Modélisation des données de sécurité pour l'apprentissage automatique

Pierre-François Gimenez
Maître de conférence à CentraleSupélec
Équipe Inria PIRAT

Académie des technologies
11 Mars 2024

Contexte

Cybersécurité

Les fonctions de la cybersécurité sont :

1. identifier les risques
2. prévenir les attaques
3. **détecter les attaques**
4. répondre aux attaques
5. remédier aux vulnérabilités

Aujourd'hui, je vais parler du point 3

Source : nist.gov

Contexte

Signatures, IA et experts

Comment détecter des attaques ?

- Signatures : le plus utilisé en détection. Mises à jour régulières nécessaires (partagées via CTI). Simple à comprendre, lève peu de faux positifs
- IA : principalement étudié en recherche. Peut détecter des attaques non-documentées (détection par anomalies), lève plus de faux positifs. Alertes plus difficiles à comprendre.
- Expert : moins d'erreurs que l'IA, mais plus lent et plus coûteux. Principalement employé pour la vérification d'alertes et pour réagir aux attaques.

En pratique, il y a une certaine fracture entre la détection par l'IA et l'analyse par les experts

Dans cet exposé

- Je vais présenter des leviers pour concilier experts et IA
- Cela passe notamment par une meilleure modélisation des données de sécurité

CentraleSupélec

Sommaire

CentraleSupélec

Tour d'horizon des données de sécurité

Données réseau

Fichier de capture réseau (pcap)

- L'ensemble des paquets transitant sur un réseau
- Typiquement capturés à un pare-feu
- Couche applicative généralement chiffrée, ce qui limite l'observabilité
- Possibilité de déchiffrement SSL, pratique peu répandue

Netflows

- Ensemble de métadonnées de description de flux entre deux IP
- Features : IP source et destination, port source et destination, nombre de paquets, durée de la connexion, flags TCP, etc.
- Plusieurs formats concurrents issus de différents outils
- Format pratique pour l'IA (données tabulaires), passe facilement à l'échelle

Quelques IDS exotiques s'intéressent aux signaux radio IoT

Données systèmes et applicatifs

Logs systèmes et applicatifs

- Évènements système (Linux Audit Logs, Windows Events Logs)
- Un ensemble hétérogène de logs textuels issus d'applications
- Généralement, 1 ligne = 1 évènement
- ... mais ce n'est pas toujours le cas
- Nécessite la création de parsers pour récupérer les informations intéressantes
- En pratique, ces parsers sont difficile à maintenir
- Des approches automatisées ont été proposées (identification de la partie variable des logs, etc.)
- Mais il est reste difficile de typer automatiquement ces données
- Format intermédiaire peu répandu : eCAR. Triplets (acteur, action, objet)

Séquences d'appels systèmes

Séquences d'appels systèmes

- Principalement utilisées en analyse de malware
- Contrairement aux logs qui s'intéressent au système en général, ces séquences portent sur une application en particulier
- Peut aller de paire avec un enregistrement des flux réseau associés à une application
- Rarement utilisées en détection d'intrusions

Il existe d'autres manières d'analyser des malwares (analyse statique par exemple), mais cela sort du périmètre de cet exposé.

CentraleSupélec

Enjeux de la représentation de données

L'approche classique

L'approche classique

Souvent, des indicateurs statistiques sont extraits de ces données :

- network flows pour les échanges réseau
- n-grammes (appels systèmes, logs, etc.)
- nombre d'appels systèmes d'un certain type dans une fenêtre temporelle
- etc.

Des modèles sont appris pour apprendre à lever des alertes. Les performances ne sont pas trop mauvaises, mais ces approches rencontrent des difficultés

Difficultés

Problèmes

- Fossé sémantique [SP10] : il est difficile de faire le lien entre une alerte "bas niveau" et son interprétation sur ce qu'elle indique sur le SI
- Un manque d'explicabilité des modèles de deep learning : souvent, en détection d'anomalies, il n'y a quasiment aucune explication des alertes
- Beaucoup de faux positifs, ce qui cause une fatigue des opérateurs : des IDS sont ignorés/désactivés
- Une vulnérabilité aux attaques adverses (mimicry attack, descente de gradient, etc.)

