



# Artemis: How CERT PL Improves the Security of the Polish Internet

Krzysztof Zajac

cert.pl



# #whoami

- Senior Threat Analysis Specialist, CERT PL
- Started as a software engineer
- Teaches offensive security at the University of Warsaw

# The purpose of this talk

# 1. Show the approach

I'll describe the non-technical environment as well

---

## 2. Show the tool

---

### 3. Encourage you to start similar projects

(if you are in the position to do so)

---

All of these are equally important!














# Purpose

After an incident, let's make sure it won't occur in other entities.

Example:

- exposed code repository on an university website caused API key leak and unauthorized data access
- let's check whether other entities have exposed code repositories!

## Index of /.git

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 <a href="#">Parent Directory</a>		-	
 <a href="#">FETCH_HEAD</a>	2019-04-03 12:19	3.8K	
 <a href="#">HEAD</a>	2019-04-03 15:04	23	
 <a href="#">ORIG_HEAD</a>	2019-04-03 12:18	41	
 <a href="#">config</a>	2019-04-03 12:18	312	
 <a href="#">description</a>	2019-04-03 12:19	73	
 <a href="#">hooks/</a>	2019-04-03 12:19	-	
 <a href="#">index</a>	2019-04-03 15:04	226K	
 <a href="#">info/</a>	2019-04-03 12:19	-	
 <a href="#">logs/</a>	2019-04-03 12:19	-	
 <a href="#">objects/</a>	2019-04-03 12:19	-	
 <a href="#">packed-refs</a>	2019-04-03 12:19	22K	
 <a href="#">refs/</a>	2019-04-03 12:19	-	



A domain →  **artemis** →

1. The following addresses contain version control system data:

- https://[REDACTED]:443/.git/

(...)



1. The following addresses contain version control system data:

- [https://\[REDACTED\]:443/.git/](https://[REDACTED]:443/.git/)

(...)

2. The following addresses contain old Joomla versions:

- [https://\[REDACTED\]:443](https://[REDACTED]:443) - Joomla 2.5.4

(...)

.gov.pl,  
schools,  
hospitals,  
universities,  
banks,  
...



# artemis



1. The following addresses contain version control system data:

- https://[REDACTED]:443/.git/

(...)

2. The following addresses contain old Joomla versions:

- https://[REDACTED]:443 - Joomla 2.5.4

(...)

# What do we check?

---

# A couple dozen modules

- Finding subdomains (e.g. cert.pl → test.cert.pl)
- Findings sites hosted on a given IP if scanning an IP range
- Domain expiration check
- Bad DNS configuration check:
  - Zone transfer
  - Subdomain takeover
- E-mail spoof protection mechanisms: SPF/DMARC
- Bad/expired TLS certificates, https:// redirect

# A couple dozen modules

- Port scanning, identifying services on a given server
- WordPress, WordPress plugin, Drupal, Joomla, and Joomla extension version check
- Closed WordPress plugins
- **Nuclei support:** thousands of vulnerabilities and misconfigurations (from Open Redirects to RCEs)
- SQLi, XSS, Local File Inclusion check
- Scripts loaded from nonexistent domains

# A couple dozen modules

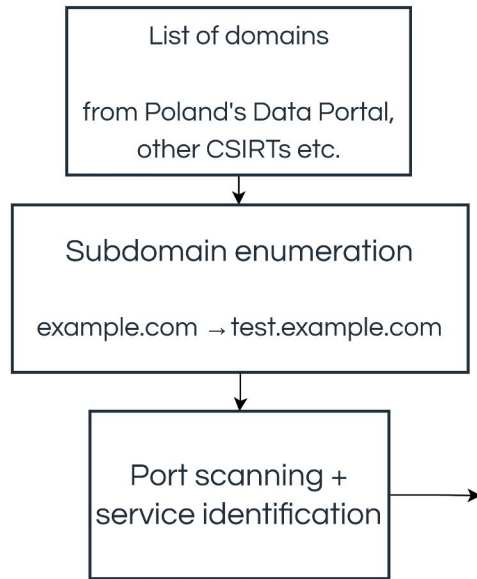
- Directory index
- Weak passwords
- Exposed code repositories
- Exposed login panels, RDP ports, databases, ...
- Accidentally published files (eg. /db.sql, /backup.zip or /wp-config.php.bak)
- **Possibility to integrate any other tool (commercial or open-source) + an example how to do that**

