Cyber-Sécurité

TD6 - 2023

Forensics des e-mail en Python

TD individuel

« La connaissance s'acquiert par l'expérience,
tout le reste n'est que de l'information.»
Albert Einstein

- **Modalité de réalisation du TD**

- **Documents et liens pour la réalisation du TD :**
  - **Python :**
    - Document Python officiel :https://docs.python.org/fr/3/
    - Document Python Fonction officiel : https://docs.python.org/3/library/functions.html
    - Index des bibliothèques officielles Python : https://docs.python.org/3/py-modindex.html
    - Communauté officielle Python : https://www.python.org/community/
  - **Jupiter-Lab :**
    - **Ensemble des exercices de ce TD devra être réaliser avec Jupiter-lab**
  - Installation de Jupyter-Lab : https://jupyter.org/install
  - Documentation de Jupiter-lab : https://docs.jupyter.org/en/latest/
  - Utilisation de Jupiterlab pour le champ **Markdown** :
    - https://jupyterlab.readthedocs.io/en/stable/user/file_formats.html

  - **Modalité de Réalisation du TD :**
    - **Document d'audit avec explication de vos recherches**
    - **Programmes Python de l'ensemble des exercices**
    - **Vous devez préparer une présentation pour expliquer l'ensemble des solutions que vous proposer pour réaliser ces exercices**
  - **Modalité de remise du TD :**
    - Votre .zip contiendra obligatoirement :
      - L'ensemble des programmes Python dans **jupyter-lab au format « ipynb »**
        - Votre ipynb devra contenir obligatoirement:
          - Votre nom et prénom, N° du TD, N°de l'exercice
          - Question et réponse à l'exercice
          - Code Python, une fonction par cellule avec commentaires
          - ...

- **Objectif - Forensics sur les mails**

- L'e-mail est un des vecteurs les plus utilisés par les attaquants pour transmettre un logiciel malveillant. Vous avez sûrement déjà reçu un e-mail contenant une facture à payer ou vous informant que vous avez été sélectionné pour toucher l'héritage d'une personne très riche ? Ce type d'e-mail est fréquent et la plupart du temps exploite le vecteur humain.

- L'envoi d'e-mail malveillant est une technique de social engineering, c'est-à-dire qu'il n'exploite pas une faille logicielle mais une faille humaine.

- Il se peut que votre adresse e-mail se retrouve dans une liste utilisée pour du spam, mais dans certains cas cet e-mail peut être ciblé, c'est à dire adapté à vous pour vous inciter encore plus à ouvrir la pièce jointe.

    - **Rappel des commandes pour Jupiter-lab**
        - le « **!** » permet de lancer une commande
            - par exemple :
                - !ping 8.8.8.8 (dns de google)
                - !ping google.fr
                - le « # » pour les commentaires

- **Généralité - Lecture du mail et analyse**

- **EML, abréviation de courrier électronique ou e-mail, est une extension de fichier pour un message électronique enregistré dans un fichier du protocole Internet Message Format pour les messages électroniques. C'est le format standard utilisé par Microsoft Outlook Express ainsi que certains autres programmes de messagerie. Étant donné que les fichiers EML sont créés pour se conformer à la norme industrielle RFC 5322, les fichiers EML peuvent être utilisés avec la plupart des clients de messagerie, des serveurs et des applications. Voir IMF (Internet Message Format) pour une description de la syntaxe du message.**

https://www.loc.gov/preservation/digital/formats/fdd/fdd000393.shtml

- **Les fichiers EML stockent généralement chaque message dans un seul fichier (contrairement à MBOX qui concatène tous les messages d'un dossier dans un seul fichier), et les pièces jointes peuvent être incluses en tant que contenu MIME dans le message ou supprimées en tant que fichier séparé, référencé à partir d'un marqueur dans le fichier EML.**

1. Exercices - Lecture mail et analyse

En vous aidant de la documentation sur DVO
ATTENTION : l'ensemble des Exercices doit être effectué via Jupiter-lab en python
Pour réaliser ces exercices, vous pouvez utiliser des bibliothèques de traitement d'e-mails : "email", GeoIP2 en Python. Ainsi que votre code python code64

1.1 Exo1.1 : Expliquer l'importance de la sécurisation des e-mails et les risques liés aux attaques par e-mail.

1.2 Exo1.2 : Détection de phishing :
- Donner des exemples de Détection de phishing
  - Proposer un système de détection de phishing en utilisant des techniques d'apprentissage automatique pour identifier les messages suspects.

1.3 Exo1.3 :Écrire un programme python qui Analyse les métadonnées :
- Les métadonnées de l'e-mail peuvent fournir des informations précieuses sur l'expéditeur, le destinataire et le contenu du message. Analyser les métadonnées peut aider à détecter les e-mails suspects, tels que les spams ou les e-mails malveillants. voici une liste non exhaustive des métadonnées courantes qui peuvent être trouvées dans un e-mail :
  - **De : adresse e-mail de l'expéditeur**
  - **À : adresse e-mail du destinataire**
  - **Cc : adresse e-mail du destinataire en copie**
  - **Cci (ou Bcc) : adresse e-mail du destinataire en copie cachée**
  - **Date : date et heure d'envoi de l'e-mail**
  - **Sujet : objet de l'e-mail**
  - **Message-ID : identifiant unique de l'e-mail**
  - **In-Reply-To : identifiant de l'e-mail auquel le message répond**
  - **Références : liste des identifiants des e-mails liés**
  - **Importance : niveau d'importance de l'e-mail**
  - **Priorité : niveau de priorité de l'e-mail**
  - **Confidentialité : niveau de confidentialité de l'e-mail**
  - **MIME-Version : version de MIME utilisée pour l'e-mail**
  - **Content-Type : type de contenu du message**
  - **Content-Transfer-Encoding : méthode d'encodage utilisée pour les parties du message**
  - **User-Agent : nom et version du client de messagerie utilisé par l'expéditeur**
  - **X-Mailer : nom et version du logiciel utilisé pour envoyer l'e-mail**
  - **X-Originating-IP : adresse IP de l'ordinateur qui a envoyé l'e-mail**
  - **X-Spam-Status : statut de spam de l'e-mail**

1.4 Exo1.4 :Écrire un programme python qui décode le fichier de mail **2023S1_C06_P00_MAIL_Mail1.eml** sur DVO
- En extraire le maximum de renseignements pour ce message .
- En déduire l'origine de ce mail

1.5 .Exo1.5 : Vérification de l'identité de l'expéditeur en python :

- Élaborer un système de vérification de l'identité de l'expéditeur pour empêcher les e-mails frauduleux d'atteindre leur destinataire.
- Pour la vérification de l'identité de l'expéditeur, vous pouvez utiliser les algorithmes tels que SPF, DKIM et DMARC.
  - Expliquer chacun de ces algorithmes.
  - Code ces algorithme en python pour la vérification de l'expéditeur

1.6 Exo1.6 :Expliquer comment l'on peut déduire si ce mail est un spam

- Écrire une fonction python qui permettra de savoir si mail est un spam ou pas

2. **- Challenge 5**

2.1 Écrire un code python qui lit et analyse le fichier « challenge5.zip »
2.2 Écrire un code python qui donne la géolocalisation « graphique » en python de l'émetteur et du destinataire