SYNACKTIV

Finding the Needle in the Haystack with Dicozorus

A New Companion for Advanced Web Fuzzing

30/06/2026

Whoami

SYNACKTIV

Vincent Herbulot

Pentester **@Synacktiv**

vincent.herbulot@synacktiv.com

@us3r777

- Pentester and Security Researcher
- My main focus is web applications
- Training instructor
  - Attacking Web Applications
  - Practical Web 0-Day Hunting

About Synacktiv

SYNACKTIV

- French offensive security company
- 189 security experts
- 5 Departments
  - Pentest / Red Team
  - Reverse
  - Development
  - Revel·IO
  - Incident Response
- We are hiring!

Overview

SYNACKTIV

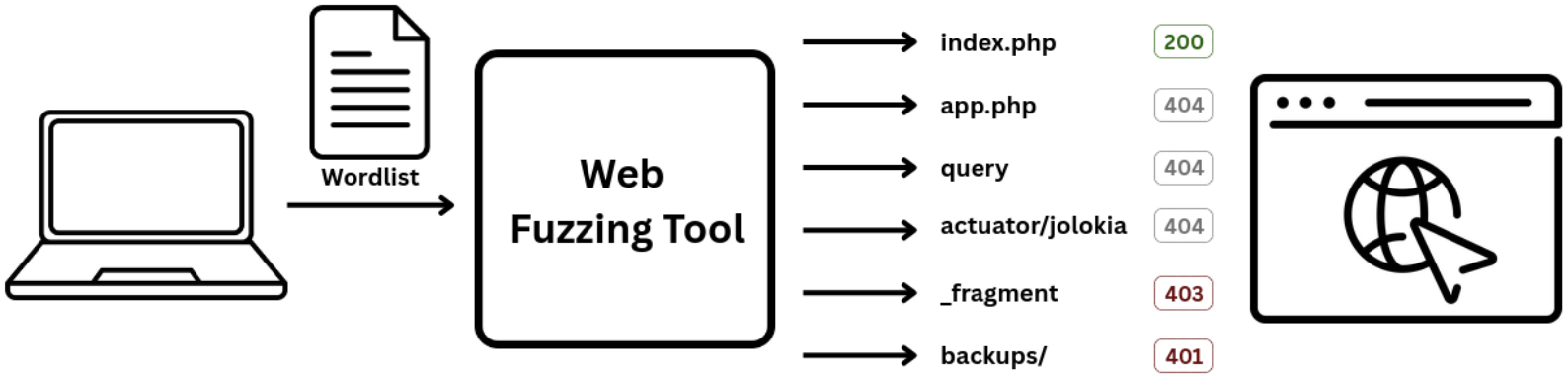
- Introduction
- Common problems with wordlists
- Improving your wordlists
- Dicozorus concepts and CLI
- Demo !
- Results
- Conclusion

Introduction

Web Fuzzing

SYNACKTIV

- Fuzzing is a critical step during web penetration tests
- Using a tool, the auditor iterates through a list of entries
- This allows them to map exposed endpoints



A critical task

SYNACKTIV

- Penetration testing / Red teams heavily rely on fuzzing
- Finding a critical endpoint can make the difference
- Results heavily depend on the quality and the relevancy of the wordlists

Synacktiv's statistics

- 50% of RedTeam initial entry points are web applications
- In 90% of the cases, initial entry points lead to a full infrastructure compromise

