УДК 004.056:004.415
DOI: 10.31673/2409-7292.2024.030004

**Pavlykevych A., Dzioban M.**

# EVENT-DRIVEN METHOD OF MEASURING EFFECTIVENESS OF INFORMATION SECURITY CONTROLS IN SOFTWARE DEVELOPMENT ENVIRONMENTS

With in growing cost of cybersecurity incidents, proper allocation of resources for information security controls (ISCs) becomes critical for every organization. This paper describes practical approach to measuring ISCs effectiveness within software development environments (SDEs) using method independent from control type and design and based on security events: an externally measured effects of control malfunctions. This together with suggested approach to calculating aggregated score for SDEs protection against categories of threats allows to develop actionable risk assessment (RA) framework for SDEs. Paper provides example of building such information security RA into overall SDE risk management framework. Proposed SDE RA methodology was implemented using Microsoft Power BI platform for analytics, with principal data supplied from SIEM (metrics on control effectiveness and adoption), ITSM (data on control implementation statis and assets per project), application used to conduct SDE RA assessments and risk options assignment. This paper demonstrates use of unified effectiveness measurement method for SDE ISCs based on security event and incident management (SIEM) as well as method of consolidating these measurements into high level metrics used to evaluate overall security of specific SDE or entire group of SDE owned by the organization. Method offers new approach for collecting meaningful benchmarking data from stakeholders without formal Information Security education and providing results, which can be directly used by non-IS professionals to drive management decisions within the organization.

**Keywords:** information security controls (ISC), cybersecurity, risk management, software development environments.

## Introduction

In today's digital landscape, the security of information is paramount for organizations as the frequency and sophistication of information security threats continue to escalate. Recent data highlights the critical nature of this issue. The volume of reported vulnerabilities continues to rise. The *Vulnerability and Threat Trends Report* (2023) reported a 25% year-over-year increase in the number of new vulnerabilities in the U.S. government's National Vulnerability Database from 2021 to 2022. Cybersecurity will remain a constant concern and there will be continued risk in 2024 from attacks against technology-enabled resources and services, including financial systems and communication infrastructure, according to the *Global Risks Report* (2023). The annual average cost of cybercrime is predicted to hit more than $23 trillion in 2027, up from $8.4 trillion in 2022, according to data cited by Anne Neuberger, U.S. Deputy National Security Advisor for cyber and emerging technologies (“Digital Press Briefing with Anne Neuberger, Deputy National Security Advisor for Cyber and Emerging Technologies,” 2023).

These disturbing statistics underscore the ongoing inadequacy of current information security practices and emphasize the urgent need for enhanced methods to safeguard sensitive and critical information. Effective implementation and operation of ISC are crucial for organizations to maintain a robust and secure information system environment. However, information security is most effective when only the most appropriate ISC are implemented. Existing literature indicates significant shortcomings in traditional ISC measurement methodologies, which often fail to facilitate an effective evaluation, prioritization, and implementation of ISC within organizations.

This research-in-progress relates to the development of a tool that can accurately measure ISC effectiveness in software development environments.

## Problem Statement

Existing approaches to ISC effectiveness evaluation rely either on specific measurements performed on each control or on various forms of expert assessments of entire control environment. Both approaches have number of deficiencies, which make them hard to implement at large number of organizations. Direct measurements require different and often sophisticated tools, which then produce results using incompatible metrics and dimensions, making any form of holistic scoring

© Pavlykevych, A., & Dzioban, M. (2024). Event-driven method of measuring effectiveness of information security controls in software development environments. Сучасний захист інформації, 3(59), 42–54. https://doi.org/10.31673/2409-7292.2024.030004.

challenging task for individual organization. While expert-based assessment cannot be performed frequently and lack granularity required to make continuous improvement decisions.

Goal of this research is to develop uniform approach towards measuring ISC effectiveness across all four categories of controls as defined by *ISO/IEC 27001* (2022): organisational, people, physical and technological.

## Related works

The purpose of this section is to review the literature about the measuring effectiveness of ISC in organizations. Following is a summary of the literature review pointing out the inadequacies of traditional ISC assessment methodologies.

### Best practices (Baseline Manuals or Frameworks)

Organizations frequently adopt industry-standard models and frameworks to integrate Information Systems Control  within their operations, as noted by Barnard (2000). These "best practices" typically include renowned examples such as the Information Technology Infrastructure Library (ITIL), Operationally Critical Threat, Asset and Vulnerability Evaluation (OCTAVE), standards from the National Institute of Standards and Technology (NIST), and the Control Objectives for Information and Related Technology (COBIT). Other tools used for ISC are standards from the International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), including ISO/IEC 177995, 27001, and 27002 (International Organization for Standardization, 2022a), alongside PROTECT, the Capability Maturity Model (CMM), and the Information Security Architecture (ISA), as stated by Veiga & Eloff (2007). However, van de Haar (2003) points out that choosing ISC best practices poses challenges because it delegates the identification of appropriate ISC to the users and offers limited assistance in pinpointing the most suitable controls for optimal information security. This approach also tends not to reflect an organization's particular limitations like budget, time constraints, and available resources, according to Barnard (2000).

### Risk Analysis and Management

Previously, identifying Information Security Controls  involved the use of Risk Analysis and Management (RAM), necessitating business analyses and risk evaluations to ascertain information security risks and their countermeasures (Barnard, 2000). Historically, RAM has been beneficial in safeguarding information security (Dhillon & Torkzadeh, 2006). Once the risks and requirements for information security are pinpointed, organizations adopt ISCs best suited to addressing these identified risks.

According to van de Haar (2003) RAM takes a subjective, bottom-up approach that might not always reflect an organization's particular constraints. Further critique points out that an over-reliance on RAM can cause more issues in achieving optimal information security rather than providing advantages (Dhillon & Torkzadeh, 2006). Additionally, through the application of RAM, there is a chance that organizations might implement unnecessary ISCs or address inconsequential problems (Dhillon & Torkzadeh, 2006).

### Information Security Checklists

Over time, the adoption of information security checklists has emerged as an approach to pinpoint aspects of Information Systems Control  (Baskerville, 1993). These checklists are commonly utilized within organizations utilizing cloud computing to discern prevalent ISCs, including potential security risks (Kalaf & ElafAyyedJebur, 2015). Dhillon & Torkzadeh (2006) acknowledge the importance of such checklists in recognizing threats to information systems and generating corresponding ISC measures to mitigate these threats. However, reliance solely on checklists can lead to a defective security strategy for information systems. Additionally, the limited analytical robustness of these checklists curtails their practicality (Baskerville, 1993), and it is further argued that they fail to adequately confront the fundamental challenge of grasping the information security issues faced by organizations (Backhouse & Dhillon, 1996).

