DOI: 10.18137/RNU.V9187.25.02.P.130 УДК 004.942

Козеев Борис Николаевич
главный специалист АО «Альфа-Банк», Москва. ORCID: 0009-0009-0993-8082
Электронный адрес: kozeev.boris2018@yandex.ru
Boris N. Kozeev
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 базовой кафедры № 252, Российский технологический университет МИРЭА, Москва. ORCID: 0000-0003-1423-1072
Электронный адрес: belikov_v@mirea.ru
Vladimir V. Belykov
Ph.D. of Military Sciences, Associate Professor of Basic Department No. 252, MIREA – Russian Technological University, Moscow. ORCID: 0000-0003-1423-1072
E-mail address: belikov_v@mirea.ru

АНАЛИЗ ОГРАНИЧЕНИЙ В ТРЕНИРОВОЧНЫХ ОКРУЖЕНИЯХ И ВНЕДРЕНИЕ СОБСТВЕННОГО СЦЕНАРИЯ БЕЗОПАСНОСТИ

Аннотация. В статье проведено исследование существующих автономных тренировочных окружений для обучения агентов информационной безопасности и выявления их актуальных функциональных возможностей и ограничений. Цели: введение в основные понятия и определение автоматизированной информационной защиты. Определение ключевых недостатков существующих решений; разработка нового тренировочного сценария на основе тренировочного окружения CybORG; оценка и описание процесса создания сценария на основе тренировочного окружения CybORG; создание рекомендаций для дальнейшего исправления функционала внедрения нового сценария; формирование рекомендаций для дальнейших исследований и разработки тренировочных окружений, направленных на повышение их эффективности в корпоративном и исследовательском плане в условиях постоянно развивающихся киберугроз. Методы: сбор и анализ информации о функционале, архитектуре и функциональных возможностях популярных платформ; аналитический метод, эксперимент по созданию собственного сценария на основе CybORG; проведение сравнительного анализа по ключевым параметрам; анализ практического применения платформ с точки зрения их корпоративного и пользовательского использования; систематизация выявленных недостатков при анализе для определения требований к доработкам; формирование рекомендаций для дальнейших исследований и совершенствования тренировочных окружений. Результаты: проведен сравнительный анализ пяти популярных платформ: Farland, NASimEmu, Yawning Titan, CybORG и CAGE Challenge. Выявление и предложение векторов доработок для недостатков текущих платформ для обучения агентов информационной безопасности. Проанализирована структура тренировочных окружений, проведено исследование об их функциональности и возможностях, предоставленных для других разработчиков. Обозначены основные понятия и определения автоматизированной информационной защиты. Разработан и реализован собственный сценарий на базе тренировочного окружения CybORG, для изучения процесса работы с функционалом создания нового сценария. Определен вектор дальнейших доработок в существующем функционале с целью использования приведенных платформ не только в академических целях, но и на корпоративном и пользовательском уровне. Выводы: анализ существующих тренировочных окружений подтвердил, что имеющиеся платформы не полностью соответствуют требованиям современных задач информационной безопасности, особенно в условиях быстро эволюционирующих кибератак. Статья предлагает новую основу для

дальнейшего развития тренировочных окружений, которая может быть расширена с учетом будущих угроз и задач в области информационной безопасности.

Ключевые слова: информационная безопасность, тренировочное окружение, автоматизированная информационная защита, обучение с подкреплением.

Для цитирования: *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

ANALYZING CONSTRAINTS IN ACO GYMS AND IMPLEMENTING OWN SECURITY SCENARIO

Abstract. The article examines the existing autonomous training environments for training information security agents and identifies their current functional capabilities and limitations.

Objectives. Introduction to the basic concepts and definitions of automated information security. Identifying the key drawbacks of existing solutions. Development of a new training scenario based on the CybORGACO Gym. Evaluation and description of the scenario creation process based on the CybORGACO Gym, creation of recommendations for further correction of the functionality of the new scenario implementation. Making recommendations for further research and development of ACO Gyms aimed at increasing their effectiveness in corporate and research terms in the context of constantly evolving cyber threats.

Methods. Collecting and analyzing information about the functionality, architecture, and functionality of popular platforms. An analytical method, an experiment to create your own script based on CybORG. Conducting a comparative analysis of key parameters. Analysis of the practical application of the platforms in terms of their corporate and user use. Systematization of the identified shortcomings during the analysis to determine the requirements for improvements. Making recommendations for further research and improvement of ACO Gyms.

Results. A comparative analysis of five popular platforms was carried out: Farland, NASimEmu, Yawning Titan, CybORG and CAGE Challenge. Identification and suggestion of improvement vectors for the shortcomings of current platforms for training information security agents. The structure of ACO Gyms has been analyzed, and a study has been conducted on their functionality and capabilities provided for other developers. The basic concepts and definitions of automated information security are outlined. A custom script based on the CybORG framework has been developed and implemented to study the process of working with the functionality of creating a new script. The vector of further improvements in the existing functionality has been determined in order to use these platforms not only for academic purposes, but also at the corporate and user levels.

Conclusions. An analysis of the existing ACO Gyms has confirmed that the existing platforms do not fully meet the requirements of modern information security challenges, especially in the context of rapidly evolving cyberattacks. The article offers a new framework for the further development of ACO Gyms, which can be expanded to take into account future threats and challenges in the field of information security.

Keywords: information security, ACO Gym, automated information protection, reinforcement learning.

For citation: 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).

Вестник Российского нового университета
Серия «Сложные системы: модели, анализ и управление», выпуск 2 за 2025 год

Введение

В настоящее время можно отметить тенденцию значительного увеличения числа кибератак. Одновременно с этим происходит их усложнение, что выражается в использовании искусственного интеллекта и автоматизации при атаках [1]. Для таких угроз необходим инструмент, позволяющий специалистам по безопасности разработать превентивную стратегию взаимодействия со злоумышленником в ответ на его атаки. Функционал для данной цели предоставляет автономная среда тренировочных окружений (далее – АСТО, тренировочные окружения), которая является частью автоматизированной информационной защиты (АИЗ) [1].

Однако современные АСТО обладают рядом системных недостатков, существенно ограничивающих их эффективность в подготовке специалистов. К ключевым проблемам относятся: низкая документированность, несоответствие функционала документации, неоптимизированный код и недостаточная реалистичность окружений. Эти ограничения требуют системного анализа и классификации.

Особое значение в контексте повышения эффективности АСТО имеет возможность создания и адаптации сценариев под конкретные задачи информационной безопасности. Разработка собственного сценария для фреймворка CybORG позволила не только проверить практическую применимость критериев оценки, но и выявить дополнительные ограничения, требующие доработки платформы.

