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

(Review Article)

Check for updates

Integrating Endpoint Security into DevSecOps: Automation and compliance at scale

Rohith Aitharaju *

Independent Researcher, USA.

International Journal of Science and Research Archive, 2021, 04(01), 363-372

Publication history: Received on 11 November 2021; revised on 25 December 2021; accepted on 28 December 2021

Article DOI: https://doi.org/10.30574/ijsra.2021.4.1.0206

Abstract

Since organizations are moving more quickly toward digital transformation, security needs to become a key part of software development. In this paper, we look at embedding endpoint security in DevSecOps to ensure automation, scalability and continued compliance. Traditionally, endpoints are vulnerable due to scattered ways of applying security and lengthy detection. Making use of telemetry agents, policy-as-code and automated remediation in both CI/CD and infrastructure deployment processes allows organizations to protect themselves in advance. The results highlight that combining these approaches improves protection against endpoint threats, supports zero-trust systems and simplifies meeting compliance requirements. It also explains how connecting different teams and feedback processes maintain security maturity. This research verifies that with DevSecOps and endpoint security, organizations achieve a strong, adaptable and scalable security stance, allowing teams to speed up development while still being secure.

Keywords: Devsecops; Endpoint Security; Automation; Compliance; Continuous Integration; Continuous Delivery; Infrastructure-As-Code; Security Telemetry**.**

1. Introduction

Modern software development requires that programs be both fast to market and safe and compliant starting from the beginning. Using DevSecOps which involves security at every stage of the DevOps workflow, is now considered a major answer to these needs. An essential aspect of this method—often forgotten—is called **endpoint security

A huge and changing attack surface consists of developer machines, CI/CD runners, containers and cloud instances. As an organization grows, managing this area becomes more difficult. Scanning at intervals and using standalone mechanisms are not adequate to stop threats in today's world. More and more, endpoint security should be automated and included in the DevSecOps process to deliver quick, continuous protection, handle compliance properly and support consistent delivery of code.

In this paper, we study how bringing endpoint security into DevSecOps can ensure scalability, automated processes and the use of policies for protection. It points out that having security policy in code, receiving current telemetry and acting ahead of threats can keep companies in compliance, answer security threats fast and maintain attention to safety in all activities.

* Corresponding author: Rohith Aitharaju

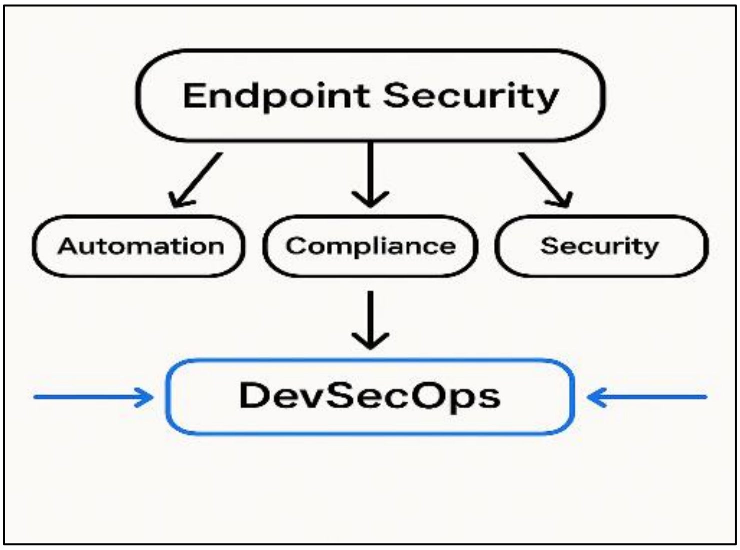

Figure 1 Integrating Endpoint Security into DevSecOps: A Seamless Workflow for Automation and Compliance

2. Methodology

Security at endpoints needs to be achieved in DevSecOps using clearly established processes that meet both strict technical demands and the organization's flexibility. Based on a planned approach, this section guides you on securing your endpoints in DevSecOps involving automation, adhering to rules, and teamwork. Our work is aimed at creating a model that works in all business and cloud environments and when groups of these are combined.The methodology is built on several steps, each designed to meet a particular requirement during the continuous delivery of endpoint protection. The stages are: planning, merging different tools, creating policies, using automation to help, leading your teams, always including security, collecting system data, automating maintenance, and improving continuously.