© Pavlykevych, A., & Dzioban, M. (2024). Event-driven method of measuring effectiveness of information security controls in software development environments. Сучасний захист інформації, 3(59), 42–54. https://doi.org/10.31673/2409-7292.2024.030004.

### Desirability Functions

An alternate method to pinpoint ISC has involved utilizing desirability functions. These functions were employed in (Otero et al., 2010) to aid organizational management in identifying the most suitable ISC within their limited resources. Essentially, desirability functions measured how desirable ISC options were within resource-constrained organizations. The study referenced in (Otero et al., 2010) conducted an analysis to assess the comprehensive quality of each ISC relative to the organization's objectives. This methodology proved sufficient for gauging the quality of ISC against various criteria specific to the application within organizations. Despite resulting in an evaluation method centered on measuring ISC quality attributes and their relevance to the organizational needs, the binary criteria used to evaluate the attributes of each ISC may not be deemed accurate enough for making decisions regarding ISC selection and implementation in organizations.

The literature just presented evidence limitations in existing ISC assessment methodologies.

## Methods of Research

### Control Effectiveness Measurement

Coming from the definition of an information security control per ISO 27001, which is “measures to prevent, detect, or correct the occurrence of an information security incident”, control effectiveness is derived as a function of alert handling per control system. DevSecOps alert processing loop provides basis for calculating control effectiveness. As each alert gets registered on Security Incident and Event Management (SIEM) system along with its status, age, criticality, related exception records, project identifier, etc., it makes SIEM obvious place to apply control effectiveness measurement.

Deriving information security control effectiveness from alert generation and handling flow makes this method universal and not dependant on actual technology details of specific controls. The idea is that whatever should be considered deviation from compliance or anomalous activity is defined by internal rules on information security control system and on SIEM event analysis level and results in generation of an alert.

Next, the measure of effectiveness of a security control is whether the number of such alerts is kept as low as possible and whether they are processed as quick as possible. Keeping number of open alerts minimal and with as short as possible duration results in minimal area of vulnerability. Lowering possibility for an external threat to exploit system under protection via open vulnerabilities is essentially a purpose of a security control – risk mitigation.

Visual representation of an exposure area is provided on the figure below (Fig. 1). It is a two-dimensional figure with horizontal dimension being time and vertical – number of open alerts. For the purposes of control effectiveness evaluation, we are counting alerts in the time window with the length of reporting period. Doing so results in metrics which are always actual for a reporting date. Calculating statistical measures of alert processing for a prior timeframe offers the benefit of showing whether the quality of security operations within a project context was sustained consistently over time, as opposed to scenarios where alerts might be managed only before the generation of performance reports.

The *Fig. 1* depicts a flow of security alerts generated from an ISC. The colour of rectangles which represent alerts indicates their level of criticality. In this case, there are alerts with two criticality levels, ‘Critical’ in red colour and ‘High’ in yellow. Rectangle length represents alert processing duration from origination to closure (or suppression by application of an ‘Exception’ record). Alerts with different criticality level are counted towards effectiveness metric calculation in the same way. The difference is that alerts of higher criticality level have shorter tolerance period as represented by white area at the beginning of each rectangle. Alerts addressed within tolerance period do not influence effectiveness metrics calculation. This period must be defined in each control Service Level Agreement (SLA) individually. Therefore, the logic behind this thinking is that even moderate flaws in system security may lead to same harm as presence of critical flaws if enough time is given for an intruder to identify and leverage them in attack kill chain. Note that there can’t be toleration

© Pavlykevych, A., & Dzioban, M. (2024). Event-driven method of measuring effectiveness of information security controls in software development environments. Сучасний захист інформації, 3(59), 42–54. https://doi.org/10.31673/2409-7292.2024.030004.

period set for alerts signalling anomalous actions and attack activity. Such alerts must be presented to Cyber Security Operation Center (CSOC) immediately.

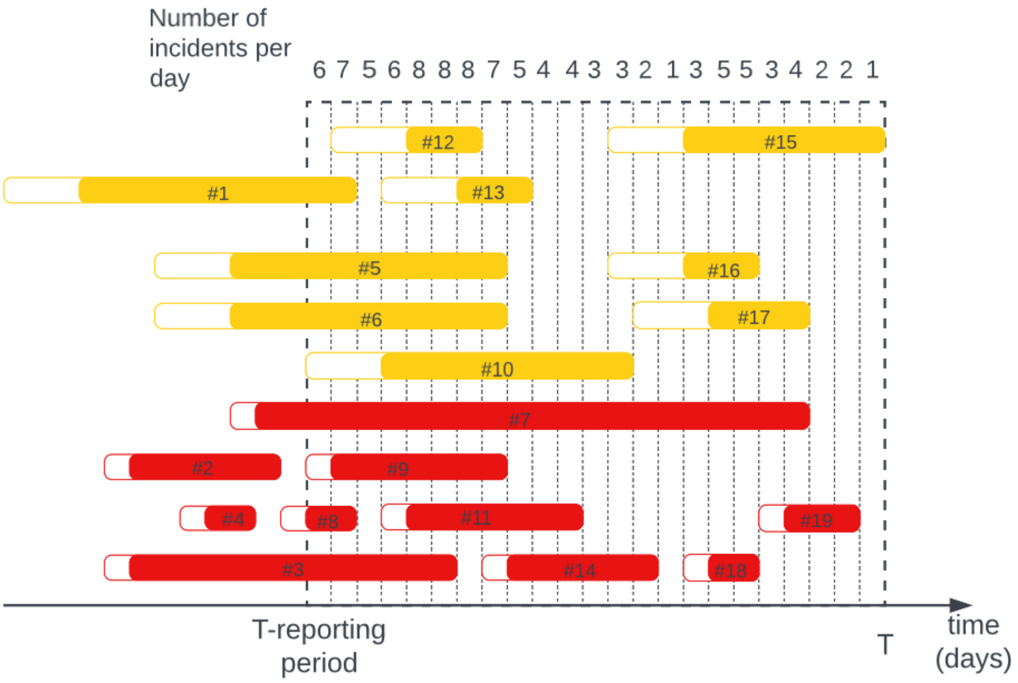


Fig. 1. Event flow example representation

Continuing the given example, we derive two metrics which in combination will indicate effectiveness of the security control and its associated operational process:

- percentile level of active incidents per day, “Cleanliness”;
- percentile level of incidents duration, “Timeliness”.