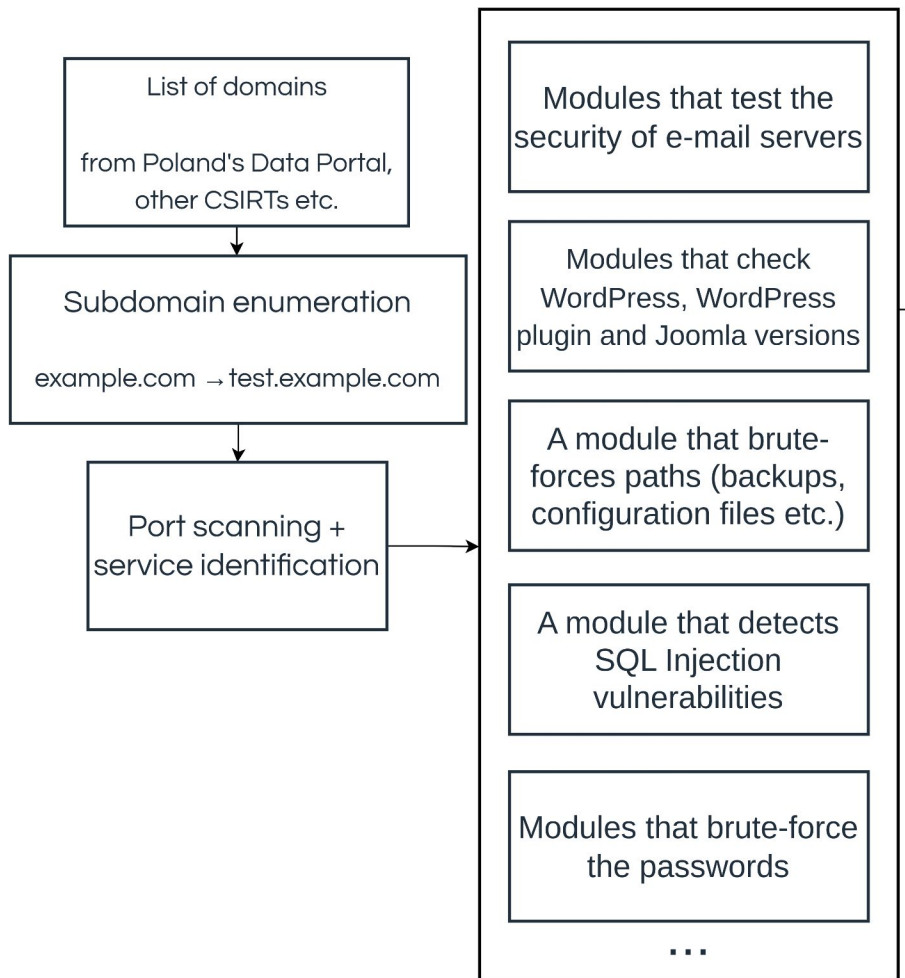
# Where the list comes from?