The trigger

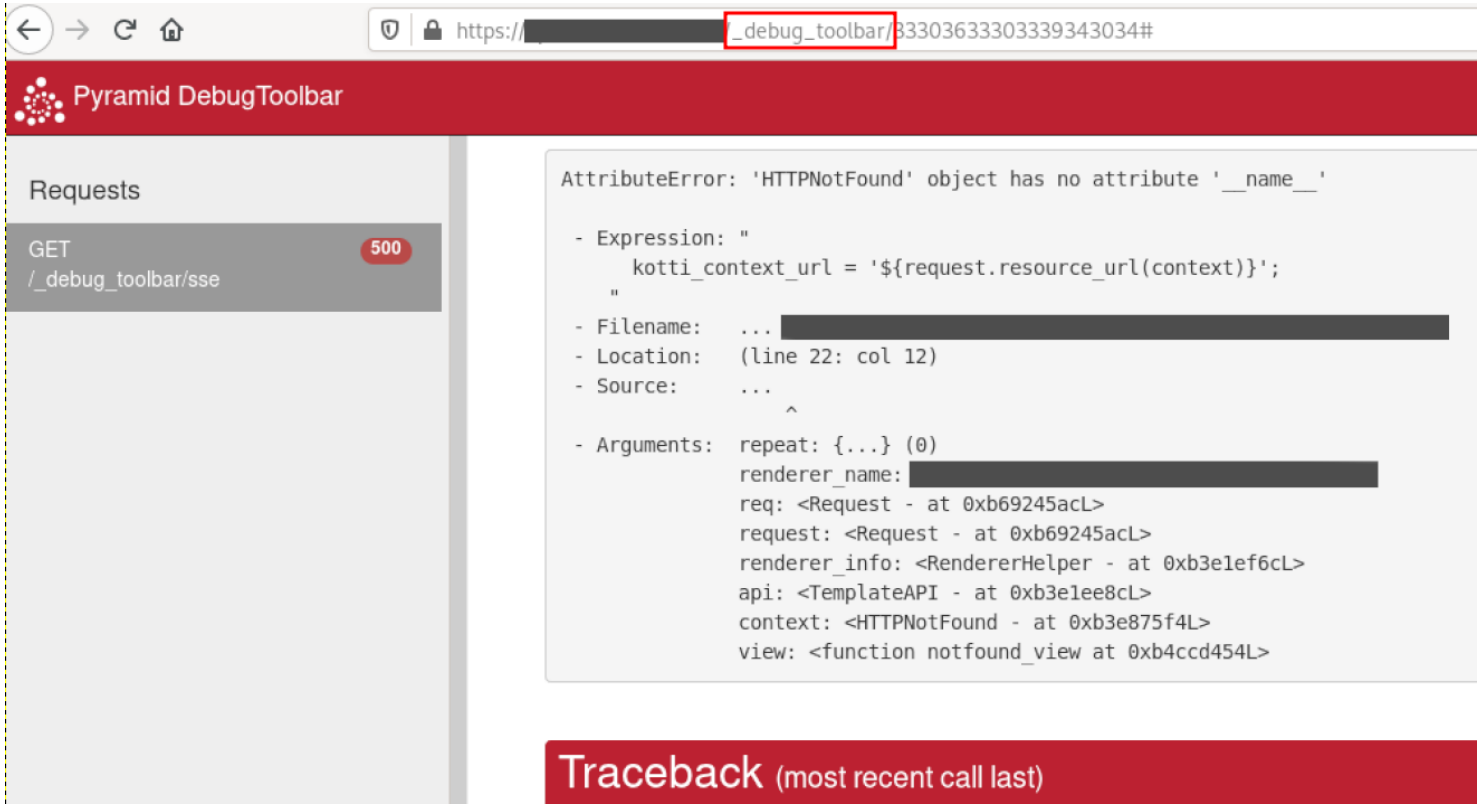
SYNACKTIV

- I used to blindfully rely on public wordlists
- During a RedTeam in 2020, we nearly missed a critical endpoint
- We had a huge target with many web applications exposed …
  - but no quick win was discovered at first

The trigger

SYNACKTIV

- In a second iteration, my colleague came up with this:



The trigger

SYNACKTIV

- The entry **_debug_toolbar/** corresponds to a debugger
  - It is provided by the Pyramid framework
  - It gives you access to a Python interpreter, allowing RCE
- Yet I couldn't find it in **any of my wordlists**
  - Not in dirsearch dicc.txt
  - Not in quickhits.txt
  - Not in rafts wordlists
  - ...
- So I decided that it was time to start maintaining my own wordlists

Common problems with wordlists

What should be improved

SYNACKTIV

- First let's analyse our current tools and processes
- What tools and wordlists are we currently using?
- What could be improved?

The tools

SYNACKTIV

- We use a variety of fuzzing tools depending on everyone's preferences
- The tool used has minimal to no impact on the results
  - As long as you know how to use your favorite tool
  - Threads / Recursivity / Filtering / Encoding …

Commonly used wordlists

SYNACKTIV

- Each pentester usually has their own favorite ones
- Some wordlists often come up:

Vulnerability-focused	Generic	Technology Specific
dicc.txt (Dirsearch)	raft-*-files.txt	php.txt / jsp.txt / asp.txt (Assetnotes)
quickhits.txt (Seclists)	raft-*-dirs.txt	Local Files Windows (Burp)
fuzz.txt (BoOom)	raft-*-words.txt	Local Files Linux (Burp)

- Most wordlists can be OK, as long as:
  - You know what's inside
  - You run it on the appropriate target/context

Common problem: missing entries

SYNACKTIV

- If the entry is not in your wordlist, you will not find it!
  - Check your wordlists
  - Are your usual high-value targets in the wordlist?


```
$ grep '_debug_toolbar' SecLists/Discovery/Web-Content/quickhits.txt; [ $? -eq 1 ] && echo "Entry not found"
Entry not found
$ grep '_debug_toolbar' dirsearch/db/dicc.txt; [ $? -eq 1 ] && echo "Entry not found"
Entry not found
$ grep '_debug_toolbar' fuzz.txt/fuzz.txt; [ $? -eq 1 ] && echo "Entry not found"
Entry not found
```

Common problem: sorting

SYNACKTIV

- Most wordlists are not sorted
  - Some of them are alphabetically sorted
    - quickhits.txt / dicc.txt (dirsearch) / hfuzz.txt

```
$ grep -n  "Telerik.Web.UI.WebResource.axd" hfuzz.txt
164749:Telerik.Web.UI.WebResource.axd
164750:Telerik.Web.UI.WebResource.axd?type=r%61u
164752:Telerik.Web.UI.WebResource.axd?type=rau
```

- When time is limited, top priority entries should be at the top of your wordlist
  - Even more important when the size of the wordlist grows
- If the wordlist is smartly sorted, adapting its size to the scope is easy

Common problem: Improper sizing

SYNACKTIV

- When the wordlist is not sorted, adjusting its size is gambling with your results
- Yet fitting the size of the wordlist to your scope is sometimes mandatory
  - Very large scope
  - Narrow scope
  - Undersized server / Network bottleneck

```
2567 quickhits.txt
9680 dicc.txt
37050 raft-large-files.txt
62281 raft-large-directories.tx
2032825 onelistforallshort.txt
```

Common problem: Irrelevant entries

SYNACKTIV

- If you are scanning a **WordPress**
  - you probably don't want to use a wordlist with .jsp or .aspx entries
- If you are scanning a **web path**
  - you probably don't want to include HTTP methods
- If you are scanning for an **LFI**
  - you will want to adapt the paths to the targeted OS (Linux / Windows / etc.)
- If entries contain **payloads in GET parameters**
  - you want to ensure that it will work with your scanning configuration

