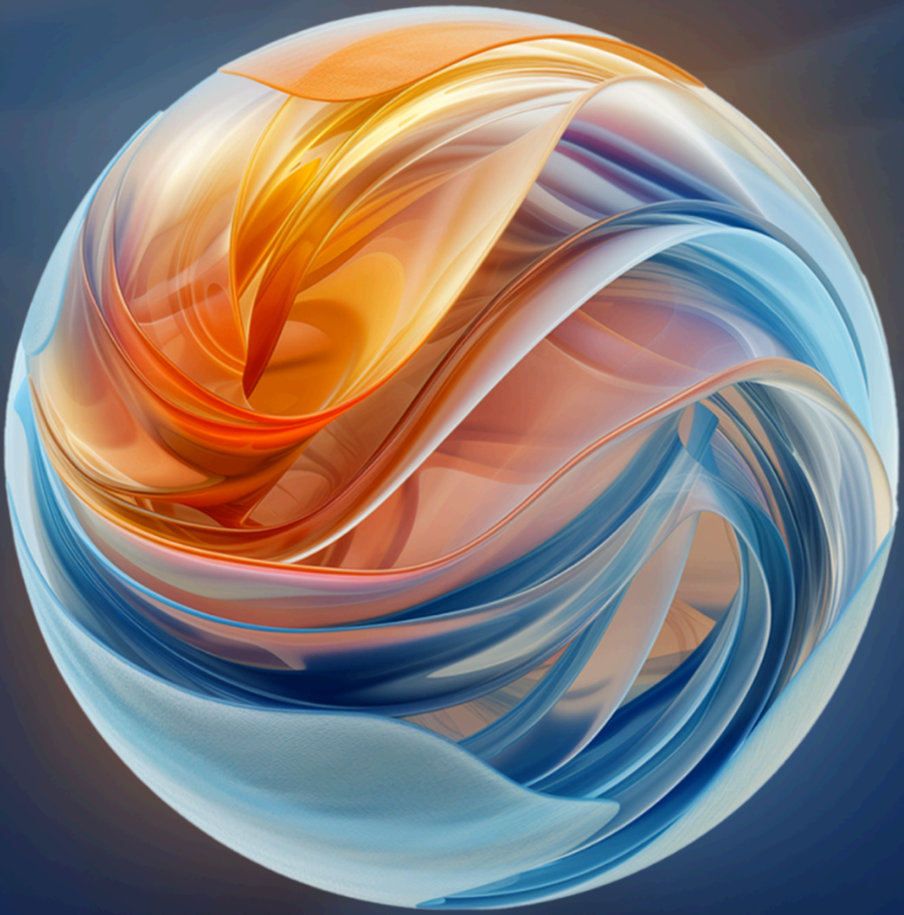
Using AI for Offensive Security



AI Technology and Risk Working Group

The permanent and official location for the AI Technology and Risk Working Group is https://cloudsecurityalliance.org/research/working-groups/ai-technology-and-risk

Acknowledgments

Initiative Leads

Adam Lundqvist
Kirti Chopra

Lead Authors

Adam Lundqvist
Kirti Chopra
Michael Roza
Sven Vetsch

Contributors

Candy Alexander
David McCrory
Elier Cruz
Govindaraj Palanisamy
John Jiang
Keith Pasley
Ken Walling
Lars Ruddigkeit
Maria Schwenger

Reviewers

Adam Lomas
Akhil Mittal
Akshat Vashishtha
Alex Rebo
Ashish Vashishtha
Ken Huang
Nancy Kramer
Neil Kessler
Patrick Schmid
Rocking Wolf Ranch
Sven Olensky
Vaibhav Malik
Venkatesh Gopal
Vivek Shitole

Co-Chairs

Chris Kirschke
Mark Yanalitis

CSA Global Staff

Josh Buker
Stephen Smith

Table of Contents

Executive Summary

The emergence of Artificial Intelligence (AI) technology, particularly Large Language Models (LLMs) and LLM powered AI Agents, has triggered a profound transformation in the landscape of offensive security, including vulnerability assessment, penetration testing, and red teaming. This shift redefines AI from a narrow use case to a versatile and powerful general-purpose technology. This paper explores the transformative potential of LLM-powered AI by examining its integration into offensive security, addressing current challenges, and showcasing AI's capabilities across five security phases: reconnaissance, scanning, vulnerability analysis, exploitation, and reporting.

Key Findings

Challenges in Offensive Security: Security teams face a shortage of skilled professionals, increasingly complex and dynamic environments, and the need to balance automation with manual testing.

AI Capabilities: AI, mainly through LLMs and AI agents, offers significant capabilities in offensive security, including data analysis, code, text generation, planning realistic attack scenarios, reasoning, and tool orchestration. These capabilities can help automate reconnaissance, optimize scanning processes, assess vulnerabilities, generate comprehensive reports, and even autonomously exploit vulnerabilities.

AI Benefits: Leveraging AI in offensive security enhances scalability, efficiency, speed, discovery of more complex vulnerabilities, and ultimately, the overall security posture.

No Silver Bullet: While promising, no single AI solution can revolutionize offensive security today. Ongoing experimentation with AI is needed to find and implement effective solutions. This requires creating an environment that encourages learning and development, where team members can use AI tools and techniques to grow their skills.

Recommendations

AI Integration: Incorporate AI to automate tasks and augment human capabilities. Leverage AI for data analysis, tool orchestration, generating actionable insights and building autonomous systems where applicable. Adopt AI technologies in offensive security to stay ahead of evolving threats.

Human Oversight: LLM-powered technologies are unpredictable, can hallucinate, and cause errors. Maintain human oversight to validate AI outputs, improve quality, and ensure technical advantage.

Governance, Risk, and Compliance (GRC): Implement robust GRC frameworks and controls to ensure safe, secure, and ethical AI use.

This paper empowers executive leadership, including Chief Information Security Officers (CISOs), cybersecurity strategists, and executives, to secure the necessary resources for enhancing offensive AI capabilities by effectively articulating the value of offensive security investments.

Introduction

Offensive Security

Offensive security involves proactively simulating an attacker's behavior by using tactics and techniques similar to those of an adversary to identify system vulnerabilities. By understanding potential weaknesses and threats, organizations can implement and enhance robust security controls, thereby reducing the risk of exploitation by malicious actors.

To maximize the effectiveness of offensive security, it is crucial to ensure it aligns with the organization's long-term goals and objectives. This approach focuses on the security risks most relevant to the organization's priorities, ensuring resources are directed toward the areas that matter most.

The rest of this document explores three approaches to offensive security:

- **Vulnerability assessment:** can be used for the automated identification of weaknesses using scanners.
- **Penetration testing:** can be used to simulate cyberattacks in order to identify and exploit vulnerabilities.
- **Red teaming:** can be used to simulate a complex, multi-stage attack by a determined adversary, often to test an organization's detection and response capabilities.

These approaches share similarities but differ in various aspects, as shown in the table below. While this table provides a helpful overview, actual practices can differ based on various factors, including organizational maturity and risk tolerance.

Type/Aspect	Vulnerability Assessment	Penetration Testing	Red Teaming
Duration	Short (hours)	Medium (days)	Long (weeks)
Risk Alignment	Indirectly tied to organizational risk	Influenced by organizational risk	Based on organizational risk
Tooling	Mainly Automated	Automated/Manual/Custom	Highly Manual/Custom
Sophistication	Low	Moderate	High
Execution	Undisguised	Undisguised	Disguised (stealth)
Cost	Low	Moderate	High
Goal	Identify and prioritize potential vulnerabilities	Determine risks associated with a range of system vulnerabilities	Measure the impact and responses of an organization as a whole

Table 1: Offensive Security Testing Practices

While the techniques, breadth, and depth of testing vary across offensive security engagements, intrusion typically follows five phases: Reconnaissance, Scanning, Vulnerability Analysis, Exploitation, and Reporting. Before commencing any engagement, it is crucial to create a Statement of Work (SoW) or a scope statement and establish the rules of engagement. These foundational documents set the stage for the five phases shown. For the remainder of this paper, we discuss these five phases, while assuming the scope and rules of engagement have been defined.

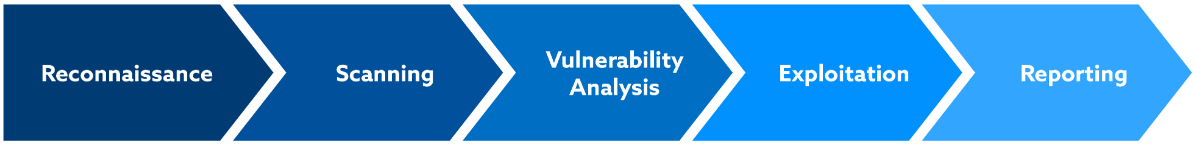


Figure 1: Offensive Security Testing Phases

Reconnaissance - Reconnaissance represents the initial phase in any offensive security strategy, aiming to gather extensive data regarding the target's systems, networks, and organizational structure.

