International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Real-Time Anomaly Detection for Zero-Day Exploits

John Komarthi

San Jose, CA
john.komarthi@gmail.com

Abstract

Zero-day exploits are cyber attacks that take advantage of vulnerabilities that are previously unknown. Lack of prior signatures or patches makes them a critical security threat. This paper is going to explore the approach based on anomalies for real-time detection of such zero-day exploits. The approach tries to flag any deviations from normal behavior to recognise potential attacks. This paper will try to explore the challenges and limitations (including model poisoning, regulatory constraints, adversarial evasion, and operational issues) and observe some zero-day exploit detection in real-world scenarios. The paper will also outline the future directions, federated learning for collaborative defense, adaptive threat modeling, integration with cyber threat intelligence (CTI), and self-healing systems.

Keywords: Zero-Day Exploits, Real-Time Anomaly Detection, Machine Learning in Cybersecurity, Behavioral Analytics, Intrusion Detection Systems (IDS), Explainable AI (XAI), Federated Learning, Cyber Threat Intelligence (CTI).

INTRODUCTION

Zero-day exploits are cyber attacks that take advantage of vulnerabilities in the software that are unknown to the vendor. The occurrence of zero-day attacks has been on the rise in recent years. Mandiant reported that 80 zero-day vulnerabilities have been reported in the wild in 2021, i.e., more than double the previous record in 2019 [1]. Zero-day attacks are especially dangerous because they have no existing patches, and traditional signature-based which rely on known signatures fail to detect them. As a result, organizations face day-zero threats with no immediate updates to deploy, creating a critical gap in security.

The impact of zero-day exploits can be severe as the attackers can gain privileged access, get their hands on sensitive data, or cause physical damage in cyber-physical systems, all of that even before the defenders identify the vulnerability. Zero-day attacks remain undetected until after the damage is done; in most cases, they are later discovered via forensic analysis or public disclosure when the vendor learns about the vulnerability.

The 'window of exposure' can span days, months, or even years, starting from the moment the vulnerability is introduced until it is discovered and a patch is released to fix it. In this window, the attackers can operate freely; the range of threat actors spans from financially motivated criminals to state-sponsored groups. This increases the risk portfolio for all industry sectors, as modern IT infrastructure is an integral part of operations virtually everywhere, and more software inevitably means more undiscovered threats [1].

Real-time anomaly detection is a promising technique used to address this gap. Unlike traditional systems, which rely on known malicious signatures, anomaly detection systems model the expected behavior of the system and continuously monitor for any deviation that could be an attack. This approach can identify zero-day exploits based on the abnormal behavior of the systems or networks, eg, unusual sequence of system calls, user actions, or network flows, even if the underlying vulnerability or the exploit technique has not been used previously.

Modern security solutions already use behavioral analysis to identify novel attacks. If the software or the network traffic displays suspicious patterns that are not observed in normal operation, these patterns become a potential candidate to be flagged for a zero-day attack. Anomaly-based detection focuses on identifying what is happening (behavior) instead of why (known exploit). This makes real-time anomaly detection perfect for finding the unexpected and unforeseen.

However, deploying real-time anomaly detection for zero-day exploits is not an easy task. The biggest challenge is the modeling of 'normal' behavior accurately; the system has to deal with false positives, but at the same time detect any anomalies on time without overwhelming the analysts with a plethora of data.

Considering these current realities, the detection and handling of zero-day exploits need a proactive approach. Anomaly detection is the method that fits the need, it is also worth noting the fact that the concept of anomaly-based detection dates back to the 1980s. Denning's seminal intrusion detection model introduced the idea of profiling normal behavior and detecting deviations [2]. This approach has been refined by researchers over the decades [3], and this has gained importance in the context of zero-day detection in cybersecurity.

PRINCIPLES OF REAL-TIME ANOMALY DETECTION

Anomaly detection is the process in which the model defines a 'baseline' behavior of the systems, users, or networks and then identifies any behavior that deviates from the baseline and flags it as potential malicious activity. Traditional signature-based detection only compares the events to the database of known malicious signatures, but anomaly-based detection works on the assumption that the patterns of the attack can vary, but generally they look deviated when compared to the regular benign behavior. This approach makes anomaly-based detection particularly powerful for detecting zero-day attacks and other novel threats.

