

<https://doi.org/10.31891/csit-2026-3-4>

### Andrii PRYTULA

Student of Faculty of Information Technologies and Computer Engineering  
Vinnitsia National Technical University,  
Vinnitsia, Ukraine

<https://orcid.org/0009-0006-9632-0712>

e-mail: [andrik.pritula@gmail.com](mailto:andrik.pritula@gmail.com)

### Leonid KUPERSHTEIN

PhD, Associate Professor of Information Protection Chair

Vinnitsia National Technical University,  
Vinnitsia, Ukraine

<https://orcid.org/0000-0001-6737-7134>

e-mail: [kupershtein.lm@gmail.com](mailto:kupershtein.lm@gmail.com)

Received: 27/07/2026

Accepted: 27/08/2026

Published: 30/09/2026

© Copyright  
2026 by the author(s)



This is an Open Access article distributed under the terms of the [Creative Commons CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)

UDC 004.056:004.85

## RISK-SENSITIVE Q-LEARNING IN A TWO-AGENT MARKOV ATTACK-DEFENSE MODEL

*A risk-sensitive method for forming a web application security testing policy based on entropic-risk Q-learning in a two-agent Markov attack-defense model is proposed. The study addresses the need to detect short critical compromise trajectories under partial observability, noisy estimation of defensive attributes, and stochastic changes in the defensive profile of the environment. Unlike risk-neutral approaches, the proposed method employs a dense reward oriented towards the time to compromise (TTC) and a post-episode evaluation of tail losses in terms of the conditional value at risk at the 0.95 confidence level (CVaR0.95), where the risk-sensitivity parameter  $\beta$  of the entropic-risk operator controls the degree of risk aversion. Two modes of the simulation environment are introduced, namely the controlled environment Env1 and the stochastic environment Env2, which are used to analyze policy stability under different levels of information uncertainty. The ToN-IoT and CIC-IDS2017 datasets were used for the statistical parameterization of the environments. Aggregation of semantically related MAL attack steps into web components reduces the dimensionality of the state representation by a factor of 8–12, which makes the training of the risk-sensitive policy computationally manageable. The obtained results show that the proposed method provides the lowest average time to compromise among all the compared approaches. In Env2 the TTC value decreased from  $36.9 \pm 1.5$  to  $27.9 \pm 1.1$  steps for ToN-IoT and from  $39.7 \pm 1.7$  to  $31.4 \pm 1.3$  steps for CIC-IDS2017, the CVaR0.95 of normalized tail losses decreased by approximately 31 %, and the number of episodes required for policy convergence was reduced by approximately 35.4%. The most effective mode is achieved at  $\beta \approx -0.6$ . The practical significance lies in the possibility of applying the method for automated penetration testing and for evaluating the robustness of web application protection mechanisms.*

*Keywords* reinforcement learning; machine learning; Markov model; risk-sensitive Q-learning; time to compromise; security testing; cybersecurity; cyberattack; cyberdefense; artificial intelligence

### Introduction

The complexity of modern web-oriented systems is growing to multi-level architectures, cloud services, and the dynamic modification of protection mechanisms. Therefore, security testing can no longer be limited to fixed verification scenarios or to a static reachability analysis of vulnerable states. Recent studies emphasize that reinforcement learning makes it possible to model sequential decision making in cybersecurity, although its application is complicated by large state spaces, partial observability, and policy instability in dynamic environments [1]. Beyond reinforcement learning specifically, adaptive information technologies for the automatic updating of threat detection models are also emphasized as a means of overcoming the limitations of static, rule-based approaches in computer systems [3]. It has also been shown that deep reinforcement learning in cybersecurity is promising for modelling multi-step attack and defense scenarios, but it requires dedicated stabilization mechanisms, risk-oriented evaluation, and a correct formulation of the reward [3].

### Related works

The first group of analogues comprises risk-neutral reinforcement learning models in which the policy is optimized mainly with respect to the average expected reward. In [4], Q-learning is used for attack graph analysis and for the approximate evaluation of multi-step compromise scenarios. However, such a formulation does not account for tail losses or for the temporal

criticality of short trajectories. In [5], deep reinforcement learning is applied to the simulation of cyberattacks based on realistic scenarios, yet the learning objective remains oriented towards the overall efficiency of the attack rather than towards the risk-sensitive detection of critical trajectories. In [6], deep reinforcement learning is used for attack path prediction, but the temporal characteristics of compromise are not considered as a separate criterion of risk-sensitive optimization.

The second group comprises models in which the interaction between the attacker and the defender is represented as a game or as a multi-agent process. In [7], self-play of attacker and defender strategies is proposed, although the authors note the difficulty of stabilizing policies in a non-stationary environment. In [8], a multi-agent Markov game for moving target defense (MTD) is constructed, but the main attention is focused on defensive dynamics rather than on TTC-oriented web application security testing. In [9], game-theoretic methods are combined with machine learning for adaptive cyber defense, yet the analysis is carried out mainly at the strategic level and does not detail the mechanism of risk-sensitive Q-function updating. In [10], a Markov game is used to counter cyber-physical attacks. However, the application domain and the target metrics differ substantially from the task of web application security testing.

The third direction is related to partial observability, noisy state estimation, and uncertainty of the defensive profile. In [11], Markov game modelling of MTD is proposed, which makes it possible to account for changes in the attack surface. Such a model, however, is not oriented towards learning a TTC-sensitive testing policy. In [12], a Markov game is applied to the allocation of cyber alerts, but the target task concerns the allocation of defensive resources rather than the detection of short compromise trajectories. In [13], deep reinforcement learning is used for adaptive MTD, yet the temporal characteristics of reaching a vulnerable state and the tail losses do not form the basis of the evaluation function.

The fourth group of sources substantiates the need for risk-sensitive reinforcement learning. The classical work [14] shows that optimizing only the mathematical expectation of the reward is insufficient in problems where the critical consequences are concentrated in unfavorable regions of the distribution. Recent studies on risk-sensitive reinforcement learning based on the conditional value at risk (CVaR) emphasize that tail criteria make it possible to control rare but critical scenarios that are ignored under risk-neutral optimization [15]. At the same time, the direct optimization of tail functionals in large state spaces is often computationally demanding, which gives rise to the need for intermediate mechanisms that provide risk-sensitive weighting of future values without an explicit enumeration of tail trajectories.

