Mise en place du showroom de l'ISIMA
Vol de données

Mardi 04 janvier 2022

Clermont Auvergne INP – ISIMA

Revue de projet – ZZ2F5 – 2021/2022

Benjamin LE CESNE

Antoine LEFEBVRE

PRÉSENTATION DU SUJET

- Qu'est ce qui peut intéresser les lycéens dans l'informatique ?
  - Hacker ses professeurs
  - S'introduire sur une machine et voler des données

- À distance avec la mise en place d'un remote shell basique
- En physique avec une clé USB

- Donner envie aux lycéens
- Servir de sensibilisation pour les professeurs

DÉMONSTRATION ET SCÉNARISATION

- Démonstration possible
  - Voler des fichiers
  - Modifier des fichiers
  - KeyLogger sur Rubber Ducky
- Scénario envisagé
  - Récupérer le sujet d'un contrôle
  - Modifier sa copie après rendu
  - Voler le mot de passe ProNote / Ent d'un professeur

CONTENU TECHNIQUE

- Langage : Python, modèle objet

- Rubber Ducky

```python
class Protocol_TCP_server:
    def __init__(self):
        self.server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self._client_connected = False
        self._server_connected = False
        self.data_size = PACKET_SIZE - HEADER_SIZE

    def start_server(self, address, port):
        self.server.bind((address, port))
        self.server.listen(1)
        self._server_connected = True
        self.connect_client()

    def connect_client(self):
        print("Waiting connection")
        client, address = self.server.accept()
        self.client = client
        self._client_connected = True

    def disconnect_client(self):
        self.client.close()
        self._client_connected = False

    def close_server(self):
        self.server.close()
        self._server_connected = False

    def is_client_connected(self):
        return self._client_connected

    def is_server_connected(self):
        return self._server_connected

    def send_file(self, file_name, dest_path):
        size = os.path.getsize(file_name)
        self.send_msg(f"{dest_path}/{file_name};{size}")
        file = open(file_name, "rb")
        if size > 1024:
            num = 0
            pourcent = 0
            nb = ceil(size / 1024)
            for i in range(nb):
                tmp = i / nb * 100 // 10
                if pourcent != tmp:
                    pourcent = tmp
                    print(f"{pourcent * 10}%")
                file.seek(num, 0)
                data = file.read(1024)
                self.client.send(data)
                num += 1024
        else:
            data = file.read()
            self.client.send(data)
        file.close()
```

PLANNING

Passé		À venir	
Octobre	Réflexion du scénario Choix technique Demande de matériel	Janvier	Fin du développement technique
Novembre	Affinage du projet Début développement technique	Février	Tests Petites retouches Rédaction rapport
Décembre	Avancée technique Prise de recul sur le projet	Mars	Préparation soutenance

TRAVAIL À TERMINER / DIFFICULTÉS

- Travail réalisé
  - Éditer les métadonnées
  - Établir une connexion TCP entre un serveur et un client
  - Envoyer et recevoir des fichiers dans les deux sens
- Travail à terminer
  - Développer le KeyLogger via la Rubber Ducky
  - Tester
- Difficultés
  - Trouver un scénario qui nous corresponde
  - Exécution automatique des clés
  - Les métadonnées