Systems will have to go through a learning phase where the typical behavioral data will be collected. This involves monitoring command sequences being executed on the host, monitoring network traffic

volumes, login frequencies at various times, etc. The output generated from this model is the learning phase to determine the baseline or normalcy, i.e, statistical ranges for certain metrics, a machine learning model which will capture normal pattern structure, or the baseline rules which are derived from clustering normal instances. Any observation outside the expected range will be flagged as an anomaly.

These are some of the commonly used techniques for anomaly detection:

- ***Statistical methods:*** This will define the normal behavior in terms of distribution and thresholds. A simple example could be tracking the average number of queries on the database per minute and flagging any prolonged increase beyond the set threshold. Sophisticated statistical models also use probabilistic measures like Gaussian models of network packet sizes to compute anomaly scores.
- ***Machine learning methods:*** A variety of ML methods are used for anomaly detection, and unsupervised or semi-supervised learning is employed. Some of the examples include clustering algorithms where the normal data forms clusters and the outliers are the flagged threats, one-class SVM's are trained to accept normal data and reject anything else and neural network-based approaches (auto encoders or recurrent neural networks) learn to reconstruct normal data and measure reconstruction error as the anomaly score. Deep learning has also been applied recently to model complex behaviors, eg, an LSTM-based model that can learn sequences of system calls and detect when a new sequence is observed (a comprehensive survey of such techniques is provided by Chandola *et al)* [3].
- ***Rule-based and heuristic methods:*** In this domain, experts define what is abnormal behavior, e.g, any network connection from an operational technology system to an unknown external IP can be considered as an anomaly in the case of a critical infrastructure network. These rules leverage the knowledge of the environment to detect anomalies and can be very effective for specific zero-day exploits (eg, PLC controller suddenly initiating a connection which was never seen).

In real-time anomaly detection, the above-mentioned techniques must be applied to the streaming data or should be continuously monitored to flag any potential incidents immediately with minimum delay. This required an architecture that is capable of handling high volumes of data (network flows, log events, etc) and performing detection on the go. Latency is extremely important in the process of real-time anomaly detection, as even a few minutes delay in flagging a potential threat can be detrimental. The attackers can achieve their objective within seconds in some cases. Because of this anomaly, detectors are designed and optimised for speed and deployed close to the data sources. For example, an agent on a host monitoring system calls in real-time, or a network sensor analyzing packets as they pass.

Anomaly detection can detect the symptoms of an attack without knowing the cause, for example, if a zero-day exploit gives the attacker access to a server, and the actions of the attacker, such as disabling services or exfiltrating data, can be anomalous and will trigger alerts. For example, Anomaly-based IDS/IPS detected ransomware outbreaks such as WannaCry without recognizing the worm's code, but by highlighting the unusual scanning and encryption behavior that was caused on networks [7].

However, it is important to observe that not all anomalies are attacks, and not all attacks can produce easily detectable anomalies. Base-rate fallacy becomes a challenge when true anomalies are rare relative to the normal operations, even a low false alarm rate can result in false positives. Tuning the anomaly detection system in a way that the sensitivity must be high enough to observe subtle deviations, but not so high that even a minor fluctuation will trigger an alert. Adaptive thresholds and feedback from the analysts are used to tune the system and set the baseline for 'normal'.

SYSTEM ARCHITECTURE AND IMPLEMENTATION

A typical system architecture for real-time anomaly detection for zero-day exploits has multiple stages, from data collection to generating the alerts.

