The Impacts of AI on Application Security Testing

kiuwan
An Idera, Inc. Company

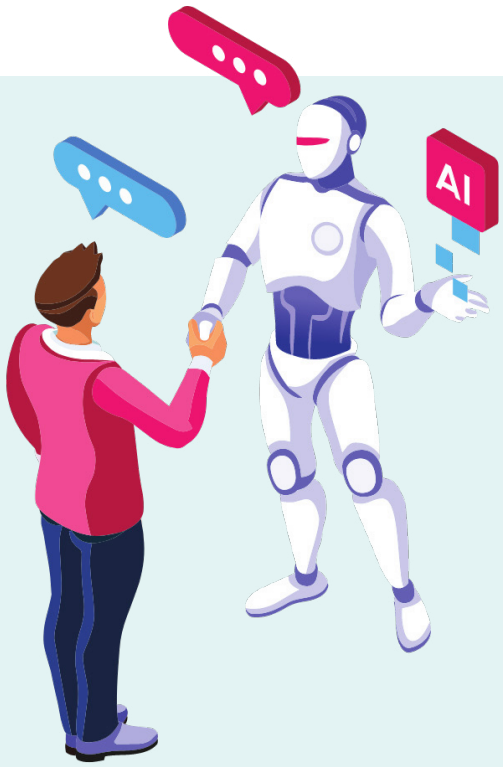
INDEX

Overview

As technology continues to evolve at an unprecedented pace, the landscape of application testing is constantly on the edge of new developments, particularly regarding the integration of artificial intelligence (AI) into testing processes.

Malicious actors are quick to embrace new technology, and AI is no exception. To stay ahead of hackers and secure all phases of the software development lifecycle (SDLC), organizations of all sizes must integrate automated, AI-based testing solutions into a comprehensive security approach for their DevSecOps teams to mitigate increasing security risks.

This ebook will provide an overview of the latest AI technologies in software development and guide stakeholders in harnessing the full potential of AI for the future of application testing. By understanding both the positive and negative potential of AI, organizations can implement this technology to enhance the safety, accuracy, and reliability of their products.



The AI market is projected to reach a staggering ***$407 billion by 2027****, experiencing substantial growth from its estimated $86.9 billion revenue in 2022.*

*Forbes

Recent Developments in Artificial Intelligence

The history of AI is characterized by periods of intense excitement and innovation followed by periods of disillusionment. The field continues to evolve, and its impact on society is profound and ever-growing.

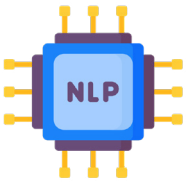
The seeds of AI were sown in the early 20th century with the development of formal logic and computational theory. Alan Turing, a British mathematician, introduced the concept of a universal machine that could simulate any computable problem, later termed the Turing Machine. However, the limitations of expert systems, along with high costs and unmet expectations, led to an AI winter marked by reduced funding and skepticism.

It wasn't until the 1990s that AI shifted toward data-driven approaches. The focus moved from rule-based systems to learning from data. Algorithms like decision trees, neural networks, and support vector machines became popular. In the 2010s, with the advent of big data and advancements in computing power, a subset of machine learning called deep learning gained prominence. Deep neural networks started outperforming other algorithms in tasks such as image and speech recognition.

These advances led to significant progress in accessible AI tools, which are increasingly integrated into everyday applications.

Deep Learning and Neural Networks

One of the most significant advancements in AI has been in deep learning, particularly through the use of deep neural networks. These networks, with multiple layers of neurons, are remarkably effective in tasks such as image and speech recognition. For example, the development of Convolutional Neural Networks (CNNs) has revolutionized image classification and object detection.



Natural Language Processing (NLP)

Major strides in NLP have enabled machines to better understand and process human language. Transformer architectures, such as BERT (Bidirectional Encoder Representations from Transformers), have set new benchmarks in various NLP tasks, such as question answering and sentiment analysis.

