



Artemis: how CERT PL improves the security of the Polish internet

Krzysztof Zajac
69th TF-CSIRT Meeting
2023-05-24, Bucharest

Purpose

- Checking the security of websites and systems used by institutions in our constituency:
 - schools
 - universities
 - government institutions
 - local government and public utility companies
 - hospitals
- Improving the security of these systems by reporting the vulnerabilities

Legal basis

Act of National Cybersecurity System Legislation (NIS1 implementation) article 26 - tasks of CSIRTs:

1. monitoring threats and incidents on a country level
2. providing information to entities in the national cybersecurity system
3. in justified cases: conducting vulnerability research of a device or software
4. developing tools to (...) detect and combat cybersecurity threats (...).

Article 32 allows us to do “any necessary technical actions” to analyze cybersecurity threats

Legal basis

- We are the registrar for the .pl domain - we can put a clause that allows us to scan in gov.pl rules.
- In some cases - agreements with other CSIRTs or institutions directly responsible for a system.
- Backup: penal code article 269c.
The Polish Criminal Code penalises breaking into someone else's IT systems, but has an explicit exception for when it's done for security purposes, without breaking anything and if the issue was immediately reported.
- NIS2 implementation - upcoming.

Design goals

- Low amount of manual vulnerability analysis: heuristics to filter true from false positives
- Low load on scanned systems: per-host rate limiting
- Reusing existing tools
- Scalability
- Easy integration of a new tool
- Flexible scanning pipeline
 - work with domains, HTTP services, WordPress instances, ...

What Artemis does?

- Finds subdomains using open-source sources (crt.sh, Common Crawl, Wayback Machine, ...):

example.com → mail.example.com, old.example.com

- Detects DNS misconfigurations:
 - Zone transfer,
 - Subdomain takeover.

What Artemis does?

- Performs port scanning and service identification (is this a website? a database?).
- Finds backups and other interesting files (e.g. /wp-config.php.bak) using brute-force.
- Brute-forces weak passwords (FTP, PostgreSQL, MySQL i WordPress).

What Artemis does?

Detects directory index:

Index of /

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
backup_20190215.zip	2019-02-15 22:21	66M	

