# Research Master's Report

## Optimal attack strategies against honeypot traps in microservice chains
## December 1, 2024

**Grégory Eyraud**

*Laboratoire d'Informatique d'Avignon (UPRES 4128)*
*339, chemin des Meinajaries*
*Agroparc – B.P. 1228*
*F-84911 Avignon cedex 9*
*gregory.eyraud@alumni.univ-avignon.fr*

ABSTRACT. *This report details the work realized during my research internship at the LIA. In the first part, I explain the problem addressed, i.e. the use of optimized cyber deception strategies to maximize the time and material resources required by the attacker to reach their target. Then, I detail the state of the art on such systems, following the system modeling and the various simulations carried out, and conclude with an analysis and comparison of my results.*

RÉSUMÉ. *Ce présent rapport détaille le travail effectué lors de mon stage de recherche réalisé au LIA. J'explique dans une première partie le problème traité, à savoir utiliser des stratégies de cybertromperies optimisées afin de maximiser les ressources temporelles et matérielles nécessaires à l'attaquant pour atteindre sa cible, je détaille ensuite l'état de l'art sur de tels systèmes, je présente ensuite la modélisation du système et les différentes simulations réalisées, et je termine par une analyse et une comparaison de mes résultats.*

KEYWORDS: *Markov chain, Optimization, Security, Networks, Stochastic process, MDP, POMDP.*

MOTS-CLÉS : *Chaîne de Markov, Optimisation, Sécurité, Réseaux, Processus stochastique, Marche aléatoire, MDP, POMDP.*

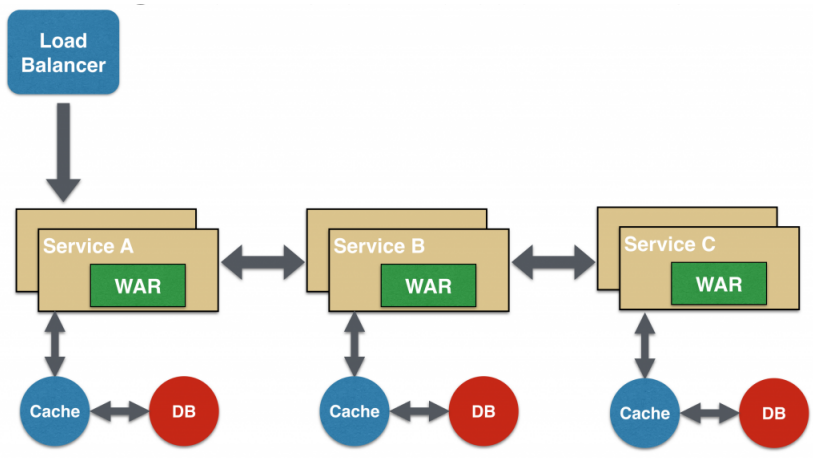


Figure 1: A representation of a chained pattern, where each microservices take the output of the last one as an input [GEE ]

## 1. Preamble

This internship has been done from February to July 2024 at the Laboratoire Informatique d'Avignon (LIA). Weekly meetings have been planned to discuss about the current progression and to planify the next tasks, and regular meetings during the weeks allowed us to solve some problems around the mathematical modeling, or to check the results obtained with the numerical simulations. Trello was use to schedule my tasks, and Git was very useful to share and save my code.

## 2. Problem's presentation

We set down a system such as a microservices chain, which could be represented as an application composed of several microservices (i.e. a group of multiple services where each ones are used for a specific task in the system) interacting with each other with HTTP/Rest protocols. It is possible to see a representation of a chained microservices with Fig 1. Furthermore we consider 2 sides:

*Intruder.* An intruder who aim to reach sensible data, where we assume are always stored or used in a specific microservice in the system, where, in our problem, is always located at the end of the microservice chain, i.e. the M-th microservice. To hit their target, the attacker use what is called lateral movement [GRA 21], a set of techniques allowing a cybercriminals to move in the chain by gaining administrators' privileges. They start from an entry point, and in our case is the first chain's microservice, and moves from one microservice to another one by one. So to hit the target M, they need to move M times in the good direction. To avoid to be trapped by the

defender, where their strategy is described next, they have a probability $p$ to back off at their starting point. To simulate the attacker moving in the chain, we assume that they can't make the difference between the good path and the honeypots when they explore, but they can return to the entry point after moving in the chain, so our problem is a controlled random walk problem [ALE 13].

*Defender.* The defender, which could be the CSIRT (Computer Security Incident Response Team), creates honeypots depending a budget K (such as computational resources) to delay the attacker and to gain time to neutralize the attacker. These resources are equally shared between all the microservices except the target, giving L possible ways for the attacker for each microservices (and L-1 give the number of honeypots for each of these states).

This 2 sides have opposite objectives: the intruder wants to minimize his time to hit the target, and the defender wants to maximize it.

As the intruder doesn't always know the length of the microservice chain, they can use a priori distribution representing their conjecture on the actual length. If they know it, they can simply back off after M steps. These 2 different conjecture make them an intelligent attacker. Otherwise, we consider the attacker as a naive intruder if they back off without optimizing their movements.

## 3. State of the art

Lateral movement is a well-known method used by cybercriminals to extract data from systems. This technique is very effective, and a lot of companies specialized in cybersecurity are trying to make industry aware if the dangers they could be exposed, like [CLO ] or [CRO ], which are famous actors.

As said in section 2, our simulated attacker move following a controlled random walk process, where the randomness depends on the lack of knowledge of the attacker on the presence of honeypots. Currently, some works have already been done to detect intrusion like in [YAN 23], where the authors tried to detect stepping-stones intrusion, i.e the first compromised computer. There are also some studies involving random walks and the maximum hitting time in dynamic graphs like the article [SAU 19] written by Thomas Sauerwald and Luca Zanetti, where they tried to study the difference in complexity between static graph and dynamic graph, by trying to reproduce networks' behaviour, which are often subject to some connectivity changes between devices.

