Information Security Management through Reflexive Security — Six Pillars in the Integration of Security, Development and Operations

CSA cloud security alliance®

The permanent and official location for Cloud Security Alliance DevSecOps is https://cloudsecurityalliance.org/group/DevSecOps/

Acknowledgements

Authors:

Michael Roza
Sean Heide
David Lewis
Eric Gauthier

Editor:

Ronald Tse

Table of contents

Introduction

Background

In today’s hyper-competitive context with increasing cybersecurity threats, especially in the cloud, organizations are under strong pressure to streamline information security management processes. The age-old problem of resource limitations is exacerbated — the tradition of information security management systems (ISMS) is seemingly too rigid in structure, yet inadequate in responsiveness to new needs.

Organizations today are faced with a number of information security management issues, including:

- Spiralling compliance governance costs;
- Global shortage of information security professionals, forcing more efficient utilization of staff;
- The notable disconnect between strategic security and operational security.

All of these factors pressure an organization’s information security processes to deliver more for less.

Learning from the Collaborative Philosophy

DevOps, the practice of applying developmental best practices such as collective collaboration to infrastructure operations, has been shown to positively impact efficiencies of development and operations teams today, especially in the cloud environment.

Given its strengthening adoption, it is necessary to consider its impact on information security itself and look to apply those practices to the arena of information security management.

Distilling from the cultural and philosophical elements of such collaborative practices, as well as from the exemplary contexts where those practices were derived from, six major principles are described in this document to demonstrate the principles of achieving

Application to security

DevOps is now broadly practiced but it has been generally separated from security practices. There is currently no standardized term in industry that cater to this aspect. A number of terms have been proposed by members of the community, including “DevOpsSec”, “DevSecOps”, “SecDevOps”, to explain the intersection of security and DevOps, as well as the amalgamation of security, development and operations practices.

These terms have been used interchangeably, but the meanings they convey can be vastly different. The definitions of these terms can also widely conflict with each other depending on the particular understanding of the reader. Moreover, there is no industry recognized definition for security automation, which is a core concept to the application of DevOps practices to security operations, leading to confusion and misinterpretations. It is therefore crucial to clarify and standardize such terms for a global and wider industry audience.

Target Audience

The target audience of this document includes those involved in the management and operational functions of risk, information security and information technology. This includes the C-suite (CISO, CIO, CTO, CRO, COO, CEO), and especially to the individuals involved in the following functional areas: automation, DevOps, quality assurance, InfoSec, governance, risk management and compliance.

1. Scope

This document defines “Reflexive Security” as a new security management approach that is built upon the interrelationships between security, development and operations necessary for protecting the security stance and the deliverables of an organization.

The benefits of combining security, development and operation practices for integrated management are also described.

Several approaches to Reflexive Security are suggested to resolve the availability gap of cyber security professionals.

2. Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 27000:2018, International Organization for Standardization and International Electrotechnical Commission. *Information technology — Security techniques — Information*

security management systems — Overview and vocabulary. Fifth edition. 2018. Geneva. https://www.iso.org/standard/73906.html .

3. Terms and definitions

For the purposes of this document, the following terms and definitions apply.

For the purposes of this document, the terms and definitions given in ISO 27000 and the following apply.

3.1.
DevOps
Application of software development methodologies to infrastructure operations.

3.2.
DevOpsSec
Application of information security principles and practices to protect processes that utilize DevOps culture, practices, and workflows

3.3.
DevSecOps (DSO)
The integration of continuous security principles, processes, and technologies into DevOps culture, practices, and workflows

3.4.
SecDevOps
Application of DevOps culture, practices, and workflows for the achievement of information security and compliance management

4. Reflexive Security

4.1. General

Reflexive Security is an approach for information security management built upon the principles of Agile and DevOps. It is a non-prescriptive framework that is purely needs-based, emphasizes collective responsibility, and considers information security and its responses to be a holistic function of the organization.

The word “Reflexive” is used for its meaning from the reflexive relation in mathematical sets, where every element in a such a relation is related to itself. In Reflexive Security, every action taken is related to the context and needs of the organization itself.