It follows from the analysis carried out that the existing approaches have three interrelated limitations. First, risk-neutral reinforcement learning models optimize the average outcome of an episode and insufficiently account for rare short trajectories of rapid compromise. Second, models of partially observable cyber defense are mainly focused on the defensive policy and do not provide a TTC-oriented formation of the testing policy. Third, multi-agent and Markov game approaches usually do not combine, within a single formal mechanism, an entropic-risk Q-update, a dense reward oriented towards the time to compromise, and a post-episode evaluation of tail losses in terms of CVaR. Accordingly, the topical scientific problem is the development of a risk-sensitive method for learning a web application security testing policy capable of detecting short critical compromise trajectories under partial observability, noisy estimation of defensive attributes, and stochastic changes in the defensive profile of the environment.

The method proposed in this paper continues a series of studies by the authors. In [16], a Markov decision process on a semantically constrained MAL attack graph is formalized. In [17], a temporal-semantic reward model based on the time to compromise is constructed. In [17], a stochastic model of two-agent attack–defense confrontation is proposed. The present paper develops this line of research by complementing the stochastic model [18] with risk-sensitive Q-learning oriented towards the detection of short critical compromise trajectories.

The aim of the study is to increase the efficiency and robustness of web application security testing policies through the application of risk-sensitive Q-learning, which provides a reduction of the time to compromise and a decrease of tail losses in unfavorable compromise scenarios. To achieve this aim, it is necessary to analyze the existing approaches to reinforcement learning, Markov game models, and risk-sensitive learning in web application security testing problems and to identify their limitations. On the basis of the stochastic model of two-agent attack–defense confrontation from [18], it is necessary to develop a risk-sensitive Q-learning method that combines an entropic-risk operator for updating the Q-function, a TTC-oriented dense reward, and a post-episode evaluation of tail losses in terms of CVaR. It is also necessary to implement the training of the testing agent using the proposed method under partial observability and to carry out an experimental study and a comparative analysis with risk-neutral and non-TTC-oriented approaches.

## **Materials and methods**

### **Risk-sensitive evaluation of security testing scenarios**

The structure of a web-oriented system is represented by a directed asset graph  $G = (N, E)$ , where  $N$  is the set of web components and  $E$  denotes the admissible transitions between them. This graph is an aggregated representation of the attack graph constructed by means of the MAL metalanguage in [16]: each web component  $N_k$  corresponds to a group of semantically related attack steps of the MAL graph, aggregated according to their functional

association with the component. Such aggregation preserves the semantic structure of possible compromise trajectories while reducing the dimensionality of the state space to a level acceptable for two-agent risk-sensitive learning. Since the dimensionality of the state representation is proportional to the number of nodes, such aggregation reduces it by a factor of 8–12 depending on the configuration, which makes the training of the risk-sensitive policy computationally manageable while preserving the semantic structure of compromise trajectories.

In web application security testing problems, a testing scenario is considered as a finite episode of interaction between the testing agent and the protection subsystem of a web-oriented information and communication system. Let  $S$  be the state space,  $A$  the set of admissible test actions,  $s_t \in S$  the state of the system at step  $t$ ,  $a_t \in A(s_t)$  the action of the testing agent, and  $T$  the horizon of the episode. Transitions between states are stochastic, and therefore all trajectories of an episode are considered on the probability space  $(\Omega, \mathcal{F}, Pr)$ , where  $\Omega$  is the set of possible trajectories,  $\mathcal{F}$  is the  $\sigma$ -algebra of events, and  $Pr$  defines the probability of the occurrence of the corresponding trajectories under a fixed policy of the testing agent and the reactions of the protection subsystem. The general structure of the proposed method is shown in Figure 1.

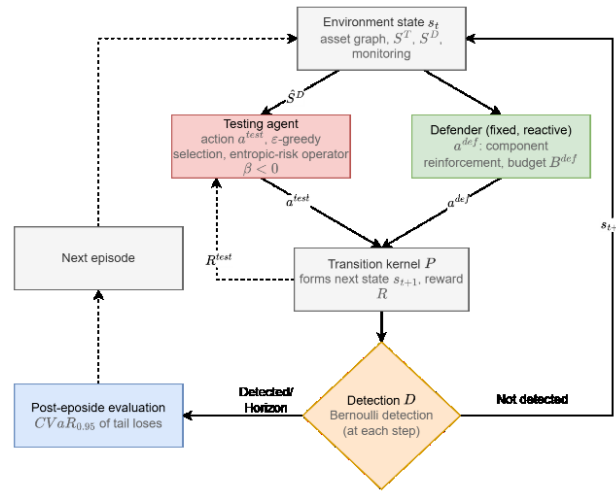


Fig. 1. Scheme of risk-sensitive Q-learning in the web application attack–defense security testing problem

The policy of the testing agent is defined as  $\pi_\theta(a | s) = Pr(a_t = a | s_t = s; \theta)$ , where  $\theta$  is the set of policy parameters. In this work the policy is not specified directly but is formed through the estimation of the Q-function and the subsequent selection of actions in accordance with the entropic-risk criterion.

The speed of reaching a vulnerable state is estimated by means of the time to compromise metric, which is interpreted as the number of steps required by the testing agent to bring the target web component  $N_{target}$  into a vulnerable state. Small TTC values correspond to short scenarios of bypassing protection mechanisms and characterize web application configurations with an increased level of practical risk. If an episode terminates with detection or does not lead to a vulnerable state before the horizon  $T$ , the TTC value is recorded as censored at the level  $T$ .

Let  $l_t$  be the instantaneous loss of the protection subsystem at time  $t$ , and let  $\gamma \in (0,1)$  be the discount factor. The risk-neutral estimate of the cumulative losses is defined as

$$J = \mathbb{E}_{pr} \left[ \sum_{t=0}^T \gamma^t l_t \right] \quad (1)$$

The estimation according to (1) averages the losses over episodes and does not distinguish between slow non-critical and short rapid compromise scenarios, whereas it is precisely the latter, with a small TTC, that are the most dangerous for security testing.

To formalize the tail region, the random cumulative loss of an episode  $L(\omega) = \sum_{t=0}^T \gamma^t l_t(\omega)$ ,  $\omega \in \Omega$ , is introduced. Here  $L$  is a random variable whose value depends on the particular trajectory  $\omega$ . The event  $L \leq z$  means that the cumulative discounted loss in an episode does not exceed the threshold value  $z$ .

