

Firmwares are weird

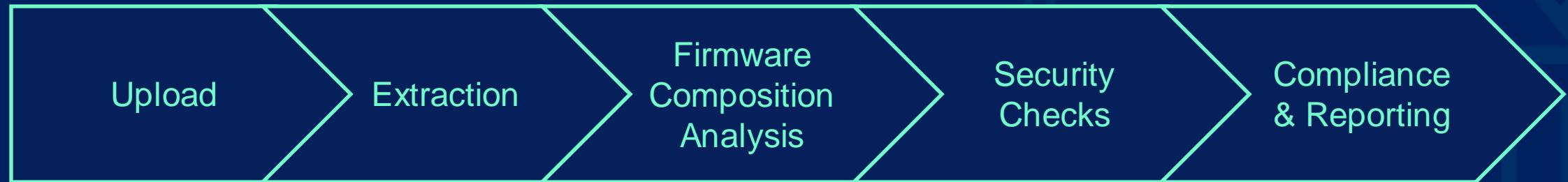
A year long journey to efficient extraction

IT-S Now Workshop
2023 June

Marton Illes
marton.illes@onekey.com

ONEKEY

NOT YOUR USUAL ANALYST WORKSTATION



NOT YOUR USUAL ANALYST WORKSTATION



NOT YOUR USUAL ANALYST WORKSTATION



NOT YOUR USUAL ANALYST WORKSTATION



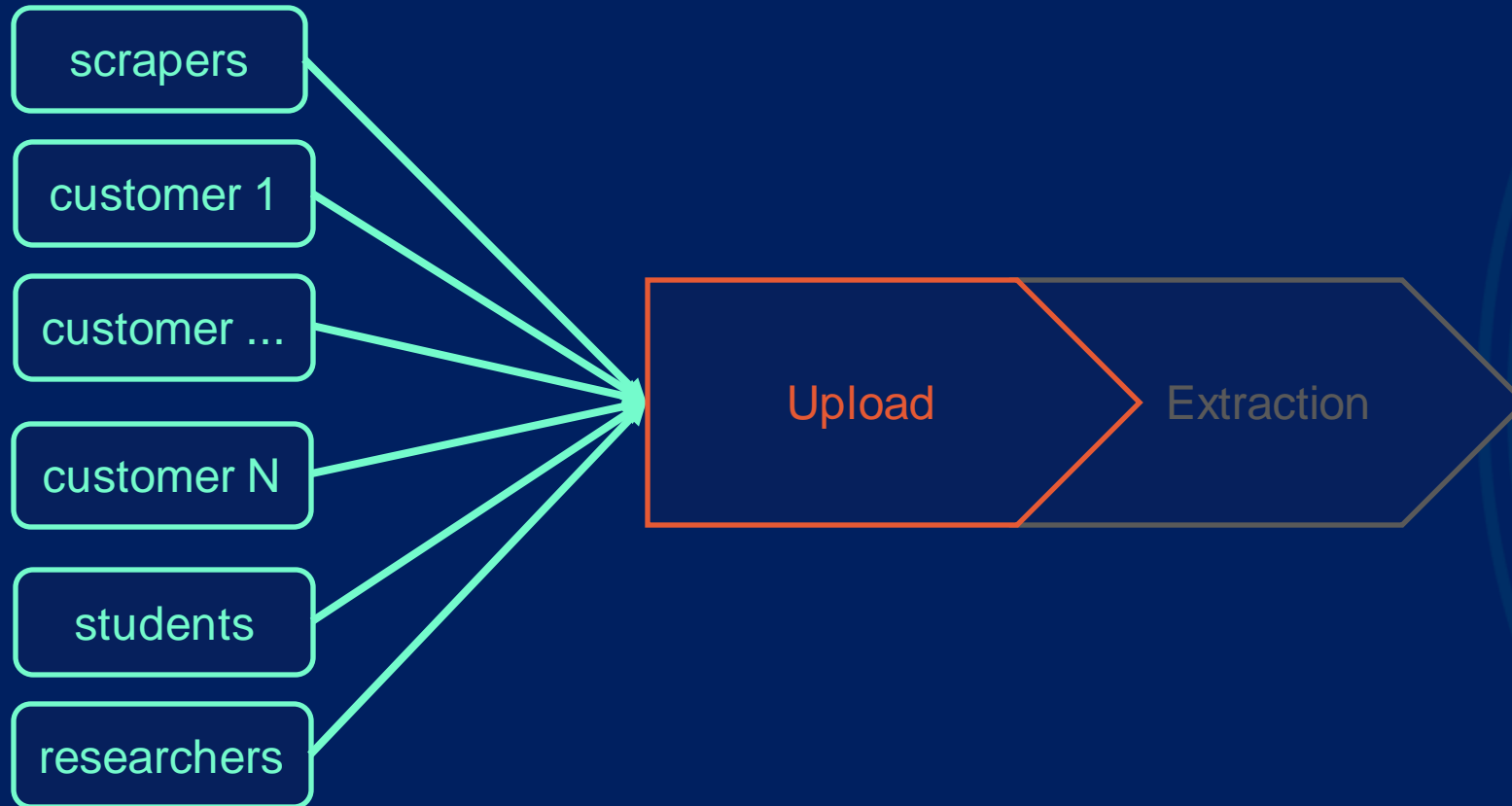
KEYSTONE

NOT YOUR USUAL ANALYST WORKSTATION



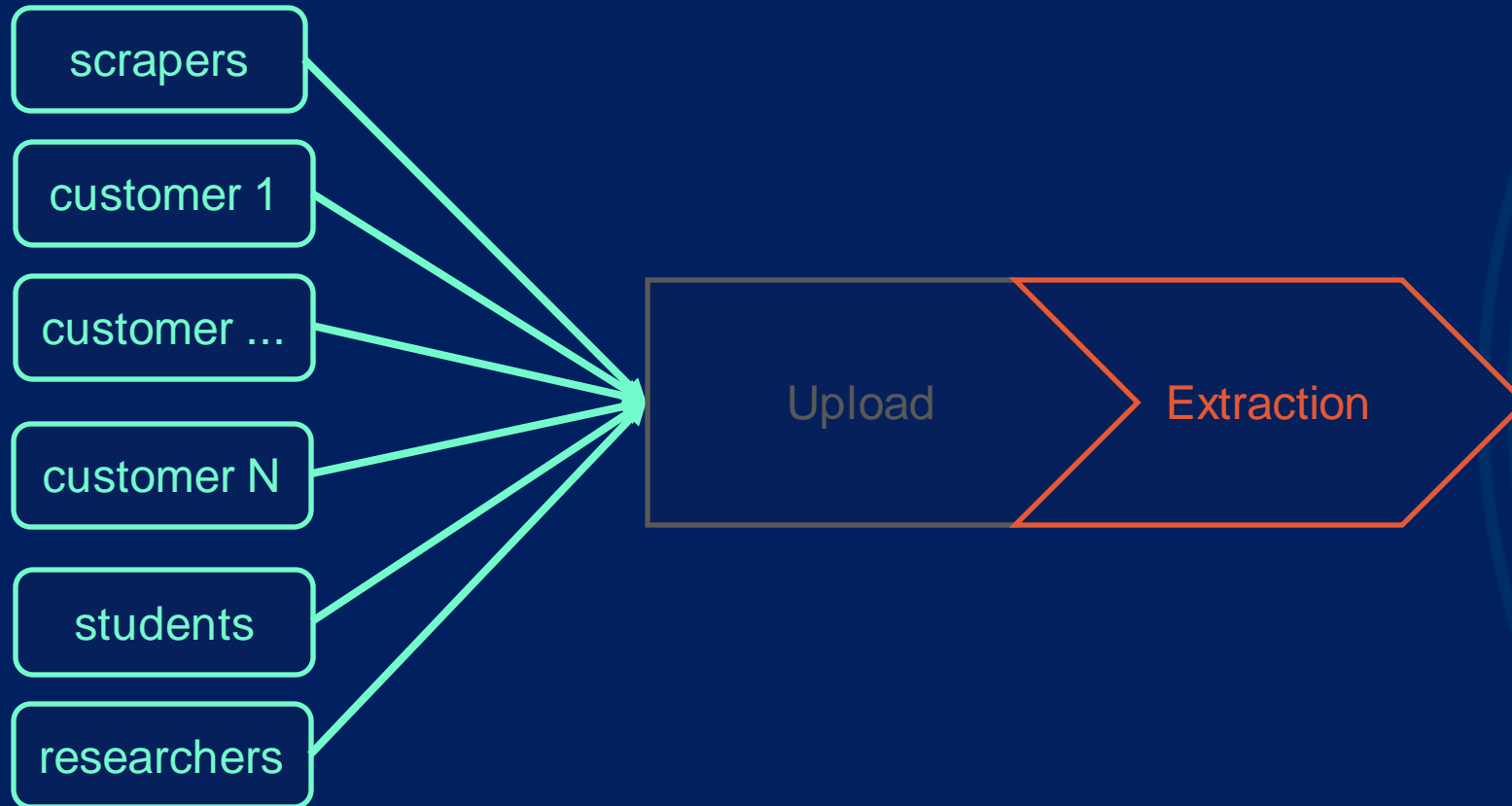
ONENET

NOT YOUR USUAL ANALYST WORKSTATION



- High amount of uploads
- Concurrency
- Untrusted users
- Source of uploads ?

NOT YOUR USUAL ANALYST WORKSTATION



- Timeouts
- Limited format support
- Memory footprint

We need to automatically extract firmwares of **arbitrary formats**, coming from **untrusted sources**, at scale.

Firmware **extraction** framework.

Parses unknown files, matching on more than 60 different **archive**, **compression**, and **file-system** formats.



<https://www.unblob.org>

Clear objectives:

