Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

TC 11 Briefing Papers

A Model For risk-Based adaptive security controls

Miguel Calvo, Marta Beltrán[*]

Department of Computing, ETSII, Universidad Rey Juan Carlos, Mostoles 28933, Madrid, Spain

ARTICLE INFO

Article history:
Received 10 January 2021
Revised 20 October 2021
Accepted 12 January 2022
Available online 15 January 2022

Keywords:
Adaptive security
Context-aware decision making
Dynamic security controls
MAPE Loop
Risk-based security

ABSTRACT

Security controls and countermeasures have shifted from static desktop-based and corporate network environments to heterogeneous, distributed and dynamic environments (e.g., cloud and mobile computing or Internet of Things). Due to this paradigm shift, adaptive and risk-based approaches have gained significant importance. These approaches allow security managers to perform context-aware decision making, adapting controls' deployment, configuration or use to every specific situation, depending on the current value of risk indicators or scores and on the level of risk tolerated by the organisation at any given time. This paper proposes a model to automatically adapt security controls to different risk scenarios in almost real-time (if required). This model is based on a three-layer architecture and a three-step flow (measurement-decision-adaptation), relying on a scalable policies&rules framework capable of integrating with different kinds of controls. Furthermore, the proposed model is validated and evaluated with an actual use case.

1. Introduction

Dynamic security allows security managers to select the best control for achieving required security levels lively. These levels may depend on context evolution, asset changes and, ultimately, on the observed risk and the risk that the organisation or users are prepared to tolerate in a given situation Poolsappasit et al. (2011).

There are no one-control-fits-all-scenarios in current security, no static configuration, deployment, use or management of a security control is able to cope with risk within all possible situations. Some configurations will work better under certain environmental conditions, others will be more secure when facing specific threat actors, and some of them will be only used when employees use specific devices or services. For example, an operator of a critical infrastructure will be required to use a second authentication factor on a service configured to use only one if the level of terrorist alert has exceeded a certain threshold on that day. An employee will not be able to download a document from a corporate server if he is working on his personal mobile phone instead of his professional laptop. Alternatively, the whitelist of allowed domains may change depending on the security team's level of control over the network from which an employee connects to the Internet.

An intelligent approach to deal with this heterogeneity, uncertainty and dynamism is an adaptive and risk-based one, capable of adjusting applied security controls to the asset's context and state and the risk these factors imply at a given moment Sion et al. (2018), Lara et al. (2019). The objective is to propose a model that autonomously changes the behaviour of security controls by monitoring the assets and their context, quantifying the risk and safeguarding assets to the desired levels (mitigating the quantified risk to its tolerated value).

The main contributions of this work are:

1. The definition of a new model capable of adapting security controls to different risk scenarios, automatically and in almost real-time: RiAS (**Ri**sk-based **A**daptive **S**ecurity).
2. The proposal of a three-layer architecture and a three-step flow (measurement-decision-adaptation) supporting this model.
3. The definition of the offline steps required to make this adaptation flow work, enabling reactive or proactive parametric, architectural and behavioural adaptations with different timing strategies and deployment approaches for the three proposed layers.
4. The specification of an easy-to-use, flexible, scalable and generic policies&rules-based engine capable of integrating with different kinds of controls.
5. A first prototype of the proposed architecture used in a real scenario, validating its functionalities and assessing its levels of

[*] Corresponding author.
E-mail addresses: miguel.calvo@urjc.es (M. Calvo), marta.beltran@urjc.es (M. Beltrán).

https://doi.org/10.1016/j.cose.2022.102612

performance and security when adapting a Web Application Filter (WAF).

The rest of this paper is organised as follows. Section 2 provides an overview of the related work and previous researches in the field. Section 3 discusses the primary motivations for this work, providing motivating examples, and research questions that need to be answered. Section 4 describes the proposed architecture and considered assumptions, presenting the new adaptation model. Section 5 details its validation and evaluation with an actual use case. Finally, Section 6 summarises our main conclusions and the most interesting lines for future research.

2. Related work

The emergence and progress of paradigms such as cloud, mobile, fog, edge Computing and the Internet of Things, the evolution of software development models and network infrastructures (increasingly software-defined) and new application domains and their requirements in terms of quality of service and experience (Industry 4.0, Smart Cities, 5G, robotics, eHealth) have forced researchers to consider a new evolution in which security capabilities become dynamic. This dynamism implies that security capabilities can be reconfigured and adapted to the context, according to the actual threats and the tolerated risk. Different researches have addressed dynamic security in the last years, proposing controls that are applied or not depending on the situation (dynamic security policies proposed in Varadharajan et al. (2019)) or configuration changes that are applied based on context (such as Graur (2017) or Bursell, 2019).

Despite these good examples, dynamic security is still an ambiguous and undefined concept. Adaptive security and risk-based security are closely related concepts that often appear in research exploring dynamic security.

2.1. On adaptive security

Adaptive systems are able to modify their composition, architecture or behaviour in response to their state and perception of the operating context. They require an adaptation logic; the well-known MAPE-K cycle Arcaini et al. (2015) proposes the following structure for this logic: a component to monitor the operating context and the managed resources (M), a component to analyse gathered data looking for significant changes (A), a component to plan adaptation (P), a component to control its execution (E), and the last component to work as a knowledge repository (K).

Previous works in the adaptive security field have shown how to apply this kind of cycle with two different approaches:

- **Context-aware security**: This approach relies on context-awareness such as geolocation, time, the reputation of a specific IP address or domain, type of device used, etc. to make security decisions. All this information, dynamically gathered and processed, can offer greater security than usual static techniques in different areas of application.
  This concept appears, in most cases, in scenarios of authentication and authorization in distributed systems, the resolution of access control can be based on different procedures or attributes depending on the context of a particular request Veloudis et al. (2016), Dasgupta et al. (2016), Ashibani et al. (2017), Kumar et al. (2018), Psarra et al. (2020).
  These context-aware mechanisms have also been applied in cryptography Fazeen et al. (2014), anomaly detection Cuadra and Aracil (2017), Yasaei et al. (2020) or security assurance Jahan et al. (2020).
- **Incremental or Intelligent security**: This approach usually combines different techniques and tools such as Big Data, Analytics or Security Information and Event Management (SIEM) to detect anomalies, outliers or deviations from standard behaviours to act accordingly. Incremental/Intelligent Security is based on collecting, standardising, and analysing data generated by networks, applications, databases, logs, and other infrastructure in real-time. This information is evaluated and processed (through machine learning, pattern recognition, etc.) to translate the data into a human-readable format that support informed decision making Mohsin et al. (2016), Liu et al. (2017), He et al. (2017), Fernández Maimó et al. (2018), Parampottupadam and Moldovann (2018), Fan et al. (2019), Sasubilli et al. (2020).

Table 1 compares previous works on Adaptive Security which, due to their similarity to RiAS, have been considered relevant for this work. The table includes those papers that aim at a dynamic non-manual adaptation of security controls considering the context or intelligent decision making, excluding very interesting works but are devoted to detection, analysis, alerting and, eventually, decision-support for human operators.

2.2. On risk-based security

On the other hand, dynamic security also includes risk-based security. This paradigm manages to identify the different risks of each of the organisation's assets, prioritising the cost of mitigation, quality of experience and usability, functionality, etc., which means reducing these risks to an acceptable level. This approach allows organisations to deal with typical security trade-offs.

It is necessary to monitor, quantify and evaluate the risks continually to obtain adequate risk-based security. It is also required to decide how risks will be treated when they appear. This treatment, to lead to efficient and effective results, should be dynamically decided and deployed.

Most of the application scenarios of this paradigm focus again on access control, good examples can be found in Chen et al. (2016), Díaz-López et al. (2016), Steinegger et al. (2016), Metoui et al. (2017) or Sepczuk and Kotulski (2018). But there are other examples in different application domains such as Industrial Control Systems Qin et al. (2018), Software Defined Networks Tripathy et al. (2018) or Data Processing Petersen and Vankempen (2019). Table 2 summarises the main features of previous works on risk-based security relevant to this research.

