DOI: 10.18137/RNU.V9187.25.04.P.106 УДК 004.942

Козеев Борис Николаевич
аспирант, МИРЭА – Российский технологический университет; главный специалист АО «Альфа-Банк», Москва. ORCID: 0009-0009-0993-8082
Электронный адрес: kozeev.boris2018@yandex.ru
Boris N. Kozeev
Postgraduate, MIREA – Russian Technological University; Chief specialist of JSC "Alpha-Bank", Moscow. ORCID: 0009-0009-0993-8082
E-mail address: kozeev.boris2018@yandex.ru

Беликов Владимир Вячеславович
кандидат военных наук, доцент кафедры, МИРЭА – Российский технологический университет, Москва. ORCID: 0000-0003-1423-1072
Электронный адрес: belikov_v@mirea.ru
Vladimir V. Belykov
Ph.D. of Military Sciences, Associate Professor, MIREA – Russian Technological University, Moscow. ORCID: 0000-0003-1423-1072
E-mail address: belikov_v@mirea.ru

CYBORG: УЛУЧШЕНИЕ СРЕДЫ ДЛЯ ЭФФЕКТИВНОГО ОБУЧЕНИЯ АГЕНТОВ КИБЕРБЕЗОПАСНОСТИ

Аннотация. В представленной научной статье исследуется процесс разработки и апробирования сценария для тренировочных окружений в области автоматизированной защиты информации (англ. Automated Cyber Defence, ACD) с использованием обучения с подкреплением (англ. Reinforcement Learning, RL). Основное внимание уделено применению алгоритма Proximal Policy Optimization (PPO) для обучения агента в среде CybORG, анализу эффективности предложенного подхода, выполнению исправлений критических недостатков в функциональности тренировочного окружения CybORG. Представлены результаты тестирования сценария, выявлены его слабые стороны и предложены доработки, направленные на оптимизацию процесса обучения. Продемонстрированы результаты внесенных изменений, значительно сказавшихся на эффективности работы с тренировочным окружением CybORG. Внесены изменения в тренировочное окружение CybORG, что позволило улучшить производительность и удобство использования. Проведенный анализ демонстрирует, что предложенные модификации способствуют более эффективному обучению агентов и упрощают интеграцию новых сценариев. На основе полученных результатов сформулированы рекомендации по дальнейшему совершенствованию тренировочных окружений автономных кибероперацій (англ. Autonomous Cyber Operation Gyms, ACOG).

Ключевые слова: обучение с подкреплением, тренировочное окружение, алгоритм PPO, CybORG, информационная безопасность.

Для цитирования: *Козеев Б.Н., Беликов В.В.* CybORG: улучшение среды для эффективного обучения агентов кибербезопасности // Вестник Российского нового университета. Серия: Сложные системы: модели, анализ, управление. 2025. № 4. С. 106 – 118. DOI: 10.18137/RNU.V9187.25.04.P.106

CYBORG: IMPROVING AN ENVIRONMENT FOR EFFECTIVE TRAINING OF CYBERSECURITY AGENTS

Abstract. The paper examines the development and testing of a training scenario for automated cyber defense (ACD) using reinforcement learning (RL). The authors focus on the application of the Proximal Policy Optimization (PPO) algorithm for agent training in the CybORG environment, analyzing the effectiveness of the proposed approach, and implementing fixes for critical flaws in the CybORG training environment. The paper presents the results of testing the scenario, identifies its weaknesses, and proposes improvements aimed at optimizing the training process. The paper demonstrates the results of the changes made which had a significant impact on the effectiveness of the CybORG training environment. The changes made to the CybORG training environment improved its performance and usability. The analysis demonstrates that the proposed modifications facilitate more efficient agent training and simplify the integration of new scenarios. Based on the results obtained, recommendations for further improvement of autonomous cyber operations training environments (ACOGyms) are formulated.

Keywords: reinforcement learning, training environment, PPO algorythm, CybORG, information security.

For citation: Kozeev B.N., Belikov V.V. (2025) CybORG: Improving an environment for effective training of cybersecurity agents. *Vestnik of Russian New University. Series: Complex Systems: Models, analysis, management*. No. 4. Pp. 106– 118. DOI: 10.18137/RNU.V9187.25.04.P.106 (In Russian).

Введение

Современные кибератаки становятся все более сложными и многоступенчатыми, включая в себя различные цели и пути атаки, что требует разработки продвинутых методов защиты [1]. Одним из перспективных направлений является использование обучения с подкреплением (RL) для создания автономных агентов, способных противостоять таким атакам. Однако эффективность упомянутых агентов во многом зависит от качества тренировочных окружений, которые должны балансировать между реализмом, масштабируемостью и эффективностью [2].