Существующие сейчас тренировочные окружения Farland [2], NASimEmu [3], YawningTitan [4], CybORG [5] и CAGE Challenge [6] характеризуются определенными достоинствами и недостатками. Для каждой задачи информационной безопасности подбирается соответствующее тренировочное окружение, что определяется наличием или отсутствием ряда критериев [7] в тренировочных окружениях.

Наличие сценария. Инфраструктура вносится в тренировочное окружение путем использования функционала добавления сценария; отсутствие подобного функционала оказывает влияние на гибкость платформы.

Наличие эмулятора. «Разрыв в реальности» возникает благодаря различиям в количестве параметров в симуляторе и реальной инфраструктуре. Эмулятор призван проверить полученную политику агента в реальной сети, поэтому его наличие является важным фактором для смягчения «разрыва в реальности».

Изменение параметров красного и синего агентов. Такие параметры, как цели агентов и их пространства действий являются важными для более точной настройки тренировочного окружения под задачу специалиста по безопасности.

Дифференциация тренировочных окружений по реалистичности среды. Правила взаимодействия злоумышленника и защитника в тренировочном окружении могут быть далеки от действительности, что необходимо учитывать при выборе тренировочного окружения под определенные задачи.

Конфигурация сценария путем изменения единого конфиг-файла. Реализация собственного сценария или внесение изменений в существующий должны внедряться путем использования единого конфиг-файла, содержащего все необходимые переменные;

Результаты обучения. После эффективного обучения агента нет доступа к просмотру человекочитаемых метрик для анализа полученной стратегии.

Обозначенные характеристики являются в том числе и недостатками при их несоответствии, решение которых необходимо для работы с тренировочными окружениями и

для их применения в реальной инфраструктуре, поддержания высокого уровня защищенности информационных систем. Следовательно, актуальной задачей становится исследование и анализ существующих тренировочных окружений с целью выявления критериев их применимости в реальной инфраструктуре.

АСТО предоставляют функционал для многократного повторения ситуации инцидента информационной безопасности с целью получения оптимальной политики действий посредством многократного запуска агента в среде. Каждая такая платформа базируется на обучении с подкреплением, что позволяет путем симуляции инфраструктуры как среды красного и синего агентов, как злоумышленника и защитника получить оптимальную политику действий при конкретном инциденте безопасности. Используя среду, агентов и взаимодействие между ними с правилами обучения с подкреплением, можно выработать АСТО.

Необходимо ограничить область применения тренировочных окружений до взаимодействия защитника и злоумышленника внутри инфраструктуры из-за многочисленных инцидентов безопасности, их природы и комплексности информационной безопасности [7]. Каждая инфраструктура отличается индивидуальным дизайном, с помощью сценария можно задать любую подходящую конфигурацию сети.

В работе представлено рассмотрение недостатков в существующих разработках, осложняющих их коммерческое, корпоративное или пользовательское применение, проведение экспериментов по написанию собственного сценария на основе CybORG.

Методы

Методологическую основу исследования составил анализ текущих разработок в области тренировочных окружений и реализация собственного сценария безопасности на основе (учений) на основе тренировочного окружения CybORG. Тренировочные окружения являются частью АСТО, использующих обучение с подкреплением для возможности изучения ситуации инцидента безопасности для выявления результативной последовательности действий. Данный подход решает несколько задач:

- упрощает работу специалистов по безопасности, поскольку ускоряет процесс анализа прошлых или будущих инцидентов безопасности [1];
- внедряет новые технологии в информационную безопасность [8].

Т.Т. Нгуен и В.Дж. Редди указывают, что «за киберпреступлениями или кибервойнами всегда стоят злоумышленники, поэтому существует острая потребность в человеческом интеллекте в сочетании с машинами для киберпространства» [8]. Д. Йу и соавторы отмечают, что «хотя многие современные технологии, такие как серверная и сетевая виртуализация, могут поддерживать эмуляцию злоумышленников, это по-прежнему требует от специалистов по кибербезопасности большого количества ручного труда» [9].

В настоящее время область применения тренировочных окружений находится на уровне исследований. Платформы для реализации данных работ создаются и улучшаются исследователями и энтузиастами, поскольку до момента их использования в технологических компаниях необходимо вывести продукт на рынок или доработать, чтобы решить задачи, поставленные перед командой специалистов по безопасности.

Обозначим ключевые аспекты тренировочных окружений.

Наличие сценария. Основная инфраструктура вносится в тренировочное окружение путем использования функционала добавления сценария, что позволяет быстро выпол-

Вестник Российского нового университета
Серия «Сложные системы: модели, анализ и управление», выпуск 2 за 2025 год

нять процесс внедрения новой инфраструктуры и, соответственно, тренировочное окружение становится более гибким.

Наличие эмулятора и симулятора. Симулятор необходим для запуска обучения без использования при этом большого количества ресурсов; в то же время для проверки полученной политики в условиях, приближенных к реальным сетям, необходим эмулятор, требующий больших вычислительных ресурсов;

Дифференциация тренировочных окружений по реалистичности среды. Различные тренировочные окружения используются для различных задач. Так, тренировочные окружения со слабой «реалистичностью» используются для проверки алгоритмов RL, а окружения, приближенные к реальности, могут использоваться в качестве инструментов по безопасности;

Изменение параметров красного и синего агентов. Изменение параметров красного и синего агентов, таких как их цели и пространство действий, используется для более точной настройки обучения под собственную инфраструктуру сети, поэтому этот критерий является ключевым.

Конфигурация сценария путем изменения единого конфиг-файла. Конфигурация параметров должна содержаться в единых конфигурационных файлах, например YAML или XML, позволяя быстро вносить изменения в функционал тренировочного окружения.

Обзор существующих тренировочных окружений

Обзор тренировочных окружений необходимо начать с указания перечня окружений, выбранных для обзора. Это платформы Farland [2], NASimEmu [3], Yawning Titan [4], CybORG [5] и CAGE Challenge [6]. Выбор данных тренировочных окружений обусловлен следующими критериями:

- наличие репозитория с открытым кодом;
- публикации и статьи;
- приближенность тренировочного окружения к реальной инфраструктуре.

Для репрезентативной выборки и сравнения тренировочных окружений следует представить результаты анализа в виде таблицы. Такой подход позволяет визуально отразить проделанную работу по исследованию и анализу.

Для комплексной оценки пяти ведущих платформ (Farland, NASimEmu, Yawning Titan, CybORG, CAGE Challenge) применена система критериев с дополнительными критериями, что позволит верифицировать выведенные критерии как актуальные и подходящие для оценки тренировочных окружений. Для систематизации анализа можно разделить выделенные критерии на две группы:

- функциональные критерии:
  – создание сценария;
  – наличие эмулятора;
  –гибкость агентов;
  – реалистичность окружения;
  – редактирование через конфиг-файл.
- технические критерии:
  – полнота документации;
  – соответствие документации функционалу.

