Analyzing Effectiveness of Security by design Model in a Cloud application development lif ecycle which uses IAC (Infrastructure As Code)

Aswani Poolakkad Bhaskaran

Cybersecurity And Human Factors

Bournemouth University

Bournemouth, United Kingdom

s5425771@bournemouth.ac.uk

***Abstract*—**Infrastructure as Code (IAC) uses DevOps while SDLC (Software Development Lifecycle). Cloud is getting big day by day and the data in the cloud environment is also getting big, protecting the data from security threats is important for both the client and the service provider. Also, the Cloud Application development lifecycle cannot be done through the traditional Software Development lifecycle because of the virtual implementation [1]. MUSA framework can be a powerful alternative. Security by Design enables the developers to ensure that the application is secured from the beginning of the development. In this paper, an analysis of the effectiveness of the security by design process in a cloud application development lifecycle with the help of the MUSA project and if Terraform is the tool for the deployment process.

Keywords—Security by Design, Threat, Cloud Computing, Infrastructure as Code.

I. INTRODUCTION

Most organizations using cloud architecture for managing their data depend on their cloud providers to keep it secure. However, with the increasing adoption of the cloud and an increase in cybercrime, trusting entirely on a service provider may not be enough. Embedding default security by design models into the cloud environment ensures a low-security risk. Meanwhile using the traditional software development lifecycle in cloud development is not feasible due to the fast developmental need, the agile nature [1], and the need for flexibility, so most of the clouds are created by using DevOps. The agile development process promises the organization a cloud with all needed features in reduced time and costs in the development process. But using DevOps with an inappropriate security measure may increase the risk of threats. To overcome the problem, European Union Agency for Network, and Information Security (ENISA) has suggested adopting Security Level Agreements between the organization and the network providers. Infrastructure as Code is one of the recent developmental techniques that can be used to create a cloud that can be fully created and configured by programs. The deployment speed and quick iterations are the main benefits of using IAC [2]. The usage of IAC (Infrastructure as Code) scripts is helpful for practitioners to configure and provision their development environments [2]. An analysis of security by design architecture and their effectiveness in cloud development in Infrastructure as Code is presented in this project. The MUSA architecture is taken as an example of security by design architecture and pretending that Terraform is used for the deployment of infrastructure as code. At the end of the paper, architecture combined with the MUSA architecture and the terraform platform is proposed.

A. Security By Design

Security by Design is a philosophy where there are many methods and techniques to explore, ensuring that the appropriate usage of one is particularly important to achieve the best output [8]. An Organization can choose its security methodology according to its needs and priorities accordingly. For example, if a financial organization is creating a security model by security by design, then there can be the Principle of least privilege and the Principle of Failing since each person in the organization has their personas. To implement the principles of Security by design need a joint effort of the creative team, operation, and the security team. The major Security principles which can be used are Principle of Least Privilege, Principle of Separation of Duties, Principle of Defense in Depth, Principle of Failing Securely, Principle of Open Design, Principle of Avoiding Security by Obscurity, Principle of Minimizing Attack Surface Area [3]. Based on Security by design, a concept called Secure Developmental life cycle (SDL)is defined and this model is successfully implemented in many organizations as of now. The Familiar examples are Cisco SDL (Security Developmental Life), Microsoft SDL, etc. And all these models use threat modeling continuously throughout the developmental process.

B. Infrastructure as code and its implementation in an organization

DevOps and agile development methods have been introduced to be more effective and efficient in the market. Infrastructure As code is a method that uses DevOps to create software i.e., the development environment in IAC is the collaboration of the Development team and the operation team. This process decreases the time taken for developing

software, so the installation process [4]. But the integration of security in this model is extremely complicated since DevOps needs a good automation capability to implement. Security is not an automated process usually it needs a workforce to continuously monitor risks and threats which leads to an inflated cost. This paved the way for implementing the concepts like SecDevOps and DevSecOps. In DevSecOps, three departments (Development, Security, and Operation) work together to achieve the same goal from the Starch.