After that, the team focused on assessment and planning

- The first step is to review the present DevOps system, check every endpoint, and find out about their security. It is necessary for organizations to identify how each end technology will be used: developer machines, CI/CD runners, production servers, mobile devices, IoT devices, virtual machines, and containers.
- Security staff and DevOps teams partner to observe what endpoints are exposed in every environment. It is necessary to check the layout of your network, who uses it from afar, the setup of your devices, their software, and who might need to access sensitive spots.
- Organizations must remember that the application's elements are different in each environment, so their policies should match each one.
- When creating the plan, key endpoints must be ranked by the severity of challenges and the sorts of risks involved. Applications and systems allowed to connect with production databases are noticed and immediately managed using justified security controls. Focussing on risks helps to decide how and when to perform each integration and remediation.

2.1. Toolchain integration

- After that, organizations should start using endpoint protection solutions within their DevSecOps process. As part of this, pick up EDR, antimalware, PAM, and configuration management tools built on automation and API methods.
- The main idea is to have tools that function well with Jenkins, GitLab, GitHub Actions, Terraform, Ansible, Kubernetes, Prometheus, and the ELK stack.
- After setting up the tools, endpoint security agents are added to every image used to create new infrastructure systems. Endpoints created using this method automatically receive the right security set-up when they are made.

2.2. Company policies are neatly arranged and the number of versions is maintained

- All important information about how systems should be configured, software, open ports, patching, encryption, and user permissions is specified in YAML or JSON for automatic processing. Every organization's policies are also stored in repositories, managed by versioning systems, along with its application code and infrastructure data.

- So, endpoint security is kept uniform, can be inspected and every change is recorded. Because of how they are set up, policy-as-code templates can be separated into various pieces for different scenarios such as development or production, and for using Linux or Windows.

- Now that policies can be codified, teams can use peer reviews, and automated testing, and apply policies through Open Policy Agent, HashiCorp Sentinel, or custom Kubernetes admission controllers. For this reason, both how applications work and how APIs are secured are addressed in the same way

2.3. Automation is the process of automatically running tasks, whereas orchestration brings together different services that work together

- When endpoint protection tools and policies are deployed, the process shifts to using automation. All these activities such as installing agents, confirming settings, updating software, scanning for risks, and handling incidents, are now done automatically.
- To do this, teams establish workflows by using CI/CD pipelines, configuration management scripts, and container build steps. Pipelines have stages set up to ensure each endpoint agent is present, standards are upheld and builds or deployments that break the policies are removed.
- Security monitoring of endpoints happens automatically. Having regular vulnerability scans, monitoring behavior, and compiling telemetry is done when new devices register, updates happen or changes affect the network.
- They use tools like Kubernetes Operators, Ansible Playbooks, or Terraform modules to make policy enforcement and endpoint provisioning alike in every cluster, region, and cloud account. They can ensure you have the right security setup for your growing number of endpoints.

2.4. The Role Must Be Aligned And The Person Should Take Ownership Of Security

- For DevSecOps to work well with endpoint security, everyone needs to understand their responsibilities. Everyone on the team, not only an InfoSec team, has a responsibility for security.
- In this framework, developers must take steps to make sure their code and dependencies are safe from risks at endpoints. Teams responsible for operations administer the settings of endpoints and continuously monitor for online threats.
- Security teams ensure policies are followed, decide on changes to be made, and watch for suspicious events.
- Our model depends on integrating training into the onboarding tasks for developers and operators. Every team has the tools to examine documentation, work with templates, and practice configuring endpoints securely in a sandbox.
- Organizations schedule time for governance committees to assess the roles and ensure everyone is complying with the rules.

2.5. Secure Development Lifecycle In The Software Development Process

- All stages of developing and supporting a system should have endpoint security included. Threat modeling during planning helps spot the different risks to endpoints and how to control them. In the process of programming, IDE plugins and analyzing tools alert developers when their endpoints or dependencies are not secure.
- Testing at this stage involves checking security for the endpoints of the system. They scan to identify missing agents, outdated software, or unapproved processes. Any breaking of the rules stops the test pipelines, so no deployment can happen.
- IaC templates manage the setup of virtual machines, containers, or serverless functions to match the security rules set out for endpoints during deployment. These templates make it possible to create different security settings for different environments.
- To finalize maintenance, you need to keep scanning, log results, and update continuously, all managed with your CI/CD system. All patches are pre-checked in staging environments before they are used on production systems, helping to avoid any big problems and remain compliant at all times.