1. ***Data Collection:*** The collection system gathers data from multiple sources in real time. For example, network traffic packets, host-based logs, user activity logs (system logs, application logs), and endpoint telemetry. A collection of a wide range of telemetry is crucial to get a comprehensive overview of the zero-day exploit coverage, eg, an attacker exploiting zero-day in a web application may also generate unusual web requests which are captured in HTTP logs, along with unusual database queries or file writes on the server. Sensors and agents are continuously fed with this raw data and send it into the detection pipeline. In cases of high-throughput technologies like Apache Kafka can be used as the messaging layer to stream data to the processing units.
2. ***Feature Extraction and Preprocessing:*** Raw data is transformed into features that are amenable to modeling. For example, the network packets can be aggregated into flows, and features like 'bytes transferred per minute per IP' or 'number of distinct external domains contacted by host per hour' can be computed. System call sequences can also be encoded in the same way via n-grams or embedding techniques and input them to an ML model.
   The step of preprocessing these extracted features includes normalization to scale features, reducing noise by filtering out benign known events, and merging the context of multiple sources. Selecting the features that will highlight the effects of potential attacks is a critical design choice of the anomaly detection system, so domain knowledge becomes key. E.g., in a privilege escalation exploit, features capturing processes spawning with new privileges or unusual access to protected files could be key.
3. ***Anomaly Detection Engine:*** One or more anomaly detection algorithms continuously process the incoming feature data and decide whether it is anomalous or not; this is the core analytical component of the system. The detection engine can be a single machine learning model or an ensemble of models and rules that operates in parallel. Auto-encoder neural network that reconstructs input and flags reconstruction error is an example of a single ML model [3]. A statistical threshold detector can detect gross outliers in the network traffic volume, while a complex ML model can catch subtler anomalies in sequence patterns. In real-time anomaly detection, systems are updated periodically or continuously to incorporate new behavior patterns. But precaution also needs to be taken with updating the normal behavior to avoid an attacker "poisoning" the model by slowly training it to accept the malicious behavior. Some advanced architectures, such as the one proposed by Touré *et al.* [4], cover the entire intrusion detection

cycle, right from data collection to new attack classification by analyzing the anomalies in network flows.

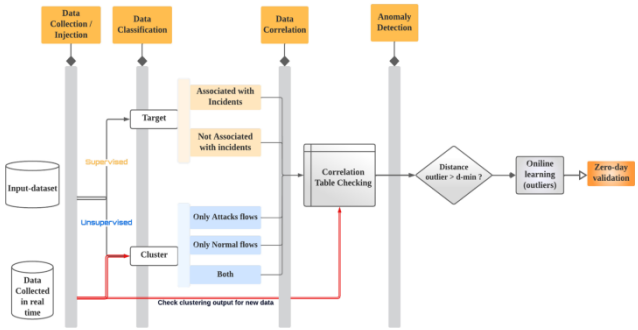

Figure 1: Zero-day detection protocol

4. ***Alert Generation and Aggregation:*** An alert is generated after the detection engine flags an anomaly. A typical alert includes the observed information, eg, "Host Z made an outbound connection to an unrecognised domain with 15x the normal data volume" along with why it was considered anomalous, "volume is 15 times above baseline". These alerts are generally forwarded to the Security Information and Event Management (SIEM) system or a security dashboard for the analysts' review. Multiple unusual behaviors on the same host around the same time can be correlated into one incident, low-level anomalies are combined into higher-level incidents using aggregation logic to reduce alert fatigue.
5. ***Response (Optional Integration):*** Even though detection is the main focus, most modern-day anomaly detection systems are integrated with response workflows. It could be as simple as forwarding the alerts to an orchestration platform or as direct as automatically triggering necessary containment processes. For instance, if a critical server is exhibiting strong anomalous behavior, the system might automatically isolate that server from the network (autonomous response). Darktrace's Antigena is an example of an automated response that reacts to anomalies in real-time.

Implementation considerations: Real-time operations must handle data that is streaming at high rates, so the anomaly detection algorithms have to be efficient. Sliding window techniques are generally used to get the window of recent activity and not store unbounded history. Data structures and databases (eg, time-series database) have to be used to retrieve baseline metrics to compare with the anomalies. In case of large enterprises, data sources are distributed, so the architecture might also be distributed with local anomaly detection systems at multiple network segments feeding the system for central analysis. The system can leverage horizontal scaling using a cloud-based implementation.

