Année universitaire : 2024/2025

Rapport du Projet de Module Sécurité Informatique

Réalisé par :

OULACHGAR Jamal
AABID Mohamed amine

Encadré par :

Prof. Najem MOUSSA

Année universitaire : 2024/2025

Sujet

Contexte et choix du sujet

Dans le cadre du module Sécurité Informatique, nous avons choisi le sujet **Reinforcement Learning & Cyber Security**, l'un des trois sujets proposés par le professeur. Ce sujet vise à explorer l'application des algorithmes d'apprentissage par renforcement (Reinforcement Learning) dans le contexte de la cybersécurité. Nous nous sommes inspirés du travail de Nicolò Romandini (décrit dans le document "Implementation of reinforcement learning algorithms in a cyber security simulation"), qui est basé sur une simulation décrite par Elderman.

Dans cette simulation, un attaquant (**Attacker**) tente de pénétrer un réseau informatique composé de plusieurs nœuds pour atteindre un nœud contenant des données sensibles (appelé **data node**), tandis qu'un défenseur (**Defender**) cherche à l'en empêcher en renforçant la sécurité du réseau. Le réseau simulé dans notre projet est composé de 4 nœuds : un nœud de départ (**START**), deux nœuds intermédiaires (**CPU1**, **CPU2**), et un nœud final (**DATA**, nœud 3). Chaque nœud est associé à un port réseau (par exemple, Port 80 pour HTTP, Port 443 pour HTTPS), ce qui reflète des scénarios réels de cybersécurité.

Contexte de la cybersécurité

Dans ce projet, nous avons pris en compte des concepts fondamentaux de la cybersécurité :

- **Ports réseau** : Chaque nœud est associé à un port (ex. : Port 80 pour HTTP, Port 443 pour HTTPS). Ces ports sont souvent ciblés dans des attaques réelles (ex. : attaques DDoS sur le port 80).
- **Commandes** : Nous avons simulé la commande **netstat -an** pour lister les ports actifs, ce qui est une pratique courante pour surveiller un réseau et détecter des activités suspectes.
- **Attaques** : L'attaquant peut lancer une attaque DDoS (Distributed Denial of Service) avec une probabilité de 10%, ciblant le port 80 du nœud 0, ce qui reflète un scénario d'attaque courante

Lien GitHub de la source initiale : (https://github.com/nickromandini/reinforcement-learning-cybersecurity/)

Lien GitHub de la source Complet de projet : (https://github.com/jamaloulachgar/CyberSecurityRL)

Année universitaire : 2024/2025

Objectif

Objectif principal

L'objectif central de ce projet est de simuler un scénario réaliste de cybersécurité, mettant en œuvre l'apprentissage par renforcement pour modéliser l'affrontement stratégique entre un attaquant et un défenseur au sein d'un réseau informatique. Ce travail vise à analyser comment ces deux agents peuvent, grâce à l'algorithme Q-Learning, adapter et optimiser leurs stratégies respectives au fil des interactions.

Objectifs spécifiques

1. **Implémenter une simulation réaliste**

   Concevoir un environnement réseau composé de 4 nœuds, chacun associé à un port courant (80, 445, 23, 443). L'attaquant a pour mission d'atteindre le nœud 3 (data node), tandis que le défenseur doit l'en empêcher en sécurisant le réseau.

2. **Exploiter l'algorithme Q-Learning**

   Permettre à l'attaquant et au défenseur d'apprendre de leurs expériences en utilisant le Q-Learning, un algorithme d'apprentissage par renforcement. Les agents ajustent leurs stratégies en fonction des récompenses positives ou négatives reçues après chaque action, dans le but d'optimiser leurs chances de succès.

3. **Visualiser les dynamiques de la simulation**

   Développer une interface graphique interactive avec JavaFX, affichant la topologie du réseau, l'état des nœuds et des ports, ainsi que les scores des deux agents. Intégrer un graphique JFreeChart pour suivre l'évolution des pourcentages de victoire et du risque global au fil des simulations.

4. **Analyser les performances et les stratégies**

   Étudier l'efficacité des différentes stratégies de défense face aux attaques, en identifiant les facteurs qui permettent à un agent (ici, le défenseur) de dominer l'autre. Cette analyse vise à mieux comprendre les dynamiques d'apprentissage et d'adaptation dans un contexte de cybersécurité.

5. **Intégrer des concepts concrets de cybersécurité**

   Incorporer des éléments réalistes tels que la gestion des ports réseau et la simulation de commandes système (ex. : `netstat -an`), afin de rapprocher la simulation des problématiques rencontrées dans des environnements informatiques réels.