Représentation de données

Représentation via des graphes

- Plusieurs propositions de représenter ces données de sécurité sous la forme d'un graphe
- En général, ces représentations reposent sur des notions plus haut niveau qui s'appuient sur des connaissances expertes
- Les nœuds et les arêtes sont généralement typés et ont des attributs

Dans la suite, je présenterai plusieurs de ces représentations

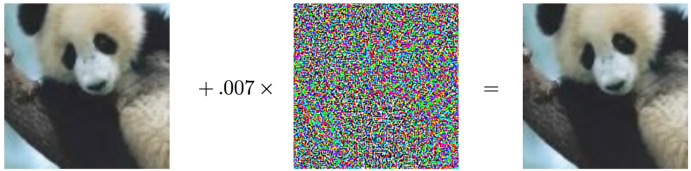
Avantages

- Cela permet une visualisation plus simple pour les opérateurs
- Le fossé sémantique est réduit car les représentations manipulent des objets plus "haut niveau"
- Cette représentation haut niveau est souvent plus difficile à manipuler par les attaquants

Robustesse contre les attaques adversariales en boîte blanche

Attaques adversariales

- Les attaques adversariales permettent notamment de changer la prédiction d'un réseau de neurones en changeant légèrement l'entrée
- Ces attaques sont faciles (descente de gradient) et transférables entre modèles
- Peu de solutions existent contre cette attaque, et elles ont un coût sur la performances
- À ma connaissance, une attaque peu rencontrée en pratique



Robustesse contre les attaques adversariales en boîte blanche

Espace du problème et espace des features

- L'espace du problème $\mathcal{P}$: l'espace des données observées. Par exemple : paquets réseau, séquence d'appels systèmes, etc.
- L'espace des features $\mathcal{F}$: l'espace des entrées du modèle. Par exemple : n-grammes, méta-données statistiques, graphe de représentation, etc.
- On passe de l'espace du problème à l'espace des features par une fonction d'extraction de features $\phi : \mathcal{P} \mapsto \mathcal{F}$
- Dans le cas des images, ϕ est l'identité
- Dans notre cas, ϕ est non-inversible et non-différentiable

Robustesse contre les attaques adversariales en boîte blanche

Une attaque en pratique

- L'attaquant dispose d'un contenu malveillant $x \in \mathcal{P}$
- Il calcule les features $x' = \phi(x)$
- L'attaquant attaque le modèle et, via une descente de gradient, obtient un vecteur d'entrée $y' \in \mathcal{F}$ qui évade le détecteur
- Puis, il cherche la donnée à fabriquer $y \in \mathcal{P}$ tel que $y' = \phi(y)$
- Plus la transformation ϕ est complexe, plus y est difficile à construire
- C'est notamment le cas quand on travaille sur des graphes "haut niveau"
- L'attaque est toujours possible mais avec une exploration en boîte noire de $\mathcal{P}$, ce qui est bien moins efficace et moins discret qu'une attaque en boîte blanche dans $\mathcal{F}$
- Exemple d'attaques sur un IDS utilisant des graphes de provenance : [GHWB23]

Difficultés

Difficultés

- Définition du graphe (types de nœuds et d'arêtes)
- Temps de construction qui n'est pas adapté à une détection temps réel
- Les modèles d'IA actuels ne peuvent pas travailler directement sur des graphes : ils travaillent soit avec des vecteurs de taille fixe, soit des séquences de données
- Pour résoudre ce problème, on utilise de l'embedding : modification de l'espace via une transformation apprise, le plus souvent pour une réduction de dimensions

