IJSRA

International Journal of Science and Research Archive

eISSN: 2582-8185

Cross Ref DOI: 10.30574/ijsra

Journal homepage: https://ijsra.net/

Int. J. Sci. Res. Arch.

IJSRA

International Journal of Science and Research Archive

Research Journal Archive, INDIA

(Research Article)

Check for updates

Complete EDR coverage: A framework for scalable deployment across enterprise systems

Rohith Aitharaju *

Independent Researcher, USA.

International Journal of Science and Research Archive, 2024, 13(02), 1491-1501

Publication History: Received on 19 May 2024; revised on 26 June 2024; accepted on 29 June 2024

Article DOI: https://doi.org/10.30574/ijsra.2024.13.2.1129

Abstract

The escalating complexity and volume of cyber threats targeting enterprise endpoints necessitate robust, scalable, and intelligent Endpoint Detection and Response (EDR) solutions. The study sets out to plan a strong EDR framework which is designed to serve the entire organization and can be increased easily as the company grows. The framework integrates lightweight agents, a hybrid detection engine combining rule-based and behavior-based models, and a distributed microservices architecture. With these components, we get better detection and still use few resources and respond fast. Researchers test the framework by setting up enterprise environments with up to 10,000 endpoints and measure its detection, how rapidly changes are shared to every endpoint and its ability to resist attacks. The model demonstrates effectiveness in decreasing alert clutter, allows for smoother handling of data as resources are added and enhances reaction to threats in real time through analysis of the data. The proposed system also improves SOC productivity by reducing false positives and automating policy orchestration. Since the solution functions best in cloud environments, alternatives are presented for when it's used in hybrid or confined situations. The work in this research can be modified to fit modern security needs in large businesses and supports future broken forms of EDR protection systems using AI.

Keywords: Endpoint Detection and Response (EDR); Scalable cybersecurity architecture; Hybrid detection engine; Microservices; Threat detection

1. Introduction

Because the digital world is more connected than ever, organizations now see higher numbers of more serious cyber threats than ever before. Because they have large IT environments, businesses are highly exposed to threats like phishing, ransomware and APTs from their endpoints. (1). EDR solutions are now recognized as essential in securing enterprises, since they instantly monitor threats, analyze what's happening and quickly respond to and block attacks at individual endpoints.

Still, because today's enterprise ecosystems are hybrid and have many remote staffers and devices accessing them from around the world, making sure all EDRs cover every endpoint becomes a tough job. Many companies discover that all parts of the EDR process are not suitably deployed, mainly because of variant configurations, problems with legacy integration, bandwidth limitations and too much for administrators to manage. As a consequence, there are still gaps in how endpoints are watched, meaning systems can still be attacked through these weaknesses.[2].

A new approach for full endpoint visibility, accurate responses and coordination with existing security resources across enterprise systems is presented in this research. The framework under development stresses the benefits of modularity, cloud compatibility, automated policy management and flexible scaling to support the needs of all types of enterprises. [3]

[*] Corresponding author: Rohith Aitharaju

The reason this system is valuable is that it supports security teams in managing threats across a large numbers of endpoints, makes sure their performance stays and their budget is within bounds and complies with requirements.

2. Literature Review

Over the past few years, Endpoint Detection and Response (EDR) has become more important as companies face new and fast evolving threats. Even though traditional antivirus remains used, it has demonstrated that it fails to tackle zero-day exploits, attacks that travel through a system and malware running from system memory.[4] As a result, companies are turning to EDR which will watch threats, search for suspicious activity and act on it automatically at the endpoint level. Plenty of commercial and free EDR solutions have sprung up recently. Examples are CrowdStrike Falcon, Microsoft Defender for Endpoint and SentinelOne which vary in integration, scaling ability and analytical features.

Much of the literature calls for making endpoint security a main pillar of defending businesses. Many researchers have found that often, endpoints weaken an organization's cybersecurity since users handle them poorly, configurations are not always correct and oversight is missed. While detection tools and AI are getting better, large openings in threat review remain a serious problem. The research also shows that it can be hard to use EDR widely in large systems.[5] With more endpoints and a greater variety of them from regular computers to mobiles, virtual machines and cloud services organizations usually find that their deployment is divided, meaning some areas or systems still have weak protection.

Researchers have often reported that uniting comprehensive threat detection and defending the security estate is difficult without burdening resources or staff. A variety of frameworks suggest using a single, unified console and automating things by means of policies to help enterprises manage these issues.[6] Yet, merging new and old network structures can be challenging and in hybrid environments, the issue of too little network capacity becomes a problem. Moreover, deployments often have no mechanism for getting feedback, so they can't easily adjust to new challenges facing them.