The threshold level of losses with the confidence parameter  $\alpha \in (0,1)$  is defined by the functional  $Var_\alpha(L) = \inf\{z \in \mathbb{R}: Pr(L \leq z) \geq \alpha\}$ . The functional  $Var_\alpha$  determines the boundary of the tail region of the loss distribution but does not characterize the average severity of the losses once this threshold is exceeded. Therefore, the conditional tail risk is used to evaluate critical testing scenarios

$$CVaR_\alpha(L) = \min_{\eta \in \mathbb{R}} \left( \eta + \frac{1}{1-\alpha} \mathbb{E}_{pr}[(L - \eta)^+] \right) \quad (2)$$

where  $(x)^+ = \max(x, 0)$ , and  $\eta$  is an auxiliary threshold parameter.

In this work  $CVaR_\alpha$  is used as a post-episode metric of tail losses. Since the direct optimization of  $CVaR_\alpha$  in a high-dimensional state space is computationally demanding, the policy is constructed by entropic-risk Q-learning, which approximates risk-sensitive behavior through the exponential weighting of unfavorable trajectories.

To account for the local temporal dynamics, the potential function  $\Phi: S \rightarrow \mathbb{R}_{\geq 0}$  is introduced, where  $\Phi(s)$

specifies the estimate of the expected time to reach a vulnerable state from state  $s$  [17], together with its local change  $\Delta\Phi_t = \Phi(s_t) - \Phi(s_{t+1})$ . If  $\Delta\Phi_t > 0$ , the current transition reduces the expected time to compromise. If  $\Delta\Phi_t < 0$ , the protection subsystem increases the temporal robustness of the web application. The potential provides an intermediate TTC-oriented signal for learning in episodes with a delayed terminal reward.

### Entropic-risk Q-learning of the testing agent

Risk-sensitive reinforcement learning is used to construct an adaptive testing policy. Let  $Q_\theta(s, a)$  be the parameterized Q-function of the testing agent with parameters  $\theta$ , and let  $Q_{\theta^-}(s, a)$  be the target Q-function whose parameters are updated with a delay in order to stabilize the learning process.

The entropic-risk operator is used to evaluate the future value

$$\mathcal{T}_\beta(Q)(s') = \frac{1}{\beta} \log \sum_{a' \in A^{test}(s')} \exp(\beta Q(s', a')) \quad (3)$$

where  $A^{test}(s')$  is the set of admissible actions in state  $s'$ , and  $\beta \in \mathbb{R}$  is the risk-sensitivity parameter. For  $\beta > 0$  the operator produces risk-seeking behavior and amplifies the influence of actions with a high expected value. For  $\beta < 0$  the operator amplifies the influence of unfavorable alternatives and produces risk-averse behavior. In the security testing problem, the mode  $\beta < 0$  is used, since it orients the policy of the testing agent towards the search for critical trajectories with a minimal expected time to compromise and a high risk for the protection subsystem.

The target of the Q-function update is defined as  $y_t = r_t^{test} + \gamma(1 - d_t)\mathcal{T}_\beta(Q_{\theta^-})(s_{t+1})$ , where  $r_t$  is the instantaneous reward of the testing agent,  $d_t \in \{0, 1\}$  is the indicator of episode termination, and  $\gamma \in (0, 1)$  is the discount factor. The parameters of the Q-function are updated by minimizing the quadratic error  $L(\theta) = \mathbb{E}_{pr}[(y_t - Q_\theta(s_t, a_t))^2]$ . The gradient update step has the form  $\theta \leftarrow \theta - \alpha \nabla_\theta L(\theta)$ , where  $\alpha > 0$  is the learning rate.

A replay buffer is used to stabilize learning, and mini-batches of transitions are randomly sampled from it during the update of the parameters. The target Q-function  $Q_{\theta^-}$  is synchronized with the main network at a fixed interval of steps, which reduces the correlation between successive updates and increases the stability of convergence.

Actions are selected according to an  $\varepsilon$ -greedy strategy: with probability  $\varepsilon$  a random action is chosen, otherwise the action with the maximal value of  $Q^{test}$ . At the initial stages of learning larger values of  $\varepsilon$  are used, after which the parameter is gradually decreased in order to move from exploration to the exploitation of the learned policy.

The instantaneous reward of the testing agent is defined as

$$r_t^{test} = \lambda_c \text{comp}(s_{t+1}) - \lambda_d \text{det}(s_{t+1}) \quad (4)$$

where  $\text{comp}(s_{t+1}) \in \{0, 1\}$  is the indicator of reaching a vulnerable state of the target component,  $\text{det}(s_{t+1}) \in \{0, 1\}$  is the indicator of detection of the testing scenario,  $\lambda_c > 0$  specifies the weight of a successful compromise, and  $\lambda_d > 0$  specifies the penalty for detection. (4) defines a sparse reward oriented towards the final outcome of an episode. To stabilize learning in environments with long trajectories, a TTC-oriented mechanism of dense reward formation is additionally used.

The defending agent implements a fixed reactive policy that is not optimized during learning. The set of defensive actions is  $A^{def}(s) = \{0, 1, \dots, |N|\}$ , where the action  $k \geq 1$  denotes reinforcement of the protection of the web component  $N_k$ , and 0 denotes the absence of a response. The compromise progress of a node is estimated by the accumulated value of its test attributes  $g_k = \sum_{j=1}^m S_{k,j}^T$ . At each step the defender applies a deterministic rule: if  $\max_k g_k > \theta^{def}$  and the response budget  $B^{def}$  is not exhausted, it reinforces the most attacked component,  $a_t^{def} = \text{argmax}_k g_k$ . Otherwise, no response is performed,  $a_t^{def} = 0$ . Reinforcement increases the defensive attribute of the selected component by a fixed amount,  $S_{k*,j}^D(t+1) = \min(S_{k*,j}^D(t) + \delta, w)$ . The budget  $B^{def}$  limits the total number of responses per episode and specifies a fixed level of defensive activity of the environment. The parameters of the rule are the same for all configurations. Such a formulation reduces the two-agent stochastic game [18] to learning the policy of the testing agent against a fixed opponent, which makes it possible to isolate the influence of risk-sensitive evaluation on the quality of detection of critical trajectories. The mutually optimal formation of the policies of both sides is considered as a direction for further research.