Тренировочные окружения, о которых идет речь в статье, являются частью автоматизированной информационной защиты, именуются тренировочными окружениями автономных кибероперациий и предлагают способы улучшения общего уровня защищенности системы [3]. Тренировочные окружения представляют собой функционал для симуляции определенного инцидента безопасности, с их помощью можно моделировать этот инцидент неограниченное количество раз [4], тем самым получая политику, с помощью которой можно эффективно минимизировать потери или предотвратить атаку.

В данной статье рассматривается апробирование разработанного сценария для тренировочного окружения CybORG [5], с помощью обучения агента [6] с использованием алгоритма PPO. Рассматриваются результаты работы по внедрению нового сценария и на его основе формируются краткосрочные доработки [7].

Цель работы – оценить эффективность сценария и предложить пути оптимизации процесса его добавления.

Статья содержит следующие разделы: «Методы «Результаты», «Обсуждение» и «Заключение». В разделе «Методы» будет приведена информация по обучению с подкреплением, продемонстрирован процесс внедрения сценария и процесс внедрения исправ-

Вестник Российского нового университета
Серия «Сложные системы: модели, анализ и управление». 2025. № 4

лений CybORG. Раздел «Результаты» содержит результаты работы, отражающие метрики обучения агента и метрики по улучшению фреймворка CybORG. В разделе «Обсуждение» приведены результаты обучения агента в новом сценарии, недостатки процесса внедрения собственного сценария, процесс анализа, вывода и реализации улучшений для существенной доработки указанного процесса. В разделе «Заключение» указана краткая выжимка по проделанному в статье исследованию.

Методы

Демонстрация процесса внедрения обучения с помощью алгоритма PPO [1] требует также проведения анализа используемых в таком окружении моделей и алгоритмов, позволяющих применять методы обучения с подкреплением. Обучение с подкреплением – это раздел машинного обучения, в котором агент обучается оптимальному поведению через взаимодействие со средой, максимизируя накопленное вознаграждение. В последние годы RL активно развивается, охватывая более сложные сценарии, включая многоагентные системы (Multi-Agent Reinforcement Learning, MARL) и игры с неполной информацией (Stochastic Games with Imperfect Information).

Для определения соответствия полученных в тренировочных окружениях стратегий агентов реальным ситуациям следует рассмотреть ключевые аспекты RL, MARL и стохастических игр с неполной информацией и дополнительно проанализировать современные алгоритмы и сложности, связанные с масштабируемостью, координацией и обучением в условиях неопределенности.

Обучение с подкреплением

В наиболее простом случае обучение с подкреплением формулируется как марковский процесс принятия решений (см. Рисунок 1), заданный кортежем (S, A, P, R, γ) [8],

где S – множество состояний;

A – множество действий;

$P(s' \mid s, a)$ – функция переходов;

$R(s, a, s')$ – функция вознаграждения;

$\gamma \in [0;1]$ – коэффициент дисконтирования.

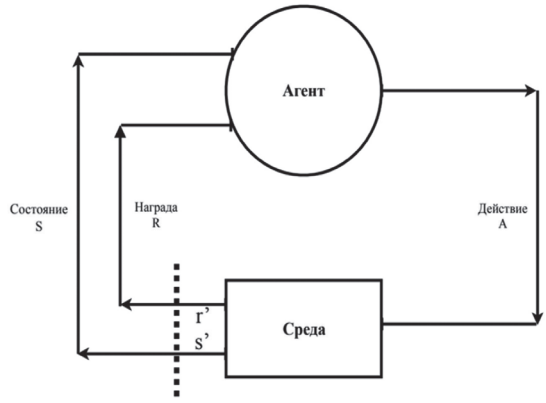

Рисунок 1. Взаимодействие агента и среды
Источник: здесь и далее рисунки выполнены авторами.

Цель агента – найти стратегию $\pi : S \to A$, максимизирующую ожидаемую дисконтированную награду [8]:

$$V^{\pi}(s) = \mathrm{E}_{\pi}\left[\sum_{t=0}^{\infty}\gamma^{t} R\left(s_t, a_t, s_{t+1}\right)\right].$$

Основными алгоритмами для решения задач обучения с подкреплением являются Q-learning, Policy Gradient, Deep Q-Networks (DQN).

Q-learning – алгоритм обучения на основе значений, использующий таблицу Q-значений для оценки полезности действий в состояниях.

Формула обновления Q-значений [9]:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[R + \gamma \max_{a'} Q(s',a') - Q(s,a)\right],$$

где $Q(s,a)$ – текущее значение действия a в состоянии s;

α – скорость обучения (learning rate);

R – награда, полученная за переход в состояние s';

γ – коэффициент дисконтирования (учитывает важность будущих наград);

$\max_{a'} Q(s',a')$ – максимальное Q-значение для следующего состояния s'.

