# Novelty Assessment Report

**Paper**: CyberGym: Evaluating AI Agents' Real-World Cybersecurity Capabilities at Scale
**PDF URL**: https://openreview.net/pdf?id=2YvbLQEdYt
**Venue**: ICLR 2026 Conference Submission
**Year**: 2026
**Report Generated**: 2025-12-30

## Abstract

AI agents have significant potential to reshape cybersecurity, making a thorough assessment of their capabilities critical. However, existing evaluations fall short, because they are based on small-scale benchmarks and only measure static outcomes, failing to capture the full, dynamic range of real-world security challenges. To address these limitations, we introduce CyberGym, a large-scale benchmark featuring 1,507 real-world vulnerabilities across 188 software projects. Adjustable to different vulnerability analysis settings, CyberGym primarily tasks agents with generating a proof-of-concept test that reproduces a vulnerability, given only its text description and the corresponding codebase. Our extensive evaluation highlights that CyberGym effectively differentiates agents' and models' cybersecurity capabilities. Even the top-performing combinations only achieve a ~20% success rate, demonstrating the overall difficulty of CyberGym. Beyond static benchmarking, we show that CyberGym leads to the discovery of 35 zero-day vulnerabilities and 17 historically incomplete patches. These results underscore that CyberGym is not only a robust benchmark for measuring AI's progress in cybersecurity but also a platform for creating direct, real-world security impact.



## Core Task Landscape

This paper addresses: **Generating Proof-of-Concept Tests for Vulnerability Reproduction in Software Codebases**
A total of **50 papers** were analyzed and organized into a taxonomy with **11 categories**.

### Taxonomy Overview

The research landscape has been organized into the following main categories:

- **Automated PoC Generation Techniques**
- **Vulnerability Detection and Validation**
- **Benchmarking and Evaluation Frameworks**
- **Vulnerability Comprehension and Exploitation Analysis**

### Complete Taxonomy Tree

- Generating Proof-of-Concept Tests for Vulnerability Reproduction in Software Codebases Survey Taxonomy
- Automated PoC Generation Techniques
  - LLM-Driven PoC Synthesis ★ (5 papers)
  - [0] CyberGym: Evaluating AI Agents' Real-World Cybersecurity Capabilities at Scale (Anon et al., 2026) View paper
  - [1] PoCGen: Generating Proof-of-Concept Exploits for Vulnerabilities in Npm Packages (Simsek Deniz, 2025) View paper
  - [3] PoCo: Agentic Proof-of-Concept Exploit Generation for Smart Contracts (Bobadilla, 2025) View paper
  - [17] LLM Agents for Automated Web Vulnerability Reproduction: Are We There Yet? (Liu Bin, 2025) View paper
  - [39] A Systematic Study on Generating Web Vulnerability Proof-of-Concepts Using Large Language Models (Zhao Mengyao, 2025) View paper
  - Dynamic Analysis and Test-Guided PoC Generation (4 papers)
  - [8] Test Suites Guided Vulnerability Validation for Node. js Applications (Changhua Luo, 2024) View paper
  - [23] Test mimicry to assess the exploitability of library vulnerabilities (Hong Jin Kang, 2022) View paper
  - [29] Automated Exploit Generation for Node.js Packages (Filipe Marques, 2025) View paper
  - [36] Octopocs: automatic verification of propagated vulnerable code using reformed proofs of concept (Seongkyeong Kwon, 2021) View paper
  - Constraint-Based and Symbolic PoC Synthesis (3 papers)
  - [37] Semfuzz: Semantics-based automatic generation of proof-of-concept exploits (Wei You, 2017) View paper
  - [41] Bug synthesis: Challenging bug-finding tools with deep faults (Subhajit Roy, 2018) View paper
  - [42] Automated test generation from vulnerability signatures (Abdulbaki Aydin, 2014) View paper
- Vulnerability Detection and Validation
  - Fuzzing and Automated Test Generation (6 papers)
  - [12] FuSeBMC: An Energy-Efficient Test Generator for Finding Security Vulnerabilities in C Programs (Kaled M. Alshmrany, 2021) View paper
  - [20] A generative and mutational approach for synthesizing bug-exposing test cases to guide compiler fuzzing (Guixin Ye, 2023) View paper
  - [27] Minerva: browser API fuzzing with dynamic mod-ref analysis (Chijin Zhou, 2022) View paper
  - [32] Intelligen: Automatic driver synthesis for fuzz testing (Mingrui Zhang, 2021) View paper
  - [35] SP-Fuzz: Fuzzing Soft PLC with Semi-automated Harness Synthesis (Seungho Jeon, 2023) View paper
  - [50] Enhancing Software Vulnerability Detection Through Adaptive Test Input Generation Using Genetic Algorithm (Mehendran, 2025) View paper
  - Static Analysis and Vulnerability Pattern Detection (4 papers)

