How can artificial intelligence be used to detect and mitigate zero-day vulnerabilities?

By Krishiv Garg

Abstract

A zero-day exploit is a cyberattack that uses unknown or unaddressed security flaws in computer software, hardware, or firmware. Zero-day vulnerabilities pose very significant threats to cyber security. While traditional methods have been effective, they are lacking in many aspects due to rapidly evolving cyber threats. Hence, this paper examines artificial intelligence techniques, including machine learning and their application in enhancing cyber security against zero-day vulnerabilities. The research delves into supervised and unsupervised models and algorithms like Naive Bayes. The findings suggest that effective solutions such as artificial intelligence-driven approaches are crucial in the face of rapidly evolving cyber threats.

Keywords: Artificial Intelligence, cybersecurity, algorithm, zero-day vulnerabilities, exploits

Introduction

A zero-day exploit is a cyberattack that uses unknown or unaddressed security flaws in computer software, hardware, or firmware [1]. Zero-day vulnerabilities are called so because the system owner has had zero days to patch the issue. IBM's X-Force threat intelligence team has recorded 7,327 zero-day vulnerabilities since 1988, which amounts to just three percent of all recorded security vulnerabilities [2]. Nonetheless, zero-day exploits pose a major threat to businesses and vendors, as they are deployed before the vendor is aware of the situation or has the opportunity to develop a patch. Zero-day exploits include malware fishing and even direct attacks on exposed systems, and since there are no existing patches, they can easily bypass conventional security measures.

Zero-day exploits can potentially cause major data breaches, financial loss and even reputational damage to multinational corporations and small businesses. One of the best examples of this would be the attack in 2017, where a hacker group called Shadow Brokers leaked an Eternalblue zero-day exploit to the public that the WannaCry ransomware worm would use. It spread to over 200 thousand computers, including computers and significant corporations like FedEx, Honda, Nissan, and the United Kingdom's National Health Service [3].

The impact of a zero-day attack is not limited to the direct cost and also includes indirect costs related to resources required for detecting and stopping the attack along with system recovery. Cybercriminals have also resorted to more sophisticated zero-day exploits, including a new technique called advanced persistent threat. This technique refers to an attack that secretly continues and stays inside a system for a long time, collecting data [4]. Some notable examples include the Iranian hacker group APT34, the Russian APT28, and many others.

Traditional cybersecurity methods like signature-based methods are ineffective because zero-day vulnerabilities are unknown and lack such signatures. Heuristic-based methods are also ineffective, as they often result in false positives and can lead to alert fatigue among security teams. The simplest, most reliable method of manual analysis of zero-day vulnerabilities is time-consuming and cannot keep up with the pace of new threats.

Contrary to traditional methods, artificial intelligence seems very promising in this field primarily due to its ability to quickly analyse vast amounts of data without exhaustion. By leveraging AI algorithms' ability to recognise patterns and perform predictive analytics, we can significantly enhance cybersecurity and ensure that we can effectively address zero-day vulnerabilities.

Traditional Methods and their flaws

Traditional methods for detecting and mitigating zero-day exploits have significantly evolved over time but often fall short. They can be broadly categorised into signature-based, heuristic-based, and manual detection methods.

Signature-based Method

Signature-based detection is an approach to vulnerability detection in which the program looks for a unique pattern, which may be a byte sequence inside network traffic or in a file. It compares it to known signatures and alerts when a match is made. A very prominent and remarkable work is SVELTE IDS. It is a hybrid IDS (intrusion detection system) that combines signature and anomaly-based detection to protect IoT devices from IPv6 RPL routing attacks. It is effective against sinkhole forwarding, altering information and selective forwarding. It uses a module

called 6LoWPAN Border Router that performs heavy calculations and a myriad of smaller modules for monitoring [5]. Despite being effective, signature-based detection methods have a significant flaw: they can only detect vulnerabilities for which they have the signature of for comparison.