Особенности:

• использует «жадную» стратегию (greedy policy) для выбора действий;

• гарантированно сходится к оптимальной политике при достаточном количестве итераций (в табличном случае);

Policy Gradient – алгоритм прямой оптимизации политики, где параметры политики обновляются в направлении увеличения ожидаемой награды.

Формула градиента стратегии [9]

$$\nabla_{\theta} J(\theta) = \mathrm{E}_{\pi}\left[\nabla_{\theta}\pi \log \pi_{\theta}(a \mid s) \bullet Q^{\pi}(s,a)\right],$$

где $J(\theta)$ – целевая функция (ожидаемая награда);

$\pi_{\theta}(a \mid s)$ – вероятностная стратегия;

$Q(s,a)$ – функция ценности действия.

Особенности:

• работает в непрерывных и дискретных пространствах действий;

• может страдать от высокой дисперсии градиентов (решается методами Actor-Critic).

Deep Q-Networks (DQN) – гибрид Q-learning и глубоких нейросетей, где Q-функция аппроксимируется нейросетью [9].

Ключевые модификации:

• Experience Replay – сохраняет переходы (s,a,r,s') в буфер и обучается на случайных батчах для уменьшения корреляции между данными;

• Target Network – используется отдельная сеть для расчета целевых значений $Q(s',a')$, которая обновляется периодически.

Формула функции потерь [9]:

$$L(\theta) = E_{(s,a,r,s')\sim D}\left[\left(r + \gamma \max_{a'} Q_{target}(s',a') - Q(s,a;\theta)^2\right)\right],$$

где D – буфер воспроизведения опыта.

Вестник Российского нового университета
Серия «Сложные системы: модели, анализ и управление». 2025. № 4

Особенности:

• устойчивость к нестабильности обучения за счет Experience Replay и Target Network;

• применяется в задачах с большими пространствами состояний (например, Atari).

Однако в реальных сценариях защиты компьютерных сетей использование всего одного агента в качестве лица, принимающего решение, является серьезным ограничением и не позволяет рассматривать несколько автономных узлов или действия одновременно как лица, защищающего компьютерную систему, так и лица, нападающего на него.

Многоагентные марковские процессы принятия решений (MARL) расширяют марковские процессы принятия решений на случай нескольких агентов, взаимодействующих в общей среде. Основные модели представлены следующими ситуациями [10]:

• совместные игры (Cooperative MARL) – агенты максимизируют общее вознаграждение (автономные подсети и узлы);

• конкурентные игры (Competitive MARL) – агенты имеют противоположные интересы (например, игры с нулевой суммой для злоумышленника и администратора безопасности);

• смешанные игры (Mixed) – сочетание кооперации и конкуренции.

MARL может быть представлен как стохастическая игра (Markov Game) [10]:

$$\left\langle N, S, \{A_i\}_{i=1}^{N}, P, \{R_i\}_{i=1}^{N}, \gamma \right\rangle,$$

где N – число агентов;

A_i – действия i -го агента;

R_i – его награда.

Основные подходы в таких моделях:

• Independent Q-learning (IQL) – каждый агент обучается как одиночный RL-агент (не учитывает других агентов);

• Centralized training with decentralized execution (CTDE) – обучение с централизованным критиком (например, алгоритм MADDPG);

• Nash Q-learning – поиск равновесия Нэша в стратегиях.

Необходимо отметить, что в реальных сценариях защиты компьютерных сетей агентам недоступна полная информация о состоянии компьютерной системы. Например, синий агент может не знать о скрытом канале управления, используемом злоумышленником, а красному агенту может быть недоступна для исследования вся сеть, или он может получить доступ к учетной записи пользователя с ограниченными правами. Стохастическая игра с неполной информацией (Partially Observable Stochastic Game, POSG) – это обобщение МППР и многоагентного обучения с подкреплением, где агенты наблюдают лишь часть состояния. Формально [10]

$$\left\langle N, S, \{A_i\}_{i=1}^{N}, \{O_i\}_{i=1}^{N}, P, \{R_i\}_{i=1}^{N}, \gamma \right\rangle,$$

где O_i – наблюдения i -го агента.

Такая модель характеризуется проблемой нестационарности: стратегии других агентов меняются, нарушая марковское свойство. К основным методам решения относятся:

• рекуррентные стратегии (использование памяти, например, DRQN);

• Belief state estimation (оценка скрытого состояния через байесовские методы).

Proximal Policy Optimization (PPO) – это алгоритм обучения с подкреплением, который оптимизирует стратегию агента, минимизируя отклонение от предыдущей политики. Это позволяет достичь стабильного обучения даже в сложных средах [11].

Основные шаги алгоритма PPO:

1) сбор данных: агент взаимодействует со средой, собирая траектории (состояния, действия, награды);

2) оценка преимуществ: используется метод Generalized Advantage Estimation (GAE) для оценки преимуществ действий;

3) оптимизация политики: политика обновляется с учетом ограничения на изменение, что предотвращает резкие отклонения.