Cleanliness metric is expressed by calculating the level of set percentile (in this example 80%) from the number of active incidents which exist per day in the reporting window. In the example below, we use a formula to display what was the number or incidents active in 80% of days in the reporting window. The result of 6 is that compared to a fixed grading defined for the given control. In this case, if in 80% of days there were up to 6 open alerts, the state of ‘cleanliness’ in considered ‘Fair’ (Fig. 2).

Similar logic is applied to qualify how quickly environment owners resolve alerts, the metric of ’Timeliness’. Here, we compare quality gradation with the number of days under which 80% of alerts were open in the reporting window. In this example, it is 11, which is graded as ‘Unsatisfactory’ (Fig. 3).

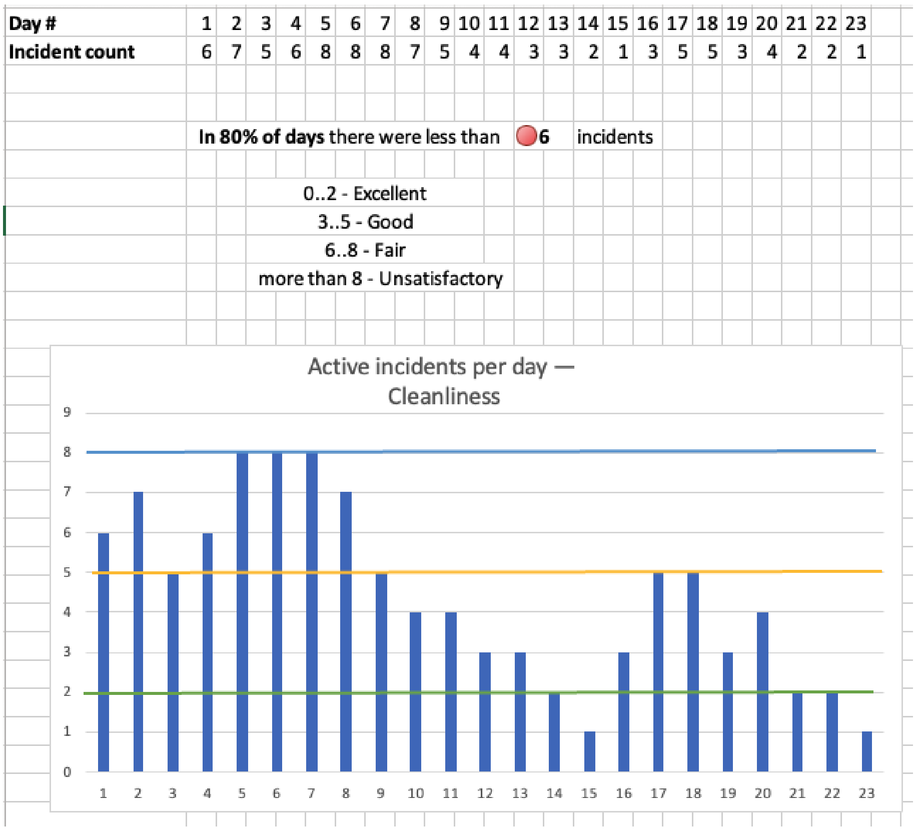
We then use fixed matrix to combine two dimensions of qualitative effectiveness metrics into a single indicator. We also assign fractional numeric value for each of defined effectivess levels for further consolidation and scoring of entire control set. See Fig. 4. Control effectiveness matrix нижe.

Individual ISC effectiveness is then used in Risk Assessment to evaluate overall effectiveness or level of security of software development environment.

**Risk Assessment**

Software Development Environment (SDE) is an environment that augments or automates the activities comprising the software development cycle, including programming-in-the-large tasks such as configuration management and programming-in-the-many tasks such as project and team management. It also defines an environment that supports largescale, long-term maintenance of software (Dart et al., n.d.).

© Pavlykevych, A., & Dzioban, M. (2024). Event-driven method of measuring effectiveness of information security controls in software development environments. Сучасний захист інформації, 3(59), 42–54. https://doi.org/10.31673/2409-7292.2024.030004.

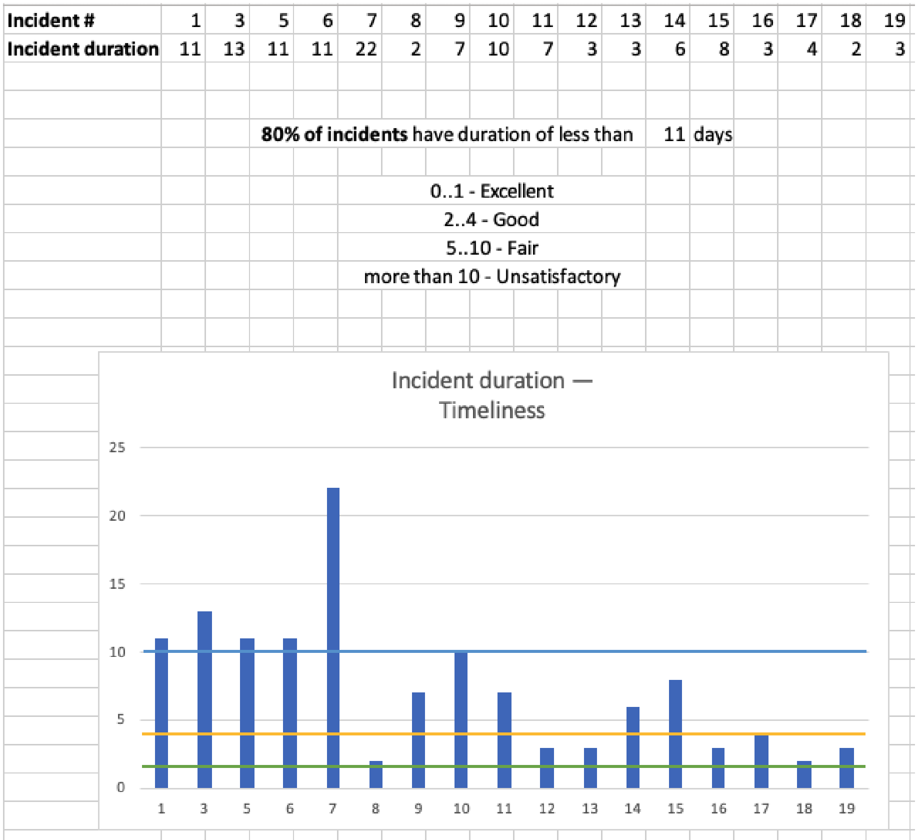


| Day # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Incident count | 6 | 7 | 5 | 6 | 8 | 8 | 8 | 7 | 5 | 4 | 4 | 3 | 3 | 2 | 1 | 3 | 5 | 5 | 3 | 4 | 2 | 2 | 1 |

