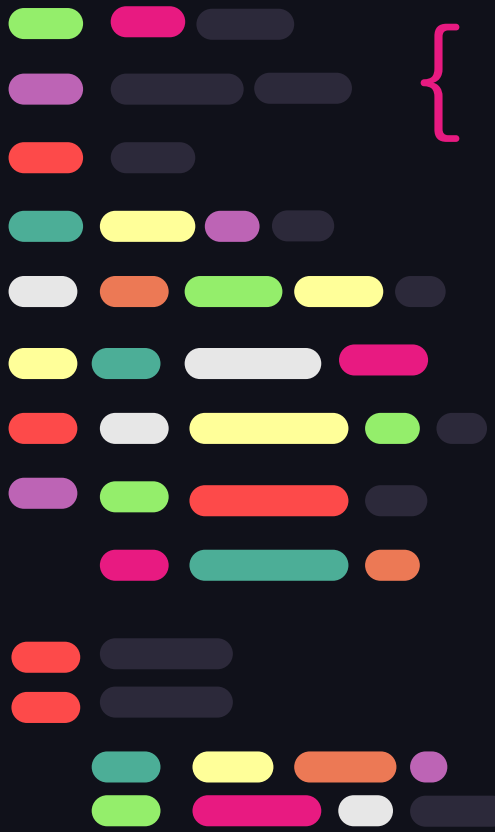


Présentation de la SAÉ 1.05 à l'IUT de Roanne

Pierre Hiltenbrand-Burianne

Sommaire

Problématique	Comment identifier le problème ?
Le but du code	Quel est l'objectif de créer ce code ?
Mise en place du cahier des charges	Quelles seront les étapes nécessaires à la création du code ?
Explication du programme	Comment fonctionne le programme ?
Utilisation du logiciel n'importe où	Comment accéder au logiciel à tout moment et depuis n'importe quel PC ?
Démonstration du programme	Quel est le résultat final ?

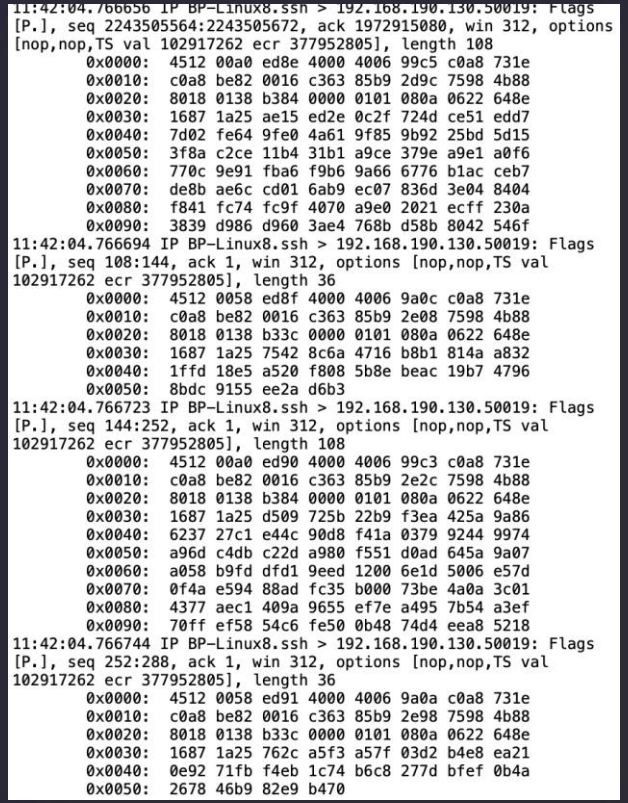
01 { ..

Comment identifier le problème ?

} ..

Ce que nous avons récupéré

Fichier texte avec une capture de réseau brute.