Scanning - Scanning entails systematically examining identified systems to uncover critical details such as live hosts, open ports, running services, and the technologies employed, e.g., through fingerprinting to identify vulnerabilities.

Vulnerability Analysis - Vulnerability analysis further identifies and prioritizes potential security weaknesses within systems, software, network configurations, and applications.

Exploitation - Exploitation involves actively exploiting identified vulnerabilities to gain unauthorized access or escalate privileges within a system.

Reporting - The reporting phase concludes the offensive security engagement by systematically compiling all findings into a detailed report.

Current Challenges in Offensive Security

Offensive security testers navigate an increasingly complex landscape, facing numerous challenges that can hinder the efficiency and effectiveness of their assessments. These challenges are further compounded by a significant shortage of skilled security testers and cybersecurity experts in general.

Expanding attack surface: The proliferation of new technologies, such as AI, blockchain, cloud computing, IoT, and an increased remote workforce, has exponentially expanded the attack surface. This makes it harder to identify and secure all potential entry points.

Advanced threats: Adversaries are employing more sophisticated techniques, such as fileless malware or living-off-the-land attacks combined with zero-day exploits, which can be difficult to detect and mitigate.

Diversity of Assessments: Offensive security teams must be proficient in conducting a wide range of assessments, including vulnerability assessments, penetration tests, and red team exercises. Each type demands specific skills, techniques, and knowledge, making it difficult for testers to maintain expertise across all domains. For instance, testing a mobile banking app requires a different skill set than assessing an IoT device.

Adapting to Dynamic Environments: Offensive security assessments often occur in dynamic environments where target systems, security controls, and configurations change rapidly. Testers must remain agile and adapt their Tactics, Techniques, and Procedures (TTPs) in real-time, modifying attack strategies, pivoting when a portion of the attack is detected, or adjusting social engineering approaches based on observed user behavior.

Balancing Automation and Manual Testing: Automated tools are essential for efficiency, but over-reliance can lead to missed vulnerabilities and distraction by false positives. Striking the right balance between automated scanning and in-depth manual analysis is an ongoing challenge.

Time-Consuming Tasks: Certain tasks, such as comprehensive authorization testing, code reviews, crafting spear-phishing emails, and exploitation of complex vulnerabilities, are inherently time consuming. They often involve extensive reconnaissance and iterative exploitation attempts.

Tool Development and Customization: Offensive security testers often need to develop or adapt scripts, tools, or frameworks to uncover unique vulnerabilities or cater to specific target environments, which can strain limited resources.

Communication and Reporting: Clear and effective communication with the stakeholders during and after the assessment is equally important. Effectively communicating technical findings, translating them into actionable recommendations, and delivering concise reports that resonate with diverse audiences can be a significant challenge. This is particularly true when there is a substantial knowledge gap between security testers and their audience, making it difficult to bridge technical complexities with business priorities and risk tolerance.

Data Analysis and Threat Intelligence: The sheer volume of data generated during an assessment can be overwhelming. Extracting meaningful insights, correlating findings, and staying up-to-date with the latest threat intelligence demands constant vigilance and advanced analysis techniques, all while contending with limited resources and staffing.

Compliance and Ethical Considerations: Offensive security testers must adhere to a growing number of strict security standards, regulations, and ethical guidelines, ensuring that their actions do not cause unintended harm or exceed the agreed-upon scope of the assessment. This is resource intensive and time consuming for testers.

The shortage of cybersecurity professionals and the growing complexity of cyber attacks have created a pressing need for innovative solutions. Artificial intelligence, particularly LLMs, offers promising avenues for addressing many of these challenges. AI could alleviate the strain on human resources, significantly augment the capabilities of offensive security testers, and enhance the effectiveness of offensive security practices in general.

Artificial Intelligence

AI encompasses a variety of technologies designed to emulate human intelligence, including natural language processing, machine learning, and robotics. While AI covers a wide range of technologies, our focus is on technologies powered by LLMs.

Large Language Models

LLMs are sophisticated deep neural networks with billions of parameters that process and generate language. Originally developed to predict subsequent words in sentences, these models now handle complex tasks across text, speech, audio, and video applications.

LLMs operate on machine learning principles involving two key phases: training and inference. During the training phase, LLMs analyze vast amounts of text data to identify linguistic patterns and learn to predict subsequent words and generate coherent language. In the inference phase, learned patterns are leveraged to process new text inputs, generate predictions, complete sentences, and provide relevant responses.

LLMs are not limited to language processing. They excel at swiftly analyzing large amounts of data, including text, code, logs, and HTTP traffic. Leveraging generative capabilities, they can create code, scripts, and emails, as well as compile summaries and reports. The most advanced LLMs demonstrate emergent abilities such as **reasoning** about text and making procedural decisions, which are crucial for **planning** and **goal-oriented tasks**. To align with the terminology used in many papers on AI and AI agents, we use the term "reason" to describe the ability to analyze text and make procedural decisions. However, we acknowledge that there is ongoing research into whether AI agents can reason in the same way humans do.

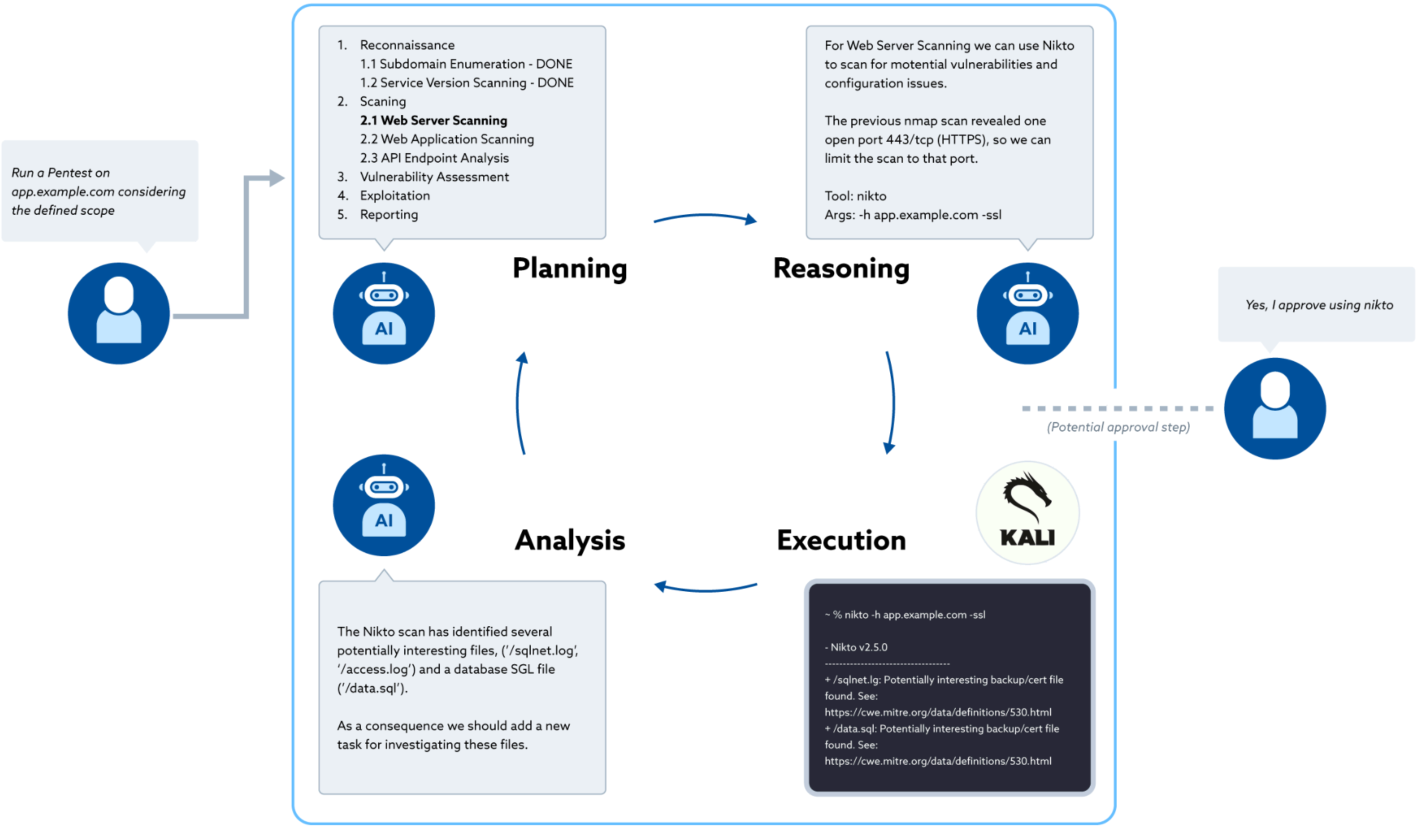
While LLMs offer impressive capabilities, they are not without their limitations. For instance, they sometimes generate plausible-sounding but factually incorrect responses, a phenomenon known as **hallucination**. This can be problematic when accurate information is essential, but may be beneficial in creative applications like brainstorming. Further, LLMs include stochastic elements that enhance the variety and naturalness of their outputs but also **reduce predictability**. As a result, similar queries might produce completely different responses, unlike in traditional computational systems.