## 4. Modeling

Each state of the chain of micro-services is denoted by $i \in \{1, \ldots, M\}$. At each state $i$ a deception component is created, i.e. $n(i)$ new edges are proposed in order to deceive the attacker on the right direction to take. Therefore, new nodes denoted

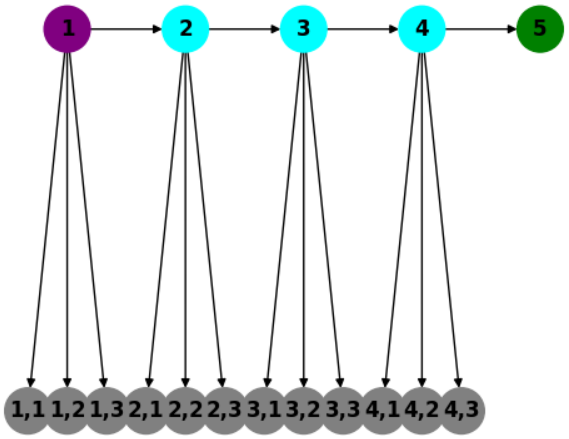


Figure 2: Uniform defense with M = 5 and K = 12. From the corresponding explanation, we have $\lfloor \frac{12}{5-1} \rfloor = 3$ honeypots allocated for each states, except the target, giving L = 4.

by $(i, 1)$ to $(i, n(i))$ define a new branch with depth 1 from initial service $i$. The number of components of this vector indicates how deep the intruder has entered the deceptive part of the system from a micro-service. For example, the node identified as $(2, 1, 3, 3)$ says that the intruder is at a depth 4 from micro-service 2. In reality, we used this representation only to follow the attacker in the chain, but we didn't really needed to use it for our modeling.

We assume that the intruder explores the chain of microservices using a simple random walk process, i.e. at each service $i \in \{1, \ldots, M\}$ and extra nodes/fake services. In the other hand, at each level $i \in \{1, \ldots, M-1\}$, the defender creates $\lfloor \frac{K}{M-1} \rfloor$ honeypots. Figure 2 is an example of a possible implementation.

We consider hereafter the reward corresponding to two specific attacker strategies: 1) the naive strategy where the attacker has a fixed probability $p$ to return at state $s_1$, 2) a threshold policy where the attacker explore the chain with a maximum span of $\sigma$ steps forward.

### 4.1. *Simple strategy*

For each state $1 < i < M$, the attacker returns with probability $p$ to state $s_0$, and a probability $1 - p$ to continue the exploration. In other words, they have a probability

$p$ to return at 1 (i.e the entry point where he starts his exploration), and as $M$ is an absorbing state, it's then possible to write:

$$P_r\left\{\text{successful path to target}\right\} = \frac{1}{L}\left(\frac{1-p}{L}\right)^{M-2} \tag{1}$$

Where $L$ is the number of branches per stage ($L-1$ is the number of honeypots). The probability to hit node $M$ from node 1 is the product of the probability to choose the good path ($L$ possibilities, only one is good) by the probability to continue the exploration M-2 times. We can simplify it as:

$$P_r\left\{\text{successful path to target}\right\} = \frac{(1-p)^{M-2}}{L^{M-1}} \tag{2}$$

Then we define $\tau_{1M}$ the number of steps to hit node M starting at the entry node as:

$$\tau_{1M} = (M-1) + \sum_{k=0}^{N_b} \tau_b(k) \tag{3}$$

With $N_b$ the number of backoff cycles before hitting the target, and $\tau_b(k)$ the length of the backoff cycle k. Then we should know how many time (in terms of expected number of hops) an intruder will take to hit M. From 3, we have to take the average number of steps for $N_b$ cycles, pondered with the probability to have $N_b$ cycles. However, this number of cycle can be infinite. Then, we have:

$$\begin{aligned}
\mathbb{E}[\tau_{1M}] &= (M-1) + \sum_{h=0}^{+\infty} \mathbb{E}[\sum_{k=1}^{N_b} \tau_b(k) | N_b = h] P_r\left\{N_b = h\right\} \\
&= (M-1) + \sum_{h=0}^{+\infty} \mathbb{E}[\sum_{k=1}^{h} \tau_b(k) | N_b = h] P_r\left\{N_b = h\right\} \\
&= (M-1) + \sum_{h=0}^{+\infty} h\mathbb{E}[\tau_b(1)] P_r\left\{N_b = h\right\} \\
\mathbb{E}[\tau_{1M}] &= (M-1) + \mathbb{E}[\tau_b(1)] \cdot \mathbb{E}[N_b],
\end{aligned} \tag{4}$$

with $\tau_b$ the length of a backoff cycle. We now need to calculate $\mathbb{E}[\tau_b(1)]$ and $\mathbb{E}[N_b]$. To do so, we're starting by defining $P_r\left\{N_b = h\right\}$ as:

$$P_r\left\{N_b = h\right\} = \rho(1-\rho)^h \tag{5}$$

With $\rho = P_r\left\{\text{successful path to target}\right\} = \frac{(1-p)^{M-2}}{L^{M-1}}$. We then use the Probability-generating function to estimate $\mathbb{E}[N_b]$ :

$$G_{N_b}(z) = \sum_{h=0}^{+\infty} z^h P_{N_b}(h) = \sum_{h=0}^{+\infty} \rho z^h (1-\rho)^h$$
$$G_{N_b}(z) = \frac{\rho}{1-(1-\rho)z} = \frac{[1-\frac{(1-\rho)^{M-2}}{L^{M-1}}]}{1-[1-\frac{(1-\rho)^{M-2}}{L^{M-1}}]z}$$

The mean value of $N_b$, i.e, $\mathbb{E}\left[N_b\right] = m_{N_b}$ can be determine by the differentiation of $G_{N_b}(z)$ and evaluate it with $z = 1$, such as:

$$m_{N_b} = \frac{\partial}{\partial z} G_{N_b}(z)\Big|_{z=1} = \frac{\rho(1-\rho)}{[1-(1-\rho)z]^2}$$
$$m_{N_b} = \frac{1-\rho}{\rho} \tag{6}$$