Reflexive Security emphasizes security across organizational roles that reacts to external and internal threats in an agile and dynamic way. It aims to be a new information security management strategy that is dynamic, interactive, effective and holistic.

4.2. Relationship with ISMS

Comparing to the traditional Information Security Management System (ISMS) approach, Reflexive Security aims to follow a similar spirit as Agile software development when compared to the Waterfall mindset.

Reflexive Security treasures dynamic information exchange within the organization. Like the immune system of the human body, information is immediately propagated to adjacent, potentially vulnerable functions given detection of a threat. A traditional ISMS approach is top-down: it requires threat information to be reported back to the higher functions, before any response can be made. In Reflexive Security, organization functions themselves are already integrated with security functions, and can provide an immediate response to address potential impact. A centralized response is important, but should not prohibit an immediate tactical approach.

The ISMS approach is about adopting best practices. However, often best practices that work in larger organizations do not work with smaller ones. In fact, they can create an unnecessary burden for smaller organizations, taking away precious time and effort and hence negatively affecting their information security stances.

Reflexive Security also alleviates the security skills shortage paradox by promoting the idea of “hybrid domain-security experts”, by delegating risks and security responsibilities to organization functions.

Top leadership commitment as required in an ISMS, while crucial, is insufficient. Top leadership has to understand the nature of information security and needs to be committed to integrating information security into each and every aspect of the organization. Top leadership must be themselves integrated into the information security function — fully understanding the risks and threats the organization faces (its risk profile) with regards to information security, setting clear bounds and requirements for personnel to adhere to.

The human link is often the weakest link in information security. Phishing, social engineering are all techniques to trick an unsuspecting individual to voluntarily compromise the security of an organization. Without full and active participation of the involved individuals, beyond awareness, information security simply cannot be achieved.

5. Six Pillars Supporting Reflexive Security

5.1. General

The six pillars can be abbreviated into the acronym RAMPAC (Responsible collectively, Automate, Measure, Pragmatic, Align and bridge, Collaborate and integrate).

5.2. Responsible collectively

Security leadership plays a shepherding role for information security within an organization. However, everyone is responsible for an organization's security. Each individual has their own security responsibility, and they must be aware of their own contribution (and potential problems) towards the organization's security stance. Edge users not only have to be security-aware, they are the first line of defence for the organization.

5.3. Pragmatic

Security should provide value, not a hindrance. For every security initiative or adopted security control, a cost-benefit should be prepared demonstrating the value of security. Without a pragmatic approach, it is easy for persons unaware of the benefits of security to dismiss security practices as mere overhead or unnecessary. A pragmatic approach provides crucial information in building business case evidence to convince stakeholders of the importance of security, giving them confidence that plans will be executed faithfully.

The traditional ISMS approach relies on a best practices approach which could easily lead to overhead. Rarely are organizations not resource-constrained, and it is necessary to balance tailoring requirements to the organization against the adoption of best practices. Requirements must be based on organizational risks and value for maximum effectiveness.

5.4. Align and bridge

Organizational risks and requirements must be fully aligned in order to derive maximum effectiveness and value from security processes. Security requirements that misalign with organizational risks represent resources potentially wasted, or worse, attention that has been taken away from critical matters that could cause organizational harm. By carefully aligning requirements throughout the organizational context, such as contractual, regulatory, product, security and compliance requirements, it is possible to extract maximum value by condensing a set of core requirements for the organization, allowing the delegation of risk management at more tactical levels for more dynamic responses and efficient usage of resources.

5.5. Automate

Automated security practices are the core of optimizing process efficiency. Processes that can be automated should be automated, and those that can't should be transformed. Automation also creates its own set of problems but automated approaches to those problems are often possible. A mantra in software development says, if one does the same thing three times, it will be time to program it. This approach applies squarely to Reflexive Security.

5.6. Measure and improve

Performance that cannot be measured cannot be improved. Without measurements, there is no understandable evidence whether information security management is effective. Reflexive Security is about achieving results, and the results must be measurable for transparency and improvement.