- [5] Eradicating the unseen: Detecting, exploiting, and remediating a path traversal vulnerability across github (Jafar Akhoundali, 2025) View paper
- [7] Untrustide: Exploiting weaknesses in vs code extensions (Elizabeth Lin, 2024) View paper
- [15] Systematic generation of XSS and SQLi vulnerabilities in PHP as test cases for static code analysis (Felix Schuckert, 2022) View paper
- [25] Secure coding practice in Java: Automatic detection, repair, and vulnerability demonstration (Ying, 2023) View paper
- Specialized Domain Vulnerability Analysis (9 papers)
- [11] Unveiling Security Vulnerabilities in Git Large File Storage Protocol (Yuan Chen, 2025) View paper
- [16] A Friend's Eye is A Good Mirror: Synthesizing {MCU} Peripheral Models from Peripheral Drivers (C Lei, 2024) View paper
- [18] Bullseye: Detecting prototype pollution in npm packages with proof of concept exploits (T Houis, 2026) View paper
- [26] {IvySyn}: Automated vulnerability discovery in deep learning frameworks (Christou, 2023) View paper
- [31] Cybersecurity vulnerability identification in system-of-systems using model-based testing (May Myat Thwe, 2022) View paper
- [44] Demo: Backdoor Through the Front Door: Demonstrating Security Flaws in the Eufy Ecosystem (Victor Goeman, 2024) View paper
- [45] Automatic creation of SQL injection and cross-site scripting attacks (Adam Kieyzun, 2009) View paper
- [48] Detecting Prototype Pollution in NPM Packages with Proof of Concept Exploits (Houis, 2025) View paper
- [49] Guided differential testing of certificate validation in SSL/TLS implementations (Yuting Chen, 2015) View paper

- Benchmarking and Evaluation Frameworks
  - AI Agent and LLM Security Benchmarks (5 papers)
  - [4] Real-Time AI Code Security Auditing: Automated Vulnerability Detection and Remediation Through Meta-Experimental Analysis (Vaddiparthy, 2025) View paper
  - [6] LLM-Driven, Self-Improving Framework for Security Test Automation: Leveraging Karate DSL for Augmented API Resilience (Daniela Delinschi, 2025) View paper
  - [10] CyberGym: Evaluating AI Agents' Cybersecurity Capabilities with Real-World Vulnerabilities at Scale (Z Wang, 2025) View paper
  - [13] SEC-bench: Automated Benchmarking of LLM Agents on Real-World Software Security Tasks (Lee Hwi-Won, 2025) View paper
  - [14] SecureAgentBench: Benchmarking Secure Code Generation under Realistic Vulnerability Scenarios (Chen Junkai, 2025) View paper
  - Vulnerability Databases and PoC Repositories (5 papers)
  - [19] Vulnerability and Attack Repository for IoT: Addressing Challenges and Opportunities in Internet of Things Vulnerability Databases (Anna Felkner, 2024) View paper
  - [30] Detecting Fake Proof-of-Concept Codes on GitHub Using Static Code Analysis (K Kita, 2025) View paper
  - [34] Understanding the reproducibility of crowd-reported security vulnerabilities (Dongliang Mu, 2018) View paper
  - [43] N-day Vulnerabilities: Detection, Bisection, and Measurement (Zheng, 2025) View paper
  - [46] Beyond the Surface: Investigating Malicious CVE Proof of Concept Exploits on GitHub (Yadmani, 2022) View paper
- Vulnerability Comprehension and Exploitation Analysis
  - Exploit Development and Attack Mechanisms (4 papers)
  - [21] Understanding and preventing open-source software supply chain attacks (Ladisa, 2024) View paper
  - [22] Counterfeit object-oriented programming: On the difficulty of preventing code reuse attacks in C++ applications (FÃ©lix Schuster, 2015) View paper
  - [40] A Study on Exploit Development (Rosy Chadha, 2022) View paper
  - [47] An Automated Identification Method for Controllable Memory-Related Fields in Proof-of-Concept Code (Fangbo Qin, 2025) View paper
  - Vulnerability Visualization and Comprehension Tools (4 papers)
  - [9] Visualizing Security Vulnerability Evolution of Software Systems (Sinhabahu, 2021) View paper
  - [24] Recycling test cases to detect security vulnerabilities (JoÃ£o Antunes, 2012) View paper
  - [28] Writing secure code (Lester E. Nichols, 2003) View paper
  - [33] Secure CodeCity A Framework For Security Vulnerability Visualization (A Abeysinghe, 2021) View paper
  - AI-Generated Code Security Assessment (2 papers)
  - [2] Is github's copilot as bad as humans at introducing vulnerabilities in code? (Owura Asare, 2023) View paper
  - [38] Modernization of a legacy codebase (Ala-Hulkko, 2025) View paper

### Narrative

Core task: generating proof-of-concept tests for vulnerability reproduction in software codebases. The field organizes around four main branches that reflect distinct stages and concerns in the vulnerability lifecycle. Automated PoC Generation Techniques encompasses methods for synthesizing exploits, ranging from traditional symbolic execution and fuzzing approaches to newer LLM-driven synthesis strategies that leverage large language models to produce test cases from vulnerability descriptions. Vulnerability Detection and Validation focuses on identifying security flaws and confirming their exploitability, including static analysis, dynamic testing, and hybrid techniques. Benchmarking and Evaluation Frameworks provides standardized datasets and metrics to assess PoC generation tools, such as SEC-bench[13] and SecureAgentBench[14], which enable systematic comparison across methods. Finally, Vulnerability Comprehension and Exploitation Analysis examines how developers and attackers understand and weaponize vulnerabilities, studying exploit development workflows and the semantics of security flaws.