So far, we have:

$$\mathbb{E}[\tau_{1M}] = (M-1) + \mathbb{E}[\tau_b] \cdot (\frac{1}{\rho} - 1) \tag{7}$$

We now need to estimate $\mathbb{E}[\tau_b]$. First:

$$P_{\tau_b}(h) = P_r\left\{\tau_b = h\right\} = \rho(1-\rho)^{h-2}, \qquad h = 2, 3, ... \tag{8}$$

Then, we once more need to use the Probability-generating function:

$$G_{\tau_b}(z) = \mathbb{E}[z^{\tau_b}] = \sum_{k=2}^{+\infty} \rho(1-\rho)^{k-2} z^k = \rho z^2 \sum_{k=0}^{+\infty} [(1-\rho)z]^k \tag{9}$$
$$G_{\tau_b}(z) = \frac{\rho z^2}{1-(1-\rho)z}$$

As before, we use the differentiation and evaluate $G_{\tau_b}(z)$ with $z = 1$ to determine $\mathbb{E}[\tau_b]$:

$$m_{\tau_b} = \frac{\partial}{\partial z} G_{\tau_b}(z)\Big|_{z=1} = \frac{z\rho z[1-(1-\rho)z] + \rho z(1-\rho)}{[1-(1-\rho)z]^2} \tag{10}$$
$$m_{\tau_b} = \frac{1+\rho}{\rho}$$

We can finally write the full expression of $\mathbb{E}[\tau_{1M}]$ as:

$$\mathbb{E}[\tau_{1M}] = (M-1) + \frac{1 + (\frac{(1-p)^{M-2}}{L^{M-1}})}{\frac{(1-p)^{M-2}}{L^{M-1}}}[\frac{1}{\frac{(1-p)^{M-2}}{L^{M-1}}} - 1] \tag{11}$$

With this expression, we're now able to estimate the mean time the attacker will need to hit M according to M, L and p. Now, we can determine an optimal value for p, named p*, where the attacker optimizes their time to hit M. For this, we differentiate $\mathbb{E}[\tau_{1M}]$ on p:

$$p* : \frac{\partial}{\partial p}\mathbb{E}[\tau_{1M}] = -\frac{1}{p^2}[\frac{L^{M-1}}{(1-p)^{M-2}} - 1] + \frac{1+p}{p}[\frac{L^{M-1}(M-2)}{(1-p)^{M-1}}]$$
$$\frac{\partial}{\partial p}\mathbb{E}[\tau_{1M}] = 0, \qquad \frac{1}{p}[\frac{L^{M-1}}{(1-p)^{M-2}} - 1] = (1+p)(\frac{L^{M-1}(M-2)}{(1-p)^{M-1}} \tag{12}$$

We have:

$$f(x) = L^{M-1} - (1-p)^{M-2} \tag{13}$$
$$g(x) = \frac{p(1+p)}{1-p} L^{M-1}(M-2) \tag{14}$$

And $p* : \quad f(p*) = g(p*)$

We need to study $f(p)$ and $g(p)$ to solve the system:

– $f$ is an increasing concave, $f(0) = L^{M-1} - 1$ and $f(1) = L^{M-1}$

– $g$ is an increasing convex, $g(0) = 0$ and $g(1) = +\infty$

We note $p_{min} \leq p* \leq p_{max}$, where $p_{min}$ solves for $g(p) = L^{M-1} - 1$ and $p_{max}$ solves for $g(p) = L^{M-1}$. We can also say:

$$f(p) = L^{M-1} - \chi(p) \tag{15}$$

with

$$\chi(p) = \begin{cases} 1 & p = 0 \\ 0 & p = 1 \end{cases} \tag{16}$$

The system is now:

$$L^{M-1} - \chi(p) = \frac{p(1-p)}{1-p} L^{M-1}(M-2) \tag{17}$$

If we consider $\frac{p(1+p)}{1-p} = A$, then from the above expression:

$$A = \frac{L^{M-1} - \chi(p)}{(M-2)L^{M-1}} \tag{18}$$

And:

$$p = -\frac{A+1}{2} + \sqrt{\frac{A+6A+1}{2}} \tag{19}$$

By series expansion $(1+x)^\alpha = \sum_{n=0}^{+\infty} \binom{n}{\alpha} x^\alpha = 1 + \alpha x + \Theta(x)$, we estimate p for M $-> +\infty$:

$$p \leq -\frac{A+1}{2} + \sqrt{\frac{A^2+2A+1+4A}{4}} \leq -\frac{A+1}{2} + \frac{A+1}{2}\sqrt{1 + \frac{4A}{(A+1)^2}}$$

$$p \leq \frac{A}{A+1} \tag{20}$$

With our above expression for A, we end with:

$$\begin{aligned} p &\leq \frac{\frac{L^{M-1} - \chi(p)}{(M-2)L^{M-1}}}{\frac{L^{M-1} - \chi(p)}{(M-2)L^{M-1}} + 1} \\ &\leq \frac{L^{M-1} - \chi(p)}{(M-1)L^{M-1} - \chi(p)} \\ p &\leq \frac{1}{M-1} \end{aligned} \tag{21}$$

The mathematical modeling finished, we performed the numerical modeling (i.e we reproduce 10000 episodes where an attacker tries to hit the target following a chain of microservices and a given $p$) to verify our results. First, we verified the average number of movements needed by the attacker following different values of $p$, with $M$ and $L$ fixed. As shown as example in figure 3 and 5, the optimal p is equal to 0.25 for a microservices chain's length M = 5, and is equal to 0.1 for M = 9. These two results confirms our assumption with equation 21: the optimal probability for a simple attacker to hit the target $p*$ depends on $M$, and can't exceed $\frac{1}{M-1}$, but can be lower (as shown with M=9, $p^*$ is lower than $\frac{1}{8}$). It also confirms another assumption: $L$ has no importance for the attacker, in the way that the optimal probability $p*$ is only depending on M. Obviously, a greater value of L will raise the average number of steps needed for the attacker.