- **Accuracy**
- **Security**
- **Extensibility**
- **Speed**



<https://www.unblob.org>

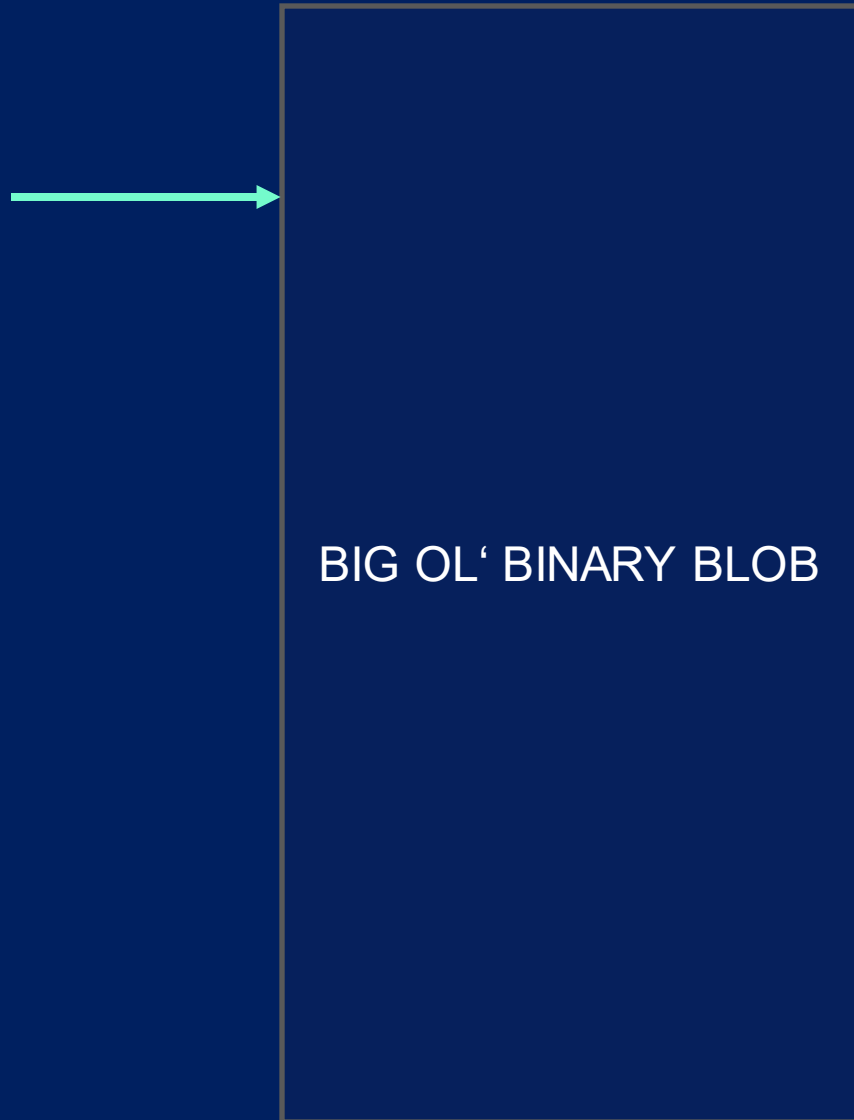
OBJECTIVE 1 : ACCURACY

BIG OL' BINARY BLOB

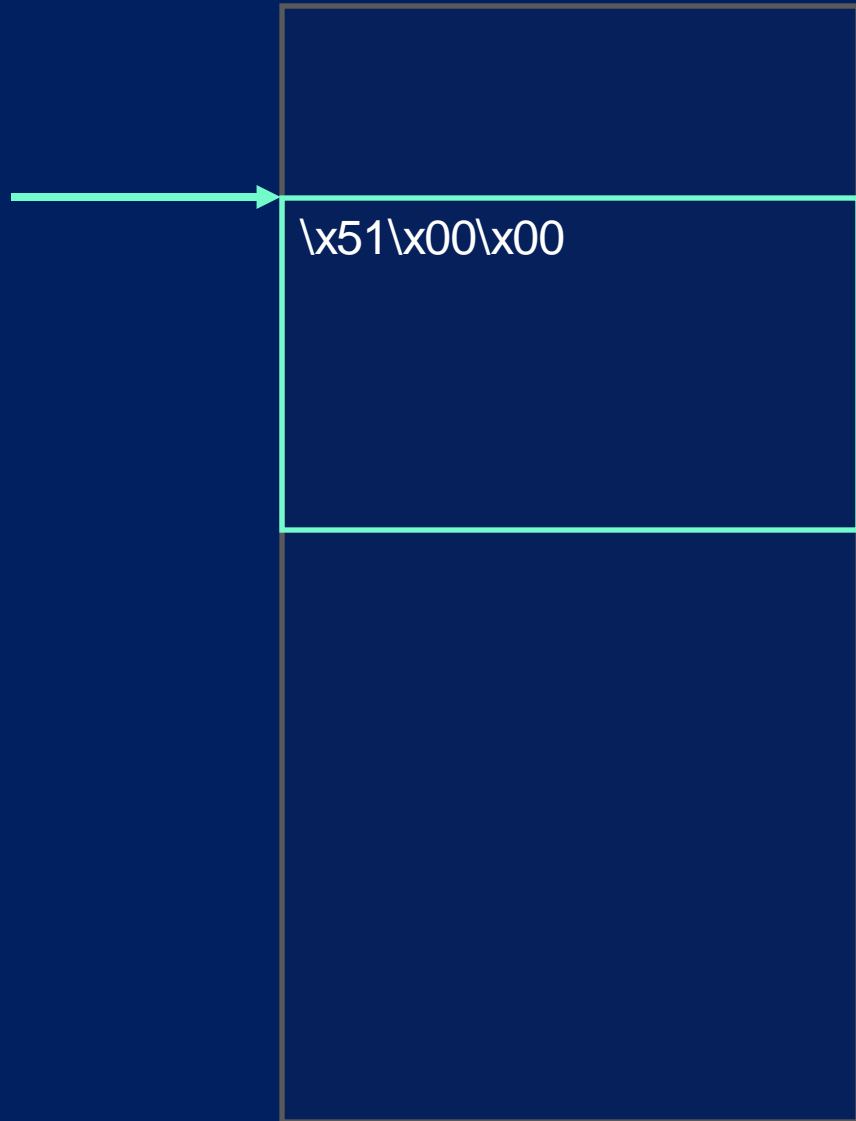
OBJECTIVE 1 : ACCURACY



OBJECTIVE 1 : ACCURACY



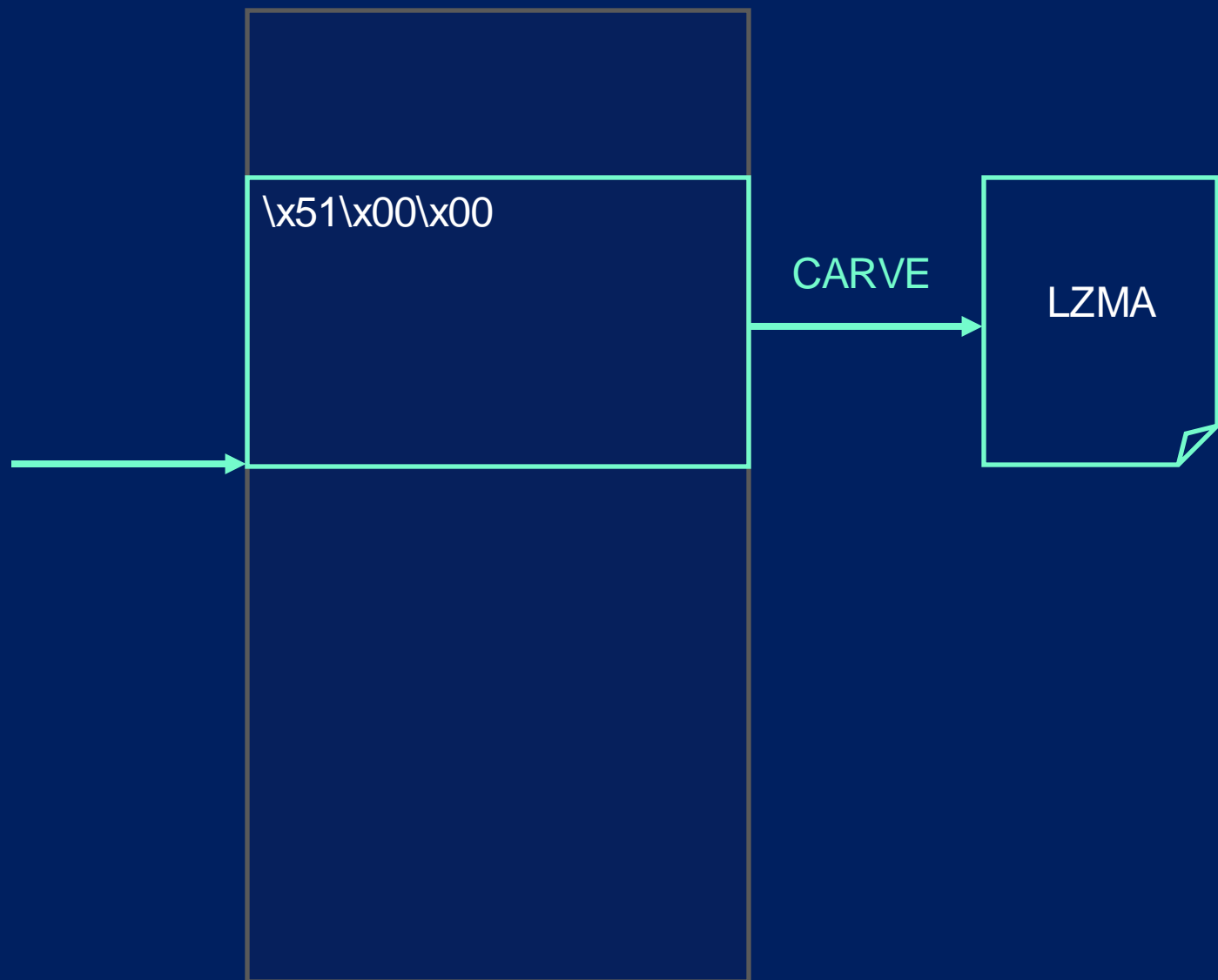
OBJECTIVE 1 : ACCURACY



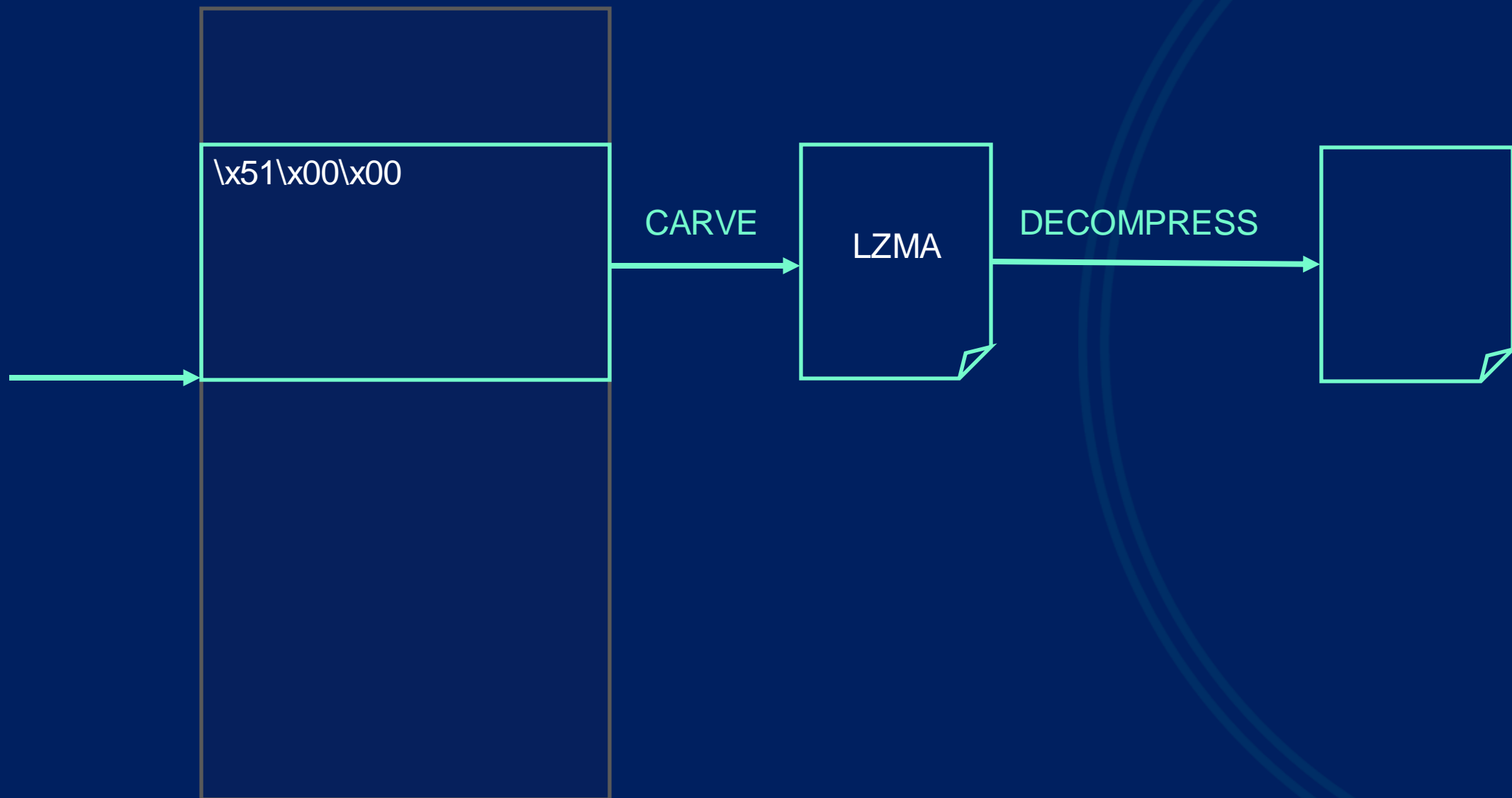
OBJECTIVE 1 : ACCURACY



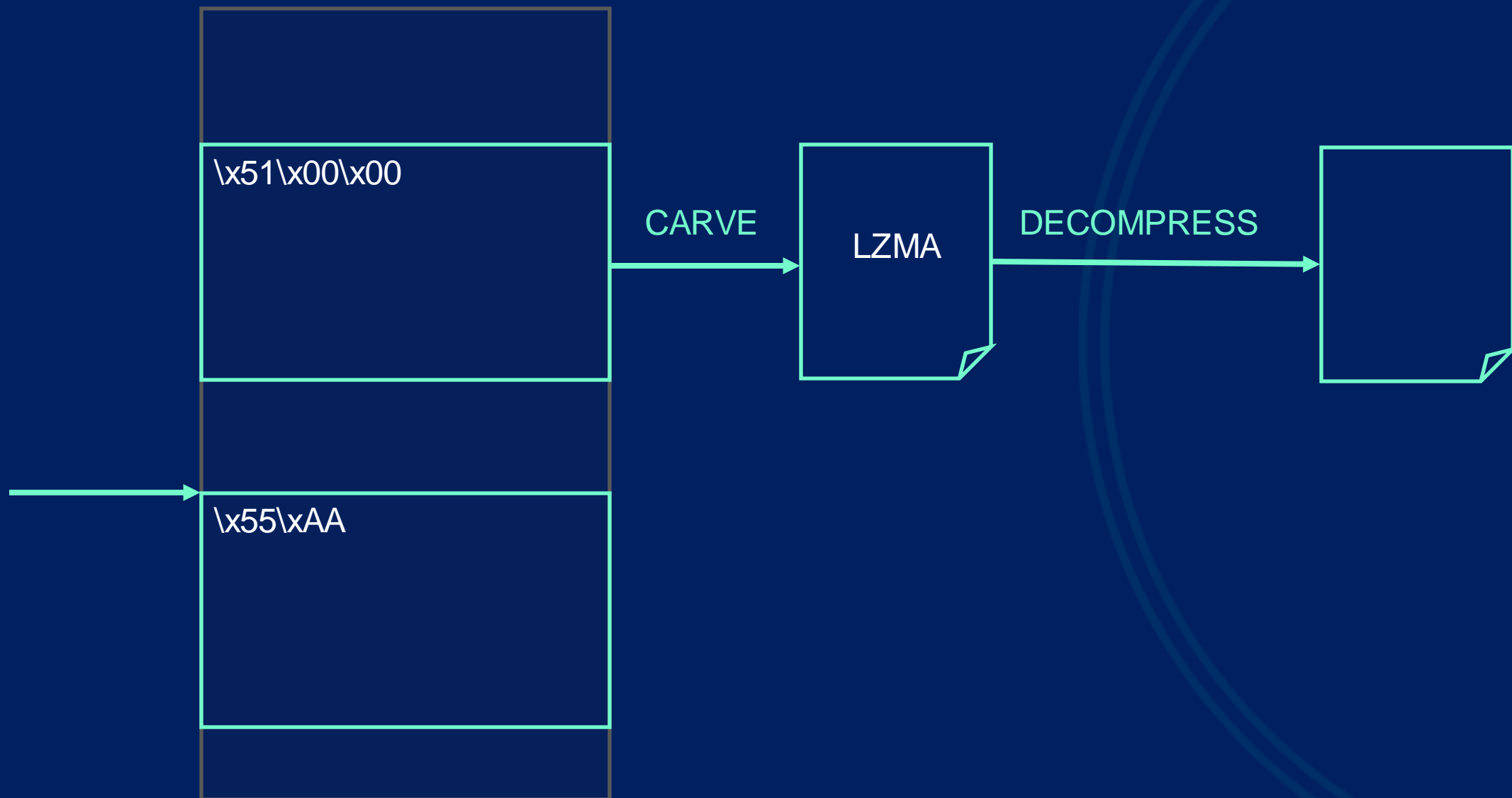
OBJECTIVE 1 : ACCURACY



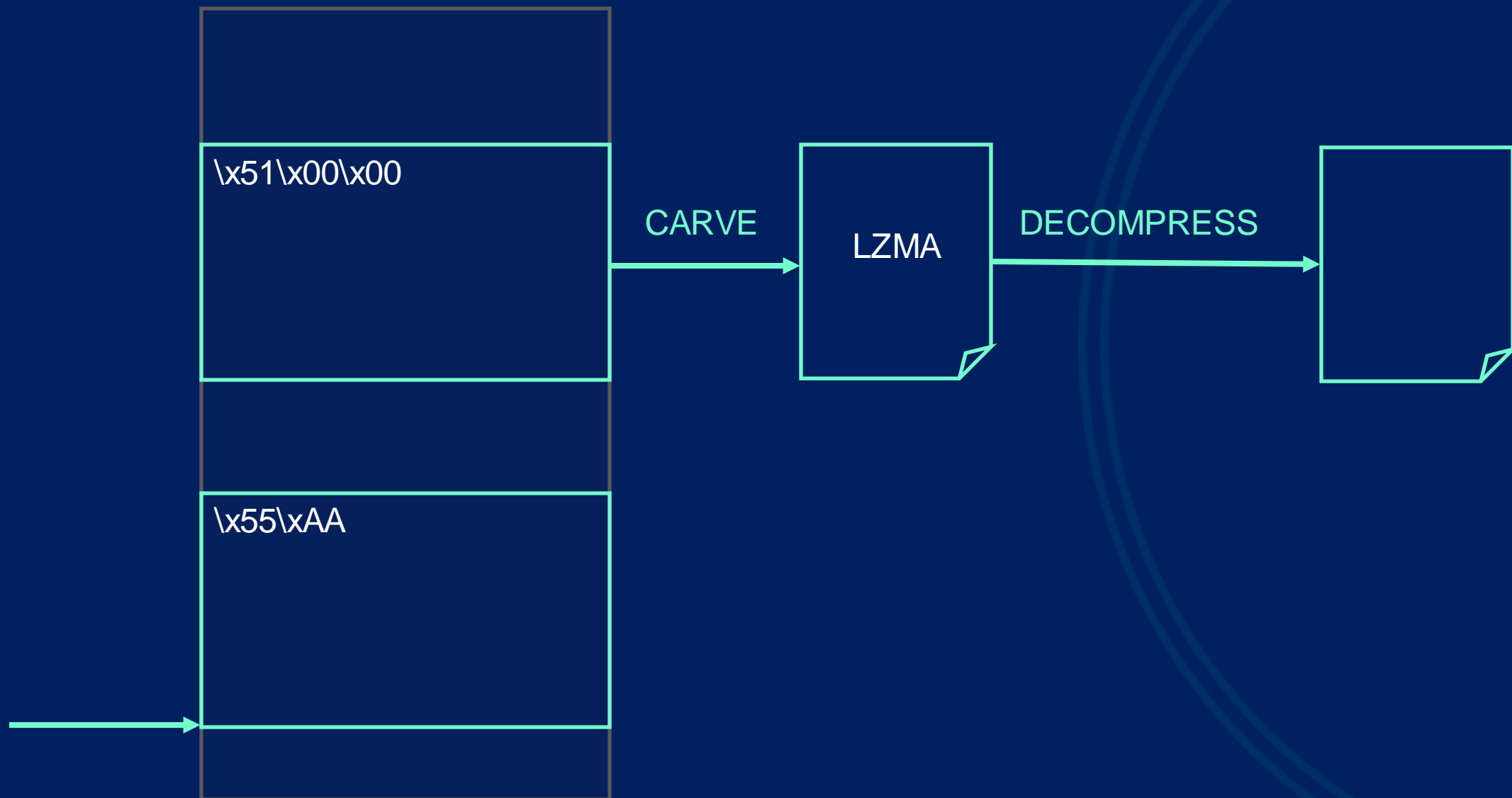
OBJECTIVE 1 : ACCURACY



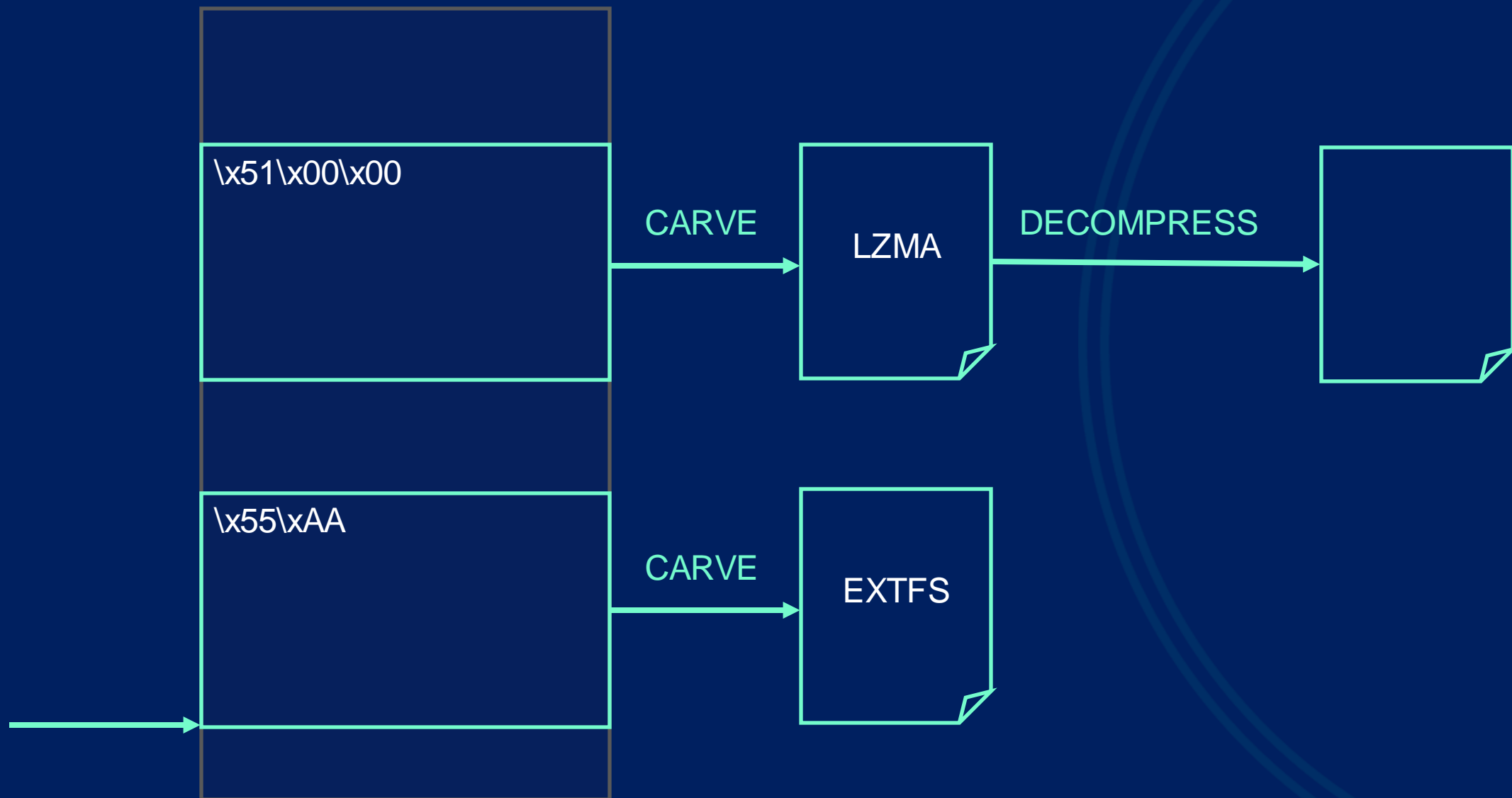
OBJECTIVE 1 : ACCURACY



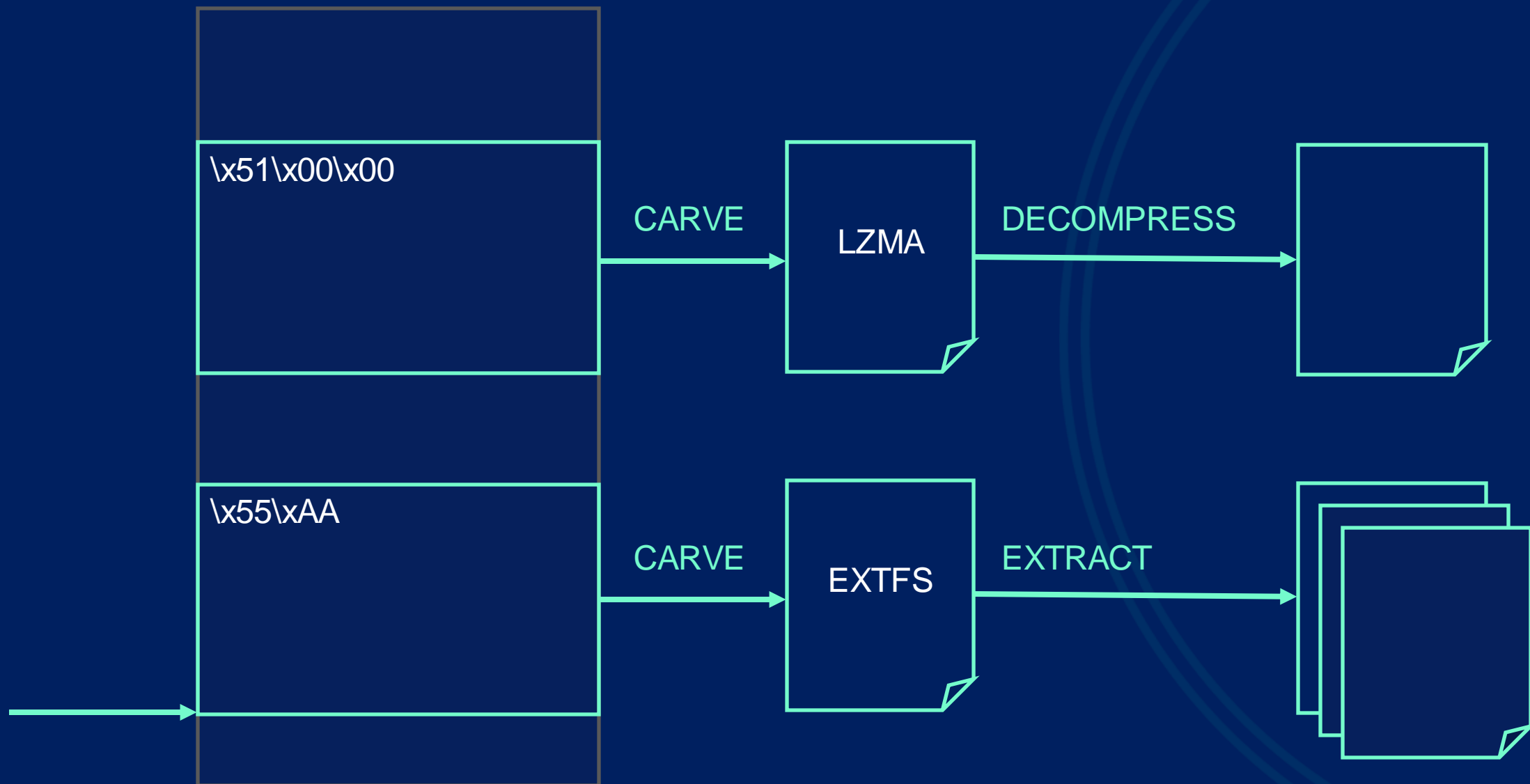
OBJECTIVE 1 : ACCURACY



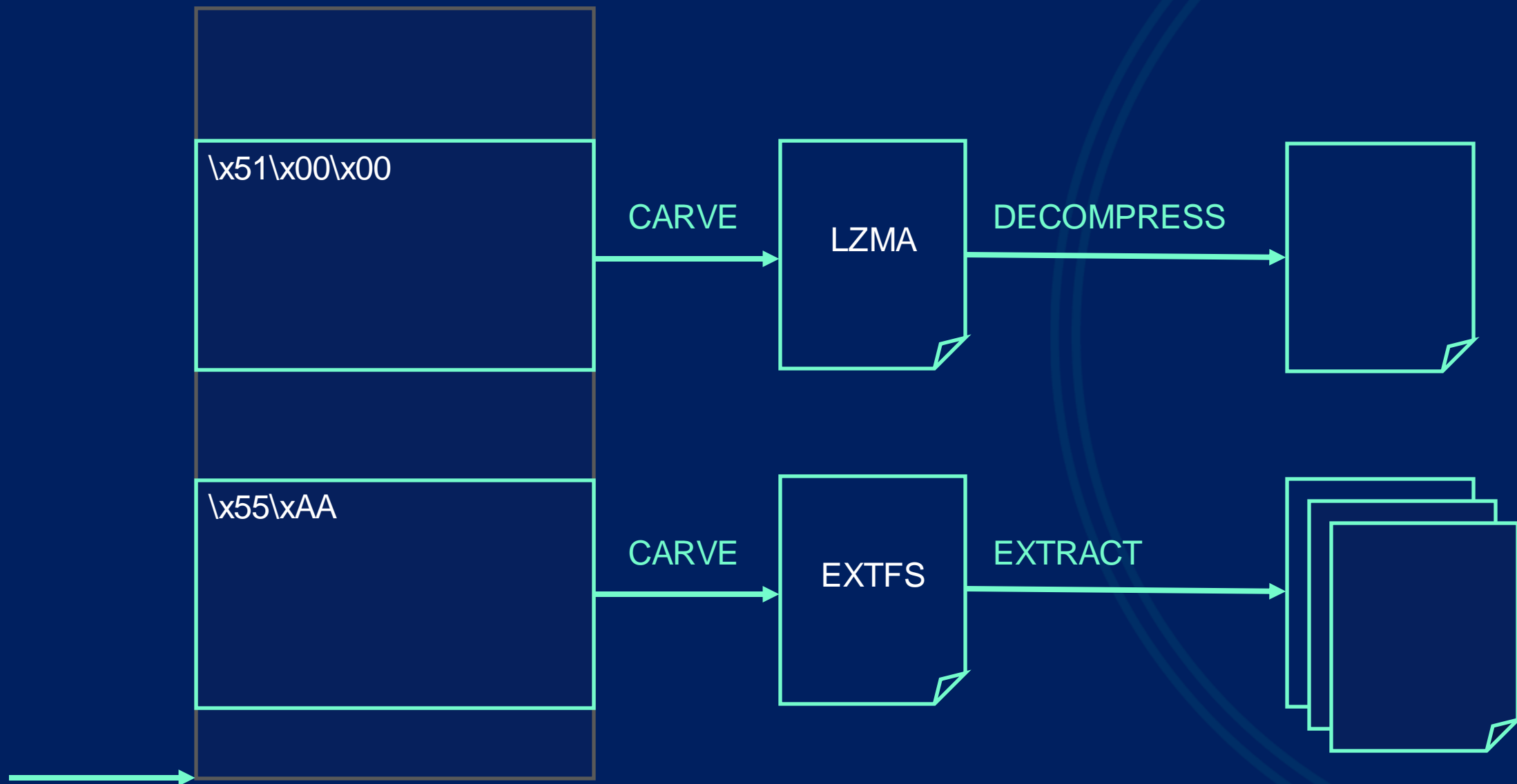
OBJECTIVE 1 : ACCURACY



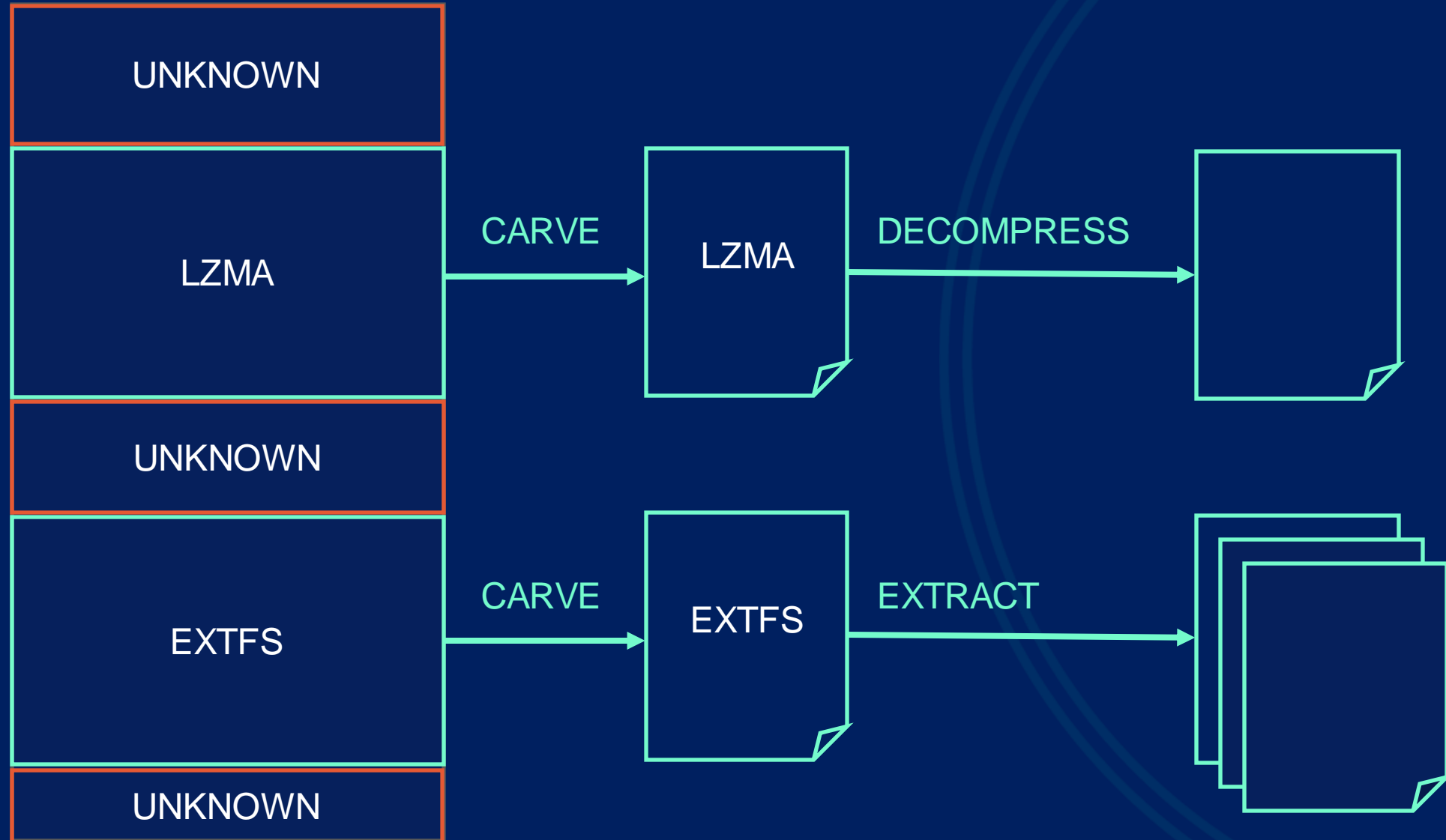
OBJECTIVE 1 : ACCURACY



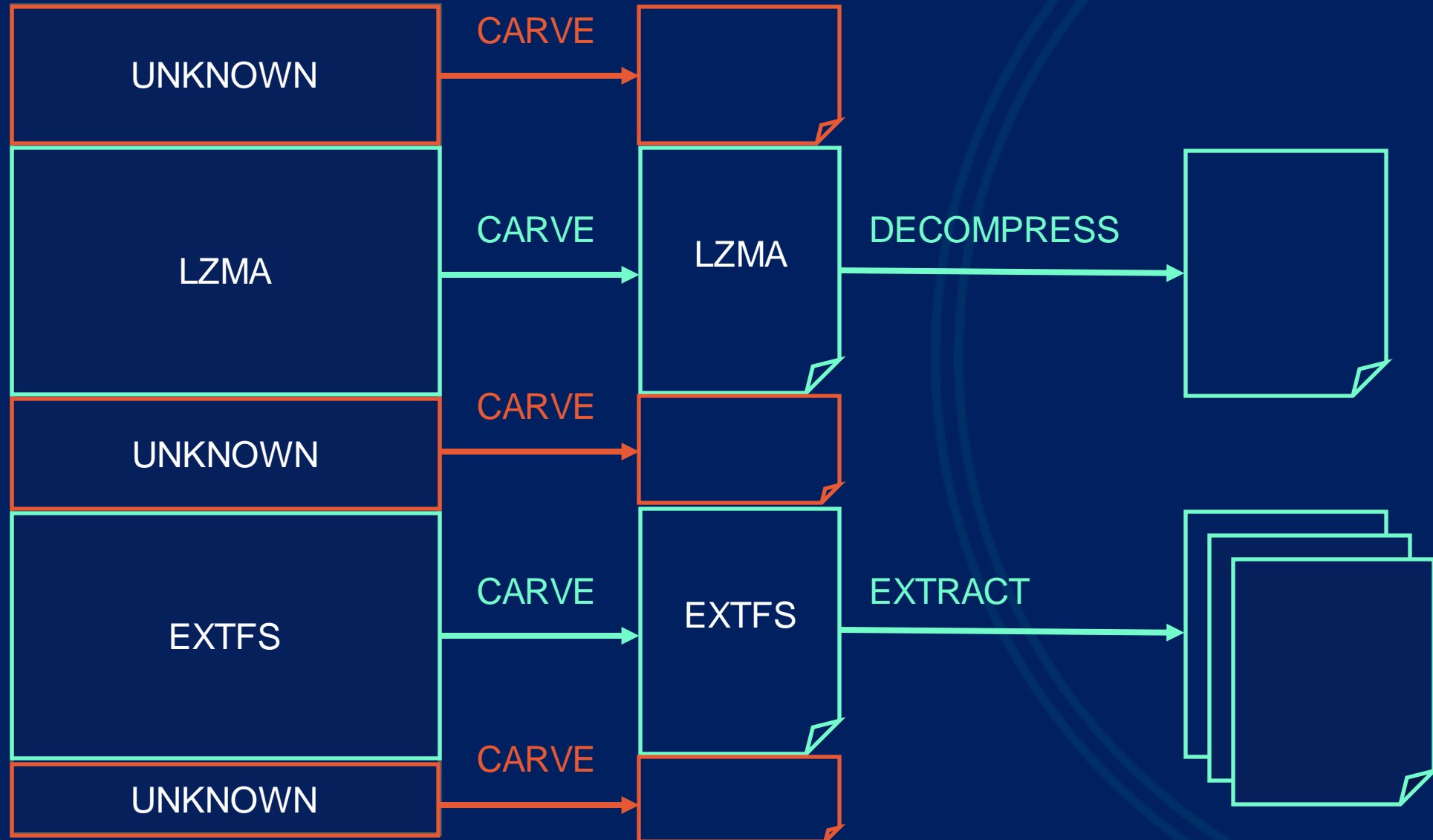
OBJECTIVE 1 : ACCURACY



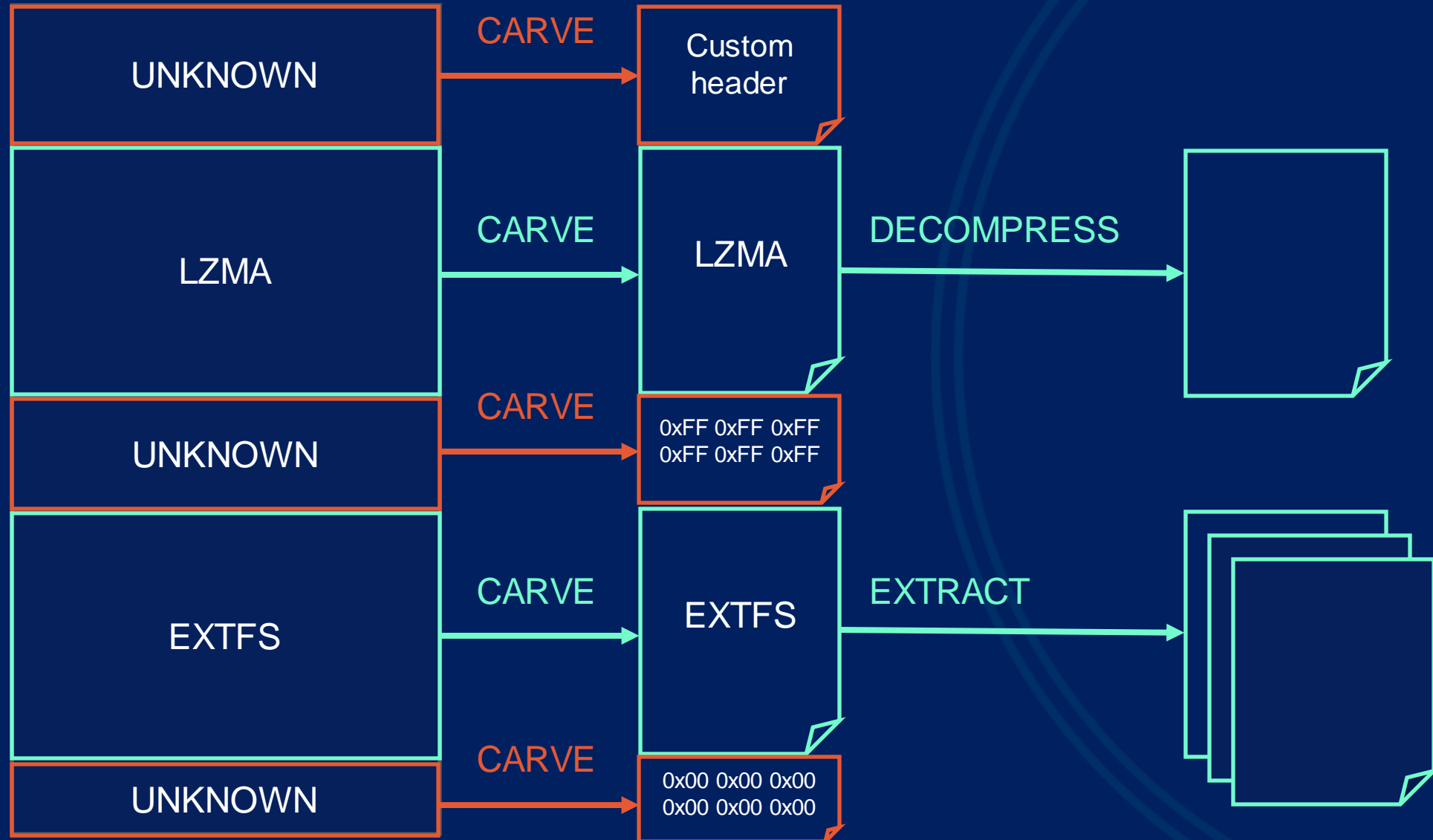
OBJECTIVE 1 : ACCURACY



OBJECTIVE 1 : ACCURACY



OBJECTIVE 1 : ACCURACY



BEING ACCURATE IN A WORLD OF EMBEDDED VENDORS WITH GREAT FIRMWARE IDEAS™

AKA HOW TO LOSE YOUR MIND IN 6 TO 8 MONTHS

THE WONDERFUL WORLD OF VENDOR FORMATS

	v2							
LE	hsqs							
BE	sqsh							

THE WONDERFUL WORLD OF VENDOR FORMATS

	v2	v3						
LE	hsqs	hsqs						
BE	sqsh	sqsh						

THE WONDERFUL WORLD OF VENDOR FORMATS

	v2	v3	v4					
LE	hsqs	hsqs	hsqs					
BE	sqsh	sqsh	sqsh					

THE WONDERFUL WORLD OF VENDOR FORMATS

	v2	v3	v4	DDWRT v3				
LE	hsqs	hsqs	hsqs	hsqt				
BE	sqsh	sqsh	sqsh	tqsh				

THE WONDERFUL WORLD OF VENDOR FORMATS

	v2	v3	v4	DDWRT v3	Broadcom v3			
LE	hsqs	hsqs	hsqs	hsqt	shsq			
BE	sqsh	sqsh	sqsh	tqsh	qshs			

THE WONDERFUL WORLD OF VENDOR FORMATS

	v2	v3	v4	DDWRT v3	Broadcom v3	??? v3		
LE	hsqs	hsqs	hsqs	hsqt	shsq	zlqs		
BE	sqsh	sqsh	sqsh	tqsh	qshs	sqlz		

THE WONDERFUL WORLD OF VENDOR FORMATS