The application of an LLM, whether confined to a chat window or used to make real-world decisions, may depend on the specific use case addressed. While chat-based LLMs are widely used for content creation and creative purposes, ongoing research explores how these models can be leveraged to solve complex problems autonomously, a significant step towards broader AI utilization.

AI Agents

AI agents are autonomous or sometimes semi-autonomous systems designed to perceive their environments and act to achieve set goals, thus shaping future interactions with the environment. These agents can use the power of LLMs to plan tasks, trigger task execution, make decisions, and interact meaningfully with the world. Unlike basic LLM applications, an AI agent using LLMs follows a cyclic approach to achieve its end goal, continuously learning and adapting from its findings and adjusting its approach. This iterative self-adaptation makes the agent effective at solving complex problems through a multistep process until the task is completed.

An agent begins by breaking down the user request into actionable and prioritized plans (**Planning**). It then reasons with available information to choose appropriate tools or next steps (**Reasoning**). The LLM cannot execute tools, but attached systems execute the tool correspondingly (**Execution**) and collect the tool outputs. Then, the LLM interprets the tool output (**Analysis**) to decide on the next steps used to update the plan. This iterative process enables the agent to continue working cyclically until the user's request is resolved.


Run a Pentest on app.example.com considering the defined scope
Planning
Reasoning
AI
Yes, I approve using nikto
(Potential approval step)
Analysis
Execution
KALI

1. Reconnaissance
   1.1 Subdomain Enumeration - DONE
   1.2 Service Version Scanning - DONE
2. Scaning
   2.1 Web Server Scanning
   2.2 Web Application Scanning
   2.3 API Endpoint Analysis
3. Vulnerability Assessment
4. Exploitation
5. Reporting

For Web Server Scanning we can use Nikto to scan for motential vulnerabilities and configuration issues.

The previous nmap scan revealed one open port 443/tcp (HTTPS), so we can limit the scan to that port.

Tool: nikto
Args: -h app.example.com -ssl

```
~ % nikto -h app.example.com -ssl

- Nikto v2.5.0
-----------------------------------
+ /sqlnet.lg: Potentially interesting backup/cert file
found. See:
https://cwe.mitre.org/data/definitions/530.html
+ /data.sql: Potentially interesting backup/cert file
found. See:
https://cwe.mitre.org/data/definitions/530.html
```

The Nikto scan has identified several potentially interesting files, ('/sqlnet.log', '/access.log') and a database SGL file ('/data.sql').

As a consequence we should add a new task for investigating these files.

Figure 2: AI Agent Phases

As shown in Figure 2, consider a simplified example of an AI agent handling a complete penetration test for a web application running at app.example.com, where we look into one cycle of the AI agent, the Web Server Scanning phase. This cycle can be detailed as follows:

1. **Planning:** Based upon the existing plan and information created in previous cycles, the agent prioritizes the task of Web Server Scanning.
2. **Reasoning:** The agent identifies the appropriate tool, in this case Nikto, for web server scanning.
3. **Execution:** The selected tool is executed against the target. Depending on its configuration, it can either trigger the tool directly or patiently await human approval, providing a flexible and user-friendly experience.
4. **Analysis:** The agent processes Nikto's output, identifying potential vulnerabilities like exposed configuration files and databases. It then adjusts its execution for future cycles by recommending changes to the plan, perhaps adding a task for further investigation of specific findings, which will be considered in the next planning phase.

A simplified model of an AI agent like this may face limitations due to the potential breadth of the context, causing the agent to lose track of their goals. To mitigate this, the concept of Multi-Agent Systems has evolved, showing promising results in complex problem solving. Such systems comprise a lineup of AI agents that collaborate to achieve tasks. Moreover, sophisticated short- and long-term memory management is essential, often involving access to an external authoritative knowledge base through Retrieval Augmented Generation (RAG) systems.

RAG contextualizes LLM output founded on an authoritative knowledge base external to its own training data set. This approach enhances the relevance and accuracy of the responses generated by the LLM. By leveraging both internal and external knowledge, the AI agent can provide more comprehensive solutions. This integration is crucial for maintaining context and extending the LLM's capability to generate output specific to an organization or domain.

AI-Powered Offensive Security

AI has the potential to transform offensive security. AI-powered tools can simulate advanced cyberattacks and identify network, system, and software vulnerabilities before malicious actors can exploit them. They can aid security teams in efficiently scaling their efforts. AI may help cover a broader range of attack scenarios, respond dynamically to vulnerability findings, adapt to different environments, and improve over time.

AI models can suggest attack paths, generate and execute unseen test cases, and learn from each interaction. AI-driven tools can also process vast amounts of data, uncover patterns unrecognizable to the human eye, and assist in discovering vulnerabilities.

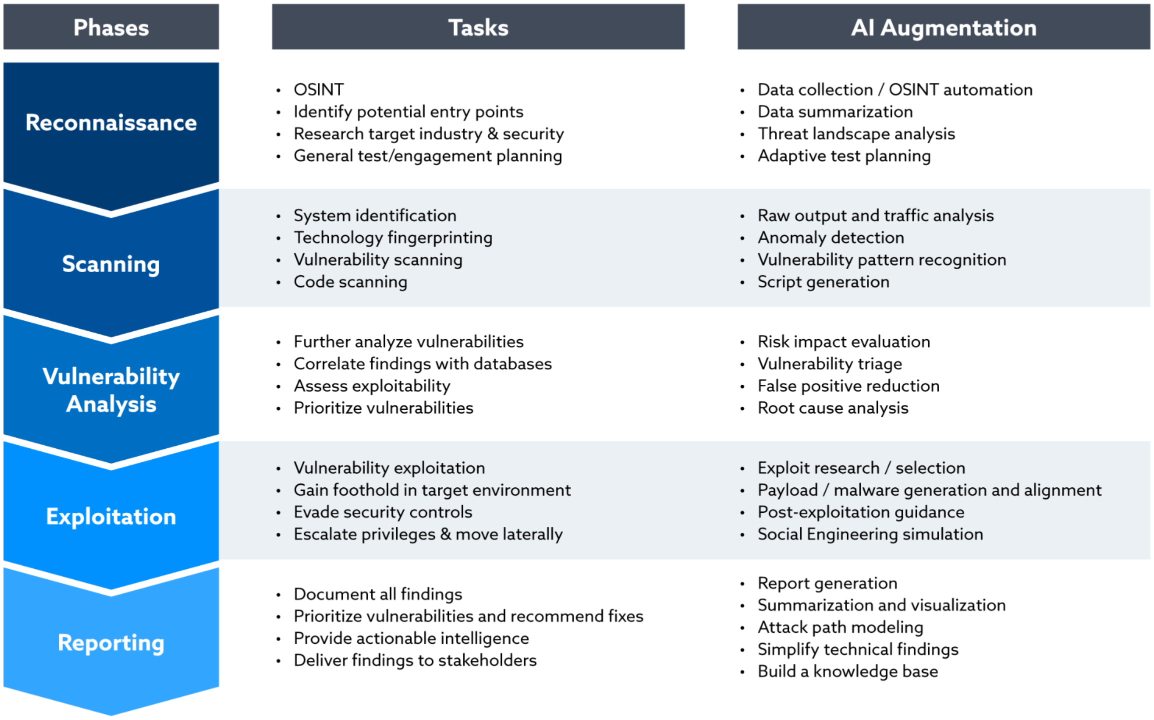
However, AI solutions are not a silver bullet. Their effectiveness is limited by the scope of their training data and algorithms. Novel or complex scenarios might fall outside their capabilities. Human expertise remains crucial for interpreting anomalies, applying judgment, and making strategic decisions with broader considerations beyond the immediate technical data. Therefore, it is essential to understand the current state-of-the-art of AI and leverage it as an augmentation tool for human security professionals.

AI Augmentation and Autonomy

As discussed above, AI agents can autonomously or semi-autonomously navigate through cycles of planning, reasoning, tool execution, and analysis to execute offensive security tasks with varying levels of human input.