Heuristic-based Method

Heuristic-based detection addresses the problem of the signature-based approach discussed in the previous paragraph by using predefined rules and patterns to identify threats rather than signatures [6]. It can detect mutated malware and zero-day vulnerabilities but is extremely prone to false positives and can lead to alert fatigue. In this phenomenon, programmers stop responding to actual errors due to many false positives/errors. According to the Cisco 2017 Security Capabilities Benchmark Study, organisations can only analyse 56% of the security alerts they receive on a given day due to numerous constraints. Half of the reviewed alarms (28%) are declared authentic, while less than half (46%) of legitimate warnings are remediated. In addition, 44% of security operations managers receive more than 5000 security warnings daily [7].

Manual Method Detection

Unlike the previous two methods discussed, manual detection methods rely on the intuition and expertise of cybersecurity experts; they include methods like log analysis, network traffic analysis, and threat hunting. Although very effective, it doesn't offer 24/7 protection and has the potential for human error.

AI Techniques for Zero-Day Vulnerability Detection

We can use artificial intelligence algorithms and machine learning models to protect ourselves against zero-day vulnerabilities by detecting them early before they can cause actual harm.

Machine Learning Models

Machine learning is a field that allows computers to learn without being programmed explicitly. Machine learning models are increasingly used to detect zero-day vulnerabilities due to their ability to learn from data and pattern identification that can easily indicate the presence of novel threats. Machine learning models have two main learning approaches: supervised and unsupervised, each offering very different applications and advantages in cyber security, especially in threat detection.

Supervised Learning

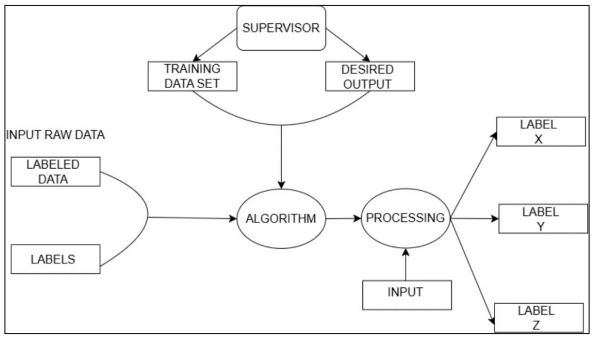

Figure 1: Simplified explanation of supervised Learning Process

As evident from the name, supervised learning is when we teach a machine learning model on a well-labelled data set, where each point is paired with the corresponding label. Eventually, the modern model learns to map input to correct output labels, enabling it to classify and predict new and unseen data based on learned patterns. We can use multiple algorithms, such as decision trees, random forests, support vector machines, and Naive Bayes.

Naive Bayes Theorem for zero-day vulnerability detection

Naive Bayes classifiers are algorithms based on the Bayes theorem:

$$P(A|B) = \frac{P(A)\times P(A)}{P(B)}$$

Where A and B are events and P(B) is not equal to 0:

- This equation will give us the probability of event A, given that event B is true.
- P(A) is the prior probability of A before the value of B is known.
- P(B) is the Marginal Probability.
- P(A|B) is a posteriori probability.
- P(B|A) is the likelihood probability.

Naive Bayes Assumption:

Given the class label, the Naive Bayes classifier assumes that the features are conditionally independent. This assumption simplifies the computation, making the classifier scalable and efficient [8]. The probability of a data point belonging to a class C is computed as:

$$P(C|X) = \frac{P(C)\times P(C)}{P(X)}$$

For multiple features $X_1, X_2, \ldots\ldots, X_n$, the equation becomes:

$$P(C|X_1, X_2, \ldots\ldots, X_n) = \frac{P(C)\times P(C)\times P(C)\ldots\ldots P(X_n|C)}{P(X_1, X_2, \ldots\ldots, X_n)}$$

This theorem has been used for spam detection; Naive Bayes can classify files as malicious or benign based on features such as file size, permissions, and embedded URL. To use this theorem, you must collect network traffic logs, including both normal and malicious traffic, over a very long period. Then, you must extract features like packet size, protocol type, source and destination IP addresses, and time between packets. Then, you would train the classifier on a labelled data set. Lastly, you would be left with a model that can be deployed to monitor network traffic and detect zero-day vulnerabilities.