3. Motivating example and research challenges

Previous works have not proposed a generic model or methodology to enable adaptive or risk-based security. The aforementioned MAPE-K cycle is one of the few general models that can serve as a basis for proposing adaptive, not necessarily risk-based, solutions.

Researches introduced in the previous section focus on designing and developing specific adaptive or risk-based approaches; almost all of them are devoted to access control within different application domains, as illustrated. This lack of a theoretical model or methodology makes it difficult to merge proposed solutions to obtain adaptive and risk-based approaches, reuse them with different security controls, or extend their utilisation.

For example, traditional firewalls have been demonstrated as a powerful solution to protect corporate networks against unwanted traffic or intrusions, potentially harmful code, etc. Firewall rules are particularly challenging to keep up to date. These rules are hard to adapt or reconfigure when the network topology is modified, when new services are offered inside the corporate network or when specific threats are more likely over a certain period. The main reason is that up to several hundred or thousand rules must be configured in current firewalls. An update might require an in-depth

Table 1
Comparison of relevant previous works on adaptive security.

Work	Application	Domain	Measurements	Adapted control
Veloudis et al. (2016)	Access control	Cloud	Security Context Element, Permission, Context Pattern	Grant or deny access to sensitive data
Dasgupta et al. (2016)	Access control	Cloud and Mobile	Latency, Throughput, Storage Space Utilization, etc.	Request different authentication factors
Ashibani et al. (2017)	Access control	IoT (Smart Home)	Usage patterns, users' profiles and calendar	Privileges and permissions (based on rules)
Kumar et al. (2018)	Access control	Mobile (Android)	App functionalities, phone state, battery status, time, etc.	App access permissions
Psarra et al. (2020)	Access control and Encryption	Cloud (eHealth)	Identity of a user, role, access patterns, type of connection, etc.	Decision based on access control policies
Fazeen et al. (2014)	Encryption	Mobile	Sensitive words	Encryption type (high-strength algorithm for sensitive information)
Jahan et al. (2020)	Security controls	IoT (Autonomous systems)	State variables, their association with the system's functions, methods, and components, etc.	Add-Delete-Modify the system's functionality
Fernández Maimó et al. (2018)	Anomaly detection	5G	Depending on the anomalies to detect	Deploy more resources, replace the deep learning framework or the detection model
RiAS	All	All	Risk indicators and scores	All

Table 2
Comparison of relevant previous works on risk-based security.

Work	Application	Domain	Measurements	Adapted control
Chen et al. (2016)	Access control	Cloud (eHealth)	PV (the vulnerability of the current environment), PT (threatening) and PI (the integrity assessment)	Decision based on access control policies if certain risk thresholds are exceeded
Díaz-López et al. (2016)	Access control	Cloud	IT components and objects features, situational conditions, people description	Decision based on access control policies if certain risk thresholds are exceeded (encryption techniques, authentication mechanisms, etc.)
Steinegger et al. (2016)	Access control	Web applications	Authentication indicators according to the requirements of the web application	Decision based on pre-configurations and access control policies if certain risk thresholds are exceeded (reject requests, re-authentication, etc.)
Metoui et al. (2017)	Access control	Threat detection	The context, the function and the trust of the application	Risk Mitigation (e.g., data anonymization) and Trust Enhancement (e.g., stronger authentication)
Sepczuk and Kotulski (2018)	Access control	Cloud and Mobile	The user's security experience, type of service and authentication mechanism	Choice of the authentication mechanism or mechanisms according to the level of risk
Qin et al. (2018)	Decision-making	Industrial Control System	Attack strategy, security strategy, recovery strategy	Turn off systems, disconnect from the network, encrypt data, disable security systems, etc.
Tripathy et al. (2018)	Decision-making	Software Defined Networks	CVSS	Re-choosing the routing path
Petersen and Vankempen (2019)	Processing of data	Information systems	Input data and different identifiers and values (e.g., source or destination host, IP, port, allowed actions, allowed programs, etc.)	Forward data to event/platform manager for presentation to personnel.
RiAS	All	All	Risk indicators and scores	All

analysis of all of them to determine which rules to change and how to change them. Security managers often do not have time to conduct this analysis, so the rules often remain unchanged (furthermore, unrelated to the faced risk or the tolerated risk). This implies either that they apply too restrictive policies (causing usability issues, restricting functionality or increasing costs, for example) or do not protect the network adequately (which ends up causing security incidents).

Fig. 1 summarises the proposed approach based on a traditional control loop: the protected system or asset is the controlled process, the security control in the loop is the controller function. With static and predictable inputs, this function could be static and yet, be capable of automatically adjusting the behaviour of the asset to keep the actual risk below the tolerated risk level. Noise and variable inputs make adaptive control necessary.

In modern control engineering, the controller can be designed to adapt, adjusting its behaviour to changes in the context, assets, or desired output. However, traditional security controls are not adaptive, mainly because this capability of adaptation would make them more complex, sophisticated and, in the end, expensive. These are the main reasons why this paper proposes to add the capability of adaptation from outside the control, relying on the ”separation of concerns” design principle, using RiAS (**Ri**sk-based **A**daptive **S**ecurity):

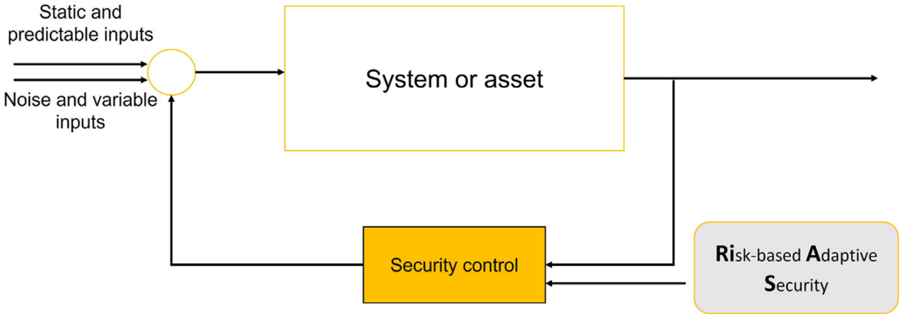
Static and predictable inputs
Noise and variable inputs
System or asset
Security control
Risk-based Adaptive Security

Fig. 1. High-level description of potential use cases.

- Resources available for adaptation: If the adaptation capability is embedded into the security control, control owners will have to limit the scope of what they adapt and how they adapt it to what they can afford to implement and reasonably manage within this specific control. With an external solution such as RiAS, resources devoted to adaptation are shared among all the security controls and are used exclusively to perform adaptations. Adaptation may be based on complex reusable mechanisms, and this will not affect controls complexity or costs. Furthermore, scalability, ease of updating, and expandability increase relying on an external solution because resources can be added to RiAS as required, its components can be modified if needed, and new layers or components can be deployed without affecting security controls design or deployment. These controls would require significant changes to perform these scale, update, or expansion processes with embedded adaptation capabilities. RiAS can be modified when required, security controls deployed in an organisation, likely not so often.
- High performance: In connection with the above, if RiAS is run as an external solution, the performance of adaptation decisions in terms of latency and throughput can be improved by adding more computing resources to the general-purpose platforms where the solution is executed. This high performance is more challenging to obtain executing adaptation decisions inside security control because computing resources devoted to adaptation will usually be minimal, and optimisations will be challenging to achieve. With RiAS, the primary responsibility is adaptation; there is no other function to optimise.
- Standardisation and reliability: If RiAS is independent of specific security controls, it is possible to standardise instrumentation, monitoring, the definition of rules and policies, etc., with solutions that are not tied to specific manufacturers, technologies or paradigms. Decision-makers, policy administrators and control owners will only have to deal with one adaptation model valid for all controls. Furthermore, adaptation mechanisms are responsible for gathering, storing, and analysing high-value information. Decision-makers must trust them to operate correctly. Outages and unreliable behaviour can have an immediate harmful impact on the organisation's security. An external solution such as RiAS can include security, fault tolerance and high-availability mechanisms that traditional security controls do not embed. Furthermore, if RiAS were to fail, the stand-alone security controls would not be affected; they would revert to traditional, static operation.
- Flexibility: An external solution such as RiAS is able to instrument different assets and their global context. On the other hand, an individual control may not have a holistic view of the complete operation environment simply because it does not have access to all distributed data sources. This capability to correlate data from different sources improves the quality of adaptation decisions. An external solution will likely offer helpful functionalities to human operators often not possible within individual security controls, not always providing the proper interfaces and APIs. Some examples could be visualising context information, analysing historical data about adaptation decisions, edition of rules and policies, etc.