However, the level of autonomy granted to an AI agent has to balance the benefits of automation and augmentation against the risks of unintended consequences, especially in critical applications. Ensuring human oversight — or keeping the '**Human in the Loop**' — is a strategic advantage. This approach is crucial for combining AI's unique strengths with human expertise, leading to the best outcomes. It maintains quality standards and accountability, especially in sensitive areas where engagements are performed in productive environments.

AI can augment or automate an existing offensive security testing process as follows:



Phases	Tasks	AI Augmentation
Reconnaissance	• OSINT • Identify potential entry points • Research target industry & security • General test/engagement planning	• Data collection / OSINT automation • Data summarization • Threat landscape analysis • Adaptive test planning
Scanning	• System identification • Technology fingerprinting • Vulnerability scanning • Code scanning	• Raw output and traffic analysis • Anomaly detection • Vulnerability pattern recognition • Script generation
Vulnerability Analysis	• Further analyze vulnerabilities • Correlate findings with databases • Assess exploitability • Prioritize vulnerabilities	• Risk impact evaluation • Vulnerability triage • False positive reduction • Root cause analysis
Exploitation	• Vulnerability exploitation • Gain foothold in target environment • Evade security controls • Escalate privileges & move laterally	• Exploit research / selection • Payload / malware generation and alignment • Post-exploitation guidance • Social Engineering simulation
Reporting	• Document all findings • Prioritize vulnerabilities and recommend fixes • Provide actionable intelligence • Deliver findings to stakeholders	• Report generation • Summarization and visualization • Attack path modeling • Simplify technical findings • Build a knowledge base

Table 2: AI potential for augmenting human tasks in offensive security phases

As the table above demonstrates, security testers can significantly benefit from using AI in all phases of an offensive security engagement. AI can be granted varying levels of autonomy, allowing for a tailored balance between automation and augmentation while adhering to regulatory and organizational policies.

The figure below provides a conceptual view of how increased dependency on AI agents might be implemented and managed.

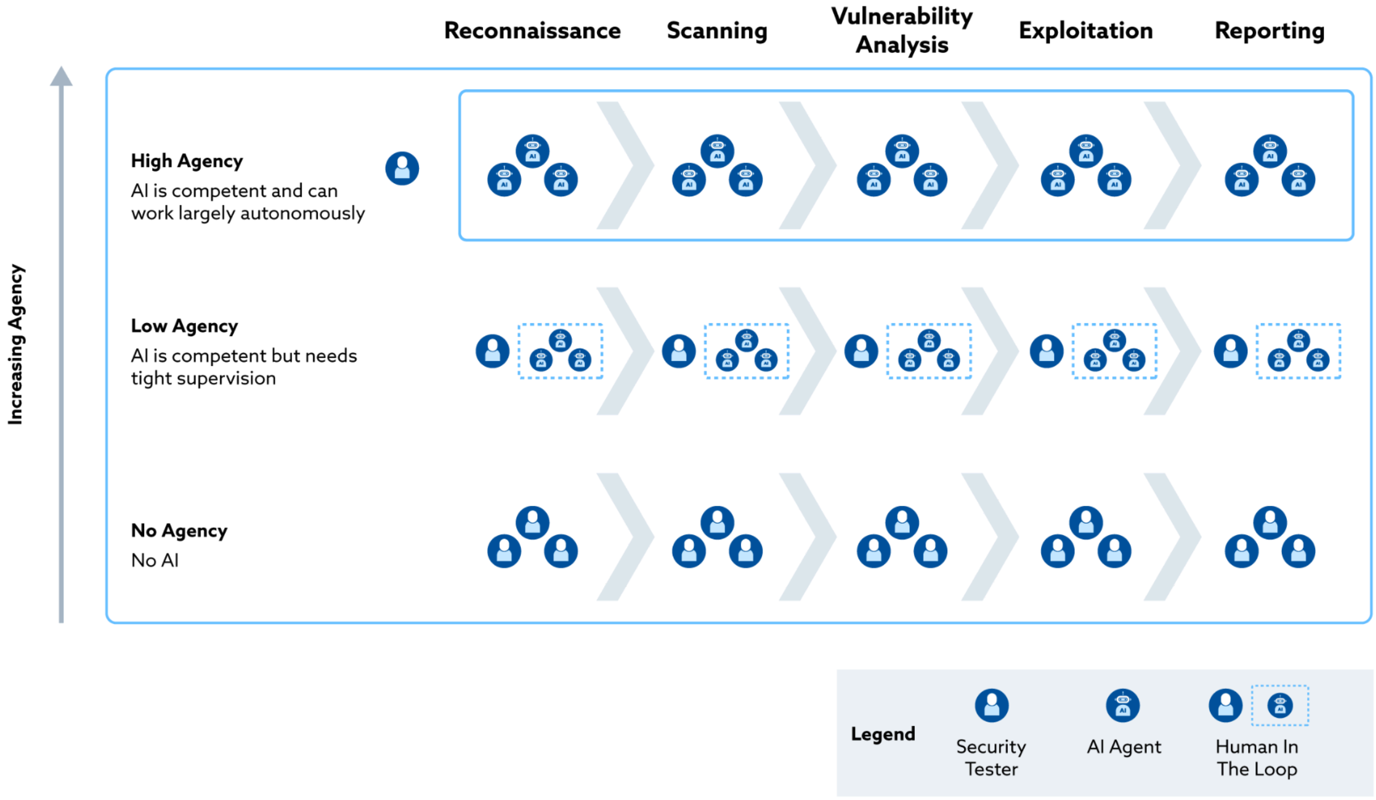


Figure 3: AI Agency in the Offensive Security Process

- **No Agency:** In the absence of AI, humans perform all tasks manually or with human-operated tools.
- **Low Agency:** Granting AI low-level agency involves assisting with specific tasks, such as data analysis or planning, while human supervision is still required for decision-making and execution.
- **High Agency:** Granting AI high-level agency enables it to autonomously execute tasks across different phases with minimal human intervention, though humans remain in the loop for oversight.

As AI-powered systems become more advanced, they can be entrusted with greater autonomy, leading to scenarios where the AI agents work autonomously throughout engagements. The extent of human intervention depends on the AI's capabilities, scope, predefined rules of engagement, and trustworthiness. This progression ensures that AI-driven tasks remain effective while leveraging the full potential of AI in offensive security. By responsibly integrating AI capabilities, irrespective of the levels of autonomy, offensive security teams can support an adaptive and resilient security strategy.

In the following sections, we explore the applicability of AI capabilities, specifically LLMs, across the aforementioned five phases of offensive security.

Reconnaissance

The objective of the reconnaissance phase is to gather detailed information about the target from publicly available sources and passive reconnaissance techniques. This includes identifying the target's network ranges, domain names, employee details, technologies in use, and potential vulnerabilities disclosed in security advisories. Tools utilized in this phase range from websites to social media to network traffic analysis and various databases, which aim to accurately map the organization's attack surface. Unlike external threat actors, offensive security testers sometimes have access to internal resources such as network diagrams, design documents, or configuration management databases (CMDBs), which accelerate and enhance reconnaissance efforts. In scenarios like white box tests, where internal knowledge is already available, reconnaissance efforts may be streamlined or even unnecessary.

Filtering through a vast amount of data to isolate pertinent information while discarding outdated or irrelevant data presents a significant challenge in reconnaissance. AI offers a solution by automating both data collection and analysis, efficiently identifying relevant information, and discarding extraneous data. Additionally, AI can act as a powerful assistant, streamlining the reconnaissance process, enabling security testers to focus on critical analysis, make better decisions faster, and develop better inference and attack strategies.

Adaptive Test Planning: AI Agents can automatically analyze extensive resources and exploit pattern repositories to generate tailored test cases for specific systems and configurations. This automation reduces the time and effort spent on manual test case creation while increasing the scope of coverage. Pentest GPT translates overarching test objectives into precise, actionable steps with increased accuracy and reliability. This adaptive approach extends to tailoring context-aware testing strategies, integrating insights from ongoing tool analysis, and data evaluation. AI Agents can dynamically prioritize actionable steps, optimizing the order based on exploitability, severity, and impact.

Data Analysis: Leveraging LLMs to analyze responses from diverse network services (e.g., SMTP, FTP, HTTP) or tools can yield insights into the target's infrastructure. This includes IP address ranges, domain names, network topology, vendor technologies, and the types of SSL/TLS ciphers, ports, and services utilized. LLMs' robust data analysis capabilities have the potential to significantly streamline research efforts and enhance the quality of findings.

Tool Orchestration for Data Gathering: AI can be leveraged to craft requests, queries, and command line arguments efficiently. One study specifically used AI to craft various queries and command-line tool arguments using natural language. Utilizing ShellGPT with GPT-3.5, the researchers automated the generation of precise and contextually appropriate command line arguments from natural language to gather information and extract actionable intelligence from databases and logs. Such an approach can assist in passive and active reconnaissance for collecting vital information such as IP address ranges, domain names, and WHOIS records.

