

# SRES-Threat

Alexis Callemard, Tanguy Becam, Hervé Hammond, Maël Nogues

7 février 2018

## 1 Objectifs

L'objectif de notre solution est de repérer les sites de hammeçonnage ciblé <sup>1</sup> sur les sites gouvernementaux.

## 2 Librairies et services utilisés

### 2.1 Sources et fichiers

Nous utilisons un dictionnaire de mots clés en plusieurs langue, à élargir en fonction du nombre de matchs possibles (assemblée, ambassade, ministère, impôt, président, taxe, passeport...), ainsi qu'un fichier *CSV* comportant une liste des sites gouvernementaux référencés dans le monde (<http://www.politicsresources.net/official.htm>).

### 2.2 CertStream

CertStream est une librairie qui fournit des mises à jour en temps réel à partir du réseau *Certificate Transparency Log*, permettant de l'utiliser comme un bloc de base pour construire des outils réagissant aux nouveaux certificats émis en temps réel

### 2.3 VirusTotal

VirusTotal permet de vérifier la présence de virus sur une *URL* grâce à une API publique qui autorise jusqu'à 4 requêtes par minutes. Elle fournit pour chaque requête un rapport comportant le nombre d'antivirus testés et ceux qui ont fourni un résultat positif.

### 2.4 IPAPI et CIRCL

*IPAPI* est un service permettant de retrouver une position *GPS* à partir d'une adresse *IP*. Cela nous permet de vérifier la position d'un potentiel site web gouvernemental d'un pays, afin de déterminer si il est dans le pays en question ou si il est ailleurs.

---

1. Cette attaque repose généralement sur une usurpation de l'identité de l'organisation, et procède par ingénierie sociale forte afin de lier le site à l'activité de l'organisation ciblée.

### 3 Fonctionnement de la solution

- Appel *API* CertStream → Récupération des certificats émis
- Extraction de l'*URL*
- Algorithme de ressemblance de chaînes : (distance de Levenshtein)
  - nombre de caractères identiques.
  - différence aux niveaux des caractères *ASCII*.
- Appel à *IPAPI* pour vérifier si l'adresse *IP* du serveur appartient au même pays que son nom de domaine. Les gouvernements utilisent leurs propres infrastructures, dans leur propre pays.
- Comparaison de la sortie d'*OCR* d'une *URL* avec l'*URL* de départ pour distinguer l'utilisation de caractère ressemblant mais n'étant pas le caractère perçu.
- Appel à l'*API* VirusTotal.
- Classifications des *URLs* qui passent les filtres sur une échelle de points.
- Archivage des sites web fichés comme sites de hameçonnage dans un fichier de journalisation.

### 4 Rapport de développement

#### 4.1 Mode de développement

Le développement de la solution a été séparé en plusieurs modules différents :

- Module de géolocalisation :
  - utilise les *APIs* `ipapi`, `bgpranking_web`, `dns_resolver`.
  - récupère l'adresse IP d'un domaine, la localisation du serveur qui l'héberge, le score *CIRCL* de cette adresse IP, et compare la localisation du serveur avec le pays défini par l'extension du domaine.
  - renvoi ces informations dans un objet de type *JSON*.
- Module de comparaisons de caractères :
  - Calcul de la distance de Levenshtein d'un nom de domaine avec les mots de notre dictionnaire.
  - Calcul de la distance de Levenshtein entre un nom de domaine et le texte extrait par reconnaissance optique d'une image créée à partir dudit nom de domaine (détection des caractères ressemblants).
- Module de requête Virus Total :
  - effectue des requêtes à l'*API* publique Virus Total et en récupère les résultats.
  - renvoie les données intéressantes du résultat dans un objet de type *tJSON*.

#### 4.2 Fonctionnement

Le programme principal utilise tous les modules pour attribuer un score à chaque domaine traité :

- Si le score obtenu est suffisamment important, on écrit une alerte sur la console avec le score obtenu par ce domaine.
- Si le score obtenu est supérieur à 100, on envoie aussi une alerte au format *STIXv2* que l'on affiche sur la console à défaut de l'envoyer à un agrégateur.

### 4.3 Utilisation

Le programme utilise python 3.6 et nécessite donc son installation. Il nécessite aussi l'installation de plusieurs dépendances python :

- `stix2` : utilisé pour la création d'alerte au format *STIXv2*.
- `ipapi` : utilisé pour récupérer la géolocalisation du serveur hébergeant un domaine.
- `dnspython` : utilisé pour récupérer l'adresse IP correspondant à un domaine.
- `bgpranking` : utilisé pour récupérer le score CIRCL d'un domaine.
- `certstream` : utilisé pour récupérer le flux des nouveaux certificats.
- `python-Levenshtein` : utilisé pour calculer les distances de Levenshtein.
- `pillow` : utilisé pour créer des images depuis du texte.
- `pytesseract` : utilisé pour la reconnaissance optique de caractères.
- `tqdm` : utilisé pour l'affichage sur la console.
- `termcolor` : utilisé pour l'affichage sur la console.

Le module `pytesseract` nécessite l'installation du programme `tesseract` (peut être nommé `tesseract-ocr`) sur votre système ainsi qu'au moins un paquet de langue.

L'installation des modules de dépendance est détaillée dans le fichier `main.py`.