Année universitaire : 2024/2025

Code

Structure du code

- **Main.java**
  - Gère la logique principale de la simulation.
  - Affiche l'interface graphique (JavaFX) : 4 nœuds (cercles), leurs ports, connexions (lignes).
  - Affiche les scores (victoires attaquant/défenseur) et un graphique JFreeChart pour les pourcentages de victoire.
- **Attacker.java**
  - Implémente l'attaquant avec Q-Learning (choix du nœud/port à attaquer).
  - 10 % de probabilité de lancer une attaque DDoS sur le nœud 0 (port 80).
  - Paramètres Q-Learning : alpha = 0.1, gamma = 0.95, epsilon = 0.04.
- **Defender.java**
  - Implémente le défenseur avec Q-Learning (choix du nœud/port à protéger).
  - Paramètres Q-Learning : alpha = 0.05, gamma = 0.91, epsilon = 0.07.
- **Simulation.java**
  - Orchestration des interactions attaquant/défenseur à chaque épisode.
  - Détermine les conditions de victoire (ex : l'attaquant gagne s'il atteint le nœud 3, sinon le défenseur gagne).
- **NetworkEnvironment.java**
  - Définit la topologie du réseau (4 nœuds, connexions via NEIGHBOURS_MATRIX, ports 80/445/23/443).
  - Simule la commande `netstat -an` pour afficher les ports actifs.

Année universitaire : 2024/2025

Améliorations

Nous avons repris le projet de Nicolò Romandini, disponible sur GitHub **(https://github.com/nickromandini/reinforcement-learning-cybersecurity/),** et y avons apporté les Améliorations apportées :

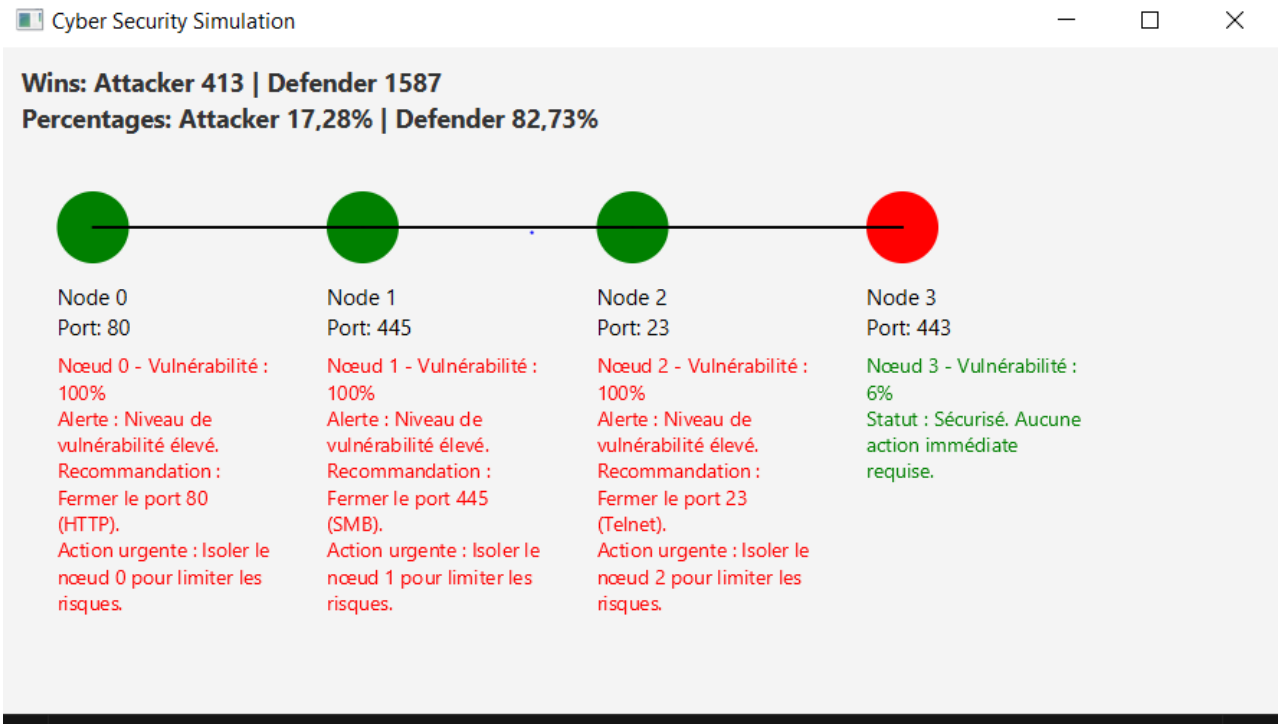
1. **Conversion Python → Java**
    - Réécriture complète du code initial Python en Java, adaptation des classes principales et remplacement de NumPy/Matplotlib par JavaFX et JFreeChart.
    - Traduction des structures de données et de l'algorithme Q-Learning.
2. **Nouvelle interface graphique (JavaFX)**
    - Visualisation des nœuds et de leurs ports.
    - Affichage dynamique des victoires, pourcentages et couleurs selon l'état de chaque nœud.
3. **Graphiques JFreeChart**
    - Suivi visuel de l'évolution des victoires et du risque global sur 10 simulations, avec personnalisation des couleurs.
4. **Simulation réaliste des ports**
    - Attribution de ports courants (HTTP, SMB, Telnet, HTTPS) et simulation de la commande netstat -an.
5. **Analyse de vulnérabilité en temps réel**
    - Ajout d'un scanner de vulnérabilité par nœud, affichage des scores et recommandations de sécurité en direct.
    - Système de pondération selon le type de port et suivi du risque global dans les graphiques

