CIRCL
Computer Incident
Response Center
Luxembourg

Beyond tools: building an open security ecosystem for autonomy and intelligence sharing

From MISP to Collective Defence

https://www.misp-project.org

Alexandre Dulaunoy - alexandre.dulaunoy@circl.lu

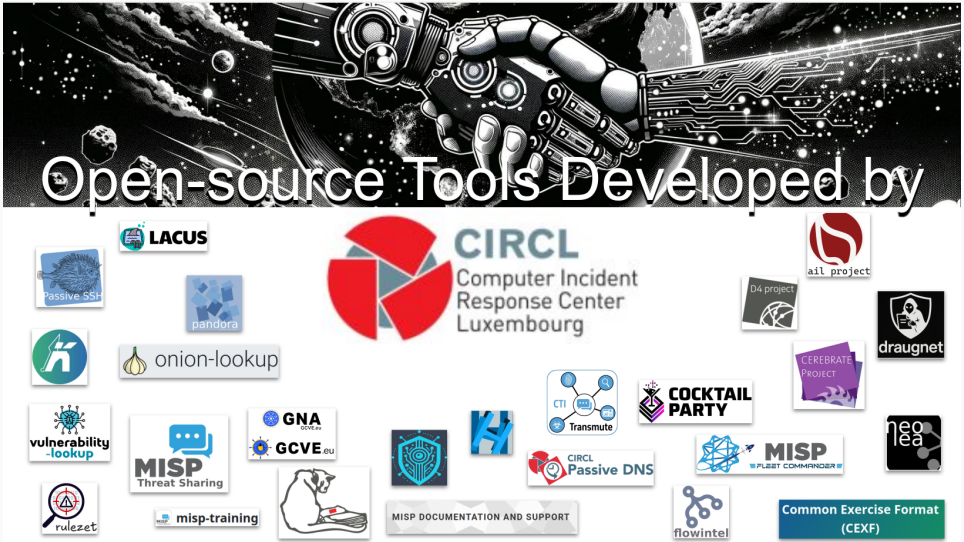
9th April 2026

CIRCL https://www.circl.lu

TLP:CLEAR

CIRCL as a NIS2 CSIRT and Open Source Security Leader

- **CIRCL is Luxembourg's national CSIRT for the economy**, designated under the NIS2 Directive.
- Core missions: incident handling, early warning, threat analysis, coordinated vulnerability disclosure, coordinated response, and support to essential/important entities.
- **Open source is central to our mandate:** CIRCL develops and maintains **17+ security projects** used worldwide by CSIRTs, ISACs, industries, intelligence community and defenders.
- Key platforms: **MISP**, **AIL**, **Vulnerability-Lookup**, **FlowIntel**, **LookyLoo**, **Cerebrate**, **GCVE.eu**.
- Open source ensures **transparency, interoperability, sovereignty** and long-term sustainability for operational security.


Open-source Tools Developed by
LACUS
Passive SSH
pandora
CIRCL
Computer Incident
Response Center
Luxembourg
ail project
D4 project
draugnet
onion-lookup
CEREBRATE
Project
vulnerability
-lookup
MISP
Threat Sharing
GNA
GCVE.eu
CTI
Transmute
COCKTAIL
PARTY
neo
lea
CIRCL
Passive DNS
MISP
FLEET COMMANDER
rulezet
misp-training
MISP DOCUMENTATION AND SUPPORT
flowintel
Common Exercise Format
(CEXF)

TLP:CLEAR

CIRCL and Open Source Tooling Strategy

- Since its creation, CIRCL has followed a simple principle: software developed for internal operational needs should also benefit the broader community.[1]
- This approach is not limited to publishing source code. Whenever possible, CIRCL also operates associated **services**[2] to make these tools directly usable and to support real-world adoption.
- The objective is not only to build tools, but to **produce, enrich, and share threat intelligence** through the platforms and services we maintain.