Формула оптимизации PPO [11]

$$L^{CLIP}(\theta) = E_t[\min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\ell, 1+\ell)\hat{A}_t],$$

где $r_t(\theta)$ – отношение вероятностей действий старой и новой политик;

$\hat{A}_t$ – оценка преимущества;

ℓ – параметр, ограничивающий изменение политики.

Интеграция PPO в CybORG

Для интеграции обучения на основе алгоритма PPO был выбран сценарий, разработанный и внедренный в работе [7] в тренировочное окружение CybORG. Такой подход позволяет верифицировать новый сценарий и проверить основные функции обучения в нем.

Новый сценарий состоит из 9 подсетей, в которых расположены различные хосты – приложения, мониторинга и СУБД [7]. Наличествующие агенты внутри сети обучаются посредством многократного запуска обучения – красный агент (злоумышленник), синий агент (защитник), зеленый агент (пользователи). Дополнительные исследования по рассмотрению тренировочного окружения CybORG также подчеркивают его пригодность к внедрению новых сценариев [6; 12; 13] и выявляют существенные недостатки на различных стадиях работы с ним [14].

Сценарий был реализован в виде Python-кода, интегрированного в среду CybORG. Для проверки корректности работы были проведены тестовые запуски, включая проверку сетевых связей, функциональности агентов и корректной работы основных функций обучения с подкреплением.

Модификация фреймворка CybORG

Используя показанную в прошлой статье [7] работу, можем реализовать следующий список доработок тренировочного окружения CybORG, которые качественно и количественно улучшат упомянутый фреймворк. Список доработок указан в Таблице.

Таблица

Модификации CybORG

Доработка	Основания для ее выбора	Позволяет получить
Создание графического интерфейса для функционала внедрения сценария в CybORG	Нет нативного функционала внедрения нового сценария	С помощью GUI процесс внедрения нового сценария происходит без ручного редактирования кода
Избавление от переопределения данных в смежных модулях CybORG	После внедрения сценария в ядре CybORG переопределяются данные	Обучение может быть запущено на любом новом сценарии без необходимости ручного исправления

Вестник Российского нового университета
Серия «Сложные системы: модели, анализ и управление». 2025. № 4

Указанные доработки основаны на процессе внедрения собственного сценария, описанного в работе [7]. Его реализация подсветила моменты в этом процессе, которые необходимо исправить для более эффективной и массовой работы с тренировочным окружением CybORG. Рассмотрим выполненные доработки подробнее.

Реализация графического интерфейса покрывает следующие процессы:

• написание Python-кода с помощью использования шаблона Jinja2 с инфраструктурой сценария;

• написание Python-кода с помощью использования шаблона Jinja2 по связям внутри новой сети;

• выбор различных типов хостов и подсетей;

• формирование файла, содержащего всю необходимую информацию о новом сценарии.

В текущем формате без использования графического интерфейса добавление подобного нового сценария сопряжено с постоянным написанием кода и большой тратой времени (около 20 часов), поэтому внедрение графического интерфейса GUI является важной доработкой для эффективной работы с тренировочным окружением CybORG.

Реализация предложенной доработки выполнена с помощью Python-модуля PyQT. Заполнение файла сценария использует технологию шаблона Jinja2, что позволяет получить от пользователя с помощью интерфейса параметры сценария и сохранить их в JSON-формате, после чего шаблонным парсером создать полноценный файл сценария. По итогам получения данных от пользователя на экране появится основная информация о сети, содержащихся в ней сегментах подсетей и хостах, сетевых доступах между подсетями.

С помощью шаблона Jinja2 реализуется формирование файла сценария на языке Python данными от пользователя. Такой подход оптимизирует создание и внедрение нового сценария для последующего обучения агента в нем.

Доработка, связанная с переопределением данных в других модулях фреймворка CybORG, направлена на перенос данных нового сценария в смежные модули, так как изначально эти данные присутствовали в виде константы, что нарушает принцип построения ПО – DRY. Передача таких данных, как имена подсетей, количество хостов в них и запущенные процессы в модуль «Враппер», значительно снизит количество ошибок при работе с фреймворком CybORG.

В качестве реализации указанного рефакторинга кода используется стандартный функционал классов в языке Python. Таким образом, полученные от пользователя данные корректно передаются в модуль «Враппер», где они без переопределения используются в дальнейших вычислениях. В связи с этим сокращается число ошибок при работе пользователя с фреймворком CybORG при внедрении нового сценария. Дополнительно введенная доработка приводит код ПО к соответствию принципам DRY и SOLID.

Результаты

Проведенная работа позволяет сделать следующие выводы относительно обучения агента в новой инфраструктуре и общих улучшений тренировочного окружения CybORG.

