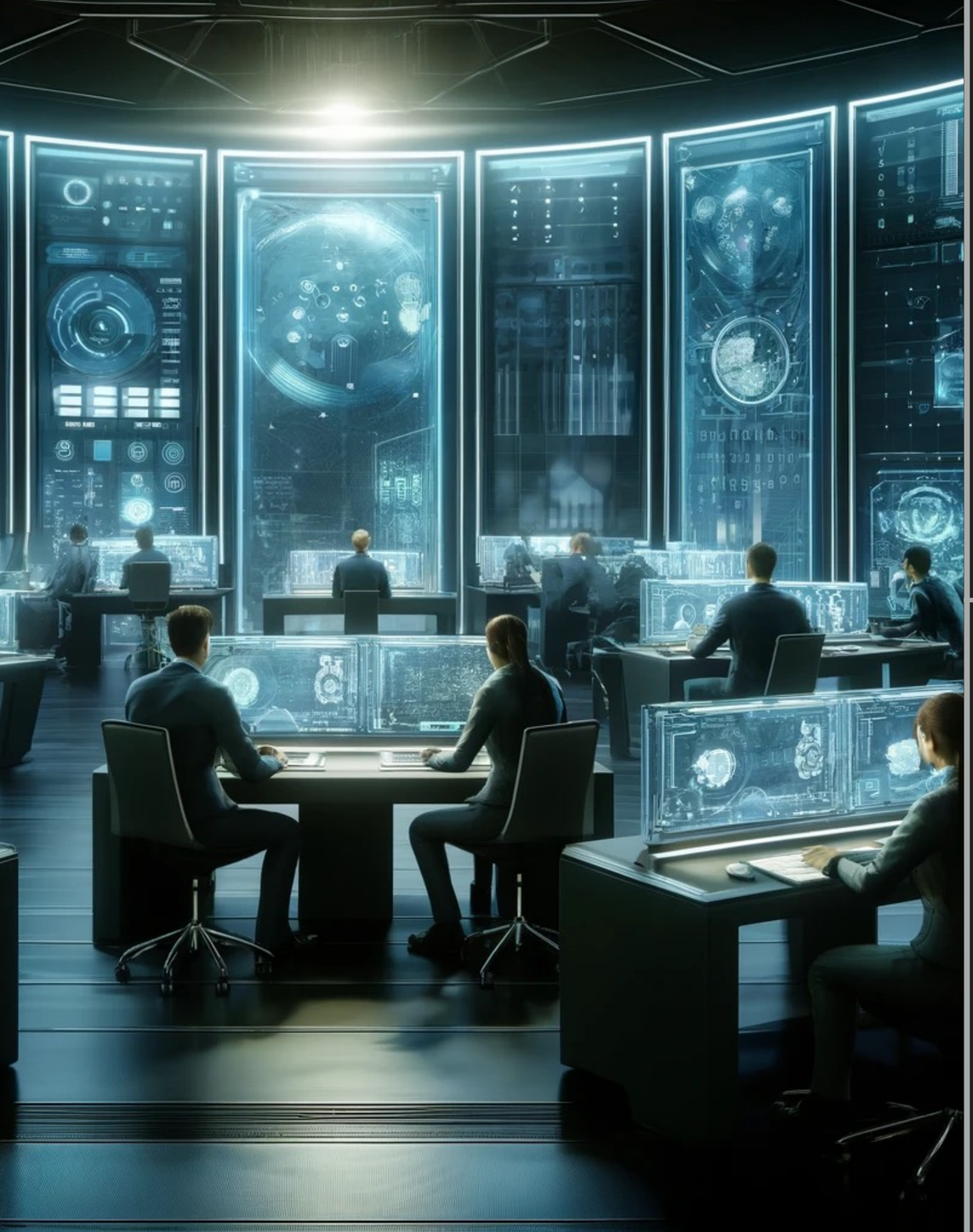


HA

NOT "HIGH AVAILABILITY"
BUT "HUNTING AUTOMATION"

Xavier Mertens | Pass-The-Salt 2024 | July 2024 | TLP:CLEAR

`WH0AMI`

- Xavier Mertens
- Freelance Consultant
- SANS Instructor (FOR610 & FOR710)
- SANS ISC Senior Handler
- Threat Hunter, Dad, Moutainbiker
- BruCON Co-Organizer

I wrote a lot of diaries that cover malicious scripts.

One of the question that was asked to me multiple times:

“How do you find them?”

Source:
https://isc.sans.edu/handler_list.html?author=1016628506&fname=Xavier&lname=Mertens

Xavier Mertens Diaries

Back to Handlers

- Analyzing Synology Disks on Linux
- Malicious PDF File Used As Delivery Mechanism
- Building a Live SIFT USB with Persistence
- Quick Forensics Analysis of Apache logs
- From JavaScript to AsyncRAT
- Using ChatGPT to Deobfuscate Malicious Scripts
- Simple Anti-Sandbox Technique: Where's The Mouse?
- Python InfoStealer With Dynamic Sandbox Detection
- MSIX With Heavily Obfuscated PowerShell Script
- A Python MP3 Player with Builtin Keylogger Capability
- A Batch File With Multiple Payloads
- Facebook AdsManager Targeted by a Python Infostealer
- macOS Python Script Replacing Wallet Applications with Rogue Apps
- One File, Two Payloads
- Are you sure of your password?
- Python Keylogger Using Mailtrap.io
- Shall We Play a Game?
- An Example of RocketMQ Exploit Scanner
- CSharp Payload Phoning to a CobaltStrike Server
- Malicious Python Script with a TCL/TK GUI
- Quasar RAT Delivered Through Updated SharpLoader
- Redline Dropped Through MSIX Package
- Visual Examples of Code Injection
- Example of Phishing Campaign Project File
- Malware Dropped Through a ZPAQ Archive
- Multiple Layers of Anti-Sandboxing Techniques
- Size Matters for Many Security Controls
- Simple Netcat Backdoor in Python Script
- Are You Still Storing Passwords In Plain Text Files?
- macOS: Who?s Behind This Network Connection?
- Python Malware Using Postgresql for C2 Communications
- More Exotic Excel Files Dropping AgentTesla
- Have You Ever Heard of the Fernet Encryption Algorithm?
- Quick Malware Triage With Inotify Tools
- From a Zalando Phishing to a RAT
- Show me All Your Windows!