**In 80% of days** there were less than ● **6** incidents

0..2 - Excellent
3..5 - Good
6..8 - Fair
more than 8 - Unsatisfactory

Fig. 2. 'Cleanliness' metric calculation example



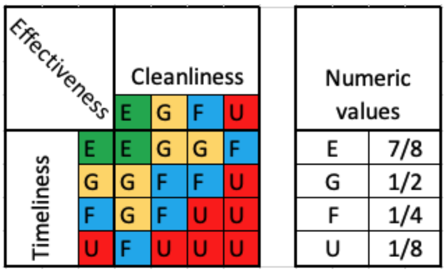
| Incident # | 1 | 3 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Incident duration | 11 | 13 | 11 | 11 | 22 | 2 | 7 | 10 | 7 | 3 | 3 | 6 | 8 | 3 | 4 | 2 | 3 |

**80% of incidents** have duration of less than 11 days

0..1 - Excellent
2..4 - Good
5..10 - Fair
more than 10 - Unsatisfactory

Fig. 3. 'Timeliness' metric calculation example

© Pavlykevych, A., & Dzioban, M. (2024). Event-driven method of measuring effectiveness of information security controls in software development environments. Сучасний захист інформації, 3(59), 42–54. https://doi.org/10.31673/2409-7292.2024.030004.



| Effectiveness | Cleanliness E | Cleanliness G | Cleanliness F | Cleanliness U |
|---|---|---|---|---|
| Timeliness E | E | G | G | F |
| Timeliness G | G | F | F | U |
| Timeliness F | G | F | U | U |
| Timeliness U | F | U | U | U |

| Numeric values | |
|---|---|
| E | 7/8 |
| G | 1/2 |
| F | 1/4 |
| U | 1/8 |

Fig. 4. Control effectiveness matrix

SDE Risk Assessment (RA) process is meant to supply information which is suitable for calculating overall risk levels to SDE primary informational asset (According to ISO27000 primary assets are information or business processes, and supporting assets are related IT systems, infrastructure and people resources) – “External customer-related information”. It does so by applying compact benchmarking method. It is implemented as “Assessment” phase of the SDE Risk Assessment process.

The main idea of the assessment interview is to compare state of a given project environment against the set of “Indicator” statements. These statements are worded as practical cases which describe general best practices for environment security configurations, processes, and composition. They are compiled from several sources of knowledge: industry standards and best practices (such as (Souppaya et al., 2022), (Joint Task Force, 2020), OWASP SAMMv2 (*SAMM*, 2020)), analysis of previous information security incidents, and regularly maintained to reflect current technologies, threats, and vulnerabilities.

SDE RA process envisions a benchmark maintenance process as a part of continuous improvement of the program. The task of such process is to regularly review indicator statements, withdraw or add items in response to changes in technologies, actual threat and vulnerability landscape, changes in the company business offerings and customers’ expectations, and the sources of information listed above.

From the risk management perspective, each indicator statement in the benchmark is a scenario describing a combination of some threat and vulnerability conditions. It is called “risk likelihood” in the ISO 27001 risk management framework.

For the purposes of more precise risk mapping and measurement within SDE RA processes, every such combination is assigned a matrix of relative risk likelihood level per information type. This is done better specify principal information security risks versus information type acting in the case described by the indicator statement.

Proposed RA method introduces four categories of data within SDE (Tab. 1):

**Application Data:** Any form of digital processed and stored by the application which is being developed within the project which is being evaluated. It includes and data which is directly accessible by the application from directly integrated services, like data bases or object storage services, and data which is being used for application testing.

**Secrets:** Any kind of information which is used to gain access to the application under development; hosts and services within the application environment; code management and integration tools; tokens, secrets, digital keys for application components integration; accounts for application testing, accounts for accessing applications and services provisioned by customer.

**Code:** All parts of the project code base, including application source code, auxiliary code for application testing and integration, configuration files, automation scripts.

**Documents:** All kinds of digital data stored or exchanged for the purposes of executing project under assessment. This includes digital documents, articles, email and instant messages, audio and video recordings, visual content.

© Pavlykevych, A., & Dzioban, M. (2024). Event-driven method of measuring effectiveness of information security controls in software development environments. Сучасний захист інформації, 3(59), 42–54. https://doi.org/10.31673/2409-7292.2024.030004.

Proposed method also introduces four principal types of risks relevant to data types listed above are following:

**Exposure**: Loss of confidentiality of the information.

**Loss**: Risk of irreversible destruction or deletion.

**Unavailability**: Risk of the data not being accessible when needed or with needed performance, mostly caused by damage to the environment it is stored or processed in.

**Loss of integrity**: Unauthorized alteration or the data, tampering.

See example of two indicators definitions below.

Table 1.

SDE RA Benchmark indicators descriptions example

| Indicator | Answer | Data Category | Exposed | Lost | Unavailable | Integrity Compromised | Potential Risk | Residual Risk | Control | Threats | Vulnerabilities |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Regular backup copies of applications systems and data are performed | Yes | App data | 0 | 4 | 4 | 4 | 21 | 21 | Backup service for project environment | Threat:<br>- Cyberattack<br>- Ransomware | Vulnerability:<br>- Backups are not being made |
| | | Secrets | 0 | 3 | 3 | 3 | | | | | |
| | | Code | 0 | 0 | 0 | 0 | | | | | |
| | | Docs | 0 | 0 | 0 | 0 | | | | | |
| Backup copies of applications systems and data saved in encrypted form on a dedicated storate with explicitly defined access policies | Yes | App data | 4 | 0 | 4 | 4 | 19 | 19 | Backup service for project environment | Threat:<br>- administrators negligence<br>- Ransomware | Vulnerability:<br>- Backups are stored insecurely |
| | | Secrets | 3 | 0 | 2 | 2 | | | | | |
| | | Code | 0 | 0 | 0 | 0 | | | | | |
| | | Docs | 0 | 0 | 0 | 0 | | | | | |

The numbers in the data type vs. risk matrix indicate relative impact level of this particular risk per data type. Note that Application Data is assigned highest value of "4", followed by "3" for Secrets, "2" for Code, and "1" for Documents. Including level of impact, gives us have complete risk description with level of risk determined.

© Pavlykevych, A., & Dzioban, M. (2024). Event-driven method of measuring effectiveness of information security controls in software development environments. Сучасний захист інформації, 3(59), 42–54. https://doi.org/10.31673/2409-7292.2024.030004.