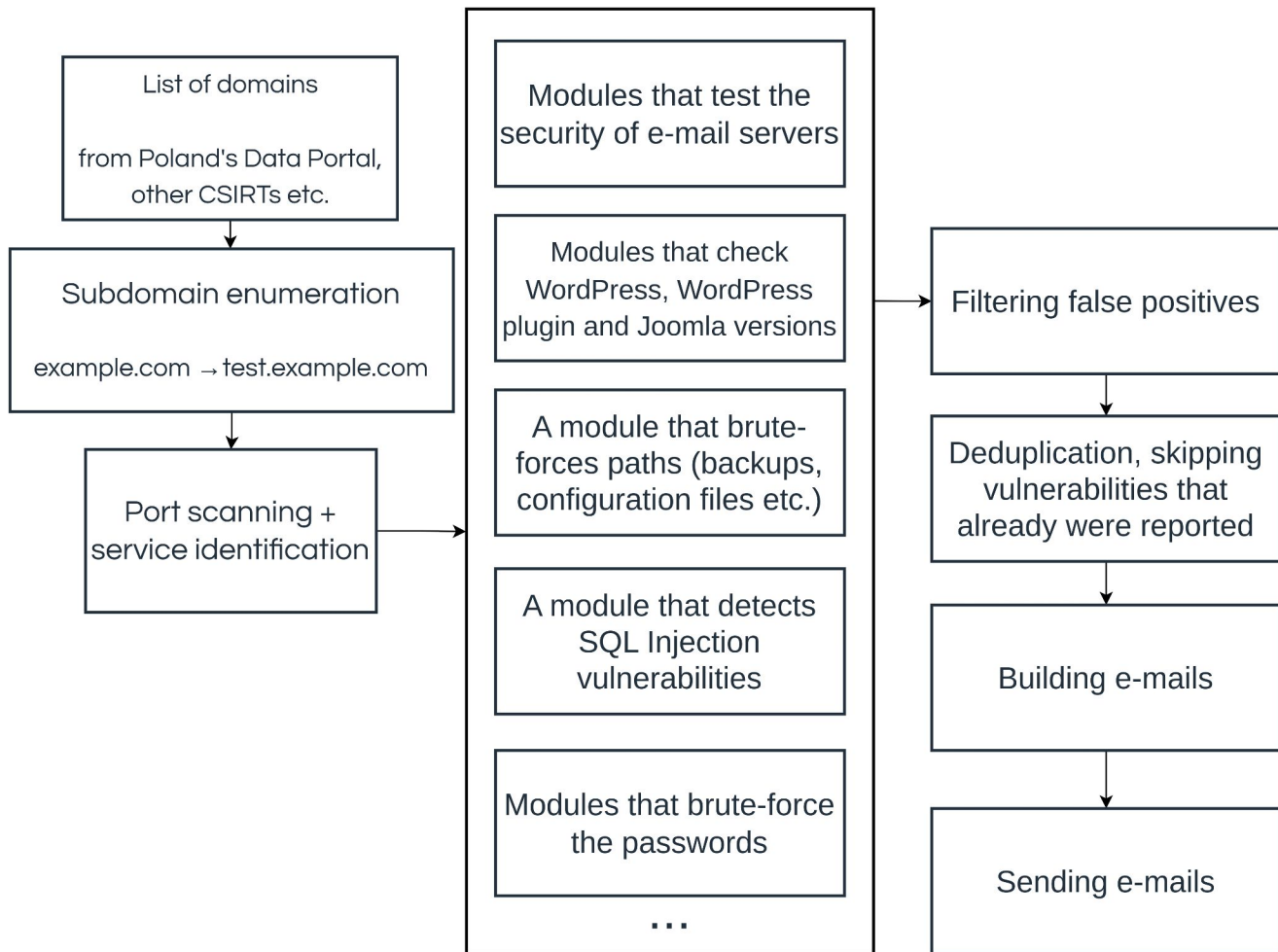
Our experiences in handling incidents.