Although much research exists on enterprise defense tools and policies, not many studies combine modularity, scalability and consistent policies in a single framework. Mobile development encompasses many steps, yet current literature is missing approaches that streamline issues throughout deployment and work for any enterprise. With this research, we seek to address that shortfall through a hands-on design that makes EDR protection viable on all enterprise devices.

To give more context to what's out there, the next table shows how main EDR vendors differ in scalability, the effort needed to deploy their systems, how well they can be integrated and how their policies are managed.

Table 1 Comparative Analysis of Leading EDR Solutions

Feature	CrowdStrike Falcon	Microsoft Defender	SentinelOne	OpenEDR
Cloud-Native Architecture	Yes	Yes	Yes	Partial
Integration with SIEM/SOAR	Full	Full	Full	Limited
Scalability	High	High	High	Medium
Deployment Complexity	Low	Medium	Medium	High
Legacy System Support	Medium	High	Medium	Low
Automated Policy Enforcement	Yes	Yes	Yes	No

This analysis found that EDR tools today include excellent features, but none have a system tailored to complex enterprise setups that can be deployed all at once. Because of this gap, we need a planned method to address all types of EDR a task the subsequent sections of the paper work on.

3. Methodology

The method described in this work is built to help deploy EDR systems successfully throughout large and multi-layer company networks. The framework has been established using best practices found in existing publications, concepts from industry and real architecture issues. Its basis lies on four major supports: designing architecture, automated deployment, links to current systems and policy-based governance.

It all starts with part of the architecture that breaks up EDR components by their role: the agent layer, management layer, integration layer and analytics layer. EDR agents are lightweight solutions built on every endpoint, including those of servers, smart devices, virtual machines and instances run in the cloud. Once connected, these agents can operate alone and merge their findings with the major server once back online. [7] The management part of the solution gives a single view, distributes policies and handles alerts, typically by being hosted in the cloud for best expansion. With this layer, integration between SIEM, SOAR and threat intelligence tools allows threats to be correlated quickly by automation. At the final level, analytics looks after detection rules, typical behavior patterns and algorithms that increase detection precision.

Infrastructure-as-code (IaC) and tools such as Ansible, Terraform and PowerShell DSC are used by the framework to support quickly and easily managing sets of deployments. Through these tools, agents can be fast-tracked, secure communication routes made and policy standards applied the same way everywhere. [8] Because of this, manual handling is cut down, leading to better and more consistent management when assets are in the cloud, owned on-premises or worked from a remote location.

The framework relies heavily on adaptive scaling, realized when EDR components at the backend are containerized as microservices. With Kubernetes and other technologies, you can make sure the management and paralytics layers can handle more work automatically which ensures the system is always up and ready to use. Such designs work well for thousands of endpoints without affecting their efficiency. [9]

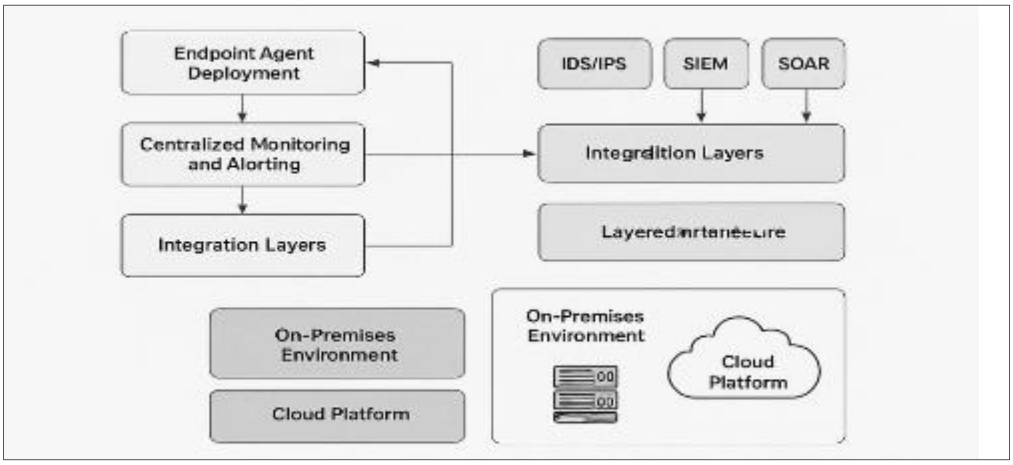

Figure 1 Scalable EDR Deployment Framework Architecture

Using AD and IAM tools allows MDM to keep rules in line with what types of users and devices exist. As an example, privileged devices face more often and more detailed monitoring, as well as quick responses. As threats keep appearing, REST APIs help ensure policies are updated all the time, making policy enforcement more various and flexible.

