

Методика тестирования на проникновение с использованием технологии искусственного интеллекта

А. Р. Порошина¹, А. В. Обухов²

Санкт-Петербургский государственный электротехнический университет
«ЛЭТИ» им. В.И. Ульянова (Ленина)

poroshina.alina02@mail.ru, aleks.obuhov@yandex.ru.

Аннотация. В статье рассматривается проблема повышения эффективности тестирования на проникновение в условиях растущей сложности киберугроз. Основное внимание уделяется интеграции технологий искусственного интеллекта (ИИ) в процессы проведения пентестов. Проведён анализ традиционных методов тестирования, а также соответствующих международных стандартов (OWASP, NIST, PTES). Выявлены ограничения ручных подходов и обоснована целесообразность применения ИИ для автоматизации сбора информации, анализа уязвимостей, моделирования атак и формирования отчётности. В результате исследования предложена методика интеллектуального тестирования на проникновение, включающая построение базы знаний, сопоставление уязвимостей с техниками MITRE ATT&CK и генерацию сценариев атак на основе входных параметров. Методика обеспечивает повышение точности обнаружения уязвимостей, ускорение анализа и адаптацию под современные ИТ-ландшафты, что подтверждает её актуальность и практическую значимость для задач информационной безопасности.

Ключевые слова: тестирование на проникновение; искусственный интеллект; кибербезопасность; уязвимости; пентест; OWASP; NIST; PTES; машинное обучение.

I. ВВЕДЕНИЕ

Рост числа целенаправленных кибератак и усложнение используемых злоумышленниками техник обуславливают необходимость применения более адаптивных и интеллектуальных подходов к обеспечению информационной безопасности [1]. Тестирование на проникновение является одним из ключевых инструментов для выявления уязвимостей до их возможной эксплуатации. Однако классические методы пентеста, основанные на ручном анализе и фиксированных сценариях, не обеспечивают достаточную скорость, масштабируемость и актуальность в условиях динамично изменяющейся ИТ-инфраструктуры.

Современные технологии искусственного интеллекта позволяют автоматизировать отдельные этапы тестирования, включая анализ конфигураций, обнаружение уязвимостей, сопоставление с базами данных атак и генерацию сценариев эксплуатации. Это позволяет значительно повысить точность тестов, сократить временные затраты и адаптироваться к новым типам угроз.

Цель данной работы — разработка и обоснование методики тестирования на проникновение с использованием ИИ. В ходе исследования проведён сравнительный анализ существующих методов и

стандартов пентеста, определены сущности для построения базы знаний, а также реализована структура применения ИИ на всех этапах тестирования.

II. АНАЛИЗ СОВРЕМЕННЫХ МЕТОДОВ ТЕСТИРОВАНИЯ НА ПРОНИКНОВЕНИЕ

В условиях стремительного роста числа кибератак и усложнения ИТ-инфраструктур тестирование на проникновение становится необходимым инструментом обеспечения кибербезопасности. Анализ существующих методов показал, что основными подходами являются тестирование по модели «черного ящика» (Black Box), «белого ящика» (White Box) и «серого ящика» (Gray Box). Каждый метод имеет свои сильные и слабые стороны, определяемые степенью осведомлённости тестирующего о целевой системе.

Также были рассмотрены три широко применяемых стандарта [2]:

- OWASP — фокусируется на уязвимостях веб-приложений [9];
- PTES — ориентирован на комплексное выполнение пентестов [10];
- NIST SP 800-115 – предлагает формализованный процесс тестирования с акцентом на стандартизацию и отчётность [11].

Несмотря на распространённость этих подходов, они часто страдают от ограничений в плане масштабируемости, скорости анализа и адаптивности к новым видам угроз. Это особенно критично в условиях постоянно меняющегося цифрового ландшафта.

III. РОЛЬ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В УЛУЧШЕНИИ ПРОЦЕССОВ ПЕНТЕСТА

Проведённый анализ подтвердил, что применение технологий искусственного интеллекта (ИИ) позволяет существенно повысить эффективность тестирования на проникновение. ИИ способен [3]:

- автоматизировать сбор информации и разведку (OSINT),
- проводить интеллектуальный анализ уязвимостей с сопоставлением по CVE, CWE и MITRE ATT&CK,
- генерировать адаптивные сценарии атак,
- выполнять категоризацию угроз и приоритизацию рисков [4],
- формировать отчёты с учётом разных целевых аудиторий.