This new model for adaptation, including the risk-based approach, implies significant research challenges. The following research questions summarise the most important:

- What type of architecture and decision flow best fit the pursued objective to provide the capacity of adaptation, from the outside to all types of security controls?
- How can risk be quantified to guide adaptation decisions? What attributes of an asset's context should be measured to know if the adaptation of security control is necessary considering risk-based criteria through Key Risk Indicators, Indicators of Compromise, Indicators of Attack or risk scores?
- What kind of adaptations are possible? How can we decide which is the best in each case? How can we express the allowed adaptations in a generic way (independent of the control)?
- How can the need for adaptation of a given control be expressed? Furthermore, how can this adaptation be made effective once the decision has been made?

The following sections of this article present a new model, RiAS, that aims to answer all these research questions.

4. Proposed model: Risk-based adaptive security

RiAS is a new model capable of adapting security controls to different risk scenarios. This section introduces the three-layer architecture and the three-step flow (measurement-decision-adaptation) proposed to support this model, as well as the assumptions under which it operates.

4.1. Underlying architecture and assumptions

The proposed model is based on a trusted agent that adapts security controls and countermeasures depending on the risk evolution. This agent can be executed in-house or consumed from a third party. The architecture supporting this model comprises three layers that can be deployed in a centralised or distributed way and may communicate explicitly or indirectly by sharing information using a knowledge repository.

These three layers are shown in Fig. 2. The Measurement layer comprises the M (Monitor) and the A (Analysis) components of the traditional MAPE-K loop. This layer is responsible for gathering data about protected assets, security controls and their context and transforming it into context-awareness to feed the following layer. The Decision layer is the P component (Plan) of the MAPE-K loop,

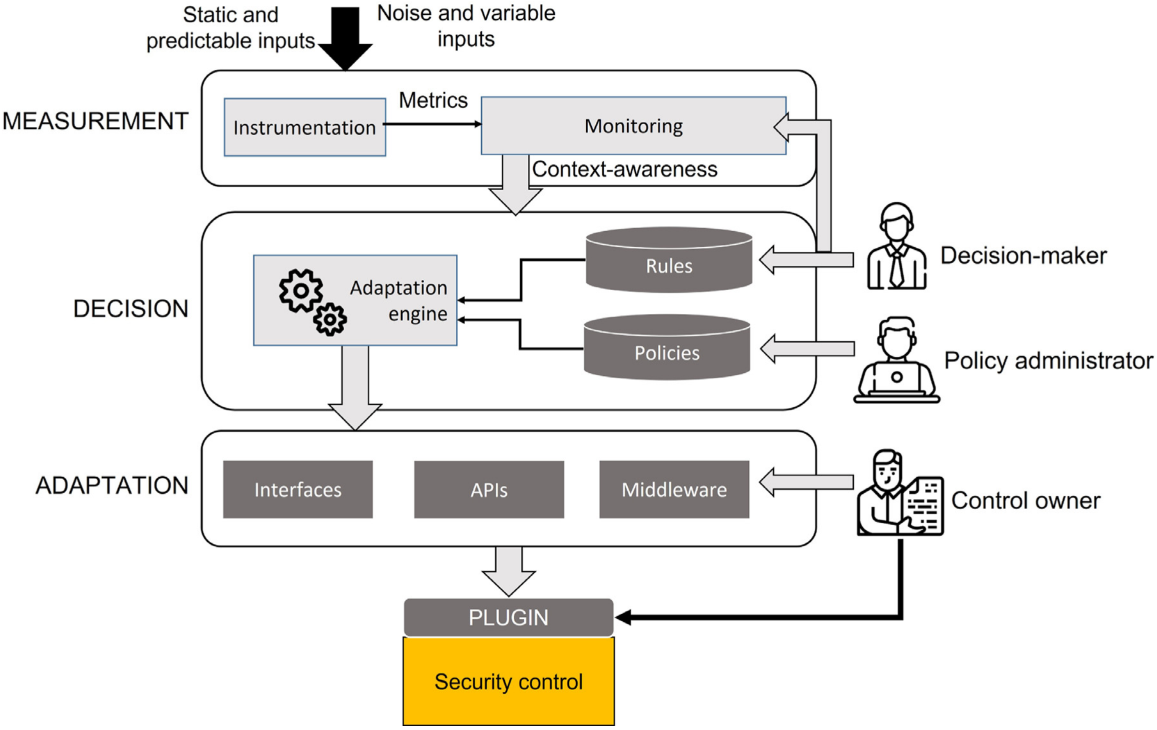

Fig. 2. Three-layer architecture supporting the proposed model.

where the adaptation engine resides. The K component (Knowledge), expressing adaptation goals and other relevant information required to make risk-based decisions through rules and policies, is also in this layer. Finally, the Adaptation layer corresponds to the Execution component (E) of the MAPE-K loop, responsible for making real the decided adaptations.

These three layers may interact with three different kinds of actors:

1. Decision-makers: Responsible for deciding how context is captured (including policy-triggers and events) and defining the rules that determine how, when or where the adaptation of security controls and countermeasures should be performed.
2. Policy administrators: Responsible for defining the policies that manage these adaptations. Therefore, for translating to a standard language the answer to a crucial question: why adapt a specific security control?
3. Control owners: Responsible for integrating the proposed architecture with specific controls and countermeasures to enable decided adaptations. They must provide the interfaces, APIs or middleware required to perform necessary changes.

4.2. Adaptation flow

The three layers composing the proposed architecture provide the necessary resources to the actors mentioned above to manage adaptation processes, govern decisions and adjust security controls, respectively. They also include the essential components to follow the adaption flow described in this section (Fig. 3).

4.2.1. Offline steps

By defining adaptation rules, decision-makers determine, offline, the kind of adaptation that should be performed in each case. Rules are specified at design time, and they can be updated, improved and completed later, after learning about their utility assessing adaptation performance.

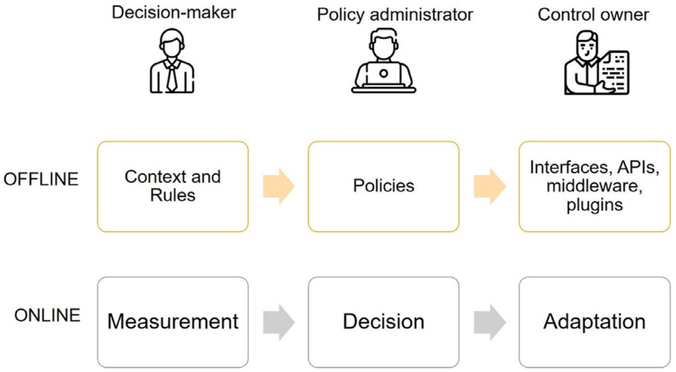

Fig. 3. Flow to perform risk-based adaptation of security controls.

The first question these rules have to answer is what should be changed in the security controls or countermeasures to observe the desired adaptation. This work categorises adaptation within three groups:

- Parametric: The adaptation is obtained by modifying the security control configuration, adjusting different parameters or tuning internal elements or components with different values.
- Architectural: In this case, the adaptation relies on some structural modification such as the removal, deactivation or addition of an element or component. Or a different deployment or interaction of the existing ones.
- Behavioural: The adaptation is performed by managing or using the security control differently, changing a protocol, flow, policy or procedure, for example.

