SAÉ1.05 : Traiter des données

...

Llascer SIMBA

Grp : B2

Contexte

Contexte Global :

- Employé dans une entreprise ayant un site d'administration en France.
- Site de production en Inde, personnel anglophone.

Problème : Une défaillance réseau à été signalée en Inde.

Objectif :
Trouver les anomalies

Travail réalisé

- Lecture du fichier txt fournis
- Extraction des informations importantes
- Création d’un fichier csvs
- Création d’un fichier markdown
- Création d’une page html

Lecture du fichier

```python
def lire_fichier(fichier):
    try:
        with open(fichier, 'r', encoding='utf-8') as file:
            lignes = file.readlines()
            print(f"{len(lignes)} lignes ont été lues du fichier '{fichier}'.")
            return lignes
    except FileNotFoundError:
        print(f"Erreur : Le fichier '{fichier}' est introuvable.")
        return []
    except Exception as e:
        print(f"Erreur lors de la lecture du fichier : {e}")
        return []
```

Extractions des informations importantes

```python
def extraire_donnees(lignes):
    donnees = []
    # Motif regex pour capturer les informations pertinentes
    pattern = re.compile(r"(\d{2}:\d{2}:\d{2}\.\d+)\sIP\s([\w\d\-\.]+)\s>\s([\d\.]+)\.(\d+):\sFlags\s\[(.*?)\].*
    print("Extraction des données en cours...")

    for ligne in lignes:
        ligne = ligne.strip()
        if not ligne.startswith("IP") and not ligne.startswith("11"):  # Ignore les lignes non pertinentes
            continue

        match = pattern.search(ligne)
        if match:
            donnees.append([
                match.group(1),  # Timestamp
                match.group(2),  # IP Source
                match.group(3),  # IP Destination
                match.group(4),  # Port Destination
                match.group(5),  # Flags
                match.group(6)   # Length
            ])
    print(f"{len(donnees)} correspondances trouvées.")
    return donnees
```

Création du fichier csv

```python
# Écriture dans un fichier CSV
def ecrire_csv(donnees, nom_csv):
    try:
        with open(nom_csv, "w", newline="") as fichier_csv:
            writer = csv.writer(fichier_csv)
            writer.writerow(["Timestamp", "IP Source", "IP Destination", "Port Destination", "Flags", "Length"])
            writer.writerows(donnees)
        print(f"Fichier CSV généré avec succès : {nom_csv} ({len(donnees)} lignes).")
    except PermissionError:
        print(f"Erreur : Impossible d'écrire dans le fichier '{nom_csv}'. Assurez-vous qu'il n'est pas ouvert.")
    except Exception as e:
        print(f"Erreur lors de l'écriture du fichier CSV : {e}")
```

Création du fichier Markdown

```python
# Génération d'un fichier Markdown
def generer_markdown(donnees, fichier_md):
    try:
        markdown = "# Rapport des Paquets Réseau\n\n"
        markdown += "| Timestamp       | IP Source      | IP Destination   | Port Destination | Flags | Length |
        markdown += "|-----------------|----------------|------------------|------------------|-------|--------|

        for donnee in donnees:
            markdown += f"| {donnee[0]} | {donnee[1]} | {donnee[2]} | {donnee[3]} | {donnee[4]} | {donnee[5]} |\

        with open(fichier_md, "w", encoding="utf-8") as fichier:
            fichier.write(markdown)
        print(f"Fichier Markdown généré avec succès : {fichier_md} ({len(donnees)} lignes).")
    except Exception as e:
        print(f"Erreur lors de la génération du fichier Markdown : {e}")
```

Exel

Timestamp	IP Source	IP Destinatior	Port Destinat	Flags	Length
11:42:04.766	BP-Linux8.ss	192.168.190	50019	P.	36
11:42:04.766	BP-Linux8.ss	192.168.190	50019	P.	36
11:42:06.669	BP-Linux8.ss	192.168.190	50245	P.	36
11:42:06.681	BP-Linux8.ss	192.168.190	50245	P.	36
11:42:06.711	BP-Linux8.ss	192.168.190	50245	P.	36
11:42:12.685	BP-Linux8.ss	192.168.190	50245	P.	36
11:42:12.689	BP-Linux8.ss	192.168.190	50245	P.	36
11:42:12.689	BP-Linux8.ss	192.168.190	50245	P.	36
11:42:32.865	BP-Linux8.ss	192.168.190	50374	P.	36
11:42:06.712	BP-Linux8.ss	192.168.190	50245	P.	68
11:42:12.689	BP-Linux8.ss	192.168.190	50245	P.	92
11:42:12.689	BP-Linux8.ss	192.168.190	50245	P.	100
11:42:13.688	BP-Linux8.ss	192.168.190	50245	P.	100
11:42:14.688	BP-Linux8.ss	192.168.190	50245	P.	100
11:42:15.688	BP-Linux8.ss	192.168.190	50245	P.	100
11:42:16.688	BP-Linux8.ss	192.168.190	50245	P.	100
11:42:17.688	BP-Linux8.ss	192.168.190	50245	P.	100
11:42:18.726	BP-Linux8.ss	192.168.190	50245	P.	100
11:42:19.726	BP-Linux8.ss	192.168.190	50245	P.	100
11:42:20.725	BP-Linux8.ss	192.168.190	50245	P.	100