Generative Models

Generative Adversarial Networks (GANs) are a class of AI models used to create new data that is similar to the training data. They have been used to generate realistic images, music, and even text. GANs have applications in art, gaming, and more.

Reinforcement Learning

Reinforcement Learning (RL) has made headlines, especially in the context of game-playing AI. Notably, AlphaGo, developed by DeepMind, defeated the world champion Go player in 2016. This was a landmark moment, as Go is a highly complex game that was considered beyond the reach of AI.

Transfer Learning and Few-Shot Learning

These techniques involve the reuse of pre-trained models on new, but related tasks with smaller datasets. This has allowed for the more efficient training of AI models, saving time and resources.

AI in Healthcare

AI has made significant inroads into healthcare, with applications such as disease diagnosis — diagnosing cancer from medical images — drug discovery, and personalized medicine.

Autonomous Systems

The development of autonomous vehicles is a major application of AI. Companies like Tesla and Waymo have been at the forefront of developing self-driving technology.

AI Ethics and Fairness

As AI systems are increasingly deployed, concerns around bias, fairness, and ethics have gained prominence. Research in explainable AI (XAI) aims to make AI models more transparent and understandable.

Edge AI

The optimization of AI algorithms to run on edge devices like smartphones and IoT devices has been a growing trend. This allows for real-time processing and decision-making without communicating with a central server.

Large Language Models

The development of extremely large neural networks specialized in processing natural language, such as GPT by OpenAI, has pushed the boundaries of what is possible in natural language generation and understanding.

Application Security *Risks Associated With AI Advancements*

AI has become an increasingly essential tool in the cybersecurity landscape. However, like any technology, AI also comes with risks and potential downsides. Some notable examples are discussed below.

AI-Powered Cyberattacks

As AI tools and techniques become more advanced, there is a growing concern that they could be used to supercharge cyberattacks, leading to serious breaches.

For example, in 2018, researchers from IBM developed DeepLocker. DeepLocker is AI-powered malware developed as a proof of concept to demonstrate how artificial intelligence could be used to create highly evasive and targeted cyberattacks. It was first unveiled at the Black Hat USA conference in 2018. DeepLocker leverages several AI and machine learning techniques to hide its malicious payload until it reaches a specific target.

How DeepLocker Works:

- **Concealing Malicious Payload:** DeepLocker hides its malicious payload in benign carrier applications, such as videoconferencing software. The malware is encrypted so that it is virtually impossible to reverse-engineer or to detect using traditional antivirus or malware scanning tools.
- **Target Identification:** DeepLocker uses AI models to identify its target. The model is trained to recognize specific attributes, such as facial recognition data, geolocation, or voice recognition, which can be used to identify the intended victim.

Trigger Conditions: DeepLocker only releases its malicious payload when the AI model confirms that it has reached the intended target. The trigger conditions could be based on a combination of factors, such as recognizing the face of the target via the webcam, detecting a specific system configuration, or identifying a certain geolocation.

Attack Execution: Once the target is identified and the trigger conditions are met, DeepLocker decrypts and executes its malicious payload. This can range from ransomware attacks to data exfiltration or any other form of cyberattack.

Implications:

Evasion of Detection: DeepLocker represents a new breed of malware that can evade traditional antivirus and malware detection methods. Because the payload is encrypted and hidden within a benign application, security tools have difficulty detecting it.

Targeted Attacks: By using AI to identify specific targets, DeepLocker can pinpoint its attacks. This contrasts with traditional malware, which often targets systems indiscriminately.

Increased Difficulty in Attribution: The highly evasive nature of DeepLocker makes it difficult to attribute attacks to specific actors. This could be exploited by state-sponsored actors or cybercriminals looking to avoid detection.

Need for Advanced Security Solutions: DeepLocker exemplifies the need for more advanced security solutions that go beyond traditional antivirus software. This includes employing AI and machine learning in defensive cybersecurity solutions to detect and counter AI-powered malware.