The second question the decision-makers need to answer through the adaptation rules is when adaptation should be performed. Three timing strategies are considered in this work:

- Periodic: Adaptation is performed, always, in specific time windows determined by a fixed period. Once every hour, once a

Table 3
Summary of the offline steps of the RiAS model.

Step	Issues	Alternatives	Responsible
		Direct metrics: KRIs, IoCs, IoAs	
		Elaborated: Risk scores	
	What to measure	Monitoring: Policy-triggers and events	
		Configuration parameters	
		Architecture	
	What to adapt	Behaviour	
		Periodic	
		Event-driven	
	When to adapt	On-demand	
		Centralized	
		Distributed	
Rules	Where to decide	Hybrid	Decision-maker
		Reactive	
	How to adapt	Predictive	
		Changes in the operating context	
		Changes in the protected asset	
Policies	Why to adapt	Changes in the security goals	Policy administrator
		Interface	
		API	
		Middleware	
Adaptation logic	Which artefact	Plugin	Control owner

day, only once on a specific day, etc. This implies that even if it is decided that an adaptation of the security control is necessary, this adaptation cannot be carried out at an arbitrary moment but is somehow programmed.

- Event-driven: In this case, adaptation is performed, once decided, always when a specific event specified by the adaptation rule occurs. The event will often be the one that has triggered the adaptation (a change in the context, in the asset, in the target risk, etc.).
- On-demand: Adaptation is performed when the control owner gives her permission to do it. The adaptation, in this case, is manual instead of automated.

The third and last decision that adaptation rules have to capture is where the adaptation decision should be made. The approach may be centralised, distributed, or hybrid, depending on the available resources and the adaptation engine's demands and complexity. The adaptation rules must specify these aspects and the required allocation of the elements composing the three RiAS layers (executing in-house, on a public cloud, on edge resources, etc.).

On the other hand, policy administrators specify the policies that manage adaptation during this offline branch of the adaptation flow by determining why these adaptations must be performed and how the control should react.

Polices can be categorised into two groups. First, reactive policies that decide about adaptation when context-awareness means that specific changes have been observed. Second, predictive policies that decide about adaptation when current context-awareness allows us to predict that specific changes will happen in the short-term, trying to be proactive and avoid risks before they emerge.

What changes should be considered to decide controls adaptation? Mainly, three types, changes in the operating context of the security control (new threats, for example), changes in the protected asset (new configuration, a new architecture, new procedures) or changes in the tolerated risk or security goals.

Finally, control owners provide the interfaces, APIs, middleware, and plugins necessary to perform the required changes to the security controls that have to be adapted during these offline steps of the adaptation flow. These agents must develop the code and logic required to apply, at a low level, the decided adaptations to the security controls.

Table 3 summarises all these offline steps of the RiAS model.

4.2.2. Measurement

This is the first step of the online branch within the RiAS flow. Its mission is to carry out instrumentation and monitoring of the operating context and the protected asset. This quantification will enable decisions to be taken later on regarding the need for adaptation at the security control.

A heterogeneous set of raw data and elaborated metrics should be obtained during this step since very different adaptations could be decided and deployed (as introduced in the previous subsection). This collection and exposure of different metrics are called Instrumentation.

The instrumentation itself may enable adaptation decisions, but sometimes, specific triggers or events specified by adaptation policies or rules (a change in the context or the asset, for example) must be monitored to enable these decisions. In the proposed model, monitoring is the process of collecting, aggregating, and analysing metrics to improve context awareness by managing metrics over periods, establishing comparisons, recognising patterns, etc.

Summarising, the difference between Instrumentation (metrics) and Monitoring (context-awareness) mirrors the difference between data and information. Data is composed of measurements that quantify behaviour, while information is produced by analysing and organising data to build context that provides value: triggers and events.

Instrumentation As introduced before, RiAS metrics may be raw measurements of behaviours that can be observed and collected or more elaborated figures. However, all these metrics should meet some common characteristics that make them useful in the proposed model:

- Accurate: Measurable, relying on timely and reliable source data, objectively quantifiable and verifiable.
- Simple: Easy to measure, understandable, non-redundant (strongly correlated metrics should be avoided; the best of them should always be considered).
- Relevant: Strongly related to the specific considered risk to its likelihood or impact.
- Standardised: Properly defined and documented, therefore, traceable and comparable, enabling benchmarking.

The detailed definition of a wide range of raw data and metrics is beyond this article's scope. The following is simply a brief classification of the metrics that can be used in the proposed model.

Key Risk Indicators (KRI) provide high-level operational variables that can be considered a reliable basis for estimating or predicting risk exposure Rodriguez and Antonucci (2017). KRIs come from risk mapping processes, determining which phases of a process or activity may be exposed to specific risks. Therefore, these indicators allow us to determine which parts of the process or activity should be changed to reduce the overall risk exposure. In our case, the security control configuration, architecture or use. Typical indicators can be technical such as downtime or time to repair or recover, time to patch, and time to solve a code vulnerability. They can also be different, such as hours devoted to employees' training and awareness or user complaints.

Policies defined relying on KRIs usually work with one of the following three approaches to trigger an adaptation decision: the decision depends on changes observed in the KRI value, the decision depends on the KRI value being above or below a certain threshold (trigger level) or the decision depends on a target KRI value that is not fulfilled.

For example, if decisions about the adaptation of a firewall have to be made, one of the considered KRIs may be network downtime. Suppose this downtime suffers significant modifications from one measurement to another. In that case, if this downtime is above a fixed threshold or if the committed target for this downtime is not being guaranteed, adaptations might be performed to change the firewall rules or its location inside the corporate network, for example.

Low-level measurements such as Indicators of Compromise (IoC) or Indicators of Attack (IoA) can be used in RiAS too, in this case, to detect situations in which the protected asset has been or is being attacked by an adversary or compromised by malicious software Mavroeidis and Bromander (2017). IoCs are forensic artefacts, simple network, applications or operating system observations whose presence is reliable proof of a past security incident. While IoAs focus on detecting, in real-time, the intent of what an attacker is trying to accomplish, the steps that this adversary must conduct to succeed and the deployed TTPs (Tactics, Techniques, and Procedures).

IoCs are used when reactive adaptation policies need to evaluate if the operating context has changed, and new threats have arisen, or new attack patterns or TTPs are available, when the protected asset has changed and performed changes have made it more vulnerable, when the new tolerated risk would not allow specific incidents to occur, etc. In these cases, typical measurements regard network traffic, and user accounts activity, databases usage, number of requests to specific resources, presence of code or exploits in the memory.

For example, a file with a given MD5 hash in a temporary directory of a laptop can be associated with a banking trojan. If decisions about the adaptation of the anti-malware suite have to be made, this IoC may be helpful in improving its installation, configuration or scanning procedure.

IoAs are used when reactive adaptation policies need to evaluate if the protected asset is under attack at this moment, trying to proactively adapt the security control to mitigate the impacts of this attack. In this case, the measurement step may be more demanding because gathered information regards code execution, lateral movement or communication with command and control tools. All in real-time in order to enable timely decisions and adaptations.

For example, if decisions about the configuration of network Windows services have to be made, and specific SMB traffic is observed, adaptations to avoid adversary lateral movements can be performed to mitigate a possible WannaCry, NotPetya or Emotet attack.

Finally, it should be noted that more elaborated metrics are instrumental when predictive adaptation policies are defined. In this case, risk scores can be computed to predict the risk that a particular change or action could imply Ruan (2017). These scores can be obtained from well-known statistical techniques or, more recently, from machine learning or deep learning techniques. For example, suppose decisions about the adaptation of an identity management system have to be made. In that case, an end user's request to an e-commerce site can be scored as suspicious (an anomaly) given her previous history, the price of the selected item, the browser used to perform the purchase, the geographical location of the IP etc. A high-risk score for this operation may lead to asking for a second authenticator, for example, adapting the authentication process to the performed prediction about risk.