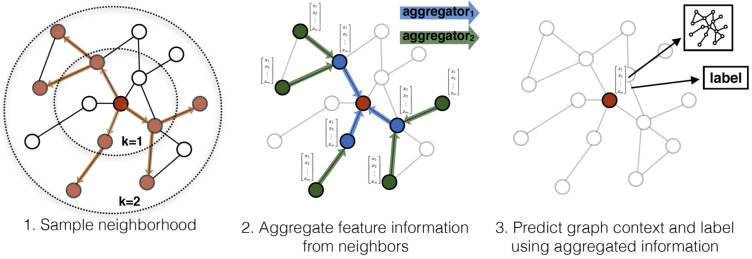
CentraleSupélec

Techniques d'IA rencontrées

Embeddings

Exemples de techniques d'embeddings

- Détection locale, nœud par nœud, ou arête par arête
- Marche aléatoire avec skip-gram (détaillé après)
- Embedding spectral
- Recherche de sous-graphes d'intérêt (identifiés manuellement)
- Graph Neural Network, e.g. GraphSAGE [HYL17] (cf. image ci-dessous)



1. Sample neighborhood

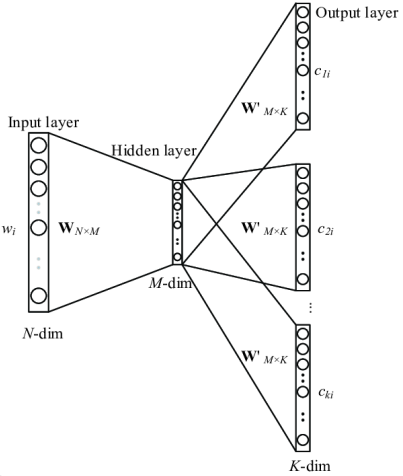
2. Aggregate feature information from neighbors

3. Predict graph context and label using aggregated information

Skip-gram

Skip-gram

- Technique issue du traitement du langage [Chu17]
- Apprentissage d'un réseau de neurones avec une structure particulière
- À partir d'une séquence de mots, on cherche à prédire le contexte d'un mot. Par exemple, pour "La cybersécurité est un enjeu national", on va essayer de prédire :
  - "la" et "est" à partir de "cybersécurité"
  - "cybersécurité" et "un" à partir de "est"
  - etc.
- Dans l'espace d'embedding appris, les mots apparaissent dans des contextes similaires ont des valeurs proches
- En pratique, des fenêtres de taille 10



Apprentissage

Contexte

Je vais traiter ici d'apprentissage non-supervisé (détection par anomalie) : on ne dispose pas d'exemples d'attaques

Apprentissage auto-supervisé

Principe : on crée une tâche qui, pour être résolue, implique que le modèle comprenne les motifs cachés dans les données

- Auto-encodeur : un réseau de neurones qui cherche à reconstruire son entrée
- Masking : une partie des données sont masquées et le modèle doit prédire ce que c'est
- Contrastive learning : deux graphes très proches doivent avoir un embedding proche, et deux graphes très différents doivent avoir un embedding différent. Ces graphes proches sont construits avec des perturbations locales (suppression d'arêtes ou de nœuds)

CentraleSupélec

État de l'art des représentations

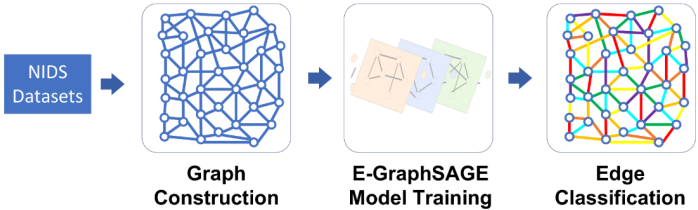
CentraleSupélec

Données réseaux

E-GraphSAGE

E-GraphSAGE

- Approche proposée par [LLS+22]
- Utilise un graphe simple
  - Les nœuds sont des adresses IP
  - Les arêtes sont des communications entre les nœuds, et sont étiquetées par les ports source et destination
- S'appuie sur GraphSAGE pour l'embedding

