an email address by replacing the characters between the first and last '@' symbols with asterisks.

    Args:
        email: The email address to mask.

    Returns:
        The masked email address.
    """

    match = re.match(r"^(.+)@(.+)$", email)
    if match:
        first_char = match.group(1)
        last_part = match.group(3)
        masked_part = "*" * len(match.group(2))
        return first_char + masked_part + last_part
    else:
        return email

# Example usage
email_address = "example.user@domain.com"
masked_email = mask_email(email_address)
print(f"Original email: {email_address}")
print(f"Masked email: {masked_email}")
```

```
MISP-5.tlp - krehan@192.168.8.128:22 - Bitvise xterm - krehan@misp5: ~
```

```
[Unit]
Description=System-wide instance of the MISP Modules
After=network.target

[Service]
User=www-data
Group=www-data
WorkingDirectory=/usr/local/src/misp-modules
Environment="PATH=/var/www/MISP/venv/bin"
ExecStart=/var/www/MISP/venv/bin/misp-modules
#ExecStart=/var/www/MISP/venv/bin/misp-modules -l 127.0.0.1 -s

[Install]
WantedBy=multi-user.target
```

The screenshot shows the MISP web interface with the following details:

- Top Bar:** Home, Event Actions, Input Filters, Global Actions, Sync Actions, Admin.
- Left Sidebar:** List Events, Add Event, Import From MISP Export, List Attributes, Search Attributes, View Proposals, Events with proposals, Export (highlighted with a red arrow), Automation.
- Main Content:** Title "Events".
 - Navigation: « previous, 1, 2, 3, next ».
 - Search: Q, My Events, Org Events.
 - Table Headers: Published, Org, Owner Org, Id, Tags.
 - Data Rows:
 - Row 1: Published (checkbox checked), Org (MISP), Owner Org (MISP), Id (145), Tags (circ:incident-classification="X", circ:incident-classification="Info leak", hophop).
 - Row 2: Published (checkbox checked), Org (MISP), Owner Org (MISP), Id (95), Tags (Type:OSINT tip:, circ:incident-classification="m").

A red arrow points to the "Export" link in the sidebar, and the text "Click to go" is overlaid on the interface.

3. Activate the Module

```
cd /var/www/MISP/misp-modules
```

```
sudo systemctl restart misp-modules
```

4. Use the Export Module in MISP

- Go to an Event
- Choose Export → Custom Format
- Select anon_export module.

Final Notes

This module:

- ❖ Applies masking to emails before export.
- ❖ Can be extended to handle IPs, phone numbers, etc.
- ❖ Helps maintain GDPR/CCPA compliance by reducing risk of sharing PII.



Practical Implementation Step: Connect MISP to Splunk for Real-Time Threat Detection

Add misp

Name *	misp_covid	Enter a unique name for the data input
Interval *	3600	Time interval of input in seconds.
Index *	misp	
MISP url *	https://covid-19.iglocska.eu	provide MISP URL. Do not end with a /
MISP API key *	provide one authkey for the instance
Check MISP certificate	<input type="checkbox"/>	
MISP CA path	<input type="text"/>	
Use proxy settings	<input type="checkbox"/>	Use proxy settings for default instance
Use a client certificate	<input type="checkbox"/>	Use a client certificate to authenticate on default instance
Client certificate file	<input type="text"/>	