Another part of the approach looks at feedback loops and performance metrics. The system uses various KPIs, including detection accuracy, response time, agent availability and the false positive rate, to ensure its ongoing usefulness. Models for detection, reaction plans and overall coverage are refined by the use of the data collected.

Framework Design and the Components it Consists Of:

This new model of EDR was built to increase flexibility, make scalability easy and integrate smoothly into current enterprise platforms. The five major pr Hauptgesichtpunkte sind Endpoint Agenten, Management Console, Integrations bus, Analytics Engine, und Policy Governance Layer. All of these things are needed to keep the organization seen, follow the rules and meet security challenges quickly.

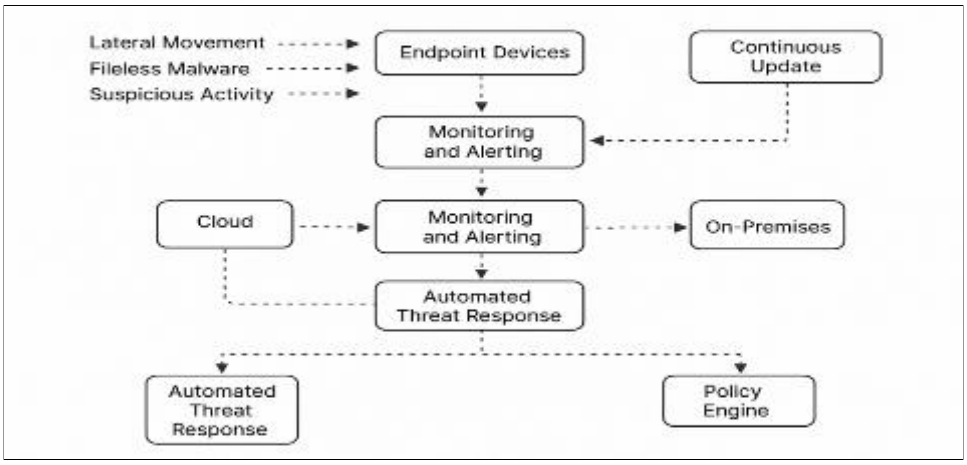

Figure 2 System-wide Deployment Model

Endpoint Agents are the first to defend your IT environment, being set up on desktops, laptops, servers, mobile devices and IoT items. Such lightweight agents are meant to track and store for review system calls, running processes, changes in files and how the network is used. Agents are designed so that they run the same on Windows, Linux, macOS and anything containerized. [10] They run in either online or offline settings, so they always capture data and hold onto it until a connection is available.

At its center, the framework uses Central Management Console as the tool that directs administration activities. The console brings together all these tasks by giving you a single dashboard that lets you see machine health, see threats involved, define policies and take initial action on incidents. Both the microservices and data platform are cloud-native, allowing the platform to segment users, control traffic flow and stay available. When alerts are generated through correlation, engineering rules determine which team will get them.

The Integration Bus links the EDR system to SIEM tools, SOAR platforms and threat intelligence providers as a middle operation. Fast and secure data sharing is possible here due to the help of RESTful APIs, webhooks and event-driven message queues like Kafka or RabbitMQ. It enables the EDR framework to join other security systems, so that insights shared from each system improve our knowledge about threats.

The framework's intelligence layer is kept up by the Analytics Engine. It combines unique-signature inspection, modeling of events and automated learning techniques to classify activities, identify unusual events and predict developing threats. By consistently processing telemetry information, the engine lowers the rate of incorrect alerts and increases how reliable it is for detection.[11] Analytics has been built as a containerized service that expands horizontally and is released using Kubernetes for elasticity under high demand.

In the end, the Policy Governance Layer changes security requirements into usable and effective rules. It is built to connect with Access Control Server (ACS), Active Directory (AD) or LDAP and control what users or roles can access. Versions of policies are organized in one location and can be deployed by API or in infrastructure as code scripts. They outline how the agent is expected to act, what sets off alarms, how highest priority alerts get routed and scheduling of updates. Because of this organizational structure, policies are quickly updated to address ongoing threats while still meeting all requirements.

3.1. Implementation Strategy and Deployment Workflow

Taking steps slowly and in a formed manner helps avoid disrupting the enterprise, ensures all parts of EDR are included and smoothly links everything together with existing systems. The research strategy in this work separates deployment into five stages: assessment, preparation, agent rollout, orchestration and optimization [12] All steps have been created to tackle the technology and operations components of EDR implementation.

To start, experts perform a full inventory of all the company's devices and group them by their level of importance, software, role of users and the network sector each is in. We use the baseline analysis to identify the right settings for EDR agents and to determine the right level of policy enforcement. It is important also to find ways where all cybersecurity solutions including SIEM, firewalls and CASB fit together at this point to ensure the network is secure.