For the protection subsystem, the corresponding instantaneous loss is defined as

$$l_t^{def} = \lambda_c \text{comp}(s_{t+1}) + \lambda_r C(a_t^{def}) \quad (5)$$

where  $C(a_t^{def}) = c \cdot 1[a_t^{def} \neq 0]$  is the constant cost of a single response, and  $\lambda_r$  is the weight of the resource expenditure. The first term penalizes the defender for reaching a vulnerable state, and the second one for the resource expenditure on the response. The quantity  $l_t^{def}$  is used to interpret the load on the protection subsystem and is not an optimization objective.

Two instances of the environment, Env1 and Env2, are used to verify the robustness of the risk-sensitive policy. Env1 corresponds to a controlled mode with minimal uncertainty and is used to verify the basic robustness of the learning process and the ability of the testing agent to learn the structure of the scenario. Env2 specifies a stochastic environment with heterogeneous defensive attributes, random initialization, and partial observability.

Let  $N$  be the set of web components of the graph and let  $N_{start} \subset N$  be the set of admissible starting nodes. In the stochastic mode the initial component of the testing agent is selected at random,  $N_0 \sim \text{Unif}(N_{start})$ ,  $N_{target} \notin$

$\mathcal{N}_{start}$ , where  $N_{target}$  is the target web component for which the reaching of a vulnerable state is evaluated.

Let  $m$  be the number of types of test actions and let  $w$  be the maximal level of the attributes of test influence and protection. For each node  $N_k \in N$ , test attributes  $S_{k,j}^T$  and defensive attributes  $S_{k,j}^D$  are introduced, where  $j = 1, \dots, m$ . The initial test attributes are set to zero values,  $S_{k,j}^T(0) = 0$ ,  $N_k \in N$ ,  $j = 1, \dots, m$ .

In the controlled environment Env1 the defensive attributes are initialized with a fixed baseline level  $S_{k,j}^D(0) = d_0$ ,  $d_0 \in \{0, \dots, w\}$ , whereas in the stochastic environment Env2 they follow a discrete uniform distribution  $S_{k,j}^D(0) \sim \text{Unif}\{d_{min}, \dots, d_{max}\}$ ,  $0 \leq d_{min} \leq d_{max} \leq w$ .

The monitoring parameter  $S_{k,m+1}^D$  is introduced separately, and it determines the node detection capability with respect to testing activity. The initial monitoring level is set as  $S_{k,m+1}^D(0) = q_1$  for Env1 and as  $S_{k,m+1}^D(0) = q_2$  for Env2, where  $q_1, q_2 \in \{0, \dots, w\}$ , and  $q_1 > q_2$ . This means that the controlled environment has a higher initial level of detection, whereas the stochastic environment leaves the testing agent a wider space for covert advancement.

Partial observability is modelled through the noisy estimation of the defensive attributes  $\hat{S}_{k,j}^D = S_{k,j}^D + \varepsilon_{k,j}$ ,  $\varepsilon_{k,j} \sim \mathcal{N}(0, \sigma^2)$ , where  $\hat{S}_{k,j}^D$  is the estimated defensive attribute is available to the testing agent, and  $\sigma^2$  is the variance of the observation noise.

To reduce the dependence of learning on rare terminal events, a TTC-oriented dense reward is used

$$r_t^{test,dense} = -\Delta\Phi_t + \frac{1}{|N|} \sum_{k \in N} \mathbf{1}_{N_k \in N_{vul}(s_t)} \quad (6)$$

where  $\Delta\Phi_t$  determines the weight of the temporal change of the potential,  $|N|$  is the number of web components of the graph,  $\mathbf{1}$  is the indicator function, and  $N_{vul}(s_t) \subseteq N$  is the set of web components that have reached a vulnerable state at time  $t$ .

The dense reward given by (6) is used in addition to the sparse one in (4), without replacing it, as an intermediate mechanism of reward formation during learning. The sparse reward evaluates the final efficiency of a testing scenario, whereas the dense TTC-oriented reward forms an intermediate signal and accelerates policy convergence in environments with long episodes and a delayed terminal event.

The generalized sequence of actions shown in Figure 1 is formalized as the following stages of the proposed method:

- 1) initialization of the environment – formation of the graph of web components, setting of the initial values of the test and defensive attributes according to the selected mode Env1 or Env2, and selection of the target web component;
- 2) observation of the state – obtaining by the testing agent of a noisy estimate of the defensive attributes under partial observability;
- 3) selection of a test action – determination of the action according to the  $\varepsilon$ -greedy strategy based on the current values of the Q-function.
- 4) response of the protection subsystem – application of the fixed reactive policy, which reinforces the most attacked web component within the response budget;
- 5) transition and reward formation – computation of the next state, of the sparse reward in (4), and of the dense TTC-oriented component in (6);
- 6) updating of the Q-function – re-evaluation of the future value by the entropic-risk operator in (3) with the risk-sensitivity parameter  $\beta$  and minimization of the quadratic error on mini-batches from the replay buffer;
- 7) verification of the episode termination condition – return to stage 2 if the testing scenario has not been detected and the horizon has not been reached;
- 8) post-episode evaluation – computation of the tail losses in terms of the  $CVaR_{0.95}$  functional and transition to the next episode.

### Datasets and simulation protocol

The experimental study is oriented towards evaluating the ability of entropic-risk Q-learning to form a policy aimed at detecting short critical trajectories. The main criteria are the time to compromise TTC, the conditional tail risk  $CVaR_{0.95}$ , the number of episodes required for policy stabilization, the robustness to observation noise, and the influence of the TTC-oriented dense reward on the learning process.

The ToN-IoT [19] and CIC-IDS2017 [20] datasets were used for the statistical parameterization of the simulation environment. Within this paper the datasets were used not as test samples for attack classification, but as a source of statistical distributions for constructing environments with different levels of information uncertainty. The ToN-IoT dataset contains network flows, HTTP events, telemetry of IoT devices, and system logs, which makes it possible to form an environment with a high heterogeneity of web-oriented interactions. The CIC-IDS2017 dataset contains multi-step attack scenarios, characteristics of application sessions, and flow features of network activity, which makes it possible to model an environment with pronounced temporal dependencies between compromise events.