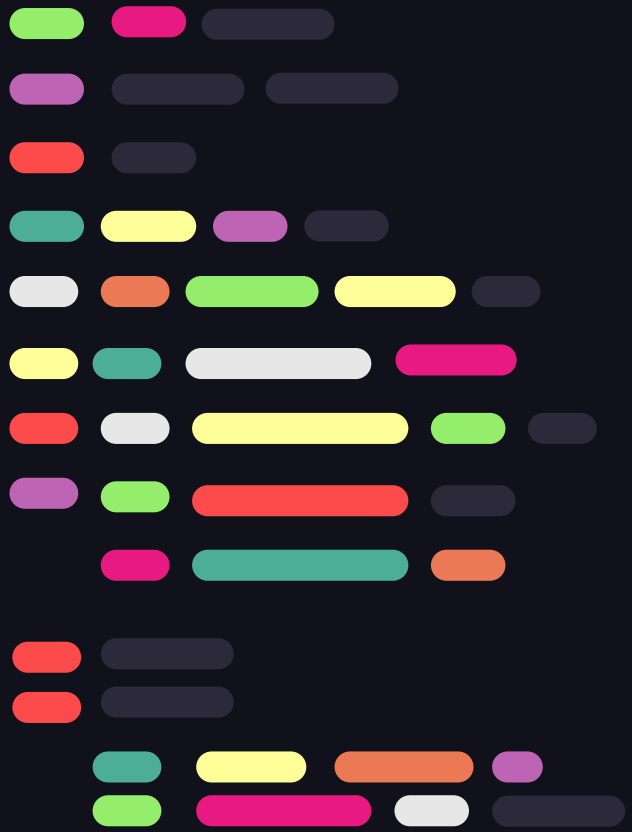


```
11:42:04.766656 IP BP-Linux8.ssh > 192.168.190.130.50019: Flags
[P.], seq 2243505564:2243505672, ack 1972915080, win 312, options
[nop,nop,TS val 102917262 ecr 377952805], length 108
        0x0000:  4512 00a0 ed8e 4000 4006 99c5 c0a8 731e
        0x0010:  c0a8 be82 0016 c363 85b9 2d9c 7598 4b88
        0x0020:  8018 0138 b384 0000 0101 080a 0622 648e
        0x0030:  1687 1a25 ae15 ed2e 0c2f 724d ce51 edd7
        0x0040:  7d02 fe64 9fe0 4a61 9f85 9b92 25bd 5d15
        0x0050:  3f8a c2ce 11b4 31b1 a9ce 379e a9e1 a0f6
        0x0060:  770c 9e91 fba6 f9b6 9a66 6776 b1ac ceb7
        0x0070:  de8b ae6c cd01 6ab9 ec07 836d 3e04 8404
        0x0080:  f841 fc74 fc9f 4070 a9e0 2021 ecff 230a
        0x0090:  3839 d986 d960 3ae4 768b d58b 8042 546f
11:42:04.766694 IP BP-Linux8.ssh > 192.168.190.130.50019: Flags
[P.], seq 108:144, ack 1, win 312, options [nop,nop,TS val
102917262 ecr 377952805], length 36
        0x0000:  4512 0058 ed8f 4000 4006 9a0c c0a8 731e
        0x0010:  c0a8 be82 0016 c363 85b9 2e08 7598 4b88
        0x0020:  8018 0138 b33c 0000 0101 080a 0622 648e
        0x0030:  1687 1a25 7542 8c6a 4716 b8b1 814a a832
        0x0040:  1ffd 18e5 a520 f808 5b8e beac 19b7 4796
        0x0050:  8bdc 9155 ee2a d6b3
11:42:04.766723 IP BP-Linux8.ssh > 192.168.190.130.50019: Flags
[P.], seq 144:252, ack 1, win 312, options [nop,nop,TS val
102917262 ecr 377952805], length 108
        0x0000:  4512 00a0 ed90 4000 4006 99c3 c0a8 731e
        0x0010:  c0a8 be82 0016 c363 85b9 2e2c 7598 4b88
        0x0020:  8018 0138 b384 0000 0101 080a 0622 648e
        0x0030:  1687 1a25 d509 725b 22b9 f3ea 425a 9a86
        0x0040:  6237 27c1 e44c 90d8 f41a 0379 9244 9974
        0x0050:  a96d c4db c22d a980 f551 d0ad 645a 9a07
        0x0060:  a058 b9fd dfd1 9eed 1200 6e1d 5006 e57d
        0x0070:  0f4a e594 88ad fc35 b000 73be 4a0a 3c01
        0x0080:  4377 aec1 409a 9655 ef7e a495 7b54 a3ef
        0x0090:  70ff ef58 54c6 fe50 0b48 74d4 eea8 5218
11:42:04.766744 IP BP-Linux8.ssh > 192.168.190.130.50019: Flags
[P.], seq 252:288, ack 1, win 312, options [nop,nop,TS val
102917262 ecr 377952805], length 36
        0x0000:  4512 0058 ed91 4000 4006 9a0a c0a8 731e
        0x0010:  c0a8 be82 0016 c363 85b9 2e98 7598 4b88
        0x0020:  8018 0138 b33c 0000 0101 080a 0622 648e
        0x0030:  1687 1a25 762c a5f3 a57f 03d2 b4e8 ea21
        0x0040:  0e92 71fb f4eb 1c74 b6c8 277d bfef 0b4a
        0x0050:  2678 46b9 82e9 b470
```