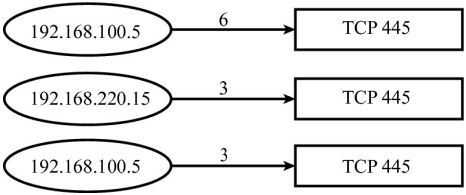


Graphes de Xiao

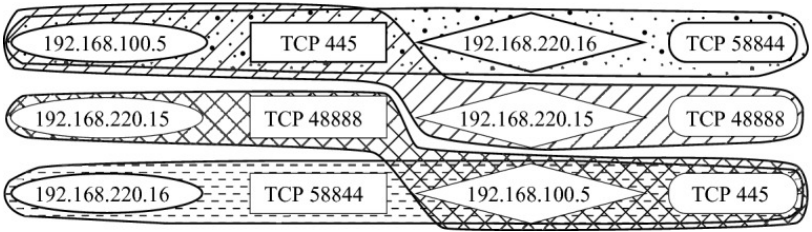
Graphes de Xiao

- Approche proposée par [XLW+20]
- Propose deux types de graphes :
  - graphe "de premier ordre", qui indique pour chaque adresse IP les ports utilisés
  - hypergraphe "de second ordre", où chaque hyperarête regroupe un quadruplet (adresse IP source, adresse IP destination, port source, port destination)
- Ce graphe est utilisé pour apprendre un espace latent en cherchant à prédire la probabilité d'apparition d'une arête en fonction des poids de chaque nœud

Graphes de Xiao



Graphe du premier ordre



Graphe du second ordre

Authentication Graph

Authentication Graph

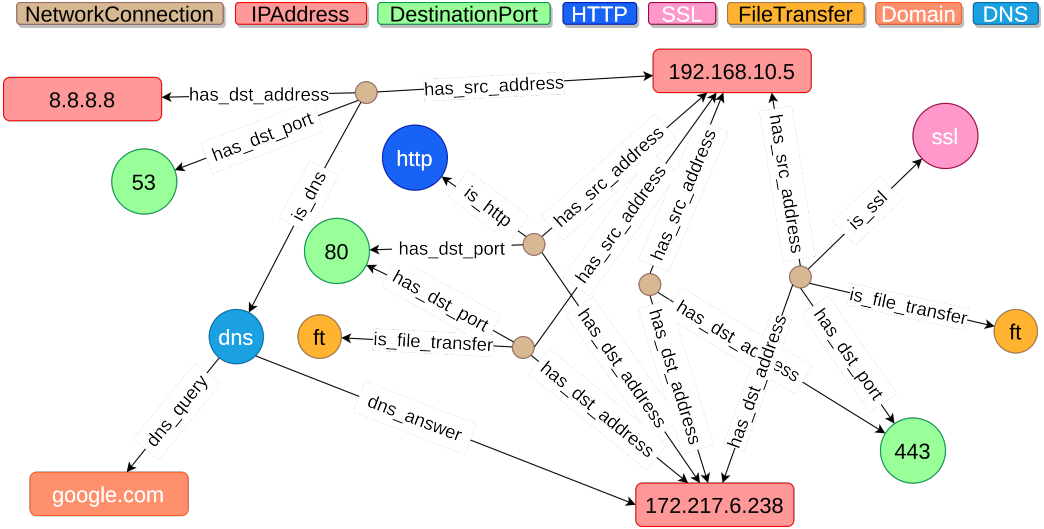
- Approche proposée par [WCYM20]
- Il s'agit d'un graphe bipartite, avec d'une part des machines, et d'autre part des utilisateurs
- Les arêtes sont pondérées par le nombre de connexions d'un utilisateur à une machine
- L'embedding est appris avec une marche aléatoire avec masquage (skip-gram)

Sec2Graph

Sec2Graph

- Approche proposée par [LTPM20]
- Les nœuds des objets réseaux : adresse IP, port, nom de domaine, protocole, etc.
- Les nœuds possèdent des attributs
- Les arêtes sont des relations
- ML : utilisation d'un auto-encodeur qui traite chaque arête indépendamment

Sec2Graph (exemple)



