Clément DURECU / ING3 – Cyber Groupe A

RAPPORT DE RESTITUTION DES RÉSULTATS DE LA CAMPAGNE DE PHISHING DU 19/09

L'ensemble des données que l'on retrouve dans ce rapport sont factices et ont été générées pour construire cette template.

GÉNÉRATION DES DONNÉES FACTICES

- **Les données ont été créées à l'aide d'un script python "csv_generate.py" en s'inspirant du csv fourni par l'outil GoPhish à l'issu d'une campagne.**
- **Le fichier .csv généré comporte plusieurs colonnes correspondant aux attributs de chaque utilisateur présent dans la campagne, chacun caractérisé par une ligne.**
- **Le script s'occupe également d'afficher les résultats dans le terminal ainsi que de fournir deux graphiques permettant des les afficher de façon plus claire.**

GÉNÉRATION DES DONNÉES FACTICES

- **Le script est découpé en plusieurs parties :**

  1) On commence par définir le nombre de valeurs, une liste de postes, de faux noms et enfin les statuts ainsi que les proportions permettant de les répartir.

```python
n = 113
statuses = (['Email Sent'] * int(n * 0.05) +
            ['Email Opened'] * int(n * 0.75) +
            ['Clicked Link'] * int(n * 0.10) +
            ['Submitted Data'] * (n - int(n * 0.05) - int(n * 0.75) - int(n * 0.10)))
random.shuffle(statuses)

# Fonction pour générer un ID aléatoire
def random_id(length=5):
    chars = string.ascii_letters + string.digits
    return ''.join(random.choices(chars, k=length))

# Fonction pour générer une IP aléatoire
def random_ip():
    return ".".join(str(random.randint(1, 255)) for _ in range(4))

# On crée une liste de faux noms, prénoms et postes
first_names = ["Lucas", "Emma", "Nathan", "Léa", "Sophie", "Julien", "Marie", "Thomas", "Clara", "Hugo", "Paul"]
last_names = ["Martin", "Bernard", "Dubois", "Durand", "Lefevre", "Moreau", "Simon", "Laurent", "Garcia", "Michel"]
positions = ["Développeur", "RH", "Commercial", "Comptable", "Chef de projet", "Stagiaire", "Technicien", "Directeur"]
base_date = datetime.now()
```

GÉNÉRATION DES DONNÉES FACTICES

2) On remplit un fichier CSV avec les valeurs aléatoires en respectant ce que l'on vient de définir puis on sauvegarde les valeurs.

```
rows = []
for status in statuses:
    send_dt = base_date + timedelta(seconds=random.randint(0, 3600))
    modified_dt = send_dt + timedelta(seconds=random.randint(0, 120))
    first = random.choice(first_names)
    last = random.choice(last_names)
    email = f"{first.lower()}.{last.lower()}@entreprise.com"
    position = random.choice(positions)
    ip = random_ip() if status in ["Email Opened", "Clicked Link", "Submitted Data"] else ""
    rows.append({
        "id": random_id(),
        "status": status,
        "ip": ip,
        "latitude": round(random.uniform(-90, 90), 6) if ip else 0,
        "longitude": round(random.uniform(-180, 180), 6) if ip else 0,
        "send_date": send_dt.isoformat() + "Z",
        "reported": random.choice(["False", "False", "False", "True"]),
        "modified_date": modified_dt.isoformat() + "Z",
        "email": email,
        "first_name": first,
        "last_name": last,
        "position": position
})
```

```
# On sauvegarde le fichier CSV généré
data_file = "phishing_results.csv"
fieldnames = rows[0].keys()
with open(data_file, "w", newline='', encoding="utf-8") as f:
    writer = csv.DictWriter(f, fieldnames=fieldnames)
    writer.writeheader()
    writer.writerows(rows)

file_path = "phishing_results.csv"
```