02 { ..

Quel est l’objectif de créer ce code ?

} ..



Analyser rapidement!

< Le code permet d’effectuer des tâches avec un rapidité des millions de fois supérieur à celle de l’homme >

Pourquoi ce langage ?

Langage Critères	Temps de dev	Analyse du Texte	Interface Graphique	Visualisation
Python	**100 %**	70 %	**90 %**	**85 %**
C	40 %	**100 %**	50 %	40 %
JavaScript	80 %	85 %	70 %	75 %

03 { ..

Mise en place du cahier des charges

} ..

Cahier des charges

Extraire les informations

Afficher une page WEB

Exporter un fichier CSV

04 { ..

Fonctionnement du Programme

} ..

Analyse fichier TCP dump

Extraction avec expression régulière Regex

```python
def parse_text_file_to_dataframe(file_path):


    main_info_pattern = re.compile(
        r"(?P<timestamp>\d{2}:\d{2}:\d{2}\.\d+)\s+IP\s+(?P<src>[\w\.\-]+)\.(?P<s
    )

    with open(file_path, "r") as file:
        file_content = file.read()

    rows = []
    for match in main_info_pattern.finditer(file_content):
        rows.append(match.groupdict())

    df = pd.DataFrame(rows)
    return df
```

Rapport Markdown et HTML

Rapport Markdown

```python
def generate_markdown_and_html_report(df):

    timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

    # Construction du contenu en Markdown
    md_content = []
    md_content.append("# Network Traffic Analysis Report\n")
    md_content.append(f"**Generated on**: {timestamp}\n\n")
    md_content.append("---\n\n")



    md_content.append("## Overview\n\n")
    md_content.append(f"- **Total packets captured**: {len(df)}\n")
    md_content.append(f"- **Unique source IPs**: {df['src'].nunique()}\n")
    md_content.append(f"- **Unique destination IPs**: {df['dst'].nunique()}\n\n")
```

Conversion Markdown vers HTML

```python
    html_content = f"""<!DOCTYPE html>

<html>
<head>
  <meta charset="utf-8">
  <title>Network Analysis Report</title>
  {css_style}
</head>
<body>
{body_html}
</body>
</html>
"""



    report_path = os.path.join(os.path.expanduser('~'), 'network_analysis_report.html')
    with open(report_path, 'w', encoding='utf-8') as f:
        f.write(html_content)

    return report_path
```