Année universitaire : 2024/2025

Analyse des résultats de la Simulation

1. Statistiques Générales

- **Nombre de victoires :**
  - Attaquant : **413**
  - Défenseur : **1587**
- **Pourcentages de réussite :**
  - Attaquant : **17,28%**
  - Défenseur : **82,73%**



Interprétation : Le défenseur domine largement la simulation, ce qui indique que les mécanismes de défense mis en place sont efficaces face aux attaques simulées. L'attaquant ne parvient à compromettre le réseau que dans moins d'un cas sur cinq.

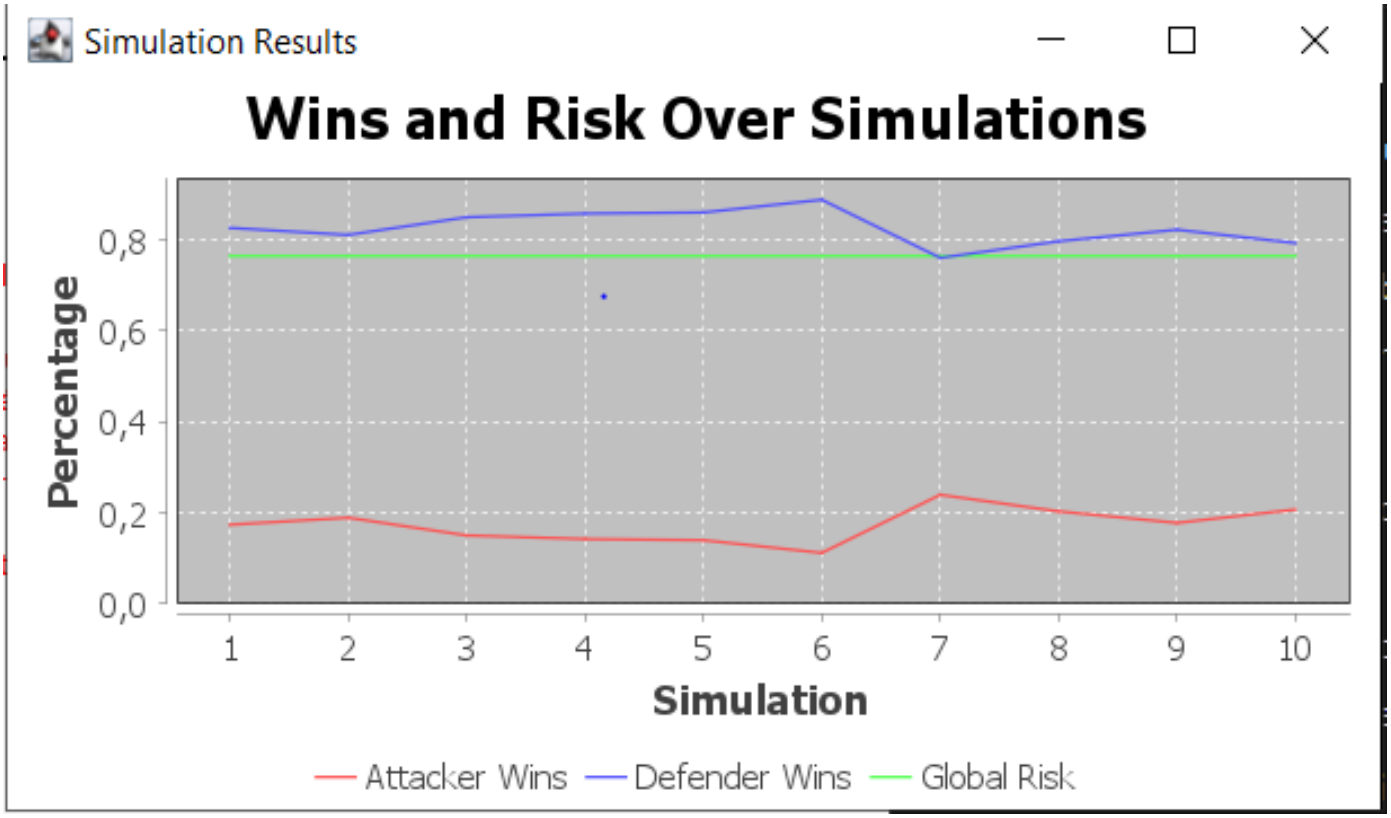
2. État des Nœuds du Réseau

- **Node 0 (Port 80 - HTTP) :**
  - Vulnérabilité : **100%**
  - Alerte : Niveau de vulnérabilité élevé.
  - Recommandation : Fermer le port 80 (HTTP).
  - Action urgente : Isoler le nœud 0 pour limiter les risques.
- **Node 1 (Port 445 - SMB) :**
  - Vulnérabilité : **100%**
  - Alerte : Niveau de vulnérabilité élevé.
  - Recommandation : Fermer le port 445 (SMB).
  - Action urgente : Isoler le nœud 1 pour limiter les risques.
- **Node 2 (Port 23 - Telnet) :**
  - Vulnérabilité : **100%**
  - Alerte : Niveau de vulnérabilité élevé.
  - Recommandation : Fermer le port 23 (Telnet).
  - Action urgente : Isoler le nœud 2 pour limiter les risques.
- **Node 3 (Port 443 - HTTPS) :**
  - Vulnérabilité : **6%**
  - Statut : Sécurisé. Aucune action immédiate requise.

Interprétation : Trois nœuds sur quatre présentent une vulnérabilité critique, principalement à cause de ports ouverts connus pour être des cibles fréquentes (HTTP, SMB, Telnet). Le nœud 3, utilisant HTTPS, est bien protégé et ne nécessite pas d'intervention.

Année universitaire : 2024/2025

1. Graphique : Victoires et Risque Global



- **Courbe rouge (Attacker Wins) :**

  Reste basse tout au long des simulations, confirmant la faible réussite des attaques.

- **Courbe bleue (Defender Wins) :**

  Se maintient à un niveau élevé, montrant la constance de la défense.

- **Courbe verte (Global Risk) :**

  Reste relativement stable, ce qui indique que le risque global du réseau ne fluctue pas fortement malgré les attaques répétées.

Interprétation : Le système de défense est robuste et parvient à maintenir un risque global maîtrisé, même si certains nœuds restent vulnérables. Cela suggère que les attaques sont soit détectées et bloquées efficacement, soit qu'elles n'atteignent pas leur objectif.