Проанализируем соответствие платформ указанным критериям (см. Таблицу).

Таблица

Соответствие платформ выделенным критериям

Категории	Среда				
	Farland	**NASimEmu**	**Yawning Titan**	**CybORG**	**CAGE Challenge**
Создание сценария	–	+	–	+	–
Полнота документации	–	+	+	+	+
Соответствие документации	–	–	–	–	–
Реалистичность окружения	+	–	–	+	+
Наличие эмуляции	+	+	–	+	+
Гибкость агентов	+	+	+	+	+
Редактирование через конфиг-файл	+	+	–	+	+

Источник: таблица составлена авторами.

В Таблице указаны значения соответствия установленным семи критериям. Соответствиям из Таблицы необходимо придать бинарное значение, в котором знак «плюс» будет означать удовлетворение критерию, а знак «минус» – неудовлетворение заданному критерию. Выполнение анализа базируется на прилагающейся к платформе документации, находящейся в GitHub-репозитории проекта, и научных статьях, опубликованных авторами. Задача, стоявшая при прочтении статей и изучении платформ, описывается как нахождение базовых аспектов тренировочных окружений. Для масштабного применения данных продуктов компаниями необходимо внести корректировки. Для последующей работы необходимо дать краткое описание каждого из фреймворков на основе полученной информации из репозиториев и публикаций авторов.

Платформа *Farland* [2] – это фреймворк для обучения с подкреплением в задачах автономной защиты сетей. Его ключевая особенность – возможность постепенного усложнения сетевых сред, что позволяет агентам плавно повышать уровень мастерства – от начального до высокого. Подход Farland основан на методах случайного автоматизированного обучения и динамического дизайна среды, что помогает агентам осваивать сложные задачи. Фреймворк включает модели противодействия, имитирующие атаки на агентов, что позволяет оценивать их устойчивость в условиях реальных угроз.

NASimEmu [3] – это фреймворк для обучения агентов глубокого обучения с подкреплением в задачах пентеста (тестирования на проникновение). Он включает симулятор и эмулятор, позволяя обучать агентов в виртуальной среде, а затем переносить их в реалистичные условия. Симулятор быстро генерирует случайные сетевые сценарии с разной топологией, конфигурацией и количеством хостов, а эмулятор работает на реальных инструментах вроде Vagrant, VirtualBox и Metasploit. Фреймворк поддерживает динамические сценарии (например, модели университетских или корпоративных сетей), где часть параметров фиксирована (топология, ОС, эксплойты), а часть варьируется (размер сети, настройки хостов). Это помогает обучать агентов, способных адаптироваться к новым, незнакомым сетям. NASimEmu позволяет тренировать и тестировать агентов сразу на нескольких сценариях, улучшая их обобщающую способность.

Фреймворк *Yawning Titan* [4] – это абстрактная графовая среда для симуляции кибербезопасности, предназначенная для обучения автономных защитных агентов. В текущей

Вестник Российского нового университета
Серия «Сложные системы: модели, анализ и управление», выпуск 2 за 2025 год

версии фреймворк поддерживает только оборонительные сценарии, где агенты противостоят вероятностным атакам «красной команды». YT разработан с упором на простоту, низкие требования к ресурсам и гибкость, он позволяет настраивать правила среды, тестировать разные алгоритмы и визуализировать результаты в виде GIF-анимаций.

CAGE Challenge 4 [6] – это серия публичных соревнований, направленных на развитие автономных агентов для киберзащиты. В рамках этих испытаний участники решают задачи, основанные на реальных сценариях кибербезопасности, в симуляции среды. Первые три этапа прошли в 2021–2022 гг., а текущий, CAGE Challenge 4 (CC4), фокусируется на защите корпоративной сети с использованием Multi-Agent Reinforcement Learning (MARL), где несколько агентов работают совместно. Для симуляции используется платформа CybORG (Cyber Operations Research Gym) [5], обеспечивающая реалистичные условия для обучения и тестирования алгоритмов, включая глубокое обучение с подкреплением (Deep RL). CAGE Challenge постепенно увеличивают сложность и реалистичность сценариев, помогая развивать технологии автономной обороны с помощью искусственного интеллекта.

CybORG (Cyber Operations Research Gym) [5] представляет собой фреймворк для исследований в области кибербезопасности, предназначенный для обучения и оценки автономных агентов. Этот инструмент предлагает унифицированный функционал для работы с различными средами – от абстрактных симуляций сетевых конфигураций до реалистичных эмулированных сред на основе облачных виртуальных машин. CybORG активно используется в серии соревнований CAGE Challenge, где тестируются возможности автономных систем киберзащиты, включая сценарии с множеством взаимодействующих агентов (Multi-Agent RL). Фреймворк отличается гибкостью и открытостью – исходный код доступен на GitHub, что позволяет исследователям адаптировать среду под конкретные задачи, например, для обучения агентов в конкретной среде.

Проанализируем обозначенные критерии:

- создание сценария;
- полнота документации;
- соответствие документации;
- реалистичность окружения;
- наличие эмуляции;
- гибкость агентов;
- редактирование через конфиг-файл.

Функционал окружения должен предоставлять возможность формирования собственного сценария. Полнота документации выражается через наличие полноценного репозитория с кодом проекта и публикаций с указанием функционала окружения и инструкции для его использования в качестве фреймворка. Необходимо проверить функционал на соответствие заявленному в документации. Следует рассмотреть пороговый критерий – реалистичность окружения – и учесть фактор правил взаимодействия злоумышленника и защитника и состав инфраструктуры сети. Рассмотрим представленные критерии более подробно.

Первый критерий подразумевает, что сценарий должен быть создан с учетом задач обучения, особенностей предметной области и потребностей исследователей. Тем самым, например, если нам необходимо обучить агента пентестингу, то в таком случае необходимо создавать среду с большим количеством действий агента.

Второй критерий подразумевает, что документация включает необходимые материалы для понимания и анализа функционала тренировочного окружения: пользовательские инструкции, возможности окружения, примеры использования и др. Полнота документации оценивается по наличию перечня функционала в самой документации или в представленных материалах научных статей.

Третий критерий основывается на том, что упомянутый функционал из второго критерия соответствует заявленному ожидаемому алгоритму работы. Иными словами, если в перечне указывается функциональная возможность добавления в операционную систему большего количества процессов, тем самым увеличивая наблюдение агента, то при соответствующем добавлении большего количества процессов ожидаемо увидеть нативную работу тренировочного окружения с увеличенным наблюдением.

Четвертый критерий указывает на максимальную реалистичность тренировочного окружения, что выражается в корректных взаимодействиях между злоумышленником и защитником, в реалистичных взаимодействиях последних с инфраструктурой и правильном поведении среды. Например, реалистичность среды в тренировочном окружении в работе [4] под вопросом, в то же время во фреймворке CybORG является очень высоким.

