Proposition équipe 5: Détection automatique d'URLs malicieuses

Jeremi Levesque LEVJ1404
Timothée Blanchy TIMB1101
Chris Tchimmegne Tchassem TCHC1301

En 2022, le FBI estime que des hameçonnages visées aux entreprises auraient causés des pertes ajustées de 2.7 milliards $USD (FBI, 2022). Les attaques deviennent de plus en plus sophistiquées et difficiles à reconnaître que ce soit pour un humain ou pour un système de filtrage. Le FBI recommande évidemment plusieurs bonnes pratiques, notamment d'examiner minutieusement les URLs faisant part du message afin de ne pas être leurré vers un site frauduleux qui tentera de soutirer des informations corporatives, financières ou personnelles à l'utilisateur. Nous proposons d'utiliser des techniques de détection automatique d'URLs malicieuses afin d'en réduire la quantité qui atteignent les utilisateurs de sorte à faire décroître le risque qu'ils accèdent à ces URLs par mégarde et par le fait même de décroître les pertes financières liées au hameçonnage. Afin d'avoir un filtrage très rapide des messages pour être déployé à grande échelle, nous proposons d'analyser la structure lexicale des URLs afin d'en déterminer la malignité. Cela évite donc d'aller vérifier le contenu de destination ce qui causerait des enjeux de délais de communication supplémentaires et en plus de nous rapprocher de l'attaquant.

URLs en entrée	Type (sortie)
icloud.com	Bénin
br-icloud.com.br	Malicieux
mp3raid.com/music/krizz_kaliko.html	Bénin
retajconsultancy.com	Malicieux
https://s.42l.fr/dc9dNseL	Bénin

Table 1: Exemples de classifications attendues d'URLs

La table 1 présente des exemples d'URLs qui seront donnés à notre solution afin que cette dernière classifie si l'URL est malicieuse ou non. On peut rapidement noter que même pour un humain renseigné il n'est trivial de distinguer la malignité de l'URL. Par exemple, *br-icloud.com.br* ressemble curieusement à *icloud.com* qui, elle, est une adresse bénigne. On ne peut donc pas simplement juger par des sous-chaînes de caractères reconnues fiables (e.g. *icloud.com*). Aussi, il devient très complexe d'évaluer la malignité d'un lien réduit comme le dernier de la table 1 par exemple. Dans ce cas, c'est un lien bénin puisqu'il redirige vers la page example.com (qui est bénin), mais comment détecter s'il est malicieux ou non?

Traditionnellement, plusieurs techniques comme l'utilisation de *blacklists* (Sun et al., 2015) (Prakash et al., 2010) (Ma et al., 2009) ainsi que des méthodes d'apprentissage machine (SVM, Régression logistique, Arbre de décisions, etc.) appliquées sur une représentation par caractéristiques des URLs ont été largement explorées. Cependant, les méthodes d'apprentissage profond ont récemment gagnées en popularité et elles se distinguent des méthodes classiques par leur performance de généralisation à de nouveaux types d'URLs. Le problème avec les méthodes classiques est qu'elles sont basées sur des caractéristiques prédéterminées ce qui permet de laisser aux attaquants la possibilité de modifier leurs URLs de sorte à les contourner et ainsi fausser le modèle. Les modèles de réseaux de neurones à convolution (Le et al., 2018) (parfois aussi utilisés avec des réseaux temporels (Das et al., 2020)) semblent être les réseaux les plus utilisés pour la détection d'URLs malicieuses dans la littérature. Ils offrent de résultats performants et une bonne capacité de généralisation à de nouveaux exemples étant donné leur capacité à découvrir automatiquement des caractéristiques discriminantes. Plus récemment, les *Transformers* offrent d'incroyables résultats dans le traitement automatique de séquences et ont donc été mis en essai pour évaluer leur performance sur les URLs notamment par (Maneriker et al., 2021), (Wang and Chen, 2022) et (Su and Su, 2023). Dans ce projet, nous proposons d'évaluer le gain de performance qu'un *transformer* pré entraîné et affiné sur notre ensemble d'entraînement peut apporter par rapport à une détection d'un réseau à convolutions.