Common problem: Junk entries

- By carefully reviewing your wordlists you may come across junk entries
  - They are usually the result of automated generation
  - They might also have been inadvertently introduced or altered

```
$ grep -P  "[^\x00-\x7F]"  onelistforallshort.txt
[...]
awst/è‰¾ç»´æ–¯ç‰¹/background/advertisingfile/advertising.asp
5-è®ºæ–‡/cnkiæŸ¥é‡æŠ¥å‘Š/è¶³çƒè®ç»ƒç›‘æŽ§è½¯ä»¶ç³»ç»Ÿçš„è®¾è®¡ä¸Žå¼€å‘-æ–‡æœ¬å¤åˆ¶æ£€æµ‹æŠ¥å‘Šå•ï¼ˆå…¨æ–‡å¯¹ç…§ï¼‰_files/redimage.aspx
/XY>.7d8T\205pZM@www.whitelisteddomain.tld⁺@localdomain.pw/
/www.whitelisteddomain.tld⁺&@localdomain.pw#⁺@www.whitelisteddomain.tld/
//www.whitelisteddomain.tld@□□□□□□□□□a□□□□□□/%2f..
  <title>payloads/env.txt at main · coffinxp/payloads · GitHub</title>
[...]
```

Common problem: Junk entries

SYNACKTIV

- Every time the wordlist is used, those entries are thrown to your target
  - This increases your chances of detection
  - This also increases the time of your scans
- Review wordlists that you frequently used

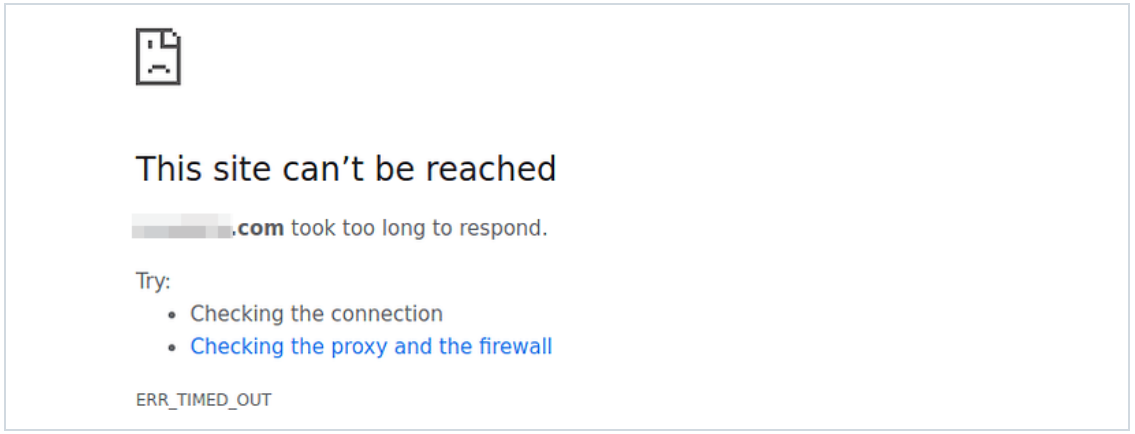
Common problem: Dangerous entries?

SYNACKTIV

- Some wordlists also contain "dangerous" entries

```
shutdown
reboot
actuator/shutdown
delete-data.php
```

- This can be fine as long as you are ok with shutting your target down …



Improving your wordlists

Improving your wordlists

SYNACKTIV

- Now how can we address the aforementioned challenges?
  - Reviewing each wordlist manually?
  - Ideally, we want to have a wordlist for each situation
  - Maintaining multiple wordlists can quickly become challenging
- To solve these problems, I developed a tool : Dicozorus.
- It is designed to:
  - Bind entries with metadata
  - Classify entries
  - Generate wordlists tailored to different contexts
  - Easily ingest and review existing wordlists

Bind entries with metadata

SYNACKTIV

- **Bind entries with metadata**
  - **Criticality**: CRITICAL | HIGH | MEDIUM | LOW | INFO | UNRANKED
  - **Count**: How many times the entry was seen
  - **Category**: Type of "vulnerability" it is attached to
    - RCE, SQLI, XXE, ADMIN_INTERFACE, KNOWN_APP, ...
  - **Tags**: list of keywords to store additional information
    - PHP, JAVA, GO
  - **Reference**: A link that identifies this entry as related to a vulnerability or a known endpoint

Classify entries

SYNACKTIV

- **Classify entries**
  - Once entries are linked to metadata (criticality, category, count), we want to be able to:
    - Sort the entries (by criticality, and count most of the time)
    - Select entries that correspond to specific criteria (criticality, tag, category, etc.)

Generate wordlists tailored to our context

SYNACKTIV

- **Generate wordlists tailored to our context**
  - With enriched / classified entries, we must be capable of:
    - Generating wordlists sorted by criticality
    - Adapting the size of the wordlist to the size of the scope
    - Filtering out entries that are not relevant to our scope