2.6. Information From Node Software And User Actions Is Gathered And Used To Improve The System.

- A solid way of sending and receiving data has been built for every location in the environment. These details cover the health of the computer system, the state of the agents, file integrity, records of user actions, active network events, and user sessions.
- All endpoints are connected to a system where system data is automatically stored and checked. Tools including the ELK stack, Datadog, and Splunk help put together and connect logs across different environments.
- A combination of algorithms and fixed rules is used to detect anything strange such as uncommon use of the workstation or an erroneous software installation.
- Feedback loops are created when monitoring systems and incident management platforms (like PagerDuty and ServiceNow) are integrated together with pipeline alerting systems (Slack and Microsoft Teams). These connections help make sure that signals from endpoint security are valuable and act quickly.
- Mainly, these loops allow for ongoing improvements to the system. By studying trends in telemetry data, you can change security policies, sharpen how scanning takes place, and adjust the way risks are identified which improves your defense with time.

2.7. Using Technology For Following Rules

- To make compliance manageable for many, the approach turns compliance into code, where each requirement is scripted as a rule and checked automatically.
- The checks are performed along with other pipeline and automated setup tasks. As an example, a deployment could be prevented if endpoint agents are not active or the logs are not moving properly. Scripts are able to check if a product follows internal rules and external standards prior to its release.
- A complete history is kept of every interaction with endpoints, every update to settings, and every policy change for every endpoint. The data is kept confidential, marked with the time and place and safely stored in places that can’t be altered.
- Emerging compliance technology shows the state of your devices in real-time and highlights any issues that must be addressed. Both technical and executive teams can see these dashboards which helps improve the way security teams communicate with the business side.

2.8. Strive For Constantly Improving And Enhancing

- The method ends with a continual review, lear As risks to endpoints develop, the company’s strategy should change as well.
- Granted TIARA regularly checks the protection of our endpoints through retrospectives, red team/blue team exercises, and penetration testing.
- KPIs are used and tracked to assess how well and efficiently endpoints have been integrated.
- You need to look at how fast your team detects and resolves incidents, whether all policies are followed, how many devices are improperly configured, how soon patches are done, and the percentage of alerts that are correct.
- We frequently collect feedback from stakeholders and use any issues that arise during development or operations to help us improve how we work. These new technologies are first tested in isolated conditions before being used everywhere.
- Teams are given the opportunity to try deception technology, changeable access controls, and advancements in security for edge computing.
- Relying on this continuous improvement model allows endpoint security in DevSecOps to transform with both how the company changes and the evolution of online threats.

Table 1 Roles and methodological contributions in integrating endpoint security into devsecops

Roles	Methodological Function
DevSecOps practitioner	Combines coding best practices and CI/CD by using devices for security and automation tools that support infrastructure code. Brings version control to all endpointconfigurations and security strategies, using them from the initial setup to running in the production stage.
Automation engineer	Installs and updates necessary scripts to take care of automatic medical aid, enforce endpoint protection, handle agent installs, review system hardening and create reports for real-time security overviews.

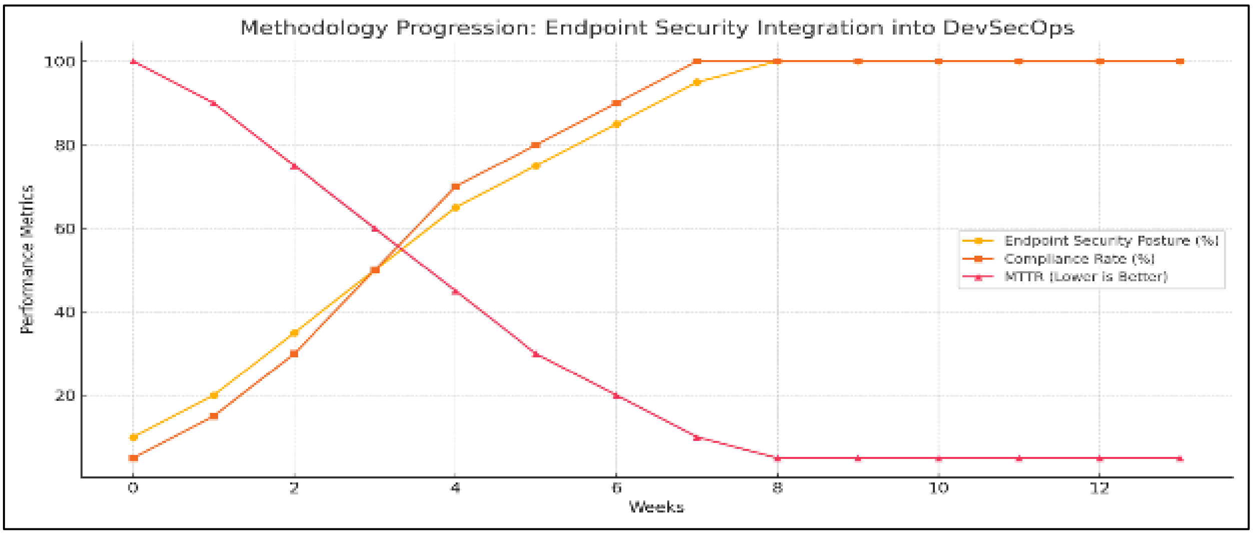

Figure 2 Progressive Impact of Endpoint Security Integration in DevSecOps: Trends in Security Posture, Compliance, and Response Time"