	v2	v3	v4	DDWRT v3	Broadcom v3	??? v3	AVM	
LE	hsqs	hsqs	hsqs	hsqt	shsq	zlqs	Header in BE, but chunks in LE.	
BE	sqsh	sqsh	sqsh	tqsh	qshs	sqlz		

THE WONDERFUL WORLD OF VENDOR FORMATS

	v2	v3	v4	DDWRT v3	Broadcom v3	??? v3	AVM	Netgear
LE	hsqs	hsqs	hsqs	hsqt	shsq	zlqs	Header in BE, but chunks in LE.	Let's use a non standard LZMA + XZ compression !
BE	sqsh	sqsh	sqsh	tqsh	qshs	sqlz		

THE WONDERFUL WORLD OF VENDOR FORMATS

	v2	v3	v4	DDWRT v3	Broadcom v3	??? v3	AVM	Netgear
LE	hsqs	hsqs	hsqs	hsqt	shsq	zlqs	Header in BE, but chunks in LE.	Let's use a non standard LZMA + XZ compression !
BE	sqsh	sqsh	sqsh	tqsh	qshs	sqlz		

THE WONDERFUL WORLD OF VENDOR FORMATS

```
computed_crc=0x3af2f56e_header_crc=0x6ef5f23a
```



ONKEY

THE WONDERFUL WORLD OF VENDOR FORMATS

```
computed_crc=0x3af2f56e header_crc=0x6ef5f23a
```

```
commit 32320fdb88bc930c0b2bc2935ada8bff5fdc9042 (394-disable-cramfs-crc)
```

```
Author: Quentin Kaiser <quentin.kaiser@onekey.com>
```

```
Date: Thu Jun 9 16:17:55 2022 +0200
```

Improve cramfs CRC checking.

We observed a sample from Cisco where the CRC value was stored in a different endianness. This is non standard but we decided to support it anyway.

We also added a check for old cramfs format. If it's an old format, the CRC check always returns true because it's not supported.

More information about CRC validation and old format:

<https://github.com/secularbird/cramfs/blob/master/cramfsck.c>

OBJECTIVE 2 : SECURITY

Extraction Attacks	Stability
Privileges	Dependencies

OBJECTIVE 2 : SECURITY

Extraction Attacks	Stability
Privileges	Dependencies

KEYSTONE

OBJECTIVE 2 : SECURITY

Extraction Attacks	Stability
Privileges	Dependencies

KEYSTONE

OBJECTIVE 2 : SECURITY

Extraction Attacks	Stability
Privileges	Dependencies

OBJECTIVE 2 : SECURITY

