Mélange de ports et d'IP par processus de décision markoviens min-max

Pierre Charreaux, Alexandre Reiffers-Masson, Francoise Sailhan, Sandrine Vaton

Mélange de ports et d'IP par processus de décision markoviens min-max

Pierre Charreaux, Alexandre Reiffers-Masson, Françoise Sailhan, Sandrine Vaton

IMT-Atlantique, laboratoire Lab-STICC

Nous introduisons une méthode optimale de mélange d'adresses IP et de ports dont l'objet est de rendre inopérante la phase d'exploration du réseau menée par l'attaquant. Nous formalisons les interactions entre attaquant et défenseur sous la forme de processus de décision markoviens min-max faiblement couplés que nous évaluons numériquement.

Mots-clefs : Défenses par cible mouvante, Processus de décision markoviens faiblement couplés

1 Introduction

La défense par cible mouvante - ou Moving Target Defense (MTD) - est une technique qui modifie l'état d'un système, en changeant par exemple les ports ouverts ou les logiciels. L'objectif est de prévenir les attaques en augmentant l'incertitude de l'attaquant et en réduisant la surface d'attaque. La défense par cible mouvante proposée introduit des changements au niveau des adresses IP et ports, qui perturbent la phase d'exploration de l'attaquant en augmentant le coût du sondage et en rendant ses investigations imprécises voire obsolètes. S'il existe plusieurs techniques de mouvement s'appliquant à plusieurs surfaces d'attaque (comme la surface d'exploration), l'une des techniques fondamentales de MTD est appelée l'IP/port shuffling et revient à changer l'adresse IP/les ports connus des serveurs. Alors que les stratégies de mouvement sont souvent modélisées sous le prisme de la théorie des jeux, nous formalisons les interactions entre attaquant et défenseur sous la forme d'un problème de processus de décision markoviens faiblement couplés, que nous résolvons numériquement pour identifier la stratégie optimale. La stratégie de l'attaquant est modélisée sous la forme d'un ensemble de chaînes de Markov contrôlées à 2 états, et celle du défenseur sous la forme d'une perturbation de celles-ci, ayant pour effet de changer les états des chaînes de Markov de l'attaquant. Alors que l'attaquant tente d'optimiser ses gains (e.g. la quantité d'informations obtenues) en réalisant les meilleures actions possibles, le défenseur essaie de réduire les gains de l'attaquant en perturbant ces actions.

2 Mélange optimal de ports et d'adresses IP

Nous considérons un réseau virtuel, dans lequel (i) un attaquant vise à découvrir les adresses IP & les ports ouverts des serveurs ainsi que les services accessibles sur ces ports et leurs vulnérabilités, (ii) un défenseur change les adresses IP et les ports des services et cherche à détecter les scans effectués par l'attaquant, à l'aide d'un système de détection d'intrusion. Un attaquant peut suivre différentes stratégies suivant qu'il souhaite (ou non) accélérer le scan du réseau au risque (ou non) de se faire détecter. En particulier, il peut faire varier (i) le nombre de ports scannés en parallèle, (ii) le délai entre deux scans, (iii) le nombre de tentatives lorsqu'aucune réponse n'est reçue à la suite d'un scan. L'outil nmap [†] propose les 5 profils [‡] de scan suivants (Tableau 1) :

— Le profil *insensé* scanne plusieurs ports en parallèle en se limitant à 2 tentatives en l'absence de réponse.

†. https ://nmap.org/

‡. Il existe un sixième profil nommé "paranoïaque" qui n'est pas mentionné ici.

— Le profil *agressif* permet un scan rapide en parallèle avec au plus 6 tentatives de scan.
— Le profil *normal* analyse plusieurs ports en parallèle avec un délai entre deux scans d'au plus 1s.
— Le profil *poli* n'effectue qu'un scan à la fois avec un délai de 0,4s entre deux scans.
— Le profil *discret* scanne un seul port à la fois et attend 15s entre deux scans.

TABLE 1 : Durée de scan et détectabilité des profils nmap dans un réseau virtuel comprenant la machine virtuelle (VM) d'un attaquant hébergeant la distribution kali, 3 VMs linux hébergeant 2 serveurs et le détecteur d'intrusion Snort.

Profil	Nombre de ports scannés en parallèle	Durée	Première détection	Nombre d'alertes
Insensé	Variable	0.11s	0.063s	1
Agressif	Variable	0.23s	0.1s	1
Normal	Variable	0.36s	0.163s	1
Poli	1	401.06s	8.63s	5
Discret	1	4h 35m 1s	Jamais	Aucune

Suivant le profil choisi, un scan prend de 0.11s à 4h35 : un scan très rapide est signalé une seule fois alors qu'un scan plus long (profil poli) est périodiquement (re)signalé.