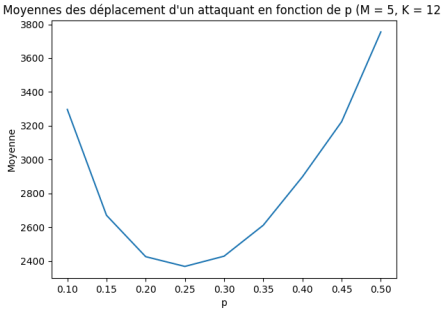


Figure 3: Attacker's average movements in function of p (M = 5 and L = 4)

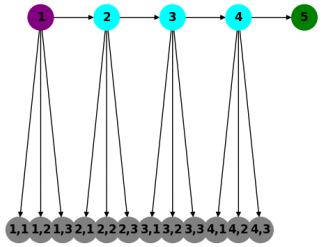


Figure 4: Microservice chain's representation for M = 5 and L = 4

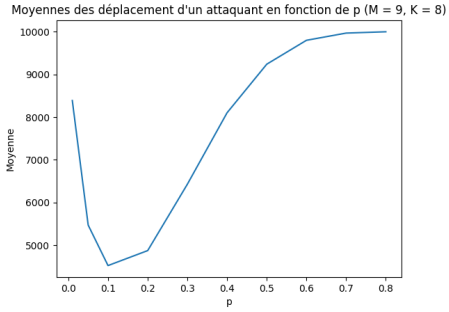


Figure 5: Attacker's average movements in function of p (M = 9 and L = 2)

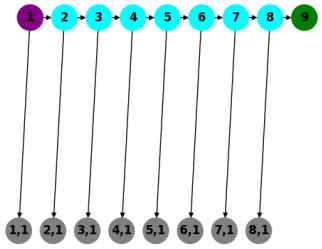


Figure 6: Microservice chain's representation for M = 9 and L = 2

We also run some simulations to verify equation 11. We first implement this equation and draw the corresponding graphic (figure 7), then we run a simulation for the same problem and and count the average number of movements needed by the attacker

to hit the target. By the superposition that could be seen at figure 8, we can validate our previous equations, and conclude with this first part.

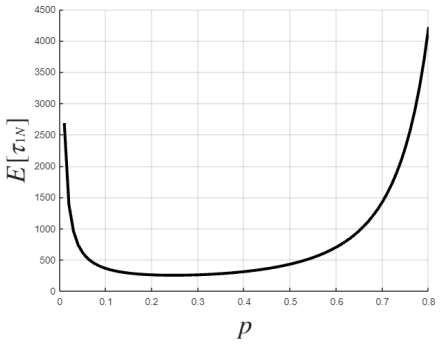


Figure 7: The average number of movements expected by the attacker for a given chain

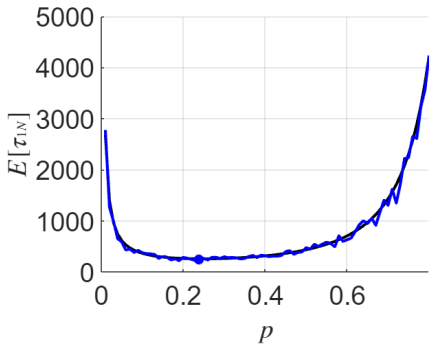


Figure 8: The equation's and simulation's curve superposition for a same microservice chain (with the simulation's curve in blue).

### 4.2. *Sophisticated strategy*

Then, we study an attacker with a more sophisticated strategy who uses a probability $p$ to return at state 1 which is evolving according to the length of the cycle. In this type of problem, we consider p which could be only strictly increasing or decreasing. For each case, we can write from our previous equation (7):

$$\mathbb{E}[\tau_{1M}] = (M-1) + \mathbb{E}[\tau_b] \cdot (\frac{1}{\rho} - 1) \tag{22}$$

But now, we have a probability to return to state 1 $p$ which is no longer constant. Instead, we have a $p_k$ for each $k$ between 2 and $M-1$, so our $\rho$ is not the same as with a constant $p$. Then, we can consider $q_k$ as:

$$q_k = \prod_{h=2}^{k-1} (1 - p_h) \tag{23}$$

Which is the probability to don't back off for a path length equal to k. $k$ starts at 2 because we assume $p_1 = 0$, and ends at $M-1$ because we assume $p_M = 1$. Then, we can write our new $\rho$ as:

$$\rho = \frac{q_M}{L^{M-1}} \tag{24}$$

So our previous equation for $\mathbb{E}[\tau_{1M}]$ is now:

$$\mathbb{E}[\tau_{1M}] = (M-1) + \mathbb{E}[\tau_b] \cdot [\frac{L^{M-1}}{\prod_{h=2}^{M-1}(1-p_h)} - 1] \tag{25}$$

And we estimate $\mathbb{E}[\tau_b]$ as:

$$\mathbb{E}[\tau_b] = \sum_{k=2}^{+\infty} k P_r \{\tau_b = k\} \tag{26}$$

As before, we first need $P_r \{\tau_b = k\}$, the probability to have a backoff cycle of length k:

$$\begin{aligned} P_r \{\tau_b = k\} &= p_k q_k \\ P_r \{\tau_b = k\} &= p_k \prod_{h=2}^{k-1}(1-p_h) \end{aligned} \tag{27}$$

So in final we have for $\mathbb{E}[\tau_{1M}]$:

$$\mathbb{E}[\tau_{1M}] = (M-1) + \sum_{k=2}^{+\infty}[k p_k \prod_{h=2}^{k-1}(1-p_h)] \cdot [\frac{L^{M-1}}{\prod_{k=2}^{M-1}(1-p_k)} - 1] \tag{28}$$

### 4.3. *Deterministic policy*

Assuming the attacker is following a threshold policy, with a fixed $\sigma > 0$ as the threshold, and the microservice chain's length M is taken following a Poisson distribution. It holds:

$$a = -1 - \gamma - ... - \gamma^{\sigma} \tag{29}$$

Where a is the discounted cost, with parameter $0 < \gamma < 1$, for the first cycle when the attacker didn't hit the target. Furthermore, $a$ is a geometric sequence where the first value is -1 and has the common ratio $\gamma$. Then we can write $a$ as:

$$a = -\frac{1-\gamma^{\sigma+1}}{1-\gamma} \tag{30}$$

Then, for the following cycles where the attacker doesn't hit the target, we have as discounted cost for the:

Second cycle : $a\gamma^{\sigma+1}$

Third cycle : $a\gamma^{2\sigma+2}$

k-th cycle :$a\gamma^{(k-1)(\sigma+1)}$

As M is taken randomly, we have 2 possibilities:

4.3.1. $\sigma < M$

Then the attacker never hit the target, so the discounted reward can be easily simplified as:

$$a_\infty = \frac{-1}{1-\gamma} \tag{31}$$

with $a_\infty$ being the discounted reward when the attacker have never hit the target after an infinite number of step.