Next attribute of the indicator description is “control”. This is a reference to an information security control most relevant to mitigate associated risk. Controls proposed to the indicators are selected from the list of SDE IS controls.

During the interview, each indicator statement should be put into correspondence with the following attributes:

“Answer”, may be set as:

- “Yes”, in case when the indicator statement regarding the project environment is true.
- “No”, in case if the configuration or practices expressed in the statement are missing in the project environment or not completely met.
- “Not applicable” if the project environment does not include systems relevant to the indicator.
- “Do not know” is set in cases when systems or practices relevant to the indicator do exist in the project environment but there is no knowledge if they satisfy the indicator statement. Actual information should be provided upon some consultation with relevant environment administrators.

“Implemented by”

- “Company”, when statements described in the indicator item are implemented by means of Company-controlled systems (ISCs).
- “Customer”, when statements described in the indicator item are implemented by means of Customer-controlled systems (ISCs).

Result of the benchmark form completion is the table which lists risks relevant to the project environment. Each indicator item with the answer or “No” is considered open risk record which must be associated with appropriate risk treatment strategy. This data is then taken to the “Risk Management” stage of the SDE RA process where associates in appropriate project roles assign risk treatment strategies. On this stage, it may be decided to introduce controls to the project environment if such are lacking and a risk mitigation strategy of “Mitigate” was selected.

Data from actual benchmark responses together with selected risk treatment strategies provide basis for profiling and comparing state of data protection on projects on the account and organization level which is performed on the next SDE RA process stage.

SDE RA process provides starting point for project environment risk profile. It can be presented on reports and is useful for making several kinds of business decision. But without continuous monitoring and reporting on the actual effectiveness of controls employed for risk mitigation, it would be a static snapshot.

Proposed method for accounting control effectiveness into resulting project-level information security risks is based on the model of “principal risk vectors” – logical paths from each data type (Application Data, Secrets, Source Code, Documents) to each principal risk (Exposure, Loss, Unavailability, Loss of integrity). Each such vector is composed by sum of individual project environment risks as identified by ZRT benchmark indicators and their respective threat and vulnerability combinations. For a given indicator, non-zero values in the risk detail matrix on intersection of data type and risk type show which principal risk vector this indicator (environment-level risk) contributes to.

The diagram below describes this approach on the example of one of principal risk vector from “Secrets” data type to the risk of “Exposure” (Fig. 5).

“Risk Likelihood” items are all contributing to the total likelihood level. For simplicity, it may be stated that the total likelihood equals to 1, however, model may be extended to sum-up weighted individual risk likelihood levels.

The next block represents influence of controls which are reducing probability of the given type of risk realization against the given data type. The more effective security controls are, the more inhibited would be the chance for security breach to happen by this risk vector. Therefore, total level of risk reduction on this vector is expressed as an average of effectiveness level of controls acting on each contributing risk likelihood.

© Pavlykevych, A., & Dzioban, M. (2024). Event-driven method of measuring effectiveness of information security controls in software development environments. Сучасний захист інформації, 3(59), 42–54. https://doi.org/10.31673/2409-7292.2024.030004.

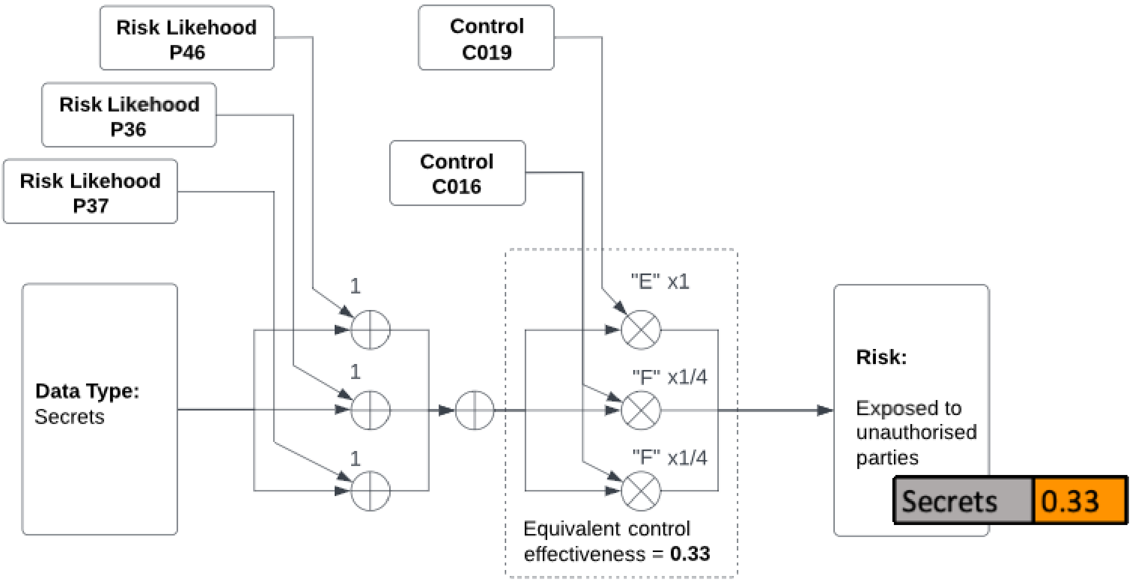


Fig. 5. Principal risk vector representation example

With control effectiveness metric represented as fractional number, it is convenient to use harmonic mean for averaging total control effectiveness on a principal risk vector. For a principal risk vector which is composed of ‘n’ risk likelihood components, the mean control effectiveness would be ‘n’ multiplied by inverse sum of reciprocals of control effectiveness which are applied to each risk likelihood component. See formula below.

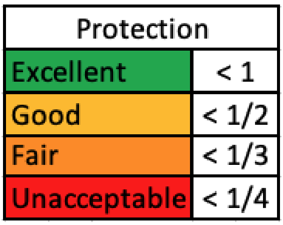
$$E_{tot} = \frac{n}{\frac{1}{E_1} + \frac{1}{E_2} + \cdots + \frac{1}{E_n}}.$$

For controls which must be applied per each resource in an environment, control effectiveness metric should be corrected by adoption factor, which is calculated as following:

$$E_{xa} = E_x \frac{R_a}{R_{tot}},$$

where $R_a$ equals to number of resources where the control in correctly applied, and $R_{tot}$ is total number of resources.

Mean effectiveness metric will always be a fractional number between 0 and 1, where higher values would indicate best possible protection against relevant risks and lower values would indicate absence of protection. With this method of calculation of mean effectiveness, we get metric which is sensitive to low values, i.e. presence of ineffective and incomplete controls would quickly influence total metric. For presentation purposes, it may be displayed as the following qualitative grade scale (Fig. 6).