GÉNÉRATION DES DONNÉES FACTICES

3) On affiche le résultat de la campagne dans la console.

```
nb_email_sent = 0
nb_email_opened = 0
nb_clicked_link = 0
nb_submitted_data = 0

with open(file_path, newline='', encoding='utf-8') as csvfile:
    reader = csv.DictReader(csvfile)
    for row in reader:
        status = row['status'].strip().lower()

        if status in ('email sent', 'email opened', 'clicked link', 'submitted data'):
            nb_email_sent += 1
        if status in ('email opened', 'clicked link', 'submitted data'):
            nb_email_opened += 1
        if status in ('clicked link', 'submitted data'):
            nb_clicked_link += 1
        if status == 'submitted data':
            nb_submitted_data += 1
```

GÉNÉRATION DES DONNÉES FACTICES

4) Enfin, on génère deux graphiques : un avec des valeurs précises et un en pourcentage.

```python
labels = ["Emails envoyés", "Mails ouverts", "Liens cliqués", "Données soumises"]
values = [nb_email_sent, nb_email_opened, nb_clicked_link, nb_submitted_data]

plt.figure(figsize=(8, 5))
colors = ["#4CAF50", "#2196F3", "#FFC107", "#F44336"]
bars = plt.bar(labels, values, color=colors, width=0.6)
for bar, value in zip(bars, values):
    plt.text(bar.get_x() + bar.get_width()/2, bar.get_height(),
             str(value), ha='center', va='bottom', fontsize=10, fontweight='bold')

plt.title("Résultats de la campagne", fontsize=14, fontweight='bold')
plt.xlabel("Statuts", fontsize=12)
plt.ylabel("Nombre d'occurrences", fontsize=12)
plt.xticks(rotation=15, fontsize=10)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

```python
nb_not_opened = max(nb_email_sent - nb_email_opened, 0)
labels_pie = ["Non lus", "Mails ouverts", "Liens cliqués", "Données soumises"]
values_pie = [nb_not_opened, nb_email_opened, nb_clicked_link, nb_submitted_data]
colors_pie = ["#9E9E9E", "#2196F3", "#FFC107", "#F44336"]

plt.figure(figsize=(6, 6))
wedges, texts, autotexts = plt.pie(
    values_pie,
    labels=labels_pie,
    colors=colors_pie,
    autopct=lambda p: f'{p:.1f}%' if p > 0 else '',
    startangle=90,
    wedgeprops={'edgecolor': 'white'}
)