CentraleSupélec

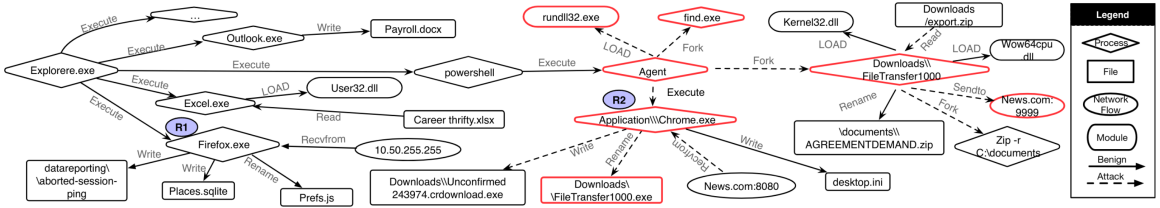
Logs systèmes et applicatifs

Graphe de provenance

Graphe de provenance

- L'utilisation de la provenance des données est ancienne
- L'idée est d'identifier, dans les logs, les acteurs, les actions, et les objets, et de les relier : acteur $-(\text{action})\rightarrow$ objet
- Les graphes de provenance sont largement utilisés dans les HIDS [HPB+20, AIH22, PH22, YXX+23, KSJ+23, CLL+23, JXN+23, RAH24]
- Les différences entre les IDS proposés portent sur l'embedding, le modèle utilisé, du contexte supervisé ou non-supervisé, etc.

Graphe de provenance (exemple issu de [RAH24])



CentraleSupélec

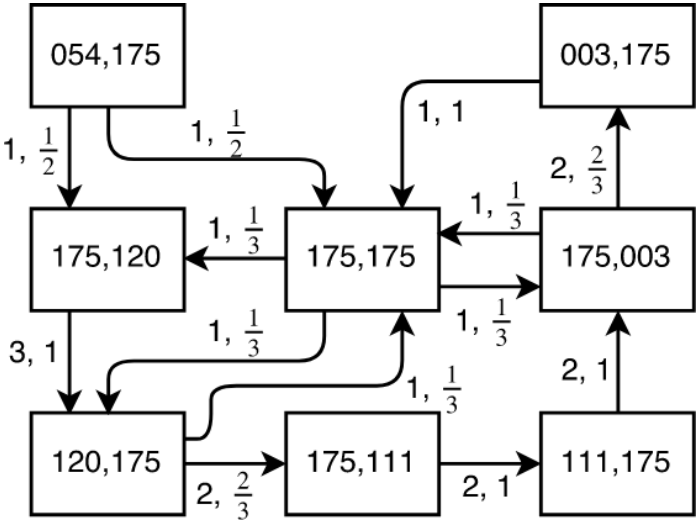
Séquences d'appels systèmes

Chaîne de Markov d'appels systèmes

Chaîne de Markov d'appels systèmes

- Simplement une chaîne de Markov issue de la séquence d'appels systèmes
- Approche utilisée par [CKYK17, GRK+18]
- N'utilise pas de deep learning

Chaîne de Markov d'appels systèmes (exemple)

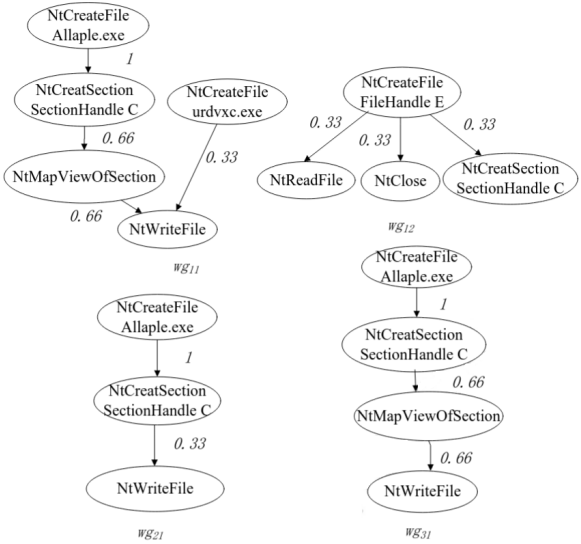


Graphe de famille de comportements

Graphe de famille de comportements

- Approche proposée par [DXCL18]
- S'intéresse à la suite d'appels systèmes qui concernent un même objet (fichier, processus, etc.), identifié par tainting
- Un algorithme d'identification de sous-graphes commun permet d'extraire des graphes caractéristiques

Graphe de famille de comportements (exemple)