THE ART OF REVERSE ENGINEERING

- Many people think that reverse engineering is the coolest part of malware analysis. It is defini! But…
- It’s time consuming and most of us don’t have time
- Malware analysis is, most of the time, performed in the scope of a security incident
- Only a few of happy people do RE as a full-time job

- Reverse engineering is not only this →
- Many scripts implements obfuscation and anti-debugging techniques that are also very exciting!
- We always need fresh meat!
- Sources?

```
Disassembly of section .text:

00402000 <.text>:
  402000:   f0 3e 09 00             lock or DWORD PTR ds:[eax],eax
  402004:   00 00                   add    BYTE PTR [eax],al
  402006:   00 00                   add    BYTE PTR [eax],al
  402008:   48                      dec    eax
  402009:   00 00                   add    BYTE PTR [eax],al
  40200b:   00 02                   add    BYTE PTR [edx],al
  40200d:   00 05 00 88 5d 01       add    BYTE PTR ds:0x15d8800,al
  402013:   00 38                   add    BYTE PTR [eax],bh
  402015:   e1 07                   loope  0x40201e
  402017:   00 03                   add    BYTE PTR [ebx],al
  402019:   00 00                   add    BYTE PTR [eax],al
  40201b:   00 cf                   add    bh,cl
  40201d:   06                      push   es
  40201e:   00 06                   add    BYTE PTR [esi],al
     ...
  402050:   9a 03 04 05 8c 01 00    call   0x1:0x8c050403
  402057:   00 1b                   add    BYTE PTR [ebx],bl
  402059:   73 01                   jae    0x40205c
  40205b:   00 00                   add    BYTE PTR [eax],al
  40205d:   0a 14 6f                or     dl,BYTE PTR [edi+ebp*2]
  402060:   02 00                   add    al,BYTE PTR [eax]
  402062:   00 0a                   add    BYTE PTR [edx],cl
  402064:   25 3a 07 00 00          and    eax,0x73a
  402069:   00 26                   add    BYTE PTR [esi],ah
  40206b:   05 8c 01 00 00          add    eax,0x18c
  402070:   1b a5 01 00 00 1b       sbb    esp,DWORD PTR [ebp+0x1b000001]
  402076:   2a 00                   sub    al,BYTE PTR [eax]
  402078:   3a 02                   cmp    al,BYTE PTR [edx]
  40207a:   28 03                   sub    BYTE PTR [ebx],al
  40207c:   00 00                   add    BYTE PTR [eax],al
  40207e:   0a 03                   or     al,BYTE PTR [ebx]
  402080:   04 28                   add    al,0x28
  402082:   01 00                   add    DWORD PTR [eax],eax
  402084:   00 2b                   add    BYTE PTR [ebx],ch
  402086:   2a 00                   sub    al,BYTE PTR [eax]
  402088:   13 30                   adc    esi,DWORD PTR [eax]
  40208a:   06                      push   es
  40208b:   00 45 00                add    BYTE PTR [ebp+0x0],al
  40208e:   00 00                   add    BYTE PTR [eax],al
  402090:   01 00                   add    DWORD PTR [eax],eax
```

SOURCES: CUSTOMERS

- Some ISC diaries have been based on real stories
- $CUSTOMER agreed to share (TLP:Clear only)
- Interesting data are always obfuscated. Only the customer who’s reading the diary will recognise some facts ;-)

SOURCES: YOU

- Manual submissions by the ISC readers. Thanks to you! ❤️

How To Contact Us

We do get a lot of mail. For issues that aren't sensitive our community forums may be more suitable. For example, if you want to learn about a particular security related topic, or you would like some input regarding a packet trace you collected (and you don't mind sharing it in public), post it to the community forums.

Submit Logs

If you would like to add logs you've collected to our database, submit them to us!

Contact Us

You may want to use this form to:

- Share your ideas about the site, or today's diary
- Submit logs which you would rather submit in private
- Notify us of any security event that should be covered

Success! CLOUDFLARE Privacy · Terms

Contact Form
We are not able to respond to you unless you provide an e-mail address!

Email Address
example@email.com

Name
enter name

Subject
enter subject

Note about attached files: Please attach any files "as found". If necessary, zip or tar multiple files into one. But try not to encrypt or obfuscate the files, as this may hinder analysis.

To anybody scanning this site for vulnerabilities: We enourage our readers to use this form to submit malicious code, viruses, spyware and all kinds of evil files. Us not filtering uploads for extensions like ".exe" is not a vulnerability, it is a feature. Submitting a vulnerability stating so shows that you have not actually read this notice and just used a lazy automated tool.

Attach a File: Choose File no file selected

Your message:

SOURCES: SPAM

- Catch-all mailboxes for lot of old or juicy domain names.
- That’s why I keep running my own mail server! (Challenging these days)
- ripMIME[1]
- Automatic extraction of interesting attachments from spams

[1] https://pldaniels.com/ripmime/

SOURCES: SPAM

```
$ grep ^@pwn3d.be /etc/postfix/virtual
@pwn3d.be       ripmime, honeypot


$ grep ^ripmime /etc/aliases
ripmime: "|/usr/bin/ripmime -i - -d /data/ripmime --prefix=ripmime- --no-nameless --syslog --paranoid"
$ crontab -l | grep ripmime


*/15 * * * * /usr/local/bin/submit2mwdb.py -d /data/ripmime -o /var/log/mwdb.json -r -v >>/var/log/
submit2mwdb.log 2>&1
```

More soon...