Пятый критерий – «наличие эмуляции» – является следствием такого феномена, как «разрыв в реальности» [5]. В действительности такое имеет место, так как симуляция фреймворка не является максимально точной копией реальной инфраструктуры, что выражается в меньшем количестве параметров в наблюдении агента. Тем самым стоит задача проверки полученной политики агента путем ее запуска в эмуляторе.

Шестой критерий – «гибкость агентов» – репрезентует наличие механизма настройки в тренировочном окружении параметров агентов, таких как изменение их цели и пространства действий.

Седьмой критерий – «редактирование через конфиг-файл» – служит для демонстрации реализации функционала сценария путем использования единого конфигурационного файла, что значительно упрощает внедрение нового сценария. Указанный критерий не привязан к формату файлу – будь то YAML, XML или, как в последней версии CAGE Challenge 4 Python, файл, формирующий словари с информацией об инфраструктуре.

Исследование автономных тренировочных окружений и внедрение нового сценария нуждается во всестороннем исследовании, которое включает качественные и количественные методы. Оценка по предложенным критериям [7] позволит обеспечить высокое качество разработки и дальнейшее использование тренировочных окружений, а сами критерии после их применения можно считать верифицированными. Внедрение новых технологий и подходов в АСТО открывает новые возможности, удовлетворяя потребности современного общества.

Анализ тренировочных окружений

Проанализируем упомянутые тренировочные окружения, основываясь на источниках, – научных работах авторов и документации в репозиториях, а также на проведенном и показанном в работе эксперименте по внедрению нового сценария. Последующий анализ будет исходить с точки зрения применения тренировочных окружений в корпоративном или пользовательском уровне. Системный анализ современных тренировочных окружений позволяет выявить системные ограничения, существенно снижающие их

Вестник Российского нового университета
Серия «Сложные системы: модели, анализ и управление», выпуск 2 за 2025 год

операциональную эффективность в их текущей реализации, иными словами, анализ необходим для выявления и получения свойств, которые являются фундаментальным для позиционирования тренировочных окружений в качестве эффективных решений для использования в корпоративных или частных целях.

Необходимо обеспечить следующие пункты:

- нативную работу программы;
- удобство в разработке собственного сценария;
- добавление новых сущностей в среду;
- подготовку агентов;
- читабельность результатов;
- визуализацию и эргономику кода, перенеся смысл данного термина к работе с кодом.

Рассмотрим недостатки в существующих тренировочных окружениях в контексте цели – построение тренировочного окружения с использованием средств обучения с подкреплением, функционал которого позволяет воспроизводить обучение агентов, получать читаемые результаты обучения и внедрять собственную инфраструктуру для собственного сценария, а также является максимально приближенным к реальности. Проанализируем и предложим поиск возможностей для их разрешения.

Тренировочные окружения Farland [2], NASimEmu [3], Yawning Titan [4], CybORG [5] и CAGE Challenge [6] отражают реальное взаимодействие со злоумышленником. Процесс решения поставленной задачи с помощью тренировочного окружения предусматривает выбор фреймворка под подходящие цели, создание в нем прототипа инфраструктуры, дальнейшее обучение агентов и последующий анализ полученных результатов их обучения.

Основная проблема текущих АСТО – это исследования, в которых среда предстает заведомо отдаленной от реальности [10]. Например, с помощью таких АСТО легко можно проверять новые алгоритмы обучения с подкреплением, тестируя их в среде, которая представляет собой набор правил, больше похожих на игровые, чем на существующие в реальном взаимодействии злоумышленника и атакующего, что является основным останавливающим фактором применения данных платформ в реальных информационных системах [10]. Такой формат исследований предоставляет возможность исследователям и энтузиастам апробировать новые алгоритмы обучения с подкреплением или проводить соревнования с полноценным сценарием, и его нарративом с последующим награждением лучших участников. Наличие опубликованных научных работ с описанием указанных АСТО позволяет проследить путь их создания и тех проблем, которые авторы разрешили своими разработками [11]. Данные работы являются невалидными по критерию текущего исследования тренировочных окружений с использованием обучения с подкреплением, звучащим как адаптированность платформы для ее использования в реальных сетевых инфраструктурах.

Второй недостаток – отсутствие встроенного редактора для сценария. Такая функция является одной из важнейших, поскольку тренировочные окружения представляют собой фреймворк, позволяющий задавать конкретную инфраструктуру, сетевые доступы, процессы, операционные системы, сервисы, пользователей и другие характеристики. За правильную передачу данной информации во фреймворк отвечает сценарий. Разработчики предоставляют фреймворк, который всегда использует только одну конкретную

инфраструктуру для обучения, как, например, в соревнованиях CAGE Challenge 1-4 [6]. В процессе обучения агент взаимодействует со средой, которую создает фреймворк, путем использования симуляции на основе сценария. В случае отсутствия сущности сценария фреймворк становится негибким, следовательно, его использование будет сопряжено с постоянным взаимодействием с одной и той же сетевой инфраструктурой.

Когда сущность сценария реализована во фреймворке, это дает возможность изменить инфраструктуру. Наличие сущности сценария во фреймворке является необходимым и достаточным аспектом для гибкой работы с ним, но в процессе внедрения могут быть допущены ошибки, замедляющие процесс интеграции новой инфраструктуры при внедрении и использовании такого рода решений:

- необходимость повторяющегося мануального ввода связей и серверов;
- исправление зависимостей внутри самого фреймворка.

В то же время претензии к решениям в разработке не всегда применимы к некоммерческим компаниям, работающим по грантам. Валидными такие претензии являются по отношению к коммерческим и зарабатывающим на таком продукте компаниям.

Третий недостаток подразумевает существование некоторого перечня функционала платформы, которым можно воспользоваться при работе с ней, например, перечень фич в документации окружения. Для решения поставленных задач с помощью определенного тренировочного окружения необходимо учесть написанное авторами в документации. В момент изучения нового тренировочного окружения и в аналитической работе по вычленению его индивидуальных функциональных особенностей, могут возникнуть трудности с неполным упоминанием функционала в тексте статей или документации [12], что приводит к необходимости мануальной проверки путем отладочных запусков или чтения кода. Приведенные работы снабжены документацией в репозитории проекта на Github и научной статьей. Таким образом, без прямого указания всех аспектов функционала сложно организовать работу с данным продуктом в корпоративном или более широком масштабе – потребуется наработка экспертизы сотрудников для настройки программы, а также возникает риск заклада функций в бизнес-процесс, не предусмотренных продуктом.

Такие риски наглядно демонстрирует последующая работа в среде CybORG –отмечается возможность создания собственного сценария и запуска эмулятора, но в зависимости от конфигурации сценария существуют еще в нескольких других файлах с кодом, а эмулятор не проработан до конца [12]. Именно поэтому требуется четкость формулировок функционала тренировочного окружения.