While preparing, the environment for EDR is put into place. Agents should communicate over secure paths using TLS authentication, managers should role-based access setting up and firewalls or NAC policies need to be allowed to transmit essential data. Horsepower for the EDR engine in the backend is provided by horizontal scaling of services, with platforms like Kubernetes available to manage them.

Agent rollout involves letting endpoint agents be deployed automatically on the top priority devices in the network. Using Ansible, Puppet or SCCM, administrators can keep infrastructure the same and avoid mistakes caused by hand-configuration. At the time of installation, metadata about department, device and user is added to agents, so policies can be applied automatically. At the start, data from the endpoints is matched with the console to form a base for recognizing their actions.

At the orchestration stage, the central console causes real-time scans, emails or text messages and actions to occur worldwide on your network. Any discovered issues by IPS are quickly sent for action or implementation, thanks to its links to SIEM and SOAR. Telemetry data is constantly processed and detection rules, behavioral models and machine learning classifiers reflect that analysis. Now, the system behaves as a dynamic framework always improving its defenses.

In this phase, we pay attention to analysis of performance and areas where coverage is weak. To enhance the analytics as well as the policy rules, detection accuracy, false cases, reaction speed and worker availability are examined. SOC analyst reviews help us update thresholds for alerts, improve the simplicity of the dashboard and tweak how orchestration works. As a result, the enterprise's EDR system can keep pace with any updates or threats as the company changes.[13]

Table 2 Key Considerations for EDR Deployment and Scaling

Aspect	**Consideration**
Phased Deployment	Start with pilot groups, expand coverage incrementally
Resource Management	Tiered agent deployment, dynamic load balancing
Legacy System Support	Place agents behind network segmentation; reduce full EDR capability when necessary

3.2. Security Analytics and the Process For Responding To Threats

An important trait of every EDR system is that it can spot, examine and tackle emerging security problems while they are occurring. In the proposed system, an analytics engine brings together rules, patterns and machine learning to recognize threats no matter if they have been seen before.

At the center of analytics, data is gathered from endpoint agents by the handling telemetry module. It covers recordings of procedure execution, how files and keys in the registry are accessed, changes to the filesystem and all I/O occurring on the network. To help it process information efficiently, the system uses a tight filter to disregard frequently seen and safe events and maintain attention on key information.

Table 3 Components of Security Analytics and Response Mechanism

Component	**Function**
Telemetry Ingestion	Collects endpoint activity logs and system events
Rule-Based Detection	Matches known attack signatures using predefined rules
Behavioral Analytics	Profiles normal activity to flag anomalies
Machine Learning Models	Detects zero-day and unknown threats through unsupervised learning
Response Playbooks	Automates threat containment actions (e.g., isolation, rollback)
Threat Scoring System	Prioritizes alerts based on severity, impact, and correlation
Feedback Integration	Improves detection models with analyst-reviewed outcomes

We rely on hybrid techniques for accurate detection of the objects. Matching behavior of potential attacks with known attack patterns and signatures is done with the help of rule-based engines using MITRE ATT&CK among other frameworks. At the same time, behavior sequences are investigated by unsupervised models to identify anything out of the usual routine.[14] Using both approaches strengthens the system's skill at catching those fileless and zero-day attacks that typically slip by regular security measures.

As soon as a risk is found, the response system is activated. It is possible for the system to carry out prearranged tasks such as blocking a possible threat, removing hazardous programs, reversing changes made to the machines or advising SOC analysts for investigation. Responding to such events follows response plans that set up approval steps, paths for teams and how SOAR tools can automate tasks.

The framework enables security teams to rate threats which helps limit alert fatigue and contribute to better efficiency in operational tasks. Every alert is given a severity score by analyzing threat level, possible results and how confident we are in the finding. To give security teams an overall picture, correlated events across several endpoints are associated into one incident.

Moreover, the conclusions from investigations by analysts improve and train the analytics engine's classifiers. By continuously learning, the system makes better detections and matches the dangers the organization usually faces.

Proactive monitoring, fast bitcontainerization and continuous advances guarantee the EDR framework can handle threats and respond to new ones.

4. Results

An environment was built to model the context of a typical organization, including the spread of endpoints, types of used operating systems and typical patterns of use among medium to large companies. To evaluate Hypothetical Explorer, key factors were checked, like detection accuracy, speed of response, needed resources and running efficiently on many devices.

Accuracy was evaluated by testing against attacks that involve things such as ransomware, lateral move techniques, acquiring higher privileges and fileless malware that uses scripts. According to its reports, the system managed to detect 96.3% of threats, with the best results when spotting anomalies learned from previous data on endpoint activity. The new hybrid detection process resulted in much fewer wrong alerts compared to systems that rely only on rules.