Recent work has seen a surge in LLM-driven approaches that promise to automate PoC creation at scale, yet these methods face challenges in balancing generality with precision. CyberGym[0] sits within the LLM-Driven PoC Synthesis cluster, employing reinforcement learning to guide language models in generating executable exploits for diverse vulnerability types. It contrasts with PoCGen[1] and PoCo[3], which also leverage LLMs but differ in their use of retrieval-augmented generation versus iterative refinement strategies. Nearby efforts like Web Vulnerability Reproduction[17] and Web PoC Generation[39] focus specifically on web application contexts, highlighting domain-specific challenges in input crafting and environment setup. A key tension across these branches is the trade-off between automation and accuracy: while LLM-based tools can rapidly produce candidate PoCs, validation remains difficult without robust execution environments and feedback mechanisms, a gap that CyberGym[0] addresses through its reinforcement learning framework.

## Related Works in Same Category

The following **4 sibling papers** share the same taxonomy leaf node with the original paper:

### 1. PoCGen: Generating Proof-of-Concept Exploits for Vulnerabilities in Npm Packages

**Authors**: Simsek Deniz, Eghbali, Aryaz, Deniz Simsek, Pradel, et al. (8 authors total) | **Year/Venue**: 2025 | **URL**: View paper

#### Abstract

Security vulnerabilities in software packages are a significant concern for developers and users alike. Patching these vulnerabilities in a timely manner is crucial to restoring the integrity and security of software systems. However, previous work has shown that vulnerability reports often lack proof-of-concept (PoC) exploits, which are essential for fixing the vulnerability, testing patches, and avoiding regressions. Creating a PoC exploit is challenging because vulnerability reports are infor...

#### Relationship Analysis

Both papers belong to the LLM-Driven PoC Synthesis category, leveraging large language models to generate proof-of-concept exploits from vulnerability descriptions. They overlap in using LLMs to understand natural language vulnerability reports and automatically generate executable PoC tests, both validating outputs through execution-based mechanisms. However, CyberGym is a large-scale benchmark (1,507 instances across 188 projects) focused on evaluating AI agents' capabilities in reproducing memory safety vulnerabilities in C/C++ codebases, while PoCGen is a specific approach combining LLMs with static/dynamic analysis to generate exploits for JavaScript/npm package vulnerabilities (path traversal, prototype pollution, command injection, code injection, ReDoS).

---

### 2. PoCo: Agentic Proof-of-Concept Exploit Generation for Smart Contracts

**Authors**: Bobadilla, Sofia, Vivi Andersson, Sofia Bobadilla, Monperrus, et al. (8 authors total) | **Year/Venue**: 2025 | **URL**: View paper

#### Abstract

Smart contracts operate in a highly adversarial environment, where vulnerabilities can lead to substantial financial losses. Thus, smart contracts are subject to security audits. In auditing, proof-of-concept (PoC) exploits play a critical role by demonstrating to the stakeholders that the reported vulnerabilities are genuine, reproducible, and actionable. However, manually creating PoCs is time-consuming, error-prone, and often constrained by tight audit schedules. We introduce POCO, an agentic...

#### Relationship Analysis

Both papers belong to the LLM-Driven PoC Synthesis category, leveraging large language models to automatically generate proof-of-concept exploits from natural language vulnerability descriptions. They overlap in their core approach of using LLMs with agentic frameworks (CyberGym uses OpenHands, PoCo uses a ReAct-style loop) to synthesize executable PoCs that reproduce vulnerabilities. However, CyberGym focuses on general software vulnerabilities across 188 diverse C/C++ projects with 1,507 instances and emphasizes large-scale benchmarking and zero-day discovery, while PoCo specifically targets smart contract vulnerabilities in Solidity, uses domain-specific tools (Foundry/Forge), and evaluates on only 23 real-world audit cases with a novel patch-based correctness methodology.

---

### 3. LLM Agents for Automated Web Vulnerability Reproduction: Are We There Yet?

**Authors**: Liu Bin, Zhao Yanjie, Bin Liu, Xu Guoai, Yanjie Zhao, et al. (8 authors total) | **Year/Venue**: 2025 | **URL**: View paper

#### Abstract

Large language model (LLM) agents have demonstrated remarkable capabilities in software engineering and cybersecurity tasks, including code generation, vulnerability discovery, and automated testing. One critical but underexplored application is automated web vulnerability reproduction, which transforms vulnerability reports into working exploits. Although recent advances suggest promising potential, challenges remain in applying LLM agents to real-world web vulnerability reproduction scenarios....

#### Relationship Analysis

Both papers belong to the LLM-Driven PoC Synthesis category, leveraging large language models to generate proof-of-concept exploits from natural language vulnerability descriptions. They share overlapping focus on using LLM agents to automate vulnerability reproduction tasks, with both evaluating state-of-the-art agents like OpenHands and Claude-Sonnet-4 on real-world vulnerabilities. However, CyberGym focuses on memory safety vulnerabilities in C/C++ codebases across 188 diverse projects (1,507 instances) with emphasis on fuzzing-based detection, while the candidate paper specifically targets web application vulnerabilities across 7 types (CSRF, XSS, SQL injection, etc.) in 80 CVEs, emphasizing HTTP-based exploitation and web-specific authentication challenges.