2.1 Formalisation des interactions attaquant-défenseur

Pour modéliser le comportement de l'attaquant et du défenseur, nous considérons des processus de décision markoviens faiblement couplés à N bras, dans lesquels chaque bras correspond à un port et N désigne le nombre de ports (un multiple de 65535, le nombre de ports par machine). L'état d'un bras correspond au fait que le port soit :

— "en cours" d'analyse/de scan : nous supposons qu'une analyse continue à moins qu'elle ne soit interrompue par le défenseur qui change par exemple de port. Elle continue [§] car l'attaquant souhaite demeurer non détecté et récupérer les informations nécessaires pour mener une attaque.
— "en attente" : une analyse n'est pas en cours car elle n'a jamais été lancée ou a été interrompue par le défenseur qui mélange les adresses IP/ports pour rendre difficile la phase de reconnaissance de l'attaquant. Cela perturbe certains bras de l'attaquant qui reviennent à l'état "en attente".

L'attaquant active une partie des bras à chaque instant ce qui limite le nombre d'analyses démarrées [¶] à un instant donné. Nous supposons que les bras sont statistiquement indifférentiables et que chaque bras activé produit une récompense (positive). L'attaquant souhaite maximiser sa connaissance des ports et donc sa récompense accumulée au fil du temps, alors que le défenseur veut limiter cette connaissance en maintenant une qualité de service suffisante. Comme certains services sont populaires (protocole HTTP) alors que d'autres le sont moins, nous supposons que certains ports fournissent des informations avec une probabilité plus élevée que d'autres et que l'attaquant peut estimer en moyenne les récompenses qu'il va recevoir.

Type de bras : chaque bras est caractérisé par un type dépeignant le degré de popularité du service correspondant. Ce type est indexé par $i \in \mathcal{I}$, où $\mathcal{I}$ est un ensemble fini. Un port k de type i ($T_k = i$) fournit une information avec une probabilité q_i.

État d'un bras : Au temps $t \in \{0, ..., T\}$, l'état $S_k(t)$ du bras k prend soit la valeur 0 soit 1. Nous désignons par $A_k(t) \in \{0, 1\}$ l'action de l'attaquant sur le bras k. Lorsque $A_k(t) = 1$ l'analyse est démarrée à l'instant t sur le port k. Lorsque $A_k(t) = 0$, l'attaquant ne démarre pas de nouvelle analyse, mais une analyse peut toutefois être en cours.

La chaîne de Markov contrôlée est la suivante :

— Lorsque $A_k(t) = 1$, $S_k(t+1) = 1$ indépendamment de $S_k(t)$. Si $S_k(t) = 0$ alors l'attaquant démarre une analyse sur le port k, sinon l'attaquant maintient l'analyse en cours.
— Lorsque $A_k(t) = 0$ et $S_k(t) = 0$, alors $S_k(t+1) = 0$. Un port sur lequel aucune analyse n'est en cours ne change pas d'état si l'attaquant ne réalise pas d'action.
— Cependant, lorsque $A_k(t) = 0$ et $S_k(t) = 1$, $S_k(t+1) = 0$ avec une probabilité β et $S_k(t+1) = 1$ avec une probabilité $1 - \beta$. Le défenseur mélange le bras avec une probabilité β, ce qui interrompt l'analyse en cours.

§. Un état absorbant "succès" peut être ajouté pour qu'une analyse qui est un succès se termine.
¶. Une alternative serait de limiter le nombre d'analyses en cours effectuées par l'attaquant.

Contrainte budgétaire : L'attaquant active au maximum $\alpha.N$ bras, à chaque instant. Nous supposons que le scan reste indétectable tant que la fraction de nouvelles analyses (correspondant à $A_k(t) = 1$) à l'instant t est inférieure à α (avec $0 \leq \alpha \leq 1$).
Récompense : Pour chaque bras en cours d'analyse, l'attaquant obtient une récompense de 1 avec la probabilité q_i et 0 autrement. L'attaquant obtient ainsi en moyenne une récompense[‖] égale à q_i.
Résolution du problème : il s'agit de résoudre une optimisation avec coût noté c (1a) d'un problème de processus de décision markoviens faiblement couplés à horizon fini (2a)-(2f), étant donné :

— $M_{s,i}^{(N)}(t)$:= la fraction des bras de type i dans l'état s au temps t, est égale à $\sum_k \mathbb{1}_{T_k=i}\mathbb{1}_{S_k=s}/N$.
— $Y_{s,a,i}^{(N)}(t)$:= la fraction des bras de type i dans l'état s au temps t pour lesquelles la décision a est prise. Cette fraction est égale à $\sum_k \mathbb{1}_{T_k=i}\mathbb{1}_{S_k=s}\mathbb{1}_{A_k=a}/N$.