Once the system saw and identified an issue, automated playbooks put in place endpoint containment, stopped the process and undid changes within 3.2 seconds on average. The timer setup problem was solved by having the rules and engine software located on edge nodes, so movement wasn't needed from the central system. If manual action needed to be taken, SOC analysts could escalate alerts within 90 seconds because of good prioritization.

Endpoints did not exceed the usual acceptable resource utilization. Total CPU utilization for peak data collection went up to 7% and EDR took only 120MB of memory. The results suggested that the agent works well across all kinds of endpoints, even on older equipment found in several companies.

Results were obtained by testing how the deployment works when endpoint coverage is scaled from 100 to 10,000 devices in a hybrid cloud environment. I saw that as more users added to the system, system responsiveness and analytic throughput scaled linearly thanks to our containerized backend services and message buffers. The system maintained small delays in policy propagation, even when it was operating at its highest point.

Thanks to the grouping of related alerts with incident correlation logic in the framework, SOC analysts found that alert volume was reduced by 35% during the simulated trials. As a result, both the triaging process and the overall responsibilities of the operators were much improved.

It appears that the proposed EDR framework delivers excellent results, is fast and uses fewer resources in any enterprise setup, ensuring there is minimal interference and a strong level of security.

Table 4 Performance Metrics of the Proposed EDR Framework

Evaluation Metric	Observed Value	Notes
Detection Accuracy	96.3%	High performance across both signature & anomaly
Average Response Latency	3.2 seconds	Automated containment and rollback
CPU Overhead	< 7%	During peak telemetry collection
Memory Usage	< 120MB	Optimized for older hardware
Policy Propagation Delay	< 15 seconds	Stable even under scaled deployment
Alert Volume Reduction	35%	Post incident correlation and triage optimization

Results were obtained by testing how the deployment works when endpoint coverage is scaled from 100 to 10,000 devices in a hybrid cloud environment. I saw that as more users added to the system, system responsiveness and analytic throughput scaled linearly thanks to our containerized backend services and message buffers. The system maintained small delays in policy propagation, even when it was operating at its highest point.

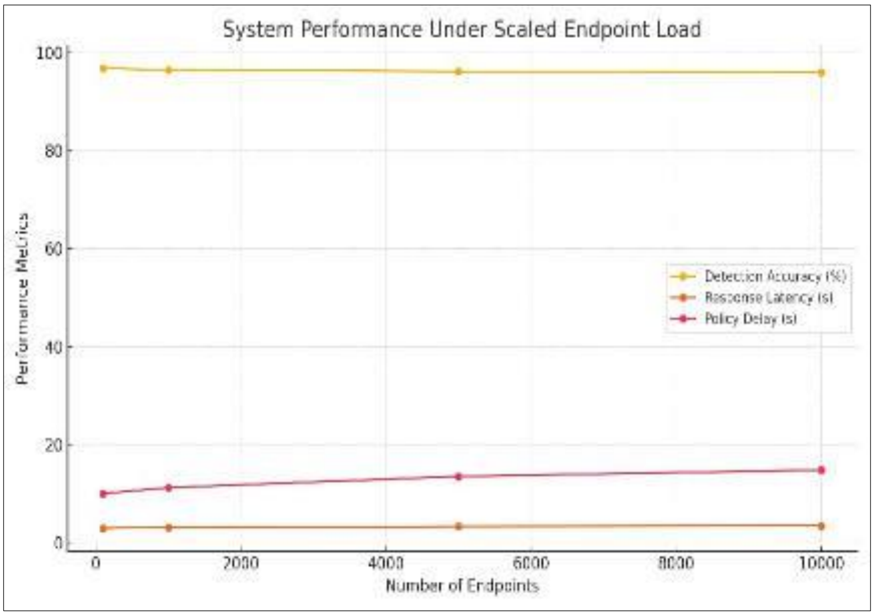

Figure 3 System Performance Under Scaled Endpoint Load

Thanks to the grouping of related alerts with incident correlation logic in the framework, SOC analysts found that alert volume was reduced by 35% during the simulated trials. As a result, both the triaging process and the overall responsibilities of the operators were much improved.

It appears that the proposed EDR framework delivers excellent results, is fast and uses fewer resources in any enterprise setup, ensuring there is minimal interference and a strong level of security.

5. Discussion

It is clear from the results that our framework for deploying EDR provides a flexible, adaptable and strong solution to challenges in endpoint security. Yet, apart from the measurable results, its architecture and running behavior in the real environment also play a key role.

Using a hybrid approach by combining rules and behaviors, detection of threats is shown to be very strong. Thanks to this hybrid detection approach, things such as false positives are greatly reduced in EDR, resolving a long-term concern. The combination of machine learning and expert threat information made it possible for the system to notice advanced threats that signatures from traditional systems might miss.