plt.title("Répartition des utilisateurs", fontsize=14, fontweight='bold')
plt.tight_layout()
plt.show()
```

CONTEXTE DE LA CAMPAGNE

- **Menée sur l'ensemble des membres d'une PME de 113 personnes.**
- **Motivée par une envie de vérifier si la campagne de sensibilisation au phishing réalisée quelques semaines plus tôt a été efficace.**
- **Lancée le 19/09 à 14h30.**
- **Mail de phishing envoyé :**

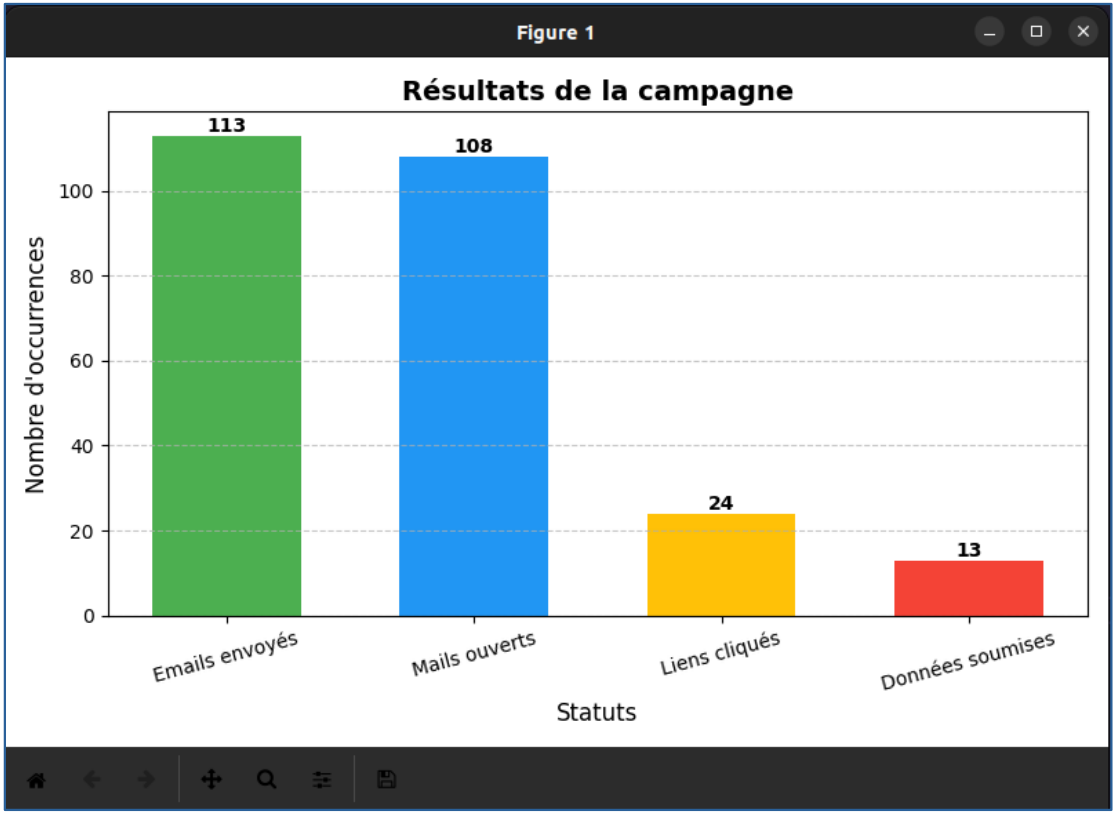

À moi

Bonjour Clément,

En raison d'une maintenance prévue demain entre 15h30 et 16h00, nous aurions besoin des identifiants de votre machine afin d'éviter d'importuner votre travail au cours de cette dernière.
Nous vous prions donc de cliquer sur ce lien et de remplir le formulaire : http://192.168.1.47:8000/click/2

Cordialement,
Jean-Claude Célestin, directeur du pôle IT.

RÉSULTATS DE LA CAMPAGNE

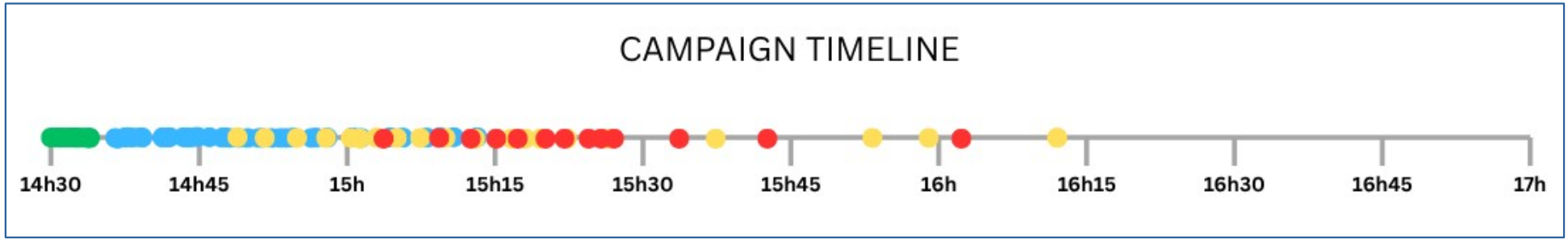


Ce que l'on constate sur ce premier graphique :

- Sur 113 mails envoyés, 108 ont été ouverts. Les 5 restants étant deux mails envoyés à des adresses inactives et trois à des personnes en congés.

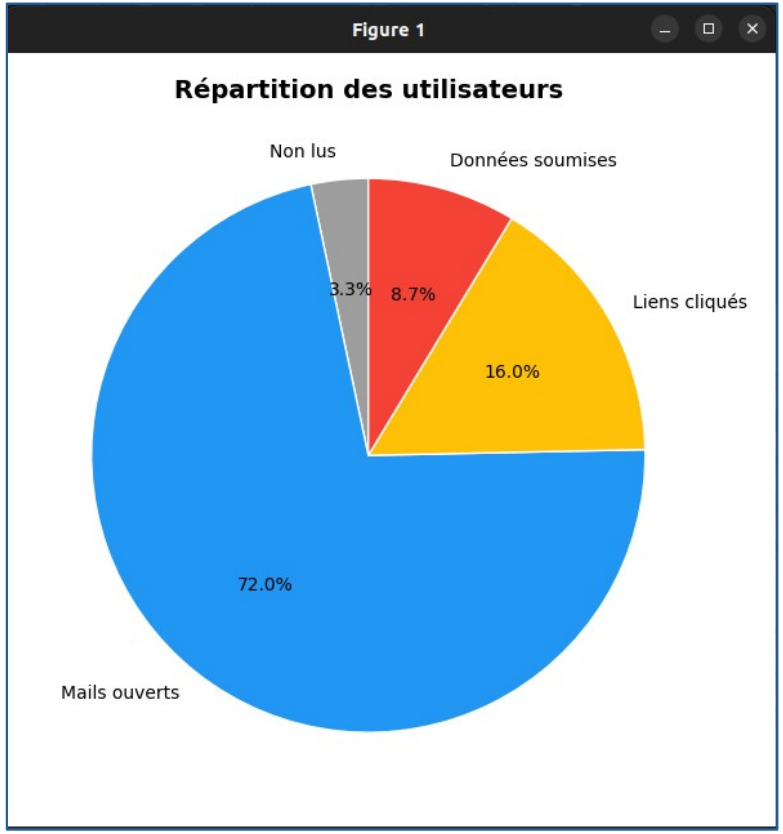
- Sur les 108 mails ouverts, 24 personnes ont cliqué sur le lien et 13 sont allées jusque remplir le formulaire avec leurs informations personnelles.

RÉSULTATS DE LA CAMPAGNE



- **Nous remarquons premièrement que les mails ont rapidement été ouverts, quasiment tous avant 15h soit moins de 30 minutes après le lancement.**