Le défenseur minimise ses coûts et les gains de l'attaquant en changeant β (1a), et l'attaquant maximise ses gains en trouvant les fractions de bras optimales (2a). Cependant, l'attaquant doit respecter une contrainte de normalisation (2b) ainsi que la condition initiale du système (2f). Pour ne pas être détecté, il doit limiter le nombre d'analyses démarrées (2e), et les fractions de ses bras doivent suivre la dynamique du système (2c),(2d)[**].

$$\underset{\beta\,=\,(\beta_i)_{i\in\mathcal{I}}}{\text{minimize}} \quad V_{\text{opt}}^{(N)}(m(0),T,\beta) + \frac{c}{T\times|\mathcal{I}|}\sum_{t=0}^{T-1}\sum_{i\in\mathcal{I}}(\beta_i(t)) \tag{1a}$$

$$V_{\text{opt}}^{(N)}(m(0),T,\beta) = \underset{Y}{\text{maximize}} \quad \sum_{t=0}^{T}\sum_{i,a} q_i\mathbb{E}(Y_{1,a,i}^{(N)}(t)) \tag{2a}$$

$$\text{subject to} \quad Y_{s,0,i}^{(N)}(t) + Y_{s,1,i}^{(N)}(t) = M_{s,i}^{(N)}(t), \forall t \in [[0,T]], \forall s \in \{0,1\}, \forall i \in \mathcal{I}, \tag{2b}$$

$$M_{1,i}^{(N)}(t+1) = (1-\beta_i(t))Y_{1,0,i}^{(N)}(t) + \sum_{s\in\{0,1\}} Y_{s,1,i}^{(N)}(t), \forall t, \forall i \in \mathcal{I}, \tag{2c}$$

$$M_{0,i}^{(N)}(t+1) = \beta_i(t)Y_{1,0,i}^{(N)} + Y_{0,0,i}^{(N)}(t), \forall t \in [[0,T-1]], \forall i \in \mathcal{I}, \tag{2d}$$

$$\sum_{i\in\mathcal{I}}\sum_{s\in\{0,1\}} Y_{s,1,i}^{(N)}(t) \leq \alpha, \forall t \in [[0,T]], \tag{2e}$$

$$M_{s,i}^{(N)}(0) = m_{s,i}(0), \forall s \in \{0,1\}, \forall i \in \mathcal{I}. \tag{2f}$$

FIGURE 1 : Formulation en processus de décision markoviens faiblement couplés de l'interaction attaquant/défenseur

Stratégie optimale de l'attaquant - Déterminer les stratégies optimales de l'attaquant, nécessite de résoudre un problème d'optimisation stochastique (Fig. 1) avec une stratégie optimale $V_{\text{opt}}^{(N)}(m(0),T,\beta)$ maximisant le nombre de ports analysés par l'attaquant. Nous résolvons ce problème d'optimisation introduit dans [GGY23] en considérant sa forme relâchée[††] dénotée $V_{\text{rel}}(m(0),T,\beta)$. Comme démontré dans [GGY23], $V_{\text{opt}}^{(N)}(m(0),T,\beta)$ converge vers $V_{\text{rel}}(m(0),T,\beta)$ lorsque $N \to \infty$. Les stratégies obtenues peuvent être adaptées pour un N fini [GGY23]. Nous résolvons cette optimisation en établissant la stratégie optimale et une stratégie approchée basée sur le waterfilling qui effectue en premier les actions les plus récompensantes : pour un budget donné (représenté par un sceau rempli d'eau) et étant donné un ensemble de conteneurs (ports de types i) à remplir caractérisés chacun par une rentabilité, la stratégie de waterfilling consiste à verser l'eau du sceau dans les conteneurs les plus rentables jusqu'à ce qu'ils soient pleins, puis (s'il en reste) dans les conteneurs moins rentables, jusqu'à ce qu'il n'y ait plus d'eau. Les résultats de la

‖. Une alternative serait de choisir une récompense avec différentes valeurs.
**. Ces équations sont écrites en moyenne pour simplifier les notations, cela ne change pas les résultats.
††. Plutôt que de considérer la contrainte (2e), nous considérons $\mathbb{E}(\sum_i\sum_s Y_{s,1,i}^{(N)}(t)) \leq \alpha$