SOURCES: VT

```
rule PowerShellSuspiciousStrings
{
    strings:
        $s1 = "Add-Exfiltration" nocase wide ascii
        $s2 = "Add-Persistence" nocase wide ascii
        $s3 = "Add-RegBackdoor" nocase wide ascii
        $s4 = "Add-ScrnSaveBackdoor" nocase wide ascii
        $s5 = "Check-VM" nocase wide ascii
        $s6 = "Do-Exfiltration" nocase wide ascii
        $s7 = "Enabled-DuplicateToken" nocase wide ascii
        $s8 = "Exploit-Jboss" nocase wide ascii
        $s9 = "Find-Fruit" nocase wide ascii
        $s10 = "Find-GPOLocation" nocase wide ascii
        $s11 = "Find-TrustedDocuments" nocase wide ascii
        $s12 = "Get-ApplicationHost" nocase wide ascii
        $s13 = "Get-ChromeDump" nocase wide ascii
        $s14 = "Get-ClipboardContents" nocase wide ascii
        $s15 = "Get-FoxDump" nocase wide ascii
        $s16 = "Get-GPPPassword" nocase wide ascii
        $s17 = "Get-IndexedItem" nocase wide ascii
        $s18 = "Get-KeyStrokes" nocase wide ascii
        $s19 = "Get-LSASecret" nocase wide ascii
        $s20 = "Get-PassHashes" nocase wide ascii
        $s21 = "Get-RegAlwaysInstalledElevated" nocase wide ascii
        $s22 = "Get-RegAutoLogon" nocase wide ascii
        $s23 = "Get-RickAstley" nocase wide ascii
        $s24 = "Get-Screenshot" nocase wide ascii
        $s25 = "Get-SecurityPackages" nocase wide ascii
        $s26 = "Get-ServiceFilePermission" nocase wide ascii
        $s27 = "Get-ServicePermission" nocase wide ascii
        $s28 = "Get-ServiceUnquoted" nocase wide ascii
        $s29 = "Get-SiteListPassword" nocase wide ascii
        […]

    condition:
        any of ($s*) and file_type contains "text" and new_file and filesize < 5MB and positives > 0 and positives < 20
}
```

SOURCES: VT

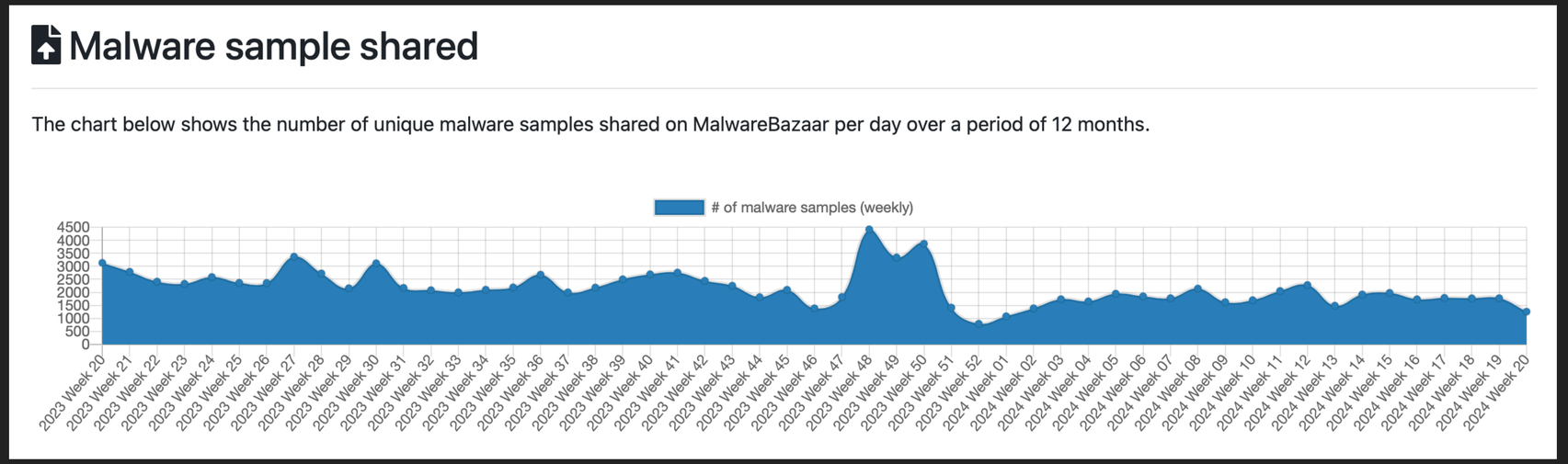
```
$ vt-hunting.py -h
usage: vt-hunting.py [-h] [-api API] [-thres THRESHOLD] [-cleanup] [-dl]
                     [-puri PROXY_URI] [-pusr PROXY_USER]
                     [-ppwd PROXY_PASSWORD] [-json JSON] [-out OUTPUT]
                     [-samples SAMPLES_DIRECTORY]

Retrieve results of VirusTotal Hunting.

optional arguments:
  -h, --help            show this help message and exit
  -api API, --api API   VirusTotal API key
  -thres THRESHOLD, --threshold THRESHOLD
                        Number of required infection to keep result (default
                        3)
  -cleanup, --cleanup   Cleanup notifications of retreived files from
                        VirusTotal
  -dl, --download       Download the samples in addition to getting
                        notifications
  -puri PROXY_URI, --proxy_uri PROXY_URI
                        Proxy URI
  -pusr PROXY_USER, --proxy_user PROXY_USER
                        Proxy User
  -ppwd PROXY_PASSWORD, --proxy_password PROXY_PASSWORD
                        Proxy User
  -json JSON, --json JSON
                        JSON file to use to store full Hunting raw result (by
                        default not done)
  -out OUTPUT, --output OUTPUT
                        File to store result (by default stdout
  -samples SAMPLES_DIRECTORY, --samples_directory SAMPLES_DIRECTORY
                        Directory where to wrote all matching samples (by
                        default not done)
```