Implementation of the Infrastructure as Code (IAC) in an organization be done with the help of applications such as AWS Cloud Formation, Amazon Image Machines, and other well-known open-source applications are Ansible, puppet, Terraform, and Chef. The implementation of IAC in an organization that already has an existing infrastructure makes the implementation process of IAC long. The Organization can use an application like Terraform to check whether there are any holes to integrate the Infrastructure As code or they can manually create a replica of the existing stack. Assuming that the deployment phase is done with the help of terraform platform.

C. *What are the major cloud security threats?*

Cloud makes data storing and data management easy leading to better productivity at low cost. When the Data in the cloud increases, the data security treats are simultaneously increasing to a certain level. Cloud Data management has challenges including Data Privacy, Insider and outsider threat, data integrity, data Location, data confidentiality, trust, etc [5]. The data-related problems in the cloud can be solved to a extend by planned Security Development Lifecycle (SDL) [6], for example, creating personas for each entity that are using the cloud and creating threat models accordingly to foresee the upcoming threat and secure accordingly.

The major physical security threats are network security issues and requirements, reliability, audit, environmental issues, physical access, maintainability, and regional threat [5]. Physical security threat is further divided into sub-threats which are mobile platforms, circumference, denial of service, port scanning, dependency, botnets, spoofing attacks. And environmental threat can be occurred due to disaster or heat issues. Mostly physical threats are fixed with Hardware specialists in the organization, and it is a time-consuming process. For Example, if a server in the organization fails due not the high heat emission the Hardware team will at least a day to retrieve the server and use it as normal. If a cloud is created using the infrastructure of code this issue can be solved i.e., In Infrastructure As code physical servers are used instead, servers are created virtually using high-end programs. To change the data arrangement or need a new installation, running a set of code in an iteration will change the data arrangement and complete the installation.

II. MUSA Security Assurance Platform

MUSA (Multi-cloud Secure Applications) Security Assurance platform is the result of the MUSA project which started in January 2015.MUSA Projects is creating open-source platforms that ensure Security in a multi-cloud environment using Security by Design. This project is EU (European Union) funded and they have successfully created a MUSA framework that can overcome the security issues in cloud environment using an SLA (Service Level Agreement).

According to the MUSA, the entire process of creation has Five main steps Modelling, Risk Assessment, CS Selection Decision Support, SLA Generation, Continuous Assurance.

Modelling: The DevOps team study about the Application they want to create and create a CPIM (Cloud Provider Independent Model).

Risk Assessment: Risk assessments are done by Analysing the security requirements and doing a complete risk assessment process

CS Selection decision support: After the Risk assessment process the DevOps teams can finales their Cloud Providers according to the requirements

SLA (Service Level Agreement) Generation: SLA Generation is done by carefully studying the Service Level Objectives (SLO) and Security Requirement metrics [9].

Security Libraries: If any of the expected security features are not available by any of the cloud service providers the organization can add their security library for protecting their data [7].

Software development lifecycle in MUSA project is created considering the DevOps environments by Security by Design.

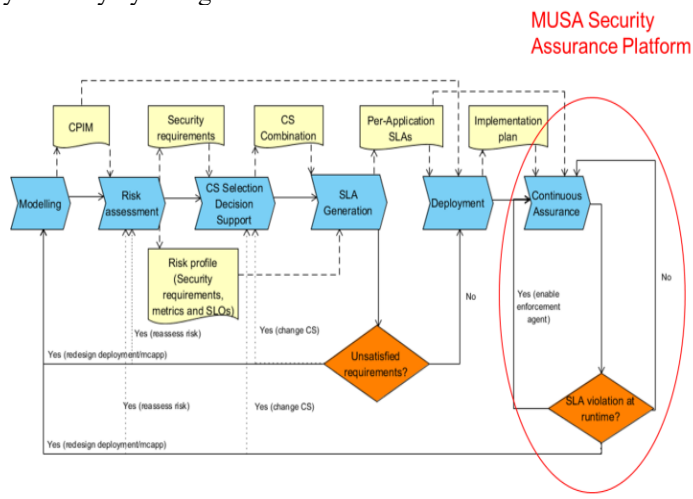

Fig 1: MUSA Security Assurance Platform [7].

