Automated Remediation Tool

ART

COMP9447 - 21T1

Week 5 Milestone

Submitted by Team 2 :

Apoorva Bhagchandani	z5276815
Apurva Shukla	z5205212
Augustine Hyunwoo Lee	z5061885
Arya Trivedi	z5269830
Sidhant Arora	z5281441
Steven Jones	z5117091

Submission Date: 19/03/21

Table of Contents

1 Introduction

1.1 What is Cloud Security?

Cloud Security is the set of policies, applications, technologies, and controls that are utilized to protect virtual information such as data, securities, infrastructure, virtual IP, and applications. Cloud Security can operate within public, private, and hybrid cloud environments in order to implement strategies to help prevent cyber threats that may compromise a system. This is a vital part of implementing cloud policies into a business. As cyber-attacks become more advanced, cloud security may be used as a prevention tool due to its variability and adaptability. It can also help restrict network usage and facilitate growth amongst other business applications.

1.2 Challenges in Cloud Security

With the exponential rise of cloud infrastructure and our resulting dependence, security has become a crucial concern for systems that are dependent on cloud services; be it for machine learning, simple web app hosting or big data analytics for the entire country. Some of the major challenges that cloud computing faces today are -

- *Inadequate Access, Credential, Identity, and Key Management*

With cloud computing comes several changes to typical internal system management practices associated with identity and access management (IAM). Although these are not new cloud security challenges, they are still however important challenges when working with cloud-based environments.

- *Interfaces and APIs with Poor Security*

In order to enable consumers to manage and utilize cloud systems, cloud computing providers release a set of software user interfaces (UIs) and APIs. These APIs are the ones that determine how secure and available the overall cloud servers services will be.

- Denial of Service (DoS) Attacks

The primary objective of DoS attacks is to disable a system, network, or machine so that it becomes inaccessible to its intended users. The development and growth of cryptocurrencies like Ripple and Bitcoin provide high value targets, increasing the number of DoS attacks and speeding up malicious tool development.

- *Data Breaches*

Cloud-based environments make it easy to share the data stored within them. These environments are accessible directly from the public Internet and include the ability to share data easily with other parties via direct email invitations or by sharing a public link to the data. These factors all increase the severity and likelihood of such a breach.

1.3 What is SOAR?

"SOAR solutions are tools that provide three key features. First, case management and workflow capabilities: Just like IT support and helpdesk teams use IT service management tools to track and control their work, Security Operations Centers (SOCs) also need tools to manage and control the work of triaging alerts, [as well as] raising, investigating, and solving incidents. Second, automate tasks from those activities via orchestration of multiple tools, such as Endpoint Detection and Response (EDR), Security Information and Event Management (SIEM) and Network Detection and Response (NDR). And third, provide a centralized way to access, query, and share threat intelligence, a vital resource for threat detection and response activities." –Augusto Barros, VP of solutions at Securonix

1.4 AWS Well-Architected Framework

For our SOAR product, we use the guidelines as mentioned by the five pillars of a Well-Architected Framework. If the foundation is not solid, structural problems can undermine the integrity and function of the building. When designing technology solutions, if the five pillars are neglected, it can become challenging to build a system that delivers on our expectations and requirements. Incorporating these pillars into our architecture will help us produce stable and efficient systems. These pillars are -

● *Operational Excellence Pillar*- The Operational Excellence pillar includes the ability to support development and run workloads effectively, gain insight into their operations, and to continuously improve supporting processes and procedures to deliver business value.

● *Security Pillar* - Before a workload is developed, best practices will need to be implemented that influence security and access control. In addition, it is pertinent to be able to identify security incidents, protect systems and services, and maintain the confidentiality and integrity of data through data protection.

● *Reliability Pillar* - To achieve reliability one must start with the foundations — the development of an environment where service quotas and network topology accommodate the workload. The workload architecture of the distributed system must be designed to prevent and mitigate failures. The workload must handle changes in demand or requirements, and it must be designed to detect failure and automatically heal itself.

● *Performance Efficiency Pillar* - Take a data-driven approach to building a high-performance architecture. Gather data on all aspects of the architecture, from the high-level design to the selection and configuration of resource types.

● *Cost Optimization Pillar* - As with the other pillars within the Well-Architected Framework, there are tradeoffs to consider, for example, whether to optimize for speed-to-market or for cost. In some cases, it’s best to optimize for speed—going to market quickly, shipping new features, or simply meeting a deadline—rather than investing in up-front cost optimization.