THE NEED FOR (SPEED) AUTOMATION

- A fact:



- We are lazy!
- All of us are facing the same problem with triage
- Automation is key!

Source: https://bazaar.abuse.ch/

DISCLAIMER

- The open source landscape is full of awesome projects
- Some of them are definitively overlapping
- This presentation is based on my own choices (freedom first!)
- No flame war please! But I’m always open to discussions (with 🍻)

DISCLAIMER (2)

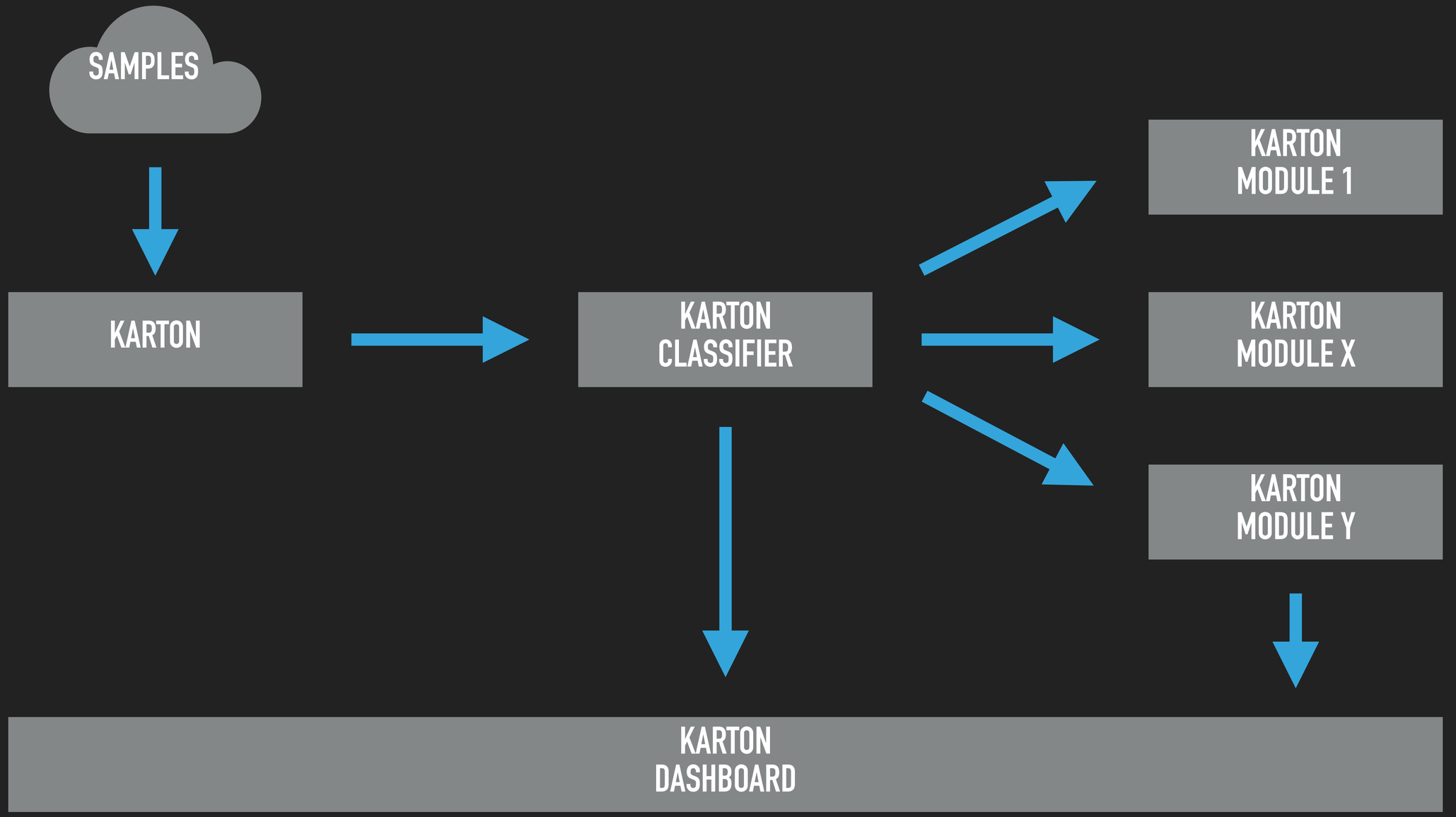
- Many organizations today deploy SOAR (Security Orchestration, Automation, and Response) tools
- “R” is not the scope of this talk
- Even if they are powerful, that don’t provide the same level of customization
- “No-code” has limitations