MatchGNet

MathGNet

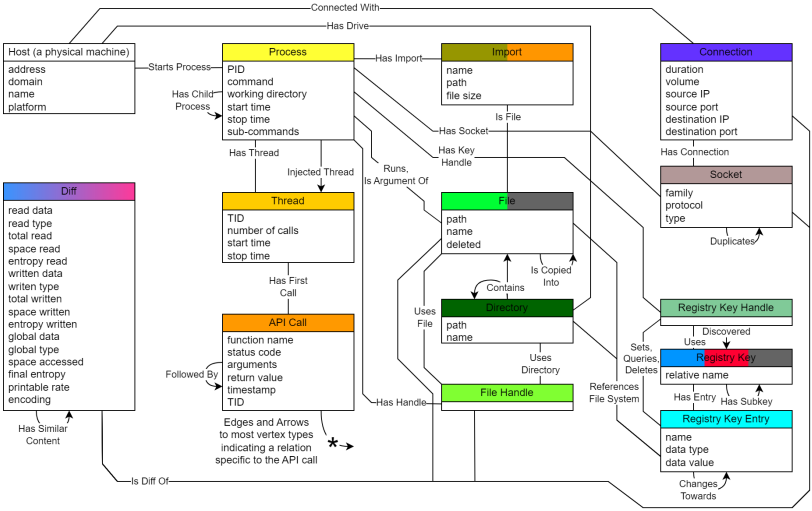
- Approche proposée par [WP19]
- Graphe avec trois types de nœud : processus, fichiers et socket réseau
- Arêtes : création d'un processus, accès à un fichier, connexion à un socket
- Embedding par GNN

BAGUETTE

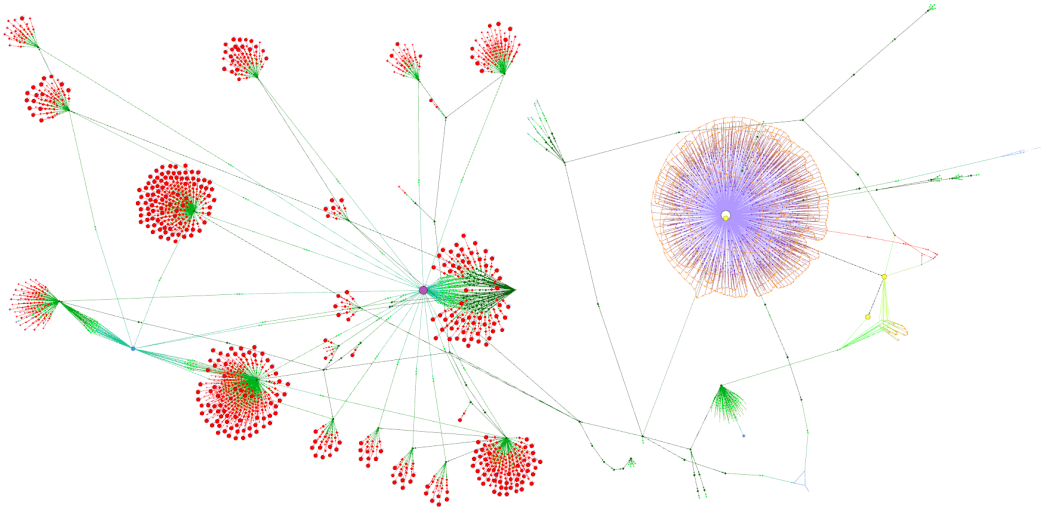
BAGUETTE

- Approche proposée par notre équipe dans [RGHVTT23]
- Graphe complexe, avec 14 types de nœuds (processus, thread, fichiers, clé de registre, socket, etc.)
- Contrairement aux graphes de provenance, il y a aussi des relations entre ressources (par exemple : relation "est un sous-répertoire de")
- Nous somme encore en train de développer l'exploitation du graphe par IA avec une approche de data mining

BAGUETTE : types de nœuds et d'arêtes



BAGUETTE (exemple)