Четвертый пункт посвящен анализу результатов обучения агента. После успешной работы агента специалистам по безопасности или исследователю необходимо проанализировать его обучение с помощью понятных метрик, помогающих оценить:

- степень защищенности системы;
- действия агента в ходе обучения;
- полученный результат;
- другие необходимые для аналитики выборки.

В результате обычно выводятся показатели, которые больше относятся к обучению с подкреплением, например, наблюдение агента на каждом шаге [10]. Из таких выходных данных проблематично сформировать позицию о защищенности системы, не предусмотрено встроенного построения графиков или целых дашбордов, данные по итогам обуче-

Вестник Российского нового университета
Серия «Сложные системы: модели, анализ и управление», выпуск 2 за 2025 год

ния часто поставляются в сыром виде в таблицах или CSV-форматах. Оптимальным вариантом является предоставление встроенного инструментария визуализации результатов с возможностью изменения масштаба зависимых метрик и построения временных рядов.

Пятый пункт связан с третьим и первым пунктами – отсутствием или неполным функционалом создания сценария. В нем рассматривается неполный функционал создания собственного сценария, что выражается в технических трудностях. Задачи, сопряженные с созданием сценария, порождают необходимость внесения доработок в уже готовый файл сценария, хранящийся в репозитории платформы. Также необходим и дополнительный рефакторинг кода в смежных файлах проекта, содержащих зависимости в виде имен переменных или других данных. Отметим, что рефакторинг тоже сопряжен с необходимостью проверки корректности работы кода и исправления найденных ошибок [13]. В случае возникновения более серьезных ошибок наилучшим решением будет дебаг кода с помощью инструментов среды разработки. Иными словами, неполноценно доработанный функционал внедрения собственного сценария сопряжен с дополнительными работами по его корректной работе и настройке. Обозначенные факторы формируют неэффективную среду использования выбранного окружения, ведь специалисту по безопасности необходимо погрузиться в код проекта, рассмотреть и исправить все зависимости, затем с помощью использования инструментов среды разработки исправить код, потом приступать к работе с платформой. Такой подход затрудняет исследование тренировочных окружений и работу с ними.

Таким образом, АИЗ является передовым направлением в области ИБ, обучение с подкреплением представляет собой инструмент для построения АСТО, а АСТО позволяют увеличить уровень общей защищенности систем и выявить оптимальную политику поведения при противодействии злоумышленнику [13]. На текущем этапе развития наличествуют определенные недостатки в платформах, которые не позволяют результативно использовать тренировочные окружения, локализация этих недостатков и показ путей решений для них являются важными составляющими процесса их улучшения.

Создание собственного сценария

В этом разделе показан эксперимент по созданию собственного сценария на основе фреймворка CybORG. Проведенный эксперимент проводится с целью апробации функционала внедрения сценария в выбранном тренировочном окружении. Продемонстрированы трудности в процессе внедрения сценария, соответствующие более точечно для указанного фреймворка.

Из-за отсутствия интерфейса для изменения сценария возникает необходимость мануальной проверки и анализа кода для его дальнейшей модификации. Сценарий представляет собой файл с кодом, содержащий словари с информацией об инфраструктуре, содержащихся в ней хостах, сетевых доступах между ними, операционных системах, запущенных процессах и пользователях.

После определения файла сценария и анализа его содержимого необходимо составить структуру собственной инфраструктуры для ее переноса во фреймворк путем записи информации о ней в найденный файл сценария. Инфраструктура представляет собой несколько связанных подсетей и хостов в них.

На Рисунке 1 показана инфраструктура, представляющая собой определенное корпоративное приложение, поставляющее свои услуги различным пользователям путем до-

ступности функционала в глобальной сети Интернет. В этой инфраструктуре расположены такие зоны, как DMZ, PVLAN1-5, CRITICAL1-2, каждая из которых выполняет свою роль и доступна из других подсетей, что обозначено на рисунке стрелками.

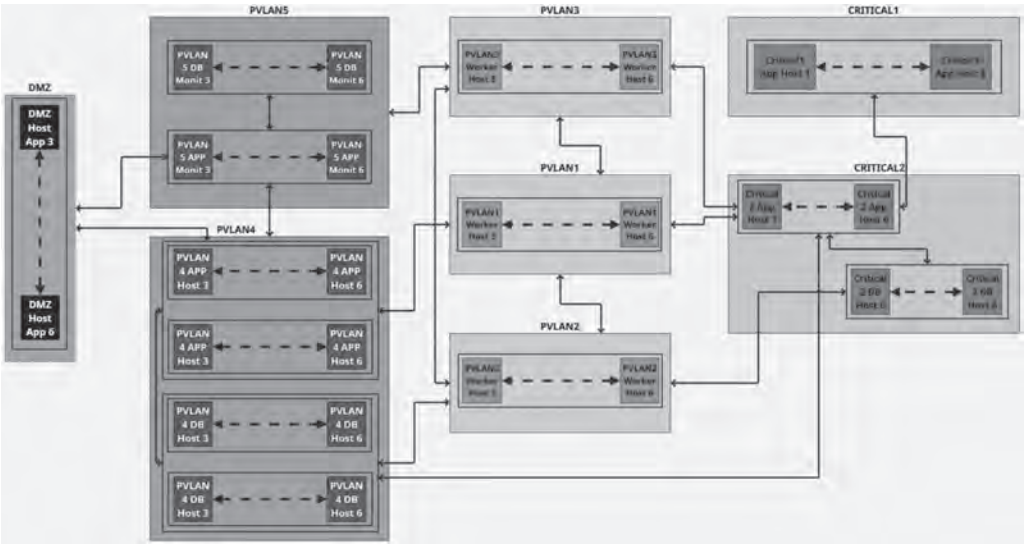

Рисунок 1. Схема инфраструктуры сценария
Источник: рисунок выполнен авторами.

Серой зоной отмечены серверы DMZ. Их задача заключается в том, чтобы выводить функционал приложения в сеть Интернет. Красная зона – это приложение, выполняющее пользовательскую бизнес-логику и мониторинг, иными словами, в этих подсетях размещен слой пользовательского интерфейса, бэкэнд приложения и система мониторинга серверов. В зеленой зоне расположены компьютеры сопровождения и развития приложения, иным словами, это рабочие места сотрудников компании. Каждая из этих подсетей ответственна за сопровождение и развитие определенного функционала, например, мониторинга или СУБД. Синим цветом выделена зона ядра приложения, то есть эта часть приложения напрямую недоступна пользователю и служит для обработки основных запросов, а сами серверы из подсети CRITICAL1 необходимы для обработки данных и их последующей отправки в другие системы, например, регулятору. Такое разделение логики приложения ориентировано на реальные сетевые инфраструктуры, тем самым такое построение сети репрезентирует текущие информационные системы.