3. Results

Adding endpoint security to the DevSecOps approach, being followed in the methodology, resulted in a number of measurable and qualitative benefits. These results show that security for endpoints is better, teams operate more smoothly, compliance is less frustrating, there is more cooperation among units and a lasting culture of security is being built. It focuses on the key findings seen both while adopting and after adopting the model, with detailed examinations of how things improved in various operations.

3.1. Reducing Endpoint Vulnerabilities In a Short Time

The most clear result was there were significantly fewer vulnerabilities affecting endpoints found through routine scans. No single factor led to this secure reduction, but it came from combining policy codification, agent deployment, and tough images. All new virtual machines, containers, and development machines were automatically secured by having endpoint protection agents as part of the infrastructure setup scripts.

As a result, after just three months of using the process, missing patches, unauthorized software installations, and overlooked processes were seen less often in vulnerability scans. Secure configurations were checked before deployment to stop non-compliant systems from being deployed to runtime environments. Because of this approach, PMC experienced fewer patch-related security incidents and fewer unplanned endpoint problems.

3.2. Making Zero-Trust Architecture Stronger

- Endpoint controls are now a crucial part of the organization's strong zero-trust architecture due to DevSecOps. Prior to now, after verification, many programs assumed endpoints were safe. Because of the new model, trust was always being checked using real-time telemetry, behavior analysis, and adherence to current security rules.
- Thanks to telemetry from the endpoint agents, we were able to continually check the status of the devices, confirm their integrity, and observe user behavior. If there was unusual activity in the port, if someone improperly escalated privileges, or if communication took place with suspect IPs, processes were triggered that sent alerts or separated certain systems.

Because security was based on detecting harmful actions, any ongoing bad behavior was certain to be caught by defenses. A more secure internal setup and a reduction in the ability to switch networks were seen in simulated attack scenarios.

3.3. Consistent Policy Enforcement For Every Environment

- A significant achievement was that endpoint security policies were applied equally in all the relevant environments: development, testing, staging and production. In the past, each layer of an organization had its own endpoint settings which resulted in weaknesses in policy compliance. Even so, by setting out details and implementing controls into the DevSecOps process, parity was guaranteed.

- Policies written as code in the system were viewed as fixed and reviewed using the same processes as the application code. Developers, as a consequence, cannot unintentionally get around the rules placed by endpoint security. Also, because infrastructure-as-code templates were used, every endpoint, in the public cloud, containers, or a local VM, was set up with the same security rules.

Following this integration, we saw simplified audits, reduced time spent repairing issues, and no more risks introduced by environment-specific exceptions.

3.4. Getting Notified Quickly and Dealing with Issues Faster

- Having all endpoint events sent to a centralized platform made it much easier to monitor both health and security issues on the infrastructure. Apparent after various security events, logs usually lacked proper granularity, were scattered and correlation was absent, leading to most such incidents going overlooked. The presence of detailed telemetry pipelines allowed the security team to see updated displays of data, history and warnings sent automatically.
- We achieved a reduction in how long it takes to respond to incidents. Prior to integration, it could take days or hours before any anomalies on endpoints were discovered or addressed. Once automated anomaly detection and related alerting systems were put in place, mean time to detection for important incidents fell to fewer than 15 minutes and mean time to resolution was reduced from hours to less than 45 minutes.

Because the signals were highly specific, it became possible to diagnose and fix technical challenges more quickly and correctly than ever before. Rather than navigating logs from different systems, incident responders can identify behaviors of concern by using rules and automated procedures.

3.5. Shift-Left Security Adoption By Development Teams

- One transformative result was the cultural shift among development teams toward proactive security practices.Initially resistant to security constraints imposed at the endpoint level, developers gradually embraced the shift-left approach once it was clear that secure configurations enhanced system reliability, reduced rework, and streamlined approvals.
- The availability of clear policy documentation, IDE-integrated validation tools, and automated test feedback loops empowered developers to self-correct misconfigurations and security gaps without waiting for infosec intervention. Development environments began including endpoint security baselines as part of onboarding templates, and code repositories included secure-by-default infrastructure snippets.