Automated Data Collection: AI agent systems, such as AutoGPT, have been used to autonomously discern potential targets by scrutinizing social media or web pages, thus laying the groundwork for a thorough external offensive security assessment.

Depending on the authority delegated to AI, it can assist the security tester or may plan and orchestrate larger parts of the reconnaissance phase.

Scanning

The objective of the scanning phase is to actively probe systems and networks to develop a detailed map of the target and its network structure, including potential vulnerabilities. Activities include detecting open ports, running services, employed technologies, and potential weaknesses using various tools. Tools and techniques involve using port scanners, vulnerability scanners, and fuzzing tools.

The primary challenge is handling the volume and complexity of data generated during the scan. Security testers must systematically analyze this data to identify critical details, which can be time consuming and prone to human error. AI can mitigate these challenges by automating the scanning process and analyzing the results faster than manual review. AI can identify patterns, correlate data, and perform intelligent continuous monitoring, enabling security testers to focus on higher-level analysis and strategy development.

Scanning Configuration: AI can analyze system configurations and recommend optimal settings for vulnerability scanners, ensuring comprehensive coverage without unnecessary overhead. Pentest GPT and the authors of this paper leverage LLMs to guide users in executing scanning tools by providing context-relevant command line arguments for offensive security tools.

Evaluation of Scanner Outputs: Large Language Models support the interpretation of tool outputs and suggest subsequent actions. This AI-powered approach enhances the depth and accuracy of vulnerability gathering about the target system. By processing and analyzing data from tools at scale, AI models can offer more precise insights, directing testers to concentrate their efforts where they are most likely to succeed.

Traffic Data Analysis: Targets generate large volumes of data from interactions that may be challenging for humans to analyze comprehensively. LLMs have been utilized to examine traffic-based data to identify vulnerabilities.

While AI can enhance efficiency during the scanning phase, validating its output is paramount to ensure the accuracy of results and help security professionals make more informed decisions throughout subsequent phases.

Vulnerability Analysis

The objective of the vulnerability analysis phase is to deeply analyze target systems and services for security flaws beyond initial scanning. While scanning provides a broad overview of the target's current state, a vulnerability analysis prioritizes potential technical security risks associated with these findings, evaluating each potential vulnerability's severity and impact.

The challenge is that vulnerability analysis remains labor intensive and resource demanding, requiring deep analysis. AI can mitigate these challenges by automating tasks, identifying zero-day vulnerabilities, and prioritizing real-time risks based on real-world threats, enabling security teams to address the most critical issues efficiently. AI can also help balance automation and manual testing by providing insights that guide in-depth manual analysis, reducing the risk of missed vulnerabilities and false positives.

False Positive Reduction: AI can be trained on vulnerability scan data to identify patterns and signatures associated with false positives. This helps to reduce the time and resources wasted investigating non-existent vulnerabilities.

Context-Aware Analysis: AI can analyze vulnerability scan results alongside system configurations, network topology, threat intelligence, and business requirements to identify potential vulnerabilities that traditional scanners might overlook. This context-aware approach offers a more holistic view of the security posture.

Interpreting Tool Outputs: Advanced AI models demonstrate exceptional skills in interpreting outputs from various testing tools, effectively guiding the vulnerability analysis process. This sophisticated capability significantly enhances the accuracy of identifying vulnerabilities, leveraging LLMs to optimize security testing strategies.

Source Code and Binary Analysis: AI's capabilities include automating source code analysis to detect vulnerabilities wherever source code access is provided, whether in open source or closed source environments for white box security tests. Recent studies have demonstrated AI's significant efficacy in scanning code snippets to identify security flaws, in several cases surpassing state-of-the-art SAST[1] tools. This application of AI can accelerate the vulnerability detection process and minimize the likelihood of human error, ensuring that even subtle or complex vulnerabilities are not overlooked.

Summarization: After scanning, it is crucial to summarize results based on high-priority vulnerabilities and critical findings. AI models can correlate scan data with information gathered during reconnaissance, providing a richer context for prioritizing vulnerabilities. Based on this comprehensive analysis, suggestions for further investigations or remedial strategies are generated.

Prioritization: AI's proficiency in deductive reasoning and result interpretation proves valuable in prioritizing vulnerabilities by risk magnitude or ease of exploitation. It swiftly digests vast amounts of text data, such as tester tool logs, significantly speeding up the assessment process compared to traditional or semi-automated approaches. Benchmark tests have shown that LLMs can accurately assess and prioritize vulnerabilities based on their exploitability.

After analyzing the vulnerabilities, a security tester, like an attacker, moves on to the exploitation phase.

Exploitation

The objective of the exploitation phase is to test how effectively identified security weaknesses can be capitalized on in real-world scenarios under controlled conditions. By simulating attacks, security testers can evaluate the effectiveness of security controls and measure the potential depth of an attacker's penetration within the organization's defenses if the scope allows. This often involves chaining multiple vulnerabilities across different systems to break through significant layers of barriers along the attack path. The attack plan and its opportunity assessment determine the choice of tools and techniques, which may include exploit tools, custom scripts, and social engineering tactics.

[1] Static Application Security Testing

The challenge is determining the most effective paths to attack. AI can mitigate these challenges by offering significant advantages in context-based planning, lining up vulnerabilities, suggesting potentially suitable combinations thereof, and identifying optimal attack paths, thereby making the exploitation process more efficient and potentially more effective. AI can also help testers adapt to dynamic environments by quickly analyzing and responding to changing circumstances, adjusting attack strategies in real time, and even autonomously exploiting zero-day vulnerabilities.

Exploitation Planning: AI systems can automatically analyze extensive security repositories and exploit databases to generate tailored test cases for specific systems and configurations. This reduces the effort required for manual test case creation and ensures comprehensive coverage. For instance, GPT-3.5 suggested realistic and feasible attack vectors like password spraying and Kerberoasting during high-level task planning.

Network Traffic Analysis and Exploitation: LLM-powered hackbots have examined network traffic, identified potential vulnerabilities, and proposed modifications to HTTP requests for exploitation.

Malware Development: AI can help develop new strains of malware that can evade traditional detection methods by obfuscating malware payloads, generating polymorphic code, and identifying potential indicators of compromise (IOCs) that need to be avoided.

Proof of Concept and Exploit Development: LLM-powered technologies can be leveraged to generate exploits and proof-of-concept scripts based on detected vulnerabilities. These proofs of concept demonstrate and explain vulnerabilities' exploitability, validating their existence and impact.

Fuzz Testing: In fuzz testing, AI can be leveraged to generate diverse inputs, potentially uncovering vulnerabilities that traditional methods miss. While AI-generated inputs can offer unique advantages through their adaptability and synthesis of information from various sources, traditional rule-based approaches may provide more consistent and predictable results. Combining both methods offers a more comprehensive approach to fuzz testing, with AI generating a wider range of inputs and rule-based methods ensuring thorough coverage of known edge cases. Additionally, LLMs can contribute to this hybrid approach by, for example, generating or selecting contextually relevant wordlists or patterns.

Social Engineering Simulations: AI can craft realistic phishing emails and social media messages and impersonate individuals, testing employee awareness and an organization's robustness against social engineering attacks. This application of AI helps identify potential human-factor vulnerabilities within security practices.

Boosting Creativity and Innovation: Security testers might overlook specific attack paths. AI can support them in exploring unconventional tactics and identifying creative ways to exploit vulnerabilities, pushing the boundaries of security testing.

Interactive Exploitation: AI-driven tools like PentestGPT guide the execution of exploitation tasks by generating intuitive commands for various security tools tailored to specific scenarios and interpreting their outputs. PentestGPT has proven efficacy in easy- to medium-difficulty challenges on HackTheBox, leading to a reported ranking among the top 1% of players in a community of over 670,000 members. Another study utilizes LLM-based systems to execute and refine attack commands on vulnerable virtual machines through SSH. For instance, the system could escalate privileges by exploiting misconfigurations in the sudoers file,[2] demonstrating AI's practical application in real-world penetration testing. ExploitFlow facilitates Game Theory and AI to produce and represent the exploitation process as dynamic attack trees, capturing the system's state at every process step.

Autonomous Exploitation: The highest level of automation is achieved using AI Agents that autonomously exploit targets. LLM-powered systems have demonstrated high capability in autonomously exploiting identified vulnerabilities, including SQL injections and XSS attacks. Furthermore, AI Agents have been reported to autonomously and without any Offensive Security specific training or fine-tuning to exploit one-day vulnerabilities with an 87% success rate in the performed tests, given relevant CVE information. A follow-up to that paper also showed that teams of AI Agents can autonomously exploit zero-day vulnerabilities. Further, the Wintermute tool has been shown to autonomously identify and execute complex privilege escalation strategies within Linux environments.