На Рисунке 1 обозначено, что инфраструктура разделена на несколько подсетей, каждая из которых выполняет определенную роль:

• CRITICAL1 – выполняет функционал основного приложения компании и необходим для связи с другими компаниями через шлюз; при сбоях в работе приложений из данной подсети ошибки наблюдаются у сотрудников компании и у потребителей, что влечет репутационные и материальные риски;

• CRITICAL2 – подразделяется на две подсети: CRITICAL2_DB и CRITICAL2_APP, которые тоже выполняют роль основного приложения – ядра системы;

Вестник Российского нового университета
Серия «Сложные системы: модели, анализ и управление», выпуск 2 за 2025 год

• PVLAN1 – выполняет роль сопровождения и разработки приложений, запущенных на хостах в подсети CRITICAL2_APP, и PVLAN4_APP_1 и PVLAN4_APP_2; при нарушении их работы возможны ошибки в работе приложений в указанных подсетях;

• PVLAN2 – выполняет роль сопровождения и разработки СУБД, запущенных на хостах подсетей CRITICAL2_DB, и PVLAN4_DB_1 и PVLAN4_DB_2; при нарушении их работы возможны ошибки в работе БД или потеря данных;

• PVLAN3 – является сопровождением и развитием систем мониторинга; на мониторинге находятся все серверы системы, кроме тех, в которых располагаются работники, то есть PVLAN1-3 и CRITICAL1;

• PVLAN4 – содержит серверы приложения PVLAN4_APP_1-2 и баз данных PVLAN4_DB_1-2; данные серверы обеспечивают работу приложения и бизнес-логики;

• PVLAN5 – содержит серверы, выполняющие роль мониторинга других серверов системы; включает в себя подсеть с базами данных PVLAN5_DB и приложения PVLAN5_APP;

• DMZ – подсеть содержит серверы DMZ, с которых происходит проксирование приложения из PVLAN4_APP в сеть Интернет.

На Рисунке 1 отображены связи между подсетями в виде сетевых доступов между ними. Все указанные связи, подсети, серверы, их операционные системы и запущенные процессы создаются в файле сценария. Изменения, вносимые в основной файл со сценарием в репозитории – scenarios.py, – изменяют структуру и связи в формировании подсети для обучения агентов. Таким образом, можно использовать приведенную на Рисунке 1 инфраструктуру в качестве среды для обучения агентов. Для сравнения с изначальной инфраструктурой из CAGE Challenge 4 необходимо ее продемонстрировать на Рисунке 2.

Для CAGE Challenge 4 характерно разбиение логики на составные модули, взаимодействующие между собой для симуляции инфраструктуры и обучения агента. Для достижения реалистичной симуляции используется сценарий с параметрами, указанными в тексте выше. Для корректной логики работы теорем обучения с подкреплением используется фреймворк OpenAIGymnasium [14], реализующий функционал обучения с подкреплением. Тем самым достигается корректность обучения агентов. Действия агента и состояния среды регулируются данными из сценария – сетевой связанностью и запущенными процессами. Состояние среды, доступное агенту, называется наблюдением, зависит от того, какая часть сети доступна агенту, и его осведомленности о процессах на серверах в сети.

Таким образом, можно заключить, что в CAGE Challenge 4 обучение агента складывается из нескольких составляющих: симуляции на основе сценария и корректной работы обучения с подкреплением. Однако в указанных составляющих наличествуют параметры, которые отличаются определенной «оберткой» (wrapper англ.), как это необходимо, например, для начальной инициации таких параметров фреймворка, как количество агентов, их начальные подсети, формула вычисления наблюдений агентов, сетевая связность подсетей и др. Заметим, что эти данные представлены в файле сценария, тем самым «обертка» их переопределяет. Такой подход решает в том числе нижеописанную проблему.

В документации к CAGE Challenge 4 подсвечивается необходимость в случайной конфигурации инфраструктуры для обучения, поскольку это позволит полученной политике агента меньше зависеть от точной архитектуры сети. Поддержка данного функционала представлена на уровне кода: количество машин в конкретной подсети и с конкретной

ролью (СУБД, приложение, АРМ) задается в определенном диапазоне, например, от 4 до 7. Реализовано обучение с учетом различных сетевых политик, которые влияют на сетевую доступность между подсетями; в процессе обучения чередуется их доступность. Такое внесение случайности в процесс обучения воздействует на полученные результаты агента, придавая им реалистичность.

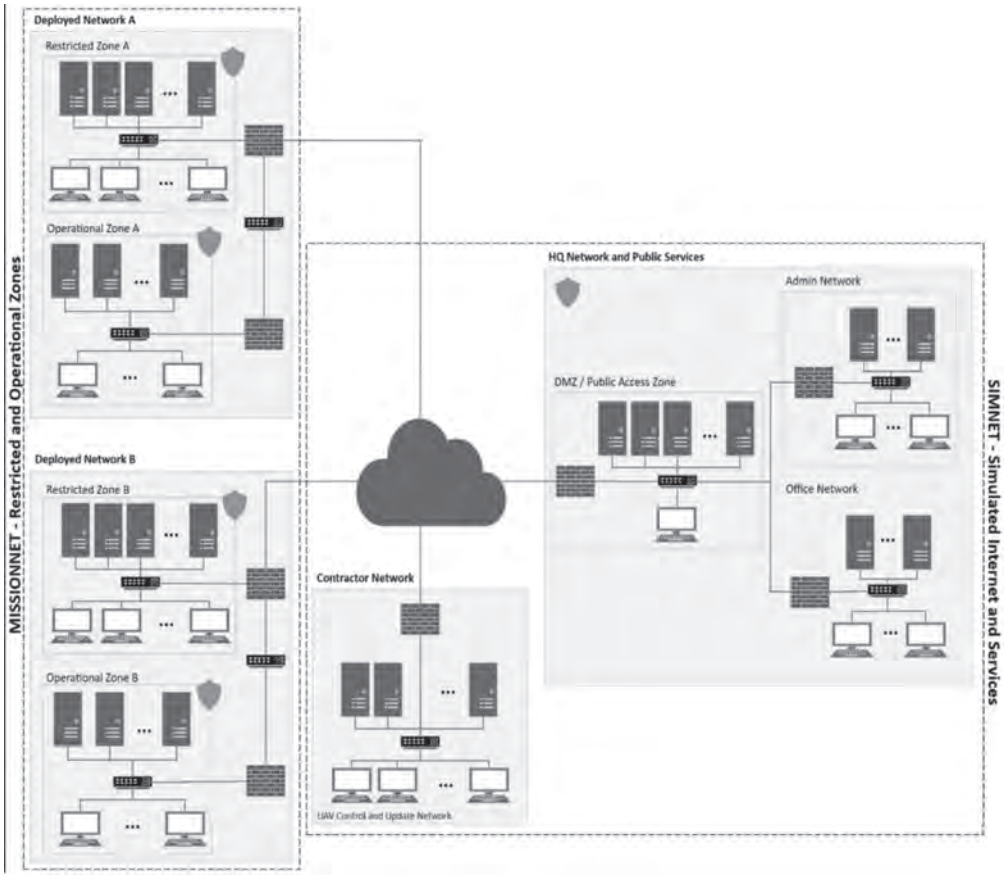

Рисунок 2. Инфраструктура CAGE Challenge 4

Источник: CAGE Challenge 4 // GitHub, Inc. URL: https://github.com/cage-challenge/cage-challenge-4?tab=readme-ov-file#challenge-details (дата обращения: 24.04.2025).