Unsupervised Learning

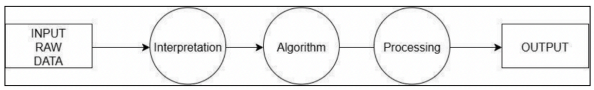

Figure 2: Simplified Unsupervised Machine Learning model

Unsupervised learning is training a model on an unlabelled data set, allowing the model to identify patterns in data without labels. This approach is useful for detecting anomalies where abnormal and normal behaviours are not explicitly labelled. As evident from the above table, after inputting raw data, the model has to interpret the data, feed it to an algorithm, and eventually process the inputs to give an output. We can use multiple algorithms like K-means clustering, DBSCAN and principal component analysis.

K-means Clustering Method

K-means clustering is an unsupervised learning algorithm that divides data into distinct clusters based on the criteria of feature similarity. It is a very useful algorithm for anomaly detection. Zero-day vulnerabilities can be detected via K-means clustering as they manifest as outliers from normal behaviour patterns.

Problem Definition:

Given a set of n data points $X=\{x_1, x_2,\ldots\ldots, x_n\}$, using K-means clustering, we will partition these points into k clusters $C=\{C_1, C_2,\ldots\ldots, C_n\}$ in a way that minimises the WCSS.

Mathematically, it is expressed as:

$$J = \sum_{i=1}^{k} \sum_{X\in C_i} \|X - \mu_i\|^2$$

Where:

- ^{k} is the number of clusters
- C_i is the set of data points in the i-th cluster
- X is a data point
- μ_i is the centroid of the i-th cluster
- $\|x-\mu_i\|^2$ is the squared Euclidean distance between data point X and centroid μ_i. [9]

To detect anomalies using this method, compute the distance of each data point after clustering. X_jTo its cluster centroid μ_i:

$$d_j = \|X_j - \mu_i\|$$

Set a threshold T for the distance. Hence, data points with distances greater than T are flagged as anomalies. This method can also detect anomalies in very large data sets.

Using AI to respond to zero-day vulnerabilities

Depending on the type of data and algorithms utilised, threat detection systems based on AI can detect a variety of dangers. Similarly, AI-based incident response systems can automate the response to cyber threats and reduce the time required to respond to an attack. These systems can analyse data from various sources and provide actionable insights to the security team to take appropriate action quickly [10]. 4.1 Automated Threat Response Systems Automated threat response systems are essential in cyber security. They use artificial intelligence and machine learning to detect, analyse, and automatically respond to cyber threats instantly and efficiently. These systems decrease response time, thereby reducing potential damage and protecting the systems. These automated systems can execute predefined responses to detected threats and adapt their actions based on the changing threat landscape. There are many ways to create automated response systems, but in this research paper, two systems, adaptive and rule-based systems, will be discussed.

Rule-Based Systems

Rule-based systems follow predefined rules to respond to threats. These are usually created by cyber security experts and are based on threat patterns and practices. To make a decision in a certain situation, a rule is typically structured and generated as an "IF[ANTECEDENT]=> THEN[CONSEQUENT]" statement, where "antecedent" represents necessary security attributes, conditions or contextual situations, and "consequent" represents the corresponding action [11]. They are very simple to implement and understand and are very reliable; rule-based systems are used in intrusion prevention systems and firewall management.

```python
# Example of a simple rule-based system in Python
def detect_threat(packet):
    # Predefined rule: block traffic from a specific IP address
    malicious_ip = "192.168.1.100"
    if packet.source_ip == malicious_ip:
        return "Block"
    else:
        return "Allow"

# Simulated network packet
class Packet:
    def __init__(self, source_ip):
        self.source_ip = source_ip

packet = Packet("192.168.1.100")
response = detect_threat(packet)
print(response)  # Output: Block
```

Figure 3: An example of a rule-based system

The example in Figure 3 detects a threat if it comes from a malicious IP. It starts by checking the IP of a packet. If the source IP is malicious, it will block the packet. But if it is any other IP, it will allow the packet to go through. This is

implemented on a large scale and leads to the creation of rule-based automated response systems.