The detection system has to be reliable, secure, and fail-safe. The detection system also needs to be protected from attackers who might target the monitoring system and blind it. The system will collect telemetry, distill that into meaningful patterns, use the anomaly detection engine to flag any behavior straying from the norm, and trigger real-time alerts.

Case Study 1: WannaCry Ransomware Worm (2017)

WannaCry was a fast-spreading worm that exploited the zero-day vulnerability in Microsoft's SMB protocol (later identified as the EternalBlue exploit). Organizations that had not yet applied Microsoft's patches and were using traditional signature-based malware detection systems were defenseless to the WannaCry ransomware worm on day zero. The network behavior of WannyCry was highly anomalous; in one recorded case, Darktrace identified the outbreak within a second of deviation from normal network patterns [7]. Once compromised, a single PC on the network started scanning the internal network for vulnerable hosts at a rate far beyond normal, triggering the anomaly detector. In addition, as multiple devices are infected within one second of the worm's propagation, the deviation in network pattern is detected [7]. This demonstrates how WannaCry's code was detected using behavior anomalies through real-time detection, even though the code was unknown on a zero-day. Darktrace's autonomous response module automatically isolated the infected machine connections.

Case Study 2: Casino Fish Tank Thermostat Breach (2017)

Some zero-day exploits are novel and are stealthy, using novel vectors. In a North American casino, the attackers infiltrated the network using a high-tech fish tank that was connected to the internet. The thermostat system sensor of the fish tank was connected to the casino's internal network and communicated telemetry over to a management service. But an anomaly detection system was already in place that quickly detected deviation and observed "anomalous data transfers to an external network from the fish tank" and triggered alarms [9].

Another similar incident occurred, where the smart tablets of the designer at an architecture firm were compromised and turned them into part of a DDoS botnet. High volume traffic from those devices was flagged using real-time anomaly detection on zero-day.

Case Study 3: Unseen Ransomware – BlackByte Attack (2021)

This case involves a complex and targeted attack on an East African financial organization, which unfolded over several days and deployed a then-unrecognized ransomware (identified as BlackByte) [8]. Initially, an external-facing VPN server was breached, mostly exploiting a zero-day vulnerability or weak credentials in the VPN software. Once infiltrated, the attacker created a new admin user, using that account to pivot via RDP into the domain controller and conducting reconnaissance. These were technically legitimate commands, but in the particular context were highly unusual.

An anomaly-based detection system that monitored the organization flagged multiple stages of the attacks as anomalies. The spike in failed login attempts on the VPN server (brute-force) was not prioritized as such incidents previously occurred, the system noticed administrative privileges outside the normal pattern for that server. Cyber AI, an automated investigation module, picked up on the unusual use of admin credentials and RDP from the VPN service, along with rare external connections to internal hosts were flagged.

Even though the attack was new at that time, defenders were alerted early on in the unknown and uncategorized phase of the attack. This helped them buy time to investigate, respond, and involve

incident response before the damage was done. In this case, the combination of network and user behavior analytics signaled the zero-day attack [8].

These case studies showcase the abilities of the detection systems and also hint at the challenges, such as distinguishing malicious patterns and dealing with attackers.

CHALLENGES AND LIMITATIONS

1. False Positives and Alert Fatigue: Complex IT environments display a lot of variable behavior, and not every deviation is an attack. An impromptu scan by the defender or an increase in traffic after a software update looks like a deviation but is not an attack. The detection system has to be tuned well so that it won't generate frequent false alarms, causing alert fatigue. Adaptive thresholds, feedback loops, and tiered alerting are used to achieve optimal balance, which is environment-specific.

2. Evolving Baselines and Concept Drift: Networks evolve, and the baseline normal behavior metrics are not static. The system workload varies over time, and if the detection model doesn't adapt, it certainly will miss attacks or flag normal behavior as anomalous. Anomaly detectors need to be retrained periodically on baseline metrics; a combination of sliding time windows, online learning algorithms, and explicit drift direction tests can signal when a model needs a refresh.