Все вышеуказанные выводы справедливы и для фреймворка CybORG, так как именно на его основе создано учение CAGE Challenge 4. Различия в CybORG, в зависимости от версий учений, не настолько значительны, и весь функционал обучения остался неизменным. В 4-й версии учений добавился модуль, позволяющий создать сценарий в виде Python-кода и успешно собрать оттуда данные для инфраструктуры.

В CAGE Challenge 4 – мультиагентное обучение, в среде задействованы сразу несколько агентов: красный (злоумышленник), синий (защитник) и зеленый (рабочие). У каждой группы агентов есть определенные области, куда они имеют доступ. Так, в текущем (разработанном) сценарии у красного и синего агентов есть доступ ко всем подсетям, в то

Вестник Российского нового университета
Серия «Сложные системы: модели, анализ и управление», выпуск 2 за 2025 год

время как у зеленого агента – только в PVLAN1-3. Все связи между подсетями и доступности подсетей для агентов реализованы в коде в файле сценария.

Дальнейшие действия носят исключительно технический характер, необходимы для последующей проверки, диагностики ошибок, корректности работы сценария и обучения агента. После перенесения собственной инфраструктуры в код следует настроить окружение для тестирования на корректность выполнения и отображения инфраструктуры. Для этого необходимо установить в Venv виртуальное окружение с Python, модуль CybORG, который поставляется в репозитории с CAGE Challenge 4. Для быстрой установки всех зависимостей разработчиками предусмотрен setup-файл, содержащий команды для настройки и установки зависимостей. Затем необходимо проверить корректность среды посредством запуска кода с тестовым участком, что реализуется с помощью вывода инфраструктуры, действий агентов, связей между серверами и прочих информационных сообщений.

Сравнительный анализ двух инфраструктур – из CAGE Challenge 4 (см. Рисунок 2) и из текущей работы (см. Рисунок 1) – можно провести по определенным численным характеристикам: количество серверов, подсетей, но это не всегда будут объективные данные. Инфраструктура создается для определенных целей, поэтому единственным релевантным критерием является реалистичность инфраструктуры, что справедливо для обеих подсетей.

В сценарии в качестве точек входа в подсети используются роутеры. Каждая подсеть имеет один дополнительный роутер, который используется только с целью сетевой доступности, пользователи и запущенные сервисы отсутствуют. На серверах добавляются пользователи и группы, а также прописываются запущенные сервисы, связанные с пользователями и PID-процессом.

После добавления в файл сценария и установки необходимых зависимостей необходимо верифицировать созданный сценарий на корректность работы и проверить локальные переменные с помощью инструментов среды разработки. Данный подход минимизирует дальнейшие риски и менее затратен, чем написание автоматического тестирования. Во время разработки наблюдались незначительные ошибки, связанные с наличием зависимостей в коде внутри других файлов, после исправления которых созданный сценарий был использован в CybORG в качестве среды. Ключевые функции – Step() и Reset() – были осуществлены эффективно [5].

Таким образом, эксперимент по созданию и внедрению собственного сценария показал, что критерий с наличием функционала добавления сценария в тренировочном окружении является неотъемлемой и важной их частью. Дополнительно проведенная работа позволила локализовать и подсветить ограничения в работе фреймворка CybORG, что позволит сформировать список доработок.

Результаты

После проведенного анализа необходимо сформулировать и структурировать полученные результаты. Анализ списка существующих тренировочных окружений показал, что имеющимся разработкам в данной области необходимы общие и частные улучшения в функционале конкретной платформы. Под общими улучшениями подразумеваются рекомендации, направленные на аспект повсеместного использования данной технологии, в том числе применений в реальных корпоративных или государственных информацион-

ных системах. Частные улучшения для каждой обозначенной платформы индивидуальны и были выработаны путем исследования и анализа приложенных научных работ и документации к проектам. Проверить функциональность и практическое использование всех перечисленных тренировочных окружений не всегда возможно из-за отсутствия открытого кода у некоторых из них и времени, затраченного на пробный запуск.

Настоящее состояние и уровень функционала современных тренировочных окружений не позволяет осуществить их корпоративное использование из-за ключевых проблемных зон:

- недостаточность полноты документации;
- недостоверность функционала в документации;
- не полностью разработанный функционал;
- нерепрезентативная реалистичность тренировочного окружения.

Это опосредованно влияет на скорость развития тренировочных окружений, поскольку личное или корпоративное использование будет замедляться, финансируемое недостаточно и только в порядке частных инициатив, будучи созданным только для энтузиастов и исследователей. В области частных улучшений применим аналогичный вывод: если не создать приведенный функционал, то выход платформы в общий доступ замедлится.

На основе разработанных критериев проведена верификация отобранных современных тренировочных окружений [2–6], что позволило, во-первых, показать применяемость разработанных критериев, во-вторых, локализовать ограничения в современных решениях. На основе проведенного анализа приведен список долгосрочных доработок в функционале тренировочных окружений, исправление которых позволит ускорить их использование в корпоративных и пользовательских целях.

Проведенный эксперимент по разработке и внедрению собственного сценария является важным этапом для демонстрации важности наличия функционала добавления сценария. Продемонстрировано, что указанный критерий является неотъемлемой и важной частью тренировочных окружений. Найдены и локализованы ограничения в работе тренировочного окружения CybORG, на основе которых можно сформировать краткосрочные доработки для улучшения опыта использования тренировочного окружения CybORG.

В разговоре о функциональной возможности использования приведенных платформ (о сложном пороге входа в исследование тренировочных окружений) можно утверждать, что указанные продукты имеют практическую ценность в персональных целях. Однако вдумчивый анализ тренировочных окружений показывает следующее: нельзя достичь такого уровня применения, как у классических моделей ML-подхода, так как последний может быть обучен на большом количестве данных с применением глубокого обучения, добиваясь возможности решать различные повседневные задачи. В случае с RL такой подход требует гораздо больших усилий, а именно построения реалистичной среды и последующего обучения агентов. Однако указанные пункты являются очень трудоемкими.