5.7. Collaborate and integrate

Security can only be achieved through collaboration, not confrontation. A security aware and collaborative culture is necessary for everyone to feel comfortable reporting potential anomalies. The human factor is often the weakest link — most security incidents are caused by inadvertent human error. Security must be integrated into every aspect of business. Dedicated security personnel rarely have in-depth business domain expertise. No one else knows the business domain and its risks as well as those responsible for it. A team of dedicated security experts are likely oblivious of domain-specific security issues. Domain knowledge and security knowledge must be fused and co-exist in every organizational process for effective security management.

6. Six Benefits of Reflexive Security

6.1. General

The six benefits can be explained with the acronym HEARTY (Human-centric, Elastic, Apt and holistic, Resilient, Tailored, Dynamic).

6.2. Human-centric

Security is integrated and internalized as an aspect of everyone's work, and requires mind-share within every employee. It is not an external responsibility enforced or required by a separate, dedicated, security function.

6.3. Elastic

Growing maturity of a Reflexive Security approach could lead to achievement of formal ISMS requirements, while being flexible enough to only target critical areas for maximum value based on actual risks. Its principles allow scaling according to organizational risks and needs, preventing permanent overhead in resource consumption.

6.4. Apt and holistic

Focused on business needs and responding to the actual risk context faced by the organization when compared to traditional information security management, from business goals to operational processes to humans, who often turn out to be the weakest link.

6.5. Resilient

Security no longer relies on a single security function, e.g. a single security department, but security practices are integrated with business processes and embedded throughout the organization. When compared to ISMS processes, an Reflexive Security approach often enables deeper security involvement amongst personnel, resulting in more resilient security activities.

6.6. Tailored

Prioritized approach to provision stronger protection to core or more vulnerable processes over those less exploitable. The traditional ISMS approach often applies the same set of controls across all processes, leading to inefficient resource allocation.

6.7. Dynamic

The protection of business goals is performed by integrating security with business processes, often down to the tactical level. This allows the organization to react faster and more effectively to threats and incidents.

7. Achieving Reflexive Security

7.1. General

Reflexive Security can be considered as the amalgamation and integration of the DevSecOps, DevOpsSec and SecDevOps practices, placed within a holistic framework. These areas of practice all derive from the integration of security and DevOps principles, together with essential concepts from Agile.

Concepts made apparent by Agile software development practices (where “Agile” is the generic term including practices like Scrum), such as incremental development, collaborative authoring, test- or behavior-driven programming, and continuous integration, have given rise to infrastructure operation practices such as continuous delivery, software-defined infrastructure and infrastructure automation.

These infrastructure-related practices are widely considered to be part of the “DevOps” repertoire, but with several conflicting definitions of what exactly the term represents, it is necessary to clarify and standardize its definition in order to evaluate and address its information security impact [1] .

[1] For example, the Wikipedia article on DevOps defines it as “a set of practices that emphasizes the collaboration and communication of both software developers and information technology (IT) professionals while automating the process of software delivery and infrastructure changes”. This definition is too vague for evaluating the impact of DevOps on information security practices.

Its practices can entail different processes and outcomes depending on organizational context. Its processes could provide organizational-wide benefits through empowering integration and interaction between development and operations (e.g., the “build” and “run” processes), or in some cases, have minimal impact in complex enterprise teams where logistics and governance controls limits the application of DevOps [2] .

The application of Agile and DevOps practices and related tools clearly have blended borders between software development and service infrastructure automation. Information security management and operations can benefit from an identical approach, with even tighter integration since it already depends on both software development and infrastructure automation.

The practices of DevSecOps, DevOpsSec and SecDevOps, as described below, provide necessary insights for the achievement of Reflexive Security. The three major aspects of how Agile and DevOps principles can help achieve Reflexive Security for information security management are shown below:

> - The impact of DevOps practices on information security
> - The application of DevOps to security operations
> - The implementation of information security processes and compliance delivery through DevOps

7.2. Securing DevOps practices

7.2.1. GENERAL

While DevOps practices can help improve the management and operations of information security processes in an organization, the execution of these practices have to be secured. Specifically, organizations have to ensure that DevOps practices introduced do not compromise information security. This aspect has been called “DevOpsSec”, or “DevOps security”. DevOps practices strongly impact existing infrastructure and operations on it. Inherent security requirements in DevOps processes must be considered in the adoption of DevOps within an organization.