---

### 4. A Systematic Study on Generating Web Vulnerability Proof-of-Concepts Using Large Language Models

**Authors**: Zhao Mengyao, Li Kaixuan, Mengyao Zhao, Zhang, Lyuye, et al. (16 authors total) | **Year/Venue**: 2025 | **URL**: View paper

#### Abstract

Recent advances in Large Language Models (LLMs) have brought remarkable progress in code understanding and reasoning, creating new opportunities and raising new concerns for software security. Among many downstream tasks, generating Proof-of-Concept (PoC) exploits plays a central role in vulnerability reproduction, comprehension, and mitigation. While previous research has focused primarily on zero-day exploitation, the growing availability of rich public information accompanying disclosed CVEs ...

#### Relationship Analysis

Both papers belong to the LLM-Driven PoC Synthesis category, leveraging large language models to generate proof-of-concept exploits from vulnerability information. While CyberGym focuses on evaluating AI agents' capabilities to generate PoCs for diverse software vulnerabilities (1,507 instances across 188 projects) using vulnerability descriptions and codebases, this candidate paper specifically targets web application vulnerabilities and systematically studies how different types of publicly disclosed information (descriptions, patches, vulnerable files) affect PoC generation success across the vulnerability disclosure timeline. The key distinction is that CyberGym emphasizes broad-scale benchmarking and zero-day discovery across general software, whereas the candidate paper provides an in-depth empirical analysis of web-specific vulnerabilities with focus on the progressive disclosure stages and adaptive reasoning strategies.

## Contributions Analysis

**Overall novelty summary.** The paper introduces CyberGym, a large-scale benchmark with 1,507 real-world vulnerabilities across 188 software projects, designed to evaluate AI agents on generating proof-of-concept tests from vulnerability descriptions. It resides in the 'LLM-Driven PoC Synthesis' leaf, which contains five papers total, indicating a moderately populated but still emerging research direction. This leaf sits within the broader 'Automated PoC Generation Techniques' branch, which also includes constraint-based and dynamic analysis approaches, suggesting the field is actively exploring multiple synthesis paradigms.

The taxonomy reveals that CyberGym's neighboring work spans several related directions: dynamic analysis and test-guided PoC generation (four papers), constraint-based symbolic synthesis (three papers), and AI agent security benchmarks (five papers). The 'Benchmarking and Evaluation Frameworks' branch, particularly 'AI Agent and LLM Security Benchmarks,' provides the closest conceptual neighbors, as these frameworks similarly assess AI capabilities on cybersecurity tasks. The taxonomy's scope notes clarify that CyberGym's focus on LLM-driven synthesis distinguishes it from purely symbolic or fuzzing-based methods, while its benchmarking component connects it to evaluation-focused research.

Among 30 candidates examined, the contribution-level analysis shows varied novelty signals. The large-scale benchmark contribution (10 candidates examined, 0 refutable) and the comprehensive AI agent evaluation (10 candidates examined, 0 refutable) appear to have limited direct overlap in the search scope. However, the platform for open-ended vulnerability discovery (10 candidates examined, 1 refutable) shows at least one candidate providing overlapping prior work. This suggests that while the benchmark scale and evaluation methodology may be distinctive, the concept of using AI for real-world vulnerability discovery has some precedent within the examined literature.

Given the limited search scope of 30 semantically similar candidates, this analysis captures the most proximate prior work but cannot claim exhaustive coverage of the field. The taxonomy structure indicates CyberGym operates in a moderately active research area with established neighboring directions, yet the specific combination of large-scale benchmarking, LLM-driven PoC synthesis, and real-world vulnerability discovery appears to differentiate it from the examined candidates. The refutable finding for one contribution warrants closer inspection of the overlapping work's scope and claims.

---

This paper presents **3 main contributions**, each analyzed against relevant prior work:

## Contribution 1: CyberGym: A large-scale, realistic cybersecurity benchmark

**Description**: The authors present CyberGym, a benchmark containing 1,507 real-world vulnerability instances from 188 diverse software projects. The benchmark tasks agents with generating proof-of-concept tests to reproduce vulnerabilities given text descriptions and codebases, using execution-based validation metrics.

This contribution was assessed against **10 related papers** from the literature. Papers with potential prior art are analyzed in detail with textual evidence; others receive brief assessments.

---

### 1. CVE-Bench: Benchmarking LLM-based Software Engineering Agent's Ability to Repair Real-World CVE Vulnerabilities

**URL**: View paper

**Brief Assessment**

CVE-Bench[51] focuses on repairing real-world CVE vulnerabilities in software repositories, not on reproducing vulnerabilities from text descriptions across diverse projects like CyberGym.

---

### 2. Detection of recurring software vulnerabilities

**URL**: View paper

**Brief Assessment**

Recurring Vulnerabilities[55] focuses on detecting recurring vulnerabilities across different software systems through code similarity analysis, not on creating benchmarks for evaluating AI agents' cybersecurity capabilities with proof-of-concept generation tasks.

---

### 3. On Security Weaknesses and Vulnerabilities in Deep Learning Systems

**URL**: View paper

**Brief Assessment**

Deep Learning Vulnerabilities[56] focuses on analyzing vulnerabilities in deep learning frameworks (TensorFlow, Caffe, etc.) through CVE databases, not on creating cybersecurity benchmarks for evaluating AI agents on vulnerability reproduction tasks.