THE ORCHESTRATOR: KARTON

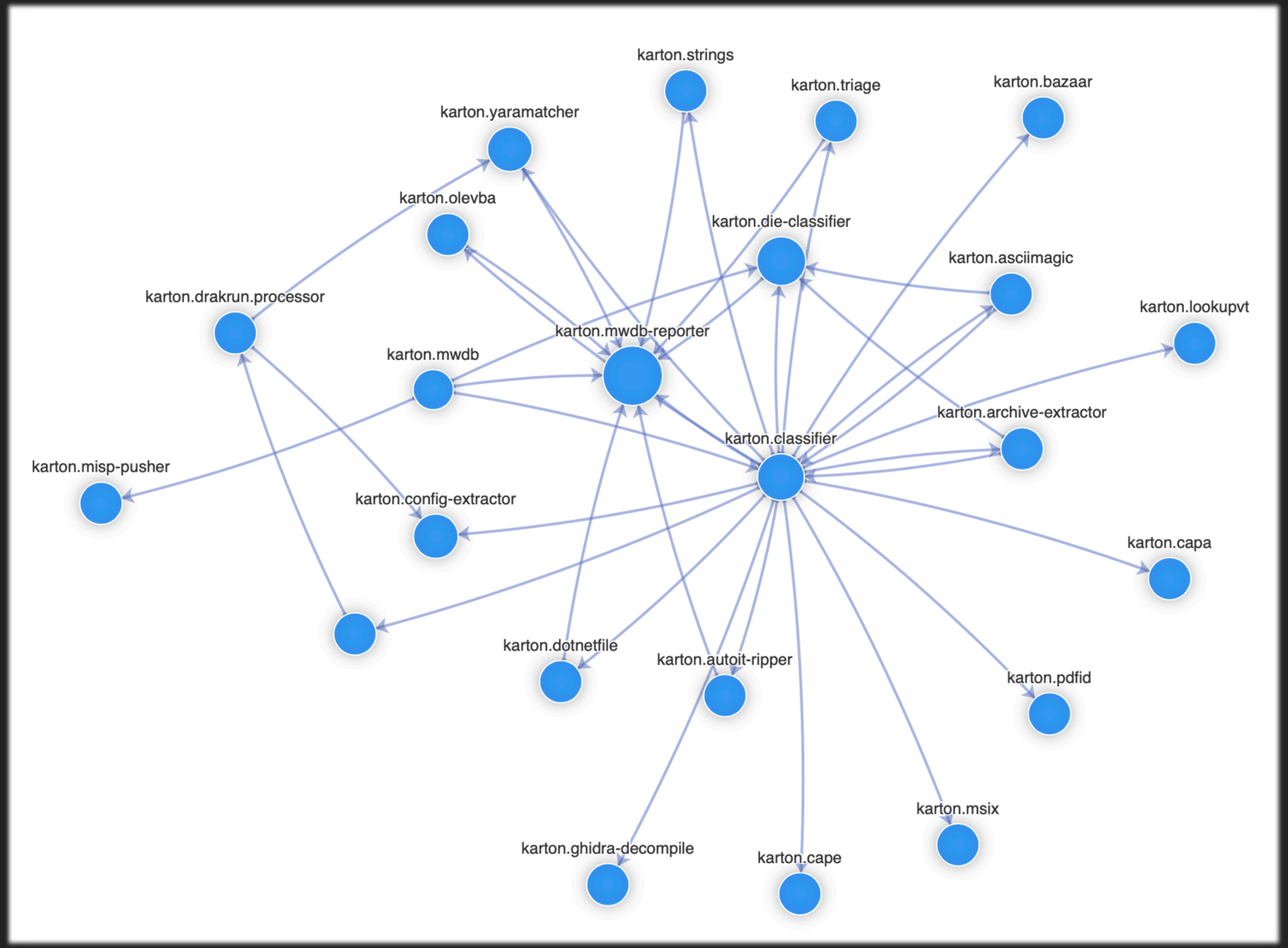
> “Karton is a robust framework for creating flexible and lightweight malware analysis backends. It can be used to connect malware* analysis systems into a robust pipeline with very little effort.”

- We need an “orchestrator”
- Developed by CERT.pl
- Core components: Python, Redis, Minio
- Easy to expand based on **your** needs