7.2.2. IMPLICATIONS

A software-defined infrastructure is highly beneficial for the experienced infrastructure engineer, where software-defined instances, containers, and network devices are programmatically created or destroyed through fingertips on the keyboard.However, it is equally true that a novice engineer could have his fingertips spin things out of control.

[2] https://devops.com/enterprise-devops-doesnt-make-sense/

Security vulnerabilities can be inadvertently created due to lack of consideration of all aspects surrounding the infrastructure, for example, lax firewall rulesets, default credentials or an increased attack surface. While software development and infrastructure operation skills are closely related, mastery in one may not necessarily translate into proficiency of the other.

7.2.3. EXAMPLE PRACTICES

- Infrastructure
  - Security baseline checking on software-defined infrastructure
  - Automated security checks on DevOps infrastructure
- Containers
  - Security baseline checking
  - Automated hardening and management
  - Vulnerability checks and file scans
  - Automated integrity and authenticity checking
- Operating Systems / VM images
  - Automatic hardening during provisioning
  - Automated integrity and authenticity checking
- Secure Coding and Testing
  - In-field compliance testing
- Release Management
  - Automated vulnerability scans integrated with development and deployment workflow
- Business Continuity
  - Automated business continuity drills
  - Automated tests for backup and restore processes
- SSL Certificates and Public Key Infrastructure
  - Verification of certificate source and validity
  - Automated testing to ensure validity of issuer
- Identity and Access Management
  - Verification of account usage
  - Continuous scans for inactive or suspicious accounts
- Encryption
  - Automated verification of data encryption processes

7.3. Implementing security operations through DevOps

7.3.1. GENERAL

DevOps provides powerful concepts that link development and infrastructure operations, and its practices can be integrated to facilitate the achievement of information security within an organization. The field of “DevSecOps”, or “DevOps for Security Operations”, applies DevOps concepts to enhance the efficiency and effectiveness of information security processes. Its aim is to facilitate the comparable paradigm shift of DevOps on infrastructure operations to security operations, through the integration with development and infrastructure operations.

7.3.2. IMPLICATIONS

On a tactical level, DSO represents the integration and automation of security controls through DevOps using automated toolchains. The CSO sets the policies following organizational goals, requirements and risk management, which could be applied programmatically.

7.3.3. EXAMPLE PRACTICES

- Infrastructure
    - Immutable infrastructures
- QA
    - Implementation of data masking of data used in development for testing
- Containers
    - Vulnerability scanning during building of container images
    - Patch management of applications and libraries inside containers
    - Hardening / secure configuration, self-healing
    - PKI / Digital signatures
    - Anti-virus scan during building of container images
    - Certify container images
    - Scan for embedded keys, hardcoded credentials, push for role based access technology, licensing compliance
    - Cryptographically sign and certify container images
    - Runtime container security delivered by monitoring container activities
- Operating Systems / VM images
    - Vulnerability scanning during building of VM images
    - Patch management of applications and libraries of the operating system
    - Hardening / secure configuration, self-healing
    - Anti-virus scan during building of VM images
    - Cryptographically sign and certify VM images
- Secure Coding and Testing
    - Inline analysis of code against top 10 OWASP vulnerabilities and provide real time feedback to developer
    - Static code analysis after code commits, builds or releases
    - Scanning for embedded keys, hardcoded credentials, push for role based access
- Release Management
    - Create new public/private keys for each release
    - Release approvals
    - Security / compliance exception management based on the thresholds

    - Only accept signed/certified container and OS images
    - Revocation of older SSL certificates / private keys / PKI after each release
- Measures
    - Builds failed due to security issues
    - Releases with security exceptions
    - Deployment time and deployment frequency
    - Releases bypassed security checks
    - Code commits
    - Iteration length

7.4. Securing information security processes with DevOps

7.4.1. GENERAL

The practice of "SecDevOps", or "Security DevOps", aims to utilize DevOps practices to the implementation of information security processes. Similar to how DevOps concepts are able to improve the collaborative nature of development and infrastructure operations, SecDevOps integrates and facilitates collaboration of development with information security processes. A common result is the automation of information security processes (the "build"), accompanied by allowing development processes to be abreast of security implications and deliverable feedback.