Ingesting and reviewing existing wordlists

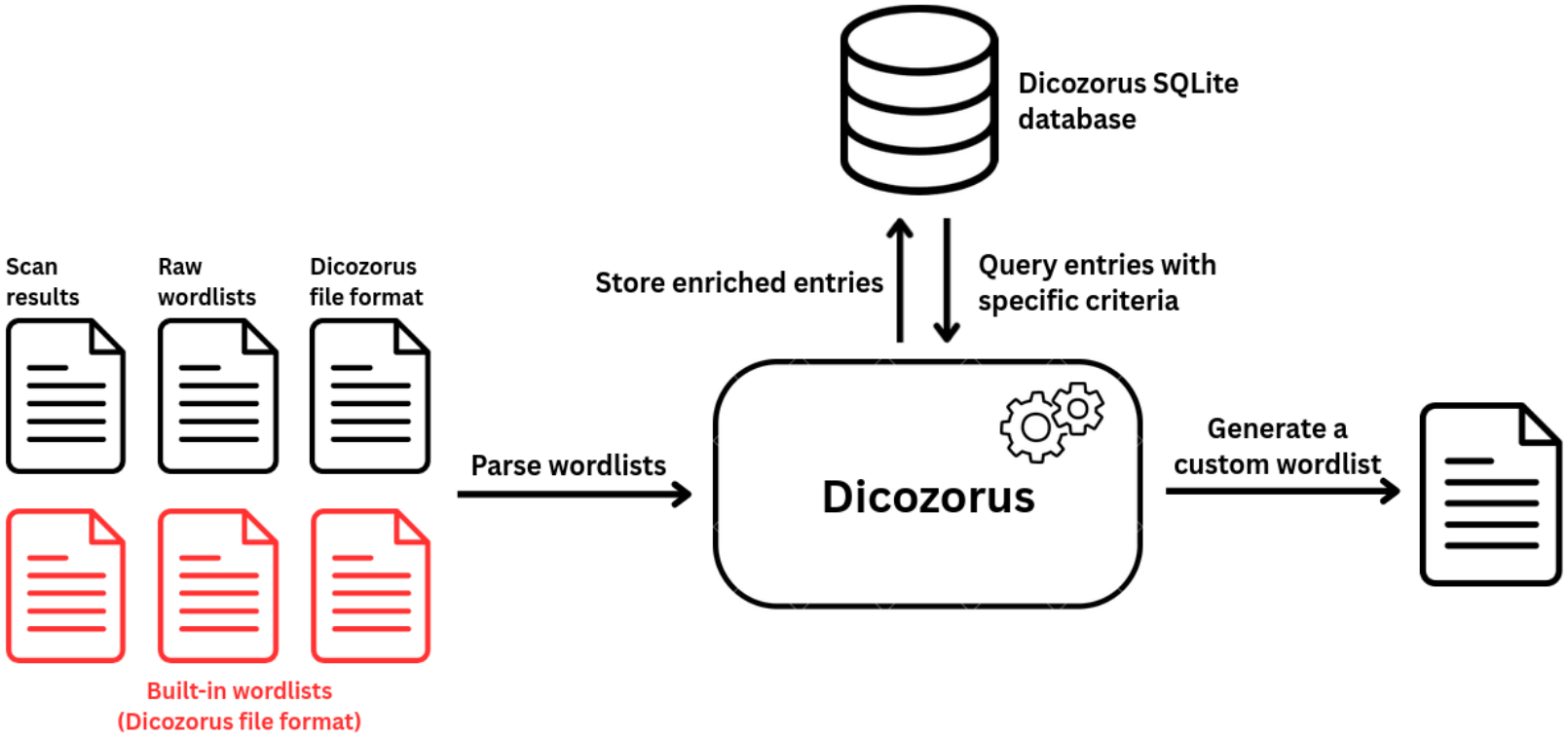
SYNACKTIV

- **Easily ingest and review existing wordlists**
  - **Ingest**: easily parse different formats
    - Raw wordlists
    - Patator / dirsearch results files
    - **Dicozorus custom format**
  - **Review**: analyze existing wordlist / entries
    - Identify missing / additional entries
    - Give metadata about known entries

Dicozorus concepts

Dicozorus overview

SYNACKTIV



Dicozorus file format

SYNACKTIV

- Just basic CSV files

```
## path, criticality, count, category, tags, reference
"_fragment","CRITICAL","1","RCE","PHP","https://www.ambionics.io/blog/symfony-secret-fragment"
```

- It can be partially filled

```
"2019/","INFO","1","","",""
```

- I usually group entries by criticality

```
critical.wordlist high.wordlist medium.wordlist low.wordlist info.wordlist
```

Dicozorus database

SYNACKTIV

- The database does not store more information than CSV files
  - Just facilitate indexing / aggregation / querying
- When feeding the wordlist into the database, Dicozorus will:
  - Merge entries with the same path
    - Keep the highest criticality
    - Increase the count
- You can choose to start with an empty database or use our built-in wordlists