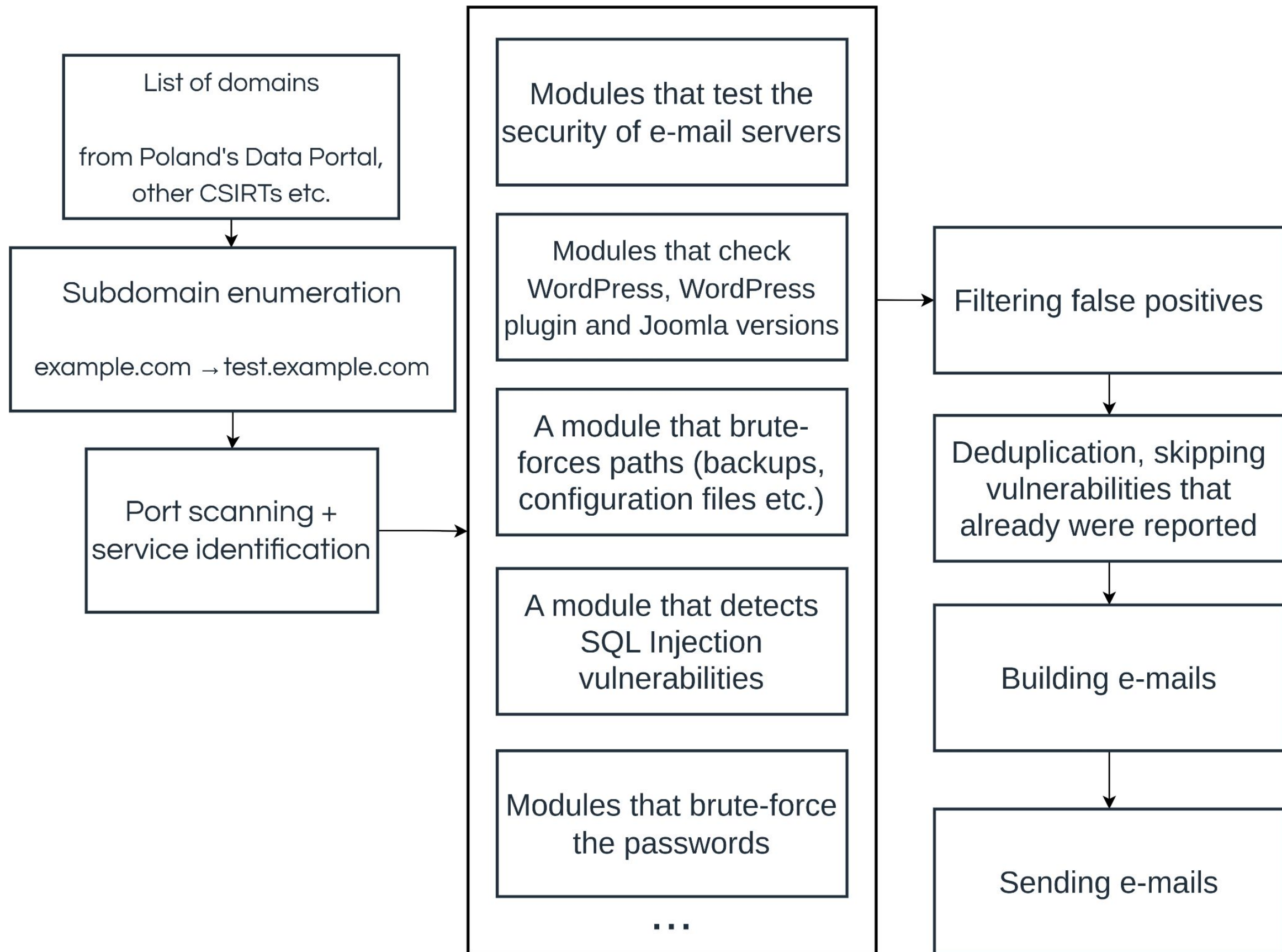
What Artemis does?

Detects known vulnerabilities using Nuclei:

CVE-2023-28343, CVE-2023-23489, CVE-2023-23488, CVE-2022-47986, CVE-2022-47966, CVE-2022-47945, CVE-2022-47003, CVE-2022-47002, CVE-2022-46169, CVE-2022-45933, CVE-2022-45917, CVE-2022-45805, CVE-2022-44877, CVE-2022-4447, CVE-2022-43769, CVE-2022-42233, CVE-2022-41840, CVE-2022-4117, CVE-2022-40881, CVE-2022-40684, CVE-2022-4063, CVE-2022-4060, CVE-2022-4050, CVE-2022-40083, CVE-2022-39952, CVE-2022-3982, CVE-2022-38637, CVE-2022-37042, CVE-2022-36642, CVE-2022-36446, CVE-2022-35914, CVE-2022-35413, CVE-2022-35405, CVE-2022-34045, CVE-2022-33965, CVE-2022-32429, CVE-2022-32409, CVE-2022-32094, CVE-2022-31814, CVE-2022-31656, CVE-2022-31499, CVE-2022-31126, CVE-2022-30525, CVE-2022-30512, CVE-2022-29775, CVE-2022-29464, CVE-2022-29383, CVE-2022-29303, CVE-2022-29078, CVE-2022-29009, CVE-2022-29007, CVE-2022-29006, CVE-2022-28219, CVE-2022-27927, CVE-2022-27593, CVE-2022-26960, CVE-2022-26833, CVE-2022-26352, CVE-2022-26148, CVE-2022-26138, CVE-2022-26134, CVE-2022-25369, CVE-2022-25125, CVE-2022-25082, CVE-2022-2488, CVE-2022-2487, CVE-2022-2486, CVE-2022-24816, CVE-2022-2467, CVE-2022-24260, CVE-2022-24112, CVE-2022-23944, CVE-2022-23898, CVE-2022-23881, CVE-2022-23178, CVE-2022-2314, CVE-2022-23131, CVE-2022-22972, CVE-2022-22965, CVE-2022-22963, CVE-2022-22954, CVE-2022-22947, CVE-2022-22536, CVE-2022-2185, CVE-2022-21587, CVE-2022-21500, CVE-2022-21371, CVE-2022-1952, CVE-2022-1609, CVE-2022-1574, CVE-2022-1391, CVE-2022-1390, CVE-2022-1388, CVE-2022-1386, CVE-2022-1329, CVE-2022-1162, CVE-2022-1057, CVE-2022-1040, CVE-2022-1020, CVE-2022-1013...