В части оценки эффективности сценария:

• сценарий успешно интегрирован в CybORG, агенты корректно взаимодействуют со средой, сценарий позволяет успешно обучать агентов, что показывают Рисунки 2 и 3;

• наблюдается реалистичное поведение красного и синего агентов, что подтверждает адекватность моделирования и успешность обучения.

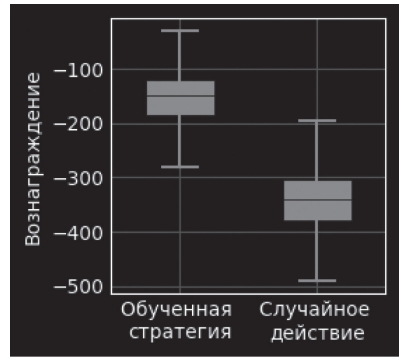

Рисунок 2. Результаты тестирования агента с обученной стратегией PPO и случайного агента

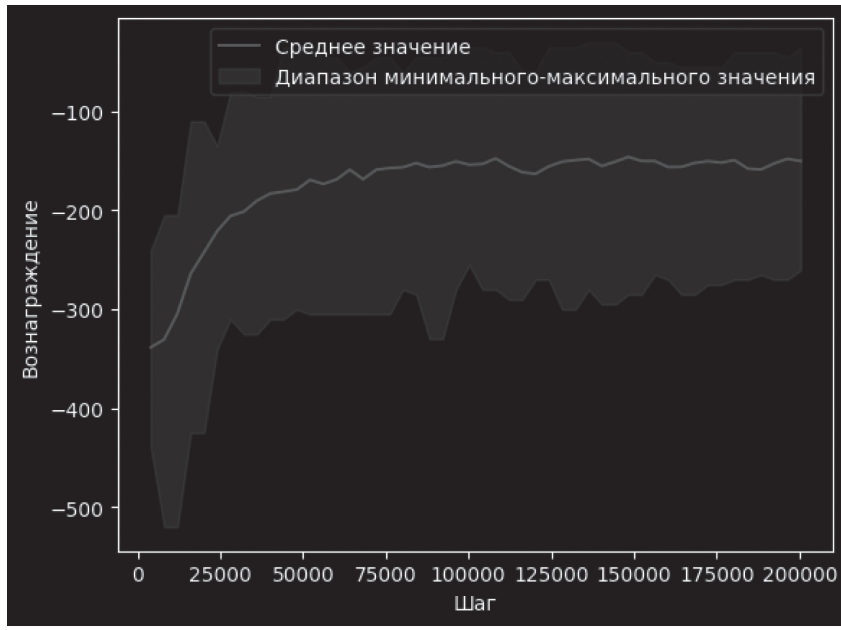

Рисунок 3. Результаты обучения PPO в новом сценарии

Результатами обучения с PPO можно считать следующие положения:

• красный агент научился эффективно использовать уязвимости, такие как перебор паролей и эксплуатация сервисов;

• синий агент разработал стратегии обнаружения и блокировки атак, включая анализ сетевого трафика и изоляцию скомпрометированных хостов;

• метрики – средняя награда агента увеличилась на 40 % после 1000 эпизодов обучения.

По сравнению с алгоритмами DQN и Policy Gradient, особенно в условиях частичной наблюдаемости (POSG), PPO показал лучшую стабильность и скорость обучения.

Вестник Российского нового университета
Серия «Сложные системы: модели, анализ и управление». 2025. № 4

Добавление графического интерфейса GUI, с помощью которого возможно реализовать внедрение нового сценария, значительно ускоряет упомянутый функционал, делает его более эргономичным и эффективным [15]. Для объективного сравнения необходимо указать метрики, которые количественно улучшила данная доработка.

Определим количество времени, требуемое на разработку и внедрение новых сценариев без использования GUI при помощи метрики heatmap с сервиса GitHub, количество которых равно 18. Примерное время, затраченное на каждый из сценариев, будем считать около 60 минут. Без GUI этот процесс занимает около 1080 минут. Обратим внимание, что среди 18 коммитов наличествуют и те, которые были направлены на исправление и тестирование корректности работы, ошибки были вызваны человеческим фактором. Следовательно, можно утверждать, что внедрение графического интерфейса дополнительно минимизирует объем ошибок при работе и время на их исправление.

При использовании GUI время на внедрение нового сценария варьируется от 4 до 13 минут в зависимости от объема среды. Будем считать, что среднее время для внедрения сценария с GUI составляет 9 минут.

Таким образом, внедренная доработка по реализации графического интерфейса ускоряет процесс внедрения нового сценария на 99,17 %, что показано на Рисунке 4.