Monitoring Previous works define context information as any information about the situation of an entity (where an entity can be a person, place or object). RiAS is designed to represent, as context, not only the metrics mentioned above but also the different types of triggers and events defined by decision-makers. Each time a new adaptation policy or an event-driven rule is defined, the Monitoring module is configured to monitor the specific trigger or event specified by this adaptation policy or rule and disseminate the results of this monitoring with the rest of context-awareness to the following layer of the RiaS architecture, the Decision layer.

4.2.3. *Decision*

Once context-awareness is obtained and updated through the measurement step, this step is responsible for making adaptation decisions by applying the rules and policies defined by decision-makers and policy-administrators.

The rules and policies required to make decisions in RiAS are defined with common semantics rules. Policies are the central element for decisions. A RiAS policy specifies whether an asset should be adapted given the current context to mitigate risk to its tolerated value. Policies are composed of the following elements:

- Name: Identifier of the policy.
- Owner: The policy-administrator who owns the policy, responsible for its definition, management, update, etc.
- Type: Reactive or predictive.
- Control: The control or set of controls affected by the policy. Different schemes composed of family names, numbers and identifiers can be used in different scenarios.
- Adaptation conditions: A list of predicates, at least one, each with an antecedent and a consequent part. If the antecedent part (dependent on information from the Monitoring module) takes the value ”true”, the consequent part is evaluated. This consequent is expressed in the form of a rule.

Rules are composed of the following elements:

- Name: Identifier of the rule.
- Owner: The decision-maker who owns the rule, responsible for its definition, management, update, etc.
- Timing: Periodic (specifying a period), event-driven (specifying the event that triggers the adaptation) or on-demand.
- Category: Parametric, architectural or behavioural.
- Control: Each one would be composed of an action and an artefact (set of specific interfaces, APIs, middleware or plugins that must be used to perform the adaptation for this specific control).

The Adaptation engine is responsible for policy evaluation; all the predicates have to be checked. If the antecedent part of a predicate is true, the rule specified in the consequent part is implemented. Therefore, the adaptation engine listens to the manual activation of policies relying on-demand rules and to event monitoring at the Monitoring module for policies relying on event-driven

Table 4
Policy example for RiaS.

Name: Firewall-adaptation. *Owner:* Security admin. *Control:* Corporate firewall. *Type:* Predictive. *Adaptation conditions:* Predicate 1: *observed*(Trigger1 when RS1 greaterThan X) → Rule 1. Predicate 2: *observed*(Trigger2 when RS2 greaterThan Y) → Rule 2.

Table 5
Rules example for RiaS.

Name: Rule 1.
Owner: Network admin. *Timing:* *Event-driven:* observed(Event1). *Category:* Parametric. *Controls:* *Action:* Corporate firewall *apply*(restrictive mode). *Artefact:* *use*(Corporate firewall configuration API).
Name: Rule 2.
Owner: Network admin. *Timing:* *On-demand.* *Category:* Architectural. *Controls:* *Action:* Corporate firewall *apply*(Anti DDoS module). *Artefact:* *use*(Network orchestration middleware).

rules that must be executed immediately. It also works with the clock, activating policies periodically when they rely on periodic adaptation rules.

For example, if a policy-maker needs to specify that the configuration of a network perimeter control must be proactively modified to mitigate DDoS attacks, an adaptation policy should be written (Table 4). This policy states that if a risk score (RS1) is above a threshold *X* and produces a triggering event (Trigger1), rule 1 should be applied. If risk score 2 (RS2) is above a greater threshold *Y* and produces a different triggering event (Trigger2), rule 2 should be applied instead. Table 5 shows these rules.

The first one is an event-driven rule that makes firewall rules more restrictive when the risk of a DDoS attack is predicted to be above a threshold. The adaptation is performed through the configuration API as soon the Monitoring module observes *Event1* (*RS1 greaterThan X*, in this case, is the same than *Trigger1*) and propagates this event.

The second is an on-demand rule that changes the firewall architecture adding a new anti-DDoS module when the risk of a DDoS attack is predicted to be above a more significant threshold. The adaptation is performed through the network orchestration middleware when the network administrator grants permission to do it (this adaptation, more complex, requires some manual intervention and, therefore, is on-demand).

4.2.4. *Adaptation*

Given that the approach chosen in this research is to separate the adaptation code and logic (provided by control owners during the offline steps) from the adapted security controls, the same artefacts may be used by the Adaptation engine to adapt different controls (re-utilization is not only possible, it is also recommendable) or different artefacts may be required to perform different adaptations (parametric, architectural or behavioural) of the same control.

For this reason, the following characteristics would be desirable in the used or consumed adaptation artefacts:

- Simplicity, adaptation artefacts should enable helpful and consistent abstractions easily discoverable and usable by the Adaptation engine.
- As decoupled of specific security controls as possible, relying on standard specifications, protocols, semantics, etc.
- As decoupled of specific implementation details as possible. Since the adaptation decision may be centralized, distributed or hybrid, the architecture layers supporting RiAS could be deployed differently. It is essential that both, the adaptation code and logic, can be consumed from the adaptation layer with independence on where it is executing.
- Organized around capabilities, not technology. Adaptation code and logic should be split into artefacts organized around capabilities (functionality). Such artefacts take a broad-stack implementation of software for that capability, including APIs, persistent storage, required external interfaces, etc. Consequently, the artefacts are cross-functional, increasing reusability and avoiding complex changes in different artefacts every time a rule, a policy or a security control is added or updated.
- Focused on automation, allowing adaptation of security controls minimizing human intervention. Furthermore, designed for failure, avoiding the blocking of the security control and capable of managing versioning and rollbacks in the event of the failure during the adaptation process.

5. Validation and evaluation

A real use case has been used to validate and evaluate the RiAS model (architecture and flow). Specifically, we are interested in assessing the proposed model's behaviour when adapting a Web Application Filter (WAF) protecting the online video-sharing platform shown in Fig. 4.

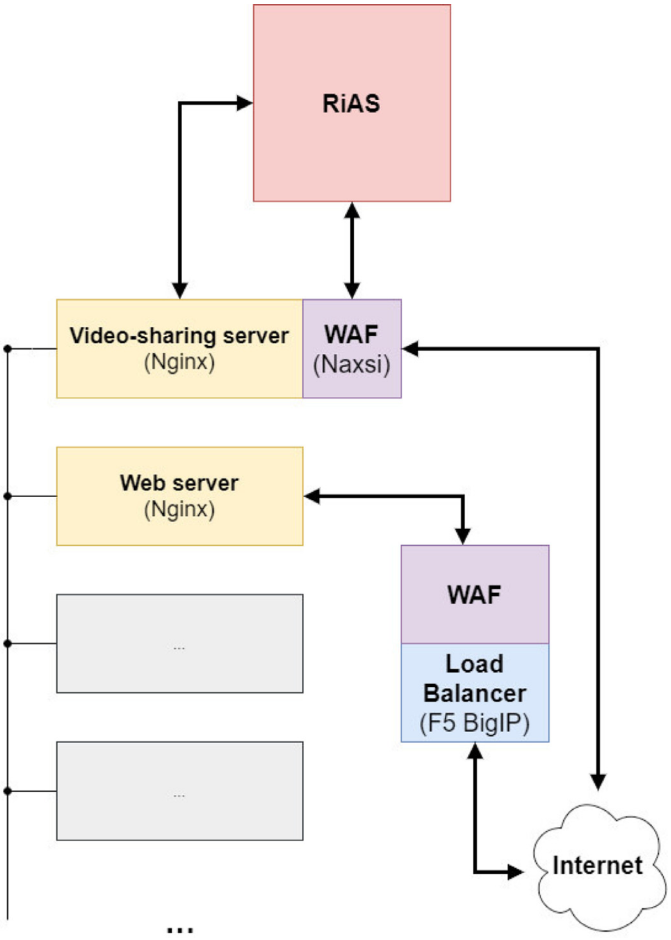

Fig. 4. Use case architecture using RiAS.