KARTON: BASIC ARCHITECTURE



THE ORCHESTRATOR: KARTON



THE ORCHESTRATOR: KARTON

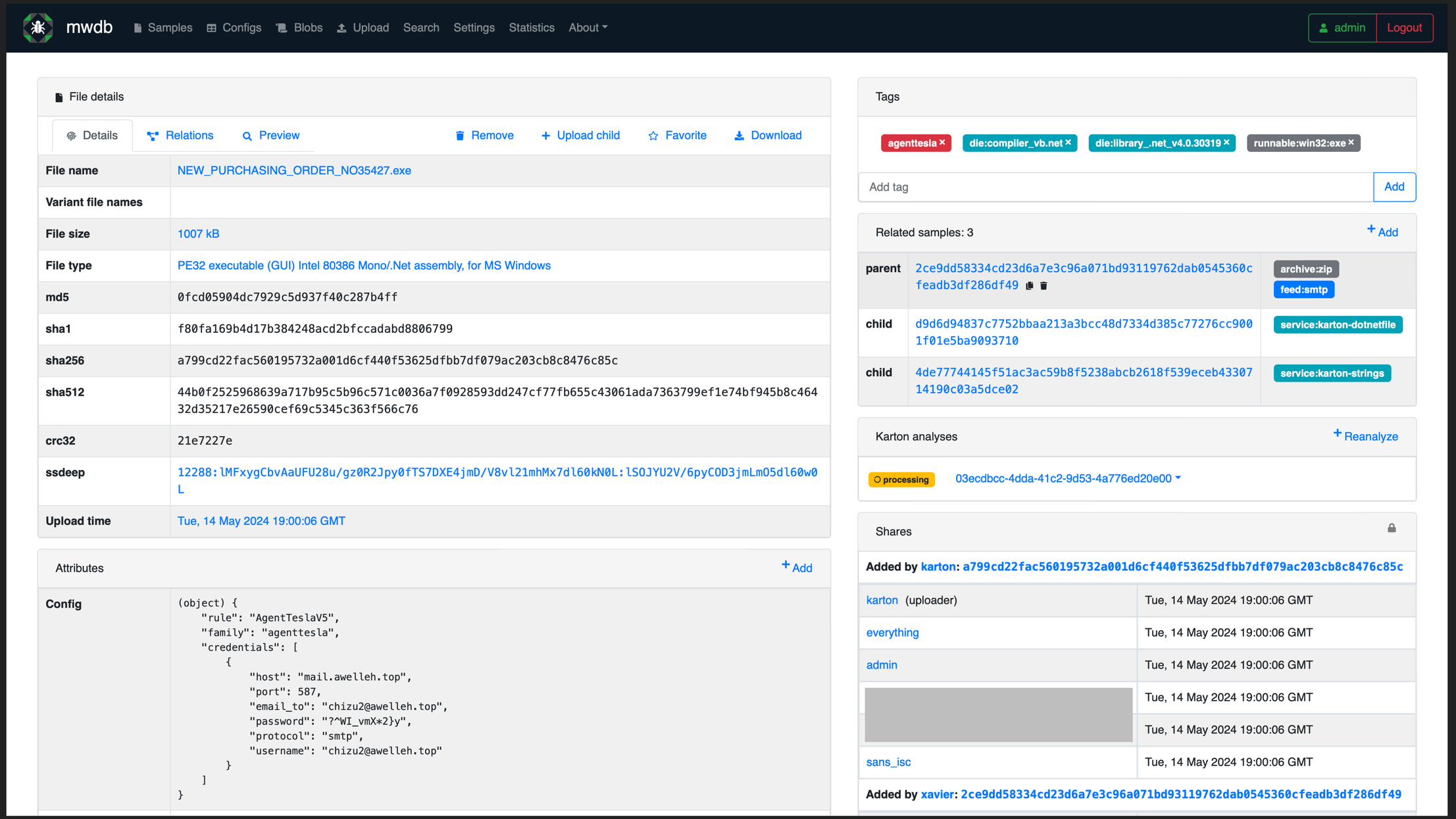
- Some Karton services (“modules”) are standard:
  - Karton-archive-extractor
  - Karton-config-extractor
  - Karton-mwdb-reporter
  - Karton-yaramatcher
  - Karton-asciimagic
  - Karton-autoit-ripper

THE REPOSITORY: MWDB

> “Malware repository component for automated malware collection/analysis systems.”

- We need a central place to store our samples
- Again developed by CERT.pl
- The key parameter is “enable_karton = 1” in mwdb.ini!

THE REPOSITORY: MWDB



EXPANDING KARTON

- It’s pretty easy to write your own Karton services
- Based on Python
- A Python library is available for an easy integration
- Everything is based on a class

EXPANDING KARTON

```
from karton.core import Karton, Task, Resource
import subprocess
import io
import os
import time
from mwdblib import MWDB

class MyTestClass(Karton):
    """
    Execute a tool against the sample
    """
    identity = "karton.test"
    filters = [
        {"type": "sample", "stage": "recognized"}
    ]

   def process(self, task: Task) -> None:
        sample_resource = task.get_resource("sample")
        self.log.info("Processing " + sample_resource.name)
        with sample_resource.download_temporary_file() as sample_file:
            results = subprocess.check_output(["/usr/local/bin/tool", sample_file.name])
        try:
            # Submit the resuts to MWDB
            mwdb = MWDB(api_key="xxxxxxx", api_url="http://127.0.0.1:5000/api/")
            sample = mwdb.query_file(sample_resource.sha256)
            # Add a comment
            sample.add_comment(results.decode("utf-8"))
        except:
            self.log.info(“ERROR: Cannot add comment")

if __name__ == "__main__":
   MyTestClass().loop()
```

EXPANDING KARTON

- My own services:
  - Karton-cape: Submit the sample to a CAPEv2 sandbox
  - Karton-pdfid: Static analysis of PDF files
  - Karton-capa: Analyze the PE file with CAPA
  - Karton-olevba: Extract VBA macros from OLE docs
  - Karton-bazaar-tags: Enrich samples with MalwareBazaar tags
  - Karton-die-classifier: Analyze the PE file with DIEC

COVERAGE OF NEW THREATS

- The modularity of services helps to address new threats in a fast way.
- Example: The (new) MSIX format

```
$ zipdump.py WSJ.msix
Index Filename                                      Encrypted Timestamp
    1 Registry.dat                                          0 2024-04-25 06:02:42
    2 User.dat                                              0 2024-04-25 06:02:42
    3 Assets/logo.png                                       0 2024-04-25 06:02:42
    4 config.json                                           0 2024-04-25 06:02:42
    5 PsfLauncher32.exe                                     0 2024-04-25 06:02:42
    6 PsfLauncher64.exe                                     0 2024-04-25 06:02:42
    7 PsfRunDll32.exe                                       0 2024-04-25 06:02:42
    8 PsfRunDll64.exe                                       0 2024-04-25 06:02:42
    9 PsfRuntime32.dll                                      0 2024-04-25 06:02:42
   10 PsfRuntime64.dll                                      0 2024-04-25 06:02:42
   11 Resources.pri                                         0 2024-04-25 06:02:42
   12 StartingScriptWrapper.ps1                             0 2024-04-25 06:02:42
   13 obhUdhwwJAgAcFkRYh.ps1                                0 2024-04-25 06:02:42
   14 VFS/ProgramFilesX64/PsfRunDll64.exe                   0 2024-04-25 06:02:42
   15 AppxManifest.xml                                      0 2024-04-25 06:02:42
   16 AppxBlockMap.xml                                      0 2024-04-25 06:02:42
   17 [Content_Types].xml                                   0 2024-04-25 06:02:42
   18 AppxMetadata/CodeIntegrity.cat                        0 2024-04-25 06:02:42
   19 AppxSignature.p7x                                     0 2024-04-25 18:03:58
```