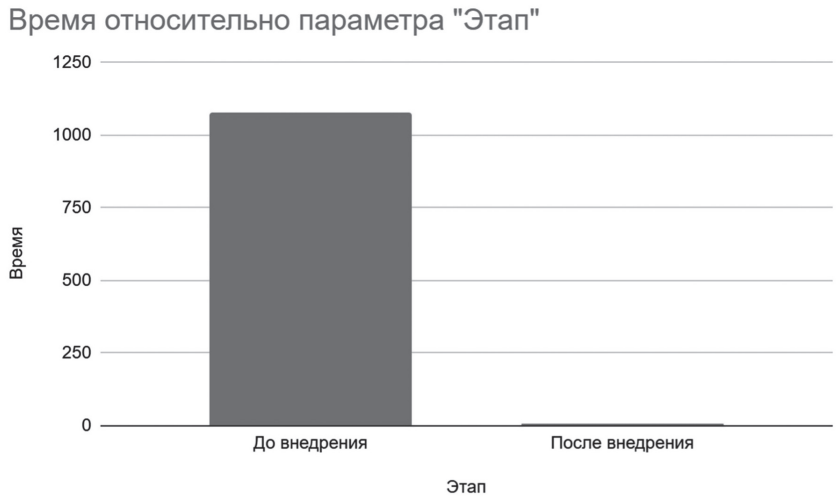

Рисунок 4. Эффективность графического интерфейса

Доработка, связанная с переопределением данных в других модулях фреймворка CybORG, аналогичным образом ускоряет процесс внедрения нового сценария, делая его эффективнее.

Для оценки эффективности воспользуемся аналогичной вышеупомянутой структурой. Таким образом, число коммитов для локализации и исправления ошибки равно 10, тем самым затраченное время составляет 600 минут. При этом после внесения изменений в код проекта затраченное время на потенциальное исправление ошибки будет составлять около 60 минут. Это объясняется тем, что все необходимые значения локализованы и указаны пользователю. Получим следующую метрику для учета эффективности указанной доработки (см. Рисунок 5).

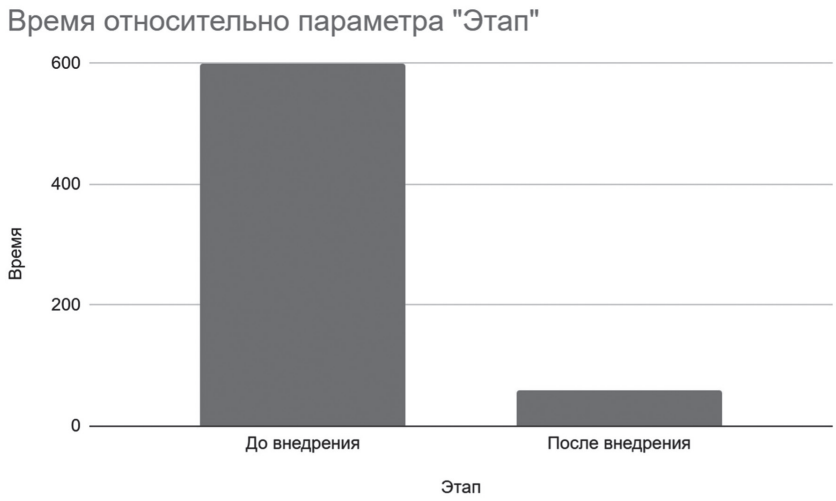

Рисунок 5. Эффективность исправления ошибки

Проведенная работа и ее результаты свидетельствуют о высокой эффективности внедренных доработок и эффективном обучении агента в новой инфраструктуре. Последнее демонстрирует, что тренировочные окружения, обладающие определенным функционалом, можно использовать в качестве инструмента информационной безопасности для анализа и построения политики защиты и увеличения уровня общей защищенности системы.

Обсуждение

В качестве дальнейших шагов в следующих работах необходимо рассмотреть основные ключевые факторы успешности и ограничения в текущем состоянии работы.

Основные факторы успешности:

• качество сценария – реалистичность сетевой топологии и правил взаимодействия во фреймворке CybORG;

• настройка PPO – оптимальный выбор гиперпараметров и функции вознаграждения для обучения;

• интерпретируемость – возможность анализа действий агента для дальнейшей оптимизации обучения;

• оптимальность доработок – реализованные доработки, направленные на основные критические нюансы работы фреймворка CybORG;

• эргономичность тренировочного окружения – графический интерфейс делает более доступным и простым в освоении новый инструмент информационной безопасности.

Ограничения:

• вычислительные ресурсы – обучение требует значительных мощностей;

• разрыв в реальности – необходимость валидации обучения агента на реальных данных;

• эффективность доработок – несмотря на выполненные доработки, во фреймворках всё еще присутствуют недостатки, которые не позволяют их эффективно использовать.