Extraction Attacks	Stability
Privileges	Dependencies

KEYSTONE

OBJECTIVE 2 : SECURITY

Extraction Attacks
Stability
Dependencies
Privileges

Audited third party extraction tools we rely on.

Found **path traversals** vulnerabilities in:

- jefferson (fixed, took over maintenance)
- ubi-reader (fixed, took over maintenance)
- Yaffshiv (fixed in our fork, reimplemented yaffs 1 & 2 in unblob)
- binwalk (fixed)

OBJECTIVE 2 : SECURITY

```
▼ ↕ 25 ■■■■ src/scripts/jefferson 📄  
54 + def is_safe_path(basedir, path, follow_symlinks=True):  
55 +     if follow_symlinks:  
56 +         matchpath = os.path.realpath(path)  
57 +     else:  
58 +         matchpath = os.path.abspath(path)  
59 +     return basedir == os.path.commonpath((basedir, matchpath))  
  
55     60  
56     61     cstruct.typedef("uint8", "uint8_t")  
57     62     cstruct.typedef("uint16", "jint16_t")  
  
⋮  
⋮  
⋮  
@@ -474,20 +479,32 @@ def dump_fs(fs, target):  
474     479         node_names.append(dirent.name.decode())  
475     480         path = "/".join(node_names)  
476     481  
477     -         target_path = os.path.join(os.getcwd(), target, path)  
482     +         target_path = os.path.realpath(os.path.join(os.getcwd(), target, path))  
483     +  
484     +         if not is_safe_path(target, target_path):  
485     +             print(f"Path traversal attempt to {target_path}, discarding.")  
486     +             continue  
487     +
```

OBJECTIVE 2 : SECURITY

```
ubireader/ubifs/output.py

29 + def is_safe_path(basedir, path, follow_symlinks=True):
30 +     if follow_symlinks:
31 +         matchpath = os.path.realpath(path)
32 +     else:
33 +         matchpath = os.path.abspath(path)
34 +     return basedir == os.path.commonpath((basedir, matchpath))

29 35