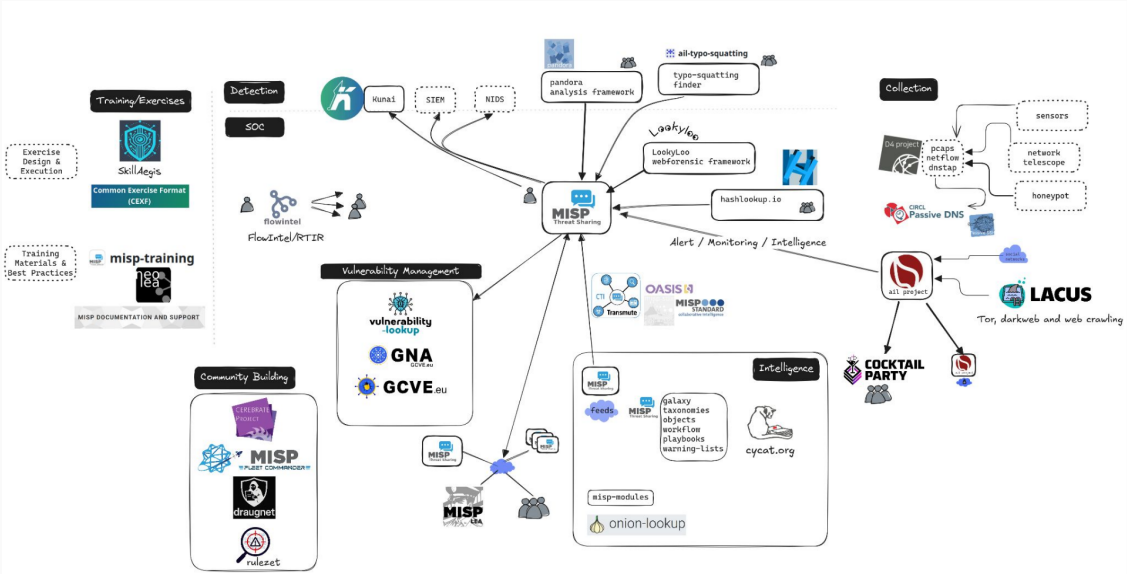
[1] *Public Money, Public Code*

[2] Publicly accessible or restricted-access services depending on the use case and constituency.

From Tools to an Operational Ecosystem

- Open development enables external review, reuse, contribution, and interoperability, while reducing dependency on closed and opaque security platforms.
- Over time, this strategy has led to an ecosystem of complementary projects supporting threat intelligence sharing, incident response, collaborative analysis, automation, and vulnerability intelligence.
- As of 2026, CIRCL maintains more than 17 open-source projects[3] and more than 250 official Git repositories, reflecting a long-term investment in open and operational cybersecurity capabilities.

[3] https://opensource-metrics.circl.lu/


Training/Exercises
Exercise Design & Execution
SkillAegis
Common Exercise Format (CEF)
Training Materials & Best Practices
misp-training
MISP DOCUMENTATION AND SUPPORT
Detection
SOC
Kunai
SIEM
NIDS
flowintel
FlowIntel/RTIR
pandora analysis framework
ail-typo-squatting
typo-squatting finder
LookyLoo webforensic framework
hashlookup.io
MISP Threat Sharing
Alert / Monitoring / Intelligence
Collection
sensors
pcaps netflow dnstap
network telescope
honeypot
Passive DNS
ail project
LACUS
Tor, darkweb and web crawling
Vulnerability Management
vulnerability-lookup
GNA
GCVE.eu
Transmute
OASIS
MISP STANDARD
COCKTAIL PARTY
Community Building
CERLEBRATE PROJECT
MISP FLEET COMMANDER
draugnet
rulezet
MISP KBA
Intelligence
feeds
galaxy
taxonomies
objects
workflow
playbooks
warning-lists
cycat.org
misp-modules
onion-lookup

TLP:CLEAR

Our Rewarding Journey

- Sixteen years ago, we began with open-source tools for CSIRTs and the BGP Ranking[4] project, which assesses the reputation of Internet Service Providers.
- The journey has been challenging, yet immensely rewarding, enhancing our **capabilities**, **expanding partnerships**, driving daily improvements, fostering continuous learning, and achievi ng **greater autonomy**.

[4] https://bgpranking.circl.lu/

First Lesson: Publish and Embrace Criticism

- Many large organizations hesitate to release open-source software due to fear of criticism.
- **Don't be afraid—release early**, release often[5].
- Early in a project, you're more likely to attract early adopters than critics, so the fear of criticism is often misplaced.
- If you follow a specific programming methodology for your open-source projects, make sure to document and share it[6].