4.3.2. $\sigma \geq M$

Then the attacker will hit the target after Nb cycles. Let S be the number of step needed to hit the target with Nb tries, and defined as:

$$S = \sum_{i=1}^{Nb-1} (\sigma + 1) + M \tag{32}$$

Note that the cost for hitting the target is a big positive value named $R_M$, and -1 otherwise. The total reward $G(\sigma)$ is defined as:

$$G_1(\sigma) = \sum_{t=0}^{S-1} -\gamma^t + R_M\gamma^S$$

$$G_1(\sigma) = -\frac{1-\gamma^S}{1-\gamma} + R_M\gamma^S \tag{33}$$

However, S is depending on the random variable Nb, such as $P_r\{Nb = h\} = P_{success}(1 - P_{success})^{h-1}$, with $P_{success} = (\frac{1}{L})^{M-1}$. Then it holds:

$$\mathbb{E}[G_1(\sigma)] = \sum_{j=1}^{+\infty} P_r\{Nb = j\} \cdot [-\frac{1-\gamma^{(j-1)(\sigma+1)+M}}{1-\gamma} + R_M\gamma^{(j-1)(\sigma+1)+M}] \tag{34}$$

4.3.3. *In general*

For any $\sigma$, the total cost need for the attacker to hit the target is:

$$G(\sigma) = \sum_{m=1}^{\sigma} G_1(\sigma) P_r\{M = m\} + \sum_{m=\sigma+1}^{+\infty} a_\infty P_r\{M = m\}$$

$$G(\sigma) = \sum_{m=1}^{\sigma} G_1(\sigma) P_r\{M = m\} + a_\infty P_r\{M > \sigma\} \tag{35}$$

Where $G_1(\sigma)$ refers to equation (34). In this general case, we add two possibilities discussed earlier weighted by their probability to occur.

We implement equation 35 to understand the importance of the a priori Poisson distribution on M, and as we can see in figure 9, $\lambda = 2$ is optimal for $\sigma < 10$, then, bigger value of lambda (like 14 or 15) seems better. However, more simulations are needed to confirm these results.

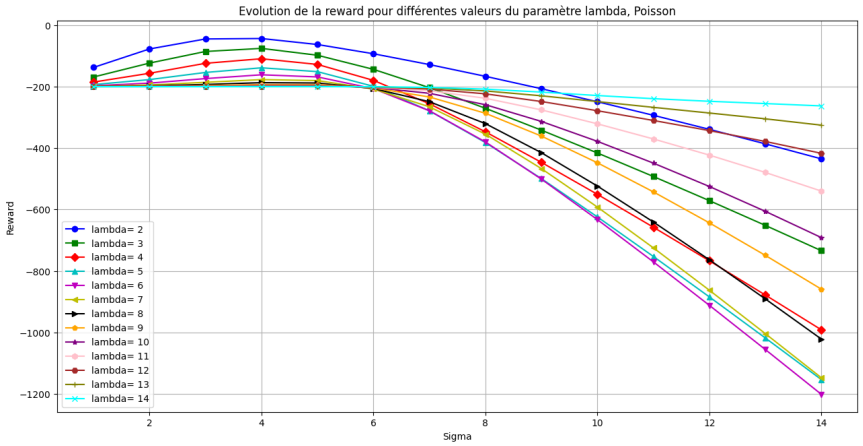


Figure 9: The evolution of the mean reward for different $\lambda$ following $\sigma$

## 4.4. *POMDP*

Our discrete time POMDP is defined by the tuple $\langle \mathcal{S}, \mathcal{A}, R, \mathcal{T}, \Omega, O, \lambda \rangle$, as described in the following.

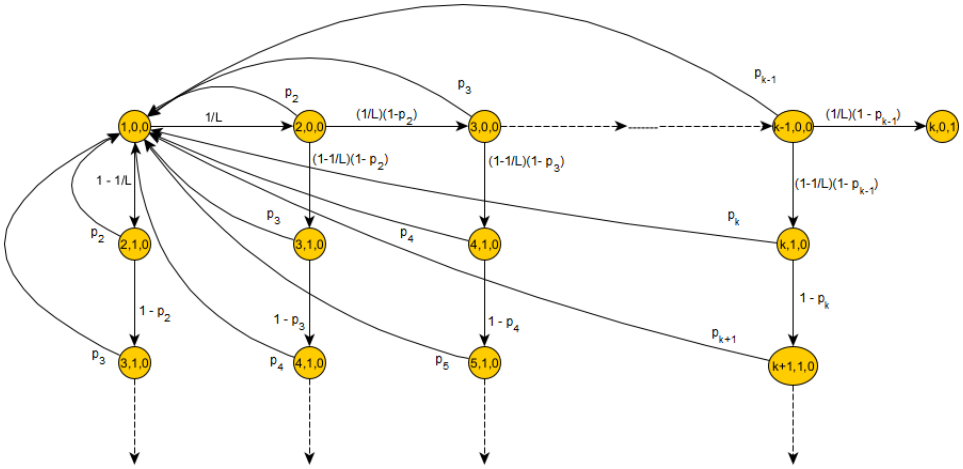


Figure 10: The transition probability diagram for each state in the POMDP modeling the attacker following a fixed politic $\pi$.