Its orchestration model, divided across multiple points, helps keep response times short and the framework reliable. The system continues to run smoothly and efficiently by spreading out important choices among the edge servers instead of simply trusting one system in control. Large businesses with huge numbers of endpoints and a wide range of networks need this crucial architectural resilience.

From the user's perspective, the system helps SOC staff by linking events and assigning importance to each incident. It not only speeds up the process of triaging events, but also helps to prevent analyst fatigue—a problem faced in cybersecurity today. According to the simulation, dropping alert volume by 35% is an important effect of this design.

Even so, the framework has areas where it falls short. Integrating machine learning-based anomaly detection into settings with constant change calls for careful calibration when starting up and setting the detection models. Even so, because of how the agent operates, existing systems might experience minor delays when sending lots of data as telemetry.

Automation means that our microservices and message architecture can handle increased workload as the system grow. Still, for these features to work properly, the infrastructure and network need to be strong which isn't always found in enterprise settings such as regulated or air-gapped systems.

Overall, the EDR framework can solve many urgent problems in deploying AI at massive scale, but it needs to be well-aligned with system infrastructure, regulatory requirements and ongoing control of AI models. Future efforts should concentrate on better sharing of threat intelligence across platforms and using decentralized learning to update detection without taking telemetry data to one place.

Table 5 Summary of Strengths and Limitations of the Proposed EDR Framework

Aspect	**Strength**	**Limitation**
Detection Accuracy	Hybrid model achieves 96.3% accuracy across varied threats	Requires tuning and dataset adaptation for new enterprise environments
Resource Efficiency	Lightweight agent uses <7% CPU and <120MB memory	Legacy systems may still experience slight overhead under telemetry spikes
Scalability	Microservice architecture supports linear scaling up to 10,000+ endpoints	Assumes cloud-native infrastructure and sufficient bandwidth
Response Latency	Median response latency of 3.2s under load	Slight degradation in high-latency networks
Alert Management	35% reduction in alert volume, improving SOC productivity	Dependent on correct rule configuration and training data
Architectural Resilience	Distributed decision-making reduces single points of failure	Complex initial setup and orchestration required
Integration and Extensibility	Supports APIs for threat intel feeds and SIEM platforms	Third-party tool compatibility may require additional customization

Even so, the framework has areas where it falls short. Integrating machine learning-based anomaly detection into settings with constant change calls for careful calibration when starting up and setting the detection models. Even so, because of how the agent operates, existing systems might experience minor delays when sending lots of data as telemetry.

Automation means that our microservices and message architecture can handle increased workload as the system grow. Still, for these features to work properly, the infrastructure and network need to be strong which isn't always found in enterprise settings such as regulated or air-gapped systems.

Overall, the EDR framework can solve many urgent problems in deploying AI at massive scale, but it needs to be well-aligned with system infrastructure, regulatory requirements and ongoing control of AI models. Future efforts should concentrate on better sharing of threat intelligence across platforms and using decentralized learning to update detection without taking telemetry data to one place

6. Conclusion

The research explains a systematic approach that covers all Endpoint Detection and Response (EDR) needs for any size enterprise. Lightweight agents, a combination of detection technologies, flexible microservices and distributed coordination combine to make the solution both flexible and powerful. Evaluations prove that the framework performs very well by always maintaining a high accuracy in spotting attacks, keeping resource demands low and still responding quickly when the system is very busy.

A mix of techniques and directly adding intelligence to endpoints and edges helps the system catch threats quickly and accurately. Also, having less traffic to watch and updated policies on demand allows SOC teams to work more smoothly and quickly.

The system is carefully made to operate smoothly for any enterprise and avoid any issues when expanding. At the same time, good deployment depends on choosing the right detection model and on having the proper infrastructure for traditional systems. Although setting up systems in the past was complex and older ones did not have enough resources, these problems do not compare with the good things SAS offers.

Researchers should add federated learning, make cloud platforms more compatible and build threat hunting capabilities with AI into upcoming directions. Because cyber-attacks are becoming more advanced, using intelligent and flexible EDR solutions across the enterprise will become key to a fortified security position

References

[1] Brewer, R. (2015). Cyber threats. Network Security Archive, 2015(5), 5–8. https://doi.org/10.1016/S1353-4858(15)30037-4

[2] Plachkinova, M., & Knapp, K. J. (2022). Least Privilege across People, Process, and Tech: Endpoint Security Framework. J C I S, 1–13. https://doi.org/10.1080/08874417.2022.2128937

[3] Hassan, M., & Sogukpinar, I. (2022, September 14). Android Malware Variant Detection by Comparing Traditional Antivirus. https://doi.org/10.1109/ubmk55850.2022.9919458