III. IAC and MUSA Platform

Security in the IAC should start from the modeling phase itself since IAC is a DevOps Principle the Development and the Operation teams are meant to work simultaneously throughout the project. IAC developers start to create code in the deployment state after the risk assessment and the selection of the Cloud Service provider. The Application development process introduced by the MUSA project can be used to create an application in the IAC as well since both are using the DevOps principle.

A. *Design*

As per the EU (European Union) data protection directive, it is recommended that even third-party data management should have a proper assessment for data protection for their clients [15]. Traditionally risk assessments are done by site

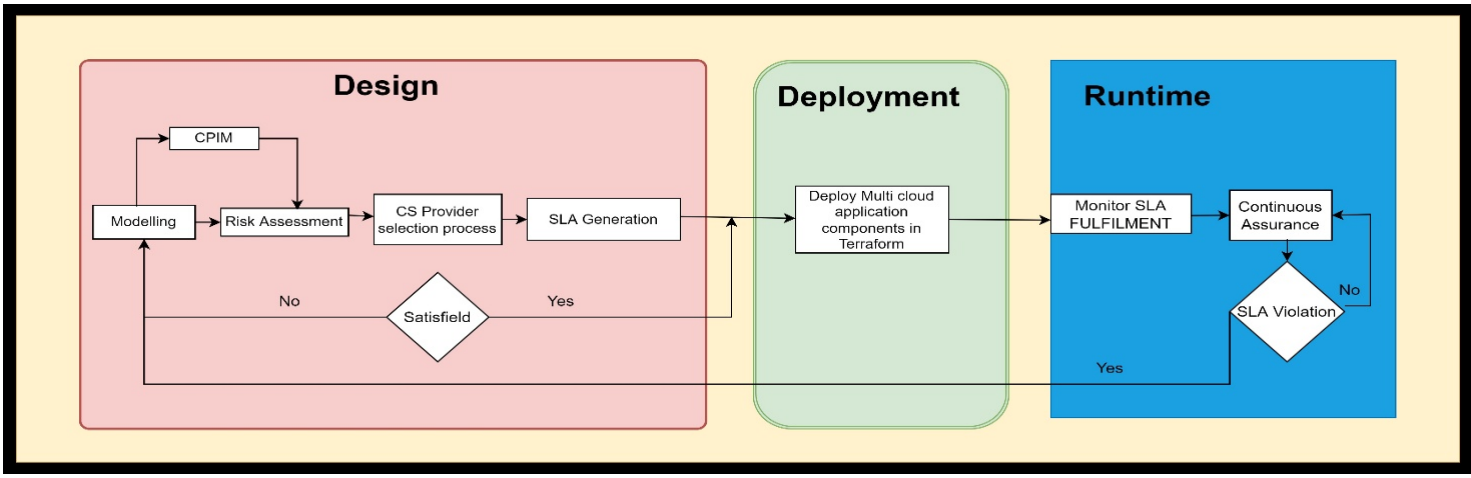

Fig 2: Implementation of MUSA Framework and Terraform

reliability engineers in the cloud-native deployments but the increase in the data and information in the cloud make this an impossible process now. So now most of the risk assessments are done automatically using the tools like CSA's CCM Framework, NIST Risk Management #Framework, ISO 27017/18, and ISO 27701 Frameworks, SOC reports covering cloud CCM controls [16].

There are five steps to do a risk assessment process [17].

1. Identify the threads and risks that can be faced by the organization
2. Identifying the relevant security measures that need to tackle the threads.
3. List out the Cloud providers and evaluate the services they give and your security needs.
4. Analyse the possible risks and their prevention
5. Accordingly select the cloud provider

Creating a well-defined Service level agreement is an important part to ensure cloud security from the provider's end. As discussed earlier the major threats in cloud computing are data breach, Misconfiguration, and inadequate change control, Lack of cloud security architecture, Insider threat, these all treats directly influence the cloud SLA. Since cloud environment always has the possibility for the treat the Infrastructure as Service sometimes is not achieved. Therefore, creating, and time management, and monitoring the SLA is a vital task. Cloud Provider SLA should contain the following parameters [11],

- Service Assurance
- Service Assurance Period
- Service Assurance Granularity
- Service Guarantee
- Service Recognition
- The service violation measurement and reporting.