While LLM-enabled AI agents can significantly benefit security testers, this phase is often a hybrid of AI and human interaction to supervise autonomous agents and ensure they don't exceed their scope and compromise organizational security.

Reporting

The objective of the reporting phase is to compile a comprehensive report detailing the entire engagement process. Activities include summarizing discovered vulnerabilities, attempted as well as successful exploits, potential impact, and recommended remediation measures. Tools and techniques involve documentation tools to create detailed reports that provide actionable recommendations.

The challenge is producing high-quality documentation that is consistent across multiple security testers and tailored to the specific target audience to ensure they can understand and act upon it. AI can mitigate these challenges by automating the report generation process, ensuring thorough, consistent, and accurate documentation tailored to the respective audience while providing actionable insights for future security improvements. Additionally, AI can assist in effective communication by translating technical findings into actionable recommendations and delivering concise reports that resonate with diverse audiences, bridging the knowledge gap between security testers and stakeholders.

Automated Reporting: AI generates comprehensive reports on offensive security engagements by summarizing findings, prioritizing risks, and recommending remediation steps. AI-powered proofreading can further automate the QA process. This saves security teams valuable time and ensures clear communication with stakeholders.

Visualization: When integrated with visualization tools, AI can enhance reporting using advanced graphical data representations, making findings more accessible and actionable for stakeholders. For example, AI can analyze vulnerabilities to generate threat landscape diagrams, interactive visualizations depicting the attack surface, relationships between vulnerabilities, and critical attack paths. These visual tools improve report clarity and help stakeholders understand and prioritize risks effectively.

[2] https://attack.mitre.org/techniques/T1548/003/The sudoers file, /etc/sudoers, describes which users can run which commands and from which terminals.

Data-Driven Insights: AI analyzes results to identify trends and patterns. This data refines future Offensive Security efforts and prioritizes security investments for maximum impact.

Generate Remediation Instruction: AI has advantages over human experts in relating vast amounts of knowledge to generate remediation instructions.

Now that we know how AI can help in Offensive Security, let's examine how threat actors already use it.

Threat Actor’s Use of AI

While we have explored the use of AI in offensive security through the lens of a security tester or researcher and some of the challenges they can overcome with the possible use of technology, we’ll be remiss not to view it through the lens of threat actors and how they are leveraging the technology. This gives us another reason to consider looking into AI for offensive security. Oftentimes, tactics used by these malicious actors overlap with methods that offensive security testers employ to identify vulnerabilities.

Threat actors are actively using AI to enhance their operations, as highlighted in a joint effort by Microsoft and OpenAI.

AI-Assisted Reconnaissance: Threat actors use AI to automate the gathering and analysis of data on technologies and vulnerabilities, significantly enhancing their reconnaissance capabilities. AI enables them to quickly process large volumes of information, precisely identifying potential targets.

AI-Powered Social Engineering: Leveraging AI, threat actors generate context-specific, convincing phishing content. By analyzing publicly available information about individuals, such as their professional backgrounds or interests, AI can craft personalized phishing emails or messages that are more likely to deceive recipients.

Malicious Code Writing: Employing AI to aid in developing and refining malicious scripts and malware, lowering the technical barrier for complex cyberattacks.

Vulnerability Research: Threat actors utilize AI to understand and identify publicly reported vulnerabilities in software and systems. AI can analyze security reports, and patch notes, and exploit databases to find exploitable weaknesses.

Bypassing Security Features: AI is employed to overcome security mechanisms like two-factor authentication or CAPTCHA. This enhances the ability to automate spam attacks and create large-scale fraudulent accounts and online profiles.

Anomaly Detection Evasion: Another tactic involves developing methods with AI to help malicious activities blend in with normal behavior or traffic. By mimicking legitimate patterns, AI helps evade detection systems, making it harder for security teams to identify and mitigate threats.

Operational Command Refinement: AI refines command and control operations, making post-compromise activities more sophisticated and harder to detect. Threat actors utilize AI to optimize command sequences, improve remote control of compromised systems, and manage data extraction processes more effectively.

By understanding and counteracting how threat actors use AI, security professionals can better protect their organizations and stay one step ahead in the ongoing battle against cyber threats.

AI-Powered Offensive Security in the Near Future

As illustrated by various examples above, AI can significantly enhance Offensive Security capabilities by increasing **scalability**, **efficiency, speed,** and the discovery of **more intricate vulnerabilities**. The level of AI's autonomy varies across these examples, with recent research indicating a progression toward greater autonomy. As AI continues to evolve, we can anticipate even higher levels of autonomy and automation, further bolstering its support across various functions within offensive security.

Lowering Barriers to Entry

AI’s introduction and increased agency mean that the entry barrier to Offensive Security is lowered. This democratization allows more individuals and businesses to participate in security testing without requiring deep expertise in vulnerabilities or techniques. For example, tools like OpenAI's GPT-4o can generate attack scripts, allowing less experienced security professionals to conduct sophisticated security testing. This democratization means more organizations can adopt robust offensive security practices. AI can automate information collection and execute standard exploits, enabling users to perform complex attacks more easily. Numerous studies outside Offensive Security have shown that lower performers benefit most from AI today, indicating that AI can empower a broader range of users to contribute to security testing efforts.

Impact on Professional Security Testers

AI’s capabilities can significantly change how professional security testers operate. By automating time-consuming tasks such as data collection, vulnerability scanning, and initial exploit attempts, AI allows security testers to focus on more strategic, creative, and complex aspects of security testing. This shift enables professionals to engage in more profound analytical work and develop new testing methodologies that better mimic sophisticated cyber threats. In essence, AI enhances productivity, enabling security testers to uncover and address more complex vulnerabilities that would be difficult to detect manually. However, this will require Offensive Security teams to develop new skills in areas such as AI model training, data preparation, and algorithm optimization to effectively leverage AI tools and techniques.

Shift-Left Offensive Security

With increased automation and shorter feedback cycles in Offensive Security, these activities can be integrated earlier in the DevSecOps process. This shift-left approach means that security considerations are embedded from the beginning of the software development lifecycle, resulting in a more proactive and fundamental impact on a business’s overall security posture. By identifying and mitigating vulnerabilities earlier, organizations can reduce the risk of security breaches and ensure more robust protection.

Maturing AI Solutions

Currently, no single AI solution on the market can revolutionize Offensive Security by itself. However, in the near future, we will likely see more commercial security solutions enabling and augmenting security testers to a much larger extent. These systems will likely integrate seamlessly with external systems and sources, such as threat intelligence feeds, social media, and dark web sources. Such integrations will enable AI-driven offensive operations to leverage real-time data on emerging vulnerabilities, exploits, and threat actor activities. By learning from these external intelligence sources, AI systems can enhance their attack simulations, making them more effective and up-to-date with the latest adversarial tactics.

Increasing Autonomy of AI Agents

As AI systems mature, the autonomy of AI agents in Offensive Security will increase rapidly. These autonomous agents will be capable of conducting more complex offensive operations with minimal human intervention, making real-time decisions, and adapting their strategies based on the evolving security landscape. This evolution will allow AI agents to perform tasks that previously required significant human expertise and oversight, further enhancing the efficiency and effectiveness of offensive security operations.

Balancing Automation and Human Oversight

Despite AI advancements, balancing automation and human oversight is currently beneficial. Human oversight ensures that AI's decisions are validated and unintended consequences are mitigated. While some studies outside Offensive Security have shown that AI alone can sometimes perform better than human-AI collaboration (e.g., AMIE), a balanced approach that leverages the strengths of both AI and human expertise should yield the best results in the near future. This combination enhances the effectiveness of security measures while maintaining the integrity and trustworthiness of cybersecurity practices.

Adversarial attacks

All advantages listed above for Offensive Security are applicable to malicious use as well. Security testers must also consider the potential for adversaries to leverage AI capabilities in their attacks. To keep the attack simulations realistic and effective, security testers need to stay current with the latest AI advancements used by adversaries. This necessitates adapting and evolving their techniques and tools at least as quickly as the adversaries do. Security testers must work with AI and continuously integrate new AI capabilities into their offensive security practices to anticipate and counteract sophisticated threats.

As we look towards a future where AI's agency in Offensive Security continues to grow, achieving a balance between automation and human insight will be essential. This approach enhances the effectiveness of security measures and maintains the integrity and trustworthiness of cybersecurity practices.

Challenges and Limitations - Risks and Mitigations

Utilizing AI in offensive security presents unique challenges and limitations. Managing large datasets and ensuring accurate vulnerability detection are significant challenges that can be addressed through technological advancements and best practices. However, inherent limitations, such as token window constraints in AI models, require careful planning and mitigation strategies.