---

### 4. Benchmarking static analysis tools for web security

**URL**: View paper

**Brief Assessment**

Web Security Benchmarking[58] focuses on benchmarking static analysis tools for detecting vulnerabilities in web applications (specifically WordPress plugins), not on creating a large-scale benchmark for evaluating AI agents' cybersecurity capabilities across diverse software projects.

---

### 5. CyberGym: Evaluating AI Agents' Cybersecurity Capabilities with Real-World Vulnerabilities at Scale

**URL**: View paper

**Brief Assessment**

CyberGym Scale[10] appears to be the same work as the original paper (identical authors, title, and content), not a prior work that could refute novelty claims.

---

### 6. SecureAgentBench: Benchmarking Secure Code Generation under Realistic Vulnerability Scenarios

**URL**: View paper

**Brief Assessment**

SecureAgentBench[14] focuses on secure code generation tasks where agents must edit existing codebases to fix vulnerabilities, whereas CyberGym evaluates agents on vulnerability reproduction tasks (generating proof-of-concept tests). The task formulations, evaluation methodologies, and objectives differ fundamentally between these two benchmarks.

---

### 7. A large-scale empirical study on vulnerability distribution within projects and the lessons learned

**URL**: View paper

**Brief Assessment**

Vulnerability Distribution Study[53] focuses on analyzing vulnerability distribution patterns within individual software projects (files, functions, types, developers), not on creating a benchmark for evaluating AI agents' cybersecurity capabilities with execution-based validation.

---

### 8. Large-scale empirical study of important features indicative of discovered vulnerabilities to assess application security

**URL**: View paper

**Brief Assessment**

Vulnerability Features Study[57] focuses on empirical analysis of features correlated with vulnerability abundance across applications, not on creating benchmarks for evaluating AI agents' cybersecurity capabilities with proof-of-concept generation tasks.

---

### 9. CVE-assisted large-scale security bug report dataset construction method

**URL**: View paper

**Brief Assessment**

CVE Dataset Construction[52] focuses on constructing labeled bug report datasets from CVE records, not on creating executable benchmarks with proof-of-concept validation for vulnerability reproduction.

---

#### 10. Cheesecloth: Zero-Knowledge Proofs of Real-World Vulnerabilities

**URL**: View paper

**Brief Assessment**

Cheesecloth[54] focuses on zero-knowledge proofs for vulnerability disclosure, not on creating benchmarks for evaluating AI agents' cybersecurity capabilities across diverse software projects.

---

### Contribution 2: Comprehensive evaluation of frontier AI agents and LLMs on cybersecurity tasks

**Description**: The authors conduct extensive experiments evaluating four state-of-the-art agent frameworks and eleven frontier LLMs on CyberGym. Their evaluation reveals that even top-performing combinations achieve only approximately 20% success rates, demonstrating CyberGym's effectiveness in differentiating agents' cybersecurity capabilities.

This contribution was assessed against **10 related papers** from the literature. Papers with potential prior art are analyzed in detail with textual evidence; others receive brief assessments.

---

#### 1. Assessing confidence in frontier AI safety cases

**URL**: View paper

**Brief Assessment**

Frontier AI Safety[64] focuses on assessing confidence in safety cases for frontier AI systems, not on evaluating AI agents' cybersecurity capabilities through benchmark tasks. The candidate addresses safety assurance methodology rather than performance evaluation on cybersecurity benchmarks.

---

#### 2. Considerations for evaluating large language models for cybersecurity tasks

**URL**: View paper

**Brief Assessment**

LLM Cybersecurity Evaluation[60] focuses on evaluation methodology and considerations for assessing LLMs on cybersecurity tasks, not on conducting comprehensive empirical evaluations of multiple frontier agents and models on a large-scale benchmark like CyberGym.

---

#### 3. Cyberpal. ai: Empowering llms with expert-driven cybersecurity instructions

**URL**: View paper

**Brief Assessment**

Cyberpal[63] focuses on fine-tuning LLMs for cybersecurity instruction-following and knowledge tasks, not on evaluating AI agents' capabilities in real-world cybersecurity scenarios like vulnerability reproduction.

---

#### 4. From vulnerability to defense: The role of large language models in enhancing cybersecurity

**URL**: View paper

**Brief Assessment**

LLM Vulnerability Defense[66] focuses on general applications of LLMs in cybersecurity (phishing detection, malware analysis, policy drafting) rather than evaluating AI agents on specific cybersecurity benchmarks like vulnerability reproduction tasks.

---

#### 5. A comprehensive survey: Evaluating the efficiency of artificial intelligence and machine learning techniques on cyber security solutions

**URL**: View paper

**Brief Assessment**

AI Cybersecurity Survey[59] is a general survey on ML/DL/RL applications in cybersecurity (malware detection, intrusion detection, vulnerability assessment). It does not evaluate frontier AI agents or LLMs on specific cybersecurity benchmarks like CyberGym does.

---

#### 6. Beyond detection: large language models and next-generation cybersecurity

**URL**: View paper

**Brief Assessment**

LLM Next-Gen Cybersecurity[65] is a survey paper discussing LLM applications across cybersecurity domains (software security, network security, content moderation, etc.) but does not present empirical evaluations of specific agent frameworks or frontier LLMs on standardized cybersecurity benchmarks like CyberGym does.