COVERAGE OF NEW THREATS

```
$ zipdump.py WSJ.msix -s 4 -d
{
  "applications": [
    {
      "id": "NOTEPAD",
      "executable": "VFS\\ProgramFilesX64\\PsfRunDll64.exe",
      "scriptExecutionMode": "-ExecutionPolicy RemoteSigned",
      "startScript": {
        "waitForScriptToFinish": false,
        "runOnce": false,
        "showWindow": false,
        "scriptPath": "obhUdhwwJAgAcFkRYh.ps1"
      }
    }
  ]
}
$ zipdump.py WSJ.msix -s 13 -d
$ezeMfzQTQkenHwzrlBwQaQI = Start-Job -ScriptBlock {
    $CUPnhUAQUPsCnVCx = (Get-WmiObject -Class Win32_OperatingSystem).Caption
    $ZSUPBUFRYPHUSeiQNHamHQoSGS = '25'
    $oGqc = 'd99844e1-4599-410e-aa0d-b504c5ca3ddf'
    $oIvwtreZuuFIjlGcbweepNMr = [System.Net.WebUtility]::UrlEncode($CUPnhUAQUPsCnVCx)
    $A = Get-WmiObject Win32_ComputerSystem | Select-Object -ExpandProperty Domain
    $NPjPfMoVbuNNDVDWtoPoyI = Get-WmiObject -Namespace "root\SecurityCenter2" -Class AntiVirusProduct
    $GpGXj = $NPjPfMoVbuNNDVDWtoPoyI | ForEach-Object {
        $_.displayName
    }
[…]
```

FEEDING MWDB

- MWDB has a REST API interface
- Easy to integrate with 3rd party tools
- Wrote my own tool to manually submit files

```
$ mwdb.py -h
Usage: mwdb.py [options] [files]

Options:
  -h, --help            show this help message and exit
  -c CONFIG, --config=CONFIG
                        configuration file for MWDB instance (default:
                        $HOME/.mwdb)
  -t TAG, --tag=TAG     tag to add to the sample
  -v SHA256, --virustotal=SHA256
                        SHA256 of file to download from VT
  -m SHA256, --malwarebazaar=SHA256
                        SHA256 of file to download from Malware Bazaar
  -n NAME, --name=NAME  Overwirte the name of the uploaded file from VT/MB
```

- Processing files from a directory (extracted by ripmime)

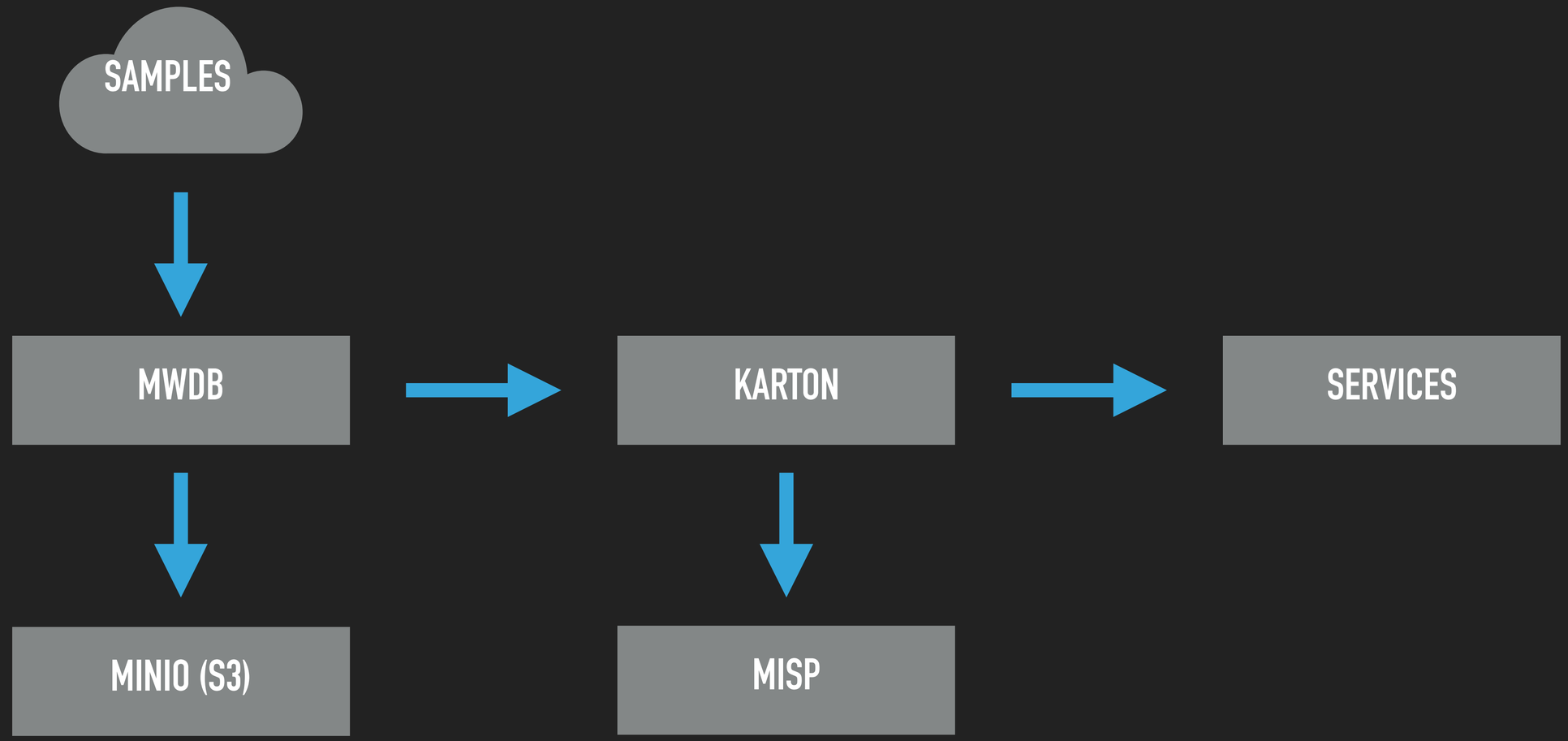
```
$ submit2mwdb.py -h
Usage: submit2mwdb.py [options]

Options:
  -h, --help            show this help message and exit
  -d DIRECTORY, --directory=DIRECTORY
                        directory to scan
  -v, --verbose         display debug information
  -r, --remove          remove processed files
  -o OUTPUT_FILE, --output=OUTPUT_FILE
                        output file (json format)
```