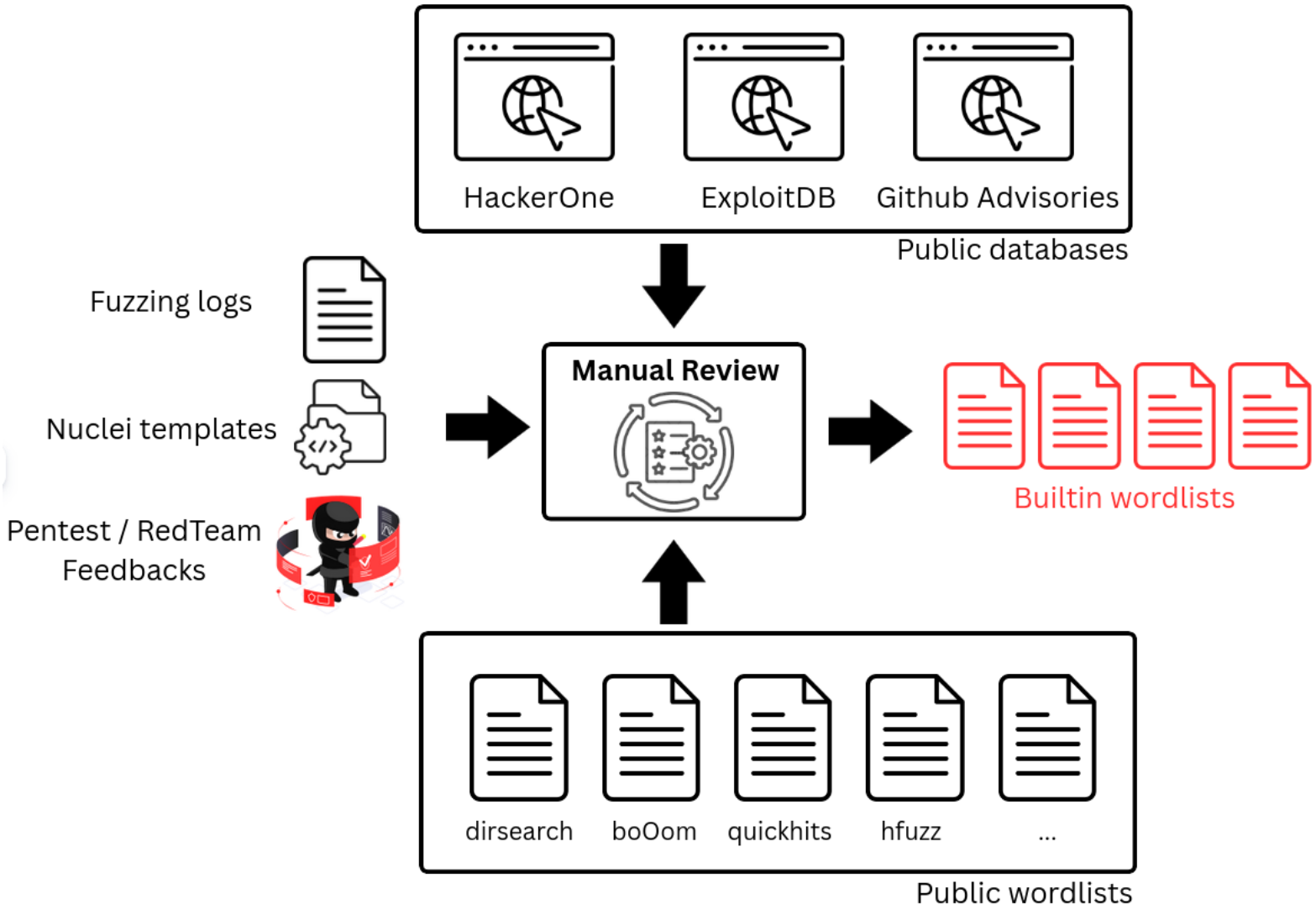
Dicozorus built-in wordlists

SYNACKTIV

- Along with Dicozorus, we provide **a set of built-in wordlists**
  - These wordlists contain already enriched entries
  - The enrichment was done by **manually reviewing each entry**
- It also contains a set of UNRANKED entries
  - They come from sources that we consider relevant
  - They were **not manually reviewed**

Built-in wordlists sources

SYNACKTIV



Dicozorus CLI

Dicozorus CLI

SYNACKTIV

- Dicozorus is a Python-based command line tool
- It provides 6 sub-commands

INIT	Initialize the Dicozorus database
FEED	Feed new files into the database
GEN	Generate a wordlist
MODIFY	Modify entries in the database
CHECK	Compare the current database against a wordlist
STATS	Give statistics of the current database

Dicozorus INIT

SYNACKTIV

- **INIT** - Initialize the Dicozorus database

```
$ dicozorus init -h
usage: dicozorus init [-h] [-e] [-F] [-m] [-D] [-W WORDLISTS_FOLDER]

options:
  -h, --help            show this help message and exit
  -e, --empty           Initialize an empty database. If not specified, database will contain entries
                        from data/*.wordlist
  -F, --force           Force database initialization (DELETE database content)
  -m, --minimal         Initialize the dicozorus database with criticality wordlists only
  -D, --dangerous       Include the dangerous.wordlist file. This one contains dangerous entries such
                        as "actuator/shutdown"
  -W, --wordlists-folder WORDLISTS_FOLDER
                        A folder that stores the initialization wordlists. By default dicozorus
                        will look in ~/.dicozorus/wordlists/
```

Dicozorus INIT

SYNACKTIV

- **INIT** - Initialize the Dicozorus database

```
$ dicozorus init -F
[+] Initializing dicozorus database
[+] Creating dicozorus tables
[+] Parsing dicozorus CSV file ~/.dicozorus/wordlists/critical.wordlist
[+] Parsing dicozorus CSV file ~/.dicozorus/wordlists/high.wordlist
[+] Parsing dicozorus CSV file ~/.dicozorus/wordlists/medium.wordlist
[+] Parsing dicozorus CSV file ~/.dicozorus/wordlists/low.wordlist
[+] Parsing dicozorus CSV file ~/.dicozorus/wordlists/info.wordlist
[+] Parsing dicozorus CSV file ~/.dicozorus/wordlists/unranked.wordlist
[+] Parsing dicozorus CSV file ~/.dicozorus/wordlists/seen.wordlist
[+] Parsing dicozorus CSV file ~/.dicozorus/wordlists/hackerone.wordlist
[+] Parsing dicozorus CSV file ~/.dicozorus/wordlists/exploitdb.wordlist
[+] Parsing dicozorus CSV file ~/.dicozorus/wordlists/bo0om.wordlist
[+] Parsing dicozorus CSV file ~/.dicozorus/wordlists/dirsearch.wordlist
```

Dicozorus FEED

SYNACKTIV

- **FEED** - Populate the database by ingesting files

```
$ dicozorus feed -h
usage: dicozorus feed [-h] [-w WORDLIST [WORDLIST ...]] [-p PATATOR [PATATOR ...]] [-d DIRSEARCH [DIRSEARCH ...]]
                      [-D DICOZORUS [DICOZORUS ...]] [-i IGNORE_COMMENTS] [-f FILTER] [-u]
                      [-c {UNRANKED,INFO,LOW,MEDIUM,HIGH,CRITICAL}]

options:
  -h, --help            show this help message and exit
  -w, --wordlist WORDLIST [WORDLIST ...]
                        Feed dicozorus with one or more wordlists.
  -p, --patator PATATOR [PATATOR ...]
                        Feed dicozorus with one or more patator CSV files
  -d, --dirsearch DIRSEARCH [DIRSEARCH ...]
                        Feed dicozorus with one or more dirsearch results files
  -D, --dicozorus DICOZORUS [DICOZORUS ...]
                        Feed dicozorus with one or more dicozorus CSV files
  -i, --ignore-comments IGNORE_COMMENTS
                        Ignore all entries starting with the specified string
  -f, --filter FILTER   Comma separated HTTP codes. Only treat entries matching specified HTTP codes.
  -u, --update-count-only
                        Do not insert any new entries in dicozorus DB. (only updatecount)
  -c, --criticality {UNRANKED,INFO,LOW,MEDIUM,HIGH,CRITICAL}
                        Set the criticality of the provided entries
```