```
| mispgetioc misp_instance=misp_covid add_description=true category="External analysis,Financial fraud,Internal reference,Network activity,Other,Payload delivery,Payload installation,Payload type,Persistence mechanism,Person ,Social network,Support Tool,Targeting data" type="ip-dst" to_ids=true geteventtag=true warning_list=true limit=0 last=1d  
| eval ip=misp_value  
| eval description = tostring(misp_event_info)."|" .tostring(misp_category).|" .tostring(misp_comment)  
| eval weight = 1  
| table description,ip,weight  
| outputlookup append=true misp_es_ip_intel
```

Here's how to **integrate MISP with Splunk** for live threat feed ingestion:

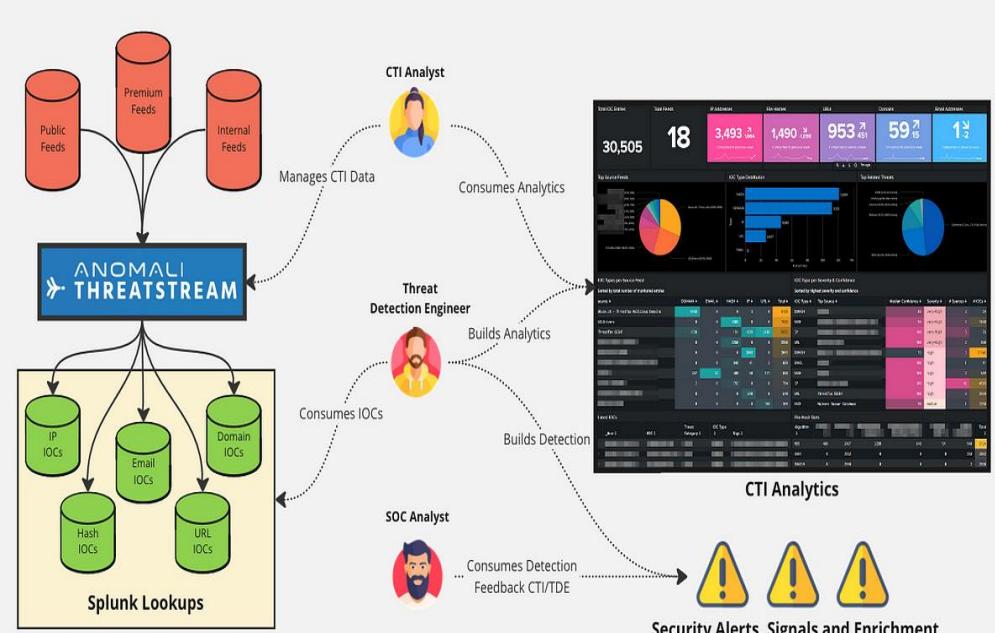
🔧 Step-by-Step Integration: MISP to Splunk

1. Install MISP42Splunk Add-on

- From: <https://github.com/MISP/misp42splunk>
`git clone https://github.com/MISP/misp42splunk.git
cp -r misp42splunk /opt/splunk/etc/apps/`

2. Configure the MISP Add-on in Splunk

- ❖ Go to the Splunk web UI
- ❖ Navigate to **Settings > Data Inputs > MISP Events**
- ❖ Add your MISP API URL, API key, and fetch schedule
- ❖ Choose data filters (e.g., by tag, org, time window)



3. Enable Scheduled Fetching of MISP Events

- Configure the fetch interval (e.g., every 15 mins)
- Choose output index (e.g., threatintel)

4. Search MISP Data in Splunk

`index=threatintel sourcetype=misp:event`

You can now correlate these with logs for **real-time alerting and detection.**

Outcome

- ❖ Real-time threat data from MISP available in SIEM.
- ❖ Uses industry protocol (API) securely.
- ❖ Can be extended to TAXII/STIX consumption later.



Practical Implementation Step: Enable and Export MISP Logs for Auditing

Home Event Actions Dashboard Galaxies Input Filters Global Actions

Edit My Profile

Change Password

My Profile

My Settings

Periodic summary settings

Set Setting

List Organisations

Strong Access Controls

Data Encryption

Automated Monitoring

Kohezion

Immutable logs

Regular Backups

Periodic audits

User admin@admin.test

ID	1
Email	admin@admin.test
Organisation	Test MISP org
Role	admin
TOTP	Yes View paper tokens



Step-by-Step: Export MISP Logs to Syslog or ELK

1. Enable Syslog Forwarding of MISP Logs

- Edit MISP's configuration file:

```
/var/www/MISP/app/Config/config.php
```

```
'Syslog' => [
```

```
    'enabled' => true,
```

```
    'facility' => LOG_LOCAL0,
```

```
    'level' => LOG_INFO
```

```
],
```

This enables MISP to log actions to the system's syslog.

2. Configure Rsyslog to Forward Logs to a SIEM

- Edit `/etc/rsyslog.conf` or create a custom file in `/etc/rsyslog.d/`

```
# Forward MISP logs to a remote server
```

```
if $programname == 'misp' then @your.siem.ip.address:514
```

- Restart rsyslog:

```
sudo systemctl restart rsyslog
```

3. Set Up a Dashboard in ELK or Splunk

- Parse logs by action , User , and IP.
- Create alerts for:
 - ❖ High number of edits/deletes in a short time
 - ❖ API key misuse
 - ❖ Failed login spikes

The screenshot shows two pages of the MISP web interface. The top part is the 'Events' page, featuring a search bar, navigation buttons ('previous', 'next'), and a table with columns for 'Published', 'Org', 'Owner org', 'Id', and 'Clusters'. The bottom part is the 'User settings management' page, showing a table of user settings with columns for 'Id', 'User', 'Setting', and 'Value'. One setting is listed: 'publish_alert_filter' with a value of '["NOT": {"Tag.name": ["\$ip.white", "\$ip.green"]}]'. Navigation buttons ('previous', 'next') are at the bottom of both pages.



Practical Implementation Step: Create an Incident Report Template in MISP

The screenshot shows the 'Add Attribute' form in the MISP web UI. The 'Category' dropdown is set to 'Artifacts dropped' and the 'Type' dropdown is set to 'md5'. The 'Value' field contains the hash 'c974ffe23d57ec909ef26b55f202047e'. Under 'Contextual Comment', there is a note about a security researcher named slipstream/RoL discovering Karma Ransomware. The 'For Intrusion Detection System' checkbox is checked. The 'Submit' button is at the bottom.

Home Event Actions Dashboard Galaxies Input Filters Global Actions Administration Logs API

View Event View Correlation Graph View Event History

Edit Event Delete Event Add Attribute

Add Object Add Attachment Add Event Report Populate from... Enrich Event Merge attributes from...

Publish Event Publish (no email) Contact Reporter Download as...

List Events Add Event

Add Attribute

Category Type

External analysis text

Distribution

Inherit event

/value

A security researcher named slipstream/RoL has discovered the Karma Ransomware, which pretends to be a Windows optimization program called Windows-TuneUp. What is worse is that this sample was discovered as software that would potentially be distributed by a pay-per-install software monetization company when people install free software downloaded from the Internet.

Contextual Comment Source Report

for Intrusion Detection System Batch Import

Submit Cancel

Step-by-Step: Use MISP to Log and Track a Security Incident

1. Create a New Incident Event in MISP

Go to MISP Web UI → Event Actions → Add Event.

Fill in:

- ❖ **Info:** Security Incident – Unauthorized Access.
- ❖ **Threat Level:** High.
- ❖ **Analysis:** Initial.
- ❖ **Distribution:** Your Organization Only.

2. Add Attributes to the Incident

Examples:

- ❖ ip-src : 192.168.1.101.
- ❖ Comment : Suspicious login attempt from internal subnet.
- ❖ Timestamp : (use current time).
- ❖ user-agent : if known from logs.

No.	Time	Source	Destination	Protocol	Length	Info
954	1122.214623	192.168.154.131	192.168.154.132	ICMP	1028	Echo (ping) request id=0xffff, seq=0/0, ttl
955	1122.214781	192.168.154.131	192.168.154.131	ICMP	1028	Echo (ping) reply id=0xffff, seq=0/0, ttl
956	1122.314715	192.168.154.131	192.168.154.132	ICMP	1028	Echo (ping) request id=0xffff, seq=0/0, ttl
957	1122.314902	192.168.154.132	192.168.154.131	ICMP	1028	Echo (ping) reply id=0xffff, seq=0/0, ttl
958	1122.414619	192.168.154.131	192.168.154.132	ICMP	1028	Echo (ping) request id=0xffff, seq=0/0, ttl
959	1122.414767	192.168.154.132	192.168.154.131	ICMP	1028	Echo (ping) reply id=0xffff, seq=0/0, ttl
960	1122.514705	192.168.154.131	192.168.154.132	ICMP	1028	Echo (ping) request id=0xffff, seq=0/0, ttl
961	1122.514842	192.168.154.132	192.168.154.131	ICMP	1028	Echo (ping) reply id=0xffff, seq=0/0, ttl
242	431.691776	192.168.154.131	192.168.154.132	ICMP	1033	Echo (ping) request id=0xffff, seq=0/0, ttl

3. Tag and Correlate

Apply relevant tags:

```
tlp:red , incident-type:unauthorized-access , actor:unknown , response:under-investigation
```

4. Export or Report the Incident

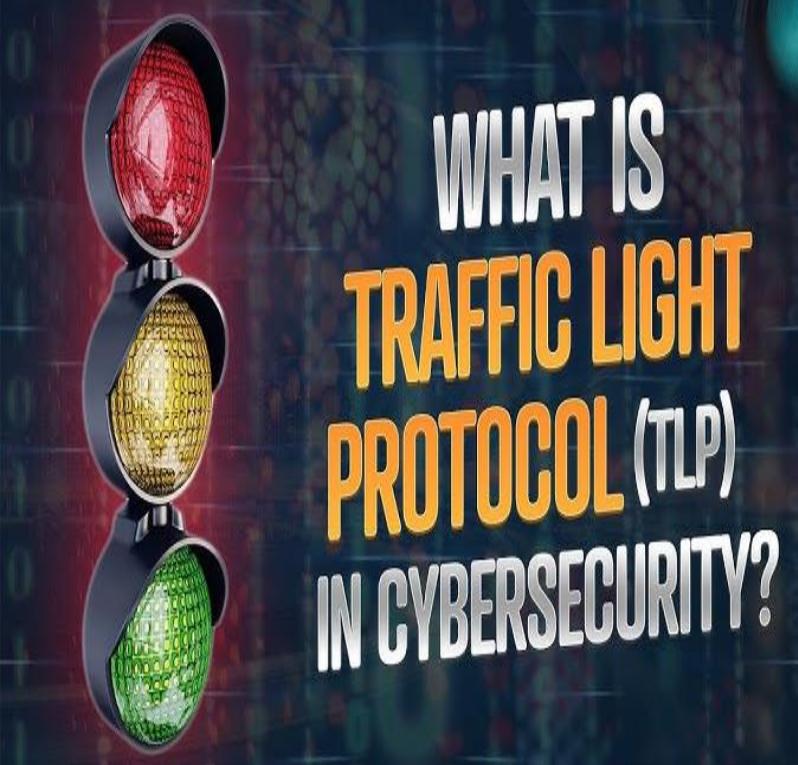
Use Export → PDF or STIX to generate a report for compliance or stakeholders.

- You can also automate this via MISP's API:

```
curl -H "Authorization: <API_KEY>" \
-H "Accept: application/json" \
-X GET https://<your_misp_url>/events/view/<event_id>
```

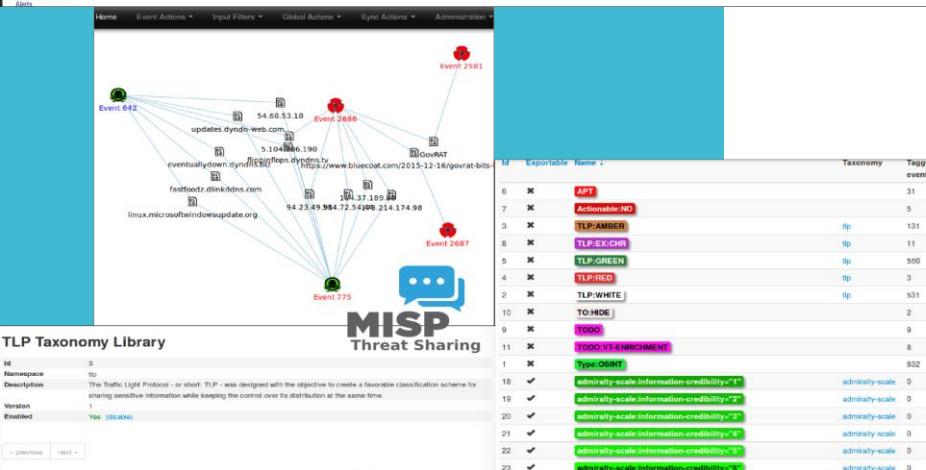
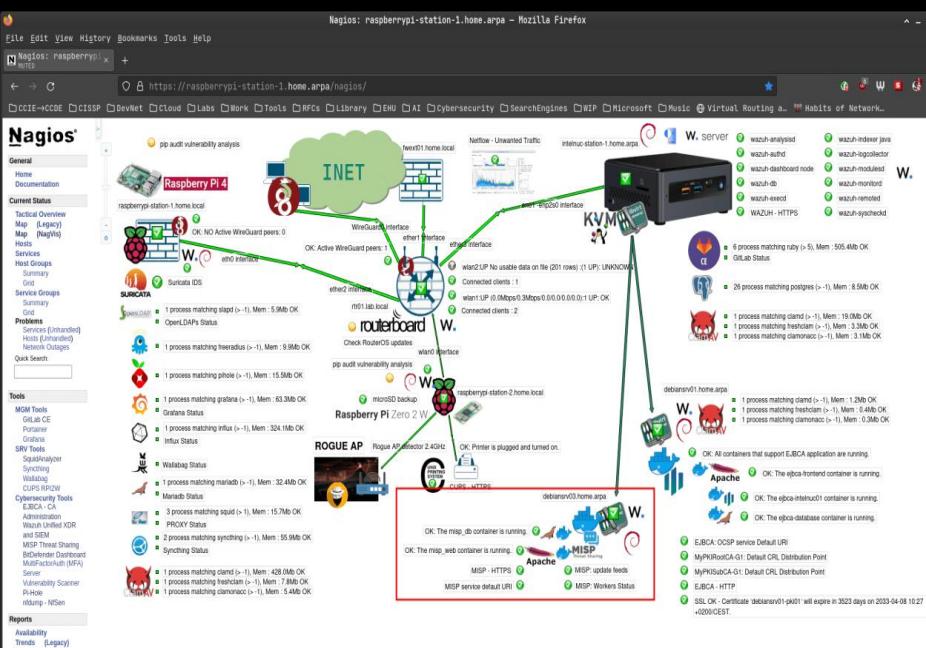
Result

- Log incidents.
- Classify and track response actions.
- Generate and export reports for audits or external communication.





Practical Implementation Step: Create a Security Configuration Guide for MISP Deployment



- Example: misp-security-configuration.md

MISP Security Configuration Guide

1. Web Server Configuration (Apache)

- Enforce HTTPS:

❖ SSLEngine on

❖ SSLCertificateFile /etc/ssl/certs/misp.crt

❖ SSLCertificateKeyFile /etc/ssl/private/misp.key.

- Add HTTP security headers:

- Header always set Strict-Transport-Security "max-age=63072000; includeSubDomains".
- Header always set Content-Security-Policy "default-src 'self'".



markdown

MISP Secure Deployment Guide

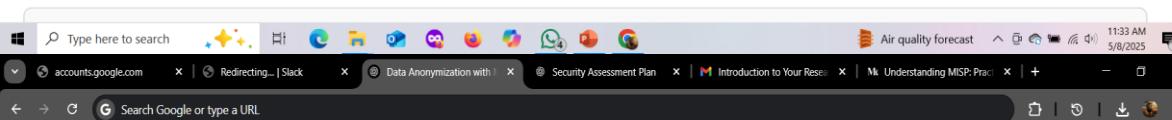
1. Authentication and Access Control

- Disable default admin credentials after setup.
- Enforce strong password policies (min. 12 chars).