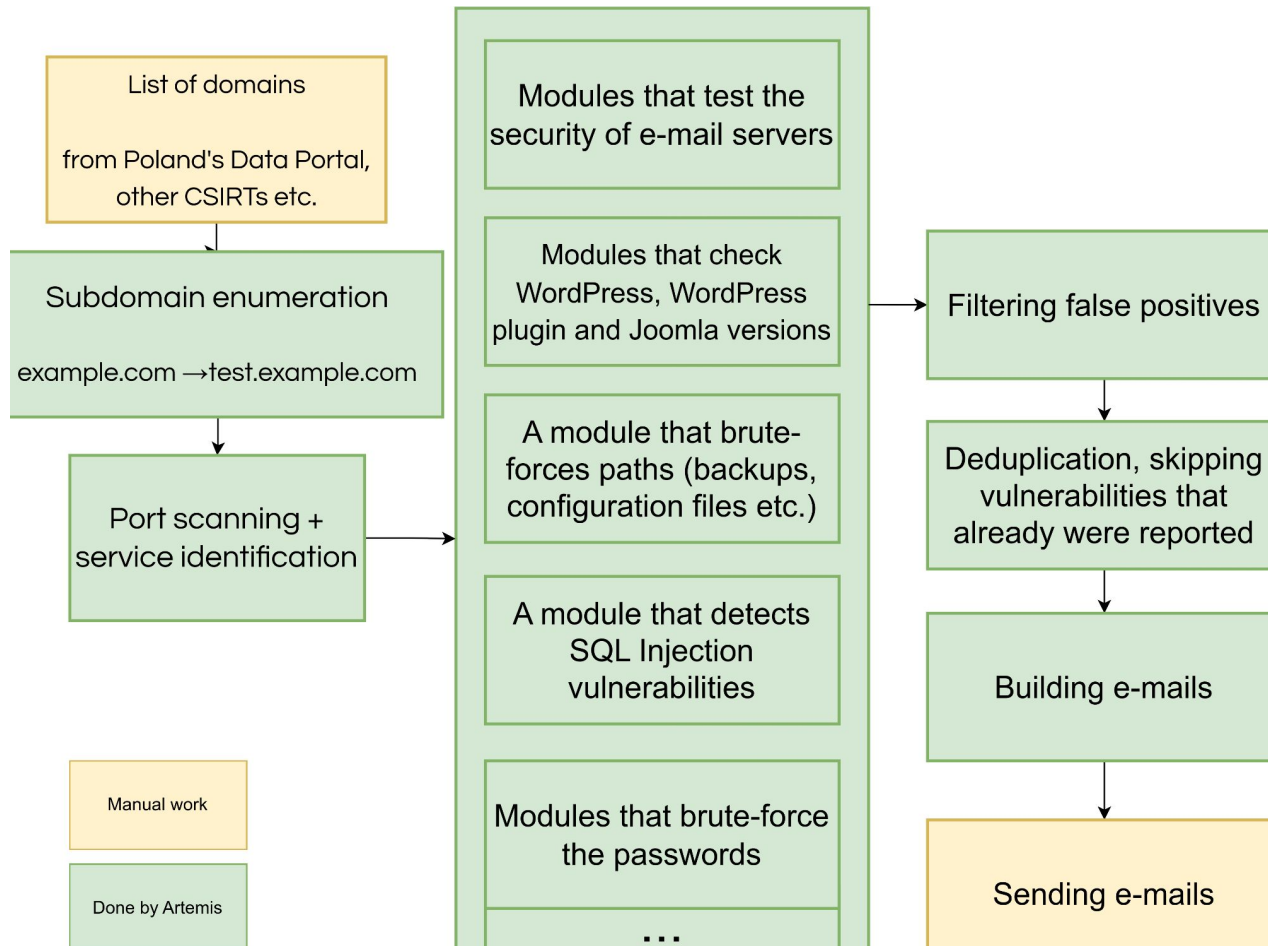
E.g.: someone got hacked because of a SQL Injection vulnerability? Let's improve SQL Injection detection capabilities!











# Example e-mail

The following addresses contain version control system data:

- `https://[REDACTED]:443/.git/`

Making a code repository public may allow an attacker to learn the inner workings of a system, and if it contains passwords or API keys - also gain unauthorized access. Such data shouldn't be publicly available.

# Example e-mail

**Such reports are sent by CERT PL to scanned entities  
(but in Polish).**

# List of domains

- Customer database (if you're e.g. a hosting provider)
- Data portals: <https://dane.gov.pl/en>
- Tools such as crt.sh: <https://crt.sh/?q=%25.gov.de>,
- Custom databases (example: [rspo.gov.pl](https://rspo.gov.pl) for schools),
- Be creative (example: [mamprawowiedziec.pl](https://mamprawowiedziec.pl)),
- Contacting the entities (slow)
- ...