Adaptive Systems

Adaptive systems are much better than rule-based systems as they use both artificial intelligence and machine learning to adjust their responses based on detecting new and changing threats. These systems learn from past incidents and change their strategies in real-time. These systems use algorithms and machine learning models to analyse data and identify patterns associated with threats. After that, they continuously monitor network traffic and system activities and change the responses based on the latest threat information. AI-driven SIEM systems and Endpoint Detection and Response systems are two primary areas where adaptive systems are used because of their proactive defence and flexibility.

However, it must be noted that they are much more complex than rule-based algorithms and are prone to generating false positives, which might lead to alert fatigue. As discussed previously, alert fatigue can decrease the overall protection of a system and lead to experts not realising threats. However, in the case of an automated response system, too many false positives would lead to the system rejecting many genuine requests.

```
import numpy as np
from sklearn.ensemble import RandomForestClassifier

# Simulated data: features of network traffic and corresponding labels (1 for threat, 0 fo
X_train = np.array([[0.1, 0.2], [0.3, 0.4], [0.5, 0.6], [0.7, 0.8]])
y_train = np.array([0, 0, 1, 1])

# Train a random forest classifier
model = RandomForestClassifier()
model.fit(X_train, y_train)

# Detect a new packet
new_packet = np.array([[0.6, 0.7]])
response = model.predict(new_packet)
if response == 1:
    print("Threat detected: Block")
else:
    print("Normal traffic: Allow")
```

Figure 4: example of an adaptive system

Conclusion

As cyberthreats rapidly evolve and cybercriminals start to leverage artificial intelligence, it is important to use artificial intelligence algorithms and machine learning to detect and protect data and systems from zero-day vulnerabilities.

As demonstrated in this paper, we can modify mathematical formulae and use other math concepts like graph theory to aid in our war against cybercrime. This paper has highlighted the importance of adaptive systems such as AI-driven SIEM systems and Endpoint Detection and Response systems in providing proactive defence and flexibility. This paper has demonstrated that automated detection and response systems are much better than traditional methods of cybersecurity. Using automated systems is much faster, more efficient and in many cases cheaper than traditional methods. As data becomes increasingly central to the daily functioning of critical systems, it is important to continue exploring and developing these technologies to ensure the security and integrity of data and systems.

Bibliography

1. IBM. What is a Zero-Day Exploit? [Internet]. IBM. 2024. Available from: https://www.ibm.com/topics/zero-day

2. IBM. IBM Security X-Force Threat Intelligence Index 2024 [Internet]. IBM. 2024. Available from: https://www.ibm.com/reports/threat-intelligence

3. Cloudflare. What Was the WannaCry Ransomware attack? [Internet]. Cloudflare. [cited 2024 Sep 15]. Available from: https://www.cloudflare.com/learning/security/ransomware/wannacry-ransomware/

4. Fortinet. What is an Advanced Persistent Threat (APT)? [Internet]. Fortinet. 2024. Available from: https://www.fortinet.com/resources/cyberglossary/advanced-persistent-threat

5. Ioulianou PP, Vassilakis V, Moscholios ID, Logothetis MD. A Signature-Based Intrusion Detection System for the Internet of Things. In: Information and Communication Technology Forum (ICTF) 2018 [Internet]. 2018. Available from: https://www.researchgate.net/publication/326376629_A_Signature-based_Intrusion_Detection_System_for_the_Internet_of_Things

6. Odii J, Anenechukwu Chukwunonso Hampo J, Onwuama NFO. Comparative Analysis of Malware Detection Techniques Using Signature Behavior and Heuristics. International Journal of Computer Science and Information Security [Internet]. 2019 Jul;17(7):33–50. Available from: https://www.researchgate.net/publication/350017172_COMPARATIVE_ANALYSIS_OF_MALWARE_DETECTION_TECHNIQUES_USING_SIGNATURE_BEHAVIOUR_AND_HEURISTICS

7. Francis R. False Positives Still Cause Threat Alert Fatigue [Internet]. CSO Online. 2017 [cited 2024 Sep 16]. Available from: https://www.csoonline.com/article/561285/false-positives-still-cause-alert-fatigue.html