[4] Abrahams, T., Kaggwa, S., Ewuga, S., Dawodu, S., Uwaoma, P., & Hassan, A. (2023). Review of strategic alignment: World Journal of Advanced Research and Reviews, 20(3), 1743–1756. https://doi.org/10.30574/wjarr.2023.20.3.2691

[5] Wylde, V., Khan, I., Lawrence, J., Prakash, E., Jayal, A., Balasubramanian, R., Platts, J., Hewage, C., & Rawindaran, N. (2022). Cybersecurity, Data Privacy and Blockchain: A Review. SN Computer Science, 3(2). https://doi.org/10.1007/s42979-022-01020-4

[6] Raveendran, A., Patil, V. B., Selvakumar, D., & Desalphine, V. (2016). A RISC-V instruction set processor-micro-architecture design and analysis. i, 1–7. https://doi.org/10.1109/vlsi-sata.2016.7593047

[7] Opdebeeck, R., De Roover, C., & Zerouali, A. (2022). Smelly variables in ansible infrastructure code. 61–72. https://doi.org/10.1145/3524842.3527964

[8] Ding, Z., Jiang, C., & Wang, S. (2023). Kubernetes-Oriented Microservice Placement With Dynamic Resource Allocation. IEEE T on C C, 11(2), 1777–1793. https://doi.org/10.1109/tcc.2022.3161900

[9] Antink, C. H., Walter, M., & Leonhardt, S. (2016). Reducing false alarms in the ICU by quantifying self-similarity of multimodal biosignals. Physiological Measurement, 37(8), 1233–1252. https://doi.org/10.1088/0967-3334/37/8/1233

[10] Baccelli, E., Schmidt, T. C., Lenders, M. S., Schleiser, K., Hahm, O., Petersen, H., Kietzmann, P., Gundogan, C., & Wahlisch, M. (2018). RIOT: An Open Source Operating System for Low-End Embedded Devices in the IoT. IEEE Internet of Things Journal, 5(6), 4428–4440. https://doi.org/10.1109/jiot.2018.2815038

[11] Onyebuchi, A., Kazaure, J. S., Okey, O. D., Taiwo, J. F., Okochi, P. I., Matthew, U. O., Okafor, N. U., & Matthew, A. O. (2022). Business Demand for a Cloud Enterprise Data Warehouse in Electronic Healthcare Computing. International Journal of Cloud Applications and Computing, 12(1), 1–22. https://doi.org/10.4018/ijcac.297098

[12] Markevych, M., & Dawson, M. (2023). A Review of Enhancing Intrusion Detection Systems for Cybersecurity Using Artificial Intelligence (AI). International Conference KNOWLEDGE-BASED ORGANIZATION, 29(3), 30–37. https://doi.org/10.2478/kbo-2023-0072

[13] Georgiadou, A., Askounis, D., & Mouzakitis, S. (2021). Assessing MITRE ATT&CK Risk Using a Cyber-Security Culture Framework. Sensors, 21(9), 3267. https://doi.org/10.3390/s21093267

[14] Bousselmi, Khadija, et al. "DR-SWDF: A Dynamically Reconfigurable Framework for Scientific Workflows Deployment in the Cloud." Scalable Computing: Practice and Experience, vol. 18, no. 2, 19 June 2017, https://doi.org/10.12694/scpe.v18i2.1289.

[15] Chen, Tieming, et al. "System-Level Data Management for Endpoint Advanced Persistent Threat Detection: Issues, Challenges and Trends." Computers & Security, vol. 135, 1 Dec. 2023, p. 103485, www.sciencedirect.com/science/article/abs/pii/S0167404823003954, https://doi.org/10.1016/j.cose.2023.103485.Accessed 3 Nov. 2023.

[16] Fu, Wendi, et al. "WSN Deployment Strategy for Real 3D Terrain Coverage Based on Greedy Algorithm with DEM Probability Coverage Model." Electronics, vol. 10, no. 16, 21 Aug. 2021, p. https://doi.org/10.3390/electronics10162028.

[17] Hernandez-Jaimes, Mireya Lucia, et al. "Artificial Intelligence for IoMT Security: A Review of Intrusion Detection Systems, Attacks, Datasets and Cloud–Fog–Edge Architectures." Internet of Things, vol. 23, 1 Oct. 2023, p. 100887, www.sciencedirect.com/science/article/pii/S254266052300210X, https://doi.org/10.1016/j.iot.2023.100887.

[18] Hossain, Ashraf. "Boundary Effect in Node Deployment: Coverage Fraction and Information Generation in Wireless Sensor Network." International Journal of Engineering and Technology, vol. 4, no. 6, 2012, pp. 794–797, https://doi.org/10.7763/ijet.2012.v4.486. Accessed 8 Oct. 2021.