What Artemis does?

- Checks e-mail configuration (SPF, DMARC, open relay).
- Detects SQL Injection vulnerabilities.
- Detects accidentally published VCS repositories.
- Performs version check for WordPress, Joomla and WordPress plugins.
- Verifies SSL/TLS configuration.



Add targets

Targets (separated with newlines)

Batch file (should contain one target per line)

Choose File

No file chosen

Tag

You may provide any string here - it will be saved in the task results in the database so that you can e.g. use the value when processing the results automatically.

Start scan

Raw results

 **artemis**

[Add targets](#) [View targets](#) [View results](#) [Task queue](#) [Restart crashed tasks](#) [API](#)

Analysis of test.local

Tasks

All interesting findings [All tasks](#)

Show entries Search:

created at	receiver	task	headers	status: reason
2023-02-20 08:47:36	port_scanner	test.local	INTERESTING origin:classifier type:domain	Found ports: 8001 (service: http ssl: False), 8002 (service: http ssl: False), 8008 (service: http ssl: False), 8011 (service: http ssl: False)
2023-02-20 08:47:44	mail_dns_scanner	test.local	INTERESTING origin:classifier type:domain	Found problems: DMARC record is not present
2023-02-20 08:47:54	directory_index	test.local:8001	INTERESTING origin:port_scanner service:http type:service	Found directories with index enabled: http://test.local:8001/files/
2023-02-20 08:47:56	vcs	test.local:8002	INTERESTING origin:port_scanner service:http type:service	Found version control system data: git
2023-02-20 08:47:56	port_scanner	213.32.88.99	INTERESTING origin:classifier type:ip	Found ports: 8001 (service: http ssl: False), 8002 (service: http ssl: False), 8008 (service: http ssl: False), 8011 (service: http ssl: False)

Example e-mail

1. The following addresses contain version control system data:

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

2. The following addresses contain old Joomla versions:

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

If a site is no longer used, we recommend shutting it down to eliminate the risk of exploitation of known vulnerabilities in older Joomla versions. Otherwise, we recommend regular Joomla core and plugin updates.

3. The following domains don't have properly configured e-mail sender verification mechanisms:

- [REDACTED].pl: Valid SPF record not found
- [REDACTED].pl: Valid DMARC record not found

Such configuration may allow an attacker to send spoofed e-mail messages from these domains.

Artemis is open-source

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

We invite you to use Artemis and add your own modules!

Not all modules are open-source yet - we are currently open-sourcing the
module to build e-mails.