30 36     def extract_files(ubifs, out_path, perms=False):
31 37         """Extract UBIFS contents to_path/

@@ -60,7 +66,11 @@ def extract_dents(ubifs, inodes, dent_node, path='', perms=False):

60 66
61 67     inode = inodes[dent_node.inum]
62 68     dent_path = os.path.join(path, dent_node.name)

63 -

69 +
70 +     if not is_safe_path(path, dent_path):
71 +         log(extract_dents, 'path traversal attempt: %s, discarding' % (dent_path))
72 +     return
73 +
```

OBJECTIVE 2 : SECURITY

```
src/yaffshiv
8      13      class Compat(object):
9      14          ...
10     15      Python2/3 compatability methods.

@@ -607,13 +612,14 @@ class YAFFSExtractor(YAFFS):
607    612          for (entry_id, file_path) in Compat.iterator(self.file_paths):
608    613              entry = self.file_entries[entry_id]
609    614              if file_path and int(entry.yaffs_obj_type) == self.YAFFS_OBJECT_TYPE_DIRECTORY:
615    615          +              file_path = os.path.join(outdir, file_path)
616    616          +
610    617              # Check the file name for possible path traversal attacks
611    617              -              if b'..' in file_path:
618    618          +              if not is_safe_path(outdir, file_path):
612    619                  sys.stderr.write("Warning: Refusing to create directory '%s': possible path traversal\n" % file_path)
613    620                  continue
614    621
615    622          +          try:
616    622          -              file_path = os.path.join(outdir, file_path)
617    623              os.makedirs(file_path)
618    624              self._set_mode_owner(file_path, entry)
619    625              dir_count += 1
```

OBJECTIVE 2 : SECURITY



Quentin Kaiser Oct 17th at 6:46 AM

Only real python developers can identify the bug in this code. Will you find it ?