*State space.* The state space is the countable set $\mathcal{S} = \{S\}$ where $S = (Y, \chi, T)$ where $Y$ is the length of the path, $\chi \in \{0, 1\}$, where 1 means the attacker is in a honeypot, and 0 otherwise, and $T$ is equal to 1 if the target has been hit, or 0 otherwise.

*Action space.* The action space is the set of actions $\mathcal{A} = \{0, 1\}$ where 0 is the action to take a further exploration step, and 1 is fall back to state $s_0 = (0, 0, 0)$, i.e., to the origin. Clearly, for $\mathcal{A}(s_0) = \{0\}$, the only action possible is to move forward, for $T = 1$, $\mathcal{A}(s) = \emptyset$ and for $T = 0$ and $Y > 0$, $\mathcal{A}(s) = \{0, 1\}$.

*Reward function.* The reward R(s,a) is equal to -1 $\forall a \in \mathcal{A}$, from any $s \in \mathcal{S}$ which is not absorbing, when the intruder doesn't hit the target. Otherwise, it is equal to a big positive value. This reward is discounted following a parameter $\gamma$, where $\gamma$ is close to 1 but is not 1.

For the sake of notation subscripts are reserved for the time index. For instance, $S_t = (Y_t, \chi_t, T_t)$ is the state at time $t$, $A_t$ the action taken at time $t$ and $R_{t+1}$ the reward gained in the corresponding transition. Also, all upper-case letters refer to a random process and lower-case letters to realizations.

*Transition probabilities.* The conditional transition probabilities provide the probability that the next state is $s'$ given the current state $S_t$ and an action $A_t$. We have the figure 10 to illustrate a fixed politic $\pi$, where $p_k$ represents the probability to choose the action 0, i.e. to back off at state $s_0 = (0, 0, 0)$. In fact:

$$\mathbb{P}(S_{t+1} = s' | S_t = s, A_t = 0) = \begin{cases} \frac{1}{L} & \text{if } s' = (k+1, 0, T_{t+1}) \text{ and } s = (k, 0, 0) \\ 1 - \frac{1}{L} & \text{if } s' = (k+1, 1, 0) \text{ and } s = (k, 0, 0) \\ 0 & \text{if } s' = (k+1, 0, T_{t+1}) \text{ and } s = (k, 1, 0) \\ 1 & \text{if } s' = (k+1, 1, 0) \text{ and } s = (k, 1, 0) \end{cases} \tag{36}$$

and it holds $\mathbb{P}(S_{t+1} = (0, 0, 0) | S_t = s, A_t = 1) = 1$.

The transition probabilities are determined by the policy adopted by the attacker. A policy $\pi_t : \mathcal{S} \to [0, 1]^{\mathcal{A}}$ associates to each state $S$ the fallback probability, i.e.,

$$\pi_t(s) = \mathbb{P}\left(A_t = 1 | S_t = s\right)$$

In particular, we define, for the sake of notation, $p_k = \mathbb{P}\left(A_t = 1 | S_t = (k, \chi, 0)\right))$, i.e., the fallback probability when the attacker is at distance $k$ from $s_0$. Note that there is no fallback when the target is reached. The transition diagram for a particular policy $\pi$ is represented by the weighed graph of Fig. 10.

*Observation space.* The set of observations is represented by the set of pairs $O_t = (Y_t, T_t)$, where $Y_t$ is the distance from origin state $s_0$ and $T_t$ is the indicator that the target is attained.

*Observation probabilities.* The conditional observation probabilities $\mathbb{P}(O_{t+1} = \overline{o} | S_{t+1} = s', A_t = 0)$ will permit us to determine the belief dynamics. In particular, the case $T_{t+1} = 1$ is uninteresting. Instead, in all the other cases the uncertainty about being in a honeytrap or not is captured by the following

$$\mathbb{P}(O_{t+1} = \overline{o} | S_{t+1} = s', A_t = 0) = \begin{cases} F_M(k) \text{ if } s' = (k+1, 0, 0) \\ \\ 1 \text{ if } s' = (k+1, 1, 0) \end{cases} \tag{37}$$

Where $F_M(k) = \mathbb{P}(M > k)$ is the a priori probability that M is strictly greater than k+1 and is defined as $\mathbb{P}(M > k+1) = \sum_{k=s'_k}^{+\infty} \beta(k)$ where $\beta(k)$ is the belief for M and is introduced in 4.4.1.

Our agent have a belief $b$ in being in the actual state, and a priori distribution $\beta$ on M's real value.

#### 4.4.1. *Belief b*

The belief $b(s')$ is updated following:

$$b(s') = \frac{\mathbb{P}\left(o | s', a\right) \sum_{s \in S} \mathbb{P}(s' | s, a) b(s)}{\sum_{s' \in S} \mathbb{P}(o | s', a) \sum_{s \in S} \mathbb{P}(s' | s, a) b(s)} \tag{38}$$

At state $s_t = (k, 0, 0)$, with the action $a_0$, we have two possibilities: $s_{t+1} = (k+1, 0, 0)$ or $s_{t+1} = (k+1, 1, 0)$. Here, we are only interested in the cases where

$T_{t+1} = 0$, because otherwise, it means that the target was hit, and that we no longer need to calculate the belief. So we have two beliefs to calculate: $b((k+1,0,0))$ and $b((k+1,1,0))$. Let's take an example with the first iteration ($s_0 = (0,0,0)$):

$$\begin{cases} b((1,0,0)) = \frac{F_M(1)\frac{1}{L}b(s_0)}{N_1+N_2} \\ b((1,1,0)) = \frac{1\times(1-\frac{1}{L})b(s_0)}{N_1+N_2} \end{cases}$$

Where $N_1 = F_M(1)\frac{1}{L}b(s_0)$ and $N_2 = (1-\frac{1}{L})b(s_0)$. As $b(s_0) = 1$, we have:

$$\begin{cases} b(1,0,0) = \frac{\frac{F_M(1)}{L}}{\frac{F_M(1)}{L}+(1-\frac{1}{L})} \\ b(1,1,0) = \frac{1-\frac{1}{L}}{\frac{F_M(1)}{L}+(1-\frac{1}{L})} \end{cases}$$

$$\begin{cases} b(1,0,0) = \frac{F_M(1)}{F_M(1)+(L-1)} \\ b(1,1,0) = \frac{L-1}{F_M(1)+(L-1)} \end{cases} \tag{39}$$

Furthermore, $\forall k \in \{0,1,...\}$:

$$\begin{cases} b(k+1,0,0) = \frac{N_{1,k+1}}{D_{k+1}} \\ b(k+1,1,0) = \frac{N_{2,k+1}}{D_{k+1}} \end{cases} \tag{40}$$

Where $N_{1,k+1} := F_M(k+1)\frac{1}{L}b(k,0)$, and $N_{2,k+1} := (1-\frac{1}{L})b(k,0) + b(k,1)$, and $D_{k+1}$ is defined by the iteration $D_{k+1} = LD_k(N_{1,k+1} + N_{2,k+1})$ with $D_0 = 1$.