While AI can enhance the efficiency and effectiveness of security measures, these complexities underscore the importance of careful integration of AI into security frameworks. Robust training and validation of AI models are crucial to ensure their reliability and performance. Moreover, stringent ethical guidelines must govern the use of AI to prevent misuse and ensure responsible application. By addressing these challenges and limitations, organizations can leverage AI to strengthen their offensive security capabilities while maintaining ethical standards and operational integrity.

Technical Challenges and Limitations

When integrating AI into Offensive Security, several direct issues can arise related to the technology's capabilities, configurations, and performance. General technical risks associated with AI integration are comprehensively outlined in resources such as OWASP's Top 10 for Large Language Model Applications and will not be elaborated on here.

Challenge/ Limitation	Description	Risk	Mitigation
Token Window Limitations	AI models have a finite capacity for processing data in a single request, making analyzing large documents cumbersome and inefficient.	Degraded performance	Utilize AI models with large context window lengths, like Google Gemini 1.5 Pro with 2 million tokens, or use data chunking techniques, like map-reduce.
Guardrails and Content Filtering	Existing public AI models have built-in restrictions that prevent them from responding effectively to specific prompts that could be deemed harmful or inappropriate.	Limited utility in dynamic security scenarios	Utilize models with flexible content filtering or develop custom models incorporating adjustable operational guidelines tailored to specific security needs without causing unintended harm or exceeding established boundaries.
Lack of Domain Knowledge	AI may not possess sufficient domain-specific knowledge for specialized security tasks.	Limited effectiveness in security scenarios	Use Retrieval-Augmented Generation (RAG) to incorporate relevant domain knowledge. Optionally enrich AI training with domain-specific data.
Hallucinations	An LLM can provide factually incorrect information that sounds plausible.	Making decisions based on information that looks true but isn't	Augmenting AI outputs with human oversight and cross-referencing against verified data sources.
Data Leakage	Inadvertently incorporating sensitive data into AI model training.	Exposing critical information	Enhance with automated data scrubbing tools and regular audits to

			identify and remove sensitive data or use self-hosted models.
False Positives	AI systems may incorrectly flag vulnerabilities.	Wasted resources and disruption to testers due to unnecessary investigations	Incorporating feedback loops into the AI system to adapt and refine its vulnerability detection algorithms can minimize false positives.
False Negatives	Conversely, AI may incorrectly miss vulnerabilities.	Undetected security vulnerabilities and potential breaches	Regularly updating the AI with the latest threat data and real-time anomaly detection techniques can enhance performance.
Loss of Scope Control	AI might autonomously expand the target list beyond the initially intended scope.	Testing unauthorized systems or exceeding permissions could lead to legal or ethical violations, data breaches, or damaging critical systems.	Adding more stringent approval processes and checks can ensure actions remain within authorized boundaries
Compromised Stealth	The detectability of AI-driven activities could be higher than expected.	The effectiveness of red teaming exercises could be compromised if the target systems quickly detect the AI's actions.	Tune the AI-powered system to keep below detection levels by limiting the tool use and speed.
Collateral Damage	AI could create self-propagating malware.	Disruption beyond the intended target due to uncontrolled self-propagating malware	Conduct pre-deployment impact assessments to predict and prevent damages.
Training Data Poisoning	Malicious alterations to training datasets.	Degraded performance and inaccurate results could lead to missed threats or wasted resources.	Constantly monitor for unusual model outputs that may indicate data tampering.
Explainability and Transparency	AI models' decision-making processes can be opaque, making it difficult to understand how they arrive at conclusions in security testing.	Unidentified biases or errors within the AI model's decision-making process lead to missed threats or wasted resources due to a lack of trust in the AI's reasoning.	Implement techniques such as feature importance scoring and model auditing to make AI decision-making processes more understandable and traceable.

Table 3: Technical Challenges and Limitations

Non-Technical Challenges and Limitations

These include broader organizational, ethical, or strategic concerns that impact the deployment and operation of AI.

Challenge/ Limitation	Description	Risk	Mitigation
Data privacy regulations limitations	Regulations restrict sending data to cloud-based AI services.	Limited access to powerful AI models hinders innovation and efficiency.	Explore on-premises AI, privacy-preserving techniques, or data anonymization methods.
Cost Concerns	Developing, training, and operating specialized AI models is expensive.	Not employing AI or at least only in a very limited capacity	Use pre-trained models, open-source tools, and cloud-based AI services to reduce costs.
Ethical Violations	AI could overstep social engineering scope or moral boundaries.	Socially unacceptable or manipulative behavior.	Regular training and updates on ethical guidelines for the AI and the team can enhance adherence.
Over-Reliance on AI	Relying on AI-aided Offensive Security to the point where other essential security practices are ignored.	Neglecting other essential security practices.	Regular drills and scenario training involving AI and human elements to ensure readiness and capability without over-reliance.
High-Value Targeting	Custom AI offensive security systems can become high-value targets for attackers.	Misusing unauthorized access to cause harm.	Implement strict access control, anomaly detection, and fail-safe mechanisms.

Table 4: Non-Technical Challenges and Limitations

While AI offers significant potential to enhance offensive security capabilities, it's crucial to acknowledge the challenges and limitations and the risks they present. Implementing appropriate mitigation strategies can help ensure AI's safe and effective integration into their security frameworks.

Governance, Risk, and Compliance (GRC)

Leveraging AI for offensive security requires an integrated approach to governance, risk, and compliance (GRC). This ensures that AI tools are used effectively and ethically, following frameworks like the NIST AI Risk Management Framework (NIST AI RMF), AI Organizational Responsibilities - Core Security Responsibilities, and the OWASP LLM AI Cybersecurity & Governance Checklist. These frameworks provide valuable assessment criteria encompassing safety, security, resilience, explainability, privacy enhancement, fairness, and transparency.

Trustworthy AI Implementation

- Safe, Secure, & Resilient: AI models must prioritize safety throughout their lifecycle, from design to deployment and decision-making. Tools like Microsoft Azure Machine Learning exemplify this by integrating robust security and resilience measures at each stage.
- Explainable & Interpretable: Effective offensive security requires understanding AI outputs to ensure valid remediation of identified issues. Tools like IBM's Watson for Cyber Security offer explainable AI (XAI) outputs detailing the reasoning behind detected threats, thereby aiding security analysts in validating and responding to AI findings.
- Privacy-Enhanced: AI models must protect individual privacy by ensuring identity, anonymity, and confidentiality. . For instance, Google's AI systems comply with GDPR, ensuring data privacy while analyzing vast amounts of data for potential threats.
- Fair: AI models must reduce biases to ensure effectiveness in diverse environments. Offensive security tools like Google's AutoAI and employ bias detection mechanisms to ensure fair and unbiased security threat assessments.
- Transparent: AI systems that are transparent allow security teams to understand and trust AI decisions. Transparent models, like those used by Palo Alto Networks, provide clear insights into AI's decision-making processes in threat detection and response.

Third-Party Risk Management

When using third-party AI models for offensive security, supplier evaluation is critical. This involves assessing data security practices, model training data, and potential information leaks.

- Adherence to Standards: Evaluate suppliers' adherence to security and privacy standards like ISO/IEC 42001:2023 and ISO/IEC 27001 and GDPR. Tools like IBM QRadar help comply with various standards while providing AI-driven offensive security threat detection.
- Security and Privacy Certifications: Request certifications for hosted AI solutions to ensure they meet regulatory requirements. For example, Amazon Web Services (AWS) AI services often provide necessary certifications and compliance assurances.
- Contractual and Insurance Coverages: Address specific risks related to the offensive security AI use case through contractual agreements and insurance coverages. This ensures that any potential issues are mitigated through legal and financial protections.

GRC Considerations

- Legal Frameworks: Understanding legal and regulatory requirements for AI in offensive security is essential. For example, AI models must comply with regional laws such as Europes GDPR and EU AI Act or Californias CPRA (CCPA as amended).

- Ethical Guidelines: AI governance must include ethical considerations. Using resources like the UNESCO Recommendation on the Ethics of AI and IEEE EAD standards can provide additional guidance. Establishing a culture of ethical AI use is crucial for maintaining trust and integrity in offensive security operations.
- Organizational Policies: Develop clear policies governing AI use in offensive security. These should complement existing legal and ethical frameworks and include specific playbooks for practical implementation.

GRC Summary

Introducing AI-aided offensive security requires careful assessment to avoid unintentionally expanding other risks. A full risk assessment aligned with the NIST AI RMF can help mitigate identified risks. This includes considering potential conflicts between the AI model's functionalities and privacy policies and preventing the exposure of sensitive information to unauthorized parties.