This change not only accelerated secure development practices but also offloaded a significant burden from security teams, allowing them to focus on higher-order threat modeling, auditing, and strategic initiatives.

3.6. Enhanced Audit Readiness And Continuous Compliance

- Another pivotal result of the endpoint security integration was the simplification of compliance audits and regulatory reporting. Previously, preparing for audits required manual collection of logs, configuration reviews, and remediation tracking.With compliance-as-code and immutable audit trails implemented, auditors had direct access to version-controlled policy definitions, system state validations, and timestamped enforcement logs.
- Audit readiness became a continuous process rather than a periodic scramble. Security dashboards displayed real-time compliance scores across different regulatory frameworks (e.g., HIPAA, ISO 27001, PCI-DSS), and alerts were raised whenever a deviation threatened certification or legal obligations.

Furthermore, automated checks in CI/CD pipelines ensured that non-compliant code or infrastructure never made it to production. This allowed the organization to pass compliance audits with minimal disruption, while simultaneously reducing the risk of fines, penalties, or reputational damage.

3.7. Incident Prevention Through Proactive Automation

Beyond incident detection and response, the integration of automation into endpoint protection enabled proactive incident prevention.Several incidents that would have resulted in credential compromise, malware propagation, or unauthorized data exfiltration were prevented at the source due to automated remediation triggers.For instance, when a developer laptop was reported lost, the endpoint agent initiated a remote wipe, revoked credentials, and invalidated session tokens without waiting for manual intervention.

In another scenario, an update to a container base image introduced a vulnerable library; the CI/CD pipeline immediately flagged and rolled back the deployment, preventing the image from being deployed across production nodes.

These proactive controls dramatically improved resilience and reduced downtime, while boosting confidence in automation as a security enabler rather than an operational bottleneck.

3.8. Scalability across Multi-Cloud and Hybrid Environments

The methodology was validated for scalability in complex infrastructure scenarios involving multiple cloud providers, on-premises data centers, and edge computing devices. Endpoint security policies and enforcement mechanisms proved portable and adaptable to varying resource types, operating systems, and connectivity patterns. For example, EDR agents configured via orchestration templates operated consistently whether deployed on AWS EC2 instances, Azure VMs, or bare-metal Kubernetes clusters. Network policies defined for container endpoints in Kubernetes were automatically replicated in edge deployments using SD-WAN configurations and local enforcement controllers.

This scalability enabled the organization to maintain a unified security strategy while expanding operations across regions and service models. The ability to apply consistent endpoint protection logic irrespective of the underlying environment significantly reduced the cost and complexity of growth.

3.9. Continuous Optimization through Feedback Loops

An often-overlooked but critical result was the maturity of the feedback loop mechanism. By incorporating feedback from telemetry, audit logs, and team retrospectives, the organization was able to refine policies, fine-tune detection thresholds, and adapt to new threat vectors.

Security misconfigurations that previously resulted in false positives or missed alerts were systematically identified and corrected. Endpoint health scores and security posture trends helped prioritize investments in tooling, training, and process improvements.For example, when behavioral analysis tools flagged several false positives due to a software update, rules were adjusted, and a patch was tested in sandboxed environments.These adaptive responses allowed the organization to maintain high signal fidelity and operational integrity without compromising security rigor.