During data pre-processing, service identifiers, duplicates, and incomplete records were removed. Numerical features were normalized to the interval  $[0,1]$ , and categorical parameters were transformed into discrete types of test

actions. Events were aggregated according to temporal adjacency and functional association with web components. For the ToN-IoT environment, the MAL graph contains 1314 attack steps, which are aggregated into 115 web components, whereas for CIC-IDS2017 the corresponding values are 512 and 58, which corresponds to a reduction of the dimensionality of the state representation by factors of 11.4 and 8.8, respectively. The intensity of suspicious HTTP requests, the frequency of failed authentications, and the atypical duration of sessions were used for the parameterization of the test attributes. The frequency of blockings, the intensity of local filtering, and the characteristics of detection events were used for the parameterization of the defensive attributes and the observation noise.

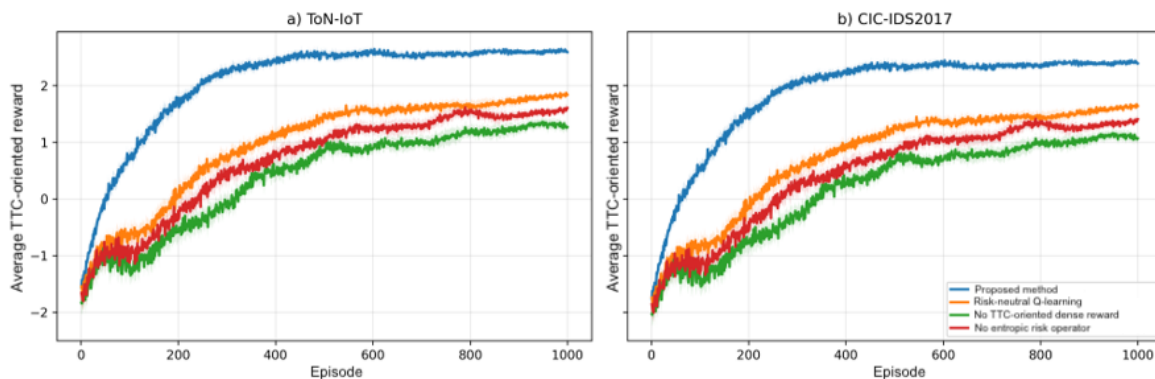
Two modes of the simulation environment were used to verify the robustness of the policy. Env1 corresponded to a controlled mode with homogeneous defensive attributes and minimal observation uncertainty, and Env2 to a stochastic mode with heterogeneous defensive attributes, random initialization of the initial states, and noisy estimation of the defensive parameters.

The same simulation protocol was used for both datasets. The horizon of an episode was  $T = 80$ , the discount factor was  $\gamma = 0.97$ , the number of types of test actions was set to  $m = 4$ , and the maximal level of the attributes was  $w = 9$ . The number of web components in the graph after aggregation was  $|N| = 115$  for ToN-IoT and  $|N| = 58$  for CIC-IDS2017. The baseline value of the risk-sensitivity parameter was set to  $\beta = -0.6$ . For each configuration, 30 independent runs of 1000 episodes each were performed. The size of the replay buffer was  $5 \cdot 10^5$  transitions, and the target Q-function was synchronized every 2500 steps. All results are presented as the mean value with a 95% confidence interval.

Three competing approaches were used for comparison. Risk-neutral Q-learning used the standard update without the entropic-risk operator and optimized the average expected reward. The method without the TTC-oriented dense reward used only the sparse terminal signal, without forming an intermediate TTC-oriented reinforcement. The method without the entropic-risk operator used the TTC-oriented reward but without risk-sensitive weighting of the future values of the Q-function. The proposed method combined the entropic-risk operator, the TTC-oriented dense reward, partial observability, and the post-episode evaluation of tail losses in terms of  $CVaR_{0.95}$ .

### Results

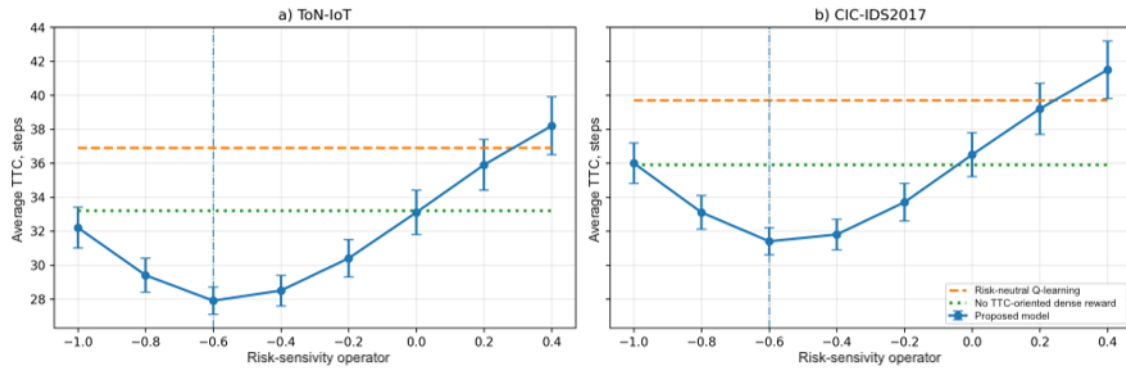
The first experiment was aimed at evaluating the stability of convergence of the testing policy. For this purpose, the dynamics of the TTC-oriented reward during learning was analyzed. Figure 2 presents the averaged results of 30 independent runs for the ToN-IoT and CIC-IDS2017 datasets.



**Fig. 2. Dynamics of the TTC-oriented reward during learning**

As can be seen in Figure 2, the proposed method provides the fastest stabilization of the policy and the lowest variability of the reward among all the compared approaches. For CIC-IDS2017, stabilization of the learning process is achieved after approximately  $440 \pm 21$  episodes, whereas risk-neutral Q-learning requires about  $675 \pm 31$  episodes. For ToN-IoT the corresponding values are given in Table 1. The greatest instability is observed for the method without the TTC-oriented reward, in which, owing to the absence of an intermediate signal, prolonged phases of local instability are formed. At the same time, the method without the entropic-risk operator demonstrates a higher sensitivity to reward fluctuations at the later stages of learning, which indicates a poorer ability to stabilize the policy in environments with partial observability.