ИИ реализует цепочку принятия решений и действий для идентификации конкретных уязвимостей на основе следующей схемы [5].



Рис. 1. Цепочка решений

Использование NLP-моделей, графов знаний и ML-классификаторов позволит сформировать интеллектуальную базу уязвимостей и методов эксплуатации, что открывает возможности для семантического поиска и автоматического подбора техник.

IV. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ

Пусть:

$V = \{v_1, v_2, \dots, v_m\}$ – множество уязвимостей, выявленных в ходе теста.

$C = \{c_1, c_2, \dots, c_k\}$ – множество классов уязвимостей, соответствующих CWE.

$A = \{a_1, a_2, \dots, a_l\}$ – множество техник атак {MITRE ATT&CK}.

$M = \{m_1, m_2, \dots, m_p\}$ – множество методов тестирования.

$E = \{e_1, e_2, \dots, e_q\}$ – множество эксплойтов или атакующих скриптов.

$S = \{s_1, s_2, \dots, s_r\}$ – множество сценариев атак, сгенерированных ИИ.

D – входной датасет: конфигурационные данные, лог-файлы, сетевые пакеты и т.д.

1. Модель сопоставления уязвимости и техники атаки

Пусть функция $\phi: V \rightarrow C$ отображает уязвимость на соответствующий класс CWE, а $\psi: C \rightarrow A$ отображает класс CWE на соответствующую технику MITRE ATT&CK.

Тогда составляющая функция:

$\theta = \psi \cdot \phi: V \rightarrow A$ позволяет сопоставить каждой уязвимости наиболее вероятную технику атаки [6].

2. Построение графа атак

Интеллектуальная система формирует направленный граф $G=(N,E)$, где:

- N – множество вершин, представляющих уязвимости, техники, цели, действия;
- $E \subseteq N \times N$ – множество ребер, соответствующих переходам между фазами атаки.

Вес ребер ω_{ij} – оценка вероятности успешного перехода между узлами $n_i \rightarrow n_j$, вычисляется на основе данных модели:

$$\omega_{ij} = P(n_j | n_i, D)$$

Оптимальный путь атаки:

$$\operatorname{argmax}_{P \subseteq G} \sum_{(i,j) \in P} \omega_{ij}$$

3. Генерация сценариев атак

Функция генерации сценария:

$$g: (V, A, M, D) \rightarrow S$$

Использует входные параметры для синтеза атакующего сценария $s \in S$, основанного на правилах и эвристиках модели ИИ.

V. ПРЕДЛОЖЕННАЯ МЕТОДИКА ТЕСТИРОВАНИЯ НА ПРОНИКНОВЕНИЕ С ИИ

Предложенная методика тестирования на проникновение с применением технологий искусственного интеллекта представляет собой последовательный, модульный процесс, обеспечивающий автоматизацию всех ключевых этапов пентеста с возможностью адаптации под различные сценарии и цели.

- Сбор требований и построение модели угроз

На начальном этапе ИИ получает входные параметры: тип тестируемой системы (веб-приложение, API, IoT, SCADA и др.), критичность активов, уровень доступа (Black/Gray/White Box), ограничения (например, запрет на DoS) и цель (поиск уязвимостей, Red Team и т.п.). Эти данные используются для построения предварительной модели угроз с учётом актуальных TTP (tactics, techniques, and procedures) из базы MITRE ATT&CK. ИИ с помощью NLP-моделей выделяет ключевые векторы риска и предлагает список необходимых разведывательных действий.

- Разведка и сбор информации

ИИ-агенты осуществляют активный и пассивный сбор данных. Используются инструменты (Nmap, Shodan, Gobuster, Wappalizer), а также LLM-модели (например, GPT) для анализа баннеров, WHOIS-информации, индексированных страниц и API-документации. ИИ классифицирует полученные данные по технологиям, версиям ПО и конфигурациям, формируя карту цифровой поверхности атаки.

- Выявление уязвимостей и их категоризация

ИИ анализирует выходные данные сканеров (OpenVAS, Nikto, Nessus) с помощью NLP и классификаторов на основе CVE и CWE. Применяется автоматическое сопоставление версий ПО с базами уязвимостей, определение типа уязвимости (SQLi, XSS, RCE и т.д.) и ранжирование по критичности (CVSS). Внутренние алгоритмы оценивают контекст (сетевой, привилегии, экспонированность) и определяют вероятность эксплуатации.