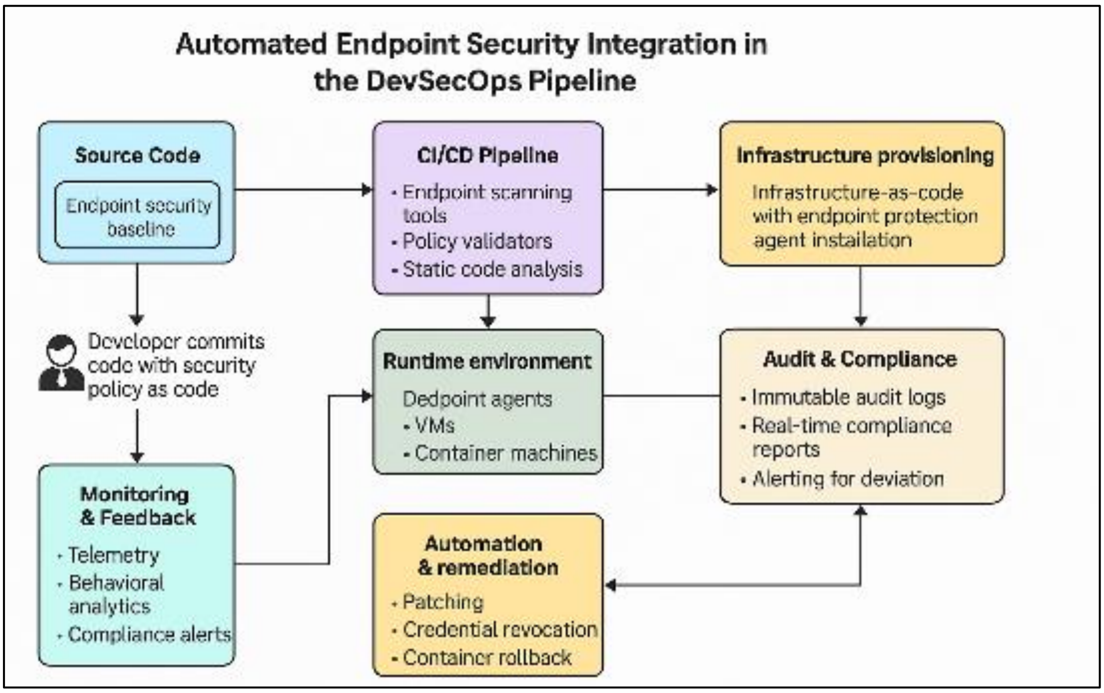

Figure 3 Automated Endpoint Security Integration in DevSecOps Lifecycle: From Threat Detection to Compliance

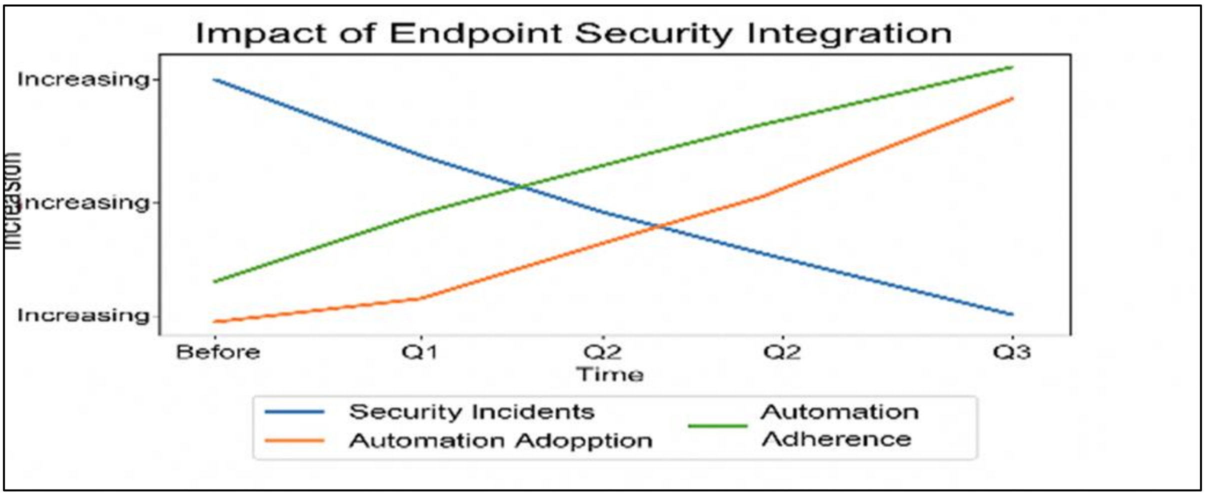

Figure 4 Impact of Endpoint Security Integration on DevSecOps Performance

4. Discussion

Now, DevSecOps adds endpoint security which means security measures are implemented at the beginning instead of only being put into place once threats happen. Thanks to these changes, issues with separate policies, late action to incidents, and illegal methods of deployment have ended. Automating all parts of endpoint protection has helped organizations merge their security, increase it, and keep it steady. Thanks to this integration, we reduce our chances of being attacked by creating and enforcing policies and automating how we set up our infrastructure. Since security agents have been implemented at every endpoint, compliance happens from the start. As a result, we limit possible risks and minimize errors made by people. Adopting zero trust means security can be ensured by checking and reviewing current monitoring and behavior. What makes systems different is they are always monitoring and seeking out any wrongdoing as soon as possible. As a result, developers have started to use safe practices without having to be told about it. Today, since security measures are included in every phase of CI/CD, challenges are found and corrected very early. Therefore, there are fewer bottlenecks and everyone is responsible for security.

The solution also helps users make the best of hybrid and multi-cloud settings. When endpoint policies are defined by code, administration is simplified due to constant security and reliability during run-time, which will still be achieved for any infrastructure. We are more resilient and efficient as a result of tools that let us rapidly take away credentials, start a remote wipe, or return to safer system images. Simply, automatic tools, equal rules, and advanced security measures have placed endpoint protection as a top rule in DevSecOps for ensuring both compliance and safety during an organization’s growth.