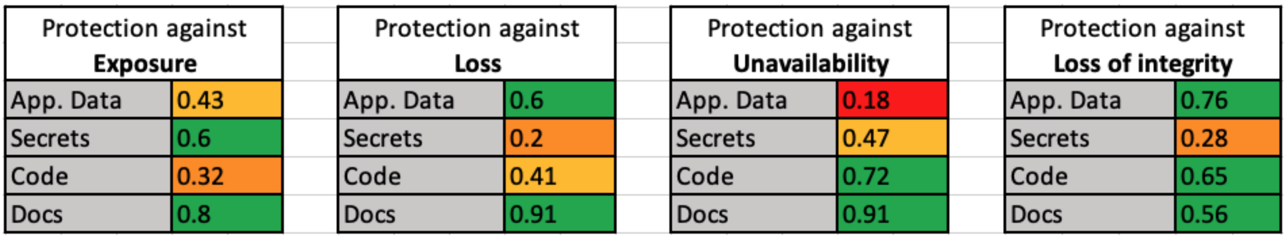


| Protection | |
|---|---|
| Excellent | < 1 |
| Good | < 1/2 |
| Fair | < 1/3 |
| Unacceptable | < 1/4 |

Fig. 5. Risk protection grades

© Pavlykevych, A., & Dzioban, M. (2024). Event-driven method of measuring effectiveness of information security controls in software development environments. Сучасний захист інформації, 3(59), 42–54. https://doi.org/10.31673/2409-7292.2024.030004.

Knowing resulting effectiveness of controls which reduce principal risk for a data type, permits presenting compact and informative report on the overall state of information protection. For each principal risk, we can display a table of resulting control effectiveness per data type (Fig. 7).



| Protection against **Exposure** | |
|---|---|
| App. Data | 0.43 |
| Secrets | 0.6 |
| Code | 0.32 |
| Docs | 0.8 |

| Protection against **Loss** | |
|---|---|
| App. Data | 0.6 |
| Secrets | 0.2 |
| Code | 0.41 |
| Docs | 0.91 |

| Protection against **Unavailability** | |
|---|---|
| App. Data | 0.18 |
| Secrets | 0.47 |
| Code | 0.72 |
| Docs | 0.91 |

| Protection against **Loss of integrity** | |
|---|---|
| App. Data | 0.76 |
| Secrets | 0.28 |
| Code | 0.65 |
| Docs | 0.56 |

Fig. 6. Risk protection grades by principal risk vectors example

Same method for measuring mean control effectiveness is applied on the risk vector which is defined for principal risk of “Contract compliance violation”, taking into calculation all controls involved for contract information security requirements implementation.

The Fig. 7 provides extended view on the way the data from ERA assessment is interpreted as principal risk vectors. Different risk likelihood items (indicators) may contribute in different way to different principal risk vectors as it is defined per each indicator characteristics and described by respective data type vs. risk matrix. There will be as many vertical columns of logical connection points as there are risk likelihood items defined. It is also illustrated that same controls may regulate different risk likelihoods.

The diagram accounts for risk treatment options selection per indicator. Only those indicator items for which mitigation strategy was selected may contribute to risk calculation as described above. Otherwise, there is no data on control effectiveness for risk which are transferred, in which case controls are implemented by a customer. The same consideration applies to risks selected to be accepted of avoided.

Statistics on the state of risk treatment options for indicators are displayed as one of the non-functional reports. Complete list of reports produced from SDE RA process data and functioning of IS controls implemented for the project environment is following:

Controls fidelity report, indicating which part of controls are of delivered by Company (and, therefore, produce constant monitoring and measurement) and which part are said to be implemented on customer’s side.

Control functional characteristics reports:

- Control effectiveness per control per project with summarization to account and organization.
- Control adoption.
- Control environment metric.
- Risk mitigation options selected per indicator with aggregation per information/risk type.
- Contract commitment support by controls (with controls contributing effectiveness averaged).

Control non-functional characteristics reports:

- Cost of control with breakdown per control/project with summarization
- Control implementation status (lead time).
- Control (product) NPS.

Process characteristics reports:

- SDE RA process progress (performance of benchmark response and risk treatment options assignment per project/account/org.).
- Process NPS.

