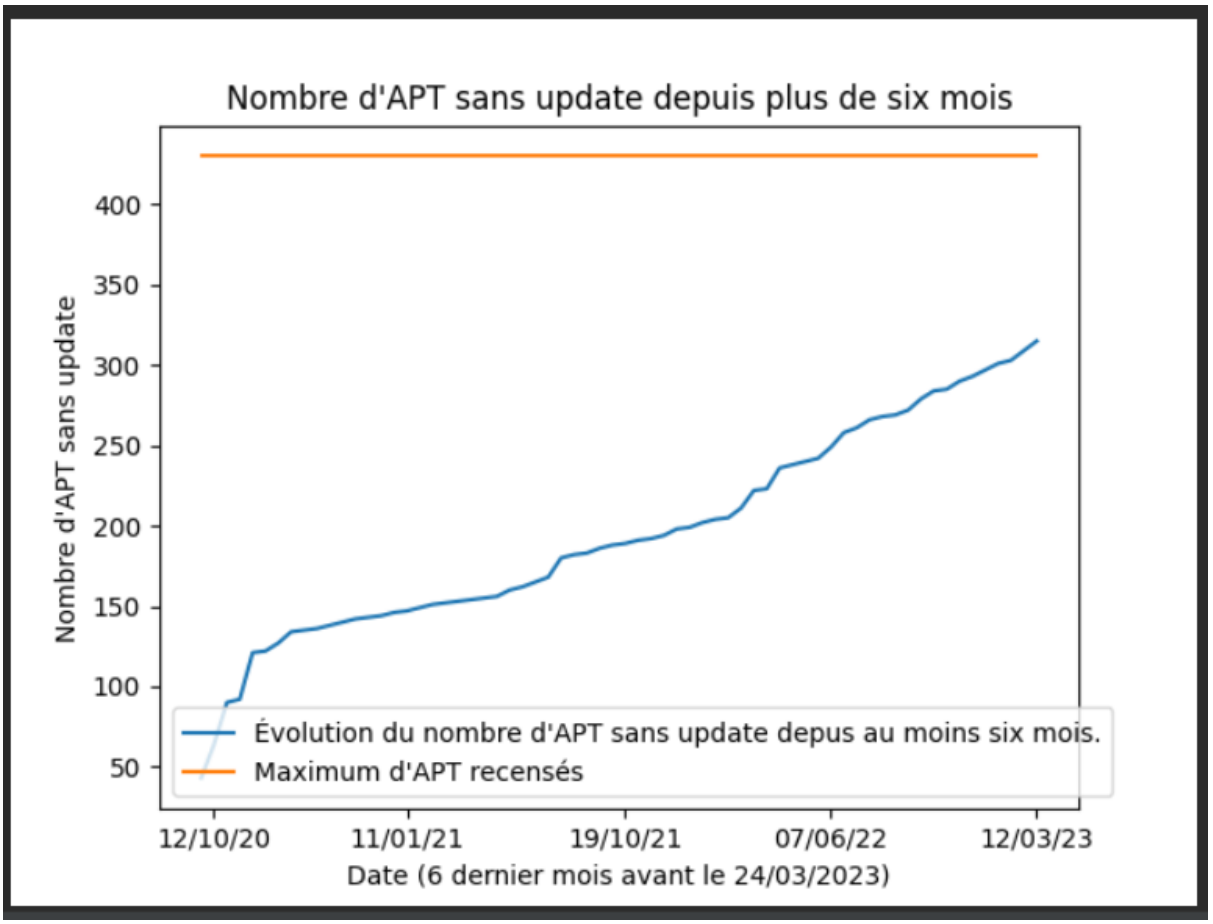
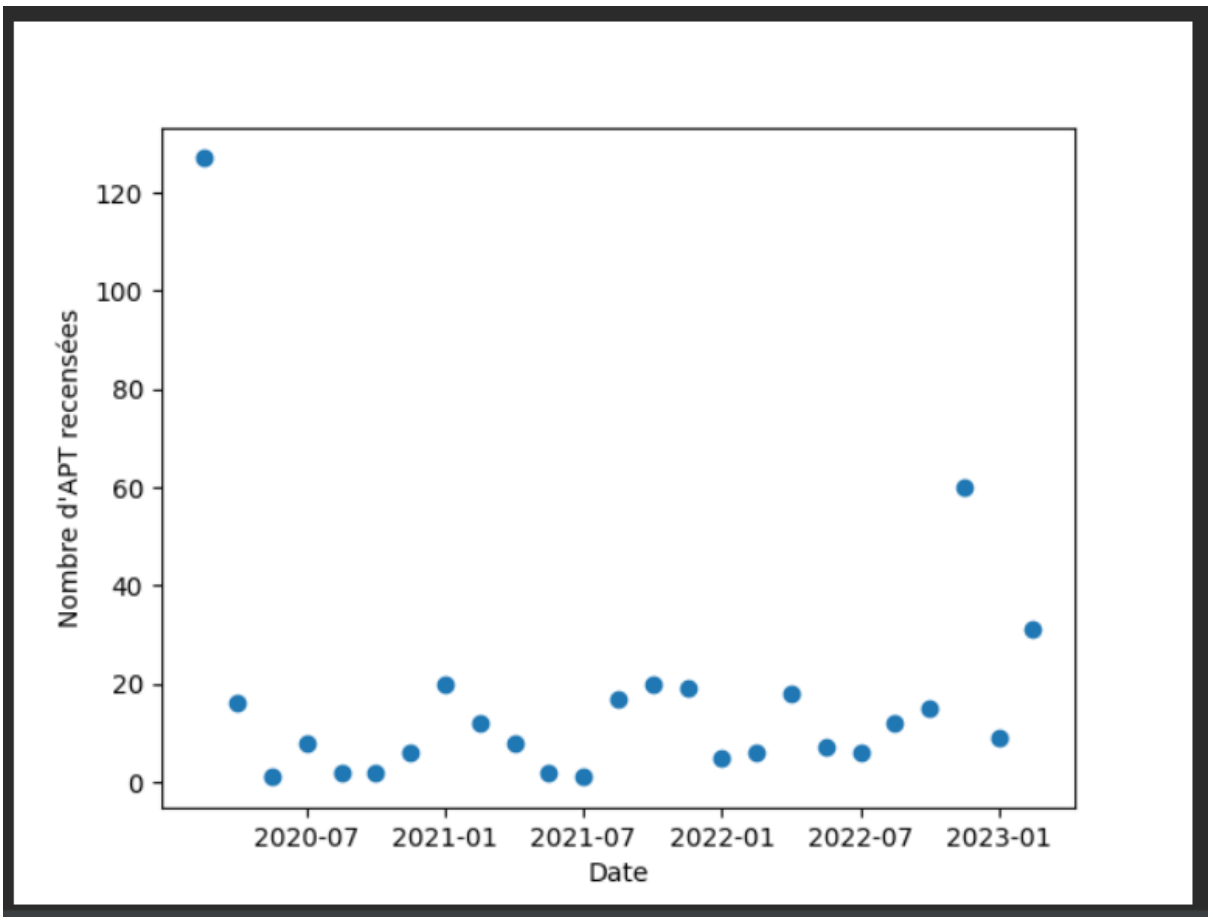
3 niveaux de progression
– I. Estimer le niveau de menace global à partir des informations sur les APT
● I.1 Trouver une base sur les APT