Data Poisoning Attacks

Data poisoning attacks involve manipulating the training data used by machine learning models in order to corrupt the output of the models. These attacks can have serious consequences, especially when the affected models are used in critical systems like autonomous vehicles, fraud detection systems, or recommendation engines.

How Data Poisoning Attacks Work:

Injection of Malicious Data: In a data poisoning attack, an adversary injects malicious data into the training dataset used by a machine learning model. This can be done by directly manipulating the data or by influencing the data collection process.

Model Training: The machine learning model is then trained using the manipulated dataset. Because the dataset includes malicious data, the model may learn incorrect or harmful associations.

Model Deployment: Once the model is deployed, it may produce incorrect or unintended outputs due to the poisoned data it was trained on. This can be exploited by an attacker.

Types of Data Poisoning Attacks:

Targeted Manipulation: The attacker may manipulate the training data to cause the model to produce specific outputs for certain inputs. For example, an attacker might poison a spam detection model to ensure that their phishing emails are not flagged as spam.

Random Noise Injection: The attacker may inject random noise into the training data to reduce the overall accuracy and reliability of the model. This could be used to undermine trust in the system.

Examples of Data Poisoning Attacks:

Autonomous Vehicle Manipulation: In 2019, Tencent's Keen Security Lab demonstrated that small stickers placed on road signs could cause an autonomous vehicle's computer vision system to misinterpret the signs. This is a form of data poisoning involving the manipulation of real-world data.

Manipulating Recommender Systems: Data poisoning can be used to manipulate recommender systems, such as those used by online retail or streaming services, by injecting biased data. This could cause the system to recommend products or content that it would not have otherwise recommended.

Implications and Mitigation:

Security and Safety Risks: Data poisoning attacks can pose significant security and safety risks, especially when the affected machine learning models are used in critical systems.

Loss of Trust: Successful data poisoning attacks can lead to a loss of trust in AI and machine learning systems, as they may be seen as unreliable or easily manipulated.

Mitigation Techniques: To defend against data poisoning attacks, it is essential to ensure the integrity and quality of training data. This can include using data sanitation techniques, validating the sources of training data, and employing anomaly detection to identify and remove malicious data. Additionally, employing robust learning algorithms that are less sensitive to outliers can help reduce the impact of data poisoning.

Evasion Attacks

Evasion attacks in the context of cybersecurity and machine learning refer to techniques used by adversaries to evade detection by AI-based systems, particularly classifiers. These attacks are designed to exploit the vulnerabilities of machine learning models to bypass security mechanisms.

How Evasion Attacks Work:

- **Understanding the Model:** Attackers first try to understand the machine learning model they want to evade. They might try to learn about the features the model relies on for classification, the decision boundaries, and other characteristics.
- **Crafting Evasive Inputs:** Once the attacker has some understanding of the model, they craft inputs that are designed to be misclassified by the model. For instance, they may slightly alter the properties of malware so that it is no longer recognized as such by an antivirus program that uses machine learning.
- **Exploiting Model Vulnerabilities:** The attacker uses the crafted inputs to exploit the vulnerabilities in the machine learning model, evading detection or causing the model to make incorrect classifications.

Types of Evasion Attacks:

- **Adversarial Examples:** These are inputs to machine learning classifiers that an attacker has intentionally modified to cause the classifier to misclassify them. For example, adding imperceptible noise to an image can cause an image classifier to label it incorrectly.
- **Feature Reduction:** By removing or altering certain features of the malicious data, an attacker may evade detection. For instance, malware can be altered to remove certain signatures or behaviors that would typically trigger detection.
- **Mimicking Benign Samples:** In this type of attack, malicious samples are designed to closely mimic benign samples, such as phishing emails crafted to resemble legitimate emails to evade spam filters.