Dicozorus GEN

SYNACKTIV

- **GEN** - Generate a wordlist matching the specified criteria

```
$ dicozorus gen -h
usage: dicozorus gen [-h] [-o OUTPUT] [-m MAX_ENTRIES] [-s SORT] [-S] [-D] [-F] [-P]
                     [--min-criticality {UNRANKED,INFO,LOW,MEDIUM,HIGH,CRITICAL} | -c CRITICALITY]
                     [-f FILTER | -t SELECT_BY_TAGS]

options:
  -h, --help            show this help message and exit
  -o, --output OUTPUT   Save the generated wordlist to the specified file
  -m, --max-entries MAX_ENTRIES
                        Maximum number of entries to generate
  -s, --sort SORT       Specify the criteria to use to sort entries. Comma separated list of values in
                        (criticality, count, name, type). Default is criticality,count
  -S, --shuffle         Print a randomly shuffled version of the selected entries
  -D                    Only output entries of type DIR. Can be combined with -F and -P. Default is -DFP
  -F                    Only output entries of type FILE. Can be combined with -D and -P. Default is
                        -DFP
  -P                    Only output entries of type PATH. Can be combined with -D and -F. Default is
                        -DFP
  --min-criticality {UNRANKED,INFO,LOW,MEDIUM,HIGH,CRITICAL}
                        Only output entries equal or greater than the specified criticality
  -c, --criticality CRITICALITY
                        Comma separated values within ["UNRANKED", "INFO", "LOW", "MEDIUM", "HIGH",
                        "CRITICAL"]. Ex:HIGH,CRITICAL
  -f, --filter FILTER   Filter out entries matching regular expression
  -t, --select-by-tags SELECT_BY_TAGS
                        Select entries based on a comma separated list of tags
```

Dicozorus GEN

SYNACKTIV

- Generate a wordlist sorted by criticality and count

```
$ dicozorus gen -o wordlist.txt
```

- Generate a 1000 entries wordlist

```
$ dicozorus gen -m 1000 -o wordlist.txt
```

- All CRITICAL and HIGH entries with a PHP tag

```
$ dicozorus gen -t PHP -c CRITICAL,HIGH -o php.txt
```

Dicozorus GEN

SYNACKTIV

- **GEN** - Sample output

```
$ dicozorus gen -F -c CRITICAL -m 20
index.php
ajax.php
search.php
login.htm
script
app
setup.cgi
xmlrpc.php
code
dig.php
spip.php
filter
fw.login.php
index.jsp
protocol.csp
query
session
upgrade_handle.php
cgi
data
```

Dicozorus GEN

SYNACKTIV

- **GEN** - Sample verbose output

```
$ dicozorus -v gen -F -c CRITICAL -m 20
[*] Executing query: SELECT * FROM dicozorus_entries WHERE (type="FILE") AND (criticity in (5)) ORDER BY criticity DESC, count DESC LIMIT 20
[*] index.php [type: FILE, criticity: CRITICAL, count: 43, category: RCE, taglist: ['PHP'], reference: https://nvd.nist.gov/vuln/detail/CVE-2020-20601]
[*] ajax.php [type: FILE, criticity: CRITICAL, count: 15, category: RCE, taglist: ['PHP'], reference: https://nvd.nist.gov/vuln/detail/CVE-2023-333440]
[*] search.php [type: FILE, criticity: CRITICAL, count: 15, category: RCE, taglist: ['PHP'], reference: https://nvd.nist.gov/vuln/detail/CVE-2021-32305]
[*] login.htm [type: FILE, criticity: CRITICAL, count: 14, category: RCE, taglist: [''], reference: https://nvd.nist.gov/vuln/detail/CVE-2020-26919]
[*] script [type: FILE, criticity: CRITICAL, count: 14, category: RCE, taglist: ['JAVA'], reference: https://jenkins.io/doc/book/managing/script-console/]
[*] app [type: FILE, criticity: CRITICAL, count: 13, category: RCE, taglist: ['JAVA'], reference: https://nvd.nist.gov/vuln/detail/CVE-2023-2735]
[*] setup.cgi [type: FILE, criticity: CRITICAL, count: 11, category: RCE, taglist: [''], reference: https://nvd.nist.gov/vuln/detail/CVE-2024-3056]
[*] xmlrpc.php [type: FILE, criticity: CRITICAL, count: 11, category: RCE, taglist: ['PHP'], reference: https://nvd.nist.gov/vuln/detail/CVE-2023-43187]
[*] code [type: FILE, criticity: CRITICAL, count: 5, category: RCE, taglist: ['PYTHON'], reference: https://nvd.nist.gov/vuln/detail/CVE-2022-24288]
[*] dig.php [type: FILE, criticity: CRITICAL, count: 5, category: RCE, taglist: ['PHP'], reference: https://nvd.nist.gov/vuln/detail/CVE-2009-1361]
[*] spip.php [type: FILE, criticity: CRITICAL, count: 5, category: RCE, taglist: ['PHP'], reference: https://nvd.nist.gov/vuln/detail/CVE-2023-27372]
[*] filter [type: FILE, criticity: CRITICAL, count: 4, category: RCE, taglist: ['JAVASCRIPT'], reference: https://nvd.nist.gov/vuln/detail/CVE-2025-1302]
[*] fw.login.php [type: FILE, criticity: CRITICAL, count: 4, category: RCE, taglist: ['PHP'], reference: https://nvd.nist.gov/vuln/detail/CVE-2020-17505]
[*] index.jsp [type: FILE, criticity: CRITICAL, count: 4, category: RCE, taglist: ['JAVA'], reference: https://nvd.nist.gov/vuln/detail/CVE-2020-9484]
[*] protocol.csp [type: FILE, criticity: CRITICAL, count: 4, category: RCE, taglist: [''], reference: https://nvd.nist.gov/vuln/detail/CVE-2025-34152]
[*] query [type: FILE, criticity: CRITICAL, count: 4, category: RCE, taglist: ['JAVASCRIPT'], reference: https://nvd.nist.gov/vuln/detail/CVE-2025-1302]
[*] session [type: FILE, criticity: CRITICAL, count: 4, category: RCE, taglist: ['TYPESCRIPT'], reference: https://nvd.nist.gov/vuln/detail/CVE-2026-22812]
[*] upgrade_handle.php [type: FILE, criticity: CRITICAL, count: 4, category: RCE, taglist: ['PHP'], reference: https://www.cvedetails.com/cve/CVE-2018-14933/]
[*] cgi [type: FILE, criticity: CRITICAL, count: 3, category: RCE, taglist: ['CGI'], reference: https://nvd.nist.gov/vuln/detail/CVE-2022-25061]
[*] data [type: FILE, criticity: CRITICAL, count: 3, category: RCE, taglist: ['JAVASCRIPT'], reference: https://nvd.nist.gov/vuln/detail/CVE-2025-1302]
```