Now, we want to be able to write $b(s_t)$ in closed form for every $s_t = (Y_t, \chi_t, T_t) \in \mathcal{S}$. As the calculation of each belief is recursive, and from the previous property, we can write the belief such as:

**Proposition 1** *It holds*

$$\begin{cases} b(k+1,0,0) = \frac{\prod_{h=0}^{k+1} F_M(h)}{D_{k+1}} \\ \\ b(k+1,1,0) = \frac{\sum_{h=0}^{k} L^h(L-1)\prod_{j=0}^{k-h} F_M(j)}{D_{k+1}} \end{cases} \tag{41}$$

*where* $D_{k+1} = \prod_{h=1}^{k+1} F_M(h) + [\sum_{h=0}^{k} L^h(L-1)\prod_{j=0}^{k-h} F_M(j)]$.

PROOF We have already proven the statement for k=1. Now let us verify that the induction step holds for every $\forall k > 1$. In fact from (38) we can write

$$b(k+1,0,0) = \frac{F_M(k+1)\frac{1}{L}b(k,0,0)}{N_{k+1,1}+N_{k+1,2}} = \frac{\frac{F_M(k+1)\prod_{h=0}^{k}F_M(h)}{LD_k}}{N_{k+1,1}+N_{k+1,2}}$$
$$= \frac{\prod_{h=0}^{k+1}F_M(h)}{LD_k(N_{k+1,1}+N_{k+1,2})} = \frac{\prod_{h=0}^{k+1}F_M(h)}{D_{k+1}} \tag{42}$$

where by assumption

$$\begin{cases} b(k,0,0) = \frac{\prod_{h=0}^{k}F_M(h)}{D_k} \\ b(k,1,0) = 1 - b(k,0,0) \end{cases} \tag{43}$$

We have proven the first part of the statement. Now, by definition:

$$b(k+1,1,0) = 1 - \frac{\prod_{h=0}^{k+1}F_M(h)}{D_{k+1}}$$
$$b(k+1,1,0) = \frac{\sum_{h=0}^{k}L^h(L-1)\prod_{j=0}^{k-h}F_M(j)}{D_{k+1}} \tag{44}$$

which concludes the proof. ◇

4.4.2. *A priori distribution $\beta(k)$ on $M$*

The conjectural distribution on M, named $\beta(k)$, is not updated in the time but is used as a belief from the intruder about M, the number of microservices. It is fixed at the beginning, and could change the agent's behavior by taking part in the belief b's calculation. It's supposed to have a finite support, since M is supposed to be finite: $\mathbb{P}(M=k) = \beta_k$ for $k = 1, ...\infty$ and $\sum_{k=1}^{\infty}\beta_k = 1$.

As the deterministic policy in 4.3,this conjectural distribution could be a Poisson distribution, where $\lambda$ is the average number of microservices in chains.

4.4.3. *Belief evolution*

**Proposition 2** *$\forall k$, $b(k+1,0,0) < b(k,0,0)$ and $b(k+1,1,0) > b(k,1,0)$.*

PROOF From (41), We have $N_{k+1,1} = \prod_{h=0}^{k+1}F_M(h)$, and $0 \leq F_M(h) \leq 1$. We need to prove that $D_{k+1} - D_k > 0$:

$$D_{k+1} - D_k = (F_M(k+1)-1)\prod_{h=0}^{k}F_M(h) + \sum_{h=0}^{k}L^h(L-1)\prod_{j=0}^{k-h}F_M(j) - \sum_{h=0}^{k-1}L^h(L-1)\prod_{j=0}^{k-h-1}F_M(j)$$
$$= (F_M(k+1)-1)\prod_{h=1}^{k}F_M(h) + L^k(L-1) > 0 \tag{45}$$

$\forall h$, $F_M(h) < 1 < L$, then:

$$\begin{aligned} D_{k+1} - D_k &= L^k(L-1) - (1 - F_M(k+1)) \prod_{h=1}^{k} F_M(h) \\ &> L^k(L-1) - L^k(1 - F_M(k+1)) \\ &> L^k \left[(L-1) - (1 - F_M(k+1))\right] \\ &> L^k \left[L - 2 + F_M(k+1)\right] \geq 0 \end{aligned}$$

because $L > 2$. ◇

4.4.4. *Belief b decline/growth speed*

**Proposition 3** *At state $S_t = (Y_t, \chi_t, T_t)$, if the next action taken is 0, i.e. take a further exploration step, then the belief to be in a honeypot at time t+1 is strictly increasing by a factor $\overline{\alpha} \leqq 1 + \frac{1}{L}$, and the belief to be in the right path is strictly decreasing following the complementary factor $\underline{\alpha} \leqq \frac{1}{L}$.*

PROOF Given equation (41), we have $\forall k$ the approximation:

$$b(k+1, 0, 0) \sim \frac{F_M(k)}{L^{k+1}(L-1)} \tag{46}$$

$$b(k+1, 0, 0) \sim \frac{F_M(k)}{L^k} \leq \frac{1}{L^k} \tag{47}$$

As $b(k+1, 0, 0) + b(k+1, 1, 0) = 1$, we have by equality the factor $\overline{\alpha} \leq 1 + \frac{1}{L}$, which concludes the proof. ◇