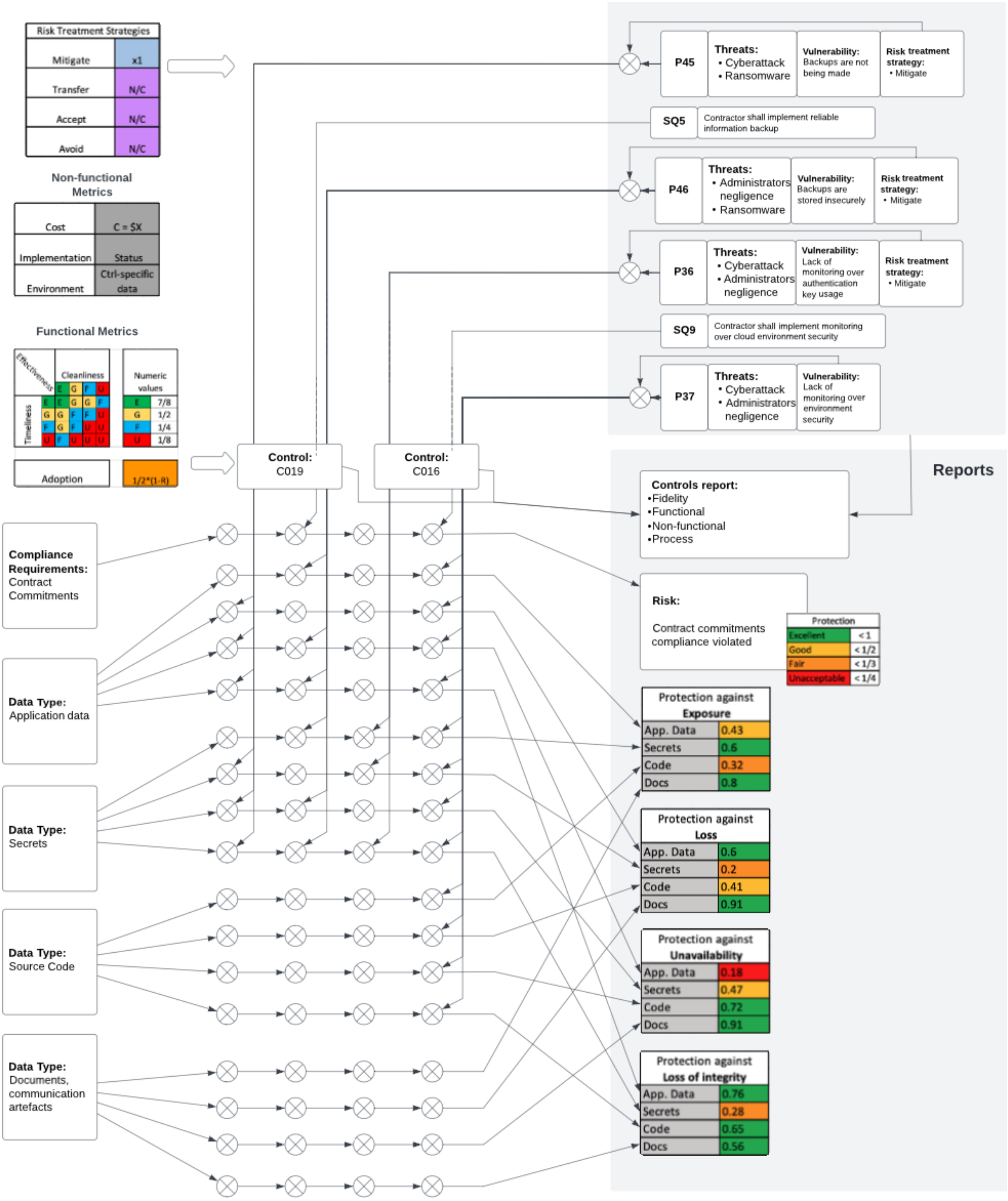

Risk Treatment Strategies
Mitigate x1
Transfer N/C
Accept N/C
Avoid N/C
Non-functional Metrics
Cost C = $X
Implementation Status
Environment Ctrl-specific data
Functional Metrics
Effectiveness
Cleanliness
Timeliness
Numeric values
E 7/8
G 1/2
F 1/4
U 1/8
Adoption 1/2*(1-R)
Control: C019
Control: C016
Compliance Requirements: Contract Commitments
Data Type: Application data
Data Type: Secrets
Data Type: Source Code
Data Type: Documents, communication artefacts
P45
Threats: • Cyberattack • Ransomware
Vulnerability: Backups are not being made
Risk treatment strategy: • Mitigate
SQ5
Contractor shall implement reliable information backup
P46
Threats: • Administrators negligence • Ransomware
Vulnerability: Backups are stored insecurely
Risk treatment strategy: • Mitigate
P36
Threats: • Cyberattack • Administrators negligence
Vulnerability: Lack of monitoring over authentication key usage
Risk treatment strategy: • Mitigate
SQ9
Contractor shall implement monitoring over cloud environment security
P37
Threats: • Cyberattack • Administrators negligence
Vulnerability: Lack of monitoring over environment security
Reports
Controls report: •Fidelity •Functional •Non-functional •Process
Risk: Contract commitments compliance violated
Protection
Excellent <1
Good <1/2
Fair <1/3
Unacceptable <1/4
Protection against Exposure
App. Data 0.43
Secrets 0.6
Code 0.32
Docs 0.8
Protection against Loss
App. Data 0.6
Secrets 0.2
Code 0.41
Docs 0.91
Protection against Unavailability
App. Data 0.18
Secrets 0.47
Code 0.72
Docs 0.91
Protection against Loss of integrity
App. Data 0.76
Secrets 0.28
Code 0.65
Docs 0.56

Fig. 7. SDE risk calculation method description

Proposed SDE RA methodology was implemented using Microsoft Power BI platform for analytics, with principal data supplied from SIEM (metrics on control effectiveness and adoption), ITSM (data on control implementation statis and assets per project), application used to conduct SDE RA assessments and risk options assignment.

© Pavlykevych, A., & Dzioban, M. (2024). Event-driven method of measuring effectiveness of information security controls in software development environments. Сучасний захист інформації, 3(59), 42–54. https://doi.org/10.31673/2409-7292.2024.030004.

The following diagram describes how metrics produced by SDE RA processes aggregate into meaningful risk records characterizing individual projects on the organization level. Together with risks reported on other aspects of projects, information security risks provide valuable information for business leaders to make decisions on investments prioritization and customer relations strategy (Fig. 9).