# List of domains

- ~~— Customer database (if you're e.g. a hosting provider)~~
- Data portals: <https://dane.gov.pl/en>
- Tools such as crt.sh: <https://crt.sh/?q=%25.gov.de>,
- Custom databases (example: [rspo.gov.pl](https://rspo.gov.pl) for schools),
- Be creative (example: [mamprawowiedziec.pl](https://mamprawowiedziec.pl)),
- Contacting the entities (slow)
- ...



# Who do we scan

- All gov.pl domains
- Local government entities
- Municipal corporations: water management, waste collection, ...
- Key Service Operators
- Banks
- Universities, schools, preschools and other educational entities
- Hospitals

# Who do we scan

- Local and country-level newspapers, TVs, information portals, etc.
- Websites of politicians, political parties, candidates, etc.
- Professional self-governments (e.g. medical chambers)
- Lists of domains provided e.g. by other CSIRTs or ministries.
- **Domains provided voluntarily by companies/users.**

# moje.cert.pl

EN: My CERT PL

---

A frontend for Artemis and other tools

# You are on moje.cert.pl

Log in to use CERT PL services that will increase the cybersecurity of your network and domains.

Log in

Create an account



# Add domains

Domain names

Domain names

Each subsequent domain in a new line

☒ **Scan the security of systems in these domains**

If you request a domain scan, its subdomains will also be scanned.

Security scanning will be performed using the Artemis system - [see more](#) .

## Unblocking the firewall

If you are using a firewall or similar solution, we recommend whitelisting the following IP addresses to ensure complete scanning results (some devices may automatically block IP addresses that scan): 195.164.49.68 , 195.164.49.69 , 195.164.49.70 , 195.164.49.71 , and 195.164.49.72 .

[Show advanced options](#)

Add

# Scan results

## example.com

Scanning from

March 3, 2025 7:32 PM ▾

The next scan will begin around **May 29, 2025**.

Low 2

Medium 1

High 0

Show 10 ▾ position

Search:

Severity



Resource

Vulnerability/Misconfiguration

Medium

example.com

The following domains do not have email sender verification mechanisms configured correctly:

- example.com: No valid DMARC record found. We recommend using all three mechanisms: SPF, DKIM, and DMARC to reduce the chance that a spoofed message will be accepted by the recipient's server.

Implementing these mechanisms will significantly increase the chance that the recipient's server will reject a forged email from the above domains. At <https://bezpiecznapoczt.cert.pl> you can verify the correct implementation of the sender verification mechanisms in your domain.

# Legal basis

Don't design law that:

- Allows scanning of a small subset of entities (e.g. *important* ones)
- Requires actions that are **not viable** in case of broad scans, such as signed agreements with entities.

Possibility to perform **broad scans** was **crucial** for the success of Artemis!

(keep an eye on the above when implementing EU NIS2)

# Statistics

**1M** domains/IPs and **1.7M** subdomains scanned **periodically**.



# Statistics

Since January 2023 we reported ~**876k** vulnerabilities and misconfigurations, including:

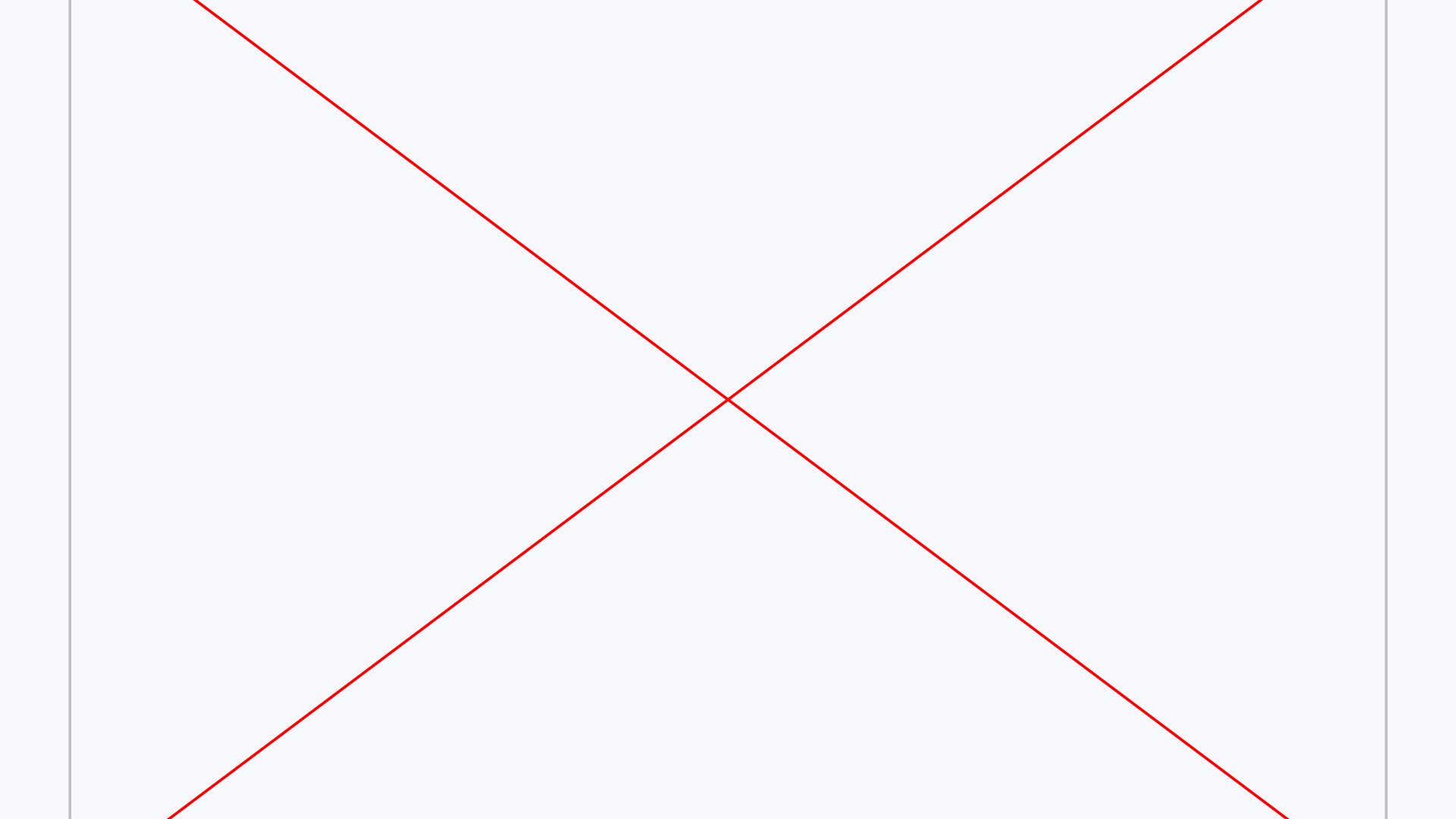
- ~**46.1k** high-severity,
- ~**531.2k** medium-severity,
- ~**299.1k** low-severity.
- For example we have almost **1000** confirmed SQL Injections (where we managed to **dump data from the**