7.4.2. IMPLICATIONS

"Security as Code" is not equivalent to security automation because it does not require the "programmatic" ability. For example, a scripted application of the DoD Security Technical Implementation Guides (STIG) may still have to be run manually during the installation of a server.

7.4.3. EXAMPLE PRACTICES

- Business Continuity
    - Automatic business continuity plan testing through Infrastructure as Code (IaC)
    - Continuous backup and restore testing with data masking
- SSL Certificates and Public Key Infrastructure
    - Monitoring and auto renewal of SSL certificates
    - Testing of expiry dates of SSL certificates
    - Private and public key rotation
- Identity and Access Management
    - Secrets management to effectively manage passwords, keys, API, tokens for application, services across IT landscape
- Infrastructure
    - Vulnerability scanning of servers, VMs, containers, appliances during runtime
    - Anti-virus scan during runtime

- Encryption
    - Encryption as a service for data in transit and data at rest
- Measures
    - Test coverage
    - Coverage of unit/integration tests
    - Coverage of functional/acceptance tests
    - CIS Benchmark, Compliance Violations
    - Time taken to achieve compliance for a new application
    - Security issues discovered across stages (aspire to zero at production)
    - Time to fix security issue

8. Relevant guidelines for implementing Reflexive Security

8.1. General

This clause provides some guidelines for organizations for the implementation of Reflexive Security. The guidelines below are meant to be indicative, not exhaustive.

8.2. Culture Shock

The goal of DevOps has been to write code and move product into production as fast as possible while maintaining a single source of truth, and removing or sidelining any barriers to their progress. The goal of security has always been too protect the organization internally and externally from access. Individually the goals are admirable until it gets to the point where DevOps deploys insecure software or security delays deployments to the point where the organization's goals are affected.

Embedding Security into DevOps can be a shock to the traditional system. Plan to deal with the issues of DevOps and Security working closely with common goals and metrics. Plan for the personal effects of automation: changed, lost and or new positions.

8.3. Cross-train

Form an operational security function by selecting key persons that have an interest or background in security from each team and delegate security related tasks to those individuals. Make these persons a representative of the operational security teams.

The operational security teams know exactly what types of operating systems are used, applications are supported for the business, etc. In some organizations, this collaboration involves embedding IT operations specialists within software development teams, thus forming a cross-functional team. This effort may also be combined with a skills matrix.

8.4. Delegate

Delegate risk management, maintenance of BCPs etc. to department heads. Freeing up resources of the dedicated security team so they can focus on the security process management and providing the framework(s). The department heads know exactly what they have in house, what risks there are.

Reflexive Security is process neutral. Organizations can use Agile, Scrum or any other development methodology. Have security engineers attend daily scrum meetings, make them part of the team.

8.5. Automate with purpose

One of the primary rules of DevSecOps is to automate early and often with the goal to embed security into the process but don't try to automate everything at once. Select those areas that will address the greatest risk for automation first. Also choose your automation tools using a selection method including testing that reflects the specifics of your environment.

9. Conclusion

Reflexive Security is a newer information security management strategy that is dynamic, interactive, holistic, and effective. It represents an integrated culture extrapolated from the amalgamation of DevSecOps, DevOpsSec and SecDevOps concepts and practices, and promotes the exemplary contexts from which those best practices were derived from.

Reflexive Security provides a set of wide-implicating and easily understandable principles that affect an organization's cybersecurity posture, especially suitable for operating under resource and personnel constraints in today's fast-paced and challenging cybersecurity landscape.

The implementation of DevSecOps, DevOpsSec and SecDevOps concepts and practices in an organization provides an excellent start in attaining the benefits of Reflexive Security. It is also straightforward for organizations that utilize Reflexive Security to formally adopt an ISMS as appropriate, by integrating ISMS processes as part of the Reflexive Security strategy.