The second experiment was devoted to the analysis of the influence of the risk-sensitivity parameter  $\beta$  on the temporal characteristics of compromise. The value of  $\beta$  was varied in the range from -1.0 to 0.4, and the average time to compromise was used as an integral indicator. Since the parameter  $\beta$  is used only in the entropic-risk operator, the dependence on  $\beta$  was analyzed only for the proposed method, whereas risk-neutral Q-learning and the method without the TTC-oriented reward are presented as reference baseline levels. The results are shown in Figure 3.



**Fig. 3. Influence of the risk-sensitivity parameter  $\beta$  on the time to compromise**

As can be seen in Figure 3a–3b, the lowest TTC values are achieved for negative values of  $\beta$ , which correspond to the risk-averse mode of evaluating the future value. For ToN-IoT the minimal average TTC is  $27.9 \pm 1.1$  steps at  $\beta = -0.6$ , whereas for risk-neutral Q-learning the corresponding level remains unchanged regardless of  $\beta$ . For CIC-IDS2017 the minimal TTC is observed in the interval  $\beta \in [-0.7; -0.5]$  and amounts to approximately  $31.4 \pm 1.3$  steps. For positive values of  $\beta$  the policy becomes risk-seeking, as a result of which the share of long inefficient trajectories increases and the average time to compromise grows. This confirms that the parameter  $\beta$  directly controls the risk sensitivity of the testing policy and determines the balance between an aggressive search for short trajectories and the stability of the learning process.

A generalized comparison of the methods is given in Table 1 separately for the environments Env1 and Env2. The  $CVaR_{0.95}$  indicator is given for Env2 as the more complex mode, in which the tail losses are the most informative. The share of successful scenarios is presented in the analysis of robustness to noise in Figure 4.

Table 1.

**Comparison of risk-sensitive Q-learning with the competing approaches in the Env1 and Env2 environments**

| Method                         | TTCEnv1,<br>ToN-IoT | TTCEnv2,<br>ToN-IoT | TTCEnv1,<br>CIC-IDS2017 | TTCEnv2,<br>CIC-IDS2017 | $CVaR_{0.95}$ of<br>normalized tail<br>losses for Env2 | Episodes to<br>convergence |
|--------------------------------|---------------------|---------------------|-------------------------|-------------------------|--------------------------------------------------------|----------------------------|
| Risk-neutral Q-learning        | $34.1 \pm 1.4$      | $36.9 \pm 1.5$      | $37.0 \pm 1.6$          | $39.7 \pm 1.7$          | $0.84 \pm 0.04$                                        | $650 \pm 28$               |
| Without TTC-oriented reward    | $31.0 \pm 1.3$      | $33.8 \pm 1.4$      | $33.6 \pm 1.4$          | $36.2 \pm 1.5$          | $0.76 \pm 0.03$                                        | $590 \pm 24$               |
| Without entropic-risk operator | $28.7 \pm 1.2$      | $31.4 \pm 1.3$      | $31.5 \pm 1.3$          | $34.6 \pm 1.4$          | $0.69 \pm 0.03$                                        | $510 \pm 21$               |
| Proposed method                | $25.8 \pm 1.0$      | $27.9 \pm 1.1$      | $29.1 \pm 1.2$          | $31.4 \pm 1.3$          | $0.58 \pm 0.02$                                        | $420 \pm 19$               |

The data in Table 1 indicate that the transition from Env1 to Env2 leads to an increase of the average time to compromise for all the methods, which is explained by the influence of information uncertainty and of the heterogeneous defensive profile. However, the proposed method demonstrates the smallest degradation of efficiency during the transition to the stochastic mode. For ToN-IoT the increase of TTC between Env1 and Env2 amounts to approximately 2.1 steps, whereas for risk-neutral Q-learning it is about 2.8 steps. For CIC-IDS2017 the corresponding difference is 2.3 and 2.7 steps. The  $CVaR_{0.95}$  indicator of normalized tail losses for Env2 decreases by approximately 31 % in comparison with the risk-neutral approach, and the number of episodes required for convergence is reduced by approximately 35.4 %.

The last experiment was aimed at evaluating the robustness of the methods to partial observability. For this purpose, the variance of the observation noise  $\sigma^2$  was gradually increased, and the share of successful compromise scenarios was used as an integral indicator.

As follows from the results in Figure 4, as  $\sigma^2$  increases, the efficiency of all the methods gradually decreases. The proposed method, however, degrades more slowly than its analogues. For ToN-IoT, even at high noise values, the share of successful scenarios remains close to 0.56, whereas for risk-neutral Q-learning it decreases to approximately 0.39. A similar tendency is observed for CIC-IDS2017 as well. The sharpest drop of efficiency is observed after  $\sigma^2 \approx 1.5$ , when local errors in the estimation of the defensive attributes begin to affect the quality of the policy substantially. At the same time, the entropic-risk operator makes it possible to account for unfavorable trajectories better, owing to which the proposed method demonstrates a higher robustness to information uncertainty.