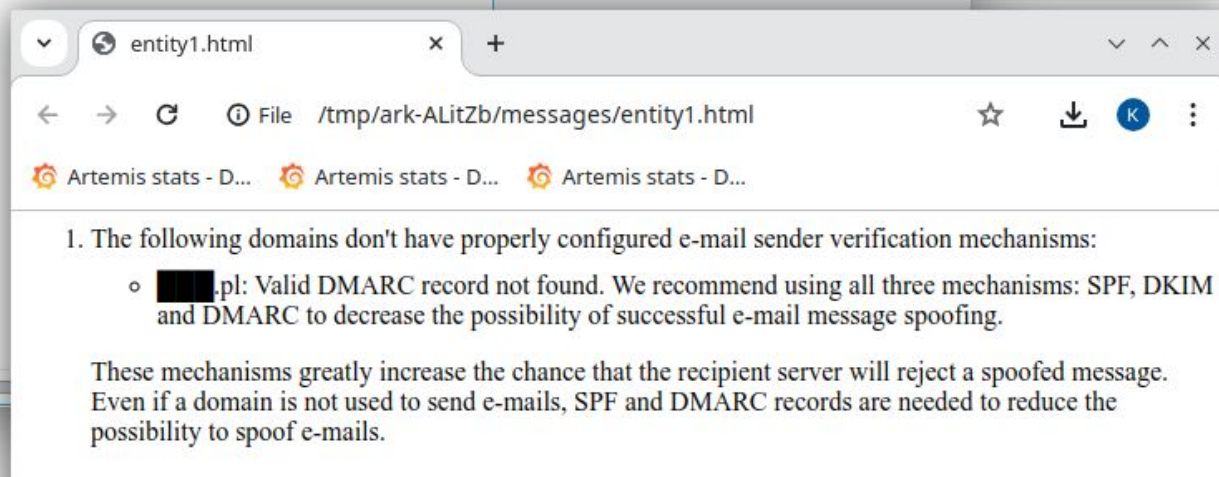
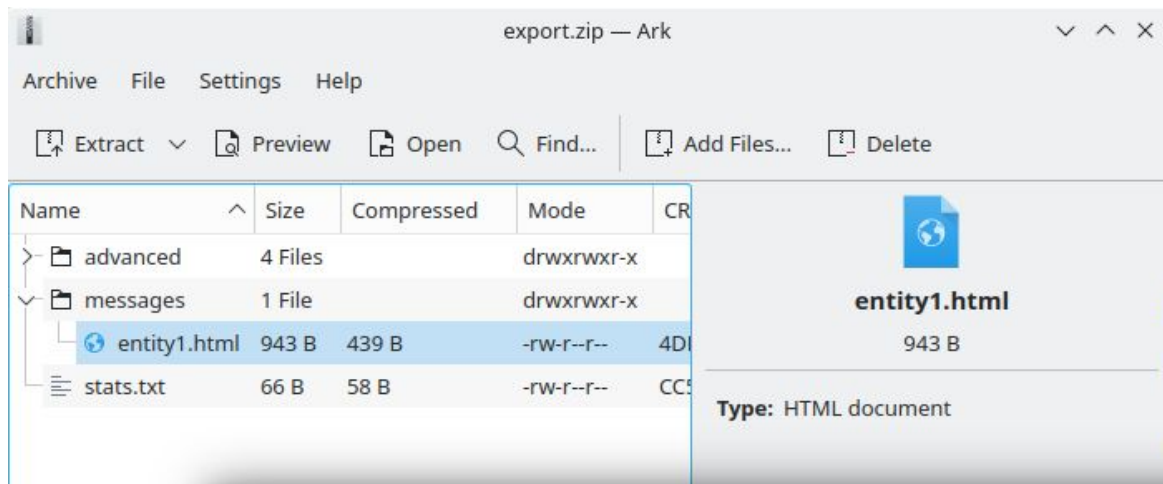
# Communication

- We already sent over **100k e-mails**.
- If an entity doesn't fix a serious issue, **we call them**. We already made **several thousands** such calls.
- Reactions are mostly positive (but we sometimes receive bug reports).
- Important: sometimes our e-mail **gives “political” support to the admins** even if they know about a problem.