Examples of Evasion Attacks:

Evading Image Classifiers: Research has shown that by adding carefully crafted noise to an image, it's possible to cause a deep neural network to misclassify the image. For example, in 2018, a group of researchers demonstrated that by altering just a few pixels in an image, they could fool an AI image classifier into believing that an image of a cat was an image of guacamole.

Bypassing Malware Detection: Malware authors often use evasion techniques to bypass machine learning-based malware detection systems. They can encrypt the malware or alter its behavior to make it look benign.

Implications and Mitigation:

Security Risks: Evasion attacks can pose significant security risks, as they allow malicious actors to bypass security measures and potentially gain unauthorized access to systems or data.

Model Robustness: Machine learning models used in security-sensitive applications must be robust against evasion attacks. This might include using techniques like adversarial training, where the model is trained on adversarial examples to improve its resilience to such attacks.

Continuous Monitoring and Updating: Since evasion techniques constantly evolve, security systems should be continuously monitored and updated to adapt to new threats.

Defense in Depth: Relying solely on machine learning models for security can be risky. Employing a defense-in-depth strategy, where multiple layers of security controls are used, can mitigate the risks of evasion attacks.

Automated Exploit Generation

AI can be used to automatically discover and exploit vulnerabilities in software at a scale and speed that would be impossible for human hackers. DARPA's Cyber Grand Challenge in 2016 showcased autonomous systems that could find and exploit software vulnerabilities without human intervention. Automated exploit generation should be of particular interest to DevOps teams.

Here's a brief explanation of how it works:

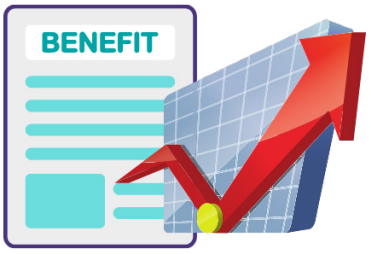
- **Analyzing Code:** AEG systems begin by analyzing the target software's code. They can use static analysis to examine the code without executing it or dynamic analysis to monitor the program's behavior during execution.
- **Identifying Vulnerabilities:** The AEG system searches for security vulnerabilities, such as buffer overflows, use-after-free errors, or injection flaws. It does this by modeling the program's behavior and looking for paths through the code that might lead to exploitable conditions.
- **Generating Exploits:** Once a vulnerability is found, the AEG system tries to create an exploit for it. This typically involves crafting input data that can trigger the vulnerability and cause the program to behave unintendedly, such as executing arbitrary code.
- **Verification:** The system tests the generated exploits to ensure they work as intended. This step is essential for confirming that the exploit can compromise the target software effectively.

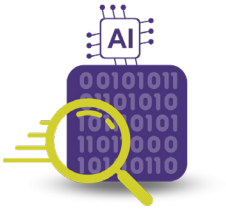
Implications and Use Cases:

- **Cybersecurity Research:** AEG can be used by cybersecurity researchers to discover and patch vulnerabilities more quickly. By automating the process, researchers can analyze a larger set of software and fend off more potential attacks.
- **Offensive Security:** Conversely, malicious actors can use AEG to find and exploit vulnerabilities for illegal or unethical purposes. This highlights the importance of using AEG responsibly and ensuring that found vulnerabilities are disclosed and patched appropriately.
- **Improving Software Security:** Software developers can use AEG as part of their development process to find and fix vulnerabilities in their code before it is released. This can lead to more secure software products.

Benefits of AI for Application Security

AI significantly affects application security by automating and improving various aspects of threat detection, analysis, and response. Some of the best use cases for AI in AppSec are discussed on the next page.





Automated Vulnerability Detection

AI-powered tools can automatically scan the codebase of applications for security vulnerabilities. By using machine learning algorithms, these tools can learn from past data and efficiently identify potential security issues, sometimes even predicting where vulnerabilities are likely to arise.