⇒ JSON contenant toutes les APT

● I.2 Analyser le contenu (Json) pour voir les plus actifs récemment

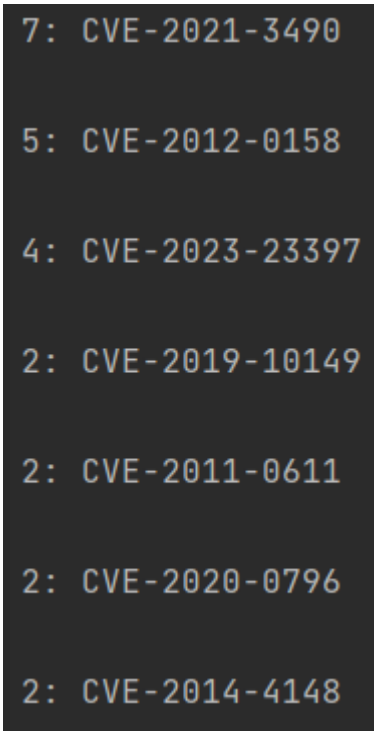




– II. Estimer la menace à partir des informations CVE
● II.1 Informations CVE

⇒ Tri par les années les plus récents

```
sers > victo > Desktop > dossier victor > Dossier ISEN > 5eme annee > analyse et gestion des risques > Ge
import json
import time

import requests
import re
from bs4 import BeautifulSoup

with open("Ressources/APT.json", "r", encoding="utf-8") as f:
    data = json.load(f)

links = []

for value in data["values"]:
    if "information" in value and "last-card-change" in value:
        year = int(value["last-card-change"][:4])
        if year in [2018, 2019, 2020, 2021, 2022, 2023]:
            links.extend(value["information"])

for link in links:
    try:
        tab = []
        # Envoyer une requête GET à la page
        response = requests.get(link)

        # Vérifier si le contenu de la réponse est du type "text/html"
        if "text/html" in response.headers["content-type"]:
            # Extraire les codes CVE de la page HTML
            soup = BeautifulSoup(response.content, "html.parser")
            cve_tags = soup.find_all(string=lambda text: "CVE-" in text)

            # Utiliser une expression régulière pour extraire le code CVE
            cve_pattern = r'CVE-\d{4}-\d{4,7}'
            for cve_tag in cve_tags:
                cve_match = re.search(cve_pattern, cve_tag)
                if cve_match:

                    cve_code = cve_match.group()
                    if not tab.__contains__(cve_code):
                        print(f"{cve_code}")
                        tab.insert(cve_code)

        time.sleep(1)
    except:
        continue
```