5.1. Use case description

This platform is deployed on different servers. In this use case, we have focused on one of them, which attend videos on demand (with hostname v-sharing.urjc.es). Due to the significant network traffic amount it has to attend, this server cannot be found behind the corporate load balancer (F5 BigIP). Furthermore, the available server resources are limited, so it is only protected with some rules in a software WAF on the server. It is also essential to know that the web portal is offered by another server (with hostname play.urjc.es) and that it is this server that provides users with the URLs of the videos they want to watch. Another WAF protects this last server on the load balancer.

Summarizing: when a user accesses a video (for example, through the URL https://play.urjc.es/video/86a24), the video file embedded in the player on the web page is requested to a different URL (https://v-sharing.urjc.es/track/86a24.mp4).

Because bandwidth traffic is very high when serving video on demand and whitelisting mode in a WAF would slow down the platform, it has been decided to use RiAS to adapt the WAF configuration on the video on demand server (v-sharing.urj.es). This server uses Nginx software to serve the videos; the selected WAF is Naxsi.

The objective in this use case is to analyze the logs generated by the video on demand server (an Nginx webserver). If a potential attack is detected, RiAS needs to apply restrictive settings to the WAF (Naxsi). Likewise, when a reasonable time has passed since the potential attack, and there is no indication that it is still going on, RiAS need to adapt the WAF configuration to be less restrictive and consume fewer machine resources.

5.2. Implementation and validation

The first prototype of RiAS has been implemented using mainly JavaScript and web technologies, with Node JS. Rules and policies have been written in JSON. The model has been run centrally, with all the modules executing on the same server. The Measurement layer relies on a MongoDB database.

5.2.1. Offline steps

The decision-maker is the WAF administrator; two different rules have been defined:

- The first, R1, of the ”event-driven” type, is responsible for applying a parametric adaptation (configuration and creation of WAF rules to work using a white list). The event that triggers this rule’s application (that has to be configured within the Monitoring module of RiAS) is the measurement of 10 or more HTTP 404 status codes in the Nginx server logs in one minute (a specific IoC).
- The second, R2, of type ”periodic”, is also in charge of applying a parametric adaptation (configuration of WAF rules to work with the default configuration). The period for this rule is at 12 a.m. (requires a short restart of the web service and takes place in an hour with little traffic).

In this way, the responsible for the video-sharing platform is the policy administrator. The adaptation policy is reactive: R1 is applied when an attack on the video server is suspected; R2 is applied when R1 has been applied, and there is no indication that the potential attack persists.

It has also been necessary to develop several plugins; the control owner is the WAF administrator again. These plugins connect to the video server to create the relevant rules in the WAF, apply the new configuration and restart the Naxsi and Nginx services when the adaptation is required.

Table 6
”IoC-HTTPcod” measurement.

date	cod	count

Table 7
”IoC-RiASrules” measurement.

date	rule

5.2.2. Measurement

The instrumentation in this use case is based on IoCs:

- On the one hand, an updated record of HTTP status codes is kept. This record is updated every minute, grouping codes by type and counting the number of occurrences of each code in that last minute. If it finds 10 (or more) HTTP 404 status code, the same process acts as antecedent in the Policy, launching the execution of rule R1 (We have called the antecedent EVENT-HTTPcod-High). This process also creates another antecedent (we have called EVENT-HTTPcod-Low), which, as with EVENT-HTTPcod-High, checks the HTTP 404 status codes and, if they have dropped from 10 a minute, launching (if another antecedent is also fulfilled) the execution of rule R2. This measurement, which we have called, IoC-HTTPcod, is collected by a plugin that, every minute, retrieves these values from the Nginx access.log of the video-on-demand server and stores them in the RiAS MongoDB database. Data is stored with the structure shown in Table 6. The *date* parameter corresponds to the date/time the data was collected, *cod* corresponds to the HTTP code and *count* to the number of occurrences of this code in the log file.
- A record of applied rules is also kept (along with the date/time of application). This second measurement, which we call IoC-RiASrules, is also collected through a plugin similar to the previous one, which retrieves the values from the RiAS logs and stores them in the MongoDB database. The data is stored as shown in Table 7. The *date* parameter corresponds to the date/time the data was collected, and the *rule* parameter is the identifier of the applied rule. In this case, there is also an antecedent, called EVENT-StrictModeWAF, which, checking if the last rule executed corresponds to R2. In case this is so, when the EVENT-HTTPcod-Low antecedent is met, rule R2 will be run.

So, IoC-HTTPcod corresponds to the measurement of the HTTP codes of the Nginx server that serves the videos on demand (context regarding the protected asset) and IoC-RiASrules to the measurement of the application of the different RiAS rules (context regarding RiAS itself and the WAF adaptation).

5.2.3. Decision

Decision making is carried out with the policy “Protect-video-server-from-a-web-attack” in Table 8 and rules in Table 9.

5.2.4. Adaptation

As has been already mentioned, the adaptation of the WAF configuration is performed using two different plugins; one is used by the rule R1 and the other by the rule R2. Both adaptations will be performed using the same artefact (which we have called CONNECTION-SSH-WithCert), which will create an SSH connection to the video on demand server authenticating with a certificate.

The two plugins share three modules:

- The first consists of a code fragment to connect to the video on demand server using the specified artefact.

Table 8
Policy for the use case.

```
[
    {
        "name": "Protect-video-server-from-a-web-attack",
        "owner": "Video platform admin",
        "type": "reactive",
        "control_id": ["WAF v-sharing.urj.es"],
        "conditions": [
            {
                "antecedent": ["EVENT-HTTPcod-High"],
                "consequent": ["R1"]
            }, {
                "antecedent": [
                    "EVENT-HTTPcod-Low",
                    "EVENT-StrictModeWAF"
                ],
                "consequent": ["R2"]
            }
        ]
    }
]
```

- The second is responsible for making a backup of the Nginx configuration file(s) and the Naxsi rules file(s) that are going to be modified.
- Finally, the third module restarts the services affected by adaptations (Nginx and Naxsi). In case of failure, it is also responsible for restoring the backup of the configuration files and rules and returning an error to the RiAS Adaptation engine (which will notify the control owner indicating that a problem has occurred when applying the adaptation).

The created plugins work as described below:

- The first plugin, RestrictedMode:
  - After connecting via the selected artefact to the video-on-demand server, check the Nginx and the Naxsi status.
  - In case Nginx and Naxsi are running, the Naxsi and Nginx rules and configuration files are backed up.
  - A query is made to the video database to retrieve all the existing videos on the platform and to be able to form all the video URLs that must be accessible.
  - Once the video URLs that must be accessible was formed, the necessary rules are created in a WAF rules file (naxsi_whitelist.rules) to allow access to these URLs through whitelist (option ”MainRule” in Naxsi). All the parameters that can be passed from the URL are also considered.
  - This rules file is included in the Nginx configuration (in the sites-enabled file).
  - The restart module for the Nginx and Naxsi services runs.
- The second plugin, called NormalMode:
  - After connecting via the selected artefact to the video-on-demand server, check the Nginx and the Naxsi status.
  - In case both are active, the Naxsi and Nginx rules and configuration files are backed up.
  - If it exists, the rules file created by RestrictedMode (naxsi_whitelist.rules) is deleted, and the Nginx configuration (sites-enabled file) is modified not to include that file, leaving only the default rules file (naxsi_default.rules).
  - The restart module for the Nginx and Naxsi services runs.

5.3. Performance analysis

Two sets of experiments have been conducted to measure the response time and the resource consumption of the first RiAS prototype. The first set of experiments has been performed during an average day (normal behaviour of the video-sharing platform users). The second set of experiments has been performed on a day when the platform is under attack.