Рекомендации:

• доработка сценария – добавление динамических угроз и уязвимостей, увеличение параметров на хостах;

Вестник Российского нового университета
Серия «Сложные системы: модели, анализ и управление». 2025. № 4

• улучшение PPO – использование иерархического PPO для сложных задач;
• интеграция с SIEM – для повышения реалистичности и валидации стратегий;
• исправление ключевых недостатков – проведение глобального рефакторинга тренировочного окружения.

Заключение

Апробирование разработанного сценария в CybORG с использованием алгоритма PPO подтвердило его эффективность для моделирования атак и защиты в компьютерных сетях. PPO продемонстрировал высокую стабильность и способность агентов обучаться сложным стратегиям. Исправление ключевых недостатков позволило значительно сократить время и уменьшить количество ошибок при внедрении нового сценария.

Дальнейшие исследования будут направлены на поиск решения для исправления долгосрочных доработок путем глобального рефакторинга тренировочного окружения, затрагивая в том числе интеграцию с реальными системами информационной безопасности для ускоренной проверки полученной политики.

Литература

1. *Vyas S., Hannay J., Bolton A., Burnap P.* Automated cyber defence: A review // arXiv. 2023. DOI: https://doi.org/10.48550/arXiv.2303.04926
2. *Tan J., Zhang T., Coumans E., Iscen A., Bai Y., Hafner D., Bohez S., Vanhoucke V.* Sim-to-Real: Learning Agile Locomotor Policies for Quadruped Robots // arXiv. 2018. DOI: https://doi.org/10.48550/arXiv.1804.10332
3. *Легкодумов А.А., Козеев Б.Н., Беликов В.В., Корольков А.В.* Обзор тренировочных окружений с использованием обучения с подкреплением // Вестник Российского нового университета. Серия: Сложные системы: модели, анализ, управление. 2025. № 3. С. 106–124. DOI: 10.18137/RNU.V9187.25.03.P.106. EDN IHCZFW.
4. *Yoo J.D., Park E., Lee G., Ahn M.K., Kim D., Seo S., Kim H.K.* Cyber attack and defense emulation agents // Applied Sciences. 2020. Vol. 10. No. 6. Article no. 2140. DOI: https://doi.org/10.3390/app10062140
5. *Standen M., Lucas M., Bowman D., Richer T.J., Kim J., Marriott D.* Cyborg: A gym for the development of autonomous cyber agents // arXiv. 2021. DOI: https://doi.org/10.48550/arXiv.2108.09118
6. *Kiely M., Bowman D., Standen M., Moir C.* On autonomous agents in a cyber defence environment // arXiv. 2023. DOI: https://doi.org/10.48550/arXiv.2309.07388
7. *Козеев Б.Н., Беликов В.В.* Анализ ограничений в тренировочных окружениях и внедрение собственного сценария безопасности // Вестник Российского нового университета. Серия: Сложные системы: модели, анализ, управление. 2025. № 2. С. 130–148. DOI: 10.18137/RNU.V9187.25.02.P.130. EDN HPZDJW.
8. *Кошманова Н.П., Трифонов Д.С., Павловский В.Е.* Управление манипулятором с помощью обучения с подкреплением // Нелинейная динамика. 2012. Т. 8. № 4. С. 689–704. EDN PKTALT.
9. *Mnih V., Kavukcuoglu K., Silver D., et al.* (2015) Human-level control through deep reinforcement learning // Nature. Vol. 518. Pp. 529–533. DOI: https://doi.org/10.1038/nature14236.
10. *Albrecht S.V., Christianos F., Schäfer L.* Multi-agent reinforcement learning: Foundations and modern approaches. MIT Press, 2024. 396 p. ISBN 0262049376.
11. *Schulman J., Wolski F., Dhariwal P., Radford A., Klimov O.* Proximal Policy Optimization Algorithms // arXiv. 2017. DOI: https://doi.org/10.48550/arXiv.1707.06347

12. *Oesch S., Chaulagain A., Weber B., Dixson M., Sadovnik A., Roberson B., Watson C., Austria P.* Towards a High Fidelity Training Environment for Autonomous Cyber Defense Agents // Proceedings of the 17th Cyber Security Experimentation and Test Workshop (CSET '24). Association for Computing Machinery. New York, USA, 2024. P. 91–99. DOI: https://doi.org/10.1145/3675741.3675752

13. *Oakley L., Oprea A*. An adaptive reinforcement learning strategy for the security game // Alpcan T., Vorobeychik Y., Baras J., Dán G. (Eds) Decision and Game Theory for Security. GameSec 2019. Series: Lecture Notes in Computer Science. Vol. 11836. Springer, Cham, 2019. Pp. 364–384. DOI: https://doi.org/10.1007/978-3-030-32430-8_22

14. *Emerson H., Bates L., Hicks C., Mavroudis V.* Cyborg++: An enhanced gym for the development of autonomous cyber agents // arXiv. 2024. DOI: https://doi.org/10.48550/arXiv.2410.16324