How to write a new module

Let's assume you want to check whether **the URL contains the string suspicious**.

```
class CustomScanner(ArtemisBase):
    # Module name that will be displayed
    identity = "custom"

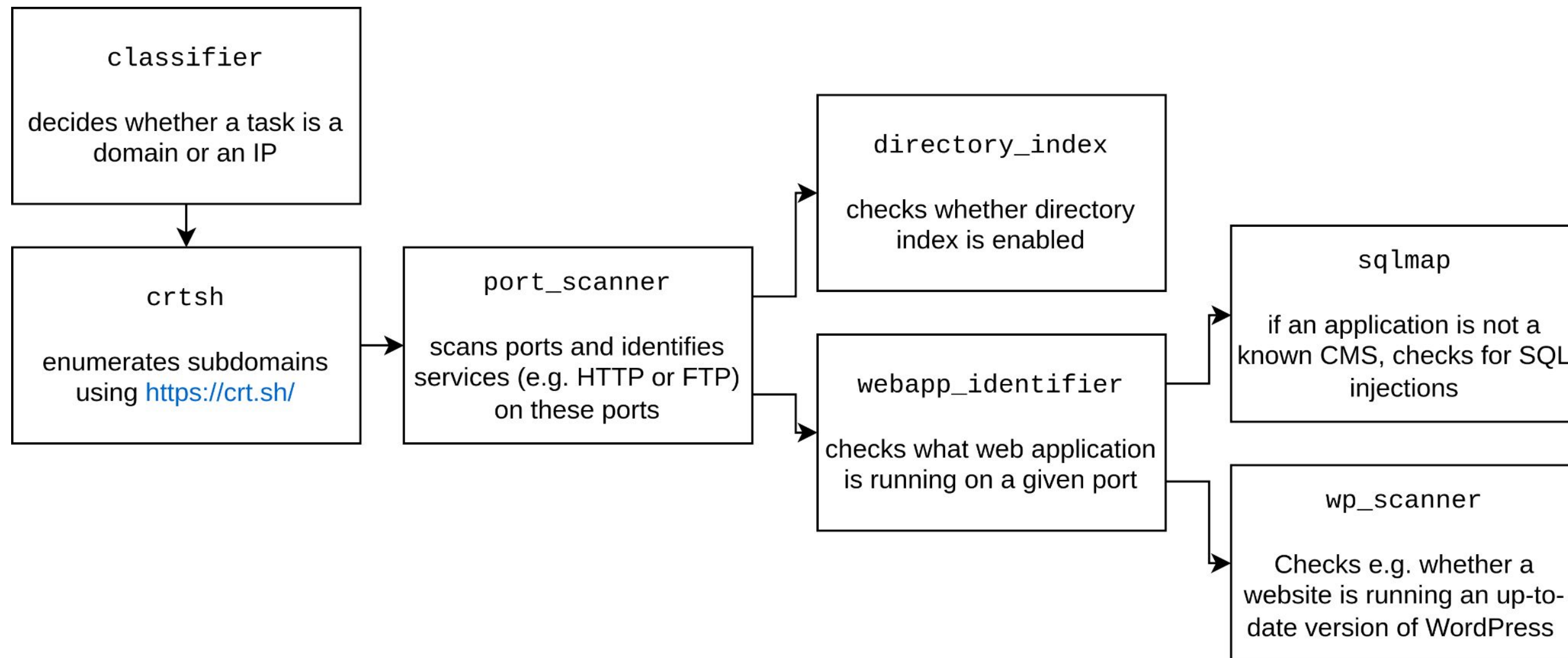
    # Types of tasks that will be consumed by the module - here,
    # open ports that were identified as containing a HTTP/HTTPS service.
    filters = [{"type": TaskType.SERVICE, "service": Service.HTTP}]

    def run(self, task: Task) -> None:
        # Will convert the identified service to the form of a URL,
        # e.g. http://domain.com:8001/
        url = get_target_url(task)

        if "suspicious" in url:
            status = TaskStatus.INTERESTING
            reason = "suspicious link detected!"
        else:
            status = TaskStatus.OK
            reason = None

        self.db.save_task_result(task=task, status=status,
                                status_reason=reason)
```


Modules can produce/consume various types of objects



Alternatives (1/2)

Osmedeus

The data flow uses text files → hard to have a robust data flow.