2 Threat Model

At first, the team researched about the various security threat modelling frameworks, mainly focusing on the STRIDE threat model. Furthermore, we investigated and aimed to identify various AWS Services which could be used in our SOAR to help tackle the previously recognized threat types.

Upon research, we decided to build a threat modelling solution which would be able to utilise Amazon GuardDuty and its finding types to tackle and streamline security operations in three key areas, namely-threat and vulnerability management, incident response, and security operations automation.

Our solution “ART” is supposed to be an overall jack of all trades type of solution that can help even the novice users to take control and be better informed about security over their AWS Account and all the resources within it.

It is a Web Application that would run on an EC2 instance in their account that would mainly focus on :

1. Sending a notification to the user based on the detected security threat.
2. Providing pre-defined playbooks and remediation steps for each GuardDuty finding type.
3. Creating Custom Flows for the various findings as defined by GuardDuty.

2.1 Use Case

A user will be able to access our “ART” user interface and be able to select the respective threat model they want to prevent for their instance and will be able to initially set their own playbook to do so. For example, a user wants to prevent the occurrence of Cryptocurrency Mining on their EC2 instance, identified as a finding type

“CryptoCurrency:EC2/BitcoinTool.B” by Amazon GuardDuty. To do this, the user will first select the resource/instance type they want to protect; in this example EC2. Further, they would select the cryptocurrency mining finding type and continue to select the action they want to perform in case of such an occurrence; for example, suspending the instance until

SecOps take further action on this finding. On creating such a playbook, the status of this instance is saved in the database by our system. By doing so, if and when Amazon GuardDuty identifies and alerts a Cryptocurrency suspicious activity on the instance, it will trigger the playbook set by the user and successfully suspend the instance until further action is taken on it.

3 High level solution overview

Our solution involves many moving parts and relies on AWS features such as GuardDuty, EventBridge and the Simple Notification Service (SNS). Our ideal flow looks like Figure A, where a user would be able to register a local user account with our platform then be able to set up flows for each type of GuardDuty finding. The GuardDuty alerts would then be hooked up to EventBridge which has a specific rule for all incoming findings. This rule would trigger an SNS Topic which then notifies our ART backend subscriber via a HTTP post request. This would finally go through some business logic to check if there exists a flow for that specific GuardDuty finding, and if it does run through the functions the user had selected. This would ensure an end to end hands off approach for automatic remediations.

A use case diagram can be seen in Figure B.