- Построение сценария атаки

ИИ строит граф атаки, в котором каждому CWE сопоставляется MITRE ATT&CK техника и подходящий инструмент (SQLmap, Burp Suite, Metasploit). Используется Neo4j или аналогичный движок графов. Сценарий включает входные данные, шаги эксплуатации, предполагаемые точки закрепления и цели постэксплуатации. LLM может генерировать команды и payload'ы с учетом специфики целевой системы [12].

- Этап атаки и постэксплуатации
- ИИ выполняет атаку согласно построенному сценарию. В случае успеха проводится закрепление, анализ прав и попытки lateral movement. Применяются инструменты BloodHound, mimikatz, linPEAS с последующим ML-анализом прав доступа, отношений между узлами и возможностей привилегий.
- Генерация отчёта и адаптация под аудиторию

ИИ агрегирует данные, структурирует их по шаблонам (OWASP, NIST, ISO/IEC), формирует резюме для управленцев, технический отчёт для специалистов и соблюдение требований для аудиторов. Включается карта атак, подробности PoC, CVE, рекомендации и визуализация путей эксплуатации.

Исходя из всего перечисленного был определен инструментарий, который можно использовать для выявления уязвимостей [7].

Инструменты														
Nmap	Masscan	UnicornsCan	Snort	Bettercap	Suricata	Responder	Nbtscan							
Netcat	NSE	OSSEC	Nagios	Arpwatch	Zeek	dnstop	Hping3	RouterSploit	OpenVAS					
Wireshark	Scapy	Metasploit	XArp	Ettercap	DNSChuf	Nessus	tcpdump							
sqlmap	Burp Suite	OWASP ZAP	Nuclei	AuthMatrix	ParamSpider	ExifTool	ClamAV							
XSSStrike	Dalfox	Hydra	Postman	JWT Tool	Nuclei	ysoserial	WhatWeb							
XSSHunter	CSRFPocGenerator	ffuf	Smuggler	marshalsec	UploadScanner									
Auditd	Lynis	CIS-CAT Pro	LinPEAS	Linux Exploit Suggester	ClamAV									
OpenSCAP	Open VAS	PowerUP	Autoruns	Commix	rkhunter	CrowdStrike								
Nessus	Vulners	Wazuh	chkconfig	BurpSuite	Metasploit	chkrootkit								
LinPEAS	WinPEAS	PowerUP	Incognito											
BeRoot	BloodHound	find	Mimikatz	SharpSploit										
AccessChk	SetACL	GTFOBins	PowerView											
Gophish	King Phisher	SET	SPF/DKIM tester	OSINT Framework										
SendEmail	CallerID Spoofing tools	LinkedIn	Flipper Zero											
Metasploit	Bettercap	SMiShing	Maltego	Bash Bunny										

Рис. 2. Инструментарий

Заметим, что одной из ключевых задач при автоматизации тестирования на проникновение с использованием ИИ является формализация и структурирование информации об уязвимостях, техниках эксплуатации и методах тестирования. Это достигается за счёт построения интеллектуальной базы знаний, в которой семантически связаны. Были выделены 6 сущностей:

- VULNERABILITY — CWE/CVE-объекты;
- ATTACK_TECHNIQUE — MITRE ATT&CK техника;
- TEST_METHOD — метод тестирования (например, fuzzing, SAST, dynamic testing);
- EXPLOIT_SCRIPT — существующий эксплойт;

- SECURITY_CONTROL — защита (WAF, IDS, sandboxing);
- ENVIRONMENT — контекст исполнения (OS, network topology, privileges).

Ниже представлены предполагаемые типы сущностей.

VULNERABILITY	ATTACK_TECHNIQUE	TEST_METHOD
ID Название Описание Тип уязвимости Уровень риска CVSS балл Дата обнаружения Источники Связанные атаки Связанные эксплойты Патч-доступность	ID Название техники Описание Тактика Связанные уязвимости Используемые инструменты Индикаторы компроментации Тип цели	Название метода Описание Тип анализа Инструменты Покрываемые уязвимости Требования к окружению Скорость/масштабируемость Интеграция CI/CD
EXPLOIT_SCRIPT	SECURITY_CONTROL	ENVIRONMENT
ID Целевая уязвимость Язык реализации Описание функционала Тип эксплуатации Предпосылки Последствия Источник Метод доставки Обход защит	Название Тип контроля Описание Метод обнаружения Покрываемые атаки/уязвимости Примеры инструментов Интеграция с SIEM Обходимость	ОС Тип устройства Сетевой контекст Привилегии Наличие защит Установленные приложения Тип архитектуры Изоляция Конфигурация

Рис. 3. Типы сущностей

Предложенная методика отличается высокой степенью адаптивности, позволяет автоматизировать повторяющиеся этапы, снижает влияние человеческого фактора и обеспечивает воспроизводимость результатов в рамках различных стандартов и моделей угроз.