IMG_1896 ▼

```
try:
    with PFS(fname) as fs:
        # The end of PFS meta data is the start of the actual data
        data = binwalk.core.common.BlockFile(fname, 'rb')
        data.seek(fs.get_end_of_meta_data())
        for entry in fs.entries():
            outfile_path = os.path.join(out_dir, entry.fname)
            if not outfile_path.startswith(out_dir):
                binwalk.core.common.warning("Urpfs extractor detected directory tra
            else:
                self._create_dir_from_fname(outfile_path)
                outfile = binwalk.core.common.BlockFile(outfile_path, 'wb')
                outfile.write(data.read(entry.fsize))
                outfile.close()

        data.close()
except KeyboardInterrupt as e:
    raise e
except Exception as e:
```



KEYWORD

OBJECTIVE 2 : SECURITY

```
try:
    with PFS(fname) as fs:
        # The end of PFS meta data is the start of the actual data
        data = binwalk.core.common.BlockFile(fname, 'rb')
        data.seek(fs.get_end_of_meta_data())
        for entry in fs.entries():
            outfile_path = os.path.join(out_dir, entry.fname)
            if not outfile_path.startswith(out_dir):
                binwalk.core.common.warning("Unpfs extractor detected directory traversal")
            else:
                self._create_dir_from_fname(outfile_path)
                outfile = binwalk.core.common.BlockFile(outfile_path, 'wb')
                outfile.write(data.read(entry.fsize))
                outfile.close()

        data.close()
except KeyboardInterrupt as e:
    raise e
except Exception as e:
    return False
```

OBJECTIVE 2 : SECURITY

```
src/binwalk/plugins/unpfs.py
@@ -104,7 +104,7 @@ def extractor(self, fname):
    data = binwalk.core.common.BlockFile(fname, 'rb')
    data.seek(fs.get_end_of_meta_data())
    for entry in fs.entries():
-       outfile_path = os.path.join(out_dir, entry.fname)
+       outfile_path = os.path.abspath(os.path.join(out_dir, entry.fname))
    if not outfile_path.startswith(out_dir):
        binwalk.core.common.warning("Unpfs extractor detected directory traversal
    else:
```

OBJECTIVE 2 : SECURITY



Quentin Kaiser

@qkaiser



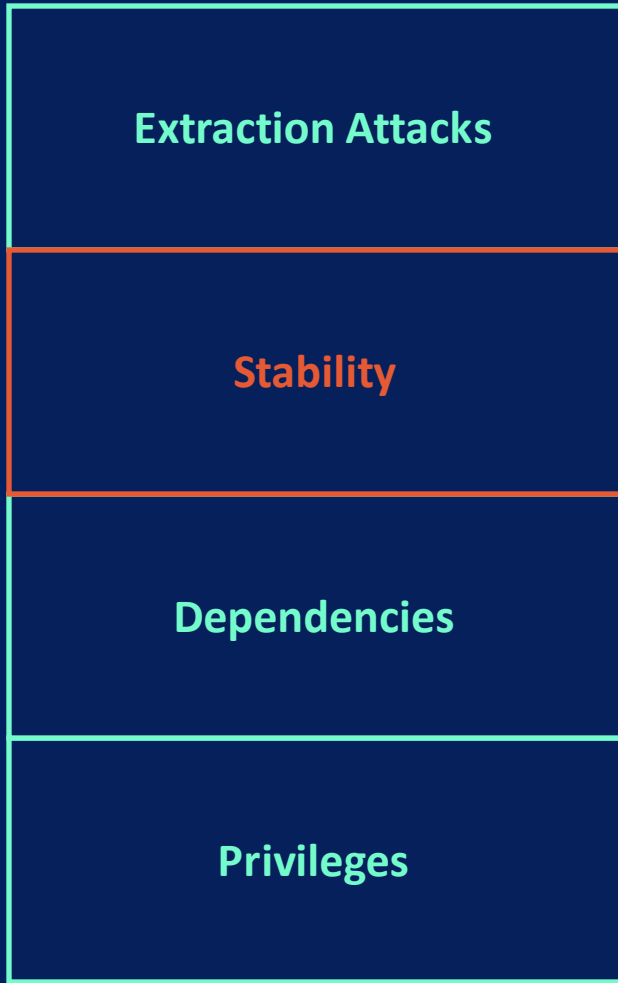
If I got my ICC embedding right, this file should be a zero day.



7:24 PM · Nov 1, 2022 · Twitter Web App

1 Retweet 1 Quote Tweet 3 Likes

OBJECTIVE 2 : SECURITY



- Fuzz testing of unblob code base.
- Found and fixed 19 bugs so far.
- Logic bugs on malformed / corrupted files mostly.

KEYSTONE

OBJECTIVE 2 : SECURITY

Extraction Attacks

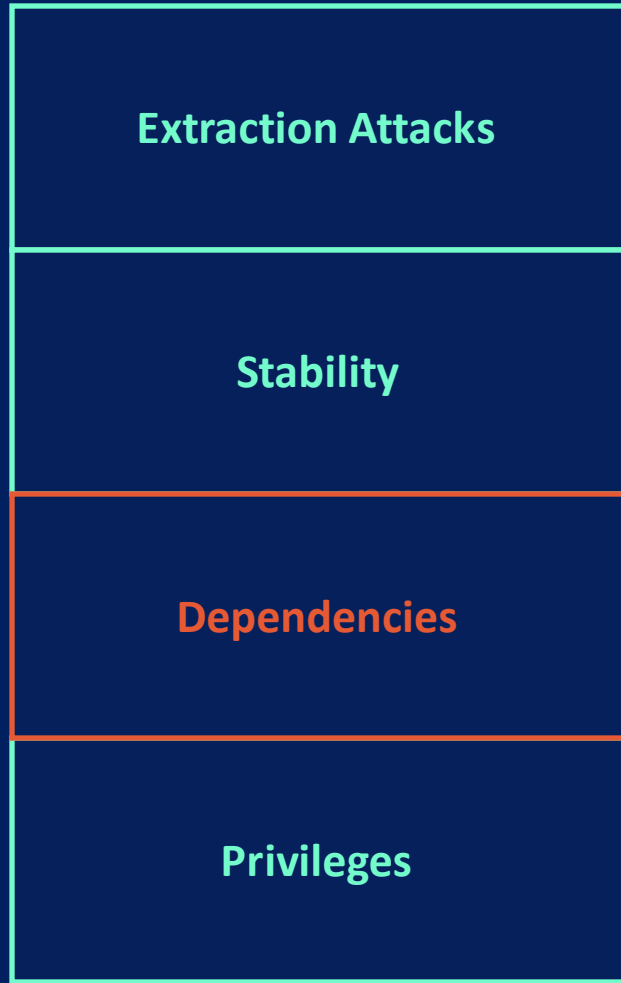
Stability

Dependencies

Privileges

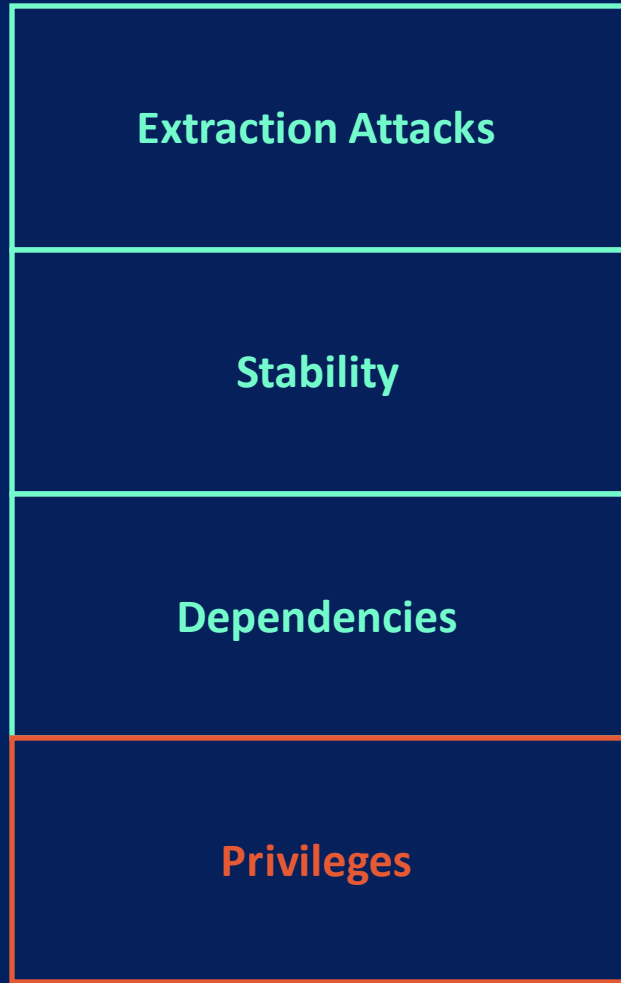
<input type="checkbox"/>	0 Open	✓ 19 Closed	Author	Label	Projects	Milestones	Assignee	Sort
<input type="checkbox"/>	✓	infinite loop in zlib handler makes unblob hang	bug	format:compression	fuzzing	1		
		#441 by QKaiser was closed on Sep 27						
<input type="checkbox"/>	✓	Unhandled zlib error in gzip handler when parsing corrupted gzip files	bug	format:compression	fuzzing	1		
		#200 by QKaiser was closed on Jan 27 ↻ v1.0 - extraction						
<input type="checkbox"/>	✓	Uncaught PermissionError in unblob on malformed JFFS2 filesystems	bug	format:filesystem	fuzzing	1		4
		#190 by QKaiser was closed on Jan 28 ↻ v1.0 - extraction						
<input type="checkbox"/>	✓	Uncaught ValueError in rar file handler on malformed rar archives	bug	format:archive	fuzzing	1		
		#189 by QKaiser was closed on Jan 26 ↻ v1.0 - extraction						
<input type="checkbox"/>	✓	Uncaught ReadError in tar file handler on malformed tar archive	bug	format:archive	fuzzing	1		
		#188 by QKaiser was closed on Jan 25 ↻ v1.0 - extraction						
<input type="checkbox"/>	✓	Uncaught InvalidHeaderError in tar file handler on malformed tar files	bug	format:archive	fuzzing	1		
		#187 by QKaiser was closed on Jan 25 ↻ v1.0 - extraction						
<input type="checkbox"/>	✓	Uncaught UnicodeDecodeError in unblob extraction on malformed CAB files	bug	format:archive	fuzzing	1		
		#186 by QKaiser was closed on Jan 28 ↻ v1.0 - extraction						
<input type="checkbox"/>	✓	Negative seek in ZIP handler on malformed file	bug	format:archive	fuzzing	1		
		#185 by QKaiser was closed on Jan 26 ↻ v1.0 - extraction						
<input type="checkbox"/>	✓	Unhandled UnicodeEncodeError in unblob through corrupted CAB files	bug	format:archive	fuzzing	1		1
		#184 by QKaiser was closed on Jan 28 ↻ v1.0 - extraction						
<input type="checkbox"/>	✓	Unhandled BadZipFile exception in ZIP handler when parsing corrupted ZIP files	bug	format:archive	fuzzing	1		
		#183 by QKaiser was closed on Jan 26 ↻ v1.0 - extraction						
<input type="checkbox"/>	✓	Unhandled UnicodeDecodeError in ZIP handler when parsing corrupted ZIP files	bug	format:archive	fuzzing	1		
		#182 by QKaiser was closed on Jan 26 ↻ v1.0 - extraction						
<input type="checkbox"/>	✓	LZ4 triggers ValueError on malformed files	bug	format:compression	fuzzing	2		
		#177 by QKaiser was closed on Jan 27 ↻ v1.0 - extraction						

OBJECTIVE 2 : SECURITY



- All our dependencies are documented and **locked** with poetry.
- Even better if you use our Nix build.

OBJECTIVE 2 : SECURITY



- No elevated privileges required.
- Opinionated decision, we can't create special files.

KEYNOTE

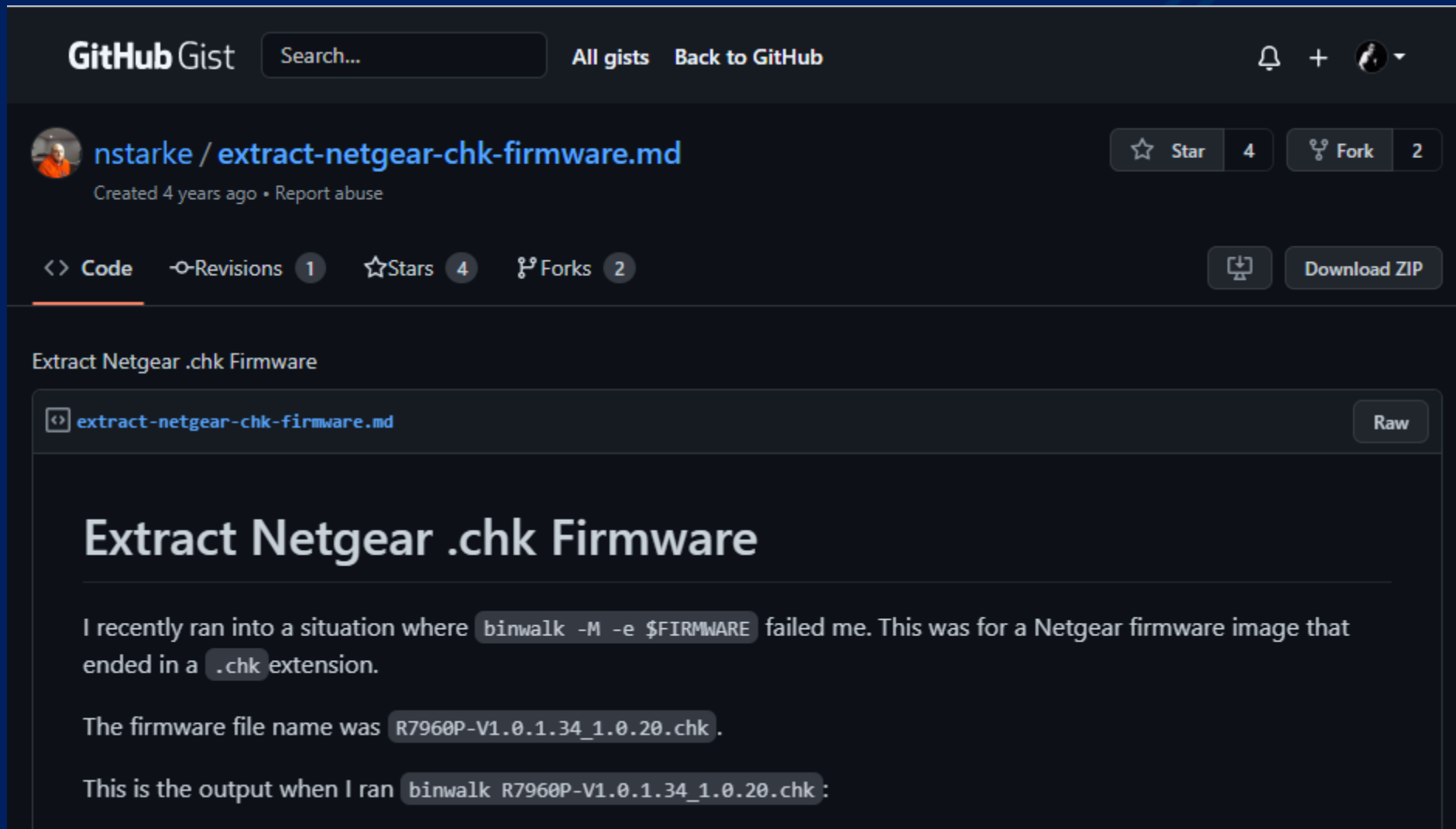
OBJECTIVE 3 : EXTENSIBILITY

A Brief History of Metasploit

Metasploit was originally developed and conceived by HD Moore while he was employed by a security firm. When HD realized that he was spending most of his time validating and sanitizing public exploit code, he began to create a flexible and maintainable framework for the creation and development of exploits. He released his first edition of the Perl-based Metasploit in October 2003 with a total of 11 exploits.