4. Recommandations Générales

- **Priorité à la fermeture des ports vulnérables** (80, 445, 23) pour réduire drastiquement la surface d'attaque.

- **Isoler les nœuds compromis** pour éviter la propagation d'une attaque.
- **Surveiller en continu** les ports ouverts et appliquer des correctifs de sécurité.
- **Renforcer la sécurité des nœuds critiques** (ceux avec des services exposés).

Conclusion

Ce projet de simulation de cybersécurité basé sur l'apprentissage par renforcement offre un environnement réaliste pour étudier les interactions entre attaquants et défenseurs dans un réseau informatique. Grâce à l'intégration d'une interface graphique intuitive et de visualisations dynamiques, il permet de suivre en temps réel l'évolution des attaques,
des défenses et du niveau de risque global du réseau.

Les résultats obtenus montrent que, même face à des vulnérabilités critiques sur plusieurs nœuds, la mise en place de stratégies de défense actives et adaptées permet de contenir efficacement la m ajorité des attaques. Le défenseur parvient à maintenir un taux de réussite élevé,
tandis que l'attaquant rencontre de grandes difficultés à compromettre durablement le réseau.

Ce projet met en évidence l'importance de la surveillance continue, de la gestion proactive des vulnérabilités (notamment la fermeture des ports sensibles) et de
l'isolation rapide des nœuds compromis.
Il constitue un outil pédagogique et expérimental précieux pour tester,
comparer et améliorer des stratégies de cybersécurité dans un cadre contrôlé.

En perspective, ce simulateur pourrait être enrichi par l'ajout de nouveaux types d'attaques, de mécanismes de défense plus sophistiqués, ou encore par l'intégration de scénarios réseau plus complexes, afin d'approfondir l'étude des dynamiques de sécurité informatique.

Références

- Elderman, R., et al. (2017). *Adversarial reinforcement learning in a cyber security simulation*. ICAART, pages 559-566.

- Romandini, N. *Implementation of reinforcement learning algorithms in a cyber security simulation*. [**GitHub](https://github.com/nickromandini/reinforcement-learning-cybersecurity**/).