Обсуждение

В настоящем разделе уделяется внимание направлению дальнейших улучшений в работе путем указания недостатков и ограничений в выполненной работе.

Созданный сценарий для тренировочного окружения показал пригодность платформы CybORG для работы с ней и наличествующие в ней ограничения. Создание сценария

Вестник Российского нового университета
Серия «Сложные системы: модели, анализ и управление», выпуск 2 за 2025 год

было сопряжено с изменением в коде платформы, что привело к добавлению минорных исправлений в продукт. Соответственно, только из разработки сценария нельзя внести изменения в работу самого тренировочного окружения и получить, например, визуализацию результатов обучения.

Выполнение анализа сопряжено с теоретической основой, поскольку только часть тренировочных окружений имеет репозиторий с открытым кодом. Из них на работоспособность и заявленные функции проверен CybORG, проведена работа с тренировочным окружением.

К вектору направления дальнейшей работы относятся:

- формирование точной и всесторонней документации;
- тщательная проработка функционала тренировочных окружений;
- исправления передачи данных из сценария в обучение;
- наличие интуитивности в использовании.

Обозначенные доработки можно результативно внедрить, поскольку они не затрагивают код тренировочного окружения, что сделает процесс внедрения исправлений более легким, а их разработку – динамичной, не связанной с рефакторингом всего кода платформы.

Заключение

Реализовано исследование АИЗ и ее значимого аспекта – АСТО, функционал которого разобран с помощью аналитического метода. Путем использования выделенных критериев проанализированы следующие тренировочные окружения: Farland [2], NASimEmu [3], Yawning Titan [4], CybORG [5] и CAGE Challenge [6]. В процессе анализа выявлены недостатки, которые не позволяют эффективно пользоваться данными разработками. Определены конкретные характеристики тренировочных окружений: создание сценария, полнота документации, соответствие документации, реалистичность окружения и наличие эмуляции. В качестве практического исследования проведена разработка и внедрение собственного сценария в тренировочное окружение CybORG [5]. В процессе работы внесены изменения в основной код проекта. Практическая работа показала эффективность тренировочных окружений, предложенная модель может выступить результативным инструментом для определения возможных путей атаки.

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. *Molina-Markham A., Miniter C., Becky P., Ridley A.* Network environment design for autonomous cyberdefense //arXiv. 2021. DOI: https://doi.org/10.48550/arXiv.2103.07583

3. *Janisch J., Pevný T., Lisý V.* NASimEmu: Network attack simulator & emulator for training agents generalizing to novel scenarios // arXiv. 2023. DOI: https://doi.org/10.48550/arXiv.2305.17246

4. *Andrew A., Spillard S., Collyer J., Dhir N.* Developing optimal causal cyber-defence agents via cyber security simulation //arXiv. 2022. DOI: https://arxiv.org/abs/2207.12355

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.2207.12355

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. *Oesch S., Austria P., Chaulagain A., Weber B., Watson C., Dixson M., Sadovnik A.* The Path to Autonomous Cyberdefense // arXiv. 2024. DOI: https://doi.org/10.48550/arXiv.2404.10788
8. *Nguyen T.T., Reddi V.J.* Deep reinforcement learning for cyber security // IEEE Transactions on Neural Networks and Learning Systems. 2021. Vol. 34. No. 8. P. 3779–3795. DOI: 10.1109/TNNLS.2021.3121870
9. *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
10. *Hammar K., Stadler R.* Finding effective security strategies through reinforcement learning and self-play // 2020 16th International Conference on Network and Service Management (CNSM). Izmir, Turkey, 2020. Pp. 1–9. DOI: 10.23919/CNSM50824.2020.9269092
11. *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
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. *Oesch S., Austria P., Chaulagain A., Weber B., Watson C., et al.* The Path to Autonomous Cyberdefense // IEEE Security & Privacy. Jan.-Feb. 2025. Vol. 23. No. 1. Pp. 38–46. DOI: 10.1109/MSEC.2024.3427640
14. *Brockman G., Cheung V., Pettersson L., Schneider J., Schulman J., Tang J., Zaremba W.* OpenAI Gym // *arXiv: Learning*. 2016. DOI: https://doi.org/10.48550/arXiv.1606.01540

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. Molina-Markham A., Miniter C., Becky P., Ridley A. (2021) Network environment design for autonomous cyberdefense. *arXiv.* DOI: https://doi.org/10.48550/arXiv.2103.07583
3. Janisch J., Pevný T., Lisý V. (2023) NASimEmu: Network attack simulator & emulator for training agents generalizing to novel scenarios. *arXiv.* DOI: https://doi.org/10.48550/arXiv.2305.17246
4. Andrew A., Spillard S., Collyer J., Dhir N. (2022) Developing optimal causal cyber-defence agents via cyber security simulation. *arXiv.* DOI: https://arxiv.org/abs/2207.12355
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.2207.12355
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. Oesch S., Austria P., Chaulagain A., Weber B., Watson C., Dixson M., Sadovnik A. (2024) The Path to Autonomous Cyberdefense. *arXiv.* DOI: https://doi.org/10.48550/arXiv.2404.10788
8. Nguyen T.T., Reddi V.J. (2021) Deep reinforcement learning for cyber security. In: *IEEE Transactions on Neural Networks and Learning Systems.* Vol. 34. No. 8. Pp. 3779–3795. DOI: 10.1109/TNNLS.2021.3121870
9. 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

Вестник Российского нового университета
Серия «Сложные системы: модели, анализ и управление», выпуск 2 за 2025 год

10. Hammar K., Stadler R. (2020) Finding effective security strategies through reinforcement learning and self-play. In: *2020 16th International Conference on Network and Service Management (CNSM)*. Izmir, Turkey, 2020. Pp. 1–9. DOI: 10.23919/CNSM50824.2020.9269092
11. Oakley L., OpreaA. (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
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: *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. Oesch S., Austria P., Chaulagain A., Weber B., Watson C., et al. (2025) The Path to Autonomous Cyberdefense. In: *IEEE Security & Privacy.* Jan.-Feb. Vol. 23. No. 1. Pp. 38–46. DOI: 10.1109/MSEC.2024.3427640The Path to Autonomous Cyberdefense
14. Brockman G., Cheung V., Pettersson L., Schneider J., Schulman J., Tang J., Zaremba W. (2016). OpenAI Gym. *arXiv: Learning*. DOI: https://doi.org/10.48550/arXiv.1606.01540

Поступила в редакцию: 28.04.2025
Поступила после рецензирования: 29.05.2025
Принята к публикации: 09.06.2025

Received: 28.04.2025
Revised: 29.05.2025
Accepted: 09.06.2025