B. Deploy

Deployment of the IAC would be a bit time-consuming in the first deployment or if attempting to change the existing platform to IAC. Terraform is an open-Source platform used to deploy and create IAC [18]. Many other sources do the same process as Amazon and Google, but the main difference is that it deploys infrastructure and applications and are easy to use [19]. Many other platforms limit the developer by the programming language where terraform is very flexible and extendable. As far as the cloud it can be used in any cloud network. In the Terraform platform, the developer is supposed to write the configuration files and configures to the terraform, and then it can be deployed in the cloud.

Initiation of the terraform in a five-step process:

1. Code the configuration files
2. AWS provider should be configured [10].

```
"aws" { "us-west-2" }
 resource "aws_instance" "helloworld"
{ ami = "ami-09dd2e08d601bff67" instance_type = "t2.micro" tags = { Name = "HelloWorld" }
```

3. Initialize Terraform
4. Deploy the EC2(Amazon Elastic Compute Cloud (Amazon EC2))
5. Cleanup the Terraform

C. Runtime

Monitoring the SLA simultaneously can ensure that there is no security breach from the cloud provider. The Agreed service level agreements can be monitored by Visual countdowns and timers, Breach notifications and warnings, Segmentation and customization, Multiple SLA metrics, SLA reporting, and analytics [13]. Detection of the SLA violation can be done by implementing some basic algorithms [12]. If any violation in SLA takes place the client can take legal action against the cloud providers and claim their compensation.

IV. Related work

Security has many Real-time entities and ensuring security is a challenging task but possible. Securing software or say organizations from threat should be in mind from the very first steps of development. Security by Design is a very practical approach to ensure security but at the same time, it needs labor, time, and high implementation cost. Nowadays Organizations only aim for efficiency in an application ignoring security. Infrastructure as Code has taken the developmental process to the next level without having any physical entities but the implementation and deployment in an organization that has an existing cloud are extremely complicated, more tools for implementation and changing existing cloud applications to IAC can be introduced to solve this problem. As already mentioned, the increasing number of cloud providers had led to the increase in cloud security threats [1] and there are many papers introduced on cloud security assessment, but the number of papers concentrated on Security by design is exceptionally low. The global market of the cloud especially IAC is ruled by huge companies and the design and implementation of their platforms are not fully revealed, the area like IAC, still have lacked proper resources that can be used.

V. CONCLUSION

Analyzing the MUSA project, creating and adopting a Security SLA is a very vital part to ensure security in cloud applications. Security by Design guarantees that the security SLA is implemented efficiently from the initial stages of the developmental process. DevOps practice is a continuous process so creating and implementing Security SLA by security by design works well the treats can be foreseen in each developmental loop.

Infrastructure is a very modern approach to cloud application creation so there should be increased studies to be done in the area which will significantly help an organization to reduce and the cost of maintaining and processing their data.

In concluding analyzing the MUSA projects, the implementation of the Security by Design model in the Cloud Application which uses DevOps model such as Infrastructure As code can be effective in building a good Security SLA, using platforms like terraform which are opensource.

VI. References

[1] D.Jagli and S. Yeddu, "CloudSDLC: Cloud Software Development Life Cycle", International Journal of Computer Applications, vol. 168, no. 8, pp. 6-10, 2017.

[2]S. Almuairfi and M. Alenezi, "Security controls in infrastructure as code", Computer Fraud & Security, vol. 2020, no. 10, pp. 13-19, 2020. Available: 10.1016/s1361-3723(20)30109-3.

[3]E. BOERSMA, "7 Application Security Principles You Need to Know | Cprime Blogs", Cprime, 2021. [Online]. Available: https://www.cprime.com/resources/blog/security-by-design-7-principles-you-need-to-know/. [Accessed: 10- Nov- 2021].