3. Lack of Ground Truth and Difficult Evaluation: After an anomaly is flagged, determining whether the anomaly is malicious or not will require further investigation. For tuning and testing, organizations use historical attack data so that the system catches recognized malicious events, but zero-day attacks by definition are unknown.

4. Adversarial Evasion and Model Poisoning: Attackers also try to evade or poison the anomaly-based detection system models.

- *Evasion (adversarial attacks at test time):* The attacker generally tries to make their activity as normal as possible. They might trickle out the data in smaller sizes below the threshold or mimic the patterns of normal traffic. Adversarial machine learning research also points out that it is possible to fool the ML models by subtly modifying the inputs. Some attackers even replay normal behavior during the attack to mimic normalcy.
- *Poisoning (attacks on the training/model):* When the anomaly detection models are updated automatically over time, attackers try and feed malicious data that results in a skewed model. Researchers Bhargava and Clifton (2018) demonstrated how anomaly detectors can be gradually trained under poisoning attacks to miss outliers [6]. Biggio *et al.* showed earlier with SVM models that poisoning a fraction of training data could significantly degrade detection [5]. Techniques like data sanitization and training algorithms are used to defend against poisoning, but are not foolproof.

5. Scalability and Performance Constraints: Real-time anomaly detection systems are computationally intensive. They need to monitor high-speed network links, thousands of endpoints in real-time, which requires a lot of processing power and memory. Deep-learning-based detectors are also resource-heavy,

and not all organizations can afford a GPU dedicated to security analytics. There is a trade-off between the speed of the algorithm and the complexity of the algorithm. The system needs to be designed in a way that can handle the load, not just detection, but also logging and storing context for each anomaly that needs to be investigated. If the system bottlenecks, fast-moving attacks can slip through before detection, hence distributed computing techniques and efficient stream processing are necessary during the model deployment.

6. Integration and Contextualization: The security team needs context when analyzing the flagged items, eg, a spike in network traffic can be from downloading a big file, and without context, defenders will end up chasing it down. Many anomaly detection systems do not incorporate threat intelligence or correlate with a recent software update, thus making anomaly alerts context-poor and requiring additional analysis. Modern anomaly detection systems try to improve this by connecting the anomalies with context using asset databases, threat intelligence, and CMDB information. At the time of deployment, the detectors need to be combined with tools like SIEM or SOAR platforms to give the full picture. Anomaly detection systems perform better when it is not a standalone solution but part of an ecosystem.

7. Privacy and Regulatory Concerns: The detection systems require broad data access rights to inspect user communications, possibly personal data, and system logs, which conflict with the privacy expectations and regulations. Especially in geographical locations where laws like GDPR are strong, it becomes harder to deploy anomaly detection systems. If the system automatically takes an action, this could be considered an automated decision that can affect the user and, under Article 22 of GDPR, needs human oversight.

Additionally, storing and processing huge amounts of log data pose compliance issues, and security teams have to ensure that sensitive personal data is handled appropriately. If the training data reflects historical bias, the anomaly detector can profile certain groups that lead to discrimination (risk of bias). Thus, ensuring fairness in anomaly detection becomes crucial.

Techniques like data pseudonymization, access controls on organizations deploying the anomaly detection systems have to navigate a fine line between comprehensive monitoring and respecting the privacy and rights of individuals at the same time. Despite the challenges, anomaly detection will remain a vital tool in zero-day detection, and most of the limitations can be mitigated with system design and operational processes.

FUTURE DIRECTION

1. Explainable AI (XAI) for Anomaly Detection: One of the major problems for advanced anomaly detection systems is their ‘black box’ nature; the systems don’t provide enough humanly understandable context for the defender to understand why the event was flagged. This reduces the analyst's trust and hinders compliance. This is why we might see increased adoption of XAI in the future. Researchers are already working on applying methods like LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations) to intrusion detection models to highlight which features or behavior contributed most to a given anomaly score [10]. Future anomaly detection dashboards can

include feature contributions, natural language summaries of anomalies, and even counterfactual analysis.