---

#### 7. Cyberseceval 3: Advancing the evaluation of cybersecurity risks and capabilities in large language models

**URL**: View paper

**Brief Assessment**

Cyberseceval[68] focuses on evaluating LLMs for cybersecurity risks (social engineering, code security, prompt injection) rather than evaluating AI agents on vulnerability reproduction tasks. The candidate does not challenge the novelty of comprehensive agent evaluation on CyberGym's specific benchmark.

---

#### 8. From Texts to Shields: Convergence of Large Language Models and Cybersecurity

**URL**: View paper

**Brief Assessment**

Texts to Shields[67] is a survey paper discussing LLM applications in cybersecurity broadly. It does not present empirical evaluations of AI agents on specific cybersecurity benchmarks like CyberGym.

---

#### 9. When llms meet cybersecurity: A systematic literature review

**URL**: View paper

**Brief Assessment**

LLMs Cybersecurity Review[61] is a systematic literature review of LLM applications in cybersecurity, not an empirical evaluation benchmark. It surveys existing work rather than conducting original experiments evaluating specific agents and models on cybersecurity tasks.

---

#### 10. Specification and Evaluation of Multi-Agent LLM Systems--Prototype and Cybersecurity Applications

**URL**: View paper

**Brief Assessment**

Multi-Agent LLM Systems[62] focuses on multi-agent system specification and evaluation for cybersecurity tasks using a schema-based architecture, not on comprehensive benchmarking of frontier agents and LLMs at scale like CyberGym's 1,507 instances across 188 projects.

---

## Contribution 3: Platform for open-ended vulnerability discovery with real-world security impact

**Description**: The authors demonstrate that CyberGym extends beyond static benchmarking to create direct security impact. Their evaluation led to the discovery of 35 zero-day vulnerabilities and 17 incomplete patches in real-world software, with responsible disclosure to maintainers.

This contribution was assessed against **10 related papers** from the literature. Papers with potential prior art are analyzed in detail with textual evidence; others receive brief assessments.

---

### 1. CAI: An Open, Bug Bounty-Ready Cybersecurity AI

**URL**: View paper

**Prior Art Analysis**

CAI[73] demonstrates that its framework extends beyond static benchmarking to create direct security impact through vulnerability discovery. The candidate paper reports discovering 35 zero-day vulnerabilities and performing responsible disclosure to maintainers, with 3 CVE assignments received and 6 vulnerabilities patched. This directly parallels the original paper's claim of discovering 35 zero-days and 17 incomplete patches, showing that similar prior work exists in creating platforms that demonstrate real-world security impact through open-ended vulnerability discovery.

**Evidence**

Evidence 1 - **Rationale**: Both papers demonstrate real-world security impact through competitive performance and vulnerability discovery, though CAI focuses more on competitive CTF scenarios while CyberGym emphasizes vulnerability discovery in open-source projects. - **Original**: beyond static benchmarking, we show that cybergym leads to the discovery of 35 zero-day vulnerabilities and 17 historically incomplete patches. these results underscore that cybergym is not only a robust benchmark for measuring ai's progress in cybersecurity but also a platform for creating direct, ... - **Candidate**: overall, cai's performance in the "ai vs human" ctf challenge highlights its ability to compete at the highest level, achieving a top-1 rank amongst ai teams, which got rewarded by a 750 usd prize, and a top-20 ranking overall despite a 3 hour-limited active time . with a strong start, it outperform...

Evidence 2 - **Rationale**: Both papers emphasize that their platforms extend beyond theoretical benchmarks to demonstrate direct, real-world security impact through vulnerability discovery and practical security testing. - **Original**: cybergym extends to creating direct, real-world security impactbeyond benchmarking, cybergym produces a direct impact on practical security, addressing limitation (ii). during our evaluation, we found that even when tasked with reproducing a specific vulnerability, the agents can inadvertently gener... - **Candidate**: beyond cybersecurity competitions, cai demonstrates real-world effectiveness, reaching top-30 in spain and top-500 worldwide on hack the box within a week, while dramatically reducing security testing costs by an average of 156 x. our framework transcends theoretical benchmarks by enabling non-profe...

---

### 2. Supporting continuous vulnerability compliance through automated identity provisioning

**URL**: View paper

**Brief Assessment**

Vulnerability Compliance[69] focuses on continuous vulnerability compliance monitoring and workload isolation in CI/CD pipelines, not on open-ended vulnerability discovery platforms or zero-day detection capabilities.

---

### 3. CVE-Bench: A Benchmark for AI Agents' Ability to Exploit Real-World Web Application Vulnerabilities

**URL**: View paper

**Brief Assessment**

CVE-Bench Web[74] focuses on exploiting known web application vulnerabilities (CVEs) through standardized attack evaluation, not open-ended vulnerability discovery. The candidate's framework evaluates agents on pre-defined attack types against existing CVEs, whereas the original contribution emphasizes discovering new zero-day vulnerabilities through open-ended exploration.

---

### 4. AI-Based Web Vulnerability Scanner: A Comprehensive Review

**URL**: View paper

**Brief Assessment**

AI Vulnerability Scanner[75] focuses on reviewing AI-based web vulnerability scanners and their effectiveness in vulnerability management. It does not present a platform for open-ended vulnerability discovery or demonstrate discovery of zero-day vulnerabilities in real-world software.