With the help of Spoonm, HD released a total rewrite of the project, Metasploit 2.0, in April 2004. This version included 19 exploits and over 27 payloads. Shortly after this release, Matt Miller (Skape) joined the Metasploit development team, and as the project gained popularity, the Metasploit Framework received ...

OBJECTIVE 3 : EXTENSIBILITY



The screenshot shows a GitHub Gist page for a file named 'extract-netgear-chk-firmware.md' by user 'nstarke'. The page has a dark theme. At the top, there is a search bar and navigation links for 'All gists' and 'Back to GitHub'. The user's profile picture and name are visible, along with the file name and creation date ('Created 4 years ago'). There are 4 stars and 2 forks. Below the file name, there are tabs for 'Code', 'Revisions', 'Stars', and 'Forks'. A 'Download ZIP' button is also present. The main content area shows the title 'Extract Netgear .chk Firmware' and the text of the gist, which describes a problem with the 'binwalk' tool and provides a solution for extracting Netgear firmware with a '.chk' extension.

GitHub Gist Search... All gists Back to GitHub

nstarke / [extract-netgear-chk-firmware.md](#) Star 4 Fork 2

Created 4 years ago • Report abuse

<> Code Revisions 1 Stars 4 Forks 2 Download ZIP

Extract Netgear .chk Firmware

`extract-netgear-chk-firmware.md` Raw

Extract Netgear .chk Firmware

I recently ran into a situation where `binwalk -M -e $FIRMWARE` failed me. This was for a Netgear firmware image that ended in a `.chk` extension.

The firmware file name was `R7960P-V1.0.1.34_1.0.20.chk`.

This is the output when I ran `binwalk R7960P-V1.0.1.34_1.0.20.chk`:

OBJECTIVE 3 : EXTENSIBILITY

The screenshot shows a GitHub repository page for 'HaToan / Decrypt-Firmware-Hikvision'. The repository is public and has 1 watch, 2 forks, and 7 stars. The main navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Security, and Insights. The repository is currently on the 'master' branch. The file list shows three files: README.md, dec_hik.py, and digicap.dav, all uploaded 14 months ago. The README.md file is selected, showing the title 'Decrypt Firmware Hikvision' and the subtitle 'Decrypt E3S'. The right sidebar contains an 'About' section with no description, a 'Releases' section with no published releases, and a 'Packages' section.

HaToan / Decrypt-Firmware-Hikvision Public Watch 1 Fork 2 Star 7

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#)

master Go to file Add file Code About

HaToan Decrypt firmware E3S ... on Sep 24, 2021 🕒 1

README.md	Decrypt firmware E3S	14 months ago
dec_hik.py	Decrypt firmware E3S	14 months ago
digicap.dav	Decrypt firmware E3S	14 months ago

☰ README.md

Decrypt Firmware Hikvision

Decrypt E3S

About
No description, website, or topics provided.

- Readme
- 7 stars
- 1 watching
- 2 forks

Releases
No releases published

Packages

OBJECTIVE 3 : EXTENSIBILITY

The screenshot shows the GitHub interface for the repository 'krolinventions / draytools', which is a public archive. At the top, there are buttons for 'Watch' (7), 'Fork' (53), and 'Star' (55). Below this is a navigation bar with links for 'Code', 'Issues' (3), 'Pull requests' (1), 'Actions', 'Projects', 'Wiki', 'Security', and 'Insights'. The main content area features a 'master' branch selector, a 'Go to file' button, and a 'Code' button. A commit history table is displayed, listing files and their commit messages. To the right, there is an 'About' section with a description: 'DrayTek Vigor password recovery, config & firmware tools - NOT MAINTAINED, CHECK'. Below this are links to the repository URL, 'Readme', 'GPL-3.0 license', '55 stars', '7 watching', and '53 forks'. At the bottom right, there is a 'Releases' section with '7 tags'.

krolinventions / draytools Public archive

Watch 7 Fork 53 Star 55

Code Issues 3 Pull requests 1 Actions Projects Wiki Security Insights

master

Go to file Code

About

DrayTek Vigor password recovery, config & firmware tools - NOT MAINTAINED, CHECK

github.com/gerard-/draytools/net...

Readme

GPL-3.0 license

55 stars

7 watching

53 forks

Releases

7 tags

File	Commit Message	Time
CHANGELOG	Fixed some null-terminated credentials (rare...	11 years ago
COPYING	first public release	11 years ago
CREDITS	refactoring & new features, now v0.3	11 years ago
INSTALL	refactoring & new features, now v0.3	11 years ago
README	added explicit "admin" username for master ...	11 years ago
VERSION	Fixed some null-terminated credentials (rare...	11 years ago
draytools.py	Improve bruteforcing by not relying on smar...	11 years ago
pydelzo.py	cfg now ok for 2700 and 2800	11 years ago

README

OBJECTIVE 3 : EXTENSIBILITY

The screenshot shows a GitHub repository page for 'yath/vigor165'. The repository is public and has 2 watchers, 2 forks, and 3 stars. The main branch is 'main'. The repository contains a README.md file and a commit history showing a commit by 'yath' on Oct 30, 2020, with 8 commits. The commit history includes folders 'decompress' and 'dump', and a file 'README.md'. The README.md file is titled 'Vigor165 notes' and contains the text: 'This repository collects some notes on my reverse engineering efforts on the DrayTek Vigor 165.' The repository is categorized as 'DrayTek Vigor165 stuff' and has 3 stars, 2 watchers, and 2 forks. The languages used in the repository are Go (47.3%), Assembly (33.8%), C (17.0%), and Makefile (1.9%).

Repository Information:

- Repository: [yath/vigor165](#) (Public)
- Watch: 2
- Fork: 2
- Star: 3

Navigation:

- Code
- Issues: 1
- Pull requests
- Actions
- Projects
- Security
- Insights

Repository Content:

- main
- Go to file
- Add file
- Code

Commit History:

Commit	Message	Time
yath	Write compressed sections as individual files ...	on Oct 30, 2020 8
decompress	Write compressed sections as individual files	2 years ago
dump	Add old memory dumping code	2 years ago
README.md	Add notes	2 years ago

Repository Details:

- About
- DrayTek Vigor165 stuff
- Readme
- 3 stars
- 2 watching
- 2 forks

Languages:

- Go 47.3%
- Assembly 33.8%
- C 17.0%
- Makefile 1.9%

README.md:

Vigor165 notes

This repository collects some notes on my reverse engineering efforts on the DrayTek Vigor 165.

OBJECTIVE 3 : EXTENSIBILITY

The screenshot shows a GitHub repository page for 'synacktiv/yealink_tools'. At the top, it indicates the repository is 'Public' and has 6 watchers, 5 forks, and 10 stars. The navigation bar includes links for Code, Issues (1), Pull requests, Actions, Projects, Security, and Insights. Below the navigation, there are buttons for 'Go to file', 'Add file', and 'Code'. The main content area displays a list of files with their commit history:

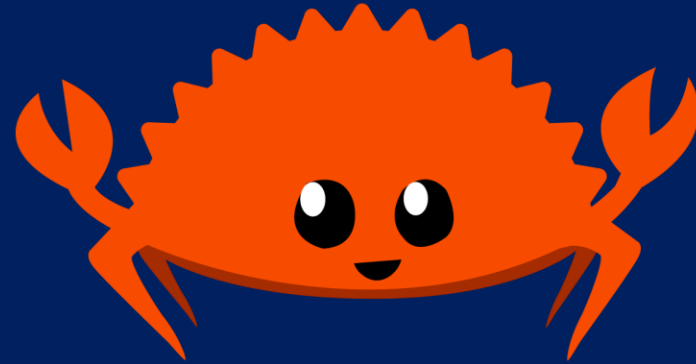
File Name	Commit Message	Time
README.md	Add README	3 years ago
yaffs.ksy	Initial commit	3 years ago
yaffs_decrypt.py	Initial commit	3 years ago
yaffs_tags.ksy	Initial commit	3 years ago
yealink_crypto.py	Initial commit	3 years ago
yealink_rom.ksy	Initial commit	3 years ago
yealink_rom_dump.py	Initial commit	3 years ago

Below the file list, the README content is partially visible, starting with 'Yealink firmware reverse engineering'. On the right side, the 'About' section describes the repository as 'Reverse engineering scripts designed for extracting Yealink VOIP upgrade files'. It also shows 10 stars, 6 watchers, and 5 forks. The 'Releases' and 'Packages' sections both indicate that no releases or packages have been published.