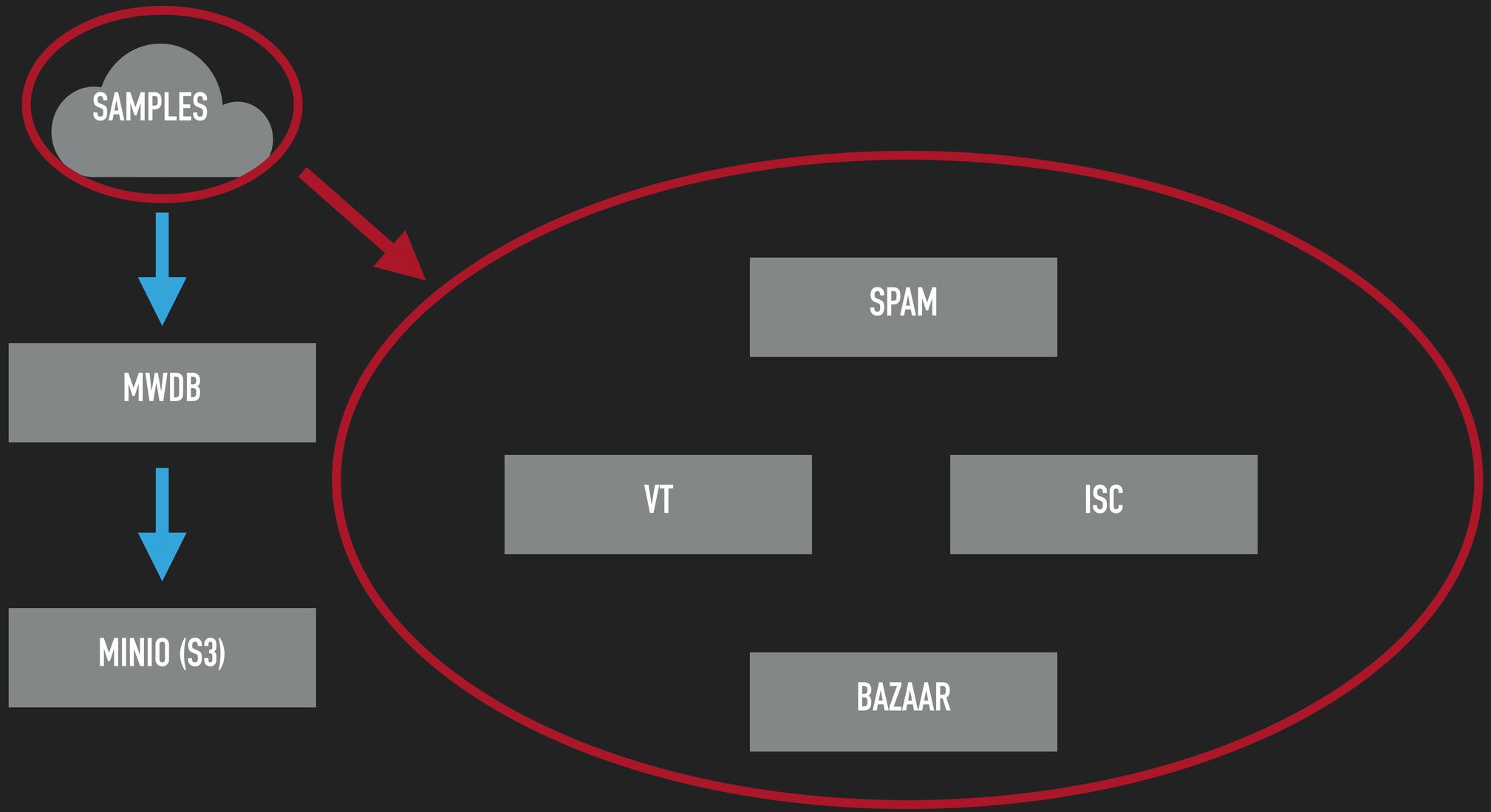
- Filering is based in MIME types

```
EXCLUDED_MIMETYPES = [
    'image/jpeg',
    'image/png',
    'image/gif',
    'text/calendar',
    'DOS/MBR boot sector;',
]
```

COMPLETE ARCHITECTURE OVERVIEW



COMPLETE ARCHITECTURE OVERVIEW



CONCLUSIONS

- Multiple sources
- Automation
- Easy to expand to cover new threats. Example: MSIX
- Next?

THANK YOU

- My scripts are not published yet but feel free to contact me
- xavier@xameco.be
- xmertens@sans.org
- xmertens@isc.sans.edu
- Happy hunting!