VI. ЗАКЛЮЧЕНИЕ

В ходе исследования была решена задача повышения эффективности тестирования на проникновение за счёт внедрения технологий искусственного интеллекта. Для этого:

1. Проведён анализ существующих методов пентеста (черный, серый, белый ящик) и применяемых стандартов (OWASP, NIST, PTES).
2. Разработана модель интеллектуального тестирования, включающая:
 - автоматическое сопоставление уязвимостей с CWE и MITRE ATT&CK;
 - построение графа атак и подбор соответствующих методов тестирования;
 - генерацию кастомизированных сценариев атак с учётом цели, контекста, ограничений и уровня доступа;
 - автоматическую категоризацию и ранжирование уязвимостей по критичности;
 - формирование отчета по итогам теста, адаптированного под разные целевые аудитории.
3. Представлена математическая модель, формализующая работу ИИ в рамках пентеста, включая выбор техники, построение сценариев и оптимизацию последовательности действий.
4. Предложена структура базы знаний на основе связей CWE ↔ GPT-тегов ↔ MITRE ATT&CK, необходимая для обучения и работы ИИ-агентов [8].

Таким образом, предложенная методика позволяет автоматизировать ключевые этапы пентеста, обеспечить воспроизводимость сценариев, снизить влияние человеческого фактора и ускорить процесс принятия решений при обнаружении уязвимостей. Методика применима как в корпоративной среде, так и в рамках аудитов информационной безопасности при ограниченных ресурсах. В дальнейшем требуется рассмотреть более детально все аспекты и проверить на практике.

СПИСОК ЛИТЕРАТУРЫ

- [1] The Rise of Cybercrime and Cyber-Threat Intelligence: Perspectives and Challenges From Law Enforcement. URL: <https://research.tue.nl/en/publications/the-rise-of-cybercrime-and-cyber-threat-intelligence-perspectives> (дата обращения 04.06.2025)
- [2] 5 Most Popular Penetration Testing Methodologies and Standards. URL: <https://zerothreat.ai/blog/top-penetration-testing-methodologies> (дата обращения 06.06.2025)
- [3] Building an AI-Driven Penetration Testing Tool. - DEV Community. URL: <https://dev.to/profm0r1arty/building-an-ai-driven-penetration-testing-tool-1o2n> (дата обращения 09.06.2025)
- [4] (PDF) Generative AI for pentesting: the good, the bad, the ugly. URL: https://www.researchgate.net/publication/378995026_Generative_AI_for_pentesting_the_good_the_bad_the_ugly (дата обращения 09.06.2025)
- [5] Hacking, The Lazy Way: LLM Augmented Pentesting. URL: <https://arxiv.org/html/2409.09493v2> (дата обращения 13.06.2025)
- [6] Canstralian/pentest_ai. URL: https://huggingface.co/Canstralian/pentest_ai (дата обращения 13.06.2025)
- [7] GitHub - arch3rPro/PentestTools: Awesome Pentest Tools Collection. URL: <https://github.com/arch3rPro/PentestTools> (дата обращения 14.06.2025)
- [8] GitHub - vishwanathakuthota/Pentest-AI: Pentest AI is. URL: <https://github.com/vishwanathakuthota/Pentest-AI> (дата обращения 06.06.2025)
- [9] OWASP Foundation, the Open Source Foundation for Application. URL: <https://owasp.org/> (дата обращения 06.06.2025)
- [10] The Penetration Testing Execution Standard. URL: http://www.pentest-standard.org/index.php/Main_Page (дата обращения 06.06.2025)
- [11] National Institute of Standards and Technology. URL: <https://www.nist.gov/> (дата обращения 06.06.2025)
- [12] GitHub - mrwadams/attackgen: AttackGen is a cybersecurity incident. URL: <https://github.com/mrwadams/attackgen> (дата обращения 15.06.2025)