2. Federated Learning and Collaborative Detection: Anomaly detection gets better with access to diverse data sets, but sharing raw data raises privacy concerns. Federated learning (FL) enables the models to be trained on decentralized data without data exposure, this also allows multiple organizations to train together and make anomaly detectors stronger. Early studies on federated anomaly-based IDS displayed better anomaly detection while keeping data localized [11]. In the future, security vendors may offer FL updates that enhance anomaly detection across clients, similar to anti-virus updates, but for anomaly detection.

3. Adaptive and Self-Learning Threat Models: The anomaly detection systems in the future are expected to be adaptive, continuously learning from evolving data and adversarial behavior. They will handle concept drift automatically, unlike traditional static models. Incorporating feedback from false positives and missed incidents refines the detection accuracy over time. The models are guided using reinforcement learning by rewarding correct identifications and penalizing errors. To counter adaptive attackers, adversarial training using simulated threats will help the model understand and resist the evasion tactics in dynamic environments.

4. Integration with Cyber Threat Intelligence (CTI): Anomaly detection on zero-day performed better when integrated with external threat intelligence (CTI) data. CTI offers details on known Indicators of Compromise (IOCs) such as domains, file hashes, IPs, etc. Even though zero-day exploits are not in threat feeds, CTi can still enrich the context of the anomaly. Some SIEM/SOAR platforms already offer enrichment but need tighter integration. This hybrid model combines anomaly detection's ability to spot the unknown with CTI's expertise, strengthening overall zero-day defense.

5. Self-Healing and Autonomous Response Systems: Self-healing cybersecurity systems represent a shift towards autonomous defense mechanisms that not only detect threats but also respond and recover without human intervention. These systems integrate detection, response, and remediation processes, enabling them to identify anomalies, contain threats, and restore normal operations swiftly. By leveraging artificial intelligence and machine learning, they can adapt to new threats, learn from past incidents, and improve over time. This approach reduces reliance on manual responses, minimizes downtime, and enhances overall system resilience.

6. Advanced Data Sources and Algorithms: Anomaly detection is evolving with richer data sources like cloud telemetry, business process logs, and enabling deeper insights. Graph-based methods, which map user, resource, time, and device relationships, are used to detect complex, multi-step attacks that traditional logs miss.

The future of zero-day detection is smarter and more autonomous. Explainable AI builds trust, federated learning broadens reach, adaptive systems handle change, CTI integration adds context, and self-healing capabilities proactively stop threats. Together, these advances aim to outpace attackers exploiting unknown vulnerabilities.

CONCLUSION

Zero-day exploits are one of the most critical security challenges that organizations face in modern times. They attack without prior warning and often cut through traditional defenses that are dependent on known threats. In this paper, we have discussed why real-time anomaly detection is vital to defend against unknown zero-day attacks. Anomaly-based detection catches the early signs of zero-day exploits in real-time by continuously analyzing the system and network behavior. In the case studies discussed, we have seen that this approach can detect a wide range of threats from rapidly spreading ransomware worms to stealthy data exfiltration through an IoT device, and complex multi-stage intrusions by advanced attackers.

At the same time, challenges have also been highlighted, the system can sometimes give out false positives, can be fooled by adaptive attackers, and needs careful designing, maintenance, and tuning. Defenders also need to follow compliance, respect privacy, and provide valuable data to analysts instead of drowning them in unnecessary and unexplained alerts. These challenges highlight that a stand-alone anomaly detection system for zero-day is not sufficient for zero-day defense; integration is a key theme. The system needs to be integrated with human expertise, other security tools, and a continuous learning process.
The future direction of anomaly-based detection systems for zero-day malware is moving towards robust and usable systems where explainable AI will help decode the alerts, making analysts more effective. A federated and collaborative approach ensures that one organization's learning can benefit multiple organizations. Adaptive models and self-healing systems enable the security systems to dynamically reconfigure and improve themselves, making the security team's life easier. Bringing in threat intelligence (CTI) and automated response systems closes the loop from detection to action, which significantly shrinks the window in which the attacker can operate freely.