resultats

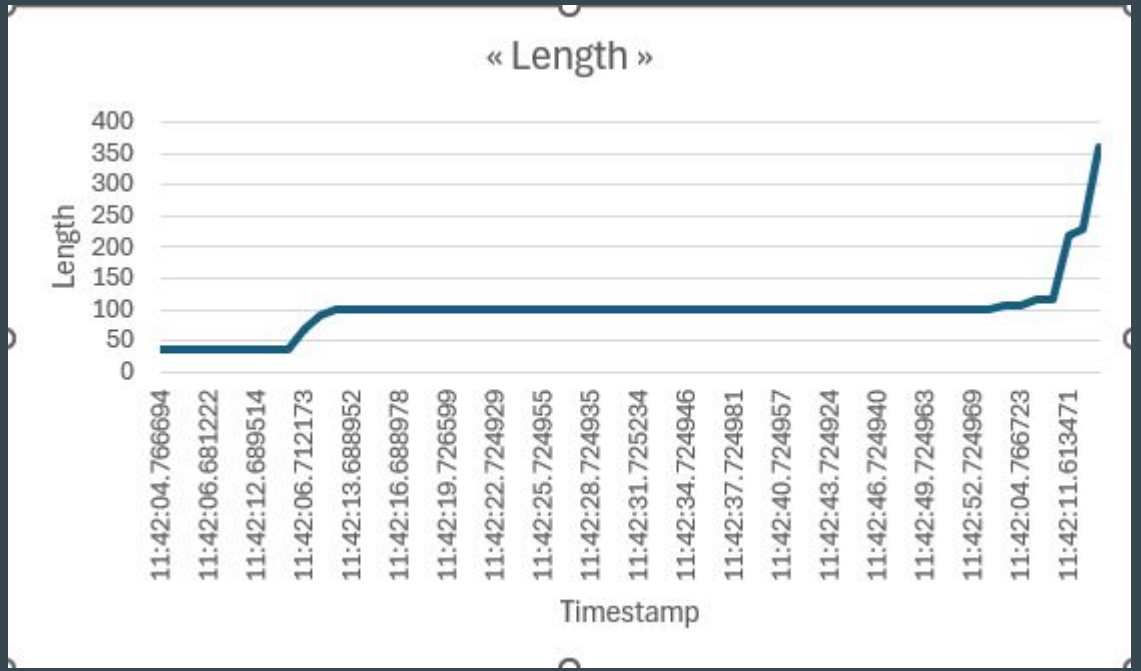
Création d'un Tableau

-Tableau créé pour pouvoir gérer chaque colonnes

Timestamp	IP Source	IP Destination	Port Destination	Flags	Length
11:42:04.766	BP-Linux8.ss	192.168.190.13(	50019	P.	36
11:42:04.766	BP-Linux8.ss	192.168.190.13(	50019	P.	36
11:42:06.669	BP-Linux8.ss	192.168.190.13(	50245	P.	36
11:42:06.681	BP-Linux8.ss	192.168.190.13(	50245	P.	36
11:42:06.711	BP-Linux8.ss	192.168.190.13(	50245	P.	36
11:42:12.685	BP-Linux8.ss	192.168.190.13(	50245	P.	36
11:42:12.689	BP-Linux8.ss	192.168.190.13(	50245	P.	36
11:42:12.689	BP-Linux8.ss	192.168.190.13(	50245	P.	36
11:42:32.865	BP-Linux8.ss	192.168.190.13(	50374	P.	36
11:42:06.712	BP-Linux8.ss	192.168.190.13(	50245	P.	68
11:42:12.689	BP-Linux8.ss	192.168.190.13(	50245	P.	92
11:42:12.689	BP-Linux8.ss	192.168.190.13(	50245	P.	100
11:42:13.688	BP-Linux8.ss	192.168.190.13(	50245	P.	100
11:42:14.688	BP-Linux8.ss	192.168.190.13(	50245	P.	100
11:42:15.688	BP-Linux8.ss	192.168.190.13(	50245	P.	100
11:42:16.688	BP-Linux8.ss	192.168.190.13(	50245	P.	100
11:42:17.688	BP-Linux8.ss	192.168.190.13(	50245	P.	100
11:42:18.726	BP-Linux8.ss	192.168.190.13(	50245	P.	100
11:42:19.726	BP-Linux8.ss	192.168.190.13(	50245	P.	100
11:42:20.725	BP-Linux8.ss	192.168.190.13(	50245	P.	100

Graphiques générés

« Port Destination » : 50245 représente la majorité des « Length ».



Conclusion

-La majorité des paquets sont d'une taille de 100

-36, 108, 116, 220, 228, 360

Paquets fragmenté ou incomplets

Changement de port de destination

-50019

-Suspect