If RiAS were not used, there would be two possible scenarios, both static. In the first one, we work with the least restrictive configuration of the WAF, based on blacklists. When the DoS attack starts, the WAF itself cannot detect or prevent it; it cannot react. So the attacker achieves his goals. The attack may be detected by adding a new IDS-type solution to the architecture, but this solution would not have the ability to react to the attack, only to raise an alert for a human operator to act. Furthermore, it would increase the cost of the architecture.

In the second, we work with the most restrictive WAF configuration, based on whitelists. All the time. This would prevent the particular attack tested in the use case, but at the cost of consuming a large amount of resources on the server all the time, even when this consumption would not be necessary. It has to be considered that the average consumption of CPU and RAM memory

Table 9
Rule 1 and Rule 2 for the use case.

```
[ {
        "name": "R1",
        "owner": "WAF administrator",
        "category": "parametric",
        "timing": {
            "type": "event-driven",
            "period": null, "event-trigger": "EVENT-HTTPcod-High"
        },
        "controls": [
            {
                "control_id": "WAF v-sharing.urj.es",
                "action": "RestrictedMode",
                "artefact": "CONNECTION-SSH-WithCert"
            } ]
}, {
        "name": "R2",
        "owner": "WAF administrator",
        "category": "parametric",
        "timing": {
            "type": "periodic",
            "period": "00 00 * * *",  "event-trigger": null
           },
        "controls": [
            {
                "control_id": "WAF v-sharing.urj.es",
                "action": "NormalMode",
                "artefact": "CONNECTION-SSH-WithCert"
            } ]
}, ]
```

at the server with the WAF configured to use blacklists and 100 concurrent users accessing the video application is 19% and 76%, respectively. On the other hand, the average CPU and RAM memory consumption at the server with the WAF configured to use whitelists and 100 concurrent users accessing the video application is 24% and 83%, respectively.

By using RiAS, the least restrictive WAF configuration can be used to save server resources and to be able to handle more concurrent user requests. However, at the first sign of increased risk, the WAF can be adapted automatically to the new scenario, working with whitelists and dealing with the potential attack. When the situation returns to normal, the WAF can switch back to the original configuration. This process could be performed as many times as necessary, automatically adapting the security control to the tolerated risk at any given time and the context of the asset being protected.

The tests were conducted with a centralised RiAS implementation running on a Linux PC with an Intel Core i7@3.6 GHz Processor and 32 GB of RAM. The results for each performance figure have been obtained by executing the experiments ten times, and computing arithmetic means obtaining the following results:

- **Response time:** The average time of a decision cycle in the first set of experiments (no necessary adaptation), from the moment the policy is recovered from the database until it is decided that no adaptation is needed, is 23.4 ms (with 2.3 ms of standard deviation). The average time of a decision cycle in the second set of experiments (required adaptation), from the moment the policy is recovered from the database until the adaptation is launched in the adaptation layer, is 49.7 ms (with 3.9 of standard deviation). Again on average, 6.1 ms are consumed by the antecedent evaluation. The consequent execution consumes the rest of the time, implying recovering the rule (27.2 ms) and triggering specified actions through available artefacts. Measurements have been performed when using R1 because R2 is periodic, and the adaptation is not performed in real-time.
- **CPU and memory usage:** The execution of RiAS with one policy and two rules has implied peaks of 0.4% of CPU utilisation and 29 MB of RAM memory.

Furthermore, scalability and usability have been assessed. Different simulations have been performed to measure scalability. More policies have been added to the Adaptation engine in these simulations, all of them working with different antecedents and consequents to avoid the effects of reuse on resource consumption. We observe that resource consumption (CPU and RAM) is linearly increased according to the number of policies up to 100. The limiting resource is the CPU; around 100 policies response times worsen significantly. This is expected in such a computer setup with limited computing resources. A more powerful server or a distributed solution should be deployed to execute RiAS managing more than 100 policies.

Regarding usability, the video platform administrator and the WAF administrator have been involved in this use case. We have asked them to translate their security requirements to different policies and rules through the RiAS prototype interfaces. They have been able to complete this task after a fast walk-through of the solution. We have also asked them to notify us if they are not satisfied with the performed adaptations during the experiments mentioned above. Some observations can be made:

- During the day of normal operation, there was one unnecessary adaptation due to a ”false positive” of the defined policy. This adaptation caused the WAF to switch to a restrictive mode without being necessary. Changes were undone by rule two in the time slot specified for it without major consequences.
- During the day with the platform under attack, the WAF adaptations were performed as soon as the selected IoC was observed (ten HTTP 404 status codes in one minute). This real-time adaptation of the WAF configuration allowed the policy manager and decision-maker to react to a context change to manage risk according to their needs.

These experiments have demonstrated the importance of properly configuring the Measurement layer (choosing the right metrics and context-awareness information) and accurately expressing the policy antecedent to avoid unnecessary adaptations and, at the same time, react within a reasonable time to context changes.

Given the inherent complexity of coordinating the three layers composing the RiAS architecture (measurement, decision and adaptation) and the three agents participating in the adaptation flow (policy administrators, decision-makers and control owners), the proposed approach may have some limitations in specific scenarios. Mainly because RiAS cannot ensure a proper configuration of the Monitoring module to trigger a specific policy or that there exists a proper artefact to perform the specific action required by a specific rule.

That is why we are adding to RiAS templates and examples of context-awareness configurations, policies, rules and artefacts for representative assets and situations that will help the different agents extract the RiAS model full potential and understand its limitations.

6. Conclusion

The continually increasing number of cyber threats and their rapid evolution raise significant security concerns regarding traditional static design, configuration and policy definition of security controls, countermeasures and safeguards. One approach to alleviate these concerns is to enable adaptive risk-based security controls to adapt to changes in the operating context, the protected asset or the security goals. However, the cost of including this dynamism, coupled with the heterogeneity and complexity of current scenarios (web, mobile, cloud, IoT, 5G, etc.), hinders the formulation of suitable solutions for different hardware and software security controls used in different application domains.

This work has proposed RiAS, a model for Risk-based Adaptive Security, allowing the adaptation of security controls through the semantic representation of adaptation policies and rules. The proposed model enables parametric, architectural or behavioural adaptation of security controls using a three-layer architecture and a three-step flow (measurement, decision and adaptation). This model has been validated in a real use case, clearly demonstrating the following characteristics: (i) It can perform different kinds of adaptation, with different timing, decision and risk-management strategies. (ii) It enables stakeholders (decision-makers, policy administrators, control owners, and users) to efficiently collaborate to define their particular security and business requirements to guide adaptation decisions. (iii) It provides automated (and almost real-time, if required) adaptation of security controls, decoupling the adaptation decision and logic from the control itself.

Further research is required to demonstrate the feasibility of our proposal in distributed environments. We are currently working on providing RiAS as a service, implementing a first distributed prototype of the solution, which could be offered using the cloud computing or edge computing paradigms. Another critical research challenge that we want to address in the short term is incorporating some memory and intelligence into the Monitoring module and the Adaptation engine to learn from past decisions and take their results into account in future decisions. If these components are able to learn, they can significantly assist policy managers and decision-makers in their work.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Miguel Calvo: Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft. **Marta Beltrán:** Conceptualization, Validation, Investigation, Writing – review & editing, Supervision, Funding acquisition.

Acknowledgements

This research has been supported by the Madrid region (EdgeData, Grant Ref. P2018/TCS-4499) and by a research contract with Rated Power (art.83 M2186). Miguel Calvo is supported by grants from the Rey Juan Carlos University (ref. C-PREDOC21-007).

References

Arcaini, P., Riccobene, E., Scandurra, P., 2015. Modeling and analyzing mape-k feedback loops for self-adaptation. In: 2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, pp. 13–23.