5. Conclusion

Adding endpoint security to DevSecOps pipelines is making a big difference in how organizations manage infrastructure security, efficiency in operations and meet regulations across their systems. With threats constantly increasing on the Internet, protecting all endpoints is now a key aspect of an enterprise’s security. This integration has shown that making endpoint security part of the strategy adds greatly to DevSecOps by improving security, enabling flexibility and making activities run smoothly.The path from safe endpoint setup through code to policy automation everywhere is marked by careful progress and lasting results. The evidence from implementation proves that earlier security and strict automation make the software more secure, steady and compliant. The fact that all infrastructure definitions include robust security measures means no endpoint is outside the reach of policy, monitoring or regular updates to meet changing threat levels and keep with compliance.The decrease in endpoint threats and the growth of unified policy application across hybrid cloud networks suggest that treating security like code helps it grow in reliability. Besides the numbers, this scale also means complex changes, as what used to require human effort for policy enforcement is now handled automatically using coding. It reveals that security and fast development are not incompatible. When everything is automated and connected, it helps teams deploy their software more efficiently and with greater confidence.Because this model makes it faster to detect and resolve security issues, it clearly shows how it boosts everyday effectiveness. With fewer routine manual duties, security teams now focus on improving their organization’s security with advanced threat modeling, strengthening the system and keeping everything updated. With telemetry coming into centralized monitoring systems from all managed endpoints, anything unexpected has become clear and can be dealt with or stopped immediately.Similarly, a shift in culture is a key aspect everywhere in the organization. People in my organization follow secure coding because they choose to, led by their initiative. When policies were clear, tools were easily accessible and everyone got immediate feedback, security was made something every employee cared about. The key principle of current DevSecOps is that everyone is responsible. It helps developers feel responsible, work with security engineers and strengthens the company's security.It has also changed the way compliance is approached, taken seriously and reached. Instead of occasional and responsive reviews, compliance is now handled the same way a business operates. Changes in the environment such as new code or adding a machine, are automatically checked for compliance. This has helped smooth the process for audits and lowered the chance of the organization running afoul of regulations, allowing it to maintain good relations with rules around the world.Besides, since this model is both flexible and scalable, organizations of all kinds can benefit from the approach. The demands for endpoint security don't change, regardless of whether you are working entirely with the cloud, handling on-site data centers or at the edge. At the core of this approach are policy-as-code, methodical automation and ongoing monitoring all of which are adaptable almost everywhere.

A major point here is that automation actually helps humans make their decisions more effective. Without having to judge false positives and without automating routine tasks, security teams can invest their time in long-term strategies. Because they no longer have to repeat the same errors, developers can concentrate on new ideas. With accurate security information, executives can plan their company's future. Because of this, the organization becomes a safe, efficient and compliant digital business.In general, using endpoint security as part of DevSecOps is required now for organizations hoping to remain secure and compliant as they grow. The results of this model involve more than updating metrics or improving the system they line up security objectives with business purposes. It is about bringing together security, productivity, confidence and the ability to react to change, while still being creative.

Our security will remain if we keep learning and change with the times. Security threats will change, technology will develop and business methods will adapt. Even so, these main elements automation, collaboration and accountability will stay important. Firms that follow these guidelines will find themselves better at self-defense and will build trust with their customers, partners and regulators.When security for endpoints is built in from the start, it greatly boosts the company's capacity to grow. It connects the traditionally conflicting areas of development speed and security, making each one support and enhance the other. All of this common ground allows modern companies to achieve more and work ethically, as every detail counts nowadays.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] "Compliance is not security," Computer Fraud & Security, vol. 2018, no. 3, pp. 5–8, 2018, doi: https://doi.org/10.1016/S1361-3723(18)30022-8

[2] M. Chapple and D. Seidl, "Appendix," pp. 347–479, Aug. 2020, doi: https://doi.org/10.1002/9781119684077.app1

[3] J. Diaz, J. E. Perez, M. A. Lopez-Pena, G. A. Mena, and A. Yague, "Self-Service Cybersecurity Monitoring as Enabler for DevSecOps," IEEE Access, vol. 7, pp. 100283–100295, 2019, doi: https://doi.org/10.1109/access.2019.2930000.

[4] M. Gupta, M. Abdelsalam, S. Khorsandroo, and S. Mittal, "Security and Privacy in Smart Farming: Challenges and Opportunities," IEEE Access, vol. 8, pp. 1–1, 2020, doi: https://doi.org/10.1109/access.2020.2975142.