References

Anupama Aggarwal, Ashwin Rajadesingan, and Ponnurangam Kumaraguru. 2012. Phishari: Automatic realtime phishing detection on twitter. In *2012 ECrime researchers summit*, pages 1–12. IEEE.

Arijit Das, Ankita Das, Anisha Datta, Shukrity Si, and Subhas Barman. 2020. Deep approaches on malicious url classification. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–6. IEEE.

FBI. 2022. Internet crime complaint center (ic3) 2022. *Internet Crime Complaint Center (IC3)*.

Hung Le, Quang Pham, Doyen Sahoo, and Steven CH Hoi. 2018. Urlnet: Learning a url representation with deep learning for malicious url detection. *arXiv preprint arXiv:1802.03162*.

Justin Ma, Lawrence K Saul, Stefan Savage, and Geoffrey M Voelker. 2009. Beyond blacklists: learning to detect malicious web sites from suspicious urls. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1245–1254.

Pranav Maneriker, Jack W Stokes, Edir Garcia Lazo, Diana Carutasu, Farid Tajaddodianfar, and Arun Gururajan. 2021. Urltran: Improving phishing url detection using transformers. In *MILCOM 2021-2021 IEEE Military Communications Conference (MILCOM)*, pages 197–204. IEEE.

Pawan Prakash, Manish Kumar, Ramana Rao Kompella, and Minaxi Gupta. 2010. Phishnet: predictive blacklisting to detect phishing attacks. In *2010 Proceedings IEEE INFOCOM*, pages 1–5. IEEE.

Ashish Singh and Pradeep Kumar Roy. 2021. Malicious url detection using multilayer cnn. In *2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, pages 340–345. IEEE.

Ming-Yang Su and Kuan-Lin Su. 2023. Bert-based approaches to identifying malicious urls. *Sensors*, 23(20):8499.

Bo Sun, Mitsuaki Akiyama, Takeshi Yagi, Mitsuhiro Hatada, and Tatsuya Mori. 2015. Autoblg: Automatic url blacklist generator using search space expansion and filters. In *2015 IEEE Symposium on Computers and Communication (ISCC)*, pages 625–631. IEEE.

Chenguang Wang and Yuanyuan Chen. 2022. Tcurl: Exploring hybrid transformer and convolutional neural network on phishing url detection. *Knowledge-Based Systems*, 258:109955.

Li Xu, Zhenxin Zhan, Shouhuai Xu, and Keying Ye. 2013. Cross-layer detection of malicious websites. In *Proceedings of the third ACM conference on Data and application security and privacy*, pages 141–152.

A Contributions

GitHub: Lien

JEREMI LEVESQUE

Contribution proposition:

- Revue de littérature de 15 articles
- Rédaction de l'énoncé du problème, l'importance du problème, ajout sur la non-trivialité de la tâche.

Contribution future:

- Fournir ressources informatiques pour l'entraînement.
- Implémentation du réseau de convolutions fine-tuned avec nos URLs.
- Implémentation du *transformer* fine-tuned RoBERT et/ou BERT sur nos URLs.
- Rédaction du rapport d'avancement.

TIMOTHÉE BLANCHY

Contribution proposition:

- Revue de littérature
- Rédaction de l'importance du problème, non-trivialité de la tâche.

Contribution future:

- Créer et gérer l'environnement pour qu'il soit propre et facile à prendre en main.

CHRIS TCHIMMEGNE TCHASSEM

Contribution proposition:

- Revue de littérature

Contribution future:

- Utiliser un dataset equilibré qui contient une proportion égale de chaque classe a classifier pour pouvoir diminuer l'erreur sur le accuracy Aider à implémenter le réseau de convolution
- Aider à implémenter le réseau de convolution pour augmenter la performance de notre model

B Dataset

- Malicious URLs dataset sur kaggle (lien, avec plus de 600 000 données classés en 4 catégories (*benign*, *defacement*, *phishing*, *malware*)

C Ressources informatiques

- Intel Core i9, 64 Go RAM, NVIDIA GTX 4090, 24 Go de mémoire dédiée.
- AMD Ryzen 7 5700G, 48Go RAM, NVIDIA GTX 3070, 8 Go de mémoire dédiée.
- Google Colab Pro si besoin.