By following established frameworks and guidelines, organizations can ensure that AI-aided offensive security implementations are safe, ethical, and compliant with regulatory standards, enhancing their overall security posture.

Conclusion

Artificial intelligence (AI) is rapidly advancing, bringing enhanced agency and automation. These developments offer new opportunities and present challenges for offensive security teams globally. Malicious actors operating outside the bounds of legal and ethical frameworks are already exploiting these advancements, highlighting the critical need for defenders to innovate proactively.

As detailed in this paper, AI technologies, particularly LLMs and AI Agents, significantly enhance offensive security by automating and scaling tasks. This improvement boosts efficiency, allows for more sophisticated and extensive assessments, and enables security teams to focus on process improvement and strategic work. Moreover, AI democratizes security testing, lowering entry barriers and addressing the shortage of skilled professionals.

Despite these benefits, no single AI solution can revolutionize offensive security practices today. Therefore, a multifaceted approach is necessary, involving continuous experimentation with AI to find and implement effective solutions. This requires creating an environment that encourages learning and development, enabling team members to grow their skills with AI tools and techniques.

To leverage AI effectively, organizations must share best practices and develop custom tools tailored to specific security needs, ideally through interdisciplinary collaboration between departments such as Data Science, Cybersecurity, Legal, etc. This ensures a holistic approach to AI integration and risk management. As AI systems integrate into workflows, they introduce new technical and organizational challenges that must be managed carefully. Vigilance is required to prevent AI-driven tools from being misused or behaving unpredictably.

Promoting a culture of responsible AI use is essential to ensure teams understand the risks and impacts of integrating AI into offensive security. This includes establishing a robust Governance, Risk, and Compliance (GRC) framework to manage these risks effectively.

In conclusion, offensive security must evolve with AI capabilities. By adopting AI, training teams on its potential and risks, and fostering a culture of continuous improvement, organizations can significantly enhance their defensive capabilities and secure a competitive edge in cybersecurity.

References

1. Jason Wei et. al. (2022), Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, https://arxiv.org/abs/2201.11903
2. Yuheng Cheng et. al. (2024), Exploring Large Language Model based Intelligent Agents: Definitions, Methods, and Prospects, https://arxiv.org/abs/2401.03428
3. Zachary Proser (2023), Retrieval Augmented Generation (RAG), https://www.pinecone.io/learn/retrieval-augmented-generation
4. Ge Wang (2019), Humans in the Loop: The Design of Interactive AI Systems, https://hai.stanford.edu/news/humans-loop-design-interactive-ai-systems
5. Sheng Lu et. al. (2023), Are Emergent Abilities in Large Language Models just In-Context Learning?, https://arxiv.org/abs/2309.01809
6. Chris Sullo (2001),Nikto, https://github.com/sullo/nikto
7. Gelei Deng et al. (2023), PentestGPT: An LLM-empowered Automatic Penetration Testing Tool, https://arxiv.org/pdf/2308.06782.pdf
8. Eric Hilario et. al. (2024), Generative AI for pentesting: the good, the bad, the ugly, https://link.springer.com/article/10.1007/s10207-024-00835-x
9. TheR1D (seen 2024/06), ShellGPT, https://github.com/TheR1D/shell_gpt
10. Significant-Gravitas (seen 2024/06), AutoGPT, https://github.com/Significant-Gravitas/AutoGPT
11. Andreas Happe, Jürgen Cito (2023), Getting pwn'd by AI: Penetration Testing with Large Language Models, https://arxiv.org/pdf/2308.00121.pdf
12. aress31 (seen 2024/06), BurpGPT, https://github.com/aress31/burpgpt
13. Legit Security Team (2023), Using AI to Reduce False Positives in Secrets Scanners, https://www.legitsecurity.com/blog/using-ai-to-reduce-false-positives-in-secrets-scanners
14. Jingyue Li et.al. (2023), Evaluating the Impact of ChatGPT on Exercises of a Software Security Course, https://arxiv.org/pdf/2309.10085.pdf
15. Shashwat et.al. (2024), A Preliminary Study on Using Large Language Models in Software Pentesting, https://arxiv.org/abs/2401.17459
16. Andreas Happe et. al. (2023), LLMs as Hackers: Autonomous Linux Privilege Escalation Attacks, https://arxiv.org/pdf/2310.11409
17. Joseph Thacker (seen 2024/06), Hero, https://github.com/jthack/hero
18. Keith McCammon (2024), How AI will affect the malware ecosystem and what it means for defenders, https://redcanary.com/blog/opinions-insights/ai-malware
19. Dijana Vukovic Grbic, Igor Dujlovic (2023), Social Engineering with ChatGPT, https://www.researchgate.net/publication/369970449_Social_engineering_with_ChatGPT
20. Víctor Mayoral Vilches et. al. (2023), ExploitFlow, cyber security exploitation routes for Game Theory and AI research in robotics, https://arxiv.org/pdf/2308.02152.pdf
21. Richard Fang et. al. (2024), LLM Agents can Autonomously Hack Websites, https://arxiv.org/html/2402.06664v1
22. Richard Fang et. al. (2024), LLM Agents can Autonomously Exploit One-day Vulnerabilities, https://arxiv.org/abs/2404.08144
23. Richard Fang et. al. (2024), Teams of LLM Agents can Exploit Zero-Day Vulnerabilities, https://arxiv.org/abs/2406.01637

24. Microsoft Threat Intelligence (2024), Staying Ahead of Threat Actors in the Age of AI, https://www.microsoft.com/en-us/security/blog/2024/02/14/staying-ahead-of-threat-actors-in-the-age-of-ai/
25. Ethan Mollick - One Useful Thing (2024), Centaurs and Cyborgs: On the Jagged Edge of AI and Human Collaboration, https://www.oneusefulthing.org/p/centaurs-and-cyborgs-on-the-jagged
26. Karthikesalingam, Natarajan (2024), AMIE: A research AI system for diagnostic medical reasoning and conversations, https://research.google/blog/amie-a-research-ai-system-for-diagnostic-medical-reasoning-and-conversations/
27. OWASP (2023), OWASP Top 10 for Large Language Model Applications, https://owasp.org/www-project-top-10-for-large-language-model-applications/
28. NIST (2024), AI Risk Management Framework, https://www.nist.gov/itl/ai-risk-management-framework
29. OWASP (2024), LLM AI Security and Governance Checklist, https://owasp.org/www-project-top-10-for-large-language-model-applications/llm-top-10-governance-doc/LLM_AI_Security_and_Governance_Checklist-v1.pdf
30. Microsoft Azure Team (2024), What is Responsible AI?, https://learn.microsoft.com/en-us/azure/machine-learning/concept-responsible-ai?view=azureml-api-2
31. IBM (seen 2024/06), What is Explainable AI, https://www.ibm.com/topics/explainable-ai
32. Google Cloud Team (seen 2024/06), Google Cloud GDPR Compliance, https://cloud.google.com/privacy/gdpr
33. GDPR.eu (2018), General Data Protection Regulation (GDPR), https://gdpr.eu/tag/gdpr/
34. IBM (seen 2024/06), Accelerating AI and model lifecycle management, https://www.ibm.com/products/watson-studio/autoai
35. Palo Alto Networks (seen 2024/06), SmartScore, https://www.paloaltonetworks.com/blog/tag/smartscore/
36. ISO (2022), ISO/IEC 42001:2023 Information technology — Artificial intelligence — Management system, https://www.iso.org/standard/81230.html
37. ISO (2022), ISO 27001, https://www.iso.org/standard/27001
38. IBM (seen 2024/06), Compliance with IBM Security QRadar SIEM, https://www.ibm.com/products/qradar-siem/compliance
39. AWS (seen 2024/06), Security Perspective Compliance and Assurance of AI/ML Systems, https://docs.aws.amazon.com/whitepapers/latest/aws-caf-for-ai/security-perspective-compliance-and-assurance-of-aiml-systems.html
40. McKinsey (2021), Getting to Know and Manage Your Biggest AI Risks, https://www.mckinsey.com/capabilities/quantumblack/our-insights/getting-to-know-and-manage-your-biggest-ai-risks
41. Council of the EU (2024), Artificial intelligence (AI) act: Council gives final green light to the first worldwide rules on AI, https://www.consilium.europa.eu/en/press/press-releases/2024/05/21/artificial-intelligence-ai-act-council-gives-final-green-light-to-the-first-worldwide-rules-on-ai/
42. California Office of the Attorney General (2024), CCPA, https://oag.ca.gov/privacy/ccpa
43. UNESCO (2022), Recommendation on the Ethics of AI, https://unesdoc.unesco.org/ark:/48223/pf0000381137
44. IEEE (seen 2024/06), EAD General Principles, https://standards.ieee.org/wp-content/uploads/import/documents/other/ead_general_principles.pdf