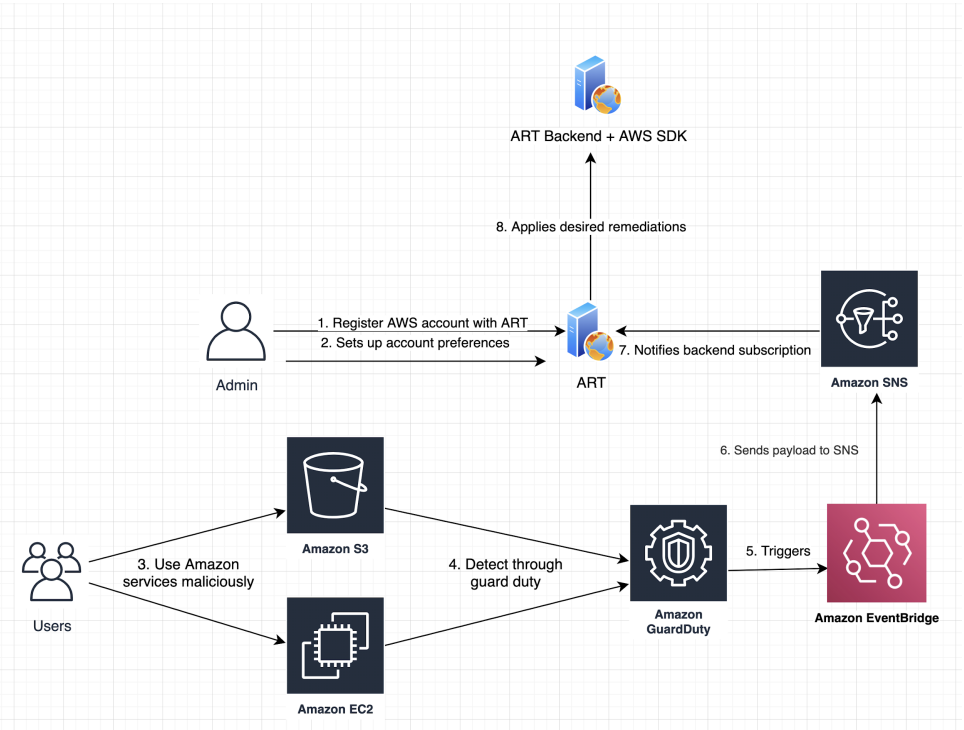

Figure A

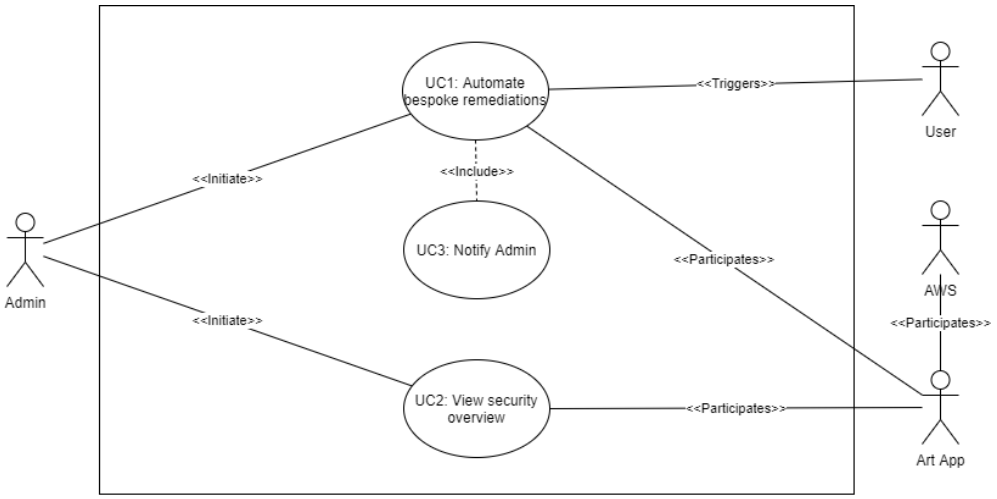

Figure B

4 Design Choices

4.1 Frontend:

For the frontend development of the application the team decided upon using ReactJS, the most popular framework for frontend development. It has several benchmark features such as VirtualDOM which improves performance and reusable components which can help the development team to standardize the code throughout the application .

4.2 Backend:

For the backend development of the application, the team decided upon using SailsJS. Built on ExpressJS, it has a convention over configuration approach alongside powerful code generation through Blueprints.

4.3 AWS Services

4.3.1 Guard Duty

The team chose AWS Guarduty as the primary method for threat detection. AWS GuardDuty is a threat detection service that identifies activity which can be associated with account

compromise, instance compromise, malicious reconnaissance, and bucket compromise. Additionally, GuardDuty detects unusual API calls, suspicious outbound communications to known malicious IP addresses, or possible data theft using DNS queries as the transport mechanism. More accurate findings are delivered through the application of machine learning enriched by threat intelligence, such as lists of malicious IPs and domains.

4.3.2 SimpleNotifcationService

The team chose SNS as a notification service. Amazon Simple Notification Service (Amazon SNS) is a fully managed service for application to application (A2A) and application to person (A2P) communication.

4.3.3 EventBridge

Eventbridge was chosen as it is a serverless event bus that makes it easier to build event-driven applications at scale using certain events generated or triggered by our application.

4.3.4 EC2

Amazon Elastic Cloud Compute was chosen to provide secure, resizable compute capacity in the cloud. It provides our team with complete control of computing resources and helps plan and organize them effectively.

5 Future Scope

The project in our eyes has a lot of potential to become the one stop solution for all security over AWS. At the moment we are only focusing on findings from GuardDuty and giving remediation options appropriate for those, but it can easily be integrated with other AWS Services such as Macie, Cognito and Inspector.

Also creating custom remediation steps may be possible in contrast to the predefined/ suggested remediations for the various findings that the time constraint of the project might allow.

We are also hoping to integrate a level of interactivity with the Slack API to have some degree of control for SecOp first responders.

Our ideal goal is to have a CloudFormation template that organisations can use to instantly create the desired resources and connections between resources (such as the SNS automatically hooking up with the IP address of the EC2 resource).