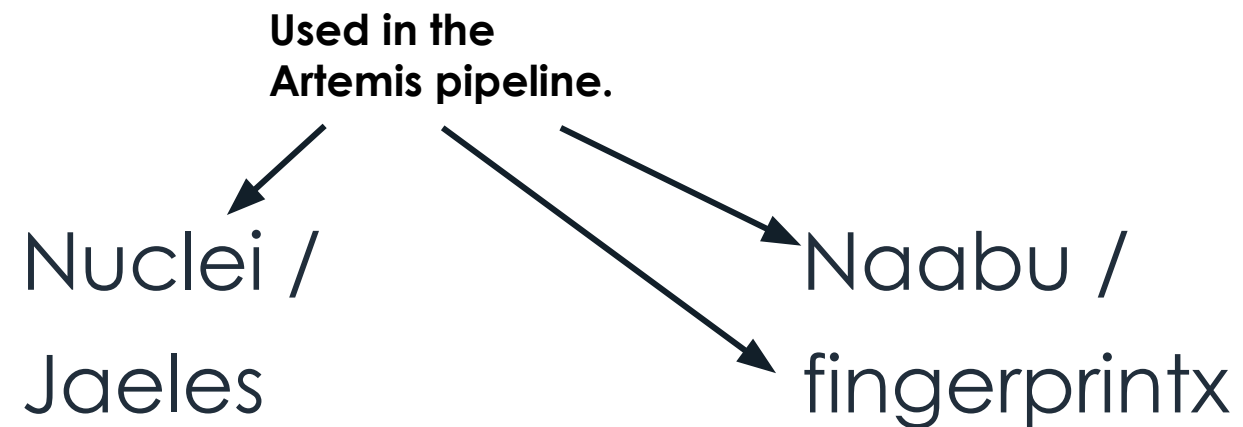
reNgin

The purpose of the system is different, we would need to manage a fork.

reconFTW

Written in Bash, therefore hard to extend in a robust way.

Alternatives (2/2)



Do not do reconnaissance.
Can be used as part of the pipeline if we already know the subdomains.

Provides only a subset of information: ports and services on these ports.

nmap

Even with script support it would be hard to adapt to e.g. enumerate subdomains.

zmap /
masscan

Solves one problem well: port scanning. Not able to build pipelines out of multiple types of tools.

Scanning

We've been scanning the websites since **January**.

We have already scanned **~31k domains and IP addresses** and **~85k** subdomains.

Reported issues in 2023 so far (January-May)

- ~21.5k SSL/TLS misconfigurations
- ~14.1k SPF/DMARC misconfigurations
- ~9.2k obsolete Joomla, WordPress or WordPress plugin versions
- ~5.4k information leaks: AXFR, directory listing, phpinfo(), etc.
- ~1.5k high/critical vulnerabilities from Nuclei or sqlmap
- 697 exposed backups, source code, database dumps or logs
- 75 exposed RDPs

~52.4k in total

Reporting and reactions

Our current workflow:

1. a package of reports is prepared semi-automatically
2. 1st line sends e-mails to best-known contacts
3. 1st line manages the follow-up communication (when needed)

Responses are mostly positive, but:

- they sometimes include bug reports (which are frequently correct!)
- sometimes the institutions report false positives
- sometimes we need to fix the contacts
- sometimes we are ignored
- sometimes the institutions fix the vulnerabilities without responding

Challenges

- Distinguishing true from false positives
Example: if we detect that /wp-config.php.bak is present, we need to check whether it is indeed an exposed configuration file. We have lots of heuristics to keep the number of false positives low.
- Rate limiting in distributed environment
Making sure no server is overloaded with requests is tricky with multiple modules.
- Scanning is slow
The biggest cause is the per-host limiting behavior.
- Deduplication
We need heuristics to detect whether two similar vulnerabilities on institution.com and www.institution.com are in fact one.
- Contact database
Maintaining an up-to-date contact database requires significant effort.
- Running a non-trivial production service
We have a medium-scale service where we sometimes need to troubleshoot unexpected administrative problems.
- Prioritizing the scans

Conclusion for administrators

Yes, they seem obvious - but following them would greatly decrease the number of problems found by Artemis.