---

### 5. Understanding software vulnerabilities in the maven ecosystem: Patterns, timelines, and risks

**URL**: View paper

**Brief Assessment**

Maven Vulnerabilities[71] focuses on analyzing existing documented vulnerabilities in the Maven ecosystem, examining patterns, timelines, and resolution delays. It does not describe a platform for open-ended vulnerability discovery or demonstrate discovery of new zero-day vulnerabilities through AI agents.

---

### 6. Securing container images through automated vulnerability detection in shift-left CI/CD pipelines

**URL**: View paper

**Brief Assessment**

Container Image Security[72] focuses on automated vulnerability detection in CI/CD pipelines for container images (Docker/Kubernetes), not on open-ended vulnerability discovery platforms or zero-day detection in diverse software projects like CyberGym.

---

### 7. Reef: A framework for collecting real-world vulnerabilities and fixes

**URL**: View paper

**Brief Assessment**

Reef[76] focuses on collecting and curating vulnerability datasets from CVEs with LLM-generated explanations, not on building an interactive platform for agents to discover zero-day vulnerabilities through open-ended exploration and responsible disclosure.

---

### 8. Transforming SOC Operations: Harnessing the Power of AI and ML for Enhanced Threat Detection

**URL**: View paper

**Brief Assessment**

SOC AI Operations[77] focuses on AI/ML integration in Security Operations Centers for vulnerability management automation, not on building platforms for open-ended vulnerability discovery or demonstrating zero-day findings through agent-based systems.

---

### 9. Specification-Guided Vulnerability Detection with Large Language Models

**URL**: View paper

**Brief Assessment**

Specification-Guided Detection[70] focuses on vulnerability detection using security specifications extracted from historical vulnerabilities, not on creating a platform for open-ended vulnerability discovery. The candidate's real-world impact comes from detecting a single CVE through specification-guided analysis, whereas the original paper emphasizes a platform approach that discovered 35 zero-days through agent-based exploration.

---

### 10. Towards LLM-Assisted Vulnerability Detection and Repair for Open-Source 5G UE Implementations

**URL**: View paper

**Brief Assessment**

5G Vulnerability Detection[78] focuses on LLM-assisted vulnerability detection in 5G UE implementations, not on creating a general platform for open-ended vulnerability discovery across diverse software projects like CyberGym does.

---

## Appendix: Text Similarity Detection

Textual similarity detection checked 34 papers and found 3 similarity segment(s) across 1 paper(s).

The following **1 paper(s)** were detected to have high textual similarity with the original paper. These may represent different versions of the same work, duplicate submissions, or papers with substantial textual overlap. Readers are advised to verify these relationships independently.

### 1. CyberGym: Evaluating AI Agents' Cybersecurity Capabilities with Real-World Vulnerabilities at Scale

**Detected in**: Contribution: contribution_1

⚠ **Note**: This paper shows substantial textual similarity with the original paper. It may be a different version, a duplicate submission, or contain significant overlapping content. Please review carefully to determine the nature of the relationship.

## References


- [0] CyberGym: Evaluating AI Agents' Real-World Cybersecurity Capabilities at Scale View paper
- [1] PoCGen: Generating Proof-of-Concept Exploits for Vulnerabilities in Npm Packages View paper
- [2] Is github's copilot as bad as humans at introducing vulnerabilities in code? View paper
- [3] PoCo: Agentic Proof-of-Concept Exploit Generation for Smart Contracts View paper
- [4] Real-Time AI Code Security Auditing: Automated Vulnerability Detection and Remediation Through Meta-Experimental Analysis View paper
- [5] Eradicating the unseen: Detecting, exploiting, and remediating a path traversal vulnerability across github View paper
- [6] LLM-Driven, Self-Improving Framework for Security Test Automation: Leveraging Karate DSL for Augmented API Resilience View paper
- [7] Untrustide: Exploiting weaknesses in vs code extensions View paper
- [8] Test Suites Guided Vulnerability Validation for Node. js Applications View paper
- [9] Visualizing Security Vulnerability Evolution of Software Systems View paper
- [10] CyberGym: Evaluating AI Agents' Cybersecurity Capabilities with Real-World Vulnerabilities at Scale View paper
- [11] Unveiling Security Vulnerabilities in Git Large File Storage Protocol View paper
- [12] FuSeBMC: An Energy-Efficient Test Generator for Finding Security Vulnerabilities in C Programs View paper
- [13] SEC-bench: Automated Benchmarking of LLM Agents on Real-World Software Security Tasks View paper
- [14] SecureAgentBench: Benchmarking Secure Code Generation under Realistic Vulnerability Scenarios View paper
- [15] Systematic generation of XSS and SQLi vulnerabilities in PHP as test cases for static code analysis View paper
- [16] A Friend's Eye is A Good Mirror: Synthesizing {MCU} Peripheral Models from Peripheral Drivers View paper
- [17] LLM Agents for Automated Web Vulnerability Reproduction: Are We There Yet? View paper
- [18] Bullseye: Detecting prototype pollution in npm packages with proof of concept exploits View paper
- [19] Vulnerability and Attack Repository for IoT: Addressing Challenges and Opportunities in Internet of Things Vulnerability Databases View paper
- [20] A generative and mutational approach for synthesizing bug-exposing test cases to guide compiler fuzzing View paper
- [21] Understanding and preventing open-source software supply chain attacks View paper
- [22] Counterfeit object-oriented programming: On the difficulty of preventing code reuse attacks in C++ applications View paper
- [23] Test mimicry to assess the exploitability of library vulnerabilities View paper
- [24] Recycling test cases to detect security vulnerabilities View paper
- [25] Secure coding practice in Java: Automatic detection, repair, and vulnerability demonstration View paper
- [26] {IvySyn}: Automated vulnerability discovery in deep learning frameworks View paper
- [27] Minerva: browser API fuzzing with dynamic mod-ref analysis View paper
- [28] Writing secure code View paper
- [29] Automated Exploit Generation for Node.js Packages View paper
- [30] Detecting Fake Proof-of-Concept Codes on GitHub Using Static Code Analysis View paper
- [31] Cybersecurity vulnerability identification in system-of-systems using model-based testing View paper
- [32] Intelligen: Automatic driver synthesis for fuzz testing View paper
- [33] Secure CodeCity A Framework For Security Vulnerability Visualization View paper
- [34] Understanding the reproducibility of crowd-reported security vulnerabilities View paper
- [35] SP-Fuzz: Fuzzing Soft PLC with Semi-automated Harness Synthesis View paper
- [36] Octopocs: automatic verification of propagated vulnerable code using reformed proofs of concept View paper
- [37] Semfuzz: Semantics-based automatic generation of proof-of-concept exploits View paper
- [38] Modernization of a legacy codebase View paper
- [39] A Systematic Study on Generating Web Vulnerability Proof-of-Concepts Using Large Language Models View paper