[5] Or perhaps, release regularly.
[6] `https://datatracker.ietf.org/doc/draft-dulaunoy-programming-methodology-framework/02/`

MISP Project and CIRCL

- CIRCL **leads the development** of the Open Source MISP threat intelligence platform[7] which is used by many military or intelligence communities, private companies, financial sector, National CERTs and LEAs globally.
- **CIRCL runs multiple large MISP communities performing active daily threat-intelligence sharing**.
- Private sector such as the financial sector can request access to one or more information sharing communities operated by CIRCL.

[7] https://www.misp-project.org/

Second Lesson: Practice Vulnerability Handling

- Leading the development of open-source tools means taking responsibility for managing vulnerabilities in your projects.
- For a CSIRT, this experience is invaluable, as it **puts you in the position of a software vendor**.
- With the MISP project being widely used in cybersecurity, effective vulnerability handling is critical.
- We learned many challenges, from release notifications and vulnerability ID assignments[8] to proper disclosure[9].

[8] https://gcve.eu/
[9] https://www.misp-project.org/security/

Third Lesson: Open Source as a Facilitator for Partnerships

- **Nothing fosters collaboration better than a successful open-source project**.
- Often, there is no need for NDAs, confidentiality agreements[10], or IPR agreements, as the open-source contribution model inherently addresses these aspects.
- In EU research projects, contributing to existing open-source projects ensures clear outcomes and deliverables, including measurable Technology Readiness Levels (TRL).

[10] Typically required for sharing private projects or defining outcomes

"Some cities have fallen into ruin and some are built upon ruins but others contain their own ruins while still growing." *Jeffrey Eugenides*

Fourth Lesson: Managing the Ruins of Software Dependencies

- Building and maintaining long-term open-source projects helps grow a community of users and contributors.
- Over time, the accumulation of outdated or obsolete software dependencies becomes an inevitable part of the development process.
- This places you in the unique position of being an active "archeologist,"[11] **navigating the complexities of aging software stacks**.
- You must address the inherent security risks of these dependencies, **just as any software vendor would**.

[11] https://hashlookup.io/

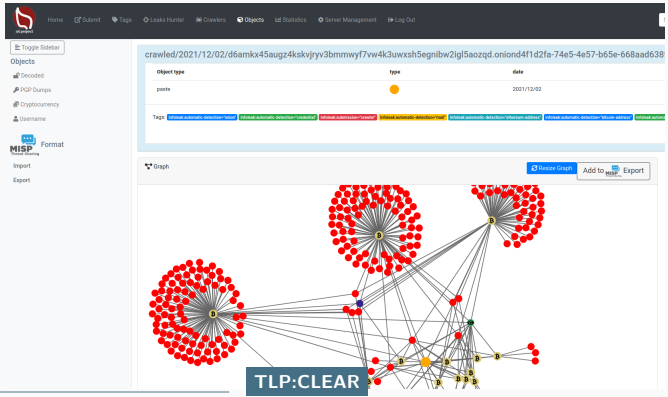
Fifth Lesson: Open Source as a Standard Generator

- Developing information security standards without open-source implementations often leads to more discussion than the creation of interoperable tools.
- **Tools and formats created by these open source implementations can serve as a strong foundation for standard definitions**. The MISP standard[12] was written based on real-world implementation, rather than by a committee disconnected from practical realities.
- Don't hesitate to draft IETF Internet-Drafts based on your implementations and open-source tools. This can be the beginning of a truly practical open standard.

[12] https://misp-standard.org/

AIL Project

- AIL Project[13] is an open source framework to collect, crawl, dig and analyse unstructured data. The framework can be used to find **information leaks**, int elligence, insights and much more. The open source framework includes crawling services (for Tor, I2P) or feeders for specific sources (Telegram, fediverse).



[13] https://ail-project.org/

Sixth Lesson: Learning from Threat Intelligence Collection

- Developing an open-source tool for intelligence collection provides valuable insights into the challenges faced by threat intelligence vendors.
- **Controlling and managing collection mechanisms** allows your CSIRT to assess the quality of intelligence and understand fluctuations over time.
- Building autonomy and enhancing your team's capabilities in intelligence collection often brings more value than relying solely on vendor-provided feeds.