- **Deuxièmement, il semblerait que beaucoup moins de personnes aient cliqué sur le lien ou rempli le formulaire à partir de 15h30. Cela correspond d'après notre taupe au moment où l'un des DRH a signalé le mail pour la première fois et a averti les équipes.**
- **Fin de la campagne à 16h13 lors de la dernière intéraction avec le mail.**

RÉSULTATS DE LA CAMPAGNE



Ce que l'on constate sur ce second graphique :

- Sur l'effectif ciblé de 113 personnes, 72% ont seulement ouvert le mail sans aller plus loin qu'une simple lecture

- 3.3% de l'effectif n'a pas ouvert le mail ; ici en raison d'anciennes boîtes mails ou bien de congés prises par les salariés

- 16% ont cliqué sur le lien fourni dans le mail de phishing

- 8.7% ont soumis des données personnelles dans le formulaire que le mail demandait de compléter

CONCLUSION DE LA CAMPAGNE

- **À l'issu de la campagne, nous pouvons donc en conclure que la campagne de sensibilisation menée plus tôt n'a pas été suffisante.**
- **En effet, sur un effectif de 113 personnes, 37 agissent de façon potentiellement dangereuse (clic sur le lien ou remplissage du formulaire) pour la société ce qui représente 24.7% de l'entreprise.**
- **De plus, il est important de noter que sans l'intervention du DRH à 15h30 signalant à tout le monde le caractère suspect du mail, le phishing aurait pu obtenir encore plus de données.**
- **À l'avenir, il faudrait insister sur les conséquences que pourraient avoir des comportements similaires sur de vrais mails de phishing pour que les collaborateurs réalisent la potentielle gravité de leurs actes.**