A simple simulation allowed us to check this proof, where we calculate the belief during some iterations following different values of L. As we can see on figure 11, the belief to be in the right path is decreasing faster for $L = 3$ than $L = 2$

4.4.5. *Optimal policy*

An optimal policy $\pi^*$ can be found by using the version of the Bellman equation for the POMDP case [HAN 13, HAU 00]. In particular, the fixed point equation provides the optimal value function $V^*$ as

$$V_t^*(b) = \max_{a \in \mathcal{A}} \left[ r(b, a) + \gamma \sum_{o \in} p(o|a, b) V_{t-1}^*(b_a^o) \right] \tag{48}$$

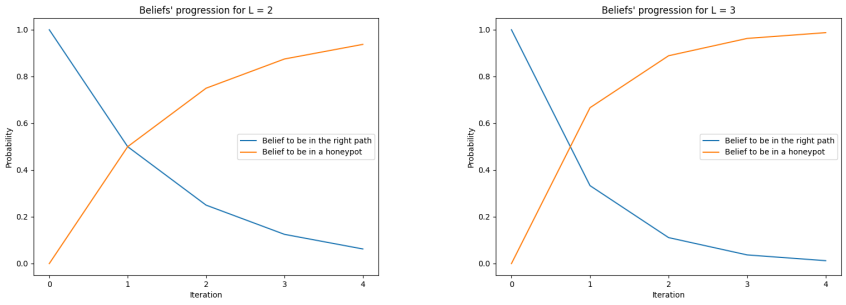


Figure 11: Evolution of the belief for L = 2 (to the left) and L = 3 (to the right). In blue we have the belief to be in the right path, and in orange to be in a honeypot.

where the expected instantaneous reward $r(b,a) = \sum_{s\in\mathcal{S}} R(s,a)b(s)$, and $(b_a^o)$ can be expressed using equation (38), whereas

$$p(o|a,b) = \sum_{s'\in S} \mathbb{P}(o|s',a) \sum_{s\in S} \mathbb{P}(s'|s,a)b(s)$$

As we have only 2 actions possible, the expression simplifies to

$$\begin{aligned} V^*(b) &= \max\left[ r(b,0) + \gamma \sum_{o\in} p(o|0,b)V^*(b_0^o), r(b,1) + \gamma \sum_{o\in} p(o|1,b)V^*(b_1^o) \right] \\ &= \max\left[ \sum_{s\in\mathcal{S}} R(s,0)b(s) + \gamma \sum_{o\in} p(o|0,b)V^*(b_0^o), -1 + \gamma V^*(b_1^o) \right] \end{aligned} \tag{49}$$

From the previous equation, it follows the following structural result

**Theorem 1** *An optimal policy is of threshold type.*

PROOF From Lemma 3, the first term appearing in the maximization is decreasing with $Y$. For the second term, it corresponds to the attacker returning to $s_0$ w.p.1. Hence, if for $b = b((k,\chi_t,0))$ it holds $\pi^*(b) = 1$, then $\pi^*(b') = 1$ for $b' = b((k+1,\chi_t,0))$ as well. However, it's not enough to conclude the proof. For now, we can't prove if $V^*(b_0^o)$ is decreasing with b. ◇

## 5. Result analysis

By analysing our different results, it holds that a naive attacker moving with a fixed probability $p$ to return at state $s_0$, even optimal, is not a good strategy. Indeed, they still have chance to back off just before hitting the target, or going through a honeypot

for a long time. A decreasing $p$ could be better, but if $p$ is decreasing too quickly, there is a high risk to have $p \to 0$, and then being lost forever in a honeypot. The opposite is true for a $p$ increasing too quickly, and where the intruder never hit the target. The threshold policy is an optimal policy, in that the attacker is able to find an optimal sigma following his a priori on M.

## 6. Perspectives

Following on from this internship, there is still some work which has to be done. For example, looking for other types of politics the attacker could use, new way for them to learn an optimal probability $p$ to back off, perform more simulations to confirm some theoretical aspects around the deterministic policy, and looking for an optimal defense following the attacker's strategy (Game theory).

## 7. Acknowledgement

I would like to thank Francesco DE PELLEGRINI and Yezekael HAYEL for their help, their tolerance and kindness toward me, and for helping me to overcome any difficulties I may have encountered.

## 8. References

[ALE 13] ALEXANDER K. S., "Controlled random walk with a target site", *Neural Information Processing Systems (NeurIPS)*, , March 15, 2013.

[CLO ] CLOUDFLARE, "What is lateral movement?", `https://www.cloudflare.com/learning/security/glossary/what-is-lateral-movement/`.

[CRO ] CROWDSTRIKE, "Lateral movement", `https://www.crowdstrike.com/cybersecurity-101/lateral-movement/`.

[GEE ] GEEKS J. C., "Microservice Design Patterns", `https://www.javacodegeeks.com/2015/04/microservice-design-patterns.html`.

[GRA 21] GRANT HO UC SAN DIEGO U. B., DROPBOX, MAYANK DHIMAN D., DEVDATTA AKHAWE FIGMA I. V. P. U. B., INSTITUTE I. C. S., SAVAGE S., GEOFFREY M. VOELKER U. S. D., DAVID WAGNER U. B., "Hopper: Modeling and Detecting Lateral Movement", *USENIX Security Symposium*, , August 11–13, 2021.

[HAN 13] HANSEN E. A., "Solving POMDPs by Searching in Policy Space", , Wed, 30 January 2013.

[HAU 00] HAUSKRECHT M., "Value-Function Approximations for Partially Observable Markov Decision Processes", *AI Access Foundation and Morgan Kaufmann Publishers*, , 2000.

[SAU 19] SAUERWALD T., ZANETTI L., "Random Walks on Dynamic Graphs: Mixing Times, Hitting Times, and Return Probabilities", , March 4, 2019.

[YAN 23] YANG J., WANG L., QIN M., NEUNDORFER N., "Detecting Stepping-Stone Intrusion and Resisting Intruders' Manipulation via Cross-Matching Network Traffic and Random Walk", *Electronics*, , 2023.