Vulnerability-Lookup

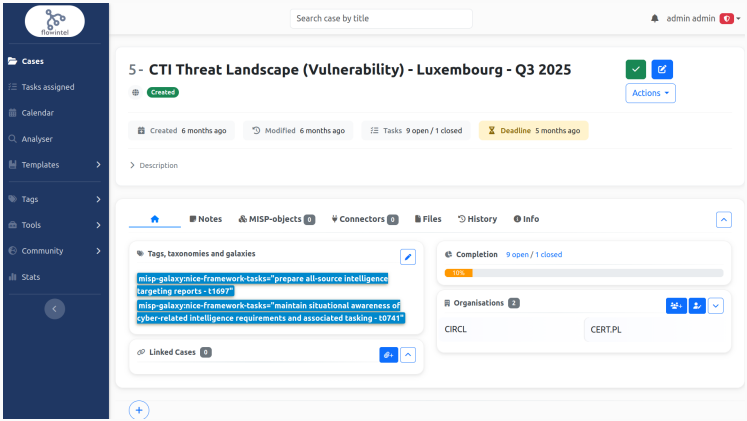
- A fast lookup API to search for vulnerabilities[14] and find correlations per vulnerability identifier.
- Modular system[15] to import **different vulnerability sources**[16].
- An API for adding new vulnerabilities, including ID assignment[17], state, and disclosure.
- Vulnerabilities can be bundled and commented allowing your CSIRT to extend vulnerability information.
- 65+ sources are already available in the default configuration.

[14] https://vulnerability.circl.lu
[15] https://www.vulnerability-lookup.org/
[16] The system is independent of the different source formats
[17] https://gcve.eu/

Seventh Lesson: Embrace Failure

- **Failure is an essential part of designing new open-source tools for CSIRTs**.
- In vulnerability management, the landscape evolves constantly, requiring tools to adapt to each new step or change in the ecosystem.
- We initially designed and maintained cve-search, but we faced setbacks with cve-portal due to the lack of proper software identifiers. This led us to develop vulnerability-lookup to address those issues.
- **Each failure is necessary to create innovative open-source software** that solves real-world problems faced by CSIRTs and their users.

flowintel - open source case management



- flowintel[18] is an open-source platform designed to assist analysts in **organizing their cases and tasks**. It features a range of tools and functionalities to enhance workflow efficiency.

[18] https://github.com/flowintel/flowintel

Eighth Lesson: Licenses and Copyright are Critical

- Don't underestimate **the importance of licensing models in open-source software**.
- Pay special attention to Contributor License Agreements (CLA) and who ultimately owns the code, regardless of the applied license.
- After experiencing open-source dependencies becoming proprietary, we've taken a strong stance against CLAs[19]. Ensuring multiple copyright ownerships is a safer approach for open-source projects.

[19] https://ossbase.org/initiatives/cla-free/

Ninth Lesson: Open Source as a Tool for Staff Retention and Skill Development

- The cybersecurity field often faces a skills shortage, with high turnover due to intellectual fatigue in some organizations.
- We found that enhancing the capabilities of our CSIRT team helped us retain skills and expand into new areas of expertise.
- **Open-source projects serve as a powerful incentive for staff autonomy**, ultimately benefiting both individual team members and the CSIRT as a whole.

Conclusion

- The open-source journey for a CSIRT team is challenging but **ultimately rewarding**. It often requires pushing back against the status quo.
- It enhances the team both economically and personally, allowing them—and other organizations—to focus on developing their staff rather than acquiring new products.
- Continuous contribution and maintenance of open-source projects can significantly boost capabilities.
- For more information, feel free to contact us: `info@circl.lu`.

References

- https://www.misp-project.org
- https://www.misp-galaxy.org
- https://github.com/misp and many more
- https://tinyurl.com/FICMISP - Overview of the open source tools and open services

Books

- *Turn the Ship Around! A True Story of Turning Followers into Leaders*, L. David Marquet, ISBN 978-1591846406
- *Tools for Conviviality*, Ivan Illich, 1973.
- *Social Architecture - Building "On-line" Communities*, Pieter Hintjens, 2016.
- *Working in Public: The Making and Maintenance of Open Source Software*, Nadia Eghbal, 2020.

Books available on `https://annas-archive.<ADDYOURTLD>/`