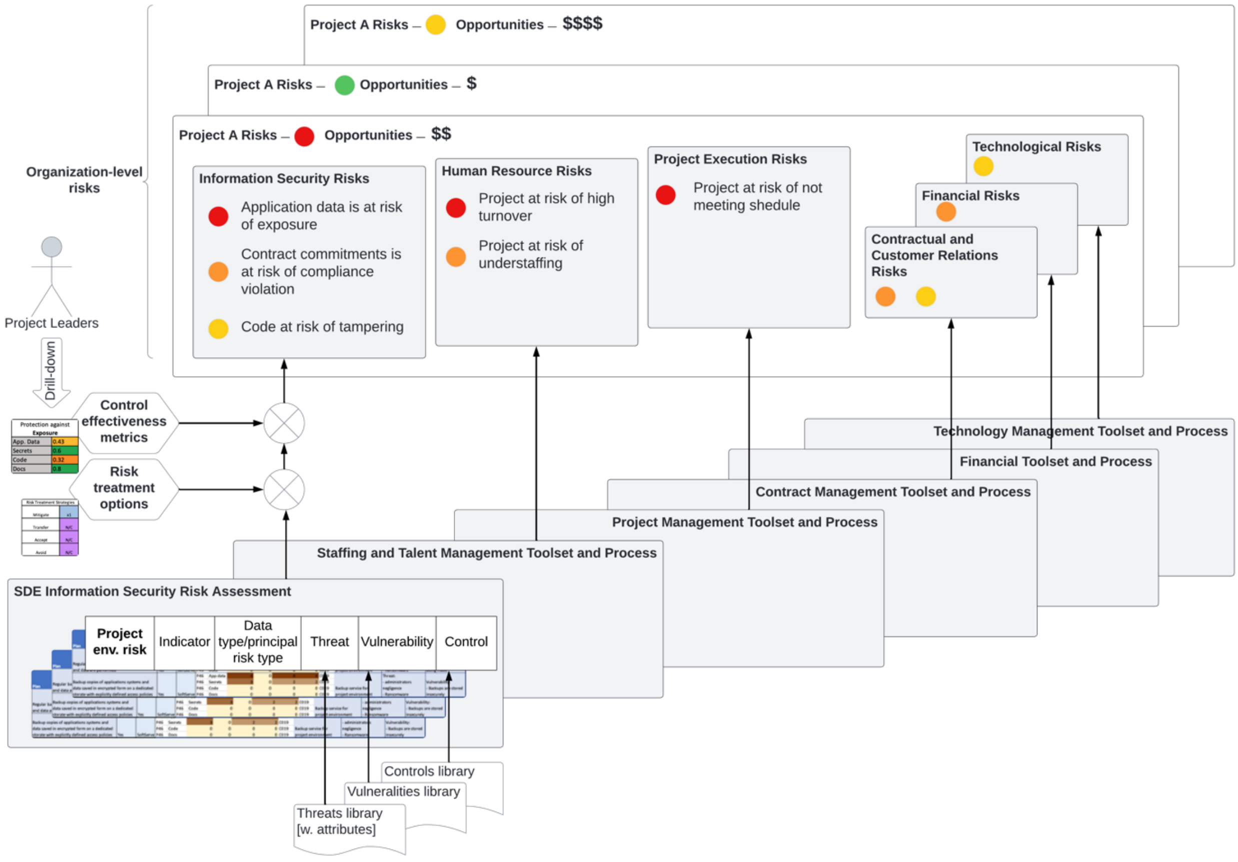

Project A Risks – ● Opportunities – $$$$
Project A Risks – ● Opportunities – $
Project A Risks – ● Opportunities – $$
Organization-level risks
Information Security Risks
Application data is at risk of exposure
Contract commitments is at risk of compliance violation
Code at risk of tampering
Human Resource Risks
Project at risk of high turnover
Project at risk of understaffing
Project Execution Risks
Project at risk of not meeting shedule
Technological Risks
Financial Risks
Contractual and Customer Relations Risks
Project Leaders
Drill-down
Control effectiveness metrics
Risk treatment options
Technology Management Toolset and Process
Financial Toolset and Process
Contract Management Toolset and Process
Project Management Toolset and Process
Staffing and Talent Management Toolset and Process
SDE Information Security Risk Assessment
Project env. risk
Indicator
Data type/principal risk type
Threat
Vulnerability
Control
Controls library
Vulnerabilities library
Threats library [w. attributes]

Fig. 9. SDE risk aggregation and reporting

## Conclusions

This paper describes practical approach to measuring ISCs effectiveness within software development environments (SDEs) using method independent from control type and design and based on security events: an externally measured effects of control malfunctions. This together with suggested approach to calculating aggregated score for SDEs protection against categories of threats allows to develop actionable risk assessment (RA) framework for SDEs. Paper provides example of building such information security RA into overall SDE risk management framework. Proposed SDE RA methodology was implemented using Microsoft Power BI platform for analytics, with principal data supplied from SIEM (metrics on control effectiveness and adoption), ITSM (data on control implementation statis and assets per project), application used to conduct SDE RA assessments and risk options assignment.

This paper demonstrates use of unified effectiveness measurement method for SDE ISCs based on security event and incident management (SIEM) as well as method of consolidating these measurements into high level metrics used to evaluate overall security of specific SDE or entire group of SDE owned by the organization. Method offers new approach for collecting meaningful benchmarking data from stakeholders without formal Information Security education and providing results, which can be directly used by non-IS professionals to drive management decisions within the organization.

© Pavlykevych, A., & Dzioban, M. (2024). Event-driven method of measuring effectiveness of information security controls in software development environments. Сучасний захист інформації, 3(59), 42–54. https://doi.org/10.31673/2409-7292.2024.030004.

## References

1. Backhouse, J., & Dhillon, G. (1996). Structures of responsibility and security of information systems. European Journal of Information Systems, 5(1), 2–9. https://doi.org/10.1057/ejis.1996.7
2. Barnard, L. (2000). A Formalized Approach to the Effective Selection and Evaluation of Information Security Controls. Computers & Security, 19, 185–194. https://doi.org/10.1016/S0167-4048(00)87829-3
3. Baskerville, R. (1993). “Information Systems Security Design Methods: Implications for Information Systems Development.” ACM Comput. Surv., 25, 375–414. https://doi.org/10.1145/162124.162127
4. Dhillon, G., & Torkzadeh, G. (2006). Value-focused assessment of information security in organizations. Information Systems Journal, 16, 293–314. https://doi.org/10.1111/j.1365-2575.2006.00219.x
5. Digital Press Briefing with Anne Neuberger, Deputy National Security Advisor for Cyber and Emerging Technologies. (2023, October 18). United States Department of State. https://www.state.gov/digital-press-briefing-with-anne-neuberger-deputy-national-security-advisor-for-cyber-and-emerging-technologies/
6. Global Risks Report. (2023, January 11). World Economic Forum. https://www.weforum.org/publications/global-risks-report-2023/
7. International Organization for Standardization. (2022a). Information security, cybersecurity and privacy protection—Information security controls (27002; Version 3). https://www.iso.org/obp/ui/en/#iso:std:iso-iec:27002:ed-3:v2:en
8. International Organization for Standardization. (2022b). Information security, cybersecurity and privacy protection—Information security management systems—Requirements (27001; Version 3). https://www.iso.org/standard/27001
9. Joint Task Force. (2020). Security and Privacy Controls for Information Systems and Organizations (SP 800-53; Version 5). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-53r5
10. Kalaf, O. & ElafAyyedJebur. (2015). IT Auditing to Assure a Secure Cloud Computing for Enterprise Applications. International Journal of Engineering Research and General Science, 3, 4.
11. Otero, A., Otero, C., & Abrar, Q. (2010). A Multi-Criteria Evaluation of Information Security Controls Using Boolean Features. International Journal of Network Security & Its Applications, 2. https://doi.org/10.5121/ijnsa.2010.2401
12. Software Assurance Maturity Model (SAMM) (2.0). (2020). https://owaspsamm.org/model/
13. Souppaya, M., Scarfone, K., & Dodson, D. (2022). Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities (800–218; 1.1). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-218
14. van de Haar, H. (2003). A model for deriving information security control attribute profiles. Computers & Security, 22, 233–244. https://doi.org/10.1016/S0167-4048(03)00311-0
15. Veiga, A., & Eloff, J. (2007). An Information Security Governance Framework. IS Management, 24, 361–372. https://doi.org/10.1145/1655168.1655170
16. Vulnerability and Threat Trends Report. (2023). Skybox Security. https://www.skyboxsecurity.com/resources/report/vulnerability-threat-trends-report-2023/

Надійшла 12.07.2024

© Pavlykevych, A., & Dzioban, M. (2024). Event-driven method of measuring effectiveness of information security controls in software development environments. Сучасний захист інформації, 3(59), 42–54. https://doi.org/10.31673/2409-7292.2024.030004.