- Integrate with LDAP or SAML for SSO and MFA.
- Use MISP role-based permissions to assign least privilege.

2. TLS/HTTPS

- Install a valid TLS certificate (e.g., Let's Encrypt).
- Enforce HTTPS in Apache configuration:

Redirect permanent / <https://your-misp-domain/>



pgsql

3. Database Security

- Use `mysql_secure_installation` to harden MySQL.
- Limit access to MySQL from localhost only.
- Enable InnoDB encryption if required (for data at rest).

4. Logging and Monitoring

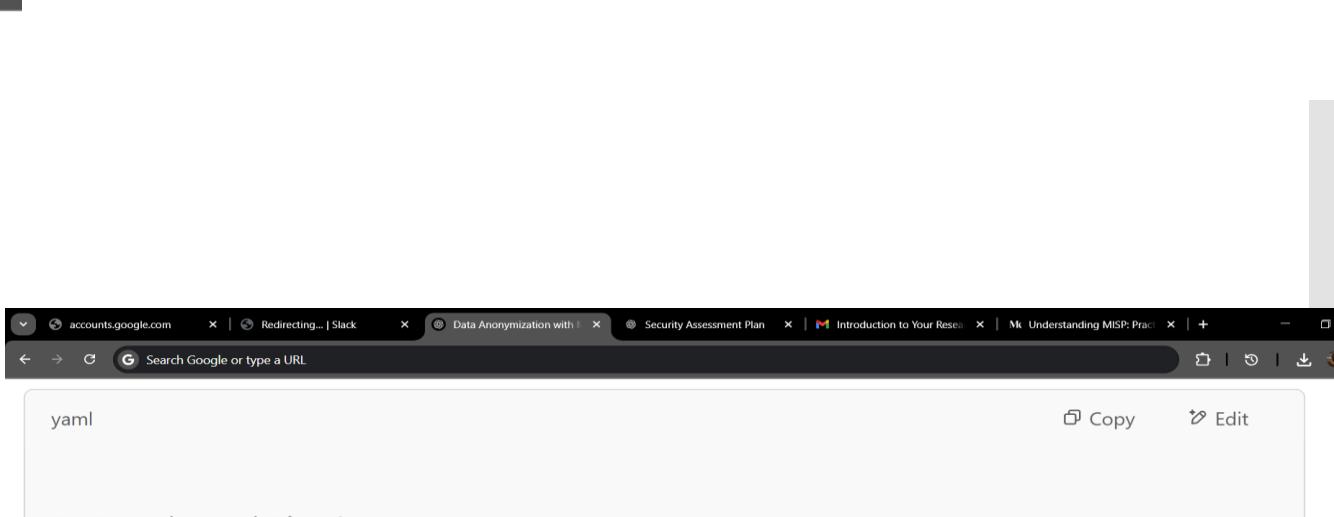
- Enable syslog integration to forward logs to a SIEM.
- Review logs weekly for unauthorized access attempts.

5. File and Permission Hardening

- Restrict file permissions on `/var/www/MISP`:

```
chown -R www-data:www-data /var/www/MISP
```

```
chmod -R 750 /var/www/MISP
```



yaml

6. Backup and Disaster Recovery

- Implement encrypted backups.
- Store backups securely off-site or in a secure cloud.

Appendix: Reference Checklists

- [x] GDPR Article 32 (Security of processing)
- [x] ISO 27001: A.9, A.12, A.13 compliance areas