Enhanced Code Analysis

Traditional static analysis tools often suffer from false positives and false negatives. AI can improve the accuracy of code analysis by learning from historical data, understanding the context, and reducing the noise in the results.

Security Testing Automation

AI can automate security testing processes such as fuzzing and penetration testing. For example, AI-based fuzzing tools can intelligently generate test cases and inputs that are more likely to expose vulnerabilities in the application.

Anomaly Detection

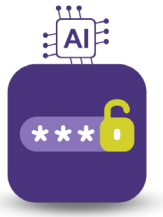
AI algorithms can monitor application logs and user behavior to detect anomalies that may indicate a security breach. By learning what normal behavior looks like, AI systems can quickly identify and alert on behavior that deviates from the norm.

Threat Intelligence and Analysis

AI can be used to aggregate and analyze data from various sources to provide real-time threat intelligence. It can identify patterns and correlations that might indicate emerging threats, helping security teams take proactive measures.

Automated Response and Remediation

In the event of a security incident, AI can help automate response actions. This can include isolating affected systems, blocking malicious IP addresses, or even suggesting code fixes for known vulnerabilities.

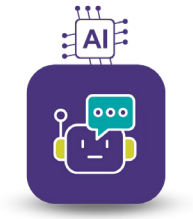


Enhanced User Authentication

AI can enhance user authentication mechanisms by incorporating behavioral biometrics. For example, an AI system can analyze keystroke dynamics, mouse movements, or other behavioral traits to confirm that the person accessing the application is who they claim to be.

Customized Security Policies

AI systems can analyze historical data to understand the specific security needs of an application and tailor security policies accordingly. This can ensure that security measures are not overly restrictive or too lax.



Chatbots and Virtual Assistants for Security

AI-powered chatbots and virtual assistants can be used to facilitate interaction with security systems, allowing developers and security teams to query the security status, receive alerts, or even execute commands through natural language processing.

Training and Awareness

AI can be used to develop training modules and simulations that adapt to the learner's progress and focus on areas where they need the most improvement. This can enhance security awareness among developers and other staff members.

Next Steps in Application Security

Staying current with security measures is indispensable in the ever-evolving technological domain. The application of AI in security frameworks necessitates a dynamic approach to counter emerging threats. One of the most efficient and effective strategies is the implementation of SAST tools (Static Application Security Testing), which play a crucial role in scanning and identifying vulnerabilities in applications' source code.

Kiuwan, a powerful end-to-end application security platform, is an example of a tool that has become invaluable to organizations, CIOs, CTOs, and developers. With its comprehensive toolset that combines SAST, Software Composition Analysis (SCA), and Quality Assurance (QA), Kiuwan empowers teams throughout the development process to identify and remediate vulnerabilities rapidly. As a trusted name for over 20 years, Kiuwan is relied upon by developers at some of the world's leading brands to ensure application security, safeguard critical data, and accelerate time to market.

What sets Kiuwan apart is its ability to detect security vulnerabilities in the source code, enforce coding guidelines, and manage open-source components. This enables developers to eliminate defects and significantly enhance application security. Furthermore, Kiuwan is aligned with prominent security s tandards, including OWASP, CWE, CVE, CPE, and NIST, providing users with the ultimate protection. Kiuwan supports over 30 major programming languages and frameworks, allowing it to cater to a wide range of needs, be it a WordPress vulnerability scanner or a Python code analyzer. This enables organizations to implement a rigorous approach to application security so they can deploy applications with confidence.

Book a demo with Kiuwan today!

*YOU KNOW **CODE**, WE KNOW **SECURITY!***

GET IN TOUCH:

Headquarters
2950 N Loop Freeway W, Ste 700
Houston, TX 77092, USA

United States **+1 732 895 9870**
Asia-Pacific, Europe, Middle East and Africa **+44 1628 684407**
contact@kiuwan.com
Partnerships: **partners@kiuwan.com**