- [40] A Study on Exploit Development View paper
- [41] Bug synthesis: Challenging bug-finding tools with deep faults View paper
- [42] Automated test generation from vulnerability signatures View paper
- [43] N-day Vulnerabilities: Detection, Bisection, and Measurement View paper
- [44] Demo: Backdoor Through the Front Door: Demonstrating Security Flaws in the Eufy Ecosystem View paper
- [45] Automatic creation of SQL injection and cross-site scripting attacks View paper
- [46] Beyond the Surface: Investigating Malicious CVE Proof of Concept Exploits on GitHub View paper
- [47] An Automated Identification Method for Controllable Memory-Related Fields in Proof-of-Concept Code View paper
- [48] Detecting Prototype Pollution in NPM Packages with Proof of Concept Exploits View paper
- [49] Guided differential testing of certificate validation in SSL/TLS implementations View paper
- [50] Enhancing Software Vulnerability Detection Through Adaptive Test Input Generation Using Genetic Algorithm View paper
- [51] CVE-Bench: Benchmarking LLM-based Software Engineering Agent's Ability to Repair Real-World CVE Vulnerabilities View paper
- [52] CVE-assisted large-scale security bug report dataset construction method View paper
- [53] A large-scale empirical study on vulnerability distribution within projects and the lessons learned View paper
- [54] Cheesecloth: Zero-Knowledge Proofs of Real-World Vulnerabilities View paper
- [55] Detection of recurring software vulnerabilities View paper
- [56] On Security Weaknesses and Vulnerabilities in Deep Learning Systems View paper
- [57] Large-scale empirical study of important features indicative of discovered vulnerabilities to assess application security View paper
- [58] Benchmarking static analysis tools for web security View paper
- [59] A comprehensive survey: Evaluating the efficiency of artificial intelligence and machine learning techniques on cyber security solutions View paper
- [60] Considerations for evaluating large language models for cybersecurity tasks View paper
- [61] When llms meet cybersecurity: A systematic literature review View paper
- [62] Specification and Evaluation of Multi-Agent LLM Systems--Prototype and Cybersecurity Applications View paper
- [63] Cyberpal. ai: Empowering llms with expert-driven cybersecurity instructions View paper
- [64] Assessing confidence in frontier AI safety cases View paper
- [65] Beyond detection: large language models and next-generation cybersecurity View paper
- [66] From vulnerability to defense: The role of large language models in enhancing cybersecurity View paper
- [67] From Texts to Shields: Convergence of Large Language Models and Cybersecurity View paper
- [68] Cyberseceval 3: Advancing the evaluation of cybersecurity risks and capabilities in large language models View paper
- [69] Supporting continuous vulnerability compliance through automated identity provisioning View paper
- [70] Specification-Guided Vulnerability Detection with Large Language Models View paper
- [71] Understanding software vulnerabilities in the maven ecosystem: Patterns, timelines, and risks View paper
- [72] Securing container images through automated vulnerability detection in shift-left CI/CD pipelines View paper
- [73] CAI: An Open, Bug Bounty-Ready Cybersecurity AI View paper
- [74] CVE-Bench: A Benchmark for AI Agents' Ability to Exploit Real-World Web Application Vulnerabilities View paper
- [75] AI-Based Web Vulnerability Scanner: A Comprehensive Review View paper
- [76] Reef: A framework for collecting real-world vulnerabilities and fixes View paper
- [77] Transforming SOC Operations: Harnessing the Power of AI and ML for Enhanced Threat Detection View paper
- [78] Towards LLM-Assisted Vulnerability Detection and Repair for Open-Source 5G UE Implementations View paper