Enhancing the accuracy of the detection algorithm, scaling to ever-growing data volumes, and validating these systems rigorously remain a continuous process for cybersecurity researchers. As for security engineers and defenders, the task is to effectively integrate anomaly detection into security operations, defining what data to monitor, how to handle alerts, and what the success rate is. The detection systems need an iterative approach where the team deploys, gets feedback, tunes the systems, and repeats while being mindful of developments in the field.

In conclusion, real-time anomaly detection systems transform the unknown and unpredictable zero-day exploits from an impossible problem to a manageable risk. This changes the approach from reacting to the threat to being proactive, where real-time system behavior is monitored and you spot any deviations and stop the attack in its tracks.

Anomaly detection

Leverages on abundant data and advanced models to uncover any subtle signs of attack. Organizations can benefit from investing in and embracing these systems and be prepared for the next zero-day attack waiting around the corner.

REFERENCES

1. J. Sadowski, **"Zero Tolerance: More Zero-Days Exploited in 2021 Than Ever Before,"** *Google Cloud Blog – Threat Intelligence (Mandiant)*, Apr. 21, 2022. cloud.google.com
2. D. E. Denning, **"An intrusion-detection model,"** *IEEE Trans. Software Engineering*, vol. SE-13, no. 2, pp. 222–232, 1987.
3. V. Chandola, A. Banerjee, and V. Kumar, **"Anomaly detection: A survey,"** *ACM Computing Surveys*, vol. 41, no. 3, Article 15, 2009.
4. A. Touré, Y. Imine, A. Semnont, T. Delot, and A. Gallais, **"A framework for detecting zero-day exploits in network flows,"** *Computer Networks*, vol. 224, p. 110476, 2024.
5. B. Biggio, B. Nelson, and P. Laskov, **"Poisoning attacks against support vector machines,"** in *Proc. 29th International Conf. Machine Learning (ICML)*, Edinburgh, UK, 2012, pp. 1467–1474.
6. R. Bhargava and C. Clifton, **"Anomaly detection under poisoning attacks,"** in *Proc. ODD v5.0 Workshop (Outlier Detection De-constructed) at KDD*, 2018link.springer.com.
7. A. Tsonchev, **"Darktrace's Response Against WannaCry Ransomware,"** *Darktrace Blog (Inside the SOC)*, May 16, 2017. darktrace.com
8. E. Foulger, **"Detecting Unknown Ransomware: A Darktrace Case Study,"** *Darktrace Blog (Inside the SOC)*, Aug. 24, 2022. darktrace.com
9. K. Townsend, **"Hacked Smart Fish Tank Exfiltrated Data to 'Rare External Destination',"** *SecurityWeek*, Jul. 26, 2017. securityweek.com
10. S. Patil *et al*., **"Explainable Artificial Intelligence for Intrusion Detection System,"** *Electronics*, vol. 11, no. 19, Article 3079, 2022. ieeexplore.ieee.org
11. S. Agrawal, A. Sharma, and D. Bhardwaj, **"Federated Learning for Intrusion Detection System: Concepts, Challenges and Future Directions,"** *Computer Communications*, vol. 195, pp. 346–361, 2022.
12. Zilliz (Milvus) Blog, **"What are the privacy concerns in anomaly detection?"** 2023. blog.milvus.ioblog.milvus.io
13. T. George, **"Self-Healing Cybersecurity Systems: A Pipe Dream or Reality?"** *SecurityWeek*, Jun. 23, 2021. securityweek.com
14. N. Marić, **"5 Examples of Zero Day Vulnerabilities and How to Protect Your Organization,"** *Bright Security Blog*, 2023. brightsec.com

Figure-1
M. Guo, C. Li, X. Zhang, and Q. Li, "Zero-day attack detection based on dynamic behavior profiling and feature reduction," *Computer Networks*, vol. 240, p. 110113, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128624003086