Dicozorus MODIFY

SYNACKTIV

- **MODIFY** - Manually edit the database content

```
$ dicozorus modify -h
usage: dicozorus modify [-h] (--add | --rm | --update) (--entry ENTRY | --wordlist WORDLIST) [--criticality CRITICALITY] [--count COUNT]
                        [--category CATEGORY] [--taglist TAGLIST] [--reference REFERENCE]

options:
  -h, --help            show this help message and exit
  --add                 Add the specified entry
  --rm                  Remove the specified entry
  --update              Update the specified entry
  --entry ENTRY         The entry to be processed
  --wordlist WORDLIST   The file containing the entries to be processed
  --criticality CRITICALITY
                        The count for the entry/entries you want to add. Default is Unranked
  --count COUNT         The count for the entry/entries you want to add. Default is 1.
  --category CATEGORY   The category for the entry/entries you want to add. Default is none.
  --taglist TAGLIST     Comma separated taglist for the entry/entries you want to add. Default is none.
  --reference REFERENCE
                        A URL to a resource describing the issue. Default is none.
```

Dicozorus CHECK

SYNACKTIV

- **CHECK** - Check an entry or a wordlist against the current database

```
$ dicozorus check -h
usage: dicozorus check [-h] [-M] (-w WORDLIST | -e ENTRY) [-c CRITICALITY] [-S]

options:
  -h, --help            show this help message and exit
  -M, --missing-in-wordlist
                        Print the entries present in the dicozorus database but not in the specified wordlist. The default behavior
                        is to check for entries missing in the dicozorus database.
  -w, --wordlist WORDLIST
                        The wordlist to be checked.
  -e, --entry ENTRY     The entry to be checked. If the specified entry is present in the database it will be printed along with
                        any information associated to this entry
  -c, --criticality CRITICALITY
                        Comma separated values within ["UNRANKED", "INFO", "LOW", "MEDIUM", "HIGH", "CRITICAL"]. Ex:HIGH,CRITICAL
  -S, --stats-only      Don't print out entries, only display statistics of the considered wordlist
```

Dicozorus CHECK

SYNACKTIV

- Checking what is in Dirsearch wordlist and missing in Dicozorus

```
$ dicozorus check -w dicc.txt.noext
HEAD
.inc.php
.mysql.txt
.xhtml
.png
.axd
.flv
.ico
.PDF
PATCH
.asa
.jpeg
.mp3
POST
getFavicon?host=burpcollaborator.net
.jpg
.0
.aspx
~shutdown
.shtml
.xls
actuator/;/shutdown
[...]
```

Dicozorus CHECK

SYNACKTIV

- Checking for specific entries in Dicozorus

```
$ dicozorus check -e '_debug_toolbar/'
_debug_toolbar/ [type: DIRECTORY, criticality: CRITICAL, count: 1, category: RCE, taglist: ['PYTHON'],
    reference: https://docs.pylonsproject.org/projects/pyramid_debugtoolbar/en/latest/]
$ dicozorus check -e 'bbs/logout.php'
bbs/logout.php [type: PATH, criticality: MEDIUM, count: 1, category: OPEN_REDIRECT, taglist: ['PHP'],
    reference: https://nvd.nist.gov/vuln/detail/CVE-2024-37656]
$ dicozorus check -e 'xwiki/'
xwiki/ [type: DIRECTORY, criticality: LOW, count: 1, category: KNOWN_APP, taglist: ['JAVA'],
    reference: https://xwiki.com/fr/]
```

Dicozorus STATS

SYNACKTIV

- **STATS** - Gives statistics of the current database

```
$ dicozorus stats
[+] Total count: 22374
[+] Entry count by criticality:
        Critical: 344
        High: 697
        Medium: 667
        Low: 1168
        Info: 4262
        Unranked: 15236
[...]
[+] Entry count by category:
        KNOWN_APP: 871
        RCE: 335
        INFO_LEAK: 280
        XSS: 171
        BACKUP: 163
        ADMIN_INTERFACE: 162
        SQL_INJECTION: 144
        CONFIG: 121
        ARBITRARY_FILE_READ: 93
[...]
```