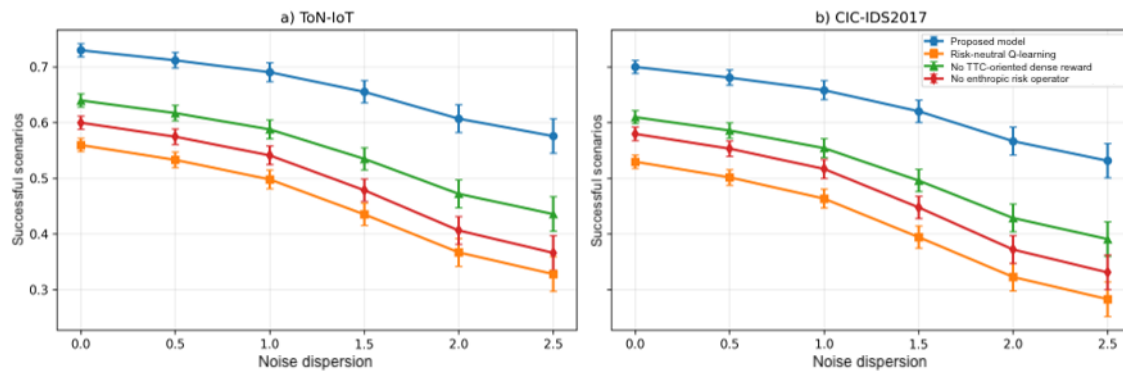


Fig. 4. Influence of the observation noise on the share of successful scenarios

### Discussion

The practical significance of the obtained results in the context of cybersecurity lies in the possibility of applying risk-sensitive Q-learning to build adaptive tools for web application security testing that are capable of detecting short critical compromise trajectories under partial observability, noisy estimation of defensive attributes, and stochastic changes in the defensive profile of the environment. The proposed method can be used for automated penetration testing, for evaluating the efficiency of web application protection mechanisms, for analyzing the robustness of attack detection systems, for modelling the adaptive behavior of an attacker, and for predicting critical compromise scenarios in web-oriented cybersecurity systems, cloud services, and distributed web infrastructures. The use of TTC-oriented risk-sensitive policies makes it possible to reduce the time of detection of potentially critical states of a system and to increase the efficiency of the automated analysis of web application security in dynamic environments with incomplete information.

The limitations of the study are related to the use of a simulation environment and of an aggregated representation of web components, which does not take into account the full structural complexity of real web infrastructures, the dynamic dependencies between multi-level services, and the adaptive change of the behavior of protection mechanisms in real time.

Further research should be directed towards the integration of risk-sensitive learning with multi-level graph models of web infrastructures, the extension of the action space of the testing agents, and the use of deep architectures with memory mechanisms for handling long dependencies.

### Conclusions

A risk-sensitive method for forming a web application security testing policy based on entropic-risk Q-learning, a TTC-oriented dense reward, and a post-episode evaluation of tail losses in terms of  $CVaR_{0.95}$  is proposed in the paper. Unlike risk-neutral approaches, the proposed method makes it possible to account for unfavorable compromise trajectories and to form a policy oriented towards the detection of short critical scenarios. The proposed method develops risk-sensitive mechanisms of policy learning within a two-agent attack–defense environment and provides an adaptive account of the temporal characteristics of compromise during web application security testing.

The experimental study on the ToN-IoT and CIC-IDS2017 datasets showed that the proposed method provides the lowest average time to compromise among all the compared approaches: in the stochastic environment Env2 the TTC value decreased from  $36.9 \pm 1.5$  to  $27.9 \pm 1.1$  steps for ToN-IoT and from  $39.7 \pm 1.7$  to  $31.4 \pm 1.3$  steps for CIC-IDS2017. The  $CVaR_{0.95}$  indicator of normalized tail losses for Env2 decreased by approximately 31 %, and the number of episodes required for policy convergence by approximately 35.4 %. The analysis of the influence of the risk-sensitivity parameter showed that the most effective mode is achieved at  $\beta \approx -0.6$ , when the policy acquires a risk-averse character and detects short critical compromise trajectories more efficiently. The study of robustness to observation noise demonstrated a higher robustness of the proposed method to information uncertainty in comparison with risk-neutral Q-learning.

### ADDITIONAL INFORMATION

#### AUTHOR CONTRIBUTIONS

Conceptualization A.P., L.K.; methodology A.P.; software A.P.; validation A.P.; formal analysis A.P.; investigation A.P.; writing – original draft preparation A.P.; writing – review and editing L.K.; supervision L.K.

#### DECLARATION ON THE USE OF GENERATIVE ARTIFICIAL INTELLIGENCE TOOLS

In preparing this work, the authors used Claude (Anthropic) for translating the manuscript from Ukrainian into English. After using this tool, the authors reviewed and edited the content and take full responsibility for the content of this publication.