15. *Run Luo, Lu Wang, Wanwei He, Longze Chen, Jiaming Li, Xiaobo Xia*. GUI-R1: A Generalist R1-Style Vision-Language Action Model For GUI Agents // arXiv. 2025. DOI: https://doi.org/10.48550/arXiv.2504.10458

References

1. Vyas S., Hannay J., Bolton A., Burnap P. (2023) Automated cyber defence: A review. *arXiv.* DOI: https://doi.org/10.48550/arXiv.2303.04926

2. Tan J., Zhang T., Coumans E., Iscen A., Bai Y., Hafner D., Bohez S., Vanhoucke V. (2018) Sim-to-Real: Learning Agile Locomotor Policies for Quadruped Robots. *arXiv.* DOI: https://doi.org/10.48550/arXiv.1804.10332

3. Legkodumov A.A., Kozeev B.N., Belikov V.V., Korolkov A.V. (2025) Overview of ACO Gym using reinforcement learning. *Vestnik of Russian New University. Series: Complex Systems: Models, analysis, management.* No. 3. Pp. 106–124. DOI: 10.18137/RNU.V9187.25.03.P.106 (In Russian).

4. Yoo J.D., Park E., Lee G., Ahn M.K., Kim D., Seo S., Kim H.K. (2020) Cyber attack and defense emulation agents. *Applied Sciences*. Vol. 10. No. 6. Article no. 2140. DOI: https://doi.org/10.3390/app10062140

5. Standen M., Lucas M., Bowman D., Richer T.J., Kim J., Marriott D. (2021) Cyborg: A gym for the development of autonomous cyber agents. *arXiv.* DOI: https://doi.org/10.48550/arXiv.2108.09118

6. Kiely M., Bowman D., Standen M., Moir C. (2023) On autonomous agents in a cyber defence environment. *arXiv.* DOI: https://doi.org/10.48550/arXiv.2309.07388

7. Kozeev B.N., Belikov V.V. (2025) Analyzing constraints in ACO Gyms and implementing your own security scenario. *Vestnik of Russian New University. Series: Complex Systems: Models, analysis, management.* No. 2. Pp. 130–148. DOI: 10.18137/RNU.V9187.25.02.P.130 (In Russian).

8. Koshmanova N.P., Trifonov D.S., Pavlovsky V.E. (2012) Reinforcement learning for manipulator control. *Russian Journal of Nonlinear Dynamics*. Vol. 8. No. 4. Pp. 689–704. (In Russian).

9. Mnih V., Kavukcuoglu K., Silver D., et al. (2015) Human-level control through deep reinforcement learning. *Nature*. Vol. 518. Pp. 529–533. DOI: https://doi.org/10.1038/nature14236

10. Albrecht S.V., Christianos F., Schäfer L. (2024) *Multi-agent reinforcement learning: Foundations and modern approaches*. MIT Press. 396 p. ISBN 0262049376.

11. Schulman J., Wolski F., Dhariwal P., Radford A., Klimov O. (2017) Proximal Policy Optimization Algorithms. *arXiv*. DOI: https://doi.org/10.48550/arXiv.1707.06347

12. Oesch S., Chaulagain A., Weber B., Dixson M., Sadovnik A., Roberson B., Watson C., Austria P. (2024) Towards a High Fidelity Training Environment for Autonomous Cyber Defense Agents. In:

Вестник Российского нового университета
Серия «Сложные системы: модели, анализ и управление». 2025. № 4

Proceedings of the 17th Cyber Security Experimentation and Test Workshop (CSET '24). Association for Computing Machinery. New York, USA. P. 91–99. DOI: https://doi.org/10.1145/3675741.3675752
13. Oakley L., Oprea A. (2019) An adaptive reinforcement learning strategy for the security game. In: Alpcan T., Vorobeychik Y., Baras J., Dán G. (Eds) *Decision and Game Theory for Security. GameSec 2019*. Series: Lecture Notes in Computer Science. Vol. 11836. Springer, Cham. Pp. 364–384. DOI: https://doi.org/10.1007/978-3-030-32430-8_22
14. Emerson H., Bates L., Hicks C., Mavroudis V. (2024) Cyborg++: An enhanced gym for the development of autonomous cyber agents. *arXiv*. DOI: https://doi.org/10.48550/arXiv.2410.16324
15. Run Luo, Lu Wang, Wanwei He, Longze Chen, Jiaming Li, Xiaobo Xia (2025) GUI-R1: A Generalist R1-Style Vision-Language Action Model for GUI Agents. *arXiv*. DOI: https://doi.org/10.48550/arXiv.2504.10458

Поступила в редакцию: 28.10.2025	Received: 28.10.2025
Поступила после рецензирования: 17.11.2025	Revised: 17.11.2025
Принята к публикации: 10.12.2025	Accepted: 10.12.2025