OBJECTIVE 4 : SPEED



Hyperscan



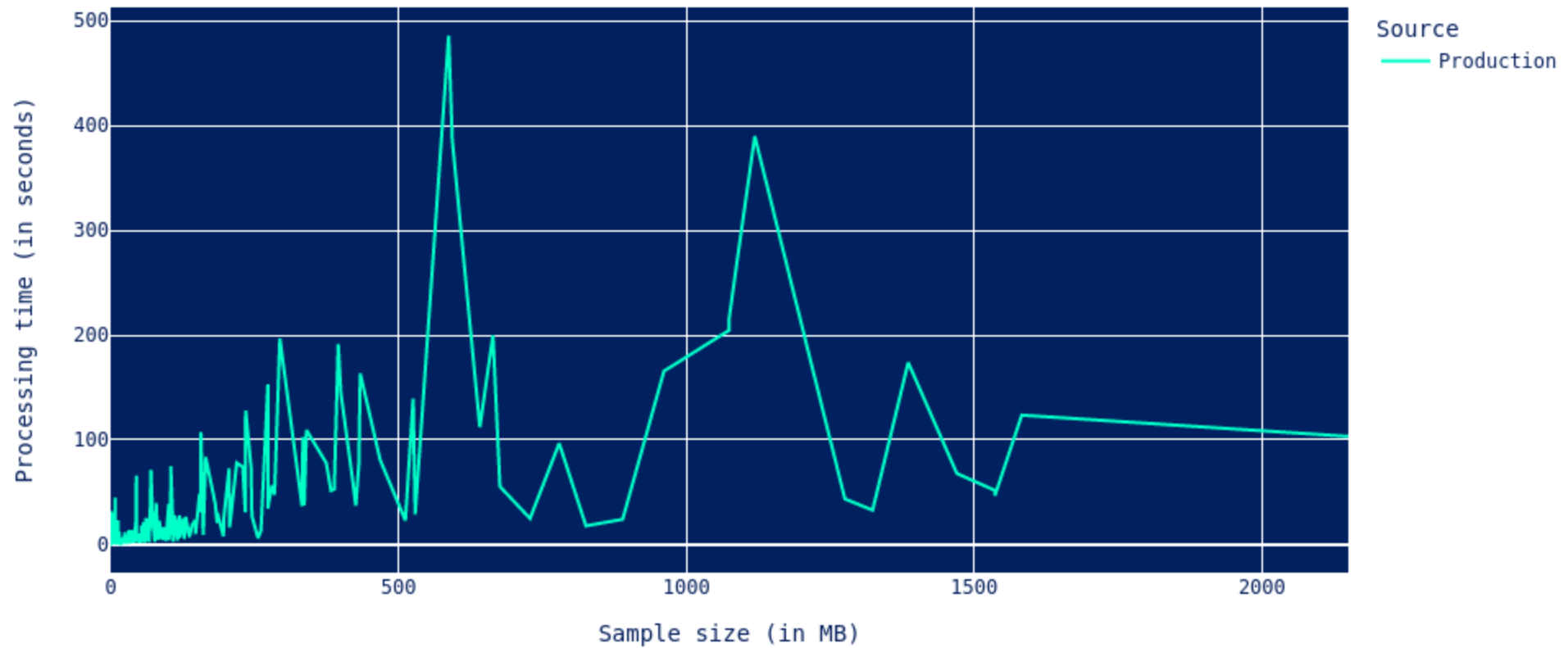
Rust



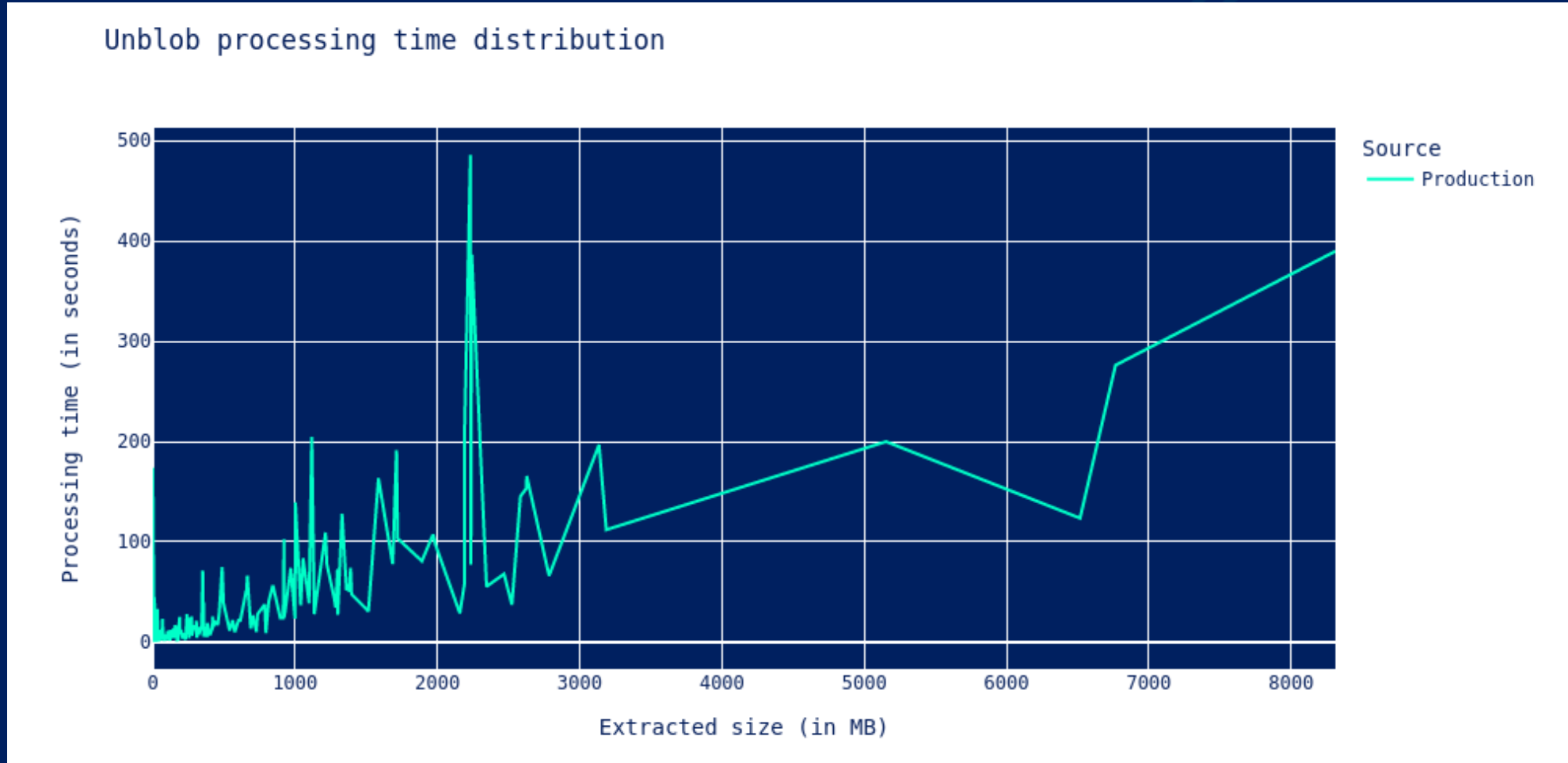
MMAP'ing

OBJECTIVE 4 : SPEED

Unblob processing time distribution



OBJECTIVE 4 : SPEED



The team behind unblob



tests00



György Kiss
kissgyorgy



János Kukovecz
kukovecz



Mücahid KIR
mucoze



Quentin Kaiser
QKaiser



Krisztián Fekete
e3krisztian



László Vaskó
vlaci



Marton Illes
martonilles



mzombor
r4bbit-r4

AGENDA

1. Getting Started
2. Extraction, Analysis & Reporting
3. Creating a Format Handler
4. Creating a Format Extractor
5. Q&A

KEYNOTE

GETTING STARTED

You can start with unblob in 3 different ways:

1. **Docker**
2. **Nix**
3. **Source**

GETTING STARTED (DOCKER)

```
docker run \  
  --rm \  
  --pull always \  
  -v /path/to/extract-dir/on/host:/data/output \  
  -v /path/to/files/on/host:/data/input \  
ghcr.io/onekey-sec/unblob:latest /data/input/path/to/file
```

GETTING STARTED (NIX)



```
nix profile install github:onekey-sec/unblob  
unblob --show-external-dependencies
```

The following executables found installed, which are needed by unblob:

7z	✓
jefferson	✓
lz4	✓
lziprecover	✓
lzop	✓
simg2img	✓
tar	✓
ubireader_extract_files	✓
ubireader_extract_images	✓
unar	✓
unromfs	✓
unsquashfs	✓
yaffshiv	✓

GETTING STARTED (SOURCE)



```
git clone https://github.com/onekey-sec/unblob.git  
cd unblob  
poetry install --no-dev
```

LET'S EXTRACT SOME STUFF !

```
unblob -h
Usage: unblob [OPTIONS] FILE

A tool for getting information out of any kind of binary blob.

You also need these extractor commands to be able to extract the supported
file types: 7z, debugfs, jefferson, lz4, lziprecover, lzop, sasquatch,
sasquatch-v4be, simg2img, ubireader_extract_files, ubireader_extract_images,
unar, yaffshiv, zstd

NOTE: Some older extractors might not be compatible.

Options:
-e, --extract-dir DIRECTORY    Extract the files to this directory. Will be
                                created if doesn't exist.
-f, --force                    Force extraction even if outputs already
                                exist (they are removed).
-d, --depth INTEGER RANGE     Recursion depth. How deep should we extract
                                containers. [default: 10; x>=1]
-n, --entropy-depth INTEGER RANGE
                                Entropy calculation depth. How deep should
                                we calculate entropy for unknown files? 1
                                means input files only, 0 turns it off.
                                [default: 1; x>=0]
-P, --plugins-path PATH       Load plugins from the provided path.
-S, --skip-magic TEXT         Skip processing files with given magic
                                prefix [default: BFLT, JPEG, GIF, PNG,
                                SQLite, compiled Java class, TrueType Font
                                data, PDF document, magic binary file, MS
                                Windows icon resource, PE32+ executable (EFI
                                application)]
-p, --process-num INTEGER RANGE
                                Number of worker processes to process files
                                parallely. [default: 12; x>=1]
--report PATH                  File to store metadata generated during the
                                extraction process (in JSON format).
-k, --keep-extracted-chunks   Keep extracted chunks
-v, --verbose                  Verbosity level, counting, maximum level: 3
                                (use: -v, -vv, -vvv)
--show-external-dependencies  Shows commands needs to be available for
                                unblob to work properly
-h, --help                    Show this message and exit.
```

UNDER THE HOOD: ARCHITECTURE

Unblob core

- CLI or python library
- Pattern matching & chunk carving
- Plugin & Handler management
- Logging & reporting
- Utility functions (parsing, converting etc.)

Unblob handlers & extractors

- Can be defined in plugins
- Format specific chunk calculation
- Format specific extraction (could use external commands)

EXTRACTORS

Extracting sometimes require external extractors, in some case also non-standard tools that could handle the “creative” extensions of various vendors.

We maintain various extractors:

- jefferson (<https://github.com/onekey-sec/jefferson>)
- ubi_reader (https://github.com/onekey-sec/ubi_reader)

We maintain various forks of extractors:

- sasquatch (<https://github.com/onekey-sec/sasquatch>)

Many other extractors are built into unblob.

CODING TIME!

FIRST HANDLER

FIRST EXTRACTOR

ONENOTE

FUTURE WORK

- Clean report of extraction process in console
- Increase meta-data extraction
- Unknown chunks auto-identification (introspection)
- Additional format support

Y
E
K
E
Y
E
N
O

CONTRIBUTE !

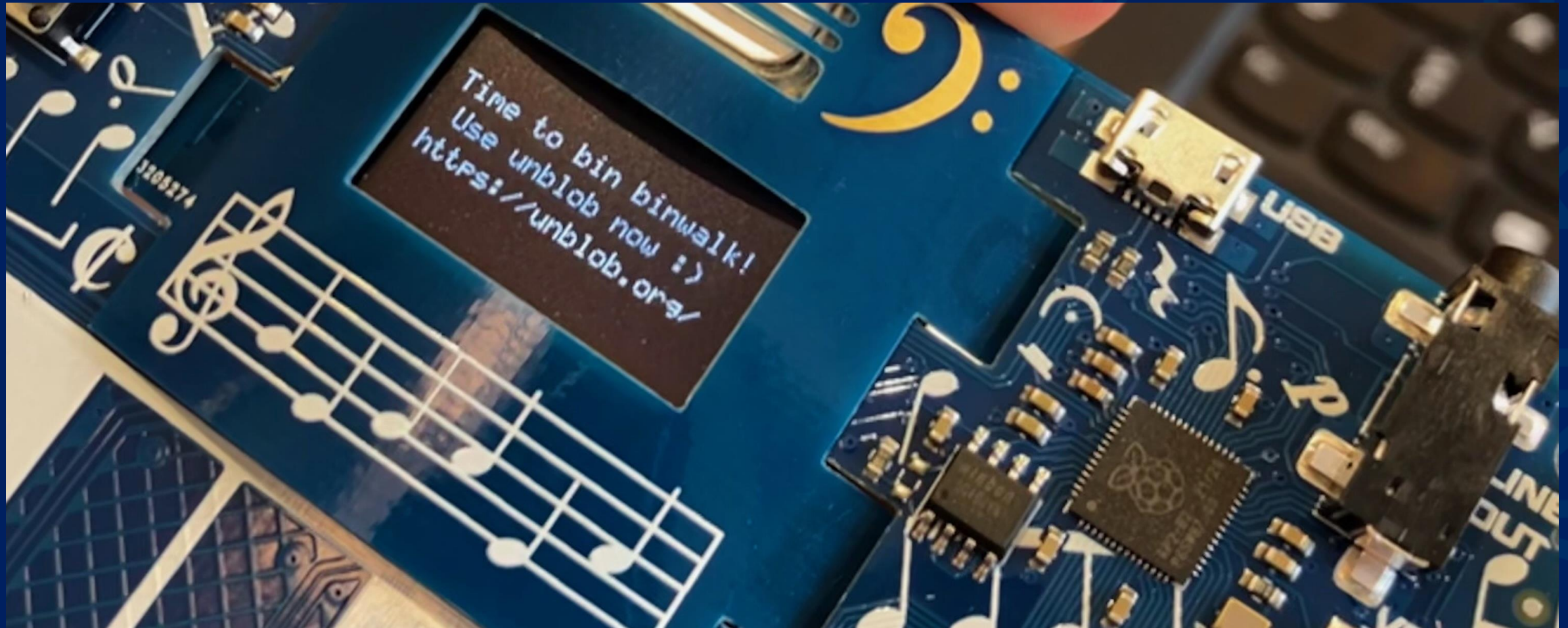
- Test it on your weirdest files
- Report bugs
- Request support for new formats
- Submit PR for new handlers and extractors

<https://unblob.org>

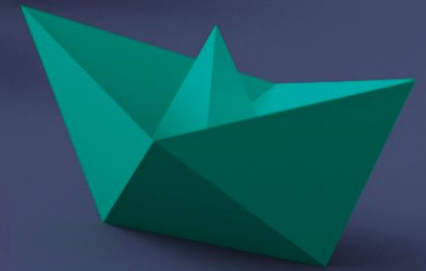
<https://github.com/onekey-sec/unblob>

ONEKEY

CONTRIBUTE !



THANK YOU for your attention.



- Incendiary emails can be sent to research@onekey.com