Demo

Results

Results

SYNACKTIV

- Dicozorus has been used internally over the past 5 years
- The built-in wordlists currently holds 22374 entries
  - 7138 entries owns a criticality
  - The higher the criticality the lower the number of entries (more or less)
  - Severity as the primary sorting criterion enables highly targeted wordlists

```
[+] Entry count by criticality:
        Critical: 344
        High: 697
        Medium: 667
        Low: 1168
        Info: 4262
        Unranked: 15236
```

Results

SYNACKTIV

- Comparing Dicozorus with existing wordlists doesn't make much sense
  - The tool is designed to maintain your own wordlists
  - Each pentester will have their own ideas of what should or should not be in a wordlist
    - You don't have to rely on the built-in database
    - You can use it as a tool to maintain / compile wordlists

Results

SYNACKTIV

- Comparing Dicozorus with existing wordlists doesn't make much sense
- **But let's do it anyway ...**

Results

SYNACKTIV

- Comparing the built-in DB with Dirsearch's wordlist
  - Entries - Dicozorus: 22374 | Dirsearch: 9680
  - Entries missing with a level equal to CRITICAL | HIGH | MEDIUM

```
$ dicozorus check -M -w dicc.txt.noext  -c CRITICAL,HIGH,MEDIUM | wc -l
1294
```

  - Some of the missing CRITICAL entries

```
birt/document [...] category: RCE, taglist: ['JAVA'], reference: https://bugs.eclipse.org/bugs/show_bug.cgi?id=538142]
_debug_toolbar/ [...] category: RCE, taglist: ['PYTHON'], reference: https://docs.pylonsproject.org/projects/pyramid_debugtoolbar/en/latest/]
functionRouter [...] category: RCE, taglist: ['JAVA'], reference: https://nvd.nist.gov/vuln/detail/CVE-2022-22963]
SamlResponseServlet [...] 1, category: RCE, taglist: ['JAVA'], reference: https://nvd.nist.gov/vuln/detail/CVE-2022-47966]
scan-engine/update [...] category: RCE, taglist: ['PYTHON'], reference: https://nvd.nist.gov/vuln/detail/CVE-2023-50094]
scriptText [...] category: RCE, taglist: ['JAVA'], reference: https://jenkins.io/doc/book/managing/script-console/]
server/executeExec [...] category: RCE, taglist: ['JAVA'], reference: https://nvd.nist.gov/vuln/detail/CVE-2025-45854]
```

Results

SYNACKTIV

- Comparing the built-in DB with quickhits.txt
  - Entries - Dicozorus: 22374 | quickhits.txt: 2567
  - Entries missing with a level equal to CRITICAL | HIGH | MEDIUM

```
$ dicozorus check -M -w quickhits.txt  -c CRITICAL,HIGH,MEDIUM | wc -l
1536
```

  - Some of the missing CRITICAL entries

```
_debug_toolbar/ [...] category: RCE, taglist: ['PYTHON'], reference: https://docs.pylonsproject.org/projects/pyramid_debugtoolbar/en/latest/]
_fragment [...] category: RCE, taglist: ['PHP'], reference: https://www.ambionics.io/blog/symfony-secret-fragment]
function/save [...] category: RCE, taglist: ['JAVA'], reference: https://nvd.nist.gov/vuln/detail/CVE-2024-0195]
jenkins/scriptText [...] category: RCE, taglist: ['JAVA'], reference: https://jenkins.io/doc/book/managing/script-console/]
ping.php [...] category: RCE, taglist: ['PHP'], reference: https://nvd.nist.gov/vuln/detail/CVE-2020-37123]
run [...] category: RCE, taglist: ['PYTHON'], reference: https://nvd.nist.gov/vuln/detail/CVE-2020-16846]
v1/tools/run [...] category: RCE, taglist: ['PYTHON'], reference: https://nvd.nist.gov/vuln/detail/CVE-2025-51482]
```

Results

SYNACKTIV

- Red Team initial entry points
  - **_debug_toolbar/** → RCE which led to further infrastructure compromise [D][Q][R]
  - **birt-viewer/document** → RCE which led to further infrastructure compromise [D][Q][R]
  - **app_dev.php** → Debugging interface that led to credential leak [R]
  - **c.php** → A webshell stored on a client internet-facing server [D][Q]
  - **script** → RCE through a Jenkins Groovy console with weak credentials [Q]
  - **xmlpserver/ReportTemplateService.xls** → XXE in Oracle Business Intelligence. Converted to RCE using another CVE [D][Q][R]
  - **Telerik.Web.UI.DialogHandler.aspx** → Arbitrary file uploads, found in 2 Red Teams[Q][R]
  - **invoker/readonly** → Pre-auth deserialization, led to further infrastructure compromise [D][Q][R]
- D = Dirsearch ; Q = Quickhits ; R = Raft
- **Update your wordlists!**

Conclusion

Conclusion

SYNACKTIV

- Dicozorus addresses a fundamental need in a pentester's arsenal: **maintaining wordlists**
- It has significantly contributed to improving our internal fuzzing processes
- Key Benefits:
  - Metadata enriched entries
  - Tailored wordlist generation
  - A built-in set of refined wordlists
  - Eases wordlist maintenance
- The tool is open-source, feel free to try it and give us some feedback
  - https://github.com/synacktiv/dicozorus
- Don't care about the tool / just want a nice wordlist
  - https://github.com/synacktiv/dicozorus/blob/main/lazy.txt

SYNACKTIV

https://github.com/synacktiv/dicozorus