Enregistrement CSV

Exportation en CSV

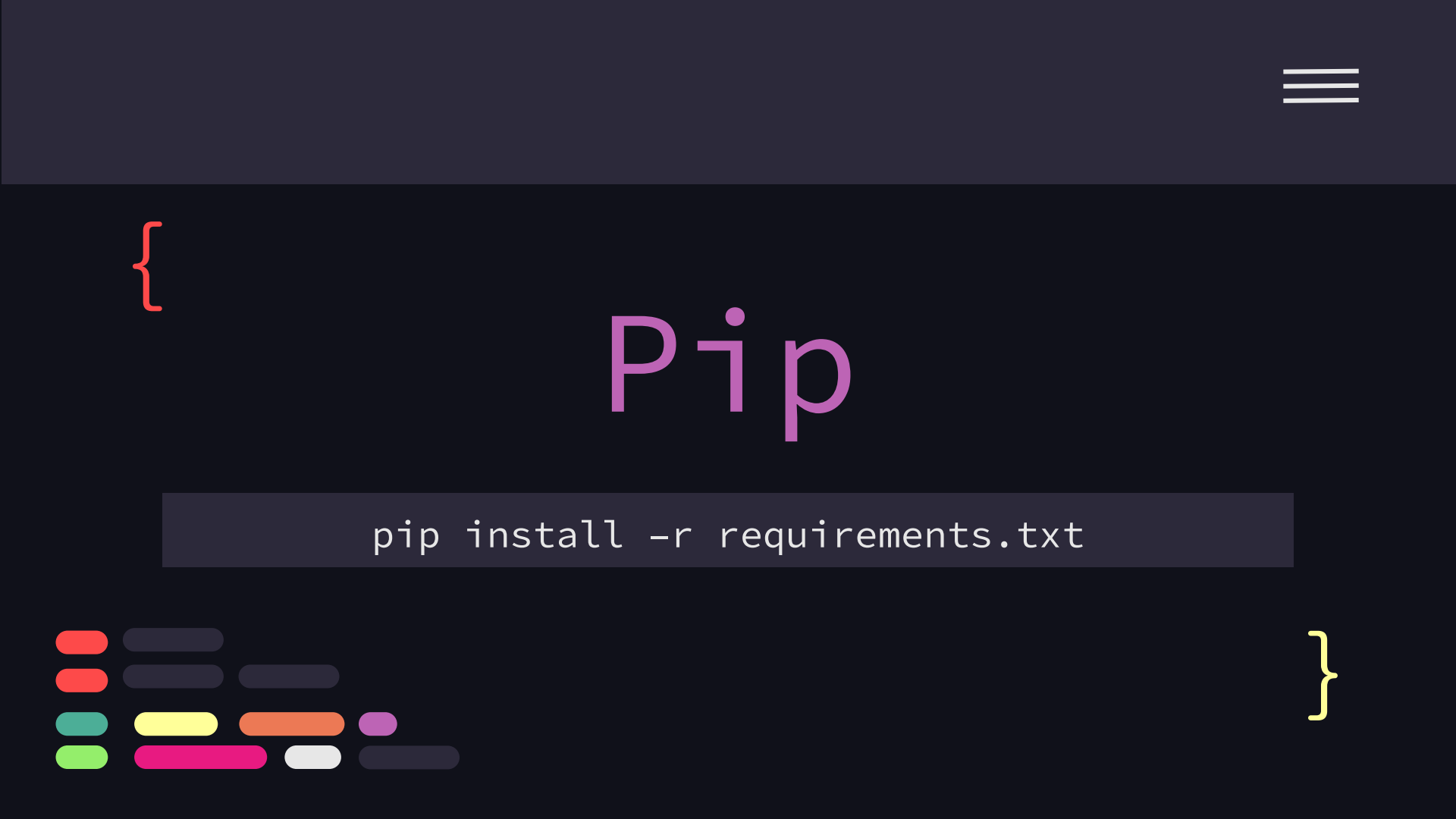
```python
def export_to_csv():

    export_file_path = asksaveasfilename(
        defaultextension='.csv',
        filetypes=[("CSV Files", "*.csv"), ("All Files", "*.*")],
        title="Export data as CSV"
    )
    if export_file_path:
        df.to_csv(export_file_path, index=False, sep=';', encoding='utf-8')
        tk.messagebox.showinfo("Success", "Data exported successfully !")
```

05 { ..

Utilisation n’importe où du programme

} ..


Git
git clone https://github.com/pierreburnn/SA-05


Pip
pip install -r requirements.txt

Python

```
python Programme.py
```

06 { ..

Démonstration du programme

} ..