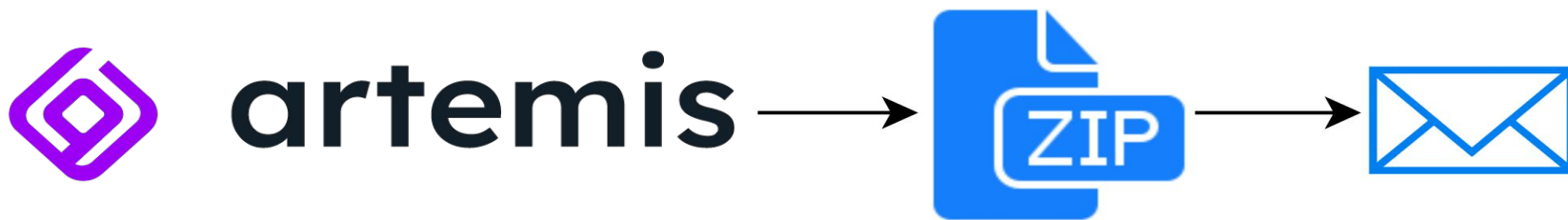
# Demo

---





# Exporting the reports



# Exporting the reports

Export: **the most important feature of Artemis** as it allows us to **scale** the scans.

And what with the problems that didn't make it to the reports?

Artemis checks whether a problem is interesting enough to be included in the reports.

# API





# Report translations

---

# Translations

Lazy way:

*translate the following file to French, leaving stuff in  
quotes after "msgid" in English but putting the  
translation after "msgstr":*



## ChatGPT

Here's the translation of the file into French:

```
#: artemis/reporting/modules/wp_scanner/template_insecure_wordpress.jinja2:3
```

```
msgid ""
```

```
"The following addresses contain WordPress versions that are no longer "
```

```
"supported and are marked as insecure in the version list from"
```

```
msgstr ""
```

```
"Les adresses suivantes contiennent des versions de WordPress qui ne sont plus "
```

```
"prises en charge et sont marquées comme non sécurisées dans la liste des "
```

```
"versions de"
```

```
#: artemis/reporting/modules/wp_scanner/template_insecure_wordpress.jinja2:19
```

```
#: artemis/reporting/modules/wp_scanner/template_old_wordpress.jinja2:14
```

```
msgid ""
```

```
"If a site is no longer used, we recommend shutting it down to eliminate "
```

```
"the risk of exploitation of known vulnerabilities in older WordPress "
```

```
"versions. Otherwise, we recommend enabling WordPress core and plugin "
```

```
"automatic updates."
```

```
msgstr ""
```

# Our approach

- **Being open:** <https://cert.pl/skanowanie/> (translation [here](#))
- Not overloading the servers (our configuration: one request per second per IP)
- **We are OK with the scan being slow**
- Making sure the vulnerabilities get fixed - e.g. **calling the scanned entities**
- Allow submitting domains voluntarily (we started in a **low-tech way** - e-mails)

# Artemis in production

Most important: **do things well enough.**

Examples:

- A module is broken? Scan using the rest until it gets fixed.
- Don't yet have a green light to scan all entities? Scan the ones you are allowed to.

# Lessons learned

- Lots of low-hanging vulnerabilities

There was a great need for such a project.

- Many good offensive tools are available

Even plain Nuclei scan or WordPress/Joomla version check would find many vulnerabilities. **Build upon other tools.**

- Iterative development contributed to the project success

Instead of building the best scanner possible, we built a MVP with a subset of modules and ran initial scans. During scans, we observed bugs, fixed them, but also added new modules.

# How to start

- Start small!
- Download Artemis (and <https://github.com/CERT-Polska/Artemis-modules-extra>)
- Set up Artemis using the [quick-start documentation](#)
- Take one list of domains (e.g. one you can get easy approval to scan), e.g. from a [data portal](#)

# How to start

- translate Artemis to your language - we have docs on how to do that.
- Scan, send the results.
- Show to the stakeholders that the scanning makes sense.
- Iterate: increase scanning coverage.
- Contact [artemis@cert.pl](mailto:artemis@cert.pl) in case of any problems.



# How to start

If you are a national CSIRT, CERT PL will be glad to help with setting up your scanning pipeline.

Good luck!

---

Questions?

# Links

<https://github.com/CERT-Polska/Artemis>

<https://github.com/CERT-Polska/Artemis-modules-extra>

[artemis@cert.pl](mailto:artemis@cert.pl)

<https://discord.com/invite/GfUW4mZmy9>