[4]B. DevOps, "Benefits of DevOps | Top 16 Advantages of DevOps You Need to Know", EDUCBA, 2022. [Online]. Available: https://www.educba.com/benefits-of-devops/. [Accessed: 12- Jan- 2022]

[5]R. Al Nafea and M. Amin Almaiah, "Cyber Security Threats in Cloud: Literature Review", Ieeexplore.ieee.org, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9491638. [Accessed: 13- Jan- 2022].

[6]S. Lipner, "Security Development Lifecycle", springer.com, 2010. [Online]. Available: https://link.springer.com/content/pdf/10.1007/s11623-010-0021-7.pdf. [Accessed: 13- Jan- 2022].

[7]E. Iturbe, "Final MUSA framework implementation", Musa-project.eu, 2021. [Online]. Available: https://www.musa-project.eu/sites/musa3.drupal.pulsartecnalia.com/files/documents/MUSA%20D1.5%20Final%20MUSA%20framework%20implementation.pdf. [Accessed: 10- Nov- 2021].

[8]"8 Security by Design Principles for Your Business Solutions - TechnologyHQ", TechnologyHQ - All about Technology, AI, blockchain, Cybersecurity, Business, 2022. [Online]. Available: https://www.technologyhq.org/8-security-design-principles-business-solutions/. [Accessed: 13- Jan- 2022].

[9]V. Casola, A. De Benedictis, M. Rak and U. Villano, "A novel Security-by-Design methodology: Modeling and assessing security by SLAs with a quantitative approach", Journal of Systems and Software, vol. 163, p. 110537, 2020. Available: 10.1016/j.jss.2020.110537.

[10]S. Jourdan and P. Pomès, Infrastructure As Code (IAC) Cookbook. Birmingham: Packt Publishing Ltd, 2017, p. 48.

[11]V. Casola, A. De Benedictis, M. Rak and E. Rios, "Security-by-design in Clouds: A Security-SLA Driven Methodology to Build Secure Cloud Applications", 2022.

[12]S. Anithakumari and K. Chandrasekaran, "Monitoring and Management of Service Level Agreements in Cloud Computing", Ieeexplore.ieee.org, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/7312156/. [Accessed: 13- Jan- 2022].

[13]"Service Level Agreement Monitoring - 5 Essential Tools", Commusoft, 2022. [Online]. Available: https://www.commusoft.co.uk/five-tools-service-level-agreement-monitoring/. [Accessed: 13- Jan- 2022].

[14] WINKLER, S. Terraform in Action video edition. [electronic resource]. 1st edition. [s. l.]: Manning Publications, 2021. Disponível em: https://search.ebscohost.com/login.aspx?direct=true&db=cat00012a&AN=bourne.1250148&site=eds-live&scope=site. Acesso em: 13 jan. 2022.

[15]"ARTICLE29 - Item Overview", Ec.europa.eu, 2022. [Online]. Available: https://ec.europa.eu/newsroom/article29/items. [Accessed: 13- Jan- 2022].

[16]E. Cayirci, A. Garaga, A. Santana de Oliveira, and Y. Roudier, "A risk assessment model for selecting cloud service providers", 2022.

[17]A. SANTANA DE OLIVEIRA, "Five Steps to Perform a Cloud Risk Assessment | SAP Blogs", Blogs.sap.com, 2022. [Online]. Available: https://blogs.sap.com/2016/04/13/five-steps-to-perform-a-cloud-risk-assessment/. [Accessed: 13- Jan- 2022].

[18]N. Singh Gill, "Top 10 Infrastructure as Code Tools to Boost Your Productivity", Nexastack.com, 2022. [Online]. Available: https://www.nexastack.com/blog/best-iac-tools. [Accessed: 13- Jan- 2022].
[19]V. A. B. Thangaraju, “The Benefits of Using Terraform as a Tool for Infrastructure-as-Code (IaC),” www.opensourceforu.com, 22-Jul-2020. [Online]. Available: https://www.opensourceforu.com/2020/07/the-benefits-of-using-terraform-as-a-tool-for-infrastructure-as-code-iac/.