CentraleSupélec

Après la levée d'alerte : défis à relever

L'explicabilité

Explicabilité

- Une alerte en tant que telle a une utilité très limitée si elle n'est pas détaillée
- Le XAI (explainable AI) est un champ de recherche qui porte sur l'explicabilité des modèles d'IA
- Des progrès sont faits en apprentissage supervisé, mais très peu en détection d'anomalies
- Nous avons récemment proposé une approche [LGH+23] pour ordonner l'importance des features dans le cas des auto-encodeurs
- Il s'agit d'une problématique trop peu étudiée
- Pas de consensus sur quelle forme doit avoir une bonne explication...
- En pratique, l'embedding se met en travers de l'explicabilité, car l'explication porte sur l'embedding plutôt que sur les données

Dérive conceptuelle

Dérive conceptuelle

- En pratique, le système à surveiller évolue régulièrement (nouveaux services, nouvelles technologies, modifications matérielles, etc.)
- La détection d'anomalie modélise le comportement normal, qui doit donc être mis à jour régulièrement
- Cette problématique, appelée "dérive conceptuelle" (*concept drift*), est rarement étudiée en recherche sur les IDS (notamment car il n'y a pas de données pour évaluer)
- Le plus souvent, la solution proposée est de réapprendre régulièrement le modèle. Cela néglige les problèmes que cela peut causer (augmentation du risque d'empoisonnement par exemple)

Human-in-the-loop

Human-in-the-loop

- Faire collaborer l'IDS avec l'expert directement
- L'IDS peut demander spécifiquement des analyses de certaines alertes pour apprendre rapidement (apprentissage actif)
- Par exemple : l'expert indique les erreurs à l'IDS qui peut éviter de les refaire à l'avenir
- Permet de réduire le taux de faux positifs
- Peu de recherche académique sur le sujet (difficile d'expérimenter avec de vrais experts...)
- Les LLMs sont une opportunité de faciliter la communiquer entre expert et IDS
- Un premier pas vers le MLOps ?

Pour aller plus loin

L'IA en cybersécurité

- Cette présentation s'est concentrée sur la représentation des données, ses avantages et ses défis
- L'IA est utilisable à plusieurs autres étapes : corrélation d'alertes dans les SIEM, réaction automatique, visualisation, qualification d'une alerte, priorisation, etc.

La sécurité de l'IA

- Un autre enjeu largement étudié dans la recherche est la vulnérabilité des modèles eux-mêmes face aux attaques
- Vol de données, évasion des détecteurs, création de portes dérobées dans un modèle, etc.

Conclusion

IA et cybersécurité

- Un grand potentiel encore peu réalisé
- Une interface difficile entre IA et cyber...
- En entrée : des représentations haut niveau, très variées, mais qui nécessitent un embedding complexe
- En sortie : peu d'explicabilité et trop de faux positifs
- Ce second problème est beaucoup moins traité mais est nécessaire pour réconcilier SOC et IA

Bibliographie I

MM Anjum, S Iqbal, and B Hamelin.
Anubis : a provenance graph-based framework for advanced persistent threat detection.
In *Proc. of the 37th ACM/SIGAPP Symp. on Applied Computing*, pages 1684–1693, 2022.

Kenneth Ward Church.
Word2vec.
Natural Language Engineering, 23(1) :155–162, 2017.

Zhi-Guo Chen, Ho-Seok Kang, Shang-Nan Yin, and Sung-Ryul Kim.
Automatic ransomware detection and analysis based on dynamic api calls flow graph.
In *Proceedings of the international conference on research in adaptive and convergent systems*, pages 196–201, 2017.

Bibliographie II

Z. Cheng, Q. Lv, J. Liang, D. Wang, Y.and Sun, T. Pasquier, and X. Han.
Kairos : Practical intrusion detection and investigation using whole-system provenance, 2023.

Yuxin Ding, Xiaoling Xia, Sheng Chen, and Ye Li.
A malware detection method based on family behavior graph.
Computers & Security, 73 :73–86, 2018.

Akul Goyal, Xueyuan Han, Gang Wang, and Adam Bates.
Sometimes, you aren't what you do : Mimicry attacks against provenance graph host intrusion detection systems.
In *30th Network and Distributed System Security Symposium*, 2023.

Bibliographie III

Martin Grimmer, Martin Max Röhling, Matthias Kricke, Bogdan Franczyk, and Erhard Rahm.
Intrusion detection on system call graphs.
Sicherheit in vernetzten Systemen, pages G1–G18, 2018.