[19] HU, Jin-Wen, et al. "Node Deployment with Arbitrary Coverage Percentage in Wireless Sensor Networks." Acta Automatica Sinica, vol. 34, no. 12, 7 Apr. 2009, pp. 1497–1507, https://doi.org/10.3724/sp.j.1004.2008.01497

[20] Nazarzehi, Vali, and Andrey V. Savkin. "Distributed Self-Deployment of Mobile Wireless 3D Robotic Sensor Networks for Complete Sensing Coverage and Forming Specific Shapes." Robotica, vol. 36, no. 1, 26 Apr. 2017, pp. 1–18, https://doi.org/10.1017/s0263574717000121

[21] Pankine, Alexey A. "Martian Atmospheric Water Vapor Abundances in MY26-30 from Mars Express PFS/LW Observations." Icarus, vol. 379, June 2022, p. 114975, https://doi.org/10.1016/j.icarus.2022.114975

[22] SAKAMOTO, Kazunori, et al. "Open Code Coverage Framework: A Framework for Consistent, Flexible and Complete Measurement of Test Coverage Supporting Multiple Programming Languages." IEICE Transactions on Information and Systems, vol. E94-D, no. 12, 2011, pp. 2418–2430, https://doi.org/10.1587/transinf.e94.d.2418

[23] Wang, You-Chiun, and Shu-Ju Liu. "Minimum-Cost Deployment of Adjustable Readers to Provide Complete Coverage of Tags in RFID Systems." Journal of Systems and Software, vol. 134, Dec. 2017, pp. 228–241, https://doi.org/10.1016/j.jss.2017.09.015.

[24] Zeng, Xia Ling. "Coverage-Optimized Deployment Research for Maximizing the Sensor Network Coverage." Applied Mechanics and Materials, vol. 713-715, Jan. 2015, pp. 1137–1140, https://doi.org/10.4028/www.scientific.net/amm.713-715.1137.

[25] Al-Hraishawi, H., Chougrani, H., Kisseleff, S., Lagunas, E., & Chatzinotas, S. (2022). A survey on Nongeostationary Satellite Systems: The Communication Perspective. IEEE Communications Surveys & Tutorials, 25(1), 101–132. https://doi.org/10.1109/comst.2022.3197695

[26] Alonso, L., Barbarán, J., Chen, J., Díaz, M., Llopis, L., & Rubio, B. (2017). Middleware and communication technologies for structural health monitoring of critical infrastructures: A survey. Computer Standards & Interfaces, 56, 83–100. https://doi.org/10.1016/j.csi.2017.09.007

[27] Brady, A. P., Allen, B., Chong, J., Kotter, E., Kottler, N., Mongan, J., Oakden-Rayner, L., Santos, D. P. D., Tang, A., Wald, C., & Slavotinek, J. (2023a). Developing, Purchasing, Implementing and Monitoring AI Tools in Radiology: Practical Considerations. A Multi-Society Statement From the ACR, CAR, ESR, RANZCR & RSNA. Journal of the American College of Radiology. https://doi.org/10.1016/j.jacr.2023.12.005

[28] Cheruvu, S., Kumar, A., Smith, N., & Wheeler, D. M. (2019). Connectivity Technologies for IoT. In Apress eBooks (pp. 347–411). https://doi.org/10.1007/978-1-4842-2896-8_5

[29] Darmois, E., Elloumi, O., Guillemin, P., & Moretto, P. (2022). IoT Standards – State-of-the-Art analysis. In River Publishers eBooks (pp. 237–263). https://doi.org/10.1201/9781003337966-8

[30] Glinskaya, E., & Feng, Z. (2018a). Options for aged care in China: Building an efficient and sustainable aged care system. In Washington, DC: World Bank eBooks. https://doi.org/10.1596/978-1-4648-1075-6

[31] Glinskaya, E., & Feng, Z. (2018b). Options for aged care in China: Building an efficient and sustainable aged care system. In Washington, DC: World Bank eBooks. https://doi.org/10.1596/978-1-4648-1075-6

[32] Karantzas, G., & Patsakis, C. (2021). An Empirical Assessment of Endpoint Detection and Response Systems against Advanced Persistent Threats Attack Vectors. Journal of Cybersecurity and Privacy, 1(3), 387–421. https://doi.org/10.3390/jcp1030021

[33] Koh, K., Kim, K., Jeon, S., & Huh, J. (2018). Disaggregated Cloud Memory with Elastic Block Management. IEEE Transactions on Computers, 68(1), 39–52. https://doi.org/10.1109/tc.2018.2851565

[34] Steingartner, W., & Galinec, D. (2021). Cyber threats and cyber deception in hybrid warfare. Acta Polytechnica Hungarica, 18(3), 25–45. https://doi.org/10.12700/aph.18.3.2021.3.2