Conclusion for administrators: updates

Detecting obsolete software versions with known bugs is **easy**.

Exploits for known vulnerabilities exist.

Conclusion for administrators: archived websites

Outside check allows to find archived or forgotten websites that can:

- use obsolete software (containing known vulnerabilities),
- be built without following of modern software engineering practices:

```
query("SELECT * FROM posts WHERE id = " . $_GET["id"])
```

Control what is exposed.

Conclusion for administrators: *security by obscurity*

Scanners can (and will) find:

- /backup.zip placed *temporarily* on the server,
- a test subdomain.

Conclusion for administrators: configuration files, logs, backups, code repositories...

`https://[domain]:443/.git`

`https://[domain]:443/uploads/`

`https://[domain]:443/config.inc`

`https://[domain]:443/config.php.save`

`https://[domain]:443/configuration.php.bak`

`https://[domain]:443/configuration.php.save`

`https://[domain]:443/wp-config.php~`

`https://[domain]:443/wp-config.php.bak`



`https://[domain]:443/wp-config.php.old`

`https://[domain]:443/wp-config.php.save`

...













Conclusion for developers: Roundcube misconfiguration - a case study



	<input type="text" value="Username"/>
	<input type="password" value="Password"/>
<input type="button" value="LOGIN"/>	

Conclusion for developers: Roundcube misconfiguration

Index of /webmail/temp

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 0a0.thumb	2020-02-17 14:13	6.8K	
 0a5.thumb	2019-12-05 22:52	8.2K	
 0b8.thumb	2020-04-01 21:13	8.4K	
 0b6.thumb	2020-02-14 08:56	6.8K	
 0bc.thumb	2019-02-03 20:35	6.8K	
 0c1.thumb	2020-04-01 20:58	6.5K	
 0c3.thumb	2019-11-15 00:10	7.2K	
 0d0.thumb	2020-04-01 20:46	9.4K	
 0e6.thumb	2020-04-01 20:46	13K	
 0ea.thumb	2020-04-01 21:13	9.7K	
 0ed.thumb	2019-10-13 23:19	8.6K	
 0ed.thumb	2020-09-15 10:11	9.4K	

Conclusion for developers: Roundcube misconfiguration

```
← → ↻ 🔒 github.com/roundcube/roundcubemail/blob/master/INSTALL
167
168 SECURE YOUR INSTALLATION
169 =====
170
171 Access through the webserver to the following directories should be denied:
172
173     /config
174     /temp
175     /logs
176
177 Roundcube uses .htaccess files to protect these directories, so be sure to
178 allow override of the Limit directives to get them taken into account. The
179 package also ships a .htaccess file in the root directory which defines some
180 rewrite rules. In order to properly secure your installation, please enable
181 mod_rewrite for Apache webserver and double check access to the above listed
182 directories and their contents is denied.
```

Why this is not a good approach? What conclusions can we draw?

Conclusion for CSIRTs

- Unfortunately, there are still low-hanging vulnerabilities
- Many good offensive tools are available
even plain Nuclei or WordPress/Joomla version check would find many vulnerabilities
- Not a huge project: currently 1 FTE: development + operations
Managing contact list and sending prepared e-mails not included.
- 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.

It is easy to start a similar
project and improve the
security of your constituency

Plans

- Develop the system:
 - add modules to detect new vulnerabilities
 - autoreporter - open source and make fully automatic
- Regularly scan multiple groups of domains (including most popular .pl domains)

Questions?

[https://github.com/
CERT-Polska/Artemis](https://github.com/CERT-Polska/Artemis)