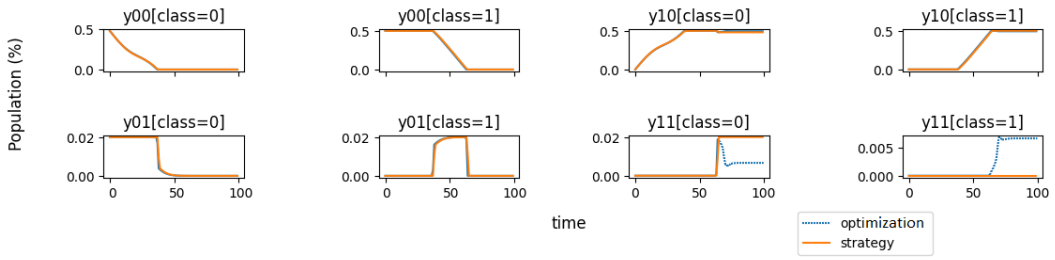


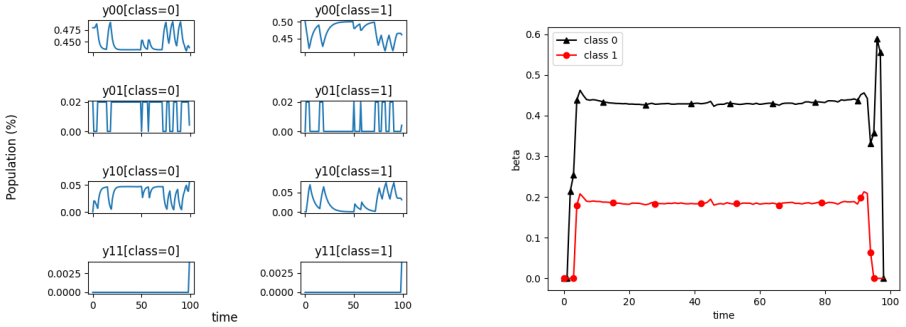
FIGURE 2 : Waterfilling (orange) et optimisation (bleue) de processus de décision markoviens faiblement couplés relaxés pour une valeur de β fixée, 2 types de port et $y_{sa}[class = i]$ avec s désignant l'état des bras, a l'action effectuée, et i le type de bras.

stratégie optimale et de Waterfilling (Figure 2) sont presque identiques à moins que l'attaquant maintienne l'analyse en cours ($y_{11}(t)$) car la stratégie de Waterfilling se focalise toujours sur les ports à forte récompense alors que l'optimisation n'en privilégie aucun quand ces ports sont en cours d'analyse.

Stratégie optimale du défenseur - Pour trouver la stratégie MTD optimale de $V_{\text{rel}}(m(0), T; \beta)$, un algorithme min-max constitué des 2 étapes suivantes s'exécute jusqu'à ce que la solution converge :

- Résoudre le problème d'optimisation ou calculer le gain de la stratégie Waterfilling en fonction de β.
- Effectuer une descente de gradient sur β en utilisant le coût défini par (1a).

Cette approche s'apparente à une descente de gradient, à moins que V_{rel} ne soit pas différentiable par rapport à β. Pour optimiser β, nous avons développé une double optimisation en utilisant les bibliothèques cvxpylayers et pytorch : 1200 variables et 200 paramètres (correspondant aux valeurs de $\beta_i(t)$ variant au cours du temps) sont nécessaires. Lors de nos évaluations, nous considérons $T = 100$, $\alpha = 0.02$, un coût associé à β égale à 10 et deux types de bras : l'un ayant une probabilité $q_0 = 0.7$ et l'autre de $q_1 = 0.3$. L'entraînement prend environ 10 minutes et comme le montre la Figure 3b, la valeur de β tend à être constante en raison du manque d'information du défenseur qui effectue une action sans connaître l'état des ports. L'attaquant (Figure 3a) quant à lui active de manière alternée les 2 types de ports.



(A) Fraction optimale d'activation des bras.

(B) Valeurs optimisées de β pour un coût linéaire.

FIGURE 3 : Solutions du problèmes d'optimisation lorsque la stratégie et β sont tous deux optimisés.

Remerciements : Ce travail a bénéficié d'une aide de l'Agence Nationale de la Recherche portant la référence ANR-23-PECL-0009 (Project TrustinClouds).

Références

[GGY23] N. Gast, B. Gaujal, and C. Yan. Linear program-based policies for restless bandits : Necessary and sufficient conditions for (exponentially fast) asymptotic optimality. *Mathematics of Operations Research*, 2023.