Site Reliability Engineering and SOAR

Angela Milova

Palo Alto - Secure the Future

SRE

1. Explain in your own terms why SRE would focus on the connections between the components within the system as much as focusing on the components themselves.

2. Consider Quality Assurance. Where would you expect to include Regulatory Standards Compliance when Diagramming future applications or solutions subsystems?

1. The core emphasis of Site Reliability Engineering (SRE) lies in the connections between system components as much as concentrating on individual components. This focus is driven by the recognition that no system operates as a mere collection of isolated components. Every component necessitates communication with others, and systems interact with one another. This means that for SRE ensuring the optimal functionality of each component within a system and guaranteeing that the overall system attains the desired performance levels, exhibiting reliability and resilience against potential threats is very important. SRE places significant importance on monitoring and analyzing both systems and components. This emphasis stems from the understanding that a compromised component can potentially jeopardize the entire system, undermining its reliability. The proactive approach of SRE involves continuous vigilance to promptly detect and address any vulnerabilities. By meticulously attending to the connections and dependencies within a system, SRE strives to fortify its reliability, mitigate risks, and sustain optimal performance (Beyer, Jones, Petoff, et. al, 2016)

2. Regulatory Standards Compliance is crucial when diagramming future applications or solution subsystems, ensuring adherence to compliance requirements throughout the development lifecycle. Some regulatory aspects that require consideration include:

a) Design: During the design phase and algorithm development, Quality Assurance must factor in all stakeholders and adhere to project standards. Additionally, depending on the sector, specific compliance requirements should be addressed to safeguard data privacy (Sharma, 2022).

b) Development Environments: Quality Assurance involves a well-specified testing phase in the development environment before the code goes live in production. It is crucial that QA addresses coding functionality and assesses security issues arising from coding processes lacking security measures (Sharma, 2022).

c) Testing: Quality Assurance must adhere to compliance rules during the testing phase, prioritizing the assurance of secure coding standards. This entails ensuring that the code is not only functional and reliable but also compliant with any regulations that the organization needs to follow (Sharma, 2022).

d) Documentation and Auditing: Quality Assurance should maintain detailed records of the testing process, results, modifications, and any necessary suggestions for the code to pass all tests (Sharma, 2022).

By integrating Regulatory Standards Compliance into each phase of the application development process, QA teams ensure that the final product aligns with both legal and industry-specific standards. This comprehensive approach not only guarantees compliance but also contributes to the creation of a robust and compliant product.

SOAR

1. Automation. Explain two governing criteria when deciding upon candidates for security automation.

2. Orchestration. Evaluate how process update frequency impacts playbook design. What future playbook design elements would you implement in order to reduce the impact of frequent updates?

1. Security Automation plays a crucial role in enhancing the overall resilience of an organization's security posture by automating routine tasks, allowing security teams to focus on strategic and complex challenges while significantly reducing response times to emerging security incidents. Two criteria that come with it are:

a) Repetitive Tasks: Identify security tasks that need to be performed repeatedly and automate them. Define the rules that should guide the automation process. These must be the tasks that follow a predictable pattern for successful automation. Additionally, part of the automation process can involve the identification of recurring false positive alerts, enabling the automated whitelisting of these incidents. This approach prevents overwhelming the security team with repetitive tasks (Mohammad,Surya, 2018).

b) Impact and Response: Evaluate tasks based on their impact and the required response time. Automation proves highly beneficial for tasks that, while traditionally performed manually, demand fast response times. Automating these tasks can be crucial for reducing response times and addressing issues in real-time. For instance, consider automating a firewall to promptly drop specific malicious traffic, ensuring a faster response compared to waiting for manual intervention from human operators (Nyre, 2019).

2. Frequent process updates can pose a problem for playbooks and their effectiveness. If a playbook relies on a tool that is constantly being automated, it can cause the playbook to be less effective and become inaccurate. Therefore, it is important to address the frequency of automation to align with the required frequency of playbook changes. Some future designs that can be implemented for playbooks are:

a) Dynamic Adaptation: Instead of hardcoding everything on the playbook, making the playbook's process more dynamic is crucial. This approach allows certain parts of the playbook to adapt to changes without requiring human intervention to modify the code or information (Sharma, 2017).

b) Modular Components: For playbook components that frequently change, it is advisable to make them modular. This ensures that these components, requiring frequent updates, can adapt without affecting the entire playbook (Sharma, 2017).

c) Testing Environment for Automation: Create a dedicated testing environment to prevent playbooks from breaking after automation. Automated testing is essential to ensure playbooks remain functional and updated following automation changes (Sharma, 2017).

d) Collaboration Between Automation and Playbook Teams: Ensure that the teams responsible for automation and updates and the teams responsible for playbooks collaborate effectively. This collaboration ensures that both teams share information seamlessly, avoiding misunderstandings or potential conflicts between teams. Such cooperation is vital for the overall success and synergy between the automation and playbook processes (Stewart, 2019).

Resources

Beyer, B., Jones, C., Petoff, J., & Murphy, N. R. (2016). *Site reliability engineering: How Google runs production systems*. " O'Reilly Media, Inc.".

Mohammad, S. M., & Surya, L. (2018). Security automation in Information technology. *International journal of creative research thoughts (IJCRT)–Volume*, *6*.

Nyre-Yu, M. M. (2019). *Determining system requirements for human-machine integration in cyber security incident response* (Doctoral dissertation, Purdue University).

Sharma, S. (2017). *The DevOps adoption playbook: a guide to adopting DevOps in a multi-speed IT enterprise*. John Wiley & Sons.

Sharma, S. (2022, September 10). *What is the standard software QA process and stages? - awsquality*. AwsQuality Technologies | Salesforce ISVPartner | AppExchange Partner.

Stewart, M. K. (2019, August 7). *Enhancing collaboration with a team Playbook*.