REFERENCES

1. Wang W. Research and challenges of reinforcement learning in cyber defense decision-making for intranet security / W. Wang, D. Sun, F. Jiang, X. Chen, C. Zhu // *Algorithms*. – 2022. – Vol. 15, № 4. – P. 134. DOI: <https://doi.org/10.3390/a15040134>
2. Isaiev T. Analysis of information technologies and methods for automatic updating of threat detection models in computer systems / T. Isaiev, O. Atamaniuk // *Computer Systems and Information Technologies*. – 2026. – № 1. – P. 172–185. DOI: <https://doi.org/10.31891/csit-2026-1-16>
3. Nguyen T. T. Deep reinforcement learning for cyber security / T. T. Nguyen, V. J. Reddi // *IEEE Transactions on Neural Networks and Learning Systems*. – 2023. – Vol. 34, № 8. – P. 3779–3795. DOI: <https://doi.org/10.1109/tnnls.2021.3121870>
4. Yousefi M. A reinforcement learning approach for attack graph analysis / M. Yousefi, N. Mtetwa, Y. Zhang, H. Tianfield // 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications. – New York, 2018. – P. 212–217. DOI: <https://doi.org/10.1109/trustcom/bigdata.2018.00041>
5. Oh S. H. Employing deep reinforcement learning to cyber-attack simulation for enhancing cybersecurity / S. H. Oh, J. Kim, J. H. Nah, J. Park // *Electronics*. – 2024. – Vol. 13, № 3. – P. 555. DOI: <https://doi.org/10.3390/electronics13030555>
6. Terranova F. Leveraging deep reinforcement learning for cyber-attack paths prediction: formulation, generalization, and evaluation / F. Terranova, A. Lahmadi, I. Chrisment // *The 27th International Symposium on Research in Attacks, Intrusions and Defenses*. – Padua, 2024. – P. 1–16. DOI: <https://doi.org/10.1145/3678890.3678902>
7. Hammar K. Finding effective security strategies through reinforcement learning and self-play / K. Hammar, R. Stadler // 2020 16th International Conference on Network and Service Management (CNSM). – Izmir, 2020. – P. 1–9. DOI: <https://doi.org/10.23919/cnsm50824.2020.9269092>
8. Yao Q. Adversarial decision-making for moving target defense: a multi-agent Markov game and reinforcement learning approach / Q. Yao, Y. Wang, X. Xiong, P. Wang, Y. Li // *Entropy*. – 2023. – Vol. 25, № 4. – P. 605. DOI: <https://doi.org/10.3390/e25040605>
9. Game Theory and Machine Learning for Cyber Security / eds.: C. A. Kamhoua, C. D. Kiekintveld, F. Fang, Q. Zhu. – Hoboken : Wiley-IEEE Press, 2021. – 546 p. DOI: <https://doi.org/10.1002/9781119723950>
10. Bitirgen K. Markov game based on reinforcement learning solution against cyber-physical attacks in smart grid / K. Bitirgen, Ü. B. Filik // *Expert Systems with Applications*. – 2024. – Vol. 255. – P. 124607. DOI: <https://doi.org/10.1016/j.eswa.2024.124607>
11. Sengupta S. General sum Markov games for strategic detection of advanced persistent threats using moving target defense in cloud networks / S. Sengupta, A. Chowdhary, D. Huang, S. Kambhampati // *Decision and Game Theory for Security*. – Cham : Springer, 2019. – P. 492–512. DOI: [https://doi.org/10.1007/978-3-030-32430-8\\_29](https://doi.org/10.1007/978-3-030-32430-8_29)
12. Dunstatter N. Solving cyber alert allocation Markov games with deep reinforcement learning / N. Dunstatter, A. Tahsini, M. Guirguis, J. Tešić // *Decision and Game Theory for Security*. – Cham : Springer, 2019. – P. 164–183. DOI: [https://doi.org/10.1007/978-3-030-32430-8\\_11](https://doi.org/10.1007/978-3-030-32430-8_11)
13. Eghesad T. Adversarial deep reinforcement learning based adaptive moving target defense / T. Eghesad, Y. Vorobeychik, A. Laszka // *Decision and Game Theory for Security*. – Cham : Springer, 2020. – P. 58–79. DOI: [https://doi.org/10.1007/978-3-030-64793-3\\_4](https://doi.org/10.1007/978-3-030-64793-3_4)
14. Mihatsch O. Risk-sensitive reinforcement learning / O. Mihatsch, R. Neuneier // *Machine Learning*. – 2002. – Vol. 49, № 2–3. – P. 267–290. DOI: <https://doi.org/10.1023/a:1017940631555>
15. Ni X. Risk-sensitive reward-free reinforcement learning with CVaR / X. Ni, G. Liu, L. Lai // *Proceedings of the 41st International Conference on Machine Learning*. – Vienna, 2024. – P. 37999–38017.
16. Prytula A. V. Modelling of cyberattack scenarios as a Markov process with a semantically constrained action space / A. V. Prytula, L. M. Kupershtein // *Cybersecurity: Education, Science, Technique*. – 2026. – № 33. – P. 555–569. DOI: <https://doi.org/10.28925/2663-4023.2026.33.1232> (in Ukrainian)
17. Prytula A. V. Reward model in reinforcement learning for the web application security testing problem [Preprint] / A. V. Prytula, L. M. Kupershtein. – ResearchGate, 2026. DOI: <https://doi.org/10.13140/RG.2.2.35201.62566> (in Ukrainian)
18. Prytula A. V. Stochastic attack–defense confrontation model for web application security testing based on reinforcement learning / A. V. Prytula, L. M. Kupershtein. – ResearchGate, 2026. DOI: <https://doi.org/10.13140/RG.2.2.17354.32966> (in Ukrainian)
19. Alsaedi A. TON IoT telemetry dataset: a new generation dataset of IoT and IIoT for data-driven intrusion detection systems / A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, A. Anwar // *IEEE Access*. – 2020. – Vol. 8. – P. 165130–165150. DOI: <https://doi.org/10.1109/access.2020.3022862>
20. Sharafaldin I. Toward generating a new intrusion detection dataset and intrusion traffic characterization / I. Sharafaldin, A. Habibi Lashkari, A. A. Ghorbani // *Proceedings of the 4th International*

Conference on Information Systems Security and  
Privacy. – Funchal, 2018. – P. 108–116. DOI:  
<https://doi.org/10.5220/0006639801080116>

Андрій ПРИТУЛА, Леонід КУПЕРШТЕЙН  
Вінницький національний технічний університет

## РИЗИК-ЧУТЛИВЕ Q-НАВЧАННЯ В ДВОАГЕНТНІЙ МАРКОВСЬКІЙ МОДЕЛІ ПРОТИБОРСТВА «АТАКА–ЗАХИСТ»

Запропоновано ризик-чутливий метод формування політики тестування безпеки вебзастосунків на основі ентропійно-ризикового Q-навчання в двоагентній марковській моделі протиборства «атака–захист». Дослідження зумовлене необхідністю виявляти короткі критичні траєкторії компрометації в умовах часткової спостережуваності, шумового оцінювання оборонних атрибутів і стохастичної зміни захисного профілю середовища. На відміну від ризик-нейтральних підходів, запропонований метод використовує щільну винагороду, орієнтовану на час до компрометації ТТС, та післяепізодне оцінювання хвостових втрат за умовним хвостовим ризиком на рівні довіри 0.95 (CVaR0.95), де параметр ризик-чутливості  $\beta$  ентропійно-ризикового оператора визначає ступінь ризико-аверсивності політики. Введено два режими модельно-симуляційного середовища – контрольований E<sub>env1</sub> і стохастичний E<sub>env2</sub>, які використовуються для аналізу стійкості політики за різного рівня інформаційної невизначеності. Для статистичної параметризації середовищ використано набори даних ToN-IoT і CIC-IDS2017. Агрегація семантично пов'язаних кроків атаки MAL-графа до рівня вебкомпонентів зменшує розмірність вектора станів у 8–12 разів, що забезпечує обчислювальну керованість навчання ризик-чутливої політики. Отримані результати показують, що запропонований метод забезпечує найменший середній час до компрометації серед усіх порівнюваних підходів. У середовищі E<sub>env2</sub> значення ТТС зменшилося з  $36.9 \pm 1.5$  до  $27.9 \pm 1.1$  кроку для ToN-IoT і з  $39.7 \pm 1.7$  до  $31.4 \pm 1.3$  кроку для CIC-IDS2017, показник CVaR0.95 нормованих хвостових втрат знизився приблизно на 31 %, а кількість епізодів до збіжності політики скоротилася приблизно на 35.4%. Найефективніший режим досягається за  $\beta \approx -0.6$ . Практичне значення полягає в можливості застосування методу для автоматизованого тестування на проникнення та оцінювання стійкості механізмів захисту вебзастосунків.

Ключові слова: навчання з підкріпленням; машинне навчання; марківська модель; ризик-чутливе Q-навчання; час до компрометації; тестування безпеки; кібербезпека; кібератака; кіберзахист; штучний інтелект