X. Han, T. Pasquier, A. Bates, J. Mickens, and M. Seltzer.
Unicorn : Runtime provenance-based detector for advanced persistent threats.
arXiv preprint arXiv :2001.01525, 2020.

Will Hamilton, Zhitao Ying, and Jure Leskovec.
Inductive representation learning on large graphs.
Advances in neural information processing systems, 30, 2017.

Z. Jia, Y. Xiong, Y. Nan, Y. Zhang, J. Zhao, and M. Wen.
Magic : Detecting advanced persistent threats via masked graph representation learning, 2023.

Bibliographie IV

I. J. King, X. Shu, J. Jang, K. Eykholt, T. Lee, and H. H. Huang.
Edgetorrent : Real-time temporal graph representations for intrusion detection.
In *Proc. of the 26th Int. Symposium on Research in Attacks, Intrusions and Defenses*, RAID '23, page 77–91, 2023.

Maxime Lanvin, Pierre-François Gimenez, Yufei Han, Frédéric Majorczyk, Ludovic Mé, and Eric Totel.
Towards understanding alerts raised by unsupervised network intrusion detection systems.
In *Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses*, RAID '23, page 135–150, New York, NY, USA, 2023. Association for Computing Machinery.

Bibliographie V

Wai Weng Lo, Siamak Layeghy, Mohanad Sarhan, Marcus Gallagher, and Marius Portmann.
E-graphsage : A graph neural network based intrusion detection system for iot.
In *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, pages 1–9. IEEE, 2022.

Laetitia Leichtnam, Eric Totel, Nicolas Prigent, and Ludovic Mé.
Sec2graph : Network attack detection based on novelty detection on graph structured data.
In *Detection of Intrusions and Malware, and Vulnerability Assessment : 17th International Conference, DIMVA 2020, Lisbon, Portugal, June 24–26, 2020, Proceedings 17*, pages 238–258. Springer, 2020.

Bibliographie VI

R. Paudel and H. H. Huang.
Pikachu : Temporal walk based dynamic graph embedding for network anomaly detection.
In *IEEE/IFIP Network Operations and Management Symp. (NOMS)*, pages 1–7, 2022.

Mati Ur Rehman, Hadi Ahmadi, and Wajih Ul Hassan.
Flash : A comprehensive approach to intrusion detection via provenance graph representation learning.
In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 139–139. IEEE Computer Society, 2024.

Vincent Raulin, Pierre-François Gimenez, Yufei Han, and Valérie Viet Triem Tong.
BAGUETTE : Hunting for Evidence of Malicious Behavior in Dynamic Analysis Reports.
In *SECRYPT 2023 - 20th International conference on security and cryptography*, pages 1–8, Rome, Italy, July 2023.

Bibliographie VII

Robin Sommer and Vern Paxson.
Outside the closed world : On using machine learning for network intrusion detection.
In *2010 IEEE symposium on security and privacy*, pages 305–316. IEEE, 2010.

Renzheng Wei, Lijun Cai, Aimin Yu, and Dan Meng.
Age : authentication graph embedding for detecting anomalous login activities.
In *Information and Communications Security : 21st International Conference, ICICS 2019, Beijing, China, December 15–17, 2019, Revised Selected Papers 21*, pages 341–356. Springer, 2020.

Shen Wang and S Yu Philip.
Heterogeneous graph matching networks : Application to unknown malware detection.
In *2019 IEEE International Conference on Big Data (Big Data)*, pages 5401–5408. IEEE, 2019.

Bibliographie VIII

Qingsai Xiao, Jian Liu, Quiyun Wang, Zhengwei Jiang, Xuren Wang, and Yepeng Yao.
Towards network anomaly detection using graph embedding.
In *Computational Science–ICCS 2020 : 20th International Conference, Amsterdam, The Netherlands, June 3–5, 2020, Proceedings, Part IV 20*, pages 156–169. Springer, 2020.

F. Yang, J. Xu, C. Xiong, Z. Li, and K. Zhang.
Prographer : An anomaly detection system based on provenance graph embedding.
In *32nd USENIX Security Symposium*, pages 4355–4372, 2023.