Ashibani, Y., Kauling, D., Mahmoud, Q.H., 2017. A context-aware authentication framework for smart homes. In: 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 1–5.

Chen, A., Xing, H., She, K., Duan, G., 2016. A dynamic risk-based access control model for cloud computing. In: 2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom), pp. 579–584.

Cuadra, A., Aracil, J., 2017. Context-aware security framework based on traffic anomaly detection indicator. Telecommun Syst 65 (2), 319–330.

Dasgupta, D., Roy, A., Nag, A., 2016. Toward the design of adaptive selection strategies for multi-factor authentication. Computers & Security 63, 85–116.

Díaz-López, D., Dólera-Tormo, G., Gómez-Mármol, F., Martínez-Pérez, G., 2016. Dynamic counter-measures for risk-based access control systems: an evolutive approach. Future Generation Computer Systems 55, 321–335.

Fan, X., Li, C., Dong, X., 2019. A real-time network security visualization system based on incremental learning. J. Visualization 22 (1), 215–229.

Fazeen, M., Bajwa, G., Dantu, R., 2014. Context-aware multimedia encryption in mobile platforms. In: Proceedings of the 9th ACM Annual Cyber and Information Security Research Conference, pp. 53–56.

Fernández Maimó, L., Perales Gómez, L., García Clemente, F.J., Gil Pérez, M., Martínez Pérez, G., 2018. A self-adaptive deep learning-based system for anomaly detection in 5g networks. IEEE Access 6, 7700–7712.

Graur, F., 2017. Dynamic network configuration in the internet of things. In: 2017 5th International Symposium on Digital Forensic and Security (ISDFS), pp. 1–4.

He, Y., Mendis, G.J., Wei, J., 2017. Real-time detection of false data injection attacks in smart grid: a deep learning-based intelligent mechanism. IEEE Trans Smart Grid 8 (5), 2505–2516.

Jahan, S., Riley, I., Walter, C., Gamble, R.F., Pasco, M., McKinley, P.K., Cheng, B.H., 2020. Mape-k/mape-sac: an interaction framework for adaptive systems with security assurance cases. Future Generation Computer Systems 109, 197–209.

Kumar, S., Shanker, R., Verma, S., 2018. Context aware dynamic permission model: A retrospect of privacy and security in android system. In: 2018 International Conference on Intelligent Circuits and Systems (ICICS), pp. 324–329.

Lara, E., Aguilar, L., Sanchez, M.A., García, J.A., 2019. Adaptive Security Based on MAPE-K: A Survey. In: Applied Decision-Making. Springer, pp. 157–183.

Liu, T., Tian, J., Gui, Y., Liu, Y., Liu, P., 2017. Sedea: state estimation-based dynamic encryption and authentication in smart grid. IEEE Access 5, 15682–15693.

Mavroeidis, V., Bromander, S., 2017. Cyber threat intelligence model: An evaluation of taxonomies, sharing standards, and ontologies within cyber threat intelligence. In: IEEE European Intelligence and Security Informatics Conference (EISIC), pp. 91–98.

Metoui, N., Bezzi, M., Armando, A., 2017. Risk-based Privacy-aware Access Control for Threat Detection Systems. In: Transactions on Large-Scale Data-and Knowledge-Centered Systems XXXVI. Springer, pp. 1–30.

Mohsin, A., Asghar, S., Naeem, T., 2016. Intelligent security cycle: A rule based run time malicious code detection technique for soap messages. In: 2016 19th International Multi-Topic Conference (INMIC), pp. 1–10.

Parampottupadam, S., Moldovann, A., 2018. Cloud-based real-time network intrusion detection using deep learning. In: 2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), pp. 1–8.

Bursell, M., 2019. Dynamic configuration in cloud computing environments. US Patent App. 16/287, 747.

Petersen, C. L., Vankempen, M., 2019. Risk based priority processing of data. US Patent App. 16/116,335.

Poolsappasit, N., Dewri, R., Ray, I., 2011. Dynamic security risk management using bayesian attack graphs. IEEE Trans Dependable Secure Comput 9 (1), 61–74.

Psarra, E., Verginadis, Y., Patiniotakis, I., Apostolou, D., Mentzas, G., 2020. A context-aware security model for a combination of attribute-based access control and attribute-based encryption in the healthcare domain. In: Barolli, L., Amato, F., Moscato, F., Enokido, T., Takizawa, M. (Eds.), Web, Artificial Intelligence and Network Applications, pp. 1133–1142.

Qin, Y., Zhang, Q., Zhou, C., Xiong, N., 2018. A risk-based dynamic decision-making approach for cybersecurity protection in industrial control systems. IEEE Transactions on Systems, Man, and Cybernetics: Systems 1–8.

Rodriguez, A., Antonucci, D., 2017. Monitoring and Review Using Key Risk Indicators (KRIs). In: The Cyber Risk Handbook: Creating and Measuring Effective Cybersecurity Capabilities. Wiley Online Library, pp. 159–170.

Ruan, K., 2017. Introducing cybernomics: a unifying economic framework for measuring cyber risk. Computers & Security 65, 77–89.

Sasubilli, S.M., Dubey, A.K., Kumar, A., 2020. Hybrid security analysis based on intelligent adaptive learning in big data. In: 2020 International Conference on Advances in Computing and Communication Engineering (ICACCE), pp. 1–5.

Sepczuk, M., Kotulski, Z., 2018. A new risk-based authentication management model oriented on user's experience. Computers & Security 73, 17–33.

Sion, L., Yskout, K., Van Landuyt, D., Joosen, W., 2018. Risk-based design security analysis. In: Proceedings of the 1st International Workshop on Security Awareness from Design to Deployment, pp. 11–18.

Steinegger, R.H., Deckers, D., Giessler, P., Abeck, S., 2016. Risk-based authenticator for web applications. In: Proceedings of the 21st European Conference on Pattern Languages of Programs.

Tripathy, B.K., Das, D.P., Jena, S.K., Bera, P., 2018. Risk based security enforcement in software defined network. Computers & Security 78, 321–335.

Varadharajan, V., Karmakar, K., Tupakula, U., Hitchens, M., 2019. A policy-based security architecture for software-defined networks. IEEE Trans. Inf. Forensics Secur. 14 (4), 897–912.

Veloudis, S., Verginadis, Y., Patiniotakis, I., Paraskakis, I., Mentzas, G., 2016. Context-aware security models for paas-enabled access control. In: Proceedings of the 6th International Conference on Cloud Computing and Services Science - Volume 1 and 2. SCITEPRESS - Science and Technology Publications, Lda, pp. 202–212.

Yasaei, R., Hernandez, F., Al Faruque, M.A., 2020. Iot-cad: Context-aware adaptive anomaly detection in iot systems through sensor association. In: 2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD), pp. 1–9.

Miguel Calvo received the B.E in Software Engineering from Rey Juan Carlos University (URJC) in 2017 and the M.S degree in Cyber Security from Oberta Catalunya University (UOC) in 2018. He is currently a PhD student at the Department of Computing at URJC. He also works as a senior software developer and systems administrator for the IT department at Universidad Rey Juan Carlos and as a visiting professor at the Master in Cybersecurity and Privacy at the same university. His research interests include adaptive cyber security, identity management, cloud security and critical infrastructures protection.

Marta Beltrán received the master's degree in Electrical Engineering from Universidad Complutense of Madrid (Spain) in 2001, the master's degree in Industrial Physics from UNED (Spain) in 2003 and the PhD degree from the Department of Computing, Universidad Rey Juan Carlos, Madrid (Spain) in 2005. She is currently working with this department as an Associate Professor. She is the leader of the GAAP research group, co-founder of the Cybersecurity Cluster and she has published extensively in high-quality national and international journals and conference proceedings in the areas of security and privacy, and parallel and distributed systems. Her current research interests are Cloud computing, Edge/Fog Computing and Internet of Things, specifically, identity management and privacy-preserving mechanisms for these paradigms.