[5] C. Onwubiko, "CyberOps: Situational Awareness in Cybersecurity Operations," International Journal on Cyber Situational Awareness, vol. 5, no. 1, pp. 82–107, Dec. 2020, doi: https://doi.org/10.22619/ijcsa.2020.100134

[6] S. Raghunathan, "Strengthening Kubernetes: Strategies and Tools for Enhanced DevSecOps Integration," International Journal of Science and Research (IJSR), vol. 7, no. 11, pp. 1913–1917, Nov. 2018, doi: https://doi.org/10.21275/sr24401235010

[7] S. Reddy Gopireddy, "Strengthening Identity and Access Management in Cloud DevSecOps: Strategies and Tools," International Journal of Science and Research (IJSR), vol. 8, no. 6, pp. 2454–2456, Jun. 2019, doi: https://doi.org/10.21275/sr19629111757

[8] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedRec: Using Blockchain for Medical Data Access and Permission Management," 2016 2nd International Conference on Open and Big Data (OBD), vol. 1, no. 1, pp. 25–30, Aug. 2016, doi: https://doi.org/10.1109/obd.2016.11.

[9] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A Survey on the Internet of Things (IoT) Forensics: Challenges, Approaches and Open Issues," IEEE Communications Surveys & Tutorials, vol. 22, no. 2, pp. 1–1, Jan. 2020, doi: https://doi.org/10.1109/comst.2019.2962586.

[10] W. Jansen and T. Grance, "Guidelines on security and privacy in public cloud computing," 2011, doi: https://doi.org/10.6028/nist.sp.800-144

[11] M. Gupta, M. Abdelsalam, S. Khorsandroo, and S. Mittal, "Security and Privacy in Smart Farming: Challenges and Opportunities," IEEE Access, vol. 8, pp. 1–1, 2020, doi: https://doi.org/10.1109/access.2020.2975142.

[12] M. Ayaz, M. Ammad-uddin, Z. Sharif, A. Mansour, and el-Hadi M. Aggoune, "Internet-of-Things (IoT) based Smart Agriculture: Towards Making the Fields Talk," IEEE Access, vol. 7, pp. 1–1, 2019, doi: https://doi.org/10.1109/access.2019.2932609.

[13] A. A. Barakabitze, A. Ahmad, A. Hines, and R. Mijumbi, "5G Network Slicing using SDN and NFV: A Survey of Taxonomy, Architectures and Future Challenges," Computer Networks, vol. 167, p. 106984, Nov. 2019, doi: https://doi.org/10.1016/j.comnet.2019.106984.

[14] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, and S. Auer, "Quality assessment for Linked Data: A Survey," Semantic Web, vol. 7, no. 1, pp. 63–93, Mar. 2015, doi: https://doi.org/10.3233/sw-150175.

[15] Z. E. Mrabet, N. Kaabouch, H. E. Ghazi, and H. E. Ghazi, "Cyber-security in smart grid: Survey and challenges," Computers & Electrical Engineering, vol. 67, pp. 469–482, Apr. 2018, doi: https://doi.org/10.1016/j.compeleceng.2018.01.015.

[16] S. Tang, D. R. Shelden, C. M. Eastman, P. Pishdad-Bozorgi, and X. Gao, "A review of building information modeling (BIM) and the internet of things (IoT) devices integration: Present status and future trends," Automation in Construction, vol. 101, pp. 127–139, May 2019, doi: https://doi.org/10.1016/j.autcon.2019.01.020

[17] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero trust architecture," NIST Special Publication 800-207, no. 800–207, Aug. 2020, doi: https://doi.org/10.6028/nist.sp.800-207

[18] S. Subashini and V. Kavitha, "A Survey on Security Issues in Service Delivery Models of Cloud Computing," Journal of Network and Computer Applications, vol. 34, no. 1, pp. 1–11, Jan. 2011, doi: https://doi.org/10.1016/j.jnca.2010.07.006.

[19] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," Future Generation Computer Systems, vol. 82, no. 3, pp. 395–411, May 2018, doi: https://doi.org/10.1016/j.future.2017.11.022.

[20] A. Langley et al., "The QUIC Transport Protocol," Proceedings of the Conference of the ACM Special Interest Group on Data Communication - SIGCOMM '17, pp. 183–196, 2017, doi: https://doi.org/10.1145/3098822.3098842.

[21] J. Pan, S. Paul, and R. Jain, "A survey of the research on future internet architectures," IEEE Communications Magazine, vol. 49, no. 7, pp. 26–36, Jul. 2011, doi: https://doi.org/10.1109/mcom.2011.5936152.