7: CVE-2021-3490 7.8

5: CVE-2012-0158 n/a

4: CVE-2023-23397 9.8

2: CVE-2019-10149 9.8

2: CVE-2011-0611 n/a

2: CVE-2020-0796 10.0

2: CVE-2014-4148 n/a

● II.2 Trouver lien CVE/CVSS

https://nvd.nist.gov/vuln/data-feeds

Feed	Updated	Download	Size (MB)
CVE-Modified	03/24/2023; 10:00:01 AM -0400	META	
		GZ	0.26 MB
		ZIP	0.26 MB
CVE-Recent	03/24/2023; 10:00:00 AM -0400	META	
		GZ	0.13 MB
		ZIP	0.13 MB
CVE-2023	03/24/2023; 3:00:01 AM -0400	META	
		GZ	0.68 MB
		ZIP	0.68 MB
CVE-2022	03/24/2023; 3:00:16 AM -0400	META	
		GZ	5.49 MB
		ZIP	5.49 MB
CVE-2021	03/24/2023; 3:00:34 AM -0400	META	
		GZ	6.00 MB
		ZIP	6.00 MB
CVE-2020	03/24/2023; 3:00:51 AM -0400	META	
		GZ	5.44 MB
		ZIP	5.44 MB
CVE-2019	03/24/2023; 3:01:05 AM -0400	META	
		GZ	4.48 MB
		ZIP	4.48 MB
CVE-2018	03/24/2023; 3:01:17 AM -0400	META	
		GZ	3.94 MB
		ZIP	3.94 MB

● II.3 Trouver technologies utilisées par Shopify

Front classique html, js framework, jQuery
Pour la partie admin: React en typescript
Base de données: MySQL
Utilisation de Ruby

● II.4 Produire un état de la menace en fonction des actifs liés à Rogenig

AD https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=active+directory (223)

Windows 11 https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=windows-11 (111)

Palo https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=palo (153)

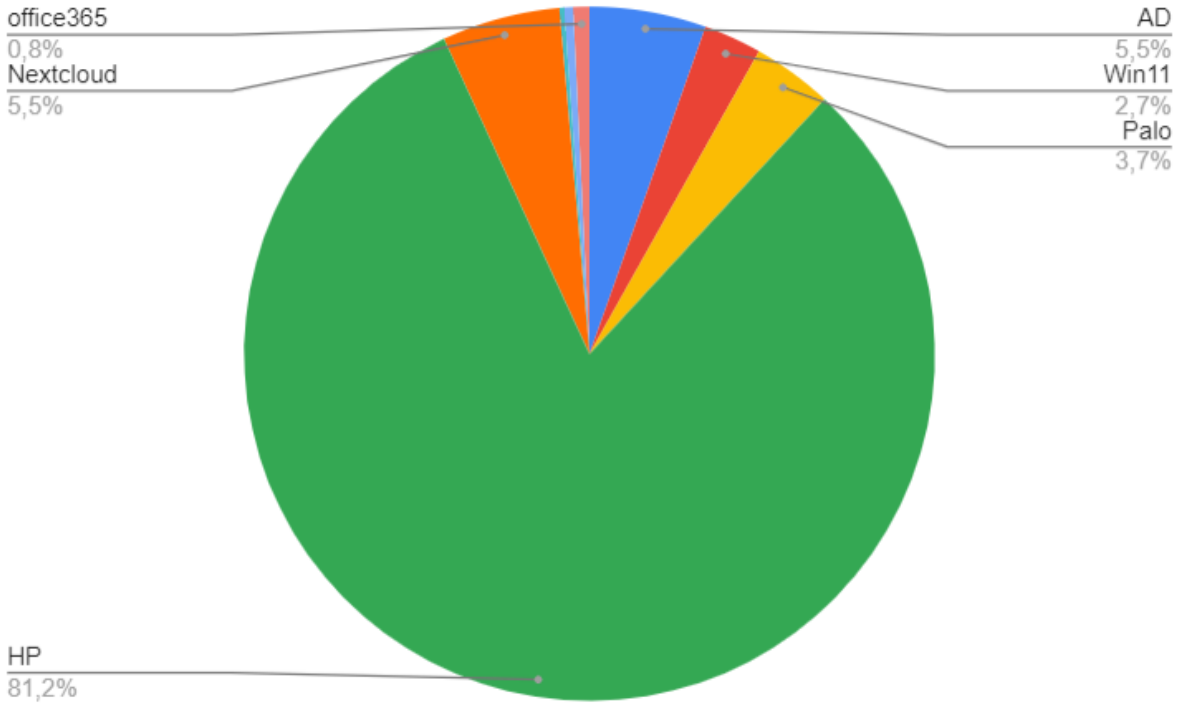
HP https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=hp (3319)

NextCloud https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=nextcloud (226)

KAV https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=kav (9)

Veeam 17

office365 31



– III. Mettre en place une veille à base de mots-clés